

**Uniwersytet Warszawski**  
**Wydział Matematyki, Informatyki i Mechaniki**

Łukasz Rogowski

# Implementacja systemu wnioskującego za pomocą DNA

*rozprawa doktorska*  
*napisana na Wydziale Matematyki i Informatyki*  
*Uniwersytetu Łódzkiego*

Promotor rozprawy:  
prof. dr hab. Tadeusz Kasiński  
Uniwersytet Łódzki  
Wydział Matematyki i Informatyki

Łódź, marzec 2021

Oświadczenie autora rozprawy:  
niniejsza rozprawa została napisana przeze mnie samodzielnie.

4 marca 2021

.....  
*Lukasz Rogowski*

Oświadczenie promotora rozprawy:  
rozprawa jest gotowa do oceny przez recenzentów.

4 marca 2021

.....  
*prof. dr hab. Tadeusz Krasieński*

## Streszczenie

Obliczenia DNA (z ang. DNA computing) są nowatorską dziedziną nauki, leżącą na pograniczu informatyki i biologii molekularnej. Przedmiotem jej badań są komputery biomolekularne, które w przyszłości mogą stanowić alternatywę dla tradycyjnych systemów komputerowych. Mają one ogromny potencjał ze względu na masowość i równoległość obliczeń. Narzędziem, wykorzystywanym do rozwiązywania problemów algorytmicznych, matematycznych czy logicznych, są tu łańcuchy DNA, a same obliczenia przeprowadzane są w odpowiednich warunkach laboratoryjnych za pomocą operacji biochemicznych. Siła tych obliczeń nie tkwi w ich prędkości a w masowej równoległości operacji, dzięki czemu problemy o naprawdę dużej złożoności mogą być rozwiązywane przy sensownych ograniczeniach czasowych. W trakcie rozwoju (od 1994 roku) w obliczeniach DNA wyodrębniło się wiele gałęzi. Między innymi: 1. rozwiązania za pomocą DNA wielu problemów informatycznych np. drogi Hamiltona, SAT problemu itp.; 2. implementacje za pomocą DNA teoretycznych modeli komputerów: automatów skończonych, automatów ze stosem, maszyn Turinga, sieci logicznych itp.; 3. implementacje za pomocą DNA systemów wnioskowań znanych z logiki. Przedstawiona rozprawa doktorska dotyczy trzeciego tematu.

Najprostszym systemem formalnym logiki matematycznej jest klasyczny rachunek zdań. Bada on zbiór zdań elementarnych (zwanym atomowymi), które mogą przyjmować tylko dwie wartości: *prawdy* lub *falszu*, oraz powstający z nich (za pomocą spójników logicznych) zbiór zdań złożonych. W otaczającym nas świecie często spotykamy się z następującym problemem: dany jest układ skończony zdań złożonych  $A$  (mogą to być dowolne zdania, niekoniecznie tautologie). Traktujemy je jako specyficzne aksjomaty czyli zdania, które uznajemy w danym układzie za prawdziwe. Wówczas dla dowolnego zdania  $p$  pytamy czy wynika ono z  $A$  za pomocą reguł wnioskowania, czyli czy  $p$  jest konsekwencją zdań z  $A$ . Realizacja rozwiązania tego problemu za pomocą łańcuchów DNA i operacji na nich jest głównym celem tej pracy.

W rozdziale 1 omówione zostały wiadomości wstępne dotyczące logiki matematycznej i klasycznego rachunku zdań w odniesieniu do prezentowanej koncepcji systemu wnioskującego. Rozdział 2 stanowi wprowadzenie do obliczeń DNA, prezentuje podstawowe modele konstrukcji biokomputerów i przedstawia niektóre prace z rozwoju tej dziedziny. W rozdziale 3 omówione zostały istniejące wcześniej koncepcje związane z implementacją logiki i wnioskowania za pomocą DNA. Rozdział 4 bardzo szczegółowo omawia wszystkie elementy nowej koncepcji. Rozdziały 5 i 6 stanowią uzupełnienie rozdziału 4. Rozdział 5 zawiera analizę poprawności oraz istniejących ograniczeń, a także omawia dodatkowe zastosowania systemu (m. in. koncepcja częściowego rozwiązania problemu SAT oraz obliczanie wartości funkcji logicznych) i możliwości dalszego rozwoju. Rozdział 6 szczegółowo opisuje wszystkie kroki eksperymentu laboratoryjnego.

Niektóre wyniki rozprawy były prezentowane w publikacjach i na konferencjach.

**Słowa kluczowe:** DNA, obliczenia DNA, wnioskowanie logiczne, logika matematyczna.

**Klasyfikacja tematyczna AMS:** 03B05, 03B70, 03B80, 92-08, 92D20.

## Abstract

### Implementation of the logic inference system by DNA

DNA computing is an innovative interdisciplinary field which combines elements of molecular biology with computer sciences and mathematics. It investigates biomolecular computers which can be one of the future alternatives to the traditional computer systems. It holds a great potential for massively parallel computation, which is one of the inherent characteristics of operating on DNA molecules. These DNA chains are used to solve algorithmic, mathematical or logical problems, through biochemical operations in laboratory environments. The compactness and reactivity of DNA strands allow for massive amounts of simultaneous parallel processing, which potentially allows us to find solutions to problems with huge computational complexity in sensible time constraints. Since the beginning of the field (starting in 1994) there are many different directions in DNA computing. Among others: 1. solving with DNA many well-known algorithmic problems i.e. Hamiltonian path problem, SAT problem etc.; 2. DNA implementations of theoretical computer models: finite automaton, push-down automaton, Turing machine, logical networks etc.; 3. DNA implementations of inference systems known from mathematical logic. Presented thesis belongs to the third theme.

The simplest formal logic system in mathematics is the classical propositional calculus. It investigates a set of logical statements, each of which can assume only one of two possible values: *true* or *false*. Propositional calculus also explores the set of composite expressions formed from these basic statements by the use of logic connectives. A common real world problem is whether we can deduce a certain conclusion  $p$  from a given set of statements  $A$ . Our set  $A$  consists of a finite amount of assertions treated as axioms which can include tautologies, but not necessarily so. We can then pose a question if some arbitrary statement  $p$  can be derived from  $A$  by rules of logical inference. The main aim of this thesis is the realisation of this type of system with DNA molecules.

Chapter 1 is an introduction to mathematical logic and propositional calculus which are used by the proposed biochemical logic inference system. Chapter 2 contains preliminary facts about DNA computing. It includes descriptions of basic models of bio-computers and brief explanations of the most important concepts from this field. Chapter 3 describes previous attempts at implementation of logic and reasoning systems by DNA molecules. Chapter 4 meticulously presents all the elements of the new biochemical inference system concept. Chapters 5 and 6 complement the previous one. Chapter 5 analyses the correctness and limitations of the proposed system and presents additional applications of it (partial solving of the SAT problem and logical functions implementation) and future improvement possibilities. Chapter 6 precisely describes all the steps of laboratory experiment.

Some of the results were previously published and shown at the conferences.

**Key words:** DNA, DNA computing, logical inference, mathematical logic.

**AMS subject classification:** 03B05, 03B70, 03B80, 92-08, 92D20.

# Spis treści

<b>Streszczenie</b> .....	<b>- 3 -</b>
<b>Abstract</b> .....	<b>- 4 -</b>
<b>Spis treści</b> .....	<b>- 5 -</b>
<b>Wstęp</b> .....	<b>- 8 -</b>
<b>Cel i teza pracy</b> .....	<b>- 12 -</b>
Szczegółowe cele pracy: .....	<b>- 12 -</b>
<b>1. Wiadomości wstępne z logiki matematycznej</b> .....	<b>- 13 -</b>
1. 1. Klasyczny rachunek zdań .....	<b>- 13 -</b>
1. 2. Reguły wnioskowania .....	<b>- 14 -</b>
1. 3. Postacie normalne zdań złożonych .....	<b>- 15 -</b>
<b>2. Wprowadzenie do obliczeń DNA</b> .....	<b>- 17 -</b>
2. 1. Budowa i podstawowe operacje na DNA .....	<b>- 17 -</b>
2. 1. 1. Ligacja .....	<b>- 18 -</b>
2. 1. 2. Restryktacja .....	<b>- 18 -</b>
2. 1. 3. Replikacja .....	<b>- 20 -</b>
2. 1. 4. Denaturacja .....	<b>- 21 -</b>
2. 1. 5. Synteza.....	<b>- 21 -</b>
2. 1. 6. Elektroforeza.....	<b>- 22 -</b>
2. 2. Modele biokomputerów .....	<b>- 23 -</b>
2. 3. Model filtrowania .....	<b>- 23 -</b>
2. 4. Model splatania .....	<b>- 26 -</b>
2. 5. Pozostałe modele obliczeń .....	<b>- 32 -</b>
<b>3. Znane implementacje systemów wnioskowania.</b> .....	<b>- 33 -</b>
3. 1. System Olgierda Unolda i Macieja Trocia .....	<b>- 33 -</b>
3. 1. 1. Model systemu wnioskującego.....	<b>- 33 -</b>
3. 1. 2. Funkcje logiczne w postaci kanonicznej.....	<b>- 36 -</b>
3. 2. System Shai Kaplana, Toma Rana i Ehuda Shapiro .....	<b>- 40 -</b>
3. 2. 1. System wnioskujący z identyfikacją przestanków.....	<b>- 40 -</b>
3. 2. 2. Klasyczny system wnioskujący .....	<b>- 42 -</b>
3. 3. Systemy grupy badawczej Jana Mulawki.....	<b>- 47 -</b>
3. 3. 1. Dowodzenie za pomocą wnioskowania regresywnego .....	<b>- 48 -</b>
3. 3. 2. Wnioskowanie z wykorzystaniem drzewa decyzyjnego .....	<b>- 49 -</b>
3. 3. 3. System wnioskujący oparty na cząsteczkach kolistych .....	<b>- 52 -</b>
3. 3. 4. Pozostałe prace.....	<b>- 55 -</b>
3. 4. System In-Hee Lee, Ji-Yoon Park i innych. ....	<b>- 55 -</b>
3. 5. Pozostałe koncepcje.....	<b>- 57 -</b>

<b>4. Rozszerzony system wnioskowania</b> .....	<b>59 -</b>
<b>4.1. Wprowadzenie</b> .....	<b>59 -</b>
<b>4.2. Kodowanie informacji</b> .....	<b>59 -</b>
4.2.1. Literały .....	60 -
4.2.2. Klauzula faktu .....	60 -
4.2.3. Pozostałe klauzule .....	61 -
4.2.4. Molekuła terminalna i wnioskowanie .....	62 -
4.2.5. Zapytania .....	64 -
<b>4. 3. Wykrycie sprzeczności</b> .....	<b>67 -</b>
<b>4. 4. Działanie systemu wnioskującego</b> .....	<b>69 -</b>
<b>4. 5. Przykłady złożonego wnioskowania</b> .....	<b>70 -</b>
<b>4. 6. Odniesienie do klasycznego rachunku zdań</b> .....	<b>74 -</b>
4. 6. 1. Podwójne zaprzeczenie .....	74 -
4. 6. 2. Modus ponens .....	74 -
4. 6. 3. Modus tollens .....	75 -
4. 6. 4. Reguła kontrapozycji.....	75 -
4. 6. 5. Przechodność implikacji.....	76 -
4. 6. 6. Prawa de Morgana.....	77 -
<b>4. 7. Podsumowanie</b> .....	<b>78 -</b>
<b>5. Analiza rozszerzonego systemu</b> .....	<b>79 -</b>
<b>5. 1. Ograniczenia systemu</b> .....	<b>79 -</b>
5. 1. 1. Liczba zmiennych możliwych do zakodowania .....	79 -
5. 1. 2. Liczba zapytań możliwych do zakodowania.....	80 -
5. 1. 3. Zapytania o dwie wartości zdania atomowego .....	81 -
5. 1. 4. Zapytania o klauzule złożone .....	82 -
5. 1. 5. Wyprowadzenie faktu z alternatywy .....	83 -
<b>5. 2. Dodatkowe zastosowania</b> .....	<b>83 -</b>
5. 2. 1. Funkcje logiczne w postaci kanonicznej.....	83 -
5. 2. 2. Częściowe rozwiązanie SAT problemu.....	87 -
<b>5. 3. Modyfikacje systemu</b> .....	<b>90 -</b>
5. 3. 1. Zastosowanie metody dowodu nie-wprost .....	90 -
5. 3. 2. Inne enzymy restrykcyjne .....	99 -
5. 3. 3. Zastosowanie fluorescencji.....	101 -
<b>6. Symulacja eksperymentów laboratoryjnych</b> .....	<b>104 -</b>
<b>6. 1. Wybór eksperymentów</b> .....	<b>105 -</b>
6. 1. 1. Wnioskowanie i reguła kontrapozycji.....	105 -
6. 1. 2. Wykrycie sprzeczności .....	105 -
6. 1. 3. Złożone wnioskowanie i prawa de Morgana .....	106 -
<b>6. 2. Przygotowanie cząsteczek DNA w laboratorium</b> .....	<b>107 -</b>
6. 2. 1. Przygotowanie fragmentów DNA do wklonowania w wektor <i>pJET 1.2</i> .....	108 -
6. 2. 2. Klonowanie elementów systemu w wektorze <i>pJET 1.2</i> .....	118 -
6. 2. 3. Selekcja przechowywanych sekwencji z „biblioteki” łańcuchów DNA .....	121 -

6. 2. 4. Selekcja fragmentów cząsteczek biorących udział w reakcji .....	- 124 -
<b>6. 3. Reakcja wnioskowania w laboratorium .....</b>	<b>- 126 -</b>
6. 3. 1. Wnioskowanie i reguła kontrapozycji .....	- 126 -
6. 3. 2. Wykrycie sprzeczności .....	- 127 -
6. 3. 3. Złożone wnioskowanie i prawa de Morgana .....	- 128 -
6. 3. 4. Wzmocnienie „sygnału” odpowiedzi systemu .....	- 130 -
6. 3. 5. Odczytanie wyniku .....	- 131 -
<b>6. 4. Podsumowanie .....</b>	<b>- 131 -</b>
<b><i>Podsumowanie .....</i></b>	<b>- 133 -</b>
<b><i>Dodatek A: „lepkie końce” o długości 4 .....</i></b>	<b>- 135 -</b>
Unikalne „lepkie końce”: .....	- 135 -
„Lepkie końce” komplementarne ze sobą: .....	- 137 -
<b><i>Bibliografia.....</i></b>	<b>- 138 -</b>
<b><i>Spis rycin .....</i></b>	<b>- 141 -</b>
<b><i>Spis tabel.....</i></b>	<b>- 154 -</b>

## Wstęp

Najprostszym systemem formalnym logiki matematycznej jest klasyczny rachunek zdań. Bada on zbiór zdań elementarnych (zwanymi atomowymi), które mogą przyjmować tylko dwie wartości: *prawdy* lub *falszu*, oraz powstający z nich (za pomocą spójników logicznych) zbiór zdań złożonych. Podstawowe problemy klasycznego rachunku zdań to:

- Które zdania złożone są zawsze prawdziwe, bez względu na wartości zdań atomowych wchodzących w jego skład (takie zdania nazywamy prawami klasycznego rachunku zdań lub inaczej tautologiami)?
- Czy wszystkie tautologie możemy „wyprowadzić” z pewnego skończonego zbioru tautologii podstawowych (zwanymi aksjomatami) za pomocą określonych reguł wnioskowania?

Oczywiście oba problemy są od dawna rozstrzygnięte. Zdania zawsze prawdziwe możemy rozpoznać za pomocą tzw. metody zero-jedynkowej, a wszystkie tautologie możemy „wyprowadzić” z pewnego zbioru podstawowego [7]. Jednak często w otaczającym nas świecie spotykamy się z pewnym innym, choć pokrewnym powyższemu, problemem: dany jest pewien układ skończony zdań złożonych  $A$ . Mogą to być dowolne zdania, niekoniecznie tautologie. Traktujemy je jako specyficzne aksjomaty, zdania które uznajemy w danym układzie jako prawdziwe. Wówczas dla dowolnego zdania  $p$  pytamy czy wynika ono z  $A$  za pomocą reguł wnioskowania, czyli czy  $p$  jest konsekwencją zdań z  $A$ . To właśnie realizacja rozwiązania tego problemu za pomocą łańcuchów DNA i operacji na nich jest głównym tematem tej pracy.

Obliczenia DNA (z ang. DNA computing) są nowatorską dziedziną stanowiącą jedną z alternatyw dla tradycyjnych systemów komputerowych, których możliwości znacznego rozwoju są już coraz bardziej na wyczerpaniu. Obok komputerów kwantowych czy obliczeń za pomocą procesorów graficznych (zapewniających wykonanie wielu jednoczesnych operacji), stanowią one ogromny potencjał ze względu na masowość i równoległość obliczeń. Narzędziem, wykorzystywanym do rozwiązywania problemów algorytmicznych, matematycznych czy logicznych, są tu łańcuchy DNA, a same obliczenia przeprowadzane są w odpowiednich warunkach laboratoryjnych za pomocą operacji biochemicznych na nich. Siła tych obliczeń nie tkwi w ich prędkości, a w masowej równoległości operacji, dzięki czemu problemy o naprawdę dużej złożoności przestają być niemożliwe do rozwiązania w skończonym czasie [20], [25], [37]. Komputer DNA (rozumiany jako odpowiednio dobrany zbiór cząsteczek DNA i enzymów reagujących w określonych warunkach laboratoryjnych) jest też najmniejszym modelem komputera jaki został dotąd zaprojektowany [46].

Historia obliczeń DNA sięga roku 1994, kiedy to Leonard Adleman (jeden ze współtwórców systemu kryptograficznego RSA) jako pierwszy zademonstrował możliwość rozwiązania problemu drogi Hamiltona w grafie (dla 7-miu wierzchołków) za pomocą łańcuchów DNA [1]. Zagadnienie to polega na rozstrzygnięciu czy w grafie skierowanym istnieje droga przechodząca przez wszystkie jego wierzchołki (przez każdy z nich tylko raz). Problem ten zaliczany jest do problemów NP-zupełnych, o których nie wiadomo czy rozwiązanie ich można znaleźć w czasie wielomianowym. Wzorem Adlemana później pojawiło się wiele kolejnych koncepcji rozwiązania innych problemów informatycznych, zarówno w formie



teoretycznych rozważań jak i implementacji laboratoryjnych, między innymi [16], [59], a sama koncepcja równoległego utworzenia wszystkich możliwych rozwiązań, a następnie sprawdzenia ich poprawności doczekała się swojego paradygmatu określanego „obliczaniem przez rzeźbienie” (z ang. *computing by carving*) [17]. Trochę inną gałęzią obliczeń DNA stały się zapoczątkowane w 2001 roku przez grupę naukowców z Instytutu Weizmanna w Izraelu (Yaakov Benenson, Ehud Shapiro i inni) koncepcje implementacji konkretnych modeli automatów. W pierwszej ich pracy został przedstawiony model 2-stanowego, 2-symbolowego niedeterministycznego programowalnego automatu skończonego zbudowanego z łańcuchów DNA oraz jednego enzymu restrykcyjnego [2]. Po raz pierwszy przedstawiona została idea naprzemiennego cięcia i łączenia się łańcuchów DNA reprezentujących poszczególne elementy automatu. Koncepcja była rozszerzana wielokrotnie, zarówno przez samych autorów [3], [4], [5] jak i innych [45], [49], gdzie przedstawiono ideę rozszerzenia do maszyny 3-stanowej. Ważną pracą w tej dziedzinie była praca [12], przedstawiająca laboratoryjną implementację takiej maszyny wykorzystującej jednocześnie dwa enzymy restrykcyjne i rozszerzającą możliwości do 6 stanów. Pojawiły się także koncepcje przedstawienia za pomocą DNA automatu ze stosem [6], [14] czy nawet maszyny Turinga [36] (tu jednak omawiany system nie jest autonomiczny czyli wymaga ciągłej ingerencji laboranta podczas przeprowadzania obliczeń, co niestety przeczy jego szybkości i masowej równoległości). Zatem temat molekularnej implementacji pojedynczego modelu o mocy obliczeniowej analogicznej do klasycznych komputerów jest wciąż otwarty.

Kolejną gałęzią obliczeń DNA są systemy bazujące na obliczeniach matematycznych, szczególnie w zagadnieniach logiki matematycznej. Pierwsza praca dotycząca implementacji bramek logicznych za pomocą DNA została przedstawiona przez Mitsunori Ogiharę i Animesha Raya w 1997 roku [23]. Autorzy udowodnili, że za pomocą DNA można przedstawić standardowy model przetwarzania informacji w informatyce, czyli model sieci logicznych. Koncepcje implementacji prostych systemów wnioskujących (zgodnych z przedstawionym wcześniej problemem „czy dowolne zdanie  $p$  jest konsekwencją skończonego zbioru zdań złożonych  $A$ , których prawdziwość przyjmujemy?” zaczęły pojawiać się w wielu pracach, między innymi w [48], [49]. Jako system splatania (wykorzystujący koncepcję Shapiro i Benensona zawierającą enzym restrykcyjny [2], [3]) zostały po raz pierwsze zaprezentowane w formie teoretycznej przez Olgierda Unolda i Macieja Trocia w roku 2001 [47], a w podobnym lecz popartym doświadczeniami laboratoryjnymi modelu przez zespół badawczy Ehuda Shapiro w roku 2009 [27]. Kilka istotnych koncepcji wnioskowania za pomocą DNA zostało przedstawionych też przez zespół badawczy prowadzony przez prof. dr hab. inż. Jana Mulawkę w [21], [22], [54], [56]. Żaden ze wspomnianych powyżej systemów nie zawiera jednak kompletnej implementacji wnioskowania z użyciem tautologii i praw wnioskowania klasycznego rachunku zdań. Większość z nich używa wyłącznie zdań atomowych np.  $p$ ,  $q$ ,  $r$ , nie przewidując możliwości kodowania ich zaprzeczeń np.  $\neg p$ ,  $\neg q$ ,  $\neg r$  [27], [47]. Dlatego głównym celem opracowania, znajdującego się w tej rozprawie autorskiego systemu wnioskującego, było zebranie zalet wszystkich dotychczas istniejących modeli oraz wyeliminowanie wad, które zawierają. Powstała koncepcja wciąż nie uwzględnia we wnioskowaniach wszystkich tautologii, ale

stanowi bardzo duży krok w kierunku stworzenia pełnego systemu wnioskowania logicznego zgodnego z klasycznym rachunkiem zdań.

Bardzo ciekawy model o zbliżonych możliwościach obliczeniowych do prezentowanego w tej pracy był rozwijany równolegle i po raz pierwszy został przedstawiony w roku 2010 przez grupę badawczą Alfonso Rodríguez Patón [29] a następnie rozwinięty wraz z Iñaki Sainz de Murieta i Petrem Sosíkem w [30]. Przedstawiona przez nich praca nie wykorzystuje enzymów restrykcyjnych, a jej wyniki mają charakter wyłącznie teoretyczny. Ich implementacja laboratoryjna wydaje się nieporównywalnie trudniejsza od prezentowanej tutaj.

W tej pracy została szczegółowo omówiona autorska koncepcja systemu wnioskującego uwzględniająca prawa klasycznego rachunku zdań oraz możliwość zastosowania jak największej liczby tautologii (*patrz.*: rozdział 4). Znacznie rozszerza ona możliwości systemów prezentowanych wcześniej. Pierwsze koncepcje i pomysły rozbudowy istniejących systemów wnioskujących zostały zaprezentowane w roku 2011 jako preprint Uniwersytetu Łódzkiego [31], a następnie wraz z pojawianiem się kolejnych ulepszeń i dalszym rozwojem systemu, w kolejnych publikacjach zarówno polskich [32], [34] jak i mających zasięg międzynarodowy [33], [35]. Przedstawiona implementacja laboratoryjna omawianej koncepcji opiera się przede wszystkim na doświadczeniach przeprowadzonych już wcześniej i zaprezentowanych między innymi w [12], [15], [27], [40], [42].

W rozdziale 1 omówione zostały wiadomości wstępne dotyczące logiki matematycznej i klasycznego rachunku zdań w odniesieniu do prezentowanej koncepcji systemu wnioskującego. Rozdział 2 stanowi wprowadzenie do obliczeń DNA, prezentuje podstawowe modele konstrukcji biokomputerów i przedstawia dwie najważniejsze prace z rozwoju tej dziedziny [1], [2]. W rozdziale 3 omówione zostały istniejące dotąd koncepcje związane z implementacją logiki i wnioskowania, które miały znaczny wpływ na końcowy wymiar nowego systemu wnioskującego i jego możliwości lub wiążą się z przedstawionymi w dalszej części jego dodatkowymi zastosowaniami. Rozdział 4 szczegółowo omawia wszystkie elementy nowej koncepcji, uwzględniając odniesienie do praw klasycznego rachunku zdań. Rozdział 5 stanowi uzupełnienie rozdziału 4, zawiera analizę poprawności i ewentualnych ograniczeń. Skupia się też na omówieniu dodatkowych zastosowań systemu (m. in. koncepcja częściowego rozwiązania problemu SAT czy obliczanie wartości funkcji logicznych) i możliwości dalszego rozwoju (związanych między innymi z odkryciem w przyszłości kolejnych enzymów restrykcyjnych). W rozdziale 6 szczegółowo przedstawiony został schemat przygotowania i przeprowadzenia właściwego doświadczenia w laboratorium, którego celem jest potwierdzenie poprawności omawianego systemu wnioskującego. Wprowadzona została również koncepcja zastosowania „biblioteki łańcuchów DNA”, która umożliwi użycie tych samych łańcuchów dla kolejnych eksperymentów przeprowadzanych w przyszłości.

Właściwe doświadczenie laboratoryjne nie zostało przeprowadzone ze względu na jego wysokie koszty, natomiast szczegółowo przedstawione w rozdziale 6 kroki eksperymentu laboratoryjnego zostały skonsultowane z Katedrą Genetyki Molekularnej Wydziału Biologii i Ochrony Środowiska Uniwersytetu Łódzkiego (z którą od lat nasza Katedra współpracuje w zakresie badań związanych z obliczeniami DNA). Możliwość przeprowadzenia takich

badan, a także poprawność przedstawionej symulacji doświadczenia laboratoryjnego została skonsultowana i potwierdzona przez specjalistów-genetyków.

Pragnę przede wszystkim podziękować mojemu promotorowi panu prof. dr hab. Tadeuszowi Krasieńskiemu, bez którego pomocy, życzliwości, wsparcia i wielu godzin spędzonych na analizie poniższych badań, ta praca nie miałaby szansy powstać. Ponadto podziękowania należą się także dla pana prof. Petra Sosíka, który przedstawił mi wiele spośród swoich badań związanych z obliczeniami DNA, a dzięki naszej współpracy opublikowany został wspólny artykuł prezentujący częściowo badania przedstawione tutaj [35]. Dziękuję również dr Sebastianowi Sakowskiemu za inspiracje i wprowadzenie mnie w zagadnienie obliczeń DNA na początku mojej drogi naukowej. Na koniec dziękuję Katedrze Genetyki Molekularnej Wydziału Biologii i Ochrony Środowiska Uniwersytetu Łódzkiego za czasochłonne i bardzo szczegółowe konsultacje planowanych przeze mnie doświadczeń laboratoryjnych.

## **Cel i teza pracy**

Celem pracy jest projekt w pełni autonomicznego systemu wnioskującego za pomocą łańcuchów DNA. Działanie systemu oparte jest na mechanizmie splatania czyli naprzemiennego cięcia i łączenia się łańcuchów DNA. Idea znacznie rozszerza znane do tej pory systemy wnioskujące, uwzględniając między innymi spójnik negacji i związane z nim prawa rachunku zdań.

Przeprowadzona w rozprawie analiza wskazuje, że możliwa jest koncepcja rozszerzonego, w pełni autonomicznego systemu wnioskującego za pomocą DNA.

### **Szczegółowe cele pracy:**

- Analiza znanych wcześniej koncepcji systemu wnioskującego za pomocą DNA, stanowiących podstawę opracowania nowego systemu wnioskującego.
- Opracowanie koncepcji w pełni autonomicznego systemu wnioskującego za pomocą DNA, znacznie rozszerzającego możliwości znanych wcześniej systemów, a także eliminującego ich wady.
- Wprowadzenie spójnika negacji i praw klasycznego rachunku zdań, które bezpośrednio się z nim wiążą. Umożliwienie zadawania zapytań, które również uwzględniają negację jak i o dowolne zdanie logiczne zapisane w dysjunkcyjnej postaci normalnej.
- Przedstawienie dodatkowych zastosowań omawianego systemu.
- Szczegółowa analiza przygotowania i przebiegu doświadczenia laboratoryjnego na konkretnych przykładach. Przedstawienie schematu pozyskiwania konkretnych cząsteczek DNA, ich przechowywania, namnażania oraz przygotowania dla konkretnego eksperymentu. Przedstawienie przebiegu możliwego eksperymentu oraz odczytania wyniku końcowego.
- Przedstawienie koncepcji utworzenia „biblioteki łańcuchów DNA”, umożliwiającej przechowywanie, namnażanie i kolejne wykorzystanie uzyskanych cząsteczek dla przyszłych eksperymentów.

# 1. Wiadomości wstępne z logiki matematycznej

Logika matematyczna to dział matematyki, który koncentruje się na analizowaniu zasad rozumowania matematycznego (w tym pośrednio rozumowania ludzkiego) oraz pojęć z nim związanych. W rozdziale tym omówiony zostanie tylko klasyczny rachunek zdań, gdyż jego implementacja za pomocą DNA jest głównym tematem pracy.

## 1. 1. Klasyczny rachunek zdań

Najprostszym systemem formalnym logiki matematycznej jest klasyczny rachunek zdań. Bada on zdania złożone, utworzone ze zdań atomowych (zmiennych zdaniowych) za pomocą spójników logicznych, w aspekcie ich prawdziwości lub fałszywości, zależnie od wartości zdań atomowych wchodzących w ich skład oraz w aspekcie „wynikania” jednych z nich ze zbioru innych.

Z formalnego punktu widzenia rozważamy następujący skończony alfabet złożony z sześciu symboli  $\Sigma = \{p, I, \rightarrow, \neg, (, )\}$  oraz zbiór  $\Sigma^*$  wszystkich możliwych słów utworzonych z  $\Sigma$ . Zdaniem atomowym lub zmienną zdaniową nazywamy dowolne słowo  $pI\dots I \in \Sigma^*$  utworzone z symbolu  $p$  i dowolnej liczby  $i \geq 1$  symboli  $I$ . Dalej będziemy je oznaczać poprzez  $p_i$ . Zdaniem złożonym (lub po prostu zdaniem) nazywamy dowolne „sensowne” słowo z  $\Sigma^*$ . Precyzyjna definicja jest następująca: zdaniem złożonym (nazywanym też formułą rachunku zdań) nazywamy każde słowo z  $\Sigma^*$  należące do najmniejszego zbioru  $F \subset \Sigma^*$  spełniającego następujące warunki:

1. każde zdanie atomowe  $p_i \in F$ ,
2. jeśli  $a \in F$ , to  $\neg(a) \in F$ ,
3. jeśli  $a \in F$  oraz  $b \in F$ , to  $(a) \rightarrow (b) \in F$ .

W dalszym ciągu pracy przyjmujemy następujące standardowe umowy i oznaczenia:

- (a)  $p_i \vee p_j$  oznacza  $\neg p_i \rightarrow p_j$  – definiujemy spójnik alternatywy,
- (b)  $p_i \wedge p_j$  oznacza  $\neg(p_i \rightarrow \neg p_j)$  – definiujemy spójnik koniunkcji,
- (c)  $p_i \leftrightarrow p_j$  oznacza  $(p_i \rightarrow p_j) \wedge (p_j \rightarrow p_i)$  – definiujemy spójnik równoważności,
- (d) gdy w zdaniu złożonym występuje tylko kilka zmiennych, to tradycyjnie oznaczamy je  $p, q, r, s, \dots$  zamiast formalnych oznaczeń  $p_1, p_2, p_3, p_4, \dots$ ,
- (e) opuszczamy zbędne nawiasy, przyjmując naturalną hierarchię spójników.

Do zdefiniowania tautologii (praw rachunku zdań) za pomocą metody zero-jedynkowej określamy pojęcie wartościowania. Jeśli przez  $V$  oznaczymy zbiór zmiennych atomowych, tzn.  $V = \{p_1, p_2, p_3, \dots\}$  to wartościowaniem nazywamy dowolną funkcję  $v$  na  $V$ , o wartościach w zbiorze dwuelementowym  $\{0, 1\}$  (interpretowanych jako *fałsz* i *prawda*):

$$v: V \rightarrow \{0, 1\}.$$

Intuicyjnie zatem wartościowanie to nadanie każdej zmiennej atomowej wartości logicznej. Wówczas w naturalny sposób rozszerzamy wartościowanie  $v$  na zbiór wszystkich zdań złożonych  $F$ , np. gdy  $v(p) = 0$  i  $v(q) = 1$ , to:

- (a)  $v(p \vee q) = v(p) \vee v(q) = 0 \vee 1 = 1$ ,
- (b)  $v(p \wedge q) = v(p) \wedge v(q) = 0 \wedge 1 = 0$ ,

$$(c) v(p \rightarrow q) = v(p) \rightarrow v(q) = 0 \rightarrow 1 = 1,$$

$$(d) v(\neg p) = \neg v(p) = \neg 0 = 1.$$

Tautologią (lub inaczej prawem rachunku zdań) nazywamy każde zdanie złożone  $a \in F$ , takie że  $v(a) = 1$  dla każdego możliwego wartościowania  $v$ . Na przykład tautologiami są zdania:  $p \vee \neg p$ ,  $p \rightarrow p$ ,  $(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p)$ , a nie są tautologiami zdania:  $p \vee q$ ,  $p \rightarrow \neg p$  itd.

Zbiór tautologii oznaczamy przez  $T \subset F$ .

W rachunku zdań dowodzi się następujących twierdzeń:

1. jeśli  $a, b \in F$  są zdaniami oraz  $a$  i  $a \rightarrow b$  są tautologiami, to  $b$  też jest tautologią,
2. jeśli  $a, b \in F$  są zdaniami,  $a$  jest tautologią oraz zmienna atomowa  $p_k$  występuje w  $a$ , to zdanie otrzymane z  $a$  poprzez zastąpienie zmiennej  $p_k$  przez zdanie  $b$  również jest tautologią. To nowe zdanie oznaczamy  $a(k/b)$ .

Pojęcie wartościowania zmiennych posłuży nam również do zdefiniowania wynikania zdania ze zbioru innych. Mianowicie, jeśli  $A \subset F$  jest zbiorem dowolnych zdań i  $a \in F$  jest zdaniem, to mówimy, że  $a$  wynika z  $A$  (w logice zapisujemy to jako  $A \models a$ ) gdy dla dowolnego wartościowania  $v$  zmiennych, jeśli zdania z  $A$  są prawdziwe (tzn.  $v(b) = 1$  dla każdego  $b \in A$ ), to również zdanie  $a$  jest prawdziwe (tzn.  $v(a) = 1$ ). Np.: jeśli  $A = \{p, p \rightarrow q\}$  i  $a = q$ , to oczywiście  $A \models a$ . Zauważmy, że tutaj  $a$  nie jest tautologią.

## 1. 2. Reguły wnioskowania

Przedstawmy również inny sposób uznawania zdań złożonych za prawdziwe lub wyprowadzania jednych ze zbioru innych. Jest to metoda aksjomatyczna z pojęciem formalnego dowodu. Ta metoda jest stosowana w „wyższych” teoriach matematycznych, gdyż nie ma prostego sposobu (takiego jak zero-jedynkowy w rachunku zdań) weryfikacji prawdziwości formuł tej teorii. Polega ona na przyjęciu pewnych zdań (formuł) za prawdziwe, a inne „wyprowadzamy” z nich za pomocą reguł wnioskowania. Zaczniemy od pojęcia reguł wnioskowania i formalnego dowodu. W rachunku zdań przyjmuje się 2 reguły wnioskowania:

1. reguła odrywania – jeśli zdanie  $a$  i implikację  $a \rightarrow b$  uznajemy za prawdziwe, to również zdanie  $b$  uznajemy za prawdziwe (*modus ponens*), a także: jeśli zdanie  $\neg b$  i implikację  $a \rightarrow b$  uznajemy za prawdziwe, to również zdanie  $\neg a$  uznajemy za prawdziwe (*modus tollens*),
2. reguła podstawiania – jeśli zdanie  $a$  uznajemy za prawdziwe i zdanie atomowe  $p_k$  występuje w  $a$ , to zdanie otrzymane z  $a$  przez zastąpienie zmiennej  $p_k$  przez dowolne zdanie  $b$  również uznajemy za prawdziwe.

Niech  $A \subset F$  będzie dowolnym zbiorem zdań oraz zdanie  $a \in F$ . Skończony ciąg zdań  $a_1, \dots, a_n$  nazywamy dowodem formalnym zdania  $a$  ze zbioru  $A$ , gdy:

1.  $a_n = a$  (ostatnim elementem ciągu jest zdanie  $a$ ),
2. dla dowolnego  $k \leq n$  albo (i)  $a_k \in A$  lub (ii)  $a_k$  jest tautologią lub (iii)  $a_k$  wynika na mocy reguły odrywania ze zdań  $a_i, a_j$ , gdzie  $i, j < k$ , występujących w ciągu przed  $a_k$  lub (iv)  $a_k$  powstaje z reguły podstawiania ze zdań  $a_i, a_j$ , gdzie  $i, j < k$ , występujących

w ciągu przed  $a_k$  (tzn.  $a_k = a_i (P_n / a_j)$ ), gdzie  $P_n$  jest zmienną występującą w  $a_i$ , za którą podstawiamy zmienną  $a_j$ ).

Zdanie  $a$  nazywamy konsekwencją zbioru zdań  $A \subset F$ , gdy istnieje dowód formalny  $a$  z  $A$  i oznaczamy to  $A \vdash a$ . Zbiór  $A$  nazywamy wówczas zbiorem aksjomatów.

Prawdziwe w rachunku zdań jest następujące twierdzenie o zupełności: jeśli  $A \subset F$  i  $a \in F$ , to  $a$  wynika z  $A$  wtedy i tylko wtedy gdy  $a$  jest konsekwencją  $A$  [7], tzn.:

$$A \models a \leftrightarrow A \vdash a.$$

Uwaga 1. W pojęciu dowodu formalnego można pominąć regułę podstawiania (której nie można całkowicie zaimplementować za pomocą operacji biochemicznych) kosztem rozszerzenia zbioru aksjomatów o wszystkie możliwe podstawienia.

Uwaga 2. W rozprawie zastosowana jest druga metoda (aksjomatyczna) uznawania zdań za prawdziwe. Nieznany jest dotąd sposób zaimplementowania metody zero-jedynkowej za pomocą łańcuchów DNA i dokonywanych na nich operacji. Przedstawiona w dalszej części pracy implementacja metody aksjomatycznej za pomocą łańcuchów DNA i operacji na nich będzie polegała, ogólnie mówiąc, na zadaniu pewnego zbioru zdań  $A$  w postaci łańcuchów DNA (zbiór aksjomatów) i zadaniu pytania (lub zbioru pytań) w postaci odpowiedniego łańcucha DNA (lub łańcuchów DNA), reprezentującego zdanie (lub zdania), o którym chcielibyśmy dowiedzieć się, czy jest konsekwencją  $A$  czy nie. Taki system wnioskujący ma w autonomiczny sposób dać odpowiedź na tak postawione pytanie.

### 1. 3. Postacie normalne zdań złożonych

Implementacje systemów wnioskujących za pomocą łańcuchów DNA będą używały zdań złożonych w postaci normalnej koniunkcyjnej (CNF czyli z ang. *conjunction normal form*). Przypomnijmy to pojęcie. Literałem nazywamy zdanie atomowe lub negację zdania atomowego np.  $\neg x$  lub  $y$ . Klauzula to alternatywa jednego lub więcej literałów, np.  $\neg x$ ,  $x \vee \neg y$ ,  $x \vee \neg y \vee \neg x$ . Zdanie złożone  $a \in F$  jest w postaci normalnej koniunkcyjnej, gdy jest ono koniunkcją klauzul. Np. zdania  $x \wedge y$ ,  $x \wedge (\neg x \vee y)$ ,  $(\neg x \vee y) \wedge (x \vee \neg y \vee z)$  są w postaci normalnej koniunkcyjnej, a zdania  $\neg(x \wedge (x \rightarrow y))$ ,  $\neg\neg x \leftrightarrow (\neg y \wedge x)$  nie są [44].

Uwaga. Istnieje inna postać normalna zdań złożonych, tzw. postać normalna alternatywna (nazywana również dysjunkcyjną postacią normalną), podobna do postaci normalnej koniunkcyjnej ale z zamianą rolami alternatywy i koniunkcji. Jest to zatem alternatywa zdań będących koniunkcjami literałów.

Każde zdanie złożone można sprowadzić do postaci normalnej koniunkcyjnej, korzystając z następujących praw rachunku zdań:

- $(a \leftrightarrow b) \leftrightarrow ((a \rightarrow b) \wedge (b \rightarrow a))$  – wyrażenie równoważności za pomocą implikacji,
- $(a \rightarrow b) \leftrightarrow (\neg a \vee b)$  – wyrażenie implikacji za pomocą negacji i alternatywy,
- $(\neg(a \vee b)) \leftrightarrow (\neg a \wedge \neg b)$  – pierwsze prawo de Morgana – negacja alternatywy,
- $(\neg(a \wedge b)) \leftrightarrow (\neg a \vee \neg b)$  – drugie prawo de Morgana – negacja koniunkcji,
- $\neg(\neg a) \leftrightarrow a$  – prawo podwójnego zaprzeczenia,
- $(a \wedge (b \vee c)) \leftrightarrow ((a \wedge b) \vee (a \wedge c))$  – rozdzielność alternatywy względem koniunkcji,

–  $(a \vee (b \wedge c)) \leftrightarrow ((a \vee b) \wedge (a \vee c))$  – rozdzielność koniunkcji względem alternatywy.

Twierdzenie. Każde zdanie złożone jest równoważne (tzn. przyjmuje te same wartości dla każdego wartościowania zmiennych) odpowiadającemu mu zdaniu w postaci normalnej koniunkcyjnej.

W rozprawie zakładamy, że wszystkie rozważane zdania są w postaci normalnej koniunkcyjnej. Algorytm sprowadzenia dowolnego zdania złożonego do koniunkcyjnej postaci normalnej wygląda następująco:

wejście:  $A$  – formuła logiki zdań; wyjście: formuła w CNF

1. Eliminacja  $\rightarrow$  oraz  $\leftrightarrow$  – zastępuj w formule  $A$ :

$B \rightarrow C$  przez  $\neg B \vee C$ ,

$B \leftrightarrow C$  przez  $(\neg B \vee C) \wedge (B \vee \neg C)$ ,

tak długo, aż otrzymana formuła zawiera wyłącznie spójniki  $\neg$ ,  $\wedge$ ,  $\vee$ .

2. Przesuwanie  $\neg$  „w głąb” – w formule otrzymanej w kroku 1 zastępuj:

$\neg(B \vee C)$  przez  $\neg B \wedge \neg C$ ,

$\neg(B \wedge C)$  przez  $\neg B \vee \neg C$ ,

$\neg\neg B$  przez  $B$ ,

tak długo, aż wszystkie negacje znajdują się bezpośrednio przed zmiennymi.

3. Rozdzielenie  $\wedge$  przez  $\vee$  – w formule otrzymanej w kroku 2 zastępuj:

$(B \wedge C) \vee D$  przez  $(B \vee D) \wedge (C \vee D)$ ,

$B \vee (C \wedge D)$  przez  $(B \vee C) \wedge (B \vee D)$ .

4. Zwróć otrzymaną formułę jako CNF( $A$ ).

Z punktu widzenia złożoności obliczeniowej, sprowadzenie dowolnego zdania do postaci normalnej koniunkcyjnej „wymaga” czasu wykładniczego w zależności od „długości” zdania wejściowego [44]. Możemy też użyć gotowych rozwiązań programistycznych, takich jak biblioteka SymPy dla języka programowania Python (szczegółowa dokumentacja, dostępna w latach 2016-2021, znajduje się pod adresem: <http://docs.sympy.org>), a przede wszystkim funkcji `to_cnf` oraz `to_dnf`, konwertujących dowolne zdanie złożone na wybraną postać normalną, wykorzystujących szereg optymalizacji przyspieszających działanie algorytmu (szczegółowa dokumentacja, dostępna w latach 2016-2021, znajduje się pod adresem: <http://docs.sympy.org/latest/modules/logic.html>).



## 2. Wprowadzenie do obliczeń DNA

### 2. 1. Budowa i podstawowe operacje na DNA

DNA (kwas deoksyrybonukleinowy) to polimer czyli substancja chemiczna składająca się z wielokrotnie powtórzonych jednostek, którego elementami są deoksyrybonukleotydy, w skrócie nazywane nukleotydami. Struktura DNA została odkryta w 1953 roku przez Jamesa Watsona i Francis Cricka [52], choć sama idea jego istnienia była znana już wcześniej. Zaproponowany model budowy został potwierdzony wieloma późniejszymi badaniami w tym również związanymi z tematyką obliczeń DNA [11], [13], [25]. Każdy nukleotyd zbudowany jest z reszty cukrowej (deoksyrybozy), grupy fosforanowej i zasady azotowej. Dwie pierwsze części są zawsze takie same, natomiast między sobą nukleotydy różnią się zasadami azotowymi (pochodnymi puryny i pirymidyny), które są właściwymi nośnikami informacji. Wewnątrz cząsteczek DNA występują cztery różne zasady: adenina (A), cytozyna (C), guanina (G) i tymina (T), w związku z czym nazwy nukleotydów łączą się z nazwami zasad.

W skład cząsteczki DNA zwykle wchodzi dwa łańcuchy, w których początek pierwszego łączy się z końcem drugiego. Owijają się one wokół wspólnej osi i tworzą prawoskrętną podwójną helisę. Zasady skierowane są do wnętrza i tworzą odpowiednie pary komplementarne – adenina łączy się zawsze z tyminą (dwa wiązania wodorowe), a cytozyna z guaniną (trzy wiązania wodorowe). Nie istnieją inne możliwości połączenia się nukleotydów. Łańcuchy biegą antyrównolegle, dzięki czemu znając jeden łańcuch możemy bez problemu odtworzyć drugi – na mocy zasady komplementarności Watsona-Cricka [52]. Łańcuch DNA wykazuje polarność (czyli nierównomierne rozłożenie cząstkowych ładunków elektrycznych, dzięki czemu cząsteczka jest dipolem elektrycznym). Jeden z końców ma grupę 5'-OH, a drugi 3'-OH. Przyjmuje się, że niezwiązana grupa 5'-OH jest ulokowana po lewej stronie zapisu (oznaczana skrótowo 5'), a grupa 3'-OH po prawej (oznaczana 3'). Wewnątrz pojedynczego łańcucha nukleotydy łączą się ze sobą za pomocą silnych wiązań fosfodiesterowych pomiędzy grupą fosforanową jednego nukleotydu, a deoksyrybozą kolejnego.

Na potrzeby skrótowego zapisu sekwencji DNA, na wszystkich rycinach znajdujących się w tej pracy wprowadza się następujące oznaczenia: koniec łańcucha oznaczany jest za pomocą 3' – lub 5' –; konkretne nukleotydy oznaczane są zależnie od rodzaju zasady azotowej A, C, G lub T, dowolny nukleotyd oznaczany jest przez N, konkretna liczba dowolnych nukleotydów zapisywana jest w nawiasach np. (4) oznacza NNNN czyli ciąg czterech dowolnych nukleotydów, (120) oznacza ciąg 120 dowolnych nukleotydów itd. Dowolna liczba dowolnych nukleotydów oznaczana jest jako (...).

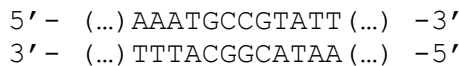
Wyróżniamy następujące podstawowe procesy biochemiczne zachodzące z udziałem fragmentów DNA: łączenie (ligacja), cięcie (trawienie, restryktacja), kopiowanie (replikacja), rozdzielanie komplementarnych nici (denaturacja), sztuczne uzyskiwanie łańcucha o dowolnej sekwencji zasad (synteza), sortowanie fragmentów DNA według ich długości (elektroforeza). Zostaną one omówione poniżej wraz z ilustrującymi ich działanie rycinami.

### 2. 1. 1. Ligacja

Nukleotydy na przeciwnych niciach łączą się ze sobą samoistnie, zgodnie z zasadą komplementarności Watsona-Cricka. W przypadku połączenia się dwóch komplementarnych fragmentów jednoniciowych enzym ligazy w rzeczywistości nie jest wykorzystywany (mimo to nazwa ta zwyczajowo używana jest dla każdego procesu łączenia się cząsteczek). Reakcja między takimi łańcuchami wygląda następująco:

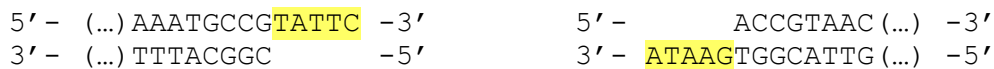


Ryc. 1a: przykładowe komplementarne łańcuchy jednoniciowe

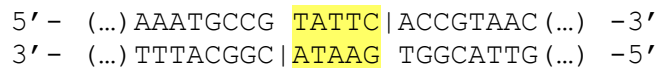


Ryc. 1b: te same łańcuchy po połączeniu się w cząsteczkę dwuniciową

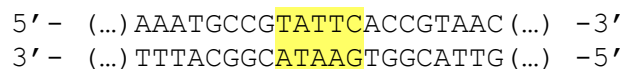
W rzeczywistości enzym ligazy wspomaga wytworzenie wiązania fosfodiesterowego pomiędzy końcem 3' a końcem 5', przez co jest istotny w przypadku łączenia się ze sobą cząsteczek o komplementarnych „lepkich końcach”. Reakcja między dwoma cząsteczkami dwuniciowymi wygląda wtedy następująco:



Ryc. 2a: przykładowe łańcuchy o komplementarnych „lepkich końcach”



Ryc. 2b: te same łańcuchy po połączeniu się nukleotydów komplementarnych



Ryc. 2c: pojedyncza cząsteczka powstała w wyniku działania enzymu ligazy

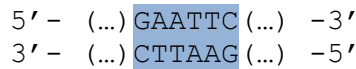
W powyższym przykładzie pokazano reakcję dla „lepkich końców” o długości 5 nukleotydów. Długość „lepkich końców” może być dowolna.

Enzym ligazy może katalizować reakcję łączenia zarówno dla pojedynczej nici, kilku fragmentów jednoniciowych czy dwuniciowych o „lepkich końcach”.

### 2. 1. 2. Restryktacja

Kolejnym procesem jest hydroliza wiązania fosfodiesterowego za pomocą enzymów restrykcyjnych czyli związków chemicznych, katalizujących precyzyjne cięcie łańcuchów DNA w określony sposób. Nazywana jest też trawieniem enzymami lub zwyczajowo „cięciem”. Enzymy restrykcyjne działają wyłącznie w odpowiednich warunkach biochemicznych (między innymi przy odpowiedniej temperaturze reakcji oraz czasie jej trwania), rozpoznają konkretną sekwencję następujących po sobie nukleotydów i tną dany łańcuch w ustalonym miejscu wewnątrz tej sekwencji lub w określonej od niej odległości. Większość z nich pozostawia „lepkie końce” (choć niektóre z nich pozostawiają „tępe końce”). Przyjrzymy się przykładowym enzymom, rozpoznawanym przez nie sekwencjom i sposobom trawienia łańcuchów:

- Enzym *EcoRI* rozpoznaje 6-cio nukleotydową sekwencję, a następnie rozcina cząsteczkę wewnątrz niej, pozostawiając zawsze identyczne „lepkie końce” o długości 4 nukleotydów:

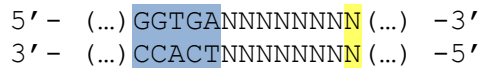


Ryc. 3a: sekwencja nukleotydów rozpoznawalna przez enzym *EcoRI*

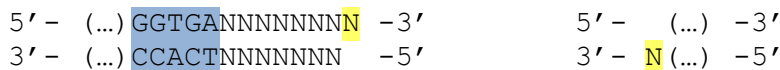


Ryc. 3b: efekt trawienia powyższej sekwencji enzymem *EcoRI*

- Enzym *HphI* rozpoznaje 5-cio nukleotydową sekwencję, a następnie rozcina cząsteczkę w odległości 8 i 7 nukleotydów za rozpoznawanym ciągiem. Pozostawia różne „lepkie końce” o długości jednego nukleotydu:

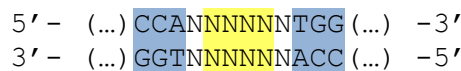


Ryc. 4a: sekwencja nukleotydów rozpoznawalna przez enzym *HphI*



Ryc. 4b: efekt trawienia powyższej sekwencji enzymem *HphI*

- Enzym *BstXI* rozpoznaje 12-to nukleotydową sekwencję, w której tylko 6 nukleotydów (3 z początku oraz 3 na końcu) musi mieć ustaloną wartość. Rozcięcie następuje wewnątrz sekwencji w obrębie 6-ciu dowolnych nukleotydów w środku sekwencji, pozostawiając „lepkie końce” o długości 4-ech nukleotydów:

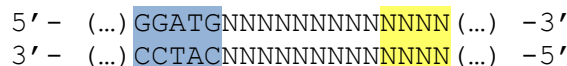


Ryc. 5a: sekwencja nukleotydów rozpoznawalna przez enzym *BstXI*

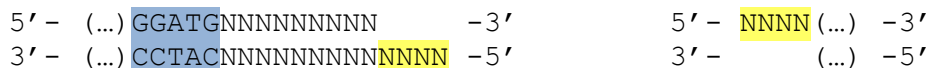


Ryc. 5b: efekt trawienia powyższej sekwencji enzymem *BstXI*

- Enzym *FokI* (wykorzystywany między innymi w pracach [27], [47] omówionych szczegółowo w rozdziale 2) rozpoznaje 5-cio nukleotydową sekwencję, a następnie rozcina cząsteczkę w odległości 9-ciu i 13-tu nukleotydów za rozpoznawanym ciągiem. Pozostawia „lepkie końce” o długości 4-ech nukleotydów:

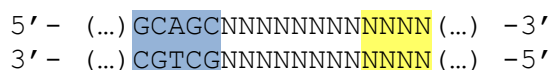


Ryc. 6a: sekwencja nukleotydów rozpoznawalna przez enzym *FokI*

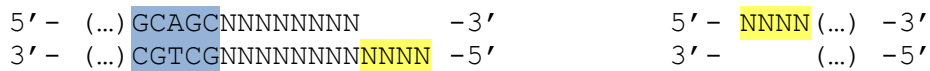


Ryc. 6b: efekt trawienia powyższej sekwencji enzymem *FokI*

- Enzym *BbvI* (wykorzystywany między innymi w pracy [47] omówionej szczegółowo w rozdziale 2 oraz w nowej koncepcji systemu wnioskującego, wprowadzonej w rozdziale 4 i będącej głównym elementem tej rozprawy) rozpoznaje 5-cio nukleotydową sekwencję, a następnie rozcina cząsteczkę w odległości 8-miu i 12-tu nukleotydów za rozpoznawanym ciągiem. Pozostawia „lepkie końce” o długości 4-ech nukleotydów:

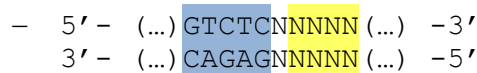


Ryc. 7a: sekwencja nukleotydów rozpoznawalna przez enzym *BbvI*

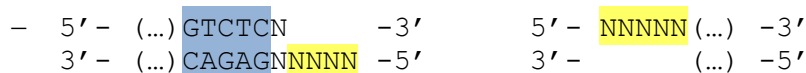


Ryc. 7b: efekt trawienia powyższej sekwencji enzymem BbvI

- Enzym *BsmAI* (wykorzystywany do przygotowania „lepkich końców” cząsteczek nowej koncepcji systemu wnioskującego tuż przed właściwą reakcją, omówioną w rozdziale 6) rozpoznaje 6-cio nukleotydową sekwencję, a następnie rozcina cząsteczkę w odległości 1-go oraz 5-ciu nukleotydów za rozpoznawanym ciągiem. Pozostawia „lepkie końce” o długości 4-ech nukleotydów:

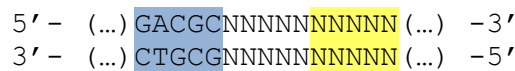


– Ryc. 8a: sekwencja nukleotydów rozpoznawalna przez enzym *BsmAI*

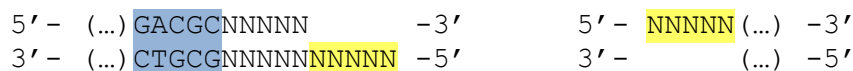


– Ryc. 8b: efekt trawienia powyższej sekwencji enzymem *BsmAI*

- Enzym *HgaI* rozpoznaje 5-cio nukleotydową sekwencję, a następnie rozcina cząsteczkę w odległości 5-ciu oraz 10-ciu nukleotydów za rozpoznawanym ciągiem. Jak dotąd jest to jedyny zidentyfikowany enzym, który pozostawia „lepkie końce” o długości 5-ciu nukleotydów:



Ryc. 9a: sekwencja nukleotydów rozpoznawalna przez enzym *HgaI*



Ryc. 9b: efekt trawienia powyższej sekwencji enzymem *HgaI*

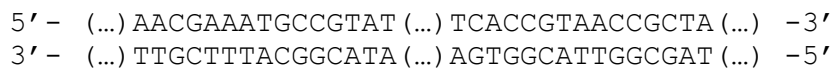
Omówione powyżej enzymy to jedynie kilka przykładowych. Dokładniejsze opisy większości dostępnych enzymów, ich działania i wymaganego środowiska znajdziemy w [25], a szczegółowe dane techniczne na stronach odpowiednich laboratoriów, między innymi: [www.neb.com](http://www.neb.com) (dostępne w latach 2016-2021), oraz za pomocą narzędzia Enzyme Finder: [enzymefinder.neb.com](http://enzymefinder.neb.com) (dostępnego od roku 2020, wcześniej jako podstrona laboratorium).

Istnieje możliwość zastosowania kilku enzymów jednocześnie, jeżeli wymagane przez nie środowisko reakcji jest takie samo. Badania laboratoryjne potwierdzające taką możliwość zostały przeprowadzone i opisane między innymi w [12] i [15].

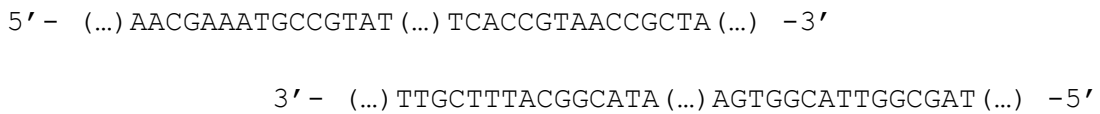
### 2. 1. 3. Replikacja

Za kopiowanie jednego łańcucha na drugi odpowiadają enzymy polimerazy DNA – zamiast jednej pojawiają się dwie dwuniciowe cząsteczki DNA, z których każda zawiera jedną nić macierzystą i jedną nowopowstałą. Obie dwuniciowe cząsteczki są identyczne. Reakcja łańcuchowa polimerazy (z ang. *Polimerase Chain Reaction, PCR*) polega na denaturacji („rozpleceniu”) podwójnej helisy na dwie pojedyncze nici, dołączeniu krótkich fragmentów komplementarnych do obu nici (tzw. „starterów”) i dobudowaniu brakującej części za pomocą wspomnianych enzymów. Polimeryzacja następuje zawsze w kierunku od 5’ do 3’. Jednokrotna reakcja podwaja liczbę kopii cząsteczek DNA, zatem wielokrotne jej zastosowanie powoduje przyrost wykładniczy.

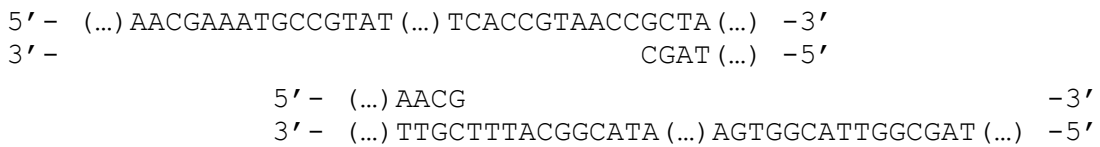
Przykładowa pojedyncza reakcja wygląda następująco:



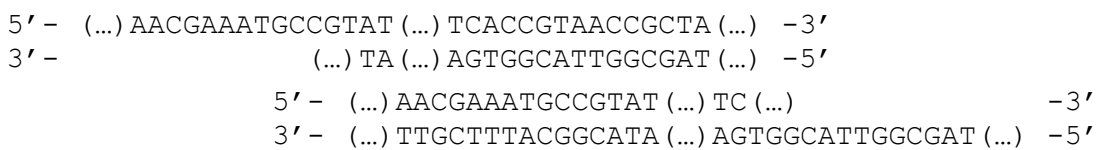
*Ryc. 10a: dwuniciowa cząsteczka DNA przed rozpoczęciem procesu replikacji*



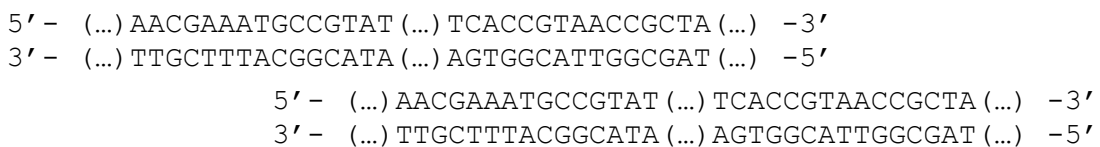
*Ryc. 10b: rozdzielenie (denaturacja) do postaci dwóch jednoniciowych łańcuchów*



*Ryc. 10c: dołączenie „starterów” czyli krótszych fragmentów komplementarnych*



*Ryc. 10d: reakcja łańcuchowa polimerazy w trakcie działania*



*Ryc. 10e: zakończona reakcja łańcuchowa polimerazy – zamiast jednej uzyskujemy dwie identyczne cząsteczki DNA*

Startery wykorzystywane w PCR mają zwyczajowo 15-25 nukleotydów (na powyższej rycinie ich długość nie została sprecyzowana – zastosowano ogólny symbol (...)), dzięki czemu nie jest możliwe aby losowo przyłączały się w wielu nieoczekiwanych miejscach.

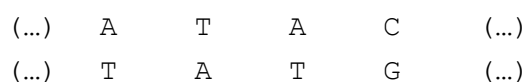
Więcej szczegółów na temat omówionego procesu możemy znaleźć między innymi w [25].

## 2. 1. 4. Denaturacja

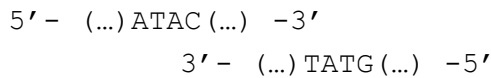
Zjawisko rozdzielania komplementarnych nici pod wpływem wysokiej temperatury lub odpowiednich substancji chemicznych nazywamy denaturacją. Wykorzystywana jest między innymi w omówionym wcześniej procesie replikacji (*patrz*: podrozdział 2.1.3). W wyniku powolnego ochładzania, nici łączą się ponownie. Zjawisko ponownego łączenia tych fragmentów nazywamy renaturacją.

## 2. 1. 5. Synteza

Sztuczne uzyskiwanie łańcucha o dowolnej sekwencji zasad nazywamy syntezą DNA. We współczesnych laboratoriach biologii molekularnej bez problemu można w ten sposób uzyskać każdą żadaną sekwencję. W pierwszej kolejności wytwarzane są dwa fragmenty jednoniciowe (nukleotyd po nukleotydzie), a następnie zlepiane są ze sobą komplementarnie.



*Ryc. 11a: synteza pojedynczych nukleotydów w odpowiedniej kolejności*



Ryc. 11b: połączenie nukleotydów do jednoniciowych oligonukleotydów



Ryc. 11c: łączenie ze sobą powstałych łańcuchów na zasadzie komplementarności

## 2. 1. 6. Elektroforeza

Sortowanie cząsteczek DNA według ich długości jest możliwe za pomocą elektroforezy w żelu poliakrylamidowym lub agarozowym. Roztwór z analizowanymi łańcuchami przenosi się na specjalną płytkę z żelazem elektroforetycznym, wzdłuż krawędzi której biegają elektrody. Technika oparta jest na właściwości, że cząsteczki DNA posiadają ujemny ładunek elektryczny (ze względu na zawartość grup fosforanowych). Pod wpływem stałego napięcia elektrycznego dwuniciowe łańcuchy przemieszczają się w kierunku dodatnio naładowanej elektrody. Natomiast użycie żelu sprawia, że układają się one zgodnie ze swoją długością – dłuższe łańcuchy przemieszczają się wolniej przez pory żelu (bez zastosowania żelu wszystkie przemieszczałyby się z tą samą prędkością) [25]. Współczesne laboratoria biologii molekularnej pozwalają posortować i odróżnić od siebie łańcuchy o różnicach długości nawet pojedynczego nukleotydu.

UJEMNY ŁADUNEK ELEKTRYCZNY

(...)

5' - ATCGCGTTCCGACT -3'

3' - TAGCGCAAGGCTGA -5'

5' - ATCGCGTTCCGACTATCG -3'

3' - TAGCGCAAGGCTGATAGC -5'

5' - (100)ATCG -3'

3' - (100)TAGC -5'

5' - ATCGCGTTCCGACTATCGATCGCGTTCCGACTATCG -3'

3' - TAGCGCAAGGCTGATAGCTAGCGCAAGGCTGATAGC -5'

5' - (100)AT -3'

3' - (100)TA -5'

(...)

DODATNI ŁADUNEK ELEKTRYCZNY

Ryc. 12a: przykładowe cząsteczki DNA ułożone w przypadkowej kolejności na płycie żelu elektroforetycznego

UJEMNY ŁADUNEK ELEKTRYCZNY

(...)

5' - (100)ATCG -3'

3' - (100)TAGC -5'

5' - (100)AT -3'

3' - (100)TA -5'

5' - ATCGCGTTCCGACTATCGATCGCGTTCCGACTATCG -3'

3' - TAGCGCAAGGCTGATAGCTAGCGCAAGGCTGATAGC -5'

5' - ATCGCGTTCCGACTATCG -3'

3' - TAGCGCAAGGCTGATAGC -5'

(Ryc. 12b, część na kolejnej stronie) →

5' - ATCGCGTTCCGACT -3'  
3' - TAGCGCAAGGCTGA -5'  
(...)

DODATNI ŁADUNEK ELEKTRYCZNY

*Ryc. 12b: te same cząsteczki DNA po zakończeniu reakcji elektroforezy w żelu (krótsze przemieściły się szybciej w kierunku dodatnim niż te dłuższe)*

## 2. 2. Modele biokomputerów

Istnieje wiele teoretycznych modeli komputerów zbudowanych z DNA i powstało wiele ich teoretycznych opisów. Znaczna część została szczegółowo omówiona w [11], [25]. Niektóre koncepcje zostały potwierdzone badaniami laboratoryjnymi, między innymi w [1], [12], [27]. Oprócz podziału na wykorzystywane narzędzia obliczeń i algorytmy (po części omówione w kolejnych podrozdziałach), komputery biomolekularne możemy też podzielić na systemy autonomiczne i nieautonomiczne.

Biomolekularne systemy autonomiczne, to takie, które nie wymagają udziału laboranta w trakcie obliczeń. Całość reakcji sprowadza się do trzech kroków: przygotowania cząsteczek wejściowych, właściwej reakcji oraz odczytania wyniku. W tym przypadku udział człowieka jest niezbędny tylko w pierwszym i ostatnim kroku, podczas gdy najważniejsza część czyli przetwarzanie danych w probówce, odbywa się samoczynnie. Przykładami systemów w pełni autonomicznych są między innymi te opisane w [2], [12], [27], [47] jak i nowa koncepcja wprowadzona w rozdziale 4. Istnieją też autonomiczne systemy, które wyłącznie w trzecim kroku (odczytywaniu wyniku) wymagają kilku nieautonomicznych ruchów – do takich właśnie należą pierwsze prace z zakresu DNA obliczeń, między innymi [1], [16].

Biomolekularne systemy nieautonomiczne wymagają ciągłej obecności laboranta, który uczestniczy praktycznie w każdym kroku reakcji i ingeruje w jej przebieg. Zwiększa to możliwości obliczeniowe, jednak czas wykonania takiego algorytmu nie jest już odpowiednio atrakcyjny. W roku 1995 Paul Rothmund przedstawił nawet teoretyczną koncepcję implementacji maszyny Turinga za pomocą DNA [36]. Gdyby istniała możliwość implementacji takiego systemu w wersji autonomicznej, prawdopodobnie możliwości obliczeniowe komputerów biomolekularnych mogłyby przewyższać możliwości komputerów tradycyjnych. Niestety jednak systemy nieautonomiczne nie są tu konkurencyjne.

## 2. 3. Model filtrowania

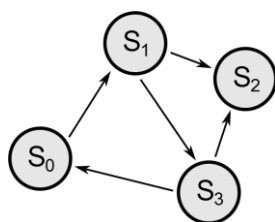
Pierwszy przykład komputera DNA został przedstawiony we wspomnianej już pracy Leonarda Adlemana z 1994 [1] (*patrz*: wstęp), który przedstawił propozycję rozwiązania znanego problemu NP-zupełnego – drogi Hamiltona w grafie, który polega na poszukiwaniu takiej drogi w grafie skierowanym (lub nieskierowanym), która przechodzi przez wszystkie jego wierzchołki, jednak nie powtarzając żadnego z nich.

Formalny model takiego schematu obliczeń został przedstawiony w pracy Richarda Liptona [16]. Wykorzystał on te same operacje molekularne co Adleman dla rozwiązania innego znanego problemu NP-zupełnego 3SAT. Idea modelu filtrowania jest następująca: wszystkie dane wejściowe kodowane są za pomocą odpowiednich jednoniciowych łańcuchów DNA, a następnie zmieszane ze sobą wewnątrz jednej probówki. W wyniku reakcji z użyciem

enzymu ligazy, łańcuchy te łączą się ze sobą na wszystkie możliwe sposoby, tworząc różne dwuniciowe łańcuchy DNA – równocześnie dzięki masowej równoległości takich operacji. Reakcja jest w pełni autonomiczna. Jeżeli istnieje konkretne rozwiązanie danego problemu, to wśród powstałych łańcuchów istnieją też takie, które właśnie to rozwiązanie reprezentują. Zatem należy je tylko „odkryć” lub inaczej „odfiltrować” (stąd przyjęta nazwa modelu) – jest to najtrudniejszy fragment omawianej reakcji, często nieautonomiczny i wymagający już udziału laboranta.

W pracy Adlemana [1] każdy z wierzchołków grafu został zakodowany w formie pojedynczej nici 5'-3', za pomocą unikalnej sekwencji nukleotydów o ustalonej długości (dowolnej lecz tej samej dla wszystkich wierzchołków). Krawędzie przejścia między wierzchołkami zostały również zakodowane w postaci pojedynczych nici 3'-5' za pomocą tej samej liczby nukleotydów, w taki sposób, że pierwsza jej połowa jest komplementarna z wierzchołkiem od którego poprowadzona jest krawędź, a druga z wierzchołkiem docelowym. Połowy tych łańcuchów nie mogą się powtarzać. W wyniku właściwej autonomicznej reakcji powstają dwuniciowe cząsteczki reprezentujące wszystkie istniejące w danym grafie drogi, włącznie z drogami zawierającymi pętle. Na nici 5'-3' będą znajdować się ciągi reprezentujące kolejne wierzchołki danej drogi, a na nici 3'-5' połączone ze sobą użyte krawędzie przejść. Druga, nieautonomiczna część reakcji polega na odfiltrowaniu wszystkich cząsteczek niezawierających którejkolwiek z sekwencji reprezentujących wierzchołki, czyli tych dróg które przez któryś z wierzchołków nie prowadzą. W ten sposób, jeżeli w próbówce wciąż pozostaną cząsteczki, to zawierają one informacje o ścieżkach przechodzących przez wszystkie wierzchołki. Ostatnim krokiem jest sprawdzenie, czy któryś z nich się nie powtarza. Możemy to uzyskać poprzez sprawdzenie długości łańcucha DNA (musi zawierać on liczbę nukleotydów kodujących pojedynczy wierzchołek pomnożoną przez liczbę wierzchołków, ponieważ drogi niezawierające któregośkolwiek z nich zostały już odfiltrowane w poprzednim kroku). Po tych operacjach, jeżeli próbówka nie pozostała pusta, istnienie drogi Hamiltona w zakodowanym grafie jest pewne.

Dla zrozumienia schematu weźmy przykład następującego grafu skierowanego:



Ryc. 13: przykład prostego grafu skierowanego o czterech wierzchołkach

Przyjmijmy następujące molekularne kodowanie wierzchołków:



Ryc. 14a: przykładowe kodowanie wierzchołka  $S_0$  w powyższym grafie



Ryc. 14b: przykładowe kodowanie wierzchołka  $S_1$  w powyższym grafie



Ryc. 14c: przykładowe kodowanie wierzchołka  $S_2$  w powyższym grafie





Ryc. 14d: przykładowe kodowanie wierzchołka  $S_3$  w powyższym grafie

oraz odpowiednio komplementarne kodowanie krawędzi:



Ryc. 15a: kodowanie przejścia  $S_0 \rightarrow S_1$  w powyższym grafie



Ryc. 15b: kodowanie przejścia  $S_1 \rightarrow S_2$  w powyższym grafie



Ryc. 15c: kodowanie przejścia  $S_1 \rightarrow S_3$  w powyższym grafie

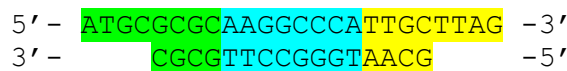


Ryc. 15d: kodowanie przejścia  $S_3 \rightarrow S_0$  w powyższym grafie

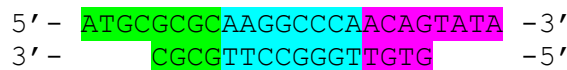


Ryc. 15e: kodowanie przejścia  $S_3 \rightarrow S_2$  w powyższym grafie

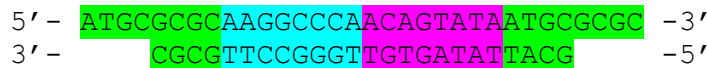
W trakcie reakcji, w wyniku odpowiedniego łączenia się cząsteczek (a właściwie ich „lepkich końców”) z użyciem enzymu ligazy, uzyskamy między innymi następujące łańcuchy dwuniciowe, reprezentujące istniejące w danym grafie różne drogi:



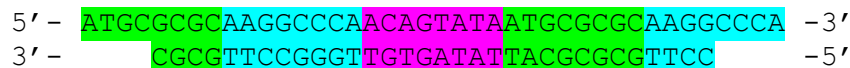
Ryc. 16a: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_2$



Ryc. 16b: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3$



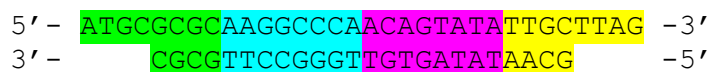
Ryc. 16c: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$



Ryc. 16d: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0 \rightarrow S_1$

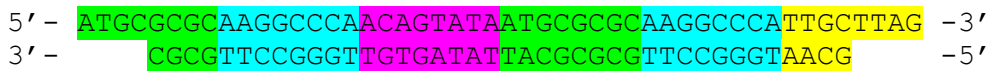


Ryc. 16e: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0 \rightarrow S_1 \rightarrow S_2$

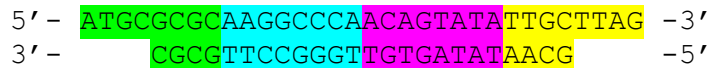


Ryc. 16f: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2$

Ze względu na istnienie w przykładowym grafie pętli, powstałych dróg będzie jeszcze więcej, ale ponieważ żadna z nich nie będzie drogą Hamiltona, możemy pominąć je i pozostać przy analizowaniu 7-miu powyższych możliwości. Jednym ze sposobów nieautonomicznego filtrowania jest podzielenie zawartości próbki na dwie części – zawierającą i niezawierającą konkretnego podłańcucha (omówione szczegółowo w [25]). Jeżeli wykonamy tę operację kolejno dla każdego z fragmentów reprezentujących każdy wierzchołek i odfiltrujemy wszystkie cząsteczki, które danego łańcucha nie zawierają, to uzyskamy próbkę zawierającą tylko te drogi, które przechodzą przez wszystkie wierzchołki. W omawianym przykładzie będą to:

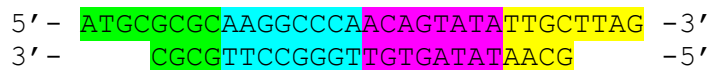


Ryc. 17a: dwuniciowa sekwencja reprezentująca drogę łączącą wszystkie wierzchołki  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0 \rightarrow S_1 \rightarrow S_2$



Ryc. 17b: dwuniciowa sekwencja reprezentująca drogę łączącą wszystkie wierzchołki  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2$

Omawiany graf zawiera 4 wierzchołki, a każdy z nich reprezentowany jest przez 8 nukleotydów. Oznacza to, że najkrótsza droga łącząca wszystkie wierzchołki będzie reprezentowana dokładnie przez  $8 \cdot 4$  czyli 32 nukleotydy, zatem po odfiltrowaniu pozostałych cząsteczek DNA względem długości, pozostanie nam jedna:



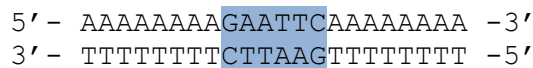
Ryc. 18: dwuniciowa sekwencja reprezentująca drogę Hamiltona  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2$

W przypadku grafów nieposiadających drogi Hamiltona, próbówka po tych operacjach pozostanie pusta. Istnieje też możliwość istnienia więcej niż jednej drogi Hamiltona w grafie.

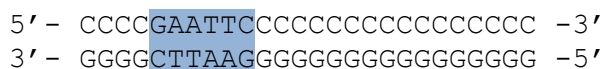
Poszukiwanie drogi Hamiltona za pomocą DNA było rozszerzane i analizowane w kolejnych pracach, między innymi w [59]. Model filtrowania, opisany w pracy Liptona na przykładzie NP-zupełnego problemu 3SAT, wykorzystywany był również w implementacji logiki [9] (omówiony pokrótce w podrozdziale 3.3).

## 2. 4. Model splatania

Innym teoretycznym modelem komputera DNA jest model splatania, który wykorzystuje naprzemienne działanie enzymu ligazy oraz enzymu restrykcyjnego. Po raz pierwszy został opisany w roku 1987 przez Toma Heada w [8], a następnie rozwinięty w [2]. Splatanie dwóch łańcuchów pierwotnie przedstawione zostało jako rozcięcie ich w konkretnych punktach (wyznaczonych przez konkretny enzym restrykcyjny), w taki sposób, że powstałe „lepkie końce” są ze sobą komplementarne, a następnie połączenie ich krzyżowo (z wykorzystaniem enzymu ligazy). Oznacza to, że jeżeli mamy łańcuchy  $a_1Ea_2$  oraz  $b_1Eb_2$  (gdzie  $E$  oznacza fragment rozpoznawany przez enzym restrykcyjny), to w wyniku ich splecenia otrzymamy dodatkowo łańcuchy  $a_1Eb_2$  oraz  $b_1Ea_2$  [25]. Weźmy przykład wykorzystujący omówiony wcześniej enzym *EcoRI* (patrz: podrozdział 2.1.2) oraz dwa łańcuchy DNA:



Ryc. 19a: przykładowa sekwencja  $a_1Ea_2$  zawierająca miejsce rozpoznawane przez enzym *EcoRI*

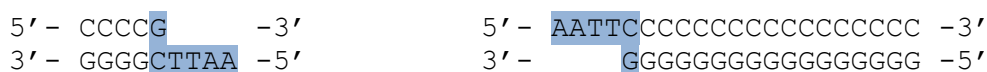


Ryc. 19b: przykładowa sekwencja  $b_1Eb_2$  zawierająca miejsce rozpoznawane przez enzym *EcoRI*

W wyniku trawienia omawianym enzymem uzyskamy:



Ryc. 20a: sekwencja  $a_1Ea_2$  rozcięta za pomocą enzymu *EcoRI*

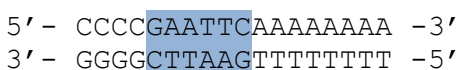


Ryc. 20b: sekwencja  $b_1Eb_2$  rozcięta za pomocą enzymu *EcoRI*

A w wyniku połączenia z użyciem enzymu ligazy uzyskamy:

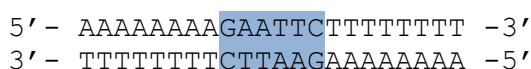


Ryc. 21a: sekwencja  $a_1Eb_2$  powstała w wyniku splecenia

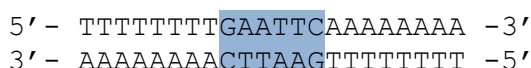


Ryc. 21b: sekwencja  $b_1Ea_2$  powstała w wyniku splecenia

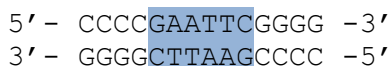
Warto zwrócić uwagę, że zarówno enzym *EcoRI*, jak i przykładowe enzymy wykorzystywane w [25] mają symetryczną sekwencję nukleotydów „lepszego końca”, co oznacza że oprócz wspomnianych molekuł  $a_1Eb_2$  oraz  $b_1Ea_2$ , powstaną też  $a_1Ea_1'$ ,  $a_2Ea_2'$ ,  $b_1Eb_1'$ ,  $b_2Eb_2'$ , gdzie przez „'” oznaczamy cząsteczkę obróconą o 180° względem postaci pierwotnej. Omawiane molekuly po powyższej reakcji będą wyglądać następująco:



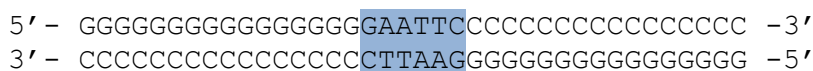
Ryc. 22a: sekwencja  $a_1Ea_1'$  powstała w wyniku splecenia



Ryc. 22b: sekwencja  $a_2Ea_2'$  powstała w wyniku splecenia



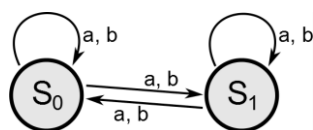
Ryc. 22c: sekwencja  $b_1Eb_1'$  powstała w wyniku splecenia



Ryc. 22d: sekwencja  $b_2Eb_2'$  powstała w wyniku splecenia

Zupełnie inne podejście do modelu splatania, zostało zaprezentowane po raz pierwszy w pracy „Programmable and autonomous computing machines made of biomolecules” przez zespół badawczy Yaakowa Benensona na łamach Nature w roku 2001 [2]. Nie wykorzystuje ono krzyżowego splatania molekuł ze sobą, a łączenie się łańcuchów DNA, powstałych w wyniku trawienia enzymami, z innymi odpowiadającymi im komplementarnie, powstałymi w wyniku innego trawienia (lub znajdującymi się w próbówce od początku reakcji). Daje to jeszcze większe zróżnicowanie nowych molekuł powstających w trakcie reakcji.

We wspomnianej pracy [2] przedstawiony został programowalny automat (czyli najprostszy model komputera) zbudowany z DNA, enzymu restrykcyjnego *FokI* (patrz: podrozdział 2.1.2) oraz enzymu ligazy. Automat ten, dzisiaj nazywany również automatem Benensona-Shapiro, jest dwustanowym niedeterministycznym automatem skończonym i umożliwia rozwiązywanie prostych problemów algorytmicznych. Ma on  $2^3$  czyli 8 możliwych ruchów, które przedstawione są na poniższej rycinie. Wybierając tylko część z dostępnych ruchów, możemy skonstruować dowolny niedeterministyczny automat skończony zawierający maksymalnie dwa stany i dwa symbole.

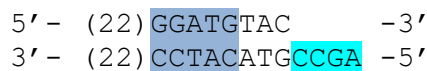


Ryc. 23: wszystkie możliwe przejścia automatu dwustanowego, dwusymbolowego

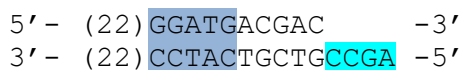
W stosunku do pierwotnej wersji systemu splatania [25], jak już zostało wspomniane, nie wykorzystuje on krzyżowego łączenia się molekuł. Poprzez zastosowanie cięcia poza sekwencją rozpoznawalną i różnych „lepkich końców” cząsteczek DNA biorących udział w reakcji, w większości wykorzystuje on do dalszego działania tylko jedną z powstałych (w wyniku trawienia enzymem) cząsteczek. W trakcie obliczeń, dla niemalże każdego „lepkiego końca”, znajdzie się jego komplementarny odpowiednik, który umożliwia połączenie się i dalszą reakcję. Główna zasada (w stosunku do pierwotnego, omówionego wcześniej modelu) naprzemiennych cięć i łączeń między łańcuchami została zachowana. Względem koncepcji pierwotnej zwiększyła się uniwersalność i deterministyczność splatania – możemy łatwiej przewidzieć, które molekuly połączą się ze sobą, jeżeli nie wszystkie posiadają tę samą sekwencję nukleotydów w obrębie „lepkiego końca”.

W automacie zaprezentowanym przez Benensona, Shapiro i innych, wszystkie jego elementy: wybrane wierzchołki (możemy wybrać automat wykorzystujący tylko jeden wierzchołek), wybrane przejścia między wierzchołkami, a także słowo wejściowe, kodowane są za pomocą odpowiednich sekwencji nukleotydów a następnie mieszane w probówce. W wyniku naprzemiennego cięcia i łączenia, do którego sprowadza się przetwarzanie słowa wejściowego, na końcu reakcji otrzymujemy (lub nie) odpowiednią sekwencję terminalną, która oznacza akceptowanie (lub nie) wprowadzonego słowa. Podobnie jak w eksperymencie Adlemana, samo wykrycie odpowiedzi jest tu działaniem nieautonomicznym i wymaga łańcuchowej reakcji polimerazy oraz elektroforezy w żelu.

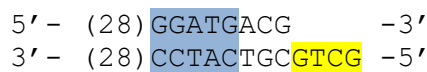
Sekwencja rozpoznawalna przez enzym restrykcyjny *FokI* znajduje się wewnątrz łańcuchów reprezentujących przejścia automatu. Między sobą różnią się one „lepkim końcem” lub długością. Przyjęto następujące kodowanie wszystkich ośmiu możliwych przejść:



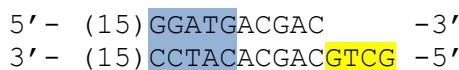
Ryc. 24a: sekwencja kodowania przejścia  $S_0 \rightarrow S_0$ , dla symbolu *a*



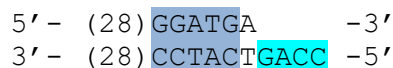
Ryc. 24b: sekwencja kodowania przejścia  $S_0 \rightarrow S_1$ , dla symbolu *a*



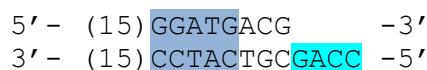
Ryc. 24c: sekwencja kodowania przejścia  $S_0 \rightarrow S_0$ , dla symbolu *b*



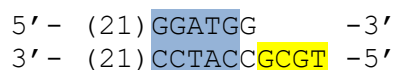
Ryc. 24d: sekwencja kodowania przejścia  $S_0 \rightarrow S_1$ , dla symbolu *b*



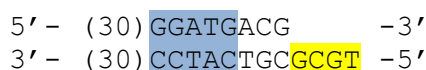
Ryc. 24e: sekwencja kodowania przejścia  $S_1 \rightarrow S_0$ , dla symbolu *a*



Ryc. 24f: sekwencja kodowania przejścia  $S_1 \rightarrow S_1$ , dla symbolu *a*

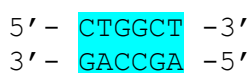


Ryc. 24g: sekwencja kodowania przejścia  $S_1 \rightarrow S_0$ , dla symbolu *b*

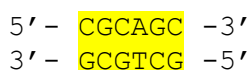


Ryc. 24h: sekwencja kodowania przejścia  $S_1 \rightarrow S_1$ , dla symbolu  $b$

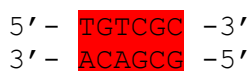
Kodowanie słowa wejściowego, należącego do alfabetu zbudowanego wyłącznie z symboli  $a$  i  $b$ , tzn.  $A \in \{a, b\}^*$  zaczyna się „lepkiem końcem” na nici 5’–3’, który koduje pierwszy symbol oraz stan początkowy. Następnie w formie dwuniciowej kodowane są kolejne symbole, zakończone odpowiednią stałą sekwencją terminalną (wprowadzoną jako trzeci symbol  $t$ ). Poszczególne symbole kodowane są w następujący sposób:



Ryc. 25a: kodowanie symbolu  $a$

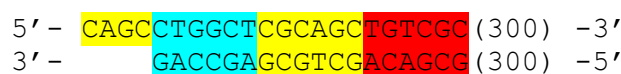


Ryc. 25b: kodowanie symbolu  $b$



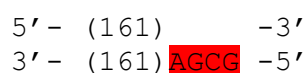
Ryc. 25c: kodowanie symbolu terminalnego  $t$

Należy zauważyć, że stan początkowy kodowany jest poprzez odpowiednie rozcięcie fragmentu kodowania symbolu – jeżeli „lepki koniec” zawiera tylko cztery ostatnie nukleotydy symbolu, oznacza to stan  $S_0$ , jeżeli zaś cztery pierwsze (zatem dwa pozostałe nukleotydy są w formie dwuniciowej), oznacza to stan  $S_1$ . Kodowanie słowa  $B=bab$  dla stanu początkowego  $S_0$  wygląda następująco:

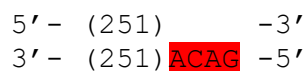


Ryc. 26: kodowanie słowa  $B=bab$  i stanu początkowego  $S_0$  zgodnie z powyższym schematem

Dla rozpoznania wyniku reakcji, niezbędne są jeszcze dwie cząsteczki detektorowe. Pierwsza odpowiada za detekcję pojawienia się symbolu terminalnego w stanie  $S_0$ , druga zaś w stanie  $S_1$ . Wyglądają one następująco:



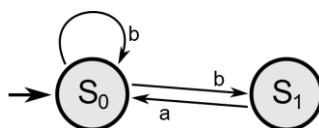
Ryc. 27a: sekwencja detektorowa dla stanu  $S_0$



Ryc. 27b: sekwencja detektorowa dla stanu  $S_1$

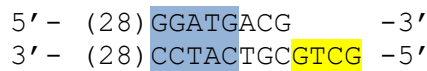
Fragmenty niezdefiniowane tutaj, ale o konkretnej ilości nukleotydów (15, 21, 22, 28 dla łańcuchów reprezentujących przejścia automatu, 300 dla kodowania słowa oraz 161, 251 dla sekwencji detektorowych) pochodzą bezpośrednio z omawianej pracy i zostały dobrane tak by rozpoznanie odpowiedzi systemu nie budziło żadnych wątpliwości.

Weźmy przykładowy automat znajdujący się na poniższej rycinie:

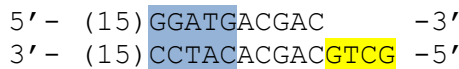


Ryc. 28: przykładowy automat uzyskany poprzez ograniczenie modelu głównego do trzech przejść

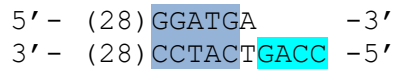
Jego przejścia zakodowane są za pomocą następujących łańcuchów DNA:



Ryc. 29a: sekwencja kodowania przejścia  $S_0 \rightarrow S_0$ , dla symbolu b

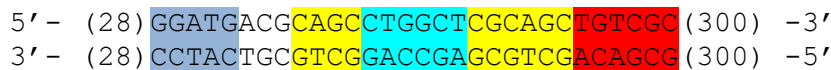


Ryc. 29b: sekwencja kodowania przejścia  $S_0 \rightarrow S_1$ , dla symbolu b



Ryc. 29c: sekwencja kodowania przejścia  $S_1 \rightarrow S_0$ , dla symbolu a

Po umieszczeniu w probówce cząsteczek reprezentujących przejścia, zakodowanego słowa wejściowego  $B=bab$  oraz cząsteczek detektorowych, rozpocznie się reakcja. Ze względu na niedeterministyczność, niektóre ruchy zachodzą równoległe (patrz: ryc. 30a, b oraz 30c, d). Omawiana reakcja będzie wyglądać następująco:

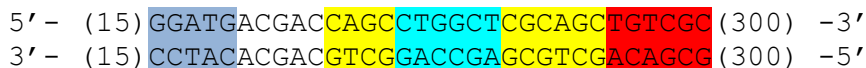


Ryc. 30a: ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_0$ , dla symbolu b

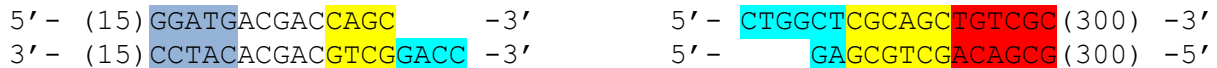


Ryc. 30b: trawienie powyższej cząsteczki enzymem restrykcyjnym FokI

(powstała cząsteczka reprezentuje stan  $S_0$  oraz symbol a; reakcja zostaje zatrzymana)



Ryc. 30c: ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_1$ , dla symbolu b

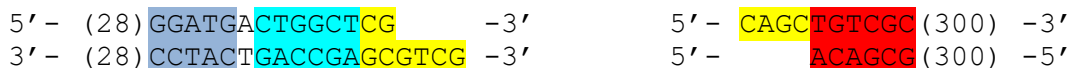


Ryc. 30d: trawienie powyższej cząsteczki enzymem restrykcyjnym FokI

(powstała cząsteczka reprezentuje stan  $S_1$  oraz symbol a)

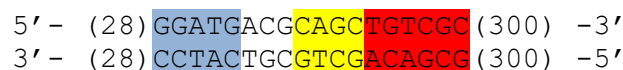


Ryc. 30e: ligacja cząsteczek słowa wejściowego i przejścia  $S_1 \rightarrow S_0$ , dla symbolu a



Ryc. 30f: trawienie powyższej cząsteczki enzymem restrykcyjnym FokI

(powstała cząsteczka reprezentuje stan  $S_0$  oraz symbol b)

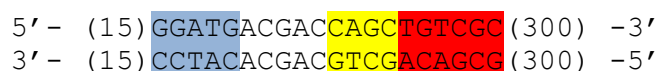


Ryc. 30g: ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_0$ , dla symbolu b

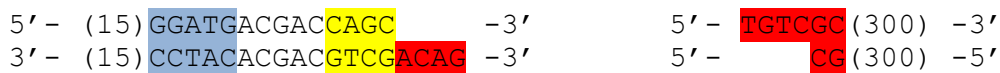


Ryc. 30h: trawienie powyższej cząsteczki enzymem restrykcyjnym FokI

(powstała cząsteczka reprezentuje symbol terminalny oraz stan  $S_0$ )



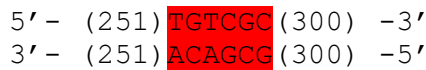
Ryc. 30i: jednoczesna ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_1$ , dla symbolu b



Ryc. 30j: trawienie powyższej cząsteczki enzymem restrykcyjnym FokI (powstała cząsteczka reprezentuje symbol terminalny oraz stan  $S_0$ )



Ryc. 30k: ligacja cząsteczki detektorowej dla stanu  $S_0$  oraz symbolu terminalnego w stanie  $S_0$



Ryc. 30l: ligacja cząsteczki detektorowej dla stanu  $S_1$  oraz symbolu terminalnego w stanie  $S_1$

Niezależnie od tego, który ze stanów traktujemy jako stan końcowy, reakcja wygląda tak samo. Akceptacja słowa wejściowego w stanie końcowym  $S_0$  reprezentowana jest przez dwuniciowy łańcuch DNA, niezawierający „lepkich końców” o długości 465 nukleotydów, zaś w stanie końcowym  $S_1$  przez analogiczny fragment o długości 557 nukleotydów. Jeżeli tylko jeden ze stanów byłby końcowym, wystarczyłoby przetestować istnienie wyłącznie jednej z tych cząsteczek. Jeżeli zaś oba stany uznawane są za końcowe, wystarczy istnienie tylko jednej spośród nich aby uznać, że dany automat akceptuje dane słowo wejściowe.

Omówiona pierwotna koncepcja automatu dwustanowego [2] była intensywnie rozwijana w kolejnych pracach. W roku 2003 zespół Benensona i Shapiro przedstawił ulepszoną wersję, w której między innymi zrezygnowano z użycia enzymu ligazy [3]. W roku 2004 grupa badawcza Olgierda Unolda z Politechniki Wrocławskiej rozszerzyła omawiany automat do trzech stanów, stosując inny enzym restrykcyjny. Uzyskali zatem automat trzystanowy, dwusymbolowy [49]. Również w 2004 Michał Soreni i inni przedstawili możliwość wykonania automatu trzystanowego i trzysymbolowego, którego działanie przetestowali także w laboratorium [45]. W roku 2008 Tadeusz Krasieński i Sebastian Sakowski pokazali rozszerzenie automatu Shapiro do sześciu stanów poprzez zastosowanie dwóch różnych enzymów restrykcyjnych w obrębie jednej mieszaniny. Zaprezentowany został automat sześciostanowy, dwusymbolowy [12]. W powstałej na podstawie tych badań rozprawie doktorskiej Sebastiana Sakowskiego została przedstawiona również teoretyczna koncepcja rozszerzenia liczby stanów do dziewięciu [40], a projekt jest dalej rozwijany, co łączy się z kolejnymi pracami, między innymi [15], [41], [42].

Oprócz omówionych implementacji automatów skończonych, zapoczątkowane przez Benensona podejście do koncepcji splatania było wykorzystywane w wielu innych modelach obliczeń DNA. Jedną z takich gałęzi są implementacje prostych systemów wnioskujących, przede wszystkim przedstawione przez zespoły Olgierda Unolda [47] i Ehuda Shapiro [27] (omawiane w rozdziale 3). Rozszerzona koncepcja systemu wnioskującego, będąca główną częścią tej pracy (wprowadzona w rozdziale 4) również wykorzystuje taką koncepcję splatania. Sedno reakcji stanowi naprzemienny ciąg ligacji i restryktacji, natomiast niektóre cząsteczki DNA powstałe w trakcie reakcji „są odrzucane” ze względu na brak komplementarnych „lepkich końców”, z którymi mogłyby ową reakcję kontynuować.

Jeszcze inna koncepcja wykorzystania splatania i pewnych enzymów restrykcyjnych dokonujących cięcia w dwie strony jednocześnie, została przedstawiona w pracy Matteo Cavaliere i jego zespołu badawczego w roku 2005. Pokazali oni ideę implementacji automatu ze stosem [6]. Pomysł ten był rozszerzany między innymi w [14].

## 2. 5. Pozostałe modele obliczeń

Do pozostałych teoretycznych modeli obliczeń za pomocą DNA, zaliczamy między innymi model oparty na „samoskładaniu” oraz model oparty na „membranach”, które nie mają konkretnego związku z nową koncepcją opisaną w tej pracy. Wspomniane pozostałe modele obliczeń zostały szczegółowo przedstawione między innymi w [11].

Ciekawą nowatorską technikę budowania komputerów DNA stanowi coraz bardziej popularny (i nie tylko teoretyczny) model DSD (z ang. *DNA strand displacement*), polegający na przemieszczeniu się nici DNA i zastąpieniu jej przez inną, bez użycia enzymów restrykcyjnych. Opiera się głównie na łączeniu dwóch komplementarnych nici i wykorzystuje proces zwany migracją rozgałęzień (przemieszczeniem rozgałęziających się nici, bądź zamianą miejscami fragmentów rozgałęzionych czy rozgałęzionego i nierozgałęzionego). Po raz pierwszy został wprowadzony w roku 2000 przez zespół badawczy Yurke [60], a od tego czasu był wielokrotnie wykorzystywany w różnych gałęziach obliczeń DNA [10]. Dla koncepcji systemu wnioskującego model DSD pojawia się między innymi w [28], [29], [30], [38], dla bramek logicznych między innymi w [24], [50], [51]. Można powiedzieć, że jest to najbardziej rozwijany w ostatnich latach model obliczeń DNA.



### 3. Znane implementacje systemów wnioskowania.

#### 3. 1. System Olgierda Unolda i Macieja Trocia

Jedną z głównych inspiracji, na bazie której powstała koncepcja rozszerzonego molekularnego systemu wnioskującego, przedstawiona została w pracy Olgierda Unolda i Macieja Trocia „Biomolekularne modele obliczeniowe” wydanej w 2005 roku na łamach czasopisma Inżynieria Komputerowa [47]. W pracy tej zawarto trzy różne modele obliczeń biomolekularnych: rozszerzenie automatu skończenie-stanowego omówionego pokrótce w podrozdziale 2.2, ideę obliczania wartości funkcji logicznych w postaci kanonicznej oraz prosty model wnioskowania. Ze względu na temat tej rozprawy szczególne omówienie wymaga ostatnia część pracy. Przedstawiony zostanie tutaj również model dotyczący funkcji logicznych ze względu na to, że zawiera unikalną koncepcję implementacji negacji, a ponadto zapoczątkowana w tej pracy idea stanowi również jedno z dodatkowych zastosowań nowego systemu wnioskującego. Omówienie jego zastosowania dla funkcji logicznych znajduje się w podrozdziale 5.2.1.

Wszystkie przedstawione modele opierają się na koncepcji systemu splatania, polegającego na naprzemiennym działaniu enzymu restrykcyjnego oraz ligazy. Wykorzystywane tutaj enzymy to *BseXI* oraz *FokI*. Taka koncepcja obliczeń została przedstawiona po raz pierwszy w [2], a w tej pracy została szczegółowo omówiona w podrozdziale 2.4.

##### 3. 1. 1. Model systemu wnioskującego

W tym punkcie omówiony zostanie prosty system wnioskujący, który na podstawie zgromadzonej wiedzy i pewnych logicznych reguł będzie w stanie otrzymywać wnioski, czyli dany zasób wiedzy poszerzać [47]. Występują tutaj dwa zasadnicze elementy: jednostki wiedzy zwane faktami (czyli przedstawione w podrozdziale 1.1 zmienne zdaniowe) oraz implikacje (w [47] nazywane też regułami wnioskowania). Dodatkowym elementem są zapytania czy dany fakt jest możliwy do wyprowadzenia przy podanych aksjomatach początkowych (zawierających odpowiednie fakty i implikacje). System dokonuje wnioskowania za pomocą reguły *modus ponens*.

Pojedyncze zmienne zdaniowe identyfikowane są za pomocą czteronukleotydowych, unikalnych, jednoniciowych łańcuchów DNA. Cząsteczki reprezentujące dany fakt (czyli zmienną zdaniową, o stałym przypisanym wartościowaniu 1, np. dla zmiennej zdaniowej  $p$ ,  $w(p)=1$ ) składają się z sekwencji rozpoznawalnej przez enzym restrykcyjny, „dystansownika” (składającego się z czterech dowolnych nukleotydów czyli NNNN) i wspomnianego „lepkiego końca” identyfikującego daną zmienną zdaniową. Między sobą różnią się wyłącznie tą ostatnią częścią. Przykładowe molekuly reprezentujące  $p$ ,  $q$  wyglądają następująco:



Ryc. 31a: schemat molekuly reprezentującej fakt  $p$

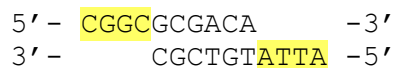


Ryc. 31b: schemat molekuly reprezentującej fakt  $q$

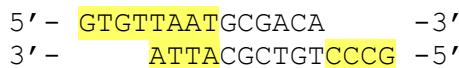
W powyższych przykładach „dystansownik” został zapisany w postaci 3'-CCCC-5' (używanej we wszystkich przykładach dotyczących omawianego systemu), jednak może on przyjmować dowolne wartości (3'-NNNN-5') i nie będzie to mieć wpływu na właściwy przebieg reakcji.

Aksjomaty w formie implikacji mogą zawierać również odpowiednio przedstawioną (i omówioną w dalszej części podrozdziału) koniunkcję i alternatywę znajdującą się w poprzedniku implikacji. Wynikiem reakcji wnioskowania będzie zawsze nowa molekula, reprezentująca pojedynczy fakt, który poszerza posiadany (przed rozpoczęciem reakcji) zasób wiedzy. W omawianym systemie nie ma możliwości wyprowadzenia negacji któregoś z faktów ani zapytania o możliwość jej wyprowadzenia. Nie ma też możliwości usunięcia już istniejącego lub wyprowadzonego faktu z aktualnej bazy wiedzy. Nowe wnioski są wyprowadzane z wykorzystaniem wnioskowania „w przód”, czyli w oparciu o aktualny zbiór faktów i regułę *modus ponens*, a niezależnie od znajdujących się w próbówce zapytań.

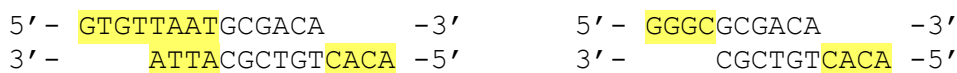
Pojedyncza implikacja reprezentowana jest przez jedną molekulę (w przypadku braku dodatkowych spójników z lewej strony implikacji lub warunków połączonych wyłącznie spójnikami „i” („^”) lub więcej molekuł gdy warunki połączone są spójnikami „lub” („v”). Wtedy musimy taką implikację „rozbić” na taką liczbę implikacji prostych ile składowych „podreguł” zawiera warunek. Przykładowe implikacje: prosta, niewykorzystująca pozostałych spójników ( $p \rightarrow q$ ), złożona z uwzględnieniem koniunkcji ( $(p \wedge q) \rightarrow r$ ) oraz koniunkcji i alternatywy ( $(p \wedge q) \vee r \rightarrow s$ ) (która musi zostać „rozbita” na dwie implikacje ( $(p \wedge q) \rightarrow s$ ) oraz ( $r \rightarrow s$ )), wyglądają następująco:



Ryc. 32a: schemat molekuly reprezentującej implikację ( $p \rightarrow q$ )



Ryc. 32b: schemat molekuly reprezentujących implikację ( $(p \wedge q) \rightarrow r$ )



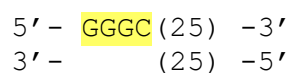
Ryc. 32c: schemat molekuly reprezentujących implikację ( $((p \wedge q) \vee r) \rightarrow s$ )

Istnieje też specjalna molekula uzupełniająca, identyczna dla wszystkich reakcji, która pozwala na przyłączenie się do molekuly wniosku i odpowiada za udzielenie odpowiedzi na zadane zapytanie. Wygląda ona następująco:



Ryc. 33: molekula uzupełniająca wnioskowania

Molekula reprezentująca zapytanie, czy dany fakt jest możliwy do wyprowadzenia na podstawie aksjomatów początkowych, składa się z czwórki identyfikującej zmienną zdaniową, o którą pytamy (komplementarnej do czwórki identyfikującej pojedynczy fakt) oraz stałej długości końca. Przykładowe zapytanie  $q?$  wygląda następująco:



Ryc. 34: schemat molekuly reprezentującej zapytanie  $q?$

Poniższy, szczegółowo omówiony przykład reakcji w tym systemie wnioskującym, zaczerpnięty jest całkowicie z omawianej pracy [47]. Weźmy następujące fakty:  $a$ ,  $d$ ,  $u$ , implikacje:  $((a \wedge b) \vee d) \rightarrow c$ ,  $((c \wedge u) \rightarrow x)$  oraz zapytanie  $x$ ?



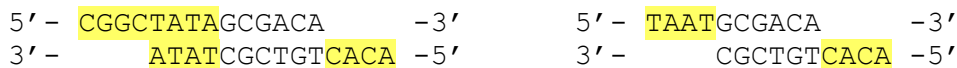
Ryc. 35a: schemat molekuly reprezentującej fakt  $a$



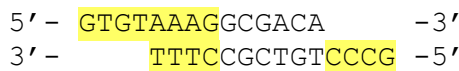
Ryc. 35b: schemat molekuly reprezentującej fakt  $d$



Ryc. 35c: schemat molekuly reprezentującej fakt  $u$



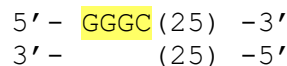
Ryc. 35d: schemat molekul reprezentujących implikację  $((a \wedge b) \vee d) \rightarrow c$



Ryc. 35e: schemat molekuly reprezentującej implikację  $(c \wedge u) \rightarrow x$

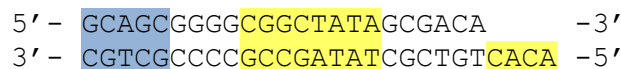


Ryc. 35f: molekula uzupełniająca wnioskowania



Ryc. 35g: molekula reprezentująca zapytanie  $x$ ?

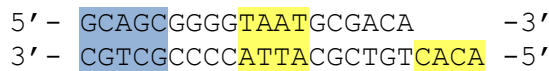
Reakcja będzie wyglądać następująco:



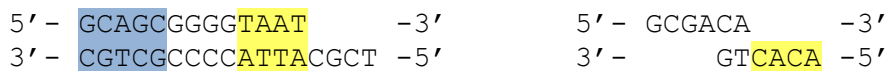
Ryc. 36a: ligacja molekuly reprezentujących  $a$  oraz  $((a \wedge b) \rightarrow c)$



Ryc. 36b: rozcięcie molekuly za pomocą enzymu restrykcyjnego *BseXI*, powstaje molekula  $(b \rightarrow c)$  (wnioskowanie z użyciem powstałej molekuly jednak nie może przebiegać dalej – brak faktu  $b$ )



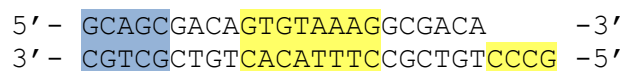
Ryc. 36c: ligacja molekuly reprezentujących  $d$  oraz  $(d \rightarrow c)$



Ryc. 36d: rozcięcie molekuly za pomocą enzymu restrykcyjnego *BseXI*



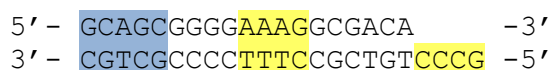
Ryc. 36e: ligacja wniosku i molekuly uzupełniającej, powstaje molekula faktu  $c$



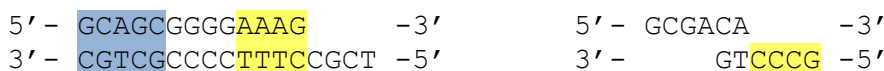
Ryc. 36f: ligacja molekul reprezentujących  $c$  oraz  $((c \wedge u) \rightarrow x)$



Ryc. 36g: rozcięcie molekuly za pomocą enzymu restrykcyjnego *BseXI*, powstaje molekula  $(u \rightarrow x)$



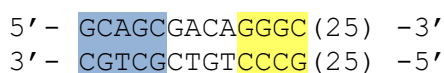
Ryc. 36h: ligacja molekuł reprezentujących  $u$  oraz ( $u \rightarrow x$ )



Ryc. 36i: rozcięcie molekuły za pomocą enzymu restrykcyjnego *BseXI*



Ryc. 36j: ligacja wniosku i molekuły uzupełniającej, powstaje molekuła faktu  $x$



Ryc. 36k: ligacja molekuł faktu  $x$  i zapytania  $x$ ?

Powstała cząsteczka wyjściowa stanowi odpowiedź na pytanie czy fakt  $x$  jest logiczną konsekwencją wprowadzonych danych wejściowych. Pojawienie się takiej cząsteczki o konkretnej długości 38 nukleotydów oznacza odpowiedź twierdzącą.

W ramach podsumowania należy zwrócić uwagę na niedoskonałości i wątpliwości związane z omówionym prostym systemem wnioskującym:

- Nie ma możliwości użycia spójnika negacji.
- Zgodnie z badaniami laboratoryjnymi, przeprowadzonymi między innymi w [27], cząsteczki o długości 6-ciu nukleotydów na każdej nici, połączone ze sobą jedynie dwoma nukleotydami (i o „lepkich końcach” długości 4-ech nukleotydów), z dużym prawdopodobieństwem rozpadną się na dwie molekuły jednoniciowe (*patrz*: ryciny 36d oraz 36i). Istnieje duże prawdopodobieństwo, że w dalszej reakcji połączą się one ze sobą ponownie, ale mogą też połączyć się na wiele innych sposobów, szczególnie w przypadku bardziej skomplikowanej reakcji.
- Molekuła finalna, na podstawie której następuje ostateczne wykrycie odpowiedzi systemu, powinna być dłuższa – w przypadku przedstawionym powyżej składa się ona z 38 nukleotydów co w trakcie rozpoznawania odpowiedzi łatwo może „pomylić się” z innymi molekułami uczestniczącymi w reakcji.
- Nie przedstawiono żadnych rozważań, dotyczących tego, które z sekwencji mogą być używane jako „lepkie końce” reprezentujące konkretne zmienne zdaniowe, a które nie, gdyż mogą „przypadkiem” utworzyć sekwencję rozpoznawalną przez enzym.
- Nie uwzględniono informacji, że użycie symboli komplementarnych do istniejącej czwórki reprezentującej dany fakt, może spowodować błędną reakcję.

Dlatego też pracę tę można uznać za inspirującą dla przedstawionych w dalszej części rozprawy rozważań, jednak z całkowitą pewnością nie można traktować tej konstrukcji jako w pełni działającego systemu wnioskowania logicznego za pomocą DNA.

### 3. 1. 2. Funkcje logiczne w postaci kanonicznej

Funkcją logiczną (lub inaczej funkcją boolowską) nazywamy dowolne odwzorowanie dowolnej wielkości zbioru zmiennych przyjmujących wartości 1 lub 0 (*prawdy* lub *fałszu*) w zbiór wartości również boolowskich 1 lub 0. Precyzyjnie, funkcją logiczną nazywamy dowolne odwzorowanie:

$$f: \{0, 1\}^n \rightarrow \{0, 1\}$$

dla pewnego  $n \geq 1$ . Przykładami takich funkcji są oczywiście funkcje zadane zdaniami złożonymi. Wtedy  $n$  jest równe liczbie zdań atomowych występujących w tym zdaniu. Na przykład zdanie złożone  $((p \wedge q \wedge r) \vee \neg s)$  reprezentuje funkcję logiczną 4 zmiennych, która np. dla wartości  $w(p) = 1$ ,  $w(q) = 1$ ,  $w(r) = 0$ ,  $w(s) = 0$  przyjmuje wartość 1.

Dokładniejszy opis funkcji logicznych znajduje się też w podrozdziale 5.2.1.

Aby rozpocząć obliczenia, niezbędny jest wzór funkcji zapisany w postaci kanonicznej czyli w formie sumy iloczynów (czyli dysjunkcyjnej postaci normalnej, *patrz*: podrozdział 1.3). Występują tutaj trzy klasy molekuł: pierwsza to argumenty podawane do funkcji, druga to zakodowane iloczyny cząstkowe, definiujące właściwą funkcję logiczną, trzecia zaś to zbiór odpowiednich detektorów, umożliwiających odczyt wyniku obliczeń. W omawianej implementacji, w przeciwieństwie do pozostałych przedstawionych tu systemów splatania, nie istnieje pojęcie wnioskowania i poszerzania zbioru wiedzy w trakcie obliczeń. System wyłącznie sprawdza wartość funkcji logicznej dla podanych argumentów (o konkretnym wartościowaniu 0 lub 1). Został on przedstawiony dlatego, że koncepcja obliczania wartości funkcji logicznych jest również jednym z dodatkowych zastosowań systemu wnioskującego będącego głównym elementem tej pracy (*patrz*: podrozdział 5.2.1).

Cząsteczki reprezentujące argumenty składają się z trzech elementów: sekwencji rozpoznawalnej przez enzym restrykcyjny, „dystansownika” (który jednocześnie koduje informację o wartościowaniu, które przyjmuje dany argument) oraz „lepkiego”, czteronukleotydowego końca identyfikującego argument na liście wejść danej funkcji (czyli analogicznie jak w poprzednim podrozdziale, identyfikującego daną zmienną zdaniową). Jednym z omówionych założeń jest brak możliwości występowania nukleotydu C jako pierwszego i czwartego elementu „lepkiego końca”. „Dystansownik” przyjmuje zawsze wartości C na nici 3'-5' oraz długość równą 3 dla argumentów o wartości 1, oraz długość 4 dla wartości 0 (zapisywanych zwyczajowo z wykorzystaniem spójnika negacji). Przykładowe cząsteczki reprezentujące argumenty  $a$ ,  $\neg a$ ,  $b$ ,  $\neg b$  wyglądają następująco:



*Ryc. 37a: schemat molekuł reprezentujących argumenty a oraz  $\neg a$*



*Ryc. 37b: schemat molekuł reprezentujących argumenty b oraz  $\neg b$*

Sama funkcja zakodowana jest w taki sposób, że każdej koniunkcji składowej odpowiada jedna molekula, zbudowana z ułożonych szeregowo odcinków DNA odpowiadających poszczególnym czynnikom. Fragmenty te odseparowane są w taki sposób, że o wartości argumentu decyduje długość „dystansownika” (każdy składnik reprezentujący wartość 1 posiada „dystansownik” o długości dwóch nukleotydów C na nici 3'-5', natomiast dla wartości 0 jest to tylko jeden nukleotyd C na nici 3'-5'). Każda taka cząsteczka zakończona jest też sekwencją terminalną 3'-AAAA-5', która sygnalizuje zakończenie obliczeń danej funkcji. Po sekwencji terminalnej umieszczony jest również pojedynczy nukleotyd C na nici 3'-5') oraz 100 dowolnych nukleotydów. Pierwsza czwórka reprezentująca pojedynczy

czynnik przedstawiona jest w formie „lepkiego końca” (oczywiście komplementarnego do „lepkiego końca” argumentów funkcji) podczas gdy wszystkie kolejne już w postaci dwuniciowej. Przykładowe składowe funkcji zawierające wyłącznie jeden argument  $a$  (o wartościowaniu 1) oraz  $\neg b$  (czyli  $b$  o wartościowaniu 0) wyglądają następująco:

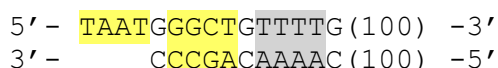


Ryc. 38a: składowa funkcji logicznej o jednym argumencie  $a$



Ryc. 38b: składowa funkcji logicznej o jednym argumencie  $\neg b$

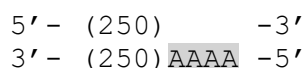
Przykładowy iloczyn składowy postaci  $\neg a \wedge \neg b$  wygląda więc następująco:



Ryc. 39: koniunkcja składowa funkcji logicznej o dwóch argumentach ( $\neg a \wedge \neg b$ )

Koniunkcje składowe o większej liczbie czynników tworzymy w analogiczny sposób, dopisując reprezentujące je czwórki nukleotydów oraz odpowiednie „dystansowniki”, tym samym dalej wydłużając molekułę.

Istnieje pojedyncza molekuła detektorowa (identyczna dla wszystkich reakcji), która wykrywa pojawienie się wartości 1 na wyjściu zadanej funkcji logicznej. Kluczowy dla obliczeń jest fakt, że tak zapisana funkcja logiczna przyjmuje wartość 1 jeżeli którykolwiek z jej iloczynów cząstkowych przyjmuje tę wartość. Poszczególne iloczyny mogą być przetwarzane równoległe (co jest jednym z ważniejszych założeń obliczeń za pomocą DNA). Częstka detektora wygląda następująco:



Ryc. 40: molekuła detektora dla wyjścia funkcji

Weźmy przykład wykorzystania tego modelu dla dwuargumentowej funkcji XOR (alternatywy wykluczającej), którą możemy przedstawić w postaci  $(a \wedge \neg b) \vee (\neg a \wedge b)$ . Przykład zaczerpnięty jest całkowicie z omawianej pracy [47]. Niezbędne molekuły wyglądają następująco:

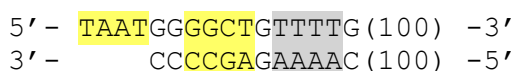


Ryc. 41a: schemat molekuł reprezentujących  $a$ : po lewej  $a$ , po prawej  $\neg a$

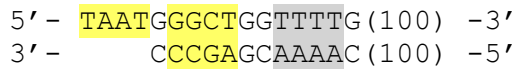


Ryc. 41b: schemat molekuł reprezentujących  $b$ : po lewej  $b$ , po prawej  $\neg b$

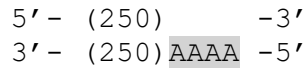
Z powyższych musimy wybrać argumenty, które podajemy na wejściu funkcji logicznej – nie możemy podać jednocześnie dwóch różnych wartości tej samej zmiennej zdaniowej.



Ryc. 42a: koniunkcja składowa funkcji logicznej ( $a \wedge \neg b$ )

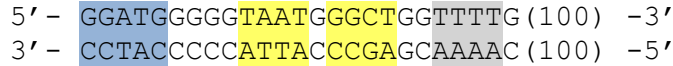


Ryc. 42b: koniunkcja składowa funkcji logicznej ( $\neg a \wedge b$ )



Ryc. 42c: molekula detektora dla wyjścia funkcji

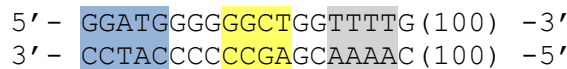
Dla argumentów  $\neg a$  i  $b$ , tzn.  $w(a) = 0$ ,  $w(b) = 1$ , reakcja wygląda następująco:



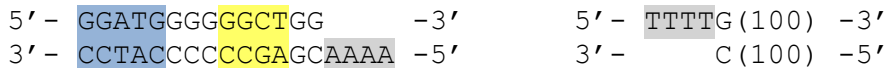
Ryc. 43a: ligacja molekuly  $\neg a$  oraz molekuly iloczynu ( $\neg a \wedge b$ )



Ryc. 43b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI



Ryc. 43c: ligacja molekuly  $b$  oraz pozostałego fragmentu molekuly koniunkcji składowej



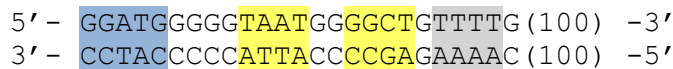
Ryc. 43d: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI



Ryc. 43e: ligacja pozostałego fragmentu i molekuly terminalnej

Tak powstała cząsteczka wyjściowa, bez „lepkich końców” oraz o konkretnej długości 355 nukleotydów oznacza odpowiedź twierdzącą, co znaczy, że wartość funkcji XOR dla podanych argumentów jest prawdziwa.

Należy zwrócić uwagę, że w tym samym czasie zajdą reakcje pozostałych molekuly:



Ryc. 44a: ligacja molekuly  $\neg a$  oraz molekuly koniunkcji ( $a \wedge \neg b$ )



Ryc. 44b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI

Dla tak powstałej cząsteczki pośredniej nie istnieje inna posiadająca odpowiedni „leпки koniec”. Reakcja w tym przypadku zostaje zatrzymana.

Omawiany system sprawdzania wartości funkcji logicznej dla podanych argumentów jest opracowany właściwie – umożliwia przetwarzanie wszystkich koniunkcji cząstkowych jednocześnie i zależnie od wartości argumentów prowadzi (lub nie) do powstania molekuly sygnalizującej odpowiedź twierdzącą. Wystarczy, że co najmniej jedna z przetwarzanych koniunkcji składowych jest spełniona, a wtedy oznacza to spełnienie danej funkcji dla wszystkich podanych argumentów. Za wadę możemy uznać konieczność użycia różnych próbek dla różnych argumentów tej samej funkcji. Ponadto należy jeszcze raz podkreślić, że nie zawiera on żadnej z możliwości wnioskowania na podstawie argumentów a wyłącznie sprawdzenie wartości zwracanej przez funkcję logiczną dla zadanych wartości argumentów wejściowych.

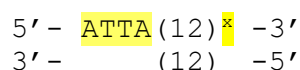
## 3. 2. System Shai Kaplana, Toma Rana i Ehuda Shapiro

Drugą, bardzo ważną inspiracją dla przedstawionych w tej rozprawie badań jest system wnioskowania przedstawiony w pracy Ehuda Shapiro, Shai Kaplana i Toma Rana „Molecular implementation of simple logic program”, wydanej w 2009 przez Nature Nanotechnologies [27]. W pracy tej zawarto dwie koncepcje systemu wnioskującego, różniące się od siebie kolejnością przeprowadzania obliczeń. Obie opierają się na modelu splatania, czyli naprzemiennych łączeniach i cięciach molekuł (*patrz*: podrozdział 2.4). Do rozpoznania odpowiedzi wykorzystano fluorescencję – umieszczenie specjalnego fragmentu na końcu molekuly powoduje pojawienie się świecenia w efekcie rozpadu danego wiązania na dwa fragmenty jednoniciowe. Badania zostały poparte doświadczeniami laboratoryjnymi.

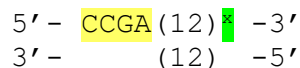
### 3. 2. 1. System wnioskujący z identyfikacją przesłanek.

Przedstawiony w pierwszej części pracy [27] system wnioskowania logicznego pozwala uzyskać informację czy na podstawie posiadanych przesłanek (aksjomatów wejściowych, których prawdziwość przyjmujemy) możliwe jest wyprowadzenie wniosku o który pytamy. Występują tutaj trzy elementy, reprezentowane przez trzy różne typu molekuł: jednostki wiedzy zwane faktami (czyli przedstawione w podrozdziale 1.1 zmienne zdaniowe), implikacje z możliwością użycia alternatywy w poprzedniku oraz zapytania.

Pojedyncze zmienne zdaniowe, analogicznie jak w przedstawionej wcześniej pracy Unolda i Trocia, identyfikowane są za pomocą unikalnych jednoniciowych, czteronukleotyдовых łańcuchów DNA. Cząsteczki faktów składają się ze wspomnianej unikalnej czwórki w formie „lepkiego końca”, odwołującego się do konkretnej zmiennej zdaniowej oraz dwunastu dowolnych nukleotydów, zakończonych fragmentem fluorescencyjnym (innego koloru dla każdego z faktów). Nie ma możliwości zakodowania negacji. Przykładowe cząsteczki reprezentujące literały *a* i *b* wyglądają następująco (gdzie kolorowy symbol w prawym górnym rogu oznacza umieszczoną, ale nieaktywną fluorescencję – świecenie nastąpi dopiero w wyniku rozpadu cząsteczki dwuniciowej na dwie jednoniciowe jako efekt końcowy przeprowadzonej reakcji):



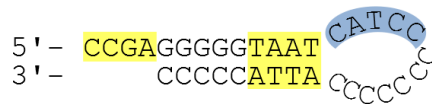
Ryc. 45a: schemat molekuly reprezentującej fakt *a*



Ryc. 45b: schemat molekuly reprezentującej fakt *b*

Molekuły implikacji zbudowane są z następujących elementów: czwórki identyfikującej następnik (w formie „lepkiego końca”), „dystansownika” składającego się z pięciu nukleotydów G/C, czwórki identyfikującej poprzednik (jednak zapisanej na przeciwnej nici, co oznacza zmianę kierunku nukleotydów) i kolistego fragmentu, składającego się z 12 nukleotydów, zawierających 5-cio nukleotydivy fragment rozpoznawalny przez enzym restrykcyjny *FokI* oraz 7 nukleotydów C. Przykładowa implikacja *a* → *b* wygląda następująco:





Ryc. 46: schemat molekuly implikacji ( $a \rightarrow b$ )

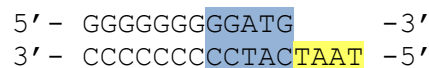
Implikacje zawierające alternatywę w poprzedniku reprezentujemy za pomocą kilku cząsteczek np. dla  $a \vee b \rightarrow c$  odpowiednikiem będzie  $a \rightarrow c$  oraz  $b \rightarrow c$ . Analogicznie będzie dla większej liczby składników. W omawianym systemie nie ma możliwości umieszczenia koniunkcji w poprzedniku implikacji.

Istnieje też specjalna jednoniciowa molekula uzupełniająca, identyczna dla wszystkich reakcji, która w trakcie przyłącza się do molekuly wniosku częściowego i tworzy molekule reprezentującą nowe zapytanie. Wygląda ona następująco.

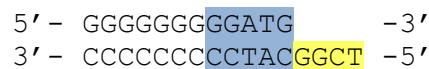


Ryc. 47: molekula uzupełniająca wnioskowania

Molekuly zapytań składają się z trzech elementów: „dystansownika” zawierającego siedem nukleotydów G/C, sekwencji rozpoznawalnej przez enzym restrykcyjny *FokI* oraz umieszczonej w formie „lepkiego końca” czwórki jednoznacznie identyfikującej fakt, o który pytamy. Przykładowe zapytania  $a?$  oraz  $b?$  wyglądają następująco:

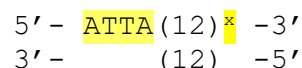


Ryc. 48a: schemat molekuly reprezentującej fakt a

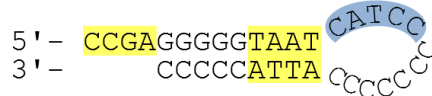


Ryc. 48b: schemat molekuly reprezentującej fakt b

W omawianej pracy przykłady przedstawione są w formie językowej np. „Sokrates jest Grekiem”, „jeżeli Sokrates jest Grekiem to jest człowiekiem”, „czy Sokrates jest człowiekiem?” itd. Do postaci klasycznego rachunku zdań przykłady te zostały już sprowadzone we wcześniejszych pracach [31], [34], a poniższy przykład zaczerpnięty jest z [34]. Weźmy następujące molekuly:  $a$ ,  $a \rightarrow b$ ,  $b?$  oraz molekule uzupełniająca:



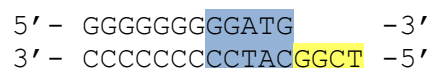
Ryc. 49a: schemat molekuly reprezentującej fakt a



Ryc. 49b: schemat molekuly reprezentującej implikację ( $a \rightarrow b$ )

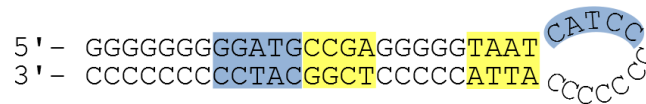


Ryc. 49c: molekula uzupełniająca wnioskowania

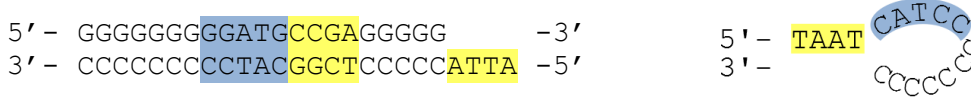


Ryc. 49d: schemat molekuly reprezentującej zapytanie b?

Reakcja będzie wyglądać następująco:



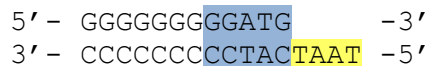
Ryc. 50a: ligacja molekuly zapytania b? i implikacji (a → b)



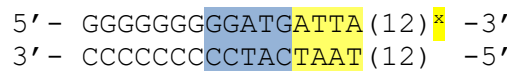
Ryc. 50b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI



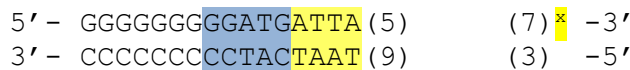
Ryc. 50c: rozwinięcie wniosku częściowego do postaci jednoniciowej



Ryc. 50d: ligacja powstałej cząstki i molekuly uzupełniającej wnioskowania (w wyniku redukcji zapytania b? i implikacji, powstaje molekula reprezentująca zapytanie a?)



Ryc. 50e: ligacja molekul argumentu a i zapytania a?



Ryc. 50f: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI



Ryc. 50g: rozpad powstałej krótkiej cząstki na dwie jednoniciowe oraz uaktywnienie fluorescencji

Uaktywnienie fluorescencji następuje w momencie rozpadu cząsteczki dwuniciowej na dwie jednoniciowe – zgodnie z doświadczeniami laboratoryjnymi, przeprowadzonymi przez autorów, tak krótka molekula nie ma szans pozostać w postaci dwuniciowej. Samo pojawienie się świecenia informuje o odpowiedzi twierdzącej systemu. Kolor światła informuje, że fakt **b** (którego dotyczyło zapytanie) został wyprowadzony z przesłanki bazowej **a**, zawierającej żółtą substancję fluorescencyjną. Możliwość sprawdzenia na podstawie których przesłanek powstał wniosek odróżnia tę koncepcję od innych.

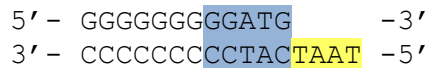
Omówiony sposób wnioskowania nazywamy wnioskowaniem „w tył”, a jego kolejność działań jest przeciwna do występującego we wszystkich pozostałych omawianych koncepcjach wnioskowania „w przód”. Podczas reakcji w probówce przetwarzane są wyłącznie zapytania oraz implikacje. Nie powstają tu w ogóle nowe molekuly faktów, otrzymane przy użyciu reguły *modus ponens*, co również ogranicza funkcjonalność systemu.

### 3. 2. 2. Klasyczny system wnioskujący

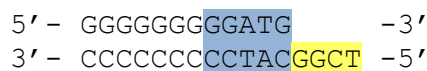
W tej samej pracy [27] przedstawiono również koncepcję wnioskowania „w przód” czyli dodawania nowych faktów do bazy wiedzy na podstawie faktów już istniejących zgodnie z regułą *modus ponens*. Koncepcja ta bardziej przypomina klasyczny sposób wnioskowania, naturalny dla naszego myślenia. Nie istnieje możliwość identyfikacji, z której przesłanki wynika pozytywna odpowiedź systemu, za to pojawia się możliwość zastosowania koniunkcji w poprzedniku implikacji a także użycie kilku różnych zapytań jednocześnie (z rozróżnieniem odpowiedzi przez zastosowane różne kolory fluorescencji). Występują tutaj trzy elementy, reprezentowane przez trzy różne typu molekul: jednostki wiedzy zwane faktami, implikacje

(z możliwością zapisania koniunkcji w poprzedniku implikacji a także alternatywy za pomocą kilku molekuł) oraz zapytania (z dołączonym fragmentem fluorescencyjnym).

Zmienne zdaniowe tu również identyfikowane są za pomocą unikalnych jednoniciowych, czteronukleotydowych łańcuchów DNA. Częsteczki faktów składają się z trzech elementów: „dystansownika” zawierającego siedem nukleotydów G/C, sekwencji rozpoznawalnej przez enzym restrykcyjny *FokI* oraz umieszczonej w formie „lepkiego końca” czwórki jednoznacznie identyfikującej daną zmienną zdaniową. Nie istnieje możliwość zastosowania negacji faktów oraz związanych z nią praw klasycznego rachunku zdań. Przykładowe molekuly reprezentujące fakty **a** oraz **b** wyglądają następująco:

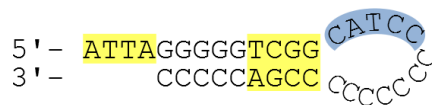


Ryc. 51a: schemat molekuly reprezentującej zapytanie a?



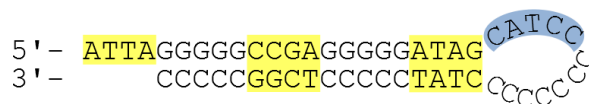
Ryc. 51b: schemat molekuly reprezentującej zapytanie b?

Implikacje zbudowane są z następujących elementów: czwórki identyfikującej poprzednik (w formie „lepkiego końca”), „dystansownika” składającego się z pięciu nukleotydów G/C, czwórki identyfikującej wniosek (jednak zapisanej na przeciwnej nici, co oznacza zmianę kierunku nukleotydów) i kolistego fragmentu, składającego się z 12 nukleotydów, zawierających 5-cio nukleotydową sekwencję rozpoznawalną przez enzym restrykcyjny *FokI* oraz 7 nukleotydów C. Przykładowa reguła wnioskowania ( $a \rightarrow b$ ) wygląda następująco:



Ryc. 52: schemat molekuly implikacji ( $a \rightarrow b$ )

Koniunkcję w poprzedniku implikacji uzyskujemy poprzez dodanie kolejnych nukleotydów z przodu molekuly – czwórki identyfikującej kolejny czynnik oraz kolejnego „dystansownika” składającego się z pięciu nukleotydów G/C (powtórzone tyle razy ile jest dodawanych czynników). Tylko pierwsze cztery nukleotydy (reprezentujące pierwszy składnik koniunkcji) przedstawione są w formie „lepkiego końca”, pozostałe zaś w formie dwuniciowej. Przykładowa reguła wnioskowania ( $a \wedge b \rightarrow c$ ) wygląda następująco:



Ryc. 53: schemat molekuly implikacji ( $a \wedge b \rightarrow c$ )

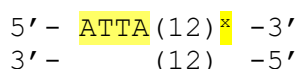
Implikacje składające się z alternatyw w poprzedniku reprezentujemy za pomocą kilku częsteczek (tak samo jak w poprzednim modelu), np. dla  $a \vee b \rightarrow c$  odpowiednikiem będzie  $a \rightarrow c$  oraz  $b \rightarrow c$ . Analogicznie dla większej liczby składników.

Istnieje też specjalna jednoniciowa molekula uzupełniająca, identyczna dla wszystkich reakcji, która w trakcie przyłącza się do molekuly wniosku częściowego i formuluje nową molekule reprezentującą fakt, oderwany na mocy reguły *modus ponens*. Wygląda ona następująco.

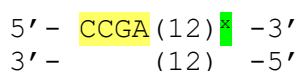


Ryc. 54: molekula uzupełniająca wnioskowania

Molekuły zapytań składają się z unikalnej, identyfikującej zdanie atomowe, czwórki nukleotydów (w formie „lepkiego końca”) oraz dwunastu dowolnych nukleotydów, zakończonych fragmentem fluorescencyjnym (innego koloru dla każdego zapytania). Przykładowe cząsteczki *a?* i *b?* wyglądają następująco (gdzie kolorowy symbol w prawym górnym rogu oznacza umieszczoną, ale nieaktywną fluorescencję – świecenie może nastąpić w wyniku rozpadu cząsteczki dwuniciowej na dwie jednoniciowe na skutek przeprowadzonej reakcji):

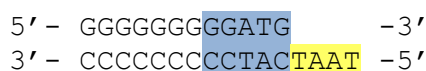


Ryc. 55a: schemat molekuly zapytania a?

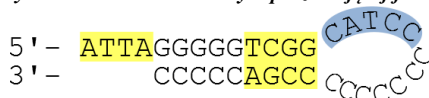


Ryc. 55b: schemat molekuly zapytania b?

Poniższy prosty przykład zaczerpnięty jest z [31], gdzie praca grupy badawczej Shapiro została szczegółowo omówiona. Weźmy następujące molekuly: *a*, *a* → *b*, *b?* oraz molekule uzupełniającą:



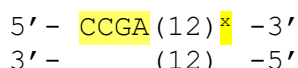
Ryc. 56a: schemat molekuly reprezentującej fakt a



Ryc. 56b: schemat molekuly reprezentującej implikację (a → b)

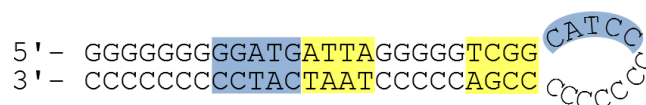


Ryc. 56c: molekula uzupełniająca wnioskowania

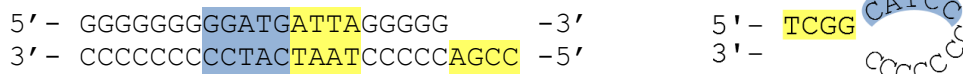


Ryc. 56d: schemat molekuly reprezentującej zapytanie b?

Reakcja będzie wyglądać następująco:



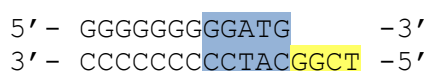
Ryc. 57a: ligacja molekuly faktu a oraz implikacji (a → b)



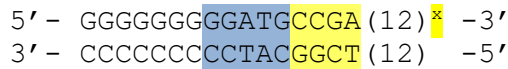
Ryc. 57b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI



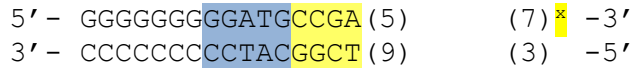
Ryc. 57c: rozwinięcie wniosku częściowego do postaci jednoniciowej



Ryc. 57d: ligacja powstałej cząstki i molekuly uzupełniającej wnioskowania (w wyniku redukcji argumentu a oraz implikacji, powstaje molekula reprezentująca fakt b)



Ryc. 57e: ligacja molekuł faktu b i zapytania b?



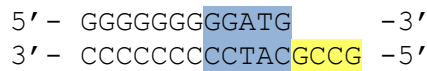
Ryc. 57f: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI



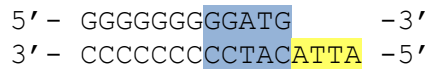
Ryc. 57g: rozpad powstałej cząstki na dwie jednoniciowe oraz uaktywnienie fluorescencji

Uaktywnienie fluorescencji następuje w momencie rozpadu tak skonstruowanej cząsteczki na dwie jednoniciowe – zgodnie z doświadczeniami laboratoryjnymi, przeprowadzonymi przez autorów, tak krótka molekula nie ma szans pozostać w postaci dwuniciowej. Pojawienie się świecenia informuje o odpowiedzi twierdzącej systemu. Kolor światła pozwala odróżnić na które z zapytań uzyskaliśmy odpowiedź twierdzącą – technika umieszczenia fragmentów fluorescencyjnych na końcu molekuł zapytań umożliwia przetwarzanie większej liczby zapytań podczas jednej reakcji w tej samej próbówce.

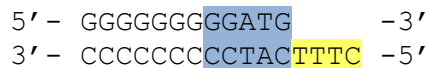
Na potrzeby porównania omawianych systemów (teoretycznie funkcjonalność przedstawiona przez zespół Ehuda Shapiro [27] jak i zespół Olgerda Unolda [47] jest taka sama), przeanalizujemy bardziej skomplikowany przykład przedstawiony w [47] i omówiony w podrozdziale 3.1.1, ale zapisując go w wersji odpowiedniej dla prezentowanego tutaj systemu wnioskującego. Weźmy następujące fakty: **a**, **d**, **u**, implikacje  $((a \wedge b) \vee d) \rightarrow c$ ,  $((c \wedge u) \rightarrow x)$  oraz zapytanie **x**:



Ryc. 58a: schemat molekuly reprezentującej fakt a



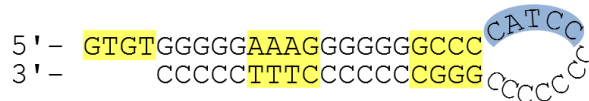
Ryc. 58b: schemat molekuly reprezentującej fakt d



Ryc. 58c: schemat molekuly reprezentującej fakt u



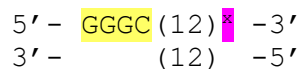
Ryc. 58d: schemat molekuł reprezentujących wnioskowanie  $((a \wedge b) \vee d) \rightarrow c$



Ryc. 58e: schemat molekuł reprezentujących wnioskowanie  $((c \wedge u) \rightarrow x)$

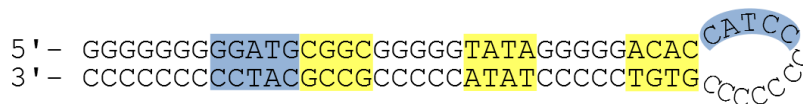


Ryc. 58f: molekula uzupełniająca wnioskowania

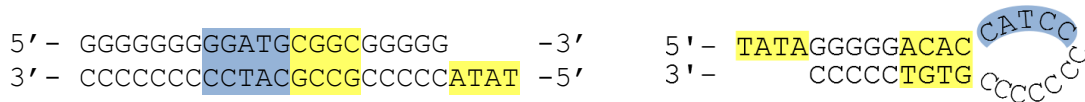


Ryc. 58g: molekuly reprezentująca zapytanie x?

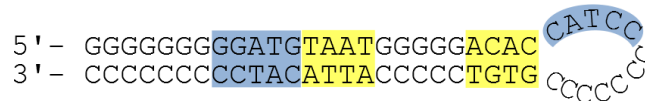
Reakcja będzie wyglądać następująco:



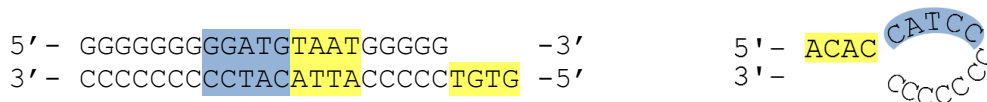
Ryc. 59a: ligacja molekuly reprezentujacych a oraz ((a ∧ b) → c)



Ryc. 59b: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI, powstaje molekula (b → c)  
 (wnioskowanie z użyciem powstałej molekuly jednak nie może przebiegać dalej – brak informacji o wartosci b)



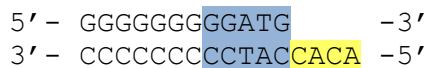
Ryc. 59c: ligacja molekuly reprezentujacych d oraz (d → c)



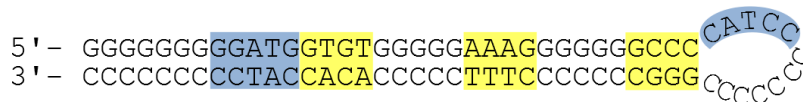
Ryc. 59d: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI



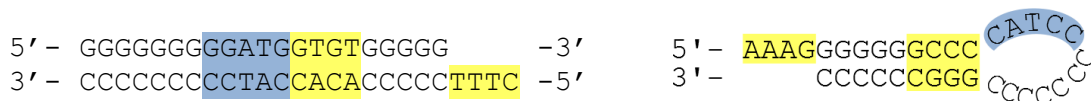
Ryc. 59e: rozwinięcie wniosku częściowego do postaci jednoniciowej



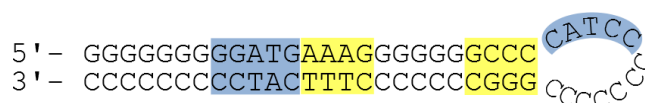
Ryc. 59f: ligacja wniosku i molekuly uzupełniającej, powstaje molekula faktu c



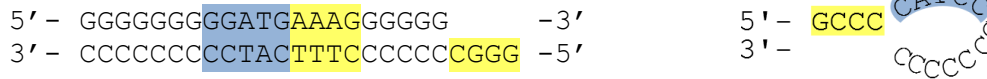
Ryc. 59g: ligacja molekuly reprezentujacych c oraz ((c ∧ u) → x)



Ryc. 59h: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI, powstaje molekula (u → x)



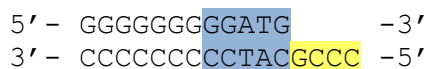
Ryc. 59i: ligacja molekuly reprezentujacych u oraz (u → x)



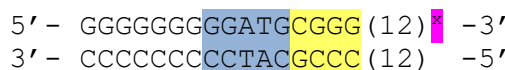
Ryc. 59j: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI



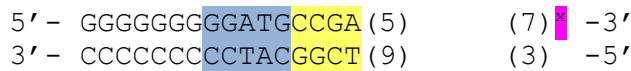
Ryc. 59k: rozwinięcie wniosku częściowego do postaci jednoniciowej



Ryc. 59l: ligacja wniosku i molekuly uzupełniającej, powstaje molekula faktu x



Ryc. 59m: ligacja molekuly faktu b i zapytania b?



Ryc. 59n: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI



Ryc. 59o: rozpad powstałej cząstki na dwie jednoniciowe oraz uaktywnienie fluorescencji

Pojawienie się świecenia w zadanym przez zapytanie  $x?$  kolorze oznacza pozytywną odpowiedź na pytanie czy fakt  $x$  jest logiczną konsekwencją wprowadzonych danych wejściowych.

W ramach podsumowania i porównania omawianych systemów, należy zwrócić uwagę na następujące rzeczy:

- W przeciwieństwie do systemu wnioskującego Unolda i Trocia, omówionego w podrozdziale 3.1.1. [47], koncepcja Shapiro zabezpiecza się przed możliwością błędnej reakcji od strony wniosku – nukleotydy reprezentujące następnik implikacji zamknięte są we fragmencie kolistym a nie „lepkim końcu”, więc nie ma możliwości „przypadkowego” połączenia się ich w nieplanowany sposób.
- Brakuje informacji na temat doboru nukleotydów reprezentujących konkretne „lepkie końce” zdań atomowych – tu również łańcuchy komplementarne do używanych powinny zostać pominięte (w tym łańcuchy komplementarne z samym sobą) ze względu na możliwość niewłaściwej reakcji między cząsteczkami faktów. Ogranicza to liczbę faktów możliwych do umieszczenia w zbiorze aksjomatów początkowych.
- Samo wnioskowanie jest wnioskowaniem prostym, nieuwzględniającym spójnika negacji ani żadnych związanych z nim aksjomatów. Zatem stanowi ono wyłącznie niewielki wycinek klasycznego rachunku zdań i tylko w niewielkim stopniu odwzorowuje zasady rozumowania matematycznego (w tym pośrednio rozumowania ludzkiego).

### 3. 3. Systemy grupy badawczej Jana Mulawki

W przedstawionym przeglądzie prac związanych z wnioskowaniem logicznym za pomocą DNA, do bardzo istotnych należy również szereg koncepcji opracowanych przez zespół badawczy Politechniki Warszawskiej, prowadzony głównie przez prof. dr hab. inż. Jana Mulawkę. Pierwsze prace pojawiły się już w 1997 roku. Dwie z przedstawionych koncepcji biokomputera (opracowane w zespole: Jan Mulawka, Piotr Węgleński, Piotr Borsuk), zostały nawet zgłoszone do opisów patentowych [21], [22]. Pierwsza z nich przedstawia sposób dowodzenia dedukcyjnego za pomocą wnioskowania regresywnego [21] (*patrz:* podrozdział 3.3.1), druga zaś sposób wnioskowania dedukcyjnego z wykorzystaniem drzewa decyzyjnego [22], poniżej przedstawiony w wersji ograniczonej do dwóch możliwych do podjęcia wyborów (*prawdy* lub *falszu*, co odpowiada logice klasycznej) (*patrz:* podrozdział 3.3.2). Kolejna bardzo istotna koncepcja, w której uwzględniono regułę przechodniości implikacji, przedstawiona została w roku 1999 podczas kongresu w Waszyngtonie [54], a następnie szczegółowo omówiona w 2000 roku na łamach międzynarodowego czasopisma „Cybernetics and Systems” (opracowana głównie w zespole: Jan Mulawka, Piotr Wąsiewicz, Tomasz Janczak) [56] (*patrz:* podrozdział 3.3.3). Z technicznego punktu widzenia, wszystkie z nich opierają się na systemie filtrowania (*patrz:* podrozdział 2.3), co oznacza że podczas reakcji

wykorzystują wyłącznie ligację znajdujących się w próbówce molekuł, a do wykrycia odpowiedzi reakcję łańcuchową polimerazy oraz elektroforezę w żelu. Wszystkie z powyższych koncepcji zostały poparte badaniami laboratoryjnymi, potwierdzającymi poprawność reakcji.

### 3. 3. 1. Dowodzenie za pomocą wnioskowania regresywnego

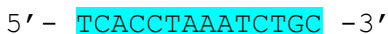
Koncepcja ta zakłada istnienie trzech różnych składowych systemu: faktów, implikacji i konkluzji (której prawdziwość należy sprawdzić, analogicznie jak w przypadku zapytań o pojedynczy fakt w innych systemach wnioskujących). Celem jest sprawdzenie czy konkluzja jest faktycznie możliwa do wyprowadzenia na podstawie pozostałych aksjomatów początkowych. W przypadku odpowiedzi twierdzącej, w próbówce pojawi się molekula nieposiadająca „lepkich końców”, odpowiadająca konkretnej ścieżce dowodzenia od bazy faktów i implikacji do oczekiwanej konkluzji. Każdy element reprezentowany jest za pomocą odpowiedniej cząsteczki jednoniciowej. Z początku tworzone są dwa roztwory: osobny dla bazy faktów i konkluzji, osobny dla bazy reguł wnioskowania. Oba dobierane są w sposób, gwarantujący niereaktywność w ramach pojedynczego roztworu. Molekuły reprezentujące reguły wnioskowania składają się z odcinków odpowiednio komplementarnych do odpowiednich faktów, konkluzji czy wniosków pozostałych reguł. Unikalne jest, że względem wszystkich pozostałych omówionych dotąd systemów, długości cząsteczek są całkowicie dowolne i mogą różnić się między sobą (różne fakty reprezentowane różnymi długościami itd.) tak długo jak zasada odpowiednich komplementarności zostaje spełniona. Nie ma możliwości wykorzystania negacji zmiennych. Molekuła reprezentująca konkluzję ma identyczną postać jakby była molekułą reprezentującą pojedynczy fakt. Dla lepszego zrozumienia przyjrzyjmy się przykładowej reakcji przedstawionej w [21]. Weźmy bazę faktów:  $a$ ,  $b$ ,  $c$ , reguł wnioskowania:  $((c \wedge a) \rightarrow k)$ ,  $((k \wedge b) \rightarrow l)$ ,  $(l \rightarrow m)$  oraz konkluzję (hipotezę, której możliwość wyprowadzenia sprawdzamy)  $m$ . Przykładowe (zaprezentowane w [21]) kodowanie molekuł wygląda następująco:



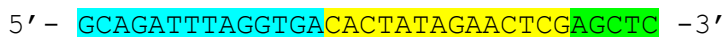
Ryc. 60a: schemat molekuly reprezentującej fakt  $a$



Ryc. 60b: schemat molekuly reprezentującej fakt  $b$



Ryc. 60c: schemat molekuly reprezentującej fakt  $c$



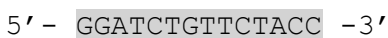
Ryc. 60d: schemat molekuly reprezentującej regułę wnioskowania  $(c \wedge a) \rightarrow k$



Ryc. 60e: schemat molekuly reprezentującej regułę wnioskowania  $(k \wedge b) \rightarrow l$  (tu zapisanej od prawej do lewej)



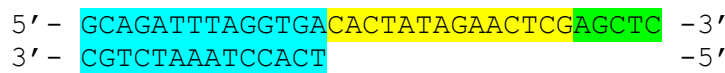
Ryc. 60f: schemat molekuly reprezentującej regułę wnioskowania  $l \rightarrow m$



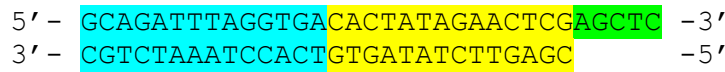
Ryc. 60g: schemat molekuly reprezentującej hipotezę  $m$



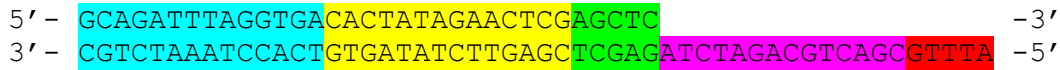
Reakcja wygląda następująco:



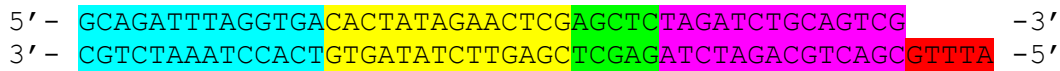
Ryc. 61a: ligacja molekuł reprezentujących  $(c \wedge a) \rightarrow k$  oraz  $c$



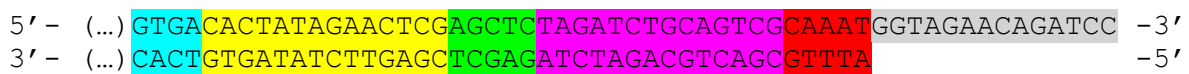
Ryc. 61b: ligacja powyższej molekuły oraz molekuły faktu  $a$



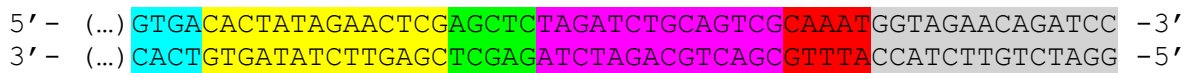
Ryc. 61c: ligacja powyższej molekuły oraz molekuły wnioskowania  $(c \wedge a) \rightarrow k$



Ryc. 61d: ligacja powyższej molekuły oraz molekuły faktu  $b$



Ryc. 61e: ligacja powyższej molekuły oraz molekuły wnioskowania  $l \rightarrow m$



Ryc. 61f: ligacja powyższej molekuły oraz molekuły reprezentującej hipotezę  $m$

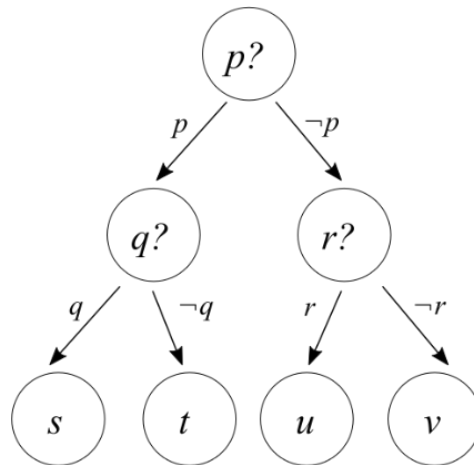
Tak powstała cząsteczka, niezawierająca „lepkich końców”, oznacza że zadana hipoteza  $m$  jest możliwa do wyprowadzenia na podstawie przesłanek wejściowych: zbioru faktów i reguł wnioskowania. Uzyskana długość cząsteczki wynika ze ścieżki dowodzenia jaką obrano – jeżeli interesuje nas wyłącznie możliwość jej wyprowadzenia, wtedy długość nie jest istotna. Gdyby zabrakło któregoś z elementów niezbędnych do wywnioskowania hipotezy, wspomniana tępo zakończona molekuła nie powstałaby w trakcie reakcji.

Omówiona koncepcja przede wszystkim wydaje się ciekawa ze względu na całkowitą dowolność konstruowania molekuł składowych i brak ograniczeń co do ich długości, co pozwala użyć dowolną liczbę zmiennych. Oczywiście różne możliwe drogi wnioskowania zostaną przetworzone w sposób równoległy co jest największą zaletą wszystkich modeli obliczeń wykorzystujących DNA. Niestety należy zwrócić uwagę na to, że kolejność składowych implikacji musi być tutaj odpowiednio dobrana dla danej reakcji aby ich łączenie przebiegło poprawnie, np. gdyby w omówionym powyżej przykładzie zamienić kolejność implikacji  $((k \wedge b) \rightarrow l)$  na  $((b \wedge k) \rightarrow l)$ , powstała w wyniku połączenia składowych cząsteczka najpewniej rozwidlałaby się, a system nie udzieliłby poprawnej odpowiedzi.

### 3. 3. 2. Wnioskowanie z wykorzystaniem drzewa decyzyjnego

Drzewo decyzyjne jest acyklicznym grafem skierowanym. Zawiera jeden węzeł początkowy, tzw. korzeń, węzły pośrednie oraz węzły końcowe, tzw. liście. Krawędzie grafu (tzw. gałęzie) mogą wychodzić z węzła początkowego jak i węzłów pośrednich. Z każdego z węzłów może wychodzić dowolna liczba gałęzi. Podejmowanie decyzji polega na wybraniu właściwej krawędzi w każdym z rozwidlających się węzłów. Jeżeli ograniczymy ich liczbę do dwóch, to takie drzewo nazywamy drzewem binarnym. Ograniczony w taki sposób system decyzyjny odpowiada kolejnej koncepcji systemu wnioskującego logiki klasycznej, wykorzystującej

również negację. Weźmy przykładowe implikacje:  $(p \wedge q) \rightarrow s$ ,  $(p \wedge \neg q) \rightarrow t$ ,  $(\neg p \wedge r) \rightarrow u$ ,  $(\neg p \wedge \neg r) \rightarrow v$ . Zapisane dla nich drzewo decyzyjne będzie wyglądać następująco:



Ryc. 62: przykładowe drzewo decyzyjne

Realizacja takiej koncepcji za pomocą molekuł DNA zakłada przedstawienie w określony sposób wszystkich składowych drzewa: korzenia, węzłów pośrednich, liści oraz gałęzi, gdzie w odniesieniu do przedstawionych wcześniej koncepcji, możemy powiedzieć, że gałęzie reprezentują fakty a węzły implikacje (gdzie kilka węzłów składa się na reprezentację pojedynczej implikacji), a każdy liść zawiera wniosek jednej z implikacji. Wszystkie elementy systemu reprezentowane są za pomocą odpowiednio dobranych jednoniciowych molekuł DNA o dowolnej długości. Korzeń zawiera fragment radioaktywny lub fluorescencyjny aby po elektroforezie w żelu (służącej do wykrycia odpowiedzi) można zidentyfikować wszystkie łańcuchy, do których został dołączony. Łańcuch zaczynający się korzeniem, przechodzący przez kolejne węzły pośrednie i kończący liściem tworzy odpowiednią ścieżkę dowodzenia. Tak powstała cząsteczka nie posiada „lepkich końców”. Przed przeprowadzeniem reakcji tworzone są dwa roztwory: pierwszy reprezentujący wszystkie węzły drzewa, do którego następnie dodaje się drugi, reprezentujący gałęzie drzewa. Oba dobierane są w sposób, gwarantujący niereaktywność w ramach pojedynczego roztworu. Przykładowe kodowanie molekuł dla powyższego przykładowego drzewa decyzyjnego wygląda następująco:



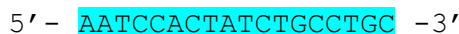
Ryc. 63a: schemat molekuly gałęzi drzewa, reprezentującej fakt  $p$



Ryc. 63b: schemat molekuly gałęzi drzewa, reprezentującej fakt  $\neg p$



Ryc. 63c: schemat molekuly gałęzi drzewa, reprezentującej fakt  $q$



Ryc. 63d: schemat molekuly gałęzi drzewa, reprezentującej fakt  $\neg q$



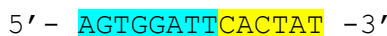
Ryc. 63e: schemat molekuly gałęzi drzewa, reprezentującej fakt  $r$



Ryc. 63f: schemat molekuly gałęzi drzewa, reprezentującej fakt  $\rightarrow r$



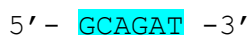
Ryc. 63g: schemat molekuly korzenia  $p$ ?



Ryc. 63h: schemat molekuly węzła pośredniego, reprezentującego  $q$ ?



Ryc. 63i: schemat molekuly węzła pośredniego, reprezentującego  $r$ ?



Ryc. 63j: schemat molekuly liścia, reprezentującego wniosek  $s$



Ryc. 63k: schemat molekuly liścia, reprezentującego wniosek  $t$



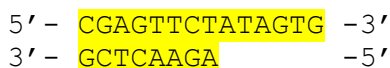
Ryc. 63l: schemat molekuly liścia, reprezentującego wniosek  $u$



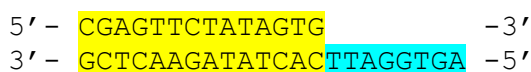
Ryc. 63m: schemat molekuly liścia, reprezentującego wniosek  $v$

Molekuly reprezentujące węzły zawsze uczestniczą w reakcji.

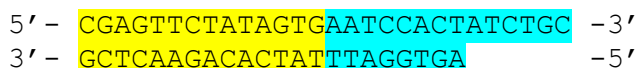
Reakcja wykorzystująca gałęzie o wartościach  $p$  i  $q$  wygląda następująco:



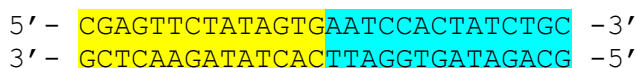
Ryc. 64a: ligacja molekul reprezentujących korzeń  $p$ ? oraz gałęzi  $q$



Ryc. 64b: ligacja powyższej molekuly oraz molekuly reprezentującej węzeł pośredni  $r$ ?

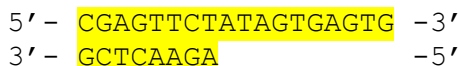


Ryc. 64c: ligacja powyższej molekuly oraz molekuly reprezentującej gałąź  $q$

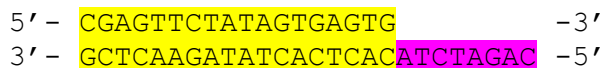


Ryc. 64d: ligacja powyższej molekuly oraz molekuly reprezentującej liść  $s$

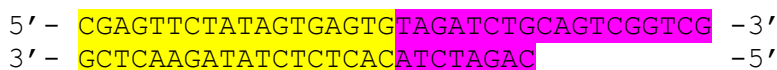
Reakcja wykorzystująca gałęzie o wartościach  $\rightarrow p$  i  $\rightarrow r$  wygląda następująco:



Ryc. 65a: ligacja molekul reprezentujących korzeń  $p$ ? oraz gałęzi  $p$



Ryc. 65b: ligacja powyższej molekuly oraz molekuly reprezentującej węzeł pośredni  $r$ ?



Ryc. 65c: ligacja powyższej molekuly oraz molekuly reprezentującej węzeł pośredni  $r$ ?



Ryc. 65d: ligacja powyższej molekuly oraz molekuly reprezentującej liść  $v$

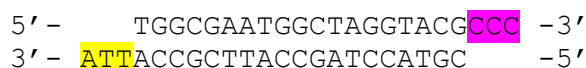
Analogicznie będą wyglądać reakcje dla pozostałych kombinacji faktów i powstałych na ich podstawie wniosków. W przypadku całkowitego braku informacji o którejs z zmiennych

zdaniowych, oczywiście reakcja nie wykona się i żadna molekula dwuniciowa niezawierająca „lepkich końców” nie powstanie.

Przedstawiona koncepcja jest ciekawa przede wszystkim dlatego, że jako jedna z pierwszych pozwala zastosować rozłączne wartościowania tej samej zmiennej, w omawianym przykładzie użyte do reprezentacji wartości *prawdy* i *falszu*. Niestety nie możemy tu wyciągać takich samych wniosków dla różnych przesłanek (każdy z liści drzewa musi dotyczyć się pojedynczego wniosku) ani powielać węzłów (np. jeżeli dla *p* o wartości *prawdy* zapytamy o wartość *q*, to dla *p* o wartości *falszu* nie możemy przeprowadzić dalszej analizy również w oparciu o *q* – spowodowałoby to błędne połączenie się ze sobą nieodpowiednio komplementarnych molekuł). Należy zwrócić uwagę, że jej zastosowanie nie kończy się wyłącznie na systemach związanych z logiką klasyczną.

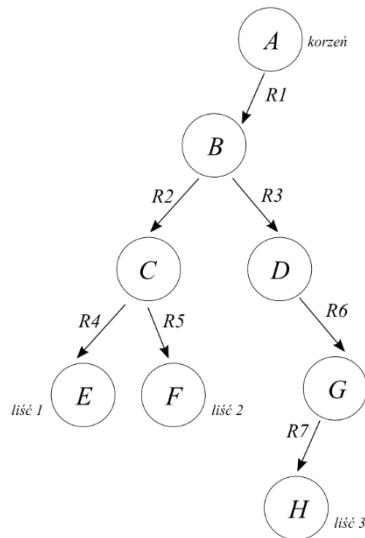
### 3. 3. 3. System wnioskujący oparty na cząsteczkach kolistych

Kolejna z koncepcji systemu wnioskującego, przedstawionych przez zespół badawczy prof. dr hab. inż. Jana Mulawki jest w pełni dedykowana dowodzeniu w logice klasycznej. Została ona po raz pierwszy przedstawiona w [54] w roku 1999 a następnie szczegółowo opisana w pracy „The inference based on molecular computing” [56] z roku 2000, której współautorami są Piotr Wąsiewicz oraz Tomasz Janczak. W wersji podstawowej (nie uwzględniającej molekuł rozwidlających się) wykorzystuje jeden rodzaj cząsteczek dwuniciowych, przedstawiający regułę wnioskowania postaci ( $p \rightarrow q$ ), która analogicznie jak w koncepcji opartej na drzewie decyzyjnym (*patrz*: podrozdział 3.3.2) odpowiada pojedynczej gałęzi. Samo drzewo może jednak przedstawiać zbiór reguł na kilka sposobów. Korzeń oraz liść (gdzie pojedynczy liść odpowiada zapytaniu o pojedynczy wniosek) kodowane są za pomocą tzw. cząsteczki DBF (nazwa użyta w [56], oznaczająca molekułę łączącą korzeń z liściem). Możemy powiedzieć, że DBF w rzeczywistości oznacza implikację (*liść*  $\rightarrow$  *korzeń*), co sprowadza się do tego, że gdy w próbówce powstanie ścieżka je łącząca, to tak skonstruowana molekula domknie całość do postaci kolistej. Domknięta molekula kolista oznacza tu odpowiedź twierdzącą systemu. Do wykrycia jej wykorzystywane są reakcja PCR i elektroforezy w żelu, co oznacza że tak skonstruowany system jest oparty na modelu filtrowania (*patrz*: podrozdział 2.3). Omawiane cząsteczki zawierają „lepkie końce” z obu stron w taki sposób, że jeden z nich identyfikuje poprzednik, a drugi następnik implikacji (gdzie pojedyncza implikacja odpowiada pojedynczej regule wnioskowania). Sekwencje nukleotydów znajdujące się w środku cząsteczki (w formie dwuniciowej) nie mają większego znaczenia dla reakcji – ich głównym zadaniem jest wydłużenie cząsteczki co pozwala zapobiec jej przypadkowemu rozpadowi. Wygląda to następująco:



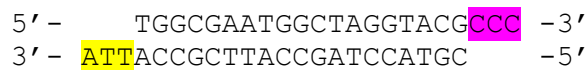
Ryc. 66: przykładowa molekula reprezentująca pojedynczą regułę wnioskowania

Weźmy drzewo reguł przedstawione w [56]:



Ryc. 67: przykładowe drzewo reguł wnioskowania

Na potrzeby zapisu molekularnego i przykładowej reakcji, ograniczymy drzewo do węzłów **A, B, C, D, E, F** (dla badań przedstawionych w [56] również zostało ono ograniczone). Przyjmijmy, że każdy węzeł (nazwany dużą literą) odpowiada za prawdziwość literału (nazwanego małą literą) odpowiadającego nazwie węzła. Np. węzeł **A**, będący korzeniem oznacza fakt **a**. Kodowanie molekularne reguł wygląda następująco:



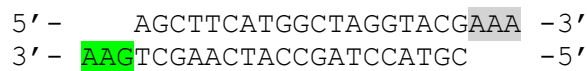
Ryc. 68a: molekula reprezentująca regułę wnioskowania R1 czyli ( $a \rightarrow b$ )



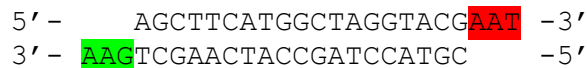
Ryc. 68b: molekula reprezentująca regułę wnioskowania R2 czyli ( $b \rightarrow c$ )



Ryc. 68c: molekula reprezentująca regułę wnioskowania R3 czyli ( $b \rightarrow d$ )

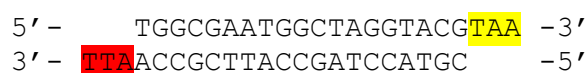


Ryc. 68d: molekula reprezentująca regułę wnioskowania R4 czyli ( $c \rightarrow e$ )



Ryc. 68e: molekula reprezentująca regułę wnioskowania R5 czyli ( $c \rightarrow f$ )

Aby sprawdzić czy istnieje możliwość znalezienia ścieżki od korzenia do węzła **F**, co jest równoznaczne zapytaniu **f?** na podstawie danej bazy reguł (i przyjęciu prawdziwości korzenia), musimy umieścić w próbówce następującą molekułę domykającą DBF:

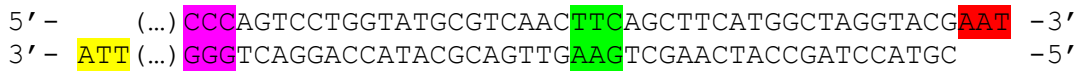


Ryc. 69: molekula DBF reprezentująca zapytanie **f?**

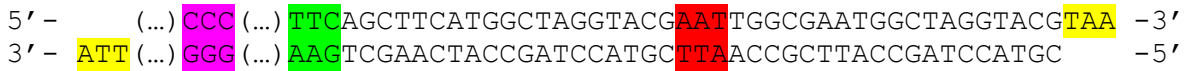
Reakcja będzie wyglądać następująco:



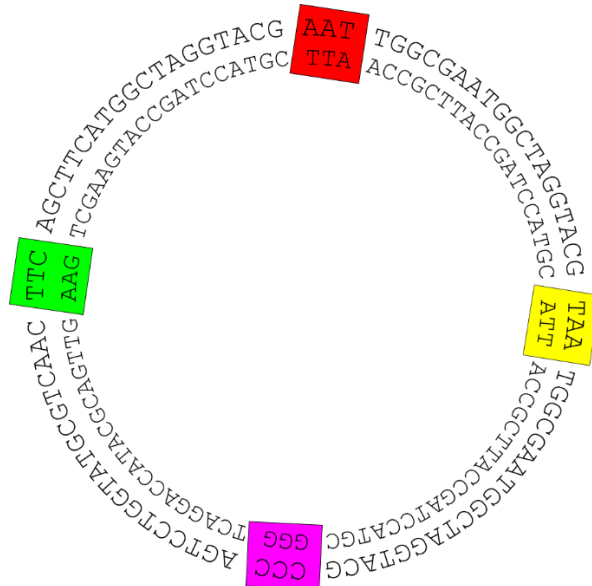
Ryc. 70a: ligacja molekuł reprezentujących reguły wnioskowania R1 oraz R2



Ryc. 70b: ligacja powyższej molekuły oraz molekuły reprezentującej regułę wnioskowania R5



Ryc. 70c: ligacja powyższej molekuły oraz molekuły DBF reprezentującej zapytanie f?



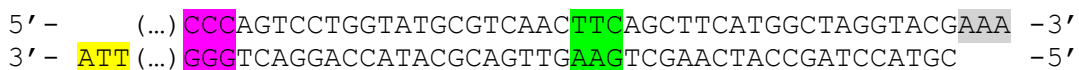
Ryc. 70d: połączenie się ze sobą komplementarnych „lepkich końców” i domknięcie molekuły do postaci kolistej

Taka molekuła oznacza odpowiedź twierdzącą systemu.

W tym samym czasie, w sposób równoległy zajdą też kolejne reakcje, nieprowadzące do uzyskania molekuły reprezentującej poprawną odpowiedź systemu na zadane pytanie:



Ryc. 71a: ligacja molekuł reprezentujących reguły wnioskowania R1 oraz R3



Ryc. 71b: ligacja powstałej wcześniej molekuły oraz molekuły reprezentującej regułę wnioskowania R4

Żadna z nich nie bierze udziału w dalszej reakcji.

W omawianej koncepcji głównym elementem nowatorskim jest pokazanie działającej w sposób autonomiczny przechodniości implikacji. W przypadku skomplikowanego drzewa, gdzie liczba możliwych reakcji byłaby dużo większa, wszystkie z możliwych połączeń powstaną w sposób równoległy, co jest główną zaletą biokomputerów. Jeżeli umieścilibyśmy w próbówce więcej niż jedno zapytanie (czyli molekułę DBF, łączącą pojedynczy liść z korzeniem) to byłoby to analogiczne z umieszczeniem zapytania o ich alternatywę – możliwość wyprowadzenia wyłącznie jednego z nich spowodowałaby pojawienie się kolistej molekuły, analogicznej jak w przedstawionej powyżej reakcji. Istnieje też możliwość

rozdzielenia różnych odpowiedzi po ich długości (w oparciu o szacowaną liczbę molekuł niezbędnych do uzyskania danej ścieżki) za pomocą reakcji elektroforezy w żelu lub rozdzielenia konkretnych sekwencji (unikalnych dla danej ścieżki) za pomocą reakcji łańcuchowej polimerazy (PCR). Za pomocą elektroforezy jest również możliwe odróżnienie molekuł kolistych od pozostałych.

Omówiona tu przechodność implikacji jest również jednym z praw klasycznego rachunku zdań, które w sposób automatyczny zawarte jest w nowej koncepcji systemu wnioskującego (*patrz*: podrozdział 4.6.4).

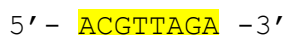
### 3. 3. 4. Pozostałe prace

Oprócz omówionych powyżej koncepcji bezpośrednio związanych z wnioskowaniem logicznym za pomocą DNA, które stanowi główny temat tej rozprawy, zaprezentowanych przez grupę badawczą Jana Mulawki zostało jeszcze kilka koncepcji, pośrednio łączących się z tematyką. Między innymi w pracach [53], [55], [57], [58] przedstawiono udoskonalaną implementację bramek logicznych wykorzystujących operacje biomolekularne, rozwijając ją w odniesieniu do starszych prac zespołów Mitsunori Ogihary [23] i Amosa Martyna [19]. W niektórych ze wspomnianych prac, oprócz odpowiedniego łączenia się molekuł, używane jest również cięcie ich za pomocą enzymów restrykcyjnych, jednak jeszcze bez wykorzystania koncepcji splatania (naprzemiennego cięcia i łączenia się molekuł, *patrz*: podrozdział 2.4). Wszystkie z omawianych koncepcji zostały potwierdzone odpowiednimi eksperymentami laboratoryjnymi.

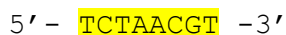
### 3. 4. System In-Hee Lee, Ji-Yoon Park i innych.

Zupełnie inna koncepcja wnioskowania logicznego, a dokładniej przeprowadzenia logicznych dowodów nie-wprost, została przedstawiona w roku 2003 w pracy „DNA Implementation of Theorem Proving with Resolution Refutation in Propositional Logic” przez In-Hee Lee, Ji-Yoon Park, Hae-Man Jang, Young-Gyu Chai oraz Byong-Tak Zhang – grupę badawczą z Korei Południowej [9]. Model ten opiera się na systemie filtrowania (*patrz*: podrozdział 2.3) co oznacza, że podczas reakcji wykorzystuje wyłącznie ligację znajdujących się w próbówce molekuł, a do wykrycia odpowiedzi reakcją łańcuchową polimerazy.

Przed zakodowaniem dowolnego zdania logicznego za pomocą DNA, musi ono zostać sprowadzone do omówionej wcześniej koniunkcyjnej postaci normalnej (*patrz*: podrozdział 1.3). Pojedynczy literał (czyli zdanie atomowe o konkretnym wartościowaniu) reprezentowany jest przez dowolnej (ale stałej dla wszystkich literałów) długości jednoniciowy łańcuch DNA. Literał o przeciwnej wartości reprezentowany jest przez jednoniciowy łańcuch komplementarny. Przykładowe molekuly reprezentujące różne wartości zdania atomowego *a* wyglądają następująco:



Ryc. 72a: schemat molekuly reprezentującej literał *a*



Ryc. 72b: schemat molekuly reprezentującej literał  $\neg a$

Klauzula zawierająca alternatywę kodowana jest poprzez sklejanie łańcuchów reprezentujących składowe literały. Przykładowa molekula reprezentująca klauzulę  $(\neg a \vee b \vee c)$  wygląda następująco:



Ryc. 73: schemat molekuly reprezentujcej klauzulę  $(\neg a \vee b \vee c)$

Literał, o którego możliwość wyprowadzenia na podstawie posiadanych przesłanek pytamy nazywamy zapytaniem. Do jego zakodowania użyjemy koncepcji dowodu nie-wprost, opierającej się na przyjęciu za prawdę zaprzeczenia informacji, o którą pytamy, a następnie sprowadzenie obliczeń do sprzeczności. Przykładowe zapytanie  $a?$  wygląda zatem identycznie jak  $\neg a$ :



Ryc. 74: schemat molekuly reprezentujcej zapytanie  $a?$

Przykładowa reakcja, przedstawiona w omawianej pracy, dla łatwiejszego jej zrozumienia wykorzystuje kodowanie pojedynczego literału wyłącznie za pomocą czterech nukleotydów. Aksjomaty biorące udział w reakcji to:  $(\neg q \vee \neg p \vee r)$ ,  $(q \vee \neg t \vee \neg s)$ ,  $s$ ,  $p$ ,  $t$  oraz zapytanie  $r?$ , które molekularnie kodujemy również jako  $\neg r$ . Kodowanie molekul wygląda zatem następująco:



Ryc. 75a: schemat molekuly reprezentujcej  $(\neg q \vee \neg p \vee r)$



Ryc. 75b: schemat molekuly reprezentujcej  $(q \vee \neg t \vee \neg s)$



Ryc. 75c: schemat molekuly reprezentujcej  $s$



Ryc. 75d: schemat molekuly reprezentujcej  $p$



Ryc. 75e: schemat molekuly reprezentujcej  $t$



Ryc. 75f: schemat molekuly reprezentujcej  $\neg r$

Reakcja wygląda następująco:



Ryc. 76a: ligacja molekul reprezentujcych  $(\neg q \vee \neg p \vee r)$  oraz  $(q \vee \neg t \vee \neg s)$



Ryc. 76b: przyłączenie się kolejnych molekul (tutaj:  $t$ )



Ryc. 76c: przyłączenie się kolejnych molekul (tutaj:  $s$ ,  $p$  oraz  $\neg r$ )

Zaplanowana w kolejnym kroku reakcja łańcuchowa polimerazy (*patrz*: podrozdział 2.1.3) nie wykona się (ponieważ powstała molekula jest dwuniciowa i nie zawiera żadnych „lepkich końców”), co tutaj oznacza odpowiedź twierdzącą tak skonstruowanego systemu – zatem  $r$  jest możliwe do wyprowadzenia na podstawie pozostałych przesłanek. W omawianym



systemie na odpowiedź twierdzącą składa się „sklejenie” wszystkich biorących udział w reakcji molekuł w jedną molekułę dwuniciową. W przypadku odpowiedzi przeczącej (dla innych danych bądź innego zapytania), któryś z fragmentów głównej molekuly nie zawierałby komplementarnego końca bądź w próbówce pozostałyby pojedyncze jednoniciowe niewykorzystane w reakcji molekuly.

Z jednej strony zrezygnowanie z użycia enzymów restrykcyjnych wydaje się ciekawym pomysłem ze względu na łatwość konstruowania molekularnych reprezentacji pojedynczych literałów oraz ich nieograniczoną liczbę (dzięki dowolnej długości takiego łańcucha jednoniciowego możemy zakodować dowolnie wiele różnych od siebie zmiennych zdaniowych). Po dokładniejszym przyjrzeniu się, koncepcja jednak zdaje się mieć sporo ograniczeń, a przede wszystkim:

- Nie ma możliwości umieszczenia w próbówce dodatkowych aksjomatów, które akurat nie biorą udziału w danej reakcji wnioskowania (i tym samym powinny nie mieć na nią żadnego wpływu).
- Nie ma możliwości zmiany kolejności składowych literałów alternatywy – ich kolejność musi być dobrana w taki sposób aby łączenie się molekuł przebiegło poprawnie, czyli w rzeczywistości reakcja powinna być częściowo zaplanowana i przeprowadzona „na papierze” przed wykonaniem jej w laboratorium. Co prawda autorzy wspominają o możliwości powstawania rozgałęzionych molekuł o trzech albo więcej końcach (z ang. *hairpin molecules*) w taki sposób, że reakcja PCR nie wystąpi, jednak taka reakcja jest dużo mniej stabilna, a ponadto zawsze ograniczona (nie ma możliwości utworzenia stabilnej molekuly o dowolnej wielkości  $n$ -końcach).
- Jeżeli w próbówce nie nastąpi żadna reakcja połączenia się łańcuchów, to reakcja PCR również nie wykona się, ponieważ do jej przeprowadzenia niezbędne są tzw. startery znajdujące się na komplementarnej nici (*patrz*: podrozdział 2.1.3), co może zostać pomylone z odpowiedzią twierdzącą systemu.

Podsumowując, najbardziej istotną inspiracją w omówionej pracy jest idea wykorzystania dowodu nie-wprost do przeprowadzenia logicznego wnioskowania. Koncepcja ta zostanie przedstawiona również jako jedno z dodatkowych zastosowań nowego systemu wnioskującego (*patrz*: podrozdział 5.3.1).

### 3. 5. Pozostałe koncepcje

Omówione powyżej koncepcje miały ważny wpływ i stanowiły inspirację dla powstałego autorskiego systemu wnioskującego, który jest najważniejszym elementem tej pracy. Oprócz nich, istnieje jeszcze sporo (mniej lub bardziej istotnych) implementacji wykorzystujących obliczenia DNA i operacje logiczne.

Jedną z nich jest przedstawiona przez Yasubumi Sakakibarę oraz Akira Suyamę w pracy „Intelligent DNA Chips: Logical Operation of Gene Expression Profiles on DNA Computers” z 2000 roku [39] implementacja sprawdzająca wartość funkcji logicznej na podstawie podanych argumentów. Wykorzystuje ona dysjunkcyjną postać normalną (*patrz*: podrozdział

1.3), model filtrowania (*patrz:* podrozdział 2.3) i wykrywanie wyniku za pomocą reakcji łańcuchowej polimerazy oraz fluorescencji.

Kolejną koncepcją jest wspomniana już we wstępie, a przedstawiona po raz pierwszy w roku 2010 przez grupę badawczą Alfonso Rodríguez Patón [29], a następnie rozwijana wraz z Iñaki Sainz de Murieta i Petrem Sosíkem [30], co zaowocowało również pracą doktorską Muriety [38]. Jest to model oparty na technice DSD (z ang. *DNA strand displacement*), wykorzystujący koniunkcyjną postać normalną (*patrz:* podrozdział 1.3), rozgałęzione molekuly o czterech i więcej końcach (z ang. *hairpin molecules*) i możliwe reakcje między nimi [26], [43], koniunkcję, alternatywę, wnioskowania *modus ponens* i *modus tollens*, wykrywanie sprzeczności oraz wykrycie wyniku wykorzystujące fluorescencję. Autorzy przedstawili również koncepcję częściowego rozwiązania problemu SAT. Przedstawione wyniki mają charakter wyłącznie teoretyczny, a podobne doświadczenia laboratoryjne nie były jeszcze przeprowadzane. Możliwości tego systemu są zbliżone do możliwości nowego systemu wnioskującego przedstawianego w tej rozprawie – opierają się natomiast na zupełnie różnych i niezwiązanych ze sobą modelach komputera DNA.

Kolejne ciekawe opracowanie wykorzystujące model DSD i wnioskowanie *modus ponens* zostało przedstawione w roku 2016 przez Raya i Mondala w [28].

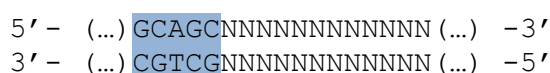
W świetle przedstawionych porównań i analiz, omawiany w dalszej części pracy rozszerzony system wnioskowania zbiera i łączy zalety większości istniejących dotąd koncepcji, a także w możliwie największym stopniu eliminuje ich istotne wady. Ponadto uwzględnia on (jako pierwszy system splatania) prawa klasycznego rachunku zdań i odwołuje się do ich zasad rozumowania matematycznego (a pośrednio również rozumowania ludzkiego). Jego implementacja laboratoryjna wzoruje się w większości na przeprowadzonych dotąd badaniach przedstawionych w [12], [15], [27].

## 4. Rozszerzony system wnioskowania

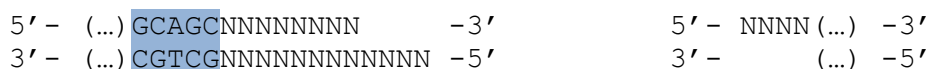
W poniższym rozdziale zostanie przedstawiona nowa koncepcja rozszerzonego systemu wnioskującego. Zostaną omówione wszystkie jej elementy składowe, schemat działania systemu z przykładami oraz odniesienie do praw klasycznego rachunku zdań (podkreślając, które z tautologii klasycznego rachunku zdań wykorzystywane są w omawianym systemie i w jaki sposób).

### 4.1. Wprowadzenie

Omawiany nowy model systemu wnioskującego, który jest najważniejszym elementem tej rozprawy, oparty jest o omówiony w rozdziale 2 model komputera DNA – system splatania. Używa on enzymu ligazy jako katalizatora reakcji łączenia się ze sobą komplementarnych łańcuchów oraz enzymu restrykcyjnego *BbvI*, który rozcina dwuniciowe łańcuchy w określony sposób, pozostawiając zawsze „lepkie końce” o długości 4 na nici 3’-5’ (patrzac w kierunku cięcia). Istotą reakcji jest naprzemienne łączenie się i cięcie molekuł z wykorzystaniem ich „lepkich końców”, omówione również w rozdziale 2. Enzym restrykcyjny *BbvI* rozpoznaje konkretny ciąg nukleotydów (5’-GCAGC-3’) i rozcina molekułę w odległości 8 i 12 nukleotydów od wspomnianej sekwencji rozpoznawalnej (gdzie N oznacza dowolny nukleotyd, patrz: podrozdział 2.1):



Ryc. 77a: ciąg nukleotydów niezbędny do rozpoznania przez enzym restrykcyjny *BbvI*



Ryc. 77b: rozcięcie za pomocą enzymu restrykcyjnego *BbvI*

### 4.2. Kodowanie informacji

Jako podstawowy zbiór aksjomatów przyjmujemy dowolne zdania logiczne, zapisane w koniunkcyjnej postaci normalnej. Oznacza to, że każde takie zdanie  $A$  jest koniunkcją dowolnej skończonej liczby klauzul  $B_1, B_2, \dots, B_n$ , czyli:

$$A \equiv B_1 \wedge \dots \wedge B_n,$$

gdzie dla każdego  $k \in \langle 1, \dots, n \rangle$ , każda z nich jest alternatywą dowolnej skończonej liczby literałów, czyli:

$$\forall k \in \langle 1..n \rangle \quad B_k \equiv b_{1,k} \vee \dots \vee b_{m,k},$$

gdzie literał jest pojedynczą zmienną zdaniową lub jej negacją. W szczególnych przypadkach indeksy  $m$  oraz  $n$  mogą równać się 1. Jeżeli dla danego  $k$ , wartość  $m$  wynosi 1, to taką klauzulę  $B_k$  nazywamy klauzulą faktu (jest to w rzeczywistości pojedynczy literał).

W omawianym systemie, każdą z klauzul  $B_i$  będziemy kodować za pomocą pojedynczej molekuly. Szczególnym przypadkiem są klauzule faktu, których kodowanie różni się od kodowania pozostałych klauzul, ale również każdej z nich odpowiada dokładnie jedna molekula.

### 4.2.1. Literały

Za reprezentację pojedynczego literału przyjmujemy unikalną sekwencję czterech nukleotydów, wyznaczoną w taki sposób, że literały reprezentujące zmienną atomową i jej negację są do siebie komplementarne, np. jeżeli  $p$  przypiszemy  $3' - AAAC - 5'$ , to łańcuch komplementarny reprezentujący  $\neg p$  ma postać  $3' - GTTT - 5'$ . W związku z przyjęciem takiej reguły, musimy zrezygnować z użycia czwórek, które są komplementarne same ze sobą np.  $3' - AATT - 5'$  (odróżnienie literału od jego negacji nie byłoby w tym przypadku możliwe). Ze względu na konstrukcję pozostałych molekuł i zastosowany enzym restrykcyjny *BbvI*, musimy odrzucić również sekwencje postaci  $3' - GNNN - 5'$  oraz  $3' - NNNC - 5'$  (gdzie N oznacza dowolny nukleotyd), ponieważ tylko w ten sposób możemy uniknąć powstania sekwencji rozpoznawalnej przez ten enzym w niepożądanym miejscu – w połączeniu ze stałą sekwencją  $3' - CGAC - 5'$ , będącą fragmentem klauzuli faktu (omówionej za moment, patrz: podrozdział 4.2.2) utworzyłaby ciąg nukleotydów rozpoznawalny przez enzym restrykcyjny *BbvI*, co akurat w tym miejscu jest niepożądane.

Seqwencje nukleotydów reprezentujące literały nie występują w reakcji samodzielnie – są one wyłącznie częścią właściwych molekuł, których budowa zostanie omówiona w dalszej części. Analiza liczby możliwych do zakodowania w ten sposób zmiennych zostanie dodatkowo omówiona w podrozdziale 5.1.

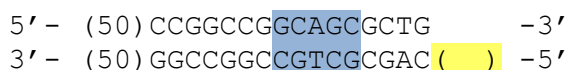
Przyjmując omówioną konstrukcję czwórek, na potrzeby uproszczenia zapisu, w dalszej części pracy zostanie użyty zapis skrótowy postaci  $3' - (p) - 5'$ , który oznacza czwórkę nukleotydów reprezentujących  $p$  oraz  $3' - (\bar{p}) - 5'$ , który oznacza  $\neg p$ .

### 4.2.2. Klauzula faktu

Każdy fakt (klauzula zawierająca tylko jeden literał), znajdujący się w podstawowym zbiorze aksjomatów, kodowany jest za pomocą pojedynczej molekuły, o innym schemacie niż dla klauzul zawierających alternatywy (które omówione zostaną dalej). Wszystkie molekuły faktów mają taką samą długość oraz tę samą sekwencję początkową, a między sobą różnią się wyłącznie „lepkim końcem” powiązany z konkretnym literałem. To właśnie te molekuły zawierają sekwencję rozpoznawalną przez enzym restrykcyjny *BbvI*, dlatego też są niezbędne dla właściwego przeprowadzenia reakcji.

Molekuła faktu składa się na nici  $3' - 5'$  z: 50 dowolnych nukleotydów, stałej sekwencji  $3' - GGCCGGC - 5'$ , sekwencji  $3' - CGTCG - 5'$  rozpoznawalnej przez enzym restrykcyjny *BbvI* (zaznaczonej kolorem niebieskim), stałej sekwencji  $3' - CGAC - 5'$  (która stanowi fragment sekwencji rozpoznawalnej przez enzym i w wyniku odpowiedniego cięcia i łączenia może utworzyć fragment rozpoznawalny w dalszej części reakcji) oraz „lepkiego końca” (zaznaczonego kolorem żółtym), złożonego z czterech nukleotydów, znajdującego się na nici  $3' - 5'$ , który odpowiada za jednoznaczną identyfikację literału.

Schemat wygląda następująco:



Ryc. 78: schemat molekuły reprezentującej klauzulę faktu

Zatem zgodnie z przyjętym powyżej uproszczonym schematem zapisu, molekuly kodowania literałów  $p$  oraz  $\neg p$  wyglądają następująco:



Ryc. 79a: molekula reprezentująca klauzulę faktu  $p$

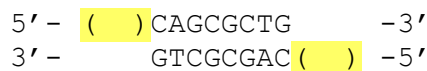


Ryc. 79b: molekula reprezentująca klauzulę faktu  $\neg p$

### 4.2.3. Pozostałe klauzule

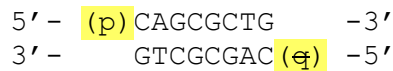
Każda klauzula zawierająca dwa lub więcej literałów połączonych alternatywami kodowana jest za pomocą pojedynczej molekuly, innej jednak niż w przypadku klauzuli faktu. Każda klauzula zawierająca dokładnie dwa literały skonstruowana jest według następującego schematu: dwa „lepkie końce”, zawierające informacje o składowych literałach alternatywy oraz stałej symetrycznej sekwencji  $3' - \text{GTCGCGAC} - 5'$  rozdzielającej łańcuchy reprezentujące literały (jej symetryczność i to, że odpowiada ona stałym sekwencjom użytym wewnątrz klauzul faktu, sprawia że w wyniku odpowiedniego wnioskowania każda ze składowych może zostać przekształcona we właściwą molekule faktu).

Schemat dla dwuliterałowej klauzuli wygląda następująco:



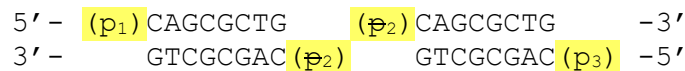
Ryc. 80: schemat molekuly reprezentującej dwuliterałowk klauzulę

Zatem zgodnie z przyjętym schematem kodowania, przykładowa molekula reprezentująca alternatywę literałów  $(p \vee \neg q)$  wyglądają następująco:



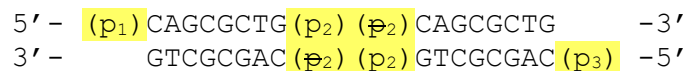
Ryc. 81: molekula reprezentująca klauzulę  $(p \vee \neg q)$

Aby zapisać klauzulę o większej (dowolnej) liczbie literałów, musimy „zlepić ze sobą” kilka molekuly powstałych według schematu powyżej. Weźmy przykładowk klauzulę  $(p_1 \vee \neg p_2 \vee p_3)$ , a następnie zapiszmy obok siebie jak wyglądałyby (zgodnie ze schematem przedstawionym powyżej) molekuly reprezentujące odpowiednio  $(p_1 \vee \neg p_2)$  oraz  $(\neg p_2 \vee p_3)$ :



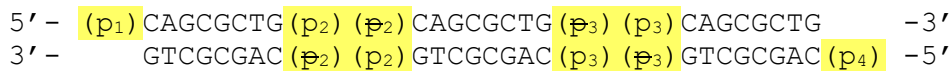
Ryc. 82: zestawione obok siebie klauzule  $(p_1 \vee \neg p_2)$  oraz  $(\neg p_2 \vee p_3)$

W celu faktycznego „zlepienia” do postaci jednej klauzuli (reprezentowanej przez jednk molekule) należy dopisać brakujące fragmenty komplementarne. Uzyskujemy molekule następującej postaci:

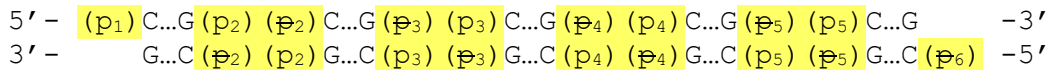


Ryc. 83: molekula reprezentująca klauzulę  $(p_1 \vee \neg p_2 \vee p_3)$

W analogiczny sposób będą zapisywane molekuly reprezentujące klauzule składające się z dowolnej liczby literałów. Przykładowe, zawierające odpowiednio cztery i sześć różnych literałów, wyglądają następująco:



Ryc. 84a: molekula reprezentująca klauzulę  $(p_1 \vee \neg p_2 \vee p_3 \vee p_4)$

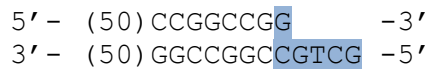


Ryc. 84b: molekula reprezentująca klauzulę  $(p_1 \vee \neg p_2 \vee p_3 \vee p_4 \vee p_5 \vee \neg p_6)$

Kolejność czwórek reprezentujących poszczególne literały składowe (jak i to, które z nich reprezentowane są w formie „lepkiego końca”) nie ma znaczenia. Jeśli w tej samej próbówce istnieje zapytanie i aksjomaty prowadzące do uzyskania odpowiedzi twierdzącej, to kolejność czwórek wpłynie wyłącznie na to w jakiej kolejności będą zachodzić kroki reakcji, a finalny efekt będzie identyczny.

#### 4.2.4. Molekuła terminalna i wnioskowanie

W omawianym systemie, samo wnioskowanie odbywa się na podstawie zapisanych w koniunkcyjnej postaci normalnej aksjomatów początkowych (*patrz*: podrozdział 1.3), zakodowanych molekularnie według powyższego schematu. Aby wnioskowanie przebiegało poprawnie, oprócz odpowiednio zakodowanych klauzul, niezbędne jest umieszczenie w próbówce także molekuly terminalnej, która uczestniczy w każdej reakcji reprezentującej proces wnioskowania. Przyjmujemy, że jej początek jest identyczny jak w przypadku molekuł reprezentujących klauzule faktu (a służy ona właśnie dopełnieniu molekuly będącej częściowym wynikiem reakcji do postaci reprezentującej klauzulę faktu, czyli właściwy wniosek). Wygląda ona następująco:

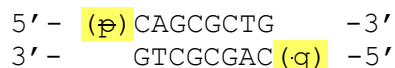


Ryc. 85: molekula terminalna wnioskowania

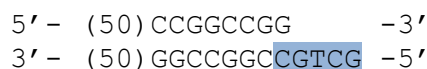
Zatem do przeprowadzenia najprostszej możliwej reakcji wnioskowania za pomocą reguły odrywania, potrzebujemy molekuly reprezentujące odpowiednio:  $p$ ,  $(\neg p \vee q)$ , molekule terminalną oraz odpowiednie środowisko (odpowiednią temperaturę reakcji, enzymem restrykcyjny *BbvI* oraz enzym ligazy). Odpowiada to wnioskowaniu *modus ponens* dla reguł:  $p, p \rightarrow q$ . Molekuly wyglądają następująco:



Ryc. 86a: molekula reprezentująca klauzulę faktu  $p$

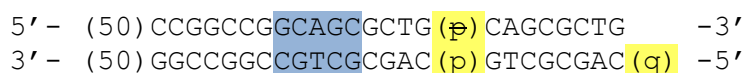


Ryc. 86b: molekula reprezentująca klauzulę  $(\neg p \vee q)$

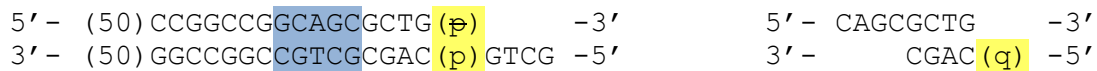


Ryc. 86c: molekula terminalna wnioskowania

Reakcja będzie wyglądać następująco:



Ryc. 87a: ligacja molekuł reprezentujących klauzule  $p$  oraz  $(\neg p \vee q)$



Ryc. 87b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 87c: ligacja molekuly terminalnej i wniosku częściowego

Warto podkreślić, że dodatkowa cząsteczka (powstała po rozcięciu enzymem; patrz: rycina 87b) nie bierze udziału w dalszych reakcjach – w przypadku przyłączenia się do niej innej molekuly, cięcie enzymem nastąpi w tym samym miejscu co poprzednio, więc obie molekuly wrócą do wcześniejszej postaci. Cząsteczka będąca głównym wynikiem reakcji reprezentuje klauzulę faktu  $q$ , czyli wynik wnioskowania (analogicznie jak po zastosowaniu reguły odrywania). Takie cząsteczki mogą brać udział w dalszych reakcjach.

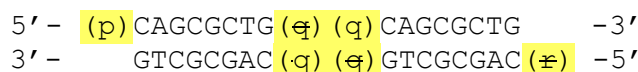
Przeanalizujemy również proste wnioskowanie dla klauzuli zawierającej trzy literały. Weźmy molekuly reprezentujące odpowiednio:  $\neg p$ ,  $r$ ,  $(p \vee q \vee \neg r)$  oraz molekule terminalną. Odpowiada to wnioskowaniom dla reguł:  $\neg p$ ,  $r$ ,  $(\neg p \wedge q) \rightarrow \neg r$ . Molekuly wyglądają następująco:



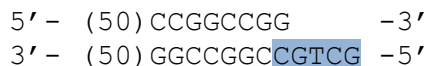
Ryc. 88a: molekula reprezentująca klauzulę faktu  $\neg p$



Ryc. 88b: molekula reprezentująca klauzulę faktu  $r$

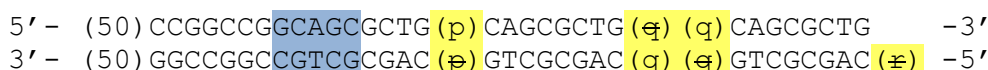


Ryc. 88c: molekula reprezentująca klauzulę  $(p \vee q \vee \neg r)$

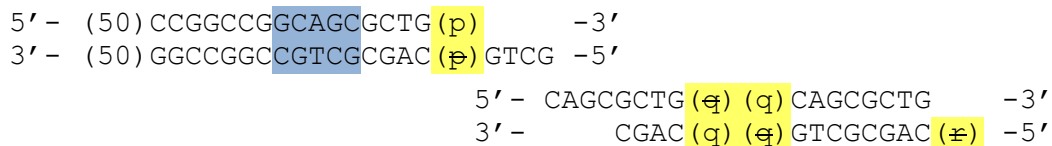


Ryc. 88d: molekula terminalna wnioskowania

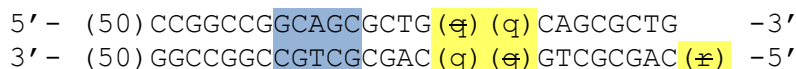
Reakcja będzie wyglądać następująco:



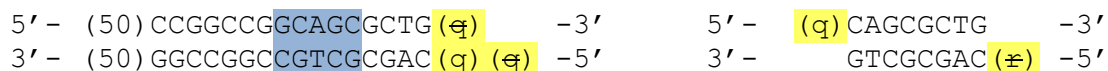
Ryc. 89a: ligacja molekul reprezentujących klauzule  $\neg p$  oraz  $(p \vee q \vee \neg r)$



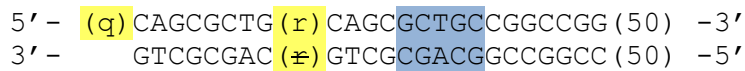
Ryc. 89b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



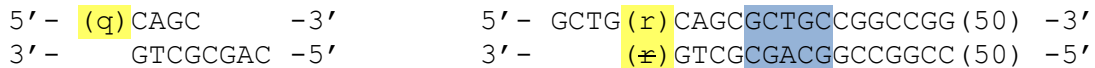
Ryc. 89c: ligacja molekuly terminalnej i wniosku częściowego



Ryc. 89d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 89e: ligacja molekuly reprezentujacej klauzule r i powstalej, reprezentujacej  $(q \vee \neg r)$



Ryc. 89f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 89g: ligacja molekuly terminalnej i wniosku częściowego

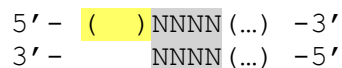
Kolejność niektórych reakcji może być zamieniona. Niektóre z nich mogą zachodzić równocześnie, jednak dla podanych przykładowych molekuł końcowy wniosek, zawsze będzie wyglądał tak samo. Tutaj wynikiem jest cząsteczka reprezentująca klauzulę faktu  $q$ , która oczywiście może brać udział w dalszych reakcjach.

#### 4.2.5. Zapytania

W omawianym systemie możemy zadać pytanie o istnienie dowolnej klauzuli faktu, zarówno takiej która znajdowała się w pierwotnym zbiorze aksjomatów jak i takiej, która została wyprowadzona podczas wnioskowania. Pojedyncze zapytanie może dotyczyć też koniunkcji kilku faktów bądź ich alternatywy, jeżeli użyjemy do jego zapisania więcej molekuł (jedna molekula reprezentuje jeden składnik alternatywy), a nawet dowolnego zdania, zapisanego w dysjunkcyjnej postaci normalnej (*patrz:* podrozdział 1.3). Podczas reakcji w jednej próbówce możemy zadać większą liczbę zapytań i otrzymać osobne odpowiedzi dla każdego z nich.

Molekula zapytania składa się z trzech elementów: stałej sekwencji czterech nukleotydów w formie „lepkiego końca”, reprezentujących literał zanegowany względem tego, o którego możliwość wyprowadzenia w systemie pytamy, co oznacza, że np. dla zapytania o  $p$  potrzebujemy czwórkę reprezentującą  $\neg p$  czyli komplementarną, zaznaczoną poniżej kolorem żółtym), następnie unikalnej (przydzielonej dla danego zapytania) czwórki nukleotydów komplementarnych z samym sobą (po odwróceniu kierunku nici) oraz łańcucha dowolnych nukleotydów o unikalnej długości identyfikującej dane zapytanie. Zastosowanie ciągów komplementarnych „z samym sobą” jest rozwiązaniem innowacyjnym i odgrywa tu kluczową rolę (jednak ze względu na ograniczoną liczbę takich ciągów, w podrozdziale 5.1.2 została dodatkowo przedstawiona alternatywna koncepcja zastosowania pozostałych ciągów czwórek dla zapytań).

Schemat wygląda następująco:



Ryc. 90: schemat molekuly reprezentujacej zapytanie

Weźmy przykładowe dwa różne zapytania:



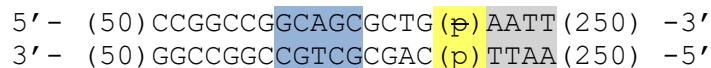
Ryc. 91a: schemat molekuly reprezentujacej zapytanie p?





Ryc. 91b: schemat molekuly reprezentujacej zapytanie  $\neg q$ ?

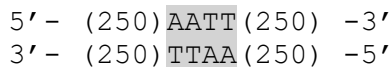
Pierwsze z nich ma unikalną długość 250 dowolnych nukleotydów oraz unikalną komplementarną do siebie czwórkę 3'-TTAA-5' na nici 3'-5'. Drugie ma długość 300 nukleotydów oraz unikalną komplementarną do siebie czwórkę 3'-TATA-5' na nici 3'-5'. Ważne jest, że żaden z tych elementów nie może powtórzyć się dla innego zapytania (zarówno długość molekuly jak i dana czwórka nukleotydów). Odpowiedzi będziemy rozróżniać po długości powstałych molekuly, za pomocą reakcji elektroforezy w żelu. W przypadku odpowiedzi pozytywnej na zapytanie  $p$ ?, otrzymamy nieposiadającą „lepkich końców” (zatem nie biorącą udziału w żadnej dalszej reakcji) molekule długości 504 nukleotydów, a w przypadku odpowiedzi twierdzącej dla  $\neg q$ ? molekule długości 604 nukleotydów (zawsze będzie to dwukrotność unikalnej długości identyfikującej zapytanie, powiększona o 4 nukleotydy jego unikalnej czwórki). Przyjrzyjmy się najprostszej możliwej reakcji z zapytaniem. Weźmy przykład molekuly reprezentujących wyłącznie  $p$  oraz  $p$ ?



Ryc. 92a: ligacja molekuly reprezentujacej  $p$  i molekuly zapytania  $p$ ?



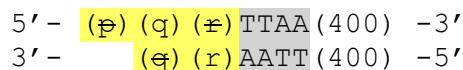
Ryc. 92b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 92c: ligacja powstałych tych samych molekuly ze sobą

W powyższym przykładzie, kluczową rolę odgrywa komplementarność „z samym sobą” unikalnego czteronukleotydowego łańcucha, reprezentującego zapytanie. Właśnie dzięki tej właściwości, w trzecim kroku reakcji następuje ligacja dwóch identycznych powstałych molekuly (należy pamiętać, że w probówce wszystkie molekuly występują wielokrotnie). W efekcie uzyskujemy cząsteczkę bez „lepkich końców”, o konkretnej długości możliwej do identyfikacji (co identyfikuje, którego z zapytań tyczy się uzyskana odpowiedź).

Aby zapisać koniunkcję w zapytaniu, wystarczy rozszerzyć molekule o kolejną czwórkę (lub czwórki) odpowiednio komplementarne do literałów o które pytamy. Zapytanie  $(p \wedge \neg q \wedge r)$ ? może wyglądać następująco:

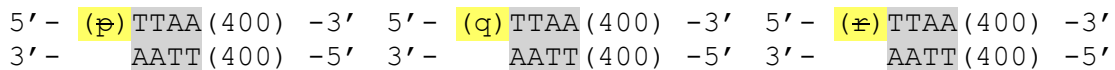


Ryc. 93: schemat molekuly reprezentujacej zapytanie  $(p \wedge \neg q \wedge r)$ ?

Kolejność wspomnianych czwórek nie ma tu żadnego znaczenia, ponieważ odpowiedź twierdzącą uzyskamy tylko jeżeli wszystkie występują w probówce. Dla tak skonstruowanej molekuly zapytania, ewentualna cząsteczka odpowiedzi twierdzącej będzie miała długość 804 nukleotydów.

Aby zapisać alternatywę w zapytaniu, wystarczy zakodować dwie (lub więcej) molekuly zapytania w taki sposób, że posiadają one tę samą czwórkę identyfikującą zapytanie oraz tej

samej długości ciąg nukleotydów dowolnej wartości. Zapytanie  $(p \vee \neg q \vee r)$ ? wygląda następująco:



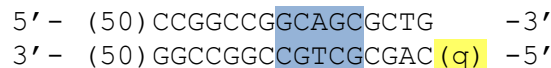
Ryc. 94: schemat molekul wspólnie reprezentujących zapytanie  $(p \vee \neg q \vee r)$ ?

Zgodnie z powyższym schematem, zapisać możemy w formie zapytania dowolne zdanie, po uprzednim przekształceniu go na dysjunkcyjną postać normalną. Zapytanie  $(p \vee (q \wedge \neg r))$ ? wygląda następująco:

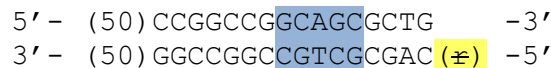


Ryc. 95: schemat molekul wspólnie reprezentujących zapytanie  $(p \vee (q \wedge \neg r))$ ?

Przypadek, w którym użyjemy cząsteczki  $p$  oraz powyższego zapytania, będzie identyczny jak pokazany wcześniej (wyłącznie pierwsza z molekul zapytania weźmie udział w reakcji, analogicznie jak na rycinie 92). Przyjrzyjmy się zatem reakcji, która będzie mieć miejsce dla cząsteczek  $q$ ,  $\neg r$  oraz powyższego zapytania  $(p \vee (q \wedge \neg r))$ ?



Ryc. 96a: molekula reprezentująca klauzulę faktu  $q$

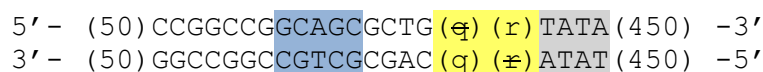


Ryc. 96b: molekula reprezentująca klauzulę faktu  $\neg r$

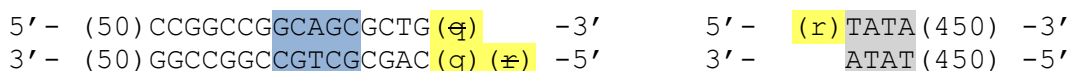


Ryc. 96c: molekuly reprezentujące zapytanie  $(p \vee (q \wedge \neg r))$ ?

Reakcja będzie wyglądać następująco:



Ryc. 97a: ligacja molekul reprezentujących klauzule  $q$  oraz cząsteczkę  $(q \wedge \neg r)$ ?



Ryc. 97b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI (po reakcji z molekulą  $q$ , zapytanie zostało sprowadzone do postaci  $(\neg r)$ ?)



Ryc. 97c: ligacja molekul reprezentujących klauzule  $\neg r$  oraz powstałą cząsteczkę  $(\neg r)$ ?



Ryc. 97d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 97e: ligacja powstałych tych samych molekul ze sobą

W wyniku reakcji otrzymaliśmy cząsteczkę bez „lepkich końców” i o konkretnej długości 904 nukleotydów, która oznacza odpowiedź twierdzącą na wprowadzone zapytanie (ponieważ dowiedzenie poprawności jednego ze składników alternatywy dowodzi poprawności całej

alternatywy). Taką samą cząsteczkę otrzymalibyśmy gdyby w próbówce znajdowała się molekula reprezentująca klauzulę faktu  $p$ . Niezależnie od wszystkich innych istniejących klauzul i równolegle przeprowadzonych reakcji, nie uzyskalibyśmy wspomnianej odpowiedzi przy braku  $p$  bądź przy braku równocześnie występujących  $q$  oraz  $\neg r$ . Gdyby w próbówce znajdowała się wyłącznie molekula reprezentująca  $q$ , reakcja nastąpiłaby częściowo (zapytanie zostałoby poprawnie sprowadzone do prostszej postaci ( $\neg r$ )?, patrz: rycina 97a, 97b), jednak nie pojawiłaby się odpowiedź twierdząca.

### 4. 3. Wykrycie sprzeczności

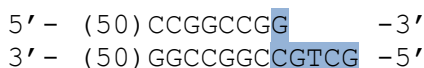
Wprowadzenie do systemu spójnika negacji (co jest zasadniczą zmianą względem systemów omówionych w podrozdziałach 3.1, 3.2 oraz 3.3) jest dużą zaletą, ale niesie też ze sobą konieczność rozpatrzenia sytuacji pojawienia się w tej samej próbówce przeciwnych sobie faktów (np.  $p$  oraz  $\neg p$ ). Omawiany system w odpowiedni sposób taką ewentualność zasygnalizuje. Ze względu na komplementarność unikalnych czwórek reprezentujących literały o przeciwnych wartościach tego samego zdania atomowego, w przypadku pojawienia się ich w tej samej próbówce nastąpi między nimi reakcja, którą nazywamy reakcją wykrycia sprzeczności. Katalizatorem tej reakcji jest cząsteczka terminalna, ta sama która bierze udział w reakcjach wnioskowania. Z matematycznego punktu widzenia, jeżeli w danym systemie logicznym pojawią się zaprzeczające sobie fakty, to możemy z niego wyprowadzić wszystko – w takiej sytuacji odpowiedź o pojawieniu się sprzeczności, możemy również potraktować jako pozytywną na wszystkie zadawane pytania. Rozważmy najprostszy przykład umieszczenia w próbówce przeciwnych literałów  $p$  oraz  $\neg p$ :



Ryc. 98a: molekula reprezentująca klauzulę faktu  $p$

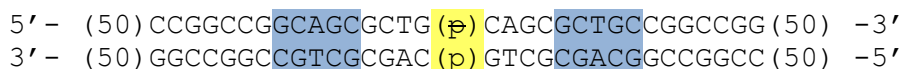


Ryc. 98b: molekula reprezentująca klauzulę faktu  $\neg p$

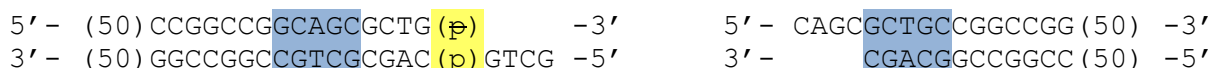


Ryc. 98c: molekula terminalna wnioskowania

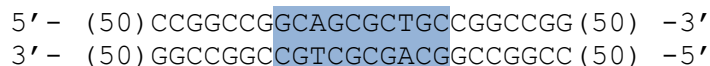
Reakcja będzie wyglądać następująco:



Ryc. 99a: ligacja molekul reprezentujących klauzule  $p$  oraz  $\neg p$



Ryc. 99b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 99c: ligacja molekuly terminalnej i powstałej cząsteczki



Ryc. 99d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 99e: ligacja powstałych molekuł ze sobą

Powstała cząsteczka niezawierająca „lepkich końców” oraz o unikalnej długości 104 nukleotydów oznacza wystąpienie sprzeczności w systemie. Zależnie od podejścia, możemy potraktować ją jako pozytywną odpowiedź na wszystkie zadane pytania (ponieważ ze sprzeczności możemy wyprowadzić dowolny wniosek), bądź jako informację o braku wiarygodności pozostałych odpowiedzi systemu. Schemat takiej reakcji będzie identyczny dla wszystkich pozostałych zdań atomowych. Oczywiście literały o różnych wartościach mogą być wnioskami powstałymi w wyniku wcześniejszych reakcji (przy skomplikowanym zbiorze aksjomatów możemy nie dostrzec ewentualnej sprzeczności przed rozpoczęciem obliczeń).

Jako konsekwencję przyjęcia długości 104 nukleotydów jako informację o wystąpieniu sprzeczności, nie możemy konstruować molekuł zapytań zawierających tylko 50 dowolnych nukleotydów na końcu, gdyż to spowodowałoby pojawienie się również 104 w przypadku odpowiedzi twierdzącej na dane zapytanie. Przyjmujemy zatem dowolne wartości większe lub równe 100 (w pojedynczej cząsteczce) dla molekuł zapytań.

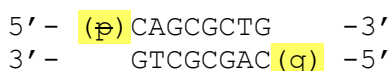
Weźmy inny przykład, zawierający klauzulę dwuelementową:  $p$ ,  $\neg q$ ,  $(\neg p \vee q)$ :



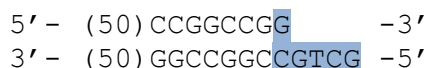
Ryc. 100a: molekula reprezentująca klauzulę faktu  $p$



Ryc. 100b: molekula reprezentująca klauzulę faktu  $\neg q$

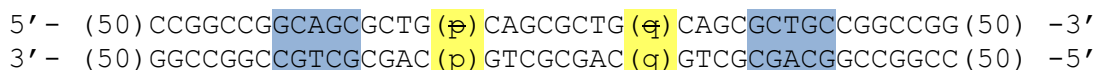


Ryc. 100c: schemat molekuly reprezentującej klauzulę  $(\neg p \vee q)$

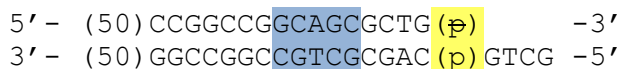


Ryc. 100d: molekula terminalna wnioskowania

Ze względu na równoległość obliczeń zachodzących w probówce, możliwości wykonania tej reakcji jest kilka (a nawet wszystkie mogą zajść równocześnie). Może nastąpić najpierw reakcja wnioskowania, według schematu przedstawionego w podrozdziale 4.2.4, w wyniku której oderwaniu nastąpią cząsteczki  $q$  i  $\neg p$ , a w konsekwencji reakcja wykrycia sprzeczności identyczna jak powyżej. Może też nastąpić równoczesna ligacja wszystkich trzech molekuł, która wygląda następująco:



Ryc. 101a: ligacja molekuł reprezentujących klauzule  $p$  oraz  $\neg q$  oraz  $(\neg p \vee q)$



Ryc. 101b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI

5' - (50) CCGGCCG **GCAGCGCTG** (ϕ) CAGC **GCTGC** CCGGCCG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (ϑ) GTCG **CGACG** GCCGGCC (50) -5'

Ryc. 101c: ligacja molekuly terminalnej i powstałej cząsteczki

5' - (50) CCGGCCG **GCAGCGCTG** (ϕ) -3'                      5' - CAGC **GCTGC** CCGGCCG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (ϑ) GTCG -5'                      3' - **CGACG** GCCGGCC (50) -5'

Ryc. 101d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI

5' - (50) CCGGCCG **GCAGCGCTGC** CCGGCCG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGACG** GCCGGCC (50) -5'

Ryc. 101e: ligacja molekuly terminalnej i powstałej cząsteczki

5' - (50) CCGGCCG **GCAGCGCTGC** CCGG -3'                      5' - CCGG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGACG** GCCGGCC -5'                      3' - (50) -5'

Ryc. 101f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI

5' - (50) CCGG (50) -3'  
 3' - (50) GGCC (50) -5'

Ryc. 101g: ligacja powstałych molekuł ze sobą

Otrzymana cząsteczka niezawierająca „lepkich końców” oraz o tej samej unikalnej długości 104 nukleotydów jest identyczna jak w przypadku przedstawionej wcześniej reakcji i oznacza wystąpienie sprzeczności w systemie.

#### 4. 4. Działanie systemu wnioskującego

W odniesieniu do modeli komputerów DNA, omówionych w rozdziałach 2 i 3, możemy potraktować rozszerzony system wnioskowania jako w pełni autonomiczny, a jego działanie (po sprowadzeniu aksjomatów wejściowych do koniunkcyjnej postaci normalnej, a zapytań do dysjunkcyjnej postaci normalnej, *patrz*: podrozdział 1.3) sprowadzić do trzech podstawowych kroków:

1. Kodowanie aksjomatów wejściowych oraz zapytań za pomocą odpowiednich molekuł DNA według ustalonego i omówionego powyżej schematu.
2. Przeprowadzenie reakcji w próbówce.
3. Odczytanie wyników za pomocą elektroforezy w żelu.

Odpowiedź systemu rozpoznajemy poprzez pojawienie się (lub nie) w próbówce molekuly o konkretnej długości i bez „lepkich końców”. Istnieją trzy możliwości:

1. Pojawienie się molekuly o konkretnej długości, odpowiadającej konkretnemu zapytaniu (a dokładnie dwukrotności unikalnej długości zapytania, powiększonej o 4 nukleotydy identyfikujące dane zapytanie, np. dla zapytania o unikalnej długości 100 będzie to 204, a dla 250 będzie to 504) – oznacza odpowiedź twierdzącą systemu na dane zapytanie.
2. Pojawienie się molekuly o długości 104 nukleotydów – oznacza pojawienie się sprzeczności w systemie.
3. Brak powyższych molekuł – oznacza, że danego wniosku nie udało się wyprowadzić na podstawie danych aksjomatów początkowych.

W przypadku zadania większej liczby pytań jednocześnie, powyższe możliwości muszą zostać przeanalizowane dla każdego z nich z osobna.

## 4. 5. Przykłady złożonego wnioskowania

Weźmy bardziej skomplikowany przykład reakcji, zaczerpnięty z [47] (omówiony już wcześniej dla systemów niezawierających spójnika negacji, *patrz*: podrozdział 3.1 oraz 3.2). Weźmy następujące aksjomaty początkowe:  $a$ ,  $d$ ,  $u$ ,  $((a \wedge b) \vee d) \rightarrow c$ ,  $((c \wedge u) \rightarrow x)$  oraz zapytanie  $x?$ . Po sprowadzeniu do koniunkcyjnej postaci normalnej, aksjomaty będą wyglądać następująco:  $(a \wedge d \wedge u \wedge (\neg a \vee \neg b \vee c) \wedge (\neg d \vee c) \wedge (\neg c \vee \neg u \vee x))$ .



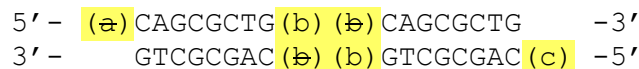
Ryc. 102a: molekula reprezentująca klauzulę faktu  $a$



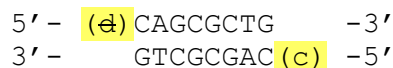
Ryc. 102b: molekula reprezentująca klauzulę faktu  $d$



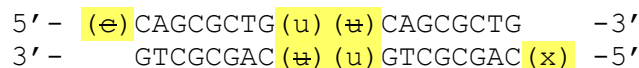
Ryc. 102c: molekula reprezentująca klauzulę faktu  $u$



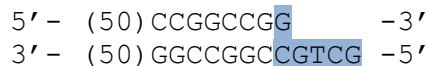
Ryc. 102d: molekula reprezentująca klauzulę  $(\neg a \vee \neg b \vee c)$



Ryc. 102e: molekula reprezentująca klauzulę  $(\neg d \vee c)$



Ryc. 102f: molekula reprezentująca klauzulę  $(\neg c \vee \neg u \vee x)$

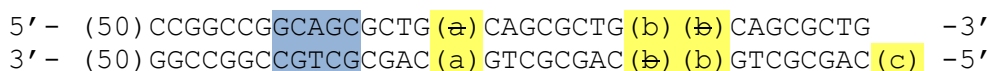


Ryc. 102g: molekula terminalna wnioskowania

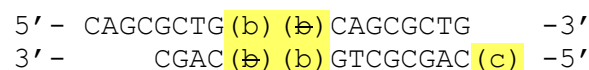
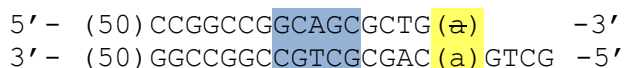


Ryc. 102h: molekula reprezentująca zapytanie  $x?$

Reakcja będzie wyglądać następująco:



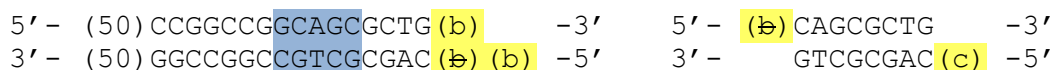
Ryc. 103a: ligacja molekuł reprezentujących klauzule  $a$  oraz  $(\neg a \vee \neg b \vee c)$



Ryc. 103b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



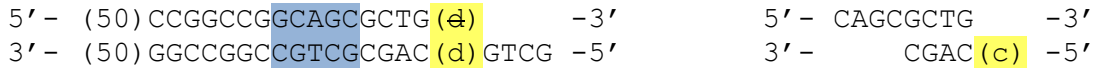
Ryc. 103c: ligacja molekuly terminalnej i wniosku częściowego



Ryc. 103d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI (powstaje molekula reprezentująca klauzulę  $(\neg b \vee c)$ )



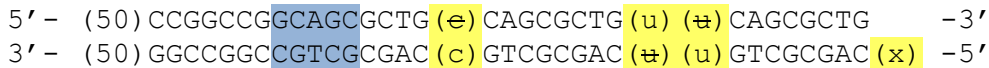
**Ryc. 103e: ligacja molekuł reprezentujących klauzule d oraz ( $-d \vee c$ )**



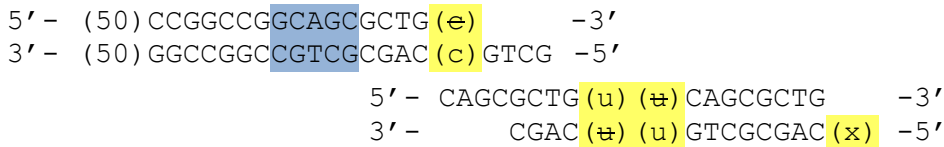
**Ryc. 103f: rozcięcie molekuły za pomocą enzymu restrykcyjnego BbvI**



**Ryc. 103g: ligacja molekuły terminalnej i wniosku częściowego (powstaje molekuła reprezentująca klauzulę faktu c)**



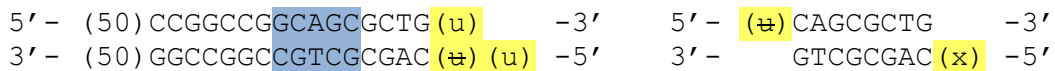
**Ryc. 103h: ligacja molekuł reprezentujących klauzule c oraz ( $-\text{c} \vee -\text{u} \vee \text{x}$ )**



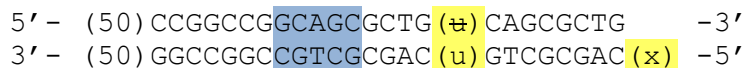
**Ryc. 103i: rozcięcie molekuły za pomocą enzymu restrykcyjnego BbvI**



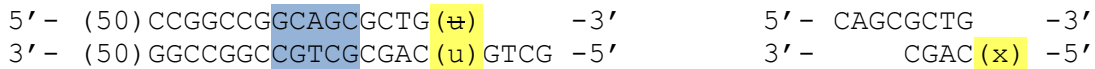
**Ryc. 103j: ligacja molekuły terminalnej i wniosku częściowego**



**Ryc. 103k: rozcięcie molekuły za pomocą enzymu restrykcyjnego BbvI (powstaje molekuła reprezentująca klauzulę ( $-\text{u} \vee \text{x}$ ))**



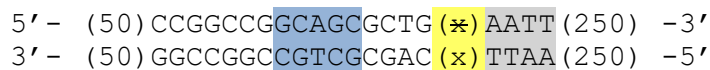
**Ryc. 103l: ligacja molekuł reprezentujących klauzule u oraz ( $-\text{u} \vee \text{x}$ )**



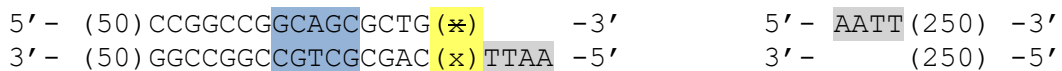
**Ryc. 103m: rozcięcie molekuły za pomocą enzymu restrykcyjnego BbvI**



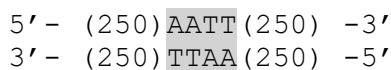
**Ryc. 103n: ligacja molekuły terminalnej i wniosku częściowego (powstaje molekuła reprezentująca klauzulę faktu x)**



**Ryc. 103o: ligacja molekuły reprezentującej x i molekuły zapytania x?**



**Ryc. 103p: rozcięcie molekuły za pomocą enzymu restrykcyjnego BbvI**



**Ryc. 103q: ligacja powstałych molekuł ze sobą**

W wyniku reakcji otrzymaliśmy cząsteczkę bez „lepkich końców” i o konkretnej długości 504 nukleotydów, która oznacza odpowiedź twierdzącą na wprowadzone zapytanie  $x$ ? czyli informuje, że na podstawie wprowadzonych aksjomatów wejściowych, zdanie  $x$  jest możliwe do wyprowadzenia.

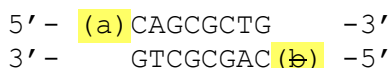
Przyjrzyjmy się jeszcze reakcji wykorzystującej jeszcze więcej odniesień do klasycznego rachunku zdań i jego tautologii. Weźmy następujące aksjomaty początkowe w koniunkcyjnej postaci normalnej:  $((a \vee \neg b) \wedge (b \vee c \vee d) \wedge (a \vee d) \wedge \neg c \wedge \neg d$  oraz zapytanie  $b$ ?:



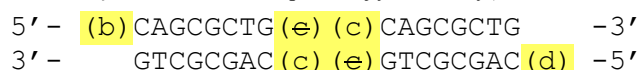
*Ryc. 104a: molekula reprezentująca klauzulę faktu  $\neg c$*



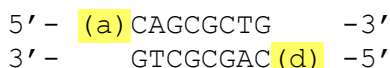
*Ryc. 104b: molekula reprezentująca klauzulę faktu  $\neg d$*



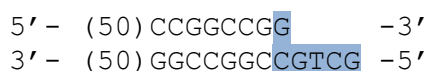
*Ryc. 104c: molekula reprezentująca klauzulę  $(a \vee \neg b)$*



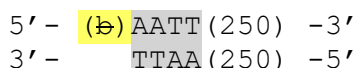
*Ryc. 104d: molekula reprezentująca klauzulę  $(b \vee c \vee d)$*



*Ryc. 104e: molekula reprezentująca klauzulę  $(a \vee d)$*

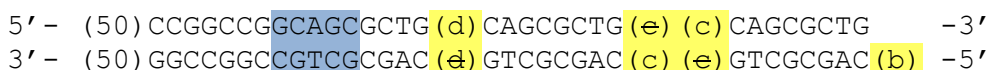


*Ryc. 104f: molekula terminalna wnioskowania*

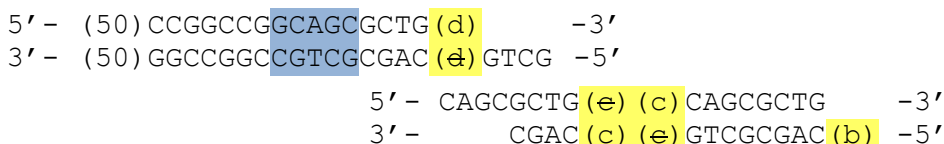


*Ryc. 104g: molekula reprezentująca zapytanie  $b$ ?*

Reakcja będzie wyglądać następująco:



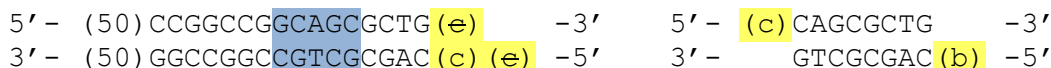
*Ryc. 105a: ligacja molekuł reprezentujących klauzule  $\neg d$  oraz  $(b \vee c \vee d)$*



*Ryc. 105b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 105c: ligacja molekuly terminalnej i wniosku częściowego*

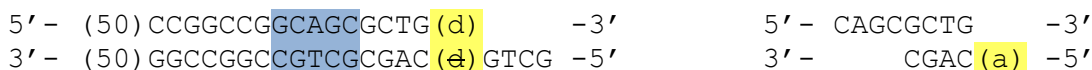


*Ryc. 105d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*

*(powstaje molekula reprezentująca klauzulę  $(b \vee c)$ )*



*Ryc. 105e: ligacja molekuł reprezentujących klauzule  $\neg d$  oraz  $(a \vee d)$*



*Ryc. 105f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



5' - (50) CCGGCCG **GCAGCGCTG** -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (a) -5'

**Ryc. 105g: ligacja molekuly terminalnej i wniosku częściowego**  
 (powstaje molekula reprezentująca klauzulę faktu a)

5' - (a) CAGCGCTG (b) CAGCGCTG (e) (c) CAGCGCTG -3'  
 3' - GTCGCGAC (b) GTCGCGAC (c) (e) GTCGCGAC (d) -5'

**Ryc. 105h: ligacja molekul reprezentujących klauzule (a ∨ -b) oraz (b ∨ c ∨ d)**

5' - (50) CCGGCCG **GCAGCGCTG** (a) CAGCGCTG (b) CAGCGCTG (e) (c) CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (a) GTCGCGAC (b) GTCGCGAC (c) (e) GTCGCGAC (d) -5'

**Ryc. 105i: ligacja molekuly reprezentującej klauzulę faktu a oraz powyższej molekuly**

5' - (50) CCGGCCG **GCAGCGCTG** (a) -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (a) GTCG -5'

5' - CAGCGCTG (b) CAGCGCTG (e) (c) CAGCGCTG -3'  
 3' - CGAC (b) GTCGCGAC (c) (e) GTCGCGAC (d) -5'

**Ryc. 105j: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI**

5' - (50) CCGGCCG **GCAGCGCTG** (b) CAGCGCTG (e) (c) CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (b) GTCGCGAC (c) (e) GTCGCGAC (d) -5'

**Ryc. 105k: ligacja molekuly terminalnej i wniosku częściowego**

5' - (50) CCGGCCG **GCAGCGCTG** (b) -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (b) GTCG -5'

5' - CAGCGCTG (e) (c) CAGCGCTG -3'  
 3' - CGAC (c) (e) GTCGCGAC (d) -5'

**Ryc. 105l: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI**

5' - (50) CCGGCCG **GCAGCGCTG** (e) (c) CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (c) (e) GTCGCGAC (d) -5'

**Ryc. 105m: ligacja molekuly terminalnej i wniosku częściowego**

5' - (50) CCGGCCG **GCAGCGCTG** (e) -3'      5' - (c) CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (c) (e) -5'      3' - GTCGCGAC (d) -5'

**Ryc. 105n: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI**  
 (powstaje molekula reprezentująca klauzulę (c ∨ d))

5' - (50) CCGGCCG **GCAGCGCTG** (d) CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (d) GTCGCGAC (c) -5'

**Ryc. 105o: ligacja molekul reprezentujących klauzule -d oraz (c ∨ d)**

5' - (50) CCGGCCG **GCAGCGCTG** (d) -3'      5' - CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (d) GTCG -5'      3' - CGAC (c) -5'

**Ryc. 105p: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI**

5' - (50) CCGGCCG **GCAGCGCTG** -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (c) -5'

**Ryc. 105q: ligacja molekuly terminalnej i wniosku częściowego**  
 (powstaje molekula reprezentująca klauzulę faktu c)

5' - (50) CCGGCCG **GCAGCGCTG** (e) CAGC **GCTGCC** CGGCCGG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (c) GTCG **CGAC** CGGCCGGCC (50) -5'

**Ryc. 105r: ligacja molekul reprezentujących klauzule c oraz -c**

5' - (50) CCGGCCG **GCAGCGCTG** (e) -3'      5' - CAGC **GCTGCC** CGGCCGG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** (c) GTCG -5'      3' - **CGAC** CGGCCGGCC (50) -5'

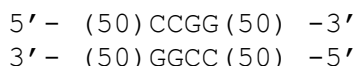
**Ryc. 105s: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI**

5' - (50) CCGGCCG **GCAGCGCTG** CGGCCGG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** CGGCCGGCC (50) -5'

**Ryc. 105t: ligacja molekuly terminalnej i powstałej cząsteczki**



Ryc. 105u: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 105v: ligacja powstałych molekuł ze sobą

Powstała cząsteczka niezawierająca „lepkich końców” oraz o unikalnej długości 104 nukleotydów oznacza wystąpienie sprzeczności w systemie dla podanych aksjomatów początkowych. Jest to celowo podany przykład rozumowania, gdzie na pierwszy rzut oka nie widać, że wspomniana sprzeczność wystąpi.

## 4. 6. Odniesienie do klasycznego rachunku zdań

Taki system uwzględnia większość praw klasycznego rachunku zdań w sposób automatyczny i dokonuje zgodnego z nimi wnioskowania (ograniczenia zostaną omówione w rozdziale 5.1). Przyjrzyjmy się jak uwzględniane są konkretne tautologie.

### 4. 6. 1. Podwójne zaprzeczenie

Jedno z najważniejszych praw klasycznego rachunku zdań mówi o podwójnym zaprzeczeniu:

$$\neg(\neg p) \leftrightarrow p.$$

Prawo to, które zgodnie z przyjętymi założeniami kodowania zmiennych zdaniowych i ich wartości, uwzględniane jest w omawianym rozszerzonym systemie wnioskowania w sposób automatyczny (zgodnie z omówioną w rozdziale 2 regułą komplementarności). Mianowicie, przyjmując czwórkę komplementarną jako charakterystyczną dla negacji danego zdania atomowego otrzymamy, że czwórką charakterystyczną dla podwójnej negacji będzie wyjściowa czwórka. Na przykład jeżeli przyjmiemy ciąg nukleotydów 3'-AAAC-5' jako reprezentujący  $p$ , to łańcuch do niego komplementarny ma postać 3'-GTTT-5'. Zatem ciąg nukleotydów 3'-GTTT-5' przyjmujemy jako reprezentujący  $\neg p$ , a wtedy łańcuch do niego komplementarny, który będzie reprezentować  $\neg(\neg p)$  ma postać 3'-AAAC-5' i jest identyczny z tym reprezentującym  $p$ , od którego wyszliśmy.

### 4. 6. 2. Modus ponens

Regułę odrywania, inaczej nazywaną *modus ponens*, w klasycznym rachunku zdań przedstawiamy następująco: jeżeli uznajemy prawdziwość implikacji, a także prawdziwość jej poprzednika, to musimy uznać również prawdziwość jej następnika. Odpowiada to następującej tautologii:

$$(p \wedge (p \rightarrow q)) \rightarrow q,$$

która po zapisaniu w koniunkcyjnej postaci normalnej, wygląda następująco:

$$(p \wedge (\neg p \vee q)) \rightarrow q,$$

a jako podstawowa reguła wnioskowania w omawianym systemie, została szczegółowo opisana wcześniej (*patrz*: podrozdziały 1.2 oraz 4.2.4).

### 4. 6. 3. Modus tollens

Regułę odwrotną do *modus ponens* przedstawiamy następująco: jeżeli uznajemy prawdziwość implikacji, a także prawdziwość negacji jej następnika, to musimy uznać również prawdziwość negacji jej poprzednika. Odpowiada to następującej tautologii:

$$(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p,$$

która po zapisaniu w koniunkcyjnej postaci normalnej, wygląda następująco:

$$(\neg q \wedge (\neg p \vee q)) \rightarrow \neg p.$$

Z pomocą reguły kontrapozycji omawiany system w sposób automatyczny dokonuje również wnioskowania odwrotnego do *modus ponens* czyli *modus tollens*, co odróżnia go od znanych i omówionych wcześniej systemów wnioskowania za pomocą DNA.

### 4. 6. 4. Reguła kontrapozycji

Dzięki regule kontrapozycji istnieje możliwość wnioskowania za pomocą *modus ponens* w „obie strony” (czyli w rzeczywistości automatycznej implementacji *modus tollens*). Jest to konsekwencja wprowadzenia spójnika negacji i odpowiedniego molekularnego kodowania zdań. Stosowanie jej odróżnia oraz znacznie rozszerza omawiany tutaj system wnioskujący od tych znanych z wcześniejszych implementacji (między innymi [27], [47], *patrz*: rozdział 3). Możliwość reakcji „w dwie strony” jest rozwiązaniem nowatorskim i została zrealizowana poprzez zastosowanie unikalnych symetrycznych molekuł reprezentujących klauzule niebędące klauzulami faktu, dzięki czemu reakcja odrywania zachodzi poprawnie niezależnie od tego, który z „lepkich końców” molekuly bierze w niej udział (szczegółowy przykład reakcji poniżej). Reguła kontrapozycji wygląda następująco:

$$(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p),$$

a włączenie jej do rozszerzonego systemu wnioskującego zostało po raz pierwszy przedstawione w jednej z poprzednich prac [32]. Po przekształceniu do postaci CNF, odpowiada ona regule przemienności alternatywy:

$$(\neg p \vee q) \leftrightarrow (q \vee \neg p).$$

Przy używanym sposobie kodowania jedna molekula reprezentuje zarówno lewą jak i prawą stronę powyższej równoważności. Zatem możemy uzyskać różne wnioski zależnie od pozostałych aksjomatów wejściowych – w przypadku istnienia literału *p*, będzie to wniosek *q* i analogicznie w przypadku istnienia literału  $\neg q$ , będzie to wniosek  $\neg p$ . Takie podejście jest możliwe dzięki zastosowaniu symetrycznych molekuł kodowania w systemie splatania.

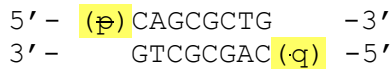
Zilustrujmy te dwie możliwości dla omawianej klauzuli  $(\neg p \vee q)$ , z wykorzystaniem (niejednocześnie) następujących molekuł:



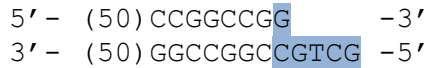
Ryc. 106a: molekula reprezentująca klauzulę faktu *p*



Ryc. 106b: molekula reprezentująca klauzulę faktu  $\neg q$



Ryc. 106c: molekula reprezentująca klauzulę  $(\neg p \vee q)$

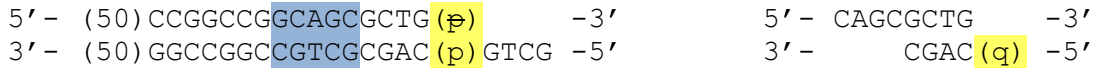


Ryc. 106d: molekula terminalna wnioskowania

Reakcja dla  $p$  oraz  $(\neg p \vee q)$  wygląda następująco:



Ryc. 107a: ligacja molekul reprezentujących klauzule  $p$  oraz  $(\neg p \vee q)$

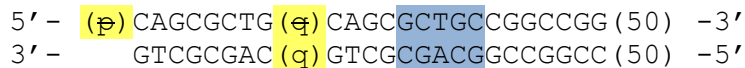


Ryc. 107b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI

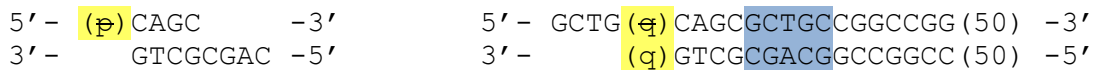


Ryc. 107c: ligacja molekuly terminalnej i wniosku częściowego

Reakcja dla  $\neg q$  oraz  $(\neg p \vee q)$  wygląda następująco:



Ryc. 108a: ligacja molekul reprezentujących klauzule  $\neg q$  oraz  $(\neg p \vee q)$



Ryc. 108b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 108c: ligacja molekuly terminalnej i wniosku częściowego (zapisana już po „odwróceniu”)

W obu przypadkach końcowym wnioskiem jest molekula reprezentująca klauzulę faktu (odpowiednio  $q$  lub  $\neg p$ ), która oczywiście może brać udział w dalszych reakcjach.

#### 4. 6. 5. Przechodność implikacji

Jedno z praw klasycznego rachunku zdań mówi o przechodności implikacji:

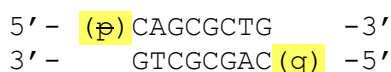
$$((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r),$$

a po zapisaniu obu stron w koniunkcyjnej postaci normalnej uzyskujemy zależność:

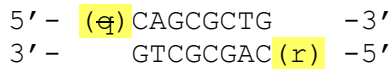
$$((\neg p \vee q) \wedge (\neg q \vee r)) \rightarrow (\neg p \vee r),$$

która ze względu na komplementarność czwórek reprezentujących przeciwne wartości tego samego zdania atomowego, będzie automatycznie spełniona w omawianej molekularnej implementacji systemu wnioskującego.

Weźmy molekuly reprezentujące przedstawione przykładowe klauzule  $(\neg p \vee q)$  oraz  $(\neg q \vee r)$ :

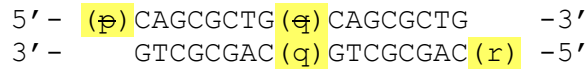


Ryc. 109a: schemat molekuly reprezentującej klauzulę  $(\neg p \vee q)$



Ryc. 109b: schemat molekuly reprezentujacej klauzule  $(\neg q \vee r)$

Ze względu na komplementarnosc, razem utworzają one nastepujacą molekule:

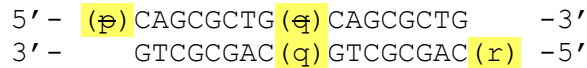


Ryc. 110: schemat molekuly reprezentujacej klauzule  $(\neg p \vee r)$

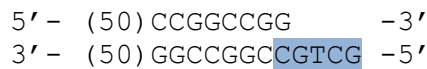
Powyzsza molekula faktycznie reprezentuje klauzule  $(\neg p \vee r)$ , a w przypadku istnienia w tej samej probowce klauzuli faktu reprezentujacej  $p$  lub klauzuli faktu reprezentujacej  $\neg r$ , nastapi poprawna reakcja wnioskowania. Weźmy omawiana molekule, molekule reprezentujaca  $p$  oraz molekule terminalna:



Ryc. 111a: molekula reprezentujaca klauzule faktu  $p$

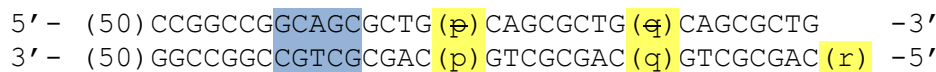


Ryc. 111b: molekula reprezentujaca klauzule  $(\neg p \vee r)$  (po uwzględnieniu przechodnosci)

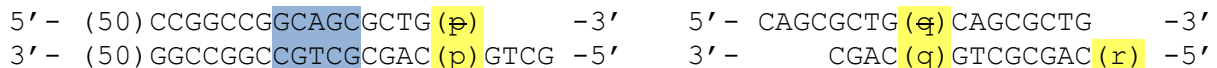


Ryc. 111c: molekula terminalna wnioskowania

Reakcja bedzie wygladac nastepujaco:



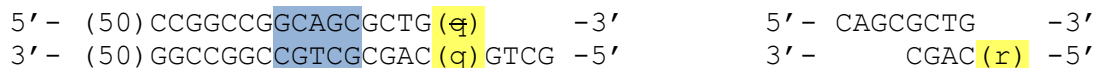
Ryc. 112a: ligacja molekul reprezentujacych klauzule faktu  $p$  oraz klauzule  $(\neg p \vee r)$



Ryc. 112b: rozcięcie molekuly za pomoc enzymu restrykcyjnego BbvI



Ryc. 112c: ligacja molekuly terminalnej i powstalej czasteczki



Ryc. 112d: rozcięcie molekuly za pomoc enzymu restrykcyjnego BbvI



Ryc. 112e: ligacja molekuly terminalnej i powstalej czasteczki

Czasteczka bedaca wynikiem reakcji reprezentuje klauzule faktu  $r$ , czyli wynik wnioskowania. Wszystkie czasteczki moga brac udzial w dalszych reakcjach.

#### 4. 6. 6. Prawa de Morgana

Prawa de Morgana w klasycznym rachunku zdań podaja sposob zaprzeczania koniunkcji oraz zaprzeczania alternatywy. Są to:

$$\neg(p \wedge q) \leftrightarrow (\neg p \vee \neg q) \quad (\text{I prawo de Morgana}),$$

$$\neg(p \vee q) \leftrightarrow (\neg p \wedge \neg q) \quad (\text{II prawo de Morgana}),$$

co w przełożeniu na system wnioskowania oparty na regułach *modus ponens* i *modus tollens* sprowadza się do wykorzystania ich w połączeniu z regułą kontrapozycji implikacji. Zatem jeżeli w prawie kontrapozycji:

$$(p \rightarrow q) \leftrightarrow (\neg q \rightarrow \neg p),$$

w miejsce  $p$  i  $q$  chcielibyśmy podstawić koniunkcję i/lub alternatywę to np. dla  $p = (p_1 \wedge p_2)$  oraz  $q = (q_1 \vee q_2)$  uzyskamy następującą serię wyrażeń równoważnych (po opuszczeniu podwójnych negacji, a następnie sprowadzeniu do koniunkcyjnej postaci normalnej):

$$((p_1 \wedge p_2) \rightarrow (q_1 \vee q_2)) \leftrightarrow (\neg(q_1 \vee q_2) \rightarrow \neg(p_1 \wedge p_2)),$$

$$((p_1 \wedge p_2) \rightarrow (q_1 \vee q_2)) \leftrightarrow ((\neg q_1 \wedge \neg q_2) \rightarrow (\neg p_1 \vee \neg p_2)),$$

$$((\neg p_1 \vee \neg p_2) \vee (q_1 \vee q_2)) \leftrightarrow ((q_1 \vee q_2) \vee (\neg p_1 \vee \neg p_2)),$$

$$(\neg p_1 \vee \neg p_2 \vee q_1 \vee q_2) \leftrightarrow (q_1 \vee q_2 \vee \neg p_1 \vee \neg p_2),$$

gdzie obie strony są identyczne. Oznacza to, że analogicznie jak w przypadku reguły kontrapozycji, działania zgodne z prawami de Morgana wykonują się automatycznie. W przypadku podstawienia wyłącznie w miejsce  $p$  lub wyłącznie  $q$ , a także podstawienie koniunkcji lub alternatywy składającej się z dowolnej liczby składników, dla ogólniejszej postaci:

$$\neg(p_1 \wedge \dots \wedge p_n) \leftrightarrow (\neg p_n \vee \dots \vee \neg p_1) \text{ (I prawo de Morgana),}$$

$$\neg(p_1 \vee \dots \vee p_n) \leftrightarrow (\neg p_n \wedge \dots \wedge \neg p_1) \text{ (II prawo de Morgana),}$$

działania również wykonują się poprawnie. Analogicznie jak dla reguły kontrapozycji, możemy uzyskać różne wnioski zależnie od pozostałych aksjomatów wejściowych, również dzięki zastosowaniu symetrycznych molekuł kodowania w systemie splatania.

#### 4. 7. Podsumowanie

W powyższym rozdziale zostały szczegółowo przedstawione wszystkie składniki nowego systemu wnioskującego, wraz ze szczegółowym schematem ich kodowania i przykładami zastosowania. Szczególnie zwrócona została uwaga na odniesienie do praw klasycznego rachunku zdań i zastosowania w systemie znanych tautologii.

Uzupełnieniem tego rozdziału są rozdziały 5 i 6. Ograniczenia systemu wnioskowania omówione zostaną w podrozdziale 5.1, dodatkowe zastosowania w podrozdziale 5.2, a ewentualne modyfikacje zwiększające możliwości obliczeniowe w podrozdziale 5.3. W rozdziale 6 szczegółowo omówiony zostanie schemat przygotowania właściwego doświadczenia w laboratorium. Propozycje konkretnych eksperymentów zostaną przedstawione w podrozdziale 6.1, szczegółowe przygotowanie cząsteczek DNA (wraz z możliwością ich dalszego przechowywania, namnażania i wykorzystania dla kolejnych eksperymentów) w podrozdziale 6.2, a właściwy przebieg reakcji i odczytywanie jej wyników w podrozdziale 6.3.

## 5. Analiza rozszerzonego systemu

W poniższym rozdziale zostanie przedstawiona analiza omawianego systemu pod względem jego ograniczeń oraz poprawności działania. Zostaną przedstawione też jego rozszerzenia i dodatkowe zastosowania (w tym nawiązujące do koncepcji przedstawionych w rozdziale 3, ale wykorzystujące wyłącznie konstrukcję molekuł przedstawioną w rozdziale 4). Mianowicie: obliczanie wartości funkcji logicznych (odniesienie do podrozdziału 3.1.2), częściowe rozwiązanie problemu SAT (odniesienie do podrozdziału 3.4) czy wykorzystanie koncepcji dowodu nie-wprost jako metody dowodzenia (odniesienie do podrozdziału 3.3). Zakończeniem pracy będzie analiza zwiększenia możliwości systemu w przypadku odkrycia nowych, nieznanych dotąd enzymów restrykcyjnych (*patrz*: podrozdział 5.3.2) i możliwości zastosowania fluorescencji, znanej z doświadczeń przeprowadzonych w [27] (*patrz*: podrozdział 5.3.3).

### 5. 1. Ograniczenia systemu

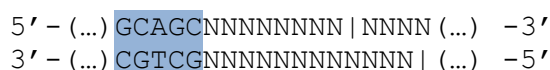
Komputery biomolekularne podlegają wielu ograniczeniom ze względów biochemicznych. Jednym z nich jest długość „lepkich końców” pozostawianych przez działanie enzymów restrykcyjnych. Przy obecnej wiedzy biochemicznej długość ta jest ograniczona do co najwyżej sześciu nukleotydów (jednak w praktyce enzymy pozostawiające „lepkie końce” długości czterech i mniej nukleotydów mają większe zastosowanie ze względu na sekwencje przez nie rozpoznawalne i odległości cięcia). Ponadto należy zadbać aby ciąg molekuł rozpoznawalny przez dany enzym restrykcyjny nie pojawił się przypadkiem w niepożądanym miejscu. Istnieją też pojedyncze przypadki ograniczeń, których nie da się całkowicie wyeliminować.

Powyższe ograniczenia i ich wpływ na działanie omawianego w tej pracy systemu wnioskowania zostaną omówione w kolejnych podpunktach tego podrozdziału.

#### 5. 1. 1. Liczba zmiennych możliwych do zakodowania

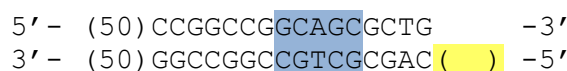
W omawianej implementacji systemu wnioskującego do zakodowania pojedynczej zmiennej posługujemy się jednoniciowymi łańcuchami o długości czterech nukleotydów. Łańcuchy takie możemy podzielić na dwie grupy: unikalne czwórki, które są komplementarne ze sobą (np. 3'-AATT-5' na komplementarnej nici będzie mieć postać 5'-TTAA-3', a zatem połączy się z drugim identycznym łańcuchem) oraz pozostałe, które w celu utworzenia podwójnej helisy potrzebują innej (komplementarnej) czwórki. Te pierwsze wykorzystane są do kodowania zapytań (zostaną omówione w kolejnym podrozdziale), natomiast te drugie wykorzystujemy do zakodowania zmiennych w taki sposób, że łańcuch komplementarny oznacza zaprzeczenie tej samej zmiennej (np. jeżeli przyjmujemy dla zmiennej *a* kod 3'-AATA-5', to reprezentacja, 3'-TATT-5' czyli 5'-TTAT-3' będzie oznaczać  $\neg a$ ). Z 256 dostępnych różnych kombinacji czwórek nukleotydów, 16 należy do grupy komplementarnych z samym sobą, zatem pozostaje do dyspozycji 240 różnych łańcuchów.

Ze względu na wykorzystywany w systemie enzym restrykcyjny *BbvI*, musimy uwzględnić dodatkowe ograniczenia. Jego sekwencja rozpoznawalna wygląda następująco:



Ryc. 113: fragment rozpoznawalny przez enzym BbvI

Schemat kodowania klauzuli faktu wygląda zaś następująco:



Ryc. 114: schemat kodowania dowolnej molekuly faktu

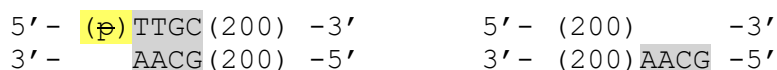
Nukleotydy znajdujące się pomiędzy właściwą sekwencją rozpoznawalną przez BbvI a czwórką identyfikującą literał stanowią niekompletny łańcuch rozpoznawalny przez ten enzym. Zatem aby nie dopełnić go „przypadkiem” i nie spowodować rozcięcia molekuly w niepożądanym miejscu, musimy odrzucić użycie nukleotydu G jako pierwszego elementu wspomnianej czwórki – inaczej uzyskalibyśmy 3'-CGACG-5', która jest już sekwencją rozpoznawaną przez wspomniany enzym (tu byłaby w nieodpowiednim miejscu). Z tej samej przyczyny, musimy odrzucić C jako czwarty element, ponieważ wtedy oznaczałoby to pojawienie się G jako pierwszego elementu w łańcuchu komplementarnym. Oznacza to, że mamy  $3 * 4 * 4 * 3 = 144$  unikalnych łańcuchów. Wśród nich znajduje się 12, które należą do grupy komplementarnych z samym sobą, co po odjęciu daje 132 możliwe czwórki, co po podzieleniu przez 2 (aby kodować pojedyncze zmienne zdaniowe i ich zaprzeczenia) daje 66 różnych możliwych do zakodowania zmiennych.

Tabela zawierająca spis wszystkich unikalnych czwórek możliwych do uzyskania, z zaznaczeniem które z nich mogą być wykorzystane do reprezentacji zmiennych atomowych omawianego systemu wnioskującego, znajduje się na końcu pracy w dodatku A.

### 5. 1. 2. Liczba zapytań możliwych do zakodowania

Z analizy przedstawionej w poprzednim punkcie wynika, że do dyspozycji mamy 16 różnych unikalnych czwórek komplementarnych z samym sobą, które możemy wykorzystać do zakodowania 16-tu różnych zapytań w obrębie tej samej próbki (tak jak zostało to przedstawione w rozdziale 4, podrozdziale 4.2.5 – pojedyncze zapytanie może dotyczyć dowolnego wyrażenia z uwzględnieniem koniunkcji i alternatywy, przedstawionego w dysjunkcyjnej postaci normalnej).

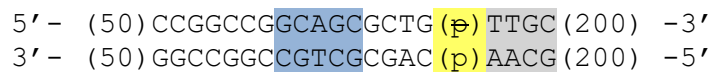
W przypadku gdybyśmy potrzebowali więcej zapytań, możemy również „zapożyczyć” unikalną czwórkę z zestawu przeznaczonego do reprezentacji literałów. Oczywiście w tym przypadku dana czwórka nie może zostać wykorzystana do reprezentacji zdania atomowego i zapytania jednocześnie. W przeciwieństwie do czwórek komplementarnych z samym sobą, wykorzystanie takiej sekwencji będzie wymagało utworzenia dwóch różnych molekuł reprezentujących jedno zapytanie, ponieważ w tym przypadku czwórka komplementarna będzie inna od czwórki bazowej. Utwórzmy przykładowe zapytanie *p?* przy użyciu sekwencji 3'-AACG-5' (oraz siłą rzeczy komplementarnego łańcucha 3'-CGTT-5') jako czwórki reprezentującej zapytanie.



Ryc. 115: schemat molekuł reprezentujących zapytanie p?



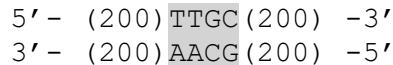
Dlaczego dodanie komplementarnej molekuly uzupełniającej (z prawej strony) jest w tym przypadku niezbędne, najłatwiej zobaczyć na przykładzie reakcji  $p$  oraz przedstawionego powyżej zapytania  $p?$ . Reakcja będzie wyglądać następująco:



Ryc. 116a: ligacja molekuly reprezentującej  $p$  i molekuly zapytania



Ryc. 116b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



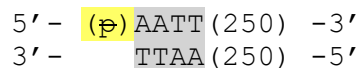
Ryc. 116c: ligacja powstałej molekuly i czwórki uzupełniającej zapytanie

Jak widać, poza wprowadzeniem dodatkowej molekuly uzupełniającej zapytanie, reakcja nie będzie niczym różnić się od przeprowadzonej dla standardowego zapytania, zawierającego unikalną czwórkę komplementarną ze sobą. W powyższym przykładzie, pojawienie się molekuly niezawierającej „lepkich końców” i o unikalnej długości 404 nukleotydów oznacza odpowiedź twierdzącą.

Tabela zawierająca spis wszystkich unikalnych czwórek możliwych do uzyskania, z zaznaczeniem które z nich mogą być wykorzystane do reprezentacji zmiennych atomowych, a zatem również zapytań zgodnie z przedstawionym powyżej schematem, znajduje się na końcu pracy w dodatku A. Ponadto znajduje się tam tabela zawierająca spis unikalnych czwórek symetrycznych, wykorzystywanych wyłącznie do reprezentacji zapytań.

### 5. 1. 3. Zapytania o dwie wartości zdania atomowego

Innym problemem jest sytuacja, w której chcielibyśmy zapytać jednocześnie o dane zdanie atomowe lub o jego negację, np. gdy chcielibyśmy się dowiedzieć czy z danych aksjomatów początkowych wynika  $p$  czy  $\neg p$  (gdyby pojawiły się oba to oczywiście zadziałałby mechanizm wykrycia sprzeczności, patrz: podrozdział 4.3). Niestety jeżeli umieścilibyśmy w jednej próbówce takie przeciwne sobie zapytania, zanim nastąpiłyby wszystkie możliwe reakcje wnioskowania, nastąpiłaby również ligacja komplementarnych ze sobą cząsteczek zapytań, co całkowicie zablokowałoby uzyskanie odpowiedzi. Weźmy przykładowe, poprawnie skonstruowane zapytania  $p?$  oraz  $\neg p?$  (w tym przypadku aksjomaty wejściowe nie mają żadnego znaczenia):

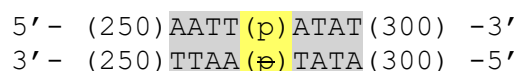


Ryc. 117a: schemat molekuly reprezentującej zapytanie  $p?$



Ryc. 117b: schemat molekuly reprezentującej zapytanie  $\neg p?$

Następnie przyjrzyjmy się ich reakcji:



Ryc. 118: ligacja cząsteczek reprezentujących zapytania  $p?$  oraz  $\neg p?$

Uzyskaliśmy cząsteczkę bez „lepkich końców” i o stałej długości 562 nukleotydów, która nie stanowi odpowiedzi na żadne zadane pytanie (zgodnie z przyjętą tutaj konwencją, cząsteczka o długości 504 nukleotydów stanowiłaby odpowiedź twierdzącą dla  $p?$ , natomiast molekula o długości 604 nukleotydów odpowiedź twierdzącą dla  $\neg p?$ ), ale blokuje ona możliwość dalszej reakcji zapytań i pojawienia się właściwych odpowiedzi.

Możemy uzyskać odpowiedź na dwa pytania o różne wartości tego samego zdania atomowego poprzez zastosowanie dwóch probówek, zawierających dokładnie te same aksjomaty wejściowe i różniących się jedynie zapytaniami – tylko w jednej z nich możemy uzyskać odpowiedź twierdzącą, ale może się zdarzyć też że nie uzyskamy jej w żadnej z probówek (gdy aksjomaty wejściowe nie są wystarczające do jej wyprowadzenia). W pojedynczej probówce możemy umieścić wiele różnych (niekolidujących ze sobą jak powyższe) zapytań, a w drugiej probówce umieścić wszystkie ich negacje.

Istnieją też przypadki większego lub bardziej złożonego zbioru zapytań, w których potrzebujemy jeszcze większej liczby probówek. Np. zapytania uwzględniające koniunkcję  $(p \wedge q)?$  oraz  $(\neg p \wedge q)?$  również musimy rozdzielić ze względu na komplementarność sekwencji reprezentujących pierwszy element. Jeżeli dodamy do zbioru zapytań trzecie postaci  $(p \wedge \neg q)?$  to również musimy je oddzielić ze względu na komplementarność sekwencji reprezentujących drugi element, które pojawiają się w formie „lepkiego końca” po poprawnej reakcji z molekułą reprezentującą fakt  $p$  (jeżeli znajdzie się ona w probówce). Zatem w takiej sytuacji potrzebujemy dokładnie trzech probówek.

Zastosowanie większej liczby probówek i rozdzielenie mogących kolidować ze sobą zapytań nie wpływa na autonomiczność omawianego systemu. Wszystkie reakcje odbywają się równocześnie i bez dodatkowej ingerencji laboranta.

#### **5. 1. 4. Zapytania o klauzule złożone**

Kolejnym szczególnym przypadkiem, spowodowanym ograniczeniami technicznymi możliwości DNA i przede wszystkim enzymów restrykcyjnych, jest zapytanie o alternatywę, czyli o klauzule wieloelementowe. W podrozdziale 4.2.5 dotyczącym zapytań, przedstawiony został sposób kodowania zapytania o alternatywę, jednak dotyczy on wyłącznie wyrażenia, w którym co najmniej jeden z jej składników jest reprezentowany w probówce przez klauzulę faktu, która w przeciwieństwie do pozostałych klauzul zawiera niezbędną sekwencję rozpoznawalną przez enzym restrykcyjny. Weźmy przykładowe zapytanie  $(p \vee q)?$  oraz dwuelementową klauzulę  $(p \vee q)$ , które teoretycznie powinny zwrócić odpowiedź twierdzącą. Niestety w reprezentacji molekularnej żadna z używanych molekuł nie zawiera sekwencji rozpoznawalnej przez używany enzym restrykcyjny, więc reakcja między nimi w podstawowej wersji omawianego systemu nie wykona się.

Rozwiązaniem takiej sytuacji jest użycie koncepcji dowodu nie wprost, gdzie zapytania kodujemy w formie negacji literalów (zawierających sekwencje rozpoznawalne przez enzym restrykcyjny), szczegółowo omówionej w podrozdziale 5.3.1.

### 5. 1. 5. Wyprowadzenie faktu z alternatywy

Istnieją aksjomaty, na podstawie których wyłącznie ze zbioru alternatyw powinna zostać wyprowadzona informacja o prawdziwości pojedynczego literału np. z  $(p \vee q)$  oraz  $(\neg p \vee q)$  możemy uzyskać (wykorzystując reprezentację implikacji za pomocą alternatywy i negacji oraz regułę przechodniości implikacji) wniosek częściowy postaci  $(q \vee q)$  a następnie z niego wyprowadzić  $q$ . Niestety tak powstała molekula również nie zawiera sekwencji rozpoznawalnej przez enzym restrykcyjny, ponieważ fizycznie nie jest molekułą reprezentującą klauzulę faktu, a zatem nie będzie ona mogła brać udziału w dalszych reakcjach na tych samych prawach co gdyby  $q$  zostało wyprowadzone na podstawie molekuł faktu  $p$  lub  $\neg p$  oraz nie zwróci poprawnej odpowiedzi w przypadku zapytania  $q$ ?

Rozwiązaniem dla takiej sytuacji jest ponownie użycie koncepcji dowodu nie wprost, gdzie zapytania kodujemy w formie negacji literałów (zawierających sekwencje rozpoznawalne przez enzym restrykcyjny), szczegółowo omówionej w podrozdziale 5.3.1.

## 5. 2. Dodatkowe zastosowania

### 5. 2. 1. Funkcje logiczne w postaci kanonicznej

Funkcją logiczną (lub inaczej funkcją boolowską) nazywamy dowolne odwzorowanie dowolnej wielkości zbioru zmiennych przyjmujących wartości 1 lub 0 (*prawdy* lub *falszu*) w zbiór wartości również boolowskich 1 lub 0. Precyzyjnie, funkcją logiczną nazywamy dowolne odwzorowanie:

$$f: \{0, 1\}^n \rightarrow \{0, 1\}$$

dla pewnego  $n \geq 1$ . Przykładami takich funkcji są oczywiście funkcje zadane zdaniami złożonymi. Wtedy  $n$  jest równe liczbie zdań atomowych występujących w tym zdaniu. Na przykład zdanie złożone  $((p \wedge q \wedge r) \vee \neg s)$  reprezentuje funkcję logiczną 4 zmiennych, która np. dla wartości  $w(p) = 1$ ,  $w(q) = 1$ ,  $w(r) = 0$ ,  $w(s) = 0$  przyjmuje wartość 1.

Podstawowym twierdzeniem dotyczącym funkcji logicznych jest to, że zachodzi fakt odwrotny – dla każdej funkcji logicznej istnieje zdanie złożone, które generuje tę funkcję. Istotnie, jeśli  $f: \{0, 1\}^n \rightarrow \{0, 1\}$  jest dowolną funkcją logiczną, to szukane zdanie złożone  $A$  jest alternatywą  $A = B_1 \vee \dots \vee B_m$ , gdzie każde  $B_i$  jest koniunkcją, która ma postać: dla dowolnego układu wartości zmiennych  $w(p_1) = a_1, \dots, w(p_n) = a_n$ , gdzie  $a_i = 0$  lub  $a_i = 1$ , dla których  $f(a_1, \dots, a_n) = 1$ ,  $B_i = q_1 \wedge \dots \wedge q_n$ , gdzie  $q_j = p_j$  gdy  $a_j = 1$ ;  $q_j = \neg p_j$  gdy  $a_j = 0$ , np. dla funkcji logicznej trzech zmiennych  $f$  zadanej tabelką:

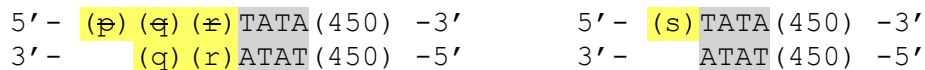
$p_1$ :	0	0	0	0	1	1	1	1
$p_2$ :	0	0	1	1	0	0	1	1
$p_3$ :	0	1	0	1	0	1	0	1
$f$ :	1	0	0	1	0	1	0	0

Tab. 1: wartościowania przykładowej funkcji  $f$

otrzymamy zdanie logiczne:

$$(\neg p_1 \wedge \neg p_2 \wedge \neg p_3) \vee (\neg p_1 \wedge p_2 \wedge p_3) \vee (p_1 \wedge \neg p_2 \wedge p_3).$$

Zauważmy, że w myśl tej konstrukcji otrzymane zdania są w dysjunkcyjnej postaci normalnej. Jeśli funkcja  $f$  reprezentuje zdanie złożone  $A = B_1 \vee \dots \vee B_m$ , gdzie każda klauzula  $B_i$  jest koniunkcją zmiennych lub ich negacji, to w naszej implementacji będzie ono reprezentowane przez  $m$  molekuł odpowiadających  $B_1, \dots, B_m$ . Molekuła pojedynczego  $B_i$  zawiera czwórki nukleotydów komplementarnych do reprezentujących wszystkie zmienne wchodzące w skład  $B_i$ . Oznacza to, że jeśli w  $B_i$  występuje zmienna  $p$ , to w molekułe umieszczamy czwórkę reprezentującą  $\neg p$ . Wszystkie czwórki wszystkich zmiennych występujących w  $B_i$  zapisujemy obok siebie (w dowolnej kolejności ze względu na przemienność koniunkcji; w taki sposób, że wyłącznie pierwsza czwórka stanowi „leпки koniec” molekuły), następnie za nimi znajduje się dowolna czwórka nukleotydów z puli symetrycznych czwórek (tzn. komplementarnych z samym sobą) oraz dowolnego łańcucha nukleotydów o ustalonej konkretnej długości. Reprezentacja molekularna danej funkcji logicznej jest identyczna jak reprezentacja pojedynczego zapytania (w tym będącego zdaniem złożonym, patrz: podrozdział 4.2.5). Na przykład funkcja logiczna reprezentowana zdaniem  $(p \wedge q \wedge r) \vee \neg s$ , będzie składała się z dwóch molekuł, przedstawiających odpowiednio  $(p \wedge q \wedge r)$  oraz  $\neg s$ . Wyglądają one następująco;



Ryc. 119: molekuły wspólnie reprezentujące funkcję logiczną  $((p \wedge q \wedge r) \vee \neg s)$

Aby obliczyć wartości funkcji logicznej  $A$ , wartości argumentów kodujemy analogicznie jak molekuły klauzul faktów. Każda z nich składa się z następujących części: 50 dowolnych nukleotydów, stałej sekwencji 3'-GGCCGGC-5', sekwencji 3'-CGTCG-5' rozpoznawalnej przez enzym restrykcyjny *BbvI* (zaznaczonej kolorem niebieskim), stałej sekwencji 3'-CGAC-5' oraz „lepkiego końca” czterech nukleotydów, który identyfikuje zmienną zdaniową i jej wartość. Przyjmijmy, że na potrzeby kodowania molekularnego, argumenty funkcji o wartości 1 odpowiadają pojedynczym zdaniom atomowym a argumenty o wartości 0 ich negacjąm.

Przeanalizujmy wartość funkcji dla argumentów:  $w(p) = 1$ ,  $w(q) = 1$ ,  $w(r) = 0$ ,  $w(s) = 0$ :



Ryc. 120a: molekuła reprezentująca argument  $p$



Ryc. 120b: molekuła reprezentująca argument  $q$

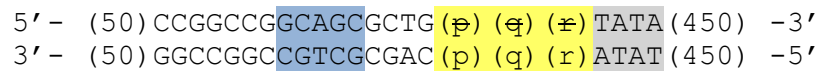


Ryc. 120c: molekuła reprezentująca argument  $\neg r$

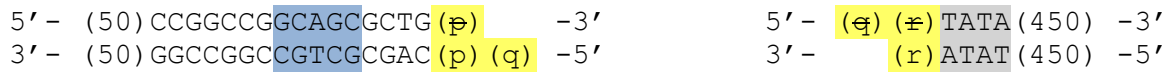


Ryc. 120d: molekuła reprezentująca argument  $\neg s$

Reakcja będzie wyglądać następująco:



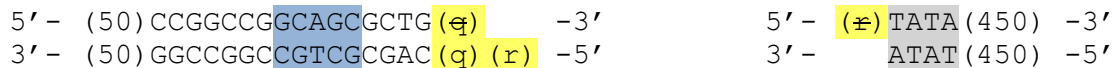
*Ryc. 121a: ligacja molekuly p oraz molekuly koniunkcji ( $p \wedge q \wedge r$ )*



*Ryc. 121b: rozcięcie za pomocą enzymu restrykcyjnego BbvI*

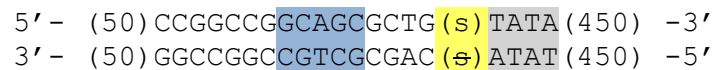


*Ryc. 121c: ligacja molekuly q oraz powstałej reszty molekuly koniunkcji ( $p \wedge q \wedge r$ )*



*Ryc. 121d: rozcięcie za pomocą enzymu restrykcyjnego BbvI*

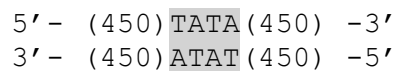
Na tym etapie następuje zatrzymanie reakcji ponieważ nie istnieje molekula komplementarna do uzyskanego „lepkiego końca” – zmienna atomowa  $r$  posiada wartość fałszu. W tym samym czasie reakcja z drugą molekułą iloczynu składowego wygląda następująco:



*Ryc. 122a: ligacja molekuly p oraz molekuly koniunkcji ( $p \wedge q \wedge r$ )*



*Ryc. 122b: rozcięcie za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 122c: ligacja powstałych molekul ze sobą*

Tak powstała cząsteczka wyjściowa, o konkretnej długości (tutaj 904 nukleotydów) oznacza odpowiedź twierdzącą – wartość danej funkcji dla podanych wartości argumentów wynosi 1.

Przeanalizujemy jeszcze wykorzystanie tego modelu dla dwuargumentowej funkcji XOR czyli alternatywy wykluczającej w postaci  $(a \wedge \neg b) \vee (\neg a \wedge b)$  (podobny przykład został omówiony w podrozdziale 3.1.2). Niezbędne molekuly wyglądają następująco:



*Ryc. 123: molekuly reprezentujące funkcję logiczną XOR czyli  $(a \wedge \neg b) \vee (\neg a \wedge b)$*

Przeanalizujemy wartość funkcji dla argumentów  $w(a) = 0$  i  $w(b) = 1$ :



*Ryc. 124a: molekula reprezentująca argument  $\neg a$*

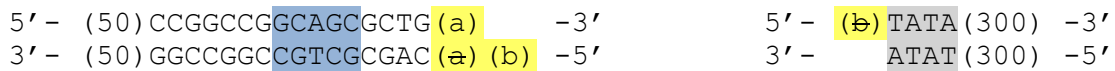


*Ryc. 124b: molekula reprezentująca argument  $b$*

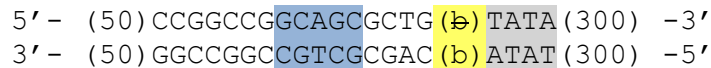
Reakcja będzie wyglądać następująco:



*Ryc. 125a: ligacja molekuly  $\neg a$  oraz molekuly koniunkcji  $(\neg a \wedge b)$*



*Ryc. 125b: rozcięcie za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 125c: ligacja molekuly b oraz powstałej reszty molekuly koniunkcji ( $\neg a \wedge b$ )*



*Ryc. 125d: ligacja molekuly b oraz powstałej reszty molekuly koniunkcji ( $\neg a \wedge b$ )*



*Ryc. 125e: ligacja powstałych molekul ze sobą*

Tak powstała cząsteczka wyjściowa, o konkretnej długości (tutaj 604 nukleotydów) oznacza odpowiedź twierdzącą – wartość funkcji XOR dla podanych argumentów wynosi 1.

Przeanalizujmy wartość funkcji dla argumentów:  $w(a) = 0$  i  $w(b) = 0$ :

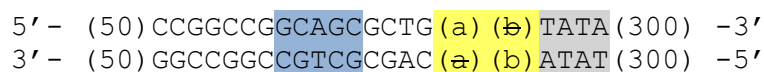


*Ryc. 126a: molekula reprezentująca argument  $\neg a$*

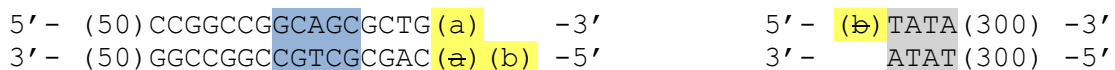


*Ryc. 126b: molekula reprezentująca argument  $\neg b$*

Reakcja będzie wyglądać następująco:



*Ryc. 127a: ligacja molekuly  $\neg a$  oraz molekuly koniunkcji ( $\neg a \wedge b$ )*



*Ryc. 127b: rozcięcie za pomocą enzymu restrykcyjnego BbvI*

W tym momencie następuje zatrzymanie reakcji, ponieważ nie istnieją cząsteczki argumentów komplementarne do żadnego spośród iloczynów składowych. Niepojawienie się w próbówce cząsteczki o konkretnej długości (tutaj 604 nukleotydów) oznacza odpowiedź przeczącą – wartość funkcji XOR dla podanych argumentów wynosi 0.

Przedstawiona propozycja otrzymywania wartości funkcji logicznej dla podanych argumentów za pomocą rozszerzonego systemu wnioskującego posiada funkcjonalność bardzo zbliżoną do prezentowanej w [47] (*patrz*: podrozdział 3.1.2). Umożliwia przetwarzanie wszystkich koniunkcji składowych w sposób równoległy i zależnie od wprowadzonych wartości argumentów każdy z nich odpowiednio prowadzi do powstania molekuly odpowiedzi twierdzącej lub nie. Odpowiedź twierdząca powstała w wyniku reakcji wyłącznie jednego z przetwarzanych iloczynów oznacza spełnienie danej funkcji dla podanych argumentów.

Główną zaletą w stosunku do koncepcji Unolda i Trocia [47] jest możliwość jednoczesnego przetwarzania wielu funkcji logicznych, z których każda wykorzystuje pewien podzbiór wspólnego zbioru argumentów o konkretnych wartościach. Samo przetwarzanie pojedynczej

funkcji logicznej wymaga mniej reakcji, ponieważ w przeciwieństwie do [47] molekuly reprezentujące przeciwne wartości tego samego zdania atomowego nie reagują ze sobą.

Przedstawione zastosowanie (sprawdzanie wartości funkcji logicznych) jest jednym z wielu możliwości wykorzystania omawianego rozszerzonego systemu wnioskującego, co również odróżnia go od rozwiązań dedykowanych wyłącznie do pojedynczego typu problemu.

### 5. 2. 2. Częściowe rozwiązanie SAT problemu

Problem SAT (z ang. *boolean satisfiability problem*), nazywany też problemem spełnialności, to pytanie czy dla danego zdania złożonego istnieje takie wartościowanie zmiennych (przypisanie im wartości *prawdy* albo *falszu* oznaczanych jako 1 lub 0) aby całe zdanie było prawdziwe. Mówimy, że zdanie nie jest spełnialne jeżeli dla każdego możliwego przypisania wartości zmiennych całość wyrażenia posiada wartość 0. Przykładowa formuła spełnialna to  $(a \wedge (\neg a \vee \neg b))$  (dla  $w(a) = 1$  oraz  $w(b) = 0$ ) a przykładowa niespełnialna (niezależnie od wartościowania) to  $(a \wedge (\neg a \vee \neg b) \wedge b)$ . Poprzez SAT oznaczamy również język wszystkich spełnialnych formuł rachunku zdań a przez UNSAT jego dopełnienie w zbiorze wszystkich zdań czyli język wszystkich niespełnialnych formuł rachunku zdań.

Problemem *k*-SAT nazywamy problem spełnialności zdań w koniunkcyjnej postaci normalnej (*patrz*: podrozdział 1.4), w których klauzule mają nie więcej niż *k* literałów. 1-SAT oraz 2-SAT mają rozwiązania w deterministycznym czasie wielomianowym, podczas gdy 3-SAT i kolejne wymagają bardziej złożonych obliczeń (gdyż dotąd nieznane są tego typu rozwiązania w deterministycznym czasie wielomianowym).

Pierwsza koncepcja rozwiązania problemu SAT za pomocą molekuł DNA, inspirowana wcześniejszą pracą Leonarda Adlemana [1] została przedstawiona przez Richarda Liptona w roku 1995 [16]. Zbiór wszystkich możliwych wartościowań zmiennych w danym zdaniu złożonym został przedstawiony w postaci grafu, w którym krawędzie reprezentują możliwe przypisania wartości poszczególnym zmiennym zdaniowym, a każdy wierzchołek rozdziela te przypisania. Każda droga w takim grafie oznacza pojedyncze przypisanie wartości zmiennych, co pozwala w dalszej reakcji określić spełnialność. Wszystkie elementy takiego grafu zostały zakodowane za pomocą odpowiednich molekuł jednoniciowych (analogicznie jak w pracy Adlemana) i umieszczone w probówce, gdzie w równoległy sposób mają miejsce wszystkie możliwe operacje łączenia się tych molekuł. Koncepcja opiera się na omówionym wcześniej modelu filtrowania (*patrz*: podrozdział 2.3).

W późniejszym czasie pojawiło się więcej różnych koncepcji rozwiązania problemu SAT za pomocą DNA. Większość z nich również opierała się na propozycji równoległego wygenerowania wszystkich możliwych przypisań wartości zmiennych zdaniowych, między innymi w [18]. Alternatywne podejście do tego zagadnienia zostało przedstawione we wspomnianej już pracy Alfonso Rodríguez-Patón, Íñaki Sainz de Murieta oraz Petra Sosika z roku 2011 [30] (*patrz*: podrozdział 3.4). Opiera się ono w rzeczywistości na poszukiwaniu zdań niespełnialnych, czyli należących do języka UNSAT, poprzez upraszczanie danego zdania złożonego z wykorzystaniem praw klasycznego rachunku zdań (analogicznie jak w systemie wnioskowania). Jeżeli ostatecznie zdanie powstałe podczas reakcji nie jest spełnialne, to wejściowe zdanie złożone jest również niespełnialne. W przeciwnym wypadku system nie

jest w stanie określić czy formuła jest spełnialna i należy wykonać obliczenia za pomocą alternatywnego rozwiązania czy klasycznego algorytmu. Odwołując się do powyższej koncepcji, istnieje możliwość użycia rozszerzonego systemu wnioskującego przedstawionego w tej rozprawie również do poszukiwania formuł niespełnialnych. Taka koncepcja została już zaprezentowana w pracy [33].

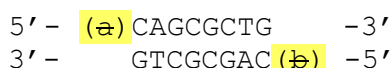
Aby zastosować omówiony wcześniej system wnioskujący, zdanie musi być przedstawione w koniunkcyjnej postaci normalnej (*patrz:* podrozdział 1.4) a następnie zakodowane za pomocą molekuł w analogiczny sposób jak dla wnioskowania z rozdzieleniem na klauzule faktu i pozostałe klauzule. Niezbędna jest również molekula terminalna wnioskowania (*patrz:* podrozdział 4.2). Jeżeli w wyniku reakcji wnioskowania na podstawie aksjomatów wejściowych, otrzymamy informację o wystąpieniu sprzeczności (*patrz:* podrozdział 4.3) to oznacza to, że zdanie bazowe jest niespełnialne. Przyjrzyjmy się reakcji dla wspomnianej wcześniej niespełnialnej formuły  $(a \wedge (\neg a \vee \neg b) \wedge b)$ . Molekuły będą wyglądać następująco:



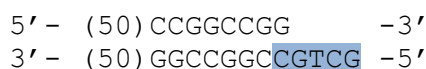
*Ryc. 128a: molekula reprezentująca klauzulę faktu a*



*Ryc. 128b: molekula reprezentująca klauzulę faktu b*



*Ryc. 128c: molekula reprezentująca klauzulę  $(\neg a \vee \neg b)$*

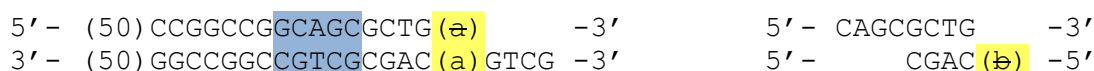


*Ryc. 128d: molekula terminalna wnioskowania*

Reakcja będzie wyglądać następująco:



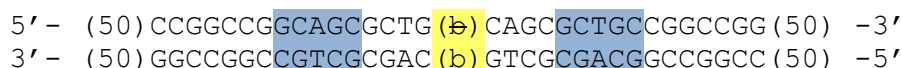
*Ryc. 129a: ligacja molekuł reprezentujących klauzule a oraz  $(\neg a \vee \neg b)$*



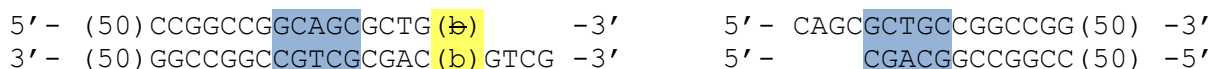
*Ryc. 129b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



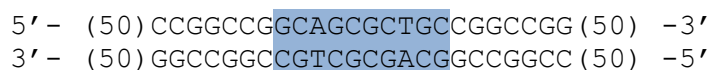
*Ryc. 129c: ligacja molekuly terminalnej i powstałej cząsteczki*



*Ryc. 129d: ligacja molekuł reprezentujących klauzule b oraz  $\neg b$  (powstałej w poprzednim kroku)*

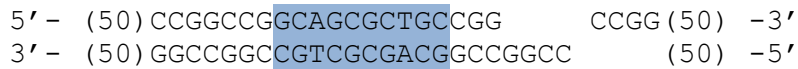


*Ryc. 129e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*

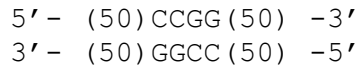


*Ryc. 129f: ligacja molekuly terminalnej i powstałej cząsteczki*





Ryc. 129g: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI

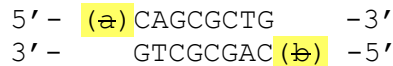


Ryc. 129h: ligacja powstałych molekul ze sobą

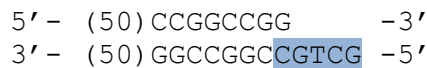
Powstała cząsteczka niezawierająca „lepkich końców” oraz o unikalnej długości 104 nukleotydów oznacza wystąpienie sprzeczności w systemie, co oznacza że analizowana formuła nie jest spełnialna dla żadnych wartości literalów. Przyjrzyjmy się również reakcji dla wspomnianej wcześniej spełnialnej formuły ( $a \wedge (\neg a \vee \neg b)$ ). Molekuły będą wyglądać następująco:



Ryc. 130a: molekula reprezentująca klauzulę faktu  $a$



Ryc. 130b: molekula reprezentująca klauzulę  $(\neg a \vee \neg b)$

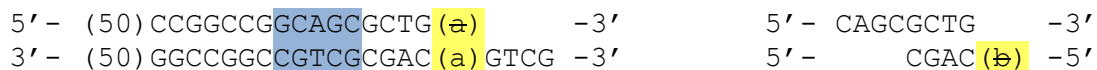


Ryc. 130c: molekula terminalna wnioskowania

Reakcja będzie wyglądać następująco:



Ryc. 131a: ligacja molekul reprezentujących klauzule  $a$  oraz  $(\neg a \vee \neg b)$



Ryc. 131b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 131c: ligacja molekuly terminalnej i powstałej cząsteczki

Reakcja zostanie zatrzymana na tym etapie ponieważ nie istnieje żadna molekula o „lepkim końcu” komplementarnym do powstałej cząsteczki. Oznacza to, że nie powstanie unikalna molekula o długości 104 nukleotydów, która informowałaby o sprzeczności w systemie, a w takim razie system nie jest w stanie określić czy dana formuła jest spełnialna. Należy zatem wykonać obliczenia za pomocą alternatywnego rozwiązania.

Warto zwrócić uwagę, że tę samą koncepcję można wykorzystać do sprawdzania czy dana formuła jest tautologią (czyli wyrażeniem zawsze prawdziwym, niezależnie od przyjętych wartościowań zmiennych zdaniowych) korzystając z prawa, mówiącego że formuła  $A$  jest tautologią wtedy i tylko wtedy gdy formuła  $\neg A$  jest niespełnialna (UNSAT).

Korzystając z omówionej koncepcji możemy sprawdzać formuły pod kątem niespełnialności dla dowolnej liczby  $k$  literalów (odpowiednik problemu  $k$ -SAT). Niedogodnością jest wyłącznie konieczność wystąpienia co najmniej jednej klauzuli faktu, ponieważ tylko te w swojej reprezentacji molekularnej zawierają sekwencję rozpoznawalną przez enzym restrykcyjny BbvI, który jest niezbędny dla rozpoczęcia właściwej reakcji.

## 5. 3. Modyfikacje systemu

### 5. 3. 1. Zastosowanie metody dowodu nie-wprost

Dowód nie-wprost (in. dowód apagogiczny lub sokratejski) to forma dowodu logicznego polegająca na sprowadzeniu do sprzeczności, czyli wyprowadzenia z zaprzeczenia dowodzonej tezy sprzeczności z założeniem lub przyjętymi aksjomatami początkowymi.

Ponieważ w omawianym systemie wnioskowania dysponujemy metodą wykrywania sprzeczności, więc możemy wykorzystać ją również do wnioskowań metodą dowodu nie-wprost. Mianowicie, jeśli chcemy rozstrzygnąć czy z niesprzecznego zbioru aksjomatów  $A$  można wywnioskować zdanie  $a$ , to po dodaniu do  $A$  zaprzeczenia  $\neg a$  powinniśmy:

1. Otrzymać sprzeczność w systemie w przypadku gdy  $a$  jest konsekwencją  $A$ .
2. Nie otrzymać sprzeczności w przypadku przeciwnym.

Daje nam to łatwy do wykorzystania, wręcz uproszczony (po zrezygnowaniu z dodatkowych molekuł reprezentujących zapytania) system dowodzenia, wciąż uwzględniający odpowiednie prawa klasycznego rachunku zdań. Zapytania mogą być zapisane w dysjunkcyjnej postaci normalnej (jak dla standardowej wersji systemu), dzięki czemu ich zaprzeczenie będzie zapisane w koniunkcyjnej postaci normalnej, tak jak w przypadku aksjomatów początkowych. Można też bezpośrednio sprowadzić zaprzeczenie zapytań do koniunkcyjnej postaci normalnej.

Metoda ta ma jednak jedną wadę. Mianowicie, nie możemy tutaj zadać jednocześnie kilku pytań w obrębie jednej próbki, tak jak miało to miejsce w podstawowej wersji systemu, gdyż w takim przypadku otrzymanie sprzeczności nie rozstrzygnęłoby, które ze zdań jest prawdziwe. Można rozwiązać ten problem poprzez sklonowanie zawartości próbki z aksjomatami początkowymi, a następnie umieszczenie w każdej z nich zaprzeczenia jednego z zapytań, o których możliwość wyprowadzenia pytamy.

W przypadku jeżeli nie posiadamy informacji o poprawności (braku sprzeczności) aksjomatów początkowych, konieczne jest przeprowadzenie reakcji w dwóch krokach:

1. Umieszczenie w próbce wyłącznie molekuł aksjomatów początkowych, a następnie sprawdzenie czy w wyniku autonomicznych reakcji między nimi nie powstanie molekula sygnalizująca wystąpienie sprzeczności.
2. Umieszczenie dodatkowo w próbce molekuł zaprzeczenia zapytania, a następnie sprawdzenie czy w wyniku autonomicznych reakcji między nimi powstanie molekula sygnalizująca wystąpienie sprzeczności. W tym przypadku pojawienie się tej molekuly oznacza odpowiedź twierdzącą na zadane pytanie (na mocy twierdzenia o dowodzeniu nie-wprost).

Przeanalizujemy możliwości omawianej modyfikacji systemu na przykładach.

## Zapytanie o klauzulę faktu:

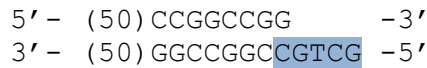
Aksjomaty początkowe:  $p$ . Zapytanie:  $p?$ , zatem zaprzeczenie zapytania:  $\neg p$ .



Ryc. 132a: molekula reprezentująca klauzulę faktu  $p$

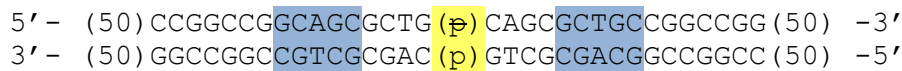


Ryc. 132b: molekula reprezentująca klauzulę faktu  $\neg p$

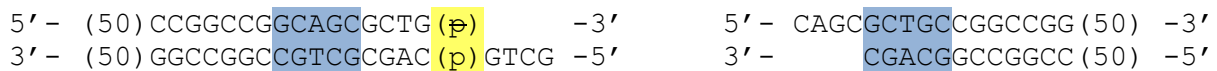


Ryc. 132c: molekula terminalna wnioskowania

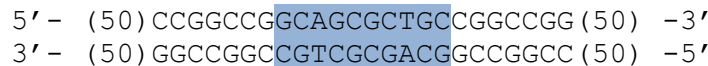
Reakcja będzie wyglądać następująco:



Ryc. 133a: ligacja molekul reprezentujących klauzule  $p$  oraz  $\neg p$



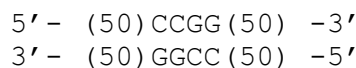
Ryc. 133b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 133c: ligacja molekuly terminalnej i powstałej cząsteczki



Ryc. 133d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 133e: ligacja powstałych molekul ze sobą

Powstała cząsteczka symbolizuje pojawienie się sprzeczności, a zatem w świetle twierdzenia o dowodzeniu nie-wprost oznacza możliwość wyprowadzenia tezy, czyli twierdzącą odpowiedź systemu na zadane pytanie.

## Zapytanie o alternatywę:

Aksjomaty początkowe:  $p$ . Zapytanie:  $(p \vee q)?$ , zatem zaprzeczenie zapytania:  $(\neg p \wedge \neg q)$ .



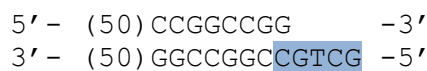
Ryc. 134a: molekula reprezentująca klauzulę faktu  $p$



Ryc. 134b: molekula reprezentująca klauzulę faktu  $\neg p$



Ryc. 134c: molekula reprezentująca klauzulę faktu  $\neg q$



Ryc. 134d: molekula terminalna wnioskowania

Reakcja będzie identyczna jak w przykładzie powyżej, ponieważ cząsteczka reprezentująca fragment zapytania  $\neg q$  nie bierze żadnego udziału w reakcji. Dla zmienionych aksjomatów początkowych, reakcja będzie analogiczna: w przypadku pojawienia się wśród nich  $q$ , udział w reakcji wezmą cząsteczki  $q$  oraz  $\neg q$ , podczas gdy cząsteczka  $\neg p$  nie będzie brała udziału w reakcji. W przypadku pojawienia się obu z nich wśród aksjomatów początkowych, obie reakcje będą przebiegać równoległe (co w tym przypadku spowoduje wyłącznie powstanie większej liczby cząsteczek sygnalizujących sprzeczność, co tak samo oznacza twierdzącą odpowiedź systemu).

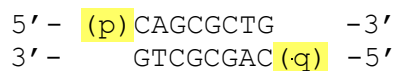
### Wyprowadzenie faktu z alternatywy:

Aksjomaty początkowe:  $(p \vee q)$ ,  $(\neg p \vee q)$ .

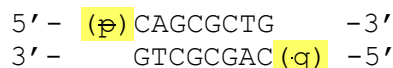
Zapytanie:  $q?$ , zatem zaprzeczenie zapytania:  $\neg q$ .



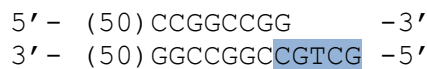
*Ryc. 135a: molekula reprezentująca klauzulę faktu  $\neg q$*



*Ryc. 135b: molekula reprezentująca klauzulę  $(p \vee q)$*

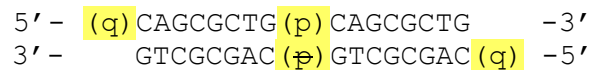


*Ryc. 135c: molekula reprezentująca klauzulę  $(\neg p \vee q)$*

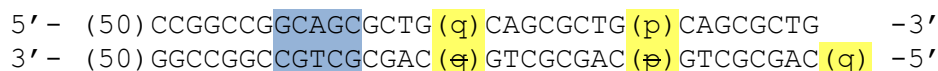


*Ryc. 135d: molekula terminalna wniosowania*

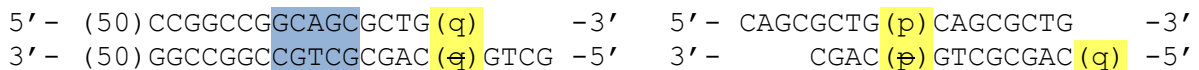
Reakcja będzie wyglądać następująco:



*Ryc. 136a: ligacja molekuł reprezentujących klauzule  $(p \vee q)$  oraz  $(\neg p \vee q)$   
(powstaje molekula odpowiadająca alternatywie  $(q \vee q)$ )*



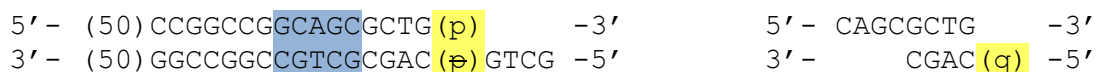
*Ryc. 136b: ligacja molekuł reprezentujących klauzule  $\neg q$  oraz  $(q \vee q)$*



*Ryc. 136c: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



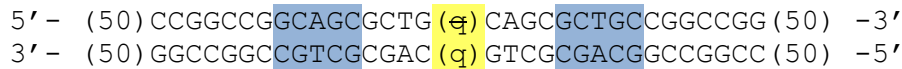
*Ryc. 136d: ligacja molekuly terminalnej i nowopowstalej cząsteczki*



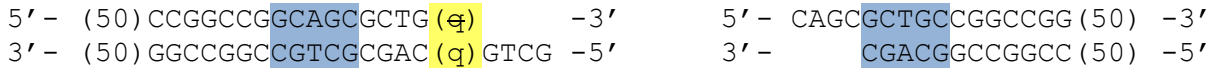
*Ryc. 136e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



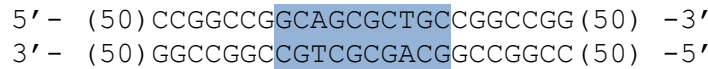
*Ryc. 136f: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $q$ )*



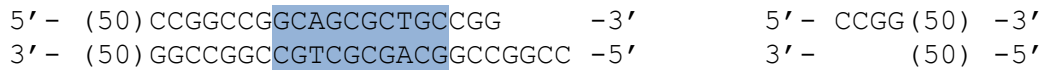
Ryc. 136g: ligacja molekuł reprezentujących klauzule  $q$  oraz  $\neg q$



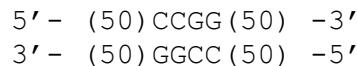
Ryc. 136h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 136i: ligacja molekuly terminalnej i powstałej cząsteczki



Ryc. 136j: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 136k: ligacja powstałych molekuł ze sobą

Powstała cząsteczka symbolizuje pojawienie się sprzeczności, a zatem w świetle twierdzenia o dowodzeniu nie-wprost oznacza możliwość wyprowadzenia tezy, czyli twierdzącą odpowiedź systemu na zadane pytanie.

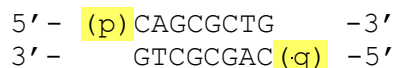
### Wyprowadzenie faktu z alternatywy i dalsze wnioskowanie:

Aksjomaty początkowe:  $(p \vee q)$ ,  $(\neg p \vee q)$ ,  $(\neg q \vee r)$ .

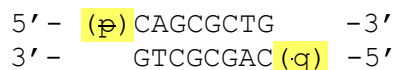
Zapytanie:  $r$ ?, zatem zaprzeczenie zapytania:  $\neg r$ .



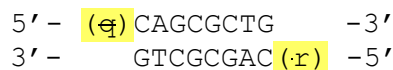
Ryc. 137a: molekula reprezentująca klauzulę faktu  $\neg r$



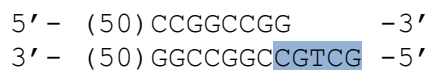
Ryc. 137b: molekula reprezentująca klauzulę  $(p \vee q)$



Ryc. 137c: molekula reprezentująca klauzulę  $(\neg p \vee q)$

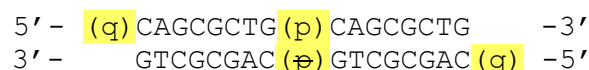


Ryc. 137d: molekula reprezentująca klauzulę  $(\neg q \vee r)$

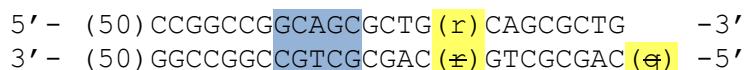


Ryc. 137e: molekula terminalna wnioskowania

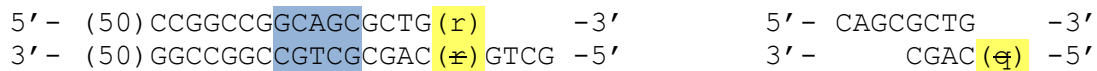
Reakcja będzie wyglądać następująco:



Ryc. 138a: ligacja molekuł reprezentujących klauzule  $(p \vee q)$  oraz  $(\neg p \vee q)$  (powstaje molekula odpowiadająca alternatywie  $(q \vee q)$ )



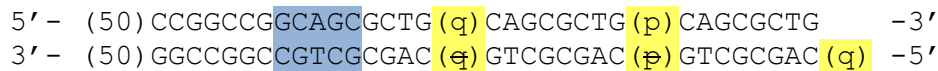
Ryc. 138b: ligacja molekuł reprezentujących klauzule  $\neg r$  oraz  $(\neg q \vee r)$



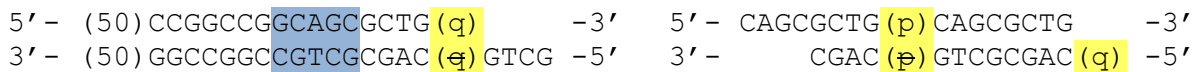
*Ryc. 138c: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



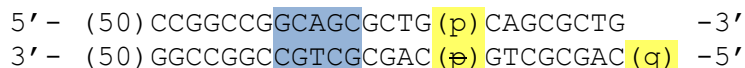
*Ryc. 138d: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg q$ )*



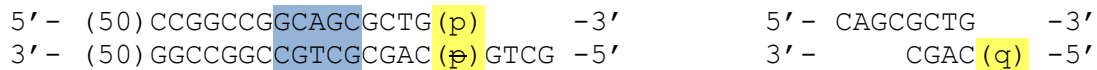
*Ryc. 138e: ligacja molekul reprezentujących klauzule  $\neg q$  oraz  $(q \vee q)$*



*Ryc. 138f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 138g: ligacja molekuly terminalnej i nowopowstalej cząsteczki*

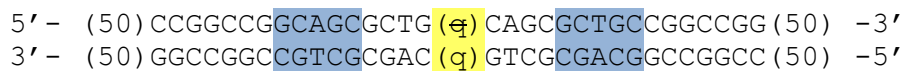


*Ryc. 138h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*

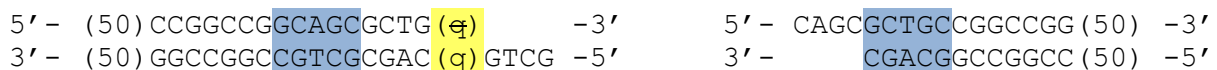


*Ryc. 138i: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $q$ )*

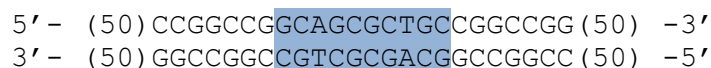
Ponieważ podczas przytoczonej reakcji powstały już cząsteczki reprezentujące zarówno  $q$  jak i  $\neg q$ , to reakcja może się zakończyć w następujący sposób (analogiczny jak we wcześniejszych reakcjach):



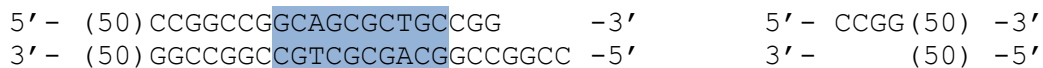
*Ryc. 139a: ligacja molekul reprezentujących klauzule  $q$  oraz  $\neg q$*



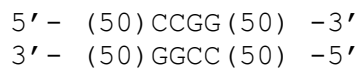
*Ryc. 139b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 139c: ligacja molekuly terminalnej i powstalej cząsteczki*



*Ryc. 139d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



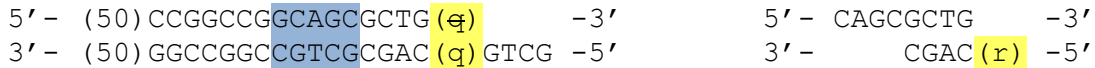
*Ryc. 139e: ligacja powstalych molekul ze sobą*

Powstała cząsteczka symbolizuje pojawienie się sprzeczności, a zatem w świetle twierdzenia o dowodzeniu nie-wprost oznacza możliwość wyprowadzenia tezy, czyli twierdzącą odpowiedź systemu na zadane pytanie.

Możliwe jest też równoległe kontynuowanie reakcji (z ponownym wykorzystaniem molekuly reprezentującej  $(\neg q \vee r)$ ), która wygląda następująco:



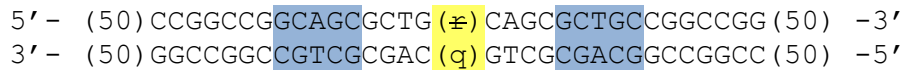
Ryc. 140a: ligacja molekuł reprezentujących klauzule q oraz  $(\neg q \vee r)$



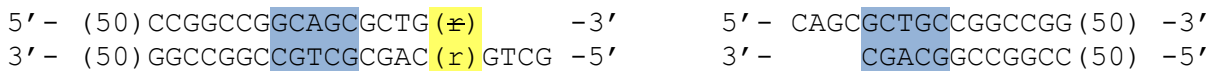
Ryc. 140b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



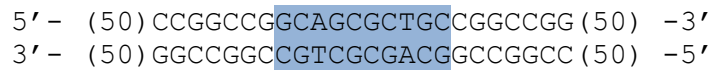
Ryc. 140c: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli r)



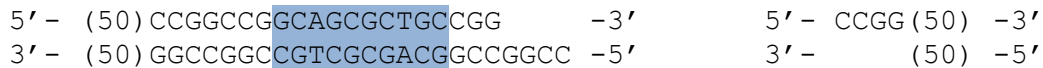
Ryc. 140d: ligacja molekuł reprezentujących klauzule q oraz  $\neg q$



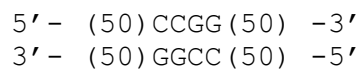
Ryc. 140e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 140f: ligacja molekuly terminalnej i powstałej cząsteczki



Ryc. 140g: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 140h: ligacja powstałych molekuł ze sobą

Powstała, identyczna do wcześniejszej, cząsteczka również symbolizuje pojawienie się sprzeczności, a zatem w świetle twierdzenia o dowodzeniu nie-wprost oznacza możliwość wprowadzenia tezy, czyli twierdzącą odpowiedź na zadane pytanie.

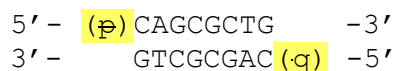
### Wyprowadzenie alternatywy z kilku alternatyw:

Aksjomaty początkowe:  $(\neg p \vee q)$ ,  $(\neg q \vee r)$ ,  $(\neg r \vee p)$ .

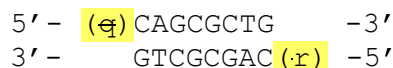
Zapytanie:  $p?$ , zatem zaprzeczenie zapytania:  $\neg p$ .



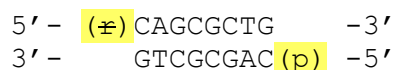
Ryc. 141a: molekula reprezentująca klauzulę faktu  $\neg p$



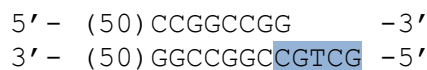
Ryc. 141b: molekula reprezentująca klauzulę  $(\neg p \vee q)$



Ryc. 141c: molekula reprezentująca klauzulę  $(\neg q \vee r)$



Ryc. 141d: molekula reprezentująca klauzulę  $(\neg r \vee p)$

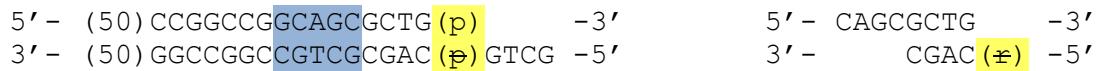


Ryc. 141e: molekula terminalna wniosowania

Reakcja będzie wyglądać następująco:



*Ryc. 142a: ligacja molekuł reprezentujących klauzule  $\neg p$  oraz  $(\neg r \vee p)$*



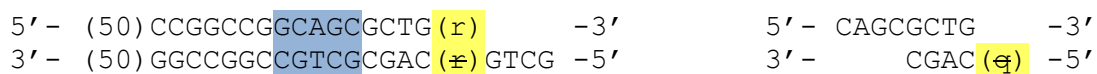
*Ryc. 142b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 142c: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg r$ )*



*Ryc. 142d: ligacja molekuł reprezentujących klauzule  $\neg r$  oraz  $(\neg q \vee r)$*



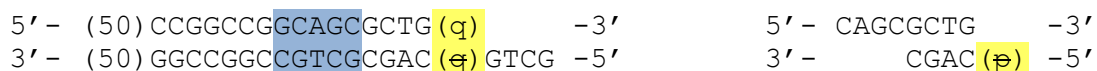
*Ryc. 142e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 142f: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg q$ )*



*Ryc. 142g: ligacja molekuł reprezentujących klauzule  $\neg q$  oraz  $(\neg p \vee q)$*



*Ryc. 142h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI*



*Ryc. 142i: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg p$ )*

Możemy powiedzieć, że powyższa reakcja „zatoczyła koło” i w jej efekcie powstała cząsteczka będąca zaprzeczeniem tezy, czyli  $\neg p$ . Dla tych aksjomatów początkowych (wyłącznie alternatyw:  $(\neg p \vee q)$ ,  $(\neg q \vee r)$ ,  $(\neg r \vee p)$ ) nie jest możliwe wyprowadzenie tylko na ich podstawie żadnego z faktów – reakcja dla zapytań  $q?$  czy  $r?$  wyglądałaby analogicznie. Oznacza to, że na podstawie tych aksjomatów nie jesteśmy w stanie stwierdzić prawdziwości danej tezy, co oczywiście jest poprawną odpowiedzią systemu wnioskującego.

Gdybyśmy zmodyfikowali powyższe alternatywy, usuwając wyłącznie w pierwszej z nich spójnik negacji, to wnioskowanie przebiegałoby inaczej (analogicznie jak we wcześniejszej części dotyczącej pojedynczej alternatywy). Dla zapytania  $p?$  (a zatem jego zaprzeczenia  $\neg p$ ) oraz aksjomatów początkowych:  $(p \vee q)$ ,  $(\neg q \vee r)$ ,  $(\neg r \vee p)$  w pierwszej kolejności uzyskalibyśmy wniosek częściowy  $(p \vee r)$ , następnie  $(p \vee p)$ , które z pomocą zapytania  $p?$  zostałyby sprowadzone do sprzeczności, która oznacza poprawną odpowiedź systemu w jego modyfikacji wykorzystującej zastosowanie metody dowodu nie-wprost.

Jeżeli zaś wrócimy do wybranych wcześniej aksjomatów początkowych (wyłącznie alternatyw:  $(\neg p \vee q)$ ,  $(\neg q \vee r)$ ,  $(\neg r \vee p)$ ), to zapytanie o dowolną, możliwą do wyprowadzenia, inną alternatywę (np.  $(q \vee \neg r)?$ ) powinno zwrócić odpowiedź twierdzącą. Przyjrzyjmy się takiej reakcji dla wspomnianych aksjomatów początkowych:



Aksjomaty początkowe:  $(\neg p \vee q)$ ,  $(\neg q \vee r)$ ,  $(\neg r \vee p)$ .

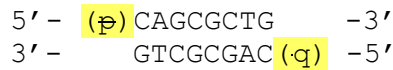
Zapytanie:  $(q \vee \neg r)?$ , zatem zaprzeczenie zapytania:  $(\neg q \wedge r)$ .



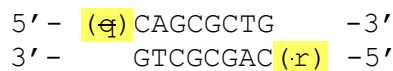
Ryc. 143a: molekula reprezentująca klauzulę faktu  $\neg q$



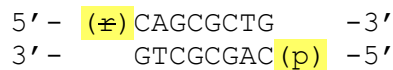
Ryc. 143b: molekula reprezentująca klauzulę faktu  $r$



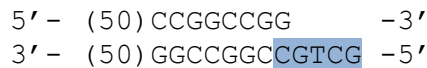
Ryc. 143c: molekula reprezentująca klauzulę  $(\neg p \vee q)$



Ryc. 143d: molekula reprezentująca klauzulę  $(\neg q \vee r)$

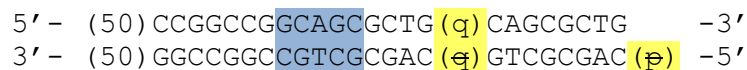


Ryc. 143e: molekula reprezentująca klauzulę  $(\neg r \vee p)$

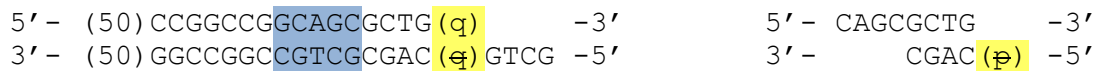


Ryc. 143f: molekula terminalna wnioskowania

Reakcja będzie wyglądać następująco:



Ryc. 144a: ligacja molekul reprezentujących klauzule  $\neg q$  oraz  $(\neg p \vee q)$



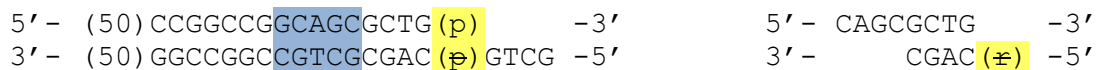
Ryc. 144b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 144c: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg p$ )



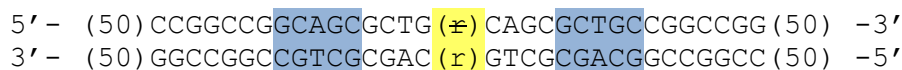
Ryc. 144d: ligacja molekul reprezentujących klauzule  $\neg p$  oraz  $(\neg r \vee p)$



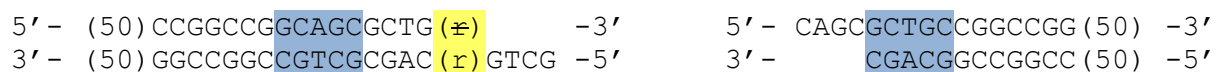
Ryc. 144e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



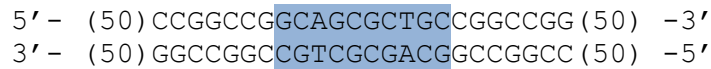
Ryc. 144f: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg r$ )



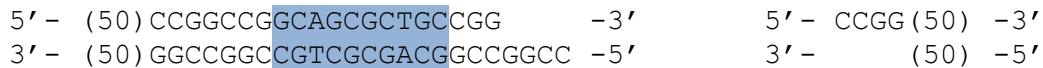
Ryc. 144g: ligacja molekul reprezentujących klauzule  $r$  oraz  $\neg r$



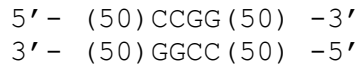
Ryc. 144h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 144i: ligacja molekuly terminalnej i powstałej cząsteczki



Ryc. 144j: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI



Ryc. 144k: ligacja powstałych molekul ze sobą

Powstała cząsteczka symbolizuje pojawienie się sprzeczności, a zatem w świetle twierdzenia o dowodzeniu nie-wprost oznacza możliwość wyprowadzenia tezy czyli twierdzącą odpowiedź systemu na zadane pytanie.

Oczywiście w przypadku zapytania o alternatywę niemożliwą do wyprowadzenia na podstawie powyższych aksjomatów początkowych (np.  $p \vee q$ ) cząsteczka symbolizująca odpowiedź twierdzącą nie pojawi się, a reakcja będzie analogiczna do tej pokazanej przy użyciu zapytania o pojedynczą klauzulę faktu (*patrz.*: pierwszy przykład w powyższej części dotyczącej wyprowadzania alternatywy z innych alternatyw).

### Wyprowadzenie większej liczby faktów wyłącznie na podstawie alternatyw

Omawiana koncepcja zastosowania dowodu nie wprost pozwoliła rozwiązać wspomniany wcześniej problem wyprowadzania informacji o prawdziwości pojedynczego literału wyłącznie na podstawie alternatyw (*patrz.*: podrozdział 5.1.3). Niestety jeżeli chcielibyśmy skorzystać z wyprowadzenia ich większej liczby w taki sposób, a następnie dokonać wnioskowania wyłącznie na ich podstawie, to ponownie spotkamy się z problemem technicznym – braku sekwencji rozpoznawalnej przez wykorzystywany enzym restrykcyjny BbvI. Zapisanie tezy składającej się wyłącznie z koniunkcji, gdzie jej zaprzeczenie będzie miało postać alternatywy, sprawi że nie brałaby udziału w reakcji żadna molekula zawierająca sekwencję rozpoznawalną przez enzym, a zatem taka reakcja wnioskowania nie jest możliwa do przeprowadzenia z użyciem tylko jednej próbki.

Przykładem takiej sytuacji jest:

Aksjomaty początkowe:  $(\neg p \vee x)$ ,  $(p \vee x)$ ,  $(\neg q \vee y)$ ,  $(q \vee y)$ .

Zapytanie:  $(x \wedge y)?$ , zatem zaprzeczenie zapytania:  $(\neg x \vee \neg y)$ .

Z alternatyw  $(\neg p \vee x)$  oraz  $(p \vee x)$  możliwe jest wyprowadzenie prawdziwości  $x$ . Z alternatyw  $(\neg q \vee y)$  oraz  $(q \vee y)$  możliwe jest wyprowadzenie prawdziwości  $y$ . Zatem z punktu widzenia logiki matematycznej odpowiedź na zadawane pytanie powinna być twierdząca. Niestety wszystkie molekuly reprezentujące te wartości będą miały postać klauzul złożonych, a co za tym idzie żadna z nich nie zawiera sekwencji rozpoznawalnej przez enzym restrykcyjny. W tym przypadku system nie zwróci odpowiedzi o prawdziwości zapytania.

Rozwiązaniem tej sytuacji ponownie okazuje się zastosowanie dwóch (lub analogicznej do liczby składowych koniunkcji) próbek. Możemy wykonać osobno reakcję dla zapytania  $x?$ , osobno dla zapytania  $y?$ , a następnie w sposób nieautonomiczny połączyć odpowiedzi i uznać prawdziwość tezy, w świetle twierdzenia o dowodzeniu nie-wprost, tylko i wyłącznie

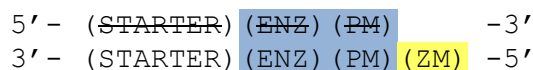
gdy w obu probówkach uzyskamy informację o sprzeczności. Zastosowanie większej liczby probówek nie wpływa na autonomiczność omawianego systemu. Wszystkie reakcje odbywają się równocześnie i bez dodatkowej ingerencji laboranta.

### 5.3.2. Inne enzymy restrykcyjne

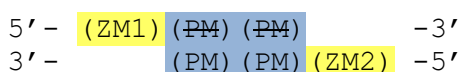
Na rynku komercyjnym mamy dostęp do sporej liczby enzymów restrykcyjnych, rozpoznających różnej długości sekwencje nukleotydów, rozcinających łańcuch wewnątrz lub w różnych od owej sekwencji odległościach, pozostawiających różnej długości „lepkie końce” lub niepozostawiające ich w ogóle (*patrz*: podrozdział 2.1.2). Po szczegółowych analizach, enzym restrykcyjny *BbvI* okazał się najlepszy na potrzeby omawianego systemu wnioskującego: rozcina łańcuch w odległości 8 i 12 nukleotydów od sekwencji rozpoznawalnej, które to liczby są wielokrotnością długości pozostawianego w wyniku cięcia „lepkiego końca”. Jego sekwencja rozpoznawalna składa się z 5-ciu nukleotydów, co pozwala też na zawarcie wyłącznie jej 4-nukleotydowego fragmentu wewnątrz niektórych molekuł (dzięki czemu enzym rozpocznie działanie dopiero w przypadku przyłączenia się brakującego fragmentu sekwencji rozpoznawalnej).

Regularnie, w wyniku nowych odkryć biochemików, pojawiają się też nowe enzymy restrykcyjne, zatem istnieje możliwość rozważenia, które z nich hipotetycznie mogłyby zostać wykorzystane aby zwiększyć możliwości obliczeniowe omawianego systemu, poprzez zwiększenie liczby możliwych do zakodowania zmiennych i zapytań (*patrz*: podrozdziały 5.1.1 i 5.1.2).

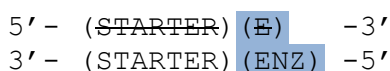
Przyjrzyjmy się raz jeszcze konstrukcji podstawowych molekuł:



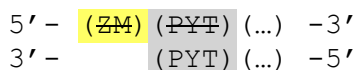
Ryc. 145a: uproszczony schemat molekuly reprezentującej klauzulę faktu



Ryc. 145b: uproszczony schemat molekuly reprezentującej klauzulę złożoną dwuelementową



Ryc. 145c: uproszczony schemat molekuly terminalnej wnioskowania



Ryc. 145d: uproszczony schemat molekuly reprezentującej zapytanie

W powyższych uproszczeniach (ENZ) oznacza sekwencję rozpoznawalną przez enzym restrykcyjny, (PM) fragment pomocniczy, będący w rzeczywistości niepełną sekwencją rozpoznawalną przez enzym restrykcyjny (zapisany w formie symetrycznej z zachowaniem kierunku nici), (E) sekwencję brakującą w (PM) (co oznacza że sklejenie (E) oraz (PM) odpowiada dokładnie ciągowi (ENZ)), (ZM) sekwencję jednoznacznie identyfikującą dany literał a (PYT) sekwencję jednoznacznie identyfikującą dane zapytanie.

Jeżeli przez  $n$  oznaczymy długość „lepkiego końca”, to możemy stwierdzić, że dla  $n = 4$  odpowiadający enzym restrykcyjny (tu: *BbvI*) rozcina molekułę w odległości  $2n$  i  $3n$  czyli 8 i 12. Długość jego sekwencji rozpoznawalnej wynosi 5, dzięki czemu fragment pomocniczy

(PM) zawierający jego fragment o długości  $n = 4$  nie reprezentuje całej sekwencji rozpoznawalnej (dzięki czemu niemożliwe jest cięcie w niepożądanym miejscu). Zatem brakujący fragment (E) posiada długość 1 (ale w przypadku innych enzymów o dłuższej sekwencji rozpoznawalnej również mógłby być dłuższy). Długość sekwencji reprezentującej pojedynczy literał czy pojedyncze zapytanie zawsze odpowiada długości pozostawianego przez enzym „lepkiego końca”, tu zatem również wynosi  $n = 4$ .

W takim razie aby „wymienić” enzym restrykcyjny na nowy, pozostawiający „lepkie końce” o długości dowolnego  $n$ , niezbędne jest aby długość jego sekwencji rozpoznawalnej była większa od  $n$  o co najmniej 1 a odległość między tą sekwencją a punktem cięcia wynosiła dokładnie  $2n$  i  $3n$ . Długości wszystkich pozostałych przedstawionych powyżej fragmentów muszą zostać również odpowiednio dopasowane do  $n$ , w taki sposób że (PM) zawiera  $n$ -nukleotydów sekwencji rozpoznawalnej, (E) pozostałe nukleotydy tej sekwencji (co oznacza że zawsze sklejenie (E) oraz (PM) odpowiada dokładnie ciągowi (ENZ)). Ciągi (ZM) i (PYT) mają długość równą  $n$ .

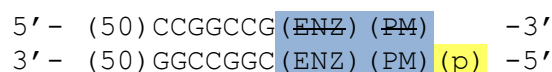
Drobna modyfikacja pojawia się dla  $n$  nieparzystych i molekuł zapytań, ponieważ nie istnieje wtedy możliwość utworzenia łańcucha (PYT), który będzie komplementarny z samym sobą. Możliwe jest wtedy zastosowanie analogicznych sekwencji jak w przypadku potrzebnej większej liczby zapytań (*patrz*: podrozdział 5.1.2) co powoduje odliczenie ich od puli reprezentującej zmienne zdaniowe. Niezbędne jest również odpowiednie dopasowanie startera, tak żeby reakcja wykrycia sprzeczności działała poprawnie (używa ona również sekwencji komplementarnej z samą sobą, zatem dla enzymów o nieparzystym  $n$ , niezbędne byłoby zastosowanie dwóch różnych starterów, które w analogicznej sytuacji będą ze sobą komplementarne).

Dla  $n$  parzystych nie występują żadne omówione w powyższym akapicie modyfikacje – startery molekuł faktu oraz zapytania będą analogiczne jak dla przypadku  $n = 4$ .

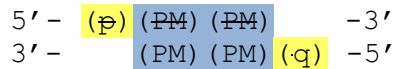
Dzięki takiej modyfikacji, jeżeli dany nowy enzym okaże się dostępny komercyjnie, otrzymujemy możliwość zakodowania co najmniej  $(3 * 4^{n-2})$  różnych zmiennych lub zapytań (pula wspólna). Dla parzystych  $n$  będzie ich jeszcze o kilka więcej (tak jak np. dla  $n = 4$ , obliczona wartość wg powyższego wzoru to 48, a uzyskana w wyniku szczegółowej analizy to 66). Znaczną różnicę wartości minimalnej widać już przy niewielkim wzroście  $n$ :

$$\begin{aligned} \text{dla } n = 5: & \quad 3 * 4^3 = 3 * 64 = 192, \\ \text{dla } n = 6: & \quad 3 * 4^4 = 3 * 256 = 768, \\ \text{dla } n = 7: & \quad 3 * 4^5 = 3 * 1024 = 3072, \\ \text{dla } n = 8: & \quad 3 * 4^6 = 3 * 4096 = 12288, \\ & \quad (\dots). \end{aligned}$$

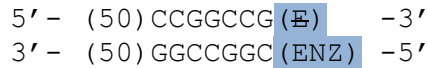
Przyjrzyjmy się przykładowej reakcji dla teoretycznego systemu o dowolnej wartości  $n$ , będącej liczbą parzystą, oraz dla następujących klauzul:  $p$ ,  $(\neg p \vee q)$  oraz zapytania  $q$ ? Molekuły wyglądają następująco:



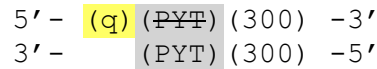
Ryc. 146a: molekula reprezentująca klauzulę faktu  $p$



Ryc. 146b: molekula reprezentująca klauzulę  $(\neg p \vee q)$

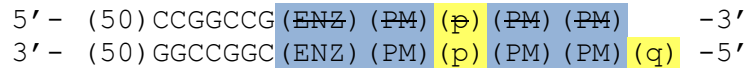


Ryc. 146c: molekula terminalna wnioskowania

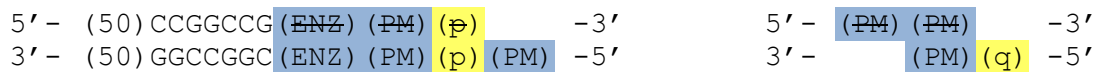


Ryc. 146d: molekula zapytania  $\neg q$ ?

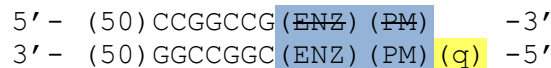
Reakcja będzie wyglądać następująco:



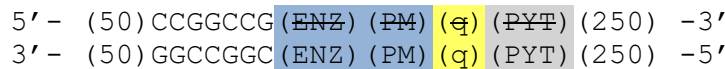
Ryc. 147a: ligacja molekuł reprezentujących klauzule p oraz  $(\neg p \vee q)$



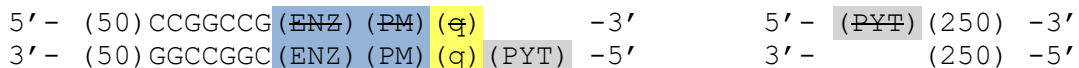
Ryc. 147b: rozcięcie molekuly za pomocą enzymu restrykcyjnego



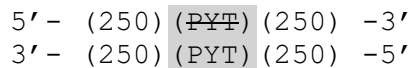
Ryc. 147c: ligacja molekuly terminalnej i wniosku częściowego



Ryc. 147d: ligacja powstałej molekuly reprezentującej q i molekuly zapytania



Ryc. 147e: rozcięcie molekuly za pomocą enzymu restrykcyjnego

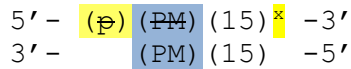


Ryc. 147f: ligacja powstałych molekuł ze sobą

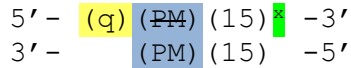
Pojawienie się molekuly zgodnej z powyższym schematem, niezależnie od wielkości  $n$ , oznacza pojawienie się odpowiedzi twierdzącej na zadane pytanie wprowadzone do systemu. Jeżeli przez  $k$  oznaczymy długość dowolnych molekuł umieszczonych na końcu molekuly zapytania, to zawsze pojawienie się molekuly o długości  $(2*k + n)$  oznacza dokładnie odpowiedź twierdzącą na zadane pytanie.

### 5. 3. 3. Zastosowanie fluorescencji

Istnieje możliwość modyfikacji przedstawionych w omawianym systemie zapytań, w taki sposób by opierając się na doświadczeniach laboratoryjnych przedstawionych w pracy zespołu badawczego Shapiro (*patrz*: podrozdział 3.2.2) [27], wykorzystać fluorescencję do wykrycia odpowiedzi twierdzącej zamiast reakcji elektroforezy w żelu. Do tej modyfikacji zostanie wykorzystana molekula terminalna wnioskowania (co za tym idzie nie ma potrzeby wprowadzania dodatkowych molekuł), która w wyniku połączenia z przeciętą wcześniej cząsteczką zapytania, dokona cięcia wywołującego świecenie. Dalej posługując się przedstawionym w poprzednim podrozdziale uproszczonym schematem, przykładowe molekuly zapytań wyglądają następująco (między sobą różnią się kolorem fluorescencji):

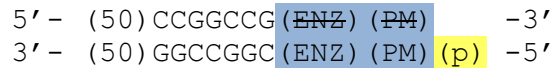


Ryc. 148a: molekula zapytania p?

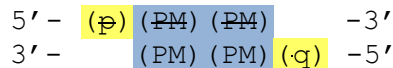


Ryc. 148b: molekula zapytania -q?

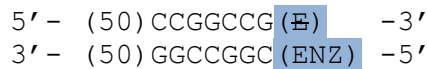
Przyjrzyjmy się przykładowej reakcji ze zmodyfikowanym zapytaniem, dla klauzul: p, ( $\neg p \vee q$ ) oraz zapytaniu q? Molekuły wyglądają następująco:



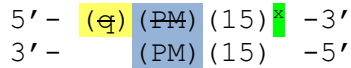
Ryc. 149a: molekula reprezentująca klauzulę faktu p



Ryc. 149b: molekula reprezentująca klauzulę ( $\neg p \vee q$ )

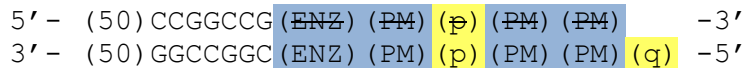


Ryc. 149c: molekula terminalna wnioskowania

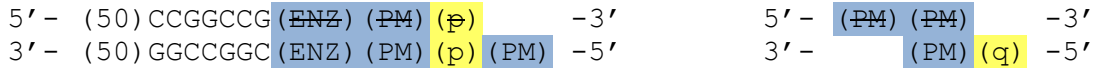


Ryc. 149d: molekula zapytania q?

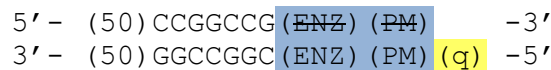
Reakcja będzie wyglądać następująco:



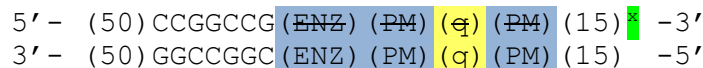
Ryc. 150a: ligacja molekul reprezentujących klauzule p oraz ( $\neg p \vee q$ )



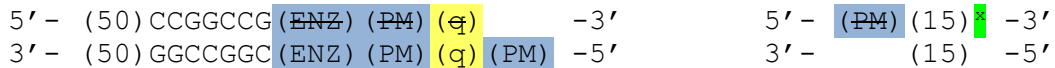
Ryc. 150b: rozcięcie molekuly za pomocą enzymu restrykcyjnego



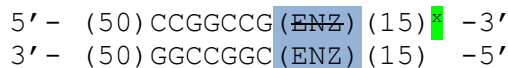
Ryc. 150c: ligacja molekuly terminalnej i wniosku częściowego



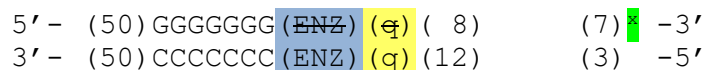
Ryc. 150d: ligacja powstałej molekuly reprezentującej q i molekuly zapytania



Ryc. 150e: rozcięcie molekuly za pomocą enzymu restrykcyjnego



Ryc. 150f: ligacja molekuly terminalnej i wniosku częściowego



Ryc. 150g: rozcięcie molekuly za pomocą enzymu restrykcyjnego



Ryc. 150h: rozpad powstałej cząstki na dwie jednonicowe oraz uaktywnienie fluorescencji

Pojawienie się świecenia w zadanym przez zapytanie  $q$ ? kolorze oznacza pozytywną odpowiedź na pytanie czy fakt  $q$  jest logiczną konsekwencją wprowadzonych danych wejściowych.

Do zalet takiej modyfikacji należy przede wszystkim zwolnienie wykorzystywanych wcześniej unikalnych sekwencji, reprezentujących każde z zapytań (gdzie niektóre z nich mogą zostać wykorzystane do reprezentacji kolejnych faktów; *patrz*: podrozdział 5.1.2). Za minus należy potraktować to, że w przypadku zadania większej liczby pytań w obrębie tej samej próbki, trudniejsza może być identyfikacja kolorów aby bezbłędnie wiedzieć, które z cząsteczek fluorescencyjnych uległy uaktywnieniu, a które nie.

## 6. Symulacja eksperymentów laboratoryjnych

W celu potwierdzenia słuszności omawianej koncepcji, w szczególności jej innowacyjnych elementów (np. symetrycznie skonstruowanych dwuniciowych fragmentów DNA, umożliwiających ich rozcięcie za pomocą enzymu restrykcyjnego *BbvI* z obu stron, czy reakcji wykrycia sprzeczności), niezbędne jest przeprowadzenie badań laboratoryjnych.

W podrozdziale 6.1 zostaną przedstawione wybrane propozycje eksperymentów (mających na celu potwierdzenie przedstawionego modelu wnioskowania, w tym: ujęcie reguły kontrapozycji, wykrycia odpowiedzi twierdzącej, wykrycia sprzeczności oraz pokazanie złożonego procesu wnioskowania). W ich opisie jak i w dalszej części pracy zamiast używanego wcześniej zapisu ogólnego np.  $3' - (p) - 5'$ , wprowadzone zostały już konkretne sekwencje nukleotydów identyfikujących poszczególne literały np.  $3' - AAAG - 5'$ . Ma to praktyczne znaczenie przy zamawianiu sekwencji DNA do przeprowadzenia eksperymentów.

W podrozdziale 6.2 przedstawione zostaną techniczne szczegóły przygotowania cząsteczek przed właściwą reakcją w laboratorium. Pokazana zostanie również koncepcja stworzenia i wykorzystania „biblioteki łańcuchów DNA”, umożliwiającej przechowywanie i dalsze zastosowanie uzyskanych sekwencji. Składa się ona z następujących kroków:

- zamówienie oligonukleotydów z firmy komercyjnej,
- połączenie jednoniciowych oligonukleotydów w cząsteczki dwuniciowe,
- wklonowanie otrzymanych fragmentów DNA do wektora plazmidowego (tutaj *pJET 1.2*) w miejsce cięcia wybranymi enzymami (tutaj *MfeI* i *PspFI*, patrz: podrozdział 6.2.2),
- wprowadzenie wektorów do komórek bakteryjnych (*Escherichia coli*), co umożliwia powielanie wprowadzonych łańcuchów wraz z powielaniem komórek bakteryjnych,
- selekcja wprowadzonych wcześniej sekwencji z wektorów plazmidowych za pomocą reakcji PCR, tuż przed eksperymentem (z utworzonej „biblioteki” pobieramy wtedy tylko fragmenty dwuniciowe niezbędne dla danej reakcji, patrz: podrozdział 6.2.3).

W podrozdziale 6.3. omówiony zostanie przebieg głównej reakcji wnioskowania (która obrazuje teoretyczne działanie systemu, przedstawione w rozdziałach 4 i 5) oraz odczytanie końcowej odpowiedzi systemu.

Analogicznie jak we wcześniejszych rozdziałach, w niektórych miejscach zastosowany został uproszczony zapis gdzie *N* oznacza dowolny nukleotyd, wyrażenia liczbowe np. (10), (50) itp. oznaczają zadaną liczbę dowolnych nukleotydów, a wyrażenie (...) sekwencję dowolnych nukleotydów o niesprecyzowanej długości. Ze względu na praktyczny charakter tego rozdziału, wspomniane uproszczenia używane są tylko tam gdzie zwiększają czytelność zapisu.

Ze względu na wysokie koszty oraz brak środków finansowych eksperymenty wciąż nie zostały przeprowadzone w laboratorium. Omówione poniżej konkretne kroki ich przeprowadzenia zostały skonsultowane z Katedrą Genetyki Molekularnej Wydziału Biologii i Ochrony Środowiska Uniwersytetu Łódzkiego, z którą od lat nasza Katedra współpracuje w zakresie badań związanych z obliczeniami DNA. Możliwość przeprowadzenia takich badań



została przez nich potwierdzona. W przyszłości, w miarę możliwości finansowych, planowana jest praktyczna implementacja przygotowanych eksperymentów.

## 6. 1. Wybór eksperymentów

W celu przedstawienia poprawności działania omawianego systemu wnioskującego, zostały wybrane trzy propozycje różnych eksperymentów. Pokazane zostaną odpowiednio: proste wnioskowanie, wykorzystujące regułę kontrapozycji (podrozdział 6.1.1), reakcja wykrycia sprzeczności (podrozdział 6.1.2) oraz złożone wnioskowanie, na którego przebieg składa się szereg prostych reakcji wnioskowania, wykorzystujących między innymi klauzule złożone oraz prawa de Morgana (podrozdział 6.1.3). Wybór został dokonany w taki sposób aby pokazać poprawność wszystkich rozszerzeń wprowadzonych względem znanych wcześniej implementacji systemów wnioskujących, w tym również dla wnioskowania złożonego.

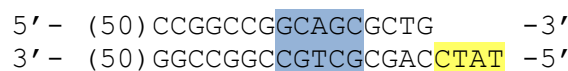
### 6. 1. 1. Wnioskowanie i reguła kontrapozycji

Pierwszym eksperymentem będzie proste wnioskowanie (*patrz:* podrozdział 4.2.4), wykorzystujące jedną klauzulę faktu, jedną klauzulę złożoną i jedno zapytanie. Wykorzystana zostanie również reguła kontrapozycji, pokazująca zaletę użycia symetrycznych cząsteczek reprezentujących klauzule złożone.

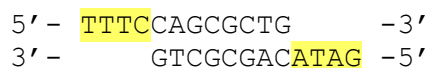
Przyjmijmy następujące aksjomaty wejściowe:  $\neg q$ ,  $(\neg p \vee q)$  oraz zapytanie  $\neg p?$ .

Przyjmijmy reprezentację  $p$  za pomocą 3'-AAAG-5' oraz  $q$  3'-ATAG-5', a zatem reprezentację  $\neg p$  za pomocą 3'-CTTT-5' oraz  $\neg q$  3'-CTAT-5'.

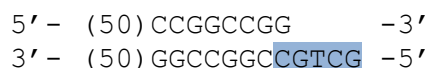
Sekwencje fragmentów DNA wyglądają następująco:



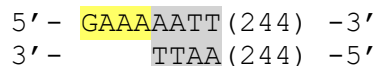
Ryc. 151a: sekwencja reprezentująca klauzulę faktu  $\neg q$



Ryc. 151b: sekwencja reprezentująca klauzulę  $(\neg p \vee q)$



Ryc. 151c: sekwencja terminalna wnioskowania



Ryc. 151d: sekwencja reprezentująca zapytanie  $\neg p?$

Proces przygotowania odpowiednich fragmentów DNA, ich reakcji oraz odczytanie wyniku zostanie szczegółowo omówiony poniżej, w kolejnych podrozdziałach.

### 6. 1. 2. Wykrycie sprzeczności

Drugim eksperymentem będzie wykrycie sprzeczności (*patrz:* podrozdział 4.3). Wykorzystana zostanie również prosta reakcja wnioskowania, w wyniku której otrzymany zostanie wniosek sprzeczny z aksjomatem wejściowym.

Przyjmijmy następujące aksjomaty wejściowe:  $p$ ,  $\neg q$ ,  $(\neg p \vee q)$ .

Przyjmijmy reprezentację  $p$  za pomocą  $3' -AAAG-5'$  oraz  $q$   $3' -ATAG-5'$ , a zatem również reprezentację  $\neg p$  za pomocą  $3' -CTTT-5'$  oraz  $\neg q$   $3' -CTAT-5'$ .

Sekwencje fragmentów DNA wyglądają następująco:

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
3' - (50) GGCCGGC **CGTCG** CGAC **AAAG** -5'

*Ryc. 152a: sekwencja reprezentująca klauzulę faktu  $p$*

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
3' - (50) GGCCGGC **CGTCG** CGAC **CTAT** -5'

*Ryc. 152b: sekwencja reprezentująca klauzulę faktu  $\neg q$*

5' - **TTTC** CAGCGCTG -3'  
3' - GTCGCGAC **ATAG** -5'

*Ryc. 152c: sekwencja reprezentująca klauzulę  $(\neg p \vee q)$*

5' - (50) CCGGCCGG -3'  
3' - (50) GGCCGGC **CGTCG** -5'

*Ryc. 152d: sekwencja terminalna wnioskowania*

Proces przygotowania odpowiednich fragmentów DNA, ich reakcji oraz odczytanie wyniku zostanie szczegółowo omówiony poniżej, w kolejnych podrozdziałach.

### 6. 1. 3. Złożone wnioskowanie i prawa de Morgana

Trzecim eksperymentem będzie złożone wnioskowanie. Wykorzystane zostanie kilka klauzul złożonych, w tym klauzula składająca się z więcej niż dwóch literałów. Użyte zostaną również: reguła przechodniości implikacji, prawa de Morgana oraz reguła kontrapozycji.

Przyjmijmy następujące aksjomaty wejściowe:  $a$ ,  $\neg d$ ,  $\neg e$ ,  $(\neg a \vee \neg b \vee c \vee d)$ ,  $(b \vee e)$ . Dla wykrycia wniosku wykorzystajmy zapytanie:  $c$ ?

Przyjmijmy reprezentację  $a$  za pomocą  $3' -TAAG-5'$ ,  $b$   $3' -TTCA-5'$ ,  $c$   $3' -TTGT-5'$ ,  $d$   $3' -CCCA-5'$ ,  $e$   $3' -ATAG-5'$ , a zatem również reprezentację  $\neg a$  za pomocą  $3' -CTTA-5'$ ,  $\neg b$   $3' -TGAA-5'$ ,  $\neg c$   $3' -ACAA-5'$ ,  $\neg d$   $3' -TGGG-5'$ ,  $\neg e$   $3' -CTAT-5'$ .

Sekwencje fragmentów DNA wyglądają następująco:

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
3' - (50) GGCCGGC **CGTCG** CGAC **TAAG** -5'

*Ryc. 153a: sekwencja reprezentująca klauzulę faktu  $a$*

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
3' - (50) GGCCGGC **CGTCG** CGAC **TGGG** -5'

*Ryc. 153b: sekwencja reprezentująca klauzulę faktu  $\neg d$*

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
3' - (50) GGCCGGC **CGTCG** CGAC **CTAT** -5'

*Ryc. 153c: sekwencja reprezentująca klauzulę faktu  $\neg e$*

5' - **ATTC** CAGCGCTG **ACTTAAGT** CAGCGCTG **AACATGTT** CAGCGCTG -3'  
3' - GTCGCGAC **TGAATTCA** GTCGCGAC **TTGTACAAGT** CGCGAC **CCCA** -5'

*Ryc. 153d: sekwencja reprezentująca klauzulę  $(\neg a \vee \neg b \vee c \vee d)$*

5' - **ACTT** CAGCGCTG -3'  
3' - GTCGCGAC **ATAG** -5'

*Ryc. 153e: sekwencja reprezentująca klauzulę  $(b \vee e)$*

5' - (50) CCGGCCGG -3'  
 3' - (50) GGCCGGC **CGTCG** -5'

Ryc. 153f: sekwencja terminalna wnioskowania

5' - **AACA**AATT (244) -3'  
 3' - **TTAA** (244) -5'

Ryc. 153g: sekwencja reprezentująca zapytanie -c?

Proces przygotowania odpowiednich fragmentów DNA, ich reakcji oraz odczytanie wyniku zostanie szczegółowo omówiony poniżej, w kolejnych podrozdziałach.

## 6. 2. Przygotowanie cząsteczek DNA w laboratorium

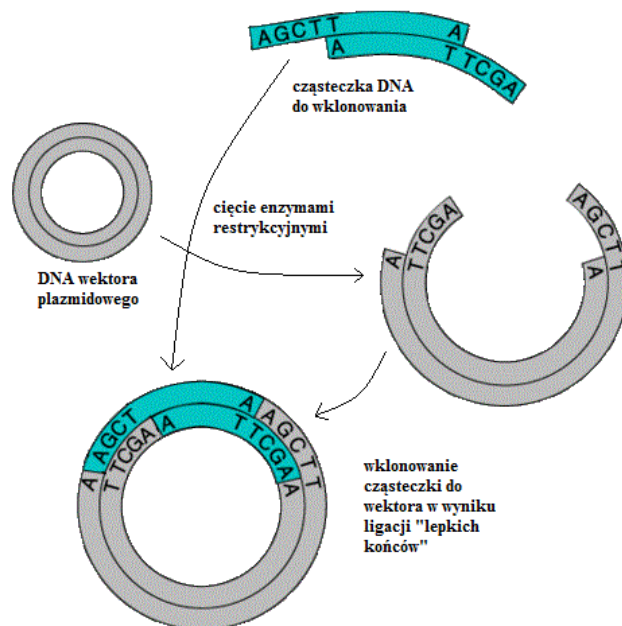
Kluczowym elementem eksperymentu jest odpowiednie przygotowanie składowych komputera DNA. Z teoretycznego punktu widzenia byłyby to wyłącznie następujące kroki: zamówienie jednoniciowych łańcuchów DNA z firmy komercyjnej, połączenie ich w cząsteczki dwuniciowe zgodnie z zasadą komplementarności, przeprowadzenie właściwej reakcji z wykorzystaniem enzymu restrykcyjnego *BbvI* i enzymu ligazy, a następnie odczytanie wyniku za pomocą elektroforezy w żelu.

Jednak z biochemicznego punktu widzenia, aby ograniczyć koszty, a także wykorzystywać cząsteczki dwuniciowe (uzyskane z zamówionych oligonukleotydów) do kolejnych doświadczeń, istnieje możliwość użycia wypełniacza, a następnie przygotowania „biblioteki łańcuchów DNA”.

Wypełniacz (z ang. *stuffer*) to odpowiednio dobrana sekwencja nukleotydów rozszerzająca każdą ze składowych systemu (z obu jej stron) do postaci o określonej wspólnej sekwencji „lepkich końców” (identycznych jak te pozostawione po rozcięciu wektora plazmidowego enzymami *MfeI* oraz *PspFI*, patrz: podrozdział 6.2.2). Wszystkie składowe omawianego komputera DNA będą różnić się między sobą wyłącznie centralną sekwencją nukleotydów. Przygotowane w ten sposób cząsteczki będą gotowe do wklonowania do wektora plazmidowego *pJET 1.2*, a każda z nich składać się będzie z następujących elementów:

- wypełniacza o „lepkim końcu” odpowiadającym miejscu rozcięcia wektora plazmidowego przez enzym restrykcyjny *MfeI*,
- sekwencji reprezentującej dany element systemu wnioskującego (innej dla reprezentacji klauzuli faktu, klauzuli złożonej, zapytania czy cząsteczki terminalnej, omówionych szczegółowo w teoretycznej części pracy),
- wypełniacza o „lepkim końcu” odpowiadającym miejscu rozcięcia wektora plazmidowego przez enzym restrykcyjny *PspFI*.

Taki proces umożliwi przygotowanie „biblioteki łańcuchów DNA” gotowych do przechowywania i ponownego użycia w podobnych eksperymentach. Uzyskane cząsteczki wektora plazmidowego (przedstawione na rycinie poniżej), z wprowadzonymi sekwencjami reprezentującymi składowe systemy wnioskującego, są następnie namnażane w komórkach bakteryjnych *Escherichia coli*. Podobny proces został szczegółowo przedstawiony i wykorzystywany między innymi w [15], [40], [41], [42].



Ryc. 154: schemat wklonowania sekwencji DNA do wektora plazmidowego

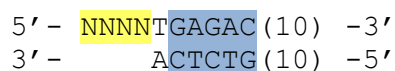
Pokazane dalej procesy nie zostały omówione we wcześniejszej części pracy ze względu na wyłącznie teoretyczny charakter rozdziałów 4 i 5. Wszystkie poniższe schematy stanowią konkretne kroki przygotowania i przeprowadzenia eksperymentu.

### 6. 2. 1. Przygotowanie fragmentów DNA do wklonowania w wektor *pJET 1.2*.

Przed zamówieniem łańcuchów z firmy komercyjnej należy każdą cząsteczkę, przedstawioną w części 6.1, dopełnić o dodatkową sekwencję wypełniacza, odpowiadającą „lepkim końcom” przygotowanego wektora *pJET 1.2*. W przypadku omawianego systemu, składają się na to dwa kroki: uzupełnienie sekwencji DNA do postaci dwuniciowej o identycznych „tępych końcach” oraz dołączenie „lepkich końców” dopasowanych do wektora plazmidowego (*patrz*: podrozdział 6.2.1.a). Dopiero tak przygotowane sekwencje nukleotydów mogą być zamawiane w firmie komercyjnej jako oligonukleotydy (*patrz*: podrozdział 6.2.1.b).

#### a) uzupełnienie przedstawionych fragmentów DNA

W pierwszym kroku uzupełnione zostają zaproponowane w części teoretycznej „lepkie końce”, w taki sposób by każdy z omawianych dotąd łańcuchów został przedstawiony w postaci dwuniciowej, zakończonej z obu stron „tępym końcem”. W przypadku cząsteczki reprezentującej klauzulę złożoną, zostanie ona uzupełniona z obu jej stron. Omawiany dodatkowy fragment będzie zawierał sekwencję nukleotydów GAGAC/CTCTG rozpoznawalną przez enzym restrykcyjny *BsmAI* (który umożliwi później wycięcie dopełnionych „lepkich końców” z powielonych cząsteczek tuż przed główną reakcją). Ma on następującą postać (gdzie za pomocą **NNNN** oznaczony został fragment komplementarny do „lepkiego końca” dowolnej cząsteczki przed omawianym rozszerzeniem):



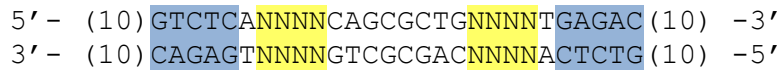
Ryc. 155: schemat sekwencji dopełniającej każdą z cząsteczek do postaci dwuniciowej o „tępych końcach”

Przykładowa sekwencja faktu po dopełnieniu wygląda następująco:



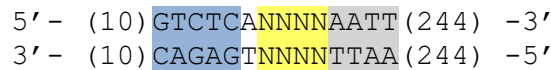
Ryc. 156: schemat sekwencji reprezentującej klauzulę faktu po dopełnieniu do postaci dwuniciowej o „tępych końcach”

Przykładowa sekwencja klauzuli złożonej po dopełnieniu (z obu stron) wygląda następująco:



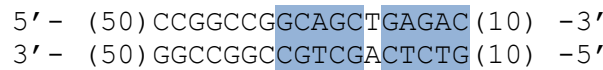
Ryc. 157: schemat sekwencji reprezentującej klauzulę złożoną po dopełnieniu do postaci dwuniciowej o „tępych końcach”

Przykładowa sekwencja zapytania po dopełnieniu wygląda następująco:



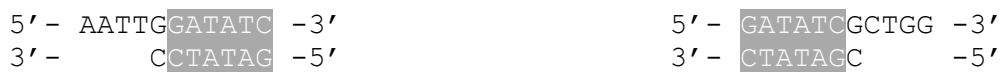
Ryc. 158: schemat sekwencji reprezentującej zapytanie po dopełnieniu do postaci dwuniciowej o „tępych końcach”

Seqwencja terminalna wnioskowania po dopełnieniu wygląda następująco:



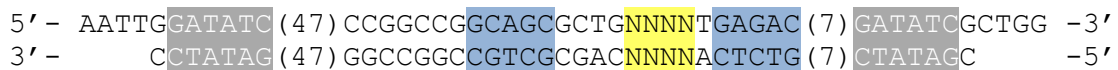
Ryc. 159: sekwencja terminalna wnioskowania po dopełnieniu do postaci dwuniciowej o „tępych końcach”

Do wklonowania powyższych fragmentów DNA w wektor plazmidowy, niezbędne jest dołączenie także sekwencji reprezentujących „lepkie końce”, umożliwiające integrację omawianych fragmentów w oczekiwane miejsce wektora, które uzyskamy po jego rozcięciu za pomocą enzymów *MfeI* oraz *PspFI* (patrz: podrozdział 6.2.2), czyli wypełniacza. W obrębie dołączanych fragmentów znajduje się również sekwencja 5'-GATATC-3' / 3'-CTATAG-5' rozpoznawalna przez enzym restrykcyjny *Eco32I*, który umożliwi wycięcie „tępych końców”, przedstawionych na rycinach 156-159, po selekcji łańcuchów z „biblioteki”, tuż przed właściwą reakcją. Dołączane w kolejnym kroku „lepkie końce” wypełniacza stanowią fragmenty sekwencji rozpoznawalnych przez enzymy *MfeI* (5'-AATTG-3') oraz *PspFI* (5'-GCTGG-3'), które zostaną użyte do nacięcia wektora plazmidowego w dwóch miejscach. Wypełniacz, służący do dopełnienia cząsteczek przedstawionych na rycinach 156-159, ma następującą postać:



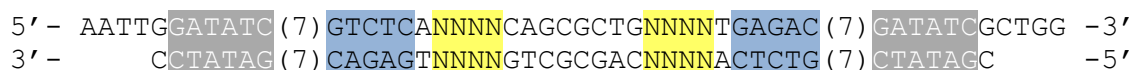
Ryc. 160: dołączane fragmenty wypełniacza, mające umożliwić jednoznaczne wklonowanie do wektora

Przykładowa cząsteczka DNA faktu po dopełnieniu wygląda następująco:



Ryc. 161: schemat sekwencji reprezentującej klauzulę faktu po dołączeniu wypełniacza

Przykładowa cząsteczka DNA klauzuli złożonej po dopełnieniu wygląda następująco:



Ryc. 162: schemat sekwencji reprezentującej klauzulę złożoną po dołączeniu wypełniacza



unikalną dla danego literału czwórkę nukleotydów oraz fragmenty rozpoznawalne przez enzymy *BbvI* i *BsmAI* (5'-GCTTCCGGCCGGCAGCGCTGNNNNTGAGACTTA-3' oraz 3'-AATCGAAGGCCGGCCGTCGCGACNNNNACTCTG-5') oraz końcową sekwencję wypełniacza rozpoznawalną przez enzymy *PspFI* i *Eco32I* (5'-GCTTGATATCGCTGG-3' oraz 3'-AATCGAACTATAGC-5').

- Sekwencja reprezentująca pojedynczą klauzulę złożoną (*patrz.*: rycina 162) dzieli się na kilka następujących fragmentów: początkową sekwencję wypełniacza rozpoznawalną przez enzymy *MfeI* oraz *Eco32I* (5'-AATTGGATATCTTA-3' oraz 3'-CCTATAG-5'), sekwencję reprezentującą daną klauzulę złożoną (zawierającą unikalne czwórki powiązanych z nią literałów) (5'-GCTTGTCTCANNNNCA-GCGCTGNNNNTGAGACTTA-3' oraz 3'-AATCGAACAGAGTNNNNGTCGCGACNNNNACTCTG-5') (lub dłuższą, podzieloną na kilka sekwencji w przypadku klauzul bardziej złożonych) oraz końcową sekwencję wypełniacza rozpoznawalną przez enzymy *PspFI* i *Eco32I* (5'-GCTTGATATCGCTGG-3' oraz 3'-AATCGAACTATAGC-5').
- Sekwencja reprezentująca pojedyncze zapytanie (*patrz.*: rycina 163) dzieli się na kilka następujących fragmentów: początkową sekwencję wypełniacza rozpoznawalną przez enzymy *MfeI* oraz *Eco32I* (5'-AATTGGATATCTTA-3' oraz 3'-CCTATAG-5'), właściwą sekwencję reprezentującą dane zapytanie (zawierającą unikalne czwórki powiązanych z nim literałów) (5'-GCTTGTCTCANNNNAAATTTTA-3' oraz 3'-AATCGAACAGAGTNNNNTTAA-5'), sekwencje uniwersalne (o długości 41 nukleotydów) (składające się z ciągu o długości 10 nukleotydów, powtórnego zadaną liczbę razy, tak by uzyskać przewidzianą dla danego zapytania długość, np. dla uzyskania 244 nukleotydów użyte zostały 4 sekwencje o długości 41 nukleotydów oraz 1 sekwencja o długości 32 nukleotydów, *patrz.*: tabele zamówień na kolejnych stronach) oraz końcową sekwencję wypełniacza rozpoznawalną przez enzymy *PspFI* i *Eco32I* (5'-GCTTGATATCGCTGG-3' oraz 3'-AATCGAACTATAGC-5').
- Sekwencja reprezentująca cząsteczkę terminalną wnioskowania (*patrz.*: rycina 164) dzieli się na kilka następujących fragmentów: początkową sekwencję wypełniacza rozpoznawalną przez enzymy *MfeI* oraz *Eco32I* (5'-AATTGGATATCTTA-3' oraz 3'-CCTATAG-5'), sekwencję uzupełniającą (o długości 41 nukleotydów) (5'-GCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTG-3' oraz 3'-CCACGATAGCCACGAATAGCCACGAATAGCCACGAATAGC-5'), właściwą sekwencję terminalną, zawierającą fragmenty rozpoznawalne przez enzymy *BbvI* oraz *BsmAI* (5'-GCTTCCGGCCGGCAGCTGAGACTTA-3' oraz 3'-AATCGAAGGCCGGCCGTCGACTCTG-5') oraz końcową sekwencję wypełniacza rozpoznawalną przez enzymy *PspFI* i *Eco32I* (5'-GCTTGATATCGCTGG-3' oraz 3'-AATCGAACTATAGC-5').

Konkretne sekwencje oligonukleotydów zamawianych w firmie komercyjnej na potrzeby przeprowadzenia poszczególnych eksperymentów (omówionych w podrozdziale 6.1), przedstawione są w poniższych tabelach.

Zamówienie dla doświadczenia przedstawionego w 6.1.1:

Sekwencje oligonukleotydów:	Opis:	Długość:
5' -AATTG <b>GATATC</b> GGT-3' 3' -C <b>CTATAG</b> -5'	Początkowa sekwencja wypełniacza (fragment rozpoznawalny przez enzymy <i>MfeI</i> oraz <i>Eco32I</i> ) – wspólna dla wszystkich cząstek.	14 7
5' -GCTT <b>GATATC</b> GCTGG -3' 3' -AGCCGAA <b>CTATAGC</b> -5'	Końcowa sekwencja wypełniacza (fragment rozpoznawalny przez enzymy <i>PspFI</i> oraz <i>Eco32I</i> ) – wspólna dla wszystkich cząstek.	15 14
5' -GCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTG-3' 3' -CCACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGC-5'	Pierwsza sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -CTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGC-3' 3' -CACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCG-5'	Druga sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -TTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCT-3' 3' -ACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGA-5'	Trzecia sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -TATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTT-3' 3' -CGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGAC-5'	Czwarta sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -ATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTA-3' 3' -GAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGACG-5'	Piąta sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -TCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTAT-3' 3' -AATAGCCACGAATAGCCACGAATAGCCACGAATAGCGACGA-5'	Szosta sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -CGGTGCTTATCGGTGCTTATCGGTGCTTATCG-3' 3' -ATAGCCACGAATAGCCACGAATAGCCACGAAT-5'	Siódma sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	32 32
5' -GCTCCGCGCG <b>GCAGC</b> GCTG <b>GATATGAGAC</b> TCG-3' 3' -CACCGAGCGCGC <b>CGTCG</b> CGAC <b>CTATACTCTG</b> -5'	Sekwencja reprezentująca klauzulę faktu (-q)	32 32



5' -GCTTGTCTCATTTTCAGCGCTGTATCTGAGACTCG-3' 3' -CCACGAA CAGAGTAAAGGTCGCGACATAGACTCTG-5'	Sekwencja reprezentująca klauzulę złożoną ( $\neg p \vee q$ )	35
5' -GCTTGTCTCAGAAAAATTTTCG-3' 3' -AATCGAA CAGAGTCTTTTAA-5'	Sekwencja reprezentująca zapytanie ( $\neg p?$ )	21
5' -GCTCCGGCCGGCAGCTGAGACTCG-3' 3' -CACCGAGGCCGGCCGTCGACTCTG-5'	Sekwencja reprezentująca cząsteczkę terminalną	24

Tab. 2: sekwencje oligonukleotydów zamawiane w firmie komercyjnej dla pierwszego eksperymentu

### Zamówienie dla doświadczenia przedstawionego w 6.1.2:

Sekwencje oligonukleotydów:	Opis:	Długość:
5' -AATTG GATATCGGT-3' 3' -C CTATAG-5'	Początkowa sekwencja wypełniacza (fragment rozpoznawalny przez enzymy <i>MfeI</i> oraz <i>Eco32I</i> ) – wspólna dla wszystkich cząstek.	14 7
5' -GCTT GATATCGCTGG -3' 3' -AGCCGAA CTATAGC-5'	Końcowa sekwencja wypełniacza (fragment rozpoznawalny przez enzymy <i>PspFI</i> oraz <i>Eco32I</i> ) – wspólna dla wszystkich cząstek.	15 14
5' -GCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTG-3' 3' -CCACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGC-5'	Sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -GCTCCGGCCGGCAGCGCTGTTTCTGAGACTCG-3' 3' -CACCGAGGCCGGCCGTCGCGACAAAGACTCTG-5'	Sekwencja reprezentująca klauzulę faktu ( $p$ )	32 32
5' -GCTCCGGCCGGCAGCGCTGGATATGAGACTCG-3' 3' -CACCGAGGCCGGCCGTCGCGACTATACTCTG-5'	Sekwencja reprezentująca klauzulę faktu ( $\neg q$ )	32 32
5' -GCTTGTCTCATTTTCAGCGCTGTATCTGAGACTCG-3' 3' -CCACGAA CAGAGTAAAGGTCGCGACATAGACTCTG-5'	Sekwencja reprezentująca klauzulę złożoną ( $\neg p \vee q$ )	35 35

5' -GCTCCGGCCG <b>GCAGCTGAGAC</b> TCG-3'	Sekwencja reprezentująca cząsteczkę terminalną	24
3' -CACCGAGGCCGGC <b>CGTCGACTCTG</b> -5'		24

Tab. 3: sekwencje oligonukleotydów zamawiane w firmie komercyjnej dla drugiego eksperymentu

### Zamówienie dla doświadczenia przedstawionego w 6.1.3:

Sekwencje oligonukleotydów:	Opis:	Długość:
5' -AATTG <b>GATATC</b> GGT-3' 3' -C <b>CTATAG</b> -5'	Początkowa sekwencja wypełniacza (fragment rozpoznawalny przez enzymy <i>MfeI</i> oraz <i>Eco32I</i> ) – wspólna dla wszystkich cząstek.	14 7
5' -GCTT <b>GATATC</b> GCTGG -3' 3' -AGCCGAA <b>CTATAGC</b> -5'	Końcowa sekwencja wypełniacza (fragment rozpoznawalny przez enzymy <i>PspFI</i> oraz <i>Eco32I</i> ) – wspólna dla wszystkich cząstek.	15 14
5' -GCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTG-3' 3' -CCACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGC-5'	Pierwsza sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -CTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGC-3' 3' -CACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCG-5'	Druga sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -TTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCT-3' 3' -ACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGA-5'	Trzecia sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -TATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTT-3' 3' -CGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGAC-5'	Czwarta sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -ATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTA-3' 3' -GAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGACG-5'	Piąta sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41
5' -TCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTAT-3' 3' -AATAGCCACGAATAGCCACGAATAGCCACGAATAGCGACGA-5'	Szósta sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	41 41

5' -CGGTGCTTATCGGTGCTTATCGGTGCTTATCG-3' 3' -ATAGCCACGAATAGCCACGAATAGCCACGAAT-5'	Siódma sekwencja 41 nukleotydów uzupełniających niezdefiniowane fragmenty (typu (10), (47), (...)).	32 32
5' -GCTCCGGCCG GCAGC GCTG ATTCT GAGACTCG-3' 3' -CACCGAGGCCGGC CGTCGCGAC TAAGACTCTG-5'	Sekwencja reprezentująca klauzulę faktu ( <i>a</i> )	32 32
5' -GCTCCGGCCG GCAGC GCTG ACCCT GAGACTCG-3' 3' -CACCGAGGCCGGC CGTCGCGAC TGGGACTCTG-5'	Sekwencja reprezentująca klauzulę faktu ( $\neg d$ )	32 32
5' -GCTCCGGCCG GCAGC GCTG GATA GAGACTCG-3' 3' -CACCGAGGCCGGC CGTCGCGAC CTATACTCTG-5'	Sekwencja reprezentująca klauzulę faktu ( $\neg e$ )	32 32
5' -GCTT GTCTCAATTC CAGCGCTG ACTTAAGT-3' 3' -AATCGAA CAGAGT TAACGTCGCGAC TGAATTCAGTCG-5'	Pierwsza sekwencja reprezentująca klauzulę złożoną ( $\neg a \vee \neg b \vee c \vee d$ )	30 37
5' -CAGCGCTG AACATGTT CAGCGCTG GGGTT GAGACTTA-3' 3' -CGAC TTGTACAAGTCGCGAC CCCAACTCTG-5'	Druga sekwencja reprezentująca klauzulę złożoną ( $\neg a \vee \neg b \vee c \vee d$ )	37 30
5' -GCTT GTCTCAACTT CAGCGCTG TATCT GAGACTCG-3' 3' -CCACGAA CAGAGT TGAAGTCGCGAC ATAGACTCTG-5'	Sekwencje reprezentujące klauzulę złożoną ( $b \vee e$ )	35 35
5' -GCTT GTCTCAGAAAATTTTCG-3' 3' -AATCGAA CAGAGT CTTTTTAA-5'	Sekwencje reprezentujące dane zapytanie ( $\neg c?$ )	21 21
5' -GCTCCGGCCG GCAGCT GAGACTCG-3' 3' -CACCGAGGCCGGC CGTCGACTCTG-5'	Sekwencja reprezentująca cząsteczkę terminalną	24 24

Tab. 4: sekwencje oligonukleotydów zamawiane w firmie komercyjnej dla trzeciego eksperymentu

### c) łączenie otrzymanych oligonukleotydów w cząsteczki dwuniciowe

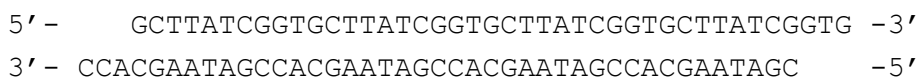
Otrzymane syntetyczne oligonukleotydy są pozbawione grupy fosforanowej na końcu 5'. Należy poddać je fosforylacji kinazą polinukleotydową. Jej rolą jest transfer fosforanu z ATP (*adenozynotrójfosforan*, z ang. *adenosine triphosphate*) na koniec 5' cząsteczki DNA, która nie posiada tego fragmentu (dlatego że powstała w wyniku syntezy) [40], [42]. W dalszej kolejności pary dopasowanych łańcuchów są mieszane i łączone we właściwe sekwencje dwuniciowe przy użyciu enzymu ligazy.

Przyjrzyjmy się poszczególnym elementom dla pierwszego eksperymentu.

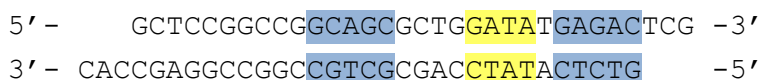
Zamówione fragmenty, niezbędne do uzyskania  $\neg q$ , to:



Ryc. 165a: początkowa sekwencja wypełniacza



Ryc. 165b: pierwsza sekwencja uzupełniająca niezdefiniowane fragmenty



Ryc. 165c: sekwencja reprezentująca klauzulę faktu  $\neg q$



Ryc. 165d: końcowa sekwencja wypełniacza

W wyniku ich ligacji otrzymamy cząsteczkę reprezentującą  $\neg q$ :

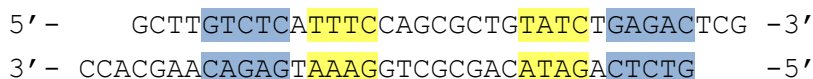


Ryc. 166: cząsteczka reprezentująca klauzulę faktu  $\neg q$

Zamówione fragmenty, niezbędne do uzyskania  $(\neg p \vee q)$ , to:



Ryc. 167a: początkowa sekwencja wypełniacza

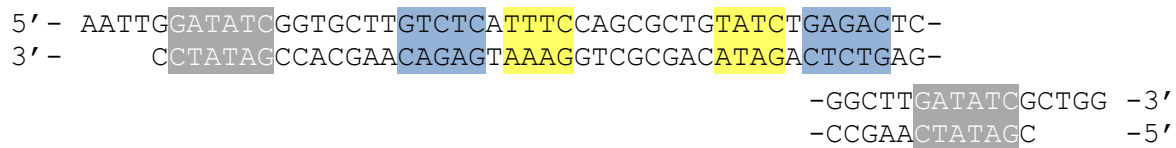


Ryc. 167b: sekwencja reprezentująca klauzulę faktu  $(\neg p \vee q)$



Ryc. 167c: końcowa sekwencja wypełniacza

W wyniku ich ligacji otrzymamy cząsteczkę reprezentującą ( $\neg p \vee q$ ):

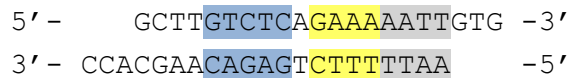


Ryc. 168: cząsteczka reprezentująca klauzulę złożoną ( $\neg p \vee q$ )

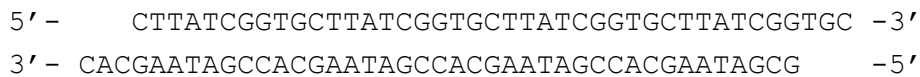
Zamówione fragmenty, niezbędne do uzyskania  $p$ ?, to:



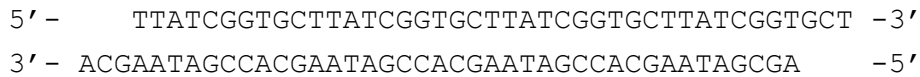
Ryc. 169a: początkowa sekwencja wypełniacza



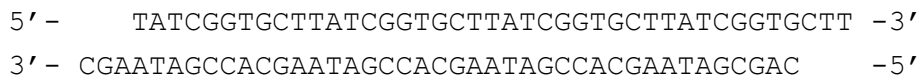
Ryc. 169b: sekwencja reprezentująca zapytanie  $p$ ?



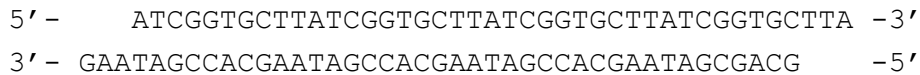
Ryc. 169c: druga sekwencja uzupełniająca niezdefiniowane fragmenty



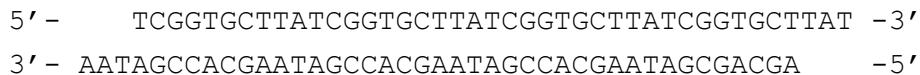
Ryc. 169d: trzecia sekwencja uzupełniająca niezdefiniowane fragmenty



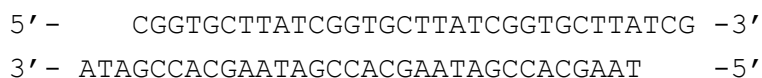
Ryc. 169e: czwarta sekwencja uzupełniająca niezdefiniowane fragmenty



Ryc. 169f: piąta sekwencja uzupełniająca niezdefiniowane fragmenty



Ryc. 169g: szósta sekwencja uzupełniająca niezdefiniowane fragmenty



Ryc. 169h: siódma sekwencja uzupełniająca niezdefiniowane fragmenty



Ryc. 169i: końcowa sekwencja wypełniacza

W wyniku ich ligacji otrzymamy cząsteczkę reprezentującą *p*?:

```

5' - AATTG GATATC GGTGCTT GTCTCA GAAAAATTGTGCTTATCGGTGCTTATCGGTGCTTATCG-
3' -      CCTATAG CCACGAACAGAGT CTTT TTAACACGAATAGCCACGAATAGCCACGAATAGC-
-GTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCT-
-CACGAATAGCGACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGACGAATAGCCACGA-
-TATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCG-
-ATAGCCACGAATAGCCACGAATAGCGACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGC-
-GTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCT-
-GACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCGACGAATAGCCACGAATAGCCACGA-
                                     -TATCGGTGCTTATCGGCTT GATATC GCTGG -3'
                                     -ATAGCCACGAATAGCCGAA CTATAGC -5'

```

Ryc. 170: cząsteczka reprezentująca zapytanie *p*?

Zamówione fragmenty, niezbędne do uzyskania cząsteczki terminalnej, to:

```

5' - AATTG GATATC GGT -3'
3' -      CCTATAG -5'

```

Ryc. 171a: początkowa sekwencja wypełniacza

```

5' -      GCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTG -3'
3' - CCACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGC -5'

```

Ryc. 171b: pierwsza sekwencja uzupełniająca niezdefiniowane fragmenty

```

5' -      GCTCCGGCCG GCAGCT GAGACTCG -3'
3' - CACCGAGGCCGGC CGTCGACTCTG -5'

```

Ryc. 171c: sekwencja reprezentująca cząsteczkę terminalną

```

5' -      GCTT GATATC GCTGG -3'
3' - AGCCGAA CTATAGC -5'

```

Ryc. 171d: końcowa sekwencja wypełniacza

W wyniku ich ligacji otrzymamy cząsteczkę terminalną:

```

5' - AATTG GATATC GGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGCTTATCGGTGGCTCCG-
3' -      CCTATAG CCACGAATAGCCACGAATAGCCACGAATAGCCACGAATAGCCACCGAGGC-
                                     -GCCGGCAGCTGAGACTCGGCTT GATATC GCTGG -3'
                                     -CGGCCGTCGACTCTGAGCCGAA CTATAGC -5'

```

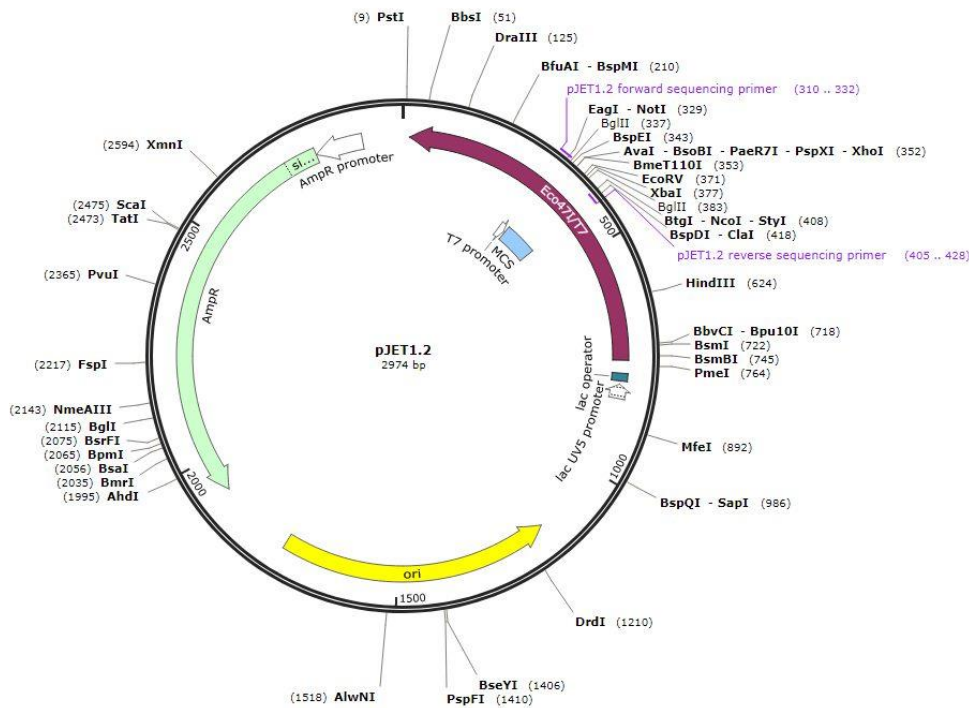
Ryc. 172: cząsteczka terminalna wnioskowania

W analogiczny sposób uzyskane zostaną fragmenty DNA reprezentujące pozostałe składowe dla pozostałych przedstawionych (lub innych) eksperymentów. Tak przygotowane łańcuchy DNA są gotowe do wklonowania w wektor *pJET 1.2* i dalszego ich powielania w komórkach bakteryjnych.

## 6. 2. 2. Klonowanie elementów systemu w wektorze *pJET 1.2*.

Do doświadczeń wybrany został łatwo dostępny wektor plazmidowy *pJET 1.2*.

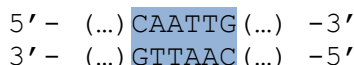
Wektor ten zawiera sekwencje rozpoznawane przez różne enzymy restrykcyjne. Jego mapa restrykcyjna została przedstawiona na rycinie 173.



Ryc. 173: Mapa restrykcyjna wektora plazmidowego pJET 1.2 z zaznaczonymi elementami genetycznymi tego wektora (źródło, dostępne w latach 2016-2021: [http://www.snapgene.com/resources/plasmid\\_files/basic\\_cloning\\_vectors/pJET1.2](http://www.snapgene.com/resources/plasmid_files/basic_cloning_vectors/pJET1.2))

Aby wprowadzić dodatkowy fragment DNA do wektora, należy za pomocą wybranych enzymów wyciąć w nim fragment, który pozostawi „lepkie” lub „tępe końce” odpowiednio dopasowane do wprowadzanego DNA. Aby umieszczenie cząsteczek zawsze przebiegało w identyczny i przewidywalny sposób, w omawianym eksperymencie wybrane zostały „lepkie końce” pozostawiane przez enzymy *MfeI* oraz *PspFI*. Oba enzymy rozpoznają 6-cio nukleotydową sekwencję, a następnie rozcinają cząsteczkę wewnątrz niej, pozostawiając „lepkie końce” o długości 4 nukleotydów:

Schemat trawienia DNA przez enzym *MfeI* wygląda następująco:

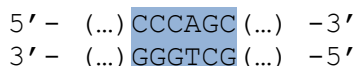


Ryc. 174a: sekwencja nukleotydów rozpoznawalna przez enzym *MfeI*



Ryc.174b: efekt trawienia powyższej sekwencji przez enzym *MfeI*

Schemat trawienia DNA przez enzym *PspFI* wygląda następująco:

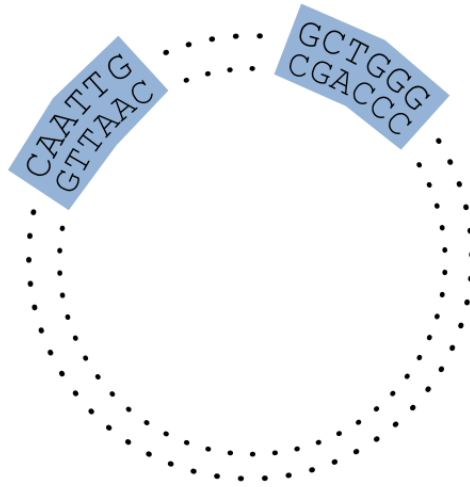


Ryc. 175a: sekwencja nukleotydów rozpoznawalna przez enzym *PspFI*



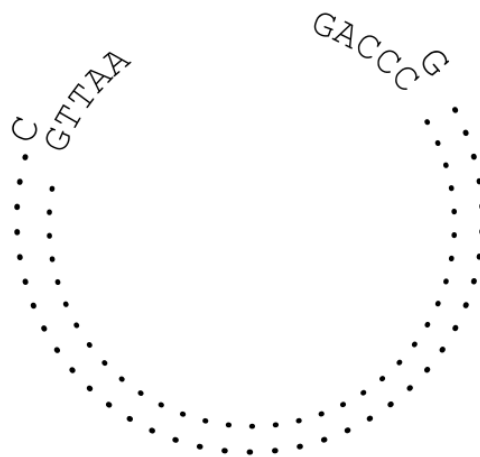
Ryc.175b: efekt trawienia powyższej sekwencji przez enzym *PspFI*

Biorąc pod uwagę wyłącznie sekwencje rozpoznawalne przez powyższe dwa enzymy, mocno uproszczony schemat wektora plazmidowego *pJET 1.2*, uwzględniający ich sekwencje rozpoznawalne, możemy przedstawić następująco:



*Ryc. 176: uproszczony schemat wektora plazmidowego pJET 1.2, uwzględniający sekwencje rozpoznawalne przez MfeI oraz PspFI*

W wyniku trawienia z udziałem enzymów restrykcyjnych MfeI oraz PspFI otrzymamy fragment wektora, zawierający „lepkie końce” dopasowane do wklonowania przygotowanych wcześniej cząsteczek DNA (*patrz*: podrozdział 6.2.1).

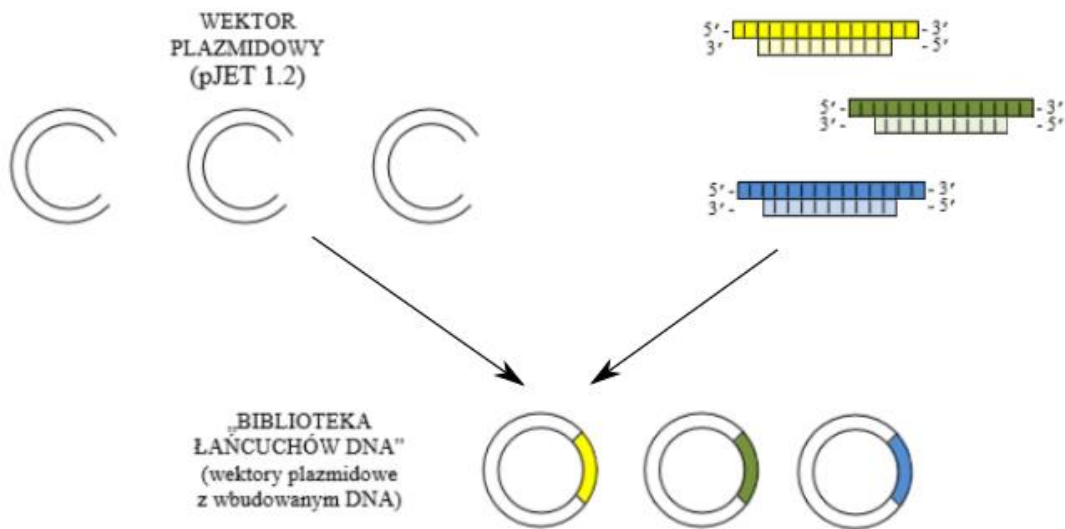


*Ryc. 177: uproszczony schemat pozostałego fragmentu wektora plazmidowego pJET 1.2*

Tak przygotowany wektor jest gotowy do połączenia z jedną dwuniciową cząsteczką DNA, reprezentującą składową systemu (np. sekwencją faktu  $-q$  przedstawioną na rycinie 166). Klonowanie przeprowadzane jest osobno dla każdej z cząsteczek składowych systemu wnioskującego.



Wklonowanie przykładowych trzech sekwencji wygląda następująco:



Ryc. 178: schemat tworzenia „biblioteki” łańcuchów DNA za pomocą wektorów plazmidowych

Wektory zawierające dodatkowe sekwencje nukleotydów są wprowadzone do komórek bakteryjnych (*Escherichia coli*) metodą transformacji. Operacja ta jest trwała i pozwala na dziedziczenie wprowadzonego elementu genetycznego również przez komórki potomne. Dzięki temu dalsze powielanie wprowadzonych łańcuchów nastąpi wraz z powielaniem komórek bakteryjnych.

### 6. 2. 3. Selekcja przechowywanych sekwencji z „biblioteki” łańcuchów DNA

Do wyselekcjonowania niezbędnych fragmentów z „biblioteki” łańcuchów DNA (czyli z poszczególnych wektorów plazmidowych *pJET 1.2*), a także odpowiedniego powielenia ich przed właściwą reakcją, zastosujemy reakcję łańcuchową polimerazy. Do zidentyfikowania i powielenia jednej sekwencji potrzebne będą dwa „startery” (oligonukleotydy o długości 15-25 nukleotydów, odpowiadające komplementarnie obu końcom 3’ selekcjonowanej sekwencji). Działanie PCR zostało szczegółowo przedstawione wcześniej (*patrz*: podrozdział 2.1.3).

Przyjrzyjmy się jeszcze dokładniej fragmentowi wektora plazmidowego *pJET 1.2* przed wklonowaniem do niego sekwencji reprezentujących składowe systemy wnioskujące:

```

5' - (...) GTT GCGCTCACTGCCAATTG (508) GCTGGGCTGTGTGCACGAAC (...) -3'
3' - (...) CAACGCGAGTGACGGTTAAC (508) CGACCCGACACACGTGCTTG (...) -5'
    
```

Ryc.179: fragment wektora plazmidowego *pJET 1.2*, uwzględniający sekwencje rozpoznawalne przez *MfeI* oraz *PspFI* (źródło, dostępne w latach 2016-2021: [http://www.snapgene.com/resources/plasmid\\_files/basic\\_cloning\\_vectors/pJET1.2/](http://www.snapgene.com/resources/plasmid_files/basic_cloning_vectors/pJET1.2/))

Fragment znajdujący się pomiędzy sekwencjami rozpoznawalnymi przez enzymy restrykcyjne *MfeI* oraz *PspFI* (oznaczony tutaj jako (508)) został wycięty i zastąpiony jednym z fragmentów reprezentujących pojedynczą składową systemy wnioskujące (*patrz*: podrozdział 6.2.2).

Interesujący nas fragment, znajdujący się w wektorze plazmidowym, wygląda zawsze zgodnie z następującym schematem (gdzie na szaro zostało zaznaczone miejsce, gdzie rozpoczyna i kończy się ciąg, który należy wyselekcjonować z wektora):



Ryc.180: fragment wektora plazmidowego pJET 1.2, zawierający schemat dowolnej sekwencji reprezentującej element systemu

Jako oligonukleotydy reprezentujące „startery” PCR dopasowujemy zatem stałe sekwencje o długości 20-tu nukleotydów, znajdujące się poza fragmentem, reprezentującym składową systemu. Będą to odpowiednio:



Ryc. 181a: sekwencja reprezentująca STARTER (komplementarny do końca 3' pierwszej nici oczekiwanego łańcucha)



Ryc. 181b: sekwencja reprezentująca STARTER (komplementarny do końca 3' drugiej nici oczekiwanego łańcucha)

Oligonukleotydy reprezentujące „startery” również zamawiamy z firmy komercyjnej.

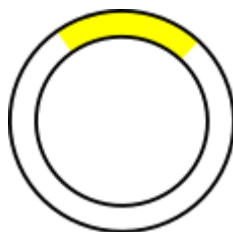
Po rozpleceniu podwójnej helisy, wspomniane startery przyłączą się następująco:



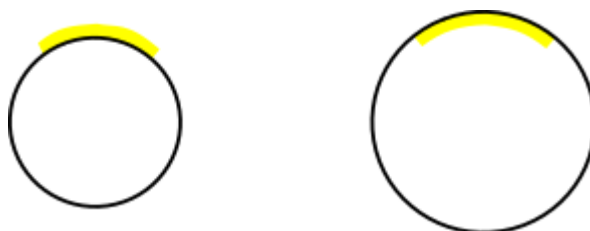
Ryc.182: schemat pierwszego kroku PCR dla dowolnej sekwencji reprezentującej element systemu i omówionych „starterów”

Na poniższych ilustracjach przedstawiony zostanie cały schemat selekcji pojedynczej sekwencji za pomocą PCR z przykładowego wektora plazmidowego pJET 1.2. Kolorem żółtym zaznaczono przeznaczony do wycięcia łańcuch reprezentujący składową systemu, (uwzględniając sekwencje komplementarne do zamówionych „starterów”). Kolorem zielonym zaznaczono przedstawione „startery” o długości 20-tu nukleotydów.

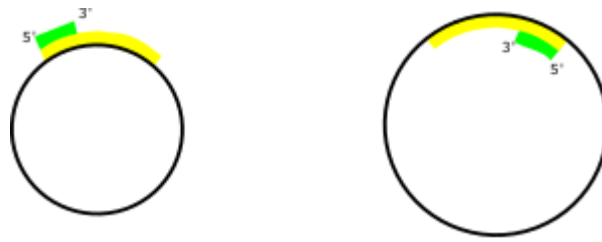
Selekcja pojedynczej cząsteczki za pomocą reakcji PCR wygląda następująco:



Ryc. 183a: wektor plazmidowy z wbudowanym DNA reprezentującym element systemu



Ryc. 183b: rozplecenie helisy do postaci dwóch oligonukleotydów (denaturacja)

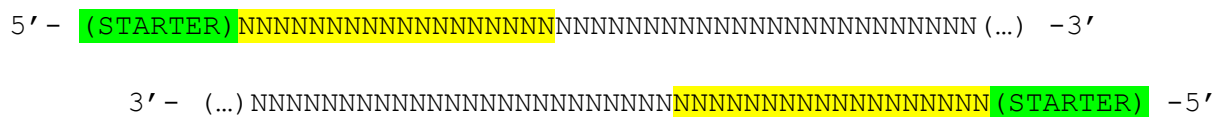


Ryc. 183c: przyłączenie się krótkich fragmentów komplementarnych („starterów”)



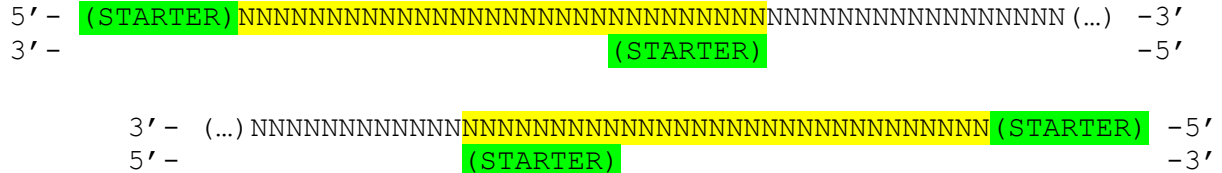
Ryc. 183d: dobudowanie komplementarnych nukleotydów w kierunku od 5' do 3'

Warto zwrócić uwagę, że podczas tej reakcji dobudowywane fragmenty nie połączą się samoistnie w dwuniciową cząsteczkę kolistą, co oznacza, że po ponownym rozpleceniu helisy do postaci jednoniciowych, tylko dwie z czterech cząsteczek będą wciąż w postaci kolistej. W dalszej reakcji przyjrzymy się dwóm pozostałym (jednoniciowym) fragmentom:

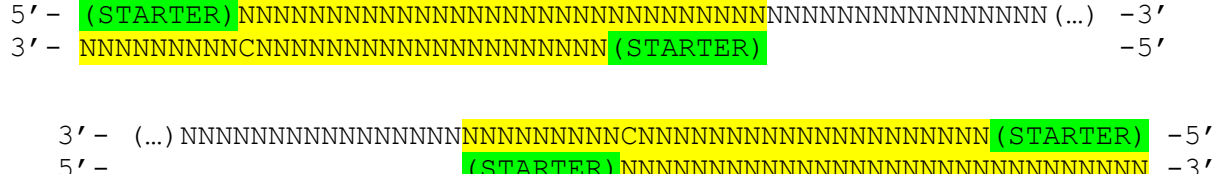


Ryc. 184: fragmenty jednoniciowe po pierwszej fazie reakcji łańcuchowej polimerazy (PCR)

Na przedstawionej „górnjej” nici żółtym kolorem oznaczony został fragment reprezentujący składową systemu wnioskującego, a zielonym odpowiednio komplementarny „starter”, który brał już udział w reakcji dobudowywania. Z drugiej jej strony (na przedstawionej „dolnej” nici) przyłączy się kolejny „starter”:

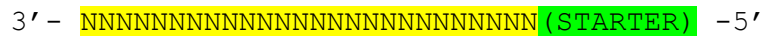


Ryc. 185a: przyłączenie się kolejnych krótkich fragmentów komplementarnych („starterów”)

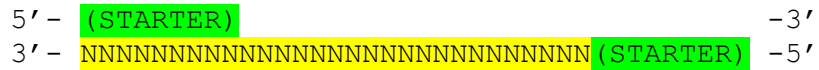
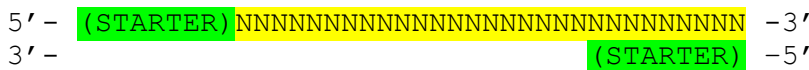


Ryc. 185b: dobudowanie komplementarnych nukleotydów w kierunku od 5' do 3'

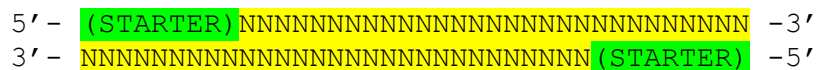
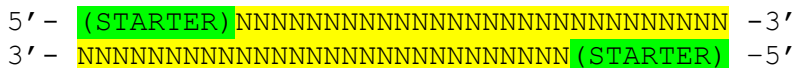
Po ponownym rozpleceniu helisy do postaci jednoniciowych, tylko dwie z czterech cząsteczek będą posiadały odpowiednią długość. To one będą namnażały się (wykładniczo) w dalszych cyklach PCR. Dwa dłuższe fragmenty, jak i dwa o postaci kolistej, też będą brać udział w cyklach PCR, również generując nowopowstałe nici o odpowiedniej długości.



Ryc. 186a: fragmenty jednoniciowe po drugiej fazie reakcji łańcuchowej polimerazy (PCR)



Ryc. 186b: przyłączenie się kolejnych krótkich fragmentów komplementarnych („starterów”)



Ryc. 186c: dobudowanie komplementarnych nukleotydów w kierunku od 5' do 3'

Za pomocą tych trzech cykli reakcji łańcuchowej polimerazy, możemy pobrać (wyselekcjonować) żądany fragment z „biblioteki” łańcuchów DNA. W wyniku przeprowadzania kolejnych cykli zostanie on namnożony w sposób wykładniczy.

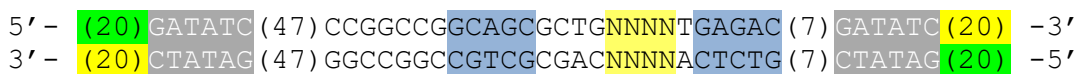
#### 6. 2. 4. Selekcja fragmentów cząsteczek biorących udział w reakcji

Aby przygotować właściwe „tępe” lub „lepkie końce” we wszystkich cząsteczkach, reprezentujących składowe systemu wnioskującego, użyjemy kolejnych dwóch enzymów, których sekwencje rozpoznawalne znajdują się wewnątrz otrzymanych łańcuchów (*patrz*: podrozdział 6.2.2). Do przygotowania „tępych końców” użyty zostanie enzym restrykcyjny *Eco32I*, a do przygotowania „lepkich końców” enzym restrykcyjny *BsmAI*. Wszystkie fragmenty wyselekcjonowane z „biblioteki” łańcuchów DNA za pomocą PCR, w sposób przedstawiony w podrozdziale 6.2.3, wyglądają zgodnie z następującym schematem:

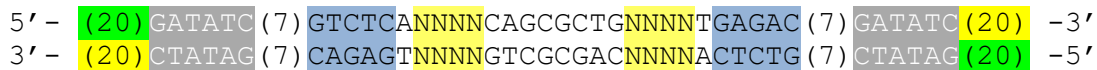


Ryc. 187: schemat wszystkich cząsteczek reprezentujących składowe systemu wnioskującego, po selekcji z wektora plazmidowego

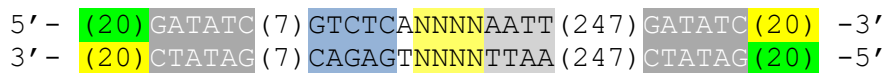
Szczegółowo, dla różnych składowych systemu, są to następujące fragmenty:



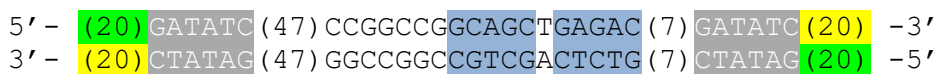
Ryc. 188a: schemat cząsteczki reprezentującej klauzulę faktu po selekcji z wektora plazmidowego



Ryc. 188b: schemat cząsteczki reprezentującej klauzulę złożoną po selekcji z wektora plazmidowego

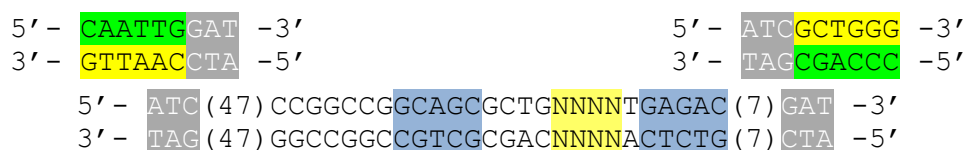


Ryc. 188c: schemat cząsteczki reprezentującej zapytanie po selekcji z wektora plazmidowego

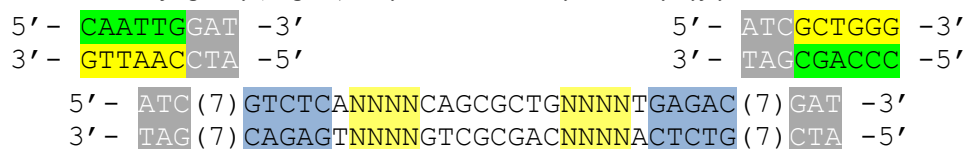


Ryc. 188d: cząsteczka terminalna wnioskowania po selekcji z wektora plazmidowego

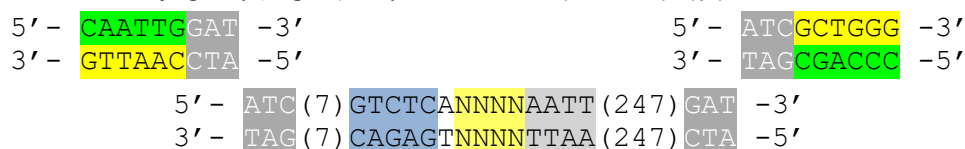
W wyniku trawienia enzymem restrykcyjnym *Eco32I*, otrzymamy:



Ryc. 189a: schemat cząsteczki reprezentującej klauzulę faktu (na dole) oraz dodatkowe fragmenty (na górze), w wyniku trawienia enzymem restrykcyjnym *Eco32I*



Ryc. 189b: schemat cząsteczki reprezentującej klauzulę złożoną (na dole) oraz dodatkowe fragmenty (na górze), w wyniku trawienia enzymem restrykcyjnym *Eco32I*

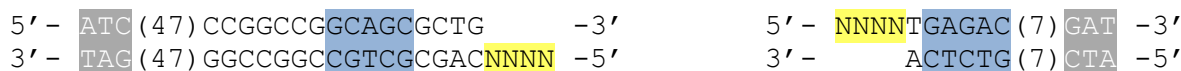


Ryc. 189c: schemat cząsteczki reprezentującej zapytanie (na dole) oraz dodatkowe fragmenty (na górze), w wyniku trawienia enzymem restrykcyjnym *Eco32I*

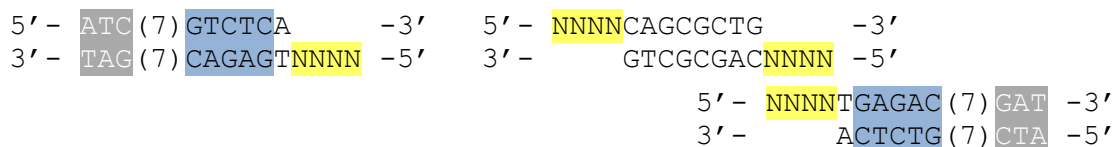


Ryc. 189d: cząsteczka terminalna (na dole) oraz dodatkowe fragmenty (na górze), w wyniku trawienia enzymem restrykcyjnym *Eco32I*

W wyniku trawienia enzymem restrykcyjnym *BsmAI*, otrzymamy:



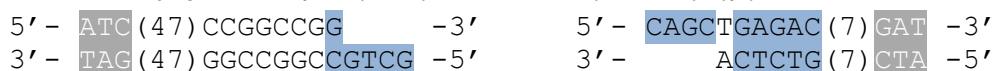
Ryc. 190a: schemat cząsteczki reprezentującej klauzulę faktu (z lewej strony) oraz dodatkowy fragment (z prawej strony), w wyniku trawienia enzymem restrykcyjnym *BsmAI*



Ryc. 190b: schemat cząsteczki reprezentującej klauzulę złożoną (u góry z prawej strony) oraz dodatkowe fragmenty (u góry z lewej strony oraz na dole z prawej strony), w wyniku trawienia enzymem restrykcyjnym *BsmAI*



Ryc. 190c: schemat cząsteczki reprezentującej zapytanie (z prawej strony) oraz dodatkowy fragment (z lewej strony), w wyniku trawienia enzymem restrykcyjnym *BsmAI*



Ryc. 190d: cząsteczka terminalna (z lewej strony) oraz dodatkowy fragment (z prawej strony), w wyniku trawienia enzymem restrykcyjnym *BsmAI*

W taki sposób przygotowane cząsteczki DNA są gotowe do wzięcia udziału we właściwej reakcji wnioskowania z udziałem enzymu restrykcyjnego *BbvI* oraz enzymu ligazy.

### 6. 3. Reakcja wnioskowania w laboratorium

Po przygotowaniu właściwych fragmentów DNA (według powyższych punktów) do wybranej reakcji wnioskowania, mieszamy je ze sobą i dodajemy enzymy *BbvI* i ligazę. Nastąpi seria naprzemiennego cięcia oraz łączenia fragmentów DNA. W jej efekcie mogą powstać cząsteczki o odpowiednich długościach, symbolizujących odpowiedzi twierdzące na któreś z zapytań (jedno lub więcej) lub informacja o wystąpieniu sprzeczności. Rozstrzygamy to za pomocą reakcji elektroforezy w żelu (*patrz*: podrozdział 2.1.6). W jej wyniku otrzymamy jednoznaczną odpowiedź:

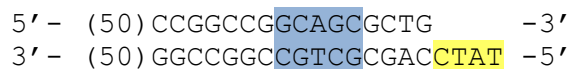
1. ewentualne wystąpienie sprzeczności, co dyskwalifikuje pozostałe odpowiedzi,
2. twierdzącą odpowiedź na niektóre i brak odpowiedzi na pozostałe z zapytań (przy braku sprzeczności).

Przebieg wielu przykładowych reakcji został już szczegółowo przedstawiony w teoretycznej części tej pracy, w rozdziałach 4 oraz 5. Poniżej zostanie przedstawiony przebieg właściwej reakcji dla zaproponowanych w podrozdziale 6.1 eksperymentów. Cząsteczki DNA biorące udział w reakcji zostały przygotowane zgodnie z procedurą omówioną w podrozdziale 6.2.

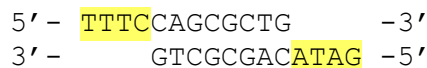
#### 6. 3. 1. Wnioskowanie i reguła kontrapozycji

Propozycja eksperymentu została przedstawiona w podrozdziale 6.1.1.

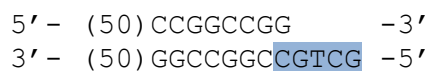
Sekwencje cząsteczek DNA wyglądają następująco:



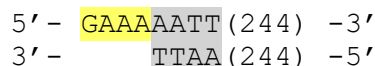
*Ryc. 191a: cząsteczka reprezentująca klauzulę faktu  $\neg q$*



*Ryc. 191b: cząsteczka reprezentująca klauzulę  $(\neg p \vee q)$*



*Ryc. 191c: cząsteczka terminalna wnioskowania*

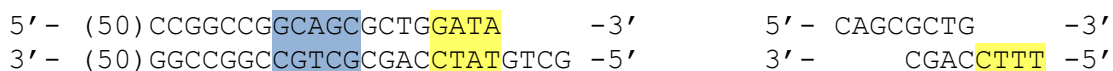


*Ryc. 191d: cząsteczka reprezentująca zapytanie  $\neg p$ ?*

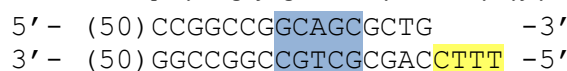
Reakcja będzie wyglądać następująco:



*Ryc. 192a: ligacja cząsteczek reprezentujących klauzule  $\neg q$  oraz  $(\neg p \vee q)$*



*Ryc. 192b: trawienie powyższego fragmentu enzymem restrykcyjnym *BbvI**



*Ryc. 192c: ligacja cząsteczki terminalnej i wniosku częściowego; powstała cząsteczka reprezentuje  $\neg p$*

5' - (50) CCGGCCG **GCAGC** GCTG **GAAA** AATT (244) -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **CTTT** TTAA (244) -5'

*Ryc. 192d: ligacja cząsteczki reprezentującej  $\neg p$  i cząsteczki zapytania  $\neg p$ ?*

5' - (50) CCGGCCG **GCAGC** GCTG **GAAA** -3'                      5' - **AATT** (244) -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **CTTT** TTAA -5'                      3' - (244) -5'

*Ryc. 192e: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI*

5' - (244) **AATT** (244) -3'  
 3' - (244) **TTAA** (244) -5'

*Ryc. 192f: ligacja nowopowstałych identycznych cząsteczek*

W wyniku reakcji otrzymamy cząsteczkę bez „lepkich końców”, o długości 504 nukleotydów, która oznacza odpowiedź twierdzącą na zadane zapytanie  $\neg p$ ? a zatem informuje, że zdanie  $\neg p$  wynika z aksjomatów wejściowych.

### 6. 3. 2. Wykrycie sprzeczności

Propozycja eksperymentu została przedstawiona w podrozdziale 6.1.2.

Sekwencje cząsteczek DNA wyglądają następująco:

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **AAAG** -5'

*Ryc. 193a: cząsteczka reprezentująca klauzulę faktu p*

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **CTAT** -5'

*Ryc. 193b: cząsteczka reprezentująca klauzulę faktu  $\neg q$*

5' - **TTTC** CAGCGCTG -3'  
 3' - GTCGCGAC **ATAG** -5'

*Ryc. 193c: cząsteczka reprezentująca klauzulę  $(\neg p \vee q)$*

5' - (50) CCGGCCG -3'  
 3' - (50) GGCCGGC **CGTCG** -5'

*Ryc. 193d: cząsteczka terminalna wnioskowania*

Reakcja będzie wyglądać następująco:

5' - (50) CCGGCCG **GCAGC** GCTG **GATA** CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **CTAT** GTCGCGAC **CTTT** -5'

*Ryc. 194a: ligacja cząsteczek reprezentujących klauzulę  $\neg q$  oraz  $(\neg p \vee q)$*

5' - (50) CCGGCCG **GCAGC** GCTG **GATA** -3'                      5' - CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **CTAT** GTCG -5'                      3' - CGAC **CTTT** -5'

*Ryc. 194b: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI*

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **CTTT** -5'

*Ryc. 194c: ligacja cząsteczki terminalnej i wniosku częściowego; powstała cząsteczka reprezentuje  $\neg p$*

5' - (50) CCGGCCG **GCAGC** GCTG **TTTC** CAGCGCTG **CGGCCG** (50) -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **AAAG** GTCGCGAC **GCCGGCC** (50) -5'

*Ryc. 194d: ligacja cząsteczek reprezentujących klauzulę p oraz  $\neg p$*

5' - (50) CCGGCCG **GCAGC** GCTG **TTTC** -3'                      5' - CAGC **GCTGC** CGGCCG (50) -3'  
 3' - (50) GGCCGGC **CGTCG** CGAC **AAAG** GTCG -5'                      3' - **CGACG** GCCGGCC (50) -5'

*Ryc. 194e: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI*

5' - (50) CCGGCCG **GCAGCGCTGC** CCGGCCG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGACG** GCCGGCC (50) -5'

*Ryc. 194f: ligacja cząsteczki terminalnej i nowopowstałej*

5' - (50) CCGGCCG **GCAGCGCTGC** CCGG -3'                      5' - CCGG (50) -3'  
 3' - (50) GGCCGGC **CGTCGCGACG** GCCGGCC -5'                      3' - (50) -5'

*Ryc. 194g: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI*

5' - (50) CCGG (50) -3'  
 3' - (50) GGCC (50) -5'

*Ryc. 194h: ligacja nowopowstałych identycznych cząsteczek*

Powstała cząsteczka niezawierająca „lepkich końców” oraz o unikalnej długości 104 nukleotydów oznacza wystąpienie sprzeczności w systemie. Zależnie od podejścia, możemy potraktować ją jako pozytywną odpowiedź na wszystkie zadane pytania (ponieważ ze sprzeczności możemy wyprowadzić dowolny wniosek), bądź jako informację o niepoprawności pozostałych odpowiedzi systemu.

### 6. 3. 3. Złożone wnioskowanie i prawa de Morgana

Propozycja eksperymentu została przedstawiona w podrozdziale 6.1.3.

Sekwencje cząsteczek DNA wyglądają następująco:

5' - (50) CCGGCCG **GCAGCGCTG** -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** **TAAG** -5'

*Ryc. 195a: cząsteczka reprezentująca klauzulę faktu a*

5' - (50) CCGGCCG **GCAGCGCTG** -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** **TGGG** -5'

*Ryc. 195b: cząsteczka reprezentująca klauzulę faktu -d*

5' - (50) CCGGCCG **GCAGCGCTG** -3'  
 3' - (50) GGCCGGC **CGTCGCGAC** **CTAT** -5'

*Ryc. 195c: cząsteczka reprezentująca klauzulę faktu -e*

5' - **ATTCCAGCGCTGACTTAAGTCAGCGCTGAAACATGTT** CAGCGCTG -3'  
 3' - GTCGCGAC **TGAATTCA** GTCGCGAC **TTGTACAA** GTCGCGAC **CCCA** -5'

*Ryc. 195d: cząsteczka reprezentująca klauzulę (-a ∨ -b ∨ c ∨ d)*

5' - **ACTT** CAGCGCTG -3'  
 3' - GTCGCGAC **ATAG** -5'

*Ryc. 195e: cząsteczka reprezentująca klauzulę (b ∨ e)*

5' - (50) CCGGCCG -3'  
 3' - (50) GGCCGGC **CGTCG** -5'

*Ryc. 195f: cząsteczka terminalna wnioskowania*

5' - **AACAAATT** (244) -3'  
 3' - **TTAA** (244) -5'

*Ryc. 195g: cząsteczka reprezentująca zapytanie -c?*

Reakcja będzie wyglądać następująco:

5' - (50) CCGGCCG **GCAGCGCTGATTCCAGCGCTGACTTAAGT** (...) **AACATGTT** (...) -3'  
 3' - (50) GGCCGGC **CGTCGCGACTAAG** GTCGCGAC **TGAATTCA** (...) **TTGTACAA** (...) **CCCA** -5'

*Ryc. 196a: ligacja cząsteczek reprezentujących klauzule a oraz (-a ∨ -b ∨ c ∨ d)*



5' - (50) CCGGCCG **GCAGC** GCTGATTC -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTAAGGTCG -5'

5' - CAGCGCTGACTTAAGTCAGCGCTGAACATGTT CAGCGCTG -3'  
 3' - CGACTGAATTCAGTCGCGACTTGTACAAATCGCGAC **CCCA** -5'

**Ryc. 196b: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI**

5' - (50) CCGGCCG **GCAGC** GCTGACTTAAGTCAGCGCTGAACATGTT CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTGAATTCAGTCGCGACTTGTACAAATCGCGAC **CCCA** -5'

**Ryc. 196c: ligacja cząsteczki terminalnej i powyższego wniosku częściowego**

5' - (50) CCGGCCG **GCAGC** GCTGACTT -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTGAATTCA -5'

5' - AAGTCAGCGCTGAACATGTT CAGCGCTG -3'  
 3' - GTCGCGACTTGTACAAATCGCGAC **CCCA** -5'

**Ryc. 196d: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI; powstała cząsteczka reprezentuje (-b v c v d)**

5' - AAGTCAGCGCTGAACATGTT CAGCGCTGGGGTGCTGGCTGCCGGCCGG (50) -3'  
 3' - GTCGCGACTTGTACAAATCGCGAC **CCCA**CGAC **CGAC**GGCCGGCC (50) -5'

**Ryc. 196e: ligacja cząsteczek reprezentujących klauzule -d oraz (-b v c v d)**

5' - AAGTCAGCGCTGAACATGTT CAGC -3'  
 3' - GTCGCGACTTGTACAAATCGCGAC -5'

5' - GCTGGGGTCAGCGCTGCCGGCCGG (50) -3'  
 3' - CCCAGTCGCGACGGCCGGCC (50) -5'

**Ryc. 196f: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI**

5' - AAGTCAGCGCTGAACATGTT CAGCGCTGCCGGCCGG (50) -3'  
 3' - GTCGCGACTTGTACAAATCGCGACGGCCGGCC (50) -5'

**Ryc. 196g: ligacja cząsteczki terminalnej i powyższego wniosku częściowego**

5' - AAGTCAGCGCTG -3'      5' - AACATGTT CAGCGCTGCCGGCCGG (50) -3'  
 3' - GTCGCGACTTGT -5'      3' - ACAAGTCGCGACGGCCGGCC (50) -5'

**Ryc. 196h: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI; powstała cząsteczka reprezentuje (-b v c)**

5' - GATACAGCGCTGAAGTCAGCGCTG -3'  
 3' - GTCGCGACTTCAAGTCGCGACTTGT -5'

**Ryc. 196i: ligacja cząsteczek reprezentujących klauzulę (b v e) oraz (-b v c); powstała cząsteczka reprezentuje (c v e)**

5' - (50) CCGGCCG **GCAGC** GCTGGATACAGCGCTGAAGTCAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTATGTCGCGACTTCAAGTCGCGACTTGT -5'

**Ryc. 196j: ligacja cząsteczek reprezentujących -e oraz (c v e)**

5' - (50) CCGGCCG **GCAGC** GCTGGATA -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTATGTCG -5'

5' - CAGCGCTGAAGTCAGCGCTG -3'  
 3' - CGACTTCAAGTCGCGACTTGT -5'

**Ryc. 196k: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI**

5' - (50) CCGGCCG **GCAGC** GCTGAAGTCAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTTCAAGTCGCGACTTGT -5'

**Ryc. 196l: ligacja cząsteczki terminalnej i powyższego wniosku częściowego**

5' - (50) CCGGCCG **GCAGC** GCTGAAGT -3'      5' - CAGCGCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTTCAAGTCG -5'      3' - CGACTTGT -5'

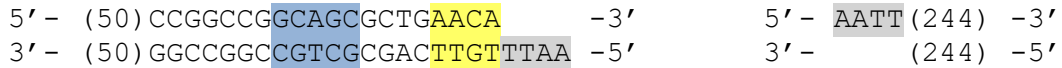
**Ryc. 196m: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI**

5' - (50) CCGGCCG **GCAGC** GCTG -3'  
 3' - (50) GGCCGGC **CGTCG** CGACTTGT -5'

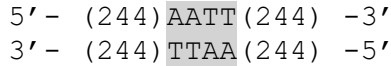
**Ryc. 196n: ligacja cząsteczki terminalnej i powyższego wniosku częściowego; powstała cząsteczka reprezentuje c**



Ryc. 196o: ligacja cząsteczki faktu *c* i cząsteczki zapytania *c*?



Ryc. 196p: trawienie powyższego fragmentu enzymem restrykcyjnym *BbvI*



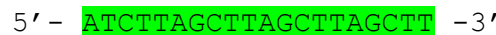
Ryc. 196q: ligacja nowopowstałych identycznych cząsteczek

W wyniku reakcji otrzymamy cząsteczkę bez „lepkich końców”, o długości 504 nukleotydów, która oznacza odpowiedź twierdzącą na zadane zapytanie *c*?, a zatem informuje, że zdanie *c* wynika z aksjomatów wejściowych.

W powyższej reakcji wykorzystano trawienie DNA z dwóch stron za pomocą enzymu *BbvI*, które ma na celu poprawną implementację reguły kontrapozycji oraz praw de Morgana. Pokazane zostało również, że w przypadku połączenia się ze sobą dwóch klauzul złożonych o komplementarnych „lepkich końcach”, wnioskowanie wciąż wykona się poprawnie, a reakcja ta odpowiada przechodności implikacji. W niektórych sytuacjach istnieje możliwość przebiegu poszczególnych kroków reakcji w innej kolejności, np. zamiast połączenia się ( $b \vee e$ ) oraz ( $\neg b \vee c$ ) może dojść najpierw do połączenia  $\neg e$  oraz ( $b \vee e$ ), co spowodowałoby w pierwszej kolejności powstanie wniosku  $\neg b$ , a następnie w wyniku reakcji z cząsteczką reprezentującą ( $\neg b \vee c$ ) powstanie wniosku *c*. Efekt końcowy reakcji wnioskowania byłby identyczny jak przedstawiono powyżej.

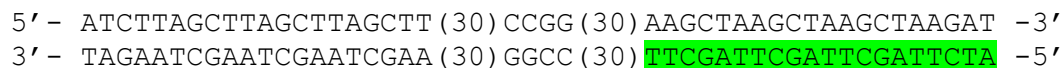
### 6. 3. 4. Wzmocnienie „sygnału” odpowiedzi systemu

Istnieje możliwość wzmocnienia „sygnału” odpowiedzi systemu, zarówno informującego o wystąpieniu sprzeczności jak i odpowiedzi twierdzącej na któreś z zapytań. W tym celu należy powielić fragmenty DNA reprezentujące uzyskane odpowiedzi systemu, za pomocą reakcji łańcuchowej polimerazy (*patrz*: podrozdział 2.1.3 oraz 6.2.3), po zakończonej reakcji splatania. Odpowiednio dobrany „starter” o długości 20-tu nukleotydów (inny niż przedstawione w podrozdziale 6.2.3), komplementarny z końcem 3’ dowolnego fragmentu DNA reprezentującego odpowiedź systemu (taka cząsteczka posiada dwa identyczne „tępe końce”) wygląda następująco:



Ryc. 197: starter biorący udział w reakcji replikacji cząsteczki reprezentującej odpowiedź systemu

W wyniku pojedynczej reakcji replikacji uzyskamy podwojenie się cząsteczek:



Ryc. 198: zakończona reakcja łańcuchowa polimerazy – uzyskujemy dwie cząsteczki reprezentujące informację o sprzeczności

5' - ATCTTAGCTTAGCTTAGCTT (230) AATT (230) AAGCTAAGCTAAGCTAAGAT -3'  
3' - TAGAATCGAATCGAATCGAA (230) TTAA (230) TTCGATTCGATTCGATTCTA -5'

5' - ATCTTAGCTTAGCTTAGCTT (230) AATT (230) AAGCTAAGCTAAGCTAAGAT -3'  
3' - TAGAATCGAATCGAATCGAA (230) TTAA (230) TTCGATTCGATTCGATTCTA -5'

Ryc. 199: zakończona reakcja łańcuchowa polimerazy – uzyskujemy dwie cząsteczki reprezentujące odpowiedź twierdzącą

W wyniku przeprowadzenia PCR w próbówce po zakończeniu właściwych obliczeń, namnożone zostaną wszystkie cząsteczki reprezentujące zarówno informację o wystąpieniu sprzeczności lub odpowiedzi twierdzących, przy użyciu tego samego, komplementarnego do wszystkich z nich, „startera”. Po zakończeniu serii replikacji, możemy być pewni, że „sygnał” informujący o odpowiedzi systemu będzie wystarczająco silny by można było odczytać go w laboratorium za pomocą reakcji elektroforezy w żelu.

### 6. 3. 5. Odczytanie wyniku

Odczytanie wyniku składa się z dwóch części: w pierwszej kolejności sprawdzenie czy pojawiła się cząsteczka sygnalizująca wystąpienie sprzeczności, a następnie w przypadku jej braku dalsze sprawdzenie istnienia cząsteczek, których długości reprezentują odpowiedzi twierdzące na dane zapytania. Do rozróżniania cząsteczek względem ich długości służy reakcja elektroforezy w żelu (*patrz*: podrozdział 2.1.6), odpowiadająca za „posortowanie” fragmentów DNA względem ich długości.

Cząsteczka o długości 104 nukleotydów oznacza wystąpienie sprzeczności. Pozostałe fragmenty DNA o długościach rzędu kilkuset oznaczają odpowiedzi twierdzące na konkretne zadane pytania, odpowiednio: fragment o długości 492 nukleotydów oznacza odpowiedź twierdzącą do zapytania, którego cząsteczka zawierała 244 nukleotydów, fragment o długości 604 nukleotydów oznacza odpowiedź twierdzącą do zapytania, którego cząsteczka zawierała 300 nukleotydów, itd. Odpowiednio, jeżeli cząsteczka zapytania składa się z  $n$  dowolnych nukleotydów, to cząsteczka odpowiedzi twierdzącej tego zapytania będzie składać się z  $(2*n + 4)$  nukleotydów. Niepojawienie się żadnej z nich oznacza brak jakiegokolwiek odpowiedzi twierdzącej, przy jednoczesnym niewystąpieniu sprzeczności.

Odpowiednia interpretacja wyników całkowicie kończy doświadczenie.

### 6. 4. Podsumowanie

W powyższym rozdziale zostały omówione wszystkie etapy przeprowadzenia właściwych eksperymentów laboratoryjnych: przygotowanie łańcuchów DNA przed zamówieniem w firmie komercyjnej, klonowanie otrzymanych łańcuchów w wektorze plazmidowym *pJET 1.2*, selekcja odpowiednich dwuniciowych fragmentów DNA przed rozpoczęciem właściwego eksperymentu, właściwa reakcja wnioskowania oraz odczytanie wyniku.

Na etapie przygotowania zostały wprowadzone dodatkowe enzymy restrykcyjne: *MfeI* i *PspFI* odpowiedzialne za odpowiednie nacięcie wektora plazmidowego a także enzymy *BsmAI* i *Eco32I* odpowiedzialne za wycięcie fragmentów DNA tuż przed główną reakcją.

Taki sposób przechowywania i powielania odpowiednich sekwencji nukleotydów w komórkach bakteryjnych umożliwia znaczne obniżenie kosztów (ograniczenie zamówienia z firmy komercyjnej do minimum), a także zachowanie kopii łańcuchów do kolejnych doświadczeń.

Całość eksperymentu sprowadza się do następujących kroków:

- synteza oligonukleotydów,
- połączenie jednoniciowych oligonukleotydów w cząsteczki dwuniciowe,
- wklonowanie dwuniciowych fragmentów DNA do wektora plazmidowego *pJET 1.2* i wprowadzenie ich do komórek bakteryjnych *Escherichia coli*,
- selekcja wybranych fragmentów DNA z wektorów plazmidowych *pJET 1.2* za pomocą reakcji łańcuchowej polimerazy (PCR) i odpowiednio dobranych „starterów”,
- trawienie pobranych fragmentów DNA enzymami *Eco32I* i *BsmAI*, które są odpowiedzialne za uzyskanie „tępych” oraz „lepkich końców” biorących udział w reakcji wnioskowania,
- dodanie enzymu *BbvI* oraz enzymu ligazy, które rozpoczną reakcję naprzemiennego cięcia i łączenia fragmentów DNA, co prowadzi do uzyskania odpowiedzi,
- ewentualne wzmocnienie sygnału informującego o odpowiedzi systemu (w tym o możliwym wystąpieniu sprzeczności) za pomocą PCR,
- odczytanie wyniku za pomocą elektroforezy w żelu.

Niestety ze względu na wysokie koszty eksperymentów oraz brak środków finansowych, wbrew początkowym planom, nie udało się przeprowadzić omawianych badań w ramach tej pracy. Omówiona powyżej koncepcja eksperymentów i zastosowanych dodatkowych elementów została dokładnie skonsultowana z Katedrą Genetyki Molekularnej Wydziału Biologii i Ochrony Środowiska Uniwersytetu Łódzkiego, która potwierdziła możliwość ich poprawnego wykonania. Oczywiście w przyszłości, w miarę możliwości finansowych, planowana jest praktyczna implementacja omówionych powyżej eksperymentów.

## Podsumowanie

Głównym celem pracy było skonstruowanie i szczegółowe przedstawienie nowej koncepcji w pełni autonomicznego systemu wnioskującego za pomocą łańcuchów DNA, opartego na mechanizmie splatania (czyli naprzemiennego cięcia i łączenia się odpowiednich łańcuchów), który w przeciwieństwie do istniejących (szczególnie prac omówionych szczegółowo w rozdziale 3) uwzględnia implementację większości praw klasycznego rachunku zdań i jego tautologii. W porównaniu z zaprezentowanymi przez inne grupy badawcze wcześniejszymi koncepcjami, można powiedzieć, że ta zdecydowanie rozszerza ich możliwości, a przede wszystkim:

- wprowadza pojęcie negacji wraz z możliwością wyciągania wniosków również na podstawie literalów o wartości *falszu* jak i wniosków o wartości *falszu*,
- aksjomaty początkowe, wyciągane wnioski czy zapytania do systemu mogą być również w formie koniunkcji czy alternatywy,
- w trakcie wnioskowania korzysta z praw klasycznego rachunku zdań, które są naturalną konsekwencją wprowadzenia spójnika negacji, między innymi wykorzystuje regułę kontrapozycji czy prawa de Morgana,
- sygnalizuje o ewentualnym pojawieniu się sprzeczności (czyli równoczesnym wystąpieniu dwóch przeciwnych wartości tego samego zdania atomowego), niezależnie czy wynikła ona z błędnie dobranych aksjomatów początkowych czy powstała w trakcie reakcji, w wyniku kolejnych wnioskowań.

Zastosowanych zostało też kilka innowacji związanych z budową cząsteczek:

- zastosowanie cząsteczek symetrycznych, które pozwalają na wykonanie (zależnie od warunków początkowych) różnych przekształceń tej samej sekwencji (np. na podstawie  $p$ ,  $(\neg p \vee q)$  wnioskujemy  $q$ , podczas gdy na podstawie  $\neg q$ ,  $(\neg p \vee q)$  wnioskujemy  $\neg p$ ; umieszczenie w próbówce tych wszystkich cząsteczek jednocześnie zwróci informację o sprzeczności),
- fragment rozpoznawalny przez enzym restrykcyjny umieszczony został w dwóch końcach tej samej cząsteczki, zatem cięcie w obrębie takiej sekwencji może pojawić się z obu jej stron (a w niektórych przypadkach jednocześnie),
- cząsteczki mogą posiadać „lepkie końce” dobrane w taki sposób, że są komplementarne z samymi sobą i w wyniku połączenia dwóch takich samych fragmentów DNA (oczywiście w obrębie jednej próbówki wszystkie cząsteczki są zwielokrotnione) „domykają się” tworząc cząsteczkę nie posiadającą „lepkich końców”, która oznacza konkretną odpowiedź na zadane pytanie.

Implementacja laboratoryjna omawianej koncepcji opiera się na przeprowadzonych już wcześniej z wynikiem pozytywnym doświadczeniach, między innymi zaprezentowanych w [12] oraz [27]. Wszystkie etapy jej przygotowania, wraz z konkretnymi przykładami doświadczeń, zostały szczegółowo przedstawione w tej pracy (rozdział 6). Ze względu na ich czasochłonność, kosztowność i brak środków finansowych nie zostały one jeszcze przeprowadzone, ale ich poprawność została potwierdzona przez specjalistów-genetyków

z Katedry Genetyki Molekularnej Wydziału Biologii i Ochrony Środowiska Uniwersytetu Łódzkiego, z którą od lat nasza Katedra współpracuje. Przeprowadzenie właściwych badań jest planowane w przyszłości.

Oprócz przedstawienia nowatorskiej koncepcji, w pracy zawarta została znaczna ilość informacji wprowadzających do tematyki logiki matematycznej (rozdział 1) oraz obliczeń DNA (szczególnie rozdziały 2 i 3), dzięki którym zrozumienie właściwego modelu powinno być łatwiejsze. W celu szczegółowego przedstawienia działania systemu, w każdej części znalazło się sporo przykładów cząsteczek DNA i przykładowych reakcji (rozdziały 4 i 5), a także dodatkowa analiza omawianego systemu i jego ograniczeń (podrozdział 5.1). Ze względu na techniczne ograniczenia narzędzi biologicznych, system ten w swojej podstawowej wersji nie odwzorowuje pełnej zgodności z logiką matematyczną, a jego rozbieżności zostały szczegółowo omówione i przeanalizowane (podrozdział 5.1). Drobne modyfikacje (omówione w pracy) pozwalają na rozwiązanie również tych problemów. Większość z nich możemy rozwiązać za pomocą sklonowania danych do kolejnej próbki (lub kilku), zastosowania zmodyfikowanego (lecz podobnego) sposobu kodowania zapytań (podrozdział 5.3.1) lub potencjalnie wynalezionych w przyszłości nowych enzymów restrykcyjnych (podrozdział 5.3.2). W pracy przedstawione zostały także dodatkowe zastosowania nowej koncepcji: częściowe rozwiązanie problemu SAT, obliczanie wartości funkcji logicznych (podrozdział 5.2) jak i wprowadzona koncepcja „biblioteki łańcuchów DNA” pozwalająca na powielanie i wykorzystanie raz zamówionych łańcuchów dla wielu doświadczeń laboratoryjnych (podrozdział 6.2).

## Dodatek A: „lepkie końce” o długości 4

W poniższym dodatku zaprezentowane zostały wszystkie możliwe do uzyskania unikalne sekwencje czwórek nukleotydów. Jest to 240 łańcuchów niesymetrycznych, co oznacza 120 komplementarnych ze sobą par sekwencji, oraz 16 łańcuchów symetrycznych, które mogą połączyć się komplementarnie z samymi sobą.

Poniżej kolorem niebieskim zostały zaznaczone pary, które nie mogą być w omawianym systemie wykorzystywane, gdyż utworzyłyby sekwencję rozpoznawalną przez enzym *BbvI* w niepożądanym miejscu (co mogłoby spowodować błędne obliczenia). Do dyspozycji spośród poniższych zostaje zatem 66 unikalnych par do reprezentacji literałów oraz 16 pojedynczych sekwencji przeznaczonych do reprezentacji zapytań. Istnieje też możliwość dostosowania par przeznaczonych do reprezentacji literałów na potrzeby reprezentacji większej liczby zapytań, co przedstawione zostało w podrozdziale 5.1.2.

### Unikalne „lepkie końce”:

<u>L.p.</u>	<u>Łańcuch</u>	<u>Łańc. komplementarny</u>	<u>L.p.</u>	<u>Łańcuch</u>	<u>Łańc. komplementarny</u>
1	3' - AAAA - 5'	3' - TTTT - 5'	2	3' - AAAC - 5'	3' - GTTT - 5'
3	3' - AAAG - 5'	3' - CTTT - 5'	4	3' - AAAT - 5'	3' - ATTT - 5'
5	3' - AACA - 5'	3' - TGTT - 5'	6	3' - AACG - 5'	3' - GGTT - 5'
7	3' - AACG - 5'	3' - CGTT - 5'	8	3' - AACT - 5'	3' - AGTT - 5'
9	3' - AAGA - 5'	3' - TCCT - 5'	10	3' - AAGC - 5'	3' - GCTT - 5'
11	3' - AAGG - 5'	3' - CCTT - 5'	12	3' - AAGT - 5'	3' - ACTT - 5'
13	3' - AATA - 5'	3' - TATT - 5'	14	3' - AATC - 5'	3' - GATT - 5'
15	3' - AATG - 5'	3' - CATT - 5'	16	3' - ACAA - 5'	3' - TTGT - 5'
17	3' - ACAC - 5'	3' - GTGT - 5'	18	3' - ACAG - 5'	3' - CTGT - 5'
19	3' - ACAT - 5'	3' - ATGT - 5'	20	3' - ACCA - 5'	3' - TGGT - 5'
21	3' - ACCC - 5'	3' - GGGT - 5'	22	3' - ACCG - 5'	3' - CGGT - 5'
23	3' - ACCT - 5'	3' - AGGT - 5'	24	3' - ACGA - 5'	3' - TCGT - 5'
25	3' - ACGC - 5'	3' - GCGT - 5'	26	3' - ACGG - 5'	3' - CCGT - 5'
27	3' - ACTA - 5'	3' - TAGT - 5'	28	3' - ACTC - 5'	3' - GAGT - 5'
29	3' - ACTG - 5'	3' - CAGT - 5'	30	3' - AGAA - 5'	3' - TTCT - 5'
31	3' - AGAC - 5'	3' - GTCT - 5'	32	3' - AGAG - 5'	3' - CTCT - 5'
33	3' - AGAT - 5'	3' - ATCT - 5'	34	3' - AGCA - 5'	3' - TGCT - 5'
35	3' - AGCC - 5'	3' - GGCT - 5'	36	3' - AGCG - 5'	3' - CGCT - 5'
37	3' - AGGA - 5'	3' - TCCT - 5'	38	3' - AGGC - 5'	3' - GCCT - 5'
39	3' - AGGG - 5'	3' - CCCT - 5'	40	3' - AGTA - 5'	3' - TACT - 5'
41	3' - AGTC - 5'	3' - GACT - 5'	42	3' - AGTG - 5'	3' - CACT - 5'
43	3' - ATAA - 5'	3' - TTAT - 5'	44	3' - ATAC - 5'	3' - GTAT - 5'

<u>L.p.</u>	<u>Łańcuch</u>	<u>Łańc. komplementarny</u>	<u>L.p.</u>	<u>Łańcuch</u>	<u>Łańc. komplementarny</u>
45	3' - ATAG - 5'	3' - CTAT - 5'	46	3' - ATCA - 5'	3' - TGAT - 5'
47	3' - ATCC - 5'	3' - GGAT - 5'	48	3' - ATCG - 5'	3' - CGAT - 5'
49	3' - ATGA - 5'	3' - TCAT - 5'	50	3' - ATGC - 5'	3' - GCAT - 5'
51	3' - ATGG - 5'	3' - CCAT - 5'	52	3' - ATTA - 5'	3' - TAAT - 5'
53	3' - ATTC - 5'	3' - GAAT - 5'	54	3' - ATTG - 5'	3' - CAAT - 5'
55	3' - CAAA - 5'	3' - TTTG - 5'	56	3' - CAAC - 5'	3' - GTTG - 5'
57	3' - CAAG - 5'	3' - CTTG - 5'	58	3' - CACA - 5'	3' - TGTG - 5'
59	3' - CACC - 5'	3' - GGTG - 5'	60	3' - CACG - 5'	3' - CGTG - 5'
61	3' - CAGA - 5'	3' - TCTG - 5'	62	3' - CAGC - 5'	3' - GCTG - 5'
63	3' - CAGG - 5'	3' - CCTG - 5'	64	3' - CATA - 5'	3' - TATG - 5'
65	3' - CATC - 5'	3' - GATG - 5'	66	3' - CCAA - 5'	3' - TTGG - 5'
67	3' - CCAC - 5'	3' - GTGG - 5'	68	3' - CCAG - 5'	3' - CTGG - 5'
69	3' - CCCA - 5'	3' - TGGG - 5'	70	3' - CCCC - 5'	3' - GGGG - 5'
71	3' - CCCG - 5'	3' - CGGG - 5'	72	3' - CCGA - 5'	3' - TCGG - 5'
73	3' - CCGC - 5'	3' - GCGG - 5'	74	3' - CCTA - 5'	3' - TAGG - 5'
75	3' - CCTC - 5'	3' - GAGG - 5'	76	3' - CGAA - 5'	3' - TTCG - 5'
77	3' - CGAC - 5'	3' - GTCG - 5'	78	3' - CGAG - 5'	3' - CTCG - 5'
79	3' - CGCA - 5'	3' - TGCG - 5'	80	3' - CGCC - 5'	3' - GGCG - 5'
81	3' - CGGA - 5'	3' - TCCG - 5'	82	3' - CGGC - 5'	3' - GCCG - 5'
83	3' - CGTA - 5'	3' - TACG - 5'	84	3' - CGTC - 5'	3' - GACG - 5'
85	3' - CTAA - 5'	3' - TTAG - 5'	86	3' - CTAC - 5'	3' - GTAG - 5'
87	3' - CTCA - 5'	3' - TGAG - 5'	88	3' - CTCC - 5'	3' - GGAG - 5'
89	3' - CTGA - 5'	3' - TCAG - 5'	90	3' - CTGC - 5'	3' - GCAG - 5'
91	3' - CTTA - 5'	3' - TAAG - 5'	92	3' - CTTC - 5'	3' - GAAG - 5'
93	3' - GAAA - 5'	3' - TTTT - 5'	94	3' - GAAC - 5'	3' - GTTC - 5'
95	3' - GACA - 5'	3' - TGTC - 5'	96	3' - GACC - 5'	3' - GGTC - 5'
97	3' - GAGA - 5'	3' - TCTC - 5'	98	3' - GAGC - 5'	3' - GCTC - 5'
99	3' - GATA - 5'	3' - TATC - 5'	100	3' - GCAA - 5'	3' - TTGC - 5'
101	3' - GCAC - 5'	3' - GTGC - 5'	102	3' - GCCA - 5'	3' - TGGC - 5'
103	3' - GCCC - 5'	3' - GGGC - 5'	104	3' - GCGA - 5'	3' - TCGC - 5'
105	3' - GCTA - 5'	3' - TAGC - 5'	106	3' - GGAA - 5'	3' - TTCC - 5'
107	3' - GGAC - 5'	3' - GTCC - 5'	108	3' - GGCA - 5'	3' - TGCC - 5'
109	3' - GGGA - 5'	3' - TCCC - 5'	110	3' - GGTA - 5'	3' - TACC - 5'
111	3' - GTAA - 5'	3' - TTAC - 5'	112	3' - GTCA - 5'	3' - TGAC - 5'



<u>L.p.</u>	<u>Łańcuch</u>	<u>Łańc. komplementarny</u>	<u>L.p.</u>	<u>Łańcuch</u>	<u>Łańc. komplementarny</u>
113	3' - GTGA - 5'	3' - TCAC - 5'	114	3' - GTTA - 5'	3' - TAAC - 5'
115	3' - TAAA - 5'	3' - TTTA - 5'	116	3' - TACA - 5'	3' - TGTA - 5'
117	3' - TAGA - 5'	3' - TCTA - 5'	118	3' - TCAA - 5'	3' - TTGA - 5'
119	3' - TCCA - 5'	3' - TGGA - 5'	120	3' - TGAA - 5'	3' - TTCA - 5'

**„Lepkie końce” komplementarne ze sobą:**

<u>L.p.</u>	<u>Łańcuch</u>	<u>L.p.</u>	<u>Łańcuch</u>	<u>L.p.</u>	<u>Łańcuch</u>	<u>L.p.</u>	<u>Łańcuch</u>
1	3' - AATT - 5'	2	3' - ACGT - 5'	3	3' - AGCT - 5'	4	3' - ATAT - 5'
5	3' - CATG - 5'	6	3' - CCGG - 5'	7	3' - CGCG - 5'	8	3' - CTAG - 5'
9	3' - GATC - 5'	10	3' - GCGC - 5'	11	3' - GGCC - 5'	12	3' - GTAC - 5'
13	3' - TATA - 5'	14	3' - TCGA - 5'	15	3' - TGCA - 5'	16	3' - TTAA - 5'

## Bibliografia

- [1] ADLEMAN L.: Molecular computation of solutions to combinatorial problems. *Science* 1994; 226: 1021-1024.
- [2] BENENSON Y, PAZ-ELIZUR T, ADAR R, KEINAN E, LIVNEH Z, SHAPIRO E.: Programmable and autonomous computing machine made of biomolecules. *Nature* 2001; 414: 430-434.
- [3] BENENSON Y, ADAR R, PAZ-ELIZUR T, LIVNEH Z, SHAPIRO E.: DNA molecule provides a computing machine with both data and fuel. *PNAS* 2003; 100: 2191-2196.
- [4] BENENSON Y, GIL B, BEN-DOR U, ADAR R, SHAPIRO E.: An autonomous molecular computer for logical control of gene expression. *Nature* 2004; 429: 423-429.
- [5] BENENSON Y.: Biomolecular computing systems: principles, progress and potential. *Nature Reviews Genetics*, 2012; 13: 455–468.
- [6] CAVALIERE M., JONOSKA N., YOGEV S., PIRAN R., KEINAN E., SEEMAN N.: Biomolecular implementation of computing devices with unbounded memory. *Lecture Notes in Computer Science* 2005; 3384: 35-49.
- [7] GRZEGORCZYK A.: Zarys logiki matematycznej. PWN, 1975.
- [8] HEAD T.: Formal language theory and DNA: an analysis of the generative capacity of specific recombinant behaviors. *Bull. Math. Biology*, 1987; 49: 737-759.
- [9] IN-HEE L, JI-YOON P, HAE-MAN J, YOUNG-GYU C, BYOUNG-TAK Z.: DNA Implementation of Theorem Proving with Resolution Refutation in Propositional Logic. *Proceeding DNA8 Revised Papers from the 8th International Workshop on DNA Based Computers: DNA Computing* 2002; 1: 156-167.
- [10] GUO Y., WEI B., XIAO S., YAO D., LI H., XU H., SONG T., LI X., LIANG H.: Recent advances in molecular machines based on toehold-mediated strand displacement reaction. *Quant Biol*; 2017; 5: 25–41.
- [11] KRASIŃSKI T, SAKOWSKI S.: Przegląd modeli i praktycznych implementacji DNA obliczeń. *Studia Informatica*, 2008; 29: 5-31.
- [12] KRASIŃSKI T, SAKOWSKI S.: Extended Shapiro Finite State Automaton. *Foundations of Computing and Decision Science*, 2008; 33: 241-255.
- [13] KRASIŃSKI T, SAKOWSKI S, POPLAWSKI T, BŁASIAK J.: Komputery DNA, *Postepy biochemii* 2011; 57 (1): 13-23.
- [14] KRASIŃSKI T, SAKOWSKI S, POPLAWSKI T.: Autonomous push-down automaton built on DNA. *Informatica* 2012; 36: 263-276.
- [15] KRASIŃSKI T, SAKOWSKI S, POPLAWSKI T.: Towards an autonomous multistate biomolecular devices built on DNA, *Nature and Biologically Inspired Computing (NaBIC)*, *IEEE*, 2014; 1: 23-28.
- [16] LIPTON R.: DNA solution of hard computational problems. *Science* 1995; 268: 542-545.
- [17] MANCA V, MARTIN-VIDE C, PAUN G.: New computing paradigms suggested by DNA computing: computing by carving, *Biosystems* 1999; 52/1-3: 47-54.
- [18] MANCA V, ZANDRON C.: A clause string DNA algorithm for SAT. In DNA 2001. LNCS, 2340, by N. Jonoska and N. C. Seeman: 172–181. *Springer, Heidelberg*, 2002.
- [19] MARTYN A, PAUL D.: DNA simulation of boolean circuits. *Technical Report CTAG-97009, Department of Computer Science, University of Liverpool in UK*, 1997.
- [20] MARTYN A.: DNA Computing. *The Encyclopedia of Complexity and System Science*. *Springer*, 2008.
- [21] MULAŃKA J, WĘGLEŃSKI P, BORSUK P.: Sposób dowodzenia dedukcyjnego za pomocą wnioskowania regresywnego. *Opis Patentowy Rzeczypospolitej Polskiej*, 1997.

- [22] MULAWKA J, WĘGLEŃSKI P, BORSUK P.: Sposób wnioskowania dedukcyjnego z wykorzystaniem drzewa decyzyjnego. *Opis Patentowy Rzeczypospolitej Polskiej*, 1997.
- [23] OGIHARA M, RAY A.: Simulating Boolean circuits on a DNA computer. *RECOMB '97 Proceedings of the first annual international conference on Computational molecular biology*, 1997; 1: 226-231.
- [24] ORDÓÑEZ-GUILLÉN N., MARTÍNEZ-PÉREZ I.: Catalytic DNA Strand Displacement Cascades Applied to Logic Programming, *in IEEE Access*, 2019; 7: 100428-100441.
- [25] PAUN G, ROZENBERG G, SALOMAA A.: DNA Computing. *New Computing Paradigms. Springer-Verlag*, 1999.
- [26] QIAN L, WINFREE E.: Scaling Up Digital Circuit Computation with DNA Strand Displacement Cascades. *Science* 2011; 6034: 1196-1201.
- [27] RAN T, KAPLAN S, SHAPIRO E.: Molecular implementation of simple logic program. *Nature Nanotechnology*, 2009; 203(2): 1-7.
- [28] RAY K., MONDAL M.: Logical Inference by DNA Strand Algebra, *New Mathematics and Natural Computation*, 2016; 12: 29-44.
- [29] RODRÍGUEZ-PATÓN A, LARREA J, SAINZ DE MURIETA I.: Inference with DNA molecules. *In: CALUDE C, HAGIYA M, MORITA K, ROZENBERG G, TIMMIS J.: UC 2010. LNCS, Springer Heidelberg*, 2010; 6079: 192.
- [30] RODRÍGUEZ-PATÓN A, SAINZ DE MURIETA I, SOSÍK P.: DNA strand displacement system running logic programs. *Biosystems*, 2014; 115: 5–12.
- [31] ROGOWSKI Ł.: Implementacje systemów logicznych za pomocą DNA – koncepcje i pomysły rozbudowy. *WMil Preprint (Wydział Matematyki i Informatyki Uniwersytetu Łódzkiego)*, 2011.
- [32] ROGOWSKI Ł.: Wnioskowanie logiczne za pomocą DNA. Reguła kontrapozycji, *Studia i Materiały Informatyki Stosowanej*, 2013; 13: 11-18.
- [33] ROGOWSKI Ł.: DNA computer-based search for unsatisfiable formulas, *PhD Interdisciplinary Journal, Special Issue: BioTech Conference*, 2013; 1: 105-110.
- [34] ROGOWSKI Ł, KRASIŃSKI T, SAKOWSKI S, BŁASIAK J, POPLAWSKI T.: Wnioskowanie logiczne przy użyciu DNA, *Postępy Biologii Komórki*, 2013; 4(40): 645-658
- [35] ROGOWSKI Ł, SOSÍK P.: The laws of natural deduction in inference by DNA computer, *The Scientific World Journal*, 2014; doi:10.1155/2014/834237.
- [36] ROTHEMUND P.: A DNA and restriction enzyme implementation of Turing machines. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science. American Mathematical Society*, 1995; 27: 75-120.
- [37] ROZENBERG G, BACK T, KOK J.: Handbook of Natural Computing, *Springer* 2012.
- [38] SAINZ DE MURIETA I.: Inference models in DNA Computing, *Rozprawa doktorska, Departamento de Inteligencia Artificial, Facultad de Informática, Politécnica de Madrid*, 2013.
- [39] SAKAKIBARAY, SUYAMA A.: Intelligent DNA chips: logical operation of gene expression profiles on DNA computers. *Proceedings 11th Workshop on Genome Informatics*, 2000: 33-42.
- [40] SAKOWSKI S.: Rozszerzenie biomolekularnego automatu Shapiro, *Rozprawa doktorska, Wydział Automatyki, Elektroniki i Informatyki, Politechnika Śląska w Gliwicach*, 2010.
- [41] SAKOWSKI S, KRASIŃSKI T, SARNIK J, BŁASIAK J, WALDMAJER J, POPLAWSKI T.: A detailed experimental study of a DNA computer with two endonucleases, *Zeitschrift für Naturforschung*, 2017; 72(7-8): 303-313.

- [42] SAKOWSKI S, KRASIŃSKI T, SARNIK J, BŁASIAK J, WALDMAJER J, POPLAWSKI T.: Biomolecular computers with multiple restriction enzymes, *Genetics and molecular biology*, 2017; 40(4): 860-870.
- [43] SAKAMOTO K, GOZU H, KOMIYA K, KIGA D, YOKOYAMA S, YOKOMORI T, HAGIVA M.: Molecular computation by DNA hairpin formation. *Science* 2000; 288: 1223–1226.
- [44] SIPSER M.: Introduction to the Theory of Computation, 2nd Edition, Int. Edition, 2006.
- [45] SORENI M, YOGEV S, KOSOV E, SHOHAM Y, KEINAN E.: Parallel biomolecular computation on surfaces with advanced finite automata. *J. Am. Chem. Soc.*, 2005; 127: 3935-3943.
- [46] UNOLD O.: Jak liczy komputer DNA, *Delta – matematyka, fizyka, astronomia, informatyka*, 2008; 7(410): 8–10.
- [47] UNOLD O, TROĆ M.: Biomolekularne modele obliczeniowe. *Inżynieria Komputerowa*, 2005: 73-85.
- [48] UNOLD O, TROĆ M.: Restriction Enzyme Computation. *Proceedings 7th International Work Conference on Artificial and Natural Neural Networks, IWANN 2003, Lecture Notes in Computer Science*, 2003; 2686: 686–693.
- [49] UNOLD O, TROĆ M, DOBOSZ T, TRUSIEWICZ A.: Extended molecular computing model. *WSEAS Transactions on Biology and Biomedicine*, 2004; 1: 15-19.
- [50] WANG F., SU H., XU J., WANG Q., ZHOU X.: High-efficiency and integrable DNA arithmetic and logic system based on strand displacement synthesis. *Nature Communications*, 2019; 10: 5390.
- [51] WANG F., LV H., LI Q., LI J., ZHANG X., SHI J., WANG L., FAN C.: Implementing digital computing with DNA-based switching circuits. *Nature Communications*, 2020; 11: 121.
- [52] WATSON J, CRICK F.: A Structure for Deoxyribose Nucleic Acid. *Nature*, 1953; 171: 737-738.
- [53] WĄSIEWICZ P, MULAŁKA J, PŁUCIENNICZAK A.: Logical operations with DNA strands. *RSCTC'98, Proceedings of 1st International Conference on Rough Sets and Current Trends in Computing*, 1998.
- [54] WĄSIEWICZ P, JANCZAK T, MULAŁKA J, PŁUCIENNICZAK A.: The inference via DNA computing. *Congress on Evolutionary Computation, Washington, USA*, 1999; 2: 988-993.
- [55] WĄSIEWICZ P, BORSUK P, MULAŁKA J, WĘGLEŃSKI P.: Implementation of Data Flow Logical Operations via self-assembly of DNA. *Lecture Notes in Computer Science*, 1999; 1586: 174-182.
- [56] WĄSIEWICZ P, JANCZAK T, MULAŁKA J.: The inference based on molecular computing. *Cybernetics and Systems*, 2000; 31: 283-315.
- [57] WĄSIEWICZ P, MALINOWSKI A, NOWAK R, MULAŁKA J, BORSUK P, WĘGLEŃSKI P, PŁUCIENNICZAK A.: DNA computing: implementation of data flow logical operations. *Future Generations Computer Systems*, 2001; 17: 361-378.
- [58] WĄSIEWICZ P, MULAŁKA J.: Molecular genetic programming. *Soft Computing. Springer*, 2001; 5: 106-113.
- [59] WOOD D.: A DNA Computing Algorithm for Directed Hamiltonian Paths. *To appear in Genetic Programming: Proceedings of the Third Annual Conference, July 22–25, 1998. Morgan Kaufmann*. 1998.
- [60] YURKE B., TURBERFIELD A., MILLS A., SIMMEL F., NEUMANN J.: A DNA-fuelled molecular machine made of DNA. *Nature*, 2000; 406: 605–608.

# Spis rycin

## 2. Wprowadzenie do obliczeń DNA.....- 17 -

### 2. 1. Budowa i podstawowe operacje na DNA .....- 17 -

- Ryc. 1a: przykładowe komplementarne łańcuchy jednoniciowe
- Ryc. 1b: te same łańcuchy po połączeniu się w cząsteczkę dwuniciową
- Ryc. 2a: przykładowe łańcuchy o komplementarnych „lepkich końcach”
- Ryc. 2b: te same łańcuchy po połączeniu się nukleotydów komplementarnych
- Ryc. 2c: pojedyncza cząsteczka powstała w wyniku działania enzymu ligazy
- Ryc. 3a: sekwencja nukleotydów rozpoznawalna przez enzym EcoRI
- Ryc. 3b: efekt trawienia powyższej sekwencji enzymem EcoRI
- Ryc. 4a: sekwencja nukleotydów rozpoznawalna przez enzym HphI
- Ryc. 4b: efekt trawienia powyższej sekwencji enzymem HphI
- Ryc. 5a: sekwencja nukleotydów rozpoznawalna przez enzym BstXI
- Ryc. 5b: efekt trawienia powyższej sekwencji enzymem BstXI
- Ryc. 6a: sekwencja nukleotydów rozpoznawalna przez enzym FokI
- Ryc. 6b: efekt trawienia powyższej sekwencji enzymem FokI
- Ryc. 7a: sekwencja nukleotydów rozpoznawalna przez enzym BbvI
- Ryc. 7b: efekt trawienia powyższej sekwencji enzymem BbvI
- Ryc. 8a: sekwencja nukleotydów rozpoznawalna przez enzym BsmAI
- Ryc. 8b: efekt trawienia powyższej sekwencji enzymem BsmAI
- Ryc. 9a: sekwencja nukleotydów rozpoznawalna przez enzym HgaI
- Ryc. 9b: efekt trawienia powyższej sekwencji enzymem HgaI
- Ryc. 10a: dwuniciowa cząsteczka DNA przed rozpoczęciem procesu replikacji
- Ryc. 10b: rozdzielenie (denaturacja) do postaci dwóch jednoniciowych łańcuchów
- Ryc. 10c: dołączenie „starterów” czyli krótszych fragmentów komplementarnych
- Ryc. 10d: reakcja łańcuchowa polimerazy w trakcie działania
- Ryc. 10e: zakończona reakcja łańcuchowa polimerazy – zamiast jednej uzyskujemy dwie identyczne cząsteczki DNA
- Ryc. 11a: synteza pojedynczych nukleotydów w odpowiedniej kolejności
- Ryc. 11b: połączenie nukleotydów do jednoniciowych oligonukleotydów
- Ryc. 11c: łączenie ze sobą powstałych łańcuchów na zasadzie komplementarności
- Ryc. 12a: przykładowe cząsteczki DNA ułożone w losowej kolejności na płycie żelu elektroforetycznego
- Ryc. 12b: te same cząsteczki DNA po zakończeniu reakcji elektroforezy w żelu

### 2. 3. Model filtrowania .....- 23 -

- Ryc. 13: przykład prostego grafu skierowanego o czterech wierzchołkach
- Ryc. 14a: przykładowe kodowanie wierzchołka  $S_0$  w powyższym grafie
- Ryc. 14b: przykładowe kodowanie wierzchołka  $S_1$  w powyższym grafie
- Ryc. 14c: przykładowe kodowanie wierzchołka  $S_2$  w powyższym grafie
- Ryc. 14d: przykładowe kodowanie wierzchołka  $S_3$  w powyższym grafie
- Ryc. 15a: kodowanie przejścia  $S_0 \rightarrow S_1$  w powyższym grafie
- Ryc. 15b: kodowanie przejścia  $S_1 \rightarrow S_2$  w powyższym grafie
- Ryc. 15c: kodowanie przejścia  $S_1 \rightarrow S_3$  w powyższym grafie
- Ryc. 15d: kodowanie przejścia  $S_3 \rightarrow S_0$  w powyższym grafie
- Ryc. 15e: kodowanie przejścia  $S_3 \rightarrow S_2$  w powyższym grafie
- Ryc. 16a: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_2$
- Ryc. 16b: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3$
- Ryc. 16c: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$
- Ryc. 16d: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0 \rightarrow S_1$
- Ryc. 16e: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0 \rightarrow S_1 \rightarrow S_2$
- Ryc. 16f: dwuniciowa sekwencja reprezentująca drogę  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2$
- Ryc. 17a: dwuniciowa sekwencja reprezentująca drogę łączącą wszystkie wierzchołki  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0 \rightarrow S_1 \rightarrow S_2$
- Ryc. 17b: dwuniciowa sekwencja reprezentująca drogę łączącą wszystkie wierzchołki  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2$
- Ryc. 18: dwuniciowa sekwencja reprezentująca drogę Hamiltona  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_2$

### 2. 4. Model splatania .....- 26 -

- Ryc. 19a: przykładowa sekwencja  $a_1Ea_2$  zawierająca miejsce rozpoznawane przez enzym EcoRI
- Ryc. 19b: przykładowa sekwencja  $b_1Eb_2$  zawierająca miejsce rozpoznawane przez enzym EcoRI

Ryc. 20a: sekwencja  $a_1Ea_2$  rozcięta za pomocą enzymu *EcoRI*  
 Ryc. 20b: sekwencja  $b_1Eb_2$  rozcięta za pomocą enzymu *EcoRI*  
 Ryc. 21a: sekwencja  $a_1Eb_2$  powstała w wyniku splecenia  
 Ryc. 21b: sekwencja  $a_2Eb_1$  powstała w wyniku splecenia  
 Ryc. 22a: sekwencja  $a_1Ea_1'$  powstała w wyniku splecenia  
 Ryc. 22b: sekwencja  $a_2Ea_2'$  powstała w wyniku splecenia  
 Ryc. 22c: sekwencja  $b_1Eb_1'$  powstała w wyniku splecenia  
 Ryc. 22d: sekwencja  $b_2Eb_2'$  powstała w wyniku splecenia  
 Ryc. 23: wszystkie możliwe przejścia automatu dwustanowego, dwusymbolowego  
 Ryc. 24a: sekwencja kodowania przejścia  $S_0 \rightarrow S_0$ , dla symbolu  $a$   
 Ryc. 24b: sekwencja kodowania przejścia  $S_0 \rightarrow S_1$ , dla symbolu  $a$   
 Ryc. 24c: sekwencja kodowania przejścia  $S_0 \rightarrow S_0$ , dla symbolu  $b$   
 Ryc. 24d: sekwencja kodowania przejścia  $S_0 \rightarrow S_1$ , dla symbolu  $b$   
 Ryc. 24e: sekwencja kodowania przejścia  $S_1 \rightarrow S_0$ , dla symbolu  $a$   
 Ryc. 24f: sekwencja kodowania przejścia  $S_1 \rightarrow S_1$ , dla symbolu  $a$   
 Ryc. 24g: sekwencja kodowania przejścia  $S_1 \rightarrow S_0$ , dla symbolu  $b$   
 Ryc. 24h: sekwencja kodowania przejścia  $S_1 \rightarrow S_1$ , dla symbolu  $b$   
 Ryc. 25a: kodowanie symbolu  $a$   
 Ryc. 25b: kodowanie symbolu  $b$   
 Ryc. 25c: kodowanie symbolu terminalnego  $t$   
 Ryc. 26: kodowanie słowa  $B=bab$  i stanu początkowego  $S_0$  zgodnie z powyższym schematem  
 Ryc. 27a: sekwencja detektorowa dla stanu  $S_0$   
 Ryc. 27b: sekwencja detektorowa dla stanu  $S_1$   
 Ryc. 28: przykładowy automat uzyskany poprzez ograniczenie modelu głównego do trzech przejść  
 Ryc. 29a: sekwencja kodowania przejścia  $S_0 \rightarrow S_0$ , dla symbolu  $b$   
 Ryc. 29b: sekwencja kodowania przejścia  $S_0 \rightarrow S_1$ , dla symbolu  $b$   
 Ryc. 29c: sekwencja kodowania przejścia  $S_1 \rightarrow S_0$ , dla symbolu  $a$   
 Ryc. 30a: ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_0$ , dla symbolu  $b$   
 Ryc. 30b: trawienie powyższej cząsteczki enzymem restrykcyjnym *FokI*  
 Ryc. 30c: ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_1$ , dla symbolu  $b$   
 Ryc. 30d: trawienie powyższej cząsteczki enzymem restrykcyjnym *FokI*  
 Ryc. 30e: ligacja cząsteczek słowa wejściowego i przejścia  $S_1 \rightarrow S_0$ , dla symbolu  $a$   
 Ryc. 30f: trawienie powyższej cząsteczki enzymem restrykcyjnym *FokI*  
 Ryc. 30g: ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_0$ , dla symbolu  $b$   
 Ryc. 30h: trawienie powyższej cząsteczki enzymem restrykcyjnym *FokI*  
 Ryc. 30i: jednoczesna ligacja cząsteczek słowa wejściowego i przejścia  $S_0 \rightarrow S_1$ , dla symbolu  $b$   
 Ryc. 30j: trawienie powyższej cząsteczki enzymem restrykcyjnym *FokI*  
 Ryc. 30k: ligacja cząsteczki detektorowej dla stanu  $S_0$  oraz symbolu terminalnego w stanie  $S_0$   
 Ryc. 30l: ligacja cząsteczki detektorowej dla stanu  $S_1$  oraz symbolu terminalnego w stanie  $S_1$

### 3. Znane implementacje systemów wnioskowania. ....- 33 -

#### 3. 1. System Olgerda Unolda i Macieja Trocia .....- 33 -

Ryc. 31a: schemat molekuly reprezentującej fakt  $p$   
 Ryc. 31b: schemat molekuly reprezentującej fakt  $q$   
 Ryc. 32a: schemat molekuly reprezentującej implikację  $(p \rightarrow q)$   
 Ryc. 32b: schemat molekuly reprezentujących implikację  $((p \wedge q) \rightarrow r)$   
 Ryc. 32c: schemat molekuly reprezentujących implikację  $((p \wedge q) \vee r) \rightarrow s$   
 Ryc. 33: molekula uzupełniająca wnioskowania  
 Ryc. 34: schemat molekuly reprezentującej zapytanie  $q?$   
 Ryc. 35a: schemat molekuly reprezentującej fakt  $a$   
 Ryc. 35b: schemat molekuly reprezentującej fakt  $d$   
 Ryc. 35c: schemat molekuly reprezentującej fakt  $u$   
 Ryc. 35d: schemat molekuly reprezentujących implikację  $((a \wedge b) \vee d) \rightarrow c$   
 Ryc. 35e: schemat molekuly reprezentującej implikację  $((c \wedge u) \rightarrow x)$   
 Ryc. 35f: molekula uzupełniająca wnioskowania  
 Ryc. 35g: molekuly reprezentująca zapytanie  $x?$   
 Ryc. 36a: ligacja molekuly reprezentujących  $a$  oraz  $((a \wedge b) \rightarrow c)$   
 Ryc. 36b: rozcięcie molekuly za pomocą enzymu restrykcyjnego *BseXI*, powstaje molekula  $(b \rightarrow c)$   
 Ryc. 36c: ligacja molekuly reprezentujących  $d$  oraz  $(d \rightarrow c)$   
 Ryc. 36d: rozcięcie molekuly za pomocą enzymu restrykcyjnego *BseXI*

Ryc. 36e: ligacja wniosku i molekuly uzupełniającej, powstaje molekula faktu  $c$   
 Ryc. 36f: ligacja molekul reprezentujących  $c$  oraz  $((c \wedge u) \rightarrow x)$   
 Ryc. 36g: rozcięcie molekuly za pomocą enzymu restrykcyjnego BseXI, powstaje molekula  $(u \rightarrow x)$   
 Ryc. 36h: ligacja molekul reprezentujących  $u$  oraz  $(u \rightarrow x)$   
 Ryc. 36i: rozcięcie molekuly za pomocą enzymu restrykcyjnego BseXI  
 Ryc. 36j: ligacja wniosku i molekuly uzupełniającej, powstaje molekula faktu  $x$   
 Ryc. 36k: ligacja molekul faktu  $x$  i zapytania  $x$ ?  
 Ryc. 37a: schemat molekul reprezentujących argumenty  $a$  oraz  $\neg a$   
 Ryc. 37b: schemat molekul reprezentujących argumenty  $b$  oraz  $\neg b$   
 Ryc. 38a: skladowa funkcji logicznej o jednym argumentem  $a$   
 Ryc. 38b: skladowa funkcji logicznej o jednym argumentem  $\neg b$   
 Ryc. 39: koniunkcja skladowa funkcji logicznej o dwóch argumentach  $(\neg a \wedge \neg b)$   
 Ryc. 40: molekula detektora dla wyjścia funkcji  
 Ryc. 41a: schemat molekul reprezentujących  $a$ : po lewej  $a$ , po prawej  $\neg a$   
 Ryc. 41b: schemat molekul reprezentujących  $b$ : po lewej  $b$ , po prawej  $\neg b$   
 Ryc. 42a: koniunkcja skladowa funkcji logicznej  $(a \wedge \neg b)$   
 Ryc. 42b: koniunkcja skladowa funkcji logicznej  $(\neg a \wedge b)$   
 Ryc. 42c: molekula detektora dla wyjścia funkcji  
 Ryc. 43a: ligacja molekuly  $\neg a$  oraz molekuly iloczynu  $(\neg a \wedge b)$   
 Ryc. 43b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI  
 Ryc. 43c: ligacja molekuly  $b$  oraz pozostalego fragmentu molekuly koniunkcji skladowej  
 Ryc. 43d: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI  
 Ryc. 43e: ligacja pozostalego fragmentu i molekuly terminalnej  
 Ryc. 44a: ligacja molekuly  $\neg a$  oraz molekuly koniunkcji  $(a \wedge \neg b)$   
 Ryc. 44b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI

### 3. 2. System Shai Kaplana, Toma Rana i Ehuda Shapiro.....- 40 -

Ryc. 45a: schemat molekuly reprezentującej fakt  $a$   
 Ryc. 45b: schemat molekuly reprezentującej fakt  $b$   
 Ryc. 46: schemat molekuly implikacji  $(a \rightarrow b)$   
 Ryc. 47: molekula uzupełniająca wnioskowania  
 Ryc. 48a: schemat molekuly reprezentującej fakt  $a$   
 Ryc. 48b: schemat molekuly reprezentującej fakt  $b$   
 Ryc. 49a: schemat molekuly reprezentującej fakt  $a$   
 Ryc. 49b: schemat molekuly reprezentującej implikację  $(a \rightarrow b)$   
 Ryc. 49c: molekula uzupełniająca wnioskowania  
 Ryc. 49d: schemat molekuly reprezentującej zapytanie  $b$ ?  
 Ryc. 50a: ligacja molekuly zapytania  $b$ ? i implikacji  $(a \rightarrow b)$   
 Ryc. 50b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI  
 Ryc. 50c: rozwinięcie wniosku częściowego do postaci jednoniciowej  
 Ryc. 50d: ligacja powstałej cząstki i molekuly uzupełniającej wnioskowania  
 Ryc. 50e: ligacja molekul argumentu  $a$  i zapytania  $a$ ?  
 Ryc. 50f: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI  
 Ryc. 50g: rozpad powstałej krótkiej cząstki na dwie jednoniciowe oraz uaktywnienie fluorescencji  
 Ryc. 51a: schemat molekuly reprezentującej zapytanie  $a$ ?  
 Ryc. 51b: schemat molekuly reprezentującej zapytanie  $b$ ?  
 Ryc. 52: schemat molekuly implikacji  $(a \rightarrow b)$   
 Ryc. 53: schemat molekuly implikacji  $(a \wedge b \rightarrow c)$   
 Ryc. 54: molekula uzupełniająca wnioskowania  
 Ryc. 55a: schemat molekuly zapytania  $a$ ?  
 Ryc. 55b: schemat molekuly zapytania  $b$ ?  
 Ryc. 56a: schemat molekuly reprezentującej fakt  $a$   
 Ryc. 56b: schemat molekuly reprezentującej implikację  $(a \rightarrow b)$   
 Ryc. 56c: molekula uzupełniająca wnioskowania  
 Ryc. 56d: schemat molekuly reprezentującej zapytanie  $b$ ?  
 Ryc. 57a: ligacja molekuly faktu  $a$  oraz implikacji  $(a \rightarrow b)$   
 Ryc. 57b: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI  
 Ryc. 57c: rozwinięcie wniosku częściowego do postaci jednoniciowej  
 Ryc. 57d: ligacja powstałej cząstki i molekuly uzupełniającej wnioskowania  
 Ryc. 57e: ligacja molekul faktu  $b$  i zapytania  $b$ ?  
 Ryc. 57f: rozcięcie molekuly za pomocą enzymu restrykcyjnego FokI  
 Ryc. 57g: rozpad powstałej cząstki na dwie jednoniciowe oraz uaktywnienie fluorescencji

Ryc. 58a: schemat molekuly reprezentujacej fakt  $a$   
 Ryc. 58b: schemat molekuly reprezentujacej fakt  $d$   
 Ryc. 58c: schemat molekuly reprezentujacej fakt  $u$   
 Ryc. 58d: schemat molekul reprezentujacych wnioskowanie  $((a \wedge b) \vee d) \rightarrow c$   
 Ryc. 58e: schemat molekul reprezentujacych wnioskowanie  $((c \wedge u) \rightarrow x)$   
 Ryc. 58f: molekula uzupelniajaca wnioskowania  
 Ryc. 58g: molekuly reprezentujaca zapytanie  $x$ ?  
 Ryc. 59a: ligacja molekuly reprezentujacych  $a$  oraz  $((a \wedge b) \rightarrow c)$   
 Ryc. 59b: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI, powstaje molekula  $(b \rightarrow c)$   
 Ryc. 59c: ligacja molekuly reprezentujacych  $d$  oraz  $(d \rightarrow c)$   
 Ryc. 59d: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI  
 Ryc. 59e: rozwinięcie wniosku czesciowego do postaci jednoniciowej  
 Ryc. 59f: ligacja wniosku i molekuly uzupelniajacej, powstaje molekula faktu  $c$   
 Ryc. 59g: ligacja molekul reprezentujacych  $c$  oraz  $((c \wedge u) \rightarrow x)$   
 Ryc. 59h: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI, powstaje molekula  $(u \rightarrow x)$   
 Ryc. 59i: ligacja molekul reprezentujacych  $u$  oraz  $(u \rightarrow x)$   
 Ryc. 59j: rozcięcie molekuly za pomoca enzymu restrykcyjnego BseXI  
 Ryc. 59k: rozwinięcie wniosku czesciowego do postaci jednoniciowej  
 Ryc. 59l: ligacja wniosku i molekuly uzupelniajacej, powstaje molekula faktu  $x$   
 Ryc. 59m: ligacja molekul faktu  $b$  i zapytania  $b$ ?  
 Ryc. 59n: rozcięcie molekuly za pomoca enzymu restrykcyjnego FokI  
 Ryc. 59o: rozpad powstalej czastki na dwie jednoniciowe oraz uaktywnienie fluorescencji

### 3. 3. Systemy grupy badawczej Jana Mulawki.....- 47 -

Ryc. 60a: schemat molekuly reprezentujacej fakt  $a$   
 Ryc. 60b: schemat molekuly reprezentujacej fakt  $b$   
 Ryc. 60c: schemat molekuly reprezentujacej fakt  $c$   
 Ryc. 60d: schemat molekuly reprezentujacej regule wnioskowania  $(c \wedge a) \rightarrow k$   
 Ryc. 60e: schemat molekuly reprezentujacej regule wnioskowania  $(k \wedge b) \rightarrow l$  (tu zapisanej od prawej do lewej)  
 Ryc. 60f: schemat molekuly reprezentujacej regule wnioskowania  $l \rightarrow m$   
 Ryc. 60g: schemat molekuly reprezentujacej hipoteze  $m$   
 Ryc. 61a: ligacja molekul reprezentujacych  $(c \wedge a) \rightarrow k$  oraz  $c$   
 Ryc. 61b: ligacja powyzszej molekuly oraz molekuly faktu  $a$   
 Ryc. 61c: ligacja powyzszej molekuly oraz molekuly wnioskowania  $(c \wedge a) \rightarrow k$   
 Ryc. 61d: ligacja powyzszej molekuly oraz molekuly faktu  $b$   
 Ryc. 61e: ligacja powyzszej molekuly oraz molekuly wnioskowania  $l \rightarrow m$   
 Ryc. 61f: ligacja powyzszej molekuly oraz molekuly reprezentujacej hipoteze  $m$   
 Ryc. 62: przykladowe drzewo decyzyjne  
 Ryc. 63a: schemat molekuly galazi drzewa, reprezentujacej fakt  $p$   
 Ryc. 63b: schemat molekuly galazi drzewa, reprezentujacej fakt  $\neg p$   
 Ryc. 63c: schemat molekuly galazi drzewa, reprezentujacej fakt  $q$   
 Ryc. 63d: schemat molekuly galazi drzewa, reprezentujacej fakt  $\neg q$   
 Ryc. 63e: schemat molekuly galazi drzewa, reprezentujacej fakt  $r$   
 Ryc. 63f: schemat molekuly galazi drzewa, reprezentujacej fakt  $\neg r$   
 Ryc. 63g: schemat molekuly korzenia  $p$ ?  
 Ryc. 63h: schemat molekuly wozla posredniego, reprezentujacego  $q$ ?  
 Ryc. 63i: schemat molekuly wozla posredniego, reprezentujacego  $r$ ?  
 Ryc. 63j: schemat molekuly liścia, reprezentujacego wniosek  $s$   
 Ryc. 63k: schemat molekuly liścia, reprezentujacego wniosek  $t$   
 Ryc. 63l: schemat molekuly liścia, reprezentujacego wniosek  $u$   
 Ryc. 63m: schemat molekuly liścia, reprezentujacego wniosek  $v$   
 Ryc. 64a: ligacja molekul reprezentujacych korzeń  $p$ ? oraz galazi  $p$   
 Ryc. 64b: ligacja powyzszej molekuly oraz molekuly reprezentujacej wozel posredni  $q$ ?  
 Ryc. 64c: ligacja powyzszej molekuly oraz molekuly reprezentujacej galaz  $q$   
 Ryc. 64d: ligacja powyzszej molekuly oraz molekuly reprezentujacej liśc  $s$   
 Ryc. 65a: ligacja molekul reprezentujacych korzeń  $p$ ? oraz galazi  $p$   
 Ryc. 65b: ligacja powyzszej molekuly oraz molekuly reprezentujacej wozel posredni  $r$ ?  
 Ryc. 65c: ligacja powyzszej molekuly oraz molekuly reprezentujacej wozel posredni  $r$ ?  
 Ryc. 65d: ligacja powyzszej molekuly oraz molekuly reprezentujacej liśc  $v$   
 Ryc. 66: przykladowa molekula reprezentujaca pojedyncza regule wnioskowania  
 Ryc. 67: przykladowe drzewo regul wnioskowania  
 Ryc. 68a: molekula reprezentujaca regule wnioskowania  $R1$  czyli  $(a \rightarrow b)$



- Ryc. 68b: molekula reprezentująca regułę wnioskowania R2 czyli ( $b \rightarrow c$ )  
 Ryc. 68c: molekula reprezentująca regułę wnioskowania R3 czyli ( $a \rightarrow d$ )  
 Ryc. 68d: molekula reprezentująca regułę wnioskowania R4 czyli ( $c \rightarrow e$ )  
 Ryc. 68e: molekula reprezentująca regułę wnioskowania R5 czyli ( $c \rightarrow f$ )  
 Ryc. 69: molekula DBF reprezentująca zapytanie  $f$ ?  
 Ryc. 70a: ligacja molekuł reprezentujących reguły wnioskowania R1 oraz R2  
 Ryc. 70b: ligacja powyższej molekuly oraz molekuly reprezentującej regułę wnioskowania R5  
 Ryc. 70c: ligacja powyższej molekuly oraz molekuly DBF reprezentującej zapytanie  $f$ ?  
 Ryc. 70d: połączenie się ze sobą komplementarnych „lepkich końców” i domknięcie molekuly do postaci kolistej  
 Ryc. 71a: ligacja molekuł reprezentujących reguły wnioskowania R1 oraz R3  
 Ryc. 71b: ligacja powstałej wcześniej molekuly oraz molekuly reprezentującej regułę wnioskowania R4

### 3. 4. System In-Hee Lee, Ji-Yoon Park i innych. .... - 55 -

- Ryc. 72a: schemat molekuly reprezentującej literal  $a$   
 Ryc. 72b: schemat molekuly reprezentującej literal  $\neg a$   
 Ryc. 73: schemat molekuly reprezentującej klauzulę ( $\neg a \vee b \vee c$ )  
 Ryc. 74: schemat molekuly reprezentującej zapytanie  $a$ ?  
 Ryc. 75a: schemat molekuly reprezentującej ( $\neg q \vee \neg p \vee r$ )  
 Ryc. 75b: schemat molekuly reprezentującej ( $q \vee \neg t \vee \neg s$ )  
 Ryc. 75c: schemat molekuly reprezentującej  $s$   
 Ryc. 75d: schemat molekuly reprezentującej  $p$   
 Ryc. 75e: schemat molekuly reprezentującej  $t$   
 Ryc. 75f: schemat molekuly reprezentującej  $\neg r$   
 Ryc. 76a: ligacja molekuł reprezentujących ( $\neg q \vee \neg p \vee r$ ) oraz ( $q \vee \neg t \vee \neg s$ )  
 Ryc. 76b: przyłączenie się kolejnych molekuł (tutaj:  $t$ )  
 Ryc. 76c: przyłączenie się kolejnych molekuł (tutaj:  $s, p$  oraz  $\neg r$ )

### 4. Rozszerzony system wnioskowania..... - 59 -

#### 4.1. Wprowadzenie ..... - 59 -

- Ryc. 77a: ciąg nukleotydów niezbędny do rozpoznania przez enzym restrykcyjny BbvI  
 Ryc. 77b: rozcięcie za pomocą enzymu restrykcyjnego BbvI

#### 4.2. Kodowanie informacji..... - 59 -

- Ryc. 78: schemat molekuly reprezentującej klauzulę faktu  
 Ryc. 79a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 79b: molekula reprezentująca klauzulę faktu  $\neg p$   
 Ryc. 80: schemat molekuly reprezentującej dwuliteralową klauzulę  
 Ryc. 81: molekula reprezentująca klauzulę ( $p \vee \neg q$ )  
 Ryc. 82: zestawione obok siebie klauzule ( $p_1 \vee \neg p_2$ ) oraz ( $\neg p_2 \vee p_3$ )  
 Ryc. 83: molekula reprezentująca klauzulę ( $p_1 \vee \neg p_2 \vee p_3$ )  
 Ryc. 84a: molekula reprezentująca klauzulę ( $p_1 \vee \neg p_2 \vee p_3 \vee p_4$ )  
 Ryc. 84b: molekula reprezentująca klauzulę ( $p_1 \vee \neg p_2 \vee p_3 \vee p_4 \vee p_5 \vee \neg p_6$ )  
 Ryc. 85: molekula terminalna wnioskowania  
 Ryc. 86a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 86b: molekula reprezentująca klauzulę ( $\neg p \vee q$ )  
 Ryc. 86c: molekula terminalna wnioskowania  
 Ryc. 87a: ligacja molekuł reprezentujących klauzule  $p$  oraz ( $\neg p \vee q$ )  
 Ryc. 87b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 87c: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 88a: molekula reprezentująca klauzulę faktu  $\neg p$   
 Ryc. 88b: molekula reprezentująca klauzulę faktu  $r$   
 Ryc. 88c: molekula reprezentująca klauzulę ( $p \vee q \vee \neg r$ )  
 Ryc. 88d: molekula terminalna wnioskowania  
 Ryc. 89a: ligacja molekuł reprezentujących klauzule  $\neg p$  oraz ( $p \vee q \vee \neg r$ )  
 Ryc. 89b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 89c: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 89d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 89e: ligacja molekuly reprezentującej klauzulę  $r$  i powstałej, reprezentującej ( $q \vee \neg r$ )

Ryc. 89f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 89g: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 90: schemat molekuly reprezentującej zapytanie  
 Ryc. 91a: schemat molekuly reprezentującej zapytanie  $p$ ?  
 Ryc. 91b: schemat molekuly reprezentującej zapytanie  $\neg q$ ?  
 Ryc. 92a: ligacja molekuly reprezentującej  $p$  i molekuly zapytania  $p$ ?  
 Ryc. 92b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 92c: ligacja powstałych tych samych molekul ze sobą  
 Ryc. 93: schemat molekuly reprezentującej zapytanie  $(p \wedge \neg q \wedge r)$ ?  
 Ryc. 94: schemat molekul wspólnie reprezentujących zapytanie  $(p \vee \neg q \vee r)$ ?  
 Ryc. 95: schemat molekul wspólnie reprezentujących zapytanie  $(p \vee (q \wedge \neg r))$ ?  
 Ryc. 96a: molekula reprezentująca klauzulę faktu  $q$   
 Ryc. 96b: molekula reprezentująca klauzulę faktu  $\neg r$   
 Ryc. 96c: molekuly reprezentujące zapytanie  $(p \vee (q \wedge \neg r))$ ?  
 Ryc. 97a: ligacja molekul reprezentujących klauzule  $q$  oraz cząsteczkę  $(q \wedge \neg r)$ ?  
 Ryc. 97b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 97c: ligacja molekul reprezentujących klauzule  $\neg r$  oraz powstałą cząsteczkę  $(\neg r)$ ?  
 Ryc. 97d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 97e: ligacja powstałych tych samych molekul ze sobą

#### 4. 3. Wykrycie sprzeczności.....- 67 -

Ryc. 98a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 98b: molekula reprezentująca klauzulę faktu  $\neg p$   
 Ryc. 98c: molekula terminalna wniosowania  
 Ryc. 99a: ligacja molekul reprezentujących klauzule  $p$  oraz  $\neg p$   
 Ryc. 99b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 99c: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 99d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 99e: ligacja powstałych molekul ze sobą  
 Ryc. 100a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 100b: molekula reprezentująca klauzulę faktu  $\neg q$   
 Ryc. 100c: schemat molekuly reprezentującej klauzulę  $(\neg p \vee q)$   
 Ryc. 100d: molekula terminalna wniosowania  
 Ryc. 101a: ligacja molekul reprezentujących klauzule  $p$  oraz  $\neg q$  oraz  $(\neg p \vee q)$   
 Ryc. 101b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 101c: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 101d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 101e: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 101f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 101g: ligacja powstałych molekul ze sobą

#### 4. 5. Przykłady złożonego wniosowania .....- 70 -

Ryc. 102a: molekula reprezentująca klauzulę faktu  $a$   
 Ryc. 102b: molekula reprezentująca klauzulę faktu  $d$   
 Ryc. 102c: molekula reprezentująca klauzulę faktu  $u$   
 Ryc. 102d: molekula reprezentująca klauzulę  $(\neg a \vee \neg b \vee c)$   
 Ryc. 102e: molekula reprezentująca klauzulę  $(\neg d \vee c)$   
 Ryc. 102f: molekula reprezentująca klauzulę  $(\neg c \vee \neg u \vee x)$   
 Ryc. 102g: molekula terminalna wniosowania  
 Ryc. 102h: molekula reprezentująca zapytanie  $x$ ?  
 Ryc. 103a: ligacja molekul reprezentujących klauzule  $a$  oraz  $(\neg a \vee \neg b \vee u)$   
 Ryc. 103b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 103c: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 103d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 103e: ligacja molekul reprezentujących klauzule  $d$  oraz  $(\neg d \vee c)$   
 Ryc. 103f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 103g: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 103h: ligacja molekul reprezentujących klauzule  $c$  oraz  $(\neg c \vee \neg u \vee x)$   
 Ryc. 103i: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 103j: ligacja molekuly terminalnej i wniosku częściowego

Ryc. 103k: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 103l: ligacja molekul reprezentujących klauzule u oraz  $(\neg u \vee x)$   
 Ryc. 103m: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 103n: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 103o: ligacja molekuly reprezentującej x i molekuly zapytania x?  
 Ryc. 103p: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 103q: ligacja powstałych molekul ze sobą  
 Ryc. 104a: molekula reprezentująca klauzulę faktu  $\neg c$   
 Ryc. 104b: molekula reprezentująca klauzulę faktu  $\neg d$   
 Ryc. 104c: molekula reprezentująca klauzulę  $(a \vee \neg b)$   
 Ryc. 104d: molekula reprezentująca klauzulę  $(b \vee c \vee d)$   
 Ryc. 104e: molekula reprezentująca klauzulę  $(a \vee d)$   
 Ryc. 104f: molekula terminalna wnioskowania  
 Ryc. 104g: molekula reprezentująca zapytanie b?  
 Ryc. 105a: ligacja molekul reprezentujących klauzule  $\neg d$  oraz  $(b \vee c \vee d)$   
 Ryc. 105b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105c: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 105d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105e: ligacja molekul reprezentujących klauzule  $\neg d$  oraz  $(a \vee d)$   
 Ryc. 105f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105g: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 105h: ligacja molekul reprezentujących klauzule  $(a \vee \neg b)$  oraz  $(b \vee c \vee d)$   
 Ryc. 105i: ligacja molekuly reprezentującej klauzulę faktu a oraz powyższej molekuly  
 Ryc. 105j: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105k: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 105l: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105m: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 105n: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105o: ligacja molekul reprezentujących klauzule  $\neg d$  oraz  $(c \vee d)$   
 Ryc. 105p: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105q: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 105r: ligacja molekul reprezentujących klauzule c oraz  $\neg c$   
 Ryc. 105s: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105t: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 105u: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 105v: ligacja powstałych molekul ze sobą

#### 4. 6. Odniesienie do klasycznego rachunku zdań.....- 74 -

Ryc. 106a: molekula reprezentująca klauzulę faktu p  
 Ryc. 106b: molekula reprezentująca klauzulę faktu  $\neg q$   
 Ryc. 106c: molekula reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 106d: molekula terminalna wnioskowania  
 Ryc. 107a: ligacja molekul reprezentujących klauzule p oraz  $(\neg p \vee q)$   
 Ryc. 107b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 107c: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 108a: ligacja molekul reprezentujących klauzule  $\neg q$  oraz  $(\neg p \vee q)$   
 Ryc. 108b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 108c: ligacja molekuly terminalnej i wniosku częściowego (zapisana już po „odwróceniu”)  
 Ryc. 109a: schemat molekuly reprezentującej klauzulę  $(\neg p \vee q)$   
 Ryc. 109b: schemat molekuly reprezentującej klauzulę  $(\neg q \vee r)$   
 Ryc. 110: schemat molekuly reprezentującej klauzulę  $(\neg p \vee r)$   
 Ryc. 111a: molekula reprezentująca klauzulę faktu p  
 Ryc. 111b: molekula reprezentująca klauzulę  $(\neg p \vee r)$  (po uwzględnieniu przechodniości)  
 Ryc. 111c: molekula terminalna wnioskowania  
 Ryc. 112a: ligacja molekul reprezentujących klauzulę faktu p oraz klauzulę  $(\neg p \vee r)$   
 Ryc. 112b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 112c: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 112d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 112e: ligacja molekuly terminalnej i powstałej cząsteczki

## 5. Analiza rozszerzonego systemu ..... - 79 -

### 5. 1. Ograniczenia systemu ..... - 79 -

- Ryc. 113: fragment rozpoznawalny przez enzym BbvI  
Ryc. 114: schemat kodowania dowolnej molekuly faktu  
Ryc. 115: schemat molekul reprezentujacych zapytanie  $p$ ?  
Ryc. 116a: ligacja molekuly reprezentujacej  $p$  i molekuly zapytania  
Ryc. 116b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
Ryc. 116c: ligacja powstalej molekuly i czwórki uzupełniajacej zapytanie  
Ryc. 117a: schemat molekuly reprezentujacej zapytanie  $p$ ?  
Ryc. 117b: schemat molekuly reprezentujacej zapytanie  $\neg p$ ?  
Ryc. 118: ligacja cząsteczek reprezentujacych zapytania  $p$ ? oraz  $\neg p$ ?

### 5. 2. Dodatkowe zastosowania ..... - 83 -

- Ryc. 119: molekuly wspólnie reprezentujace funkcję logiczną  $((p \wedge q \wedge r) \vee \neg s)$   
Ryc. 120a: molekula reprezentujaca argument  $p$   
Ryc. 120b: molekula reprezentujaca argument  $q$   
Ryc. 120c: molekula reprezentujaca argument  $\neg r$   
Ryc. 120d: molekula reprezentujaca argument  $\neg s$   
Ryc. 121a: ligacja molekuly  $p$  oraz molekuly koniunkcji  $(p \wedge q \wedge r)$   
Ryc. 121b: rozcięcie za pomocą enzymu restrykcyjnego BbvI  
Ryc. 121c: ligacja molekuly  $q$  oraz powstalej reszty molekuly koniunkcji  $(p \wedge q \wedge r)$   
Ryc. 121d: rozcięcie za pomocą enzymu restrykcyjnego BbvI  
Ryc. 122a: ligacja molekuly  $p$  oraz molekuly koniunkcji  $(p \wedge q \wedge r)$   
Ryc. 122b: rozcięcie za pomocą enzymu restrykcyjnego BbvI  
Ryc. 122c: ligacja powstalych molekul ze sobą  
Ryc. 123: molekuly reprezentujace funkcję logiczną XOR czyli  $((a \wedge \neg b) \vee (\neg a \wedge b))$   
Ryc. 124a: molekula reprezentujaca argument  $\neg a$   
Ryc. 124b: molekula reprezentujaca argument  $b$   
Ryc. 125a: ligacja molekuly  $\neg a$  oraz molekuly koniunkcji  $(\neg a \wedge b)$   
Ryc. 125b: rozcięcie za pomocą enzymu restrykcyjnego BbvI  
Ryc. 125c: ligacja molekuly  $b$  oraz powstalej reszty molekuly koniunkcji  $(\neg a \wedge b)$   
Ryc. 125d: ligacja molekuly  $b$  oraz powstalej reszty molekuly koniunkcji  $(\neg a \wedge b)$   
Ryc. 125e: ligacja powstalych molekul ze sobą  
Ryc. 126a: molekula reprezentujaca argument  $\neg a$   
Ryc. 126b: molekula reprezentujaca argument  $\neg b$   
Ryc. 127a: ligacja molekuly  $\neg a$  oraz molekuly koniunkcji  $(\neg a \wedge b)$   
Ryc. 127b: rozcięcie za pomocą enzymu restrykcyjnego BbvI  
Ryc. 128a: molekula reprezentujaca klauzulę faktu  $a$   
Ryc. 128b: molekula reprezentujaca klauzulę faktu  $b$   
Ryc. 128c: molekula reprezentujaca klauzulę  $(\neg a \vee \neg b)$   
Ryc. 128d: molekula terminalna wnioskowania  
Ryc. 129a: ligacja molekul reprezentujacych klauzule  $a$  oraz  $(\neg a \vee \neg b)$   
Ryc. 129b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
Ryc. 129c: ligacja molekuly terminalnej i powstalej cząsteczki  
Ryc. 129d: ligacja molekul reprezentujacych klauzule  $b$  oraz  $\neg b$  (powstalej w poprzednim kroku)  
Ryc. 129e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
Ryc. 129f: ligacja molekuly terminalnej i powstalej cząsteczki  
Ryc. 129g: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
Ryc. 129h: ligacja powstalych molekul ze sobą  
Ryc. 130a: molekula reprezentujaca klauzulę faktu  $a$   
Ryc. 130b: molekula reprezentujaca klauzulę  $(\neg a \vee \neg b)$   
Ryc. 130c: molekula terminalna wnioskowania  
Ryc. 131a: ligacja molekul reprezentujacych klauzule  $a$  oraz  $(\neg a \vee \neg b)$   
Ryc. 131b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
Ryc. 131c: ligacja molekuly terminalnej i powstalej cząsteczki

- Ryc. 132a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 132b: molekula reprezentująca klauzulę faktu  $\neg p$   
 Ryc. 132c: molekula terminalna wnioskowania  
 Ryc. 133a: ligacja molekuł reprezentujących klauzule  $p$  oraz  $\neg p$   
 Ryc. 133b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 133c: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 133d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 133e: ligacja powstałych molekuł ze sobą  
 Ryc. 134a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 134b: molekula reprezentująca klauzulę faktu  $\neg p$   
 Ryc. 134c: molekula reprezentująca klauzulę faktu  $\neg q$   
 Ryc. 134d: molekula terminalna wnioskowania  
 Ryc. 135a: molekula reprezentująca klauzulę faktu  $\neg q$   
 Ryc. 135b: molekula reprezentująca klauzulę  $(p \vee q)$   
 Ryc. 135c: molekula reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 135d: molekula terminalna wnioskowania  
 Ryc. 136a: ligacja molekuł reprezentujących klauzule  $(p \vee q)$  oraz  $(\neg p \vee q)$   
 Ryc. 136b: ligacja molekuł reprezentujących klauzule  $\neg q$  oraz  $(q \vee q)$   
 Ryc. 136c: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 136d: ligacja molekuly terminalnej i nowopowstałej cząsteczki  
 Ryc. 136e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 136f: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $q$ )  
 Ryc. 136g: ligacja molekuł reprezentujących klauzule  $q$  oraz  $\neg q$   
 Ryc. 136h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 136i: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 136j: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 136k: ligacja powstałych molekuł ze sobą  
 Ryc. 137a: molekula reprezentująca klauzulę faktu  $\neg r$   
 Ryc. 137b: molekula reprezentująca klauzulę  $(p \vee q)$   
 Ryc. 137c: molekula reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 137d: molekula reprezentująca klauzulę  $(\neg q \vee r)$   
 Ryc. 137e: molekula terminalna wnioskowania  
 Ryc. 138a: ligacja molekuł reprezentujących klauzule  $(p \vee q)$  oraz  $(\neg p \vee q)$   
 Ryc. 138b: ligacja molekuł reprezentujących klauzule  $\neg r$  oraz  $(\neg q \vee r)$   
 Ryc. 138c: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 138d: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg q$ )  
 Ryc. 138e: ligacja molekuł reprezentujących klauzule  $\neg q$  oraz  $(q \vee q)$   
 Ryc. 138f: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 138g: ligacja molekuly terminalnej i nowopowstałej cząsteczki  
 Ryc. 138h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 138i: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $q$ )  
 Ryc. 139a: ligacja molekuł reprezentujących klauzule  $q$  oraz  $\neg q$   
 Ryc. 139b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 139c: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 139d: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 139e: ligacja powstałych molekuł ze sobą  
 Ryc. 140a: ligacja molekuł reprezentujących klauzule  $q$  oraz  $(\neg q \vee r)$   
 Ryc. 140b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 140c: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $r$ )  
 Ryc. 140d: ligacja molekuł reprezentujących klauzule  $q$  oraz  $\neg q$   
 Ryc. 140e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 140f: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 140g: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 140h: ligacja powstałych molekuł ze sobą  
 Ryc. 141a: molekula reprezentująca klauzulę faktu  $\neg p$   
 Ryc. 141b: molekula reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 141c: molekula reprezentująca klauzulę  $(\neg q \vee r)$   
 Ryc. 141d: molekula reprezentująca klauzulę  $(\neg r \vee p)$   
 Ryc. 141e: molekula terminalna wnioskowania  
 Ryc. 142a: ligacja molekuł reprezentujących klauzule  $\neg p$  oraz  $(\neg r \vee p)$

Ryc. 142b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 142c: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg r$ )  
 Ryc. 142d: ligacja molekul reprezentujących klauzule  $\neg r$  oraz  $(\neg q \vee r)$   
 Ryc. 142e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 142f: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg q$ )  
 Ryc. 142g: ligacja molekul reprezentujących klauzule  $\neg q$  oraz  $(\neg p \vee q)$   
 Ryc. 142h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 142i: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg p$ )  
 Ryc. 143a: molekula reprezentująca klauzulę faktu  $\neg q$   
 Ryc. 143b: molekula reprezentująca klauzulę faktu  $r$   
 Ryc. 143c: molekula reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 143d: molekula reprezentująca klauzulę  $(\neg q \vee r)$   
 Ryc. 143e: molekula reprezentująca klauzulę  $(\neg r \vee p)$   
 Ryc. 143f: molekula terminalna wnioskowania  
 Ryc. 144a: ligacja molekul reprezentujących klauzule  $\neg q$  oraz  $(\neg p \vee q)$   
 Ryc. 144b: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 144c: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg p$ )  
 Ryc. 144d: ligacja molekul reprezentujących klauzule  $\neg p$  oraz  $(\neg r \vee p)$   
 Ryc. 144e: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 144f: ligacja molekuly terminalnej i wniosku częściowego (powstanie klauzuli  $\neg r$ )  
 Ryc. 144g: ligacja molekul reprezentujących klauzule  $r$  oraz  $\neg r$   
 Ryc. 144h: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 144i: ligacja molekuly terminalnej i powstałej cząsteczki  
 Ryc. 144j: rozcięcie molekuly za pomocą enzymu restrykcyjnego BbvI  
 Ryc. 144k: ligacja powstałych molekul ze sobą  
 Ryc. 145a: uproszczony schemat molekuly reprezentującej klauzulę faktu  
 Ryc. 145b: uproszczony schemat molekuly reprezentującej klauzulę złożoną dwuelementową  
 Ryc. 145c: uproszczony schemat molekuly terminalnej wnioskowania  
 Ryc. 145d: uproszczony schemat molekuly reprezentującej zapytanie  
 Ryc. 146a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 146b: molekula reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 146c: molekula terminalna wnioskowania  
 Ryc. 146d: molekula zapytania  $\neg q$ ?  
 Ryc. 147a: ligacja molekul reprezentujących klauzule  $p$  oraz  $(\neg p \vee q)$   
 Ryc. 147b: rozcięcie molekuly za pomocą enzymu restrykcyjnego  
 Ryc. 147c: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 147d: ligacja powstałej molekuly reprezentującej  $q$  i molekuly zapytania  
 Ryc. 147e: rozcięcie molekuly za pomocą enzymu restrykcyjnego  
 Ryc. 147f: ligacja powstałych molekul ze sobą  
 Ryc. 148a: molekula zapytania  $p$ ?  
 Ryc. 148b: molekula zapytania  $\neg q$ ?  
 Ryc. 149a: molekula reprezentująca klauzulę faktu  $p$   
 Ryc. 149b: molekula reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 149c: molekula terminalna wnioskowania  
 Ryc. 149d: molekula zapytania  $q$ ?  
 Ryc. 150a: ligacja molekul reprezentujących klauzule  $p$  oraz  $(\neg p \vee q)$   
 Ryc. 150b: rozcięcie molekuly za pomocą enzymu restrykcyjnego  
 Ryc. 150c: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 150d: ligacja powstałej molekuly reprezentującej  $q$  i molekuly zapytania  
 Ryc. 150e: rozcięcie molekuly za pomocą enzymu restrykcyjnego  
 Ryc. 150f: ligacja molekuly terminalnej i wniosku częściowego  
 Ryc. 150g: rozcięcie molekuly za pomocą enzymu restrykcyjnego  
 Ryc. 150h: rozpad powstałej cząstki na dwie jednoniciowe oraz uaktywnienie fluorescencji

## **6. Symulacja eksperymentów laboratoryjnych.....- 104 -**

### **6. 1. Wybór eksperymentów.....- 105 -**

Ryc. 151a: sekwencja reprezentująca klauzulę faktu  $\neg q$   
 Ryc. 151b: sekwencja reprezentująca klauzulę  $(\neg p \vee q)$   
 Ryc. 151c: sekwencja terminalna wnioskowania  
 Ryc. 151d: sekwencja reprezentująca zapytanie  $\neg p$ ?

- Ryc. 152a: sekwencja reprezentująca klauzulę faktu  $p$
- Ryc. 152b: sekwencja reprezentująca klauzulę faktu  $\neg q$
- Ryc. 152c: sekwencja reprezentująca klauzulę  $(\neg p \vee q)$
- Ryc. 152d: sekwencja terminalna wnioskowania
- Ryc. 153a: sekwencja reprezentująca klauzulę faktu  $a$
- Ryc. 153b: sekwencja reprezentująca klauzulę faktu  $\neg d$
- Ryc. 153c: sekwencja reprezentująca klauzulę faktu  $\neg e$
- Ryc. 153d: sekwencja reprezentująca klauzulę  $(\neg a \vee \neg b \vee c \vee d)$
- Ryc. 153e: sekwencja reprezentująca klauzulę  $(b \vee e)$
- Ryc. 153f: sekwencja terminalna wnioskowania
- Ryc. 153g: sekwencja reprezentująca zapytanie  $\neg c?$

## 6. 2. Przygotowanie cząsteczek DNA w laboratorium.....- 107 -

- Ryc. 154: schemat wklonowania sekwencji DNA do wektora plazmidowego
- Ryc. 155: schemat sekwencji dopełniającej każdą z cząsteczek do postaci dwuniciowej o „tępych końcach”
- Ryc. 156: schemat sekwencji reprezentującej klauzulę faktu po dopełnieniu do postaci dwuniciowej o „tępych końcach”
- Ryc. 157: schemat sekwencji reprezentującej klauzulę złożoną po dopełnieniu do postaci dwuniciowej o „tępych końcach”
- Ryc. 158: schemat sekwencji reprezentującej zapytanie po dopełnieniu do postaci dwuniciowej o „tępych końcach”
- Ryc. 159: sekwencja terminalna wnioskowania po dopełnieniu do postaci dwuniciowej o „tępych końcach”
- Ryc. 160: dołączane fragmenty wypełniacza, mające umożliwić jednoznaczne wklonowanie do wektora
- Ryc. 161: schemat sekwencji reprezentującej klauzulę faktu po dołączeniu wypełniacza
- Ryc. 162: schemat sekwencji reprezentującej klauzulę złożoną po dołączeniu wypełniacza
- Ryc. 163: schemat sekwencji reprezentującej zapytanie po dołączeniu wypełniacza
- Ryc. 164: sekwencja terminalna wnioskowania po dołączeniu wypełniacza
- Ryc. 165a: początkowa sekwencja wypełniacza
- Ryc. 165b: pierwsza sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 165c: sekwencja reprezentująca klauzulę faktu  $\neg q$
- Ryc. 165d: końcowa sekwencja wypełniacza
- Ryc. 166: cząsteczka reprezentująca klauzulę faktu  $\neg q$
- Ryc. 167a: początkowa sekwencja wypełniacza
- Ryc. 167b: : sekwencja reprezentująca klauzulę faktu  $(\neg p \vee q)$
- Ryc. 167c: końcowa sekwencja wypełniacza
- Ryc. 168: cząsteczka reprezentująca klauzulę złożoną  $(\neg p \vee q)$
- Ryc. 169a: początkowa sekwencja wypełniacza
- Ryc. 169b: sekwencja reprezentująca zapytanie  $p?$
- Ryc. 169c: druga sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 169d: trzecia sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 169e: czwarta sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 169f: piąta sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 169g: szósta sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 169h: siódma sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 169i: końcowa sekwencja wypełniacza
- Ryc. 170: cząsteczka reprezentująca zapytanie  $p?$
- Ryc. 171a: początkowa sekwencja wypełniacza
- Ryc. 171b: pierwsza sekwencja uzupełniająca niezdefiniowane fragmenty
- Ryc. 171c: sekwencja reprezentująca cząsteczkę terminalną
- Ryc. 171d: końcowa sekwencja wypełniacza
- Ryc. 172: cząsteczka terminalna wnioskowania
- Ryc. 173: Mapa restrykcyjna wektora plazmidowego pJET 1.2 z zaznaczonymi elementami genetycznymi tego wektora
- Ryc. 174a: sekwencja nukleotydów rozpoznawalna przez enzym MfeI
- Ryc. 174b: efekt trawienia powyższej sekwencji przez enzym MfeI
- Ryc. 175a: sekwencja nukleotydów rozpoznawalna przez enzym PspFI
- Ryc. 175b: efekt trawienia powyższej sekwencji przez enzym PspFI
- Ryc. 176: uproszczony schemat wektora plazmidowego pJET 1.2, uwzględniający sekwencje rozpoznawalne przez MfeI oraz PspFI
- Ryc. 177: uproszczony schemat pozostałego fragmentu wektora plazmidowego pJET 1.2
- Ryc. 178: schemat tworzenia „biblioteki” łańcuchów DNA za pomocą wektorów plazmidowych
- Ryc. 179: fragment wektora plazmidowego pJET 1.2, uwzględniający sekwencje rozpoznawalne przez MfeI oraz PspFI
- Ryc. 180: fragment wektora plazmidowego pJET 1.2, zawierający schemat dowolnej sekwencji reprezentującej element systemu
- Ryc. 181a: sekwencja reprezentująca STARTER (komplementarny do końca 3' pierwszej nici oczekiwanego łańcucha)
- Ryc. 181b: sekwencja reprezentująca STARTER (komplementarny do końca 3' drugiej nici oczekiwanego łańcucha)
- Ryc. 182: schemat pierwszego kroku PCR dla dowolnej sekwencji reprezentującej element systemu i omówionych „starterów”
- Ryc. 183a: wektor plazmidowy z wbudowanym DNA reprezentującym element systemu

Ryc. 183b: rozplecenie helisy do postaci dwóch oligonukleotydów (denaturacja)  
 Ryc. 183c: przyłączenie się krótkich fragmentów komplementarnych („starterów”)  
 Ryc. 183d: dobudowanie komplementarnych nukleotydów w kierunku od 5' do 3'  
 Ryc. 184: fragmenty jednoniciowe po pierwszej fazie reakcji łańcuchowej polimerazy (PCR)  
 Ryc. 185a: przyłączenie się kolejnych krótkich fragmentów komplementarnych („starterów”)  
 Ryc. 185b: dobudowanie komplementarnych nukleotydów w kierunku od 5' do 3'  
 Ryc. 186a: fragmenty jednoniciowe po drugiej fazie reakcji łańcuchowej polimerazy (PCR)  
 Ryc. 186b: przyłączenie się kolejnych krótkich fragmentów komplementarnych („starterów”)  
 Ryc. 186c: dobudowanie komplementarnych nukleotydów w kierunku od 5' do 3'  
 Ryc. 187: schemat wszystkich cząsteczek reprezentujących składowe systemy wnioskujące, po selekcji z wektora plazmidowego  
 Ryc. 188a: schemat cząsteczki reprezentującej klauzulę faktu po selekcji z wektora plazmidowego  
 Ryc. 188b: schemat cząsteczki reprezentującej klauzulę złożoną po selekcji z wektora plazmidowego  
 Ryc. 188c: schemat cząsteczki reprezentującej zapytanie po selekcji z wektora plazmidowego  
 Ryc. 188d: cząsteczka terminalna wnioskowania po selekcji z wektora plazmidowego  
 Ryc. 189a: schemat cząsteczki repr. klauzulę faktu oraz dodatkowe fragmenty, w wyniku trawienia enzymem restrykcyjnym Eco32I  
 Ryc. 189b: schemat cząsteczki repr. klauzulę złożoną oraz dodatkowe fragmenty, w wyniku trawienia enzymem restrykcyjnym Eco32I  
 Ryc. 189c: schemat cząsteczki repr. zapytanie oraz dodatkowe fragmenty, w wyniku trawienia enzymem restrykcyjnym Eco32I  
 Ryc. 189d: cząsteczka terminalna oraz dodatkowe fragmenty, w wyniku trawienia enzymem restrykcyjnym Eco32I  
 Ryc. 190a: schemat cząsteczki repr. klauzulę faktu oraz dodatkowy fragment, w wyniku trawienia enzymem restrykcyjnym BsmAI  
 Ryc. 190b: schemat cząsteczki repr. klauzulę złożoną oraz dodatkowe fragmenty, w wyniku trawienia enzymem restrykcyjnym BsmAI  
 Ryc. 190c: schemat cząsteczki repr. zapytanie oraz dodatkowy fragment, w wyniku trawienia enzymem restrykcyjnym BsmAI  
 Ryc. 190d: cząsteczka terminalna oraz dodatkowy fragment, w wyniku trawienia enzymem restrykcyjnym BsmAI

### 6. 3. Reakcja wnioskowania w laboratorium .....- 126 -

Ryc. 191a: cząsteczka reprezentująca klauzulę faktu  $\rightarrow q$   
 Ryc. 191b: cząsteczka reprezentująca klauzulę  $(\rightarrow p \vee q)$   
 Ryc. 191c: cząsteczka terminalna wnioskowania  
 Ryc. 191d: cząsteczka reprezentująca zapytanie  $\rightarrow p?$   
 Ryc. 192a: ligacja cząsteczek reprezentujących klauzulę  $\rightarrow q$  oraz  $(\rightarrow p \vee q)$   
 Ryc. 192b: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
 Ryc. 192c: ligacja cząsteczki terminalnej i wniosku częściowego; powstała cząsteczka reprezentuje  $\rightarrow p$   
 Ryc. 192d: ligacja cząsteczki reprezentującej  $\rightarrow p$  i cząsteczki zapytania  $\rightarrow p?$   
 Ryc. 192e: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
 Ryc. 192f: ligacja nowopowstałych identycznych cząsteczek  
 Ryc. 193a: cząsteczka reprezentująca klauzulę faktu p  
 Ryc. 193b: cząsteczka reprezentująca klauzulę faktu  $\rightarrow q$   
 Ryc. 193c: cząsteczka reprezentująca klauzulę  $(\rightarrow p \vee q)$   
 Ryc. 193d: cząsteczka terminalna wnioskowania  
 Ryc. 194a: ligacja cząsteczek reprezentujących klauzulę  $\rightarrow q$  oraz  $(\rightarrow p \vee q)$   
 Ryc. 194b: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
 Ryc. 194c: ligacja cząsteczki terminalnej i wniosku częściowego; powstała cząsteczka reprezentuje  $\rightarrow p$   
 Ryc. 194d: ligacja cząsteczek reprezentujących klauzulę p oraz  $\rightarrow p$   
 Ryc. 194e: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
 Ryc. 194f: ligacja cząsteczki terminalnej i nowopowstałej  
 Ryc. 194g: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
 Ryc. 194h: ligacja nowopowstałych identycznych cząsteczek  
 Ryc. 195a: cząsteczka reprezentująca klauzulę faktu a  
 Ryc. 195b: cząsteczka reprezentująca klauzulę faktu  $\rightarrow d$   
 Ryc. 195c: cząsteczka reprezentująca klauzulę faktu  $\rightarrow e$   
 Ryc. 195d: cząsteczka reprezentująca klauzulę  $(\rightarrow a \vee \rightarrow b \vee c \vee d)$   
 Ryc. 195e: cząsteczka reprezentująca klauzulę  $(b \vee e)$   
 Ryc. 195f: cząsteczka terminalna wnioskowania  
 Ryc. 195g: cząsteczka reprezentująca zapytanie  $\rightarrow c?$   
 Ryc. 196a: ligacja cząsteczek reprezentujących klauzulę a oraz  $(\rightarrow a \vee \rightarrow b \vee c \vee d)$   
 Ryc. 196b: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
 Ryc. 196c: ligacja cząsteczki terminalnej i powyższego wniosku częściowego  
 Ryc. 196d: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI; powstała cząsteczka reprezentuje  $(\rightarrow b \vee c \vee d)$   
 Ryc. 196e: ligacja cząsteczek reprezentujących klauzulę  $\rightarrow d$  oraz  $(\rightarrow b \vee c \vee d)$   
 Ryc. 196f: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
 Ryc. 196g: ligacja cząsteczki terminalnej i powyższego wniosku częściowego  
 Ryc. 196h: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI; powstała cząsteczka reprezentuje  $(\rightarrow b \vee c)$   
 Ryc. 196i: ligacja cząsteczek reprezentujących klauzulę  $(b \vee e)$  oraz  $(\rightarrow b \vee c)$ ; powstała cząsteczka reprezentuje  $(c \vee e)$



Ryc. 196j: ligacja cząsteczek reprezentujących  $\neg e$  oraz  $(c \vee e)$   
Ryc. 196k: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
Ryc. 196l: ligacja cząsteczki terminalnej i powyższego wniosku częściowego  
Ryc. 196m: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
Ryc. 196n: ligacja cząsteczki terminalnej i powyższego wniosku częściowego; powstała cząsteczka reprezentuje  $c$   
Ryc. 196o: ligacja cząsteczki faktu  $c$  i cząsteczki zapytania  $c$ ?  
Ryc. 196p: trawienie powyższego fragmentu enzymem restrykcyjnym BbvI  
Ryc. 196q: ligacja nowopowstałych identycznych cząsteczek  
Ryc. 197: starter biorący udział w reakcji replikacji cząsteczki reprezentującej odpowiedź systemu  
Ryc. 198: zakończona reakcja łańcuchowa polimerazy – uzyskujemy dwie cząsteczki reprezentujące informację o sprzeczności  
Ryc. 199: zakończona reakcja łańcuchowa polimerazy – uzyskujemy dwie cząsteczki reprezentujące odpowiedź twierdzącą

## Spis tabel

**5. Analiza rozszerzonego systemu .....- 79 -**

**5. 2. Dodatkowe zastosowania .....- 83 -**

*Tab. 1: wartościowania przykładowej funkcji f*

**6. Symulacja eksperymentów laboratoryjnych.....- 104 -**

**6. 2. Przygotowanie cząsteczek DNA w laboratorium.....- 107 -**

*Tab. 2: sekwencje oligonukleotydów zamawiane w firmie komercyjnej dla pierwszego eksperymentu*

*Tab. 3: sekwencje oligonukleotydów zamawiane w firmie komercyjnej dla drugiego eksperymentu*

*Tab. 4: sekwencje oligonukleotydów zamawiane w firmie komercyjnej dla trzeciego eksperymentu*