1. (6 punktów) Niedeterministyczny automat skończony nazwiemy jednoznacznym, jeśli dla każdego akceptowanego słowa istnieje dokładnie jeden bieg akceptujący. Napisać algorytm sprawdzający, czy niedeterministyczny automat skończony jest jednoznaczny.

Odpowiedź. Niech $\mathcal{A}$ automat, którego jednoznaczność chcemy sprawdzić. Dla zbiorów P, R stanów, oznaczmy przez $L_{P,R}$ język słów, dla których automat ma bieg, który zaczyna się w stanie ze zbioru P, a kończy w stanie ze zbioru R. Załóżmy, że automat *nie jest* jednoznaczny. Wówczas istnieje słowo z dwoma różnymi biegami akceptującymi. Załóżmy, że na jednej pozycji pierwszy bieg ma stan p, a drugi bieg ma stan q. Wówczas widzimy, przy oznaczeniu przez I, F stanów początkowych i akceptujących, że zachodzi

$$L_{I,\{p\}} \cap L_{I,\{p\}} \neq \emptyset \qquad \text{i} \qquad L_{\{p\},F} \cap L_{\{q\},F} \neq \emptyset$$

Powyższy warunek też jest wystarczający dla jednoznaczności, a więc jest jej równoważny. Warunek można sprawdzić w czasie wielomianowym: dla każdej pary różnych stanów sprawdzamy dwa razy niepustość przecięcia dwóch języków. (Można napisać szybszy algorytm.)

2. (6 punktów) Rozważmy język zawierający słowa postaci $x = y + z$, gdzie $x, y, z \in \{0,1\}^*$ są takie, że rozwinięcie dwójkowe x jest sumą rozwinięć dwójkowych y i z. Na przykład do języka należy słowo

$$10110 = 101 + 00010.$$

Czy język ten jest bezkontekstowy?

Odpowiedź. Język nie jest bezkontekstowy. Idea jest następująca: w szczególnym przypadku sumy $x = y + 0$, poprawność sumy jest tym samym co sprawdzenie równości dwóch słów x, y, co nie jest własnością bezkontekstową.

Oto dowód. Rozważmy przecięcie języka z zadania z regularnym wzorcem.

$$10^*10^* = 10^*10^* + 0.$$

Gdyby język z zadania był bezkontekstowy, to i owo przecięcie byłoby bezkontekstowe, a pokażemy, że nie jest. Przecięcie to jest językiem

$$L = \{10^n 10^m = 10^n 10^m + 0 : n, m \in \mathbb{N}\}.$$

Zupełnie standardowe zastosowanie lematu o pompowaniu pokaże, że język L nie jest bezkontekstowy. Załóżmy więc, że L jest bezkontekstowy, a M to stała z lematu o pompowaniu. Wówczas można pompować słowo

$$w = 10^M 10^M = 10^M 10^M + 0.$$

Zgodnie z lematem o pompowaniu, istnieje podział

$$w = uxyzv \qquad \text{taki, że dla każdego } i \in \mathbb{N} \text{ zachodzi } ux^i yz^i v \in L$$

przy czym jedno ze słów x, z jest niepuste, a słowo xyz ma długość co najwyżej M. Gdyby któreś z pompowanych słów x, z zawierało jedynkę, to po jednokrotnym zapompowaniu zaburzony zostaje wzorzec regularny. A więc oba pompowane słowa zawierają wyłącznie zera. Ponieważ słowo xyz jest długości co najwyżej M, to pompowane są zera z jednego bloku, ewentualnie z dwóch kolejnych bloków (spośród czterech bloków zer w słowie w). A więc po zapompowaniu słowo wypada z L, która to sprzeczność dowodzi, że L nie jest bezkontekstowy.

3. (6 punktów) Pokazać, że nie jest całkowicie rozstrzygalny następujący problem: dany niedeterministyczny automat ze stosem, pytanie czy akceptuje pewne słowo postaci w^n, gdzie w jest dowolnym słowem, a $n \geqslant 2$. Można skorzystać z tego, że nie jest całkowicie rozstrzygalny problem: czy przecięcie dwóch języków bezkontekstowych jest niepuste.

Odpowiedź.

Do problemu z zadania zredukujemy następujący problem, który nie jest całkowicie rozstrzygalny: dane dwa języki bezkontekstowe, pytania czy mają niepuste przecięcie. Problem jest oczywiście nierozstrzygalny dla języków, których alfabet nie zawiera symbolu #, bo można zawsze zmienić nazwy liter. Na podstawie dwóch języków bezkontekstowych L i K tworzymy automat ze stosem, który rozpoznaje język $L\#K\#$. Język ten zawiera słowo postaci w^n wtedy i tylko wtedy, gdy zawiera słowo postaci $(w\#)^2$, wtedy i tylko wtedy gdy L i K mają niepuste przecięcie.

4. (6 punktów *) Rozważmy nastepujący wariant problemu QBF. Pytamy o prawdziwość kwantyfikowanej formuły boole-owskiej

$$Q_1 x_1 \cdots Q_n x_n \ \varphi(x_1, \ldots, x_n) \qquad Q_1, \ldots, Q_n \in \{\forall, \exists\}$$

gdzie zmienne $x_1, \ldots, x_n$ mają wartości ze zbioru {prawda, fałsz}, formuła φ nie zawiera kwantyfikatorów, a co najwyżej $\log n$ z kwantyfikatorów jest $\forall$. Czy problem ten jest PSPACE-zupełny, czy może tylko NP-zupełny?

Odpowiedź. Problem jest NP-zupełny. Jest NP-trudny, bo jest uogólnieniem problemu SAT. Pokażemy, że należy do NP. Rozważmy instancję problemu, a więc formułę jak w treści zadania. Załóżmy, że kwantyfikatory uniwersalne dotyczą zmiennych o numerach

$$1 \leqslant n_1 < \cdots < x_{n_k} \leqslant n$$

gdzie $k \leqslant \log n$. Zmienne kwantyfikowane egzystencjalnie dzielimy na k bloków $X_0, \ldots, X_k$, które są podzielone zmiennymi kwantyfikowanymi uniwersalnie:

$$X_0 = \{x_1, \ldots, x_{n_1-1}\} \quad X_1 = \{x_{n_1+1}, \ldots, x_{n_2-1}\} \quad \cdots \quad X_k = \{x_{n_k+1}, \ldots, x_n\}$$

Świadkiem prawdziwości formuły jest pełne drzewo o głębokości k, które wygląda w następujący sposób. (Głębokości wierzchołków liczymy tak, że korzeń ma głębokość 0, a liście mają głębokość k.) W wierzchołku na głębokości $i \in \{0, \ldots, k\}$ jest napisane wartościowanie dla zmiennych z bloku X_i. Idea jest taka, że dla wierzchołka na głębokości i, lewe poddrzewo odpowiada wartościowaniom gdzie x_{n_i} ma wartość 0, a prawe poddrzewo odpowiada wartościowaniom, gdzie zmienna x_{n_i} ma wartość 1. W związku z czym ze ścieżką od liścia do korzenia możemy stowarzyszyć wartościowanie wszystkich zmiennych: dla egzystencjalnych patrzymy na etykiety wierzchołków ze ścieżki, a dla uniwersalnych patrzymy na ciąg skrętów (lewo,prawo) na ścieżce.

Nietrudno zauważyć, że formuła jest prawdziwa wtedy i tylko wtedy gdy istnieje drzewo opisane powyżej, takie że w każdym liściu stowarzyszone wartościowanie spełnia formułę φ. Drzewo takie ma 2^k wierzchołków, a w każdym z nich jest zapisanych co najwyżej n bitów. Ponieważ $k \leqslant \log n$, to drzewo ma co najwyżej n wierzchołków, a więc zapisanie go zajmuje około n^2 bitów. A więc drzewo może być zgadnięte przez niedeterministyczną wielomianową maszynę Turinga.

Imię, nazwisko, nr indeksu *(kod)*

(9 punktów) Wybierz 9 z poniższych pytań i wybierz odpowiedź tak/nie (bez uzasadnienia). Za prawidłowe odpowiedzi dajemy +1 punkt, za złe -1 punkt. Punkty policzymy za 9 najgorszych odpowiedzi, czyli nie opłaca się odpowiadać na więcej niż 9 pytań.

1. Czy jest całkowicie rozstrzygalny problem: dany niedeterministyczny automat ze stosem, pytanie czy automat akceptuje wszystkie słowa długości parzystej?

   Odpowiedź. Nie, bo można do niego zredukować następujący problem, który nie jest całkowicie rozstrzygalny: czy automat akceptuje wszystkie słowa. Redukcja wygląda tak. Dla automatu ze stosem $\mathcal{A}$ nietrudno obliczyć automat $\mathcal{B}$, który akceptuje słowa

   $$\{a_1a_1a_2a_2\cdots a_na_n : a_1\cdots a_n \text{ jest akceptowane przez } \mathcal{A}\}.$$

   Automat $\mathcal{B}$ akceptuje wszystkie słowa długości parzystej wtedy i tylko wtedy gdy automat $\mathcal{A}$ akceptuje wszystkie słowa.

2. Czy jest całkowicie rozstrzygalny problem: dany niedeterministyczny automat ze stosem, pytanie czy automat akceptuje wyłącznie słowa długości parzystej?

   Odpowiedź. Problem jest całkowicie rozstrzygalny. Aby odpowiedzieć na to pytanie, przecinamy język z regularnym językiem słów nieparzystej długości. Przecięcie jest nowym automatem ze stosem, który można obliczyć, a potem sprawdzić czy akceptuje przynajmniej jedno słowo.

3. Czy dopełnienie języka, który nie jest całkowicie rozstrzygalny, może być bezkontekstowe?

   Odpowiedź. Nie. Każdy język bezkontekstowy jest całkowicie rozstrzygalny, a więc jego dopełnienie musi być całkowicie rozstrzygalne.

4. Czy równoważne (pod względem akceptowanych języków) są niedeterministyczne automaty ze stosem i niedeterministyczne automaty ze stosem, które mogą czytać wejście dwa razy (po przeczytaniu wejścia pierwszy raz, dostają separator # i potem wejście drugi raz)?

   Odpowiedź. Nie. Automat nowego typu możne rozpoznać język $a^nb^nc^n$, który nie jest bezkontekstowy. W pierwszym podejściu sprawdza, czy jest tyle samo a co b, a w drugim podejściu sprawdza czy jest tyle samo b co c.

5. Czy jest częściowo rozstrzygalny problem: dana maszyna Turinga, pytanie czy akceptuje wszystkie słowa długości pierwszej?

   Odpowiedź. Nie. Do problemu z zadania można zredukować problem, który nie jest nawet częściowo rozstrzygalny: czy dana deterministyczna maszyna Turinga odrzuca słowo puste. (Jest to dopełnienie problemu stopu, gdyby zarówno problem stopu i jego dopełnienie były częściowo rozstrzygalne, to problem stopu by był całkowicie rozstrzygalny.) Redukcja wygląda tak. Dla deterministycznej maszyny Turinga M konstruujemy maszynę M' nad alfabetem jednoliterowym, która akceptuje słowo a^p wtedy i tylko wtedy gdy p jest n-tą liczbą pierwszą i maszyna

M odrzuca słowo puste w n krokach. Maszyna M' oblicza n-tą liczbę pierwszą p, symuluje M i zatrzymuje symulację po p krokach. Przekształcenie $M \mapsto M'$ jest redukcją, bo M odrzuca słowo puste wtedy i tylko wtedy gdy M' akceptuje wszystkie słowa długości pierwszej.

6. Czy jest całkowicie rozstrzygalny problem: dana maszyna Turinga, pytanie czy akceptuje pewne słowo długości pierwszej?

   Odpowiedź. Nie. Gdyby problem był całkowicie rozstrzygalny, to i jego dopełnienie byłoby całkowicie rozstrzygalne, a poprzednim zadaniu pokazaliśmy, że nie jest.

7. Czy dla każdego $n \in \mathbb{N}$ istnieje niedeterministyczny automat skończony jednoznaczny o n stanach, taki że każdy równoważny automat deterministyczny ma wykładniczo więcej stanów?

   Odpowiedź. Tak. Niech alfabet to $\{a, b\}$. Język „n-ta litera od końca to a" jest rozpoznawany przez niedeterministyczny jednoznaczny automat o $n+1$ stanach, a jego minimalny deterministyczny automat ma 2^n stanów.

8. Czy dla nieskończenie wielu $n \in \mathbb{N}$ istnieje niedeterministyczny automat skończony o n stanach, taki że najkrótsze akceptowane słowo ma długość $n+2$?

   Odpowiedź. Nie. Taki automat nie istnieje dla żadnego n. Każdy niepusty automat niedeterministyczny o n stanach akceptuje słowo długości $\leqslant n$.

9. Rozważmy wariant wyrażeń regularnych, gdzie oprócz zwykłej gwiazdki, mamy też gwiazdkę L^K, gdzie L, K są wyrażeniami. Taka gwiazdka generuje język:

   $$\bigcup_{n:K \text{ generuje pewne słowo długości } n} L^n.$$

   Czy takie wyrażenia generują tylko języki regularne?

   Odpowiedź. Tak. Zwykłe wyrażenie regularne dla języka L^K tworzymy tak: bierzemy zwykłe wyrażenie regularne dla K i podstawiamy w miejsce każdej litery zwykłe wyrażenie regularne dla L.

10. Rozważmy model niedeterministycznego automatu skończonego, gdzie kryterium akceptacji to "co najmniej dwa biegi akceptują". Czy takie automaty rozpoznają wszystkie języki regularne i nie rozpoznają nieregularnych?

    Odpowiedź. Tak. Każdy język regularny L można rozpoznawać w taki sposób: bierzemy rozłączną sumę deterministycznego automatu dla L, oraz deterministycznego automatu który akceptuje wszystkie słowa. Taki automat ma jeden bieg akceptujący na słowach spoza L, a dwa biegi na słowach z L.

    Należy też pokazać, że automaty takie nie wykraczają poza języki regularne. Przy oznaczeniach z zadania 1 z części nietestowej, język akceptowany przez automat z treści zadania to:

    $$\bigcup_{q \neq p \in Q} (L_{I,\{p\}} \cap L_{I,\{q\}}) \cdot (L_{\{p\},F} \cap L_{\{q\},F}).$$

Powyższy język jest regularny, bo jest zbudowany z języków regularnych korzystając z sum, przecięcia i konkatenacji.

11. Czy dla każdych dwóch języków bezkontekstowych ich przecięcie jest w klasie PTIME?

    Odpowiedź. Tak. Uruchamiamy wielomianowy algorytm CYK dla obu języków po kolei.

12. Czy $L - K$ jest częściowo rozstrzygalny, jeśli L jest częściowo rozstrzygalny, a K bezkontekstowy?

    Odpowiedź. Tak. Najpierw uruchamiamy algorytm dla L. Jeśli się zatrzyma i zaakceptuje, to uruchamiamy algorytm CYK dla K i akceptujemy tylko jeśli słowo nie należy do K.

13. Czy jest bezkontesktowy zbiór słow nad alfabetem $\{a, b\}$, które nie są postaci yxx^R, gdzie $|x| = |y|$.

    Odpowiedź. Tak. Słowo należy do języka wtedy i tylko wtedy, gdy ma długość niepodzielną przez 3, lub ma długość $3n$ i da się je przedstawić w postaci

    $$\underbrace{y}_{\text{długość } n} \quad \underbrace{x_1 \sigma x_2}_{\text{długość } n} \quad \underbrace{x_2' \sigma' x_1'}_{\text{długość } n}$$

    gdzie słowa x_1 i x_1' mają tę samą długość, a σ, σ' są różnymi literami. Słowa długości niepodzielnej przez trzy są oczywiście językiem bezkontekstowym, skupmy się więc na warunku dla słów o długościach postaci $3n$. Jeśli długości słów x_1 i x_2 oznaczymy przez k i m, to po przestawieniu nawiasów, drugi warunek opisuje słowa postaci

    $$\underbrace{u_1}_{\text{długość } 2k} \quad \underbrace{u_2}_{\text{długość } m+1} \quad \sigma \quad \underbrace{u_3}_{\text{długość } 2m} \quad \sigma' \quad \underbrace{u_4}_{\text{długość } k} .$$

    Takie słowa generuje gramatyka, która najpierw tworzy $u_2 \sigma u_3$, a potem resztę słowa.

14. Czy jest bezkontekstowy język $\{a^x b^y c^z : \text{zbiór } \{x, y\} \text{ jest inny niż zbiór } \{z, z+1\}\}$

    Odpowiedź. Tak. Słowo należy do języka wtedy i tylko wtedy jeśli zachodzi któryś z następujących warunków:

    - odległość $|x - y|$ nie jest dokładnie 1; lub
    - x nie jest żadną z liczb $\{z, z+1\}$; lub
    - y nie jest żadną z liczb $\{z, z+1\}$.

    Wszystkie trzy warunki można sprawdzić gramatyką.

15. Załóżmy, że dwa automaty niedeterministyczne skończone o n stanach akceptują te same słowa długości $\leqslant n^3$. Czy akceptują te same słowa w ogólności?

Odpowiedź. Nie. Dla każdego n można napisać niedeterministyczny o wielomianowej liczbie stanów $f(n)$, który akceptuje wszystkie słowa z wyjątkiem jednego:

$$bin(0)\#bin(1)\#bin(2)\# \cdots \#bin(2^n - 1)\#.$$

(W powyższym, $bin(i)$ oznacza binarny zapis liczby naturalnej i.) Automat ten zgadza się z automatem który akceptuje wszystkie słowa na słowach o długościach aż do $n \cdot 2^n$, a dla odpowiednio dużych n liczba ta jest większa niż $\geqslant f(n)^3$.