Reprezentacja wiedzy: Bazy Wiedzy

Wojciech Jaworski

Instytut Informatyki
Uniwersytet Warszawski

Baza danych a baza wiedzy

- Baza danych: kolekcja danych zapisanych zgodnie z określonymi regułami.
- Baza wiedzy: sformalizowana reprezentacja danych dotyczących danej dziedziny (obszaru tematycznego) umożliwiająca wnioskowanie.
- Struktura (zbiór pojęć i relacji między nimi) bazy danych zadana jest przez jej schemat, struktura bazy wiedzy to ontologia.
- W bazie danych informacje są zapisane strukturze danych równoważnej logice EC.
- Rolę teorii opisującej właściwości bytów pełnią wyzwalacze, procedury składowane w bazie i aplikacja korzystająca z bazy.
- W bazie wiedzy mamy jeden język służący do reprezentacji struktury, danych i teorii.
- Wiedzę wydobywamy z bazy wiedzy za pomocą wnioskowania.
- Procedury wnioskujące są oddzielone od wiedzy.

Systemy eksperckie (ekspertowe)

- Aplikacje działające w oparciu o bazy wiedzy.
- Dzięki oddzieleniu wnioskowania od wiedzy systemy eksperckie posiadają zdolność do wyjaśnień, uzasadnienia swojego działania.
- Szkielety systemów ekspertowych: systemy ekspertowe pozbawione wiedzy.
- CLIPS (C Language Integrated Production System)

Spis treści

Projekt CYC (enCYClopedia)

- Projekt ma na celu stworzenie bazy wiedzy zawierającej ogólną i potoczną wiedzę o świecie (common sense knowledge) umożliwiającą wnioskowanie automatyczne na podobieństwo wnioskowania ludzkiego.
- Komponenty:
  - język reprezentacji wiedzy CycL;
  - powszechna ontologia świata (pojęcia, indywidua i relacje między nimi);
  - teoria świata (zbiór stwierdzeń, faktów, reguł, heurystyk, itp. wiążących za sobą pojęcia z ontologii);
  - system wnioskowania.
- W chwili obecnej, Cyc zawiera:
  - prawie 500.000 pojęć;
  - w tym około 26.000 typów relacji;
  - około 5.000.000 faktów (twierdzeń) dotyczących tych pojęć.

Wiedza zawarta w CYC'u

- Thing, Intangible Thing, Individual, Relations, Sets, Collections, Paths, Logic, Math
- Space; Spatial Paths; Borders, Geometry; Movement; Artifacts; Events & Scripts; Time
- Physical Objects; Materials, Parts & Statics; State Change Dynamics; Physical Agents; Plans & Goals; Actors & Actions; Agents
- Weather; Natural Geography; Chemistry; Physical Artifacts; Astronomy; Earth & Solar System
- Organization; Organizational Actions; Types of Organizations; Organizational Plans; Law; Political Geography; Nations, Governments & Geopolitics; Business; Military Organizations
- Living Things; Ecology; Plants & Animals; Social Behavior
- Sentient Beings; Emotion, Perception & Belief; Human Anatomy & Physiology; Products & Devices; Conceptual Works; Literature & Works of Art; Human Behavior & Actions; Vehicles, Buildings & Weapons; Mechanical & Electrical Devices; Software; Language; Social Relations & Culture
- Human Activities; Business & Commerce; Politics & Warfare; Sports, Recreation & Entertainment; Professions & Occupations; Purchasing & Shopping; Social Activities; Transportation & Logistics; Travel & Communication; Everyday Living
- Domain-Specific Knowledge (Chemistry, Computer Security, Healthcare, ...)
- Domain-Specific Facts and Data

Język CycL

- Język reprezentacji wiedzy używany w Cyc'u.
- Rachunek predykatów pierwszego rzędu rozszerzony o obsługę równości, domniemań, skolemizację i elementy logiki drugiego rzędu takie jak:
  - kwantyfikowanie po predykatach
  - predykaty metajęzykowe.
- CycL korzysta z *circumscription*, zakłada istnienie unikalnych nazw, i, gdy jest to stosowne, wykorzystuje założenie zamkniętego świata.
- Składnia CycL oparta jest na LISPie.
- *Circumscription* — logika, w której dla zadanej teorii ograniczamy się do jej minimalnych modeli (podobna do CWA).

System wnioskowania (inference engine)

- System wnioskowania umożliwia odpowiadanie na pytania zadawane przez użytkownika.
- Klasyczne metody wnioskowania w logice nie są skalowalne do baz wiedzy rozmiaru takiego jak Cyc.
- System wnioskujący korzysta z klasycznych reguł wnioskowania w logice takich jak modus ponens, modus tollens oraz uniwersalna i egzystencjalna kwantyfikacja.
- Ich szczególnymi przypadkami są mechanizmy wnioskowania takie jak dziedziczenie, czy automatyczna klasyfikacja.
- Cyc wykonuje zachłanne przeszukiwanie w przestrzeni dowodów sterowane heurystykami.
- Cyc zawiera także moduły wnioskowania dedykowane do obsługi konkretnych klas wnioskowania, na przykład określania należenia do kolekcji, bądź ich rozłączności, wnioskowania o równości, czasie, czy wnioskowania matematycznego.

Mikroteorie — struktura bazy wiedzy

- Wiedza zawarta w Cyc'u podzielona jest na tysiące "mikroteorii".
- Każda z nich jest zbiorem stwierdzeń, które dotyczą tego samego zestawu pojęć.
- Mikroteorie skupiają się na danej dziedzinie wiedzy, mają określony poziom szczegółowości, dotyczą określonego przedziału czasu, itp.
- Mechanizm mikroteorii pozwala reprezentować stwierdzenia, które są w ogólności sprzeczne.
- Mikroteorie umożliwiają skupienie procesu wnioskowania, co zwiększa wydajność systemu.
- Podział wiedzy na mikroteorie jest zgodny ze sposobem w jaki jest ona reprezentowana w mózgu.

Podsystem przetwarzania języka naturalnego

- Komponenty:
- Leksykon
  - słowom z języka angielskiego są przypisane stałe np. `#$Light-TheWord`,
  - stałym tym za pomocą formuł CycL są przypisane kategorie gramatyczne, walencja, pojęcia reprezentowane przez słowo np. `#$LightEnergy`, czy `#$LightingDevice`;
- Parser składniowy
  - działa w oparciu o phrase-structure grammar;
  - zwraca wszystkie możliwe drzewa rozbioru;
- Interpreter semantyczny
  - zamienia drzewa rozbioru składniowego w formuły CycL;
  - działa zgodnie z zasadą kompozycjonalności;
  - czasownikom są przypisane szablony, w które są wstawiane pojęcia odpowiadające ich argumentom.

Podsystem przetwarzania języka naturalnego

- Baza wiedzy w przetwarzaniu języka naturalnego stanowi model (nadaje strukturę), który rozszerzany jest informacjami wydobytymi z tekstów.
- Wiedza zawarta w Cyc'u umożliwia rozstrzyganie niejednoznaczności semantycznych.
- Aktualnie Cyc potrafi zinterpretować jedynie fragment języka angielskiego.
- Celem jest umożliwienie komunikacji z użytkownikiem w języku naturalnym.

Szyna Integracji Semantycznej (Semantic Integration Bus)

- Narzędzie do integracji Cyc'a z zewnętrznymi źródłami danych takimi, jak bazy danych, arkusze kalkulacyjne, strony internetowe, dokumenty tekstowe.
- Zewnętrzne informacje są udostępniane w bazie wiedzy jako wirtualne fakty, co umożliwia wykorzystanie ich podczas wnioskowania.
- Cyc interpretuje rekordy bazodanowe jako niejawne formuły CycL, podobnie przetwarza pola tekstowe.
- Integracja odbywa się półautomatycznie.

Uczenie i rozwój

- W zamierzeniu docelowym CYC ma się uczyć sam.
- Aktualnie potrafi uczyć się nowych pojęć na podstawie Wikipedii.
- Nowe stwierdzenia są nieustannie dodawane do bazy wiedzy za pomocą metod ręcznych i automatycznych.
- Dużą liczbę faktów Cyc dodaje samemu jako efekt wnioskowania.
- Obecnie Cyc zawiera ok. 10% wiedzy potrzebnej do tego, by się rozwijać samemu.

Dystrybucje

- OpenCyc (`opencyc.org`) jest bazą uproszczoną (stwierdzenia ograniczone do opisu generalizacji i należenia do pojęcia), dostępną na zasadach licencji Open Source.
- OpenCyc dostępny jest jako aplikacja, ontologia w formacie OWL oraz jako sieć semantyczna (Semantic Web) pod adresem `sw.opencyc.org`.
- ResearchCyc (`researchcyc.cyc.org`) to pełna zawartość Cyc'a udostępniana licencji do celów badawczych.

Rozproszona sztuczna inteligencja

- Podział wiedzy na mikroteorie pozwala wnioskować w środowisku rozproszonym.
- Odpowiada to sposobowi w jaki działają ludzie, polegającym na specjalizacji i współpracy:
  - przeciętny człowiek posiada wiedzę potrzebną mu do komunikacji z innymi,
  - ale nikt nie posiada sumy wiedzy,
  - zamiast tego mamy ekspertów, których prosimy o pomoc.
- Implementacja powyższego zjawiska wygląda następująco:
  - każdy agent posiada ogólną bazę wiedzy dotyczącą podstawowych faktów
  - oraz szczegółową wiedzę o dziedzinie, w której się specjalizuje
- Jeśli agent podczas wnioskowania stwierdzi, że problem wykracza poza jego wiedzę, komunikuje się z agentem-ekspertem.
- Wspólna podstawowa ontologia jest konieczna, by tą komunikację umożliwić.

Platforma do budowania aplikacji wspomagających wykonywanie analiz

- Zapytania zadawane w języku naturalnym (a nie w SQL).
- Odpowiedzi odnajdywane w wyniku wnioskowania (a nie dopasowania do wzorca) wraz z uzasadnieniem.
- Baza wiedzy ogólnej uzupełniana o wiedzę dziedzinową.
- Przykłady:
  - Analizowanie danych medycznych dotyczących populacji, chorób, diagnozowania itp.
    - ★ *Which patients had a heart attack less than 2 weeks prior to a coronary artery bypass graft (CABG) between 2000 and 2005?*
  - Analizy finansowe: określanie produktywności korporacji, trendów rynkowych, wpływu wydarzeń na rynek...
    - ★ *What type of derivative, whose underlyer is equity in African corporations, is most heavily invested in by GS corporate customers in Japan?*
  - Modelowanie wzorców zachowań i norm w różnych kulturach. Przewidywanie w jaki sposób dane postępowanie zostanie odebrane przez członków danej społeczności.

CycSecure: Network Intrusion Protection

- Przewiduje ataki, zaleca metody naprawy
- Baza wiedzy o lukach w zabezpieczeniach i ich wpływie na działanie sieci.
- Model sieci, na którym są wykonywane symulacje
- Symulowanie ataku hakerów

Spis treści

Wordnet (wordnet.princeton.edu)

- Sieć semantyczna (Semantic Network) zawierająca słowa (rzeczowniki, czasowniki, przymiotniki i przysłówki) pogrupowane zgodnie z relacją synonimii (synsety).
- 117 000 synsetów (pojęć).
- Słowa wieloznaczne należą jednocześnie do kilku synsetów.
- Każdy synset ma krótką definicję i zdanie ilustrujące jego użycie.
- Synsety są ze sobą powiązane relacją bycia pojęciem bardziej ogólnym/szczegółowym (hiperonimia, hiponimia, subsumpcja, relacja *is a*), która tworzy hierarchię pojęć.
- Wordnet odróżnia pojęcia od indywiduów i posiada relację należenia do pojęcia.
- Wprowadzona jest relacja bycia częścią (meronimia). Jest ona dziedziczona przez pojęcia bardziej szczegółowe.
- Występują też relacje wiążące słowa o wspólnym rdzeniu.
- Słowosieć (plWordnet, `/plwordnet.pwr.wroc.pl`) jest polską wersją Wordnetu. Zawiera 73839 synsetów.

GeoNames

- Otwarta baza danych geograficznych `www.geonames.org`.
- Ponad 10 milionów nazw geograficznych.
- 7,5 miliona unikalnych miejsc.
- Nazwy są podzielone na 645 kategorii np. (region administracyjny II stopnia, zatoka, pojezierze, wioska, gazociąg itp.).
- Z nazwą skojarzone są jej aliasy i różne wersje językowe: New York, Bandaraya New York,Big Apple,Cathair Nua Eabhraic,Eabhraig Nuadh,Efrog Newydd,Evrek Nowydh,Nowy Jork,...
- Informacje o przynależności administracyjnej, współrzędnych geograficznych, liczbie mieszkańców, wysokości nad poziomem morza.

Freebase

- Freebase is a large collaborative knowledge base consisting of metadata composed mainly by its community members.
- It is an online collection of structured data harvested from many sources, including individual, user-submitted wiki contributions.
- Freebase aims to create a global resource which allows people (and machines) to access common information more effectively.
- Unlike the W3C approach to the semantic web, which starts with controlled ontologies, Freebase adopts a folksonomy approach, in which people can add new categories (much like tags), in a messy sprawl of potentially overlapping assertions.
- 46,295,148 Topics — entities (25 November 2014)
- 2,677,948,113 Facts (25 November 2014)

Graphs

- Freebase data is stored in a data structure called a graph.
- A graph is composed on nodes connected by edges.
- In Freebase, the nodes are defined using /type/object and edges are defined using /type/link.
- By storing the data as a graph, Freebase can quickly traverse arbitrary connections between topics and easily add new schema without having to change structure of the data.

Topics

- Freebase has over 39 million topics about real-world entities like people, places and things.
- Since Freebase data is represented a graph, these topics correspond to the nodes in the graph.
- Examples of the types of topics found in Freebase:
  - Physical entities, e.g., Bob Dylan, the Louvre Museum, the Saturn planet,
  - Artistic/media creations, e.g., The Dark Knight (film), Hotel California (song),
  - Classifications, e.g., noble gas, Chordate,
  - Abstract concepts, e.g., love,
  - Schools of thoughts or artistic movements, e.g., Impressionism.
- Some topics are notable because they hold a lot of data (e.g., Wal-Mart), and some are notable because they link to many other topics, potentially in different domains of information.
- For example, abstract topics like love, poverty, chivalry, etc. don't have many properties associated with them but they appear often as book subjects, film subjects, etc. making them more notable.

Types

- Any given topic can be seen for many different perspectives for example:
  - Bob Dylan was a song writer, singer, performer, book author, and film actor;
  - Leonardo da Vinci was a painter, a sculptor, an anatomist, an architect, an engineer, ...;
  - Love is a book subject, film subject, play subject, poetry subject, ...;
  - Any city is a location, potentially a tourist destination, and an employer of civil servants.
- In order to capture this multi-faceted nature of many topics, the concept of types is introduced in Freebase.
- Topics in Freebase can have any number of types assigned to them.
- The topic about Bob Dylan is assigned several types: the song writer type, the music composer type, the music artist (singer) type, the book author type, etc.

Properties

- Each type carries a different set of properties germane to that type. For example,
  - The music artist type contains a property that lists all the albums that Bob Dylan has produced as well as all the music instruments he was known to play;
  - The book author type contains a property that lists all the books Bob Dylan has written or edited, as well as his writing school of thoughts or movement;
  - The company type contains many property for listing a company's founders, board members, parent company, divisions, employees, products, year-by-year revenue and profit records, etc.
- Thus, a type can be thought of as a conceptual container of properties that are most commonly needed for describing a particular aspect of information.

Spis treści

Składnia CyCL

- Stałe poprzedzamy znakami `#$`, np `#$Dog`.
- Predykaty piszemy w nawiasach podając nazwę predykatu a następnie argumenty: `(#$isa #$Burek #$Dog)`
- Stałe reprezentujące funkcje zawierają w nazwie `Fn`, np `#$BirthFn`.
- Termy piszemy w nawiasach podając nazwę funkcji a następnie argumenty: `(#$BirthFn #$Burek)`
- Spójniki logiczne: `#$and`, `#$or`, `#$not`, `#$implies`:

```
(#$or (#$isa #$Burek #$Dog)
      (#$not (#$isa #$Burek #$Dog)))
```

- Kwantyfikatory: `#$forAll`, `#$thereExists`:

```
(#$forAll ?X (#$implies (#$isa ?X #$Dog)
                        (#$not (#$isa ?X #$Cat))))
```

- Zmienne poprzedzamy `?`, np `?X`.

Kolekcje i indywidua

- `#$Collection` — pojęcie, kolekcja
- `#$Individual` — indywiduum, przedmiot, instancja pojęcia
- `(#$isa X Y)` oznacza:
  `X` jest obiektem należącym pojęcia `Y`;
  `X` jest instancją kolekcji `Y`.
- `(#$genls X Y)` oznacza, że
  pojęcie `Y` jest bardziej ogólne od pojęcia `X`;
  każda instancja kolekcji `X` jest również instancją kolekcji `Y`.
- `#$genls` jest częściowym porządkiem (zwrotna, antysymetryczna i przechodnia)
- `(#$disjointWith X Y)` — pojęcia `X` i `Y` są rozłączne.

Interfejs OpenCyc

Index Frame

- Służy do wyświetlania informacji o wybranej stałej z bazy wiedzy.
- Zawiera wykaz operacji, które mogą być wykonane na wyświetlanej stałej, jak również listę typów asercji jej dotyczących.
- Klikając na zielony znak plus obok nazwy relacji zadajemy pytanie o możliwe wartości jednego z jej argumentów, np. `(#$isa #$RoadVehicle ?ARG2)`.
- Kliknięcie na czerwony romb wyświetla hierarchię pojęć.

Assertion Display Frame

- Zawiera asercje dotyczące wybranej stałej.
- Jej zawartość zależy od opcji wybranej w Index Frame.
- Aby zobaczyć szczegółowe informacje o wybranej asercji należy kliknąć na kulkę.
- Znaczenia kolorów kulek:
  - biały — monotonicznie prawdziwe
  - żółty — prawdziwe przez domniemanie
  - niebieski — Forward Rule
  - fioletowy — Backward Rule
  - czerwony — zanegowane
  - zielony — wywnioskowane

Arność i typy argumentów

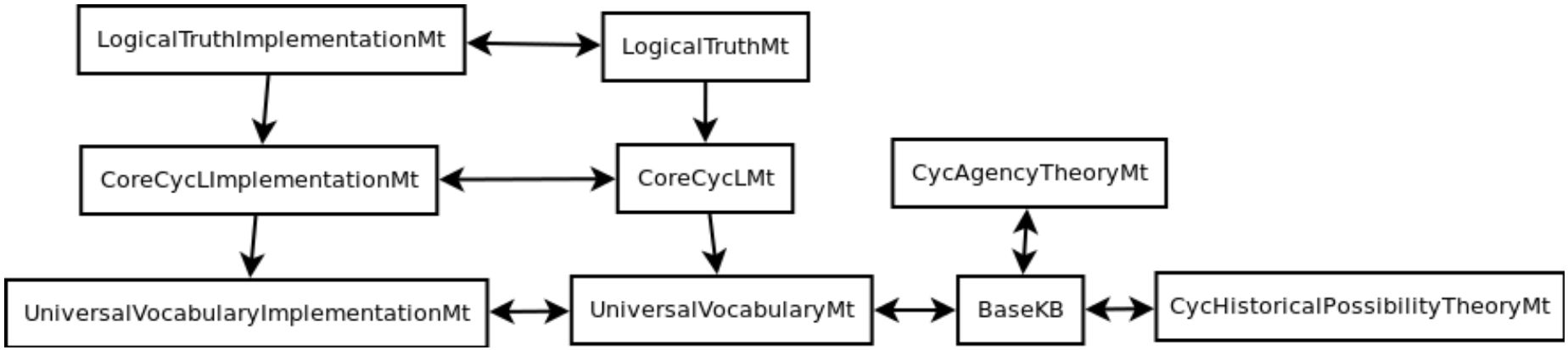
- `#$arity` - predykat określający liczbę argumentów funkcji lub predykatu: `(#$arity #$BirthFn 1) (#$arity #$arity 2)`
- `#$argIsa`, `(#$argIsa #$performedBy 1 #$Action)` oznacza, że pierwszy argument `#$performedBy` musi być instancją `#$Action`.
- `#$argGenl (#$argGenl #$penaltyForInfraction 2 #$Event)` oznacza, że drugi argument `#$penaltyForInfraction` musi być podkolekcją `#$Event`.
- `#$arg[N]Isa`, `#$arg[N]Genl`
- `#$resultIsa` wskazuje kolekcję, do której należy wynik funkcji.
- `#$resultGenl` wskazuje kolekcję, której podkolekcją jest wynik funkcji.
- `#$UnaryFunction`,`#$BinaryPredicate`, `#$BinaryFunction`

Kolekcje kolekcji

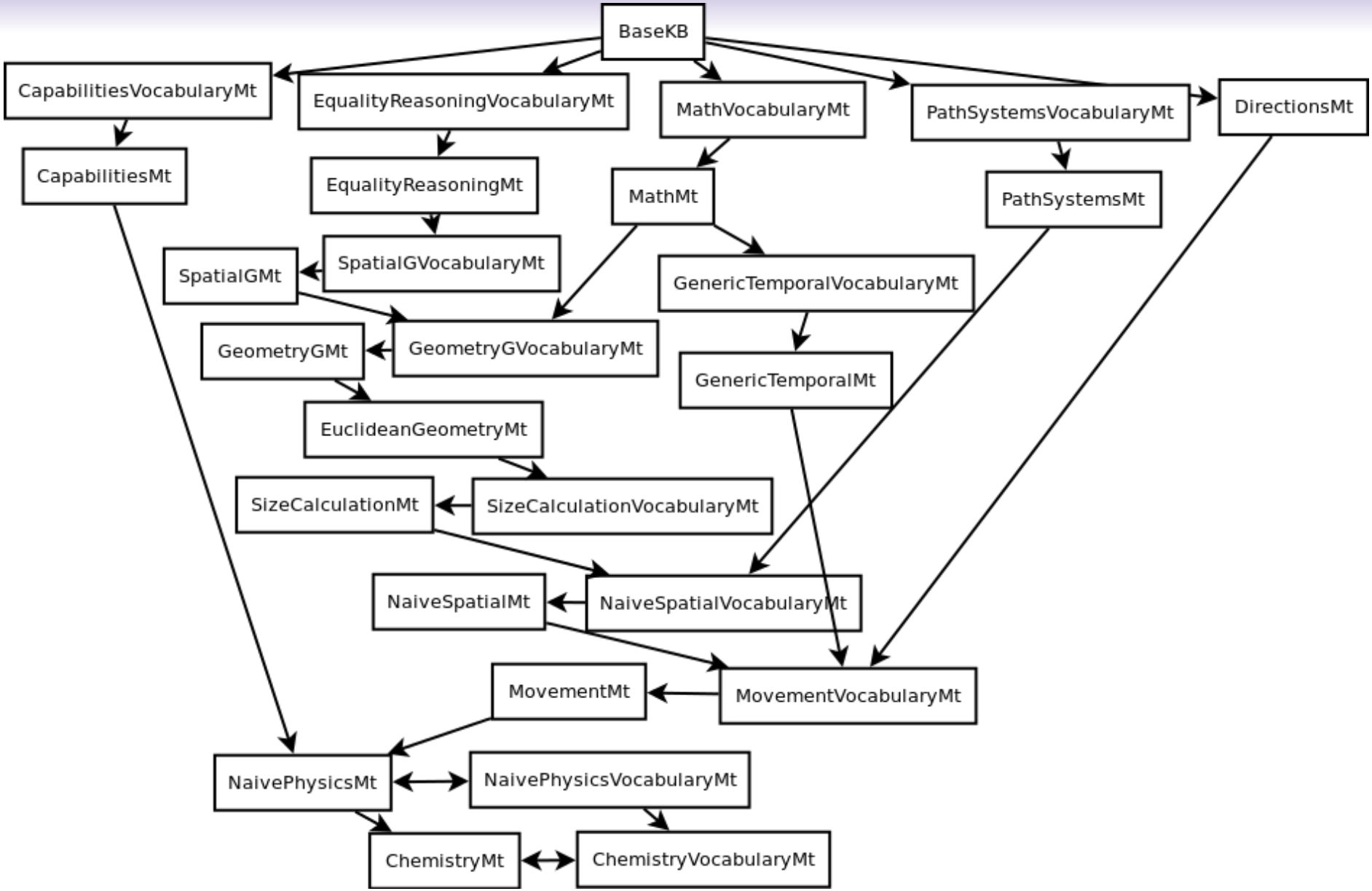
- Kolekcje, których instancjami są kolekcje nazywamy typami.
- `COLTYPE`, w predykacie `(#$typeGenls COLTYPE COL)` oznacza typ (kolekcję kolekcji) zaś `COL` kolekcję.
- `(#$typeGenls COLTYPE COL)` oznacza, że każda instancja `COLTYPE` posiada `COL` jako generalizację.
- Na przykład: `(#$typeGenls #$CreditCardTypeByBrand #$CreditCard)` oznacza, że każdy typ karty kredytowej jest podkolekcją kolekcji wszyskich kart kredytowych.
- Uwaga: nie oznacza to, że każda podkolekcja `CreditCard` jest instancją `CreditCardTypeByBrand`.
- `#$IndividualDenotingFunction` funkcje zwracające indywidua `(#$resultIsa ?FUNCTION #$Individual)`
- `#$CollectionDenotingFunction` funkcje zwracające kolekcje `(#$resultIsa ?FUNCTION #$Collection)`

Mikroteorie

- Fakty są prawdziwe jedynie w obrębie mikroteorii.
- `(#$ist MT FORMLA)` means that the Cyc formula `FORMLA` is true in the microtheory `MT`.
- `(#$genlMt MT-1 MT-2)` means that every assertion which is true in `MT-2` is also true in `MT-1`.
- Relecja `#$genlMt` zadaje na mikroteoriach częściowy porządek.
- Najbardziej ogólne mikroteorie



Mikroteorie składajce się na chemię



Reifikacja

- A non-atomic term (NAT) is a denoting term like a constant. NATs are formed by applying a denotational function to a denoting term, ex. `#$GovernmentFn`
- Some functions return concepts that we want to “reify” and keep in the KB. When a new NAT is created using a reifiable function, that new term is itself reified (kept around separately in the KB) and becomes a Non- Atomic Reified Term, or NART,
- Other functions return concepts that we don’t want to store separately in the KB, ex. `#$TimesFn`
- When a new NAT is created using an unrefiable functions, it does not get created as a persistent term in the KB. Unrefiable functions result in Non- Atomic Unreified Terms, such as (#$TimesFn 3 7).