

LAB: Regułowe systemy ekspertowe

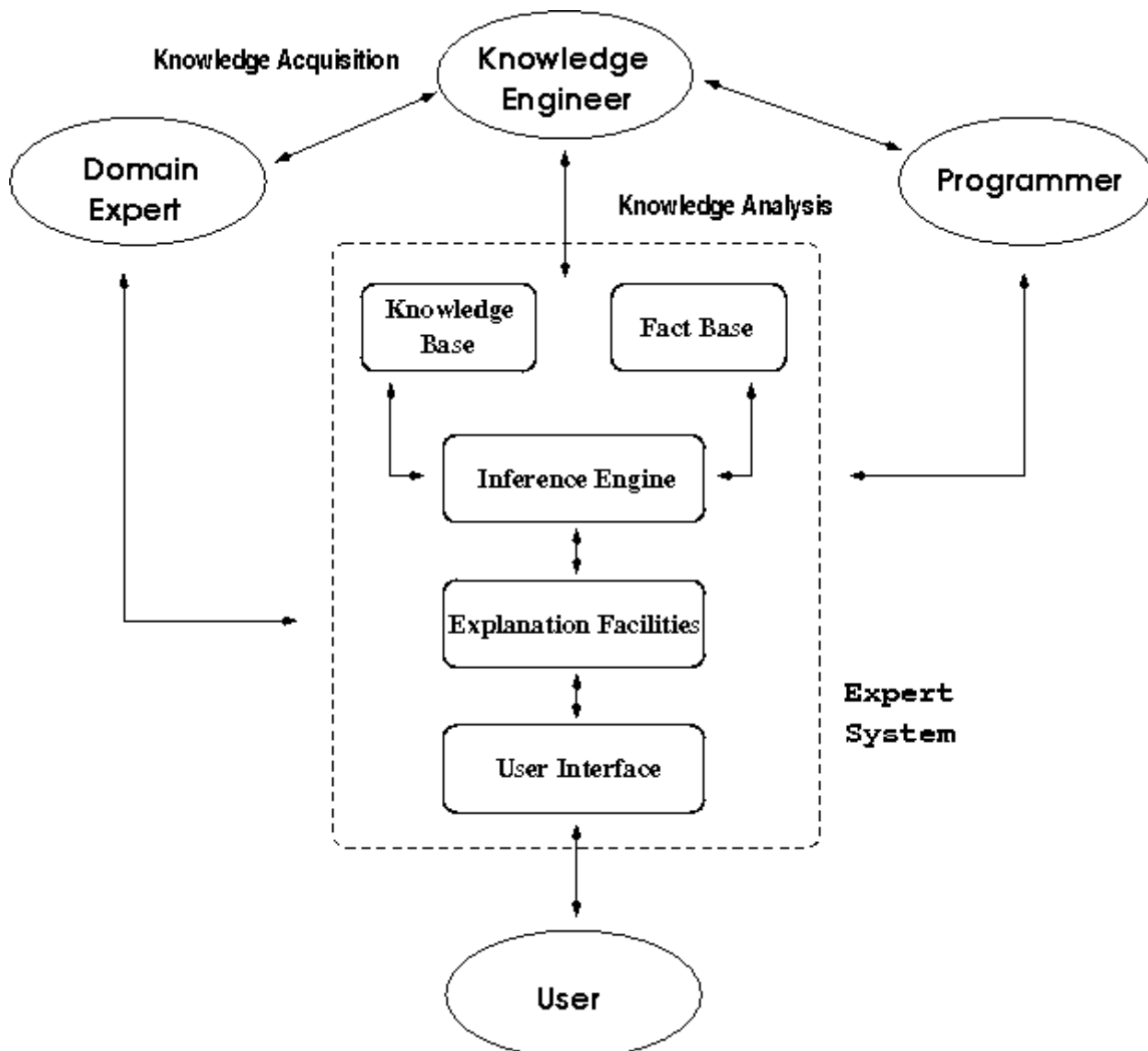
Celem ćwiczenia jest zapoznanie się z regułową reprezentacją wiedzy i systemami ekspertowymi, które je wykorzystują.

1 Wstęp

W systemie ekspertowym można wyróżnić następujące elementy:

- baza wiedzy, czasami dzielona na: właściwą bazę wiedzy (czyli tą którą system dysponuje stale, od początku/uruchomienia) i bazę faktów, które system odkrywa, dostaje, wypracowuje,
- mechanizm wnioskujący, który przeprowadza właściwy proces wnioskowania, tj. odnajduje rozwiązanie/odpowiedź
- mechanizm wyjaśniający, dlaczego jest to odpowiedź poprawna/dopuszczalna,
- interfejs użytkownika, pozwalający na komunikację z systemem.

Rysunek 1: Struktura Systemu Ekspertowego.



2 CLIPS

Rodzina

1. Ściągnij plik family1.clp.zip
2. Po ściągnięciu pliku należy zawartą w nich wiedzę wczytać do bazy wiedzy a następnie uruchomić wnioskowanie przy pomocy komendy:

```
clips -f family1.clp
```

3. Po poprawnym załadowaniu na ekranie powinna ukazać się lista zdefiniowanych faktów.
4. Tutaj także warto zwrócić uwagę czy nie wystąpiły jakieś błędy.

W CLIPS-ie mamy dostępne wnioskowanie w przód, a więc na podstawie istniejących faktów sprawdzane będą warunki reguł i te reguły, których warunki są spełnione zostaną uruchomione. Cykl ten powtarza się aż do momentu kiedy żadna reguła nie zostanie uruchomiona. Informacją wejściową dla algorytmu wnioskowania w przód jest **zbiór faktów** w bazie wiedzy.

- Na podstawie otrzymanego rezultatu uruchomienia wnioskowania odpowiedz na następujące pytania:
 - Jakie reguły są uruchamiane?
 - Jakie fakty aktywują te reguły?
 - Jak wyjaśnić otrzymany rezultat?
 - Ile faktów zostało dodanych do bazy wiedzy w trakcie wnioskowania?
 - Czy takiego wyniku można było się spodziewać?

Aby reguła opisująca **dziecko** mogła zostać uruchomiona należy zdefiniować regułę dodającą do bazy wiedzy fakt mówiący kto jest **rodzicem** kogo (zmiana bazy wiedzy spowoduje uruchomienie kolejnego cyklu wnioskowania):

```
(defrule rodzic
  (or (mama ?x ?y) (tata ?x ?y))
=>
  (assert (rodzic ?x ?y))
)
```

Uruchom wnioskowanie ponownie i odpowiedz na pytania:

- Ile teraz jest faktów w bazie CLIPS-a?
- Jakie są aktywacje reguł w CLIPS-ie przed rozpoczęciem wnioskowania (rezultat wywołania funkcji (agenda))?
- Dlaczego wśród nich nie ma reguł definiujących **dziecko**?

Zadanie 1

Rozbuduj bazę reguł dodając do niej reguły definiujące relacje rodzinne takie jak

- syn, córka
- małżeństwo
 - Dwie osoby są małżeństwem jeżeli:
 - są różnej płci,
 - mają wspólne dziecko.
 - Definiując regułę przy pomocy powyższych warunków, algorytm wnioskowania wyszuka nam tylko małżeństwa posiadające dzieci. Małżeństwa, które nie posiadają dzieci są niemożliwe do znalezienia na podstawie dostępnej wiedzy.
- rodzeństwo
 - Dwie osoby są rodzeństwem jeżeli
 - mają wspólnego rodzica
 - są dwiema różnymi osobami
- dziadkowie, dziadek, babcia
- wnuk, wnuczka, wnuczek
- kuzynostwo
- ciocia, wujek

- Jak przy pomocy reguł definiujących **siostrę** oraz **rodzica** zdefiniować pojęcie **ciocia**?
- Jak przy pomocy reguł definiujących **brata** oraz **rodzica** zdefiniować pojęcie **wujek**?

Truth maintenance

Truth Maintenance System w systemach regułowych jest odpowiedzialny za utrzymanie spójności wiedzy w nich zawartych. Niespójność może być różna: niespójność logiczna wiedzy, niespójność materialna (w odniesieniu do interpretacji wiedzy w przyjętym świecie).

- Czym różnią się te rodzaje niespójności ?
- Które z nich mogą wystąpić w systemach produkcyjnych ?

1. Ściągnij plik family2.clp.zip
2. Otwórz i zapoznaj się z jego zawartością. Zwróć uwagę na:
 - Definicję faktów **wiek** określających wiek danej osoby.
 - Definicję modułów **defmodule** - jaką rolę odgrywają moduły ?
 - Dwie reguły zdefiniowane w pliku:
 - **przyznaj-bilet::bilet-uczen** która dla każdej niepełnoletniej osoby w dodaje fakt mówiący, że należy się jej bilet dziecięcy. Następnie reguła ta wyświetla kompletną listę faktów znajdujących się w bazie wiedzy.
 - **zwiększ-wiek::zwiększ** która zwiększa wiek dla danej osoby.
3. Uruchom ściągnięty plik. Aby łatwiej było przeglądać rezultaty wyświetlane podczas uruchomienia, można użyć następującej komendy

```
clips -f family2.clp | less
```

Zaobserwuj jak zwiększany jest wiek poszczególnych osób oraz zidentyfikuj moment kiedy dodawany jest fakt typu **bilet**. Jakiego typu i dla kogo ten bilet jest dodawany ?

Zadanie 1

Zdefiniuj regułę **przyznaj-bilet::bilet-dorosly** w analogiczny sposób jak zdefiniowana została reguła **przyznaj-bilet::bilet-uczen**. Uruchom otrzymany model i zaobserwuj jakie bilety istnieją w bazie wiedzy na końcu procesu wnioskowania.

- Jak wyjaśnić taki rezultat ?
- Czy baza wiedzy w takim wypadku jest spójna ?

Zadanie 2

Popraw otrzymany model poprzez zastosowanie mechanizmu *Truth Maintenance System*

- Jaka konstrukcja języka CLIPS umożliwia jego wykorzystanie ?
- Po poprawnym zastosowaniu mechanizmu na końcu wnioskowania powinien być dokładnie jeden fakt opisujący bilet dla każdej osoby.
- Wśród wyświetlanych wiadomości wskaż miejsca, gdzie usuwane są *niepotrzebne* bilety.

3 Drools (dla zainteresowanych)

Model systemu PLOC

Celem systemu jest określenie wysokości składki ubezpieczenia na podstawie danych wejściowych. Danymi wejściowymi dla systemu są dane na temat kierowcy: wiek, okres posiadania prawo jazdy, liczba wypadków w ostatnim roku, dotychczasowa klasa ubezpieczeniowa. Kolejnym elementem istotnym przy wyliczaniu składki ubezpieczenia są dane na temat pojazdu: pojemność silnika, wiek samochodu, liczba miejsc, badanie techniczne. Ostatnim elementem są zniżki oraz zwyczajki za: liczbę rat, inne ubezpieczenia, ciągłość ubezpieczenia, liczbę samochodów ubezpieczonych. Wyliczenie składki ubezpieczenia przebiega w trzech etapach. Pierwszym etapem jest ustalenie stawki podstawowej na

podstawie pojemności samochodu. Dane o wysokości składki podstawowej są podane dla rejestracji w Szczecinie. Drugim etapem jest tzw. tabela bonus malus. Zawarte w niej są zniżki i wyżki wynikające z bezszkodowego przebiegu ubezpieczenia. Towarzystwa ubezpieczeniowe udzielają maksymalnie 60 % zniżki z tego tytułu. Za każdy bezszkodowy 12 miesięczny okres ubezpieczenia, klient przesuwa się o jedną klasę w dół tabeli. Trzecim etapem jest uwzględnienie pozostałych zniżek i wyżek m.in. za wiek, dodatkowe ubezpieczenia itp.

Kompletna lista reguł: pl-oc.pdf

Uruchomienie modelu

1. Ściągnij plik ploc-drools.drl.zip
2. W celu uruchomienia bazy wiedzy zapisanej w ściągniętym pliku, uruchamiamy narzędzie Eclipse i tworzymy nowy projekt:

I. Tworzymy nowy projekt File ⇒ New ⇒ Project...

- a. Na liście odnajdujemy pozycję Drools i ją rozwijamy.
- b. Wybieramy Drools Project
- c. Klikamy Next >
- d. W kolejnym korku kreatora podajemy nazwę projektu np. DroolsTest.
- e. Klikamy Next >
- f. W kolejnym kroku pozostawiamy zaznaczone tylko dwie pierwsze opcje.
- g. Klikamy Next >
- h. W kroku Drools Runtime z listy Generate code compatible with: wybieramy Drools 5.1 or above.
- i. Klikamy Finish

II. W worksspace pojawia się nam nowy projekt do którego dodajemy klasy pozwalające na uruchomienie projektu:

- a. Rozwijamy drzewo projektu DroolsTest ⇒ src ⇒ main ⇒ java ⇒ com ⇒ sample.
- b. Klikamy na plik *.java. W tym pliku:
 - Usuwamy definicję klasy Message
 - Usuwamy w funkcji main wszystkie linie, które korzystają z instancji tej klasy.
 - Zmieniamy nazwę pliku z regułami z Sample.drl na ploc-drools.drl.
 - Do głównej klasy dodajemy definicję typu enumerowanego:

```
public enum PaymentType {  
    single, instalments  
}
```

- c. Rozwijamy drzewo projektu DroolsTest ⇒ src ⇒ main ⇒ rules.
- d. Usuwamy plik Sample.drl.
- e. W miejsce usuniętego pliku dodajemy wcześniej ściągnięty plik ploc-drools.drl:
 - a. Klikamy prawym przyciskiem na pobrany plik i wybieramy Kopiuj.
 - β. Klikamy prawym przyciskiem na katalog rules i menu kontekstowego wybieramy opcję Wklej.

III. Wybieramy polecenie Run as ⇒ Java application.

- IV. Jeżeli wszystko przebiegło pomyślnie to w konsoli środowiska Eclipse powinny pojawić się komunikaty generowane przez reguły.

Zadania

Zadanie 1

Wśród uruchamianych reguł można zauważyć, że atrybut `DriverClass` klasy `Driver` jest zwiększany wielokrotnie:

- Czy jest to poprawne zachowanie ? Jeżeli nie, to dlaczego i jakie powinno być poprawne ?
- Jak sprawić aby tylko jedna z tych reguł została uruchomiona ?
Wskazówka: poczytaj o możliwych atrybutach reguł i spróbuj znaleźć ten który pozwoli osiągnąć żądany efekt: atrybuty reguł [http://docs.jboss.org/drools/release/6.0.1.Final/drools-docs/html_single/#d0e6643]

Zadanie 2

Model bazy wiedzy jest niekompletny i zawiera tylko część reguł z modułu `base-charge-modifiers` (agenda-group „`base-charge-modifiers`”). Na podstawie definicji innych reguł dopisz 4 ostatnie reguły podwyższające cenę ubezpieczenia zawarte w pliku `pl-oc.pdf` tzn.:

1. liczba miejsc pojazdu (6-9): +20%,
2. brak ważnego badania technicznego: +20%,
3. opłata na 2 raty: +10%,
4. brak zaświadczenia o przebiegu ubezpieczenia: +60%.

Zadanie 3

Dopisz regułę, która po przyznaniu wszystkich zniżek i zwyżek obliczy wartość końcową polisy ubezpieczenia.

- Zdefiniuj tą regułę jako element modułu `calculation` (agenda-group „`calculation`”).
- W definicji reguły można wykorzystać element `accumulate`.
- Reguła ta ma zapisywać wartość polisy w polu `Value` obiektu typu `Result`.
- Reguła powinna wyświetlać finalną wartość do zapłaty.
- Poprawna wartość, jaka powinna zostać obliczona to: `865.95`.

Zadanie 4*

Spróbuj zdefiniować powyższą regułę bez użycia `accumulate`.