

## Laboratorium logiki obliczeniowej

### Podstawy Programowania w logice

#### Elementy języka Prolog

Będziemy rozpatrywać następujące trzy elementy (inaczej - wyrażenia) języka Prolog:

- **stałe**
- **zmiennie**
- **symbole predykatywne**

Przyjmujemy, że każde powyższe wyrażenie jest ciągiem znaków alfanumerycznych (bez polskich liter diakrytyzowanych), może też zawierać znaki podkreślenia.

Za pomocą stałych reprezentuje się dowolne byty (inaczej - obiekty) należące do opisywanej rzeczywistości, jak np. osoby, przedmioty, liczby, itp. Stałe rozpoczynają się od małej litery (np. adam, a1\_b2); mogą także być ciągami samych cyfr (np. 537). Z kolei zmiennie muszą się rozpoczynać od litery wielkiej, np. X, Ya54, Osoba.

Symbole predykatywne służą do nazywania związków zachodzących między obiektami. Przykładowo, dwuargumentowy symbol predykatywny `wieksza_od` może oznaczać, że pierwszy argument jest liczbą większą o liczbę będącej drugim argumentem. Szczególnym przypadkiem są symbole predykatywne jednoargumentowe, opisujące własności obiektów, np. symbol `liczba_nieparzysta` może oznaczać liczbę niepodzielną (całkowicie) przez 2.

#### Podstawy programowania w języku Prolog

Dalej będziemy rozpatrywać rzeczywistość złożoną z liczb naturalnych (w skrócie – liczb) oraz zera, co do którego przyjmujemy, że nie jest liczbą naturalną. Założmy, że chcemy stwierdzić w Prologu, iż liczba 17 jest większa od 0. Robimy to za pomocą następującego wyrażenia, zwanego **faktem**:

```
wieksza_od(17,0).
```

Należy zwrócić uwagę na kropkę, która jest obowiązkowym znakiem kończącym dowolny fakt. Możemy także sformułować fakt bardziej ogólny, mówiący że każda liczba (naturalna) jest większa od 0:

```
wieksza_od(X,0).
```

Należy zauważyć, że zmienna (lub zmiennie) występujące w fakcie są domyślnie poprzedzone frazą "dla każdego". Oprócz faktów, w Prologu tworzy się wyrażenia zwane **regułami**. Reguła, podobnie jak fakt, postuluje zachodzenie relacji jednak pod pewnym warunkiem, podczas gdy fakt czyni to bezwarunkowo. Powiedzmy, że chcemy wyrazić, iż za "liczbę dużą" uważamy dowolną liczbę większą od 1000000. Innymi słowy, każda liczba jest duża pod warunkiem, że jest większa od miliona. W tym celu posługujemy się następującą regułą

```
liczba_duza(X) :- wieksza_od(X,1000000).
```

Każda reguła, podobnie jak fakt, jest zakończona kropką. Warunki w regułach mogą mieć postać złożoną. Przykładowo chcemy stwierdzić, że "liczba średnia" to liczba większa od tysiąca i zarazem mniejsza od miliona. Wyraża to następująca reguła

```
liczba_srednia(X) :- wieksza_od(X,1000), mniejsza_od(X,1000000).
```

Symbol przecinka, rozdzielający warunki składowe, jest odpowiednikiem logicznej koniunkcji (tj. logicznego "i"). Warunki zachodzenia relacji mogą także być połączone spójnikiem alternatywy (tj. logicznym "lub"). Wyrażenie tego jest nieco bardziej złożone. Musimy bowiem sformułować tyle reguł, ile jest składników alternatywy i każdy składnik umieścić osobnej regule.

Rozpatrzmy pojęcie roku przestępnego. Zgodnie z definicją, rokiem przestępnym jest każdy taki rok, którego numer jest liczbą podzielną (całkowicie) przez 4 i niepodzielną przez 100, lub gdy jest to liczba podzielna przez 400. Zauważmy, że na mocy tej definicji rok 1900 nie był przestępny, ale rok 2000 był rokiem przestępnym. Pojęcie roku przestępnego wyrażają następujące dwie reguły:

```
rok_przestepny(X) :- podzielna_przez(X,4), niepodzielna_przez(X,100).
rok_przestepny(X) :- podzielna_przez(X,400).
```

Ponadto, do wyrażania alternatywnych okoliczności zachodzenia relacji, oprócz reguł można także używać faktów. Przykładowo chcemy stwierdzić, że liczbą parzystą jest każda liczba podzielna przez 2 lub zero. Wyrażają to następujące reguła oraz fakt:

```
liczba_parzysta(X) :- podzielna_przez(X,2).
liczba_parzysta(0).
```

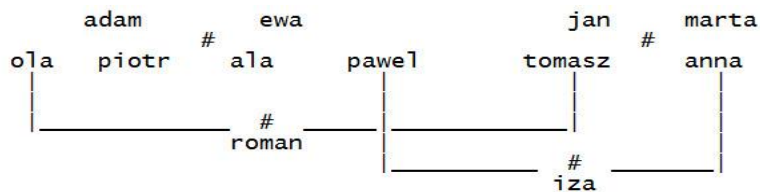
Rozpatrzmy regułę, której warunki zawierają zmienną (lub zmienne) niewystępujące po lewej stronie symbolu :-

```
liczba_o_duzym_dzielniku(X) :- podzielna_przez(X,Y), wieksza_od(Y,1000).
```

Każda taka zmienna (w tym przypadku Y) jest domyślnie poprzedzona frazą "istnieje takie Y, że...". Powyższą regułę możemy odczytać jako stwierdzenie, że liczbą o dużym dzielniku jest każda liczba X pod warunkiem, iż istnieje takie Y, że Y jest dzielnikiem X i Y jest liczbą większą od 1000.

### Przykładowy program w języku Prolog

W języku Prolog, program jest opisem pewnej rzeczywistości, mającym postać zbioru faktów oraz reguł. Poniżej, bierzemy pod uwagę przykładową rzeczywistość złożoną z kobiet i mężczyzn. Do ich oznaczania używa się stałych odpowiadających imionom. Relacje rodzinne, zachodzące między tymi osobami przedstawia schematycznie rysunek 1.



Rys. 1: relacje rodzinne w pewnym zbiorze osób

Osoby występujące nad symbolem # mają dzieci wymienione bezpośrednio pod tym symbolem. Przykładowo, dziećmi pary adam i ewa są ola, piotr, ala oraz pawel. Oto program w języku Prolog opisujący *fragment* rozpatrywanej rzeczywistości.

```
kobieta(ewa).
kobieta(marta).
meczyczna(adam).
meczyczna(piotr).
meczyczna(roman).
rodzic(adam,ola).
rodzic(adam,piotr).
rodzic(adam,pawel).
rodzic(ewa,ola).
rodzic(ewa,piotr).
rodzic(ewa,pawel).
rodzic(ola,roman).
```

Program ów możemy rozszerzyć o poniższą regułę definiującą relację bycia czyimś synem, która zachodzi między każdymi takimi osobami X oraz Y, że Y jest rodzicem X i dodatkowo X jest mężczyzną.

```
syn(X,Y) :-
    rodzic(Y,X),
    meczyczna(X).
```

Inny przykład, to reguła opisująca relację bycia czyimś wnukiem. Relacja ta zachodzi między dowolnymi dwiema osobami X oraz Y pod warunkiem, że istnieje Z takie że Y jest rodzicem Z i Z jest rodzicem X, i dodatkowo X jest mężczyzną.

```
wnuk(X,Y) :-
    rodzic(Y,Z),
    rodzic(Z,X),
    meczyczna(X).
```

## Uruchomienie programu w języku Prolog

W celu uruchomienia programu w języku Prolog, należy wstępnie wprowadzić ten program do odpowiedniego **systemu wykonawczego**. Uruchomienie programu następuje przez skierowanie do systemu tzw. **zapytania**, na które system konstruuje odpowiedź w oparciu o program. Utworzona odpowiedź jest wynikiem końcowym.

Zapytanie ma postać zbliżoną do reguły, przy czym lewa strona symbolu `:-` jest pusta. Wyróżnia się dwa rodzaje zapytań zależnie od tego, czy występują w nich zmienne, czy też nie. Są to odpowiednio:

- **zapytania o rozstrzygnięcie**
- **zapytania o obiekt**

W **zapytaniach o rozstrzygnięcie** nie ma zmiennych; oto przykład:

```
:- rodzic(adam,ola).
```

Jeżeli system wykonawczy stwierdzi na podstawie programu, że zachodzi relacja zawarta w zapytaniu, wówczas jest zwracana odpowiedź twierdząca, czyli wyrażenie `true` albo `yes`, zależnie od implementacji systemu wykonawczego. W przeciwnym przypadku pada odpowiedź przecząca, tj. `false` albo `no`.

Zapytania drugiego rodzaju, czyli **zapytania o obiekt**, zawierają co najmniej jedną zmienną. Rozpatrzmy przykład takiego zapytania:

```
:- rodzic(X,ola).
```

Zgodnie z wcześniejszymi stwierdzeniami, każda zmienna występująca w zapytaniu (w tym przypadku `X`) jest domyślnie poprzedzona frazą "Czy istnieje takie `X`, że...". Odpowiedzią jest dowolny obiekt, dla którego zachodzi relacja opisana w zapytaniu. Należy zauważyć, że na zapytanie sformułowane wyżej istnieją dwie odpowiedzi, tj. `adam` oraz `ewa`. System zwraca te odpowiedzi w formie:

```
X=adam
```

```
X=ewa
```

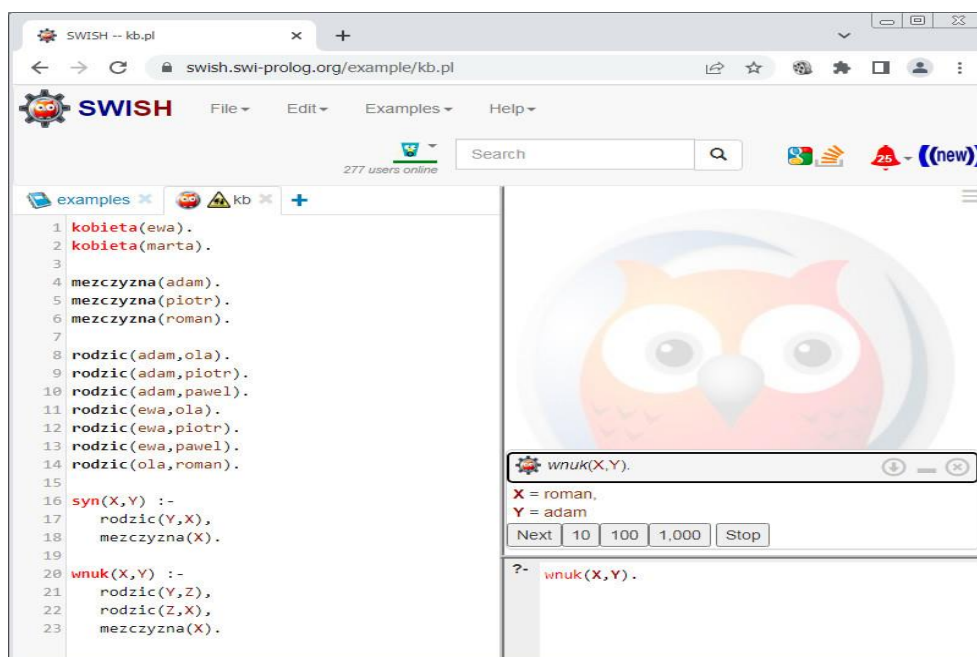
Ściślej ujmując, z przyczyn realizacyjnych system wykonawczy zwraca tylko jedną odpowiedź. W celu uzyskania odpowiedzi kolejnej, użytkownik musi wykonać działania opisane w dalszej części tekstu. Po wyczerpaniu się wszystkich odpowiedzi, prośba o kolejną odpowiedź skutkuje wygenerowaniem wyrażenia `false` albo `no`.

## Przykładowy system wykonawczy języka Prolog

Jako przykładowy system wykonawczy języka Prolog wykorzystamy platformę SWISH, dostępną zdalnie pod adresem URL:

<https://swish.swi-prolog.org/>

Interfejs prezentowanej platformy przedstawia rysunek 2.



Rys. 2: Interfejs systemu SWISH

W celu użycia systemu należy wykonać następujące kroki.

1. Klikając myszką wykonać polecenie „Create a PROGRAM”
2. Wprowadzić program prologowy do okienka-zakładki znajdującego się po lewej stronie, tak jak to przedstawiono na rysunku (nazwa okienka-zakładki dowolna)
3. Umieścić zapytanie w okienku oznaczonym symbolem ?- odgrywającym rolę znaku zachęty. **Uwaga:** pomijamy przy tym symbol :- ograniczając się tylko do podania wyrażenia po lewej stronie tego symbolu. Na rysunku widnieje odpowiednio sformułowane zapytanie o to, czy istnieją X oraz Y takie, że X jest wnukiem Y.
4. Uruchomić program przez naciśnięcie kombinacji klawiszy Ctrl-Enter
5. Odpowiedź pojawi się w oknie nad okienkiem zawierającym zapytanie. W celu uzyskania kolejnej odpowiedzi, należy nacisnąć przycisk Next.

### **Zadania**

1. Uzupełnić przykładowy program opisujący relacje rodzinne tak, aby w pełni odpowiadał rzeczywistości przedstawionej na rysunku 1. W tym celu należy dodać nowe fakty dotyczące relacji kobieta, mezczyzna oraz rodzic.
2. Rozszerzyć program wymieniony w poprzednim zadaniu o reguły opisujące warunki zachodzenia relacji:
  - matka
  - córka
  - rodzeństwo
  - ciocia