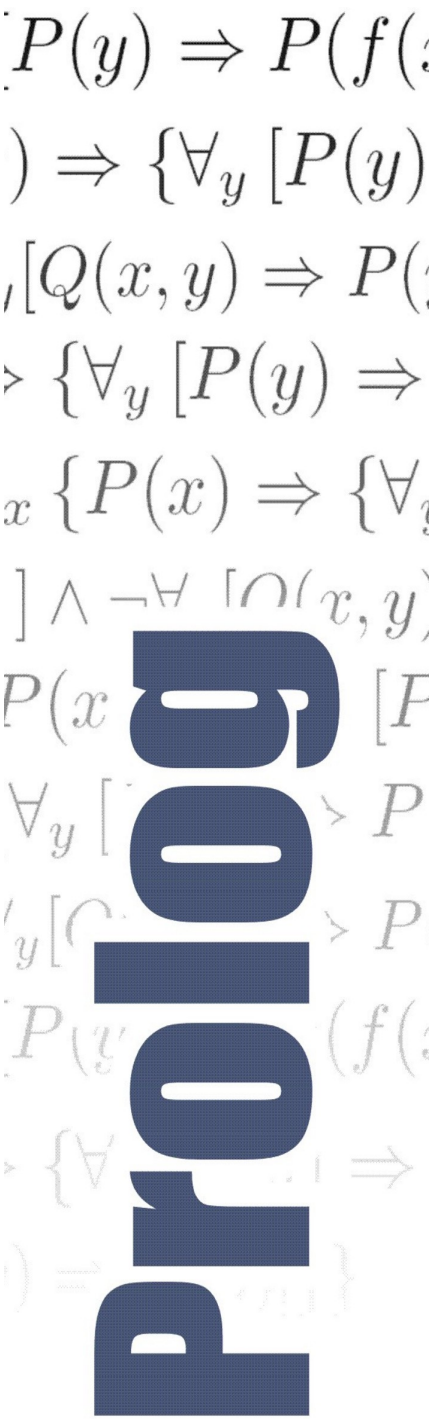
Katedra Wytrzymałości Materiałów
i Metod Komputerowych Mechaniki
www.kwmimkm.polsl.pl

Wydział Mechaniczny Technologiczny
Politechnika Śląska



Inżynieria wiedzy

Instrukcja do zajęć laboratoryjnych

4. Podstawy programowania w języku Prolog

Opracował: mgr inż. Jacek Ptaszny
jacek.ptaszny@polsl.pl

Gliwice 2008

1 Cel ćwiczenia

Wykonując ćwiczenie zapoznasz się z podstawami programowania w języku Prolog.

2 Zanim przejdziemy dalej

Przypomnij sobie:

- Jakimi narzędziami tworzy się systemy ekspertowe?
- Jakie istnieją języki sztucznej inteligencji?
- Jakie są zalety i wady języków sztucznej inteligencji w stosunku do języków systemów ekspertowych?

3 Kilka informacji na temat języka Prolog

Programowanie w języku logiki (ang. logic programming) narodziło się na początku lat siedemdziesiątych ubiegłego wieku, w wyniku prac w zakresie automatycznego dowodzenia twierdzeń i sztucznej inteligencji. Język Prolog (fr. Programmation en Logique) został opracowany w roku 1972 przez Alaina Colmarauera i Philippe'a Roussela, w Marsylii (Francja). Pierwszy kompilator został opracowany przez Davida H. D. Warrena, w Edynburgu (Szkocja). Język ten został wybrany jako podstawa języka maszynowego komputerów piątej generacji, których program budowy rozpoczęto w 1982 roku z inicjatywy rządu japońskiego.

Istnieje wiele dostępnych nieodpłatnie kompilatorów i interpreterów języka Prolog, m.in.:

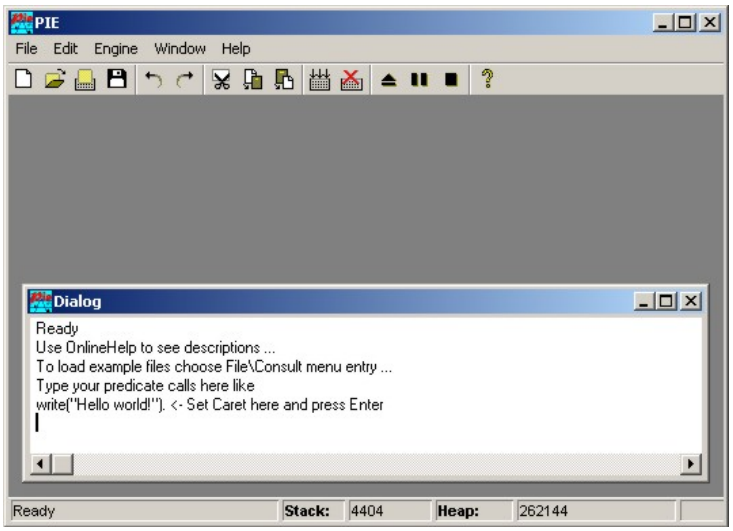
- Amzi! Prolog, www.amzi.com,
- SWI-Prolog, www.swi-prolog.org,
- Visual Prolog, www.visual-prolog.com.

Na zajęciach będziemy używać programu PIE (ang. Prolog Inference Engine, http://www.visual-prolog.com/vip/example/pie/pie.htm). Do samodzielnych ćwiczeń w domu możesz wykorzystać również inny program.

4 Zaczynamy!

4.1 Uruchom program PIE

Zlokalizuj plik PIE.exe i uruchom program. Na ekranie pojawi się okno programu wraz z oknem dialogowym:



4.2 Utwórz nowy plik programu

Wybierz polecenia **File -> New** w menu głównym. W oknie głównym zostanie otwarte nowe okno edycyjne.

4.3 Zapisz plik

Wybierz polecenia **File -> Save As..** i zapisz plik w wybranym katalogu, ustalając jego nazwę.

4.4 Wprowadź fakty

W oknie edycyjnym wpisz:

```
jest_ojcem(adam, piotr).
jest_ojcem(piotr, andrzej).
jest_ojcem(andrzej, marek).
```

 Program napisany w Prologu stanowi ciąg faktów i reguł, a więc bazę wiedzy.

 Pierwszy fakt należy rozumieć następująco: "Adam jest ojcem Piotra". Podobnie w przypadku pozostałych faktów.

 Powyższe fakty zostały wyrażone za pomocą **predykatów**. Predykat określa tutaj relację pomiędzy obiektami. W tym przypadku predykat to relacja "jest ojcem". Ścisłą definicję predykatu, zaczerpniętą z książki [4], podano poniżej:

> ***PREDYKAT*** *(ang. predicate)*
> *(1) Funktor zdaniotwórczy od argumentów nazwowych. Np. słowo "świeci" w zwrotach "Księżyc świeci", "x świeci" itp.*
> *(2) Wyrażenie złożone z funktora zdaniotwórczego od argumentów nazwowych oraz ze zmiennych nazwowych; inaczej: funkcja zdaniowa argumentów nazwowych, czyli funkcja propozycjonalna. Np. "x świeci", "x = y", "x leży między y i z".*
> *(3) Wyrażenie, które opisuje pewną własność lub relację. [...]*
> *Predykat opisujący własność nazywa się predykatem jednoargumentowym (ang. monadic predicate), predykat opisujący relację nazywa się dwu-, trzy- lub więcej-argumentowym (ang. dyadic predicate, triadic predicate etc.), w zależności od liczby członów danej relacji.*
> *Dział logiki dotyczący predykatów nazywa się rachunkiem predykatów lub rachunkiem kwantyfikatorów.*

 Argumentami predykatów są **termy**. Termy są stałymi, zmiennymi lub strukturami. W powyższych faktach termami są **stałe**: *adam, piotr, andrzej* oraz *marek*.

 Stałe służą do nazywania obiektów lub relacji. Dzielimy je na atomy i liczby całkowite. Atomy rozpoczynają się od małych liter. Dozwolone jest użycie znaku podkreślenia. Jeśli w wyrażeniu atomowym musi na początku wystąpić duża litera, cyfra lub inne znaki specjalne, to taki atom musi być ujęty między znakami apostrofu.

 Pamiętaj, że definicje faktów i reguł należy kończyć kropką.

 Pamiętaj o częstym zapisywaniu wyników swojej pracy.

4.5 Wprowadź reguły

Do treści programu dopisz:

```
jest_dziadkiem(X, Y) :- jest_ojcem(X, Z), jest_ojcem(Z, Y).
jest_pradziadkiem(X, Y) :- jest_ojcem(X, Z), jest_dziadkiem(Z, Y).
```

 Powyższe reguły należy rozumieć następująco:

- X jest dziadkiem Y jeśli X jest ojcem Z oraz Z jest ojcem Y.
- X jest pradziadkiem Y jeśli X jest ojcem Z oraz Z jest dziadkiem Y.

 Ogólna postać reguł jest następująca:

$$P_1(A_1, A_2, ..., A_k) \ :- \ P_2(B_1, B_2, ..., B_l), ..., P_n(C_1, C_2, ..., C_m).$$

gdzie P_1 jest konkluzją, natomiast $P_{2,...,n}$ są warunkami, które muszą być spełnione jednocześnie. $A_{1,...,k}$, $B_{1,...,l}$ oraz $C_{1,...,m}$ są argumentami predykatów. Symbol :- pełni rolę słowa "jeśli". Wyrażenia o powyższej postaci nazywane są **klauzulami Horna**.

 W definicji reguł użyto nazw zmiennych: X, Y oraz Z. Zasięg zmiennej nie wykracza poza regułę, w której zmienna ta została użyta.

 Nazwy zmiennych zaczynają sie dużymi literami.

4.6 Zapisz program

...za pomocą polecenia **File -> Save**.

4.7 Wczytaj program do obszaru roboczego

Skorzystaj z polecenia **consult**. W oknie poleceń wpisz:

```
consult('plik.pro').
```

wstawiając zamiast *plik.pro* odpowiednią nazwę pliku.

 Możesz również wybrać polecenie **File -> Consult** z menu głównego.

4.8 Zadaj pytanie

Aby dowiedzieć się kto jest dziadkiem Marka, wpisz w oknie poleceń:

```
jest_dziadkiem(X, marek).
```

a następnie naciśnij klawisz Enter. Program powinien odpowiedzieć:

```
X = piotr
1 solution
```

 W języku Prolog postawione pytanie nazywa się **celem**.

 System poszukuje wszystkich wartości zmiennych występujących w predykacie definiującym cel spełniających ten predykat, i wyświetla je. Proces ten nazywany jest **unifikacją**.

4.9 Zadaj kolejne pytania

Wpisz następujące pytania, zastanów się nad ich znaczeniem i przeanalizuj odpowiedzi których udzieli system:

```
jest_ojcem(adam, X).
jest_ojcem(X, andrzej).
jest_ojcem(X, Y).
jest_dziadkiem(X, marek).
jest_pradziadkiem(adam, X).
```

 W przypadku dwóch ostatnich pytań nie ma na liście faktów żadnego predykatu spełniającego bezpośrednio postawione cele. Procedura wnioskująca sformułowała odpowiedzi korzystając z reguł istniejących w bazie wiedzy, wprowadzonych w punkcie 4.5.

4.10 Sprawdź, w jaki sposób program osiąga rozwiązanie

Wybierz polecenie **Engine -> Trace calls** w menu głównym i zadaj jeszcze raz pytanie:

```
jest_dziadkiem(adam, X).
```

Przeanalizuj sposób wnioskowania (kolejność wywoływanych predykatów).

 Celem wykonania programu w języku Prolog jest wykazanie prawdziwości hipotezy i znalezienie zmiennych występujących w predykacie reprezentującym hipotezę. Procedura wnioskowania działa w sposób regresywny. Oznacza to, że poszukiwanie celu głównego rozpoczyna się od przeszukania listy faktów. Jeśli operacja ta zakończy się niepowodzeniem, procedura poszukuje celu w konkluzjach reguł. Po odnalezieniu celu w konkluzji pewnej reguły, warunki tej reguły stają się podcelem i dalej wnioskowanie prowadzone jest w sposób rekurencyjny. Gdy wszystkie warunki reguły są spełnione jest ona uruchamiana, a fakt występujący w jej konkluzji jest umieszczany na liście faktów.

 Procedura wnioskująca korzysta z mechanizmu nawracania (ang. backtracking). Jest to pojęcie związane z przeszukiwaniem przestrzeni stanów, odwzorowanej za pomocą drzewa, w głąb. Jeśli bieżący podcel nie zostanie udowodniony, to procedura powraca do podcelu, bezpośrednio poprzedzającego bieżący podcel. Dzięki temu mechanizmowi Prolog znajduje wszystkie możliwe rozwiązania.

4.11 Sprawdź, czy Adam jest dziadkiem

Zadaj pytanie:

```
jest_dziadkiem(adam, _).
```

 Zamiast zmiennej można użyć znaku podkreślenia, jeśli wartość argumentu predykatu nie jest ważna.

System powinien odpowiedzieć:

```
True
1 Solution
```

4.12 Przeprowadź śledztwo

Napisz program rozwiązujący zagadkę kryminalną[1]. Do utworzenia bazy wiedzy wykorzystaj wymienione poniżej zdania.

Dla ułatwienia, relacje które posłużą do nazwania predykatów oraz konkluzji oznaczono pojedynczym podkreśleniem, natomiast nazwy obiektów oznaczono podwójnym podkreśleniem.

Nazwy predykatów i obiektów nie muszą dokładnie odpowiadać podkreślonym wyrazom, ale powinny jednoznacznie oddawać ich sens i znaczenie.

Oto przykład predykatu dla pierwszego zdania:

```
osoba(tomasz, 55, mezczyzna, stolarz).
```

Predykaty:

1. Tomasz jest osobą 55-letnią i jest stolarzem.
2. Krzysztof jest osobą 25-letnią i jest piłkarzem.
3. Krzysztof jest osobą 25-letnią i jest rzeźnikiem.
4. Piotr jest osobą 25-letnią i jest złodziejem.
5. Anna jest osobą 39-letnią i jest chirurgiem.
6. Anna miała romans z Piotrem.
7. Anna miała romans z Krzysztofem.
8. Agnieszka miała romans z Piotrem.
9. Agnieszka miała romans z Tomaszem.
10. Agnieszka została zamordowana.
11. Agnieszka została zamordowana przez uderzenie kijem golfowym.
12. Tomasz był pobrudzony krwią.
13. Agnieszka była pobrudzona krwią.
14. Krzysztof był pobrudzony błotem.
15. Piotr był pobrudzony czekoladą.
16. Anna była pobrudzona krwią.
17. Tomasz posiada sztuczną nogę.
18. Piotr posiada rewolwer.
19. Uderzenie sztuczną nogą powoduje podobne obrażenia jak uderzenie kijem golfowym.

[1] Opracowano na podstawie przykładu z książki [3]

20. Uderzenie nogą od stołu powoduje podobne obrażenia jak uderzenie kijem golfowym.
21. Uderzenie nożyczkami powoduje podobne obrażenia jak uderzenie nożem.
22. Uderzenie butem piłkarskim powoduje podobne obrażenia jak uderzenie kijem golfowym.

 Oto przykład reguły dla pierwszego ze zdań określających reguły:

```
prawdopodobnie_posiada(X, buty_pilkarskie) :- osoba(X, _, _, pilkarz).
```

Reguły:

1. Osoba prawdopodobnie posiada buty piłkarskie jeśli jest piłkarzem.
2. Osoba prawdopodobnie posiada nożyczki jeśli jest chirurgiem.
3. Osoba prawdopodobnie posiada narzędzie zbrodni jeśli je posiada.
4. Osoba jest podejrzana jeśli prawdopodobnie posiada narzędzie, którego uderzenie powoduje podobne obrażenia jak uderzenie narzędziem, którym została uderzona Agnieszka.
5. Motywem osoby będącej mężczyzną jest zazdrość jeśli mężczyzna miał romans z Agnieszką.
6. Motywem osoby będącej kobietą jest zazdrość jeśli kobieta miała romans z tym samym mężczyzną co Agnieszka.
7. Motywem osoby będącej mężczyzną są pieniądze jeśli jest on złodziejem.
8. Mordercą jest osoba podejrzana, posiadająca motyw i która była pobrudzona tą samą substancją co Agnieszka.

 Odpowiedz na pytania:

- Kto jest podejrzany o morderstwo?
- Jakie motywy zbrodni miały poszczególne osoby?
- Kto jest mordercą?

5 Podsumowanie

Wykonując ćwiczenie nauczyłeś(aś) się:

- formułować predykaty,
- formułować reguły,
- formułować cele,

w języku Prolog.

Literatura

[1] Bramer M., *Logic programming with Prolog*. Springer Science + Business Media, 2005.

[2] Cholewa W., Pedrycz W., *Systemy doradcze*. Wydawnictwo Politechniki Śląskiej, Gliwice 1987.

[3] Kendal S., Creen M., *An Introduction to Knowledge Engineering*. Springer-Verlag, London 2007.

[4] Marciszewski W. (red.), *Mała encyklopedia logiki*, Zakład Narodowy Imienia Ossolińskich - Wydawnictwo, Wrocław - Warszawa - Kraków 1970.

[5] Michalik K., *Prolog*. Aitech, Katowice, www.aitech.pl.

[6] Mulawka J. J., *Systemy ekspertowe*. WNT, Warszawa 1996.

[7] Russel S., Norvig P., *Artificial intelligence: A Modern Approach*. Prentice Hall, 2002.

[8] Rutkowski L., *Metody i techniki sztucznej inteligencji*. WNT, Warszawa 2005.