Wstęp do Prolog

Wykonali:
Rafał Denkiewicz, Rafał Rutkowski,
Kacper Rękawiecki

Prolog

Prolog to komputerowy język programowania. Jego początki sięgają roku 1970, od tego czasu używano go w aplikacjach związanych z przetwarzaniem symbolicznym, w takich dziedzinach, jak:

-relacyjne bazy danych,
-logika matematyczna,
-rozwiązywanie problemów abstrakcyjnych,
-przetwarzanie języka naturalnego,
-automatyzacja projektowania,
-symboliczne rozwiązywanie równań,
-analiza struktur biochemicznych,
-różne zagadnienia z dziedziny sztucznej inteligencji.

Prolog

Programowanie w Prologu nie polega na opisywaniu algorytmu, jak to ma miejsce w tradycyjnych językach programowania. Zamiast tego programiści Prologu zajmują się raczej formalnymi relacjami i obiektami związanymi z danym problemem, badając, które relacje są „prawdziwe” dla szukanego rozwiązania. Tak więc Prolog może być uważany za język opisowy i deklaratywny. Programowanie w Prologu polega przede wszystkim na opisaniu znanych faktów i relacji dotyczących problemu, w mniejszym stopniu na podawaniu kolejnych kroków algorytmu. Kiedy programuj emy w Prologu, sposób pracy komputera częściowo wynika z deklaratywnej semantyki Prologu, częściowo z tego, że Prolog na podstawie danego zbioru faktów może wnioskować nowe fakty, a jedynie częściowo na podstawie jawnie podanych przez programistę instrukcji sterujących.

Obiekty i relacje

Prolog to język programowania używany do rozwiązywania problemów dotyczących obiektów i relacji między nimi.

Prolog stanowi praktyczną i wydajną implementację szeregu aspektów „inteligentnego”wykonywania programu: braku determinizmu, równoległości i wywoływania procedur według wzorca. Prolog zawiera ujednoliconą strukturę danych, term, na bazie której tworzone są wszystkie dane oraz same programy Prologu. Program prologowy składa się ze zbioru klauzul, a każda klauzula to albo fakt opisujący pewną informację, albo reguła mówiąca, jak rozwiązanie można powiązać z danymi faktami. Reasumując, kiedy mówimy o obiektach w Prologu, nie chodzi nam o struktury danych dziedziczące pola i metody z klasy, ale o byty, które można opisać termami.

Obiekty i relacje

Obiekt w sensie Prologu jest czymś, co możemy nazwać bytem. Nie definiujemy z czego się on składa i co można z nim zrobić, ale jaki jest. Dodatkowo, dla każdego obiektu definiujemy relacje jakim obiekt ten podlega. Przy pomocy obiektów opisujemy interesujący nas wycinek świata. Działanie programu prologowego objawia się możliwością stawiania pytań związanych z uprzednio opisanym światem.

Najprostszym sposobem opisu świata (problemu), jest podanie faktów z nim związanych, jak na przykład:

```
ciezszy(pomarancz,jablko).
ciezszy(jablko,mandarynka).
ciezszy(arbuz,pomarancz).
ciezszy(jablko,winogrono).
```

Powyższe fakty stwierdzają, że
- ciezszy(pomarancz,jablko). – pomarańcz jest cięższa od jabłka,
- ciezszy(jablko,mandarynka). – jabłko jest cięższe od mandarynki,
- itd.

Obiekty i relacje

Powyżej określiliśmy kilka obiektów (pomarancz, jablko, itd) i powiązaliśmy je między sobą relacją cięższy wyrażającą, który z obiektów jest cięższy od innych. Istotne jest to, że nadal nie jest nam znana masa żadnego z obiektów - one po prostu nie posiadają cech.

Po „uruchomieniu" takiego programu, możemy zadawać pytania związane z rzeczywistością jaką opisuje:
?- ciezszy(pomarancz,jablko).
Yes

W ten oto sposób otrzymujemy twierdzącą odpowiedz na pytanie: Czy pomarańcz jest cięższa od jabłka?. Będąc precyzyjniejszym, to pytanie brzmi:Czy wiadomo coś na temat tego, że pomarańcz jest cięższa od jabłka?. Jest to istotne rozróżnienie, gdyż wówczas reakcję
?- ciezszy(winogrono,arbuz).
No

należy odczytywać jako: Nic nie wiadomo na temat tego, że winogrono jest cięższe od arbuza. Nie oznacza to jednak, że tak nie jest.
Taka interpretacja jest właściwsza, co pokazuje kolejny przykład:
?- ciezszy(arbuz,winogrono).
No

Obiekty i relacje

Uwaga:

Obiekty mogą pozostawać we wzajemnych zależnościach wyrażanych za pomocą implikacji, której poprzednikiem i następnikiem są określone formuły zdaniowe. Możemy zatem definiować jedne obiekty poprzez inne w ten sposób, że z prawdziwości pewnych predykatów określonych na znanych już obiektach i na tym definiowanym wynika prawdziwość jakiegoś predykatu określonego (między innymi) na obiekcie definiowanym.

Struktura i składnia

Praca z Prologiem składa sie zwykle z następujących etapów:

1. Definiowanie obiektów poprzez definiowanie faktów dotyczących obiektów i związków między nimi.
2. Definiowanie reguł dotyczących obiektów i związków miedzy nimi.
3. Zapytania o obiekty i związki miedzy nimi.

Podczas zapisywania programu stosujemy nastepującą konwencje:

1. Nazwy relacji i obiektów musza zaczynać sie małymi literami.
2. Nazwy rozpoczynajace sie od dużej litery oznaczają zmienne.
3. Najpierw zapisujemy relacje, a potem, rozdzielone przecinkami i ujęte w nawias okragły, obiekty których ona dotyczy.
4. Nazwy obiektów wystepujacych w nawiasach nazywamy **argumentami**.
5. Nazwe relacji znajdującej sie przed nawiasem nazywamy **predykatem**.

Struktura i składnia

6. Nie można jako predykatu użyć zmiennej. Innymi słowy, nie można się dowiedzieć jaka relacja
   łączy obiekty jaś i małgosia *X(jas,malgosia).*
7. Fakt i regułę kończymy znakiem kropki.
8. Kolejność obiektów umieszczonych w nawiasie jest dowolna ale trzeba stosować ją konsekwentnie. O ile bowiem dobrze znanym faktem jest to, że Ala lubi swojego kota, to nie oznacza to, że kot ten lubi Ale.
9. Zbiór faktów i reguł nazywamy baza danych.
10. Składnia reguły jest nastepująca <lewaCzesc> :- <prawaCzesc>, co mozemy czytac jako *lewaCzęść zachodzi (jest prawdą), gdy zachodzi prawaCzęść (jest prawda)*, gdzie
    – <lewaCzęść> to predykat i ewentualne argumenty umieszczone w nawiasach okrągłych, np. *lubi(X,Y)*
    silnia(0,X)
    – <prawaCzęść> to jedno lub więcej wyrażeń atomowych połączonych operatorami logicznymi: i (,) oraz lub (;) i poprzedzonych ewentualnie operatorem negacji (\+).

Praca z programem - zapytania

Praca z programem Prologowym także odbywa się inaczej niż w innych językach programowania. Raczej trudno mówić o uruchamianiu programu i jego działaniu jako samodzielnej i niezależnej aplikacji, gdyż programy Prologu z natury są raczej interakcyjne. Bardziej adekwatnym kreśleniem zamiast uruchamianie wydaje się być formułowanie zapytań lub też interakcyjny tryb zapytanie–odpowiedź.

Zapisany program wczytujemy poleceniem (znaki ?-są tzw. znakiem zachęty):

?- [plikBezRozszerzenia].

i od tego momentu możemy formułować zapytania, np.

?- posiada(piotr,ksiazka).

Zapytanie to, w języku naturalnym brzmiałoby: Czy Piotr ma książkę?

Na potrzeby przetwarzania przez Prolog należy czytać je jednak trochę inaczej: Czy istnieje fakt mówiący, że Piotr ma książkę?

Praca z programem - zapytania

Kiedy mamy już jakies fakty, możemy zadawać dotyczące ich zapytania. W Prologu zapytanie wyglada tak samo jak fakt, ale umieszcza się przed nim specjalny symbol - pytajnik i minus (?-).

?- posiada(maria,gazeta).

Każde wpisane pytanie musi być zakończone kropką, po której naciskamy klawisz Enter celem wysłania go do systemu Prolog.

Prolog przeszuka całą dostępną bazę wiedzy (w postaci faktów i reguł) i jeśli zostanie znalezione coś co pasuje do zapytania i zwraca przy tym wartość logiczną prawda, wówczas zostanie zwrócona odpowiedź yes; w przeciwnym razie no. Raz jeszcze zaznaczamy, że no nie oznacza „nie”, ale „nie wiem”. Sam proces przeszukiwania, odbywa się linia po linii, czyli fakty i reguły rozpatrywane są w kolejności ich umieszczenia w pliku.

Zamiast szukać odpowiedzi na pytanie “Czy Piotr ma książkę?”, możemy chcieć zapytać “*Co ma Piotr?”,* co w języku Prologu bardziej należy czytać jako: Jeśli Piotr ma X, to X jest tym czego szuka.

?- posiada(piotr,X).

Praca z programem - zapytania

Majac więcej faktów:
lubi(jas,piernik).
lubi(jas,malgosia).
lubi(malgosia,cukierek).
lubi(malgosia,piernik).

możemy konstruować zapytania złożone, np.

?- lubi(jas,malgosia), lubi(malgosia,jas).
czyli: Czy prawdą jest, że Jaś lubi Małgosię i Małgosia lubi Jasia? lub

?- lubi(jas,X), lubi(malgosia,X).
czyli: Szukam tego wszystkiego co lubi zarówno Jas jak i Małgosia.

Odpowiedź na pytanie o to co lubi Jaś lub Małgosia uzyskamy zapytaniem:
?- lubi(jas,X); lubi(malgosia,X).

Fakty

Najpierw omówimy fakty opisujace obiekty. Załóżmy, że chcemy w Prologu zanotowac fakt, że „Jan lubi Marie”. Fakt ten dotyczy dwóch obiektów, Jana i Marii, oraz zawiera relacje „lubienia”. W prologu fakty zapisuje sie w postaci:

lubi(jan,maria)

Trzeba pamietac o kilku rzeczach:

- Nazwy wszystkich relacji i obiektów musza sie zaczynac małymi literami, jak powyzej: lubi, jan, maria.

1 Najpierw zapisuje sie relacje, a potem jej obiekty rozdzielone przecinkami i ujete w nawias okragły.

1 Fakt musi sie konczyc kropka ().

Fakty

Relacje moga miec dowolna liczbe argumentów. Jesli chcemy zdefiniowac predykat *gra*, w którym podamy dwóch graczy i nazwe gry, bedziemy potrzebowali trzech argumentów. Oto przykład takiego faktu:

gra(jan,maria,futbol)

Terminologia:

- Nazwy obiektów wystepujace w nawiasach nazywamy argumentami.
- Nazwe relacji znajdującą sie przed nawiasem nazywamy predykatem. Wobec tego predykat *gra* ma trzy argumenty.
- Zbiór faktów nazywamy bazą danych.
- Nazwy obiektów i relacji sa całkowicie dowolne.

Termy

Program Prologu składa sie z termów. Wyrózniamy cztery rodzaje termów: atomy (ang. atoms), liczby (ang. numbers), zmienne (ang. variables) i termy złozone (ang. compound terms). Atomy i liczby wspólnie okreslane sa jako stałe (ang. constants). Zbiór złozony z atomów i termów złozonych nazywany jest tez zbiorem predykatów1. Kazdy term zapisywany jest jako ciag znaków pochodzacych z nastepujacych czterech kategorii:

- duze litery: A-Z
- małe litery: a-z
- cyfry: 0-9
- znaki specjalne: % + - * / \ ~ ^ < > : . ? @ # $ &

Zbiór ten uzupełnia znak podkreslenia (_), który zwykle traktowany jest jak litera.

Termy

1. **Atomy**

Atom jest ciagiem znaków utworzonym z

- małych i duzych liter, cyfr i znaku podkreslenia z zastrzezeniem, ze pierwszym znakiem musi byc mała litera, np.

  jaś, a, aLA, x_y_z, abc
- dowolnego ciagu znaków ujetego w apostrofy, np.

  'To tez jest atom'
- symboli, np. ?- lub :- .

Termy

2. Liczby

W SWI-Prologu dostepne sa zarówno liczby całkowite jak i rzeczywiste

-17, 23, 99.9, 123e-3

3. Zmienne

Zmienna jest ciagiem znaków utworzonym z małych i duzych liter, cyfr i znaku podkreslenia z zastrzezeniem, ze pierwszym znakiem musi byc duza litera lub znak podkreslenia, np.

X, Kto, _123, X_1_2, _

Termy

3. Zmienne cd.

Ostatni z wymienionych przykładów, pojedynczy znak podkreslenia, to tak zwana **zmienna anonimowa**. Korzystamy z niej zawsze wtedy, gdy interesuje nas tylko czy cos jest prawdą, ale zupełnie nie interesuje nas co, np.

Czy ktoś lubi Jasia?

```
?- lubi(_,jas).
```

Należy pamiętać, że wielokrotnym wystąpieniom zmiennej anonimowej w jednym wyrażeniu mogą być przypisane różne wartości.

```
?- a(1,2)=a(X,Y).
X = 1,
Y = 2
?- a(1,2)=a(X,X).
No
?- a(1,2)=a(_,_).
Yes
```

Termy

Przykład:
Czy ktos lubi Jasia?
?- lubi(_,jas).

4. Termy złozone

Term złozony, inaczej struktura, to obiekt złozony z innych obiektów, czyli atomów, liczb, zmiennych a takze innych termów złozonych. Termy złozone maja postać
f(arg_1,...,arg_n)
gdzie arg_1,. . . ,arg_n sa termami, natomiast f jest atomem (nazwa relacji). Korzystajac z mozliwosci zagniezdzania innych termów w termach złozonych, mozemy lepiej opisac interesujacy nas problem.

Termy

Przykład
Fakty
posiada(piotr,auto).
posiada(marcin,auto).
pozwalaja jedynie stwierdzic, ze obiekty piotr i marcin zwiazane sa relacja posiada z obiektem auto, czyli mówiac „normalnie”, Piotr i Marci maja auto. Trudno powiedziec jakie to jest auto i czy przypadkiem to nie jest to samo auto. Zapisując te fakty inaczej
posiada(piotr,auto(nissan,almera)).
posiada(marcin,auto(fiat,punto)).
maAuto(X) :- posiada(X,auto(_,_)).

Termy

Wciąż mamy możliwość dowiedzenia się czy obaj mają auto, ale jeśli będziemy chcieli, to możemy zapytać o coś bardziej szczegółowego, np. marka i model.

?- maAuto(piotr).
Yes
?- posiada(piotr,auto(X,Y)).
X = nissan,
Y = almera

Reguły

W Prologu reguł używa sie do zapisania, że jakis fakt zależy od grupy innych faktów. W jezyku polskim do stworzenia reguły używamy słówka „jesli”, na przykład:

Używam parasola, jesli pada.

Jan kupuje wino, jesli jest ono tansze od piwa.

Reguł używa sie też do zapisywania definicji, na przykład:

X jest ptakiem jesli:

X jest zwierzeciem i

X ma pióra.

Reguły

Reguła składa sie z głowy i tresci, które połaczone sa symbolem składajacym sie z dwukropka i myslnika (!). Głowa reguły składa się tylko z jednego predykatu, natomiast ciało reguły może być koniunkcją dowolnej liczby warunków (oddzielonych przecinkami – w Prologu oznaczają one logiczne AND).

Zapiszmy w Prologu regułę, że maria lubi coś, jeśli tylko to cos jest tanie i zdrowe.

lubi(maria, X) :- tanie(X), zdrowe(X).

Aby spełniona była przesłanka reguły, spełnione muszą być wszystkie jej podcele.

Unifikacja (dopasowanie)

Operator dopasowania oznaczany jest jako „=”. Mówimy, że dwa termy można zunifikować, gdy można je do siebie dopasować. Dopasowanie dwóch termów X i Y zachodzi gdy:

- jeśli X jest zmienną a Y dowolnym termem. W takiej sytuacji pod X podpisywana jest wartość Y, np.

  ?-jedzie(jan, auto) = X.

- obydwa termy są identycznymi stałymi (liczby całkowite i atomy zawsze są sobie równe), np.

  papier = papier

  1203 = 1203

- X i Y są termami złożonymi i jednocześnie mają taką samą liczbę argumentów, ten sam symbol funkcyjny oraz zachodzi dopasowanie pomiędzy wszystkimi argumentami obu termów.

Unifikacja termów

Oto kilka przykładów unifikacji termów:

- ?- a = a.
  Yes
- ?- a = X.
  X = a
  Yes
- ?- a = b.
  No
- ?- f(X, a) = f(b, Y).
  X = b
  Y = a
  Yes

Czym są w Prologu stałe,zmienne.

Stała: Stałe w Prologu, podobnie jak w innych językach programowania, nie mogą zmieniać swojej wartości. Zapisuje się je jako ciągi liter, cyfr i znaku podkreślenia rozpoczynające się od małej litery. Szczególnymi stałymi w Prologu są liczby całkowite i rzeczywiste. Dowolny ciąg znaków może być w Prologu również stałą ale w tym celu należy go ująć między cudzysłowy.

Stałe dzielą się na:

- liczby;
- atomy (są zbudowane z dowolnych symboli ewentualnie ujętych w pojedynczy cudzysłów przy czym zawsze zaczynają się małą literą)

Zmienna: relacja może być rozumiana jako ’funkcja’ określona na pewnym zbiorze obiektów. Wówczas przez zmienną rozumiemy dowolny ale nie ustalony element z dziedziny jakiejś relacji. Nazwy zmiennych są atomami rozpoczynającymi się zawsze wielką literą. Np.: weźmy relację student/2, która jest prawdziwa jeśli argumentami są nazwisko studenta i nazwa przedmiotu na egzaminie z którego ściągał. Wstawiając do niej zmienne (Kowalski, Origami) otrzymamy predykat prawdziwy dla par utworzonych przez konkretne nazwisko i odpowiadający mu przedmiot(y).

Czym są w Prologu stałe,zmienne.

Zmienna cd:

Jeśli bazą programu jest zbiór:

```
student(a,teoria_pola).
student(b,mechanika_kwantowa).
student(c,wychowanie_fizyczne).
student(d,szkolenie_bhp).
student(d,geometria_różniczkowa).
```

to w wyniku śledztwa przeprowadzonego w Prologu otrzymamy:

```
1 ?- student(Kowalski,Origami).
Kowalski = a,
Origami = teoria_pola ;
Kowalski = b,
Origami = mechanika_kwantowa ;
Kowalski = c,
Origami = wychowanie_fizyczne ;
Kowalski = d,
Origami = szkolenie_bhp ;
Kowalski = d,
Origami = geometria_różniczkowa ;
```

Operatory

Mamy do dyspozycji operatory pozwalające porównywać liczby:

- **X=:=Y (**X i Y są tą samą liczbą)

  ?- 2=:=2.
  true.
- **X=\=Y** (X i Y są różnymi liczbami)

  ?- 2=\=3.
  true.
- **X<Y** (X jest mniejsze od Y)

  ?- 1<2.
  true.
- **X>Y** (X jest większe od Y)

  ?- 2>3.
  false.
- **X=<Y** (X jest mniejsze lub równe Y)

  ?- 2=<3.
  true.
- **X>=Y** (X jest większe lub równe Y)

  ?- 4>=1.
  false.

Dziękujemy za uwagę!

Zadanie 1:

Proszę przyjrzeć się poniższemu, prostemu programowi (klasyczny przykład „Rodzina”).

```
kobieta(ola).
    kobieta(lila).

    mezczyzna(tomek).
    mezczyzna(max).

    rodzic(ola,max).
    rodzic(tomek,max).
    rodzic(tomek,lila).
```

Proszę zapoznać się z propozycjami poniższych pytań i zapisać je w Prologu.

- Kto jest mężczyzną?
- Czy Tomek jest mężczyzną?
- Czy Ola jest rodzicem Max-a?
- Czyim rodzicem jest Ola?

Zadanie 2:

Utworzyć plik zawierający następujące fakty:
- Piotr lubi góry.
- Arek jest wysoki.
- Piotr dał Hani tulipany.
- Michał podróżuje autobusem do Paryża.
- Trojkąt, kwadrat, okrąg to figury.

Zadanie 3:

Utworzyć plik zawierający reguły opisujące następujące zdania o świecie:
- Każdy człowiek je tylko mięso lub ser lub owoce.
- Jeśli X jest babcią Y, to Y jest wnukiem lub wnuczką.
- Dwie osoby są rodzeństwem, jeśli mają tych samych rodziców.
- X jest figurą jeśli jest trojkatem lub kwadratem lub okręgiem.

Zadanie 4:

Napisać program obliczający silnię.

Zadanie 5:

Napisać program obliczający n-ty wyraz ciągu Fibonacciego.