



Podstawy sztucznej inteligencji

INSTRUKCJA DO LABORATORIUM NR 11:
PROJEKTOWANIE SYSTEMU EKSPERCKIEGO W
ŚRODOWISKU PROLOG

1 Cel laboratorium

Celem laboratorium jest wprowadzenie do środowiska Prolog oraz zaprojektowanie podstawowego systemu eksperckiego w tym środowisku.

2 Podstawy programowania w języku Prolog

W tej sekcji zostanie przedstawiona podstawowa logika języka programowania Prolog. Prolog jest językiem programowania logicznego, który jest często stosowany do tworzenia systemów eksperckich. **Systemy eksperckie** to programy komputerowe, które używają wiedzy i reguł wnioskowania do rozwiązywania problemów, które zwykle wymagają ludzkiej ekspertyzy. W Prologu systemy te działają na zasadzie faktu i reguł.

2.1 Fakty i reguły w języku Prolog

W Prologu fakty są podstawowymi stwierdzeniami, które opisują właściwości obiektów lub relacje między obiektami. Na przykład, fakt jest ptakiem(kanarek). mówi, że kanarek jest ptakiem. Fakty te są używane do budowania bazy danych, z której system ekspercki może korzystać. Reguły w Prologu służą do reprezentowania logiki i procedur wnioskowania. Są one formą warunkowych stwierdzeń, które pozwalają na wnioskowanie nowych faktów na podstawie już istniejących. Reguła składa się z głowy i ciała, gdzie ciało (zawierające warunki) musi być prawdziwe, aby głowa (wniosek) również była prawdziwa. Na przykład reguła może latać(X) :- jest ptakiem(X), nie jest pingwinem(X). oznacza, że "X może latać, jeśli X jest ptakiem i X nie jest pingwinem".

Na przykład, jeśli zapytasz system, czy kanarek może latać, Prolog sprawdzi fakty i reguły, aby stwierdzić, czy kanarek spełnia warunki wymagane do lotu. Jeśli wszystkie warunki reguły są spełnione, system wnioskuje, że kanarek może latać.

2.2 Środowisko Prolog, tworzenie skryptu i kompilacja

Całe oprogramowanie do środowiska Prologa możemy ściągnąć z linku: <https://www.swi-prolog.org/download/stable>. Wybieramy wersję zgodną z naszym systemem operacyjnym (32 lub 64-bitowy).

Po zainstalowaniu uruchamiamy aplikację **SWI-Prolog**. Ukaże nam się konsola, w której będziemy pracować. Skrypt Prolog tworzymy w dowolnym edytorze tekstowym i zapisujemy go z rozszerzeniem .pl. Aby załadować skrypt Prologa podajemy ścieżkę do skryptu. Przykład poniżej:

```
1      ['C://Users//Jola//Desktop//expert-sys.pl'] .
```

Uruchamiamy diagnozę za pomocą polecenia:

```
1      diagnoza .
```

Następnie musimy odpowiedzieć na ciąg pytań zadanych przez system ekspercki.

```
?- ['C://Users//Jola//Desktop//expert-sys.pl'].
true.

?- diagnoza.
Czy uwazasz, ze jestes chory? (tak/nie)
|: tak.
Czy masz objaw: goraczka? (tak/nie)
|: tak.
Czy masz objaw: kaszel? (tak/nie)
|: tak.
Czy masz objaw: bol_glowy? (tak/nie)
|: tak.
Mozliwa diagnoza to: grypa
true.

?-
```

Za nawiasem kwadratowym powinna znaleźć się kropka.

3 Zadania

Proszę o realizację następujących zadań.

1. Dodaj do skryptu definicję nowej choroby grypa_zoladkowa, która charakteryzuje się objawami: nudności, wymioty, goraczka. Następnie uruchom skrypt i sprawdź, czy nowa choroba zostanie poprawnie zdiagnozowana, jeśli podasz odpowiednie objawy.
2. Zaktualizuj regułę pytaj_o_objawy tak, aby po pytaniu, czy użytkownik jest chory, zadawała dodatkowe pytanie o jego wiek (np. „Czy masz

mniej niż 18 lat?”). Zmienna Wiek powinna być następnie wykorzystywana do podejmowania decyzji o dalszym zadawaniu pytań lub diagnozie.

3. Zaktualizuj skrypt tak, aby mógł rozpoznawać sytuacje, w których użytkownik może mieć więcej niż jedną chorobę na podstawie podanych objawów. Skrypt powinien wyświetlać wszystkie możliwe diagnozy, a nie tylko pierwszą pasującą.
4. Zaktualizuj skrypt tak, aby na podstawie zebranych objawów nie tylko podawał nazwę choroby, ale również wyświetlał rekomendację dotyczącą postępowania (np. „Skonsultuj się z lekarzem” lub „Zażyj leki dostępne bez recepty”).
5. Dodaj nową funkcję wyjaśnij(Choroba), która na podstawie diagnozy zwróci szczegółowe informacje na temat tej choroby, takie jak możliwe przyczyny, czas trwania choroby, oraz rekomendacje dotyczące leczenia. Funkcja powinna być wywoływana po postawieniu diagnozy.
6. Dodaj do skryptu pytania o styl życia użytkownika, takie jak „Czy palisz papierosy?”, „Czy uprawiasz regularnie sport?”, które mogą wpływać na diagnozę. Skrypt powinien uwzględniać te odpowiedzi podczas diagnozy.
7. * Rozbuduj system o obsługę alergii sezonowych, które mogą być aktywne tylko w określonych porach roku. Skrypt powinien pytać użytkownika o aktualną porę roku i na tej podstawie wykluczać lub potwierdzać diagnozy związane z alergiami.

4 Sprawozdanie

W sprawozdaniu przedstaw wyniki swoich eksperymentów:

- Imię i nazwisko
- Nazwę ćwiczenia
- Opis algorytmu
- Zestawienie wyników (wykresy, tabele z komentarzem).
- Wnioski końcowe Skrypt programu (załącz jako osobny plik z rozszerzeniem .pl w kursie Podstawy sztucznej inteligencji na portalu e-nauka).

Skrypt programu (załącz jako osobny plik z rozszerzeniem .pl w kursie Podstawy sztucznej inteligencji na portalu e-nauka).