Laboratorium nr 3 - Sztuczna Inteligencja
Arytmetyka liczb w PROLOGU

Logiczne programowanie nie posiada zaimplementowanej arytmetyki liczb - programista musi sam zdefiniować arytmetykę w logiczny sposób. Najpierw programista powinien zdefiniować czym jest liczba, a następnie zdefiniować narzędzia pozwalające na operowaniu na liczbach.

Liczby będą reprezentowane przez termy zawierające stałą zero oraz funkcję następnika $s(x) = x + 1$. Tak więc dla przykładu $zero$, $s(zero)$, $s(s(zero))$, $s(s(s(zero)))$ będą reprezentować liczby całkowite: 0, 1, 2 i 3.

Zaprogramujmy teraz predykat, który będzie generował liczby nieujemne całkowite:

$isnumber(zero).$
$isnumber(s(X)){:}-isnumber(X).$

Posiadając powyższą definicję, możemy również zdefiniować relację równoważności liczb oraz relację częściowego porządku:

$isequal(X, X){:}-isnumber(X).$
$isequal\big(s(X), s(Y)\big){:}-isequal(X, Y).$

$lessthanequal(zero, X){:}-isnumber(X).$
$lessthanequal(s(X), s(Y)){:}-lessthanequal(X, Y).$

Zad.1 Przetestuj powyższe predykaty. Sprawdź w jaki sposób Prolog sprawdza poprawność zapytań. W jaki sposób wyświetlić liczby mniejsze od zadanej liczby, np. od 4 (w sensie od $s(s(s(s(zero))))$)?

Zad.2 Przeanalizuj w jaki sposób możemy zdefiniować operację dodawania. W rozwiązaniu zadania pomogą dwa następujące fakty:

Jeżeli dodamy zero do jakiejś liczby to otrzymamy tą samą liczbę.
Jeżeli $x + y = z$ to $(x + 1) + y$ musi być równe $z + 1$.

Powyższe fakty możemy zapisać w Prologu jako:

$$add(zero, X, X){:}-isnumber(X).$$
$$add\big(s(X), Y, s(Z)\big){:}-add(X, Y, Z).$$

Przetestuj zaprogramowaną operację dodawania.

Zad.3. Wzorując się na powyższych przykładach zaprogramuj poniższe predykaty:

a) $odd(X)$ jest prawdziwe jeżeli X jest liczbą nieparzystą.
b) $even(X)$ jest prawdziwe jeżeli X jest liczbą parzystą.
c) $times(X, Y, Z)$ jest prawdziwe jeżeli $XY = Z$.
d) $quotient(X, Y, Q)$ jest prawdziwy jeżeli $X/Y = Q$ (w arytmetyce liczb naturalnych)

e) $remainder(X, Y, R)$ jest prawdziwy jeżeli X dzielone przez Y daje resztę R
f) $fact(N, X)$ jest prawdziwe jeżeli $X = N!$.
g) $fibonacci(N, X)$ jest prawdziwe jeżeli X jest N-tą liczbą Fibonacciego.

Zad.4. Zdefiniuj predykat $shownumber(X, N)$, który jest prawdziwy gdy symbol X odpowiada liczbie naturalnej N. Dla przykładu $shownumber(s(zero),1)$ jest prawdą. Następnie sprawdź wynik następujący zapytań:

?- shownumber(s(s(zero))),X).
?- shownumber(Y,5).

Podpowiedź: *Wykorzystaj wbudowaną arytmetykę w Prologu (spr. w dokumentacji: Arithmetic).*

Zad.5 Wykorzystaj predykat shownumber(X,Y), aby zdefiniować:

a) times(X,Y,Z)
b) quotient(X,Y,Q) oraz remainder(X,Y,R) dla Y=zero?