ZMIENNE i WYRAŻENIA MATEMATYCZNE

1. Zmienne

Zmienne w CLIPS-ie zapisujemy ze znakiem zapytania na początku:

?nazwa_zmiennej

np. ?x ?y ?a1 ?e

Aby do zmiennej przypisać wartość, najczęściej używamy polecenia (bind ...) np.
(bind ?x 82.5) – oznacza, że do zmiennej x przypisujemy wartość 82.5
(bind ?y ?x) - oznacza, że do zmiennej y przypisujemy wartość zmiennej x
(bind ?a1 (+ ?x ?y 12)) – oznacza, że do zmiennej a1 przypisujemy wartość (x+y+12)

Aby wartość zmiennej była „widziana” także przez inne reguły, potwierdzamy fakt:
(assert (x ?x)), (assert (y ?y)) lub (assert (a1 ?a1))

Gdy do zmiennej przypisujemy nową wartość i potwierdzamy fakt, lista faktów staje się dłuższa. Maszyna wnioskująca stara się dopasować każdy fakt do lewej strony każdej reguły, przez co system staje się mniej wydajny. Dobrze jest więc usuwać niepotrzebne już fakty. W tym celu należy przywiązać do faktu po lewej stronie reguły adres faktu:

Przykład:

```
(deffacts fakty (x 5)(y 2))
(defrule usuwanie_faktow
?fakt1<-(x ?x)
?fakt2<-(y ?y)
=>
(printout t " x = " ?x crlf " y = " ?y crlf)
(retract ?fakt1 ?fakt2))
```

```
CLIPS> (reset)
= = > f-0     (initial-fact)
= = > f-1     (x  5)
= = > f-2     (y  2)
CLIPS> (run)
x = 5
y = 2
< = = f-1     (x  5)
< = = f-2     (y  2)
CLIPS>
```

Jak widać, przy użyciu jednej komendy (retract ….) można usunąć kilka faktów.

3. Operacje matematyczne

W CLIPS-ie mamy następujące operatory matematyczne:

+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
**	potęgowanie

W wyżej wymienionych operacjach (z wyjątkiem potęgowania) mogą być więcej niż dwa argumenty. Obliczanie wartości wyrażenia następuje z lewej do prawej strony, np. dla x=8, y=4, z=2, mamy:

```
(+  ?x  ?y  ?z)      daje 14
(-  ?x  ?y  ?z)      daje 2
(*  ?x  ?y  ?z)      daje 64
(/  ?x  ?y  ?z)      daje 1
```

Wykładnik w operacji potęgowania może być ułamkowy np.

```
CLIPS> (**  8 (/  1  3))        ;osiem do jednej-trzeciej
```

Wyrażenia w CLIPS-ie zapisujemy w następujący sposób, np. (x + y * z) – x/z w CLIPS zostanie zapisane jako: (- (+ x (* y z)) (/ x z))

CLIPS operując na wyrażeniach liczbowych próbuje zachować typ wyniku taki sam jak typ argumentów, tzn. działanie na dwóch argumentach zmiennoprzecinkowych daje rezultat zmiennoprzecinkowy, na dwóch argumentach całkowitych – rezultat całkowity (z wyjątkiem dzielenia), argument zmiennoprzecinkowy i całkowity dają rezultat zmiennoprzecinkowy.

Możemy jawnie zmienić jeden typ na inny, przez użycie funkcji (float ...) oraz (integer ...)

```
CLIPS> (float (+  2  3))
CLIPS> (integer (+  2.0  3.0))
```

4. Funkcje matematyczne

```
(min  ?x  ?y  ?z)      ;zwraca najmniejszą liczbę z podanych
(max  ?x  ?y  ?z)      ;zwraca największą liczbę z podanych
(mod  ?x  ?y)          ;zwraca resztę z dzielenia pierwszej liczby przez drugą
(log  ?x)              ;zwraca logarytm naturalny z ?x
(log10  ?x)            ;zwraca logarytm dziesiętny z ?x
(exp  ?x)              ;zwraca e podniesione do potegi ?x
(sqrt  ?x)             ;zwraca pierwiastek kwadratowy z ?x
(trunc  ?x)            ;zwraca całkowitą część liczby ?x
(abs  ?x)              ;zwraca wartość bezwzględną z ?x
(length  $?x)          ;zwraca liczbę pól w zmiennej wielopolowej $?x
(nth  n  $?x)          ;zwraca n-te pole zmiennej wielopolowej $?x
(round  ?x)            ;zaokrągla ?x do najbliższej liczby całkowitej
```

5. Funkcje trygonometryczne

```
(cos  ?x)      (sin  ?x)      (tan  ?x)      (cot  ?x)
```

Uwaga: Wszystkie argumenty muszą być podawane w radianach !
Funkcja (deg-rad ?beta) zamienia stopnie na radiany
Funkcja (rad-deg ?beta) zamienia radiany na stopnie
Liczba (pi) musi być podawana w nawiasach

6. Przykład – część programu obliczającego pierwiastki równania kwadratowego

```
(defrule naglowek
     (initial-fact)
=>
     (printout t "Program oblicza pierwiastki rownania kwadratowego : "
crlf)
     (printout t "a*x^2 + b*x + c = 0" crlf crlf)
     (printout t "Podaj a : ")
     (bind ?a (read))
     (printout t "Podaj b : ")
     (bind ?b (read))
     (printout t "Podaj c : ")
     (bind ?c (read))
     (assert (a ?a)(b ?b)(c ?c)))

(defrule licz-delte
     (a ?a)
     (b ?b)
     (c ?c)
=>
     (bind ?delta (- (* ?b ?b)(* 4 ?a ?c)))
     (assert (delta ?delta)))

(defrule delta-rowna-zero
     (delta ?delta)
     (a ?a)
     (b ?b)
     (test (= ?delta 0))
=>
     (bind ?x1 (/ (- 0 ?b)(* 2 ?a)))
     (bind ?x2 ?x1)
     (assert (x1 ?x1)(x2 ?x2)))

(defrule pisz-wynik
     (x1 ?x1)
     (x2 ?x2)
=>
     (printout t "Rozwiazanie : x1 = " ?x1 " x2 = " ?x2 crlf))
```

7. Zadanie

a) do powyższego przykładu dopisać regułę obliczająca pierwiastki, gdy delta > 0,

$\Delta>0$ istnieją dwa rozwiązania $x_1 = \frac{-b+\sqrt{\Delta}}{2a}$ oraz $x_2 = \frac{-b-\sqrt{\Delta}}{2a}$,

b) do powyższego przykładu dopisać regułę obliczająca pierwiastki, gdy delta < 0,

gdy $\Delta<0$ istnieją dwa rozwiązania zespolone postaci $x_1 = \frac{-b}{2a} - \frac{\sqrt{|\Delta|}}{2a} j$ oraz $x_2 = \frac{-b}{2a} + \frac{\sqrt{|\Delta|}}{2a} j$

c) do powyższego przykładu dopisać regułę wypisująca wynik zespolony

d) napisać program rozwiązujący układ równań postaci:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

wiedząc, że: $W = a_1b_2 - b_1a_2$; $W_1 = c_1b_2 - c_2b_1$; $W_2 = a_1c_2 - a_2c_1$;

Jeżeli $W \neq 0$, to układ ma dokładnie jedno rozwiązanie:

$$x = \frac{W_1}{W}\ ;\ y = \frac{W_2}{W}$$

Jeżeli $W = 0$ i $W_1 \neq 0$ lub $W_2 \neq 0$, to układ jest sprzeczny
Jeżeli $W = 0$ i $W_1 = W_2 = 0$, to układ jest nieoznaczony