



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

Wydział Mechaniczny Technologiczny
Politechnika Śląska

Inżynieria wiedzy

Instrukcja do zajęć laboratoryjnych

1. Podstawy programowania w języku CLIPS



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

Gliwice 2008

1 Cel ćwiczenia

Wykonując ćwiczenie zapoznasz się z podstawowymi informacjami dotyczącymi języka CLIPS oraz nauczysz się podstaw programowania przy użyciu tego języka.

2 Zanim przejdziemy dalej

Przypomnij sobie lub znajdź w literaturze lub innych dostępnych źródłach, odpowiedzi na następujące pytania:

- Co to są systemy ekspertowe?
- Jakie narzędzia służą do tworzenia systemów ekspertowych?

3 Kilka informacji na temat języka CLIPS

CLIPS (skrót od *C Language Integrated Production System*) jest językiem opracowanym przez Software Technology Branch (STB), NASA/Lyndon B. Johnson Space Center, do celów tworzenia systemów ekspertowych. Pierwsza wersja języka ukazała się w 1986 roku i od tego czasu CLIPS jest ciągle udoskonalany. Język jest używany przez tysiące specjalistów z różnych dziedzin, na całym świecie.

CLIPS do reprezentacji wiedzy wykorzystuje następujące elementy:


- fakty i reguły - elementy programowania deklaratywnego,
- funkcje - elementy programowania proceduralnego,
- obiekty - elementy programowanie obiektowego.

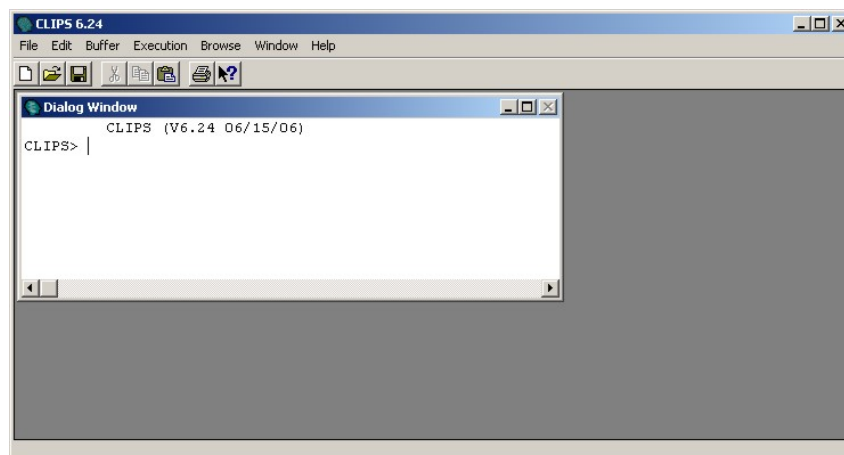
CLIPS jest oprogramowaniem typu Public Domain - można go używać nieodpłatnie. Więcej informacji dotyczących tego języka znajdziesz na stronie internetowej <http://clipsrules.sourceforge.net/>.

4 Zaczynamy!

W rozdziale tym zostaną przedstawione czynności, które będziesz wykonywał(a), wraz z informacjami pomocniczymi oraz opisem skutków wykonanych działań. Czytaj uważnie, wykonuj czynności bez pośpiechu a następnie porównuj dokładnie ich skutki z opisem. Zwracaj uwagę na składnię poleceń. Do dzieła!


4.1 Uruchom program CLIPS

Zlokalizuj i uruchom plik  CLIPSWin.exe. Na ekranie monitora pojawi się okno programu:



W oknie *Dialog window* pojawi się linia poleceń wraz ze znakiem zachęty:


```
CLIPS>
```


 Kolejne czynności będą polegały na wpisywaniu poleceń zrozumiałych dla języka CLIPS. Wpisuj tylko tekst poprzedzony znakiem zachęty widocznym powyżej (po wpisaniu tekstu naciskaj klawisz Enter). Linie poniżej to odpowiedź programu na wydane przez Ciebie polecenia. Taki tryb wykonywania poleceń przez program CLIPS nazywany jest bezpośrednim (ang. top level).

4.2 Wprowadź pierwszy fakt


Do wprowadzania faktów służy polecenie **assert**. W oknie poleceń wpisz:

```
CLIPS> (assert (lubi Tomek mechanike))  
<Fact-0>
```

 Zdefiniowałeś(aś) fakt składający się z trzech atomów (wyrazów). Atom jest ciągiem znaków, który nie podlega podziałowi na mniejsze jednostki leksykalne.

 Wyróżnia się trzy typy atomów:


1. symboliczne - zaczynające się od litery (mogą dalej zawierać cyfry, myślnik lub znak podkreślenia),
2. literalne - zawierające dowolne znaki, zamknięte w cudzysłowie,
3. numeryczne - liczby zmiennoprzecinkowe.

 CLIPS umieszcza wprowadzane fakty na liście i przypisuje każdemu z nich kolejny numer. CLIPS poinformował Cię, że wprowadzonemu faktowi został przyporządkowany numer 0.

 Pamiętaj, że w CLIPSie wszystkie polecenia należy zamykać w nawiasie okrągłym.

4.3 Wprowadź jeszcze raz ten sam fakt


```
CLIPS> (assert (lubi Tomek mechanike))  
FALSE
```


 Pojedynczy fakt może być zdefiniowany tylko raz. Stąd odpowiedź programu: FALSE.

4.4 Wprowadź grupę faktów

Użyj poleceń **deffacts** oraz **reset**:

```
CLIPS> (deffacts lubia(lubi Ania matematyke)  
(lubi Rafal mechanike)  
(lubi Kasia matematyke))  
CLIPS> (reset)
```


 Za pomocą polecenia 'deffacts' można wprowadzać grupę faktów o zadanej nazwie (w tym przypadku 'lubia'). Do wprowadzenia na listę faktów zdefiniowanych w ten sposób służy polecenie **reset**.


 Polecenie 'reset' najpierw usuwa z listy wszystkie istniejące fakty, ustawia fakt początkowy 'initial-fact' jako pierwszy (o numerze 0) i dopiero po tych operacjach umieszcza na liście kolejno zdefiniowane fakty.

4.5 Wyświetl listę faktów

Użyj polecenia **facts**:

```
CLIPS> (facts)
f-0 (initial-fact)
f-1 (lubi Ania matematyke)
f-2 (lubi Rafal mechanike)
f-3 (lubi Kasia matematyke)
For a total of 4 facts.
```

 Polecenie 'facts' wyświetla indeksy (numery) oraz fakty do których są one przypisane.

 Listę faktów można na bieżąco śledzić uaktywniając okno 'Facts', poprzez wybór opcji Window -> 1 Facts Window, w menu górnym.

4.6 Wprowadź po raz trzeci pierwszy z poznanych faktów


Ponieważ fakt wyrażający że Tomek lubi mechanikę został usunięty z listy faktów przez operację 'reset', należy go wprowadzić jeszcze raz.

```
CLIPS> (assert (lubi Tomek mechanike))
<Fact-4>
```


4.7 Zdefiniuj pierwszą regułę


Użyj polecenia **defrule**:


```
CLIPS> (defrule lubia-mechanike (lubi Tomek mechanike) (lubi Rafal mechanike) =>
(assert (lubia-mechanike chlopcy)))
```

 Definicja reguły ma następującą składnię:

```
(defrule nazwa_reguły (Warunek 1)...(Warunek n) => (Polecenie 1)...(Polecenie m))
```

 Regułę należy rozumieć następująco: jeśli są spełnione (jednocześnie) warunki Warunek 1 ... Warunek n, wykonaj polecenia Polecenie 1 ... Polecenie m.

 Część reguły znajdującą się po lewej stronie znaku => określa się skrótem 'LHS' (ang. left hand side), natomiast część występującą po prawej stronie znaku określa się skrótem 'RHS' (ang. right hand side).

 Reguła o danej nazwie może być definiowana wiele razy. Za każdym razem stara definicja jest wtedy zastępowana nową.

4.8 Wyświetl definicję reguły

Użyj polecenia **ppdefrule**:

```
CLIPS> (ppdefrule lubia-mechanike)
(defrule::MAIN lubia-mechanike
(lubi Tomek mechanike)
(lubi Rafal mechanike)
=>
(assert (lubia-mechanike chlopcy)))
```

4.9 Zdefiniuj drugą regułę

Zdefiniuj regułę o nazwie 'lubia-matematyke', gdzie w LHS wystąpią fakty dotyczące dziewczyn. Niech w RHS zostanie wprowadzony fakt 'lubia-matematyke dziewczyny'. Wzoruj się na przykładzie z punktu 4.7.


4.10 Wyświetl listę reguł


Użyj polecenia **rules**:


```
CLIPS> (rules)
lubia-mechanike
lubia-matematyke
For a total of 2 defrules.
```

4.11 Wyświetl agendę

```
CLIPS> (agenda)
0 lubia-matematyke f-3, f-1
0 lubia-mechanike f-4, f-2
For a total of 2 activations.
```

 Jak się pewnie domyślasz, agenda to lista aktywacji reguł, których LHS są spełnione przez istniejące fakty.

 Agendę można na bieżąco śledzić uaktywniając okno 'Agenda', poprzez wybór opcji Window -> 2 Agenda Window, w menu głównym.

 Zauważ, że CLIPS wypisał reguły oraz fakty w kolejności 'od końca'. CLIPS wykorzystuje zasadę odkładania elementów na stosy, tzn. elementy wprowadzane jako ostatnie są przetwarzane jako pierwsze. Dotyczy to zarówno faktów jak i reguł.


4.12 Uruchom procedurę wnioskującą...

...za pomocą polecenia **run** oraz sprawdź listę faktów:

```

CLIPS> (run)
CLIPS> (facts)
f-0 (initial-fact)
f-1 (lubi Ania matematyke)
f-2 (lubi Rafal mechanike)
f-3 (lubi Kasia matematyke)
f-4 (lubi Tomek mechanike)
f-5 (lubia-matematyke dziewczyny)
f-6 (lubia-mechanike chlopcy)
For a total of 7 facts.


```


 Reguły mogą być uruchamiane przy wykorzystaniu tych samych faktów tylko raz! Aby ponownie uruchomić zdefiniowane reguły należy usunąć wszystkie fakty i zdefiniować je powtórnie. W przeciwnym wypadku uruchomienie reguł nie przyniesie żadnego skutku. Sprawdź!

4.13 Usuń fakty 5. i 6.

Użyj polecenia **retract**:

```
CLIPS> (retract 5 6)
```

 Pojedyncze fakty usuwa się za pomocą polecenia 'retract'. Argumentem polecenia jest lista indeksów faktów przeznaczonych do usunięcia.

 Aby usunąć fakty za pomocą polecenia 'retract' musisz znać ich indeksy. Nie można usuwać faktów posługując się ich treścią.

4.14 Sprawdź, kto lubi matematykę


Zdefiniuj regułę, która zidentyfikuje indeksy odpowiednich faktów, wyświetli informację oraz usunie użyte fakty z listy:

```

CLIPS> (defrule kto-lubi-matematyke ?x<-(lubi ?imie matematyke) =>
(printout t ?imie " lubi matematyke." crlf) (retract ?x))
CLIPS> (run)
Kasia lubi matematyke.
Ania lubi matematyke.
CLIPS> (facts)
(f-0) initial-fact
(f-2) lubi Rafal mechanike
(f-4) lubi Tomek mechanike
For a total of 3 facts.


```

 Znak zapytania przed wyrazem oznacza, że wprowadzana jest zmienna, np. **?x**.

 Powyższa reguła jest uruchamiana dla wszystkich faktów, których pierwszy atom to wyraz 'lubi', drugi atom jest dowolny, natomiast trzeci atom to wyraz 'matematykę'. Dla kolejnych faktów zmiennej ?imie przypisywane są odpowiednie atomy.

 Zmienna nie może odnosić się do pierwszego z atomów faktu.

 Operator `<-` przypisuje zmiennej `?x` indeksy faktów spełniających LHS definiowanej reguły.

 Polecenie **`printout`** służy do wyświetlania komunikatów. Dalej następuje lista argumentów. Pierwszy z nich określa urządzenie do którego kierowany jest komunikat (**`t`** oznacza ekran monitora). Kolejne argumenty definiują komunikat. Wyświetlone mogą być zmienne (np. **`?imie`**) lub **ciągi znaków zamknięte w znak cudzośćlowu**. Ostatni argument, **`crLf`**, oznacza, że po wypisaniu komentarza kursor ma przejść do nowej linii.


4.15 Sprawdź również, kto lubi mechanikę

Wzorując się na regule przedstawionej w poprzednim poleceniu, utwórz analogiczną regułę dotyczącą osób, które lubią mechanikę. Uruchoń regułę i sprawdź wynik jej działania.

4.16 Usuń reguły

Użyj polecenia **`clear`**. Następnie wyświetl fakty i reguły.


```
CLIPS> (clear)
CLIPS> (facts)
CLIPS> (rules)
```

 Polecenie `'clear'` usuwa wszystkie reguły oraz wszystkie fakty, wraz z faktem `'initial-fact'`.

4.17 Wczytaj fakt z klawiatury

Wykorzystaj polecenie **`read`**:

```
CLIPS> (assert (student (read) (read)))
Jan
Kowalski
<Fact-0> CLIPS>(facts)
f-0 student Jan Kowalski
For a total of 1 fact.
```

 Polecenie `read` wczytuje każdorazowo pojedynczy atom faktu z urządzenia zewnętrznego. Urządzeniem domyślnym jest klawiatura.

 Nie wolno wczytywać pierwszego atomu występującego w fakcie.

4.18 Zakończ pracę z programem CLIPS

```
CLIPS> (exit)
```

5 Podsumowanie

Wykonując wszystkie powyższe polecenia zapoznałeś(aś) się z podstawowymi zasadami programowania w języku CLIPS. Nauczyłeś(aś) się jak:

- definiować, przeglądać i usuwać fakty,
- definiować, przeglądać i usuwać reguły,
- uruchamiać procedurę wnioskującą,
- przeszukiwać listę faktów,
- wyświetlać komunikaty,
- wczytywać fakty z klawiatury.

Literatura

- [1] Cholewa W., Pedrycz W., *Systemy doradcze*. Wydawnictwo Politechniki Śląskiej, Gliwice 1987.
- [2] Giarratano J. C., *CLIPS User's Guide*, <http://clipsrules.sourceforge.net/>.
- [3] Kendal S., Creen M., *An Introduction to Knowledge Engineering*. Springer-Verlag, London 2007.
- [4] Mulawka J. J., *Systemy ekspertowe*. WNT, Warszawa 1996.
- [5] Rutkowski L., *Metody i techniki sztucznej inteligencji*. WNT, Warszawa 2005.
- [6] Russel S., Norvig P., *Artificial intelligence: A Modern Approach*. Prentice Hall, 2002.