ОБЩИ СВЕДЕНИЯ ЗА CLIPS

CLIPS (C Language Integrated Production System) е разработка на NASA’s Johnson Space Center. Представлява инструментално средство за изграждане на експертни системи.

История

- Прототип: 1985 г. (език за работа с продукционни правила, чийто интерпретатор извършва прав извод на основата на алгоритъма RETE).
- Версия 3.0: 1986 г. (първата версия на CLIPS, достъпна за потребители извън NASA).
- Версия 5.0: 1991 г. (добавени са две нови парадигми: процедурно програмиране и обектно-ориентирано програмиране).

История (продължение)

- Версия 6.0: 1997 г. (интеграция на трите парадигми: декларативни знания, процедурни знания, ООП).
- Версия 6.1: 1998 г. (коригирани са някои недостатъци в работата на интерпретатора на правилата и интерфейса на CLIPS).
- Версия 6.2: 2002 г. (усъвършенствани са средствата за създаване на приложения с използване на CLIPS).

- Версия 6.3: 2008 г. (подобрена е работата на интерпретатора в случаите на работа с голям брой факти/екземпляри; добавена е възможност за работа с 64-битови цели числа; поддръжка на Microsoft Windows VC++ Express 2008 и Borland Turbo C++ 2006; подобрени възможности за създаване на C++ библиотеки и DLL).

Разпространение на CLIPS

Сега CLIPS се разпространява като public domain software от основните си автори, които вече не работят за NASA. Достъпна е компилирана версия, както и source код на C/C++.

http://clipsrules.sourceforge.net/

JESS (Java Expert System Shell) е развит вариант на подмножество на CLIPS, реализиран на Java.

Ф А К Т И

Добавяне на факти:

```
(assert <fact-specifier>+)
```

Изтриване на факти:

```
(retract <fact-specifier>+)
```

Пример

```
CLIPS> (clear)
CLIPS> (assert (student ivan))
<Fact-0>
CLIPS> (assert (student peter))
<Fact-1>
CLIPS> (facts)
f-0    (student ivan)
f-1    (student peter)
For a total of 2 facts.
CLIPS>
```

Примери за наредени факти

```
(student)
(student ivan)
(grocery-list bread milk eggs)
```

Конструкция deftemplate

```
(deftemplate <deftemplate-name> [<comment>]
  <slot-definition>*)
<slot-definition> ::= <single-slot-definition> |
  <multislot-definition>
<single-slot-definition> ::=
  (slot <slot-name> <template-attribute>*)
<multislot-definition> ::=
  (multislot <slot-name> <template-attribute>*)
<template-attribute> ::= <default-attribute> |
  <constraint-attribute>
<default-attribute> ::=
  (default ?DERIVE | ?NONE | <expression>*) |
  (default-dynamic <expression>*)
```

Примерни дефиниции

```
(deftemplate student
  (slot name)
  (slot fac-number)
  (multislot exams))
```

```
(deftemplate person
  (slot name
     (type SYMBOL)
     (default ?DERIVE))
  (slot age
     (type INTEGER)
     (range 1 100)
     (default 30))
```

```
(slot weight
   (allowed-values light heavy)
   (default light))
(multislot hobbies
   (type SYMBOL)))
```

Конструкция deffacts

Синтаксис:

```
(deffacts <deffacts-name>
          [<comment>]
          <fact-specifier>*)
```

Пример

```
(deftemplate person
  (slot name)
  (slot age)
  (multislot friends))
(deffacts people
  (person (name Joe) (age 20))
  (person (name Bob) (age 20))
  (person (name Joe) (age 34))
  (person (name Sue) (age 34))
  (person (name Sue) (age 20)))
```