

CLIPS

C Language Integrated
Production System

Šta je CLIPS ?

- Jezik za razvoj ekspertnih sistema
- Razvijen u NASA Johnson Space Centru
- Prva verzija – 1986. godine.
- <http://www.jsc.nasa.gov/~clips/CLIPS.html>

ZAPAMTITI

**CLIPS RAZLIKUJE MALA I VELIKA
SLOVA!!!**

Komponente CLIPS-a

- Jezik produkcionih pravila – Production rule language, Rule-based language
- Proceduralni jezik – Procedural language
- COOL – CLIPS Object Oriented Language

Komponente jezika zasnovanog na pravilima:

- Lista činjenica (fact list)
 - ✓ Sadrži podatke na osnovu kojih se zaključuje
- Skup pravila (rule set)
 - ✓ Omogućava da se od početnog stanja dođe do rešenja problema

Veza između činjenica i pravila:

- MEHANIZAM ZA ZAKLJUČIVANJE
(Inference Engine)
 - ✓ Pravilima pridružuje odgovarajuće činjenice
 - ✓ Bira pravilo koje će “okinuti”
 - ✓ Izvršava akcije određene tim pravilom

Dakle postoje 3 načina predstavljanja znanja u CLIPS-u:

- *Pravila* – za heurističko znanje bazirano na iskustvu.
- *Definisane funkcije i generičke funkcije* – za proceduralno znanje.
- *Objekti* - classes, message-handlers, abstraction, encapsulation, inheritance, polymorphism.

- CLIPS softver se može razvijati korišćenjem samo pravila, pravila i funkcija, samo objekata ili kombinacijom svega navedenog.
- CLIPS se može potpuno integrisati sa C jezikom. CLIPS je akronim za C Language Integrated Production System.
- CLIPS se može koristiti samostalno, može biti pozivan iz proceduralnog jezika ili se proceduralni kod može definisati kao spoljna funkcija i pozivati iz CLIPS-a.

Tokens & Fields

- tokens
 - grupe karaktera koje za CLIPS imaju posebno značenje,
- fields
 - naročito važna grupa tokena
 - ima 8 tipova fields-ova – primitivnih tipova podataka

Tipovi podataka u CLIPSu:

- float
- integer
- symbol
- string
- external-address
- fact-address
- instance-name
- instance-address

- **float**

broj sa decimalnom tačkom ili u notaciji eksponenta

1.5 -2.89 6.4e7

- **integer**

označen ili neoznačen ceo broj

3 -21 +5

- **symbol**

Bilo koji niz karaktera koji počinje bilo kojim ASCII karakterom koji se može printati. Pod završetkom simbola se podrazumeva bilo koji od graničnika: space, tab, carriage return, line feed, ”, (,), &, |, <, ~, ;. Smeju počinjati sa “<”. Ne smeju počinjati sa “\$?” , “?” , ali smeju sadržati ove karaktere.

Hello 2fast @+-*% TRUE

- **string**

Počinje i završava se navodnicima

“Ovo je string“

Ovo je string

“\“Ovo je string\““

“Ovo je string“

“\\Ovo je string\\“

\Ovo je string\

- **external-address**

Adresa spoljnog podatka koji vraća neka funkcija napisana u, na primer, C jeziku koja je integrisana u CLIPS.

- **fact address**

Adresa činjenice

- **instance-name, instance-address**

Tipovi podataka koji služe za referenciranje na pojedine instance (pojavljivanja) određenih klasa u slučaju korišćenja COOLa - objektno orijentisane ekstenzije CLIPSa

CLIPS prompt (top-level) izgleda ovako:

CLIPS>

Odavde se komande unose direktno u CLIPS.

Sve komande se kucaju unutar zagrada.

Primer:

```
CLIPS> (exit) ;; izlazak iz CLIPS-a
```

FACTS

- Činjenica predstavlja neku informaciju
- Sastoji se od naziva relacije (tipa symbol) iza kog slede nula ili više slotova (tipa symbol) i njima pridružene vrednosti

```
(osoba (ime "Petar Petrovic")  
      (godine 22)  
      (boja-ociju plava)  
      (boja-kose sredja))
```

*redosled slotova u činjenici nije bitan

Konstrukcija `deftemplate`

- Grupe činjenica koje imaju isto ime relacije i sadrže slične informacije mogu se opisati `deftemplate` konstrukcijom.

```
(deftemplate <naziv-relacije>  
  (slot naziv-slota) |  
  (multislot naziv-multislota)*)
```


Deftemplate konstrukcija koja odgovara činjenici osoba

```
(deftemplate osoba  
  (slot ime)  
  (slot godine)  
  (slot boja-ociju)  
  (slot boja-kose))
```

naziv relacije

naziv slota

Multifield slots

- Slotovi koji su u odgovarajućoj `deftemplate` konstrukciji određeni ključnom reči `slot` mogu imati tačno jednu vrednost – *single-field slots*.
- Slotovi koji su u odgovarajućoj `deftemplate` konstrukciji određeni ključnom reči `multislot` mogu imati nula ili više vrednosti – *multifield slots*.

Primer:

```
(deftemplate licni_podaci
  (multislot ime)
  (slot godine)
  (slot visina)
  (slot tezina)
  (multislot
   krvni_pritisak)
)
```

Činjenica koja odgovara napravljenom šablonu:

naziv relacije

(`licni_podaci`
 (`ime Andreja Andric`)
 (`godine 20`)
 (`tezina 70`)
 (`visina 178`)
 (`krvni_pritisak 120 80`))

naziv slota **vrednosti slota (polja)**

Ordered facts

- Činjenice koje smo do sada upoznali zovu se *non-ordered (deftemplate) facts*, jer se sastoje od imenovanih polja čiji redosled navođenja nije bitan.
- Činjenice sa imenom relacije koje nema odgovarajući deftemplate zovu se *ordered fact*.

Ordered facts

- Ordered facts se sastoje od imena relacije i jednog ili nijednog multifeild slota.
- Slot nije imenovan

Primer


naziv relacije

vrednosti slota (polja)

(neparni-brojevi **1 3 5 7 9**) **neimenovani multifold slot**



(deftemplate neparni-brojevi (multifold slot vrednosti))



(neparni-brojevi (vrednosti 1 3 5 7 9))

Ordered vs. non-ordered

- Non-ordered činjenice treba upotrebljavati kada god je to moguće.
- Ordered činjenice koristiti kao flag-ove (tada nemaju slotove).
- Ordered činjenice koristiti kada činjenica treba da ima samo jedan slot.

Dodavanje i uklanjanje činjenica

- Sve činjenice u CLIPS-u se nalaze u listi činjenica (*fact list*)
- Činjenice se mogu dodavati u listu, i uklanjati iz liste.

Dodavanje činjenica u listu

Komanda assert. Sintaksa:

(assert <fact>+)

```
CLIPS> (deftemplate osoba
  (slot ime)
  (slot godine)
  (slot boja-ociju)
  (slot boja-kose))
```

```
CLIPS> (assert (osoba (ime "Pera Peric")
  (godine 22)
  (boja-ociju plava)
  (boja-kose smedja)))
```

<Fact-0>

```
CLIPS> (assert (neparni-brojevi 1 3 5 7 9))
```

<Fact-1>

```
CLIPS>
```

Nakon primene komande assert CLIPS nam vraća adresu činjenice, ubačene ovom komandom, u listi činjenica

Prikazivanje liste činjenica:

Naredba za prikazivanje liste činjenica je `facts`

```
CLIPS> (facts)
```

```
f-0 (osoba (ime "Petar Petrovic")
```

```
      (godine 22)
```

```
      (boja-ociju plava)
```

```
      (boja-kose sredja))
```

```
f-1 (neparni-brojevi 1 3 5 7 9)
```

```
For a total of 2 facts.
```

```
CLIPS>
```

identifikator činjenice – njena adresa u listi činjenica

Uklanjanje činjenice iz liste

Naredba `retract`. Sintaksa:

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

```
CLIPS> (retract 0)
```

```
CLIPS> (retract 0 1)
```

Adresa činjenice
koju uklanjamo



Modifikacija činjenica

- Vrednosti polja u slotovima se menjaju komandom modify. Sintaksa:

```
(modify <fact-index> <slot-name> <slot-value>+)
```

Modifikacija činjenica

```
CLIPS> (modify 0 (godine 23))
```

```
<Fact-1>
```

```
CLIPS>(facts)
```

```
f-1 (osoba (ime "Petar  
Petrovic")
```

```
(godine 23)
```

```
(boja-ociju plava)
```

```
(boja-kose sredja))
```

```
For a total of 1 fact
```

```
CLIPS>
```

modify=retract+assert

Duplikacija činjenica

- Komanda **duplicate** radi isto što i komanda **modify**, s tim što se original ne uklanja iz liste.
- Komande **modify** i **duplicate** ne mogu se upotrebljavati sa **ordered** činjenicama.

Konstrukcija **deffacts**

Više činjenica definisanih komandom `deftemplate` se može odjednom ubaciti u listu činjenica i komandom **deffacts**. Najčešće se koristi za ubacivanje činjenica koje su poznate pre izvršenja programa, a koje treba na početku ubaciti u listu. Sintaksa:

```
(deffacts <deffacts-ime><facts>*)
```

Reset

- Nakon komande **defacts** mora se uneti komanda **reset** koja uklanja sve činjenice iz liste i u listu ubacuje činjenicu (**initial-fact**) i sve činjenice zadate **defacts** komandom.

Primer

```
CLIPS>(deffacts ljudi
  (osoba (ime "Mika Mikic")
    (godine 45)
    (boja-ociju siva)
    (boja-kose plava))
  (osoba (ime "Laza Lazic")
    (godine 37)
    (boja-ociju zelena)
    (boja-kose crna)))
CLIPS>(reset)
```

Zadatak 1

- Definisati deftemplate za činjenicu koja bi sadržala informacije o nekom skupu: naziv, elemente i naziv njegovog nadskupa.

Zadatak 2

- Sledeće skupove predstaviti kao činjenice određene prethodno definisanim deftemplate-om:

$A = \{1, 2, 3\}$

$B = \{1, 2, 3, \text{crveno}, \text{zeleno}\}$

$C = \{\text{crveno}, \text{zeleno}, \text{plavo}, \text{belo}\}$