

# Nametanje ograničenja na vrednost polja u slotu upotrebom *Field constraints* -a

- ~ (not)
- | (or)
- & (and)

- *Field constraints* se mogu kombinovati na razne načine u cilju definisanja svih vrednosti koje neko polje u uslovu pravila može da poprими prilikom proveravanja zadovoljenosti uslova.

~ (not)

- ~ ima najviši prioritet.
- ~ deluje na vrednost (ili uslov) koja sledi tako što ne dozvoljava da dato polje poprimi tu vrednost (ili ispuni taj uslov).

# PRIMER:

```
(defrule najdi-osobe-koje-nemaju-smedju-kosu
  (osoba (ime ?ime)(boja-kose ~smedja))
  =>
  (printout t ?ime " nema smedju kosu." crlf))
```

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

(deffacts ljudi
  (osoba (ime "Mika Mikic")
    (godine 45)
    (boja-ociju plava)
    (boja-kose plava))
  (osoba (ime "Laza Lazic")
    (godine 37)
    (boja-ociju zelena)
    (boja-kose crna))
  (osoba (ime "Mara Maric")
    (godine 28)
    (boja-ociju plava)
    (boja-kose smedja)))
```

```
CLIPS> (reset)
==> f-0      (initial-fact)
==> f-1      (osoba (ime "Mika Mikic") (godine 45) (boja-
ociju plava) (boja-kose plava))
==> Activation 0  nadji-osobe-koje-nemaju-smedju-kosu: f-1
==> f-2      (osoba (ime "Laza Lazic") (godine 37) (boja-
ociju zelena) (boja-kose crna))
==> Activation 0  nadji-osobe-koje-nemaju-smedju-kosu: f-2
==> f-3      (osoba (ime "Mara Maric") (godine 28) (boja-
ociju plava) (boja-kose smedja))
```

```
CLIPS> (run)
FIRE      1 nadji-osobe-koje-nemaju-smedju-kosu: f-2
Laza Lazic nema smedju kosu.
FIRE      2 nadji-osobe-koje-nemaju-smedju-kosu: f-1
Mika Mikic nema smedju kosu.
```

```
CLIPS>
```

# | (or)

- | ima najnižji prioritet
- | uslov deluje na dve vrednosti (ili dva uslova) koje povezuje tako što dozvoljava da dato polje poprimi bilo koju od tih vrednosti (ili ispuni bilo koji od tih uslova).

# PRIMER:

```
(defrule najdi-osobe-koje-ima-ju-tamnu-kosu
  (osoba (ime ?ime)(boja-kose sredja | crna))
  =>
  (printout t ?ime " ima tamnu kosu." crlf))
```



# PRIMER:

```
CLIPS> (reset)
==> f-0      (initial-fact)
==> f-1      (osoba (ime "Mika Mikic") (godine 45) (boja-
ociju plava) (boja-kose plava))
==> f-2      (osoba (ime "Laza Lazic") (godine 37) (boja-
ociju zelena) (boja-kose crna))
==> Activation 0      nadji-osobe-koje-imaju-tamnu-kosu: f-2
==> f-3      (osoba (ime "Mara Maric") (godine 28) (boja-
ociju plava) (boja-kose smedja))
==> Activation 0      nadji-osobe-koje-imaju-tamnu-kosu: f-3
```

```
CLIPS> (run)
FIRE      1 nadji-osobe-koje-imaju-tamnu-kosu: f-3
Mara Maric ima tamnu kosu.
FIRE      2 nadji-osobe-koje-imaju-tamnu-kosu: f-2
Laza Lazic ima tamnu kosu.
```

```
CLIPS>
```

# & (and)

- & ima srednji prioritet
- & uslov deluje na dva uslova koja povezuje tako što zahteva da dato polje ispuni oba uslova.
- & se koristi zajedno sa  $\sim$  i  $|$ , jer samostalno nema praktičnu vrednost.

& je koristan kada želimo da istovremeno dodelimo vrednost promenljivoj i da nametnemo ograničenje na vrednosti koje ta promenljiva može da poprими.

Neka u prethodnom primeru pored toga što želimo da pronađemo osobe sa tamnom kosom, želimo i da znamo tačnu boju njihove kose. Tada boju kose moramo vezati za neku promenljivu i dodati ograničenje da je boja smeđa ili crna.

```
(defrule osobe-koje-imaju-tamnu-kosu
(osoba (ime ?ime)
        (boja-kose ?boja&smedja|crna))
```

=>

```
(printout t ?ime " ima kosu " ?boja "
boje." crlf))))
```

& se često kombinuje sa ~. Na primer, sledeće pravilo pronalazi osobe čija kosa nije ni smeđa niti crna.

```
(defrule osobe-koje-nemaju-tamnu-kosu
(osoba (ime ?ime)
        (boja-kose ?boja&~smedja&~crna))
```

=>

```
(printout t ?ime " ima kosu " ?boja " boje."
  crlf)))
```

# Kombinovanje field constraints-a

Neka je potrebno utvrditi da li postoje dve osobe sa određenim opisom. Prva osoba treba da ima plave ili zelene oči i da nema crnu kosu. Druga osoba treba da ima oči različite od prve osobe i kosu koja je ili riđa, ili iste boje kao kod prve osobe.

```
(defrule nadji-po-opisu
(osoba (ime ?ime1)
      (boja-ociju ?oci1&plava|zeleno)
      (boja-kose ?kosa1&~crna))
(osoba (ime ?ime2&~?ime1)
      (boja-ociju ?oci2&~?oci1)
      (boja-kose ?kosa2&ridja|?kosa1))
=>
(printout t ?ime1 " ima " ?oci1 " oci i "
?kosa1 " kosu." crlf)
(printout t ?ime2 " ima " ?oci2 " oci i "
?kosa2 " kosu." crlf))
```

# Zadatak 7

Napisati pravilo koje pronalazi dve osobe od kojih prva nema zelene oči, a ima plavu ili smeđu kosu, a druga osoba ima istu boju očiju kao prva, a nema ni riđu, niti smeđu kosu.