or Uslovni element

- Ako je ispunjen bilo koji od uslova koji se nalaze pod njegovim dejstvom onda je or uslovni element zadovoljen.

Sledeća dva pravila napisana su bez korišćenja **or**

```
(deftemplate nepogoda (slot tip))
(deftemplate protivpozarni-sistem
                        (slot tip)(slot status))

(defrule iskljuci-struju1
(nepogoda (tip poplava))
=>
(printout t “Iskljuci struju” crlf))

(defrule iskljuci-struju2
(protivpozarni-sistem (tip prskalica-za-vodu)
                        (status ukljucen))
=>
(printout t “Iskljuci struju” crlf))
```

Ova pravila se mogu iskombinovati u jedno korišćenjem or uslovnog elementa

```
(defrule iskljuci-struju
(or (nepogoda (tip poplava))
    (protivpozarni-sistem (tip prskalica-za-vodu)
                          (status ukljucen)))
=>
(printout t “Iskljuci struju” crlf))
```

Drugi uslovi mogu se dodati na LHS pravila pored uslova povezanih or uslovnim elementom

```
(defrule iskljuci-struju
(struja (status ukljucena))
(or (nepogoda (tip poplava))
   (protivpozarni-sistem (tip prskalica-za-vodu)
                         (status ukljucen)))
=>
(printout t “Iskljuci struju” crlf))
```

Kako spreciti da ovo pravilo okida više puta

```
(defrule iskljuci-struju
?s <- (struja (status ukljucena))
(or (nepogoda (tip poplava))
    (protivpozarni-sistem (tip prskalica-za-vodu)
                          (status ukljucen)))
=>
(modify ?s (status iskljucena))
(printout t “Iskljuci struju” crlf))
```

Ili

```
(defrule iskljuci-struju
?s <- (struja (status ukljucena))
(or ?razlog <- (nepogoda (tip poplava))
    ?razlog <- (protivpozarni-sistem
                          (tip prskalica-za-vodu)
                          (status ukljucen)))
=>
(retract ?razlog)
(modify ?s (status iskljucena))
(printout t “Iskljuci struju” crlf))
```

not uslovni element

CLIPS dozvoljava da se neko pravilo izvrši u slučaju kada neki uslov NIJE ispunjen. To se postiže korišćenjem **not** uslovnog elementa.

not CE

- not uslovni element je zadovoljen ako uslov koji je pod njegovim dejstvom nije zadovoljen.
- Jedan not može negirati samo jedan uslov.
- Promenljive kojima je prethodno dodeljena vrednost mogu se slobodno koristiti unutar not CE. Međutim, promenljive kojima se vrednost prvi put dodeljuje unutar not CE mogu se koristiti samo u uslovu koji je pod dejstvom not-a.

PRIMER:

```
(defrule rule2
(temperatura visoka)
(ventil otvoren)
(not (postoji greska))
=>
(printout t "Zatvorite ventil“ crlf))

(deffacts f1
(temperatura visoka)
(ventil otvoren))
```

PRIMER:

```
CLIPS> (reset)
==> f-0     (initial-fact)
==> f-1     (temperatura visoka)
==> f-2     (ventil otvoren)
==> Activation 0      rule2: f-
  1,f-2,

CLIPS> (run)
FIRE    1 rule2: f-1,f-2,
Zatvorite ventil
```

Korišćenje promenljivih unutar negiranih uslova:

```
(defrule najveci-broj
(broj ?x)
(not (broj ?y & : (> ?y ?x)))
 =>
(printout t “Najveci broj je “ ?x
 crlf))
```

Sledeće pravilo utvrđuje datume kada niko od “poznatih” osoba nema rođendan

```
(defrule nema-rodjendana
(da-li-ima-rodjendana-na-dan ?datum))
(not (osoba (rodjendan ?datum)))
=>
(printout t “Nema rodjendana na “
  ?datum crlf))
```

```
(defrule nema-rodjendana
(not (osoba (rodjendan ?datum)))
(da-li-ima-rodjendana-na-dan ?datum))
=>
(printout t “Nema rodjendana na “ ?datum crlf))
```

Ako uslovi zamene mesta pravilo neće funkcionisati ispravno. Promenljive kojima se vrednost dodeli pod dejstvom **`not`** CE zadržavaju tu vrednost samo u opsegu delovanja **`not`**-a, tako da vrednost dodeljena promenljivoj **`?datum`** u prvom uslovu neće biti poznata u drugom uslovu. Takođe, prvi uslov neće biti zadovoljen ako u listi činjenica postoji činjenica o bilo kojoj osobi.

and CE

- CLIPS već pretpostavlja postojanje implicitnog and uslovnog elementa kojim su povezani svi uslovni elementi na LHS.
- Explicitno navođenje and uslovnog elementa se upotrebljava kako bi se vršilo miksovanje ostalih uslovnih elemenata.
- Ako su zadovoljeni svi uslovi koji se nalaze pod njegovim dejstvom onda je and uslovni element zadovoljen.

Kombinacija not i and CE

```
(defrule nema-identicnih-rodjendana
  (not (and (osoba (ime ?ime))
                   (rodjendan ?datum))
            (osoba (ime ~?ime)
                   (rodjendan ?datum))))
   =>
  (printout t “Nema dve osobe koje istog
              dana imaju rodjendan” crlf))
```

Kombinovanje not i test

```
(not (test (> ?x ?y)))
```

CLIPS tumači kao:

```
(not (and (initial-fact)
          (test (> ?x ?y))))
```

Tako da ako **`(initial-fact)`** ne postoji u listi uslov nikada neće biti ispunjen. Zato je bolje koristiti:

```
(test (not (> ?x ?y)))
```

Exists Conditional Element

- exists CE pruža mehanizam kojim se utvrđuje da li je određena grupa uslovnih elemenata zadovoljena bar jednim skupom činjenica iz liste.

PRIMER:

```
(deftemplate hero
(slot name)
(slot status))

(deffacts goal-and-heroes
(goal save-the-day)
(hero (name Superman)(status unoccupied))
(hero (name Spiderman)(status unoccupied))
(hero (name Batman)(status unoccupied)))

(defrule save-the-day
(goal save-the-day)
(exists (hero (status unoccupied)))
=>
(printout t "The day is saved." crlf))
```

PRIMER:

```
CLIPS> (reset)
CLIPS> (agenda)
0 save-the-day: f-1,
For a total of 1 activation.

CLIPS> (facts)
f-0 (initial-fact)
f-1 (goal save-the-day)
f-2 (hero (name Superman) (status unoccupied))
f-3 (hero (name Spiderman) (status
  unoccupied))
f-4 (hero (name Batman) (status unoccupied))
For a total of 5 facts.
```

PRIMER:

```
CLIPS> (matches save-the-day)
Matches for Pattern 1
f-1
Matches for Pattern 2
f-0
Matches for Pattern 3
f-2
f-3
f-4
Partial matches for CEs 1 - 2
f-1,
Activations
f-1,
```

forall conditional element

- forall CE pruža mehanizam kojim se utvrđuje da li je jedna grupa određenih uslovnih elemenata zadovoljena za svako pojavljivanje drugog određenog uslovnog elementa.

Neka postoji određeni broj zgrada u industrijskom postrojenju u kojima može izbiti požar i neka je potrebno utvrditi da li su sve zgrade u kojima je požar evakuisane i da li vatrogasci gase požar u svakoj od zapaljenih zgrada:

```
(deftemplate nepogoda
(slot tip)(slot lokacija))

(deftemplate vatrogasna-jedinica
(slot naziv)(slot lokacija))

(deftemplate zgrada
(slot status)(slot lokacija))

(defrule svi-pozari-su-pod-kontrolom
  (forall (nepogoda (tip pozar)(lokacija ?gde))
          (vatrogasna-jedinica (lokacija ?gde))
          (zgrada (status evakuisana)(lokacija ?gde)))
=>
(printout t "Sve zapaljene zgrade su evakuisane i
             vatrogasci gase pozare u njima" crlf))
```

```
CLIPS> (watch activations)
CLIPS> (reset)
==> Activation 0      svi-pozari-su-pod-kontrolom: *
CLIPS> (assert (nepogoda (tip pozar)(lokacija zgrada-8)))
<== Activation 0      svi-pozari-su-pod-kontrolom: *
<Fact-1>
CLIPS> (assert (zgrada (status evakuisana)(lokacija
zgrada-8)))
<Fact-2>
CLIPS> (assert (vatrogasna-jedinica (naziv A)(lokacija
zgrada-8)))
==> Activation 0      svi-pozari-su-pod-kontrolom: *
<Fact-3>
```

```
CLIPS> (assert (vatrogasna-jedinica (naziv B)(lokacija
zgrada-5)))
<Fact-4>
CLIPS> (assert (nepogoda (tip pozar)(lokacija zgrada-5)))
<== Activation 0      svi-pozari-su-pod-kontrolom: *
<Fact-5>
CLIPS> (assert (zgrada (status evakuisana)(lokacija
zgrada-5)))
==> Activation 0      svi-pozari-su-pod-kontrolom: *
<Fact-6>
CLIPS> (retract 4)
<== Activation 0      svi-pozari-su-pod-kontrolom: *
CLIPS> (retract 5)
==> Activation 0      svi-pozari-su-pod-kontrolom: *
CLIPS> (run)
Sve zapaljene zgrade su evakuisane i
             vatrogasci gase pozare u njima
```

Zadatak 9

Sledeća pravila prepisati u jedno pravilo korišćenjem and i or uslovnih elemenata

```
(defrule rule-1
  (fact-a)
  (fact-d)
   =>)

(defrule rule-2
  (fact-b)
  (fact-c)
  (fact-e)
  (fact-f)
  =>)
```

```
(defrule rule-3
  (fact-a)
  (fact-e)
  (fact-f)
  =>)

(defrule rule-4
  (fact-b)
  (fact-c)
  (fact-d)
  =>)
```