

# Produksijski sustavi

## predavanja iz Inteligentnih sustava

Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

Fakultet elektrotehnike i računarstva

Sveučilište u Zagrebu

# Sustavi temeljeni na pravilima

- Uvod
- Produkcijaska pravila
- Ekspertni sustavi
- Zaključivanje ulančavanjem unaprijed
- Zaključivanje ulančavanjem unatrag

# Sustavi temeljeni na znanju

- Općeniti pristupi 50-tih i 60-tih (npr, GPS) nisu postigli ambiciozan cilj općenitog postupka za rješanje problema
- alternativan pristup: postići korisnu funkcionalnost uz:
  - manje sofisticiran prikaz znanja
  - klase problema iz usko definiranih domena
- pravila oblika “ako – onda” naprema predikatnoj logici, ekspertno znanje naprema općem znanju
- naglasak na (komercijalnim) primjenama: npr, medicina, geologija, financije, šah, ...
- sinonimi:
  - sustavi temeljeni na pravilima
  - produkcijski sustavi
  - ekspertni sustavi

# Formalizam za rukovanje znanjem

- moderni ekspertni sustavi provode zaključivanje nad složenim bazama znanja
- tražimo formalizam za predstavljanje znanja koji bi bio dovoljno efikasan i za praktične primjene
- rezolucija opovrgavanjem u predikatnoj logici je zdrava i potpuna, ali...
- **poluodlučljivost**: postupak ne mora završiti ako postavljeni upit ne slijedi iz zadanog skupa formula
- **eksponencijalna složenost zaključivanja**: čak i u slučajevima u kojima postupak završava, ne mora završiti u stvarnom vremenu (NP težak problem)
- prethodna loša svojstva mnoge su odbila od korištenja formalnih logika za velike ekspertne sustave

# Suočavanje sa stvarnim svijetom

- Putevi do efikasnog zaključivanja:
  - odustati od formalne korektnosti:  
oblikovati postupak koji se statistički “dobro” ponaša za zanimljive distribucije klauzula
  - odustati od **poluodlučljivosti**  
(postupak ponekad ne dokazuje ispravnu formulu)
  - koristiti **manje izražajan** podskup pune predikatne logike  
(zaključivanje u domeni **Hornovih klauzula** je formalno dobro definirano, a znatno efikasnije)

# Hornove klauzule

- podskup klauzula predikatne logike prvog reda
- **važno**: rezolucijski postupak nad H.k. je **zatvoren** (rezolucija dvije H.k. ponovo je H.k.) te ima **polinomnu** složenost.
- glavno svojstvo: imaju najviše jedan pozitivni literal

$$\neg A_1 \vee \neg A_2 \vee \dots \vee \neg A_n \vee B$$

- mogu se pisati u obliku **produkcijskog pravila** (“ako–onda”):

$$\text{ako } A_1 \wedge A_2 \wedge \dots \wedge A_n \text{ onda } B$$

- Hornova klauzula može imati:
  - niti jedan pozitivni literal (upit),
  - niti jedan negirani literal (činjenica), ili,
  - i pozitivni i negativne literale (pravilo).

# Ograničenja Hornovih klauzula i Prologa

- **glavno ograničenje**: nema disjunkcije na desnoj strani implikacije, odnosno negiranih literala na lijevoj:

**ako**  $A_1 \wedge A_2 \wedge \dots \wedge A_n$  **onda**  $B_1 \vee B_2$

**ako**  $\neg A_1 \wedge A_2 \wedge \dots \wedge A_n$  **onda**  $B$

- definitne Hornove klauzule (činjenice i pravila) izravno se preslikavaju u stavke Prologa:

$B :- A_1 \wedge A_2 \wedge \dots \wedge A_n.$

- **dodatno ograničenje Prologa**: nema negacije na desnoj strani implikacije (odnosno u glavi stavka):

$\neg B :- A_1 \wedge A_2 \wedge \dots \wedge A_n.$

# Produksijska pravila

- **ipak**, produksijska pravila su korisna jer predstavljaju relativno prirodan i intuitivan formalizam za izražavanje ljudskog znanja;
- npr:
  - Ako** je temperatura pacijenta veća od  $38^{\circ}\text{C}$  **onda** treba prepisati lijek za snižavanje temperature.
  - Ako** je svjetlo na semaforu crveno **onda** se zaustavi.
- svako produksijsko pravilo predstavlja jedno zrnce znanja; model ljudske spoznaje temeljen na mozaiku takvih pravila (Newell i Simon, Human problem solving, 1972)
  - dugotrajna memorija: ukupan skup produksijskih pravila
  - kratkotrajna memorija: znanje koje se privremeno pohranjuje tijekom postupka zaključivanja
  - spoznajni proces: pronalazi pravila koja će biti aktivirana, rješava konflikte



# Dobra svojstva produkcijskih pravila

- **modularnost:** pravilo — mali, relativno nezavisan dio znanja;
- **proširivost:** baza znanja lako se proširuje novim pravilima;
- **jednostavno održavanje:** pravila se jednostavno mijenjaju, neovisno o ostalim pravilima;
- **transparentnost:** donesene odluke mogu se obrazložiti; konkretno, moguće je odgovoriti na dvije vrste pitanja:
  1. Kako si došao do tog zaključka?  
(takav podatak je potreban primjerice kod sustava koji imaju savjetodavnu ulogu pri upravljanju složenim postrojenjima npr. nuklearnom elektranom čiji pogrešan rad može imati nesagledive posljedice)
  2. Zašto ti treba taj podatak?  
(ekspertni sustav može tražiti dodatne podatke koji su povezani sa značajnim izdacima, primjerice kod sustava za upravljanje geološkim istraživanjem — korisnik tada može odlučiti opravdava li precizniji odgovor dodatna ulaganja)

# Ekspertni sustavi

- Ekspertni sustav— program koji obavlja posao stručnjaka za neko usko određeno područje
- prednosti računalnih eksperata:
  - sinteza znanja većeg broja ljudskih eksperata
  - nema utjecaja umora i emocija
  - jednostavna replikacija
  - neograničen vijek trajanja,
- Značajna područja primjene:
  - medicinska, elektronička i elektromehanička dijagnostika
  - planiranje prihoda, financije, bankarstvo
  - planiranje eksperimenata u biologiji, kemiji, genetici
  - oblikovanje računalnih i mikroelektroničkih sustava
  - edukacijski sustavi

# Temeljne operacije ekspertnog sustava

- **rješavanje problema** — ekspertni sustav posjeduje i primjenjuje znanje s područja primjene;
- **obrazloženje rješenja** — poput "živih" stručnjaka, ekspertni sustav mora moći pojasniti svoje odluke i zaključke;
- **dijalog s korisnikom** — rješenje se pronalazi interaktivno, korisnik može utjecati na tok rješavanja;
- **rad s nepotpunim i neizrazitim podacima** — potrebno je omogućiti primjenu pravila kao npr. "tjelesna temperatura malo veća od 37° upućuje na upalu pluća".
- **održavanje konzistentnosti** (engl. truth maintenance): razrješavanje konflikata između novih činjenica i postojećeg znanja (monotona logika često nedovoljna)

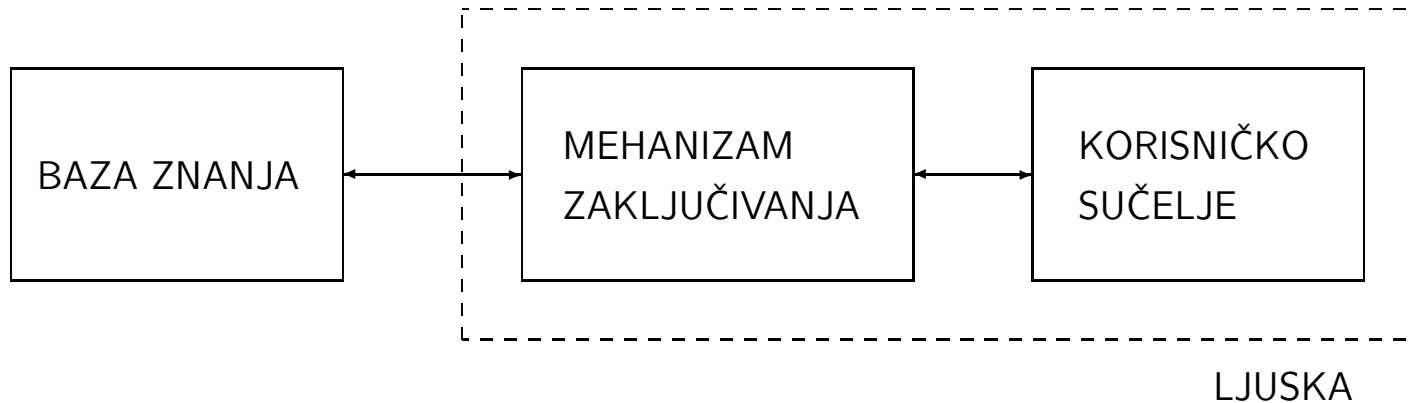
# Monotona naspram nemonotone logike

- **monotona logika:** zaključci dobiveni valjanim zaključivanjem ostaju valjani
- **nemonotona logika:** nove činjenice mogu biti u kontradikciji s postojećim znanjem — time se staro znanje obezvrijeđuje
- Primjer:
  - neka se baza znanja sastoji od  $P, P \rightarrow Q$
  - zaključujemo  $Q$ , zaključak dodajemo u bazu znanja
  - međutim, neka je naknadno nedvosmisleno ustanovljeno  $\neg P$
  - TMS mora ukloniti i  $P$  i  $Q$

# Definicija ekspertnog sustava

- ekspertni sustav definiran je sa:
  - bazom znanja  
(skup produkcijskih pravila)
  - radnom memorijom  
(skup činjenica, opis tekućeg stanja svijeta u postupku zaključivanja)
  - ciklusom podudaranje-djelovanje
    - pronalaženje primjenljivih (omogućenih) pravila  
(formiranje konfliktnog skupa)
    - odabir jednog pravila iz konfliktnog skupa  
(razrješavanje konflikta)
    - izvršavanje pravila  
(upis novodobivenih činjenica u bazu)

# Struktura ekspertnog sustava



- **baza znanja** sadrži specifično znanje o području primjene: činjenice, pravila i metode za rješavanje problema iz domene
- **mehanizam zaključivanja** definira postupak pronalaženja rješenja problema, na temelju elemenata baze znanja.
- **korisničko sučelje** omogućava korisniku uvid u postupak zaključivanja i proširivanje baze znanja.
- ekspertni sustav sastoji se od znanja o domeni (baza znanja) i generičkih postupaka za njegovu primjenu (ljuska)

# Ljuske ekspertnih sustava

- poznatije ljuske ekspertnih sustava: CLIPS, OPS5, EMycin
- osnovni sintaktički oblici literala:
  - (objekt, atribut, vrijednost)  
npr: (moj\_auto, boja, crvena)
  - (atribut, relacija, vrijednost)  
npr: (temperatura, veća\_od, 20)
  - ...
- Npr:
  - **if** (auto ?x) **and** (svjetla ?x slaba)  
**then** (provjeri\_baterije ?x)
  - radna memorija:  
(auto V90) (svjetla V90 slaba)
  - rezultat primjene pravila: (provjeri\_baterije V90)

# Zaključivanje u ekspertnom sustavu

- postupak zaključivanja (ulančavanje) → stvaranje lanaca međuzaključaka koji povezuju opis problema sa zaključkom
- u svakom koraku, postupak zaključivanja proširuje jedan ili više lanaca (hipotetiziranih rješenja problema)
- pojam lanac nije najprecizniji: ne radi se o linearnoj strukturi!
- dva glavna načina napredovanja su:
  - **ulančavanje pravila prema naprijed**: rekurzivno stvaranje svih mogućih zaključaka na temelju poznatih podataka, u nadi da ćemo jednom doći do cilja
  - **ulančavanje unatrag**: zadani zaključak se pokušava poduprijeti na sve moguće načine sve dok ne dođemo do znanih podataka ili ne iscrpimo sve mogućnosti



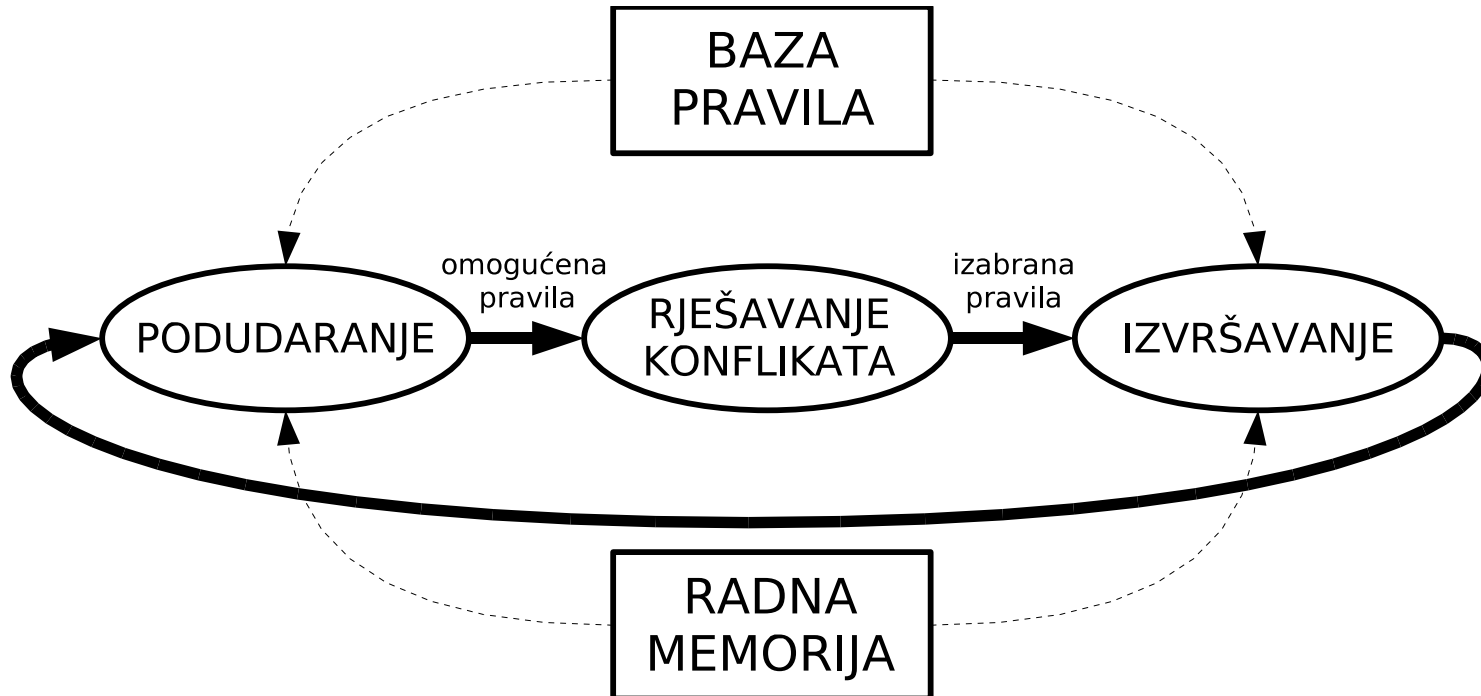
# Ulančavanje unaprijed

- za ovakav proces kažemo da je vođen podacima (dogadajima), odnosno organiziran od dna prema vrhu
- metoda izbora kad imamo malo početnih podataka i mali faktor grananja lanaca (premise se u prosjeku nalaze u lijevoj strani malog broja pravila), te puno prihvatljivih zaključaka
- tipičan primjer: automatsko oblikovanje računalnog sustava prema potrebama korisnika (XCON, DEC)
- prikladne problemske domene: oblikovanje, planiranje, raspoređivanje, nadzor, dijagnostika
- prikladno kad god je zadatak sustava inherentno kreativan (potrebno sintetizirati nove tvorevine ili ponašanja)
- nedostatak: potencijalno veliki zahtjevi za radnom memorijom ukoliko postupak zastrani (razmatra duge lance koji ne vode do cilja)

# Ulančavanje unatrag

- ulančavanje je vođeno hipotetiziranim ciljem (organizacija od vrha prema dnu)
- metoda izbora kada ima malo zanimljivih zaključaka, te mnogo početnih podataka uz veliko grananje (premise se prosječno nalaze u lijevim stranama većeg broja pravila)
- tipičan primjer: programski jezik Prolog
- prikladne problemske domene: klasifikacija, edukacija, dijagnostika
- prikladno kad želimo potvrditi ili opovrgnuti relativno malen broj unaprijed poznatih hipoteza
- nedostatak: nemogućnost kreiranja originalnih hipoteza

# Ulančavanje unaprijed



- Postupak stroja za zaključivanje (engl. inference engine):
  - podudaranje činjenica iz RM s LHS svih pravila
  - razrješavanje konflikta: odabir jednog pravila iz s.o.p.
  - izvršavanje pravila: rezultira novom činjenicom (pravilom) ili nekom drugom operacijom

# Ulančavanje unaprijed – primjer

- zadan je produkcijski sustav za sortiranje nizova slova a,b i c
- produkcijska pravila: (1)  $ba \rightarrow ab$ ; (2)  $ca \rightarrow ac$ ; (3)  $cb \rightarrow bc$ .
- postupak zaključivanja:

korak	radna memorija	konfliktni skup	odabrano pravilo
0	cbaca	3,1,2	1
1	cabca	2	2
2	acbca	3,2	2
3	acbac	3,1	1
4	acabc	2	2
5	aacbc	3	3
6	aabcc	$\emptyset$	stop

# Ulančavanje unaprijed – primjer

- skup pravila:
  1.  $p \wedge q \rightarrow \text{cilj}$
- strategija rješavanja konflikta: odabir pravila koje do sada nije palilo
  2.  $r \wedge s \rightarrow p$
  3.  $w \wedge r \rightarrow q$
  4.  $t \wedge u \rightarrow q$
  5.  $v \rightarrow s$
- postupak rješavanja:
  6. početak  $\rightarrow v \wedge r \wedge q$

	radna memorija	konfliktni skup	odabir
0	početak	6	6
1	početak, v, r, q	6, 5	5
2	početak, v, r, q, s	6, 5, 2	2
3	početak, v, r, q, s, p	6, 5, 2, 1	1
4	početak, v, r, q, s, p, cilj	6, 5, 2, 1	stop

# Ulančavanje unaprijed – primjer

- skup pravila:
  - 3 ako  $\text{oblik}=\text{okrugao} \wedge \text{promjer}<10\text{cm}$   
onda voćka=stablo
- strategija: odabir pravila s najmanjim rednim brojem
  - 4 ako  $\text{broj\_sjemenki}=1$   
onda sjemenka=koštunjasta
  - 11 ako  $\text{voćka}=\text{stablo} \wedge \text{boja}=\text{crvena} \wedge$   
 $\text{sjemenka}=\text{koštunjasta}$   
onda voće=višnja
- rješavanje:

	radna memorija	konfliktni skup	odabir
0	promjer=2cm,oblik=okrugao, broj_sjemenki=1,boja=crvena	3,4	3
1	voćka=stablo	3,4	4
2	sjemenka=koštunjasta	3,4,11	11
3	voće=višnja	3,4,11	stop

# Algoritam rete

- faza podudaranja je kritična za performansu sustava sa ulančavanjem unaprijed
- pretpostavke:
  1. u sustavu ima ukupno  $r$  pravila i  $f$  činjenica
  2. pravila u prosjeku imaju  $p$  premisa u lijevoj strani
  3. (tipično,  $r = 150$ ;  $f = 20$ ;  $p = 4$ )
- složenost podudaranja:  $r \cdot f^p$  usporedbi ( $24 \cdot 10^6!$ )
- složenost podudaranja se može smanjiti pod pretpostavkama:
  - više pravila u radnoj memoriji imaju slične premise (uvjeti takvih pravila mogu se usporedno provjeravati) 2.
  - radna memorija se tijekom ciklusa značajno ne mijenja (znanje o sličnostima može se inkrementalno obnavljati)

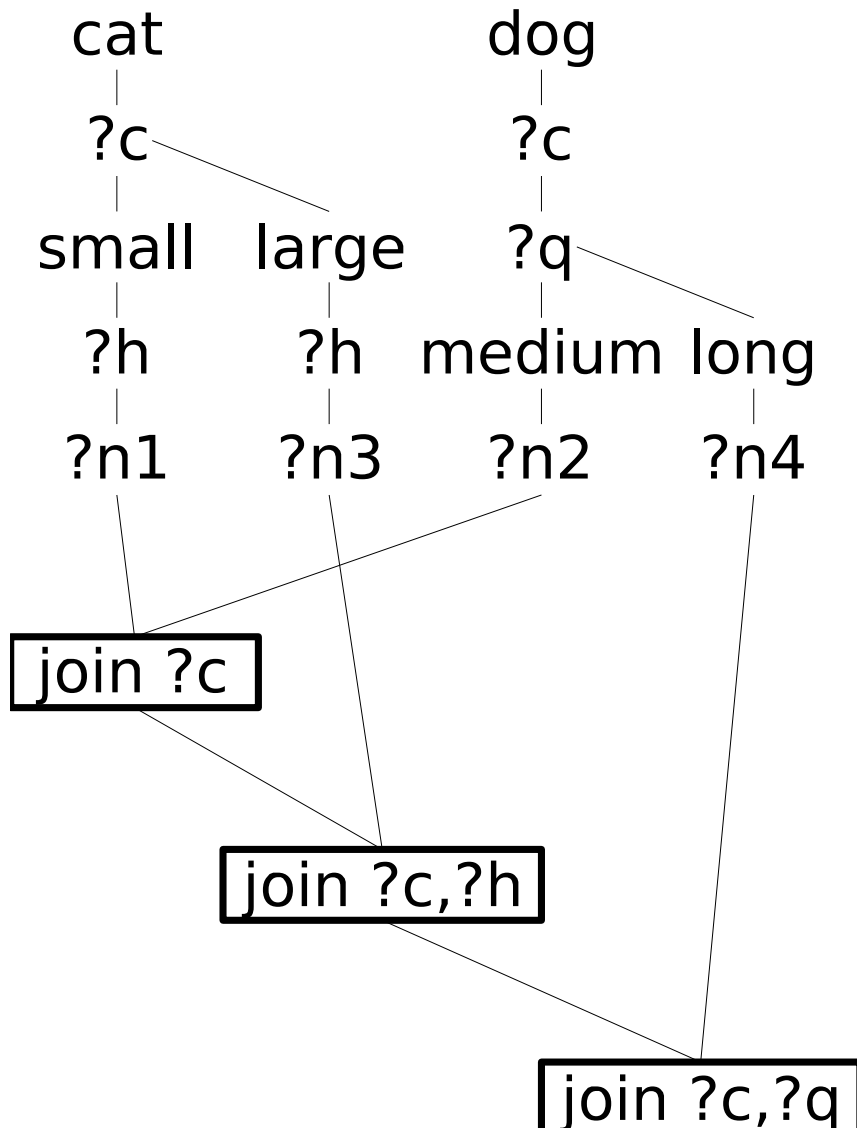
# Algoritam rete - primjer

- znanje o sličnostima uvjeta izražava se:
  1. **mrežom uzoraka** (skup stabala izgrađenih od svih premisa)
  2. **mrežom udruživanja** (veze između varijabli pojedinih premisa, za svako pravilo)
- za primjer, razmotrimo pravilo u kojima su premise oblika (`animal-type, color, size, hair-length, name`):

```
if ((cat,?c,small,?h, ?n1)
    (dog,?c,?q ,medium,?n2)
    (cat,?c,large,?h, ?n3)
    (dog,?c,?q ,long, ?n4))
then ...
```
- pravilo bi imalo smisla koristiti u ekspertnom sustavu za planiranje obaveza životinja-glumaca :-)



# Algoritam rete - primjer



- izgrađene mreže sadrže informaciju o strukturalnoj sličnosti pojedinih premisa: ta informacija može dramatično smanjiti broj usporedbi u fazi podudaranja
- za svako pravilo, podatak o omogućenosti dobiva se iz odgovarajućeg najdonjeg čvora mreže udruživanja

# Načini rješavanja konflikta

- važan činioc: broj pravila koja se istovremeno izvršavaju (jedno, više njih ili sva)
- rješavanje konflikta prema uređaju pravila:
  - odabir prvog pravila
  - odabir prvog pravila koje daje nove činjenice
  - odabir pravila koje slijedi prethodno odabranom
- prema složenosti pravila (broju premisa): odabir najsloženijeg ili najjednostavnijeg pravila
- prema uređenju podataka: odabir pravila koje se podudara s najstarijim odnosno najnovijim podacima
- upotrebom meta-produkcijskih pravila

# Zaključivanje ulančavanjem unatrag

- implementacija ovakvog zaključivanja najčešće se temelji na stogu hipoteza (Prolog + radna memorija)
- početno, na stogu se nalazi samo hipoteza koju valja dokazati; postupak uspješno završava kada se stog isprazni
- u svakoj iteraciji, skida se cilj s vrha stoga i:
  - provjerava se da li se cilj nalazi u bazi znanja: ako da, vraća se vrijednost cilja (**istina** + varijable ili **laž**)
  - ukoliko je cilj “interaktivan”, zahtijeva se od korisnika da navede ili istinitost, ili vrijednosti varijabli uz koje cilj vrijedi
  - za svako pravilo čija se desna strana podudara s ciljem, premise pravila se postavljaju na stog
  - inače, cilj se ne može dokazati, vraća se **laž**.
- **vraćanje** (engl. backtracking): kad postupak dođe u “slijepu” ulicu, hipoteza se pokušava zadovoljiti na neki drugi način

# Ulančavanje unatrag – primjer

- pravila iz baze za raspoznavanje životinja:

```
rule1::
  if
    Animal has hair
  or
    Animal gives milk
  then
    Animal isa mammal.

rule3::
  if
    Animal isa mammal and
  (
    Animal eats meat
  or
    Animal has pointedTeeth and
    Animal has claws and
    Animal has forwardPointingEyes
  )
  then
    Animal isa carnivore.

rule5::
  if
    Animal isa carnivore and
    Animal has tawnyColour and
    Animal has blackStripes
  then
    Animal isa tiger.

upit(_ eats _, 'Animal' eats 'What').
upit(_ has _, 'Animal' has 'Property').
```

- interaktivni ciljevi definirani su predikatom upit/2
- <http://www.zemris.fer.hr/education/is/lv/lab/A/index.php>

# Ulančavanje unatrag – primjer

% primjer dijaloga sa ekspertnim sustavom

Question, please:

mirko isa tiger.

Is it true: mirko has hair?

yes.

Is it true: mirko eats meat?

no.

Is it true: mirko has pointedTeeth?

yes.

Is it true: mirko has claws?

why.

To investigate, by rule3, mirko isa carnivore

To investigate, by rule5, mirko isa tiger

This was your question.

<...>

mirko isa tiger is true

Would you like to see how?

yes.

mirko isa tiger

was derivedBy rule5 from

mirko isa carnivore

was derivedBy rule3 from

mirko isa mammal

was derivedBy rule1 from

mirko has hair

was told

and

mirko has pointedTeeth

was told

and

mirko has claws

was told

and

mirko has forwardPointingEyes

was told

and

mirko has tawnyColour

was told

and

mirko has blackStripes

was told