Uvod u umjetnu inteligenciju

8. Ekspertni sustavi

prof. dr. sc. Jan Šnajder
izv. prof. dr. .sc. Marko Čupić
prof. dr. sc. Bojana Dalbelo Bašić

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva

Ak. god. 2021./2022.

Outline

Sadržaj

GPS vs. ES

- Rana UI (1950. i 1960.) bila je usredotočena na razvoj sofisticiranih postupaka zaključivanja (npr., automatsko dokazivanje teorema)
- Cilj ranih sustava bilo je **rješavanje općih problema** (engl. *general problem solving, GPS*).
- Rani UI sustavi:
  - nisu se oslanjali na velike količine znanja specifičnog za neki zadatak
  - pokušali su ostvariti inteligenciju a da nužno ne modeliraju ljudsko rasuđivanje
  - intenzivno su se oslanjali na **pretraživanje**
- Neuspjeh razvoja GPS-a doveo je do nastanka alternativnog pristupa: **sustavi temeljeni na znanju** (engl. *knowledge based systems*) ili **ekspertni sustavi**

Ljudsko znanje

- Sustavi temeljeni na znanju teže eksplicitno obuhvatiti što ljudi znaju i kako koriste to znanje
- Kako najjednostavnije prikazati **ljudsko znanje**?
- Većina ljudskog znanja (ekspertnog znanja) može se prikazati pomoću **ako–onda pravila**

Pravilo 1

Ako je temporatura pacijenta veća od 38°C, onda treba propisati lijekove za snižavanje povišene tjelesne temperature.

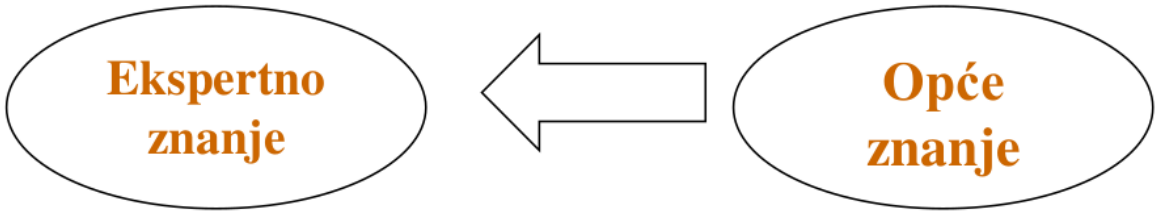
Pravilo 2

Ako je na semaforu crveno svjetlo, onda se zaustavi.

- Ova se pravila također nazivaju **produkcijska pravila**. Sustavi koji ih koriste nazivaju se **produkcijski sustavi**, **sustavi temeljeni na znanju** ili **ekspertni sustavi**

Opće vs. ekspertno znanje

- Važan korak za uspješno rješavanje AI problema: smanjivanje opsega (domene) problema



- **Ekspertno znanje** (domensko znanje) je znanje **usko usredotočeno na specifičnu domenu** (npr., medicina, financije, šah, i sl.), za razliku od znanja potrebnog za rješavanje općih problema
- Ne teži se ostvariti širinu niti općenitost!

Intelektualno kloniranje

Opća namjera jedna je vrsta **intelektualnog kloniranja**: pronađite osobe sa **znanjem koje je važno i rijetko** (npr., stručni medicinski dijagnostičar, šahist, kemičar), razgovarajte s njima da biste utvrdili koje to specijalizirano znanje posjeduju i na koji način razmišljaju, a zatim to **znanje i razmišljanje utjelovite u računalni program**.

(MITECTS, p430)

Ekspertni sustavi vs. drugi pristupi

- **Ljudsko rasuđivanje vs. dokazivanje teorema**
  - ES ne razmišljaju značajno brže niti drugačije od običnih ljudi
  - Njihova ekspertiza posljedica je toga što imaju mnogo više znanja o konkretnom zadatku od običnog čovjeka
- **Znanje vs. pretraživanje**
  - Rani sustavi pokušali su zaobići potrebu za znanjem primjenom učinkovitih tehnika pretraživanja (npr., igranje šaha)
  - ekspertni sustavi ubrzo su pokazali da znanje može eliminirati potrebu za pretraživanjem
  - Kombinacija pretraživanja i znanja: pretraživanje usmjereno znanjem (e.g., DENDRAL)

Znanje vs. pretraživanje: DENDRAL

- DENDRAL ("Dendritic Algorithm"): ES za **analitičku kemiju** razvijen na Sveučilištu Stanford 1960-ih, u svrhu određivanje strukture kemijskog spoja iz njegovog masene spektrometrije

- **Generiraj-i-testiraj** (engl. *generate-and-test*): generiranje svih mogućih struktura i testiranje svake od njih kako bi utvrdio daje li opaženi maseni spektar. Međutim, to je vrlo brzo dovelo do **kombinatoričke eksplozije**
- U suradnji sa **stručnjacima kemičarima**, tvorci DENDRAL-a uspjeli su odrediti koje znakove kemičari uočavaju u spektru, što je omogućilo da se **pretraživanje usredotoči** na određene podvrste molekula

Sadržaj

Temeljne ideje

1. **Produkcijski sustavi**
   - Logičar E. Post 1943. predlaže produkcijska pravila kao formalan model izračunavanja
   - sustav za manipulaciju nizovima znakova koji, krenuvši od konačno mnogo nizova, opetovano transformira nizove primjenjujući pravila iz predefiniranog konačnog skupa pravila
   - Postov kanonski sustav svediv je na sustav prepisivanja nizova (engl. *string rewriting system*). Oba su formalizma Turing-potpuna
2. **Ljudski kognitivni proces**
   - Knjiga Allena Newella i Herberta A. Simona "Human problem solving" iz 1972.: ljudska kognicija može se modelirati ako–onda pravilima
   - jedno pravilo je jedna granula znanja (engl. *chunk of knowledge*)
   - osjeti stimuliraju mozak podržajima, koji aktiviraju znanje u našoj dugoročnoj memoriji $\Rightarrow$ ako–onda pravila
   - kratkoročna memorija pohranjuje privremeno znanje nastalo zaključivanjem (granule znanja koje možemo istovremeno razmatrati) $\Rightarrow$ nove činjenice izvedene pravilima

Rani ekspertni sustavi

- DENDRAL (Feigenbaum i dr. 1971)
  - otkrivanje molekularne strukture spojeva na temelju spektrometrije
- MYCIN (Davis 1977)
  - dijagnosticiranje i preporučivanje liječenja bakterijskih krvnih infekcija
  - oko 450 relativno nezavisnih ako–onda pravila, zaključivanje ulančavanjem unazad
  - prvi sustav sa svim karakteristikama koje su obilježile kasnije ekspertne sustave
- XCON/R1 (McDermott 1978)
  - sustav za konfiguriranje računala DEC VAX i PDP-11
  - oko 10,000 pravila, zaključivanje ulančavanjem unaprijed
- PROSPECTOR (Duda 1979)
  - procjena postojanja rudače na temelju geografskih karakteristika
  - koristi neizrazitu logiku (engl. *fuzzy logic*) za modeliranje nesigurnih zaključaka $\Rightarrow$ TBD iduće predavanje

Ekspertni sustavi: Uspjeh ili neuspjeh?

- Početak/sredina 1980-ih: prvi val komercijalnih ekspertnih sustava
- Razvoj je skup i nespretan (napredniji sustavi iziskuju na tisuće vrlo skupih čovjek-sati)
- U razdoblju od 1987. do 1992. od većine se sustava odustalo, no iz razloga koji nisu tehničke ni ekonomske prirode (menadžerski i oganizacijski problemi, npr., usklađivanje razvoja ekspertnih sustava s poslovom strategijom, pravni problemi, itd.)

$\Rightarrow$ Uspjeh ekspertnih sustava u tehničkome smislu ne jamči njihovu široku prihvaćenost niti dugoročnu uporabu

- Pa ipak, mišljenja o uspjehu ekspertnih sustava se razilaze. . .

Ekspertni sustavi: Uspjeh ili neuspjeh?

In the 1990s and beyond, the term expert system and the idea of a standalone AI system mostly dropped from the IT lexicon. There are two interpretations of this. One is that **expert systems failed**: the IT world moved on because expert systems did not deliver on their over hyped promise. The other is the mirror opposite, that expert systems were simply **victims of their success**: as IT professionals grasped concepts such as rule engines, such tools migrated from being standalone tools for developing special purpose expert systems, to being one of many standard tools.

(Wikipedia: Expert system)

Once touted as potentially revolutionizing business operations, you don't hear much about expert systems these days, although proponents claim **more enterprises than you might expect have adopted them**.

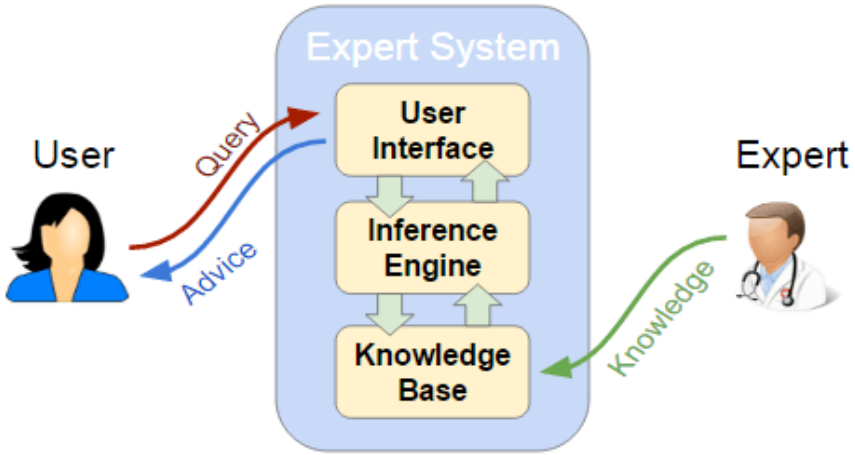
(D. Haskin, "Years After Hype, 'Expert Systems' Paying Off For Some")

Sadržaj

Stroj za zaključivanje vs. baza znanja

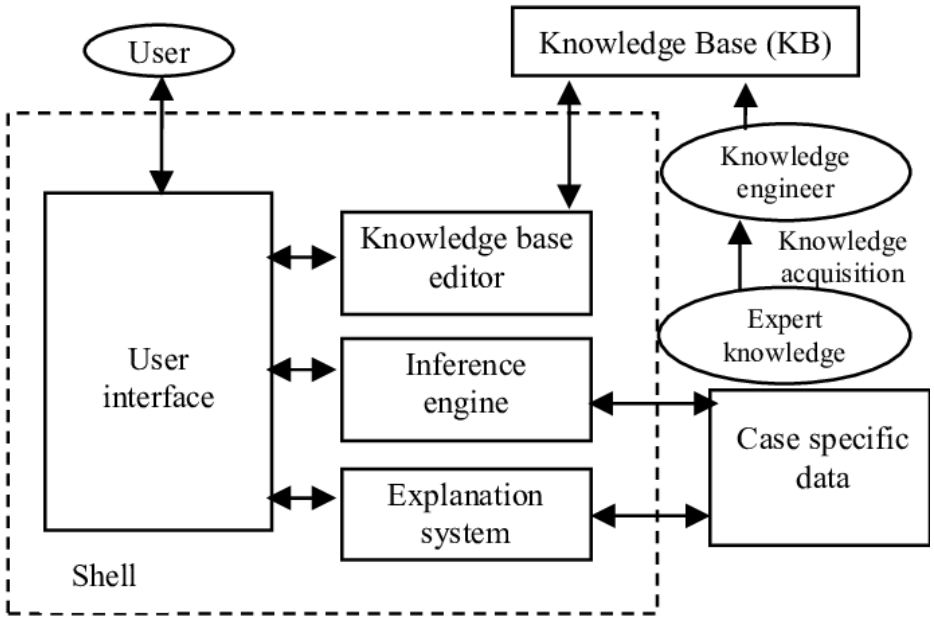
Ekpertni sustavi koriste različite tehnologije za prikazivanje znanja, no svima su zajedničke dvije arhitekturne karakteristike:

1. Razlikovanje između **stroja za zaključivanje** (engl. *inference engine*) i **baze znanja** (engl. *knowledge base*)
   - stroj za zakljčivanje dohvaća pravila i činjenice iz baze znanja
2. Uporaba **deklarativnog stila prikazivanja znanja**
   - pravila su podatkovne strukture s vlastitom semantikom, a ne dio programskog koda u kojem je implementirano stroj za zaključivanje



Ljuska ekspertnog sustava

- Stroj za zaključivanje odvojen je od baze znanja $\Rightarrow$ različite baze mogu se koristiti kao "plug-inovi"
- **Ljuska ekspertnog sustava** (engl. *expert system shell*): alat za izgradnju ekspertnog sustava
  - stroj za zaključivanje
  - uređivač baze znanja
  - korisničko sučelje i modul za objašnjavanje zaključka

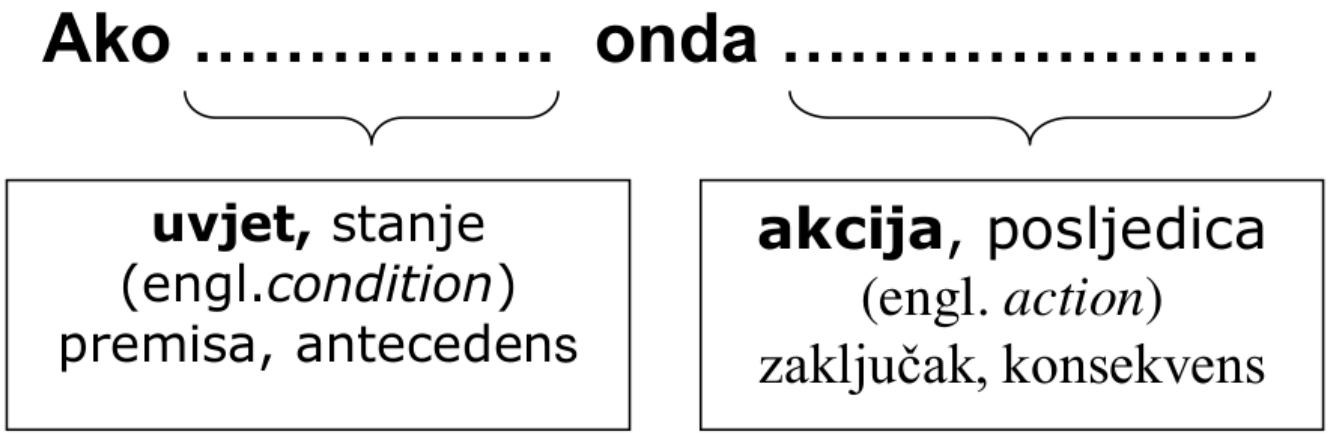


Popularne ljuske ekspertnih sustava

- CLIPS ("C Language Integrated Production System"), NASA (1985)
  - najšire korištena ljuska, implementirana u C-u
- JESS ("Java Expert System Shell"), Friedman-Hill & Ernest (2003)
  - verzija CLIPS-a u Javi
- PyCLIPS
  - sučelje prema CLIPS-u iz Pythona
- PyKE ("Python Knowledge Engine")
  - Prologom inspirirana ljuska implementirana u Pythonu
- ES-builder
  - edukacijski eksperti sustav, za zaključivanje koristi Prolog

Ako–onda pravila

- Znanje u ekspertnim sustavima prikazano je u obliku ako–onda pravila (produkcijska pravila)



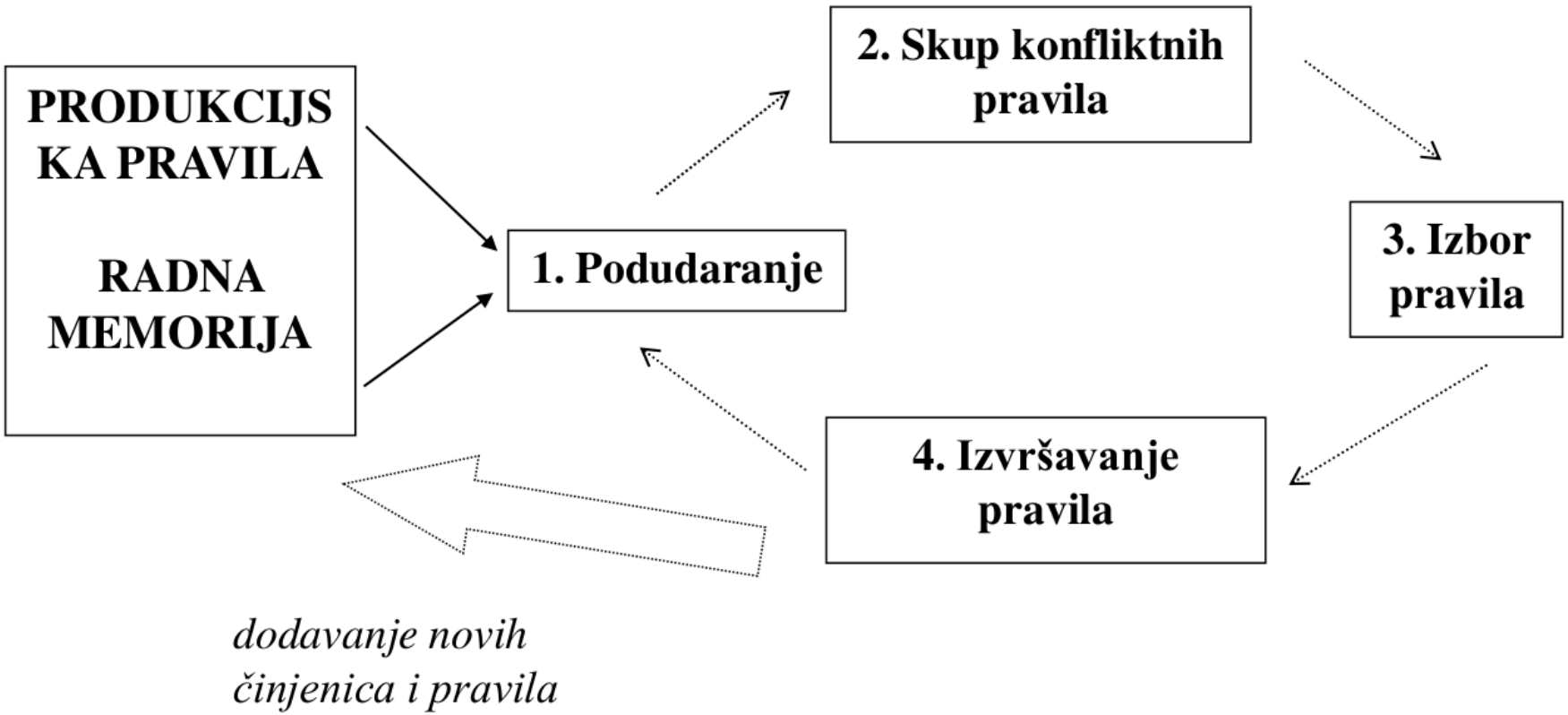
- Nalikuje **implikaciji** u logici ($A \rightarrow B$), ali dvije su ključne razlike:
  - implikacija je formula logike koja ima istinosnu vrijednost
  - konzekvent implikacije B također je formula, dok je konzekvent ako–onda pravila akcija (dodavanje nove činjenice, ali i brisanje činjenica, izvođenje koda, npr., ispis na ekran, itd.)

Sadržaj

Komponente postupka zaključivanja

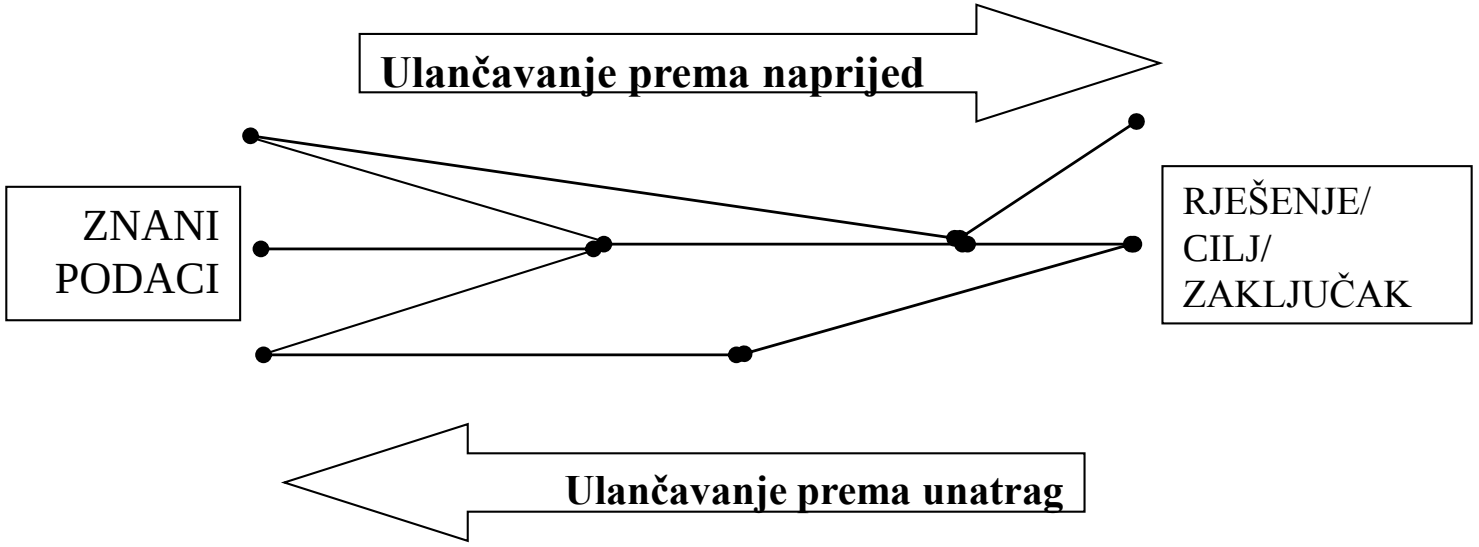
- **Radna memorija** (engl. *working memory*) je dio baze znanja koji:
  - pohranjuje činjenice koje je dodao korisnik prije pokretanja zaključivanja ili nove činjenice izvedene postupkom zaključivanja
  - ne pohranjuje ih trajno (kao kratkoročna memorija kod ljudi)
- **Stroj za zaključivanje** je upravljački mehanizam koji izvodi:
  - **podudaranje** – podudaranje činjenica iz baze znanja s lijevom stranom (uvjetom) ako–onda pravila (koristi učinkovite tehnike za podudaranje uzoraka, npr., algoritam Rete)
  - **razrješavanje konflikta** – ako je omogućeno više od jednog pravila, odabire jedno od njih, npr., pravilo s najvišim predefiniranim prioritetom (engl. "salience")
  - **primjena pravila** ("paljenje pravila") – izvršenje desne strane (akcija) ako–onda pravila, što rezultira dodavanjem novih činjenica (engl. "assertion") u radnu memoriju, brisanjem činjenica (engl. *"retraction"*) iz radne memorije ili nekom drugom akcijom (u bazu znanja također se mogu dodavati nova pravila)

Ciklus zaključivanja



14. PROCES ZAKLJUČIVANJA U SUSTAVU TEMELJENOM NA PRAVILIMA

- Niz sastavljen od višestrukih zaključaka koji povezuju dani početni opis problema s rješenjem naziva se **LANAC**



- Postupak zaključivanja, tj. automatsko napredovanje kroz lanac, naziva se ulančavanje

14. PROCES ZAKLJUČIVANJA U SUSTAVU TEMELJENOM NA PRAVILIMA

Postoje dva glavna načina napredovanja prema zaključcima:

1. **ULANČAVANJE PRAVILA PREMA NAPRIJED**

- → započinjanje sa znanim podacima i napredovanje prema zaključku
  (*engl.* ***forward chaining (forchaining), data driven processing***, event driven, bottom-up, antecedent, pattern directed **processing ⇔ reasoning)**

2. **ULANČAVANJE PRAVILA UNATRAG**

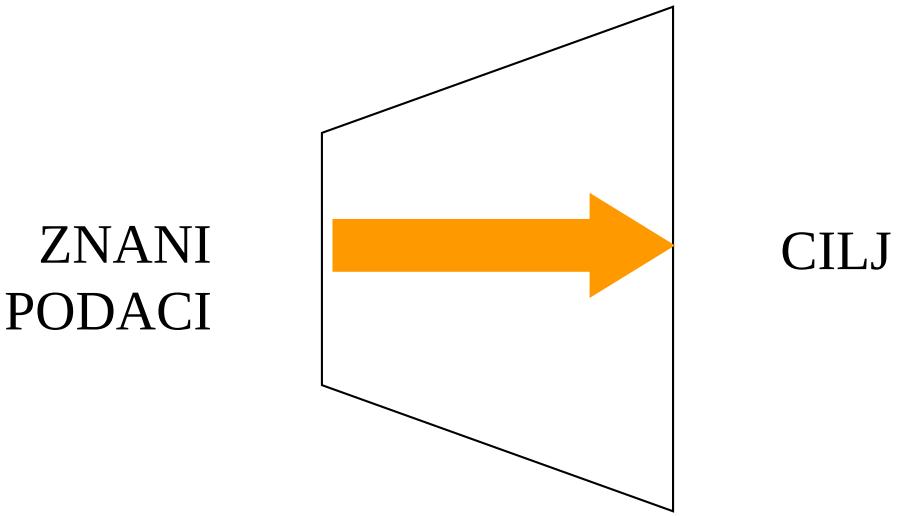
- → izbor mogućeg zaključka (hipoteza) i pokušaj dokazivanja valjanosti hipoteze traženjem valjanih potpora (dokaza, engl. *evidence*).

  (engl. ***backward chaining (backchaining), goal driven processing***, goal driven, top-down, consequent, expectation driven processing)

14. PROCES ZAKLJUČIVANJA U SUSTAVU TEMELJENOM NA PRAVILIMA

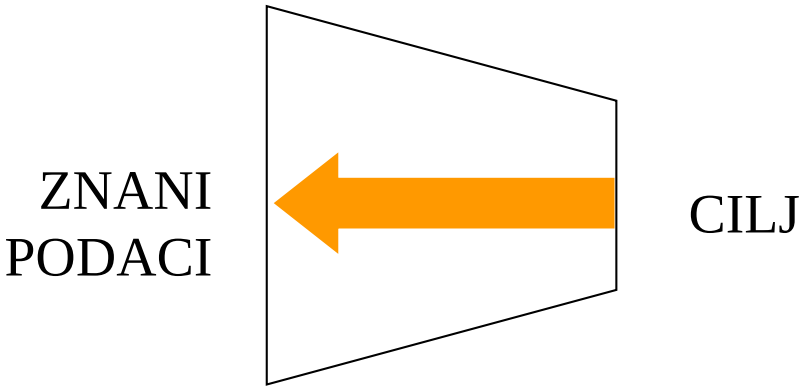
- Ulančavanje prema naprijed - kada ima malo podataka i puno mogućih rješenja

  (za problemske domene koje uključuju sintezu:
  za dizajniranje, planiranje, raspoređivanje, za nadzor i dijagnostiku sustava za rad u stvarnim vremenu)



14. PROCES ZAKLJUČIVANJA U SUSTAVU TEMELJENOM NA PRAVILIMA

- Ulančavanje unatrag razuman je izbor kada je malo mogućih zaključaka/ciljeva i puno znanih podataka. (Za probleme dijagnosticiranja, klasificiranja)



- Izbor metode zaključivanja ovisi o osobinama problemske domene i o načinu zaključivanja eksperta
- Moguće je implementirati i obostrano (bidirekcionalno) zaključivanje

15. ULANČAVANJE PREMA NAPRIJED

Primjer

- Baza pravila za određivanje vrsta voća
- Parametri (tj. varijable) i njihove vrijednosti su:

Oblik:	izdužen		Promjer:	> 10 cm
	okrugli			< 10 cm
	zaoblen		Vrsta_voćke:	loza
Površina:	glatka			stablo
	hrapava		Voće:	banana
Boja:	zelena			lubenica
	žuta			dinja
	žuto-smeđa			kanalupe
	crvena			jabuka
	plava			marelica
	narančasta			višnja
Broj_sjemenki	> 1			breskva
	= 1			šljiva
Vrsta_sjemenke	višestruke			naranča
	koštunjasta			

15. ULANČAVANJE PREMA NAPRIJED

PRAVILA

1	AKO	Oblik = izdužen & Boja = zelena ili žuta
	ONDA	Voće = banana
2	AKO	Oblik = okrugli ili zaobljen & Promjer > 10 cm
	ONDA	Vrsta_voćke = loza
3	AKO	Oblik = okrugli & Promjer < 10 cm
	ONDA	Vrsta_voćke = stablo
4	AKO	Broj_sjemenki = 1
	ONDA	Vrsta_sjemenke = koštunjasta
5	AKO	Broj_sjemenki > 1
	ONDA	Vrsta_sjemenke = višestruke
6	AKO	Vrsta_voćke = loza & Boja = zelena
	ONDA	Voće = lubenica

15. ULANČAVANJE PREMA NAPRIJED

7	AKO	Vrsta_voćke = loza & Površina = glatka & Boja = žuta
	ONDA	Voće = dinja
8	AKO	Vrsta_voćke = loza & Površina = hrapava & Boja = žuto-smeđa
	ONDA	Voće = kantalupe
9	AKO	Vrsta_voćke = stablo & Boja = narančasta & Vrsta_sjemenke = koštunjasta
	ONDA	Voće = marelica
10	AKO	Vrsta_voćke = stablo & Boja = narančasta & Vrsta _sjemenke = višestruke
	ONDA	Voće = naranča
11	AKO	Vrsta_voćke = stablo & Boja = crvena & Vrsta_sjemenke = koštunjasta
	ONDA	Voće = višnja
12	AKO	Vrsta_voćke = stablo & Boja = narančasta & Vrsta _sjemenke = koštunjasta
	ONDA	Voće = breskva
13	AKO	Vrsta_voćke = stablo & Boja = žuta ili zelena ili crvena & Vrsta _sjemenke = višestruke
	ONDA	Voće = jabuka
14	AKO	Vrsta_voćke = stablo & Boja = plava & Vrsta _sjemenke = koštunjasta
	ONDA	Voće = šljiva

15. ULANČAVANJE PREMA NAPRIJED

- Znani podaci: Promjer = 2 cm
  Oblik = okrugli
  Broj_sjemenki = 1
  Boja = crvena

Strategija izbora pravila: pravilo s najmanjim brojem

	Radna memorija	*Skup konfliktnih pravila*	*Pravilo koje pali*
0	Promjer = 2 cm Oblik = okrugli Broj_sjemenki = 1 Boja = crvena	3,4	3
1	Vrsta_voćke: stablo	3, 4	4
2	Vrsta_sjemenke=koštunjasta	3, 4, 11	11
3	Voće = višnja	3, 4, 11	STOP

18. ULANČAVANJE PRAVILA UNATRAG

Ulančavanje unatrag

- bitno se razlikuje od ulančavanja unaprijed iako oba načina zaključivanja ispituju i primjenjuju pravila

- započinje sa željenim ciljem (hipoteza) i ispituje se da li postojeće činjenice podržavaju izvođenje vrijednosti za taj zaključak

- Sistem započinje praznom bazom činjenica. Zadaje se lista ciljeva za koju sustav pokušava izvesti vrijednosti

18. ULANČAVANJE PRAVILA UNATRAG

Koraci:

1. Oblikuj stog inicijalno sastavljen od najvažnijih ciljeva (hipoteza) koje treba dokazati

2. Na vrhu stoga je hipoteza koju treba dokazati.
   Ako je stog prazan, onda je KRAJ

3. Izdvoji sva pravila koja mogu zadovoljavati dani cilj (tj. izdvoji sva pravila čija se DESNA strana podudara s ciljem)

18. ULANČAVANJE PRAVILA UNATRAG

4. Za svako od tih pravila učini redom:

a) **Ako** su sve premise pravila zadovoljene (svaki parametar premise ima vrijednost sadržanu u radnoj memoriji)

tada izvrši pravilo, tj. DESNU stranu tog pravila, tj. dodaj zaključke u radnu memoriju. Ne razmatraj više pravila za taj cilj - vrijednost cilja upravo je izvedena paljenjem tog pravila.

Ako je cilj bio vršni cilj **tada** ukloni cilj sa stoga & vrati se na korak 2.

Ako je cilj bio međucilj **tada** ukloni cilj sa stoga & vrati se privremeno suspendiranom cilju

18. ULANČAVANJE PRAVILA UNATRAG

b) **Ako** se vrijednost parametra nađena u memoriji ne podudara sa vrijednošću parametra premise

onda ne izvršavaj to pravilo

c) **Ako** premise pravila nisu zadovoljene zato jer jedna od parametarskih vrijednosti te premise nije u radnoj memoriji,

tada potraži pravilo čija desna strana izvodi vrijednost tog parametra.

Ako barem jedno takvo pravilo postoji

tada odredi taj parametar kao podcilj, tj.

postavi taj parametar na vrh stoga &

idi na korak 2

18. ULANČAVANJE PRAVILA UNATRAG

d) **Ako** korak (c) ne može naći pravilo koje izvodi potrebnu vrijednost tekućeg parametra

tada pitaj korisnika za tu vrijednost parametra & dodaj vrijednost u radnu memoriju.

Idi na korak 4a i razmatraj sljedeću premisu tekućeg pravila

5. **Ako** su sva pravila koja mogu zadovoljavati tekući cilj provjerena i ako ni jedno nije uspjelo izvesti vrijednost cilja

tada cilj ostaje neodređen. Makni cilj sa stoga i prijeđi na korak 2

18. ULANČAVANJE PRAVILA UNATRAG

Primjer ulančavanja unatrag

- Neka je naša početna hipoteza da se radi o komadu voća
- **hipoteza = (voće).** Imajući na umu da se radi o višnji - slijedimo sada rad sustava ulančavanjem unatrag, da bi vidjeli je li moguće izvesti da je voće višnja
- U drugom koraku izdvajamo SVA pravila koja izvode vrijednost za voće

18. ULANČAVANJE PRAVILA UNATRAG

Cilj (stog)	**RADNA MEMORIJA**		**KONFLIKTNI SKUP**	**parametar koji se provjerava**
(**voće**)			**1**,6,7,8,9,10,11,12, 13,14	oblik nije u RM i nije na RHS od nekog pravila - **?oblik?**
(voće)	oblik = okrugli	(~~R1~~)	**6**,7,8,9,10,11, 12,13,14	vrsta_voćke –RHS od pravila 2 i 3
(**vrsta_voćke** je trenutni cilj; cilj **voće** je privremeno suspendiran)			2,3,6,7,8,9,10,11, 12,13,14	oblik – nalazi se u RM – OK promjer – nije u RM i nije na RHS ni jednog **pravila** **?promjer?**
(vrsta_voćke voće)	promjer < 10 cm	(~~R2~~)	**3-PALI,** 6,7,8,9,10,11, 2,13,14	obje premise od R3 su u RM
(**voće**)	vrsta_voćke= stablo		**6**,7,8,9,10,11, 12,13,14	prva premisa od R6 nije zadovoljena
(voće)		(~~R6~~)	**7**,8,9,10,11, 12,13,14	prva premisa od R7 nije zadovoljena
(voće)		(~~R7~~)	**8**,9,10,11, 12,13,14	prva premisa od R8 nije zadovoljena

18. ULANČAVANJE PRAVILA UNATRAG

(voće)		(~~R8~~)	**9**,10,11, 12,13,14	vrsta_voćke je u RM; 2. premisa nije RM i nije na RHS od ni jednog pravila **? boja ?**
(**voće**)	boja = crvena	(~~R9~~)	**10**,11, 12,13,14	druga premisa od R10 nije zadovoljena
(**voće**)		(~~R10~~)	11, 12,13,14	
….	…	..	…	…
(**voće**)			**11 - PALI**	voće=višnja

? … ? – upit korisniku. Sva uklonjena pravila sa vrha stoga za koje ne piše da pale nisu bila zadovoljena. Ulančavanje ide unatrag sve dok se desni dio pravila pojavljuje u nekom drugom pravilu na lijevoj strani

Sadržaj

EKSPERTNI SUSTAVI

Kroz primjere

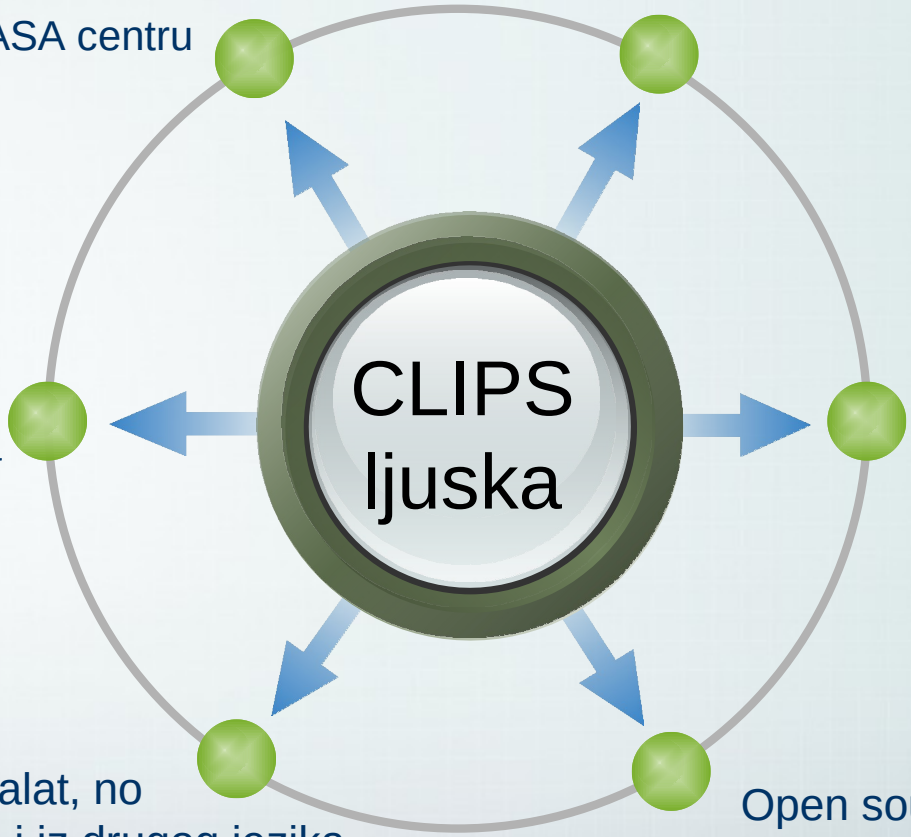
Marin Japec

“I'm sorry Dave, I'm afraid I can't do that. ***Hall 9000***

CLIPS

Sintaksa nalik LISP-u (okrugle zagrade kao delimiteri)

Razvijen 1984, u NASA centru



Više paradigmatsko programsko okruženje:
(3 načina prikaza znanja)

1. pravilima
2. proceduralno
3. objektno-orijentirano

Potpuno okruženje za razvoj ekspertnih sustava

Radi kao samostalan alat, no moguće ga je pozvati i iz drugog jezika

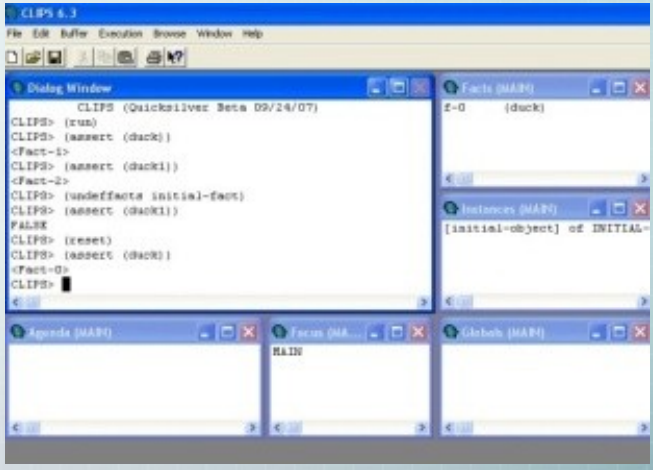
Open source

CLIPS

- Razlikuje velika i mala slova
- Činjenice (*facts*) - Baza činjenica predstavlja početno stanje problema
- Pravila (*rules*) - Baza pravila sadrži operatore koji pretvaraju stanje problema u rješenje

Mehanizam zaključivanja(*inference engine*) u CLIPS-u radi u tri koraka:
1.Uspoređuje činjenice sa pravilima
2.Izabire koje pravilo izvršiti
3.Izvršava odgovarajuću akciju zadanu uz pravilo

"**Watch**"	*CLIPS> (watch rules)*
"**Reset**"	*CLIPS> (reset)*
"**Run**"	*CLIPS> (run)*



CLIPS-FACTS!

ČINJENICE

- Pregledavanje baze činjenica — *CLIPS> (facts)*
- Dodavanje podatka u bazu činjenica: “**assert** “ — *CLIPS> (assert (Homer voli pivo))*
- Brisanje činjenice: “**retract**” — *CLIPS> (retract 0)*
- Brisanje svih činjenica: “**clear**” — *CLIPS> (clear)*
- Definiranje više činjenica odjednom: “**deffacts**” —

```
CLIPS> (deffacts Simpsoni
(Homer voli pivo)
(Marge ima plavu kosu) )
```

- Ili...učitavanje iz datoteke! Potrebno napraviti “**reset**” (tek tada dodajemo ih u bazu)
- Korištenje predložaka(*templates*): “**deftemplate**”

```
(deftemplates Simpson
(slot ime (type STRING) )
(slot godine (type NUMBER)
 (default 36)) )
```

→

```
CLIPS> (deffacts Simpsoni
(Simpson (ime Homer) )
(Simpson (ime Marge) (godine (34) ) )
```

CLIPS-RULES

PRAVILA

Sintaksa:

(defrule <imePravila>
<komentar(opcija)>
<deklaracija(opcija)>
<premisa1>
...
<premisaN>
=>
<akcija1>
...
<akcijaM>
)

Npr:

(defrule navike
"Homerove navike"
(salience 10)
(Homer drži pivo u ruci)
(Moe razgovara sa Homerom)
=>
(assert (Homer se nalazi u baru))
(assert (Homer je sretan))
)

Sa varijablama !

(defrule navike
(?osoba drži pivo u ruci)
(Moe razgovara sa ?osoba))
=>
(assert (?osoba se nalazi u baru))
(assert (?osoba je sretan))
)

- Važnost pravila: "**salience**", raspon: [-10 000,10 000], veći broj, veća važnost, default 0

Vidi Clips User's Guide za više informacija!

Sažetak

- Ekspertni sustavi (sustavi temeljeni na znanju) oponašaju **rasuđivanje stručnjaka u nekoj uskoj domeni**, ne pokušavajući ostvariti rješavanje općenitih problema
- Znanje prikazuju **ako–onda pravilima** ("granule znanja")
- Jasno odvajaju između **baze znanja** (pohranjuje pravila i činjenice) i **stroja za zaključivanje** (provodi podudaranje uzoraka, razrješavanje konflikta i paljenje pravila)
- **Ljuska ekspertnog sustava** sačinjena je od stroja za zaključivanje i korisničkog sučelja te se može kombinirati s različitim bazama znanja
- Ovisno o primjeni, zaključivanje može biti izvedeno kao **ulančavanje unaprijed** ili **ulančavanje unazad**
- **CLIPS** je široko korištena ljuska s unaprijednim zaključivanjem

Sljedeća tema: Modeliranje neizvjesnosti