Oblikovanje programske podrške

2012./2013. grupa P01

Temelji formalne verifikacije

Logika

Prof.dr.sc. Vlado Sruk

Sveučilište u Zagrebu
Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroel., računalne i inteligentne sustave

Literatura

- Clarke E., Grumberg O., Peled D.: *Model Checking*, MIT Press 1999.

Pripremio: Nikola Bogunović

Prilagodio: Vlado Sruk

U pripremi materijala osim literature upotrijebljeni su i drugi izvori, te zahvaljujem autorima.

Sadržaj

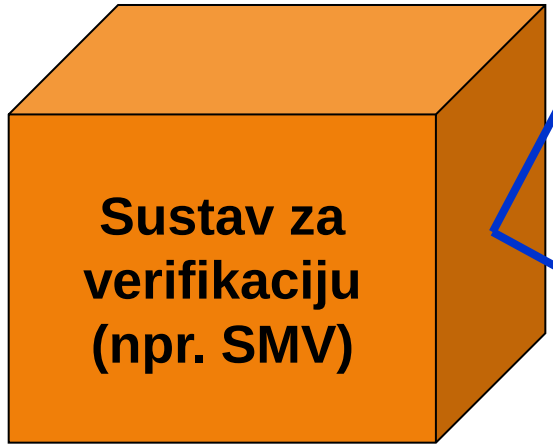
- Formalna matematička logika
  - propozicijska
  - predikatna
- Preslikavanje formula predikatne logike u normalizirane klauzule
- Postupci formalne verifikacije računalnih sustava

- Cilj:
  - Osnove logike i formalnog promatranja apstraktno i simbolički
  - Motiviranje rasuđivanja sposobnost i razvoj logičnog razumijevanja problema
  - Promocija uporabe logika i njene primjenjivosti za rješavanje probleme u računarstvu
  - Osigurati osnovne elemente logičkih argumenta, analize i formulacije procesa, bitne u razumijevanju i primjeni računarstva

Formalna verifikacija

- Formalna verifikacija programske potpore metodom provjere modela (engl. Model checking)
- Postupak provjere da *formalni model* izvedenog sustava *(I),* odgovara *formalnoj specifikaciji (S)* s matematičkom izvjesnošću

I = Implementacija (model sustava koji se verificira). Izraženo povezanim strojevima s konačnim brojem stanja (FSM).



S =Specifikacija (željeno ponašanje). Izraženo u vremenskoj logici.

$I \not\models S$

DA = model sustava logički zadovoljava specifikaciju

NE - ispis pogrešnog izvođenja programa

Formalna verifikacija

- formalna verifikacija nastoji dokazati logičku zadovoljivost (tj. da model zadovoljava specifikaciju) te za njezinu primjenu su potrebna osnovna znanja iz područja:
  - Formalna (matematička) logika
    - posebice definicija “logičke zadovoljivosti” i sl.
  - Modeliranje implementacije strojevima s konačnim brojem stanja.
  - Izražavanje specifikacije (tj. željenog ponašanja) vremenskom logikom kao proširenjem klasične matematičke logike.

Primjer

```
void merge (int a[ ], a_len, b[ ], b_len, *c)
{
int i = 0, j = 0, k = 0;
while (k < a_len+b_len) {
  if (a[i] < b[j]) {
     c[k] = a[i];
     i++; }
  else {
     c[k] = b[j];
     j++; };
  k++;
}
```

Specifikacija?

Program/implementacija?

FORMALNA (MATEMATIČKA) LOGIKA

Formalna (matematička) logika

- Različiti tipovi logike se razlikuju po sadržaju svojih “primitiva”.
- Dva su temeljna pogleda na logiku:
  - *Ontološki:* Što postoji u svijetu.
  - *Epistemološki:* Kakvo je stanje znanja (što agent vjeruje).

Language	Ontological Commitment (What exists in the world)	Epistemological Commitment (What an agent believes about facts)
Propositional logic	facts	true/false/unknown
First-order logic	facts, objects, relations	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Probability theory	facts	degree of belief 0…1
Fuzzy logic	degree of truth	degree of belief 0…1

Klasična logika zasniva se na pojmu istinitosti.

naš interes

Formalna (matematička) logika

- *Logike su* **formalni jezici** *koji predstavljaju informaciju na način da se mogu automatizirano izvoditi zaključci.*
- *Sintaksa* definira strukturu rečenice u jeziku.
- *Semantika* definira značenje rečenica (definira istinitost rečenice u svijetu u kojem ju promatramo).
- **Postoji mnogo logika:**
  - *Propozicijska i predikatna logika*
  - **Logike višega reda**
  - **Modalne logike**
    - **Epistemička logika**
    - *Vremenska logika*
    - **…**
  - **Opisna logika**
  - **Nemonotona logika**
  - **…**

u ovom kolegiju

Formalna (matematička) logika

- Logika određuje postupke ispravnog rasuđivanja.
- Primjer 1:

  Pretpostavka (premisa) 1: 1. Svaki čovjek je smrtan.
  Pretpostavka (premisa) 2: 2. Sokrat je čovjek.
  Zaključak: 3. Sokrat je smrtan.
  (Ako su istinite rečenice 1 i 2, “logički slijedi” rečenica 3.)

- Primjer 2:

  1. Svaki α ima obilježje β.
  2. γ je α.
  3. γ ima obilježje β.

- **Zaključak 3** "Logički slijedi" samo na temelju oblika (forme), a ne na temelju sadržaja (konteksta).

- ***Matematička ili formalna logika*** daje sustav zaključivanja u kojem je "logički izveden" zaključak barem tako dobar kao polazne pretpostavke.

  - Temelj: formalan sustav (definicija formalnog sustava slijedi kasnije).
  - Niti jedan formalan sustav ne može osigurati istinite polazne pretpostavke.

Propozicijska logika

- Logika sudova, iskaza, tvrdnji
- *engl. propositional logic, propositional calculus*

- Sintaksa: Logika iskaza preslikava deklarativne rečenice (koje mogu biti istinite ili lažne) u sustav simbola.
  - Npr.: “Sokrat je mudar.” preslikava se u simbol P.

- Sustav propozicijske logike sastoji se od:
  - PS: P, Q, ... PS je prebrojiv skup atoma, simboličkih varijabli, simbola
  - Logički operatori (vezice):
    - $\neg$ (ne, not, ~) negacija
    - $\wedge$ (i, and, &) konjunkcija
    - $\vee$ (ili, or, |) disjunkcija
    - $\Rightarrow$ (ako, if, $\supset$, $\rightarrow$) implikacija
    - $\Leftrightarrow$ (akko, iff, $\equiv$, $\leftrightarrow$) ekvivalencija
    - Rezervirani simboli:
    - F (false, $\varnothing$, 0, $\perp$) konstanta (neistinitost)
    - T (true, 1) konstanta (istinitost)
    - (), . znakovi zagrada, zareza i točke
  - Def. (rekurzivno) ispravno formiran složeni iskaz, ili formula (*engl. well-formed formula - wff*) :
    - 1. Svaki atom je formula.
    - 2. Ako su P i Q formule, onda su formule: ($\neg$P), ($\neg$Q), (P$\wedge$Q), (P$\vee$Q), (P$\Rightarrow$Q), (P$\Leftrightarrow$Q).

Semantika

- Pridruživanje obilježja istinitosti (T, F) atomičkim simbolima = *Interpretacija*
  - *I: PS → BOOL*

  gdje je BOOL={ T, F }, tj. funkcija s kodomenom T ili F (istinito ili lažno).

- Semantika *dvaju složenih atomičkih simbola* prikazuje se istinitosnom tablicom.
  - 2 suda = 2^2 = 4 interpretacije, 2^4 = 16 istinitosnih tablica
- Neke važnije tablice istinitosti za povezivanje dva simbola:

	P	Q	implikacija (P ⇒ Q)	ekvivalencija (P ⇔ Q)	kontradikcija (), ⊥	tautologija T
I_1 :	T	T	T	T	F	T
I_2 :	T	F	F	F	F	T
I_3 :	F	T	T	F	F	T
I_4 :	F	F	T	T	F	T

Svojstva implikacije (P ⇒ Q)

- To je *materijalna implikacija* i nije potpuno intuitivna prirodnom jeziku. Namjera materijalne implikacije je modelirati *uvjetnu konstrukciju, (a ne uzročno-posljedičnu vezu*), tj..:
- "ako P tada Q", tj.. ako je P istinit, tada je (P ⇒ Q) istinito samo ako je Q istinito.
- Primjeri koji pokazuju *neintuitivni aspekt* materijalne implikacije:

  (2 + 2 = 4) ⇒ ("Zagreb je glavni grad Hrvatske")
- je istinita formula jer su prethodna (P) i posljedična (zaključna) (Q) tvrdnja istinite.

  (2 + 2 = 4) ⇒ ("London je glavni grad Hrvatske")
- je neistinita formula jer je posljedična tvrdnja (Q) neistinita.

Svojstva implikacije (P ⇒ Q)

- Što je s formulama gdje je prethodna tvrdnja neistinita, a zaključna istinita ili neistinita:

- *(2 + 2 = 5) ⇒ (“Zagreb je glavni grad Hrvatske”)*
- *(2 + 2 = 5) ⇒ (“London je glavni grad Hrvatske”)*

- U prirodnom jeziku mogli bi ovakvim formulama implikacije pridijeliti bilo istinitost ili neistinitost, a možda čak i tvrditi da ako je prethodna tvrdnja (P) neistinita, implikacija ne mora biti ni istinite ni neistinite.

- U formalnoj logici prihvaćena je **konvencija**:
- *Ako je P neistinit, tada je implikacija (P ⇒ Q) istinita, neovisno o istinitosti Q.*

Svojstva implikacije (P ⇒ Q)

- Zašto ima smisla implikaciju proglasiti istinitom ako je P neistinit ?
- Koje su moguće opcije:
- Za P = istinito suglasni smo s istinitosti implikacije (istinita ili neistinita ovisno o Q).
- Za P = neistinito postoje 4 moguće tablice:

P	Q	1:	2:	3:	4:	
F	*F*	*F*	*F*	*T*	*T*	
F	*T*	*F*	*T*	*F*	*T*	*što izabrati ?*
T	F	F	F	F	F	
T	T	T	T	T	T	nije sporno!

- Ako 1., to je konjunkcija, ako 2., to je Q, ako 3., to je ekvivalencija.
- Dakle preostaje jedino 4. tablica.

Semantička pravila

- Izračunavanje istinitosti složene formule (evaluacija):
- Primjer 1: P_1, P_2 istinite, Q_1, Q_2 neistinite, a bilo koja formula (istinita ili ne).
- *Istinite* su formule:

  $\neg Q_1$
  $(P_1 \wedge P_2)$
  $(P_1 \vee A)$
  $(A \vee P_1)$
  $(A \Rightarrow P_1)$
  $(Q_1 \Rightarrow A)$
  $(P_1 \Leftrightarrow P_2)$
  $(Q_1 \Leftrightarrow Q_2)$

  Neistinite su formule:

  $\neg P_1$
  $(Q_1 \wedge A)$
  $(A \wedge Q_1)$
  $(Q_1 \vee Q_2)$
  $(P_1 \Rightarrow Q_1)$
  $(P_1 \Leftrightarrow Q_1)$
  $(Q_1 \Leftrightarrow P_1)$
  () - prazna formula

- Primjer 2: Izračunavanja istinitosti složene formule $(Q \vee (((\neg Q) \wedge P) \Rightarrow R))$
- ima 3 propozicijska simbola P, Q, R:
- *Interpretacija* (jedan od mogućih svjetova, ovdje 2^3 mogućih interpretacija)
- Neka je jedna interpretacija I: P=T, Q=F, R=F, izračunavanje (evaluacija) istinitosne vrijednosti daje formuli:
  - $(Q \vee (((\neg Q) \wedge P) \Rightarrow R))$ *neistinitu* vrijednost.

- **Semantika uključuje interpretaciju i evaluaciju.**

Pravila ekvivalencije

- Definicija: Dvije formule su semantički ekvivalentne ili jednake
  - ako imaju jednaku (istu) istinitosnu vrijednost za svaku interpretaciju I.
- Ekvivalencija u slijedećim pravilima može se provjeriti tablicom istinitosti za sve interpretacije. Provjera koincidencije istinitosnih tablica **nije u općem slučaju dovoljna**, ali definicija je ispravna.

$(A \wedge \neg A) = ()$	kontradikcija
$(\neg(\neg A)) = A$	dvostruka negacija
$(A \wedge A) = A$	jednaka važnost (idempotencija)
$(A \vee A) = A$	jednaka važnost
$(A \vee B) = (B \vee A)$	komutativnost
$(A \wedge B) = (B \wedge A)$	komutativnost
$((A \vee B) \vee C) = (A \vee (B \vee C))$	asocijativnost
$((A \wedge B) \wedge C) = (A \wedge (B \wedge C))$	asocijativnost
$(A \wedge (B \vee C)) = ((A \wedge B) \vee (A \wedge C))$	distributivnost
$(A \vee (B \wedge C)) = ((A \vee B) \wedge (A \vee C))$	distributivnost
$(\neg(A \vee B)) = ((\neg A) \wedge (\neg B))$	De Morganov zakon
$(\neg(A \wedge B)) = ((\neg A) \vee (\neg B))$	De Morganov zakon
$(A \Rightarrow B) = ((\neg A) \vee B)$	eliminacija uvjeta
$(A \Leftrightarrow B) = ((A \Rightarrow B) \wedge (B \Rightarrow A))$	eliminacija dvostrukog uvjeta
$(A \Rightarrow B) = ((\neg B) \Rightarrow (\neg A))$	transpozicija

Formalan sustav

- Definiramo formalan sustav kao dvojku: {Γ, L} gdje je
  - Γ - konačan skup ispravno definiranih (formiranih) formula (wff)
  - L – konačan skup pravila zaključivanja
- Neka temeljna pravila zaključivanja (jedan mogući skup L)
- Generiraju dodatne istinite formule (mehanički) bez razumijevanja konteksta (značenja).
  - Pogodna za strojnu primjenu.
  - Semantički korespondiraju sa semantikom "istinitosti".

Ako P=*T*, Q=*T*, generiraj $(P \wedge Q) = T$	(uvođenje konjunkcije)
Ako P=*T*, (P⇒Q)=*T*, generiraj $Q = T$	("modus ponens")
Ako ¬Q=*T*, (P⇒Q)=*T*, generiraj $\neg P$	("modus tolens")
Ako (P∧Q)=*T*, generiraj $(Q \wedge P) = T$	(komutativnost ∧)
Ako (P∧Q)=*T*, generiraj *P*=*T*, *Q*=*T*	(∧ eliminacija)
Ako P=*T* (odnosno Q=*T*), generiraj $(P \vee Q) = T$	(uvođenje disjunkcije)
Ako [¬(¬P)]=*T*, generiraj *P*=*T*	(eliminacija negacije)

Definicije obilježja u formalnom sustavu

- Sekvencija formula $\{\omega_1, \omega_2, \ldots, \omega_n\}$ ili pojedina formula ω_i je
- *teorem (dokaz, dedukcija)* iz skupa formula Γ, ako je u skupu Γ,
- ili se može izvesti iz Γ korištenjem pravila zaključivanja L .
- $\Gamma \vdash_L \{\omega_1, \omega_2, \ldots, \omega_n\}$ sekvencija formula je teorem
- $\Gamma \vdash_L \omega_i$ formula ω_i je teorem
-

Zarez označava konjunkciju

- Npr. (skup Γ sadrži dvije *istinite* formule): $\Gamma = \{ P, (P \Rightarrow Q) \}$
- Korištenjem pravila “Modus ponens” (iz skupa dopustivih pravila L), izvodimo da je istinita nova formula *Q,* te je ta formula Q *teorem (dokaz, dedukcija) skupa* Γ.

- Skup Γ je *konzistentan* akko (ako i samo ako)
- ne sadrži formule na temelju kojih bi ω_i i $\neg\omega_i$ (istovremeno) bili teoremi.

- $\Gamma = \{ P, (P \Rightarrow Q) \}$ je *konzistenatan*.

- $\Gamma = \{ P, \neg P, (P \Rightarrow Q) \}$ je *nekonzistentan ili kontradiktoran* jer su P i $\neg$P istovremeno teoremi (nalaze se u samom skupu Γ).

- $\Gamma = \{ P, \neg Q, (P \Rightarrow Q) \}$ je *nekonzistentan* jer sadrži $\neg$Q, a pravilom “Modus ponens” može se izvesti Q, dakle $\neg$Q i Q bi istovremeno bili teoremi.

Obilježja u formalnom sustavu

- Neka se u formalnom sustavu {Γ, L} izvodi neki teorem ωi (tražimo odgovor da li je ωi teorem ili ne).

- Sustav je *odrediv* ili odlučljiv (*engl. decidable*), akko postoji algoritam koji će u konačnom vremenu odrediti ili ne teorem ωi (dati u konačnom vremenu dati odgovor da li teorem ωi postoji ili ne).

- Formalan sustav {Γ, L} je *poluodrediv* ili poluodlučljiv (*engl. semidecidable*), akko postoji algoritam koji će u konačnom vremenu odrediti teorem ako on postoji. Algoritam završava u konačnom vremenu s odgovorom "da" (za teorem ωi), ali ne mora završiti u konačnom vremenu s odgovorom "ne"(tj. ωi nije teorem).

- Formalan sustav je *neodrediv* ili neodlučljiv (*engl. undecidable*) ako nije odrediv ni poluodrediv.

Interpretacija i evaluacija

- *Semantika* u formalnom sustavu povezane su
  - *interpretacijom* (*pridruživanjem* istinitosti atomima) i
  - *evaluacijom* (*izračunavanjem* istinitosti složene formule).
- Neka *interpretacija je model* formalnog sustava ako evaluira *sve* njegove formule *u istinito* (vrijedi i za svaku formulu pojedinačno).
- Npr.: interpretacija I: {P=T, Q=F, R=F} formule $(Q \vee (((\neg Q) \wedge P) \Rightarrow R))$ *nije model* jer ta interpretacija formuli daje neistinitu vrijednost.
- Skup formula je *zadovoljiv* (*engl. satisfiable*) ako ima model (*barem jedan*). Vrijedi i za pojedinačne formule. (SAT problem (zadovoljivost) - temeljni NP problem!!)
- Sukladno ranijoj definiciji, *nezadovoljiv* (nekonzistentan, kontradiktoran) skup formula *nema nijedan model*.
- Skup formula Γ *implicira* ili *povlači* (*engl. entails*) formulu ω, ako je *svaki model* od Γ ujedno i model od ω.
  - Formula ω je tada *logička posljedica* skupa formula Γ.
  - $\Gamma \models \omega$ (svaki model od Γ je model formule ω)
- Formula je *valjana* ili *tautologija* (*engl. valid*)
  - ako je istinita za svaku interpretaciju i evaluaciju.
  - $\models \omega$ (svaka interpretacija je model formule ω)

Primjeri logičkih posljedica

- Svaka interpretacija koja lijevoj strani od znaka $\models$ daje istinitost mora i desnoj strani dati istinitost.
- $(P \wedge Q) \models P$

  lijeva strana = T samo za (P=T, Q=T), a to daje i desnoj strani =T, dakle gornji izraz vrijedi (P je logička posljedica ($P \wedge Q$).
- $(P \vee Q) \models P$

  lijeva strana je istinita za (P=F,Q=T; P=T,Q=F; P=T,Q=T), ali desna za interpretaciju (P=F,Q=T) nije istinita, te P nije logička posljedica ($P \vee Q$).
- $\{\neg Q, (P \vee Q)\} \models P$ (zarez predstavlja konjunkciju $\wedge$)

  skup Γ na lijevoj strani je istinit samo za Q=F, P=T, a to daje istinitost i desnoj strani, te je P logička posljedica navedenog skupa Γ.
- $P \models (Q \vee \neg Q)$

  također vrijedi, jer za svaku interpretaciju za koju je lijeva strana istinita (P=T) i desna stana je istinita (desna strana je uvijek istinita).

Primjeri logičkih posljedica

- Γ = $(A \vee C) \wedge (B \vee \neg C)$ = Knowledge Base = KB
  - dvije konjunkcijom povezane formule (umjesto $\wedge$ može se koristiti zarez).

Neka je: $\alpha = (A \vee B)$

$KB \models \alpha$?

A	B	C	$A \vee C$	$B \vee \neg C$	KB	α
$False$	$False$	$False$	$False$	$True$	$False$	$False$
$False$	$False$	$True$	$True$	$False$	$False$	$False$
$False$	$True$	$False$	$False$	$True$	$False$	$True$
$False$	$True$	$True$	$True$	$True$	$True$	$True$
$True$	$False$	$False$	$True$	$True$	$True$	$True$
$True$	$False$	$True$	$True$	$False$	$False$	$True$
$True$	$True$	$False$	$True$	$True$	$True$	$True$
$True$	$True$	$True$	$True$	$True$	$True$	$True$

Ispravnost i kompletnost form. sust.

- Formalan sustav {Γ, L} je *ispravan* (*engl. sound*)
  - ako $\Gamma \models \omega_i$ kadgod je $\Gamma \vdash_L \omega_i$,
  - tj. svaka pravilima dokazana formula je ujedno i logička posljedica skupa Γ.

$$\Gamma \vdash_L \omega_i \text{ implicira } \Gamma \models \omega_i$$

- Formalan sustav {Γ, L} je *kompletan* (*engl. complete*)
  - ako $\Gamma \vdash_L \omega_i$ kadgod je $\Gamma \models \omega$,
  - tj. svaku logičku posljedicu skupa Γ moguće je dokazati pravilima L.

$$\Gamma \models \omega \text{ implicira } \Gamma \vdash_L \omega_i$$

- U *ispravnom i kompletnom* formalnom sustavu {Γ, L} vrijedi:

$$\Gamma \models \omega = \Gamma \vdash_L \omega_i$$

- Većina interesantnih formalnih sustava je nekompletno, a vrlo malo ih je odredivo.
- *Propozicijska logika je ispravna, kompletna i odrediva* (npr. preslikavanjem u tablicu istinitosti), jer operira s *konačnim* skupom simbola.

Proriteti operatora

- Primjeri:
- Prioritet logičkih operatora:
- Najviši:

$\neg$	negacija
$\wedge$	konjunkcija
$\vee$	disjunkcija
$\Rightarrow$	implikacija

- Najniži: $\Leftrightarrow$ ekvivalencija

Formule:	Obilježja:
1. P	zadovoljiva ali ne i valjana (interpretacija P=T je model, dok interpretacija P=F nije model).
2. $(P \vee \neg P)$	valjana (tautologija), sve interpretacije (dvije) P=T, P=F, su modeli (formula je istinita).
3. $(P \wedge \neg P)$	kontradiktorna (nezadovoljiva), nema modela.
4. **()**	*kontradiktorna (nezadovoljiva).*
5. $P \Rightarrow (Q \Rightarrow P)$	valjana (tautologija), sve interpretacije (ima ih 4: FF, FT, TF, TT) su modeli.
6. $(P \wedge Q)$	zadovoljiva. Ima samo jedan model: P=T, Q=T.

Normalni oblici logičkih formula

- Svaka propozicijska formula može se preslikati (ekvivalentna je) formuli u *disjunkcijskom normalnom obliku (DNF)* :
  - $(k1_1 \wedge \ldots \wedge k1_n) \vee (k2_1 \wedge \ldots \wedge k2_m) \vee \ldots \vee (kp_1 \wedge \ldots \wedge kp_r)$
- Svaka propozicijska formula može se preslikati (ekvivalentna je) formuli u *konjunkcijskom normalnom obliku (CNF)* :
  - $(k1_1 \vee \ldots \vee k1_n) \wedge (k2_1 \vee \ldots \vee k2_m) \wedge \ldots \wedge (kp_1 \vee \ldots \vee kp_r)$
  - *CNF* = konjunkcija klauzula
- gdje su:
  - k_i = *literal* (negirani ili nenegirani atomički simbol - atom)
  - *klauzula* = disjunkcija literala. Npr.: $(k2_1 \vee \ldots \vee k2_m)$

Konverzija propozicijske formule u CNF oblik

- Svaka formula u propozicijskoj logici može se konvertirati u konjunkciju klauzula (CNF):
- Npr: $\neg(P \Rightarrow Q) \vee (R \Rightarrow P)$

  1. Eliminiraj implikaciju uporabom ekvivalentnog "$\vee$" oblika:

     $\neg(\neg P \vee Q) \vee (\neg R \vee P)$

  2. Reduciraj doseg negacije (pomak u desno) uporabom DeMorganovih pravila, te eliminiraj dvostruke negacije:

     $(P \wedge \neg Q) \vee (\neg R \vee P)$

  3. Pretvori u CNF asocijativnim i distribucijskim pravilima:

     $(P \vee \neg R \vee P) \wedge (\neg Q \vee \neg R \vee P)$,

- te dalje:

  $(P \vee \neg R) \wedge (\neg Q \vee \neg R \vee P)$ = **CNF oblik**

Postupci pojednostavljivanja klauzula

- 1. Uporaba temeljnih pravila:
  - Npr.:

    $(P \vee R \vee P)$ pojednostavi (spoji) u $(P \vee R)$

    $(P \vee \neg P \vee Q)$ izostavi cijelu jer je evidentno valjana (T)

- 2. Podrazumijevanje (*engl subsumption*) *klauzula*
  - Klauzula ω_1 podrazumijeva klauzulu ω_2 , ako su literali u ω_1 podskup literala u ω_2.
  - Npr.:
  - $(P \vee R)$ podrazumijeva klauzule $(P \vee R \vee Q)$

Usporedba CNF i DNF oblika

- Mogu dati brze odgovore na česta pitanja:
- DNF nam govori da li je **formula zadovoljiva**.
  - Ako su sve disjunkcije neistinite ($\perp$), ili sve sadrže komplementarne literale (npr. ($A \wedge \neg A$)) ne postoji niti jedan model za taj DNF. Inače je formula zadovoljiva.
- CNF nam govori da li je ili nije formula **tautologija** (valjana, uvijek istinita).
  - Ako sve klauzule sadrže istinitost (T) ili sve sadrže komplementarne literale (npr ($A \vee \neg A$)) formula je tautologija. Inače, za formulu postoji barem jedna interpretacija (pridruživanje istinitosti atomičkim simbolima) koja nije model (ne zadovoljava formulu) pa formula nije tautologija.
- Preslikavanje CNF u DNF i obrnuto je računalno vrlo skupo (vremenski i prostorno

Semantička ekvivalencija

- Ranija definicija: Dvije formule su semantički *ekvivalentne* ili *jednake*
  - ako imaju *jednaku (istu) istinitosnu vrijednost za svaku interpretaciju* ***I***.
- Npr. Da li su ekvivalentne dvije formule: $((P \wedge Q) \Rightarrow P)$ i $(R \vee \neg R)$?
  - *DA!* sukladno gornjoj definiciji (obje su valjane -- tautologije), ali usporedba istinitosnih tablica nema smisla (simboli i tablice su različite).

- *Definicija ekvivalencije preko pojma logičke posljedice ($\not\models$)*
- Ako dvije ekvivalentne formule imaju jednaku istinitosnu vrijednost za svaku interpretaciju, može se definirati:
- *Dvije formule α i β su semantički ekvivalentne (oznake $(\alpha \Leftrightarrow \beta)$ ili $(\alpha \equiv \beta)$) akko vrijedi: $(\alpha \models \beta)$ i $(\beta \models \alpha)$.*
- Ranija tablica pravila ekvivalencije daje: $(\alpha \Leftrightarrow \beta) = (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.
  - Ako su α i β ekvivalentne, formula $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$ mora biti **uvijek** istinita:

    $I \models (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$
  - *Semantička* ekvivalencija je na taj način identična *dokazivoj* ekvivalenciji.
- ako želiš dokazati ekvivalentnost, dokaži da je $((\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha))$ tautologija, tj. *da je njena negacija nezadovoljiva*
  - Primjer ekvivalentne formule: $((P \wedge Q) \Rightarrow R) \Leftrightarrow (P \Rightarrow (Q \Rightarrow R))$

Teorem dedukcije

- *Dokazivanje logičke posljedice preko (ne)zadovoljivosti*
- Formula ψ je *logička posljedica* formule φ, tj. $\varphi \models \psi$, akko je formula $(\varphi \Rightarrow \psi)$ tautologija (valjana).
- Dokaz:
- Akko je $(\varphi \Rightarrow \psi)$ tautologija (uvijek istinita), onda iz tablice za implikaciju proizlazi da kada je φ istini i ψ mora biti istinit. To je upravo definicija logičke posljedice.

φ	ψ	$(\varphi \Rightarrow \psi)$
F	F	T
F	T	T
T	F	F
T	*T*	*T*

- Budući da $(\varphi \Rightarrow \psi)$ mora bit tautologija, to njena negacija
- $\neg(\varphi \Rightarrow \psi) = \neg(\neg\varphi \vee \psi) = (\varphi \wedge \neg\psi)$ mora biti nezadovoljiva. Dakle:

> - *$\varphi \models \psi$ akko je $(\varphi \wedge \neg\psi)$ nezadovoljiva*

Primjena teorema dedukcije

- dokazivanje obaranjem - *1*
- φ možemo zamisliti kao konjunkciju istinitih formula (bazu formula, bazu znanja). To su početne pretpostavke – aksiomi nekog problema.

Npr. neka φ predstavlja skup istinitih formula (povezanih konjunkcijom):

$A1 \wedge$
$A2 \wedge$
$\ldots \wedge$
Ap

Tražimo dokaz:

Da li je neka formula ψ logička posljedica skupa formule danim sa φ ?

Teorem dedukcije kaže: Ako želimo dokazati da je neka formula ψ logička posljedica formule φ, moramo dokazati *nezadovoljivost* formule $(\varphi \wedge \neg\psi)$.

- Dakle formulu ψ negiramo i dodamo formulama φ , te uporabom dopustivih pravila pokušavamo pokazati kontradiktornost (nezadovoljivost) u $(\varphi \wedge \neg\psi)$.

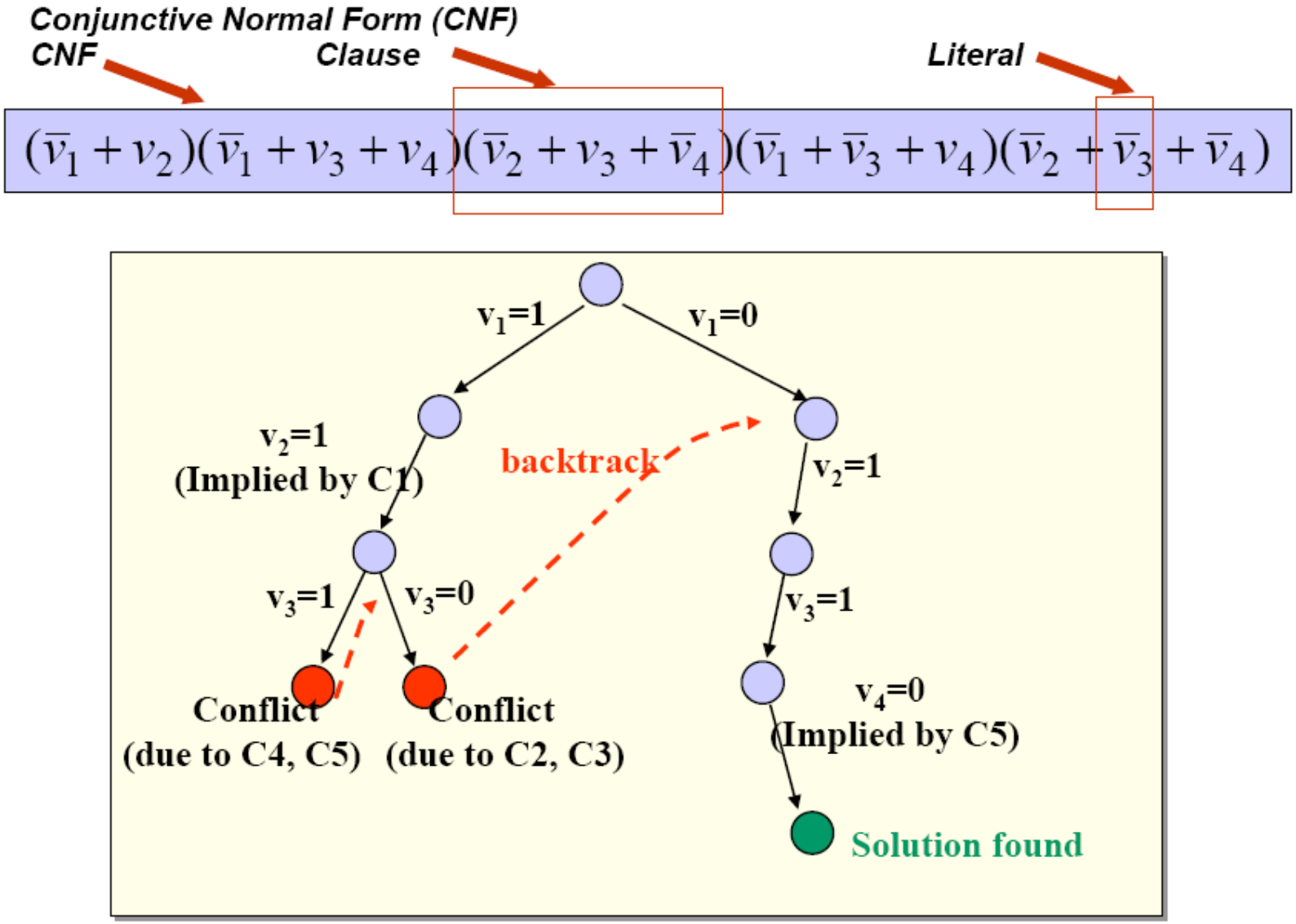
- *Nezadovoljivost možemo dokazati ako primjenom pravila uspijemo generirati praznu formulu "( )" (jer je ona sigurno nezadovoljiva).*

SAT problem

- Problem zadovoljivosti - temeljni NP problem
- Tražimo model skupa formula Γ (interpretaciju koja evaluira sve formule u skupu Γ u istinito. To je ekvivalentno traženju modela *jedne* složene formule koja se sastoji iz *konjunkcije svih formula u* Γ.
- Γ skup formula je najčešće dan u *CNF* obliku:

  $(k1_1 \vee \ldots \vee k1_p) \wedge (k2_1 \vee \ldots \vee k2_r) \wedge \ \ldots \wedge (kp_1 \vee \ldots \vee kp_s)$
- Iscrpna procedura rješavanja *CNF* SAT problema sistematski pridjeljuje istinitosne vrijednosti atomičkim propozicijskim simbolima.
  - Za *n atoma* 2^n *pridruživanja*.
  - Eksponencijalna složenost, računalno neizvedivo u općem slučaju.
- Za *DNF – polinomska složenost* jer postoji konačan broj literala, a dovoljno je pronaći zadovoljivost u samo jednom disjunkcijskom članu.
- CNF 2SAT - polinomska kompleksnost (do 2 literala u klauzuli)
- *CNF 3SAT* - *NP kompletno* (3 literala u klauzuli)
- *Zadovoljivost formule u CNF obliku s 3 i više literala je NP kompletno.*
- Mnogi stohastički algoritmi troše eksponencijalno vrijeme u najgorem slučaju, ali polinomsko u srednjem (očekivanom).

Primjer: Donošenja odluke o zadovoljivosti



Primjena teorema dedukcije

- ***dokazivanje SAT rješavačem***
- Neka *istinite* formule predstavljaju skup Γ:
  1. P
  2. $(P \Rightarrow Q)$
  3. $(Q \Rightarrow S)$
- U CNF obliku: $\Gamma = [\,(P) \;\wedge\; (\neg P \vee Q) \;\wedge\; (\neg Q \vee S)\,]$

- Da li je neka formula **S** logička posljedica skupa Γ : $\Gamma \models S$?

- Teorem dedukcije:
- S je logička posljedica Γ ako je $(\Gamma \wedge \neg S)$ nezadovoljiva.

- Skupu Γ **dodajemo negaciju formule** koju želimo dokazati ($\neg S$):

  $[\,(P) \wedge (\neg P \vee Q) \wedge (\neg Q \vee S) \wedge (\neg \mathbf{S})\,]$

- Sat sustavom pokušamo naći bar jedan model (zadovoljivost).
  Ako SAT sustav pokaže da formulu nije moguće zadovoljiti (nema modela), zaključujemo:

  S je doista logička posljedica skupa Γ.

PREDIKATNA LOGIKA

PREDIKATNA LOGIKA

- Logika predikata prvoga reda – *FOPL* (*engl. predicate logic, predicate calculus, first order predicate logic*)

  1. P: Svi ljudi su smrtni.
  2. Q: Sokrat je čovjek.
  3. R: Sokrat je smrtan.

- U propozicijskoj logici nikako se iz 1 i 2 ne može zaključiti 3.

- FOPL uvodi objekte, relacije, obilježja, funkcije (pobliži opis izjave).

Sintaksa predikatne logike

- *Atomički predikat:*
  - pred_simbol: osnovno obilježje u rečenici (predikat)
  - t_i = članovi: objekti ili odnosi u rečenici
  - dva načina zapisa:
    - (pred_simb t1 t2 … tn) – infiks notacija (LISP)
    - pred_simb(t1 t2 … tn) – prefiks notacija (Prolog)
- ***Članovi (t_i)*** :
  - Konstante: objekti u nekom svijetu (blok1, sokrat, …).
  - Rezervirane konstante: T, F.
  - Varijable: razred objekata ili obilježja; mogu poprimiti vrijednosti iz svoje domene;
    - (Npr.: X, Y, …).
- Funkcije:
  - veza između objekata - (fun_simb t1 t2 … tn)
  - Npr.: (cos X), (otac_od abel kain)

- Formalna def. člana:

  1. Konstanta je član.
  2. Varijabla je član.
  3. Ako je fun_simb funkcijski simbol sa n-argumenata, a t_1, t_2, …, t_n su članovi, tada je (fun_simb t_1 t_2 … t_n) član.

- ***Logički operatori*** (vezice): $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$

- ***Kvantifikacijski simboli*** (uz varijable, pobliže određuju istinitost rečenice):
  - $\exists$ (postoji, za_neki, exist) - egzistencijski ili partikularni kvantifikator (barem jedan).
  - $\forall$ (za_svaki, svi, for_all) - univerzalni kvantifikator (svi), ima središnju ulogu u izražavanju generalizacije.

Ispravno definiran složeni predikat ili formula

- 1. svaki atomički predikat je formula.
- 2. ako je S_i formula, tada su formule:
  $(\neg S), (S_1 \wedge S_2), (S_1 \vee S_2), (S_1 \Rightarrow S_2), (S_1 \Leftrightarrow S_2)$.
- 3. ako je X varijabla, a S formula, tada su formule: $\exists X\ S(X)$, $\forall X\ S(X)$.
  (oznaka S(X) = formula S u kojoj postoji varijabla X)
- Negirani ili nenegirani atomički predikat naziva se *literal*.
- *Dopuna pravilima ekvivalencije (*P(X), Q(X) su wff s varijablom X):

$(\neg(\forall X\ P(X))) = \exists X\ (\neg P(X))$ - analogno De Morgan
$(\neg(\exists X\ Q(X))) = \forall X\ (\neg Q(X))$
$\exists X\ P(X) = \exists Y\ P(Y)$ - simbol varijable nije bitan, ali je bitan doseg, uvijek unutar jedne formule
$\forall X\ P(X) = \forall Y\ P(Y)$

- Primjer ispravno definirane složene formule u infiks notaciji :

$(\forall X\ \forall Y\ (((\text{otac } X\ Y) \vee (\text{majka } X\ Y)) \Rightarrow (\text{roditelj } X\ Y)))$

Semantika predikatne logike

- Skup ispravno definiranih složenih predikata ili formula (wff) odnosi se na neku domenu razmatranja D.
- Interpretacija I je proces preslikavanja elemenata iz domene D svakoj
- pojedinoj konstanti, varijabli, i funkciji, te atomičkom predikatu, tako da:
  - Simbolu T uvijek je pridružena istinita vrijednost.
  - Simbolu F uvijek je pridružena neistinita vrijednost.
  - Svakoj konstanti pridruži se jedan element iz D.
  - Svakom funkcijskom simbolu pridruži se jedan element iz D.
  - Svakoj varijabli se pridruži neprazan podskup iz D (dozvoljene supstitucije).
- Svaka funkcija f, sa m argumenata, definira interpretacijom i *evaluacijom* preslikavanje iz D^m u D, tj.: f: $D^m \rightarrow D$ (pridruživanje jednog elementa iz D).
- Svaki predikat P, s brojem članova n, definira *interpretacijom i evaluacijom* svojih članova preslikavanje iz D^n u $\{T, F\}$, tj. P: $D^n \rightarrow \{T, F\}$ (istinito ili ne).
- Vrijednosti wff formula složenih logičkim operatorima date su odgovarajućim istinitosnim tablicama.
- Vrijednost $\forall X$ P(X) je T, ako P(X) je T, za sve vrijednosti X date sa I, a F inače.
- Vrijednost $\exists X$ P(X) je T, ako P(X) je T, barem za jednu vrijednost X danoj sa I, a F inače.

Semantika predikatne logike

- Određivanje istinitosti wff svodi se na *interpretaciju + evaluaciju*

- Primjeri pridruživanja istinitosti:
- 1. (prijatelj ivan ana)
  - predikat je T, ako u D postoji objekt Ana koja je prijatelj Ivanu.

- 2. X je domena prirodnih brojeva
  - ∀X (veci X 10) atomički predikat je F
  - ∃X (veci X 10) atomički predikakt je T

- ∀ - u određivanju T potrebne sve supstitucije varijable
  - (problem ako je domena beskonačna)
- ∃ - u određivanju T potrebna jedan supstitucija za koju T
  - (problem ako je domena beskonačna i predikat F)

- Skup svih istinitih predikata iz domene D = *stanje svijeta*
  - *engl. state of the world*

Primjeri preslikavanja

- Preslikavanja prirodnog jezika u formule predikatne logike

  1. “Nitko nije savršen.” (infiks notacija)

¬∃X (savrsen X), ili ∀X (¬(savrsen X)), tj. "svi su nesavršeni"

  2. “Svi košarkaši su visoki.” (infiks notacija)

∀X ((kosarkas X) ⇒ (visok X))

  - “za svaki X u domeni razmatranja vrijedi da ako je (X) košarkaš tada je visok”

- *Ispravna uporaba univerzalnog kvantifikatora ∀ (1)*

- Neka je okvir razmatranja (skup objekata): { Garfield, Feliks, računalo}

- Preslikaj u pred. logiku: "Sve mačke su sisavci."

- Za sve objekte u okviru razmatranja vrijedi: ako su mačke tada su sisavci.

  ∀ x [mačka(x) ⇒ sisavac(x)] (prefiks notacija)

Ispravna uporaba univerzalnog kvantifikatora ∀

- ∀ x [mačka(x) ⇒ sisavac(x)] (vrijedi za sve objekte x)

- Dokaz: Supstitucija svih objekata u formulu (konjunkcija formula jer ∀):

[mačka(Garfield) ⇒ sisavac(Garfield)] ∧ [mačka(Feliks) ⇒ sisavac(Feliks)] ∧ [mačka(računalo) ⇒ sisavac(računalo)] ∧ ... (ostali objekti ako ih ima)

prva [ ]: T (vidi tablicu za ⇒)

druga [ ]: T (vidi tablicu za ⇒)

treća [F ⇒ T] = T (vidi tablicu za ⇒)pa je i treća formula =T !!!!!

- Ako bi preslikali: *∀ x [mačka(x) ∧ sisavac(x)]*

- Supstitucija svih objekata daje:

[mačka(Garfield) ∧ sisavac(Garfield)] ∧ [mačka(Feliks) ∧ sisavac(Feliks)] ∧ [mačka(računalo) ∧ sisavac(računalo)] ∧ … (ostali objekti ako ih ima)

- *mačka(računalo) = F* - daje neistinitu cijelu formulu !!

Ispravna uporaba egzistencijskog kvantifikatora ∃

- Neka je okvir razmatranja (kao i prije): { Garfield, Feliks, računalo}
- Preslikaj u predikatnu logiku: "Garfield ima brata koji je mačka."
- Postoji barem jedan (neki) objekt i takav da su mu obilježja istinita.

  $\exists x$ [brat(x, Garfield) $\wedge$ mačka(x)]

- Dokaz supstitucijom svih objekata u formulu (disjunkcija formula jer $\exists$):

  [brat(Garfield, Garfield) $\wedge$mačka(Garfield)] $\vee$
  [brat(Feliks, Garfield) $\wedge$ mačka(Feliks)] $\vee$
  [brat(računalo, Garfield) $\wedge$ mačka(računalo)] $\vee$ … (ostali ako ih ima)

- Prva [ ] neistinita, ali idemo dalje jer su […] povezane disjunkcijom.
- Drugi red istinit, cijela formula je istinita (dalje ne moramo ispitivati).

Ispravna uporaba egzistencijskog kvantifikatora

- Ako bi preslikali: $\exists x\ [brat(x, Garfield) \Rightarrow mačka(x)]$

- Supstitucija svih objekata u disjunkciju formula daje:
  [brat(Garfield, Garfield) $\Rightarrow$ mačka(Garfield)] $\vee$
  [brat(Feliks, Garfield) $\Rightarrow$ mačka(Feliks)] $\vee$
  [brat(računalo, Garfield) $\Rightarrow$ mačka(računalo)] $\vee$ … (ostali objekti ako ih ima)

- Implikacija je istinta ako je atomički izraz na lijevioj strani neistinit !
- Npr. ako je: *[brat(računalo, Garfield) $\Rightarrow$ mačka(računalo)]* istinito,
  - cijela je formula istinita !!
- Egzistencijski kvantificirana implikacijska formula je istinita ako
  - u okviru razmatranja postoji barem jedan objekt
  - za koji je premisa implikacije neistinita (desna strana može biti T ili F).
- Takva rečenica ne daje nikakvu potvrdnu informaciju.
- Zaključak:
  - $\forall$ ide uz $\Rightarrow$
  - $\exists$ ide uz $\wedge$

Primjer: predikatne logike

- Preslikaj u predikatnu logiku: "Niti jedan student ne sluša sve predmete."
- Rješenje: Definiramo predikate (formalna logika ne definira predikate):

  S(x) - x je student (prefiks notacija)
  L(x) - x je predmet
  B(x, y) - x sluša y

- Prioriteti logičkih operatora: $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$.
- Preporuka: koristi zagrade za prioritet i doseg kvantif.

a) "Ne postoji x koji je student i takav da sluša sve predmete.

$$\neg\exists x \, [\, S(x) \wedge \forall y \, (L(y) \Rightarrow B(x, y)\,)\,]$$

= obilježje

b) "Za svaki x vrijedi: ako je student postoji predmet (bar jedan) koji ne sluša"

$$\forall x \, [\, S(x) \Rightarrow (\, \exists y \, (L(y) \wedge \neg B(x, y)\,)\,)\,]$$

= obilježje

- Evidentno: 1) $\exists$ ide uz $\wedge$, $\forall$ ide uz $\Rightarrow$
  2) pomicanjem negacije u a) slijedi b) - DeMorgan

Obilježja predikatne logike

- Zadovoljivost
- Model
- Logička posljedica
- Kontradiktornost
- Pravila zaključivanja

} kao u propozicijskoj logici

- Logika predikata višega reda
- Predikatna logika:
  - Kvantifikacija samo varijabli (objekata u domeni D), a ne na odnose (predikatni ili funkcijski simbol) u domeni D.
- Logika višega reda:

  $\forall$(Voli) (Voli ivo ana)
  - kvantifikacija na predikatnom (ili funkcijskom) simbolu.

Obilježja propozicijske i predikatne logike

- Većina interesantnih formalnih logičkih sustava je nekompletna, a vrlo malo ih je odredivo.

- Propozicijska logika je:
  - Ispravna, kompletna i odrediva (npr. preslikavanjem u tablicu istinitosti), jer operira s konačnim skupom simbola.

- Predikatna logika je:
  - Poluodrediva (ako teorem postoji, dokazat će se, a ako ne postoji može se ali i ne mora dokazati).
  - "Čista" (npr. bez aritmetike) predikatna logika je ispravna i kompletna (Gödel).

1. Deduktivni pristup

- Opis sustava (implementacija) dana skupom formula Γ. Treba dokazati da je formula φ (specifikacija sustava) logička posljedica skupa Γ.

$\Gamma \models \varphi$ dokaži da su *svi modeli* Γ ujedno i modeli φ

ili

$\Gamma \vdash_L \varphi$ *dokaži* (izvedi) φ uporabom skupa pravila L
(simboličko izvođenje programa)

- Obilježja:
  - Problem predstavljanja.
  - Zahtijeva stručno vođenje (strategije, jednakost, …).
  - Primjena ograničena na Ulazno/Izlazne sustave (terminirajuće).
  - Može se koristiti za sustave s beskonačnim brojem stanja.

2. Provjera modela (engl. "model checking")

- $\Gamma \models \varphi$ provjeravamo isključivo zadovoljivost (da li je φ-specifikacija istinita u jednom modelu Γ-implementacije, tj. za jednu interpretaciju)

- Obilježja:
  - Ograničeno na modele s konačnim brojem stanja = FSM.
  - Primjena u reaktivnim sustavima (neterminirajućim).
  - Automatizirano izvođenje.

- Pazi :
- Višestruka semantika ("overloading") znaka $\models$: /
  - *logička posljedica* (formalna logika) i
  - *zadovoljivost* (provjera modela)
  - Treba uvijek navesti kontekst u kojem se koristi ova oznaka.

PROVJERA MODELA

- 1981. Clarke & Emerson, Model Checking
- Fokus: strojevi s konačnim brojem stanja i prijelazi
  - nije “općenito” dokazivanje teorema
- Zasniva se na uporabi:
  - Linearne vremenska logike
    - engl. Linear Temporal Logic
    - Boolean + always, until, eventually, ...
  - vremenska logika s grananjem
    - engl. Computation Tree Logic
    - + “for all futures”; “for some futures”

Tipične primjene formalne logike

1. Matematika (dokazivanje teorema)
2. Formalna logika (dopuna teorije)
3. Zagonetke (imitacija racionalnog rasuđivanja)
4. Oblikovanje računalnih sustava
5. Automatizirano upravljanje temeljem istinitih formula
6. …

Diskusija

Preslikavanje formula predikatne

- *Normalizirana klauzula* je univerzalno kvantificirana disjunkcija literala.
- *Temeljna ili osnovna klauzula* (*engl. ground clause*): klauzula bez varijabli.
- Preslikavanje *ne zadržava jednakost* (ekvivalenciju), ali *zadržava zadovoljivost.*

- 1. Eliminirati sve implikacije i ekvivalencije, te umjesto njih koristiti disjunkcije i konjunkcije, npr.:
- $(P \Rightarrow Q) = ((\neg P) \vee Q)$
- $(P \Leftrightarrow Q) = (((\neg P) \vee Q) \wedge (P \vee (\neg Q)))$

- 2. Pomaknuti negacije do jediničnih formula, npr.:
- $(\neg(\neg P)) = P$
- $(\neg(\forall X\ A(X))) = (\exists X\ (\neg A(X)))$
- $(\neg(\exists X\ A(X))) = (\forall X\ (\neg A(X)))$

- 3. Preimenovati varijable uz kvantifikatore u istoj složenoj formuli. Npr.:
- $(\forall X\ (P(X)) \Rightarrow (\exists X\ Q(X)))$
- Formule P i Q trebaju sadržati različite varijable (npr. X, Y),
- $(\forall X\ (P(X)) \Rightarrow (\exists Y\ Q(Y)))$
- Doseg varijable je samo unutar formula.
- Ponekad je korisno formule standardizirati (posebno označiti varijable).

- 4. Eliminirati kvantifikatore (skolemizirati - Toraf Skolem).
- 4.1 Univerzalni kvantifikator (podrazumijeva se, te se ispušta)

  $(\forall X ((\text{prizma } X) \Rightarrow (\text{geom\_tijelo } X)))$;
  - sve prizme se geom. tijela pišemo jednostavnije: $((\text{prizma } X) \Rightarrow (\text{geom\_tijelo } X))$
- 4.2 Egzistencijski kvantifikator
  - Pretvorba varijable u novu konstantu ili funkciju, čiji su članovi univerzalno kvantificirane varijable.
- 4.2.1 Egzistencijski kvantifikator nije u dosegu univerzalnog:

  $(\exists X((\text{nudist } X) \wedge (\text{demokrat } X)))$
  - X superponiramo s novom jedinstvenom konstantom – skolem konstanta
  - Dajemo ime nečemu što mora postojati, jer ako je formula istinita, mora postojati bar jedna supstitucija.

  Odaberemo npr: X = ab_1, te slijedi:

  $((\text{nudist ab\_1}) \wedge (\text{demokrat ab\_1}))$

- 4.2.2 Egzistencijski kvantifikator u dosegu univerzalnog:

(∀X (∃Y P(X,Y))) ; predikat P sadrži varijable X i Y

- Postoji neki Y i nekako ovisi o (odabranom) X. Dajemo novi simbol za Y, (uvodimo funkcijski član - skolem funkciju), jer je Y različit za svaki odabrani X.

- Y = f(X), pa uz izostavljanje ∀ slijedi:

P(X, f(X)) ; prefiks notacija

- Npr. "Svaka osoba ima majku."

∀X ∃Y (majka X Y) ; infiks notacija

Varijabla X označuje svaku osobu.
Y označuje određenu (ne svaku), jer ovisi o X.
Y zamjenjujemo s funk. članom (m X), pa (uz ispuštanje ∀):

(majka X (m_funkcija X)) ; u infiks notaciji

Pojednostavljenje klauzula podrazumijevanjem

- *Podrazumijevanje* odbacuje manje općenite klauzule.

- Upotrebljavamo prefiks notaciju:
- Npr. stariji(otac_od(X), X) je općenitija jedinična klauzula od
  - stariji(otac_od(ana), ana)

- Npr. žena(X, Y) $\vee$ žensko(X) je općenitija od
  - žena(ana, ivan) $\vee$ žensko(ana)

- Npr. P(a, X) $\vee$ P(Y, b) je općenitija od
  - P(a, b)