Logică și Structuri Discrete -LSD

Cursul 10 – Logică propozițională

dr. ing. Cătălin Iapă

catalin.iapa@cs.upt.ro

De data trecută:

Logică - noțiuni generale

Logică Propozițională

Sintaxa

Semantica

Diagrame de decizie binară

Forma normală conjunctivă

Sintaxa logicii propoziționale

Un *limbaj* e definit prin
simbolurile sale
și *regulile* după care combinăm corect simbolurile (*sintaxa*)

Simbolurile logicii propoziționale:
propoziții: notate deobicei cu litere p, q, r , etc.
operatori (conectori logici): negație $\neg$, implicație $\rightarrow$, paranteze ( )

Formulele logicii propoziționale: definite prin *inducție structurală*
(construim formule complexe din altele mai simple)

O formulă e:
orice *propoziție* (numită și formulă atomică)
$(\neg\alpha)$ dacă α este o formulă
$(\alpha \rightarrow \beta)$ dacă α și β sunt formule
(α, β numite *subformule*)

Alți operatori (conectori) logici

Deobicei, dăm definiții *minimale* (cât mai puține cazuri)
(orice raționament ulterior trebuie făcut pe toate cazurile)

Operatorii cunoscuți pot fi definiți folosind $\neg$ și $\rightarrow$:

$\alpha \wedge \beta \stackrel{def}{=} \neg(\alpha \rightarrow \neg\beta)$ (ȘI)

$\alpha \vee \beta \stackrel{def}{=} \neg\alpha \rightarrow \beta$ (SAU)

$\alpha \leftrightarrow \beta \stackrel{def}{=} (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$ (echivalență)

Omitem parantezele redundante, definind precedența operatorilor.

Ordinea precedenței: $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$

Implicația e asociativă *la dreapta*! $p \rightarrow q \rightarrow r = p \rightarrow (q \rightarrow r)$

Semantica unei formule: funcții de adevăr

Definim riguros cum calculăm valoarea de adevăr a unei formule = dăm o *semantică* (înțeles) formulei (formula=noțiune *sintactică*)

O *funcție de adevăr* v atribuie oricărei formule o *valoare de adevăr* $\in$ {T, F} astfel încât:

$v(p)$ e definită pentru fiecare *propoziție* atomică p.

$$v(\neg\alpha) = \begin{cases} \mathrm{T} & \text{dacă } v(\alpha) = \mathrm{F} \\ \mathrm{F} & \text{dacă } v(\alpha) = \mathrm{T} \end{cases}$$

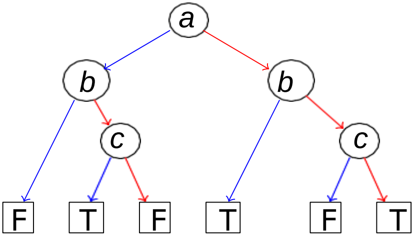
$$v(\alpha \rightarrow \beta) = \begin{cases} \mathrm{F} & \text{dacă } v(\alpha) = \mathrm{T} \text{ și } v(\beta) = \mathrm{F} \\ \mathrm{T} & \text{în caz contrar} \end{cases}$$

Arbore de decizie binar

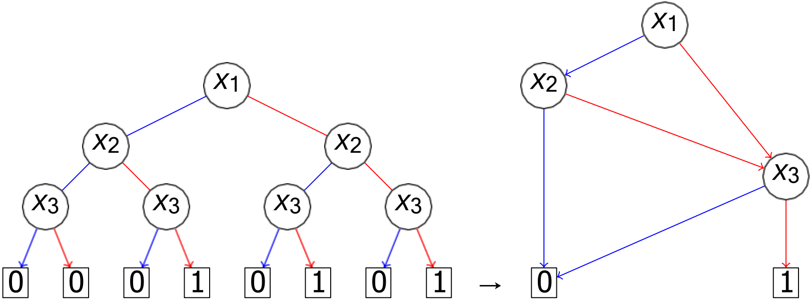
$$f = (a \vee b) \wedge (a \vee \neg c) \wedge (\neg a \vee \neg b \vee c)$$

$$f|_{a=T} = T \wedge T \wedge (\neg b \vee c) = \neg b \vee c$$

$$f|_{a=F} = b \wedge \neg c \wedge T = b \wedge \neg c$$



De la arbore la diagramă de decizie binară



arbore de decizie binar *diagramă de decizie binară*

Forma normală conjunctivă (conjunctive normal form)

folosită pentru a determina dacă o formulă e *realizabilă* (poate fi T)

Def: *Forma normală conjunctivă*
= *conjuncție* $\wedge$ de *clauze*

clauză = *disjuncție* $\vee$ de *literali*

literal = propoziție sau negația ei (p sau $\neg p$)

$(a \vee \neg b \vee \neg d)$ clauză
$\wedge (\neg a \vee \neg b)$ clauză
$\wedge (\neg a \vee c \vee \neg d)$ ...
$\wedge (\neg a \vee b \vee c)$ clauză

Exemplu: forma normală conjunctivă

Lucrăm *din exterior*

1) ducem *negațiile înăuntru* până la propoziții r. *de Morgan*

dubla negație dispare $\neg\neg A = A$

înlocuim implicațiile dinspre exterior când ajungem la ele

$p \to q = \neg p \vee q \qquad \neg(p \to q) = p \wedge \neg q$

2) ducem disjuncția $\vee$ înăuntrul conjuncției $\wedge$ *distributivitate*

Exemplu:

$$
\begin{aligned}
& \neg((a \wedge b) \vee ((a \to (b \wedge c)) \to c)) \\
= \; & \neg(a \wedge b) \wedge \neg((a \to (b \wedge c)) \to c)) \\
= \; & (\neg a \vee \neg b) \wedge ((a \to (b \wedge c)) \wedge \neg c) \\
= \; & (\neg a \vee \neg b) \wedge (\neg a \vee (b \wedge c)) \wedge \neg c \\
= \; & (\neg a \vee \neg b) \wedge (\neg a \vee b) \wedge (\neg a \vee c) \wedge \neg c
\end{aligned}
$$

În cursul de azi

Cum determinăm dacă o formulă e *realizabilă*?
algoritm folosit în rezolvarea multor probleme

Ce înseamnă o *demonstrație* logică?

Realizabilitatea unei formule în logică propozițională (SAT-problem/ satisfiability)

Demonstrație vs consecință logică

Realizabilitatea unei formule propoziționale (satisfiability)

Se dă o formulă în *logică propozițională*.

Există vreo atribuire de valori de adevăr care o face adevărată ?
= e *realizabilă* (engl. *satisfiable*) formula ?

$$
\begin{aligned}
& (a \vee \neg b \vee \neg d) \\
\wedge\ & (\neg a \vee \neg b) \\
\wedge\ & (\neg a \vee c \vee \neg d) \\
\wedge\ & (\neg a \vee b \vee c)
\end{aligned}
$$

Găsiți o atribuire care satisface formula?

Formula e în *formă normală conjunctivă* (conjunctive normal form)
= conjuncție de disjuncții de *literali* (pozitiv sau negat)

Fiecare conjunct (linie de mai sus) se numește *clauză*

Reguli în determinarea realizabilității

Simplificăm problema, știind că vrem **formula adevărată**
(NU se aplică la simplificarea formulelor în formule echivalente!)

R1) Un literal *singur într-o clauză* are o singură valoare utilă:

în $a \wedge (\neg a \vee b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$ — a trebuie să fie T

în $(a \vee b) \wedge \neg b \wedge (\neg a \vee \neg b \vee c)$ — b trebuie să fie F

(altfel formula are valoarea F)

Reguli pentru determinarea realizabilității (cont.)

R2a) Dacă un literal e T, *pot fi șterse clauzele* în care apare (ele sunt adevărate, le-am rezolvat)

R2b) Dacă un literal e F, *el poate fi șters* din clauzele în care apare (nu poate face clauza adevărată)

Exemplele anterioare se simplifică:

$$a \wedge (\neg a \vee b \vee c) \wedge (\neg a \vee \neg b \vee \neg c) \xrightarrow{a=\mathrm{T}} (b \vee c) \wedge (\neg b \vee \neg c)$$

$$(a \vee b) \wedge \neg b \wedge (\neg a \vee \neg b \vee c) \xrightarrow{b=\mathrm{F}} a$$

(și de aici $a = T$, deci formula e realizabilă)

Reguli pentru determinarea realizabilității (cont.)

R3) Dacă *nu mai sunt clauze*, formula e realizabilă (cu atribuirea construită)

Dacă obținem o *clauză vidă*, formula *nu e realizabilă* (fiind vidă, nu putem s-o facem T)

$(a \vee b) \wedge a \wedge (a \vee \neg b \vee c) \xrightarrow{a=\mathrm{T}} (\mathrm{T} \vee b) \wedge \mathrm{T} \wedge (\mathrm{T} \vee \neg b \vee c) \xrightarrow{R2a}$

ștergem toate clauzele (conțin T, le-am rezolvat)

$\Rightarrow$ formulă realizabilă (cu $a = \mathrm{T}$)

$a \wedge (\neg a \vee b) \wedge (\neg b \vee c) \wedge (\neg a \vee \neg b \vee \neg c)$

$\xrightarrow{a=\mathrm{T}} b \wedge (\neg b \vee c) \wedge (\neg b \vee \neg c)$

$\xrightarrow{b=\mathrm{T}} c \wedge \neg c \quad \xrightarrow{c=\mathrm{T}} \quad \emptyset$ ($\neg c$ devine clauza vidă $\Rightarrow$ nerealizabilă)

Reguli pentru determinarea realizabilității (cont.)

Dacă *nu mai putem face reduceri* după aceste reguli ?

$a \wedge (\neg a \vee b \vee c) \wedge (\neg b \vee \neg c) \quad \xrightarrow{a=T} \quad (b \vee c) \wedge (\neg b \vee \neg c)$?

R4) Alegem o variabilă și *despărțim pe cazuri* (încercăm):

- cu valoarea F
- cu valoarea T

O soluție pentru *oricare* caz e bună (nu căutăm o soluție anume).

Dacă *niciun caz* nu are soluție, formula *nu e realizabilă*.

Realizabilitatea unei formule în logică propozițională (SAT-problem/ satisfiability)

Demonstrație vs consecință logică

Sintaxă și semantică

Pentru logica propozițională, am discutat:

Sintaxa: o formulă are *forma*:
propoziție sau ($\neg$ *formulă*) sau (*formulă* $\rightarrow$ *formulă*)

Semantica: calculăm *valoarea de adevăr* (înțelesul), pornind de la cea a propozițiilor

$$v(\neg\alpha) = \begin{cases} T & \text{dacă } v(\alpha) = F \\ F & \text{dacă } v(\alpha) = T \end{cases}$$

$$v(\alpha \rightarrow \beta) = \begin{cases} F & \text{dacă } v(\alpha) = T \text{ și } v(\beta) = F \\ T & \text{în caz contrar} \end{cases}$$

Deducții logice

Deducția ne permite să demonstrăm o formulă în mod *sintactic* (folosind doar structura ei)

E bazată pe o *regulă de inferență* (de deducție)

$$\frac{A \qquad A \rightarrow B}{B}$$

modus ponens

(din A și $A \rightarrow B$ deducem/inferăm B; A, B formule oarecare)

și un set de *axiome* (formule care pot fi folosite ca premise/ipoteze)

A1: $\alpha \rightarrow (\beta \rightarrow \alpha)$

A2: $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$

A3: $(\neg\beta \rightarrow \neg\alpha) \rightarrow (\alpha \rightarrow \beta)$

în care α, β etc. pot fi înlocuite cu *orice* formule

A1 - A3 sunt tautologii

Deducție (demonstrație)

Informal, o deducție (demonstrație) e o *înșiruire de afirmații* în care fiecare *rezultă* (poate fi derivată) din cele *anterioare*.

Riguros, definim:

Fie H o mulțime de formule (ipoteze).

O *deducție* (demonstr.) din H e un șir de formule $A_1, A_2, \ldots, A_n$, astfel ca $\forall i \in [1, n]$

1. A_i este o *axiomă*, sau

2. A_i este o *ipoteză* (o formulă din H), sau

3. A_i rezultă prin *modus ponens* din A_j, A_k anterioare ($j, k < i$)

Spunem că A_n *rezultă* din H (e *deductibil*, e o *consecință*).
Notăm: $H \vdash A_n$

Exemplu de deducție

Demonstrăm că $A \to A$ pentru orice formulă A

(1) $A \to ((A \to A) \to A))$ A1 cu $\alpha = A$, $\beta = A \to A$

(2) $A \to ((A \to A) \to A)) \to ((A \to (A \to A)) \to (A \to A))$ A2 cu $\alpha = \gamma = A$, $\beta = A \to A$

(3) $(A \to (A \to A)) \to (A \to A)$ MP(1,2)

(4) $A \to (A \to A)$ A1 cu $\alpha = \beta = A$

(5) $A \to A$ MP(3,4)

Verificarea unei demonstrații e un proces simplu, mecanic (verificăm motivul indicat pentru fiecare afirmație; o simplă comparație de șiruri de simboluri).

Găsirea unei demonstrații e un proces mai dificil.

Alte reguli de deducție

Modus ponens e suficient pentru a formaliza logica propozițională dar sunt și alte reguli de deducție care simplifică demonstrațiile

$$\frac{p \to q \qquad \neg q}{\neg p}$$ *modus tollens (reducere la absurd)*

$$\frac{p}{p \vee q}$$ *generalizare (introducerea disjuncției)*

$$\frac{p \wedge q}{p}$$ *specializare (simplificare)*

$$\frac{p \vee q \qquad \neg p}{q}$$ *eliminare (silogism disjunctiv)*

$$\frac{p \to q \qquad q \to r}{p \to r}$$ *tranzitivitate (silogism ipotetic)*

Deducția (exemplu)

Fie $H = \{a,\ \neg b \vee d,\ a \rightarrow (b \wedge c),\ (c \wedge d) \rightarrow (\neg a \vee e)\}$.
Arătați că $H \vdash e$.

(1) a	ipoteză, H_1
(2) $a \rightarrow (b \wedge c)$	ipoteză, H_3
(3) $b \wedge c$	modus ponens (1, 2)
(4) b	specializare (3)
(5) d	eliminare (4, H_2)
(6) c	specializare (3)
(7) $c \wedge d$	(5) și (6)
(8) $\neg a \vee e$	modus ponens (7, H_4)
(9) e	eliminare (1, 8)

Consecința logică (semantică)

Interpretare = atribuire de adevăr pentru propozițiile unei formule. O formulă poate fi adevărată sau falsă într-o interpretare.

Def.: O mulțime de formule $H = \{H_1, \ldots, H_n\}$ *implică* o formulă C dacă *orice interpretare* care satisface (formulele din) H satisface C

Notăm: $H \models C$

(C e o *consecință logică* / consecință semantică a ipotezelor H)

Consecința logică (semantică)

Ca să stabilim consecința semantică trebuie să *interpretăm* formule (cu valori/funcții de adevăr)

$\Rightarrow$ lucrăm cu *semantica* (înțelesul) formulelor

Exemplu: arătăm $\{A \vee B, C \vee \neg B\} \models A \vee C$

Cazul 1: $v(B)$ = T. Atunci $v(A \vee B)$ = T și $v(C \vee \neg B) = v(C)$. Dacă $v(C) = T$, atunci $v(A \vee C) = T$, deci afirmația e adevărată.

Cazul 2: $v(B)$ = F. La fel, reducem la $\{A\} \models A \vee C$ (adevărat).

Consistență și completitudine

$H \vdash C$: *deducție* (pur sintactică, din axiome și reguli de inferență)
$H \vDash C$: *implicație, consecință semantică* (valori de adevăr)

Consistență:

Dacă H e o mulțime de formule, și C este o formulă astfel ca $H \vdash C$, atunci $H \vDash C$

(Orice teoremă e *validă*;

orice afirmație obținută prin deducție e *întotdeauna adevărată*).

Consistență și completitudine

$H \vdash C$: *deducție* (pur sintactică, din axiome și reguli de inferență)
$H \vDash C$: *implicație, consecință semantică* (valori de adevăr)

Completitudine:

Dacă H e o mulțime de formule, și C e o formulă astfel ca $H \vDash C$, atunci $H \vdash C$.

(Orice tautologie e o teoremă,
orice consecință semantică poate fi *dedusă* din *aceleași ipoteze*).

Consistență și completitudine

$H \vdash C$: *deducție* (pur sintactică, din axiome și reguli de inferență)
$H \models C$: *implicație, consecință semantică* (valori de adevăr)

Logica propozițională e *consistentă și completă*:

Ca să demonstrăm o formulă, putem arăta că e *validă*.

Pentru aceasta, verificăm că *negația ei nu e realizabilă*.

Vă mulțumesc!

Bibliografie

Conținutul cursului se bazează pe materialele de anii trecuți de la cursul de LSD, predat de conf. dr. ing.Marius Minea și ș.l. dr. ing. Casandra Holotescu (http://staff.cs.upt.ro/~marius/curs/lsd/index.html)