# Logica propozițiilor

## Curs 3

! Propozițiile logice sunt modele ale afirmațiilor propoziționale care sunt fie adevărate, fie false.

### Sintaxa logicii propozițiilor.

- alfabetul:

$\Sigma_p$ = Var_propoziționale ∪ Conective ∪ {, }

Var_propoziționale = $\{p, q, r, \dots\}$

Conective = $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$

$\neg$ negația
$\wedge$ conjuncția
$\vee$ disjuncția
$\rightarrow$ implicația
$\leftrightarrow$ echivalența

- regulile de formare a formulelor propoziționale

$F_p$ = mulțimea formulelor prop. corect construite
= cea mai mică mulțime de formule ce se poate construi cu regulile:
  - baza: $p_i \in F_p$, $i = 1, 2, \dots$
  - inducția: dacă $U, V \in F_p$ atunci:

$\boxed{\neg U \in F_p}$

$\boxed{U \wedge V \in F_p}$

$\boxed{U \vee V \in F_p}$

$\boxed{U \rightarrow V \in F_p}$

$\boxed{U \leftrightarrow V \in F_p}$

  - închiderea: toate formulele din $F_p$ se obțin doar prin aplicarea regulilor precedente de un nr. finit de ori

### Semantica logicii propoziționale.

Scopul definirii semanticii logicii propoziționale este de a atribui un înțeles, o valoare de adevăr, formulelor propoziționale.

Domeniul semantic: $\{F, T\}$

$\neg F = T$

$\neg T = F$

## Semantica conectivelor

| $p$ | $\neg p$ |
|---|---|
| T | F |
| F | T |

$\uparrow$ – nand $\quad p \uparrow q := \neg(p \wedge q)$

$\downarrow$ – nor $\quad p \downarrow q := \neg(p \vee q)$

$\oplus$ – xor $\quad p \oplus q := \neg(p \leftrightarrow q)$

| $p$ | $q$ | $p \wedge q$ | $p \vee q$ | $p \to q$ | $p \leftrightarrow q$ | $p \uparrow q$ | $p \downarrow q$ | $p \oplus q$ |
|---|---|---|---|---|---|---|---|---|
| T | T | T | T | T | T | F | F | F |
| T | F | F | T | F | F | T | F | T |
| F | T | F | T | T | F | T | F | T |
| F | F | F | F | T | T | T | T | F |

## Interpretarea (Def). Concepte semantice (Def)

O interpretare a formulei $U(p_1, p_2, \dots, p_n) \in F_p$ este o funcție $i: \{p_1, p_2, \dots, p_n\}$ care asociază valori de adevăr variabilelor propoziționale și poate fi extinsă la o funcție $i: F_p \to \{T, F\}$ folosind relațiile:

$i(\neg p) = \neg i(p)$

$i(p \vee q) = i(p) \vee i(q)$

$i(p \wedge q) = i(p) \wedge i(p)$

$i(p \to q) = i(p) \to i(q)$

$i(p \leftrightarrow q) = i(p) \leftrightarrow i(q)$

→ interpretările – evaluează formulele conform conectivelor

tabela de adevăr = evaluează formula în toate cele $2^n$ interpretări

MODEL: $i(U) = T$

ANTIMODEL: $i(U) = F$

, unde $i: \{p_1, p_2, p_3, \dots, p_n\} \to \{T, F\}$

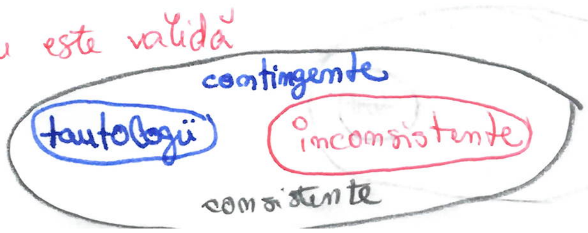
U se numește:

→ consistentă (realizabilă) $\Leftrightarrow$ $\exists$ cel puțin un model (deci poate fi evaluată ca adevărată)

→ validă (tautologie): $\vDash U$ $\Leftrightarrow$ U evaluată „T" în $\forall$ interpretare
$\Updownarrow$
toate interpretările sunt modele

→ inconsistentă (nerealizabilă) $\Leftrightarrow$ U nu are niciun model
$\Updownarrow$
toate interpretările sunt antimodele

→ contingentă $\Leftrightarrow$ consistentă, dar nu este validă



## Metoda tabelelor de adevăr

| | $p$ | $q$ | $r$ | $\neg p \vee q$ | $r \vee p$ | $(\neg p \vee q) \wedge (r \vee p)$ | $(\neg p \wedge r) \vee (q \wedge r) \vee (q \wedge p)$ | $p \uparrow \neg p$ | $p \downarrow \neg p$ |
|---|---|---|---|---|---|---|---|---|---|
| $i_1$ | T | T | T | T | T | T | T | T | F |
| $i_2$ | T | T | F | T | T | T | T | T | F |
| $i_3$ | T | F | T | F | T | F | F | T | F |
| $i_4$ | T | F | F | F | T | F | F | T | F |
| $i_5$ | F | T | T | T | T | T | T | T | F |
| $i_6$ | F | T | F | T | F | F | F | T | F |
| $i_7$ | F | F | T | T | T | T | T | T | F |
| $i_8$ | F | F | F | T | F | F | F | T | F |

$U(p,q,r) = (\neg p \vee q) \wedge (r \vee p)$
$V(p,q,r) = (\neg p \wedge r) \vee (q \wedge r) \vee (q \wedge p)$ } contingente & consistente

$p \uparrow \neg p$ ↑ tautologie; $p \downarrow \neg p$ ↑ inconsistentă

modele pentru $U$: $i_1, i_2, i_5, i_7$

$i_1 : \{p,q,r\} \to \{T,F\}$ $i_1(p)=T$, $i_1(q)=T$, $i_1(r)=T$ și $i_1(U)=T$

antimodele pt $U$: $i_3, i_4, i_6, i_8$

## Metasimboluri – relații semantice între formule

- Formula $V$ este consecință logică a formulei $U$, notație: $U \models V \iff$
$\iff \forall i : F_p \to \{T,F\}$ astfel încât $i(U)=T$, au loc $i(V)=T$

- Formulele $U(p_1, p_2, \ldots, p_n) \in F_p$ și $V(p_1, p_2, \ldots, p_n) \in F_p$ sunt logic echivalente, notație $U \equiv V \iff$ tabelele lor de adevăr sunt identice, adică $\forall i : F_p \to \{T,F\}$, $i(U) = i(V)$

ex: $U \equiv V$
$U \models \neg p \vee q$

## Concepte semantice pentru mulțimi de formule

### Mulțime consistentă (realizabilă)

O mulțime $\{U_1, U_2, \ldots, U_n\}$ de formule se numește consistentă (realizabilă) $\iff$ formula $U_1 \wedge U_2 \wedge U_3 \wedge \ldots \wedge U_n$ este consistentă, adică $\exists i : F_p \to \{T,F\}$ astfel încât $i(U_1 \wedge U_2 \wedge \ldots \wedge U_n) = T$, $i$ se numește model al mulțimii $\{U_1, U_2, \ldots, U_n\}$

## Mulțime inconsistentă (nerealizabilă, contradictorie)

O mulțime $\{U_1, U_2, \dots, U_n\}$ de formule se numește inconsistentă (nerealizabilă) $\Leftrightarrow$ formula $U_1 \wedge U_2 \wedge \dots \wedge U_n$ este inconsistentă, adică $\forall i: F_P \to \{T, F\}$ astfel încât $i(U_1 \wedge U_2 \wedge \dots \wedge U_n) = F$, $i$ se numește anti-model al mulțimii $\{U_1, U_2, \dots, U_n\}$

## Consecință logică

Formula $V$ este consecință logică a mulțimii de formule $\{U_1, U_2, \dots, U_n\}$ și se ~~numește~~ notează $U_1, U_2, \dots, U_n \models V \Leftrightarrow \forall i: F_P \to \{T, F\}$ astfel încât $i(U_1 \wedge U_2 \wedge \dots \wedge U_n) = T$, are loc $i(V) = T$.

Formulele $U_1, U_2, \dots, U_n$ se numesc premise, ipoteze, fapte, iar $V$ se numește concluzie.

### Teoremă

Fie $S = \{U_1, U_2, \dots, U_n\}$ o mulțime de formule propoziționale.

1. Dacă $S$ este o mulțime consistentă, atunci $\forall j\ 1 \le j \le n$ $S \setminus \{U_j\}$ este o mulțime consistentă.
2. Dacă $S$ este o mulțime consistentă și $V$ este o formulă validă (tautologie), atunci mulțimea $S \cup \{V\}$ este consistentă.
3. Dacă $S$ este o mulțime inconsistentă, atunci $\forall V \in F_P$ mulțimea $S \cup \{V\}$ este inconsistentă.
4. Dacă $S$ este o mulțime inconsistentă și $U_j$ este o formulă validă, unde $1 \le j \le n$, atunci mulțimea $S \setminus \{U_j\}$ este inconsistentă.

### Teoremă

Fie $U_1, U_2, \dots, U_n, V$ formule propoziționale

- $\models U \Leftrightarrow \neg U$ inconsistentă
  (O formulă este tautologie $\Leftrightarrow$ negația ei este inconsistentă)
- $U \models V \Leftrightarrow \models U \to V \Leftrightarrow$ mulțimea $\{U, \neg V\}$ este inconsistentă
- $U \equiv V \Leftrightarrow \models U \leftrightarrow V$
- $U_1, U_2, \dots, U_n \models V \Leftrightarrow \models U_1 \wedge U_2 \wedge \dots \wedge U_n \to V$
  $\Leftrightarrow$ mulțimea $\{U_1, U_2, \dots, U_n, \neg V\}$ este inconsistentă.

## Echivalențe logice

Legile lui DeMorgan

$\neg(U \wedge V) \equiv \neg U \vee \neg V$ și $\neg(U \vee V) \equiv \neg U \wedge \neg V$

Legile de absorbție

$U \wedge (U \vee V) \equiv U$ și $U \vee (U \wedge V) \equiv U$

Legile de comutativitate

$U \wedge V \equiv V \wedge U$ și $U \vee V \equiv V \vee U$

Legile de asociativitate

$U \wedge (V \wedge Z) \equiv (U \wedge V) \wedge Z$ și $U \vee (V \vee Z) \equiv (U \vee V) \vee Z$

Legile de distributivitate

$U \wedge (V \vee Z) \equiv (U \wedge V) \vee (U \wedge Z)$ și $U \vee (V \wedge Z) \equiv (U \vee V) \wedge (U \vee Z)$

Legile de idempotență

$U \wedge U \equiv U$ și $U \vee U \equiv U$

## Principiul dualității

pt orice echivalență logică $U \equiv V$ care conține doar conectivele $\neg, \wedge, \vee, \uparrow, \downarrow$ $\exists$ o altă echivalență logică, $U' \equiv V'$, unde $U', V'$ sunt formule obținute din $U, V$ prin interschimbarea conectivelor logice duale: $(\wedge, \vee)$, $(\uparrow, \downarrow)$ și a valorilor de adevăr $T, F$

Conective duale
$(\wedge, \vee)$, $(\uparrow, \downarrow)$, $(\leftrightarrow, \oplus)$
Valori de adevăr duale: $T$ și $F$
Concepte duale:
tautologie și formulă inconsistentă

## Forme normale

literal = variabilă propozițională sau negația sa ex $p, \neg q, \neg r, r$

(un literal este pur dacă negația nu apare în clauză)

clauză = disjuncția unui nr finit de literali

ex $p \vee \neg q \vee r$

cub = conjuncția unui nr finit de literale

ex $p \wedge \neg q \wedge r$

formă normală disjunctivă (FND) = disjuncție de cuburi

ex $\underbrace{(\neg p \wedge q \wedge r)}_{\text{cub}} \underset{\text{disjuncție}}{\vee} (\neg p \wedge q \wedge \neg r) \vee (p \wedge q \wedge \neg r)$

forma normală conjunctivă (FNC) = conjuncție de clauze

ex $\underbrace{(\neg p \vee q \vee r)}_{\text{clauză}} \underset{\text{conj}}{\wedge} (\neg p \vee \neg r) \wedge (p \vee q)$

## Proprietate

Fie mulțimea de literali $\{l_1, l_2, \dots, l_m\}$. Următoarele afirmații sunt echivale

- clauza $\vee_{i=1}^{m} l_i$ este validă
- cubul $\wedge_{i=1}^{m} l_i$ este inconsistent
- în mulțimea $\{l_1, l_2, \dots, l_m\}$ există cel puțin o pereche de literali opuși, adică $\exists i, j \in \{1, \dots, m\}$ astfel încât $l_i = \neg l_j$

ex $\neg p \vee q \vee p \equiv T$

$p \wedge n \wedge \neg n \equiv F$

**Teoremă**: Orice formulă admite o FNC și o FND logic echivalente cu

## Algoritmul de normalizare

! FNC: $(\vee) \wedge (\vee) \wedge \dots \wedge (\vee)$

FND: $(\wedge) \vee (\wedge) \vee \dots \vee (\wedge)$

**Pas 1**: înlocuirea:

$U \to V$ : $\neg U \vee V$

$U \leftrightarrow V$ : $(\neg U \vee V) \wedge (\neg V \vee U)$

**Pas 2**: – legile lui deMorgan (dinspre exterior spre interior)

! negația va preceda doar variabilele propoziționale

– eliminarea negațiilor multiple: $\neg\neg U \equiv U$

**Pas 3**: – aplicarea legilor distributivității

pt. FND: $U \wedge (V \vee Z) \equiv (U \wedge V) \vee (U \wedge Z)$

pt FNC: $U \vee (V \wedge Z) \equiv (U \vee V) \wedge (U \vee Z)$

**Pas 4**: simplificarea formei obținute folosind alte echivalențe logice.

**Teoremă**:

- O formulă în FNC este tautologie $\Longleftrightarrow$ toate clauzele sale sunt valide
- O formulă în FND este inconsistentă $\Longleftrightarrow$ toate cuburile sale sunt inconsistente.

! Obs.:

Prima parte a teoremei furnizează o metodă directă de rezolvare a problemei decizionale (verificarea dacă o formulă este tautologie).

FND furnizează toate modelele formulei inițiale, prin găsirea interpretărilor care evaluează cuburile componente ca adevărate

FNC furnizează toate anti-modelele formulei inițiale prin găsirea interpretărilor care evaluează clauzele componente ca false.

# Curs 4

Sisteme axiomatice
- axiomele geometriei
- axiomele aritmeticii

## sistemul axiomatic al calculului propozițional

- propus de Hilbert; deductiv, formal.

$P = (\underbrace{\Sigma_P, F_P}_{\text{curs } 3}, A_P, R_P)$

inferență = operație logică de trecere de la un enunț la altul și în care ultimul enunț este dedus din primul

### Axiome și reguli de inferență

$A_P = \{A_1, A_2, A_3\}$ scheme axiomatice

$A_1: U \to (V \to U)$

$A_2: (U \to (V \to Z)) \to ((U \to V) \to (U \to Z))$

$A_3: (U \to V) \to (\neg V \to \neg U)$

$R_P = \{mp\}$ - o singură regulă de inferență „modus ponens"

- $U, U \to V \vdash V$

„din faptele $U$ și $U \to V$ se deduce (inferă) $V$"

### Deducția

#### Definiția deducției

- Fie formulele $U_1, U_2, \ldots, U_m$ numite ipoteze și $V$ formulă propozițională. Spunem că $V$ este deductibilă din $U_1, U_2, \ldots, U_m$ și notăm $U_1, U_2, \ldots, U_{m-1}, U_m \vdash V$, dacă există o secvență de formule $(f_1, f_2, \ldots, f_m)$ a.î. $f_m = V$ și $\forall i \in \{1, \ldots, m\}$ avem:
  - $f_i \in A_P$;
  - $f_i \in \{U_1, U_2, \ldots, U_m\}$
  - $f_j, f_k \vdash_{mp} f_i$; $j < i$ și $k < i$

- Secvența $(f_1, f_2, \ldots, f_m)$ se numește deducția lui $V$ din $U_1, U_2, \ldots, U_m$

### Noțiunea de teoremă

O formulă $U \in F_P$, astfel încât $\emptyset \vdash U$ (sau $\vdash U$) se numește teoremă.

! Obs: Teoremele sunt formule care sunt deductibile doar din axiome și folosind regula modus ponens.

## Teorema de deducție și inversa sa

**Teorema de deducție**

Dacă $U_1, U_2, \dots, U_{n-1}, U_n \vdash V$, atunci $U_1, U_2, \dots, U_{n-1} \vdash U_n \to V$

**Inversa teoremei de deducție**

Dacă $U_1, U_2, \dots, U_{n-1} \vdash U_n \to V$ atunci $U_1, U_2, \dots, U_{n-1}, U_n \vdash V$

**Consecințele teoremei de deducție**

- $\vdash U \to ((U \to V) \to V)$
- $\vdash (U \to V) \to ((V \to Z) \to (U \to Z))$ legea silogismului
- $\vdash (U \to (V \to Z)) \to (V \to (U \to Z))$ legea permutării premiselor
- $\vdash (U \to (V \to Z)) \to (U \wedge V \to Z)$ legea reuniunii premiselor
- $\vdash (U \wedge V \to Z) \to (U \to (V \to Z))$ legea separării premiselor

## Proprietățile logicii propozițiilor

Probleme decizionale: „Este o formulă propozițională o teoremă sau nu?"
„Este o formulă deductibilă dintr-o mulțime de formule

| Teorema de corectitudine: Dacă $\vdash U$ atunci $\vDash U$ (Validitatea sintactică implică validitatea semantică) | Teorema de completitudine: Dacă $\vDash U$ atunci $\vdash U$ (Validitatea semantică implică validitatea sintactică) |
|---|---|
| Teorema de corectitudine și completitudine: $\vdash U$ dacă și numai dacă $\vDash U$ | |

↳ Consecințe

Logica propozițiilor este:

1) necontradictorie: nu pot avea loc simultan $\vdash U$ și $\vdash \neg U$
2) coerentă: nu orice formulă propozițională este teoremă
3) decidabilă: se poate decide dacă o f.p. este sau nu teoremă.

# Curs 5

## Metoda tabelelor semantice

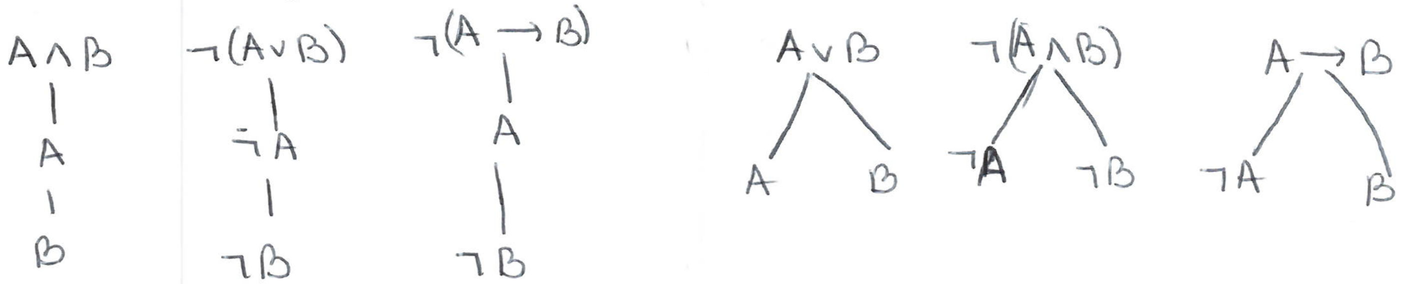
- introdusă de Smullyan
- se bazează pe considerații semantice
- încearcă să construiască modelele unei formule date (FND)

$\Rightarrow \models U$ prin respingere, $\neg U$ nu are modele

## Clase de formule

clasa $\alpha$ - formule de tip conjunctiv

clasa $\beta$ - formule de tip disjunctiv




## Arborele binar de descompunere a unei formule

→ rădăcina ⇒ formula U

→ fiecare ramură a arborelui care conține o formulă va fi extinsă cu subarborele corespunzător regulii de descompunere care se aplică formulei

→ extinderea se încheie
- dacă s-au descompus toate formulele
- dacă pe ramură apare o formulă și negația

ramură
- închisă ⊗ dacă conține o formulă și negația ei
- deschisă ⊙

ramură completă
- închisă
- toate formulele au fost descompuse

## Tipuri de tabelă semantică

O tabelă se numește
- închisă dacă toate ramurile sale sunt închise
- deschisă dacă o tabelă are cel puțin o ramură deschisă
- completă dacă toate ramurile ei sunt complete

## ! Obs

- Procesul de construire a unei tabele semantice este unul nedeterminist deoarece regulile de descompunere se pot aplica în orice ordine și la un moment dat se pot alege mai multe ramuri pentru extindere
  ⟹ unei formule i se pot asocia mai multe tabele semantice, care sunt echivalente.

→ se recomandă utilizarea regulilor de tip $\alpha$ înaintea celor $\beta$

- formulele de pe aceeași ramură a unei tabele semantice sunt legate între ele prin $\wedge$, iar ramificarea prin $\vee$
- tabela semantică este o reprezentare grafică a FND, fiecare ramură reprezintă un cub (conjuncția tuturor literalilor de pe acea ramură), iar arborele este disjuncția tuturor ramurilor sale.

- unei formule consistente i se asociază o tabelă completă deschisă iar fiecare ramură deschisă furnizează cel puțin un model pt formulă
- o tabelă semantică închisă asociată indică că formula este inconsistentă, adică nu există nicio interpretare în care formula să fie adevărată

### Teorema de corectitudine și completitudine
- a metodei tabelelor semantice -

O formulă $U$ este teoremă (tautologie) $\Longleftrightarrow$ $\exists$ o tabelă semantică închisă pentru formula $\neg U$.

### Teoremă

$U_1, U_2, \dots, U_m \vdash Y$ (echivalent cu $U_1, U_2, \dots, U_m \models Y$) dacă și numai dacă există o tabelă semantică închisă pentru formula $U_1 \wedge U_2 \wedge \dots \wedge U_m \wedge \neg Y$

# Curs 6

## Metoda rezoluției

- Robinson (1965)
- automată sintactică, prin respingere
- corectă și completă !
- scop: verificarea consistenței / inconsistenței

**Sistem axiomatic (formal)** $\longrightarrow Res = (\Sigma_{Res}, F_{Res}, A_{Res}, R_{Res})$

- alfabetul:

$\Sigma_{Res} = \Sigma_p \setminus \{\wedge, \rightarrow, \leftrightarrow\}$

- mulțimea formulelor bine-formate

$F_{Res} \cup \{\square\}$

$F_{Res}$ – mulțimea tuturor clauzelor ce se pot forma folosind $\Sigma_{Res}$

$\square$ – clauza vidă

- mulțimea axiomelor

$A_{Res} = \emptyset$

- mulțimea regulilor de inferență conține doar: regula rezoluției.

$A \vee \ell,\ B \vee \neg\ell \vdash_{Res} A \vee B$, $\ell$ literal, $A, B \in F_{Res}$

**Terminologie**

- clauzele:
  $C_1 = A \vee \ell$  $C_2 = B \vee \neg\ell$
- notație: $C_3 = Res_\ell(C_1, C_2)$
  $\Rightarrow C_3$ rezolventul clauzelor $C_1, C_2$

clauze părinte: $C_1, C_2$

caz particular: $C_1 = \ell$, $C_2 = \neg\ell$
$Res_\ell(C_1, C_2) = \square$ – inconsistență

! **Observație**

Rezoluția ca și regulă de inferență este o generalizare a regulilor
→ modus ponens
→ modus tollens
→ silogismului

MP: $U, U \rightarrow V \vdash V \Rightarrow U, \neg U \vee V \vdash V$

MT: $U \rightarrow V \vdash \neg V \overset{ITD}{\Rightarrow} U \rightarrow V, \neg V \vdash \neg U$
$\Rightarrow \neg U \vee V, \neg V \vdash \neg U$

SIL: $U \rightarrow V, V \rightarrow Z \vdash U \rightarrow Z \Rightarrow \neg U \vee V, \neg V \vee Z \vdash \neg U \vee Z$

! Notație $S \vdash_{Res} \square$ "din mulțimea S de clauze s-a derivat clauza vidă prin aplicarea algoritmului rezoluției propoziționale"

## Teorema de corectitudine și completitudine

Teorema de corectitudine
Dacă $S \vdash_{Res} \square$ atunci $S$ inconsistentă

Teorema de completitudine
Dacă $S$ este inconsistentă atunci $S \vdash_{Res} \square$

## Teorema de corectitudine și completitudine T.C.C.

! Mulțimea $S$ este inconsistentă dacă și numai dacă $S \vdash_{Res} \square$

Teoreme:

- $U$ este tautologie $\Leftrightarrow FNC(\neg U) \vdash_{Res} \square$
- $U_1, U_2, \dots, U_m \models V \Leftrightarrow U_1, U_2, \dots, U_m \vdash V \Leftrightarrow FNC(U_1 \wedge U_2 \wedge \dots \wedge U_m \wedge \neg V) \vdash_{Res} \square$

↳ tautologie ↳ deductibilă

$S_i \stackrel{not}{=} FNC(U_i)$, $i = \overline{1,m}$

$S_{m+1} \stackrel{not}{=} FNC(\neg V)$

$S_1 \cup S_2 \cup \dots \cup S_m \cup S_{m+1} \vdash_{Res} \square$

! exemple seminar + curs

## Automatizarea procesului rezolutiv

### Strategia eliminării

- procedura Davis-Putman

O mulțime $S$ poate fi simplificată păstrând (in)consistența ei prin:

→ eliminarea clauzelor tautologice (nu pot contribui la derivarea clauzei vide)

ex: $\neg p \vee q \vee p \vee r$

→ eliminarea clauzelor subsumate de alte clauze din $S$:

clauza $C_1$ este subsumată de $C_2$ dacă există o clauză $C_3$ a.î. $C_1 = C_2 \vee C_3$

→ eliminarea clauzelor care conțin literali puri în $S$.

Un literal este pur dacă negația sa nu apare în nici o clauză din $S$

Dacă $C = l$ este o clauză unitate din $S$, se șterg toate clauzele care-$l$ conțin pe $l$ și $\neg l$ din clauzele rămase.

## Strategia saturării pe nivele (algoritm)

Date de intrare: S – o mulțime de clauze
Date de ieșire: S consistentă / inconsistentă
// Se generează mulțimile de clauze $S^0, S^1, \ldots, S^k$ ce repr. nivelele

$S^0 = S$
$k = 0$

Repetă
$k = k+1$
$S^k = \{ Res(C_1, C_2) \mid C_1 \in S^0 \cup S^1 \cup \ldots \cup S^{k-1}, C_2 \in S^{k-1} \}$
$S^k = S^k \setminus (S^0 \cup S^1 \cup \ldots \cup S^{k-1})$
Până când $\square \in S^k$ sau $S^k = \emptyset$

Dacă $\square \in S^k$
atunci „S inconsistentă"
altfel „S consistentă"

(ex. curs)

## Strategia mulțimii suport

- se evită aplicarea regulii de rezoluție asupra unor clauze dintr-o submulțime consistentă a mulțimii inițiale de clauze pt că rezolvenții obținuți sunt irelevanți în procesul de derivare a $\square$
- această strategie a fost inspirată de faptul că:
  în general, mulțimea premizelor (faptelor) unei deducții este consistentă, deci rezolvarea unor clauze din această mulțime consistentă nu poate duce la derivarea clauzei vide (inconsistența)

Def: Fie S o mulțime de clauze. O submulțime Y a lui S se numește mulțime suport a lui S, dacă $S \setminus Y$ este consistentă

Rezoluția mulțimii suport este rezoluția a două clauze care nu aparțin ambele mulțimii $S \setminus Y$

## Rafinările rezoluției

- impun restricții asupra clauzelor care rezolvă, pentru a eficientiza procesul rezolutiv.

Notație: $S \vdash^{st}_{Res} \square$ „din mulțimea S de clauze s-a derivat clauza vidă prin aplicarea strategiei st a rezoluției propoziționale"

## Completitudinea și corectitudinea

- toate rafinările și strategiile rezolutive păstrează completitudinea și corectitudinea
- combinarea lor poate impune prea multe restricții și deși mulțimea inițială de clauze este inconsistentă, s-ar putea să nu se poată deriva clauza vidă.

→ sunt complete

- rezoluția generală + strategia eliminării
- rezoluția generală + strategia mulțimii suport
- rezoluția generală + strategia mulțimii suport + strategia eliminării
- rezoluția liniară + strategia eliminării
- rezoluția liniară + strategia mulțimii suport

→ NU sunt complete

- rezoluția blocării + strategia eliminării
- rezoluția blocării + strategia mulțimii suport
- rezoluția blocării + rezoluția liniară
- rezoluție unitară
- rezoluția de intrare

# Rezoluția blocării (lock resolution)

- Boyer, 1971
- fiecare apariție de literal din mulțimea de clauze este indexat arbitrar cu un întreg
- restricția: literalii care rezolvă din clauzele părinți trebuie să aibă cei mai mici indici din aceste clauze.

→ literalii din rezolvenți moștenesc indicii de la clauzele părinți, iar în cazul moștenirii a doi literali identici, se păstrează cel cu indicele mai mic

– se recomandă combinarea cu strategia saturării pe nivele

## T. C. C.

### Teorema de completitudine

Fie S o mulțime de clauze în care fiecare literal este indexat în mod arbitrar cu un întreg. Dacă S este inconsistentă, atunci există o deducție din mulțimea S a clauzei vide prin rezoluția blocării.

### Teorema de corectitudine

Fie S o mulțime de clauze în care fiecare literal este indexat în mod arbitrar cu un întreg. Dacă din S se deduce prin rezoluția blocării clauza vidă, atunci S este inconsistentă.

### ! Observație

- $C_1 = q \vee p \qquad C_4 = \neg q \vee \neg p$

$\square \equiv F$

$U \wedge \neg U \equiv F$

$A \vee l,\ B \vee \neg l \vdash_{res} A \vee B$

$Res_p(C_1, C_4) = q \vee \neg q \equiv T$

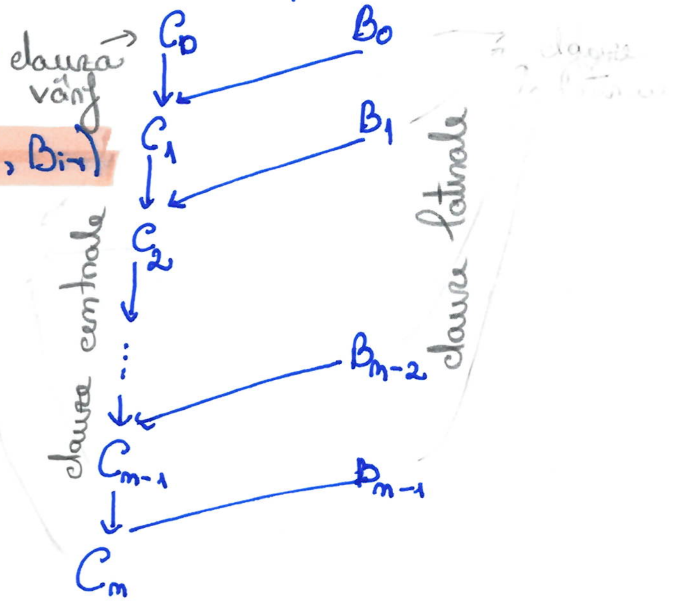
$Res_q(C_1, C_4) = p \vee \neg p \equiv T$

! $\neg(q \vee p) \equiv \neg q \wedge \neg p \neq \neg q \vee \neg p$

# Rezoluția liniară

- Loveland, 1970
- ! procesul rezolutiv este liniar: la fiecare pas una dintre clauzele părinte este rezolventul obținut la pasul anterior
- arborele de derivare:

$\forall i = \overline{1,m}$, $C_i = Res(C_{i-1}, B_{i-1})$




## Teorema de corectitudine și completitudine

Mulțimea S de clauze este inconsistentă, dacă și numai dacă $S \vdash^{lin}_{Res} \square$

! Observație

- rezoluția liniară furnizează o strategie la nivel de implementare: căutare cu revenire
  - → la fiecare iterație, pt clauza centrală pot exista mai multe posibile clauze laterale.
  - → după ce au fost utilizate toate posibilele clauze laterale, dar nu s-a obținut clauza vidă, se revine la iterația precedentă
  - → consistența este demonstrată DOAR DUPĂ O CĂUTARE COMPLETĂ fără derivarea clauzei vide.

## Cazuri particulare

Rezoluția unitară (unit): clauzele centrale au cel puțin o clauză părinte unitară (conține un singur literal)

Rezoluția de intrare (input): clauzele laterale sunt clauze inițiale (de intrare)

Teorema de echivalență dintre rezoluția unit și cea input

Fie mulțimea S de clauze. $S \vdash^{input}_{Res} \square \Longleftrightarrow S \vdash^{unit}_{Res} \square$

corectitudinea: Dacă $S \vdash^{input/unit}_{Res} \square$ atunci S este inconsistentă.

incompletitudinea: există mulțimi inconsistente de clauze din care nu se poate deriva clauza vidă folosind rez. input sau rez. unit

# TIPURI DE METODE

| | SEMANTICE | SINCTACTICE |
|---|---|---|
| DIRECTE | - tabela de adevăr<br>- FNC | - deducția (mp) |
| prin RESPINGERE | - FND<br>- tabelă semantică | - rezoluția (generală, strategia eliminării, strategia saturării pe nivele strategia mulțimii suport rafinare rez. blocării rafinarea rez liniare imput, unit) |

Notițe

Propoziții : — sunt adevărate sau false :

„Partea întreagă a nr 3,12 este 3." $\stackrel{not}{=}$ p → A

„$2+9=10$" $\stackrel{not}{=}$ r → F

„$5>7$" $\stackrel{not}{=}$ q → F

Predicate — apar variabile

„$x+7=11$, x nr real" → predicat ✓

$P(x)$ : „$x+5<10$, $x\in\mathbb{R}$" → predicat unar

$P(x,y)$ : „$x+y=7$, $x,y\in\mathbb{N}$" → predicat binar (2 variabile)

$P(x,y,z)$ : „$x^2+y^2=z^2$, $x,y,z\in\mathbb{N}$" → predicat ternar (3 variabile)

∇ dacă atribuim variabilelor valori obținem propoziții adevărate sau false
∘ (interpretare)

$P(x)$ : „$x+5<10$, $x\in\mathbb{R}$"
↳ domeniul de definiție