– Curs 2 – Semantica LP

Curs 1 : – Logica propozițională informală
↳ ambiguități
– Logica propozițională (formal)
↳ sintaxa LP

Limbajul logicii prop :
$\{ p, q, r, p_1, r', \ldots \} = A$ – var prop.
$\cup \{ \neg, \wedge, \vee \}$
$\cup \{ (,) \}$

def. inductivă

$\mathbb{LP}$ – cea mai mică mulțime a.î.

$p \in \mathbb{LP}$

$(p) \notin \mathbb{LP}$

CB : $A \subseteq \mathbb{LP}$

CI 1 : Dacă $\varphi \in \mathbb{LP}$, atunci $\neg\varphi \in \mathbb{LP}$

CI 2 : Dacă $\varphi_1, \varphi_2 \in \mathbb{LP}$, atunci $(\varphi_1 \wedge \varphi_2) \in \mathbb{LP}$

CI 3 : Dacă $\varphi_1, \varphi_2 \in \mathbb{LP}$, atunci $(\varphi_1 \vee \varphi_2) \in \mathbb{LP}$

p $\quad (p \vee q)$ $\quad \neg(p \vee q)$ $\quad \neg p$ $\quad \neg\neg p$

Ex de funcții recursive :

subf – mulțimea de subformule a unei formule date.

$\neg(p \vee q)$

$subf(\neg(p \vee q)) = \{ \neg(p \vee q), (p \vee q), p, q \}$

$$subf : \mathbb{LP} \to 2^{\mathbb{LP}}$$

$2^X = \mathcal{P}(X)$ - mulțimea părților lui X

$X = \{0,1,2\}$

$2^X = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$

$|2^X| = 2^{|X|}$

$$subf(\varphi) = \begin{cases} \{\varphi\}, & \text{dacă } \varphi \in A \\ \{\varphi\} \cup subf(\varphi'), & \text{dacă } \varphi = \neg\varphi',\ \varphi' \in \mathbb{LP} \\ \{\varphi\} \cup subf(\varphi_1) \cup subf(\varphi_2), & \text{dacă } \varphi = (\varphi_1 \land \varphi_2),\ \varphi_1, \varphi_2 \in \mathbb{LP} \\ \{\varphi\} \cup subf(\varphi_1) \cup subf(\varphi_2), & \text{dacă } \varphi = (\varphi_1 \lor \varphi_2),\ \varphi_1, \varphi_2 \in \mathbb{LP} \end{cases}$$

CB: $\varphi = p \quad \varphi = q$

CI 1: $\varphi = \neg\underbrace{(p \lor q)}_{\varphi'}$

CI 2: $\varphi = (\underbrace{\neg p}_{\varphi_1} \land \underbrace{\neg q}_{\varphi_2})$

$$subf(\underbrace{\neg\overbrace{(p \lor q)}^{\varphi'}}_{\varphi}) = \{\neg(p \lor q)\} \cup subf((\overset{\varphi_1}{p} \lor \overset{\varphi_2}{q}))$$

$$= \{\neg(p \lor q)\} \cup \{(p \lor q)\} \cup subf(p) \cup subf(q)$$

$$= \{\neg(p \lor q), (p \lor q)\} \cup \{p\} \cup \{q\}$$

$$= \{\neg(p \lor q), (p \lor q), p, q\}$$

arb - calculează arborele de sintaxă abstractă a unei formu

$$arb : \mathbb{LP} \to Trees$$

$arb(\varphi) =$

- φ, dacă $\varphi \in A$ (var. prop)

- , dacă $\varphi = \neg\varphi'$, $\varphi' \in \mathbb{LP}$

- 

, dacă $\varphi = (\varphi_1 \wedge \varphi_2)$, $\varphi_1, \varphi_2 \in LP$ — CI2

- 

, dacă $\varphi = (\varphi_1 \vee \varphi_2)$, $\varphi_1, \varphi_2 \in LP$

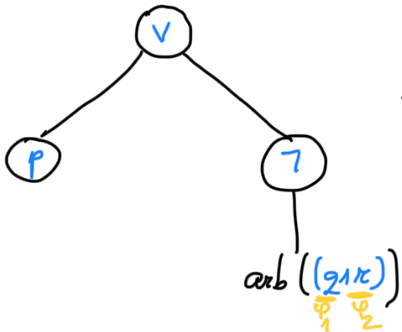
$$arb\Big((\underbrace{p}_{\varphi_1} \vee \underbrace{\neg(q \wedge r)}_{\varphi_2}) \Big) =$$

$p \in LP$

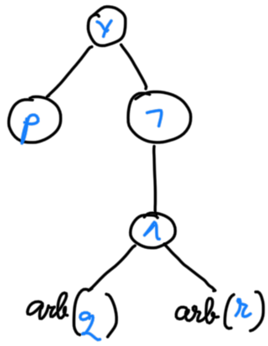
$q \in LP$ $\Big|$ $\xRightarrow{CI2}$ $(q \wedge r) \in LP$ $\Rightarrow$ ..... $\xRightarrow{CI3}$ $\varphi_1 = p$, $\varphi_2 = \neg(q \wedge r)$

$r \in LP$

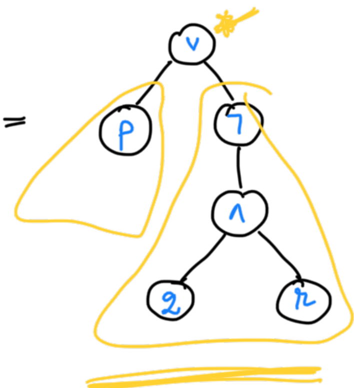
=



=



=



$\longleftrightarrow$ $\underbrace{(p \vee \neg(q \wedge r))}_{\text{string}}$

- height : $LP \to \mathbb{N}$
- size : $LP \to \mathbb{N}$
- prop : $LP \to 2^A$

Demonstrați prin inducție structurală.

ind matematică

CB: $n=0$ $\quad P(0)$ "A"

CI: pp. $P(k) \Rightarrow P(k+1)$

$\mathbb{N}$ = cea mai mică mult a.î.

CB: $0 \in \mathbb{N}$

CI: dacă $k \in \mathbb{N}$, $\underbrace{k+1}_{succ(k)} \in \mathbb{N}$

ind structurală (LP)

CB: $P(\varphi)$, $\varphi \in A$

CI 1: pp. $P(\varphi') \Rightarrow P(\neg\varphi')$

CI 2: pp. $P(\varphi_1)$, $P(\varphi_2)$ $\Rightarrow P((\varphi_1 \wedge \varphi_2))$

CI 3: ~ - - - .

Semantica logicii propoziționale

$B = \{0, 1\}$

0 – fals, 1 – adevărat

$\overline{} : B \to B$ – negația logică

$\bar{0} = 1$ și $\bar{1} = 0$

$+ : B \times B \to B$ – sau logic (disjuncția)

$0+0=0$

$0+1=1$

$1+0=1$

$1+1=1$

$\cdot : B \times B \to B$ – conjuncția logică

$0 \cdot 0 = 0$

$0 \cdot 1 = 0$

$1 \cdot 0 = 0$

$1 \cdot 1 = 1$

$(B, \overline{}, +, \cdot)$ – algebră booleană

Atribuiri

O atribuire de valori de adevăr este orice funcție $\tau : A \to B$

Ex: $\tau_1 : A \to B$

$$\begin{cases} \tau_1(p) = 1 \\ \tau_1(q) = 0 \\ \tau_1(r) = 1 \\ \tau_1(a) = 0 \text{ pt orice } a \in A \setminus \{p, q, r\} \end{cases}$$

Ex: $\tau' : A \to B$

$\tau'(a) = 0$, pt orice $a \in A$

Valoarea de adevăr a unei formule φ într-o atribuire τ

$\hat{\tau} : LP \to B$ - extensia homomorfică a lui τ

$$\hat{\tau}(\varphi) = \begin{cases} \tau(\varphi) & \text{, dacă } \varphi \in A \\ \overline{\hat{\tau}(\varphi')} & \text{, dacă } \varphi = \neg\varphi',\ \varphi' \in LP \\ \hat{\tau}(\varphi_1) \bullet \hat{\tau}(\varphi_2) & \text{, dacă } \varphi = (\varphi_1 \wedge \varphi_2),\ \varphi_1, \varphi_2 \in LP \\ \hat{\tau}(\varphi_1) + \hat{\tau}(\varphi_2) & \text{, dacă } \varphi = (\varphi_1 \vee \varphi_2),\ \varphi_1, \varphi_2 \in LP \end{cases}$$

$\varphi = p$

$\neg p$

$(p \wedge q)$

$$\hat{\tau}_1(\neg\underbrace{(p \vee q)}_{\varphi'}) = \overline{\hat{\tau}_1((p \vee q))} = \overline{\hat{\tau}_1(p) + \hat{\tau}_1(q)} = \overline{\tau_1(p) + \tau_1(q)} =$$

$$= \overline{1 + 0} = \overline{1} = 0$$

$\to \neg(p \vee q)$ este falsă în atribuirea τ_1

φ este adevarat în τ

↳ τ satisface φ

τ este model pt φ

φ ține în τ

τ face φ adevarat

$\tau \vDash \varphi$

Def: $\tau \vDash \varphi$ ddacă $\hat{\tau}(\varphi) = 1$

$\tau \nvDash \varphi$ ddacă $\hat{\tau}(\varphi) = 0$

φ este satisfiabilă ddacă există o atribuire $\tau : A \to B$

a.î. $\tau \vDash \varphi$ $(\hat{\tau}(\varphi) = 1)$

Ex: $\neg(p \vee q)$ $\hat{\tau}'(\neg(p \vee q)) = \dots = 1$

$(p \wedge \neg p)$ - nu este satisfiabilă - contradicție

φ este validă ddacă pt orice atribuire $\tau : A \to B$,

avem $\tau \vDash \varphi$

$\neg(p \vee q)$ - nu este validă $(\hat{\tau}_1(\neg(p \vee q)) = 0)$

$(p \vee \neg p)$ - validă.

φ - satisfiabilă, dar nu este validă $\approx$ contingență

$\varphi_1, \varphi_2 \in LP$ sunt echivalente și notăm $\varphi_1 \equiv \varphi_2$

ddacă pt orice atribuire $\tau : A \to B$, avem $\hat{\tau}(\varphi_1) = \hat{\tau}(\varphi_2)$

Ex: $p \equiv \neg\neg p$

Dem: Fie $\tau : A \to B$ o atribuire arbitrară

$$\left.\begin{aligned} \hat{\tau}(p) &= \tau(p) \\ \hat{\tau}(\neg\neg p) &= \overline{\hat{\tau}(\neg p)} = \overline{\overline{\hat{\tau}(p)}} = \hat{\tau}(p) \end{aligned}\right\} \Rightarrow p \equiv \neg\neg p$$

Consecință semantică

$\Gamma = \{\varphi_1, \varphi_2, \dots \varphi_n\}$

$\Gamma \vDash \varphi$

φ este consecință semantică din Γ ddacă

pt orice atribuire $\tau : A \to B$ a.î. $\hat{\tau}(\varphi_1) = \hat{\tau}(\varphi_2) = \dots = \hat{\tau}(\varphi_n)$:

avem și $\hat{\tau}(\varphi) = 1$

Ex: $\Gamma = \{p, (\neg p \vee q)\}$

$\Gamma \vDash q$

Fie $\tau : A \to B$ arbitrar a.î. $\hat{\tau}(p) = 1$ (1)

și $\hat{\tau}((\neg p \vee q)) = 1$

și arătăm că $\hat{\tau}(q) = 1$ → concluzia dorită

$\hat{\tau}((\neg p \vee q)) = 1 \Rightarrow \hat{\tau}(\neg p) + \hat{\tau}(q) = 1 \Rightarrow$

$\Rightarrow \overline{\hat{\tau}(p)} + \hat{\tau}(q) = 1$

Dar $\hat{\tau}(p) = 1$

$\Rightarrow \overline{1} + \hat{\tau}(q) = 1$

$\Rightarrow 0 + \hat{\tau}(q) = 1 \Rightarrow \hat{\tau}(q) = 1 \Rightarrow$

$\Rightarrow \Gamma \vDash q$ (deoarece τ a fost ales arbitrar)

$\vDash$ \models