

Logică pentru informatică - Săptămâna 10

Forme normale ale formulelor de ordinul I

Ștefan Ciobâcă

January 14, 2018

1 Formule echivalente

În diverse contexte, anumite formule pot avea același înțeles. De exemplu, formulele $\forall x.P(x, x)$ și $\forall y.P(y, y)$ au același înțeles în orice context. Vom numi astfel de formule echivalente. Anumite formule au același înțeles doar pentru o anumită interpretare a simbolurilor predicative și funcționale. De exemplu, dacă lucrăm într-o structură în care simbolul predicativ P este interpretat printr-un predicat simetric, formulele $P(x, y)$ și respectiv $P(y, x)$ au același înțeles. Astfel de formule se numesc echivalente în structura respectivă.

Acest aspect este surprins de următoarele definiții.

Definiția 1.1. Două formule $\varphi_1 \in LP1$ și $\varphi_2 \in LP1$ sunt echivalente în structura S dacă, pentru orice S -atribuire α ,

$$S, \alpha \models \varphi_1 \text{ dacă } S, \alpha \models \varphi_2.$$

Faptul că φ_1 și φ_2 sunt echivalente în structura S se notează $\varphi_1 \stackrel{S}{\equiv} \varphi_2$.

Exemplul 1.1. Continuăm exemplele din cursul anterior. Considerăm semnatura $\Sigma = (P, f, i, e)$, Σ -structura $S_1 = (\mathbb{Z}, =, +, -, 0)$ și S_1 -atribuirea α_1 prezentate în cursul anterior.

1. Avem că $P(x, y) \stackrel{S_1}{\equiv} P(y, x)$. De ce?

Fie α o S_1 -atribuire oarecare.

Avem $S_1, \alpha \models P(x, y)$ dacă (prin definiția relației \models) $\alpha(x) = \alpha(y)$ dacă (prin simetria relației de egalitate) $\alpha(y) = \alpha(x)$ dacă (prin definiția relației \models) $S_1, \alpha \models P(y, x)$.

Deci, pentru orice S_1 -atribuire α , avem: $S_1, \alpha \models P(x, y)$ dacă $S_1, \alpha \models P(y, x)$, care este chiar definiția $P(x, y) \stackrel{S_1}{\equiv} P(y, x)$.

2. Avem că $P(x_1, x_3) \not\stackrel{S_1}{\equiv} P(x_2, x_3)$. De ce?

Deoarece există o S_1 -atribuire $\alpha : \mathcal{X} \rightarrow \mathbb{Z}$, definită prin $\alpha(x_1) = 42, \alpha(x_2) = 7, \alpha(x_3) = 42$ (pentru restul variabilelor nu este relevantă valoarea lor în atribuire) cu proprietatea că

$$\begin{aligned} S_1, \alpha &\models P(x_1, x_3) \text{ (deoarece } 42 = 42), \text{ dar} \\ S_1, \alpha &\not\models P(x_2, x_3) \text{ (deoarece } 42 \neq 7). \end{aligned}$$

Cu alte cuvinte, două formule sunt echivalente într-o anumită structură S dacă, evaluând valoarea de adevăr a formulelor în structura S , obținem același rezultat pentru ambele formule (ambele adevărate sau ambele false), indiferent de atribuirea α cu care lucrăm.

În cazul în care structura nu este fixată, avem următoarea definiție:

Definiția 1.2. Două formule $\varphi_1 \in \text{LP1}$ și $\varphi_2 \in \text{LP1}$ sunt echivalente dacă, pentru orice structură S și pentru orice S -atribuire α ,

$$S, \alpha \models \varphi_1 \text{ ddacă } S, \alpha \models \varphi_2.$$

Faptul că φ_1 și φ_2 sunt echivalente se notează $\varphi_1 \equiv \varphi_2$.

Exemplul 1.2. Continuăm exemplul anterior.

1. Avem că $P(x, y) \not\equiv P(y, x)$. De ce?

Deoarece există o Σ -structură, și anume chiar structura $S_5 = (\mathbb{Z}, <, +, -, 0)$ definită în cursul anterior. Observați că singura diferență între S_1 și S_5 este faptul că simbolul predicativ P este interpretat prin predicatul $=$ în S_1 , în timp ce în S_5 este interpretat prin $<$ (relația mai mic strict peste numere întregi). Fie S_5 -atribuirea $\alpha_6 : \mathcal{X} \rightarrow \mathbb{Z}$, definită prin $\alpha(x) = 2, \alpha(y) = 3$ (pentru restul variabilelor nu este relevantă valoarea lor în atribuire).

Avem $S_5, \alpha_6 \models P(x, y)$, deoarece $2 < 3$, dar $S_5, \alpha_6 \not\models P(y, x)$, deoarece $3 \not< 2$. Deci formulele $P(x, y)$ și $P(y, x)$ nu sunt echivalente (chiar dacă sunt echivalente în structura S_1).

2. Avem că $\forall x.P(x, z) \equiv \forall y.P(y, z)$. De ce?

Fie S o Σ -structură oarecare cu domeniul D și $\alpha : \mathcal{X} \rightarrow D$ o S -atribuire oarecare.

Avem că

$$\begin{aligned} S, \alpha &\models \forall x.P(x, z) && \text{ddacă} \\ \text{pentru orice } u \in D, &\text{avem } S, \alpha[x \mapsto u] \models P(x, z) && \text{ddacă} \\ \text{pentru orice } u \in D, &\text{avem } P^S(\overline{\alpha[x \mapsto u]}(x), \overline{\alpha[x \mapsto u]}(z)) = 1 && \text{ddacă} \\ \text{pentru orice } u \in D, &\text{avem } P^S(u, \alpha(z)) = 1 && \text{ddacă} \\ \text{pentru orice } u \in D, &\text{avem } P^S(\overline{\alpha[y \mapsto u]}(y), \overline{\alpha[y \mapsto u]}(z)) = 1 && \text{ddacă} \\ \text{pentru orice } u \in D, &\text{avem } S, \alpha[y \mapsto u] \models P(y, z) && \text{ddacă} \\ S, \alpha &\models \forall y.P(y, z). \end{aligned}$$

Deci, pentru orice Σ -structură S , pentru orice S -atribuire α , avem că

$$S, \alpha \models \forall x.P(x, z) \text{ dacă } S, \alpha \models \forall y.P(y, z),$$

care este chiar definiția faptului că $\forall x.P(x, z) \equiv \forall y.P(y, z)$.

2 Substituții

Definiția 2.1. O substituție este o funcție $\sigma : \mathcal{X} \rightarrow \mathcal{T}$, cu proprietatea că $\sigma(x) \neq x$ pentru un număr finit de variabile $x \in \mathcal{X}$.

Definiția 2.2. Dacă $\sigma : \mathcal{X} \rightarrow \mathcal{T}$ este o substituție, atunci mulțimea $\text{dom}(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$ se numește domeniul substituției σ .

Observație 2.1. Prin definiție, domeniul unei substituții este o mulțime finită.

Definiția 2.3. Dacă $\sigma : \mathcal{X} \rightarrow \mathcal{T}$ este o substituție, atunci extensia substituției σ la mulțimea termenilor este funcția $\sigma^\# : \mathcal{T} \rightarrow \mathcal{T}$, definită astfel:

1. $\sigma^\#(x) = \sigma(x)$, pentru orice $x \in \mathcal{X}$;
2. $\sigma^\#(c) = c$, pentru orice simbol constant $c \in \mathcal{F}_0$;
3. $\sigma^\#(f(t_1, \dots, t_n)) = f(\sigma^\#(t_1), \dots, \sigma^\#(t_n))$, pentru orice simbol funcțional $f \in \mathcal{F}_n$ de aritate $n \in \mathbb{N}$ și orice termeni $t_1, \dots, t_n \in \mathcal{T}$.

Substituțiile se notează cu $\sigma, \tau, \sigma_0, \tau_1, \sigma'$, etc.

Observație 2.2. Dacă $t \in \mathcal{T}$ este un termen, atunci $\sigma^\#(t) \in \mathcal{T}$ este termenul obținut din t prin aplicarea substituției σ sau termenul obținut prin aplicarea substituției σ asupra termenului t .

Practic, pentru a obține $\sigma^\#(t)$ din t , fiecare apariție a unei variabile x din t este înlocuită cu termenul $\sigma(x)$.

Exemplul 2.1. Fie substituția $\sigma_1 : \mathcal{X} \rightarrow \mathcal{T}$ definită astfel:

1. $\sigma_1(x_1) = x_2$;
2. $\sigma_1(x_2) = f(x_3, x_4)$;
3. $\sigma_1(x) = x$ pentru orice $x \in \mathcal{X} \setminus \{x_1, x_2\}$.

Fie termenul $t = f(f(x_1, x_2), f(x_3, e))$. Avem că:

$$\begin{aligned} \sigma_1^\#(t) &= \sigma_1^\#(f(f(x_1, x_2), f(x_3, e))) \\ &= f(\sigma_1^\#(f(x_1, x_2)), \sigma_1^\#(f(x_3, e))) \\ &= f(f(\sigma_1^\#(x_1), \sigma_1^\#(x_2)), f(\sigma_1^\#(x_3), \sigma_1^\#(e))) \\ &= f(f(\sigma_1(x_1), \sigma_1(x_2)), f(\sigma_1(x_3), e)) \\ &= f(f(x_2, f(x_3, x_4)), f(x_3, e)). \end{aligned}$$

Practic, prin aplicarea unei substituții asupra unui termen, se înlocuiesc (simultan) toate aparițiile variabilelor din domeniul substituției cu termenii asociați acestora.

Notăție 2.1. Dacă $\text{dom}(\sigma) = \{x_1, \dots, x_n\}$, atunci substituția σ se mai poate scrie în felul următor:

$$\sigma = \{x_1 \mapsto \sigma(x_1), \dots, x_n \mapsto \sigma(x_n)\}.$$

Atenție, nu este vorba de o mulțime, ci de o notație pentru substituție.

Exemplul 2.2. Pentru substituția din exemplul anterior, avem

$$\sigma_1 = \{x_1 \mapsto x_2, x_2 \mapsto f(x_3, x_4)\}.$$

Definiția 2.4. Dacă $\sigma : \mathcal{X} \rightarrow \mathcal{T}$ este o substituție, $V \subseteq \mathcal{X}$ este o submulțime de variabile, atunci restricția substituției σ la mulțimea V este o nouă substituție notată $\sigma|_V : \mathcal{X} \rightarrow \mathcal{T}$, definită astfel:

1. $\sigma|_V(x) = \sigma(x)$ pentru orice $x \in V$;
2. $\sigma|_V(x) = x$ pentru orice $x \in \mathcal{X} \setminus V$.

Exemplul 2.3. $\sigma_1|_{\{x_1\}} = \{x_1 \mapsto x_2\}$.

Practic, prin restricția unei substituții la o mulțime de variabile, se scot celelalte variabile din domeniul substituției.

Definiția 2.5. Pentru orice substituție $\sigma : \mathcal{X} \rightarrow \mathcal{T}$, extensia lui σ la mulțimea formulelor este funcția $\sigma^b : LP1 \rightarrow LP1$, definită astfel:

1. $\sigma^b(P(t_1, \dots, t_n)) = P(\sigma^\#(t_1), \dots, \sigma^\#(t_n))$;
2. $\sigma^b(\neg\varphi) = \neg\sigma^b(\varphi)$;
3. $\sigma^b(\varphi_1 \wedge \varphi_2) = \sigma^b(\varphi_1) \wedge \sigma^b(\varphi_2)$;
4. $\sigma^b(\varphi_1 \vee \varphi_2) = \sigma^b(\varphi_1) \vee \sigma^b(\varphi_2)$;
5. $\sigma^b(\varphi_1 \rightarrow \varphi_2) = \sigma^b(\varphi_1) \rightarrow \sigma^b(\varphi_2)$;
6. $\sigma^b(\varphi_1 \leftrightarrow \varphi_2) = \sigma^b(\varphi_1) \leftrightarrow \sigma^b(\varphi_2)$;
7. $\sigma^b(\forall x.\varphi) = \forall x.(\rho^b(\varphi))$, unde $\rho = \sigma|_{\text{dom}(\sigma) \setminus \{x\}}$;
8. $\sigma^b(\exists x.\varphi) = \exists x.(\rho^b(\varphi))$, unde $\rho = \sigma|_{\text{dom}(\sigma) \setminus \{x\}}$.

Practic, pentru a obține formula $\sigma^b(\varphi)$ din formula φ , fiecare apariție liberă a variabilei x din formula φ este înlocuită cu termenul $\sigma(x)$.

Exemplul 2.4.

$$\begin{aligned}
& \sigma_1^b \left((\forall x_2. P(x_1, x_2)) \wedge P(x_2, x_2) \right) \equiv \\
& \sigma_1^b \left((\forall x_2. P(x_1, x_2)) \right) \wedge \sigma_1^b \left(P(x_2, x_2) \right) \equiv \\
& (\forall x_2. \sigma_1|_{\{x_1\}}^b \left(P(x_1, x_2) \right)) \wedge P(\sigma_1^\#(x_2), \sigma_1^\#(x_2)) \equiv \\
& (\forall x_2. P(\sigma_1|_{\{x_1\}}^\#(x_1), \sigma_1|_{\{x_1\}}^\#(x_2))) \wedge P(\sigma_1(x_2), \sigma_1(x_2)) \equiv \\
& (\forall x_2. P(\sigma_1|_{\{x_1\}}(x_1), \sigma_1|_{\{x_1\}}(x_2))) \wedge P(f(x_3, x_4), f(x_3, x_4)) \equiv \\
& (\forall x_2. P(\sigma_1(x_1), x_2)) \wedge P(f(x_3, x_4), f(x_3, x_4)) \equiv \\
& (\forall x_2. P(x_2, x_2)) \wedge P(f(x_3, x_4), f(x_3, x_4)).
\end{aligned}$$

Observație 2.3. *Atenție! Aparițiile legate ale variabilelor nu sunt înlocuite prin aplicarea substituției.*

3 Forma normală prenex

Definiția 3.1. *O formulă φ este în formă normală prenex dacă*

$$\varphi = Q_1x_1.Q_2x_2.\dots.Q_nx_n.\varphi',$$

unde:

1. $Q_i \in \{\forall, \exists\}$ (pentru orice $1 \leq i \leq n$);
2. φ' nu conține cuantificatori.

Practic, o formulă este în formă normală prenex (FNP), dacă toți cuantificatorii sunt “în fața formulei”.

Exemplul 3.1. 1. formula $\forall x.\exists y.(P(x, y) \wedge \neg P(z, y))$ este în formă normală prenex;

2. formula $\forall x.((\exists y.P(x, y)) \wedge \neg P(z, y))$ nu este în formă normală prenex.

Orice formulă poate fi adusă în formă normală prenex, fapt surpins de următoarea teoremă:

Teorema 3.1. *Pentru orice formulă $\varphi \in LP1$, există $\varphi' \in LP1$ astfel încât:*

1. φ' este în formă normală prenex;
2. $\varphi \equiv \varphi'$.

Formula φ' este o formă normală prenex a formulei φ .

În continuare, demonstrăm teorema de mai sus prin intermediul unui algoritim care calculează φ' pornind de la φ . Pentru a prezenta algoritmul, avem nevoie de următoarea leamnă:

Lema 3.1 (Lema redenumirii). Fie $\varphi \in LP1$ o formulă și $x, y \in \mathcal{X}$ două variabile cu proprietatea că $y \notin \text{free}(\varphi)$.

Atunci au loc echivalențele:

$$\forall x.\varphi \equiv \forall y.(\sigma^b(\varphi)) \text{ și } \exists x.\varphi \equiv \exists y.(\sigma^b(\varphi)),$$

unde $\sigma = \{x \mapsto y\}$.

Exemplul 3.2. Fie formula $\forall x.P(x, y)$. Deoarece $z \notin \text{free}(P(x, y))$, avem prin lema redenumirii că $\forall z.P(z, y)$.

Atenție, $\forall x.P(x, y) \not\equiv \forall y.P(y, y)$ (echivalența nu poate fi explicată prin lema redenumirii deoarece $y \in \text{free}(P(x, y))$ și nici nu are loc).

Suntem acum pregătiți să prezentăm, schițat, demonstrația Teoremei 3.1.

Proof. Se aplică următoarele echivalențe, de la stânga la dreapta:

1. $(\forall x.\varphi_1) \wedge \varphi_2 \equiv \forall x.(\varphi_1 \wedge \varphi_2)$, dacă $x \notin \text{free}(\varphi_2)$;
2. $(\forall x.\varphi_1) \vee \varphi_2 \equiv \forall x.(\varphi_1 \vee \varphi_2)$, dacă $x \notin \text{free}(\varphi_2)$;
3. $(\exists x.\varphi_1) \wedge \varphi_2 \equiv \exists x.(\varphi_1 \wedge \varphi_2)$, dacă $x \notin \text{free}(\varphi_2)$;
4. $(\exists x.\varphi_1) \vee \varphi_2 \equiv \exists x.(\varphi_1 \vee \varphi_2)$, dacă $x \notin \text{free}(\varphi_2)$;
5. $\neg \forall x.\varphi \equiv \exists x.\neg \varphi$;
6. $\neg \exists x.\varphi \equiv \forall x.\neg \varphi$.

În cazul în care una dintre primele patru echivalențe nu poate fi aplicată din cauza restricției $x \notin \text{free}(\varphi_2)$, trebuie să aplicăm de asemenea lema redenumirii pentru a redenumi convenabil variabila legată x .

De asemenea, vom folosi, când este necesar, comutativitatea conectorilor \wedge și \vee , adică:

1. $\varphi_1 \wedge \varphi_2 \equiv \varphi_2 \wedge \varphi_1$;
2. $\varphi_1 \vee \varphi_2 \equiv \varphi_2 \vee \varphi_1$.

Efectul primelor 6 echivalențe de mai sus este de muta cuantificatorii, în arborele formulei, deasupra conectorilor \wedge, \vee, \neg , asigurând astfel terminarea algoritmului și faptul că într-un final toți cuantificatorii vor fi cât mai apropiați de rădăcina arborelui.

Pentru a trata conectorii $\rightarrow, \leftrightarrow$, putem folosi “traducerea” lor cu ajutorul conectorilor \wedge, \vee, \neg :

1. $\varphi_1 \rightarrow \varphi_2 \equiv \neg \varphi_1 \vee \varphi_2$;
2. $\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$.

□

În continuare, dăm un exemplu de calcul al unei forme normale prenex pentru o formulă, folosind algoritmul de mai sus.

Exemplul 3.3. Fie formula $\varphi = (\forall x. \neg(P(x, x) \wedge \neg \exists y. P(x, y))) \wedge P(x, x)$.

Nu putem aplica prima echivalență pentru a aduce cuantificatorul $\forall x$ în fața formulei deoarece $x \in \text{free}(P(x, x))$. Așadar trebuie să aplicăm întâi lema reenumirii (L.R.):

$$\begin{aligned}
\varphi &= (\forall x. \neg(P(x, x) \wedge \neg \exists y. P(x, y))) \wedge P(x, x) \\
&\stackrel{\text{L.R.}}{\equiv} (\forall z. \neg(P(z, z) \wedge \neg \exists y. P(z, y))) \wedge P(x, x) \\
&\stackrel{1}{\equiv} \forall z. (\neg(P(z, z) \wedge \neg \exists y. P(z, y)) \wedge P(x, x)) \\
&\stackrel{6}{\equiv} \forall z. (\neg(P(z, z) \wedge \forall y. \neg P(z, y)) \wedge P(x, x)) \\
&\stackrel{\text{commutativity}}{\equiv} \forall z. (\neg((\forall y. \neg P(z, y)) \wedge P(z, z)) \wedge P(x, x)) \\
&\stackrel{1}{\equiv} \forall z. (\neg(\forall y. (\neg P(z, y) \wedge P(z, z))) \wedge P(x, x)) \\
&\stackrel{\text{commutativity}}{\equiv} \forall z. (\neg(\forall y. (P(z, z) \wedge \neg P(z, y))) \wedge P(x, x)) \\
&\stackrel{5}{\equiv} \forall z. ((\exists y. \neg(P(z, z) \wedge \neg P(z, y))) \wedge P(x, x)) \\
&\stackrel{3}{\equiv} \forall z. \exists y. (\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x)).
\end{aligned}$$

Așadar, am găsit că formula $\forall z. \exists y. (\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x))$ este o formă normală prenex a formulei $(\forall x. \neg(P(x, x) \wedge \neg \exists y. P(x, y))) \wedge P(x, x)$.

Când facem calculele, de obicei nu marcăm explicit aplicarea unei comutativități și scriem mai pe scurt, în felul următor:

$$\begin{aligned}
\varphi &= (\forall x. \neg(P(x, x) \wedge \neg \exists y. P(x, y))) \wedge P(x, x) \\
&\stackrel{\text{L.R.}}{\equiv} (\forall z. \neg(P(z, z) \wedge \neg \exists y. P(z, y))) \wedge P(x, x) \\
&\stackrel{1}{\equiv} \forall z. (\neg(P(z, z) \wedge \neg \exists y. P(z, y)) \wedge P(x, x)) \\
&\stackrel{6}{\equiv} \forall z. (\neg(P(z, z) \wedge \forall y. \neg P(z, y)) \wedge P(x, x)) \\
&\stackrel{1}{\equiv} \forall z. (\neg(\forall y. (P(z, z) \wedge \neg P(z, y))) \wedge P(x, x)) \\
&\stackrel{5}{\equiv} \forall z. ((\exists y. \neg(P(z, z) \wedge \neg P(z, y))) \wedge P(x, x)) \\
&\stackrel{3}{\equiv} \forall z. \exists y. (\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x)).
\end{aligned}$$

4 Formule închise

Definiția 4.1. O formulă $\varphi \in \text{LP1}$ este închisă dacă $\text{free}(\varphi) = \emptyset$.

Cu alte cuvinte, dacă o formulă nu are variabile libere, aceasta se numește închisă. Formulele închise se mai numesc și *propoziții* (engl. sentences).

Definiția 4.2. O formulă care nu este închisă se numește deschisă.

Exemplul 4.1. Formula $\forall z.\exists y.(\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x))$ nu este închisă (este deschisă), deoarece $\text{free}(\forall z.\exists y.(\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x))) = \{x\}$.

Definiția 4.3. Fie $\varphi \in \text{LP1}$ o formulă și $\text{free}(\varphi) = \{x_1, \dots, x_n\}$ mulțimea variabilelor libere ale acesteia.

Formula

$$\exists x_1.\exists x_2.\dots.\exists x_n.\varphi$$

se numește închiderea existențială a formulei φ .

Observație 4.1. Închiderea existențială a unei formule este o formulă închisă.

Exemplul 4.2. Închiderea existențială a formulei $\forall z.\exists y.(\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x))$ este $\exists x.\forall z.\exists y.(\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x))$.

Definiția 4.4. Două formule $\varphi_1 \in \text{LP1}$ și $\varphi_2 \in \text{LP1}$ sunt echisatisfiabile, dacă:

1. sau atât φ_1 cât și φ_2 sunt satisfiabile;
2. sau nici φ_1 și nici φ_2 nu sunt satisfiabile.

Cu alte cuvinte, singurele cazuri când două formule nu sunt echisatisfiabile sunt când una dintre formule este satisfiabilă, iar cealaltă nu este satisfiabilă.

Teorema 4.1. Orice formulă este echisatisfiabilă cu închiderea ei existențială.

Definiția 4.5. Fie $\varphi \in \text{LP1}$ o formulă și $\text{free}(\varphi) = \{x_1, \dots, x_n\}$ mulțimea variabilelor libere ale acesteia.

Formula

$$\forall x_1.\forall x_2.\dots.\forall x_n.\varphi$$

se numește închiderea universală a formulei φ .

Observație 4.2. Închiderea universală a unei formule este o formulă închisă.

Exemplul 4.3. Închiderea universală a formulei $\forall z.\exists y.(\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x))$ este $\forall x.\forall z.\exists y.(\neg(P(z, z) \wedge \neg P(z, y)) \wedge P(x, x))$.

Teorema 4.2. O formulă este validă dacă și numai dacă închiderea ei universală este validă.