Logică pentru informatică - Săptămâna 10
Forme normale ale formulelor de ordinul I

Ștefan Ciobâcă

December 14, 2017

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.

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

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

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

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

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

   Fie α o S_1-atribuire oarecare.

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

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

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

Deoarece există o S_1-atribuire $\alpha : \mathcal{X} \to \mathbb{Z}$, 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

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

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:

Definition 1.2. *Două formule $\varphi_1 \in$ LP1 și $\varphi_2 \in$ 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 $\varphi_1 \equiv \varphi_2$.

Example 1.2. *Continuăm exemplul anterior.*

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

   Deoarece există o Σ-structură, să îi spunem S_6, definită astfel: $S_6 = (\mathbb{Z}, <, +, -, 0)$. Observați că singura diferență între S_1 și S_6 este faptul că simbolul predicativ P este interpretat prin predicatul $=$ în S_1, în timp ce în S_6 este interpretat prin $<$ (relația mai mic strict peste numere întregi). Fie S_6-atribuirea $\alpha_6 : \mathcal{X} \to \mathbb{Z}$, definită prin $\alpha(x) = 2, \alpha(y) = 3$ (pentru restul variabilelor nu este relevantă valoarea lor în atribuire).

   Avem $S_6, \alpha_6 \models P(x, y)$, deoarece $2 < 3$, dar $S_6, \alpha_6 \not\models P(y, x)$, 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 $\alpha : \mathcal{X} \to D$ o S-atribuire oarecare.

   Avem că

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

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

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

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

2 Substituţii

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

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

Remark 2.1. *Prin definiție, domeniul unei subtituții este o mulțime finită.*

Definition 2.3. *Dacă $\sigma : \mathcal{X} \to \mathcal{T}$ este o substituție, atunci* extensia *substituției σ la mulțimea termenilor este funcția $\sigma^\sharp : \mathcal{T} \to \mathcal{T}$, definită astfel:*

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

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

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

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

Example 2.1. *Fie substituția $\sigma_1 : \mathcal{X} \to \mathcal{T}$ 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:

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

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

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

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

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

Example 2.2. *Pentru substituția din exemplul anterior, avem*

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

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

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

Example 2.3. $\sigma_1|_V = \{x_1 \mapsto x_2\}$.

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

Definition 2.5. *Pentru orice substituție* $\sigma : \mathcal{X} \to \mathcal{T}$*,* extensia *lui* σ *la mulțimea formulelor este funcția* $\sigma^\flat : \texttt{LP1} \to \texttt{LP1}$*, definită astfel:*

1. $\sigma^\flat(P(t_1, \ldots, t_n)) = P(\sigma^\sharp(t_1), \ldots, \sigma^\sharp(t_n))$*;*
2. $\sigma^\flat(\neg\varphi) = \neg\sigma^\flat(\varphi)$*;*
3. $\sigma^\flat(\varphi_1 \wedge \varphi_2) = \sigma^\flat(\varphi_1) \wedge \sigma^\flat(\varphi_2)$*;*
4. $\sigma^\flat(\varphi_1 \vee \varphi_2) = \sigma^\flat(\varphi_1) \vee \sigma^\flat(\varphi_2)$*;*
5. $\sigma^\flat(\varphi_1 \to \varphi_2) = \sigma^\flat(\varphi_1) \to \sigma^\flat(\varphi_2)$*;*
6. $\sigma^\flat(\varphi_1 \leftrightarrow \varphi_2) = \sigma^\flat(\varphi_1) \leftrightarrow \sigma^\flat(\varphi_2)$*;*
7. $\sigma^\flat(\forall x.\varphi) = \forall x.(\sigma'^\flat(\varphi))$*, unde* $\sigma' = \sigma|_{dom(\sigma)\setminus\{x\}}$*;*
8. $\sigma^\flat(\exists x.\varphi) = \exists x.(\sigma'^\flat(\varphi))$*, unde* $\sigma' = \sigma|_{dom(\sigma)\setminus\{x\}}$*.*

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

Example 2.4.

$$\begin{aligned}
&\sigma_1^{\flat}\Big(\big(\forall x_2.P(x_1,x_2)\big) \wedge P(x_2,x_2)\Big) \equiv \\
&\sigma_1^{\flat}\Big(\big(\forall x_2.P(x_1,x_2)\big)\Big) \wedge \sigma_1^{\flat}\Big(P(x_2,x_2)\Big) \equiv \\
&\big(\forall x_2.\sigma_1|^{\flat}_{\{x_1\}}\Big(P(x_1,x_2)\Big)\big) \wedge P(\sigma_1^{\sharp}\Big(x_2\Big), \sigma_1^{\sharp}\Big(x_2\Big)) \equiv \\
&\big(\forall x_2.P(\sigma_1|^{\sharp}_{\{x_1\}}\Big(x_1\Big), \sigma_1|^{\sharp}_{\{x_1\}}\Big(x_2\Big))\big) \wedge P(\sigma_1\Big(x_2\Big), \sigma_1\Big(x_2\Big)) \equiv \\
&\big(\forall x_2.P(\sigma_1|_{\{x_1\}}\Big(x_1\Big), \sigma_1|_{\{x_1\}}\Big(x_2\Big))\big) \wedge P(f(x_3,x_4), f(x_3,x_4)) \equiv \\
&\big(\forall x_2.P(\sigma_1\Big(x_1\Big), x_2)\big) \wedge P(f(x_3,x_4), f(x_3,x_4)) \equiv \\
&\big(\forall x_2.P(x_2,x_2)\big) \wedge P(f(x_3,x_4), f(x_3,x_4)).
\end{aligned}$$

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

3 Forma normală prenex

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

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

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".

Example 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:

Theorem 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 algoritm care calculează φ' pornind de la φ. Pentru a prezenta algoritmul, avem nevoie de următoarea lemă:

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

Atunci au loc echivalențele:

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

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

Example 3.2. *Fie formula $\forall x.P(x, y)$. Deoarece $z \notin free(P(x, y))$, 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 $y \in free(P(x, y))$ 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 $x \notin free(\varphi_2)$;
2. $(\forall x.\varphi_1) \vee \varphi_2 \equiv \forall x.(\varphi_1 \vee \varphi_2)$, dacă $x \notin free(\varphi_2)$;
3. $(\exists x.\varphi_1) \wedge \varphi_2 \equiv \exists x.(\varphi_1 \wedge \varphi_2)$, dacă $x \notin free(\varphi_2)$;
4. $(\exists x.\varphi_1) \vee \varphi_2 \equiv \exists x.(\varphi_1 \vee \varphi_2)$, dacă $x \notin free(\varphi_2)$;
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 $x \notin free(\varphi_2)$, 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.

Example 3.3. *Fie formula* $\varphi = \Big(\forall x.\neg\big(P(x,x) \wedge \neg\exists y.P(x,y)\big)\Big) \wedge P(x,x)$.

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

$$\begin{array}{rcl}
\varphi & = & \Big(\forall x.\neg\big(P(x,x) \wedge \neg\exists y.P(x,y)\big)\Big) \wedge P(x,x) \\
& \overset{L.R.}{\equiv} & \Big(\forall z.\neg\big(P(z,z) \wedge \neg\exists y.P(z,y)\big)\Big) \wedge P(x,x) \\
& \overset{1}{\equiv} & \forall z.\Big(\neg\big(P(z,z) \wedge \neg\exists y.P(z,y)\big) \wedge P(x,x)\Big) \\
& \overset{6}{\equiv} & \forall z.\Big(\neg\big(P(z,z) \wedge \forall y.\neg P(z,y)\big) \wedge P(x,x)\Big) \\
& \overset{commutativity\ \wedge}{\equiv} & \forall z.\Big(\neg\big((\forall y.\neg P(z,y)) \wedge P(z,z)\big) \wedge P(x,x)\Big) \\
& \overset{1}{\equiv} & \forall z.\Big(\neg\big(\forall y.(\neg P(z,y) \wedge P(z,z))\big) \wedge P(x,x)\Big) \\
& \overset{commutativity\ \wedge}{\equiv} & \forall z.\Big(\neg\big(\forall y.(P(z,z) \wedge \neg P(z,y))\big) \wedge P(x,x)\Big) \\
& \overset{5}{\equiv} & \forall z.\Big(\big(\exists y.\neg(P(z,z) \wedge \neg P(z,y))\big) \wedge P(x,x)\Big) \\
& \overset{3}{\equiv} & \forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x)).
\end{array}$$

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* $\Big(\forall x.\neg\big(P(x,x) \wedge \neg\exists y.P(x,y)\big)\Big) \wedge P(x,x)$.

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

$$\begin{array}{rcl}
\varphi & = & \Big(\forall x.\neg\big(P(x,x) \wedge \neg\exists y.P(x,y)\big)\Big) \wedge P(x,x) \\
& \overset{L.R.}{\equiv} & \Big(\forall z.\neg\big(P(z,z) \wedge \neg\exists y.P(z,y)\big)\Big) \wedge P(x,x) \\
& \overset{1}{\equiv} & \forall z.\Big(\neg\big(P(z,z) \wedge \neg\exists y.P(z,y)\big) \wedge P(x,x)\Big) \\
& \overset{6}{\equiv} & \forall z.\Big(\neg\big(P(z,z) \wedge \forall y.\neg P(z,y)\big) \wedge P(x,x)\Big) \\
& \overset{1}{\equiv} & \forall z.\Big(\neg\big(\forall y.(P(z,z) \wedge \neg P(z,y))\big) \wedge P(x,x)\Big) \\
& \overset{5}{\equiv} & \forall z.\Big(\big(\exists y.\neg(P(z,z) \wedge \neg P(z,y))\big) \wedge P(x,x)\Big) \\
& \overset{3}{\equiv} & \forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x)).
\end{array}$$

4 Formule închise

Definition 4.1. *O formulă* $\varphi \in$ `LP1` *este* închisă *dacă* $free(\varphi) = \emptyset$.

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

Definition 4.2. *O formulă care nu este închisă se numește* deschisă.

Example 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* $free(\forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))) = \{x\}$.

Definition 4.3. *Fie* $\varphi \in$ `LP1` *o formulă și* $free(\varphi) = \{x_1, \ldots, x_n\}$ *mulțimea variabilelor libere ale acesteia.*

Formula

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

se numește închiderea existențială a formulei φ.

Remark 4.1. *Închiderea existențială a unei formule este o formulă închisă.*

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

Definition 4.4. *Două formule* $\varphi_1 \in$ `LP1` *și* $\varphi_2 \in$ `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.

Theorem 4.1. *Orice formulă este echisatisfiabilă cu închiderea ei existențială.*

Definition 4.5. *Fie* $\varphi \in$ `LP1` *o formulă și* $free(\varphi) = \{x_1, \ldots, x_n\}$ *mulțimea variabilelor libere ale acesteia.*

Formula

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

se numește închiderea universală *a formulei* φ.

Remark 4.2. *Închiderea universală a unei formule este o formulă închisă.*

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

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