Logică pentru informatică - note de curs

Universitatea Alexandru Ioan Cuza, Iași
Facultatea de Informatică
Anul universitar 2021-2022

Ștefan Ciobâcă
Andrei Arusoaie
Rodica Condurache
Cristian Masalagiu

Cuprins

Capitolul 1

Motivație și introducere

Logica de ordinul I, pe care o vom studia în continuare, este o extensie a logicii propoziționale, extensie care aduce un plus de expresivitate. Expresivitatea adițională este necesară pentru a putea modela anumite afirmații care nu pot fi exprimate în logica propozițională.

În logica propozițională, nu putem exprima într-un mod natural următoarea afirmație: *Orice om este muritor*.

Pentru a modela o afirmație în logica propozițională, identificăm întâi propozițiile atomice. Apoi asociem fiecărei propoziții atomice o variabilă propozițională. Propozițiile atomice sunt propozițiile care nu pot fi împărțite în alte propoziții mai mici, care să fie conectate între ele prin conectorii logici $\neg$, $\wedge$, $\vee$, $\rightarrow$ și respectiv $\leftrightarrow$.

Observăm că afirmația *Orice om este muritor* nu poate fi descompusă în afirmații indivizibile legate între ele prin conectorii logicii propoziționale, după cum este descris mai sus. Așadar, în logica propozițională, afirmația este atomică. Asociem întregii afirmații o variabilă propozițională $\mathtt{p} \in A$.

Acum să modelăm afirmația *Socrate este om.* Evident, acestei a doua afirmații trebuie să îi asociem o altă variabilă propozițională $\mathtt{q} \in A$. Să presupunem că știm că $\mathtt{p}$ și $\mathtt{q}$ sunt adevărate. Formal, știm că lucrăm cu o atribuire $\tau : A \to B$ astfel încât $\tau(\mathtt{p}) = 1$ și $\tau(\mathtt{q}) = 1$. Putem trage concluzia ca afirmația *Socrate este muritor* este adevărată în atribuirea τ?

Nu, deoarece afirmației *Socrate este muritor* ar trebui să îi asociem o a treia variabilă propozițională $\mathtt{r}$ și nu putem trage nicio concluzie asupra lui $\tau(\mathtt{r})$ din faptul că $\tau(\mathtt{p}) = 1$ și $\tau(\mathtt{q}) = 1$. Deci, din semantica logicii propoziționale, nu putem trage concluzia că $\mathtt{r}$ este adevărată în orice atribuire în care $\mathtt{p}$ și $\mathtt{q}$ sunt adevărate, în ciuda faptului că, dacă *orice om este muritor* și *Socrate este om* atunci sigur *Socrate este muritor*. Această diferență între realitate și modelarea noastră ne indică faptul că modelarea nu este suficient

de bună.

Logica de ordinul I aduce, în plus față de logica propozițională, noțiunea de *cuantificator* (existențial sau universal) și noțiunea de *predicat*. Cuantificatorul universal este notat cu $\forall$ (de la litera *A* întoarsă – *all* în limba engleză), iar cuantificatorul existențial este notat cu $\exists$ (de la litera *E* întoarsă – *exists* în limba engleză).

Un predicat este o afirmație a cărei valoare de adevăr depinde de zero sau mai mulți parametri. De exemplu, pentru afirmația de mai sus, vom folosi două predicate: $\mathtt{O}$ și $\mathtt{M}$. Predicatul $\mathtt{O}$ va fi definit astfel: $\mathtt{O}(\mathtt{x})$ va fi adevărat când $\mathtt{x}$ este om. Predicatul $\mathtt{M}(\mathtt{x})$ este adevărat când $\mathtt{x}$ este muritor. Deoarece predicatele de mai sus au fiecare câte un singur argument/parametru, ele se numesc predicate *unare*. Predicatele generalizează variabilele propoziționale prin faptul că pot primi argumente. De fapt, variabilele propoziționale pot fi văzute ca predicate fără argumente.

Astfel, afirmația *orice om este muritor* va fi modelată prin formula

$$(\forall \mathtt{x}.(\mathtt{O}(\mathtt{x}) \to \mathtt{M}(\mathtt{x}))),$$

care este citită astfel: *pentru orice* $\mathtt{x}$*, dacă* $\mathtt{O}$ *de* $\mathtt{x}$*, atunci* $\mathtt{M}$ *de* $\mathtt{x}$. Afirmația *Socrate este om* va fi modelată prin formula $\mathtt{O}(\mathtt{s})$, unde $\mathtt{s}$ este o *constantă* prin care înțelegem Socrate, la fel cum prin constanta $\mathtt{0}$ ne referim la numărul natural *zero*. De exemplu, $\mathtt{O}(\mathtt{s})$ este adevărat (deoarece $\mathtt{s}$ denotă un om), dar $\mathtt{O}(\mathtt{l})$ este fals dacă, spre exemplu, $\mathtt{l}$ este o constantă care ține locul cățelului *Lăbuș*.

Afirmația *Socrate este muritor* va fi reprezentată prin $\mathtt{M}(\mathtt{s})$ (deoarece constanta $\mathtt{s}$ se referă la Socrate). Afirmația $\mathtt{M}(\mathtt{s})$ este adevărată deoarece Socrate este muritor; la fel și afirmația $\mathtt{M}(\mathtt{l})$ este adevărată.

Vom vedea că în logica de ordinul I, formula $\mathtt{M}(\mathtt{s})$ este consecință a formulelor $(\forall \mathtt{x}.(\mathtt{O}(\mathtt{x}) \to \mathtt{M}(\mathtt{x})))$ și respectiv $\mathtt{O}(\mathtt{s})$. În acest sens, logica de ordinul I este suficient de expresivă pentru a explica din punct de vedere teoretic raționamentul prin care putem deduce că *Socrate este muritor* din faptul că *Orice om este muritor* și din faptul că *Socrate este om*.

Capitolul 2

Structuri și signaturi

Cu siguranță ați întâlnit deja mai multe formule din logica de ordinul I, fără să știți neapărat că aveți de a face cu logica de ordinul I. Fie următoare formulă:

$$\varphi = \Big(\forall \mathsf{x}.(\forall \mathsf{y}.(\mathsf{x} < \mathsf{y} \rightarrow \exists \mathsf{z}.(\mathsf{x} < \mathsf{z} \wedge \mathsf{z} < \mathsf{y})))\Big).$$

Formula folosește un simbol $<$ căruia îi corespunde un predicat binar $<$ (adică o relație binară) definit astfel: $<(x, y)$ este adevărat dacă x este mai mic strict decât y. Pentru multe predicate binare (inclusiv pentru $<$), pentru a simplifica scrierea, folosim notația infixată $(x < y)$ în loc de notația prefixată $(<(x, y))$.

Este formula φ de mai sus adevărată? Formula afirmă că între orice două valori ale variabilelor x, y există o a treia valoare, a variabilei z. Formula este adevărată dacă domeniul variabilelor $\mathsf{x}, \mathsf{y}, \mathsf{z}$ este $\mathbb{R}$, dar este falsă dacă domeniul este $\mathbb{N}$ (între orice două numere reale există un al treilea, dar între două numere naturale consecutive nu există niciun alt număr natural).

În general, formulele de ordinul I se referă la o anumită *structură matematică*.

Definiția 1 (Structură matematică)**.** *O* structură matematică *este un triplet* $S = (D, Pred, Fun)$*, unde:*

- D *este o mulțime nevidă numită* domeniu*;*
- *fiecare* $P \in Pred$ *este predicat (de o aritate oarecare) peste mulțimea* D*;*
- *fiecare* $f \in Fun$ *este funcție (de o aritate oarecare) peste mulțimea* D*.*

Iată câteva exemple de structuri matematice:

1. $(\mathbb{N}, \{<, =\}, \{+, 0, 1\})$;

   Domeniul structurii este mulțimea numerelor naturale. Structura conține două predicate: $<$ și $=$, ambele de aritate 2. Predicatul $<$ este predicatul *mai mic* pe numere naturale, iar predicatul $=$ este predicatul de *egalitate* a numerelor naturale.

   Funcția binară $+ : \mathbb{N}^2 \to \mathbb{N}$ este funcția de adunare a numerelor naturale, iar structura conține și constantele $0 \in \mathbb{N}$ și $1 \in \mathbb{N}$.

2. $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$;

   Această structură conține două predicate binare, $<$ și $=$, precum și patru funcții peste $\mathbb{R}$: funcția binară $+$, funcția unară $-$ și constantele $0, 1 \in \mathbb{R}$.

3. $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$;

   Această structură este similară cu structura precedentă, dar domeniul este mulțimea numerelor întregi.

4. $(B, \emptyset, \{\cdot, +, \bar{}\})$;

   Această structură este o algebră booleană, unde domeniul este mulțimea valorilor de adevăr, iar funcțiile sunt cele cunoscute din prima jumătate a semestrului. Astfel de structuri, fără niciun predicat, se numesc *structuri algebrice*.

5. $(\mathbb{R}, \{<\}, \emptyset)$.

   Această structură conține doar un predicat de aritate 2 (relația *mai mic* peste $\mathbb{R}$) și nicio funcție. Structurile care nu conțin funcții se numesc structuri relaționale. Structurile relaționale cu domeniul finit se mai numesc baze de date relaționale și se studiază în anul 2.

Când avem o formulă de ordinul I și dorim să îi evaluăm valoarea de adevăr, trebuie să fixăm structura în care lucrăm. Revenind la formula de mai devreme:

$$\varphi = \Big(\forall x.(\forall y.(x < y \to \exists z.(x < z \wedge z < y)))\Big),$$

avem că această formulă este adevărată în structura $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ (între orice două numere reale distincte există cel puțin un număr real) dar este falsă în structura $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$ (deoarece nu între orice două numere întregi putem găsi un alt număr întreg – de exemplu între două numere întregi consecutive nu există niciun întreg). În primul caz, domeniul variabilelor x, y, z este $\mathbb{R}$ și simbolului $<$ îi corespunde predicatul $<\subseteq \mathbb{R}^2$. În al doilea caz, domeniul variabilelor x, y, z este $\mathbb{Z}$ și simbolului $<$ îi corespunde predicatul $<\subseteq \mathbb{Z}^2$.

Este posibil ca două structuri diferite să aibă un set de predicate și de funcții cu același nume. De exemplu, chiar structurile de mai devreme, $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ și respectiv $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$. Deși predicatul $< \subseteq \mathbb{R}^2$ este diferit de predicatul $< \subseteq \mathbb{Z}^2$, ele au același nume: $<$.

În general, în Matematică și în Informatică, nu facem diferența între un predicat și numele lui, respectiv între o funcție și numele funcției, dar în Logică diferența este extrem de importantă. În particular, dacă ne referim la numele unei funcții vom folosi sintagma *simbol funcțional*, iar dacă ne referim la numele unui predicat vom folosi sintagma *simbol predicativ*. De ce este importantă diferența dintre un simbol predicativ și un predicat? Deoarece vom avea (ne)voie să asociem simbolului predicativ diverse predicate, analog modului în care unei variabile într-un limbaj de programare imperativ îi putem asocia diverse valori.

Când ne interesează doar numele funcțiilor și predicatelor (nu și funcțiile și respectiv predicatele în sine), vom utiliza signaturi:

Definiția 2 (Signatură)**.** *O* signatură Σ *este un tuplu* $\Sigma = (\mathcal{P}, \mathcal{F})$ *unde* $\mathcal{P}$ *este o mulțime de simboluri predicative și* $\mathcal{F}$ *este o mulțime de simboluri funcționale. Fiecare simbol* s *(predicativ sau funcțional) are asociat un număr natural pe care îl vom numi* aritatea *simbolului și îl vom nota cu* $ar(s)$.

Unei signaturi îi putem asocia mai multe structuri:

Definiția 3 (Σ-structuri)**.** *Dacă* $\Sigma = (\mathcal{P}, \mathcal{F})$ *este o signatură, o* Σ-structură *este orice structură* $S = (D, Pred, Fun)$ *astfel încât fiecărui simbol predicativ (sau funcțional) îi corespunde în mod unic un predicat (respectiv, o funcție).*

Exemplul 4. *Fie* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$ *unde* $\mathtt{P}, \mathtt{Q}$ *sunt simboluri predicative de aritate* $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$ *și* $\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}$ *sunt simboluri funcționale cu arități:* $ar(\mathtt{f}) = 2$, $ar(\mathtt{i}) = 1$ *și* $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$.

Avem că $(\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ *și respectiv* $(\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$ *sunt* Σ*-structuri.*

Observație. *Dupa cum se poate observa și în Exemplul 4, pentru simboluri predicative (e.g.,* $\mathtt{P}$, $\mathtt{Q}$*) vom utiliza o culoare diferită față de culoarea simbolurilor funcționale (e.g.,* $\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}$*). Pentru predicatele și funcțiile din structuri vom utiliza fontul obișnuit pentru formule matematice.*

De reținut!

Structură = domeniu + predicate + funcții

Signatură = simboluri predicative + simboluri funcționale

Unei signaturi Σ îi putem asocia mai multe structuri, numite Σ-structuri.

Notație. *Mulțimea simbolurilor predicative dintr-o Σ-structură de aritate n este notată cu $\mathcal{P}_n = \{P \mid ar(P) = n\}$, iar mulțimea simbolurilor funcționale de aritate n este notată cu $\mathcal{F}_n = \{f \mid ar(f) = n\}$. Pentru cazul particular $n = 0$, $\mathcal{F}_0$ reprezintă mulțimea simbolurilor constante (simboluri funcționale de aritate 0).*

Capitolul 3

Sintaxa logicii de ordinul I

În acest capitol vom prezenta sintaxa formulelor din logica cu predicate de ordinul I. Pentru logica de ordinul I limbajul (mulțimea de șiruri de simboluri) este determinat de alegerea unei signaturi Σ. Practic, există mai multe limbaje de ordinul I, câte un limbaj pentru fiecare signatură Σ.

În continuare, vom presupune fixată o signatură Σ cu simboluri predicative $\mathcal{P}$ și simboluri funcționale $\mathcal{F}$.

3.1 Alfabetul

Ca și formulele din logica propozițională, formulele din logica de ordinul I sunt șiruri de simboluri peste un anumit alfabet. Spre deosebire de logica propozițională, alfabetul este mai bogat și conține următoarele simboluri:

1. *conectori logici* deja cunoscuți: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \bot$, precum și doi *cuantificatori*: $\forall$, $\exists$;

2. *variabile*: vom fixa o mulțime infinit numărabilă de variabile, notată $\mathcal{X} = \{\mathsf{x}, \mathsf{y}, \mathsf{z}, \mathsf{x}', \mathsf{y}', \mathsf{x}_1, \mathsf{z}'', \ldots\}$ (a nu se confunda cu mulțimea variabilelor propoziționale din logica propozițională; sunt două noțiuni fundamental diferite și din acest motiv utilizăm altă culoare pentru a le reprezenta);

3. simboluri auxiliare: $($, $)$, $.$, $,$, $($, $)$, și $,$;

4. simboluri suplimentare, care sunt specifice fiecărei signaturi $\Sigma = (\mathcal{P}, \mathcal{F})$ în parte: simbolurile funcționale din mulțimea $\mathcal{F}$ și respectiv simbolurile predicative din mulțimea $\mathcal{P}$.

3.2 Termen

Definiția 5. *Mulţimea* termenilor, $\mathcal{T}$, *este cea mai mică mulţime care satisface următoarele proprietăți:*

1. $\mathcal{F}_0 \subseteq \mathcal{T}$ *(orice simbol constant este termen);*
2. $\mathcal{X} \subseteq \mathcal{T}$ *(orice variabilă este termen);*
3. *dacă* $f \in \mathcal{F}_n$ *(cu* $n > 0$*) şi* $t_1, \ldots, t_n \in \mathcal{T}$*, atunci* $f(t_1, \ldots, t_n) \in \mathcal{T}$ *(un simbol funcțional de aritate* n *aplicat unui număr de exact* n *termeni este termen).*

Observație. *Deoarece definiția mulțimii termenilor depinde de* $\Sigma = (\mathcal{P}, \mathcal{F})$*, elementele mulțimii* $\mathcal{T}$ *se mai numesc* Σ*-termeni.*

Practic, termenii se construiesc aplicând simboluri funcționale peste simboluri constante și variabile.

Exemplul 6. *Fie signatura* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$ *definită în Exemplul 4, unde* $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$*,* $ar(\mathtt{f}) = 2$*,* $ar(\mathtt{i}) = 1$*,* $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$*. Iată câteva exemple de termeni:* $\mathtt{a}$*,* $\mathtt{b}$*,* $\mathtt{x}$*,* $\mathtt{y}$*,* $\mathtt{x}_1$*,* $\mathtt{y}'$*,* $\mathtt{i(a)}$*,* $\mathtt{i(x)}$*,* $\mathtt{i(i(a))}$*,* $\mathtt{i(i(x))}$*,* $\mathtt{f(a,b)}$*,* $\mathtt{i(f(a,b))}$*,* $\mathtt{f(f(x,a),f(y,y))}$*.*

Exercițiul 7. *Identificați în lista de mai jos* Σ*-termenii:*

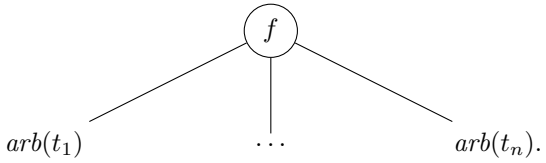
1. $\mathtt{i(i(x))}$*;*
2. $\mathtt{i}$*;*
3. $\mathtt{f(x,x)}$*;*
4. $\mathtt{P(a,b)}$*;*
5. $\mathtt{i(a,a)}$*;*
6. $\mathtt{f(i(x),i(x))}$*;*
7. $\mathtt{f(i(x,x))}$*;*
8. $\mathtt{a(i(x))}$*.*

Termenii (sau, în mod echivalent, *termii*), sunt notaţi cu t, s, t_1, t_2, s_1, t', etc. Deşi termenii sunt scrişi în mod uzual ca un şir de simboluri, ei au asociat un arbore de sintaxă abstractă definit după cum urmează:

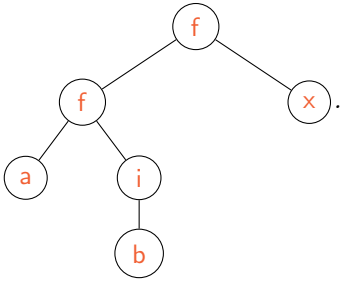
1. dacă $t = c$ și $c \in \mathcal{F}_0$, atunci $arb(t) =$ 

2. dacă $t = x$ și $x \in \mathcal{X}$, atunci $arb(t) =$ (x)

3. dacă $t = f(t_1, \ldots, t_n)$ și $f \in \mathcal{F}_n$ $(n > 0)$, $t_1, \ldots, t_n \in \mathcal{T}$, atunci

$arb(t) =$ 

Observație. *Deși formal termenii sunt definiți ca fiind șiruri de simboluri peste alfabetul descris mai sus, aceștia trebuie înțeleși ca fiind arbori. De altfel, în orice software care lucrează cu termeni, aceștia sunt memorați sub formă de arbori cu rădăcină. Iată arborele atașat termenului* f(f(a, i(b)), x)*:*

$arb($f(f(a, i(b)), x)$) =$ 

Exercițiul 8. *Calculați arborii de sintaxă pentru termenii din Exemplul 6.*

3.3 Formule atomice

Definiția 9 (Formulă atomică)**.** *O* formulă atomică *este orice șir de simboluri de forma* $P(t_1, \ldots, t_n)$*, unde* $P \in \mathcal{P}_n$ *este un simbol predicativ de aritate* $n \geq 0$*, iar* $t_1, \ldots, t_n \in \mathcal{T}$ *sunt termeni. Dacă* $n = 0$*, scriem* P *în loc de* $P()$*.*

Exemplul 10. *Continuând Exemplul 6, folosim signatura*

$$\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\}),$$

unde $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$*,* $ar(\mathtt{f}) = 2$*,* $ar(\mathtt{i}) = 1$*,* $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$*.*

Iată câteva exemple de formule atomice: P(a,b)*,* P(x, y)*,* Q(i(i(x)), f(x,x))*,* Q(a, b)*,* P(f(f(a, i(x)), b), i(x))*.*

Exercițiul 11. *Explicați de ce* P(a)*,* P*,* i(i(x)) *nu sunt formule atomice peste signatura din Exemplul 10.*

3.4 Formule de ordinul I

Definiția 12 (Formule de ordinul I)**.** *Mulţimea formulelor de ordinul I, notată* $\mathbb{LPI}$*, este cea mai mică mulţime astfel incât:*

1. *(cazul de bază) orice formulă atomică este formulă (adică* $P(t_1, \ldots, t_n) \in \mathbb{LPI}$ *pentru orice simbol predicativ* $P \in \mathcal{P}_n$ *și orice termeni* $t_1, \ldots, t_n \in \mathcal{T}$*;*
2. *(cazurile inductive) pentru orice formule* $\varphi, \varphi_1, \varphi_2 \in \mathbb{LPI}$*, pentru orice variabilă* $x \in \mathcal{X}$*, avem că:*
   (a) $\neg\varphi_1 \in \mathbb{LPI}$*;*
   (b) $(\varphi_1 \wedge \varphi_2) \in \mathbb{LPI}$*;*
   (c) $(\varphi_1 \vee \varphi_2) \in \mathbb{LPI}$*;*
   (d) $(\varphi_1 \to \varphi_2) \in \mathbb{LPI}$*;*
   (e) $(\varphi_1 \leftrightarrow \varphi_2) \in \mathbb{LPI}$*;*
   (f) $(\forall x.\varphi) \in \mathbb{LPI}$*;*
   (g) $(\exists x.\varphi) \in \mathbb{LPI}$*.*

Observație. *În Definiția 12, regăsim conectorii logici* $\neg, \wedge, \vee, \to$ *și respectiv* $\leftrightarrow$ *din logica propozițională. Locul variabilelor propoziționale (deocamdată, la nivel sintactic) este luat de simbolurile predicative de aritate* 0*. Construcțiile* $(\forall x.\varphi)$ *și* $(\exists x.\varphi)$ *sunt noi.*

Exemplul 13. *Continuând Exemplul 6, folosim signatura* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$*, unde* $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$*,* $ar(\mathtt{f}) = 2$*,* $ar(\mathtt{i}) = 1$ *și* $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$*.*
Iată câteva exemple de formule din logica de ordinul I:

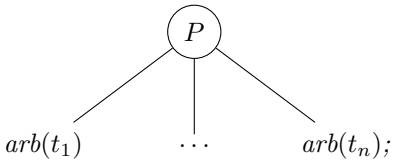
1. $\mathtt{P(a,b)}$*;*
2. $\mathtt{Q(a,b)}$*;*
3. $\mathtt{P(a,x)}$*;*
4. $\neg\mathtt{P(a,b)}$*;*
5. $(\mathtt{P(a,b)} \wedge \neg\mathtt{Q(a,b)})$*;*
6. $(\mathtt{P(a,b)} \vee \neg\mathtt{Q(x,y)})$*;*
7. $(\mathtt{P(a,b)} \to \mathtt{P(a,b)})$*;*
8. $((\mathtt{P(a,b)} \to \mathtt{P(a,b)}) \leftrightarrow (\mathtt{P(a,b)} \to \mathtt{P(a,b)}))$*;*

9. $(\forall x.P(a,x))$;

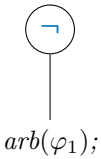
10. $(\exists x.\neg Q(x,y))$.

Definiția 14 (Arborele de sintaxă abstractă asociat formulelor din $\mathbb{LPI}$)**.** *Formulele au asociat un arbore de sintaxă abstractă definit în cele ce urmează:*

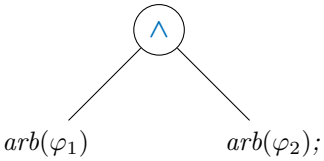
1. *dacă* $\varphi = P(t_1,\ldots,t_n)$, *atunci* $arb(\varphi) =$



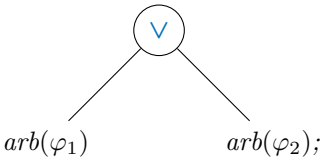
2. *dacă* $\varphi = \neg\varphi_1$, *atunci* $arb(\varphi) =$



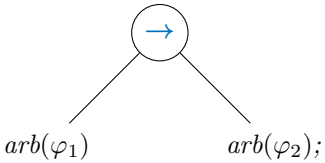
3. *dacă* $\varphi = (\varphi_1 \wedge \varphi_2)$, *atunci* $arb(\varphi) =$



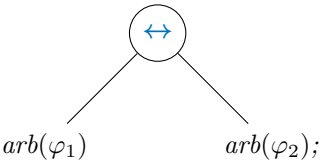
4. *dacă* $\varphi = (\varphi_1 \vee \varphi_2)$, *atunci* $arb(\varphi) =$

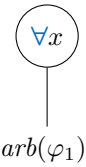


5. *dacă* $\varphi = (\varphi_1 \to \varphi_2)$, *atunci* $arb(\varphi) =$



6. *dacă* $\varphi = (\varphi_1 \leftrightarrow \varphi_2)$, *atunci* $arb(\varphi) =$



7. *dacă* $\varphi = (\forall x.\varphi_1)$, *atunci* $arb(\varphi) =$ 

8. *dacă* $\varphi = (\exists x.\varphi_1)$, *atunci* $arb(\varphi) =$ 

Exercițiul 15. *Calculați arborele de sintaxă asociat formulelor din Exemplul 13.*

3.5 Paranteze

Parantezele (și) sunt folosite pentru a marca ordinea efectuării operațiilor logice (și, sau, not, etc.) În continuare, vom renunța la parantezele care nu sunt necesare, la fel ca în cazul logicii propoziționale: dacă o formulă poate fi interpretată ca un arbore de sintaxă abstractă în două sau mai multe moduri, se vor folosi paranteze pentru a stabili arborele dorit.

De exemplu, $\varphi_1 \vee \varphi_2 \wedge \varphi_3$ ar putea fi înţeleasă ca $((\varphi_1 \vee \varphi_2) \wedge \varphi_3)$ sau ca $(\varphi_1 \vee (\varphi_2 \wedge \varphi_3))$. Pentru a evita scrierea formulelor cu prea multe paranteze, se stabilesc *priorităţi*. Ordinea priorității operatorilor este:

$$\bot, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists,$$

unde $\neg$ este cel mai prioritar, iar $\exists$ este cel mai puțin prioritar. Pentru a evita situațiile de ambiguitate (i.e., atunci când există mai multe moduri de a construi arborele de sintaxă), este recomandată folosirea de paranteze suplimentare.

Datorită priorității conectorilor logici, formula $\mathtt{P(a,a)} \vee \mathtt{P(b,b)} \wedge \mathtt{P(a,b)}$ va fi întotdeauna înţeleasă ca $(\mathtt{P(a,a)} \vee (\mathtt{P(b,b)} \wedge \mathtt{P(a,c)}))$ (deoarece $\wedge$ este prioritar faţă de $\vee$). Ca analogie, la fel se întâmplă şi în cazul aritmeticii: $1 + 2 * 3$ va fi înteles ca $1 + (2 \times 3)$, deoarece $\times$ are prioritate în faţa lui $+$ ($\times$ este similar cu $\wedge$ şi $+$ cu $\vee$).

Exercițiul 16. *Scrieți formulele din Exemplul 13 cu cât mai puține paranteze.*

În Secțiunea 3.12 vom detalia modul în care cuantificatorii interacționează cu ceilalți conectori logici. Vom vedea că pentru cuantificatori avem niște reguli suplimentare în afară de priorități.

3.6 Modelarea în $\mathbb{LPI}$ a afirmațiilor din limba română

În această secțiune vom explica care este signatura folosită pentru a modela în logica de ordinul întâi afirmațiile: *Orice om este muritor*, *Socrate este om* și respectiv *Socrate este muritor*.

În primul rând, identificăm predicatele din text. Avem două predicate unare *este om* și respectiv *este muritor*. Alegem simbolul predicativ $\mathtt{O}$ pentru primul predicat și simbolul predicativ $\mathtt{M}$ pentru al doilea predicat. De asemenea, în text avem și o constantă: Socrate. Alegem simbolul funcțional $\mathtt{s}$ de aritate 0 pentru această constantă. Așadar, pentru a modela afirmațiile de mai sus, vom lucra cu signatura

$$\Sigma = (\{\mathtt{O}, \mathtt{M}\}, \{\mathtt{s}\}),$$

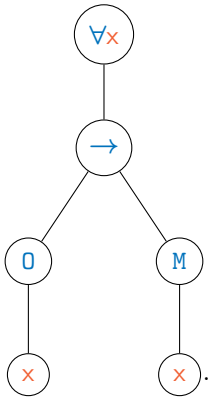
unde $\mathtt{O}$ și $\mathtt{M}$ sunt simboluri predicative de aritate $ar(\mathtt{O}) = ar(\mathtt{M}) = 1$, iar $\mathtt{s}$ este un simbol funcțional de aritate $ar(\mathtt{s}) = 0$, adică un simbol constant.

Afirmația *orice om este muritor* va fi modelată prin formula de ordinul I

$$(\forall \mathtt{x}.(\mathtt{O}(\mathtt{x}) \to \mathtt{M}(\mathtt{x}))),$$

al cărei arbore de sintaxă abstractă este:

$arb\Big((\forall \mathtt{x}.(\mathtt{O}(\mathtt{x}) \to \mathtt{M}(\mathtt{x})))\Big) =$



.

Afirmația *Socrate este om* o vom modela prin formula atomică $\mathtt{O}(\mathtt{s})$, iar afirmația *Socrate este muritor* o vom modela prin formula atomică $\mathtt{M}(\mathtt{s})$.

Pentru signatura $\Sigma = (\{\mathtt{O}, \mathtt{M}\}, \{\mathtt{s}\})$ stabilită mai sus, există mai multe Σ-structuri posibile. Un exemplu este structura $S = (D, \{\mathtt{O}^S, \mathtt{M}^S\}, \{\mathtt{s}^S\})$ definită astfel:

1. D este mulțimea tuturor ființelor de pe Pământ;

2. $\mathtt{O}^S(x)$ este adevărat pentru orice ființă x care este și om;
3. $\mathtt{M}^S(x)$ este adevărat pentru orice ființă x (toate elementele domeniului sunt muritoare);
4. $\mathtt{s}^S$ este Socrate (Socrate, fiind o ființă, aparține mulțimii D).

Anticipând puțin (vom discuta despre semantica formulelor de ordinul I în Capitolul 4), toate cele trei formule discutate în această secțiune, adică $(\forall \mathtt{x}.(\mathtt{O}(\mathtt{x}) \to \mathtt{M}(\mathtt{x})))$, $\mathtt{O}(\mathtt{s})$ și respectiv $\mathtt{M}(\mathtt{s})$, sunt adevărate în structura S definită mai sus. De fapt, calitatea raționamentului *orice om este muritor; Socrate este om; deci: Socrate este muritor* este dată de faptul că formula $\mathtt{M}(\mathtt{s})$ este în mod necesar adevărată în *orice* structură în care formulele $\mathtt{O}(\mathtt{s})$ și $(\forall \mathtt{x}.(\mathtt{O}(\mathtt{x}) \to \mathtt{M}(\mathtt{x})))$ sunt adevărate, nu doar în structura S de mai sus.

3.7 Modelarea în $\mathbb{LPI}$ a afirmațiilor despre aritmetică

Fie signatura $\Sigma = (\{<, =\}, \{+, -, 0, 1\})$, unde $<$ și $=$ sunt simboluri predicative de aritate 2, $+$ este simbol funcțional de aritate 2, $-$ este simbol funcțional de aritate 1, iar 0 și 1 sunt simboluri constante. Iată câteva formule care fac parte din limbajul de ordinul I asociat signaturii Σ:

1. $\Big(\forall \mathtt{x}.\big(\forall \mathtt{y}.\big(<(\mathtt{x}, \mathtt{y}) \to \exists \mathtt{z}.(<(\mathtt{x}, \mathtt{z}) \wedge <(\mathtt{z}, \mathtt{y}))\big)\big)\Big)$;
2. $\Big(\forall \mathtt{x}.\big(\forall \mathtt{y}.\big(\exists \mathtt{z}.(=(+(\mathtt{x}, \mathtt{y}), \mathtt{z}))\big)\big)\Big)$;
3. $\Big(\forall \mathtt{x}.(<(0, \mathtt{x}) \vee =(0, \mathtt{x}))\Big)$;
4. $\Big(\forall \mathtt{x}.\big(\exists \mathtt{y}.(=(\mathtt{x}, -(\mathtt{y})))\big)\Big)$;
5. $=(+(\mathtt{x}, \mathtt{y}), \mathtt{z})$.

De multe ori, în cazul simbolurilor predicative și simbolurilor funcționale binare, se folosește notația infixată (e.g., $\mathtt{x}<\mathtt{y}$ în loc de $<(\mathtt{x}, \mathtt{y})$). În acest caz, putem scrie formulele de mai sus în felul următor:

1. $\Big(\forall \mathtt{x}.\big(\forall \mathtt{y}.\big(\mathtt{x} < \mathtt{y} \to \exists \mathtt{z}.(\mathtt{x} < \mathtt{z} \wedge \mathtt{z} < \mathtt{y})\big)\big)\Big)$;
2. $\Big(\forall \mathtt{x}.\big(\forall \mathtt{y}.\big(\exists \mathtt{z}.(\mathtt{x} + \mathtt{y} = \mathtt{z})\big)\big)\Big)$;
3. $\Big(\forall \mathtt{x}.(0 < \mathtt{x} \vee 0 = \mathtt{x})\Big)$;

4. $\Big(\forall x.\big(\exists y.(x = -(y))\big)\Big)$;
5. $x + y = z$.

Două dintre Σ-structurile posibile sunt $S_1 = (\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$ și $S_2 = (\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$, unde predicatele și funcțiile sunt cele cunoscute de la matematică (cu precizarea că $-$ este funcția minus unar).

Anticipând cursul următor referitor la semantica formulelor de ordinul I, prima formulă este falsă în S_2 și adevărată în S_1. A doua formulă și a patra formulă sunt adevărate atât în S_1 cât și în S_2. A treia formula este falsă atât în S_1 cât și în S_2. Valoarea de adevăr a celei de-a cincea formule nu depinde doar de structura în care evaluăm formula, ci și de valorile variabilelor x, y, z. Deoarece variabilele x, y, z nu apar cuantificate în formula numărul 5, acestea se numesc *libere*. Formula 5 este *satisfiabilă* atât în structura S_1 cât și în structura S_2, deoarece în ambele cazuri există valori pentru variabilele x, y, z care să facă formula adevărată (e.g. valorile $1, 2, 3$ pentru x, y și respectiv z).

3.8 Variabilele unei formule

Cu $vars(\varphi)$ notăm variabilele care apar în formula φ. De exemplu, vom avea că $vars\big((\forall z.(P(x, y)))\big) = \{x, y, z\}$. Definim funcția $vars : \mathbb{LPI} \to 2^{\mathcal{X}}$ în cele ce urmează.

În primul rând, definim o funcție $vars : \mathcal{T} \to 2^{\mathcal{X}}$ (atenție, domeniul este $\mathcal{T}$) ca fiind funcția care asociază unui termen (din mulțimea $\mathcal{T}$) mulțimea *variabilelor care apar* în acel termen. Toate definițiile care urmează vor fi definiții inductive, care oglindesc definițiile sintactice corespunzătoare. Le vom denumi simplu *definiții recursive*, fără a mai preciza explicit cazurile de bază sau pe cele inductive. Amintim că $2^{\mathcal{X}}$ denotă mulțimea tuturor submulțimilor lui $\mathcal{X}$. Totodată, amintim că pentru o signatură fixată Σ, notăm cu $\mathcal{P}_n$ mulțimea simbolurilor predicative de aritate n din Σ, iar cu $\mathcal{F}_n$ mulțimea simbolurilor funcționale de aritate n din Σ.

Definiția 17. *Funcția $vars : \mathcal{T} \to 2^{\mathcal{X}}$ este definită recursiv după cum urmează:*

1. *$vars(c) = \emptyset$, dacă $c \in \mathcal{F}_0$ este un simbol constant;*
2. *$vars(x) = \{x\}$, dacă $x \in \mathcal{X}$ este o variabilă;*
3. *$vars(f(t_1, \ldots, t_n)) = \bigcup_{i \in \{1,\ldots,n\}} vars(t_i)$.*

Putem acum defini (inductiv, prin imbricare) funcția extinsă și notată omonim, $vars : \mathbb{LPI} \to 2^{\mathcal{X}}$, care asociază unei formule din $\mathbb{LPI}$ mulțimea de variabile ale formulei (adică, variabilele care apar în formulă):

Definiția 18. *Funcția vars* : $\mathbb{LPI} \to 2^{\mathcal{X}}$ *este definită recursiv după cum urmează:*

1. $vars(P(t_1, \ldots, t_n)) = \bigcup_{i \in \{1,\ldots,n\}} vars(t_i)$;
2. $vars(\neg\varphi) = vars(\varphi)$;
3. $vars((\varphi_1 \wedge \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
4. $vars((\varphi_1 \vee \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
5. $vars((\varphi_1 \to \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
6. $vars((\varphi_1 \leftrightarrow \varphi_2)) = vars(\varphi_1) \cup vars(\varphi_2)$;
7. $vars\big((\forall x.\varphi)\big) = vars(\varphi) \cup \{x\}$;
8. $vars\big((\exists x.\varphi)\big) = vars(\varphi) \cup \{x\}$.

Să observăm că variabila x este adăugată corespunzător în mulțimea de variabile care se construiește chiar dacă ea apare doar imediat după simbolurile $\exists$ sau $\forall$.

Exemplul 19. *Fie formula* φ:

$$\Big(\big(\forall \mathtt{x}.(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \exists \mathtt{y}.(\mathtt{P}(\mathtt{z},\mathtt{f}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{y})))\big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big).$$

Avem $vars(\varphi) = \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$.

Exercițiul 20. *Fie signatura* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{a}, \mathtt{b}\})$, *unde* $ar(\mathtt{P}) = ar(\mathtt{Q}) = 2$, $ar(\mathtt{f}) = 2$, $ar(\mathtt{i}) = 1$, $ar(\mathtt{a}) = ar(\mathtt{b}) = 0$. *Calculați* $vars(\varphi)$ *pentru fiecare formulă* φ *de mai jos:*

1. $\mathtt{P}(\mathtt{x},\mathtt{y})$;
2. $\mathtt{Q}(\mathtt{a},\mathtt{b})$;
3. $\mathtt{P}(\mathtt{a},\mathtt{x})$;
4. $\neg\mathtt{P}(\mathtt{x},\mathtt{z})$;
5. $(\mathtt{P}(\mathtt{x},\mathtt{x}) \wedge \neg\mathtt{Q}(\mathtt{x},\mathtt{z}))$;
6. $(\mathtt{P}(\mathtt{x},\mathtt{b}) \vee \neg\mathtt{Q}(\mathtt{z},\mathtt{y}))$;
7. $((\mathtt{P}(\mathtt{x},\mathtt{b}) \to \mathtt{P}(\mathtt{x},\mathtt{z})) \leftrightarrow (\mathtt{P}(\mathtt{x},\mathtt{b}) \to \mathtt{P}(\mathtt{a},\mathtt{z})))$;
8. $(\forall \mathtt{x}.\mathtt{P}(\mathtt{a},\mathtt{x}))$;
9. $(\exists \mathtt{x}.\neg\mathtt{Q}(\mathtt{x},\mathtt{y}))$;
10. $\Big((\exists \mathtt{x}.\neg\mathtt{Q}(\mathtt{x},\mathtt{y})) \wedge (\forall \mathtt{y}.\mathtt{P}(\mathtt{y},\mathtt{x}))\Big)$.

3.9 Domeniul de vizibilitate al unui cuantificator - analogie cu limbajele de programare

Într-un limbaj de programare, putem declara mai multe variabile cu același nume. De exemplu, în C, putem avea următorul cod:

```
/*  1:*/ int f()
/*  2:*/ {
/*  3:*/   int s = 0;
/*  4:*/   for (int x = 1; x <= 10; ++x) {
/*  5:*/     for (int y = 1; y <= 10; ++y) {
/*  6:*/       s += x * y * z;
/*  7:*/       for (int x = 1; x <= 10; ++x) {
/*  8:*/         s += x * y * z;
/*  9:*/       }
/* 10:*/     }
/* 11:*/   }
/* 12:*/   return s;
/* 13:*/ }
```

În acest fragment de cod, sunt declarate trei variabile, două dintre variabile având același nume, și anume `x`. Domeniul de vizibilitate al variabilei `x` declarate la linia 4 este dat de liniile $4-11$, iar domeniul de vizibilitate al variabile `x` declarată la linia 7 este dat de liniile $7-9$. Astfel, orice apariție a numelui `x` între liniile $7-9$ se referă la cea de-a doua declarație a variabilei, în timp ce orice apariție a numelui `x` între liniile $4-11$ (cu excepția liniilor $7-9$) se referă la prima declarație a lui `x`. De exemplu, apariția lui `x` de la linia 6 se referă la variabila `x` declarată la linia 4. Apariția lui `x` de la linia 8 se referă la variabila `x` declarată la linia 7.

Liniile $4-11$ reprezintă domeniul de vizibilitate al primei declarații a variabilei `x`, iar liniile $7-9$ reprezintă domeniul de vizibilitate al celei de-a două declarații a variabilei `x`.

Un fenomen similar se întâmplă în formulele logicii de ordinul I. De exemplu, în formula $\Big(\forall x.\big(\forall y.(P(x,y) \wedge P(x,z) \wedge (\exists x.P(x,y)))\big)\Big)$, variabila x este cuantificată de două ori (prima dată universal, a doua oară existențial). O cuantificare a unei variabile se numește *legare* (engl. *binding*), din motive istorice. O *legare* este similară, din punctul de vedere al domeniului de vizibilitate, cu definirea unei variabile într-un limbaj de programare.

Astfel, domeniul de vizibilitate D_1 al variabilei x cuantificate universal este $\big(\forall y.(P(x,y) \wedge P(x,z) \wedge (\exists x.P(x,y)))\big)$, în timp ce domeniul de vizibilitate D_2 al variabilei x cuantificate existențial este $P(x,y)$:

$$\Big(\forall x.\ \underbrace{\big(\forall y.\big(P(x,y) \wedge P(x,z) \wedge (\exists x.\ \overbrace{P(x,y)}^{D_2})\big)\big)}_{D_1}\Big)$$

Aparițiile unei variabile cuantificate care sunt prezente în domeniul de vizibilitate al acesteia se numesc *legate*, în timp ce aparițiile din afara domeniului de vizibilitate se numesc *libere*.

3.10 Apariții libere și legate ale variabilelor

În această secțiune vom stabili formal conceptul de apariție/variabilă legată și de apariție/variabilă liberă. Aparițiile libere/legate ale unei variabile în logica de ordinul I sunt, ca o analogie, similare cu variabilele globale/locale într-un limbaj de programare.

Mai departe vom utiliza noțiunea de *arbore de sintaxă abstractă* pentru formulele din logica de ordinul I.

Definiția 21 (Apariţie liberă)**.** *O* apariţie liberă a unei variabile x într-o formulă φ *este dată de un nod în arborele formulei etichetat cu* x *și care are proprietatea că, mergând din nod înspre rădăcină, nu întâlnim niciun nod etichetat cu* $\forall x$ *sau cu* $\exists x$*.*

Definiția 22 (Apariţie legată)**.** *O* apariţie legată a unei variabile x într-o formulă φ *este dată de un nod în arborele formulei etichetat cu* x *și care are proprietatea că, mergând din nod înspre rădăcină, întâlnim măcar un nod etichetat cu* $\forall x$ *sau cu* $\exists x$*.*

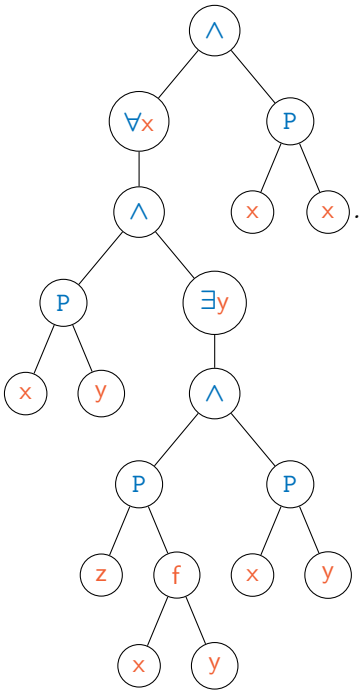
Cel mai apropiat astfel de nod etichetat cu $\forall x$ *sau cu* $\exists x$ *este cuantificarea care* leagă *apariția în cauză a variabilei* x*.*

Exemplul 23. *Considerăm în continuare formula*

$$\varphi = \Big(\Big(\forall x.\big(P(x,y) \wedge \exists y.\big(P(z,f(x,y)) \wedge P(x,y)\big)\big)\Big) \wedge P(x,x)\Big).$$

Arborele de sintaxă abstractă al formulei φ *este:*

$arb(\varphi) =$



În formula φ de mai sus, variabila $\mathtt{x}$ are două apariții libere. Variabila $\mathtt{y}$ are o apariție liberă. Variabila $\mathtt{z}$ are o apariție liberă. Toate aparițiile libere ale variabilelor în formula φ sunt marcate prin subliniere:

$$\varphi = \Big(\big(\forall \mathtt{x}.\big(\mathtt{P}(\mathtt{x},\underline{\mathtt{y}}) \wedge \exists \mathtt{y}.\big(\mathtt{P}(\underline{\mathtt{z}},\mathtt{f}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{y})\big)\big)\big) \wedge \mathtt{P}(\underline{\mathtt{x}},\underline{\mathtt{x}})\Big).$$

Toate aparițiile legate ale variabilelor în formula φ sunt marcate prin dublă subliniere în următoarea formulă:

$$\varphi = \Big(\big(\forall \mathtt{x}.\big(\mathtt{P}(\underline{\underline{\mathtt{x}}},\mathtt{y}) \wedge \exists \mathtt{y}.\big(\mathtt{P}(\mathtt{z},\mathtt{f}(\underline{\underline{\mathtt{x}}},\underline{\underline{\mathtt{y}}})) \wedge \mathtt{P}(\underline{\underline{\mathtt{x}}},\underline{\underline{\mathtt{y}}})\big)\big)\big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big).$$

Observație. *În restul capitolelor vom face o distincție clară între aparițiile (libere sau legate ale) variabilelor într-o formulă și mulțimile variabilelor de acest tip (libere sau legate). Nodurile etichetate cu $\forall x$ și respectiv $\exists x$ nu vor fi considerate ca desemnând nici o apariție liberă, nici o apariție legată a lui x, ci ca fiind simple noduri prin care se fixează/denumește un cuantificator (sau, prin care se leagă variabila x).*

3.11 Variabile libere și variabile legate

Mulţimea variabilelor unei formule φ care au cel puţin o apariţie liberă se notează $free(\varphi)$.

Definiția 24. *Funcția* $free : \mathbb{LPI} \to 2^{\mathcal{X}}$ *este definită recusiv în modul următor:*

1. $free(P(t_1, \ldots, t_n)) = vars(t_1) \cup \ldots \cup vars(t_n)$;
2. $free(\neg\varphi) = free(\varphi)$;
3. $free((\varphi_1 \wedge \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
4. $free((\varphi_1 \vee \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
5. $free((\varphi_1 \to \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
6. $free((\varphi_1 \leftrightarrow \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
7. $free\big((\forall x.\varphi)\big) = free(\varphi) \setminus \{x\}$;
8. $free\big((\exists x.\varphi)\big) = free(\varphi) \setminus \{x\}$.

Exemplul 25. *Pentru formula*

$$\varphi = \Big(\big(\forall \mathtt{x}.\big(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \exists \mathtt{y}.(\mathtt{P}(\mathtt{z},\mathtt{f}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{y}))\big)\big) \wedge \mathtt{P}(\mathtt{x},\mathtt{x})\Big).$$

avem că $free(\varphi) = \{\mathtt{x}, \mathtt{y}, \mathtt{z}\}$.

Exercițiul 26. *Calculați* $free(\varphi)$ *pentru fiecare formulă* φ *din Exercițiul 20.*

Cu $bound(\varphi)$ notăm mulțimea variabilelor legate într-o formulă, cu alte cuvinte mulțimea acelor variabile x cu proprietatea că există în formulă cel puțin un nod etichetat cu $\forall x$ sau cu $\exists x$.

Definiția 27. *Funcția* $bound : \mathbb{LPI} \to 2^{\mathcal{X}}$ *este definită recursiv astfel:*

1. $bound(P(t_1, \ldots, t_n)) = \emptyset$;
2. $bound(\neg\varphi) = bound(\varphi)$;
3. $bound((\varphi_1 \wedge \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;
4. $bound((\varphi_1 \vee \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;
5. $bound((\varphi_1 \to \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;
6. $bound((\varphi_1 \leftrightarrow \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;

7. $bound((\forall x.\varphi)) = bound(\varphi) \cup \{x\}$;
8. $bound((\exists x.\varphi)) = bound(\varphi) \cup \{x\}$.

Exercițiul 28. *Calculați $bound(\varphi)$ pentru fiecare formulă φ din Exercițiul 20.*

Exercițiul 29. *Calculați $bound(\varphi)$, unde*

$$\varphi = \Big(\big(\forall x.\big(P(x,y) \wedge \exists y.\big(P(z,f(x,y)) \wedge P(x,y)\big)\big)\big) \wedge P(x,x)\Big).$$

Definiția 30. *Variabilele legate ale unei formule φ sunt elementele mulțimii $bound(\varphi)$.*

Definiția 31. *Variabilele libere ale unei formule φ sunt elementele mulțimii $free(\varphi)$.*

Observație. *Mulțimile $free(\varphi)$ şi $bound(\varphi)$ pot avea elemente în comun.*

Observație. *O variabilă poate avea mai multe apariții într-o formulă.*

După cum am precizat anterior, trebuie făcută diferența între o apariție liberă a unei variabile într-o formulă *și o* variabilă liberă a unei formule. *Apariția liberă este indicată printr-un un nod din arborele formulei, în timp ce variabila liberă este un element al mulțimii $\mathcal{X}$.*

Similar, trebuie făcută diferența între o apariție legată a unei variabile într-o formulă *și o* variabilă legată a unei formule. *Apariția legată este indicată de un nod în arborele formulei, în timp ce variabila este un element al mulțimii $\mathcal{X}$.*

3.12 Domeniul de vizibilitate și parantetizarea formulelor

Acum că am înțeles ce este domeniul de vizibilitate a unei variabile legate (apelând și la arborele formulei), putem clarifica un aspect referitor la ordinea de prioritate a conectorilor logici (dacă privim formula doar ca text/cuvânt). Am stabilit deja că ordinea de prioritate este $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \forall, \exists$, dar cuantificatorii $\forall$ și $\exists$ interacționează într-un mod mai subtil cu ceilalți conectori logici. Mai precis, textual, o formulă fără paranteze se (re)parantetizează astfel încât domeniul de vizibilitate al fiecărui cuantificator să se extindă *cât mai mult spre dreapta*.

De exemplu, formula:

$$\forall x.P(x,x) \vee \neg\exists y.P(x,y) \wedge P(x,x)$$

se parantezează în felul următor:

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

3.13 Fișă de exerciții

Exercițiul 32. *Identificați o signatură pentru afirmațiile de mai jos și apoi modelați aceste afirmații ca formule în logica de ordinul I:*

Ion este student. Orice student învață la Logică. Oricine învață la Logică trece examenul. Orice student este om. Există un om care nu a trecut examenul. Deci nu toți oamenii sunt studenți.

Exercițiul 33. *Fie structura* $S = (\mathbb{R}, \{Nat, Int, Prim, Par, >\}, \{+, 0, 1, 2\})$, *unde Nat, Int, Prim, Par sunt predicate unare cu următoarea semnificație:*

- $Nat(u)$ = u este număr natural*;*
- $Int(u)$ = u este număr întreg*;*
- $Prim(u)$ = u este număr prim*;*
- $Par(u)$ = u este număr par.

Predicatul binar $>$ *este relația "mai mare" peste numere reale. Funcția* $+$ *este funcția de adunare a numerelor reale. Constantele* $0, 1, 2$ *sunt chiar numerele* $0, 1, 2$.

1. *Propuneți o signatură* Σ *pentru structura* S *de mai sus.*
2. *Modelați următoarele afirmații ca formule de ordinul I în signatura asociată structurii* S *de mai sus:*
   (a) *Orice număr natural este și număr întreg.*
   (b) *Suma oricăror două numere naturale este număr natural.*
   (c) *Oricum am alege un număr natural, există un număr prim care este mai mare decât numărul respectiv.*
   (d) *Dacă orice număr natural este număr prim, atunci zero este număr prim.*
   (e) *Oricum am alege un număr prim, există un număr prim mai mare decât el.*
   (f) *Suma a două numere pare este un număr par.*

(g) *Orice număr prim mai mare decât* 2 *este impar.*

(h) *Orice număr prim poate fi scris ca suma a patru numere prime.*

(i) *Suma a două numere pare este un număr impar.*

Exercițiul 34. *Dați exemplu de* 5 *termeni peste signaturile de la Exercițiul 33 și calculați arborele de sintaxă abstractă al acestor termeni.*

Exercițiul 35. *Dați exemplu de* 5 *formule peste signatura de la Exercițiul 33 și calculați arborele de sintaxă abstractă al acestora.*

Exercițiul 36. *Calculați arborele e sintaxă abstractă al următoarelor formule (indicație: puneți paranteze în jurul formulelor, în ordinea de prioritate a conectorilor):*

1. $\mathtt{P}(\mathtt{x}) \vee \mathtt{P}(\mathtt{y}) \wedge \neg \mathtt{P}(\mathtt{z})$;
2. $\neg\neg \mathtt{P}(\mathtt{x}) \vee \mathtt{P}(\mathtt{y}) \rightarrow \mathtt{P}(\mathtt{x}) \wedge \neg \mathtt{P}(\mathtt{z})$;
3. $\forall \mathtt{x}.\forall \mathtt{y}.\neg\neg \mathtt{P}(\mathtt{x}) \vee \mathtt{P}(\mathtt{y}) \rightarrow \mathtt{P}(\mathtt{x}) \wedge \neg \mathtt{P}(\mathtt{z})$;
4. $\forall \mathtt{x}.\forall \mathtt{y}.\neg\neg \mathtt{P}(\mathtt{x}) \vee \mathtt{P}(\mathtt{y}) \rightarrow \exists \mathtt{x}.\mathtt{P}(\mathtt{x}) \wedge \neg \mathtt{P}(\mathtt{x})$;
5. $\forall \mathtt{x}'.\neg\forall \mathtt{x}.\mathtt{P}(\mathtt{x}) \wedge \exists \mathtt{y}.\mathtt{Q}(\mathtt{x},\mathtt{y}) \vee \neg \mathtt{Q}(\mathtt{z},\mathtt{z}) \rightarrow \exists \mathtt{z}'.\mathtt{P}(\mathtt{z}')$.

Exercițiul 37. *Marcați aparițiile libere și respectiv aparițiile legate ale variabilelor în formulele de mai jos:*

1. $\varphi_1 = (\forall \mathtt{x}.\mathtt{P}(\mathtt{x},\mathtt{x}) \wedge \mathtt{P}(\mathtt{x},\mathtt{y})) \wedge \mathtt{P}(\mathtt{x},\mathtt{z})$;
2. $\varphi_2 = (\forall \mathtt{x}.\mathtt{P}(\mathtt{f}(\mathtt{x},\mathtt{x}),\mathtt{i}(\mathtt{x})) \wedge \exists \mathtt{y}.(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge \mathtt{P}(\mathtt{x},\mathtt{z})))$.

Exercițiul 38. *Identificați domeniul de vizibilitate, pentru fiecare cuantificator, în formulele φ_1 și φ_2 date în Exercițiul 37.*

Exercițiul 39. *Găsiți variabilele, variabilele libere și respectiv variabilele legate ale formulelor φ_1 și φ_2 date în Exercițiul 37.*

Exercițiul 40. *Fie $A = \{\mathtt{p},\mathtt{q},\mathtt{r},\ldots\}$ mulțimea variabilelor propoziționale. Vom considera signatura $\Sigma_{\mathbb{LP}} = (A,\emptyset)$, unde variabilele propoziționale din A sunt simboluri predicative de aritate* 0.

1. *Demonstrați că pentru orice formulă $\varphi \in \mathbb{LP}$ avem $\varphi \in \mathbb{LPI}$ (peste signatura $\Sigma_{\mathbb{LP}}$).*
2. *Demonstrați că pentru orice formulă fără cuantificatori $\varphi \in \mathbb{LPI}$ peste signatura $\Sigma_{\mathbb{LP}}$, avem că $\varphi \in \mathbb{LP}$.*

Capitolul 4

Semantica formulelor logicii de ordinul I

Sintaxa logicii de ordinul I explică care sunt, din punct de vedere sintactic, formulele logicii de ordinul I. Semantica logicii de ordinul I se referă la *înțelesul* formulelor. Semantica unei formule (sau înțelesul formulei) va fi o valoare de adevăr. Ca și la logica propozițională, în general, valoarea de adevăr a unei formule depinde nu doar de formulă, ci și de *structura* în care evaluăm formula.

Reamintim că o signatură $\Sigma = (\mathcal{P}, \mathcal{F})$ este o pereche formată dintr-o mulțime de simboluri predicative $\mathcal{P}$ și o mulțime de simboluri funcționale $\mathcal{F}$. Fiecare simbol are atașat un număr natural numit aritatea simbolului.

În acest capitol vom utiliza signatura $\Sigma = (\{\mathtt{P}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{e}\})$, unde $\mathtt{P}$ este simbol predicativ de aritate 2, iar $\mathtt{f}$, $\mathtt{i}$ și $\mathtt{e}$ sunt simboluri funcționale de aritate 2, 1 și respectiv 0. Altfel spus, $\mathcal{P}_2 = \{\mathtt{P}\}, \mathcal{P}_1 = \emptyset, \mathcal{P}_0 = \emptyset, \mathcal{F}_2 = \{\mathtt{f}\}, \mathcal{F}_1 = \{\mathtt{i}\}$ și $\mathcal{F}_0 = \{\mathtt{e}\}$.

Reamintim și că, dacă $\Sigma = (\mathcal{P}, \mathcal{F})$ este o signatură, prin Σ-structură înțelegem orice tuplu $S = (D, Pred, Fun)$ cu proprietatea că:

1. D este o mulțime nevidă numită domeniul structurii S;
2. pentru fiecare simbol predicativ $P \in \mathcal{P}$ există un predicat $P^S \in Pred$ de aritate corespunzătoare;
3. pentru fiecare simbol funcțional $f \in \mathcal{F}$ există o funcție $f^S \in Fun$ de aritate corespunzătoare.

Exemplul 41. *Mai jos avem câteva exemple de Σ-structuri:*

1. $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$;

2. $S_2 = (\mathbb{R}^*, \{=\}, \{\times, \cdot^{-1}, 1\})$;
3. $S_3 = (\mathbb{N}, \{=\}, \{+, s, 0\})$;
4. $S_4 = (\mathbb{N}, \{<\}, \{+, s, 0\})$;
5. $S_5 = (\mathbb{Z}, \{<\}, \{+, -, 0\})$.

Structura S_1 are domeniul $\mathbb{Z}$ (mulțimea numerelor întregi), predicatul asociat simbolului predicativ P *este $=$ (predicatul de egalitate pentru numere întregi), funcția $+$ este funcția de adunare a numerelor întregi asociată simbolului funcțional* f*, $-$ este funcția minus unar asociată simbolului funcțional* i*, iar simbolul constant* e *are asociată constanta* 0.

Structura S_2 are domeniul $\mathbb{R}^*$ (mulțimea numerelor reale strict pozitive), predicatul asociat simbolului predicativ P *este $=$ (predicatul de egalitate pentru numere reale pozitive), funcția $\times$ este funcția de înmulțire a numerelor reale asociată simbolului funcțional* f*, $\cdot^{-1}$ este funcția unară asociată simbolului funcțional* i *care calculează inversul unui număr real (e.g. $5^{-1} = \frac{1}{5}$, iar $\left(\frac{1}{10}\right)^{-1} = 10$), iar simbolul constant* e *are asociată constanta* 1.

Structura S_3 are domeniul $\mathbb{N}$ (mulțimea numerelor naturale), predicatul asociat simbolului predicativ P *este $=$ (predicatul de egalitate pentru numere naturale), funcția $+$ este funcția de adunare a numerelor naturale asociată simbolului funcțional* f*, s este funcția succesor (care asociază unui număr natural următorul număr natural – e.g., $s(7) = 8$) asociată simbolului funcțional* i*, iar simbolul constant* e *are asociată constanta* 0.

Structura S_4 are domeniul $\mathbb{N}$ (mulțimea numerelor naturale), predicatul asociat simbolului predicativ P *este $<$ (relația* mai mic *peste numere naturale), funcția $+$ este funcția de adunare a numerelor naturale asociată simbolului funcțional* f*, s este funcția succesor (care asociază unui număr natural următorul număr natural – e.g., $s(7) = 8$) asociată simbolului funcțional* i*, iar simbolul constant e are asociată constanta* 0.

Structura S_5 este similară cu S_1, doar că simbolul predicativ P *are asociată relația* mai mic *în loc de* egal.

Folosind notațiile de mai sus, avem că $\mathtt{P}^{S_4} = <$, $\mathtt{f}^{S_2} = \times$, *iar* $\mathtt{e}^{S_1} = 0$.

4.1 Atribuiri

Asemănător cu logica propozițională, pentru a obține valoarea de adevăr a unei formule într-o structură, trebuie să pornim cu fixarea unor valori concrete pentru simbolurile sintactice din alfabetul peste care este construită formula. În cazul de față, începem cu variabilele.

Definiția 42 (Atribuire)**.** *Fie Σ o signatură și S o Σ-structură cu domeniul D. Se numește S-atribuire este orice funcție*

$$\alpha : \mathcal{X} \to D.$$

Exemplul 43. *Funcția $\alpha_1 : \mathcal{X} \to \mathbb{Z}$, definită ca mai jos, este o S_1-atribuire:*

1. $\alpha_1(\mathsf{x}_1) = 5$;
2. $\alpha_1(\mathsf{x}_2) = 5$;
3. $\alpha_1(\mathsf{x}_3) = 6$;
4. $\alpha_1(x) = 0$ *pentru orice* $x \in \mathcal{X} \setminus \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3\}$.

Exemplul 44. *Funcția $\alpha_2 : \mathcal{X} \to \mathbb{Z}$, definită ca mai jos, este o S_1-atribuire:*

1. $\alpha_2(\mathsf{x}_1) = 6$;
2. $\alpha_2(\mathsf{x}_2) = 5$;
3. $\alpha_2(\mathsf{x}_3) = 6$;
4. $\alpha_2(x) = 0$ *pentru orice* $x \in \mathcal{X} \setminus \{\mathsf{x}_1, \mathsf{x}_2, \mathsf{x}_3\}$.

Acum, având la dispoziție o atribuire α, putem calcula valoarea unui termen într-o asemenea atribuire. Pentru aceasta, vom folosi de fapt extensia lui α, notată $\overline{\alpha}$,

$$\overline{\alpha} : \mathcal{T} \to D,$$

dată în definiția care urmează.

Definiția 45 (Valoarea unui termen într-o atribuire)**.** *Dându-se o S-atribuire α și un termen $t \in \mathcal{T}$ peste signatura Σ,* valoarea termenului t în atribuirea α *este un element al domeniului D notat cu $\overline{\alpha}(t)$ și calculat recursiv astfel:*

1. $\overline{\alpha}(c) = c^S$ *dacă $c \in \mathcal{F}_0$ (i.e., c este un simbol constant);*
2. $\overline{\alpha}(x) = \alpha(x)$ *dacă $x \in \mathcal{X}$ (i.e., x este o variabilă);*
3. $\overline{\alpha}(f(t_1, \ldots, t_n)) = f^S(\overline{\alpha}(t_1), \ldots, \overline{\alpha}(t_n))$ *dacă $f \in \mathcal{F}_n$ este un simbol funcțional de aritate n, iar $t_1, \ldots, t_n$ sunt termeni.*

Exemplul 46. *Continuând Exemplul 43, unde α_1 este o S_1-atribuire, avem:*

$$\begin{aligned}\overline{\alpha_1}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1),\mathsf{e})) &= \overline{\alpha_1}(\mathsf{i}(\mathsf{x}_1)) + \overline{\alpha_1}(\mathsf{e})\\ &= -(\overline{\alpha_1}(\mathsf{x}_1)) + \mathsf{e}^{S_1}\\ &= -(\alpha_1(\mathsf{x}_1)) + 0\\ &= -5 + 0\\ &= -5.\end{aligned}$$

Așadar, valoarea termenului $\mathsf{f}(\mathsf{i}(\mathsf{x}_1),\mathsf{e})$ în atribuirea α_1 este -5.

Definiția 47 (Actualizarea unei atribuiri)**.** *Dându-se o atribuire α, o variabilă $x \in \mathcal{X}$ și un element $u \in D$, notăm cu $\alpha[x \mapsto u]$ o nouă atribuire, care coincide cu α, exceptând valoarea variabilei x, care devine acum u:*

$$\alpha[x \mapsto u] : \mathcal{X} \to D, \text{ a.î.}$$

1. $(\alpha[x \mapsto u])(x) = u$;
2. $(\alpha[x \mapsto u])(y) = \alpha(y)$, *pentru orice* $y \in \mathcal{X} \setminus \{x\}$.

Exemplul 48. *De exemplu, atribuirea $\alpha_1[\mathsf{x}_1 \mapsto 6]$ este exact atribuirea α_2 definită în exemplele de mai sus. Valoarea termenului $\mathsf{f}(\mathsf{i}(\mathsf{x}_1),\mathsf{e})$ în atribuirea $\alpha_1[\mathsf{x}_1 \mapsto 6]$, notată cu $\overline{\alpha_1[\mathsf{x}_1 \mapsto 6]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1),\mathsf{e}))$, este -6.*

Exercițiul 49. *Calculați valorile de mai jos:*

1. $\overline{\alpha_1[\mathsf{x}_1 \mapsto 10]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1),\mathsf{e}))$;
2. $\overline{\alpha_1[\mathsf{x}_2 \mapsto 10]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1),\mathsf{e}))$;
3. $\overline{\alpha_1[\mathsf{x}_2 \mapsto 10][\mathsf{x}_1 \mapsto 10]}(\mathsf{f}(\mathsf{i}(\mathsf{x}_1),\mathsf{e}))$.

4.2 Valoarea de adevăr a unei formule de ordinul I

În acest moment avem ingredientele pentru a defini formal valoarea de adevăr a unei formule de ordinul I, construită peste o signatură Σ. Această valoare se poate calcula doar într-o Σ-structură S, cu ajutorul unei S-atribuiri α.

Notațiile folosite sunt similare cu cele pentru logica propozițională. Astfel, notăm faptul că o formulă φ este adevărată într-o structură S cu o atribure α prin $S, \alpha \models \varphi$. Faptul că o formulă φ nu este adevărată într-o structură S cu o atribuire α se notează cu $S, \alpha \not\models \varphi$.

Notația $S, \alpha \models \varphi$ se mai citește *S satisface φ cu atribuirea α*, iar $S, \alpha \not\models \varphi$ se mai citește *S nu satisface φ cu atribuirea α*.

Definiția 50. *Faptul că o structură S satisface o formulă φ cu o anumită atribuire α (echivalent, φ este adevărată în structura S cu atribuirea α) se definește inductiv astfel (prima linie din enumerarea care urmează desemnează cazul de bază, restul reprezentând cazurile inductive):*

1. $S, \alpha \models P(t_1, \ldots, t_n)$ *ddacă* $P^S(\overline{\alpha}(t_1), \ldots, \overline{\alpha}(t_n))$;
2. $S, \alpha \models \neg\varphi$ *ddacă* $S, \alpha \not\models \varphi$;
3. $S, \alpha \models (\varphi_1 \wedge \varphi_2)$ *ddacă* $S, \alpha \models \varphi_1$ *și* $S, \alpha \models \varphi_2$;
4. $S, \alpha \models (\varphi_1 \vee \varphi_2)$ *ddacă* $S, \alpha \models \varphi_1$ *sau* $S, \alpha \models \varphi_2$;
5. $S, \alpha \models (\varphi_1 \to \varphi_2)$ *ddacă* $S, \alpha \not\models \varphi_1$ *sau* $S, \alpha \models \varphi_2$;
6. $S, \alpha \models (\varphi_1 \leftrightarrow \varphi_2)$ *ddacă (1) atât* $S, \alpha \models \varphi_1$, *cât și* $S, \alpha \models \varphi_2$, *sau (2)* $S, \alpha \not\models \varphi_1$ *și* $S, \alpha \not\models \varphi_2$;
7. $S, \alpha \models (\exists x.\varphi)$ *ddacă există* $u \in D$ *astfel încât* $S, \alpha[x \mapsto u] \models \varphi$;
8. $S, \alpha \models (\forall x.\varphi)$ *ddacă pentru orice* $u \in D$, *avem că* $S, \alpha[x \mapsto u] \models \varphi$.

Exemplul 51. *Vom lucra în continuare peste signatura* $\Sigma = (\{\mathtt{P}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{e}\})$, Σ*-structura* $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$ *definită la începutul capitolului și* S_1*-atribuirile* α_1, α_2.

Avem că

$$\begin{aligned} S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \ &\textit{ddacă} & \mathtt{P}^{S_1}(\overline{\alpha_1}(\mathtt{x}_1), \overline{\alpha_1}(\mathtt{x}_1)) \\ &\textit{ddacă} & \overline{\alpha_1}(\mathtt{x}_1) = \overline{\alpha_1}(\mathtt{x}_1) \\ &\textit{ddacă} & \alpha_1(\mathtt{x}_1) = \alpha_1(\mathtt{x}_1) \\ &\textit{ddacă} & 5 = 5. \end{aligned}$$

Din moment ce $5 = 5$, *rezultă că* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$, *adică formula* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *este adevărată în structura* S_1 *cu atribuirea* α_1. *Altfel spus,* S_1 *satisface* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *cu atribuirea* α_1.

Exemplul 52. *Continuând exemplul anterior, avem*

$$\begin{aligned} S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \ &\textit{ddacă} & P^{S_1}(\overline{\alpha_1}(\mathtt{x}_1), \overline{\alpha_1}(\mathtt{x}_3)) \\ &\textit{ddacă} & \overline{\alpha_1}(\mathtt{x}_1) = \overline{\alpha_1}(\mathtt{x}_3) \\ &\textit{ddacă} & \alpha_1(\mathtt{x}_1) = \alpha_1(\mathtt{x}_3) \\ &\textit{ddacă} & 5 = 6. \end{aligned}$$

Din moment ce $5 \neq 6$, *rezultă că* $S_1, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$, *adică formula* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *este falsă în structura* S_1 *cu atribuirea* α_1. *Altfel spus* S_1 *nu satisface* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *cu atribuirea* α_1.

Exemplul 53. *Continuând exemplul anterior, avem*

$$
\begin{array}{rr}
S_1, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \ \textit{ddacă} & S_1, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \\
\textit{ddacă} & \textit{nu } P^{S_1}(\overline{\alpha_1}(\mathtt{x}_1), \overline{\alpha_1}(\mathtt{x}_3)) \\
\textit{ddacă} & \textit{nu } \overline{\alpha_1}(\mathtt{x}_1) = \overline{\alpha_1}(\mathtt{x}_3) \\
\textit{ddacă} & \overline{\alpha_1}(\mathtt{x}_1) \neq \overline{\alpha_1}(\mathtt{x}_3) \\
\textit{ddacă} & \alpha_1(\mathtt{x}_1) \neq \alpha_1(\mathtt{x}_3) \\
\textit{ddacă} & 5 \neq 6.
\end{array}
$$

Din moment ce $5 \neq 6$, *rezultă că* $S_1, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$, *adică formula* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *este adevărată în structura* S_1 *cu atribuirea* α_1. *Altfel spus,* S_1 *satisface* $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *cu atribuirea* α_1.

Exemplul 54. *Continuând exemplul anterior, avem*

$$
\begin{array}{ll}
S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \wedge \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) & \textit{ddacă} \\
S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \ \textit{și} \ S_1, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) & \textit{ddacă} \\
\ldots \ \textit{și} \ \ldots & \textit{ddacă} \\
5 = 5 \ \textit{și} \ 5 \neq 6. &
\end{array}
$$

Din moment ce $5 = 5$ *și* $5 \neq 6$, *rezultă că* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \wedge \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$.

Exemplul 55. *Continuând exemplul anterior, avem*

$S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \vee \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *dacă* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$ *sau* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$.

Am stabilit deja că $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3)$, *deci* $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \vee \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ *(chiar dacă* $S_1, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$*).*

Exemplul 56. *Continuând exemplul anterior, avem*

$$
\begin{array}{ll}
S_1, \alpha_1 \models \exists \mathtt{x}_1 . \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) & \textit{ddacă} \\
\textit{există } u \in D \ \textit{a.î.} \ S_1, \alpha_1[\mathtt{x}_1 \mapsto u] \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) & \textit{ddacă} \\
\textit{există } u \in D \ \textit{a.î.} \ \mathtt{P}^{S_1}(\overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_1), \overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_3)) & \textit{ddacă} \\
\textit{există } u \in D \ \textit{a.î.} \ \overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_1) = \overline{\alpha_1[\mathtt{x}_1 \mapsto u]}(\mathtt{x}_3) & \textit{ddacă} \\
\textit{există } u \in D \ \textit{a.î.} \ \alpha_1[\mathtt{x}_1 \mapsto u](\mathtt{x}_1) = \alpha_1[\mathtt{x}_1 \mapsto u](\mathtt{x}_3) & \textit{ddacă} \\
\textit{există } u \in D \ \textit{a.î.} \ u = \alpha_1(\mathtt{x}_3) & \textit{ddacă} \\
\textit{există } u \in D \ \textit{a.î.} \ u = 6. &
\end{array}
$$

Din moment ce există u (putem alege $u = 6$) a.î. $u = 6$, avem că $S_1, \alpha_1 \models \exists x_1.\mathtt{P}(x_1, x_3)$.

Exemplul 57. *Continuând exemplul anterior, avem*

$S_1, \alpha_1 \models \forall x_1.\exists x_3.\mathtt{P}(x_1, x_3)$ *ddacă*
pentru orice $u \in D$, *avem că* $S_1, \alpha_1[x_1 \mapsto u] \models \exists x_3.\mathtt{P}(x_1, x_3)$ *ddacă*
pt. orice $u \in D$, *există* $v \in D$ *a.î.* $S_1, \alpha_1[x_1 \mapsto u][x_3 \mapsto v] \models \mathtt{P}(x_1, x_3)$ *ddacă*
... *ddacă*
pentru orice $u \in D$, *avem că există* $v \in D$ *a.î.* $u = v$.

Din moment ce pentru orice număr întreg u, există un număr întreg v a.î. $u = v$, avem că $S_1, \alpha_1 \models \forall x_1.\exists x_3.\mathtt{P}(x_1, x_3)$.

Exercițiul 58. *Arătați că* $S_1, \alpha_1 \models \forall x_1.\exists x_3.\mathtt{P}(x_1, \mathtt{i}(x_3))$.

4.3 Satisfiabilitate într-o structură fixată

Definiția 59 (Satisfiabilitate într-o structură fixată)**.** *O formulă φ* este satisfiabilă într-o structură *S dacă există o S-atribuire α cu proprietatea că*

$$S, \alpha \models \varphi.$$

Exemplul 60. *Formula* $\mathtt{P}(x_1, x_3)$ *este satisfiabilă în structura S_1, deoarece există o atribuire, de exemplu α_2, cu proprietatea că* $S_1, \alpha_2 \models \mathtt{P}(x_1, x_3)$.

Exercițiul 61. *Arătați că formula* $\neg\mathtt{P}(x_1, x_1)$ *nu este satisfiabilă în structura S_1 (deoarece, pentru orice atribuire α aleasă, avem că* $S_1, \alpha \not\models \neg\mathtt{P}(x_1, x_1)$*).*

4.4 Validitate într-o structură fixată

Definiția 62 (Validitate într-o structură fixată)**.** *O formulă φ* este validă într-o structură *S dacă pentru orice S-atribuire α, avem că*

$$S, \alpha \models \varphi.$$

Exemplul 63. *Formula* $\mathtt{P}(x_1, x_3)$ *nu este validă în structura S_1, deoarece există o atribuire, și anume α_1, cu propritatea că* $S_1, \alpha_1 \not\models \mathtt{P}(x_1, x_3)$.

Exercițiul 64. *Arătați că formula* $\mathtt{P}(x_1, x_1)$ *este validă în structura S_1 (deoarece orice atribuire α am alege,* $S_1, \alpha \models \mathtt{P}(x_1, x_1)$*).*

4.5 Satisfiabilitate

Definiția 65 (Satisfiabilitate)**.** *O formulă φ* este satisfiabilă *dacă există o structură S și o S-atribuire α cu proprietatea că*

$$S, \alpha \models \varphi.$$

Exemplul 66. *Formula $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ este satisfiabilă, deoarece există o structură (de exemplu S_5) și o S_5-atribuire (de exemplu α_1) astfel încât $S_5, \alpha_1 \models \neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ (deoarece $5 \not< 5$).*

Să subliniem faptul că, deoarece S_5 și S_1 au același domeniu, atribuirea α_1 este atât o S_1-atribuire cât și o S_5-atribuire.

Observație. *O formulă poate să nu fie satisfiabilă într-o structură fixată (de exemplu $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ nu este satisfiabilă în structura S_1) și totuși să fie satisfiabilă (vezi Exemplul 66 unde aceeași formulă $\neg \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ este satisfiabilă).*

4.6 Validitate

Definiția 67 (Validitate)**.** *O formulă φ* este validă *dacă, pentru orice structură S și pentru orice S-atribuire α, avem*

$$S, \alpha \models \varphi.$$

Exemplul 68. *Formula $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ nu este validă, deoarece $S_5, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$. Pe de altă parte, formula $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1) \to \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ este validă.*

Observație. *O formulă poate să fie validă într-o structură fixată (de exemplu $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ este validă în structura S_1) și totuși să nu fie validă (de exemplu, $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$ nu este validă, deoarece $S_5, \alpha_1 \not\models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$).*

4.7 Consecință semantică

Definiția 69. *O formulă φ* este consecință semantică a formulelor $\varphi_1, \ldots, \varphi_n$ *într-o structură fixată S, notat $\varphi_1, \ldots, \varphi_n \models_S \varphi$, dacă, pentru orice S-atribuire α pentru care $S, \alpha \models \varphi_1$, $S, \alpha \models \varphi_2$, $\ldots$, $S, \alpha \models \varphi_n$, avem că $S, \alpha \models \varphi$.*

Exemplul 70. *Avem că $\mathtt{P}(\mathtt{x}, \mathtt{y}) \models_{S_1} \mathtt{P}(\mathtt{y}, \mathtt{x})$, deoarece, pentru orice S_1-atribuire α cu proprietatea că $S_1, \alpha \models \mathtt{P}(\mathtt{x}, \mathtt{y})$ (adică $\alpha(\mathtt{x}) = \alpha(\mathtt{y})$), avem și că $S_1, \alpha \models \mathtt{P}(\mathtt{y}, \mathtt{x})$ (adică $\alpha(\mathtt{y}) = \alpha(\mathtt{x})$).*

Avem că $\mathtt{P}(\mathtt{x}, \mathtt{y}) \not\models_{S_5} \mathtt{P}(\mathtt{y}, \mathtt{x})$, deoarece, pentru atribuirea $\alpha(\mathtt{x}) = 5$, $\alpha(\mathtt{y}) = 6$, avem că $S_5, \alpha \models \mathtt{P}(\mathtt{x}, \mathtt{y})$ (adică $5 < 6$), dar $S_5, \alpha \not\models \mathtt{P}(\mathtt{y}, \mathtt{x})$ ($6 \not< 5$).

Definiția 71. *O formulă φ este consecință semantică a formulelor $\varphi_1, \ldots, \varphi_n$, notat $\varphi_1, \ldots, \varphi_n \models \varphi$, dacă*

$$\varphi_1, \ldots, \varphi_n \models_S \varphi$$

pentru orice structură S.

Exemplul 72. *Avem că* $\mathtt{P(x,y)} \not\models \mathtt{P(y,x)}$, *deoarece există o structură (și anume S_5) astfel încât* $\mathtt{P(x,y)} \not\models_{S_5} \mathtt{P(y,x)}$.

Exercițiul 73. *Arătați că:*

$$\forall \mathtt{x}.\forall \mathtt{y}.\forall \mathtt{z}.(\mathtt{P(x,y)} \wedge \mathtt{P(y,z)} \rightarrow \mathtt{P(x,z)}), \mathtt{P}(\mathtt{x}_1, \mathtt{x}_2), \mathtt{P}(\mathtt{x}_2, \mathtt{x}_3) \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3).$$

Desigur că, în cele de mai sus (similar cu logica propozițională), lista $\varphi_1, \varphi_2, \ldots, \varphi_n$ denotă de fapt o mulțime având respectivele elemente.

4.8 Fișă de exerciții

Amintim mai jos structurile din Exemplul 41:

1. $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$;
2. $S_2 = (\mathbb{R}^*, \{=\}, \{\times, \cdot^{-1}, 1\})$;
3. $S_3 = (\mathbb{N}, \{=\}, \{+, s, 0\})$;
4. $S_4 = (\mathbb{N}, \{<\}, \{+, s, 0\})$.
5. $S_5 = (\mathbb{Z}, \{<\}, \{+, -, 0\})$.

Aceste structuri vor fi utilizate în exercițiile de mai jos.

Exercițiul 74. *Stabiliți dacă:*

1. $S_1, \alpha_1 \models \mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
2. $S_1, \alpha_1 \models \neg\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
3. $S_1, \alpha_1 \models \neg\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3) \wedge \mathtt{P}(\mathtt{x}_1, \mathtt{x}_1)$;
4. $S_1, \alpha_1 \models \exists \mathtt{x}_3.\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
5. $S_1, \alpha_1 \models \forall \mathtt{x}_2.\exists \mathtt{x}_3.\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
6. $S_1, \alpha_1 \models \exists \mathtt{x}_3.\forall \mathtt{x}_2.\mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$;
7. $S_1, \alpha_2 \models \forall \mathtt{x}_2.\exists \mathtt{x}_3.\mathtt{P}(\mathtt{x}_2, \mathtt{i}(\mathtt{x}_3))$;

Exercițiul 75. *Găsiți pentru fiecare dintre itemii de mai jos câte o S_2-atribuire α_3 astfel încât:*

1. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_2)$;
2. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{f}(\mathtt{x}_1, \mathtt{x}_2), \mathtt{x}_3)$;
3. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{f}(\mathtt{x}_1, \mathtt{x}_2), \mathtt{i}(\mathtt{x}_3))$;
4. $S_2, \alpha_3 \models \mathtt{P}(\mathtt{x}, \mathtt{e})$;
5. $S_2, \alpha_3 \models \exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$;
6. $S_2, \alpha_3 \models \forall \mathtt{y}.\exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$.

Exercițiul 76. *Arătați că următoarele formule sunt valide în S_2:*

1. $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$;
2. $\forall \mathtt{x}.\mathtt{P}(\mathtt{f}(\mathtt{x}, \mathtt{e}), \mathtt{x})$;
3. $\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{i}(\mathtt{x})))$.

Exercițiul 77. *Arătați că formula $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{y}))$ nu este validă în S_3.*

Exercițiul 78. *Găsiți o formulă care să fie satisfiabilă în S_1 dar nu în S_3.*

Exercițiul 79. *Găsiți o formulă fără variabile libere care să fie satisfiabilă în S_5 dar nu în S_4.*

Exercițiul 80. *Arătați că formula $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{y})$ nu este validă.*

Exercițiul 81. *Arătați că formula $(\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{x})) \to \exists \mathtt{x}_2.\mathtt{P}(\mathtt{x}_1, \mathtt{x}_2)$ este validă.*

Exercițiul 82. *Arătați că formula $\forall \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{y}, \mathtt{x})$ nu este validă.*

Exercițiul 83. *Arătați că formula $\forall \mathtt{x}.\neg \mathtt{P}(\mathtt{x}, \mathtt{x})$ este satisfiabilă.*

Exercițiul 84. *Arătați că formula $\forall \mathtt{x}.\neg \mathtt{P}(\mathtt{x}, \mathtt{x}) \wedge \exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{x})$ nu este satisfiabilă.*

Exercițiul 85. *În Exercițiul 40 (din Capitolul 3) am arătat că $\mathbb{LPI}$ este o extensie sintactică a lui $\mathbb{LP}$. Pe scurt, dacă $A = \{\mathtt{p}, \mathtt{q}, \mathtt{r}, \ldots\}$ este o mulțime de variabile propoziționale atunci construim o signatura $\Sigma_{\mathbb{LP}} = \{A, \emptyset\}$, unde variabilele propoziționale din A sunt simboluri predicative de aritate 0.*

Semantica formulelor $\mathbb{LPI}$ construite peste signatura $\Sigma_{\mathbb{LP}}$ este consistentă cu semantica formulelor $\mathbb{LP}$. Fie $\tau : A \to B$ o atribuire. Fie $S = (D, \{a^S \mid a \in A\}, \emptyset)$ o $\Sigma_{\mathtt{LP}}$-structură, unde D este orice mulțime nenulă și $a^S = \tau(a)$, pentru orice $a \in A$.

Demonstrați că pentru orice $\varphi \in \mathbb{LP}$, avem că $\tau \models \varphi$ ddacă $S, \alpha \models \varphi$ pentru orice S-atribuire $\alpha : \mathcal{X} \to D$.

Capitolul 5

Deducția naturală

În acest capitol, vom prezenta deducția naturală pentru logica de ordinul I. Vom defini noțiunea de *substituție*, vom reaminti câteva dintre noțiunile specifice deducției naturale (discutate în prealabil la logica propozițională) și vom prezenta sistemul deductiv extins împreună cu proprietățile sale (corectitudine și completitudine).

Observație. *Regulile sistemului deductiv al deducției naturale include regulile discutate deja la logica propozițională. În acest capitol, acestea din urmă sunt reluate și exemplificate pe formule de ordinul I. Deducția naturală pentru logica de ordinul I include reguli pentru cuantificatori. Acestea sunt reguli care nu au fost studiate anterior la logica propozițională.*

5.1 Substituţii

Definiția 86. *O* substituție *este o funcție $\sigma : \mathcal{X} \to \mathcal{T}$, cu proprietatea că $\sigma(x) \neq x$ pentru un număr finit de variabile $x \in \mathcal{X}$.*

Definiția 87. *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.*

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

Definiția 88. *Dacă $\sigma : \mathcal{X} \to \mathcal{T}$ este o substituție, atunci* extensia *unică a substituției σ la mulțimea termenilor este funcția $\sigma^\sharp : \mathcal{T} \to \mathcal{T}$, definită recursiv 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* $t_1,\ldots,t_n \in \mathcal{T}$.

De regulă, substituţiile se notează cu $\sigma, \tau, \sigma_0, \tau_1, \sigma'$, etc.

Observație. *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, toate apariţile unei variabile x din t sunt înlocuite simultan cu termenul corespunzător $\sigma(x)$.

Exemplul 89. *Fie substituția* $\sigma_1 : \mathcal{X} \to \mathcal{T}$ *definită astfel:*

1. $\sigma_1(\mathsf{x}_1) = \mathsf{x}_2$;
2. $\sigma_1(\mathsf{x}_2) = \mathsf{f}(\mathsf{x}_3, \mathsf{x}_4)$;
3. $\sigma_1(x) = x$ *pentru orice* $x \in \mathcal{X} \setminus \{\mathsf{x}_1, \mathsf{x}_2\}$.

Fie termenul $t = \mathsf{f}(\mathsf{f}(\mathsf{x}_1, \mathsf{x}_2), \mathsf{f}(\mathsf{x}_3, \mathsf{e}))$. *Avem că:*

$$\begin{aligned}
\sigma_1^\sharp\Big(t\Big) &= \sigma_1^\sharp\Big(\mathsf{f}(\mathsf{f}(\mathsf{x}_1,\mathsf{x}_2),\mathsf{f}(\mathsf{x}_3,\mathsf{e}))\Big)\\
&= \mathsf{f}(\sigma_1^\sharp\Big(\mathsf{f}(\mathsf{x}_1,\mathsf{x}_2)\Big),\sigma_1^\sharp\Big(\mathsf{f}(\mathsf{x}_3,\mathsf{e})\Big))\\
&= \mathsf{f}(\mathsf{f}(\sigma_1^\sharp\Big(\mathsf{x}_1\Big),\sigma_1^\sharp\Big(\mathsf{x}_2\Big)),\mathsf{f}(\sigma_1^\sharp\Big(\mathsf{x}_3\Big),\sigma_1^\sharp\Big(\mathsf{e}\Big)))\\
&= \mathsf{f}(\mathsf{f}(\sigma_1\Big(\mathsf{x}_1\Big),\sigma_1\Big(\mathsf{x}_2\Big)),\mathsf{f}(\sigma_1\Big(\mathsf{x}_3\Big),\mathsf{e}))\\
&= \mathsf{f}(\mathsf{f}(\mathsf{x}_2,\mathsf{f}(\mathsf{x}_3,\mathsf{x}_4)),\mathsf{f}(\mathsf{x}_3,\mathsf{e})).
\end{aligned}$$

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

Notație. *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 doar de o ***notație*** *pentru substituții.*

Exemplul 90. *Pentru substituția din Exemplul 89, avem*

$$\sigma_1 = \{\mathsf{x}_1 \mapsto \mathsf{x}_2, \mathsf{x}_2 \mapsto \mathsf{f}(\mathsf{x}_3, \mathsf{x}_4)\}.$$

Definiția 91. *Dacă $\sigma : \mathcal{X} \to \mathcal{T}$ este o substituție și $V \subseteq \mathcal{X}$ 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* $x \in \mathcal{X} \setminus V$.

Exemplul 92. *Pentru substituția din Exemplul 89, avem* $\sigma_1|_{\{x_1\}} = \{x_1 \mapsto x_2\}$ *și* $\sigma_1|_{\{x_2\}} = \{x_2 \mapsto f(x_3, x_4)\}$.

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

Definiția 93. *Pentru orice substituție $\sigma : \mathcal{X} \to \mathcal{T}$,* extensia *lui σ la mulțimea formulelor este funcția $\sigma^\flat : \mathbb{LPI} \to \mathbb{LPI}$, 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. $\sigma^\flat\big((\varphi_1 \wedge \varphi_2)\big) = (\sigma^\flat(\varphi_1) \wedge \sigma^\flat(\varphi_2))$;
4. $\sigma^\flat\big((\varphi_1 \vee \varphi_2)\big) = (\sigma^\flat(\varphi_1) \vee \sigma^\flat(\varphi_2))$;
5. $\sigma^\flat\big((\varphi_1 \to \varphi_2)\big) = (\sigma^\flat(\varphi_1) \to \sigma^\flat(\varphi_2))$;
6. $\sigma^\flat\big((\varphi_1 \leftrightarrow \varphi_2)\big) = (\sigma^\flat(\varphi_1) \leftrightarrow \sigma^\flat(\varphi_2))$;
7. $\sigma^\flat\big((\forall x.\varphi)\big) = \big(\forall x.(\rho^\flat(\varphi))\big)$, *unde* $\rho = \sigma|_{dom(\sigma)\setminus\{x\}}$;
8. $\sigma^\flat\big((\exists x.\varphi)\big) = \big(\exists x.(\rho^\flat(\varphi))\big)$, *unde* $\rho = \sigma|_{dom(\sigma)\setminus\{x\}}$;

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

Exemplul 94. *Utilizând substituția din Exemplul 89, avem că:*

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

Observație. *Atenție: aparițiile legate ale variabilelor nu sunt înlocuite prin aplicarea substituției! În Exemplul 94, apariția lui* $\mathtt{x}_2$ *în* $(\forall \mathtt{x}_2.\mathtt{P}(\mathtt{x}_1, \mathtt{x}_2))$ *este legată.*

Notație. *Conform Notației 5.1 pentru substituțiile care au domeniul finit mai utilizăm notația* $\{x_1 \mapsto \sigma(x_1), \ldots, x_n \mapsto \sigma(x_n)\}$. *De multe ori vom utiliza substituții pentru care nu vom mai asocia un nume, deoarece ele sunt foarte simple având forma:* $\{x \mapsto t\}$. *Pentru a exprima faptul că aplicăm această substituție unei formule, conform notațiilor noastre, ar trebui să scriem* $\{x \mapsto t\}(\varphi)$. *Însă în literatură sunt preferate alte notații pe care le vom utiliza și noi. O variantă este să scriem* $\varphi[t/x]$. *O altă variantă este* $\varphi[x \mapsto t]$. *În acest document vom prefera ultima notație, adică* $\varphi[x \mapsto t]$.

5.2 Secvențe

Definiția 95 (Secvență)**.** *O* secvență *este o pereche de forma:*

$$\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi,$$

unde $\{\varphi_1, \ldots, \varphi_n\} \subseteq \mathbb{LPI}$ *este o mulțime de formule iar* $\varphi \in \mathbb{LPI}$ *este o formulă.*

Câteodată citim notația $\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$ sub forma φ *este consecință sintactică din* $\{\varphi_1, \ldots, \varphi_n\}$. De multe ori, vom nota cu $\Gamma = \{\varphi_1, \ldots, \varphi_n\}$ mulțimea de ipoteze și în acest caz vom scrie secvența ca $\Gamma \vdash \varphi$.

Observație. *Ca și în cazul logicii propoziționale, este permisă scrierea fără acolade, adică* $\varphi_1, \ldots, \varphi_n \vdash \varphi$, *în loc de* $\{\varphi_1, \ldots, \varphi_n\} \vdash \varphi$. *Totuși trebuie să ținem cont că în partea stângă a simbolului* $\vdash$ *este tot timpul o mulțime. Această notație fără acolade ne permite să scriem* $\varphi_1, \ldots, \varphi_n, \psi \vdash \varphi$ *în loc de* $\{\varphi_1, \ldots, \varphi_n\} \cup \{\psi\} \vdash \varphi$, *ceea ce ușurează citirea secvențelor.*

Exemplul 96. *În multe dintre exemplele din acest material vom lucra cu o signatură* $\Sigma = (\{\mathtt{P}, \mathtt{Q}\}, \{\mathtt{a}, \mathtt{b}, \mathtt{f}, \mathtt{g}\})$, *unde simbolurile predicative* $\mathtt{P}$ *și* $\mathtt{Q}$ *au aritate 1, simbolurile funcționale* $\mathtt{f}$ *și* $\mathtt{g}$ *au aritate 1, iar simbolurile* $\mathtt{a}$ *și* $\mathtt{b}$ *sunt constante (de aritate 0).*

Exemplul 97. *Fie signatura* Σ *din Exemplul 96. Iată câteva exemple de secvențe:*

1. $\{\mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$;
2. $\{\forall \mathtt{x}.\mathtt{Q}(\mathtt{x}), \mathtt{P}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$;
3. $\{\exists \mathtt{x}.\mathtt{Q}(\mathtt{x})\} \vdash \mathtt{Q}(\mathtt{a})$.

Mai târziu vom vedea că primele două secvențe de mai sus sunt valide, *iar ultima secvență nu este validă.*

Uneori este convenabil să scriem secvenţele fara acolade, ca în exemplul următor:

Exemplul 98. *Secvențele din Exemplul 97 pot fi scrise fără acolade astfel:*

1. $\mathtt{P(a)}, \mathtt{Q(a)} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$;
2. $\forall \mathtt{x}.\mathtt{Q(x)}, \mathtt{P(a)} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$;
3. $\exists \mathtt{x}.\mathtt{Q(x)} \vdash \mathtt{Q(a)}$.

5.3 Reguli de inferență

Definiția 99. *O* regulă de inferență *este un tuplu format din:*

1. *o mulțime de secvențe $S_1, \ldots, S_n$, care se numesc* ipotezele *regulii;*
2. *o secvență S care se numește* concluzia *regulii;*
3. *o* condiție *de aplicare a regulii;*
4. *un nume.*

O regulă de inferență se notează în felul următor:

$$\textsc{nume}\ \frac{S_1 \qquad \ldots \qquad S_n}{S}\ \textit{condiție.}$$

Observație. *Regulile de inferență care au $n = 0$ ipoteze, se numesc axiome. De asemenea, condiția de aplicare poate să lipsească.*

Exemplul 100. *Iată câteva exemple de reguli de inferență pe care le-am întâlnit și la logica propozițională:*

$$\wedge i\ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,} \qquad \wedge e_2\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2.}$$

Ca și în logica propozițională, toate cele trei reguli de inferență de mai sus sunt corecte. *Niciuna dintre cele trei reguli de mai sus nu are o condiție atașată. Iată și un exemplu de regulă cu $n = 0$ ipoteze, dar cu o condiție:*

$$\textsc{Ipoteză}\ \frac{}{\Gamma \vdash \varphi_1}\ \varphi_1 \in \Gamma.$$

Mai jos avem un exemplu de regulă de inferență incorectă (într-un sens pe care îl vom preciza mai târziu, dar care poate fi deja intuit):

$$\text{REGULĂ INCORECTĂ}\ \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2).}$$

Observație. *Ipotezele regulii de inferență, precum și concluzia, sunt de fapt* scheme *de secvențe și nu secvențe propriu-zise. Aceste scheme pot fi instanțiate, adică o regulă de inferență (prezentată ca mai sus) are mai multe* instanțe, *obținute prin înlocuirea variabilelor matematice $\varphi, \varphi', \Gamma$ cu formele concrete. De exemplu, iată două instanțe ale regulii $\wedge i$ de mai sus:*

$$\wedge i\ \frac{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(a)} \qquad \{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{Q(a)}}{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)});}$$

$$\wedge i\ \frac{\{\mathtt{P(a)}, \mathtt{Q(a)}, \mathtt{Q(b)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)}) \qquad \{\mathtt{P(a)}, \mathtt{Q(a)}, \mathtt{Q(b)}\} \vdash \mathtt{P(a)}}{\{\mathtt{P(a)}, \mathtt{Q(a)}, \mathtt{Q(b)}\} \vdash ((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \wedge \mathtt{P(a)}).}$$

În prima instanță, am înlocuit variabila matematică Γ cu mulțimea de formule $\{\mathtt{P(a)}, \mathtt{Q(a)}\}$, variabila matematică φ cu formula $\mathtt{P(a)}$ și variabila matematică φ' cu formula $\mathtt{Q(a)}$. Exercițiu: stabiliți singuri cu ce am înlocuit fiecare variabilă matematică în cea de-a doua instanță.

Iată un exemplu de regulă care nu este instanță a regulii $\wedge i$ (exercițiu: explicați de ce nu):

$$?\ \frac{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(a)} \qquad \{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{Q(a)}}{\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)});}$$

5.4 Sistem deductiv

Definiția 101. *Un* sistem deductiv *este o mulțime de reguli de inferență.*

Exemplul 102. *Fie sistemul deductiv D_1, format din următoarele patru reguli de inferență:*

$$\text{IPOTEZĂ}\ \frac{}{\Gamma \vdash \varphi,}\ \varphi \in \Gamma \qquad \wedge i\ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,}$$

$$\wedge e_2\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2.}$$

5.5 Demonstrație formală

Definiția 103 (Demonstrație formală)**.** *O* demonstrație formală *într-un sistem deductiv este o listă de secvențe*

1. S_1
2. S_2

...

n. S_n,

cu proprietatea că fiecare secvență S_i este justificată de o regulă de inferență a sistemului deductiv din secvențele anterioare $(S_1, \ldots, S_{i-1})$, în sensul în care S_i este concluzia unei instanțe a unei reguli de inferență din sistemul deductiv, regulă care folosește ca ipoteze doar secvențe alese dintre $S_1, \ldots, S_{i-1}$. În plus, dacă regula de inferență are condiție, această condiție trebuie să fie adevărată. Să notăm și faptul că orice prefix al unei demonstrații este tot o demonstrație.

Exemplul 104. *Iată un exemplu de demonstrație formală în sistemul D_1 introdus mai sus:*

1. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(a)}$; *(*Ipoteză*)*
2. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{Q(a)}$; *(*Ipoteză*)*
3. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$; *($\wedge i$, 1, 2)*
4. $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{Q(a)} \wedge (\mathtt{P(a)} \wedge \mathtt{Q(a)}))$. *($\wedge i$, 2, 3)*

Ca și în cazul logicii propoziționale, fiecare linie este adnotată cu numele regulii de inferență aplicate, plus liniile la care se găsesc ipotezele necesare aplicării (în aceeași ordine folosită pentru prezentarea sistemului deductiv).

Observație. *Definiția demonstrației formale în logica de ordinul întâi este aceeași ca în cazul logicii propoziționale. Totuși, vom vedea mai târziu că pentru aplicarea regulilor de inferență noi, asociate cuantificatorilor, vom folosi adnotări suplimentare.*

Definiția 105 (Secvență validă)**.** *O secvență $\Gamma \vdash \varphi$ este* validă într-un sistem deductiv *D dacă există o demonstrație formală $S_1, \ldots, S_n$ în D astfel încât $S_n = \Gamma \vdash \varphi$.*

Exemplul 106. *Secvența $\{\mathtt{P(a)}, \mathtt{Q(a)}\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$ este validă în sistemul deductiv D_1 de mai sus, deoarece este ultima secvență din următoarea demonstrație formală:*

1. $\{\mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash \mathtt{P}(\mathtt{a})$; *(Ipoteză)*
2. $\{\mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash \mathtt{Q}(\mathtt{a})$; *(Ipoteză)*
3. $\{\mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$; *($\wedge i$, 1, 2)*

Observație. *Atenție! Nu confundați noțiunea de* secvență validă într-un sistem deductiv *cu noțiunea de* formulă validă.

5.6 Deducția naturală

Deducția naturală este un sistem deductiv pentru logica de ordinul I. De fapt, sistemul deductiv pentru logica de ordinul I include toate regulile de deducție studiate la logica propozițională. În plus, pentru logica de ordinul I mai apar reguli noi, și anume cele de introducere și eliminare a cuantificatorilor. În această secțiune vom prezenta în detaliu fiecare regulă de inferență care aparține deducției naturale în logica de ordinul I.

5.6.1 Regulile pentru conjuncții

Am văzut deja regulile de introducere și de eliminare pentru conectorul "și":

$$\wedge i\ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,} \qquad \wedge e_2\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2.}$$

Regulile de inferență mimează raționamentul uman, bazat în esență pe o semantică intuitivă a noțiunii de adevăr:

1. Regula de introducere a conectorului $\wedge$ ne indică că putem demonstra o conjuncție $(\varphi_1 \wedge \varphi_2)$ din ipotezele Γ dacă știm deja că fiecare parte a conjuncției, φ_1 și respectiv φ_2, sunt consecințe ale ipotezelor Γ.

   Cu alte cuvinte, pentru a arăta o conjuncție dintr-un set de ipoteze, este suficient să stabilim individual că fiecare parte a conjuncției este o consecință a ipotezelor.

2. Pentru conectorul $\wedge$ avem două reguli de eliminare. Prima regulă de eliminare a conectorului $\wedge$ ne precizează că dacă am stabilit deja că o conjuncție $(\varphi_1 \wedge \varphi_2)$ este consecința unei mulțimi Γ de ipoteze, atunci și partea stângă a conjuncției, φ_1, este consecință a mulțimii Γ.

   A doua regulă este simetrică față de prima și ne permite să concluzionăm că și partea dreaptă a unei conjuncții este consecința unei mulțimi de formule dacă și conjuncția este consecința a aceleiași mulțimi de formule.

Iată un exemplu de demonstrație formală care utilizează regulile de inferență pentru conectorul $\wedge$:

1. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$; (Ipoteză)
2. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$; (Ipoteză)
3. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \mathtt{P}(\mathtt{a})$; ($\wedge e_1$, 1)
4. $\{(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$. ($\wedge i$, 3, 2)

5.6.2 Regulile pentru implicații

Regula de eliminare a implicației, numită și *modus ponens* în latină, este una dintre cele mai importante reguli de inferență pe care le aplicăm.

$$\rightarrow e \ \frac{\Gamma \vdash (\varphi_1 \rightarrow \varphi_2) \qquad \Gamma \vdash \varphi_1}{\Gamma \vdash \varphi_2}$$

Regula ne arată că, presupunând că am demonstrat $(\varphi_1 \rightarrow \varphi_2)$ (din Γ) și în plus am demonstrat și φ_1 (tot din Γ), atunci putem demonstra φ_2 (din Γ).

Iată un exemplu de demonstrație formală care folosește regula de eliminare a implicației:

1. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$; (Ipoteză)
2. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \mathtt{P}(\mathtt{a})$; ($\wedge e_1$, 1)
3. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$; (Ipoteză)
4. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$. ($\rightarrow e$, 3, 1)

Această demonstrație arată că secvența $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$ este validă, adică formula $\forall \mathtt{x}.\mathtt{P}(\mathtt{x})$ este o consecință a mulțimii de formule $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\}$. Observați ordinea în care apar liniile 3 și 1 în explicația liniei 4: urmează aceeași ordine, fixată prin regula de inferență.

Exercițiul 107. *Arătați că sunt valide următoarele secvențe:*

1. $\{((\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), \mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$;
2. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), \mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{Q}(\mathtt{a}) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$.

Regula de introducere a implicației este mai subtilă. Pentru a arăta că o implicație $(\varphi_1 \to \varphi_2)$ decurge din Γ, *presupunem* φ_1 (în plus față de Γ) și arătăm φ_2. Regula poate fi scrisă în două moduri echivalente, care se deosebesc doar prin faptul că prima regulă folosește convenția de notație referitoare la acoladele din jurul premiselor unei secvențe, în timp ce în a doua regulă acoladele care marchează mulțimea apar explicit:

$$\to i\ \frac{\Gamma, \varphi_1 \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \to \varphi_2),} \qquad \to i\ \frac{\Gamma \cup \{\varphi_1\} \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \to \varphi_2).}$$

Ceea ce este important de observat și de înțeles la regula de introducere a implicației este că mulțimea de premise se schimbă. Dacă în concluzie avem că formula $(\varphi_1 \to \varphi_2)$ decurge din Γ, în ipoteză trebuie să arătăm că φ_2 decurge din premisele $\Gamma \cup \{\varphi_1\}$. Cu alte cuvinte, la modul intuitiv, pentru a demonstra o implicație $(\varphi_1 \to \varphi_2)$, presupunem antecedentul φ_1 și arătăm consecventul φ_2.

Exemplul 108. *Să arătăm că secvența* $\{\} \vdash (\mathtt{P(a)} \to \mathtt{P(a)})$ *este validă:*

1. $\{\mathtt{P(a)}\} \vdash \mathtt{P(a)}$; *(Ipoteză)*
2. $\{\} \vdash (\mathtt{P(a)} \to \mathtt{P(a)})$. *($\to i$, 1)*

Exemplul 109. *Să arătăm că secvența* $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)})\} \vdash (\mathtt{P(a)} \to \mathtt{P(b)})$ *este validă:*

1. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash (\mathtt{P(a)} \to \mathtt{Q(a)})$; *(Ipoteză)*
2. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash \mathtt{P(a)}$; *(Ipoteză)*
3. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash \mathtt{Q(a)}$; *($\to e$, 1, 2)*
4. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash (\mathtt{Q(a)} \to \mathtt{P(b)})$; *(Ipoteză)*
5. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)}), \mathtt{P(a)}\} \vdash \mathtt{P(b)}$; *($\to e$, 4, 3)*
6. $\{(\mathtt{P(a)} \to \mathtt{Q(a)}), (\mathtt{Q(a)} \to \mathtt{P(b)})\} \vdash (\mathtt{P(a)} \to \mathtt{P(b)})$. *($\to i$, 5)*

Exercițiul 110. *Arătați că următoarele secvențe sunt valide:*

1. $\{((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \to \mathtt{P(b)}), \mathtt{P(a)}, \mathtt{Q(a)}\} \vdash \mathtt{P(b)}$;
2. $\{((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \to \mathtt{P(b)})\} \vdash (\mathtt{P(a)} \to (\mathtt{Q(a)} \to \mathtt{P(b)}))$;
3. $\{(\mathtt{P(a)} \to (\mathtt{Q(a)} \to \mathtt{P(b)}))\} \vdash ((\mathtt{P(a)} \wedge \mathtt{Q(a)}) \to \mathtt{P(b)})$.

5.6.3 Regulile pentru disjuncții

Conectorul $\vee$ are două reguli de introducere:

$$\vee i_1 \frac{\Gamma \vdash \varphi_1}{\Gamma \vdash (\varphi_1 \vee \varphi_2),} \qquad \vee i_2 \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \vee \varphi_2).}$$

Prima regulă ne arată că dacă știm φ_1 (din Γ), atunci știm și $(\varphi_1 \vee \varphi_2)$ (din Γ), indiferent de φ_2. A doua regulă de eliminare este simetrică, pentru partea dreaptă a disjuncției.

Exemplul 111. *Să arătăm că secvența* $\{(\mathtt{P(a)} \wedge \mathtt{Q(a)})\} \vdash (\mathtt{P(a)} \vee \mathtt{Q(a)})$ *este validă:*

1. $\{(\mathtt{P(a)} \wedge \mathtt{Q(a)})\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$; *(*Ipoteză*)*
2. $\{(\mathtt{P(a)} \wedge \mathtt{Q(a)})\} \vdash \mathtt{P(a)}$; *($\wedge e_1$, 1)*
3. $\{(\mathtt{P(a)} \wedge \mathtt{Q(a)})\} \vdash (\mathtt{P(a)} \vee \mathtt{Q(a)})$. *($\vee i_1$, 2)*

O altă demonstrație formală pentru aceeași secvență este:

1. $\{(\mathtt{P(a)} \wedge \mathtt{Q(a)})\} \vdash (\mathtt{P(a)} \wedge \mathtt{Q(a)})$; *(*Ipoteză*)*
2. $\{(\mathtt{P(a)} \wedge \mathtt{Q(a)})\} \vdash \mathtt{Q(a)}$; *($\wedge e_2$, 1)*
3. $\{(\mathtt{P(a)} \wedge \mathtt{Q(a)})\} \vdash (\mathtt{P(a)} \vee \mathtt{Q(a)})$. *($\vee i_2$, 2)*

Regula de eliminare a disjuncției este ușor mai complicată, fiind o altă regulă în care mulțimea de premise variază de la ipoteză la concluzie:

$$\vee e \frac{\Gamma \vdash (\varphi_1 \vee \varphi_2) \qquad \Gamma, \varphi_1 \vdash \varphi' \qquad \Gamma, \varphi_2 \vdash \varphi'}{\Gamma \vdash \varphi'}$$

Prima ipoteză a regulii, $\Gamma \vdash (\varphi_1 \vee \varphi_2)$, este ușor de înțeles: pentru a "elimina" o disjuncție, trebuie să avem o disjuncție printre ipoteze (disjuncție pe care să o "eliminăm"). Ultimele două ipoteze ale regulii de eliminare a disjuncției trebuie înțelese intuitiv după cum urmează. Din prima ipoteză știm $(\varphi_1 \vee \varphi_2)$ (din Γ); cu alte cuvinte, măcar una dintre formulele φ_1 și respectiv φ_2 decurge din Γ. Ipotezele 2 și 3 ne indică faptul că, indiferent care dintre formulele φ_1 și respectiv φ_2 ar avea loc, în orice caz φ' are loc. Adică dacă presupunem φ_1 (în plus față de Γ), φ' are loc, iar dacă presupunem φ_2 (în plus față de Γ), φ' tot are loc. Și atunci concluzia ne indică că φ' are loc indiferent care dintre φ_1 și respectiv φ_2 ar avea loc.

Exemplul 112. *Să arătăm că secvența* $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$ *este validă:*

1. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash \mathtt{P}(\mathtt{a})$; *(Ipoteză)*
2. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{P}(\mathtt{a})\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$; *($\vee i_2$, 1)*
3. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{Q}(\mathtt{a})\} \vdash \mathtt{Q}(\mathtt{a})$; *(Ipoteză)*
4. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$; *($\vee i_1$, 1)*
5. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))$; *(Ipoteză)*
6. $\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{Q}(\mathtt{a}) \vee \mathtt{P}(\mathtt{a}))$. *($\vee e$, 5, 2, 4)*

Observați cu atenție modul în care mulțimea de premise variază de la o secvență la alta pe parcursul demonstrației formale, respectând regulile de inferență.

Exercițiul 113. *Găsiți o demonstrație formală pentru secvența*

$$\{(\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})), (\mathtt{P}(\mathtt{a}) \to \mathtt{P}(\mathtt{b})), (\mathtt{Q}(\mathtt{a}) \to \mathtt{P}(\mathtt{b}))\} \vdash \mathtt{P}(\mathtt{b}).$$

5.6.4 Regulile pentru negații

Regulile pentru introducerea și respectiv eliminarea negației sunt prezentate împreună cu o regulă pentru eliminarea lui $\bot$:

$$\neg i\ \frac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi} \qquad \neg e\ \frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot} \qquad \bot e\ \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi}$$

Să ne readucem aminte că $\bot$ este un conector logic de aritate 0. Cu alte cuvinte, conectorul $\bot$ este de sine stătător o formulă. Semantica formulei $\bot$ este că este falsă în orice atribuire. Cu alte cuvinte, $\bot$ este o contradicție.

Prima regulă dintre cele de mai sus, cea de introducere a negației, este ușor de explicat intuitiv: cum putem arăta că o formulă de forma $\neg\varphi$ decurge din premisele Γ? Presupunem, în plus față de premisele Γ, că avem φ și arătăm că din Γ și φ decurge o contradicție ($\Gamma, \varphi \vdash \bot$). În acest fel, arătăm că $\neg\varphi$ decurge din Γ.

A doua regulă, pentru eliminarea negației, ne indică faptul că dacă atât o formulă φ, cât și negația sa, $\neg\varphi$, decurg din aceeași mulțime de premise Γ, atunci din Γ decurge și o contradicție, $\bot$. O mulțime Γ din care decurge o contradicție se numește și mulțime *inconsistentă* de formule.

A treia regulă indică că, dacă Γ este o mulțime inconsistentă de formule, atunci orice formulă φ decurge din Γ.

Nu există nicio regulă pentru introducerea conectorului $\bot$ (sau, regula de eliminare a negației se poate considera ca fiind și regula de introducere a lui $\bot$).

Exemplul 114. *Să arătăm că secvența* $\{\mathtt{P(a)}\} \vdash \neg\neg\mathtt{P(a)}$ *este validă:*

1. $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \mathtt{P(a)}$; *(Ipoteză)*
2. $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \neg\mathtt{P(a)}$; *(Ipoteză)*
3. $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \bot$; *($\neg e$, 1, 2)*
4. $\{\mathtt{P(a)}\} \vdash \neg\neg\mathtt{P(a)}$. *($\neg i$, 3)*

Exemplul 115. *Să arătăm că secvența* $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \mathtt{P(b)}$ *este validă:*

1. $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \mathtt{P(a)}$; *(Ipoteză)*
2. $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \neg\mathtt{P(a)}$; *(Ipoteză)*
3. $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \bot$; *($\neg e$, 1, 2)*
4. $\{\mathtt{P(a)}, \neg\mathtt{P(a)}\} \vdash \mathtt{P(b)}$. *($\bot e$, 3)*

Eliminarea dublei negații

La logica propozițională am întâlnit și următoarea regulă pentru eliminarea dublei negații:

$$\neg\neg e \ \frac{\Gamma \vdash \neg\neg\varphi}{\Gamma \vdash \varphi}$$

Exemplul 116. *Să arătăm că secvența* $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}\} \vdash \mathtt{P(a)}$ *este validă:*

1. $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}, \neg\mathtt{P(a)}\} \vdash \neg\mathtt{P(a)}$; *(Ipoteză)*
2. $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}, \neg\mathtt{P(a)}\} \vdash (\neg\mathtt{P(a)} \to \mathtt{Q(a)})$; *(Ipoteză)*
3. $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}, \neg\mathtt{P(a)}\} \vdash \mathtt{Q(a)}$; *($\to e$, 2, 1)*
4. $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}, \neg\mathtt{P(a)}\} \vdash \neg\mathtt{Q(a)}$; *(Ipoteză)*
5. $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}, \neg\mathtt{P(a)}\} \vdash \bot$; *($\neg i$, 4, 3)*
6. $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}\} \vdash \neg\neg\mathtt{P(a)}$; *($\neg i$, 5)*
7. $\{(\neg\mathtt{P(a)} \to \mathtt{Q(a)}), \neg\mathtt{Q(a)}\} \vdash \mathtt{P(a)}$. *($\neg\neg e$, 6)*

Exemplul 117. *Să arătăm că secvența* $\{\} \vdash (\mathtt{P(a)} \vee \neg \mathtt{P(a)})$ *este validă:*

1. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)}), \mathtt{P(a)}\} \vdash \neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)})$; *(Ipoteză)*
2. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)}), \mathtt{P(a)}\} \vdash \mathtt{P(a)}$; *(Ipoteză)*
3. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)}), \mathtt{P(a)}\} \vdash (\mathtt{P(a)} \vee \neg \mathtt{P(a)})$; *$(\vee i_1$, 2)*
4. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)}), \mathtt{P(a)}\} \vdash \bot$; *$(\neg e$, 1, 3)*
5. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)})\} \vdash \neg \mathtt{P(a)}$; *$(\neg i$, 4)*
6. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)})\} \vdash (\mathtt{P(a)} \vee \neg \mathtt{P(a)})$; *$(\vee i_2$, 5)*
7. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)})\} \vdash \neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)})$; *(Ipoteză)*
8. $\{\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)})\} \vdash \bot$; *$(\neg e$, 7, 6)*
9. $\{\} \vdash \neg\neg(\mathtt{P(a)} \vee \neg \mathtt{P(a)})$; *$(\neg i$, 8)*
10. $\{\} \vdash (\mathtt{P(a)} \vee \neg \mathtt{P(a)})$. *$(\neg\neg e$, 9)*

5.6.5 Eliminarea cuantificatorului universal.

Regula pentru eliminarea cuantificatorului universal este:

$$\forall e \ \frac{\Gamma \vdash (\forall x.\varphi)}{\Gamma \vdash \varphi[x \mapsto t]} \ vars(t) \cap bound(\varphi) = \emptyset$$

Regula de eliminare a cuantificatorului universal este foarte simplă: practic, dacă știm că $(\forall x.\varphi)$ este o consecință sintactică din Γ atunci putem instanția variabila legată x cu orice termen t cu condiția ca t să nu conțină variabile legate în φ.

Exercițiul 118. *Întrebare: regula de eliminare de mai sus mai are sens dacă x nu apare în φ? De exemplu, din $\Gamma \vdash (\forall x.\mathtt{P(a)})$ putem deduce că $\Gamma \vdash \mathtt{P(a)}[\mathtt{x} \mapsto \mathtt{b}]$?*

Exemplul 119. *Să ne amintim un exemplu discutat anterior în care aveam două afirmații:* Orice om este muritor *și* Socrate este om. *Putem trage concluzia că* Socrate este muritor*? Pentru a răspunde la întrebare, am putea încerca să demonstrăm secvența*

$$\{\forall \mathtt{x}.(\mathtt{Om(x)} \rightarrow \mathtt{Muritor(x)}), \mathtt{Om(s)}\} \vdash \mathtt{Muritor(s)},$$

unde `Om` *și* `Muritor` *sunt predicate de aritate 1 iar* `s` *este o constantă (simbol funcțional de aritate 0) asociată numelui Socrate. Iată demonstrația formală a secvenței:*

1. $\{\forall \mathtt{x}.(\mathtt{Om}(\mathtt{x}) \rightarrow \mathtt{Muritor}(\mathtt{x})), \mathtt{Om}(\mathtt{s})\} \vdash \forall \mathtt{x}.(\mathtt{Om}(\mathtt{x}) \rightarrow \mathtt{Muritor}(\mathtt{x}))$ (Ipoteză)
2. $\{\forall \mathtt{x}.(\mathtt{Om}(\mathtt{x}) \rightarrow \mathtt{Muritor}(\mathtt{x})), \mathtt{Om}(\mathtt{s})\} \vdash (\mathtt{Om}(\mathtt{s}) \rightarrow \mathtt{Muritor}(\mathtt{s}))$ $(\forall e, 1, \mathtt{s})$
3. $\{\forall \mathtt{x}.(\mathtt{Om}(\mathtt{x}) \rightarrow \mathtt{Muritor}(\mathtt{x})), \mathtt{Om}(\mathtt{s})\} \vdash \mathtt{Om}(\mathtt{s})$ (Ipoteză)
4. $\{\forall \mathtt{x}.(\mathtt{Om}(\mathtt{x}) \rightarrow \mathtt{Muritor}(\mathtt{x})), \mathtt{Om}(\mathtt{s})\} \vdash \mathtt{Muritor}(\mathtt{s})$ $(\rightarrow e, 2, 3)$

La pasul 2 al demonstrației am utilizat regula $\forall e$*, care instanțiază în formula* $\forall \mathtt{x}.(\mathtt{Om}(\mathtt{x}) \rightarrow \mathtt{Muritor}(\mathtt{x}))$ *variabila legată* $\mathtt{x}$ *cu* $\mathtt{s}$*:* $(\mathtt{Om}(\mathtt{s}) \rightarrow \mathtt{Muritor}(\mathtt{s}))$*. În limbaj natural, am dedus prin raționament sintactic că dacă* Socrate este om *şi* Orice om este muritor *atunci* Socrate este muritor*.*

5.6.6 Introducerea cuantificatorul existențial.

Există o oarecare dualitate a regulilor pentru introducerea și eliminarea cuantificatorilor. Astfel, regula de introducere a cuantificatorului existențial de mai jos poate văzută ca duala regulii de eliminare a cuantificatorului universal:

$$\exists i \ \frac{\Gamma \vdash \varphi[x \mapsto t]}{\Gamma \vdash (\exists x.\varphi)} \ vars(t) \cap bound(\varphi) = \emptyset$$

Regula ne indică faptul că putem deduce $(\exists x.\varphi)$ atunci când $\varphi[x \mapsto t]$ este consecință semantică din Γ. Informal, dacă există un x concret – și anume t – astfel încât $\varphi[x \mapsto t]$ este adevărată (cu condiția ca t să nu conțină variabile legate în φ) vom trage concluzia că $(\exists x.\varphi)$ este adevărată. Aici, t joacă rolul unui *martor* pentru care formula din concluzie este adevărată.

Exemplul 120. *Să arătăm că secvența* $\{\mathtt{P}(\mathtt{a})\} \vdash \exists \mathtt{x}.\mathtt{P}(\mathtt{x})$ *este validă:*

1. $\{\mathtt{P}(\mathtt{a})\} \vdash \mathtt{P}(\mathtt{a})$ (Ipoteză)
2. $\{\mathtt{P}(\mathtt{a})\} \vdash \exists \mathtt{x}.\mathtt{P}(\mathtt{x})$ $(\exists i, 1)$

Observați că în acest caz, metavariabila φ *este* $\mathtt{P}(\mathtt{x})$ *din regula* $\exists_e$*, iar* $\varphi[x \mapsto a]$ *este* $\mathtt{P}(\mathtt{x})[\mathtt{x} \mapsto \mathtt{a}]$*, adică* $\mathtt{P}(\mathtt{a})$*.*

Exemplul 121. *Să arătăm că secvența* $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \rightarrow \mathtt{Q}(\mathtt{x})), \mathtt{P}(\mathtt{a})\} \vdash \exists \mathtt{x}.\mathtt{Q}(\mathtt{x})$ *este validă:*

1. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \rightarrow \mathtt{Q}(\mathtt{x})), \mathtt{P}(\mathtt{a})\} \vdash \forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \rightarrow \mathtt{Q}(\mathtt{x}))$ (Ipoteză)
2. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \rightarrow \mathtt{Q}(\mathtt{x})), \mathtt{P}(\mathtt{a})\} \vdash \mathtt{P}(\mathtt{a})$ (Ipoteză)
3. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \rightarrow \mathtt{Q}(\mathtt{x})), \mathtt{P}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \rightarrow \mathtt{Q}(\mathtt{a}))$ $(\forall e, 1, \mathtt{a})$
4. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \rightarrow \mathtt{Q}(\mathtt{x})), \mathtt{P}(\mathtt{a})\} \vdash \mathtt{Q}(\mathtt{a})$ $(\rightarrow e, 3, 2)$
5. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \rightarrow \mathtt{Q}(\mathtt{x})), \mathtt{P}(\mathtt{a})\} \vdash \exists \mathtt{x}.\mathtt{Q}(\mathtt{x})$ $(\exists i, 4)$

5.6.7 Introducerea cuantificatorului universal.

Regula de introducere a cuantificatorului universal este:

$$\forall i \ \frac{\Gamma \vdash \varphi[x \mapsto x_0]}{\Gamma \vdash (\forall x.\varphi)} \ x_0 \notin vars(\Gamma, \varphi)$$

Regula de mai sus ne spune că vom putea deriva concluzia $\Gamma \vdash (\forall x.\varphi)$ dacă vom arăta că $\varphi[x \mapsto x_0]$ este consecință sintactică din Γ, unde x_0 este o variabilă *nouă*: ea nu mai apare în alte formule și asupra ei nu avem nici o constrângere.

Exemplul 122. *Să arătăm că secvența* $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \forall \mathtt{x}.\mathtt{Q}(\mathtt{x})$ *este validă:*

1. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x}))$ (Ipoteză)
2. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$ (Ipoteză)
3. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash (\mathtt{P}(\mathtt{x}_0) \to \mathtt{Q}(\mathtt{x}_0))$ $(\forall e, 1, \mathtt{x}_0)$
4. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \mathtt{P}(\mathtt{x}_0)$ $(\forall e, 2, \mathtt{x}_0)$
5. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \mathtt{Q}(\mathtt{x}_0)$ $(\to e, 3, 4)$
6. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \to \mathtt{Q}(\mathtt{x})), \forall \mathtt{x}.\mathtt{P}(\mathtt{x})\} \vdash \forall \mathtt{x}.\mathtt{Q}(\mathtt{x})$ $(\forall i, 5)$

Observați că pentru secvențele 3, 4 și 5 utilizăm variabila nouă $\mathtt{x}_0$*. La nivel intuitiv,* $\mathtt{Q}(\mathtt{x}_0)$ *va avea loc pentru orice* $\mathtt{x}_0$*.*

Exercițiul 123. *Arătați că următoarele secvențe sunt valide:*

1. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \wedge \mathtt{Q}(\mathtt{x}))\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$;
2. $\{\forall \mathtt{x}.\mathtt{Q}(\mathtt{x}), \mathtt{P}(\mathtt{a})\} \vdash \mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})$;
3. $\{\forall \mathtt{x}.\mathtt{P}(\mathtt{x}), \forall \mathtt{x}.\mathtt{Q}(\mathtt{x})\} \vdash \forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \wedge \mathtt{Q}(\mathtt{x}))$.

5.6.8 Eliminarea cuantificatorului existențial

Regula pentru eliminarea cuantificatorului existențial este următoarea:

$$\exists e \ \frac{\Gamma \vdash (\exists x.\varphi) \qquad \Gamma \cup \{\varphi[x \mapsto x_0]\} \vdash \psi}{\Gamma \vdash \psi} \ x_0 \notin vars(\Gamma, \varphi, \psi)$$

Prima ipoteză a regulii este $\Gamma \vdash (\exists x.\varphi)$, care, la nivel intuitiv, ne asigură că există cel puțin un termen (pot fi mai mulți) care îl poate înlocui x astfel încât φ este consecință sintactică din Γ. Nu știm însă care sunt acești termeni (în cazul în care sunt mai mulți). Știm doar că măcar unul există și îi vom nota generic cu x_0. Pentru a demonstra concluzia, adică ψ este consecință sintactică din Γ, va trebui să facem o analiză de cazuri după toți x_0. Practic, acest lucru este sumarizat de cea de-a doua ipoteză a regulii unde trebuie arătat că ψ este consecință sintactică din $\Gamma \cup \{\varphi[x \mapsto x_0]\}$.

Exemplul 124. *Să arătăm că secvența* $\{\forall x.(P(x) \to Q(x)), \exists x.P(x)\} \vdash \exists x.Q(x)$ *este validă:*

1. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x)\} \vdash \exists x.P(x)$ (Ipoteză)
2. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash P(x_0)$ (Ipoteză)
3. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash \forall x.(P(x) \to Q(x))$ (Ipoteză)
4. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash (P(x_0) \to Q(x_0))$ $(\forall e, 3, x_0)$
5. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash Q(x_0)$ $(\to e, 4, 2)$
6. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x), P(x_0)\} \vdash \exists x.Q(x)$ $(\exists i, 5)$
7. $\{\forall x.(P(x) \to Q(x)), \exists x.P(x)\} \vdash \exists x.Q(x)$ $(\exists e, 1, 6)$

Observați că pentru a demonstra secvența 7, am utilizat secvența 1 și secvența 6. Aceasta din urmă a fost demonstrată de pașii 2, 3, 4 și 5, unde am utilizat ca ipoteză și formula $P(x_0)(= P(x)[x \mapsto x_0])$.

5.6.9 Alte reguli

O altă regulă utilă, care nu ține de un anumit conector, este regula de extindere, care a fost prezentă și în capitolul dedicat logicii propoziționale:

$$\textsc{Extindere}\ \frac{\Gamma \vdash \varphi}{\Gamma, \varphi' \vdash \varphi}$$

Această regulă ne indică faptul că, dacă φ este consecință a unei mulțimi de formule Γ, atunci φ este consecință și a mulțimii $\Gamma \cup \{\varphi'\}$ (indiferent de φ'). Cu alte cuvinte, putem extinde oricând mulțimea de premise ale unei secvențe valide și obținem o nouă secvență validă.

Exemplul 125. *Arătăm că secvența* $\{P(a), \neg Q(a), P(f(a)), (P(b) \wedge Q(b))\} \vdash \neg\neg P(a)$ *este validă:*

1. $\{P(a), \neg P(a)\} \vdash P(a)$; *(Ipoteză)*

2. $\{P(a), \neg P(a)\} \vdash \neg P(a)$; *(Ipoteză)*

3. $\{P(a), \neg P(a)\} \vdash \bot$; *($\neg e$, 1, 2)*

4. $\{P(a)\} \vdash \neg\neg P(a)$; *($\neg i$, 3)*

5. $\{P(a), \neg Q(a)\} \vdash \neg\neg P(a)$; *(Extindere, 4)*

6. $\{P(a), \neg Q(a), P(f(a))\} \vdash \neg\neg P(a)$; *(Extindere, 5)*

7. $\{P(a), \neg Q(a), P(f(a)), (P(b) \wedge Q(b))\} \vdash \neg\neg P(a)$. *(Extindere, 6)*

5.7 Sistemul deductiv al deducției naturale

Deducția naturală pentru logica de ordinul I este sistemul deductiv alcătuit din toate regulile din secțiunile precedente. Iată aici sumarizate toate regulile:

$$\wedge i\ \frac{\Gamma \vdash \varphi_1 \qquad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \qquad \wedge e_1\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,}$$

$$\wedge e_2\ \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2,} \qquad \to e\ \frac{\Gamma \vdash (\varphi_1 \to \varphi_2) \qquad \Gamma \vdash \varphi_1}{\Gamma \vdash \varphi_2,}$$

$$\to i\ \frac{\Gamma, \varphi_1 \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \to \varphi_2),} \qquad \vee i_1\ \frac{\Gamma \vdash \varphi_1}{\Gamma \vdash (\varphi_1 \vee \varphi_2),} \qquad \vee i_2\ \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \vee \varphi_2),}$$

$$\vee e\ \frac{\Gamma \vdash (\varphi_1 \vee \varphi_2) \qquad \Gamma, \varphi_1 \vdash \varphi' \qquad \Gamma, \varphi_2 \vdash \varphi'}{\Gamma \vdash \varphi',}$$

$$\neg e\ \frac{\Gamma \vdash \varphi \qquad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot,} \qquad \neg i\ \frac{\Gamma, \varphi \vdash \bot}{\Gamma \vdash \neg\varphi,} \qquad \bot e\ \frac{\Gamma \vdash \bot}{\Gamma \vdash \varphi,}$$

$$\textsc{Ipoteză}\ \frac{}{\Gamma \vdash \varphi}\ \varphi \in \Gamma, \qquad \textsc{Extindere}\ \frac{\Gamma \vdash \varphi}{\Gamma, \varphi' \vdash \varphi,} \qquad \neg\neg e\ \frac{\Gamma \vdash \neg\neg\varphi}{\Gamma \vdash \varphi.}$$

$$\forall e\ \frac{\Gamma \vdash (\forall x.\varphi)}{\Gamma \vdash \varphi[x \mapsto t]}\ vars(t) \cap bound(\varphi) = \emptyset$$

$$\exists i\ \frac{\Gamma \vdash \varphi[x \mapsto t]}{\Gamma \vdash (\exists x.\varphi)}\ vars(t) \cap bound(\varphi) = \emptyset$$

$$\forall i\ \frac{\Gamma \vdash \varphi[x \mapsto x_0]}{\Gamma \vdash (\forall x.\varphi)}\ x_0 \notin vars(\Gamma, \varphi)$$

$$\exists e\ \frac{\Gamma \vdash (\exists x.\varphi) \qquad \Gamma \cup \{\varphi[x \mapsto x_0]\} \vdash \psi}{\Gamma \vdash \psi}\ x_0 \notin vars(\Gamma, \varphi, \psi)$$

Desigur că putem folosi în demonstrații și regulile derivate (pe care le-am prezentat în cazul logicii propoziționale).

5.8 Corectitudinea și completitudinea deducției naturale pentru logica de ordinul I

Teorema 126 (Corectitudinea deducției naturale)**.** *Pentru orice mulțime de formule Γ și orice formulă φ, dacă secvența $\Gamma \vdash \varphi$ este validă, atunci $\Gamma \models \varphi$.*

Exercițiul 127. *Demonstrați Teorema 126.*

Teorema 128 (Completitudinea deducției naturale)**.** *Pentru orice mulțime de formule Γ și orice formulă φ, dacă $\Gamma \models \varphi$ atunci secvența $\Gamma \vdash \varphi$ este validă.*

Demonstrația teoremei de completitudine depășește nivelul cursului.

Observație. *De remarcat că, folosind teoremele de corectitudine și respectiv de completitudine, relația $\vdash$ coincide cu relația $\models$, deși au definiții cu totul diferite.*

5.9 Fișă de exerciții

Exercițiul 129. *Arătați că următoarele secvențe sunt valide:*

1. $\{((\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))\} \vdash (\mathtt{Q}(\mathtt{a}) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$;
2. $\{((\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), \forall \mathtt{x}.\mathtt{Q}(\mathtt{x})\} \vdash (\forall \mathtt{x}.\mathtt{Q}(\mathtt{x}) \wedge \mathtt{Q}(\mathtt{a}))$;
3. $\{((\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))\} \vdash (\forall \mathtt{x}.\mathtt{P}(\mathtt{x}) \wedge (\mathtt{Q}(\mathtt{a}) \wedge \mathtt{P}(\mathtt{a})))$;
4. $\{((\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a})) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), \mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$;
5. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \forall \mathtt{x}.\mathtt{P}(\mathtt{x})), \mathtt{P}(\mathtt{a}), \mathtt{Q}(\mathtt{a})\} \vdash (\mathtt{Q}(\mathtt{a}) \wedge \forall \mathtt{x}.\mathtt{P}(\mathtt{x}))$;
6. $\{(\mathtt{P}(\mathtt{a}) \rightarrow \mathtt{P}(\mathtt{b})), (\mathtt{Q}(\mathtt{a}) \rightarrow \mathtt{P}(\mathtt{b}))\} \vdash ((\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a})) \rightarrow \mathtt{P}(\mathtt{b}))$;
7. $\{\neg(\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))\} \vdash (\neg\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{Q}(\mathtt{a}))$;
8. $\{\neg(\neg\mathtt{P}(\mathtt{a}) \vee \neg\mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$;
9. $\{\neg(\neg\mathtt{P}(\mathtt{a}) \wedge \neg\mathtt{Q}(\mathtt{a}))\} \vdash (\mathtt{P}(\mathtt{a}) \vee \mathtt{Q}(\mathtt{a}))$;

Exercițiul 130. *Stabiliți care dintre secvențele de mai jos sunt valide:*

1. $\{\forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \wedge \mathtt{Q}(\mathtt{x}))\} \vdash \forall \mathtt{x}.\mathtt{P}(\mathtt{x})$;
2. $\{\forall \mathtt{x}.\mathtt{Q}(\mathtt{x}), \mathtt{P}(\mathtt{a})\} \vdash (\mathtt{P}(\mathtt{a}) \wedge \mathtt{Q}(\mathtt{a}))$;
3. $\{\forall \mathtt{x}.\mathtt{P}(\mathtt{x}), \forall \mathtt{x}.\mathtt{Q}(\mathtt{x})\} \vdash \forall \mathtt{x}.(\mathtt{P}(\mathtt{x}) \wedge \mathtt{Q}(\mathtt{x}))$;
4. $\{\exists \mathtt{x}.\exists \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{y})\} \vdash \exists \mathtt{y}.\exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{y})$;
5. $\{\exists \mathtt{x}.\forall \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{y})\} \vdash \forall \mathtt{y}.\exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{y})$; *Dar invers:* $\{\forall \mathtt{y}.\exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{y})\} \vdash \exists \mathtt{x}.\forall \mathtt{y}.\mathtt{P}(\mathtt{x}, \mathtt{y})$*?*
6. $\{\neg(\exists \mathtt{x}.\mathtt{P}(\mathtt{x}))\} \vdash \forall \mathtt{x}.\neg\mathtt{P}(\mathtt{x})$;
7. $\{\forall \mathtt{x}.\neg\mathtt{P}(\mathtt{x})\} \vdash \neg(\exists \mathtt{x}.\mathtt{P}(\mathtt{x}))$;

Capitolul 6

Forme normale ale formulelor de ordinul I

În acest capitol vom defini noţiunile de formule echivalente și cea de formă normală Prenex (FNP), formă normală Skolem (FNS) și formă normală Skolem Clauzală (FNSC) corespunzătoare unei formule din logica de ordinul I.

6.1 Formule echivalente

În diverse contexte, anumite formule pot avea același înțeles. De exemplu, formulele $(\forall \mathtt{x}.\mathtt{P}(\mathtt{x},\mathtt{x}))$ și $(\forall \mathtt{y}.\mathtt{P}(\mathtt{y},\mathtt{y}))$ au același înțeles în orice context. Un alt exemplu de formule cu același înțeles este $\neg(\forall \mathtt{x}.\mathtt{Q}(\mathtt{x}))$ și $(\exists \mathtt{x}.\neg\mathtt{Q}(\mathtt{x}))$. 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 $\mathtt{P}$ este interpretat printr-un predicat simetric, formulele $\mathtt{P}(\mathtt{x},\mathtt{y})$ și respectiv $\mathtt{P}(\mathtt{y},\mathtt{x})$ au același înțeles. Astfel de formule se numesc *echivalente în structura respectivă*.

Acest aspect este surprins de următoarele definiții.

Definiția 131. *Două formule $\varphi_1 \in \mathbb{LPI}$ și $\varphi_2 \in \mathbb{LPI}$ 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 $\varphi_1 \overset{S}{\equiv} \varphi_2$.

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.

Exemplul 132. *Continuăm exemplele din cursurile anterioare. Considerăm signatura* $\Sigma = (\{\mathtt{P}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{e}\})$ *și* Σ*-structura* $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$.

1. *Avem că* $\mathtt{P}(\mathtt{x}, \mathtt{y}) \overset{S_1}{\equiv} \mathtt{P}(\mathtt{y}, \mathtt{x})$. *De ce?*
   Fie α *o* S_1*-atribuire oarecare.*

   Avem $S_1, \alpha \models \mathtt{P}(\mathtt{x}, \mathtt{y})$ *ddacă (prin definiția relației* $\models$*)* $P^{S_1}(\bar{\alpha}(\mathtt{x}), \bar{\alpha}(\mathtt{y}))$
   ddacă $\bar{\alpha}(\mathtt{x}) = \bar{\alpha}(\mathtt{y})$
   ddacă $\alpha(\mathtt{x}) = \alpha(\mathtt{y})$
   ddacă (prin simetria relației de egalitate) $\alpha(\mathtt{y}) = \alpha(\mathtt{y})$
   ddacă $\bar{\alpha}(\mathtt{y}) = \bar{\alpha}(\mathtt{x})$
   ddacă (prin definiția relației $\models$*)* $S_1, \alpha \models \mathtt{P}(\mathtt{y}, \mathtt{x})$.

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

2. *Avem că* $\mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \overset{S_1}{\not\equiv} \mathtt{P}(\mathtt{x}_2, \mathtt{x}_3)$. *De ce?*
   Deoarece există o S_1*-atribuire* $\alpha : \mathcal{X} \to \mathbb{Z}$*, definită prin* $\alpha(\mathtt{x}_1) = 42, \alpha(\mathtt{x}_2) = 7, \alpha(\mathtt{x}_3) = 42$ *(pentru restul variabilelor nu este relevantă valoarea lor în atribuire) cu proprietatea că*

   $$S_1, \alpha \models \mathtt{P}(\mathtt{x}_1, \mathtt{x}_3) \text{ (deoarece } 42 = 42\text{), dar}$$
   $$S_1, \alpha \not\models \mathtt{P}(\mathtt{x}_2, \mathtt{x}_3) \text{ (deoarece } 42 \neq 7\text{).}$$

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

Definiția 133. *Două formule* $\varphi_1 \in \mathbb{LPI}$ *și* $\varphi_2 \in \mathbb{LPI}$ 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 134. *Continuăm exemplul anterior. Avem că* $\mathtt{P}(\mathtt{x}, \mathtt{y}) \not\equiv \mathtt{P}(\mathtt{y}, \mathtt{x})$. *De ce?*
Deoarece există o Σ*-structură și o atribuire în structura respectivă astfel încât cele două formule să ia valori de adevăr diferite.*

Fie structura $S_5 = (\mathbb{Z}, \{<\}, \{+, -, 0\})$ *definită în cursul anterior şi* S_5*-atribuirea* $\alpha_6 : \mathcal{X} \to \mathbb{Z}$*, definită prin* $\alpha_6(\mathtt{x}) = 2, \alpha_6(\mathtt{y}) = 3$ *şi* $\alpha_6(z) = 1$ *pentru orice* $z \in \mathcal{X} \setminus \{\mathtt{x}, \mathtt{y}\}$.

Observaţi că singura diferenţă între S_1 *şi* S_5 *este faptul că simbolul predicativ* $\mathtt{P}$ *este interpretat prin predicatul* $=$ *în* S_1*, în timp ce în* S_5 *este interpretat prin* $<$ *(relaţia mai mic strict peste numere întregi).*

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

Exemplul 135. *Avem că* $(\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{z})) \equiv (\forall \mathtt{y}.\mathtt{P}(\mathtt{y}, \mathtt{z}))$*. De ce?*

Fie S *o* Σ*-structură oarecare cu domeniul* D *şi* $\alpha : \mathcal{X} \to D$ *o* S*-atribuire oarecare. Avem că*

$$
\begin{array}{ll}
S, \alpha \models (\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{z})) & \textit{ddacă} \\
\textit{pentru orice } u \in D, \textit{ avem } S, \alpha[\mathtt{x} \mapsto u] \models \mathtt{P}(\mathtt{x}, \mathtt{z}) & \textit{ddacă} \\
\textit{pentru orice } u \in D, \textit{ avem } P^S\Big(\overline{\alpha[\mathtt{x} \mapsto u]}(\mathtt{x}), \overline{\alpha[\mathtt{x} \mapsto u]}(\mathtt{z})\Big) & \textit{ddacă} \\
\textit{pentru orice } u \in D, \textit{ avem } P^S\Big(u, \alpha(\mathtt{z})\Big) & \textit{ddacă} \\
\textit{pentru orice } u \in D, \textit{ avem } P^S\Big(\overline{\alpha[\mathtt{y} \mapsto u]}(\mathtt{y}), \overline{\alpha[\mathtt{y} \mapsto u]}(\mathtt{z})\Big) & \textit{ddacă} \\
\textit{pentru orice } u \in D, \textit{ avem } S, \alpha[\mathtt{y} \mapsto u] \models \mathtt{P}(\mathtt{y}, \mathtt{z}) & \textit{ddacă} \\
S, \alpha \models (\forall \mathtt{y}.\mathtt{P}(\mathtt{y}, \mathtt{z})). &
\end{array}
$$

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

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

care este chiar definiţia faptului că $(\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{z})) \equiv (\forall \mathtt{y}.\mathtt{P}(\mathtt{y}, \mathtt{z}))$.

6.2 Forme normale și Substituții

Pentru a defini formele normale pentru formulele din logica de ordinul I, reamintim noţiunea de *substituţie* :

- O *substituţie* este o funcţie $\sigma : \mathcal{X} \to \mathcal{T}$, cu proprietatea că $\sigma(x) \neq x$ pentru un număr finit de variabile $x \in \mathcal{X}$;
- *Domeniul substituţiei* σ este multimea $dom(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$;
- *Extensia* substituţiei σ la mulţimea termenilor este funcţia $\sigma^\sharp : \mathcal{T} \to \mathcal{T}$, 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 $t_1,\ldots,t_n \in \mathcal{T}$.

Dacă $t \in \mathcal{T}$ este un termen, atunci $\sigma^\sharp(t) \in \mathcal{T}$ este termenul obținut din t prin *aplicarea* substituției σ (fiecare apariţie a unei variabile x din t este înlocuită cu termenul $\sigma(x)$).

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

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

- *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 $x \in \mathcal{X} \setminus V$.

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

- *Extensia* lui σ la mulțimea formulelor este funcția $\sigma^\flat : \mathbb{LPI} \to \mathbb{LPI}$, 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. $\sigma^\flat((\forall x.\varphi)) = (\forall x.(\rho^\flat(\varphi)))$, unde $\rho = \sigma|_{dom(\sigma)\setminus\{x\}}$;
8. $\sigma^\flat((\exists x.\varphi)) = (\exists x.(\rho^\flat(\varphi)))$, unde $\rho = \sigma|_{dom(\sigma)\setminus\{x\}}$.

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

6.3 Forma normală Prenex

Definiția 136. *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".

Exemplul 137. *Formula* $(\forall x.(\exists y.(P(x,y) \wedge \neg P(z,y))))$ *este în formă normală prenex;*
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 138. *Pentru orice formulă* $\varphi \in \mathbb{LPI}$*, există* $\varphi' \in \mathbb{LPI}$ *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 următoarele rezultate:

Teorema 139 (Teorema de înlocuire)**.** *Fie* $\varphi, \varphi' \in \mathbb{LP}1$ *astfel încât* $\varphi \equiv \varphi'$ *și* $\varphi_1 \in PL1$ *ce contine* φ *ca subformulă.*
Fie formula φ_2 *obținută din* φ_1 *prin înlocuirea unei apariții a lui* φ *cu* φ'*.*
Atunci $\varphi_1 \equiv \varphi_2$.

Lema 140 (Lema redenumirii)**.** *Fie* $\varphi \in \mathbb{LPI}$ *o formulă și* $x, y \in \mathcal{X}$ *două variabile cu proprietatea că* $y \notin free(\varphi)$.
Atunci au loc echivalențele:

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

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

Cu alte cuvinte, comform lemei redenumirii, în formula $\forall x.\varphi$ putem înlocui cuantificatorul $\forall m$ (respectiv $\exists m$) cu un cuantificator $\forall y$ (respectiv $\exists y$) la alegere cu condiția ca y să nu fie variabilă a formulei φ și să obținem o formulă echivalentă. De asemenea, aparițiile libere ale variabilei x în φ trebuie înlocuite cu y prin aplicarea substituției $\sigma = \{x \mapsto y\}$ asupra formulei φ.

Exemplul 141. *Fie formula* $(\forall x.P(x,y))$*. Deoarece* $z \notin free(P(x,y))$*, avem prin lema redenumirii că* $(\forall x.P(x,y)) \equiv (\forall z.P(z,y))$.
Atenție, $(\forall x.P(x,y)) \not\equiv (\forall y.P(y,y))$ *(echivalența nu poate fi explicată prin lema redenumirii deoarece* $y \in free((\forall x.P(x,y)))$ *și nici nu are loc).*

Suntem acum pregătiți să prezentăm, schițat, demonstrația Teoremei 138.

Demonstrație: Se aplică Teorema 139 (de inlocuire) în care sunt folosite Lema 140 (redenumirii) și 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 mai întai lema redenumirii pentru a redenumi convenabil variabila legată x.

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

7. $(\varphi_1\wedge\varphi_2) \equiv (\varphi_2\wedge\varphi_1)$;
8. $(\varphi_1\vee\varphi_2) \equiv (\varphi_2\vee\varphi_1)$.

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 $\wedge, \vee, \neg$:

9. $(\varphi_1\rightarrow\varphi_2) \equiv \neg\varphi_1\vee\varphi_2$;
10. $(\varphi_1\leftrightarrow\varphi_2) \equiv ((\varphi_1\rightarrow\varphi_2) \wedge (\varphi_2\rightarrow\varphi_1))$.

q.e.d.

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

Exemplul 142. *Fie formula* $\varphi = \Big(\big(\forall x.\neg(P(x,x) \wedge \neg\exists y.P(x,y))\big) \wedge P(x,x)\Big)$.

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.):*

$$
\begin{array}{rcl}
\varphi & = & \Big(\forall x.\neg\big(P(x,x) \wedge \neg\exists y.P(x,y)\big)\Big) \wedge P(x,x) \\
& \overset{L.R.}{\equiv} & \Big(\forall z.\neg\big(P(z,z) \wedge \neg\exists y.P(z,y)\big)\Big) \wedge P(x,x) \\
& \overset{1}{\equiv} & \forall z.\Big(\neg\big(P(z,z) \wedge \neg\exists y.P(z,y)\big) \wedge P(x,x)\Big) \\
& \overset{6}{\equiv} & \forall z.\Big(\neg\big(P(z,z) \wedge \forall y.\neg P(z,y)\big) \wedge P(x,x)\Big) \\
& \overset{7(\text{comutativitate } \wedge)}{\equiv} & \forall z.\Big(\neg\big((\forall y.\neg P(z,y)) \wedge P(z,z)\big) \wedge P(x,x)\Big) \\
& \overset{1}{\equiv} & \forall z.\Big(\neg\big(\forall y.(\neg P(z,y) \wedge P(z,z))\big) \wedge P(x,x)\Big) \\
& \overset{7(\text{comutativitate } \wedge)}{\equiv} & \forall z.\Big(\neg\big(\forall y.(P(z,z) \wedge \neg P(z,y))\big) \wedge P(x,x)\Big) \\
& \overset{5}{\equiv} & \forall z.\Big(\big(\exists y.\neg(P(z,z) \wedge \neg P(z,y))\big) \wedge P(x,x)\Big) \\
& \overset{3}{\equiv} & \forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x)).
\end{array}
$$

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:

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

6.3.1 Fișă de exerciții

Fie $\Sigma = (\{P, Q\}, \{f, i, e\})$ unde $P \in \mathcal{P}_2$, $Q \in \mathcal{P}_1$, $f \in \mathcal{F}_2$, $i \in \mathcal{F}_1$ si $e \in \mathcal{F}_0$.

Exercițiul 143. *Arătați că următoarele echivalențe au loc:*

1. $P(e,x) \overset{S}{\equiv} P(e,f(x,x))$ *unde* S *este* Σ*-structura* $S = (\mathbb{N}, \{<, Par\}, \{+, s, 0\})$.
2. $\neg\forall x.\varphi \equiv \exists x.\neg\varphi$;
3. $\neg\exists x.\varphi \equiv \forall x.\neg\varphi$

4. $((\forall x.\varphi_1) \wedge \varphi_2) \equiv \forall x.(\varphi_1 \wedge \varphi_2)$, *dacă* $x \notin free(\varphi_2)$;
5. $((\forall x.\varphi_1) \vee \varphi_2) \equiv \forall x.(\varphi_1 \vee \varphi_2)$, *dacă* $x \notin free(\varphi_2)$;
6. $((\exists x.\varphi_1) \wedge \varphi_2) \equiv \exists x.(\varphi_1 \wedge \varphi_2)$, *dacă* $x \notin free(\varphi_2)$;
7. $((\exists x.\varphi_1) \vee \varphi_2) \equiv \exists x.(\varphi_1 \vee \varphi_2)$, *dacă* $x \notin free(\varphi_2)$;
8. $\forall x.(P(x,x) \wedge Q(x)) \equiv (\forall x.P(x,x)) \wedge (\forall x.Q(x))$
9. $\forall x.(P(x,x) \vee Q(x)) \not\equiv (\forall x.P(x,x)) \vee (\forall x.Q(x))$
10. $\exists x.(P(x,x) \wedge Q(x)) \not\equiv (\exists x.P(x,x)) \wedge (\exists x.Q(x))$
11. $\exists x.(P(x,x) \vee Q(x)) \equiv (\exists x.P(x,x)) \vee (\exists x.Q(x))$

Exercițiul 144. *Fie substituția* $\sigma : \mathcal{X} \to \mathcal{T}$ *astfel încât* $\sigma(x) = i(y)$, $\sigma(y) = f(x,z)$ *și* $\sigma(z) = z$ *pentru* $z \in \mathcal{X} \setminus \{x,y\}$. *Aplicați substituția* σ *pentru următoarele formule:*

1. $\varphi = (\forall x.P(x,y) \to P(i(y),x)$
2. $\varphi = P(x,y) \wedge \exists y.Q(y \to \forall x.P(x,y)$

Exercițiul 145. *Calculați câte o FNP pentru fiecare dintre formulele:*

1. $\forall x.(P(x,y) \wedge \exists x.P(x,x))$
2. $(\exists z.P(x,y)) \vee P(z,z)$;
3. $(\exists z.P(x,z)) \wedge (\forall x.P(x,z))$;
4. $(\exists z.P(x,z)) \to P(x,x)$.
5. $\neg(\exists x.P(x,y) \vee \forall z.P(z,z)) \wedge \exists y.P(x,y)$
6. $\forall x.\exists y.P(x,y) \to \neg\exists x.\neg\exists y.P(x,y)$

6.4 Formule închise

Definiția 146. *O formulă* $\varphi \in \mathbb{LPI}$ *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).

Definiția 147. *O formulă care nu este închisă se numește* deschisă.

Exemplul 148. *Formula* $(\forall x.P(x,x) \wedge \exists y.P(y,x))$ *este o formulă închisă deoarece* $free((\forall x.P(x,x) \wedge \exists y.P(y,x))) = \emptyset$.

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

Definiția 149. *Fie* $\varphi \in \mathbb{LPI}$ *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.

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

Exemplul 150. *Închiderea existențială a formulei* $(\forall z.(\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))))$ *este* $\Big(\exists x.(\forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))))\Big)$.

Definiția 151. *Două formule* $\varphi_1 \in \mathbb{LPI}$ *și* $\varphi_2 \in \mathbb{LPI}$ *sunt* echisatisfiabile, *dacă:*

1. *atât* φ_1 *cât și* φ_2 *sunt satisfiabile;* ***sau***
2. *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 152. *Orice formulă este echisatisfiabilă cu închiderea ei existențială.*

Definiția 153. *Fie* $\varphi \in \mathbb{LPI}$ *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.

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

Exemplul 154. *Închiderea universală a formulei* $(\forall z.(\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))))$ *este* $\Big(\forall x.(\forall z.(\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))))\Big)$.

Teorema 155. *O formulă este validă dacă și numai dacă închiderea ei universală este validă.*

6.5 Forma normală Skolem

Definiția 156 (FNS). *O formulă φ este în formă normală Skolem (prescurtat, FNS) dacă*

$$\varphi = \forall x_1 \ldots . \forall x_n . \varphi',$$

unde:

1. *φ' nu conține cuantificatori și*
2. *$free(\varphi') \subseteq \{x_1, \ldots, x_n\}$.*

Cu alte cuvinte, o formulă este în formă normală Skolem dacă conține doar cuantificatori universali aflați la "începutul" formulei, iar toate aparițiile variabilelor din formulă sunt apariții legate (avem cuantificatori pentru toate variabilele ce apar în formulă).

Observație. *O formulă aflată în FNS este obligatoriu închisă, deoarece toate variabilele libere ale formulei φ' sunt cuantificate universal în φ (datorită condiției 2), și deci nu mai pot exista variabile libere în φ.*

Exemplul 157. *În continuare, vom lucra peste signatura $\Sigma = (\{\mathtt{P}, \mathtt{Q}, \mathtt{R}\}, \{\mathtt{f}, \mathtt{i}, \mathtt{e}\})$, unde $\mathtt{P}, \mathtt{Q}, \mathtt{R}$ sunt simboluri predicative de aritate $2, 1$ și respectiv 3, $\mathtt{f}$ și $\mathtt{i}$ sunt simboluri funcționale de aritate 2 și respectiv 1, iar $\mathtt{e}$ este simbol functional de aritate 0 (constantă).*

Exemple de formule în FNS:

$$\forall \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{i}(\mathtt{e})) \qquad \forall \mathtt{x}.\forall \mathtt{y}.\Big(\mathtt{P}(\mathtt{f}(\mathtt{x}, \mathtt{e}), \mathtt{y}) \wedge \neg(\mathtt{R}(\mathtt{x}, \mathtt{i}(\mathtt{f}(\mathtt{y}, \mathtt{y})), \mathtt{e}) \vee \mathtt{Q}(\mathtt{y}))\Big)$$

Exemple de formule care nu sunt în FNS:

$$\exists \mathtt{x}.\mathtt{P}(\mathtt{x}, \mathtt{x}) \qquad \forall \mathtt{x}.\Big(\mathtt{Q}(\mathtt{e}) \wedge \neg(\mathtt{Q}(\mathtt{x}) \vee \mathtt{Q}(\mathtt{y}))\Big) \qquad \mathtt{Q}(\mathtt{e}) \wedge \forall \mathtt{x}.\mathtt{Q}(\mathtt{x})$$

În cazul formulelor care nu sunt în FNS, motivele pentru care acestea nu sunt în FNS sunt urmatoarele: prima formulă conține cuantificator existențial; în cea de-a doua formulă nu avem cuantificator pentru variabila y; iar în cea de-a treia formulă, cuantificatorul $\forall \mathtt{x}$ nu se află la începutul formulei.

Teorema 158 (Teorema de aducere în FNS). *Pentru orice formulă $\varphi \in \mathbb{LPI}$, există o formulă $\varphi' \in \mathbb{LPI}$ astfel încât:*

1. *φ' este în formă normală Skolem;*
2. *φ și φ' sunt echisatisfiabile.*

Demonstrație: [Schiță de demonstrație]

1. Calculăm o formulă φ_1, aflată în FNP și echivalentă cu formula φ (folosind Teorema de aducere în FNP din cursul precedent);

2. Calculăm o formulă φ_2, închiderea existențială a lui φ_1 (φ_2 va fi echisatisfabilă cu φ_1 și deci cu φ);

3. Aplicăm în mod repetat lema de Skolemizare pe formula φ_2, lemă prezentată mai jos.

Rezultatul va fi o formulă aflată în FNS, echisatisfabilă cu formula de la care am plecat.

q.e.d.

Lema 159 (Skolem)**.** *Fie $\varphi = \forall x_1.\forall x_2.\ldots.\forall x_k.\exists x.\varphi'$, unde $k \geq 0$, $\varphi' \in \mathbb{LPI}$ (φ' poate conține alți cuantificatori). Cu alte cuvinte, există k cuantificatori universali înainte de primul cunatificator existențial.*

Fie $f \in \mathcal{F}_k$ un simbol funcțional de aritate k care nu apare în φ (un simbol funcțional proaspăt – engl. fresh).

Avem că φ este echisatisfiabilă cu

$$\forall x_1.\ldots.\forall x_k.(\sigma^\flat(\varphi')),$$

unde $\sigma = \{x \mapsto f(x_1, \ldots, x_k)\}$.

Demonstrație: [Schiță de demonstrație]

Considerând că formula φ este peste signatura $\Sigma = (\mathcal{P}, \mathcal{F})$, observăm că formula nou obținută este o formulă peste signatura $\Sigma' = (\mathcal{P}, \mathcal{F} \cup \{f\})$ ce conține, pe lângă simbolurile predicative și cele funcționale din signatura Σ, și simbolul funcțional f de aritate k.

Implicația directă:

Presupunem că există o Σ-structură S și o atribuire α astfel încât $S, \alpha \models \varphi$ și găsim o Σ'-structură S' și o atribuire α' astfel încât

$$S', \alpha' \models \forall x_1.\forall x_2.\ldots.\forall x_k.(\sigma^\flat(\varphi'))$$

Atenție! Structura S' este peste signatura Σ' care este mai bogată decât signatura structurii S (apare simbolul funcțional f de aritate k, care este nou). Rezultatele calculate de funcția f sunt în concordanță cu valorile alese pentru variabila x în functie de valorile variabilelor $x_1 \ldots x_k$ (știm că există o astfel de valoare aleasă astfel încât φ să fie satisfăcută de S și α). În schimb, atribuirea α' poate coincide cu atribuirea α.

Implicația inversă:

Presupunem că există o Σ'-structură S' și o atribuire α' astfel încât $S', \alpha' \models \forall x_1.\forall x_2.\ldots.\forall x_k.(\sigma^\flat(\varphi'))$.

Găsim o Σ-structură S și atribuirea σ astfel încât $S, \alpha \models \varphi$. ($S$ este obținută din S' prin eliminarea funcției f). Din nou, atribuirea σ poate fi identică cu σ'.

q.e.d.

Exercițiul 160. *Completați demonstrația de mai sus.*

Exemplul 161. *Calculăm o formă normală Skolem pentru formula*

$$\varphi = \forall \mathsf{x}.\exists \mathsf{y}.\forall \mathsf{z}.\exists \mathsf{z}'.(\mathtt{P}(\mathsf{x}, \mathsf{y}) \leftrightarrow \mathtt{P}(\mathsf{z}, \mathsf{z}')).$$

Prin Lema 159, avem că φ este echisatisfiabilă cu

$$\varphi_1 = \forall \mathsf{x}.\forall \mathsf{z}.\exists \mathsf{z}'.(\mathtt{P}(\mathsf{x}, \mathsf{g}(\mathsf{x})) \leftrightarrow \mathtt{P}(\mathsf{z}, \mathsf{z}')),$$

unde g este un simbol funcțional nou, de aritate 1.

Aplicând din nou Lema 159, avem că formula φ_1 este echisatisfiabilă cu

$$\varphi_2 = \forall \mathsf{x}.\forall \mathsf{z}.(\mathtt{P}(\mathsf{x}, \mathsf{g}(\mathsf{x})) \leftrightarrow \mathtt{P}(\mathsf{z}, \mathsf{h}(\mathsf{x}, \mathsf{z}))),$$

unde h este un simbol funcțional nou, de aritate 2.

În concluzie, φ_2 este în FNS și est echisatisfiabilă cu φ, deci este o formă normală Skolem a formulei φ.

Observație. *Signatura formei normale Skolem a unei formule este mai bogată decât signature formulei de la care am plecat, din cauza adăugării simbolurilor Skolem.*

6.6 Forma normală conjunctivă

Definiția 162 (Literal). *O formulă $\varphi \in \mathbb{LPI}$ se numește* literal *dacă există un simbol predicativ $P \in \mathcal{P}_n$ de aritate $n \geq 0$ și n termeni $t_1, \ldots, t_n \in \mathcal{T}$ astfel încât*

$$\varphi = P(t_1, \ldots, t_n) \text{ sau } \varphi = \neg P(t_1, \ldots, t_n).$$

Cu alte cuvinte, un literal este o formulă atomică, sau negația unei formule atomice.

Exemplul 163. *Exemple de literali:*

$$\mathtt{P}(\mathsf{x}, \mathsf{x}) \quad \neg\mathtt{P}(\mathsf{x}, \mathsf{i}(\mathsf{y})) \quad \neg\mathtt{R}(\mathsf{a}, \mathsf{b}, \mathsf{c}) \quad \neg\mathtt{P}(\mathsf{x}, \mathsf{x}) \quad \mathtt{Q}(\mathsf{i}(\mathsf{x})) \quad \mathtt{R}(\mathsf{a}, \mathsf{f}(\mathsf{x}), \mathsf{b})$$

Exemple de formule care nu sunt literali:

$$\mathtt{P}(\mathsf{x}, \mathsf{y}) \wedge \mathtt{P}(\mathsf{x}, \mathsf{y}) \qquad \neg\neg\mathtt{P}(\mathsf{x}, \mathsf{x}) \qquad \forall \mathsf{x}.\mathtt{P}(\mathsf{x}, \mathsf{x})$$

Definiția 164 (Clauză). *O formulă $\varphi \in \mathbb{LPI}$ se numește* clauză *dacă există n literali $\varphi_1, \ldots, \varphi_n \in \mathbb{FOL}$ astfel încât*

$$\varphi = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n.$$

Exemplul 165. *Exemple de clauze:*

$$\mathtt{P}(\mathtt{x},\mathtt{x}) \vee \mathtt{R}(\mathtt{x},\mathtt{e},\mathtt{y}) \vee \neg\mathtt{P}(\mathtt{x},\mathtt{f}(\mathtt{e},\mathtt{y})) \qquad \mathtt{P}(\mathtt{x},\mathtt{x}) \qquad \square \qquad \mathtt{P}(\mathtt{x},\mathtt{x})$$

$$\mathtt{P}(\mathtt{x},\mathtt{x}) \vee \mathtt{R}(\mathtt{x},\mathtt{f}(\mathtt{e},\mathtt{e}),\mathtt{y}) \vee \neg\mathtt{P}(\mathtt{x},\mathtt{f}(\mathtt{e},\mathtt{y})) \vee \neg\mathtt{P}(\mathtt{e},\mathtt{i}(\mathtt{x}))$$

Observație. *Un caz particular este reprezentat de clauza vidă, notată $\square$, care este disjuncția a 0 literali. Clauza vidă este o formulă nesatisfiabilă.*

Un alt caz particular este reprezentat de literali. Orice literal este și clauză, fiind considerat disjuncția unui singur literal.

Definiția 166 (FNC). *O formulă φ este* în formă normală clauzală *(sau, echivalent,* în formă normală conjunctivă*) dacă există $n \geq 1$ clauze $\varphi_1, \ldots, \varphi_n$ astfel încât*

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n.$$

Exemplul 167. *Următoarele formule sunt în FNC:*

$$(\mathtt{P}(\mathtt{x},\mathtt{x}) \vee \mathtt{Q}(\mathtt{x})) \wedge (\neg\mathtt{P}(\mathtt{x},\mathtt{y}) \vee \mathtt{R}(\mathtt{x},\mathtt{y},\mathtt{e}))$$

$$(\mathtt{P}(\mathtt{x},\mathtt{y}) \vee \mathtt{Q}(\mathtt{i}(\mathtt{x})) \vee \neg\mathtt{Q}(\mathtt{e})) \wedge (\neg\mathtt{P}(\mathtt{x},\mathtt{x})) \wedge (\neg\mathtt{Q}(\mathtt{f}(\mathtt{z},\mathtt{z})) \vee \mathtt{R}(\mathtt{x},\mathtt{z}))$$

6.7 Forma normală Skolem clauzală

Definiția 168. *O formulă φ* este în formă normală Skolem clauzală *(prescurtat FNSC) dacă*

1. *φ este în formă normală Skolem și*
2. *φ' este în formă normală clauzală, unde: $\varphi = \forall x_1 \ldots . \forall x_n . \varphi'$, iar φ' nu are cuantificatori (cu alte cuvinte, φ' este subformula obținută din φ prin ștergerea cuantificatorilor).*

Exemplul 169. *Exemple de formule în FNSC:*

$$\forall \mathtt{x}.\forall \mathtt{y}.\Big((\mathtt{P}(\mathtt{x},\mathtt{x}) \vee \neg\mathtt{Q}(\mathtt{i}(\mathtt{x})) \wedge (\mathtt{P}(\mathtt{e},\mathtt{y}) \vee \neg\mathtt{Q}(\mathtt{e}))\Big)$$

$$\forall \mathtt{x}.\forall \mathtt{y}.\Big(\mathtt{Q}(\mathtt{e}) \wedge (\neg\mathtt{R}(\mathtt{x},\mathtt{e},\mathtt{y}) \vee \mathtt{Q}(\mathtt{i}(\mathtt{y})))\Big)$$

$$\forall \mathtt{x}.\forall \mathtt{y}.\forall \mathtt{z}.(\mathtt{P}(\mathtt{x},\mathtt{y}) \wedge (\mathtt{Q}(\mathtt{x}) \vee \mathtt{R}(\mathtt{x},\mathtt{y},\mathtt{z})) \wedge \neg\mathtt{Q}(\mathtt{x})).$$

Exemple de formule care nu sunt în FNSC:

$$\exists x.Q(x) \qquad \forall x.\Big(Q(e) \wedge (\neg R(x,y,z) \vee Q(y))\Big)$$

$$\forall x.\forall y.\Big(Q(e) \wedge \neg(Q(x) \vee Q(y))\Big)$$

In cazul ultimelor formule, motivele pentru care acestea nu sunt in FNSC sunt urmatoarele:

- $\exists x.Q(x)$ *conține cuantificator existențial (neacceptat în FNS)*
- $\forall x.\Big(Q(e) \wedge (\neg R(x,y,z) \vee Q(y))\Big)$ *nu este in FNS deoarece contine variabile libere (y si z)*
- $\forall x.\forall y.\Big(Q(e) \wedge \neg(Q(x) \vee Q(y))\Big)$ *este în FNS, dar formula* $\varphi' = \Big(Q(e) \wedge \neg(Q(x) \vee Q(y))\Big)$ *nu este în formă normală conjunctivă*

Teorema 170. *Pentru orice formulă* $\varphi \in \mathbb{LPI}$ *aflată în FNS există o formulă* $\varphi' \in \mathbb{LPI}$ *astfel încât:*

1. φ' *este în FNSC și*
2. $\varphi \equiv \varphi'$.

Demonstrație: [Schiță de demonstrație]
Se aplică de la stânga la dreapta următoarele echivalențe:

1. $\varphi_1 \leftrightarrow \varphi_2 \equiv ((\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1))$
2. $(\varphi_1 \rightarrow \varphi_2) \equiv (\neg\varphi_1 \vee \varphi_2)$
3. $\neg\neg\varphi \equiv \varphi$;
4. $\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$;
5. $\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$;
6. $\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$.

De asemenea, se permite folosirea liberă a asociativității și comutativității conectorilor $\vee$ și $\wedge$.

Primele două echivalențe elimină orice folosire a cuantificatorilor $\leftrightarrow$ și $\rightarrow$.

Următoarele trei echivalențe se asigură că negațiile pot fi aplicate doar formulelor atomice.

Ultima echivalență se asigură că $\vee$-ul nu apare peste $\wedge$.

În final, arborele sintactic asociat formulei rezultate va avea spre rădăcină, după cuantificatorii universali, $\wedge$, urmat de un nivel de $\vee$, urmat de $\neg$, urmat de formule atomice, ceea ce înseamnă că formula rezultată este în FNSC.

q.e.d.

Observație. *Un rezultat al teoremelor 158 și 170 este faptul că pentru orice formulă $\varphi \in \mathbb{LPI}$ există o formulă $\varphi' \in \mathbb{LPI}$ astfel încât φ' este în FNSC iar formulele φ și φ' sunt echisatisfiabile.*

Pentru a găsi această formulă, trebuie să urmăm pașii următori:

1. *Calculăm formula φ_1 în FNP echivalentă cu φ;*
2. *Calculăm φ_2 închiderea existențială pentru φ_1;*
3. *Aplicăm lema de Skolemizare pentru a elimina cuantificatorii existențiali și obținem formula φ_3;*
4. *Aplicăm echivalențele din teorema 170 pentru a pune formula φ_3 în FNSC.*

Exemplul 171. *Calculăm o FNSC pentru formula $\varphi = (\forall x.P(x,z)) \wedge (\exists y.\neg P(y,z))$.*

$$\begin{aligned} \varphi = & \quad (\forall x.P(x,z)) \wedge (\exists y.\neg P(y,z)) \\ \equiv & \quad \forall x.(P(x,z) \wedge (\exists y.\neg P(y,z))) \\ \equiv & \quad \forall x.\exists y.(P(x,z) \wedge \neg P(y,z)) \end{aligned}$$

Așadar, o formă normală prenex (FNP) a formulei φ este

$$\varphi_1 = \forall x.\exists y.(P(x,z) \wedge \neg P(y,z))$$

Continuăm prin calcularea închiderii existentiale pentru φ_1. Aceasta este

$$\varphi_2 = \exists z.\forall x.\exists y.(P(x,z) \wedge \neg P(y,z)).$$

Conform Teoremei 152, φ_2 este echisatisfiabil cu φ_1.

În continuare aplicăm lema de scolemizare (Lema 159) pentru eliminarea cuantificatorilor existentiali din φ_2. Astfel, φ_2 este echisatisfiabil cu

$$\varphi_3 = \forall x.\exists y.(P(x,c) \wedge \neg P(y,c))$$

unde c este un simbol constant nou (deoarece în φ_2, cuantificatorul $\exists z$ nu are alti cuantificatori universali in față). Aplicand din nou lema de skolemizare, obtinem că φ_3 este equisatisfiabil cu

$$\varphi_4 = \forall x.(P(x,c) \wedge \neg P(g(x),c))$$

unde g este un simbol functional nou de aritate 1 (deoarece în φ_3, cuantificatorul $\exists y$ este precedat doar de cuantificatorul universal $\forall x$). Formula φ_4 obtinuta este o FNSC pentru φ deoarece este in FNS si formula $\varphi' = (P(x,c) \wedge \neg P(g(x),c))$ este in FNC.

6.7.1 Fișă de exerciții

Exercițiul 172. *Gasiți câte o FNSC pentru următoarele formule:*

1. $\neg\Big((\forall \mathtt{x}.\mathtt{Q}(\mathtt{x})) \to (\exists \mathtt{x}.\mathtt{Q}(\mathtt{x}))\Big)$;
2. $(\exists \mathtt{z}.\mathtt{P}(\mathtt{x},\mathtt{z})) \wedge (\forall \mathtt{x}.\mathtt{P}(\mathtt{x},\mathtt{z}))$;

Capitolul 7

Rezoluția pentru LP1

Rezoluția pentru LP1 este o metodă de demonstrare a nesatisfiabilității unei formule de ordinul I aflată în FNSC. Avantajul rezoluției este că este făcută mecanic, printr-o metodă inventată de Robinson în anii 1960 numită unificare; așadar este mai ușor pretabilă spre a fi mecanizată (implementată într-un program pe calculator).

7.1 Unificare

Definiția 173 (Unificator)**.** *O substituție σ este unificator al termenilor t_1 și t_2 dacă $\sigma^\sharp(t_1) = \sigma^\sharp(t_2)$.*

Exemplul 174. *Fie termenii $t_1 = \mathsf{f}(\mathsf{x}, \mathsf{h}(\mathsf{y}))$ și $t_2 = \mathsf{f}(\mathsf{h}(\mathsf{z}), \mathsf{z}')$. Un unificator pentru t_1 și t_2 este:*

$$\sigma = \{\mathsf{z} \mapsto \mathsf{a}, \mathsf{x} \mapsto \mathsf{h}(\mathsf{a}), \mathsf{z}' \mapsto \mathsf{h}(\mathsf{y})\} \qquad (\sigma^\sharp(t_1) = \mathsf{f}(\mathsf{h}(\mathsf{a}), \mathsf{h}(\mathsf{y})) = \sigma^\sharp(t_2)).$$

Un alt unificator al celor doi termeni este:

$$\sigma_1 = \{\mathsf{x} \mapsto \mathsf{h}(\mathsf{z}), \mathsf{z}' \mapsto \mathsf{h}(\mathsf{y})\} \qquad (\sigma_1^\sharp(t_1) = \mathsf{f}(\mathsf{h}(\mathsf{z}), \mathsf{h}(\mathsf{y})) = \sigma_1^\sharp(t_2)).$$

Definiția 175 (Termeni unificabili)**.** *Doi termeni sunt* unificabili *dacă au cel puțin un unificator.*

Exemplul 176. *Termenii $t_1 = \mathsf{f}(\mathsf{x}, \mathsf{y})$ și $t_2 = \mathsf{h}(\mathsf{z})$ nu au unificator, deci nu sunt unificabili. De ce?*

Pentru orice substituție σ avem $\sigma^\sharp(t_1) = \sigma^\sharp(\mathsf{f}(\mathsf{x}, \mathsf{y})) = \mathsf{f}(\sigma^\sharp(\mathsf{x}), \sigma^\sharp(\mathsf{y})) \neq \mathsf{h}(\sigma^\sharp(\mathsf{z})) = \sigma^\sharp(\mathsf{h}(\mathsf{z})) = \sigma^\sharp(t_2)$ (intuitiv, orice substituție am aplica peste cei doi

termeni, în primul caz rezultatul are ca funcție principală f*, iar în al foilea caz avem* h*).*

Termenii $t_1 = \mathsf{x}$ *și* $t_2 = \mathsf{h(x)}$ *nu au unificator. De ce?*

Să presupunem că ar exista un unificator al lor, σ*. Din moment ce* $\sigma^\sharp(t_1) = \sigma^\sharp(t_2)$*, înseamnă că arborii abstracti corespunzatori termenilor* $\sigma^\sharp(t_1)$ *și* $\sigma^\sharp(t_2)$ *au același număr de noduri. Notăm cu noduri*(t) *numărul de noduri ale arborelui corespunzator unui termen* t*.*

Avem că noduri$(\sigma^\sharp(t_2)) = $ *noduri*$(\sigma^\sharp(\mathsf{h(x)})) = 1 +$ *noduri*$(\sigma^\sharp(\mathsf{x})) = 1 +$ *noduri*$(\sigma^\sharp(t_1)) >$ *noduri*$(\sigma^\sharp(t_1))$*, ceea ce reprezintă o contradicție, deoarece trebuia să avem noduri*$(\sigma^\sharp(t_2)) =$ *noduri*$(\sigma^\sharp(t_1))$*. Prin urmare, presupusul unificator* σ *nu există.*

Definiția 177 (Compunerea a două substituții)**.** *Fie* σ_1, σ_2 *două substituții. Substituția* $\sigma_2 \circ \sigma_1 : \mathcal{X} \to \mathcal{T}$*, denumită* compunerea substituțiilor σ_1 și σ_2*, este definită astfel:*

- $(\sigma_2 \circ \sigma_1)(x) = \sigma_2^\sharp(\sigma_1(x))$*, pentru orice* $x \in \mathcal{X}$*.*

Exercițiul 178. *Verificați că funcția* $\sigma_2 \circ \sigma_1$ *este într-adevăr o substituție (adică că mulțimea acelor variabile* x *cu proprietatea că* $(\sigma_2 \circ \sigma_1)(x) \neq x$ *este finită).*

Exemplul 179. *Continuând exemplul anterior, fie substituțiile* $\sigma = \{\mathsf{z} \mapsto \mathsf{a}, \mathsf{x} \mapsto \mathsf{h(a)}, \mathsf{z'} \mapsto \mathsf{h(y)}\}$*,* $\sigma_1 = \{\mathsf{x} \mapsto \mathsf{h(z)}, \mathsf{z'} \mapsto \mathsf{h(y)}\}$ *și respectiv* $\sigma_2 = \{\mathsf{z} \mapsto \mathsf{a}\}$*. Avem că* $\sigma = \sigma_2 \circ \sigma_1$ *deoarece:*

$$\sigma_2 \circ \sigma_1(\mathsf{x}) = \sigma_2^\sharp(\sigma_1(\mathsf{x})) = \sigma_2^\sharp(\mathsf{h(z)}) = \mathsf{h(a)} = \sigma(\mathsf{x})$$
$$\sigma_2 \circ \sigma_1(\mathsf{z}) = \sigma_2^\sharp(\sigma_1(\mathsf{z})) = \sigma_2^\sharp(\mathsf{z}) = \mathsf{a} = \sigma(\mathsf{z})$$
$$\sigma_2 \circ \sigma_1(\mathsf{z'}) = \sigma_2^\sharp(\sigma_1(\mathsf{z'})) = \sigma_2^\sharp(\mathsf{h(y)}) = \mathsf{h(y)} = \sigma(\mathsf{z'})$$
$$\sigma_2 \circ \sigma_1(y') = \sigma_2^\sharp(\sigma_1(y')) = \sigma_2^\sharp(y') = y' = \sigma(y'), \text{ for any } y' \in \mathcal{X} \setminus \{x, z, z'\}$$

Definiția 180 (Substituție mai generală)**.** *O substituție* σ_1 *este mai generală decât o substituție* σ *dacă* σ *se poate obține prin compunerea substituției* σ_1 *cu o altă substituție* σ_2*:* $\sigma = \sigma_2 \circ \sigma_1$*.*

Exemplul 181. *De exemplu,* $\sigma_1 = \{\mathsf{x} \mapsto \mathsf{h(z)}, \mathsf{z'} \mapsto \mathsf{h(y)}\}$ *este mai generală decât* $\sigma = \{\mathsf{z} \mapsto \mathsf{a}, \mathsf{x} \mapsto \mathsf{h(a)}, \mathsf{z'} \mapsto \mathsf{h(y)}\}$*, deoarece* $\sigma = \sigma_2 \circ \sigma_1$*, unde* σ_2 *este definită în exemplul de mai sus.*

7.2 Cel mai general unificator

Definiția 182 (Cel mai general unificator)**.** *O substituție* σ *este* cel mai general unificator al termenilor t_1 și t_2 *dacă:*

1. *σ este unificator al termenilor t_1, t_2 și*
2. *σ este o substituție mai generală decât orice unificator al t_1, t_2.*

Exemplul 183. *Fie $t_1 = \mathsf{f(x, a)}$ și $t_2 = \mathsf{f(y, a)}$. Unificatorul $\{\mathsf{y} \mapsto \mathsf{x}\}$ este mai general decât $\{\mathsf{x} \mapsto \mathsf{a}, \mathsf{y} \mapsto \mathsf{a}\}$ și orice alt unificator al celor doi termeni.*

Exemplul 184. *Substituția $\sigma_1 = \{\mathsf{x} \mapsto \mathsf{h(z)}, \mathsf{z'} \mapsto \mathsf{h(y)}\}$ este cel mai general unificator al termenilor $t_1 = \mathsf{f(x, h(y))}$ și $t_2 = \mathsf{f(h(z), z')}$.*

Teorema 185 (Teorema existenței celui mai general unificator)**.** *Orice doi termeni unificabili au un cel mai general unificator.*

Observație. *În general, cel mai general unificator nu este unic.*

Exemplul 186. *Un unificator pentru termenii $\mathsf{h(x)}$ și $\mathsf{h(y)}$ este substituția $\{\mathsf{x} \mapsto \mathsf{a}, \mathsf{y} \mapsto \mathsf{a}\}$ (dar nu este cel mai general unificator).*

Un cel mai general unificator este $\{\mathsf{x} \mapsto \mathsf{y}\}$. Un alt cel mai general unificator este $\{\mathsf{y} \mapsto \mathsf{x}\}$.

În continuare, vom prezenta un algoritm pentru calculul unui cel mai general unificator.

În acest scop, avem nevoie de generalizarea noțiunii de unificare pentru mai multe perechi de termeni.

7.3 Problemă de unificare

Definiția 187 (Problemă de unificare)**.** *O* problemă de unificare $\mathbb{P}$ *este:*

- *sau o mulțime*
$$\mathbb{P} = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$$
formată din n perechi de termeni
- *sau simbolul special*
$$\mathbb{P} = \bot \text{ (citit bottom).}$$

Definiția 188 (Soluție a unei probleme de unificare)**.** *O substituție σ este o* soluție *a unei probleme de unificare $\mathbb{P}$ dacă:*

1. *problema este de forma $\mathbb{P} = \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$ și*
2. *σ este unificator pentru t_i și t'_i, pentru orice $i \in \{1, \ldots, n\}$.*

Definiția 189 (Mulțimea soluțiilor unei probleme de unificare)**.** *Cu $unif(\mathbb{P})$ notăm mulțimea soluțiilor unei probleme de unificare $\mathbb{P}$:*

$$unif(\mathbb{P}) = \{\sigma \mid \sigma \text{ este soluție a problemei } \mathbb{P}\}.$$

Observație. *Prin definiția noțiunii de soluție a unei probleme de unificare, dacă* $\mathbb{P} = \bot$*, atunci* $unif(\mathbb{P}) = \emptyset$*.*

Exemplul 190. *Fie* $\mathbb{P} = \{f(x, a) \doteq f(y, a)\}$*. Avem că* $unif(\mathbb{P}) = \{\{x \mapsto z, y \mapsto z\}, \{x \mapsto y\}, \ldots\}$*.*

Definiția 191 (Cea mai generală soluție)**.** *Substituția* σ *este cea mai generală soluție pentru o problemă de unificare* $\mathbb{P} = \{t_1 \doteq t_1', \ldots, t_n \doteq t_n'\}$ *dacă:*

1. σ *este soluție pentru* $\mathbb{P}$*:* $\sigma^\sharp(t_i) = \sigma^\sharp(t_i')$*, pentru orice* $1 \leq i \leq n$*;*
2. σ *este mai generală decât orice altă soluție pentru* P*.*

Observație. *Observați că în cazul în care* $unif(\mathbb{P}) \neq \emptyset$ *(problema de unificare are soluții), atunci există cel putin o cea mai generală soluție pentru* $\mathbb{P}$*.*

Notație. *Cu* $mgu(\mathbb{P})$ *notăm o cea mai generală soluție a problemei de unificare* P *(dacă problema* $\mathbb{P}$ *are soluție).*

Cu $mgu(t_1, t_2)$ *notăm un cel mai general unificator al termenilor* t_1, t_2 *(dacă termenii sunt unificabili).*

Observație. $mgu(t_1, t_2) = mgu(\{t_1 \doteq t_2\})$*.*

Definiția 192 (Formă rezolvată)**.** *O problemă de unificare* $\mathbb{P}$ este în formă rezolvată *dacă* $\mathbb{P} = \bot$ *sau* $\mathbb{P} = \{x_1 \doteq t_1', \ldots, x_n \doteq t_n'\}$ *și* $x_i \notin vars(t_j)$ *pentru orice* $i, j \in \{1, \ldots, n\}$*.*

Exemplul 193. *Următoarele probleme de unificare sunt în formă rezolvată:*

- $\mathbb{P}_1 = \{x_3 \doteq f(g(a, a), a), x_2 \doteq a, x_1 \doteq a\}$
- $\mathbb{P}_2 = \bot$

Următoarele probleme de unificare NU sunt în formă rezolvată:

- $\mathbb{P}_3 = \{f(x, a) \doteq f(y, a)\}$ *(deoarcec în "perechea"* $f(x, a) \doteq f(y, a)$ *termenul din stânga nu este o variabilă)*
- $\mathbb{P}_4 = \{x_1 \doteq f(x_2, x_3), x_2 \doteq x_3\}$ *(deoarece, chiar dacă avem doar variabile în partea stângă,există variabila* x_2 *în partea stângă a* $\doteq$ *și apare și în partea dreaptă a* $\doteq$*)*

De ce este utilă forma rezolvată a problemelor de unificare?

Lema 194. *Dacă* $\mathbb{P} = \{x_1 \doteq t_1', \ldots, x_n \doteq t_n'\}$ *este în formă rezolvată, atunci* $\{x_1 \mapsto t_1', \ldots, x_n \mapsto t_n'\}$ *este cea mai generală soluție a problemei* $\mathbb{P}$*.*

Următoarele reguli pot fi folosite pentru aducerea unei probleme de unificare în formă rezolvată:

Ștergere	$\mathbb{P} \cup \{t \doteq t\} \Rightarrow \mathbb{P}$
Descompunere	$\mathbb{P} \cup \{f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n)\} \Rightarrow$ $\mathbb{P} \cup \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$
Orientare	$\mathbb{P} \cup \{f(t_1, \ldots, t_n) \doteq x\} \Rightarrow \mathbb{P} \cup \{x \doteq f(t_1, \ldots, t_n)\}$
Eliminare	$\mathbb{P} \cup \{x \doteq t\} \Rightarrow \sigma^\sharp(\mathbb{P}) \cup \{x \doteq t\}$ dacă $x \notin vars(t), x \in vars(\mathbb{P})$(unde $\sigma = \{x \mapsto t\}$)
Conflict	$\mathbb{P} \cup \{f(t_1, \ldots, t_n) \doteq g(t'_1, \ldots, t'_m)\} \Rightarrow \bot$ dacă $f \neq g$
Occurs check	$\mathbb{P} \cup \{x \doteq f(t_1, \ldots, t_n)\} \Rightarrow \bot$ dacă $x \in vars(f(t_1, \ldots, t_n))$

Transformările de mai sus au următoarele proprietăți:

Lema 195 (Progres)**.** *Dacă $\mathbb{P}$ nu este în formă rezolvată, atunci există $\mathbb{P}'$ astfel încât $\mathbb{P} \Rightarrow \mathbb{P}'$.*

Lema 196 (Păstrarea soluțiilor)**.** *Dacă $\mathbb{P} \Rightarrow \mathbb{P}'$, atunci $unif(\mathbb{P}) = unif(\mathbb{P}')$.*

Lema 197 (Terminare)**.** *Nu există o secvență infinită $\mathbb{P} \Rightarrow \mathbb{P}_1 \Rightarrow \mathbb{P}_2 \Rightarrow \ldots \Rightarrow \mathbb{P}_i \Rightarrow \ldots$.*

Corolarul 198. *Regulile precedente constituie un algoritm de calcul al unei cele mai generale soluții pentru o problemă de unificare, dacă aceasta există.*

Exemplul 199.

$$\mathbb{P} = \{f(g(x_1, a), x_2) \doteq x_3, f(x_2, x_2) \doteq f(a, x_1)\} \overset{\text{Descompunere}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_2 \doteq a, x_2 \doteq x_1\} \overset{\text{Eliminare}}{\Rightarrow}$$

$$\{f(g(x_1, a), a) \doteq x_3, x_2 \doteq a, a \doteq x_1\} \overset{\text{Orientare}}{\Rightarrow}$$

$$\{f(g(x_1, a), a) \doteq x_3, x_2 \doteq a, x_1 \doteq a\} \overset{\text{Eliminare}}{\Rightarrow}$$

$$\{f(g(a, a), a) \doteq x_3, x_2 \doteq a, x_1 \doteq a\} \overset{\text{Orientare}}{\Rightarrow}$$

$$\{x_3 \doteq f(g(a, a), a), x_2 \doteq a, x_1 \doteq a\}.$$

Concluzie: $\{x_3 \mapsto f(g(a, a), a), x_2 \mapsto a, x_1 \mapsto a\}$ *este cea mai generală soluție a problemei de unificare inițiale.*

Exemplul 200.

$$\mathbb{P} = \{f(g(x_1, a), x_2) \doteq x_3, f'(x_2) \doteq f'(x_3)\} \overset{\text{Descompunere}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_2 \doteq x_3\} \overset{\text{Orientare}}{\Rightarrow}$$

$$\{x_3 \doteq f(g(x_1, a), x_2), x_2 \doteq x_3\} \overset{\text{Eliminare}}{\Rightarrow}$$

Explicati de ce nu se mai poate aplica orientare

$$\{x_3 \doteq f(g(x_1, a), x_3), x_2 \doteq x_3\} \overset{\text{Occurs check}}{\Rightarrow}$$

$$\bot.$$

Concluzie: $unif(\mathbb{P}) = \emptyset$.

Exemplul 201.

$$\mathbb{P} = \{f(g(x_1, a), x_2) \doteq x_3, f(g(x_4, x_5)) \doteq f(x_3)\} \overset{\text{Descompunere}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, g(x_4, x_5) \doteq x_3\} \overset{\text{Orientare}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_3 \doteq g(x_4, x_5)\} \overset{\text{Eliminare}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq g(x_4, x_5), x_3 \doteq g(x_4, x_5)\} \overset{\text{Conflict}}{\Rightarrow}$$

$$\bot.$$

Concluzie: $unif(\mathbb{P}) = \emptyset$.

7.4 Rezoluție de ordinul I

Rezoluția pentru logica de ordinul I este un sistem deductiv pentru clauze alcătuit din următoarele două reguli de inferență:

$$\text{Rezoluție Binară} \quad \frac{P(t_1, \ldots, t_n) \vee C_1 \quad \neg P(t'_1, \ldots, t'_n) \vee C_2}{\sigma^\flat(C_1 \vee C_2)} \quad \begin{array}{l} V_1 \cap V_2 = \emptyset \\ \sigma = mgu(\{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}) \end{array}$$

unde $V_1 = vars(P(t_1, \ldots, t_n) \vee C_1)$ și $V_2 = vars(\neg P(t'_1, \ldots, t'_n) \vee C_2)$.

$$\text{Factorizare Pozitivă} \quad \frac{P(t_1, \ldots, t_n) \vee P(t'_1, \ldots, t'_n) \vee C}{\sigma^\flat(P(t_1, \ldots, t_n) \vee C)} \quad \sigma = mgu(\{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\})$$

Observație.

- *În cazul în care clauzele care reprezintă ipotezele regulii* Rezoluție Binară *au variabile în comun ($V_1 \cap V_2 \neq \emptyset$), variabilele uneia dintre clauze trebuie redenumite înainte de a aplica regula (vezi exemplul de mai jos);*
- *În cazul în care problema de unificare care apare în regula de rezoluție nu are soluție, regula nu poate fi aplicată.*
- *Regula de factorizare pozitivă are o singură ipoteză.*
- *În cazul în care problema de unificare care apare în regula de factorizare nu are soluție, regula nu poate fi aplicată.*

- *Regula de factorizare pozitivă este necesară pentru completitudine (vezi exerciții seminar).*

Teorema 202 (Teorema rezoluției). *O formulă $\varphi = \forall x_1 \ldots \forall x_n.(C_1 \wedge C_2 \wedge \ldots C_m)$, aflată în FNSC, este nesatisfiabilă dacă și numai dacă $\square$ se poate obține din clauzele $C_1, \ldots, C_m$, aplicând regulile* Rezoluție Binară *și* Factorizare Pozitivă.

Exemplul 203. *Să demonstrăm că $\forall x.(P(x) \wedge (\neg P(h(x)) \vee Q(f(x))) \wedge (\neg Q(f(g(a)))))$ este nesatifiabilă, prin rezoluție de ordinul I:*

1. $P(x)$
2. $\neg P(h(x)) \vee Q(f(x))$
3. $\neg Q(f(g(a)))$
4. $Q(f(x))$ *rezoluție binară între 1 și 2:*
$$\frac{P(x') \qquad \neg P(h(x)) \vee Q(f(x))}{\sigma^{\flat}(Q(f(x)))}\ \sigma = \{x' \mapsto h(x)\} = mgu(\{x' \doteq h(x)\})$$
5. $\square$ *rezoluție între 4 și 3:*
$$\frac{Q(f(x)) \qquad \neg Q(f(g(a)))}{\sigma^{\flat}(\square)}\ \sigma = \{x \mapsto g(a)\} = mgu(\{f(x) \doteq f(g(a))\})$$

Exemplul 204. *Ne propunem să demonstrăm că $\varphi = \Big(\forall x.(P(x) \to Q(x))\Big) \wedge P(e) \to Q$ este validă folosind rezoluția pentru $\mathbb{LPI}$.*

Formula φ este validă dacă și numai dacă $\neg\varphi$ este nesatisfiabilă. Pentru aceasta, trebuie să punem formula $\neg\varphi$ în FNSC. Primul pas este de a pune formula $\neg\varphi$ în FNP:

$$\begin{aligned}
\neg\varphi = &\ \neg\Big((\forall x.(P(x) \to Q(x)) \wedge P(e)) \to Q(e)\Big)\\
\equiv &\ \neg\Big(\neg(\forall x.(P(x) \to Q(x)) \wedge P(e)) \vee Q(e)\Big)\\
\equiv &\ \neg\Big((\exists x.\neg((P(x) \to Q(x)) \wedge P(e))) \vee Q(e)\Big)\\
\equiv &\ \neg\Big(\exists x.\neg((P(x) \to Q(x)) \wedge P(e)) \vee Q(e)\Big)\\
\equiv &\ \forall x.\neg\Big(\neg((P(x) \to Q(x)) \wedge P(e)) \vee Q(e)\Big) \text{ în FNP}
\end{aligned}$$

Așadar formula $\varphi_1 = \forall x.\neg\Big(\neg((P(x) \to Q(x)) \wedge P(e)) \vee Q(e)\Big)$ este o FNP pentru φ și $\varphi \equiv \varphi_1$. Formula φ_1 este de asemenea închisă și nu are cuantificatori existențiali si deci nu necesită calcularea închiderii existențiale sau aplicării Skolemizării. Așadar, este in FNS.

Acum trebuie să punem formula φ_1 în FNSC:

$$\begin{aligned}\varphi_1 = &\ \forall x.\neg\Big(\neg\big((P(x) \to Q(x)) \wedge P(e)\big) \vee Q(e)\Big)\\ \equiv &\ \forall x.\neg\neg\big((P(x) \to Q(x)) \wedge P(e)\big) \wedge \neg Q(e)\\ \equiv &\ \forall x.\big((P(x) \to Q(x)) \wedge P(e)\big) \wedge \neg Q(e)\\ \equiv &\ \forall x.(\neg P(x) \vee Q(x)) \wedge P(e) \wedge \neg Q(e) \text{ in CSNF}\end{aligned}$$

Formula $\varphi_2 = \forall x.(\neg P(x) \vee Q(x)) \wedge P(e) \wedge \neg Q(e)$ este o FNSC pentru φ_1 și $\varphi_2 \equiv \varphi_1$.

Acum putem aplica rezoluția folosind clauzele din φ_2 pentru a demonstra că aceasta nu este satisfiabilă.

1. $\neg P(x) \vee Q(x)$
2. $P(e)$
3. $\neg Q(e)$
4. $\neg P(e)$ *Rezoluție binară între 1 și 3:*
$$\frac{\neg P(x) \vee Q(x) \qquad \neg Q(e)}{\sigma^\flat(\neg P(e))}\ \sigma = \{x \mapsto e\} = mgu(\{x \doteq e\})$$
5. $\square$ *Rezoluție binară între 2 și 4:*
$$\frac{P(e) \qquad \neg P(e)}{\sigma^\flat(\square)}\ \sigma = \emptyset = mgu(\{e \doteq e\})$$

Din aceasta rezultă că φ_2 nu este satisfiabilă.

Dar, $\varphi_2 \equiv \varphi_1$ și $\neg\varphi \equiv \varphi_1$ și deci $\neg\varphi \equiv \varphi_2$. Asta înseamnă că $\neg\varphi$ nu este satisfiabilă, și deci φ este validă.

7.5 Fișă de exerciții

Exercițiul 205. *Rezolvați următoarele probleme de unificare:*

1. $\{f(x,y) \doteq f(y,x), g(x) \doteq g(z)\}$;
2. $\{f(x,y) \doteq f(y,x), g(x) \doteq a\}$;
3. $\{f(f(x,y),z) \doteq f(y,x), g(z) \doteq g(a)\}$;
4. $\{f(g(x),y) \doteq f(y,z), z \doteq h(a)\}$;
5. $\{x_1 \doteq f(x_2,x_2), x_2 \doteq f(x_3,x_3), x_3 \doteq f(x_4,x_4)\}$.

Exercițiul 206. *Arătați că următoarele formule aflate în FNSC sunt nesatisfiabile, folosind rezoluția de ordinul I:*

1. $\forall x.\forall y.\forall z.\Big(\big(\neg P(x,z) \vee R(x,x,z)\big) \wedge \big(\neg R(e,x,e)\big) \wedge \big(P(e,y)\big)\Big)$;
2. $\forall x.\forall y.\Big(\big(\neg P(x,y) \vee Q(x) \vee Q(y)\big) \wedge \big(\neg Q(i(i(e)))\big) \wedge \big(P(i(x),i(x))\big)\Big)$.

Exercițiul 207. *Stabiliți prin rezoluție de ordinul I că următoarele formule sunt valide:*

1. $\Big(\forall x.(P(x) \to Q(x))\Big) \wedge P(e) \to Q(e)$
2. $\Big(\big(\forall x.\forall y.\forall z.(P(x,y) \wedge P(y,z) \to P(x,z))\big) \wedge P(x,y) \wedge P(y,x)\Big) \to P(x,x)$;
3. $(\forall x.Q(x)) \to (\exists x.Q(x))$;
4. $(\neg\forall x.Q(x)) \leftrightarrow (\exists x.\neg Q(x))$;
5. $(\neg\exists x.Q(x)) \leftrightarrow (\forall x.\neg Q(x))$;
6. $(\exists y.\forall x.P(x,y)) \to (\forall x.\exists y.P(x,y))$;
7. $(\forall x.(P(x,x) \leftrightarrow Q(x))) \to (P(e,e) \to Q(e))$.