

Logică pentru informatică - note de curs

Universitatea Alexandru Ioan Cuza, Iași

Facultatea de Informatică

Anul universitar 2021-2022

Ștefan Ciobâcă

Andrei Arusoai

Rodica Condurache

Cristian Masalagiu

Cuprins

1	Motivație și introducere	5
2	Structuri și semnături	7
2.1	Fișă de exerciții	10
3	Sintaxa logicii de ordinul I	13
3.1	Alfabetul	13
3.2	Termen	14
3.3	Formule atomice	15
3.4	Formule de ordinul I	16
3.5	Modelarea în \mathbb{LPI} a afirmațiilor din limba română	18
3.6	Modelarea în \mathbb{LPI} a afirmațiilor despre aritmetică	19
3.7	Fișă de exerciții	20
4	Semantica formulelor logicii de ordinul I	23
4.1	Atribuire	24
4.2	Valoarea de adevăr a unei formule de ordinul I	26
4.3	Satisfiabilitate într-o structură fixată	29
4.4	Validitate într-o structură fixată	29
4.5	Satisfiabilitate	30
4.6	Validitate	30
4.7	Consecință semantică	30
4.8	Mulțime consistentă de formule	31
4.9	Fișă de exerciții	31
5	Variabilele unei formule	33
5.1	Domeniul de vizibilitate al unui cuantificator - analogie cu limbajele de programare	35
5.2	Apariții libere și legate ale variabilelor	36
5.3	Variabile libere și variabile legate	38
5.4	Domeniul de vizibilitate și parantetizarea formulelor	39

6	Deducția naturală	41
6.1	Substituții	41
6.2	Secvențe	44
6.3	Reguli de inferență	45
6.4	Sistem deductiv	47
6.5	Demonstrație formală	47
6.6	Deducția naturală	48
6.6.1	Regulile pentru conjuncții	48
6.6.2	Regulile pentru implicații	49
6.6.3	Regulile pentru disjuncții	51
6.6.4	Regulile pentru negații	52
6.6.5	Eliminarea cuantificatorului universal.	54
6.6.6	Introducerea cuantificatorului existențial.	55
6.6.7	Introducerea cuantificatorului universal.	56
6.6.8	Eliminarea cuantificatorului existențial	57
6.6.9	Alte reguli	58
6.7	Sistemul deductiv al deducției naturale	58
6.8	Corectitudinea și completitudinea deducției naturale pentru logica de ordinul I	59
6.9	Fișă de exerciții	60

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 $p \in A$.

Acum să modelăm afirmația *Socrate este om*. Evident, acestei a doua afirmații trebuie să îi asociem o altă variabilă propozițională $q \in A$. Să presupunem că știm că p și q sunt adevărate. Formal, știm că lucrăm cu o atribuire $\tau : A \rightarrow B$ astfel încât $\tau(p) = 1$ și $\tau(q) = 1$. 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 r și nu putem trage nicio concluzie asupra lui $\tau(r)$ din faptul că $\tau(p) = 1$ și $\tau(q) = 1$. Deci, din semantica logicii propoziționale, nu putem trage concluzia că r este adevărată în orice atribuire în care p și 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: $\mathbf{0}$ și \mathbf{M} . Predicatul $\mathbf{0}$ va fi definit astfel: $\mathbf{0}(x)$ va fi adevărat când x este om. Predicatul $\mathbf{M}(x)$ este adevărat când 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

$$(\forall x. (\mathbf{0}(x) \rightarrow \mathbf{M}(x))),$$

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

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

Vom vedea că în logica de ordinul I, formula $\mathbf{M}(s)$ este consecință a formulelor $(\forall x. (\mathbf{0}(x) \rightarrow \mathbf{M}(x)))$ și respectiv $\mathbf{0}(s)$. 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 semnături

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:

$$\varphi = \left(\forall x. (\forall y. (x < y \rightarrow \exists z. (x < z \wedge z < y))) \right).$$

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 x, y, 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 $+$: $\mathbb{N}^2 \rightarrow \mathbb{N}$ 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. $(B, \emptyset, \{., +, \bar{\cdot}\})$;

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:

$$\varphi = \left(\forall x. (\forall y. (x < y \rightarrow \exists z. (x < z \wedge z < y))) \right),$$

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 semnături:

Definiția 2 (Signatură). O semnatură 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 semnături îi putem asocia mai multe structuri:

Definiția 3 (Σ -structuri). Dacă $\Sigma = (\mathcal{P}, \mathcal{F})$ este o semnatură, 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 $\Sigma = (\{\mathbf{P}, \mathbf{Q}\}, \{\mathbf{f}, \mathbf{i}, \mathbf{a}, \mathbf{b}\})$ unde \mathbf{P}, \mathbf{Q} sunt simboluri predicative de aritate $ar(\mathbf{P}) = ar(\mathbf{Q}) = 2$ și $\mathbf{f}, \mathbf{i}, \mathbf{a}, \mathbf{b}$ sunt simboluri funcționale cu aritățile: $ar(\mathbf{f}) = 2$, $ar(\mathbf{i}) = 1$ și $ar(\mathbf{a}) = ar(\mathbf{b}) = 0$.

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

Observație. După cum se poate observa și în Exemplul 4, pentru simboluri predicative (e.g., \mathbf{P}, \mathbf{Q}) vom utiliza o culoare diferită față de culoarea simbolurilor funcționale (e.g., $\mathbf{f}, \mathbf{i}, \mathbf{a}, \mathbf{b}$). 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 semnături Σ îi putem asocia mai multe structuri, numite Σ -structuri.

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

2.1 Fișă de exerciții

Exercițiul 5. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

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 6. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

Suma a două numere pare este un număr par.

Exercițiul 7. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

În jocul de șah, regina poate efectua o mutare dintr-o căsuță într-alta ddacă tura sau nebunul poate efectua aceeași mutare.

Exercițiul 8. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

Suma a două numere mai mari decât zero este mai mare decât zero.

Exercițiul 9. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

Numărul 7 este prim.

Exercițiul 10. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

Orice număr par mai mare decât 2 este suma a două numere prime.

Exercițiul 11. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

Dacă Pământul este plat, atunci $2 + 2 = 5$.

Exercițiul 12. *Identificați predicatele și funcțiile din textul de mai jos și stabiliți aritatea acestora. Care este domeniul acestora?*

Pentru orice $\epsilon \in (0, \infty)$, există $\delta_\epsilon \in (0, \infty)$ astfel încât pentru orice $x \in \mathbb{R}$ cu $d(x_0, x) < \delta_\epsilon$, avem $d(f(x_0), f(x)) < \epsilon$.

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 semnături Σ . Practic, există mai multe limbaje de ordinul I, câte un limbaj pentru fiecare semnătură Σ .

În continuare, vom presupune fixată o semnătură 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: $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \perp$, precum și doi *cuan-tificatori*: \forall, \exists ;
2. *variabile*: vom fixa o mulțime infinit numărabilă de variabile, notată $\mathcal{X} = \{x, y, z, x', y', x_1, z'', \dots\}$ (a nu se confunda cu mulțimea variabilelor propoziționale din logica propozițională; sunt două noțiuni fundamentale diferite și din acest motiv utilizăm altă culoare pentru a le reprezenta);
3. simboluri auxiliare: $(,), \cdot, \cdot\cdot, (,),$ și $;$;
4. simboluri suplimentare, care sunt specifice fiecărei semnături $\Sigma = (\mathcal{P}, \mathcal{F})$ în parte: simbolurile funcționale din mulțimea \mathcal{F} și respectiv simbolurile predicative din mulțimea \mathcal{P} .

3.2 Termen

Definiția 13. Mulțimea termenilor, \mathcal{T} , este cea mai mică mulțime 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 $n > 0$) și $t_1, \dots, t_n \in \mathcal{T}$, atunci $f(t_1, \dots, t_n) \in \mathcal{T}$ (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 14. Fie signatura $\Sigma = (\{\mathbf{P}, \mathbf{Q}\}, \{\mathbf{f}, \mathbf{i}, \mathbf{a}, \mathbf{b}\})$ definită în Exemplul 4, unde $ar(\mathbf{P}) = ar(\mathbf{Q}) = 2$, $ar(\mathbf{f}) = 2$, $ar(\mathbf{i}) = 1$, $ar(\mathbf{a}) = ar(\mathbf{b}) = 0$. Iată câteva exemple de termeni: \mathbf{a} , \mathbf{b} , \mathbf{x} , \mathbf{y} , \mathbf{x}_1 , \mathbf{y}' , $\mathbf{i}(\mathbf{a})$, $\mathbf{i}(\mathbf{x})$, $\mathbf{i}(\mathbf{i}(\mathbf{a}))$, $\mathbf{i}(\mathbf{i}(\mathbf{x}))$, $\mathbf{f}(\mathbf{a}, \mathbf{b})$, $\mathbf{i}(\mathbf{f}(\mathbf{a}, \mathbf{b}))$, $\mathbf{f}(\mathbf{f}(\mathbf{x}, \mathbf{a}), \mathbf{f}(\mathbf{y}, \mathbf{y}))$.

Exercițiul 15. Identificați în lista de mai jos Σ -termenii:

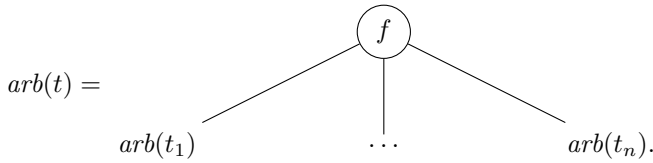
1. $\mathbf{i}(\mathbf{i}(\mathbf{x}))$;
2. \mathbf{i} ;
3. $\mathbf{f}(\mathbf{x}, \mathbf{x})$;
4. $\mathbf{P}(\mathbf{a}, \mathbf{b})$;
5. $\mathbf{i}(\mathbf{a}, \mathbf{a})$;
6. $\mathbf{f}(\mathbf{i}(\mathbf{x}), \mathbf{i}(\mathbf{x}))$;
7. $\mathbf{f}(\mathbf{i}(\mathbf{x}, \mathbf{x}))$;
8.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

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

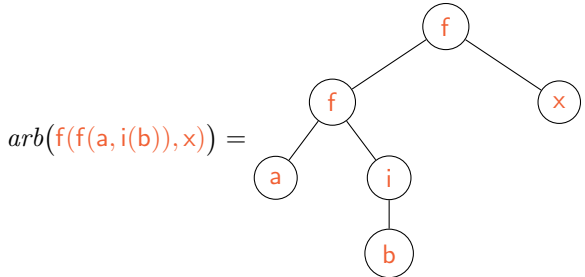
1. dacă $t = c$ și $c \in \mathcal{F}_0$, atunci $arb(t) = \bigcirc(c)$

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

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



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 memorati sub formă de arbori cu rădăcină. Iată arborele atașat termenului $f(f(a, i(b)), x)$:



Exercițiul 16. Calculați arborii de sintaxă pentru termenii din Exemplul 14.

3.3 Formule atomice

Definiția 17 (Formulă atomică). O formulă atomică este orice șir de simboluri de forma $P(t_1, \dots, t_n)$, unde $P \in \mathcal{P}_n$ este un simbol predicativ de aritate $n \geq 0$, iar $t_1, \dots, t_n \in \mathcal{T}$ sunt termeni. Dacă $n = 0$, scriem P în loc de $P()$.

Exemplul 18. Continuând Exemplul 14, folosim signatura

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

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

Iată câteva exemple de formule atomice: $\mathbf{P}(\mathbf{a}, \mathbf{b})$, $\mathbf{P}(\mathbf{x}, \mathbf{y})$, $\mathbf{Q}(\mathbf{i}(\mathbf{i}(\mathbf{x})), \mathbf{f}(\mathbf{x}, \mathbf{x}))$, $\mathbf{Q}(\mathbf{a}, \mathbf{b})$, $\mathbf{P}(\mathbf{f}(\mathbf{f}(\mathbf{a}, \mathbf{i}(\mathbf{x})), \mathbf{b}), \mathbf{i}(\mathbf{x}))$.

Exercițiul 19. Explicați de ce $\mathbf{P}(\mathbf{a})$, $\mathbf{P}, \mathbf{i}(\mathbf{i}(\mathbf{x}))$ nu sunt formule atomice peste signatura din Exemplul 18.

3.4 Formule de ordinul I

Definiția 20 (Formule de ordinul I). *Mulțimea formulelor de ordinul I, notată \mathbb{LPI} , este cea mai mică mulțime astfel încât:*

1. (cazul de bază) orice formulă atomică este formulă (adică $P(t_1, \dots, t_n) \in \mathbb{LPI}$ pentru orice simbol predicativ $P \in \mathcal{P}_n$ și orice termeni $t_1, \dots, t_n \in \mathcal{T}$;
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) $(\varphi_1 \rightarrow \varphi_2) \in \mathbb{LPI}$;
- (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 20, regăsim conectorii logici $\neg, \wedge, \vee, \rightarrow$ 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 21. Continuând Exemplul 14, folosim signatura $\Sigma = (\{\mathbf{P}, \mathbf{Q}\}, \{\mathbf{f}, \mathbf{i}, \mathbf{a}, \mathbf{b}\})$, unde $ar(\mathbf{P}) = ar(\mathbf{Q}) = 2$, $ar(\mathbf{f}) = 2$, $ar(\mathbf{i}) = 1$ și $ar(\mathbf{a}) = ar(\mathbf{b}) = 0$.

Iată câteva exemple de formule din logica de ordinul I:

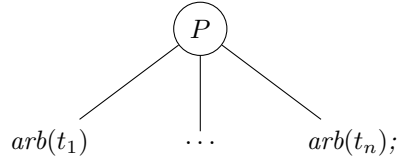
1. $\mathbf{P}(\mathbf{a}, \mathbf{b})$;
2. $\mathbf{Q}(\mathbf{a}, \mathbf{b})$;
3. $\mathbf{P}(\mathbf{a}, \mathbf{x})$;
4. $\neg \mathbf{P}(\mathbf{a}, \mathbf{b})$;
5. $(\mathbf{P}(\mathbf{a}, \mathbf{b}) \wedge \neg \mathbf{Q}(\mathbf{a}, \mathbf{b}))$;
6. $(\mathbf{P}(\mathbf{a}, \mathbf{b}) \vee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

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

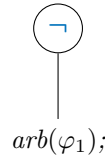
10. $(\exists x.\neg Q(x, y)).$

Definiția 22 (Arborele de sintaxă abstractă asociat formulelor din \mathbb{LPI}).
Formulele au asociat un arbore de sintaxă abstractă definit în cele ce urmează:

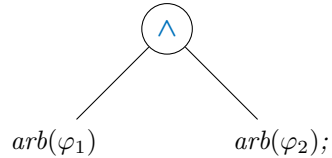
1. dacă $\varphi = P(t_1, \dots, t_n)$, atunci $arb(\varphi) =$



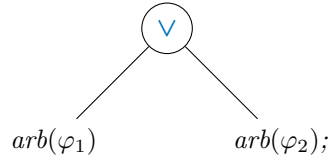
2. dacă $\varphi = \neg\varphi_1$, atunci $arb(\varphi) =$



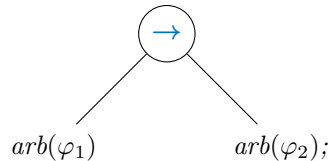
3. dacă $\varphi = (\varphi_1 \wedge \varphi_2)$, atunci $arb(\varphi) =$



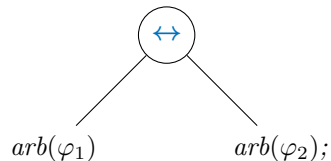
4. dacă $\varphi = (\varphi_1 \vee \varphi_2)$, atunci $arb(\varphi) =$



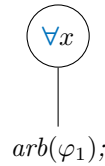
5. dacă $\varphi = (\varphi_1 \rightarrow \varphi_2)$, atunci $arb(\varphi) =$



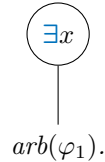
6. dacă $\varphi = (\varphi_1 \leftrightarrow \varphi_2)$, atunci $arb(\varphi) =$



7. dacă $\varphi = (\forall x.\varphi_1)$, atunci $\text{arb}(\varphi) =$



8. dacă $\varphi = (\exists x.\varphi_1)$, atunci $\text{arb}(\varphi) =$



Exercițiul 23. Calculați arborele de sintaxă asociat formulelor din Exemplul 21.

3.5 Modelarea în 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 **O** pentru primul predicat și simbolul predicativ **M** pentru al doilea predicat. De asemenea, în text avem și o constantă: Socrate. Alegem simbolul funcțional **s** de aritate 0 pentru această constantă. Așadar, pentru a modela afirmațiile de mai sus, vom lucra cu signatura

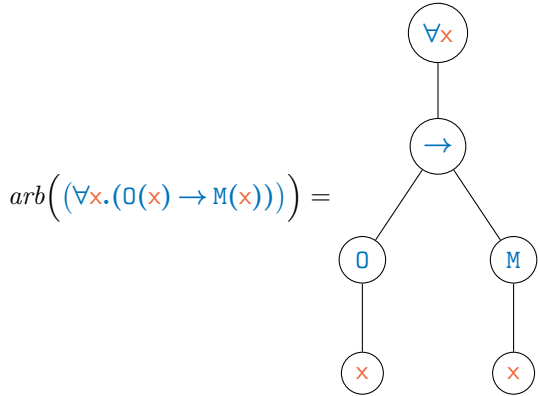
$$\Sigma = (\{\mathbf{O}, \mathbf{M}\}, \{\mathbf{s}\}),$$

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

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

$$(\forall x.(\mathbf{O}(x) \rightarrow \mathbf{M}(x))),$$

al cărei arbore de sintaxă abstractă este:



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

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

1. D este mulțimea tuturor ființelor de pe Pământ;
2. $O^S(x)$ este adevărat pentru orice ființă x care este și om;
3. $M^S(x)$ este adevărat pentru orice ființă x (toate elementele domeniului sunt muritoare);
4. 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 $(\forall x. (O(x) \rightarrow M(x)))$, $O(s)$ și respectiv $M(s)$, 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 $M(s)$ este în mod necesar adevărată în *orice* structură în care formulele $O(s)$ și $(\forall x. (O(x) \rightarrow M(x)))$ sunt adevărate, nu doar în structura S de mai sus.

3.6 Modelarea în 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. $(\forall x.(\forall y.(<(x, y) \rightarrow \exists z.(<(x, z) \wedge <(z, y)))));$
2. $(\forall x.(\forall y.(\exists z.(= (+ (x, y), z)))));$
3. $(\forall x.(<(0, x) \vee = (0, x)));$
4. $(\forall x.(\exists y.(= (x, -(y)))));$
5. $= (+ (x, y), z).$

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

1. $(\forall x.(\forall y.(x < y \rightarrow \exists z.(x < z \wedge z < y))));$
2. $(\forall x.(\forall y.(\exists z.(x + y = z))));$
3. $(\forall x.(0 < x \vee 0 = x));$
4. $(\forall x.(\exists y.(x = -(y))));$
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 formulă 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).

3.7 Fișă de exerciții

Exercițiul 24. *Identificați o semnătură 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 25. 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 semnătură Σ pentru structura S de mai sus.
2. Modelați următoarele afirmații ca formule de ordinul I în semnătura 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 26. Dați exemplu de 5 termeni peste semnăturile de la Exercițiul 25 și calculați arborele de sintaxă abstractă al acestor termeni.

Exercițiul 27. Dați exemplu de 5 formule peste semnătura de la Exercițiul 25 și calculați arborele de sintaxă abstractă al acestora.

Exercițiul 28. *Calculați arborele de sintaxă abstractă al următoarelor formule:*

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

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 semnătură $\Sigma = (\mathcal{P}, \mathcal{F})$ 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 semnătura $\Sigma = (\{\mathbf{P}\}, \{\mathbf{f}, \mathbf{i}, \mathbf{e}\})$, unde \mathbf{P} este simbol predicativ de aritate 2, iar \mathbf{f} , \mathbf{i} și \mathbf{e} sunt simboluri funcționale de aritate 2, 1 și respectiv 0. Altfel spus, $\mathcal{P}_2 = \{\mathbf{P}\}$, $\mathcal{P}_1 = \emptyset$, $\mathcal{P}_0 = \emptyset$, $\mathcal{F}_2 = \{\mathbf{f}\}$, $\mathcal{F}_1 = \{\mathbf{i}\}$ și $\mathcal{F}_0 = \{\mathbf{e}\}$.

Reamintim și că, dacă $\Sigma = (\mathcal{P}, \mathcal{F})$ este o semnătură, 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 29. *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 \mathbf{P} este = (predicatul de egalitate pentru numere întregi), funcția + este funcția de adunare a numerelor întregi asociată simbolului funcțional \mathbf{f} , - este funcția minus unar asociată simbolului funcțional \mathbf{i} , iar simbolul constant \mathbf{e} are asociată constanta 0.

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

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

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

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

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

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 30 (Atribuire). *Fie Σ o semnătură și S o Σ -structură cu domeniul D . Se numește S -atribuire orice funcție*

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

Exemplul 31. *Funcția $\alpha_1 : \mathcal{X} \rightarrow \mathbb{Z}$, definită ca mai jos, este o S_1 -atribuire:*

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

Exemplul 32. *Funcția $\alpha_2 : \mathcal{X} \rightarrow \mathbb{Z}$, definită ca mai jos, este o S_1 -atribuire:*

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

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 $\bar{\alpha}$,

$$\bar{\alpha} : \mathcal{T} \rightarrow D,$$

dată în definiția care urmează.

Definiția 33 (Valoarea unui termen într-o atribuire). *Dându-se o S -atribuire α și un termen $t \in \mathcal{T}$ peste semnatura Σ , valoarea termenului t în atribuirea α este un element al domeniului D notat cu $\bar{\alpha}(t)$ și calculat recursiv astfel:*

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

Exemplul 34. Continuând Exemplul 31, unde α_1 este o S_1 -atribuire, avem:

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

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

Definiția 35 (Actualizarea unei atribuii). 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 :

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

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

Exemplul 36. De exemplu, atribuirea $\alpha_1[x_1 \mapsto 6]$ este exact atribuirea α_2 definită în exemplele de mai sus. Valoarea termenului $\mathbf{f}(\mathbf{i}(\mathbf{x}_1), \mathbf{e})$ în atribuirea $\alpha_1[x_1 \mapsto 6]$, notată cu $\overline{\alpha_1[x_1 \mapsto 6]}(\mathbf{f}(\mathbf{i}(\mathbf{x}_1), \mathbf{e}))$, este -6 .

Exercițiul 37. Calculați valorile de mai jos:

1. $\overline{\alpha_1[x_1 \mapsto 10]}(\mathbf{f}(\mathbf{i}(\mathbf{x}_1), \mathbf{e}))$;
2. $\overline{\alpha_1[x_2 \mapsto 10]}(\mathbf{f}(\mathbf{i}(\mathbf{x}_1), \mathbf{e}))$;
3. $\overline{\alpha_1[x_2 \mapsto 10][x_1 \mapsto 10]}(\mathbf{f}(\mathbf{i}(\mathbf{x}_1), \mathbf{e}))$.

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 semnătură Σ . Această valoare se poate calcula doar într-o Σ -structură S , cu ajutorul unei S -atribuii α .

Notățiile folosite sunt similare cu cele pentru logica propozițională. Astfel, notăm faptul că o formulă φ este adevărată într-o structură S cu o atribuire 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 $S, \alpha \not\models \varphi$.

Notăția $S, \alpha \models \varphi$ 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 38. *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. $S, \alpha \models P(t_1, \dots, t_n)$ ddacă $P^S(\bar{\alpha}(t_1), \dots, \bar{\alpha}(t_n))$;
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. $S, \alpha \models (\varphi_1 \rightarrow \varphi_2)$ 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 39. *Vom lucra în continuare peste signatura $\Sigma = (\{P\}, \{f, i, e\})$, Σ -structura $S_1 = (\mathbb{Z}, \{=\}, \{+, -, 0\})$ definită la începutul capitolului și S_1 -atribuirile α_1, α_2 .*

Avem că

$$\begin{array}{ll}
 S_1, \alpha_1 \models P(\mathbf{x}_1, \mathbf{x}_1) & \text{ddacă} \quad P^{S_1}(\bar{\alpha}_1(\mathbf{x}_1), \bar{\alpha}_1(\mathbf{x}_1)) \\
 & \text{ddacă} \quad \bar{\alpha}_1(\mathbf{x}_1) = \bar{\alpha}_1(\mathbf{x}_1) \\
 & \text{ddacă} \quad \alpha_1(\mathbf{x}_1) = \alpha_1(\mathbf{x}_1) \\
 & \text{ddacă} \quad 5 = 5.
 \end{array}$$

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

Exemplul 40. *Continuând exemplul anterior, avem*

$$\begin{array}{ll}
 S_1, \alpha_1 \models P(\mathbf{x}_1, \mathbf{x}_3) & \text{ddacă} \quad P^{S_1}(\bar{\alpha}_1(\mathbf{x}_1), \bar{\alpha}_1(\mathbf{x}_3)) \\
 & \text{ddacă} \quad \bar{\alpha}_1(\mathbf{x}_1) = \bar{\alpha}_1(\mathbf{x}_3) \\
 & \text{ddacă} \quad \alpha_1(\mathbf{x}_1) = \alpha_1(\mathbf{x}_3) \\
 & \text{ddacă} \quad 5 = 6.
 \end{array}$$

Din moment ce $5 \neq 6$, rezultă că $S_1, \alpha_1 \not\models P(x_1, x_3)$, adică formula $P(x_1, x_3)$ este falsă în structura S_1 cu atribuirea α_1 . Altfel spus S_1 nu satisface $P(x_1, x_3)$ cu atribuirea α_1 .

Exemplul 41. Continuând exemplul anterior, avem

$S_1, \alpha_1 \models \neg P(x_1, x_3)$	ddacă	$S_1, \alpha_1 \not\models P(x_1, x_3)$
	ddacă	nu $P^{S_1}(\overline{\alpha_1}(x_1), \overline{\alpha_1}(x_3))$
	ddacă	nu $\overline{\alpha_1}(x_1) = \overline{\alpha_1}(x_3)$
	ddacă	$\overline{\alpha_1}(x_1) \neq \overline{\alpha_1}(x_3)$
	ddacă	$\alpha_1(x_1) \neq \alpha_1(x_3)$
	ddacă	$5 \neq 6$.

Din moment ce $5 \neq 6$, rezultă că $S_1, \alpha_1 \models \neg P(x_1, x_3)$, adică formula $\neg P(x_1, x_3)$ este adevărată în structura S_1 cu atribuirea α_1 . Altfel spus, S_1 satisface $\neg P(x_1, x_3)$ cu atribuirea α_1 .

Exemplul 42. Continuând exemplul anterior, avem

$S_1, \alpha_1 \models P(x_1, x_1) \wedge \neg P(x_1, x_3)$	ddacă
$S_1, \alpha_1 \models P(x_1, x_1)$ și $S_1, \alpha_1 \models \neg P(x_1, x_3)$	ddacă
... și ...	ddacă
$5 = 5$ și $5 \neq 6$.	

Din moment ce $5 = 5$ și $5 \neq 6$, rezultă că $S_1, \alpha_1 \models P(x_1, x_1) \wedge \neg P(x_1, x_3)$.

Exemplul 43. Continuând exemplul anterior, avem

$S_1, \alpha_1 \models P(x_1, x_3) \vee P(x_1, x_1)$ dacă $S_1, \alpha_1 \models P(x_1, x_3)$ sau $S_1, \alpha_1 \models P(x_1, x_1)$.

Am stabilit deja că $S_1, \alpha_1 \models P(x_1, x_3)$, deci $S_1, \alpha_1 \models P(x_1, x_3) \vee P(x_1, x_1)$ (chiar dacă $S_1, \alpha_1 \not\models P(x_1, x_1)$).

Exemplul 44. Continuând exemplul anterior, avem

$S_1, \alpha_1 \models \exists x_1. P(x_1, x_3)$	ddacă
există $u \in D$ a.î. $S_1, \alpha_1[x_1 \mapsto u] \models P(x_1, x_3)$	ddacă
există $u \in D$ a.î. $P^{S_1}(\overline{\alpha_1[x_1 \mapsto u]}(x_1), \overline{\alpha_1[x_1 \mapsto u]}(x_3))$	ddacă
există $u \in D$ a.î. $\overline{\alpha_1[x_1 \mapsto u]}(x_1) = \overline{\alpha_1[x_1 \mapsto u]}(x_3)$	ddacă
există $u \in D$ a.î. $\alpha_1[x_1 \mapsto u](x_1) = \alpha_1[x_1 \mapsto u](x_3)$	ddacă
există $u \in D$ a.î. $u = \alpha_1(x_3)$	ddacă
există $u \in D$ a.î. $u = 6$.	

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

Exemplul 45. Continuând exemplul anterior, avem

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

Exercițiul 46. Arătați că $S_1, \alpha_1 \models \forall x_1. \exists x_3. P(x_1, i(x_3))$.

4.3 Satisfiabilitate într-o structură fixată

Definiția 47 (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 48. Formula $P(x_1, x_3)$ este satisfiabilă în structura S_1 , deoarece există o atribuire, de exemplu α_2 , cu proprietatea că $S_1, \alpha_2 \models P(x_1, x_3)$.

Exercițiul 49. Arătați că formula $\neg P(x_1, x_1)$ nu este satisfiabilă în structura S_1 (deoarece, pentru orice atribuire α aleasă, avem că $S_1, \alpha \not\models \neg P(x_1, x_1)$).

4.4 Validitate într-o structură fixată

Definiția 50 (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 51. Formula $P(x_1, x_3)$ nu este validă în structura S_1 , deoarece există o atribuire, și anume α_1 , cu proprietatea că $S_1, \alpha_1 \not\models P(x_1, x_3)$.

Exercițiul 52. Arătați că formula $P(x_1, x_1)$ este validă în structura S_1 (deoarece orice atribuire α am alege, $S_1, \alpha \models P(x_1, x_1)$).

4.5 Satisfiabilitate

Definiția 53 (Satisfiabilitate). *O formulă φ este satisfiabilă dacă există o structură S și o S -atribuire α cu proprietatea că*

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

Exemplul 54. Formula $\neg P(x_1, x_1)$ este satisfiabilă, deoarece există o structură (de exemplu S_5) și o S_5 -atribuire (de exemplu α_1) astfel încât $S_5, \alpha_1 \models \neg P(x_1, x_1)$ (deoarece $5 \not\leq 5$).

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 $\neg P(x_1, x_1)$ nu este satisfiabilă în structura S_1) și totuși să fie satisfiabilă (vezi Exemplul 54 unde aceeași formulă $\neg P(x_1, x_1)$ este satisfiabilă).

4.6 Validitate

Definiția 55 (Validitate). *O formulă φ este validă dacă, pentru orice structură S și pentru orice S -atribuire α , avem*

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

Exemplul 56. Formula $P(x_1, x_1)$ nu este validă, deoarece $S_5, \alpha_1 \not\models P(x_1, x_1)$. Pe de altă parte, formula $P(x_1, x_1) \rightarrow P(x_1, x_1)$ este validă.

Observație. O formulă poate să fie validă într-o structură fixată (de exemplu $P(x_1, x_1)$ este validă în structura S_1) și totuși să nu fie validă (de exemplu, $P(x_1, x_1)$ nu este validă, deoarece $S_5, \alpha_1 \not\models P(x_1, x_1)$).

4.7 Consecință semantică

Definiția 57. *O formulă φ este consecință semantică a formulelor $\varphi_1, \dots, \varphi_n$ într-o structură fixată S , notat $\varphi_1, \dots, \varphi_n \models_S \varphi$, dacă, pentru orice S -atribuire α pentru care $S, \alpha \models \varphi_1$, $S, \alpha \models \varphi_2$, ..., $S, \alpha \models \varphi_n$, avem că $S, \alpha \models \varphi$.*

Exemplul 58. Avem că $P(x, y) \models_{S_1} P(y, x)$, deoarece, pentru orice S_1 -atribuire α cu proprietatea că $S_1, \alpha \models P(x, y)$ (adică $\alpha(x) = \alpha(y)$), avem și că $S_1, \alpha \models P(y, x)$ (adică $\alpha(y) = \alpha(x)$).

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

Definiția 59. *O formulă φ este consecință semantică a formulelor $\varphi_1, \dots, \varphi_n$, notat $\varphi_1, \dots, \varphi_n \models \varphi$, dacă*

$$\varphi_1, \dots, \varphi_n \models_S \varphi$$

pentru orice structură S .

Exemplul 60. *Avem că $P(x, y) \not\models P(y, x)$, deoarece există o structură (și anume S_5) astfel încât $P(x, y) \models_{S_5} P(y, x)$.*

Exercițiul 61. *Arătați că:*

$$\forall x. \forall y. \forall z. (P(x, y) \wedge P(y, z) \rightarrow P(x, z)), P(x_1, x_2), P(x_2, x_3) \models P(x_1, x_3).$$

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

4.8 Mulțime consistentă de formule

Definiția 62. *O mulțime de formule Γ este consistentă într-o structură S dacă, prin definiție,*

$$\text{există } \alpha \text{ a.î. } S, \alpha \models \varphi \text{ pentru orice } \varphi \in \Gamma.$$

Exemplul 63. *Avem că $\{P(x, y), P(y, x)\}$ este consistentă în S_1 , dar nu în S_4 .*

Definiția 64. *O mulțime de formule Γ este consistentă dacă, prin definiție, există o structură în care este consistentă.*

Exemplul 65. *Avem că $\{P(x, y), P(y, x)\}$ este consistentă (deoarece este consistentă în măcar o structură, de exemplu S_1).*

Exemplul 66. *Avem că $\{P(x, y), \neg P(x, y)\}$ nu este consistentă.*

4.9 Fișă de exerciții

Amintim mai jos structurile din Exemplul 29:

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 67. *Stabiliți dacă:*

1. $S_1, \alpha_1 \models P(x_2, x_3)$;
2. $S_1, \alpha_1 \models \neg P(x_2, x_3)$;
3. $S_1, \alpha_1 \models \neg P(x_2, x_3) \wedge P(x_1, x_1)$;
4. $S_1, \alpha_1 \models \exists x_3. P(x_2, x_3)$;
5. $S_1, \alpha_1 \models \forall x_2. \exists x_3. P(x_2, x_3)$;
6. $S_1, \alpha_1 \models \exists x_3. \forall x_2. P(x_2, x_3)$;
7. $S_1, \alpha_2 \models \forall x_2. \exists x_3. P(x_2, i(x_3))$;

Exercițiul 68. *Găsiți pentru fiecare dintre itemii de mai jos câte o S_2 -atribuire α_3 astfel încât:*

1. $S_2, \alpha_3 \models P(x_1, x_2)$;
2. $S_2, \alpha_3 \models P(f(x_1, x_2), x_3)$;
3. $S_2, \alpha_3 \models P(f(x_1, x_2), i(x_3))$;
4. $S_2, \alpha_3 \models P(x, e)$;
5. $S_2, \alpha_3 \models \exists y. P(x, i(y))$;
6. $S_2, \alpha_3 \models \forall y. \exists x. P(x, i(y))$.

Exercițiul 69. *Arătați că următoarele formule sunt valide în S_2 :*

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

Exercițiul 70. *Arătați că formula $\forall x. \exists y. P(x, i(y))$ nu este validă în S_3 .*

Exercițiul 71. *Găsiți o formulă care să fie satisfiabilă în S_1 dar nu în S_3 .*

Exercițiul 72. *Arătați că formula $\forall x. \exists y. P(x, y)$ nu este validă.*

Exercițiul 73. *Arătați că formula $(\forall x. P(x, x)) \rightarrow \exists x_2. P(x_1, x_2)$ este validă.*

Exercițiul 74. *Arătați că formula $\forall x. \exists y. P(y, x)$ nu este validă.*

Exercițiul 75. *Arătați că formula $\forall x. \neg P(x, x)$ este satisfiabilă.*

Exercițiul 76. *Arătați că formula $\forall x. \neg P(x, x) \wedge \exists x. P(x, x)$ nu este satisfiabilă.*

Capitolul 5

Variabilele unei formule

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

În primul rând, definim o funcție $\text{vars} : \mathcal{T} \rightarrow 2^{\mathcal{X}}$ (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 semnătură 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 77. *Funcția $\text{vars} : \mathcal{T} \rightarrow 2^{\mathcal{X}}$ este definită recursiv după cum urmează:*

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

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

Definiția 78. *Funcția $\text{vars} : \mathbb{LPI} \rightarrow 2^{\mathcal{X}}$ este definită recursiv după cum urmează:*

1. $\text{vars}(P(t_1, \dots, t_n)) = \bigcup_{i \in \{1, \dots, n\}} \text{vars}(t_i)$;

2. $\text{vars}(\neg\varphi) = \text{vars}(\varphi);$
3. $\text{vars}((\varphi_1 \wedge \varphi_2)) = \text{vars}(\varphi_1) \cup \text{vars}(\varphi_2);$
4. $\text{vars}((\varphi_1 \vee \varphi_2)) = \text{vars}(\varphi_1) \cup \text{vars}(\varphi_2);$
5. $\text{vars}((\varphi_1 \rightarrow \varphi_2)) = \text{vars}(\varphi_1) \cup \text{vars}(\varphi_2);$
6. $\text{vars}((\varphi_1 \leftrightarrow \varphi_2)) = \text{vars}(\varphi_1) \cup \text{vars}(\varphi_2);$
7. $\text{vars}((\forall x.\varphi)) = \text{vars}(\varphi) \cup \{x\};$
8. $\text{vars}((\exists x.\varphi)) = \text{vars}(\varphi) \cup \{x\}.$

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 79. Fie formula φ :

$$\left(\left(\forall x.(P(x, y) \wedge \exists y.(P(z, f(x, y)) \wedge P(x, y))) \right) \wedge P(x, x) \right).$$

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

Exercițiul 80. Fie semnatura $\Sigma = (\{P, Q\}, \{f, i, a, b\})$, unde $\text{ar}(P) = \text{ar}(Q) = 2$, $\text{ar}(f) = 2$, $\text{ar}(i) = 1$, $\text{ar}(a) = \text{ar}(b) = 0$. Calculați $\text{vars}(\varphi)$ pentru fiecare formulă φ de mai jos:

1. $P(x, y);$
2. $Q(a, b);$
3. $P(a, x);$
4. $\neg P(x, z);$
5. $(P(x, x) \wedge \neg Q(x, z));$
6. $(P(x, b) \vee \neg Q(z, y));$
7. $((P(x, b) \rightarrow P(x, z)) \leftrightarrow (P(x, b) \rightarrow P(a, z)));$
8. $(\forall x.P(a, x));$
9. $(\exists x.\neg Q(x, y));$
10. $\left((\exists x.\neg Q(x, y)) \wedge (\forall y.P(y, x)) \right).$

5.1 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 doua declarații a variabilei x .

Un fenomen similar se întâmplă în formulele logicii de ordinul I. De exemplu, în formula $(\forall x.(\forall y.(P(x, y) \wedge P(x, z) \wedge (\exists x.P(x, y)))))$, 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 $(\forall y.(P(x, y) \wedge P(x, z) \wedge (\exists x.P(x, y))))$, în timp ce domeniul de vizibilitate D_2 al variabilei x cuantificate existențial este $P(x, y)$:

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

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*.

5.2 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 81 (Apariție liberă). *O apariție 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 $\exists x$.*

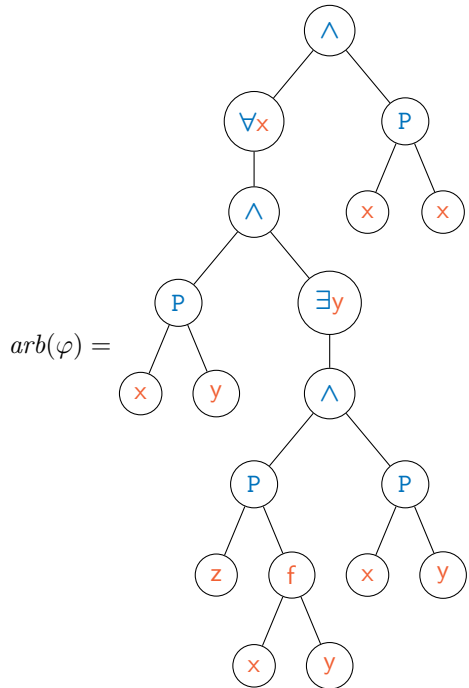
Definiția 82 (Apariție legată). *O apariție legată a unei variabile x într-o formulă φ este dată de un nod în arborele formulei etichetat cu x și care are proprietatea că, mergând din nod înspre rădăcină, întâlnim măcar un nod etichetat cu $\forall x$ sau cu $\exists x$.*

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 83. *Considerăm în continuare formula*

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

Arborele de sintaxă abstractă al formulei φ este:



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

$$\varphi = \left(\left(\forall x. \left(P(x, \underline{y}) \wedge \exists y. \left(P(\underline{z}, f(x, \underline{y})) \wedge P(x, y) \right) \right) \right) \wedge P(\underline{x}, \underline{x}) \right).$$

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

$$\varphi = \left(\left(\forall x. \left(P(\underline{x}, \underline{\underline{y}}) \wedge \exists y. \left(P(\underline{z}, f(\underline{x}, \underline{\underline{y}})) \wedge P(\underline{x}, \underline{\underline{y}}) \right) \right) \right) \wedge P(\underline{x}, \underline{x}) \right).$$

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/denumeste un cuantificator (sau, prin care se leagă variabila x).

5.3 Variabile libere și variabile legate

Mulțimea variabilelor unei formule φ care au cel puțin o apariție liberă se notează $free(\varphi)$.

Definiția 84. Funcția $free : \mathbb{LPI} \rightarrow 2^{\mathcal{X}}$ este definită recursiv în modul următor:

1. $free(P(t_1, \dots, t_n)) = vars(t_1) \cup \dots \cup vars(t_n)$;
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. $free((\varphi_1 \rightarrow \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
6. $free((\varphi_1 \leftrightarrow \varphi_2)) = free(\varphi_1) \cup free(\varphi_2)$;
7. $free((\forall x.\varphi)) = free(\varphi) \setminus \{x\}$;
8. $free((\exists x.\varphi)) = free(\varphi) \setminus \{x\}$.

Exemplul 85. Pentru formula

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

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

Exercițiul 86. Calculați $free(\varphi)$ pentru fiecare formulă φ din Exercițiul 80.

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 87. Funcția $bound : \mathbb{LPI} \rightarrow 2^{\mathcal{X}}$ este definită recursiv astfel:

1. $bound(P(t_1, \dots, t_n)) = \emptyset$;
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. $bound((\varphi_1 \rightarrow \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;
6. $bound((\varphi_1 \leftrightarrow \varphi_2)) = bound(\varphi_1) \cup bound(\varphi_2)$;

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

$$8. \text{bound}((\exists x.\varphi)) = \text{bound}(\varphi) \cup \{x\}.$$

Exercițiul 88. Calculați $\text{bound}(\varphi)$ pentru fiecare formulă φ din Exercițiul 80.

Exercițiul 89. Calculați $\text{bound}(\varphi)$, unde

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

Definiția 90. Variabilele legate ale unei formule φ sunt elementele mulțimii $\text{bound}(\varphi)$.

Definiția 91. Variabilele libere ale unei formule φ sunt elementele mulțimii $\text{free}(\varphi)$.

Observație. Mulțimile $\text{free}(\varphi)$ și $\text{bound}(\varphi)$ 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 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} .

5.4 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 cuantorii \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 cuantor 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:

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

Capitolul 6

Deducția naturală

În capitolele anterioare am văzut că pentru a stabili, de exemplu, că două formule sunt echivalente, este necesar un raționament la nivel semantic (adică raționament care folosește noțiunile semantice de valoare de adevăr, structură, etc.).

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.*

6.1 Substituții

Definiția 92. *O substituție este o funcție $\sigma : \mathcal{X} \rightarrow \mathcal{T}$, cu proprietatea că $\sigma(x) \neq x$ pentru un număr finit de variabile $x \in \mathcal{X}$.*

Definiția 93. *Dacă $\sigma : \mathcal{X} \rightarrow \mathcal{T}$ este o substituție, atunci mulțimea $\text{dom}(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$ se numește domeniul substituției σ .*

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

Definiția 94. Dacă $\sigma : \mathcal{X} \rightarrow \mathcal{T}$ este o substituție, atunci extensia unică a substituției σ la mulțimea termenilor este funcția $\sigma^\# : \mathcal{T} \rightarrow \mathcal{T}$, definită recursiv astfel:

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

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

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

Practic, pentru a obține $\sigma^\#(t)$ din t , toate aparițiile unei variabile x din t sunt înlocuite simultan cu termenul corespunzător $\sigma(x)$.

Exemplul 95. Fie substituția $\sigma_1 : \mathcal{X} \rightarrow \mathcal{T}$ definită astfel:

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

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

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

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

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

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

Atenție, nu este vorba de o mulțime, ci doar de o **notăție** pentru substituții.

Exemplul 96. Pentru substituția din Exemplul 95, avem

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

Definiția 97. Dacă $\sigma : \mathcal{X} \rightarrow \mathcal{T}$ 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 $\sigma|_V : \mathcal{X} \rightarrow \mathcal{T}$, definită astfel:

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

Exemplul 98. Pentru substituția din Exemplul 95, avem $\sigma_1|_{\{\mathbf{x}_1\}} = \{\mathbf{x}_1 \mapsto \mathbf{x}_2\}$ și $\sigma_1|_{\{\mathbf{x}_2\}} = \{\mathbf{x}_2 \mapsto \mathbf{f}(\mathbf{x}_3, \mathbf{x}_4)\}$.

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

Definiția 99. Pentru orice substituție $\sigma : \mathcal{X} \rightarrow \mathcal{T}$, extensia lui σ la mulțimea formulelor este funcția $\sigma^b : \mathbb{LPI} \rightarrow \mathbb{LPI}$, definită astfel:

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

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

Exemplul 100. *Utilizând substituția din Exemplul 95, avem că:*

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

Observație. *Atenție: aparițiile legate ale variabilelor nu sunt înlocuite prin aplicarea substituției! În Exemplul 100, apariția lui x_2 în $(\forall x_2. P(x_1, x_2))$ este legată.*

Notăție. *Conform Notăției 6.1 pentru substituțiile care au domeniul finit mai utilizăm notația $\{x_1 \mapsto \sigma(x_1), \dots, x_n \mapsto \sigma(x_n)\}$. 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 $\varphi[x \mapsto t]$.*

6.2 Secvențe

Definiția 101 (Secvență). *O secvență este o pereche de forma:*

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

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

Câteodată citim notația $\{\varphi_1, \dots, \varphi_n\} \vdash \varphi$ sub forma φ este consecință sintactică din $\{\varphi_1, \dots, \varphi_n\}$. De multe ori, vom nota cu $\Gamma = \{\varphi_1, \dots, \varphi_n\}$ 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 $\varphi_1, \dots, \varphi_n \vdash \varphi$, în loc de $\{\varphi_1, \dots, \varphi_n\} \vdash \varphi$. 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 $\varphi_1, \dots, \varphi_n, \psi \vdash \varphi$ în loc de $\{\varphi_1, \dots, \varphi_n\} \cup \{\psi\} \vdash \varphi$, ceea ce ușurează citirea secvențelor.

Exemplul 102. În multe dintre exemplele din acest material vom lucra cu o semnătură $\Sigma = (\{P, Q\}, \{a, b, f, g\})$, unde simbolurile predicative P și Q au aritate 1, simbolurile funcționale f și g au aritate 1, iar simbolurile a și b sunt constante (de aritate 0).

Exemplul 103. Fie semnatura Σ din Exemplul 102. Iată câteva exemple de secvențe:

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

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

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

Exemplul 104. Secvențele din Exemplul 103 pot fi scrise fără acolade astfel:

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

6.3 Reguli de inferență

Definiția 105. O regulă de inferență este un tuplu format din:

1. o mulțime de secvențe S_1, \dots, 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:

$$\text{NUME} \frac{S_1 \quad \dots \quad S_n}{S} \text{ condiție.}$$

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

Exemplul 106. Iată câteva exemple de reguli de inferență pe care le-am întâlnit și la logica propozițională:

$$\wedge_i \frac{\Gamma \vdash \varphi_1 \quad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}, \quad \wedge_{e1} \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1}, \quad \wedge_{e2} \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2}.$$

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:

$$\text{IPOTEZĂ} \frac{}{\Gamma \vdash \varphi_1} \varphi_1 \in \Gamma.$$

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:

$$\wedge_i \frac{\{\mathbf{P(a)}, \mathbf{Q(a)}\} \vdash \mathbf{P(a)} \quad \{\mathbf{P(a)}, \mathbf{Q(a)}\} \vdash \mathbf{Q(a)}}{\{\mathbf{P(a)}, \mathbf{Q(a)}\} \vdash (\mathbf{P(a)} \wedge \mathbf{Q(a)})};$$

$$\wedge_i \frac{\{\mathbf{P(a)}, \mathbf{Q(a)}, \mathbf{Q(b)}\} \vdash (\mathbf{P(a)} \wedge \mathbf{Q(a)}) \quad \{\mathbf{P(a)}, \mathbf{Q(a)}, \mathbf{Q(b)}\} \vdash \mathbf{P(a)}}{\{\mathbf{P(a)}, \mathbf{Q(a)}, \mathbf{Q(b)}\} \vdash ((\mathbf{P(a)} \wedge \mathbf{Q(a)}) \wedge \mathbf{P(a)})}.$$

În prima instanță, am înlocuit variabila matematică Γ cu mulțimea de formule $\{\mathbf{P(a)}, \mathbf{Q(a)}\}$, variabila matematică φ cu formula $\mathbf{P(a)}$ și variabila matematică φ' cu formula $\mathbf{Q(a)}$. 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):

$$? \frac{\{\mathbf{P(a)}, \mathbf{Q(a)}\} \vdash \mathbf{P(a)} \quad \{\mathbf{P(a)}, \mathbf{Q(a)}\} \vdash \mathbf{Q(a)}}{\{\mathbf{P(a)}, \mathbf{Q(a)}\} \vdash (\mathbf{P(a)} \wedge \mathbf{Q(a)})};$$

6.4 Sistem deductiv

Definiția 107. *Un sistem deductiv este o mulțime de reguli de inferență.*

Exemplul 108. *Fie sistemul deductiv D_1 , format din următoarele patru reguli de inferență:*

$$\begin{array}{l} \text{IPOTEZĂ} \frac{}{\Gamma \vdash \varphi, \varphi \in \Gamma} \quad \wedge i \frac{\Gamma \vdash \varphi_1 \quad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2),} \quad \wedge e_1 \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1,} \\ \wedge e_2 \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2.} \end{array}$$

6.5 Demonstrație formală

Definiția 109 (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, \dots, 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, \dots, 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 110. *Iată un exemplu de demonstrație formală în sistemul D_1 introdus mai sus:*

1. $\{P(a), Q(a)\} \vdash P(a);$ (IPOTEZĂ)
2. $\{P(a), Q(a)\} \vdash Q(a);$ (IPOTEZĂ)
3. $\{P(a), Q(a)\} \vdash (P(a) \wedge Q(a));$ ($\wedge i$, 1, 2)
4. $\{P(a), Q(a)\} \vdash (Q(a) \wedge (P(a) \wedge Q(a))).$ ($\wedge i$, 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 111 (Secvență validă). *O secvență $\Gamma \vdash \varphi$ este validă într-un sistem deductiv D dacă există o demonstrație formală S_1, \dots, S_n în D astfel încât $S_n = \Gamma \vdash \varphi$.*

Exemplul 112. *Secvența $\{P(a), Q(a)\} \vdash (P(a) \wedge Q(a))$ este validă în sistemul deductiv D_1 de mai sus, deoarece este ultima secvență din următoarea demonstrație formală:*

$$1. \{P(a), Q(a)\} \vdash P(a); \quad (\text{IPOTEZĂ})$$

$$2. \{P(a), Q(a)\} \vdash Q(a); \quad (\text{IPOTEZĂ})$$

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

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

6.6 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.

6.6.1 Regulele pentru conjuncții

Am văzut deja regulile de introducere și de eliminare pentru conectorul “și”:

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

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

6.6.2 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.

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

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

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

Exercițiul 113. *Arătați că sunt valide următoarele secvențe:*

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

Regula de introducere a implicației este mai subtilă. Pentru a arăta că o implicație $(\varphi_1 \rightarrow \varphi_2)$ 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 notatie referitoare la acoladele din jurul premiselor unei secvențe, în timp ce în a doua regulă acoladele care marchează mulțimea apar explicit:

$$\rightarrow i \frac{\Gamma, \varphi_1 \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \rightarrow \varphi_2)}, \quad \rightarrow i \frac{\Gamma \cup \{\varphi_1\} \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \rightarrow \varphi_2)}.$$

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 $(\varphi_1 \rightarrow \varphi_2)$ 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 $(\varphi_1 \rightarrow \varphi_2)$, presupunem antecedentul φ_1 și arătăm consecventul φ_2 .

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

1. $\{P(a)\} \vdash P(a);$ (IPOTEZĂ)
2. $\{\} \vdash (P(a) \rightarrow P(a)).$ ($\rightarrow i$, 1)

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

1. $\{(P(a) \rightarrow Q(a)), (Q(a) \rightarrow P(b)), P(a)\} \vdash (P(a) \rightarrow Q(a));$ (IPOTEZĂ)
2. $\{(P(a) \rightarrow Q(a)), (Q(a) \rightarrow P(b)), P(a)\} \vdash P(a);$ (IPOTEZĂ)
3. $\{(P(a) \rightarrow Q(a)), (Q(a) \rightarrow P(b)), P(a)\} \vdash Q(a);$ ($\rightarrow e$, 1, 2)
4. $\{(P(a) \rightarrow Q(a)), (Q(a) \rightarrow P(b)), P(a)\} \vdash (Q(a) \rightarrow P(b));$ (IPOTEZĂ)

$$5. \{(P(a) \rightarrow Q(a)), (Q(a) \rightarrow P(b)), P(a)\} \vdash P(b); \quad (\rightarrow e, 4, 3)$$

$$6. \{(P(a) \rightarrow Q(a)), (Q(a) \rightarrow P(b))\} \vdash (P(a) \rightarrow P(b)). \quad (\rightarrow i, 5)$$

Exercițiul 116. Arătați că următoarele secvențe sunt valide:

$$1. \{((P(a) \wedge Q(a)) \rightarrow P(b)), P(a), Q(a)\} \vdash P(b);$$

$$2. \{((P(a) \wedge Q(a)) \rightarrow P(b))\} \vdash (P(a) \rightarrow (Q(a) \rightarrow P(b)));$$

$$3. \{(P(a) \rightarrow (Q(a) \rightarrow P(b)))\} \vdash ((P(a) \wedge Q(a)) \rightarrow P(b)).$$

6.6.3 Regulile pentru disjuncții

Conectorul \vee are două reguli de introducere:

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

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

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

$$1. \{(P(a) \wedge Q(a))\} \vdash (P(a) \wedge Q(a)); \quad (\text{IPOTEZĂ})$$

$$2. \{(P(a) \wedge Q(a))\} \vdash P(a); \quad (\wedge e_1, 1)$$

$$3. \{(P(a) \wedge Q(a))\} \vdash (P(a) \vee Q(a)). \quad (\vee i_1, 2)$$

O altă demonstrație formală pentru aceeași secvență este:

$$1. \{(P(a) \wedge Q(a))\} \vdash (P(a) \wedge Q(a)); \quad (\text{IPOTEZĂ})$$

$$2. \{(P(a) \wedge Q(a))\} \vdash Q(a); \quad (\wedge e_2, 1)$$

$$3. \{(P(a) \wedge Q(a))\} \vdash (P(a) \vee Q(a)). \quad (\vee i_2, 2)$$

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

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

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

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

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 119. *Găsiți o demonstrație formală pentru secvența*

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

6.6.4 Regulile pentru negații

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

$$\neg i \frac{\Gamma, \varphi \vdash \perp}{\Gamma \vdash \neg \varphi} \quad \neg e \frac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg \varphi}{\Gamma \vdash \perp} \quad \perp e \frac{\Gamma \vdash \perp}{\Gamma \vdash \varphi}$$

Să ne readucem aminte că \perp este un conector logic de aritate 0. Cu alte cuvinte, conectorul \perp este de sine stătător o formulă. Semantica formulei \perp este că este falsă în orice atribuire. Cu alte cuvinte, \perp 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 ($\Gamma, \varphi \vdash \perp$). 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, \perp . 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 \perp (sau, regula de eliminare a negației se poate considera ca fiind și regula de introducere a lui \perp).

Exemplul 120. Să arătăm că secvența $\{P(a)\} \vdash \neg\neg P(a)$ este validă:

1. $\{P(a), \neg P(a)\} \vdash P(a)$; (IPOTEZĂ)
2. $\{P(a), \neg P(a)\} \vdash \neg P(a)$; (IPOTEZĂ)
3. $\{P(a), \neg P(a)\} \vdash \perp$; ($\neg e$, 1, 2)
4. $\{P(a)\} \vdash \neg\neg P(a)$. ($\neg i$, 3)

Exemplul 121. Să arătăm că secvența $\{P(a), \neg P(a)\} \vdash P(b)$ este validă:

1. $\{P(a), \neg P(a)\} \vdash P(a)$; (IPOTEZĂ)
2. $\{P(a), \neg P(a)\} \vdash \neg P(a)$; (IPOTEZĂ)
3. $\{P(a), \neg P(a)\} \vdash \perp$; ($\neg e$, 1, 2)
4. $\{P(a), \neg P(a)\} \vdash P(b)$. ($\perp e$, 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 122. Să arătăm că secvența $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a)\} \vdash P(a)$ este validă:

1. $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a), \neg P(a)\} \vdash \neg P(a);$ (IPOTEZĂ)
2. $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a), \neg P(a)\} \vdash (\neg P(a) \rightarrow Q(a));$ (IPOTEZĂ)
3. $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a), \neg P(a)\} \vdash Q(a);$ ($\rightarrow e$, 2, 1)
4. $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a), \neg P(a)\} \vdash \neg Q(a);$ (IPOTEZĂ)
5. $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a), \neg P(a)\} \vdash \perp;$ ($\neg i$, 4, 3)
6. $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a)\} \vdash \neg \neg P(a);$ ($\neg i$, 5)
7. $\{(\neg P(a) \rightarrow Q(a)), \neg Q(a)\} \vdash P(a).$ ($\neg \neg e$, 6)

Exemplul 123. Să arătăm că secvența $\{\} \vdash (P(a) \vee \neg P(a))$ este validă:

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

6.6.5 Eliminarea cuantificatorului universal.

Regula pentru eliminarea cuantificatorului universal este:

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

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

Exemplul 125. *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*

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

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

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

6.6.6 Introducerea cuantificatorului 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:

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

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

1. $\{P(a)\} \vdash P(a)$ (IPOTEZĂ)

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

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

Exemplul 127. Să arătăm că secvența $\{\forall x.(P(x) \rightarrow Q(x)), P(a)\} \vdash \exists x.Q(x)$ este validă:

$$1. \{\forall x.(P(x) \rightarrow Q(x)), P(a)\} \vdash \forall x.(P(x) \rightarrow Q(x)) \quad (\text{IPOTEZĂ})$$

$$2. \{\forall x.(P(x) \rightarrow Q(x)), P(a)\} \vdash P(a) \quad (\text{IPOTEZĂ})$$

$$3. \{\forall x.(P(x) \rightarrow Q(x)), P(a)\} \vdash (P(a) \rightarrow Q(a)) \quad (\forall e, 1, a)$$

$$4. \{\forall x.(P(x) \rightarrow Q(x)), P(a)\} \vdash Q(a) \quad (\rightarrow e, 3, 2)$$

$$5. \{\forall x.(P(x) \rightarrow Q(x)), P(a)\} \vdash \exists x.Q(x) \quad (\exists i, 4)$$

6.6.7 Introducerea cuantificatorului universal.

Regula de introducere a cuantificatorului universal este:

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

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 128. Să arătăm că secvența $\{\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x)\} \vdash \forall x.Q(x)$ este validă:

$$1. \{\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x)\} \vdash \forall x.(P(x) \rightarrow Q(x)) \quad (\text{IPOTEZĂ})$$

$$2. \{\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x)\} \vdash \forall x.P(x) \quad (\text{IPOTEZĂ})$$

$$3. \{\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x)\} \vdash (P(x_0) \rightarrow Q(x_0)) \quad (\forall e, 1, x_0)$$

$$4. \{\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x)\} \vdash P(x_0) \quad (\forall e, 2, x_0)$$

$$5. \{\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x)\} \vdash Q(x_0) \quad (\rightarrow e, 3, 4)$$

$$6. \{\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x)\} \vdash \forall x.Q(x) \quad (\forall i, 5)$$

Observați că pentru secvențele 3, 4 și 5 utilizăm variabila nouă x_0 . La nivel intuitiv, $Q(x_0)$ va avea loc pentru orice x_0 .

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

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

6.6.8 Eliminarea cuantificatorului existențial

Regula pentru eliminarea cuantificatorului existențial este următoarea:

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

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

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

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 $P(x_0)(= P(x)[x \mapsto x_0])$.

6.6.9 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:

$$\text{EXTINDERE} \frac{\Gamma \vdash \varphi}{\Gamma, \varphi' \vdash \varphi}$$

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 131. *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. $\{P(a), \neg P(a)\} \vdash \perp;$ ($\neg e$, 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)

6.7 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:

$$\begin{array}{c}
 \wedge i \frac{\Gamma \vdash \varphi_1 \quad \Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}, \quad \wedge e_1 \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_1}, \\
 \wedge e_2 \frac{\Gamma \vdash (\varphi_1 \wedge \varphi_2)}{\Gamma \vdash \varphi_2}, \quad \rightarrow e \frac{\Gamma \vdash (\varphi_1 \rightarrow \varphi_2) \quad \Gamma \vdash \varphi_1}{\Gamma \vdash \varphi_2}, \\
 \rightarrow i \frac{\Gamma, \varphi_1 \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \rightarrow \varphi_2)}, \quad \vee i_1 \frac{\Gamma \vdash \varphi_1}{\Gamma \vdash (\varphi_1 \vee \varphi_2)}, \quad \vee i_2 \frac{\Gamma \vdash \varphi_2}{\Gamma \vdash (\varphi_1 \vee \varphi_2)}, \\
 \vee e \frac{\Gamma \vdash (\varphi_1 \vee \varphi_2) \quad \Gamma, \varphi_1 \vdash \varphi' \quad \Gamma, \varphi_2 \vdash \varphi'}{\Gamma \vdash \varphi'}, \\
 \neg e \frac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg \varphi}{\Gamma \vdash \perp}, \quad \neg i \frac{\Gamma, \varphi \vdash \perp}{\Gamma \vdash \neg \varphi}, \quad \perp e \frac{\Gamma \vdash \perp}{\Gamma \vdash \varphi}, \\
 \text{IPOTEZĂ} \frac{}{\Gamma \vdash \varphi} \varphi \in \Gamma, \quad \text{EXTINDERE} \frac{\Gamma \vdash \varphi}{\Gamma, \varphi' \vdash \varphi}, \quad \neg \neg e \frac{\Gamma \vdash \neg \neg \varphi}{\Gamma \vdash \varphi}, \\
 \forall e \frac{\Gamma \vdash (\forall x. \varphi)}{\Gamma \vdash \varphi[x \mapsto t]} \text{ vars}(t) \cap \text{bound}(\varphi) = \emptyset \\
 \exists i \frac{\Gamma \vdash \varphi[x \mapsto t]}{\Gamma \vdash (\exists x. \varphi)} \text{ vars}(t) \cap \text{bound}(\varphi) = \emptyset \\
 \forall i \frac{\Gamma \vdash \varphi[x \mapsto x_0]}{\Gamma \vdash (\forall x. \varphi)} x_0 \notin \text{vars}(\Gamma, \varphi) \\
 \exists e \frac{\Gamma \vdash (\exists x. \varphi) \quad \Gamma \cup \{\varphi[x \mapsto x_0]\} \vdash \psi}{\Gamma \vdash \psi} x_0 \notin \text{vars}(\Gamma, \varphi, \psi)
 \end{array}$$

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

6.8 Corectitudinea și completitudinea deducției naturale pentru logica de ordinul I

Teorema 132 (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 133. *Demonstrați Teorema 132.*

Teorema 134 (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.*

6.9 Fișă de exerciții

Exercițiul 135. *Arătați că următoarele secvențe sunt valide:*

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

Exercițiul 136. *Stabiliți care dintre secvențele de mai jos sunt valide:*

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