Logică pentru Informatică - Săptămâna 12
Forme normale ale formulelor de ordinul I
Exerciții pentru Seminar

1. Arătați că următoarele echivalențe au loc:

   (a) $P(e, x) \stackrel{S}{\equiv} P(e, f(x, x))$ unde S este Σ-structura $S = (\mathbb{N}, \{<\}, \{+, s, 0\})$.

   (b) $\neg\forall x.\varphi \equiv \exists x.\neg\varphi$;

   (c) $\neg\exists x.\varphi \equiv \forall x.\neg\varphi$

   (d) $(\forall x.\varphi_1) \wedge \varphi_2 \equiv \forall x.(\varphi_1 \wedge \varphi_2)$, dacă $x \notin \mathit{free}(\varphi_2)$;

   (e) $(\forall x.\varphi_1) \vee \varphi_2 \equiv \forall x.(\varphi_1 \vee \varphi_2)$, dacă $x \notin \mathit{free}(\varphi_2)$;

   (f) $(\exists x.\varphi_1) \wedge \varphi_2 \equiv \exists x.(\varphi_1 \wedge \varphi_2)$, dacă $x \notin \mathit{free}(\varphi_2)$;

   (g) $(\exists x.\varphi_1) \vee \varphi_2 \equiv \exists x.(\varphi_1 \vee \varphi_2)$, dacă $x \notin \mathit{free}(\varphi_2)$;

   (h) $\forall x.(P(x) \wedge Q(x)) \equiv (\forall x.P(x)) \wedge (\forall x.Q(x))$

   (i) $\forall x.(P(x) \vee Q(x)) \not\equiv (\forall x.P(x)) \vee (\forall x.Q(x))$

   (j) $\exists x.(P(x) \wedge Q(x)) \not\equiv (\exists x.P(x)) \wedge (\exists x.Q(x))$

   (k) $\exists x.(P(x) \vee Q(x)) \equiv (\exists x.P(x)) \vee (\exists x.Q(x))$

2. Fie substituţia $\sigma : \mathcal{X} \to \mathcal{T}$ astfel încât $\sigma(x) = i(y)$, $\sigma(y) = f(x, z)$ şi $\sigma(z) = z$ pentru $z \in \mathcal{X} \setminus \{x, y\}$. Aplicaţi substituţia σ pentru urmatoarele formule:

   (a) $\varphi = (\forall x.P(x, y)) \to P(i(y), x)$

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

3. Calculați câte o FNP pentru fiecare dintre formulele:

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

Soluţie:

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

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

(c) $(\exists z.P(x,y)) \vee P(z,z)$;

(d) $(\exists z.P(x,z)) \wedge (\forall x.P(x,z))$;

(e) $(\exists z.P(x,z)) \rightarrow P(x,x)$.

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

(g) $\forall x.\exists y.P(x,y) \rightarrow \neg\exists x.\neg\exists y.P(x,y)$

4. Vom folosi demonstratorul Z3 disponibil la adresa `https://rise4fun.com/z3` peste signatura $\Sigma = (\{<,\leq,>,\geq,=\},\{+,*,0,1,2,3,4,\ldots\})$ și Σ-structura $S = (\mathbb{Z},\{<,\leq,>,\geq,=,+,*,0,1,2,3,4,\ldots\})$. Vom determina dacă anumite formule sunt satisfiabile în S, iar dacă sunt satisfiabile să găsim o atribuire care le face adevărate.

   Reamintim că o formulă φ este satisfiabilă într-o structură S dacă există o S-atribuire α cu proprietatea că $S,\alpha \models \varphi$.

   De exemplu, formula:

$$> (x, +(y,2)) \wedge = (x, +(*(2,z),10)) \wedge \leq (+(z,y),100)$$

   sau, folosind notația infixată:

$$x > y + 2 \wedge x = 2 * z + 10 \wedge z + y \leq 100$$

   este satisfiabilă în structura S de mai sus, iar un martor al satisfiabilității este atribuirea $\alpha : \mathcal{X} \to \mathbb{Z}$, definită prin $\alpha(x) = 10, \alpha(y) = 0, \alpha(z) = 0$. Pentru a testa satisfiabilitatea formulei de mai sus, folosim codul:

```
(declare-const x Int) ;; declaram toate variabilele libere
(declare-const y Int)
(declare-const z Int)
```

```
(assert (and (> x (+ y 2)) ;; introducem formula pe care dorim
             (= x (+ (* 2 z) 10)) ;; sa o testam daca e adevarata
             (<= (+ z y) 1000))) ;; folosind sintaxa Z3

(check-sat) ;; verificam daca formula e satisfiabila
```

Deoarece formula este satisfiabilă în S, putem folosi următoarea comandă pentru a obține o atribure în care formula este adevărată.

```
(get-model) ;; afiseaza o atribuire care face formula adevarata
```

Modelați următoarele afirmații ca formule de ordinul I peste signatura Σ definită mai sus și folosiți Z3 pentru a determina dacă acestea sunt sau nu satisfiabile.

(a) x este mai mare decât 100, y este mai mic decât 42, iar produsul $x \times y$ este mai mic decât 10. Variabile libere: x, y.

(b) x este un număr par mai mare decât 11. Variabile libere: x. Hint: putem exprima "x este par" prin $\exists x'.(= (x, *(2, x')))$. În Z3, scriem `(exists ((xp Int)) (= x (* 2 xp)))` corespunzător formulei $\exists x'.(= (x, *(2, x')))$.

(c) x este impar, y este par și $x + y$ este mai mare decât 42. Variabile libere: x, y.

(d) x este impar, y este par și $x + y$ este par. Variabile libere: x, y.

(e) $x * y$ este impar, $x + y$ este par, $x > 10$ și $y < 0$. Variabile libere: x, y.

(f) Suma oricăror două numere pare este număr par. Variabile libere: niciuna.

(g) Folosiți Z3 pentru a explica jocul următor: te gândești la un număr, aduni 4 la el, înmulțești rezultatul cu 2, scazi 6 din rezultat, împarți la 2 și la final, scazi din ce obții numărul la care te-ai gândit; rezultatul este întotdeauna 1, indiferent cu ce număr ai început.

Ghid rapid de sintaxa Z3:

Math	Z3
$x \times y$	(*xy)
$x + y$	(+xy)
$\varphi_1 \wedge \varphi_2$	(and φ_1 φ_2)
$\varphi_1 \rightarrow \varphi_2$	(=> φ_1 φ_2)
$\exists x.\varphi$	(exists ((x Int)) φ)