Синтаксис и семантика на език на предикатното смятане от първи ред

Иво Стратев

9 октомври 2019 г.

Съдържание

Предстои да бъде въведен един формален език. Този език ще има напълно недвусмислен синтаксис и ясна семантика. Една от целите ни с такъв език ще бъде да описваме (говорим) и работим в различни светове. Мета език е езика, който използваме за да опишем формалния (обектния) език.

1 Забележки

Смятам, че използването на различи цветове ще помогне в разбирането и осмислянето на нещата, защото ще послужи за ясно разграничаване между обектен език и мета език. За това ще използва три цвята вместо един.

- В син цвят ще бъдат оцветявани буквите (символитите) от азбуката/ите. Тоест например (ще бъде буквата лява отваряща кръгла скоба.
- В червен цвят ще бъдат оцветявани символи, чието значние ще бъде или символ или редици от букви (символи). Целта е да е по-ясно, че използвайки тези символи за строене на думи ще имаме предвид, че те биват заместени синтактично в строените думи или че значението на даден обект е символ. Така например ако $\varphi = f(x)$, то $s(\varphi, c) = s(f(x), c)$. Ако пък кажем $\alpha \in \{a, b, \dots\}$, то ще разбираме, че α е някой от символите (буквите) $a, b, \dots$.
- В черен цвят ще бъдат всички останали символи, които интерпретираме по стандартния за нас начин. Тоест в черен цвят ще бъде всичко от мета езика, който за нас ще бъде българския език, смесен с езика на математиката, който сме развили до момента (възприели до сега).

Също така смятам, че използването на различни скоби би улеснило разчитането на "дълги"формули, когато не е въведен приоритет на съждителните връзки.

И така сега ще кажем какво значи един език $\mathcal{L}$ да бъде език на предикатното смятане от първи ред.

2 Синтаксис

2.1 Логическа част

Състои се от следните азбуки (множества от символи):

- азбука на съждителните връзки $\{\neg, \vee, \&, \implies, \iff\}$;
- азбука на кванторите $\{\exists, \forall\}$;
- азбука на помощните символи $\{[, <, (, ,,), >,]\}$;

- азбука на (обектовите/индивидните) променливи $Var_{\mathcal{L}}$ (изброимо множество от символи). Обикновенно за нас елементи на това множество ще бъдат буквите $x\ y\ z\ w$ с или без индекси. Например напълно възможно е $Var_{\mathcal{L}} = \{x, y, z, x_\pi, y_{\frac{325}{66}}, z_\epsilon\}$;

- езика $\mathcal{L}$ е възможно да съдържа, но не е задължително да съдържа символ $\doteq$, който ще наричаме символ за формално равенство.

2.2 Нелогическа част

Състои се от следните азбуки (множества от символи, всяко от които е изброимо):

- множество на константните символи $Const_{\mathcal{L}}$ Обикновенно за нас елементи на това множество ще бъдат буквите $a\ b\ c\ d$ с или без индекси. Например напълно възможно е $Const_{\mathcal{L}} = \{a, b, c, d_\pi, a_2, b_\epsilon\}$;

- множество на функционалните символи $Func_{\mathcal{L}}$ Обикновенно за нас елементи на това множество ще бъдат буквите $f\ g\ h$ с или без индекси. Например напълно възможно е $Func_{\mathcal{L}} = \{f, g, h, f_\pi, g_2, h_\epsilon\}$;

- множество на предикатите символи $Pred_{\mathcal{L}}$ Обикновенно за нас елементи на това множество ще бъдат буквите $p\ q\ r\ t$ с или без индекси. Например напълно възможно е $Pred_{\mathcal{L}} = \{p, q, r, t_\pi, p_2, q_\epsilon\}$.

2.3 Арност

Арност ще наричаме функция $\#_{(\_)} : Func_{\mathcal{L}} \cup Pred_{\mathcal{L}} \to \mathbb{N} \setminus \{0\}$. Тази функция за нас неформално ще ни казва броя на аргументите на всеки функционален и предикатен символ. Което като програмисти ще значи, че забраняваме претоварването (overloading) на символите нещо, с което сме свикнали от езика C++ . Това е с цел наше улеснение, не обратното :) .

2.4 Сигнатура

Сигнатура на езика $\mathcal{L}$ ще наричаме множеството

$$Const_{\mathcal{L}} \cup Func_{\mathcal{L}} \cup Pred_{\mathcal{L}}\ (\ \cup \{\doteq\}\ )$$

Може би най-близкото за нас нещо до сигнатура е интерфейс (абстрактен клас) (Type class) от програмирането.

2.5 Пример до тук

Нека $\mathcal{L}$ включва формално равенство и още нека:

- $Const_{\mathcal{L}} = \{a, b\}$;

- $Func_{\mathcal{L}} = \{f, g\}$;

- $Pred_{\mathcal{L}} = \emptyset$.

Света, в който бихме искали да работим (да опишем) е някой конкретен пръстен. Тогава $\#f = \#g = 2$ и сигнатурата в случая е множеството $\{a, b, f, g, \doteq\}$.
Предстои ни да се сблъскаме с понятието терм, което не е нищо друго от име за обект от света, в който желаем да работим/опишем. Какво е терм ще дефинираме индуктивно.

2.6 Терм

База:

- $(\forall \chi \in Var_{\mathcal{L}})[\chi \in Term_{\mathcal{L}}]$ (всяки символ за променлива е терм);
- $(\forall c \in Const_{\mathcal{L}})[c \in Term_{\mathcal{L}}]$ (всяки символ за константа е терм);

Имаме единствено правило чрез което да конструираме нови термове от вече получени такива.

Ако $\delta \in Func_{\mathcal{L}}$ и τ_1 е терм, ..., $\tau_{\#\delta}$ е терм, то $\delta(\tau_1, \ldots, \tau_{\#\delta})$ също е терм. (символичното прилагане на функционален символ към термове е терм).

Термове получени чрез горното правило могат да бъдат получени само след краен брой пъти неговото прилагане.

Множеството на термовете от езика $\mathcal{L}$ ще означаваме с $Term_{\mathcal{L}}$.

2.6.1 Примери

Надграждаме примера, който разглеждахме като искаме $Var_{\mathcal{L}} = \{x, y, z\}$. Тогава следните редици от символи са термове:

- a;
- b;
- x;
- y;
- $f(a, b)$;
- $f(a, g(b, b))$;
- $f(f(g(b, a), b), g(a, a))$.

2.7 Атомарна формула

Ако $\rho \in Pred_{\mathcal{L}}$ и τ_1, ..., $\tau_{\#\rho}$ са термове, то $\rho < \tau_1, \dots, \tau_{\#\rho} >$ е атомарна формула.

Ако $\mathcal{L}$ е с формално равенство и α е терм и β е терм, то $< \alpha \doteq \beta >$ също е атомарна формула.

Сега ще дефинираме напълно формално, какво ще означава за нас формула.

2.8 Формула

База: Всяка атомарна формула е формула.
Можем да конструираме нови формули от други формули чрез краен брой прилагания на следите три правила:

- Ако φ е формула, то $[\neg\varphi]$ е формула;
- Ако φ е формула, ψ е формула и $\sigma \in \{\vee, \&, \implies, \iff\}$, то $[\varphi\sigma\psi]$ е формула;
- Ако φ е формула, $\chi \in Var_{\mathcal{L}}$ и $\Lambda \in \{\exists, \forall\}$, то $[\Lambda\chi\varphi]$ е формула.

Множеството от формули от езика $\mathcal{L}$ ще бележим с $\mathcal{F}_{\mathcal{L}}$.

2.8.1 Примери

Надграждаме още примера за пръстен. Следните думи са формули:

- $< x \doteq x >$;
- $< a \doteq y >$;
- $[\neg < a \doteq y >]$;
- $[< f(a, b) \doteq x > \& < g(x, y) \doteq y >]$;
- $[\forall x < x \doteq x >]$;
- $[\exists x < x \doteq x >]$;
- $[[\forall x < x \doteq x >] \implies [< f(g(a, y), x) \doteq x > \& < g(x, y) \doteq y >]]$;
- $[\forall x[\exists y < f(x, y) \doteq a >]]$.

Така вече можем да кажем, че думите от език на предикатното смятане от първи са формулите от този език.

3 Семантика

Няма как да дефинираме семантика без да кажем какво значи всеки символ от езика (как да го интерпретираме). И така първо ще дефиираме понятието структура.

3.1 Структура

Структура за нас ще бъде наредена двойка (A, I), за която

- A е непразно множество, което ще наримаче универсум, домей или носител на структурата;
- I ще е изображение, което ще наричаме интерпретация;
- Ако $\epsilon \in Const_{\mathcal{L}}$, то $I(\epsilon) \in A$ (интерпретацията на всеки символ за константа е фиксиран елемент на носителя);
- Ако $\pi \in Func_{\mathcal{L}}$, то $I(\pi)\ : A^{\#\pi} \to A$ (интерпретацията на всеки функционален символ е функция между носителя на степен арността на функционалния символ и носителя);
- Ако $\rho \in Pred_{\mathcal{L}}$, то $I(\rho)\ \subseteq A^{\#\rho}$ (интерпретацията на всеки предикатен символ е релация с домейн носителя на степен арността на предикатния символ);

Неформално можем да кажем, че интерпретацията на всеки:

- символ за константа е константа;
- функционален символ е функция;
- предикатен символ е релация.

Обикновенно за структурите ще ползваме голяма ръкописна буква, а за носителя/универсума на структурата голяма печатна буква.

Тоест в случая $\mathcal{A} = (A, I)$.

И ако имаме структура $\mathcal{A}$, то нейния носител ще означаваме с $|\mathcal{A}|$.

Също така ако $\xi \in Const_{\mathcal{L}} \cup Func_{\mathcal{L}} \cup Pred_{\mathcal{L}}$, то вместо $I(\xi)$ ще ползваме означението $\xi^{\mathcal{A}}$.

3.1.1 Пример

Разширяваме още примера, който разглеждаме.
Една структура за еика, който разглеждаме е следната:

- $\mathcal{A} = (\mathbb{Z}, I)$;

- $I(a) = a^{\mathcal{A}} = 0_{\mathbb{Z}}$;
- $I(b) = b^{\mathcal{A}} = 1_{\mathbb{Z}}$;
- $I(f) = f^{\mathcal{A}} = +_{\mathbb{Z}}$;
- $I(g) = g^{\mathcal{A}} = \cdot_{\mathbb{Z}}$.

За да дефинираме семантиката на $Lang$ ще ни трябва да кажем какво значи да оценим една променлива. Идея, с която до болка сме привикнали когато пишем програми. Надявам се всеки е запознат с начина за оценяване в езика Scheme (Racket) - модела на средите. Ще се опитаме да формализираме този модел.

3.2 Оценка

Оценка в универсума на една структура $\mathcal{A}$ на променливите от езика $\mathcal{L}$ ще наричаме функция $\nu : Var_{\mathcal{L}} \rightarrow |\mathcal{A}|$.

Нека разширим оценката, до функция над термове.

3.3 Стойност на терм

Нека рекурсивно дефинираме разширената функция
$\$\ (\_)\ \$^{\mathcal{A}}_{\nu} : Term_{\mathcal{L}} \rightarrow |\mathcal{A}|$ по следния начин:

- $(\forall \chi \in Var_{\mathcal{L}})[\ \$\ \chi\ \$^{\mathcal{A}}_{\nu} \stackrel{def}{=} \nu(\epsilon)\ ]$;
- $(\forall \epsilon \in Const_{\mathcal{L}})[\ \$\ \epsilon\ \$^{\mathcal{A}}_{\nu} \stackrel{def}{=} \epsilon^{\mathcal{A}}\ ]$;
- Ако $\pi \in Func_{\mathcal{L}}, \tau_1 \in Term_{\mathcal{L}}, \ldots, \tau_{\#\pi} \in Term_{\mathcal{L}}$, то
$\$\ \pi(\tau_1, \ldots, \tau_{\#\pi})\ \$^{\mathcal{A}}_{\nu} \stackrel{def}{=} \pi^{\mathcal{A}}(\$\ \tau_1\ \$^{\mathcal{A}}_{\nu}, \ldots, \$\ \tau_{\#\pi}\ \$^{\mathcal{A}}_{\nu})$.

3.3.1 Пример

Нека пресметнем стойостта на следния терм

$$g(f(b, x), y)$$

в структурата $\mathcal{A}$ от предния пример, при оценка $\nu : \{x, y, z\} \rightarrow \mathbb{Z}$, за която $\nu(x) = 3,\ \nu(y) = -5,\ \nu(z) = 7$.

$$
\begin{aligned}
\$\ g(f(b,x),y)\ \$_\nu^{\mathcal{A}} &= \\
g^{\mathcal{A}}(\$\ f(b,x)\ \$_\nu^{\mathcal{A}}, \$\ y\ \$_\nu^{\mathcal{A}}) &= \\
\cdot_{\mathbb{Z}}(f^{\mathcal{A}}(\$\ b\ \$_\nu^{\mathcal{A}}, \$\ x\ \$_\nu^{\mathcal{A}}), \nu(y)) &= \\
\cdot_{\mathbb{Z}}(+_{\mathbb{Z}}(b^{\mathcal{A}}, \nu(x)), -5) &= \\
\cdot_{\mathbb{Z}}(+_{\mathbb{Z}}(1,3), -5) &= \\
\cdot_{\mathbb{Z}}(4, -5) &= \\
-20 &
\end{aligned}
$$

3.4 Модифицирана оценка

За да разберем, защо ще се нуждаем от следващата дефиниция нека разгледаме следния израз на езика Scheme

$$((lambda\ (x)\ (+\ x\ ((lambda\ (x)\ (*\ x\ 3))\ 2)))\ 5)$$

Горният израз по стойност е еквивалентен със следния математически

$$(x + (3x)(2))(5)$$

И сега може би е по-ясно, че при пресмятането на израза $3x$, е в сила $x = 2$, а във външното пресмятане е в сила $x = 5$.

Ако сега разгледаме формулата

$$[[\exists x < x \doteq b >] \ \&\ [\exists x < x \doteq a >]]$$

Без да сме въвели какво значи семантика на формула нека кажем неформално, че искаме семантиката на една формула да е нейната вярност в света, в който я разглеждаме.

И така ако искаме разглежданата формула да е вярна в пръстена, който образуват целите числа, тоест в света, който разглеждаме в примерите. То бихме искали да е истина следното изречение на български: Същесвува цяло число x, равно на цялото число 1 и същесвува цяло число x, равно на цялото число 0. Ясно е че това е вярно и в първия случай x е 1 във втория 0.

Ясно е, че се нуждаем от начин да "свързваме"стойност с променлива при разглеждането на формула или подформула. Тоест трябва ни начин, по който да моделираме промяна на оценката на дадена формула.

Въвеждаме понятието модифицирана оценка. Ако имаме една оценка ν и променлива χ и искаме нейната оценка да е a, който е фиксиран елемент на универсума на структурата, която разглеждаме. То $\nu^{\chi}{}_{a}$ е модифицираната оценка на ν, която оценява χ с a. Тоест

$$\nu^{\chi}{}_{a}(\gamma) = \begin{cases} a, & \gamma = \chi \\ \nu(\gamma), & \gamma \neq \chi \end{cases}$$

3.4.1 Пример

Ако модефицираме стойността на променливата y да бъде 5 в предния пример, ще получим

$$\$ \ g(f(b,x),y) \ \$^{\mathcal{A}}_{\nu^{y}{}_{5}} = 20$$

3.5 Една последна дефиниця преди въвеждането на Семантика

Нека означаваме константите "истина"с И и "лъжа"с Л.
Нека $h_{\neg}, h_{\vee}, h_{\&}, h_{\implies}, h_{\iff}$ са стандартните булеви функции: негация, дизюнкция, конюнкция, импликация и биимпликация.
И нека $Eval \ : \ \{\neg, \vee, \&, \implies, \iff\} \to \{h_{\neg}, h_{\vee}, h_{\&}, h_{\implies}, h_{\iff}\}$, която на символ съпоставя съответната булева функция. Тоест

$$\begin{aligned} Eval(\neg) &= h_{\neg} \\ Eval(\vee) &= h_{\vee} \\ Eval(\&) &= h_{\&} \\ Eval(\implies) &= h_{\implies} \\ Eval(\iff) &= h_{\iff} \end{aligned}$$

3.6 Семантика на формула в структура $\mathcal{A}$ за езика (отговаряща на сигнатурата на езика) при оценка ν

Дефинираме функция $\| \ (\_) \ \|^{\mathcal{A}}_{\nu} \ : \ \mathcal{F}_{\mathcal{L}} \to \{\text{И}, \text{Л}\}$ по следните правила:

- Ако $\rho \in Pred_{\mathcal{L}}, \tau_1 \in Term_{\mathcal{L}}, \ldots, \tau_{\#\rho} \in Term_{\mathcal{L}}$, то
$\| \ \rho < \tau_1, \ldots, \tau_{\#\rho} > \ \|^{\mathcal{A}}_{\nu} = \text{И} \iff (\$ \ \tau_1 \ \$^{\mathcal{A}}_{\nu}, \ldots, \$ \ \tau_{\#\rho} \ \$^{\mathcal{A}}_{\nu}) \in \rho^{\mathcal{A}}$.

- Ако $\mathcal{L}$ е с формално равенство и $\tau \in Term_{\mathcal{L}}$ и $\gamma \in Term_{\mathcal{L}}$, то
$\| \ < \tau \doteq \gamma > \ \|^{\mathcal{A}}_{\nu} = \text{И} \iff \$ \ \tau \ \$^{\mathcal{A}}_{\nu} = \$ \ \gamma \ \$^{\mathcal{A}}_{\nu}$.

- Ако $\varphi \in \mathcal{F}_{\mathcal{L}}$, то $\| \ [\neg\varphi] \ \|^{\mathcal{A}}_{\nu} = h_{\neg}(\| \ \varphi \ \|^{\mathcal{A}}_{\nu})$.

- Ако $\varphi \in \mathcal{F}_{\mathcal{L}}, \sigma \in \{\vee, \&, \implies, \iff\}, \psi \in \mathcal{F}_{\mathcal{L}}$, то
$\| \ [\varphi\sigma\psi] \ \|^{\mathcal{A}}_{\nu} = Eval(\sigma)(\| \ \varphi \ \|^{\mathcal{A}}_{\nu}, \| \ \psi \ \|^{\mathcal{A}}_{\nu})$.

- Ако $\varphi \in \mathcal{F}_{\mathcal{L}}, \chi \in Var_{\mathcal{L}}$, то
$\| [\exists\chi\varphi] \|_{\nu}^{\mathcal{A}} = \text{И} \iff$ съществува $a \in |\mathcal{A}|$ и $\| \varphi \|_{\nu^{\chi}{}_{a}}^{\mathcal{A}} = \text{И}$.

- Ако $\varphi \in \mathcal{F}_{\mathcal{L}}, \chi \in Var_{\mathcal{L}}$, то
$\| [\forall\chi\varphi] \|_{\nu}^{\mathcal{A}} = \text{И} \iff$ за всеки елемент $a \in |\mathcal{A}|$ е в сила $\| \varphi \|_{\nu^{\chi}{}_{a}}^{\mathcal{A}} = \text{И}$.

3.7 Модел

Казваме, че структура $\mathcal{A}$ и оценка ν са модел за формула φ тогава и само тогава когато $\| \varphi \|_{\nu}^{\mathcal{A}} = \text{И}$. Ако $\mathcal{A}$ и оценка ν са модел за φ, то ще го означаваме с $\mathcal{A} \underset{\nu}{\models} \varphi$.

3.8 Примери

Нека проверим дали структурата и оценката от примера, който разглеждаме са модели за:

- $< x \doteq y >$;
- $[\exists z < f(x,z) \doteq a >]$;
- $[\exists x\ [\forall y\ [\exists z < f(y,z) \doteq x >]]]$

3.8.1 Формула 1

$\| < x \doteq y > \|_{\nu}^{\mathcal{A}} = \text{И} \iff \$\ x\ \$_{\nu}^{\mathcal{A}} = \$\ y\ \$_{\nu}^{\mathcal{A}} \iff 3 = -5 \iff \text{Л}$, Следователно $\mathcal{A}$ и ν не са модел.

3.8.2 Формула 2

$$\begin{aligned}
\| [\exists z < f(x,z) \doteq a >] \|_{\nu}^{\mathcal{A}} = \text{И} &\iff \\
\text{съцесвува } \alpha \in \mathbb{Z} \text{ и } \| < f(x,z) \doteq a > \|_{\nu^{z}{}_{\alpha}}^{\mathcal{A}} = \text{И} &\iff \\
\text{съцесвува } \alpha \in \mathbb{Z} \text{ и } \$\ f(x,z)\ \$_{\nu^{z}{}_{\alpha}}^{\mathcal{A}} = \$\ a\ \$_{\nu^{z}{}_{\alpha}}^{\mathcal{A}} &\iff \\
\text{съцесвува } \alpha \in \mathbb{Z} \text{ и } 3 + \alpha = 0 &\iff \\
3 + (-3) = 0 &\iff \text{И}
\end{aligned}$$

Тоест избирайки $\alpha = -3$ получаваме истина, което означава че съцесвува елемент α с желаното свойство. Следователно $\mathcal{A} \underset{\nu}{\models} [\exists z < f(x,z) \doteq a >]$

3.8.3 Формула 3

Проверете сами, че $\mathcal{A} \underset{\nu}{\models} [\exists x\ [\forall y\ [\exists z < f(y,z) \doteq x >]]]$. =)