

Metalogika

Jerzy Pogonowski

Zakład Logiki i Kognitywistyki UAM
www.kognitywistyka.amu.edu.pl
<http://logic.amu.edu.pl/index.php/Dydaktyka>
pogon@amu.edu.pl

Arytmetyzacja składni

Plan wykładu

Przygotowujemy się do przedstawienia kilku ważnych twierdzeń metalogicznych (Gödla, Rossera, Tarskiego, Löba).

- Przypomnimy *aksjomatyczną teorię arytmetyki*: PA (Arytmetykę Peana pierwszego rzędu).
- Powiemy, co to znaczy, że zbiory, relacje i funkcje rekurencyjne są *reprezentowalne* w PA.
- Pokażemy także, jak dokonuje się *arytmetyzacji składni*, czyli jak koduje się w PA pojęcia składniowe.
- Podamy definicję *hierarchii arytmetycznej*.

Język PA

Alfabet symboli:

- stałe logiczne: $\neg, \wedge, \vee, \rightarrow, \equiv, \forall, \exists$,
- zmienne indywidualne: x_1, x_2, \dots ,
- predykat (2-arg.) identyczności: \doteq ,
- stała indywidualna (zero): $\underline{0}$,
- symbol funkcyjny (1-arg.) następnika: \underline{s} ,
- symbole funkcyjne (2-arg.) dodawania $\underline{+}$ oraz mnożenia $\underline{\times}$,
- symbole pomocnicze: nawias lewy (oraz nawias prawy).

Wyrażeniem języka PA jest dowolny skończony ciąg symboli alfabetu języka PA.

Uwaga. Skorzystamy z możliwości zastąpienia notacji infiksowej przez prefiksową.

Termy

Zbiór termów jest najmniejszym zbiorem wyrażeń języka PA takim, że:

- stała $\underline{0}$ jest termem,
- każda zmienna indywidualna x_1, x_2, \dots jest termem,
- jeśli t jest termem, to $\underline{s}(t)$ jest termem,
- jeśli t_1 i t_2 są termami, to $(t_1 \underline{+} t_2)$ oraz $(t_1 \underline{\times} t_2)$ są termami.

Ze względów natury psychologicznej piszemy zwykle:

- $(t_1 \underline{+} t_2)$ zamiast $\underline{+}(t_1, t_2)$
- $(t_1 \underline{\times} t_2)$ zamiast $\underline{\times}(t_1, t_2)$.

Formuły

Zbiór formuł jest najmniejszym zbiorem wyrażeń języka PA takim, że:

- jeśli t_1, t_2 są termami, to $t_1 \doteq t_2$ jest formułą,
- jeśli ψ jest formułą, a x_i zmienną indywidualną, to formułami są też $\neg\psi, \forall x_i\psi, \exists x_i\psi$,
- jeśli φ, ψ są formułami, to formułami są także: $(\varphi \wedge \psi), (\varphi \vee \psi), (\varphi \rightarrow \psi), (\varphi \equiv \psi)$.

Stosujemy konwencje opuszczania nawiasów, znane z elementarnej logiki. Dla wygody, opuszczamy też czasem indeks przy zmiennych indywidualnych.

Aksjomaty logiczne

Aksjomatami logicznymi są wszystkie podstawienia formuł języka PA za zmienne zdaniowe w następujących formułach języka KRZ:

- | | |
|--|--|
| 1) $p \rightarrow (q \rightarrow p)$ | 8) $(p \rightarrow q) \rightarrow ((p \rightarrow r) \rightarrow p \rightarrow (q \wedge r))$ |
| 2) $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$ | 9) $p \rightarrow (p \vee q)$ |
| 3) $(p \rightarrow q) \rightarrow (\neg q \rightarrow \neg p)$ | 10) $q \rightarrow (p \vee q)$ |
| 4) $\neg\neg p \rightarrow p$ | 11) $(p \rightarrow r) \rightarrow ((q \rightarrow r) \rightarrow ((p \vee q) \rightarrow r))$ |
| 5) $p \rightarrow \neg\neg p$ | 12) $(p \equiv q) \rightarrow (p \rightarrow q)$ |
| 6) $(p \wedge q) \rightarrow p$ | 13) $(p \equiv q) \rightarrow (q \rightarrow p)$ |
| 7) $(p \wedge q) \rightarrow q$ | 14) $(p \rightarrow q) \rightarrow ((q \rightarrow p) \rightarrow (p \equiv q))$ |

Aksjomaty identyczności

Aksjomatami identyczności dla PA są:

- $x \doteq x$
- $x \doteq y \rightarrow y \doteq x$
- $(x \doteq y \wedge y \doteq z) \rightarrow x \doteq z$
- $x \doteq y \rightarrow \underline{s}(x) \doteq \underline{s}(y)$
- $x \doteq y \rightarrow x \underline{+} z \doteq y \underline{+} z$
- $x \doteq y \rightarrow z \underline{+} x \doteq z \underline{+} y$
- $x \doteq y \rightarrow x \underline{\times} z \doteq y \underline{\times} z$
- $x \doteq y \rightarrow z \underline{\times} x \doteq z \underline{\times} y$

Aksjomaty pozalogiczne

Aksjomatami specyficznymi PA są:

- (A1) $\underline{s}(x) \doteq \underline{s}(y) \rightarrow x = y$
- (A2) $\neg \underline{0} \doteq \underline{s}(x)$
- (A3) $x \underline{+} \underline{0} \doteq x$
- (A4) $x \underline{+} \underline{s}(y) \doteq \underline{s}(x \underline{+} y)$
- (A5) $x \underline{\times} \underline{0} \doteq \underline{0}$
- (A6) $x \underline{\times} \underline{s}(y) \doteq (x \underline{\times} y) \underline{+} x$
- (A7) $(\varphi(\underline{0}) \wedge \forall x(\varphi(x) \rightarrow \varphi(\underline{s}(x)))) \rightarrow \forall x \varphi(x)$

(A7) jest schematem (nieskończenie wielu) aksjomatów, zwanym *schematem indukcji matematycznej*.

Reguły wnioskowania

Regułami wnioskowania są:

- reguła podstawiania: $\frac{\psi}{\psi(x/t)}$, o ile term t jest podstawialny za zmienną x w formule ψ ,
- reguła odrywania: $\frac{\varphi \rightarrow \psi \quad \varphi}{\psi}$
- reguła generalizacji: $\frac{\psi}{\forall x \psi}$
- reguła ($O\forall$): $\frac{\varphi \rightarrow \forall x \psi}{\varphi \rightarrow \psi}$
- reguła ($D\forall$): $\frac{\varphi \rightarrow \psi}{\varphi \rightarrow \forall x \psi}$, o ile x nie jest zmienną wolną w φ
- reguła ($O\exists$): $\frac{\exists x \varphi \rightarrow \psi}{\varphi \rightarrow \psi}$
- reguła ($D\exists$): $\frac{\exists x \varphi \rightarrow \psi}{\varphi \rightarrow \psi}$, o ile x nie jest zmienną wolną w ψ .

Semantyka

Wykorzystujemy semantykę dla teorii pierwszego rzędu, podaną w elementarnym kursie logiki.

Spośród wszystkich modeli PA (a jest ich bardzo wiele!) wyróżnia się **model standardowy** \mathfrak{N}_0 , w którym:

- uniwersum stanowią wszystkie liczby naturalne $\{0, 1, 2, 3, \dots\}$
- interpretacją stałej $\underline{0}$ jest liczba zero,
- interpretacją predykatu $\underline{=}$ jest relacja równości $=$,
- interpretacją symbolu $\underline{+}$ jest operacja dodawania $+$,
- interpretacją symbolu $\underline{\times}$ jest operacja mnożenia \cdot ,
- interpretacją symbolu \underline{s} jest funkcja następnika $s: s(n) = n + 1$.

Arytmetyka Robinsona

- Aksjomat indukcji nie może zostać zastąpiony żadnym równoważnym mu skończonym zbiorem aksjomatów.
- Nie można zastąpić go także żadną równoważną (nawet nieskończoną) liczbą przypadków szczególnych, tj. aksjomatu indukcji dla formuł o dowolnie z góry ograniczonej liczbie kwantyfikatorów.

System w powyższym języku, o aksjomatach pozalogicznych (A1)–(A6) wraz z aksjomatem (A0) $\neg x \doteq \underline{0} \rightarrow \exists y(x \doteq \underline{s}(y))$ nazywa się **Arytmetyką Robinsona** i bywa oznaczany Q .

- Q jest istotnie słabszy od PA: można w nim dowodzić **konkretnych** prawd arytmetycznych (np. tego, że $2 + 2 = 4$), ale nie wielu praw ogólnych (np. tego, że dodawanie jest przemienne).

Wybrane proste twierdzenia arytmetyczne

Jeśli ψ jest twierdzeniem PA, to piszemy: $PA \vdash \psi$.

Wygodnie jest wprowadzić pewne zdefiniowane predykaty, np. *mniejszość*:

$$x \leq y \equiv_{df} \exists z (\neg z \doteq \underline{0} \wedge z \underline{+} x \doteq y).$$

Wtedy interpretacją \leq w \mathfrak{N}_0 jest relacja mniejszości $<$.

Uwaga. Nie myl predykatu \leq (podkreślony symbol $<$) z predykatem nazywającym relację „mniejsze lub równe”, którą zapisujemy \leq . Odróżniaj predykat identyczności \doteq od relacji równości $=$, stałą $\underline{0}$ od liczby 0, symbol funkcyjny $\underline{+}$ od funkcji następnika s , itd.

Niektóre proste twierdzenia arytmetyczne są potrzebne w dowodach twierdzeń o reprezentowalności oraz twierdzeń dotyczących arytmetyzacji składni. Podamy wybrane z nich.

Dygresja

Odróżniliśmy (poprzez kształt stosowanych symboli) stałe pozalogiczne arytmetyki od ich semantycznych interpretacji.

Dla pełnej (pedantycznej) poprawności trzeba byłoby również np. używać innych oznaczeń na stałe logiczne w języku przedmiotowym, a inne w tej roli w metajęzyku.

Podobnie, kształt zmiennych języka przedmiotowego powinien być różny od zmiennych metajęzykowych (przebiegających, w naszym przypadku, liczby naturalne).

Nie robimy tego. **Ufamy.** Ufamy, że kontekst użycia symbolu pozwala na uniknięcie nieporozumień w rozumieniu, co w tekście porabia ten symbol, jaki jest jego status, itd.

Wybrane proste twierdzenia arytmetyczne

W PA można udowodnić:

- łączność i przemienność dodawania i mnożenia,
- rozdzielność mnożenia względem dodawania,
- prawa skracania dla dodawania i mnożenia,
- zgodność porządku z operacjami arytmetycznymi.

- $PA \vdash \neg x \leq 0$.
- $PA \vdash x \leq s(y) \equiv (x \leq y \vee x \doteq y)$.
- $PA \vdash x \leq y \vee x \doteq y \vee y \leq x$.

Wybrane proste twierdzenia arytmetyczne

Niech formuła φ' powstaje z formuły φ poprzez zastąpienie niektórych wystąpień formuł ψ_1, \dots, ψ_n odpowiednio formułami ψ'_1, \dots, ψ'_n i założymy, że: $PA \vdash \psi_1 \equiv \psi'_1, \dots, PA \vdash \psi_n \equiv \psi'_n$. Wtedy: $PA \vdash \varphi \equiv \varphi'$.

Niech term t powstaje z termu t' poprzez (poprawne!) zastąpienie niektórych wystąpień termów t_1, \dots, t_n odpowiednio termami t'_1, \dots, t'_n , a formuła φ' powstaje z formuły φ poprzez takie samo zastąpienie wymienionych termów. Założymy, że: $PA \vdash t_1 \doteq t'_1, \dots, PA \vdash t_n \doteq t'_n$. Wtedy: $PA \vdash t \doteq t'$ oraz $PA \vdash \varphi \equiv \varphi'$.

Powyższe twierdzenia zachodzą nie tylko dla PA, lecz także dla dowolnej teorii pierwszego rzędu.

Liczebniki

Definicja liczebników.

- Term $\underline{0}$ jest liczebnikiem.
- Jeśli term α jest liczebnikiem, to term $\underline{s}(\alpha)$ jest liczebnikiem.
- Liczebnikami są tylko termy opisane w powyższy sposób.

Oznaczmy: $\bar{n} = \underbrace{\underline{s}(\underline{s}(\dots \underline{s}(\underline{0}) \dots))}_{n \text{ razy}}$.

\bar{n} jest zatem liczebnikiem nazywającym liczbę n .

Uwaga. Liczebniki są symbolami językowymi, liczby naturalne są elementami uniwersum modelu standardowego.

Słaba reprezentowalność relacji

- Formuła φ języka PA o n zmiennych wolnych **słabo reprezentuje** w PA relację $R \subseteq \omega^n$, wtedy i tylko wtedy, gdy dla dowolnych liczb naturalnych k_1, \dots, k_n zachodzi równoważność:
 $R(k_1, \dots, k_n)$ wtedy i tylko wtedy, gdy $PA \vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$.
- Relację $R \subseteq \omega^n$ nazywamy **słabo reprezentowalną** w PA, jeśli istnieje formuła języka PA, która słabo reprezentuje R .

Uwaga. Formuła φ słabo reprezentuje R w PA wtedy i tylko wtedy, gdy zachodzą implikacje:

- Jeśli $R(k_1, \dots, k_n)$, to $\vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$.
- Jeśli $\vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$, to $R(k_1, \dots, k_n)$.

Mocna reprezentowalność relacji

- Formuła φ języka PA o n zmiennych wolnych ***mocno reprezentuje*** w PA relację $R \subseteq \omega^n$, wtedy i tylko wtedy, gdy dla dowolnych liczb naturalnych k_1, \dots, k_n zachodzą implikacje:
 - Jeśli $R(k_1, \dots, k_n)$, to $PA \vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$.
 - Jeśli $\neg R(k_1, \dots, k_n)$, to $PA \vdash \neg\varphi(\overline{k_1}, \dots, \overline{k_n})$.
- Relację $R \subseteq \omega^n$ nazywamy ***mocno reprezentowalną*** w PA, jeśli istnieje formuła języka PA, która mocno reprezentuje R .

Uwaga. Każda relacja mocno reprezentowalna w PA jest też słabo reprezentowalna w PA, lecz nie na odwrót.

Reprezentowalność relacji

Jeśli PA jest niesprzeczna oraz R jest mocno reprezentowana w PA przez formułę φ , to zachodzą następujące równoważności:

- $R(k_1, \dots, k_n)$ wtedy i tylko wtedy, gdy $PA \vdash \varphi(\overline{k_1}, \dots, \overline{k_n})$.
- $\neg R(k_1, \dots, k_n)$ wtedy i tylko wtedy, gdy $PA \vdash \neg\varphi(\overline{k_1}, \dots, \overline{k_n})$.

Na mocy powyższego twierdzenia, relacja R jest mocno reprezentowalna w PA wtedy i tylko wtedy, gdy istnieje formuła φ języka PA taka, że:

- R jest słabo reprezentowana przez φ ,
- $\neg R$ jest słabo reprezentowana przez $\neg\varphi$.

Reprezentowalność funkcji

- Formuła φ języka PA o $n + 1$ zmiennych wolnych **reprezentuje** w PA funkcję $f : \omega^n \rightarrow \omega$ wtedy i tylko wtedy, gdy dla dowolnych liczb naturalnych k_1, \dots, k_n :

$$\text{PA} \vdash \forall y (\varphi(\overline{k_1}, \dots, \overline{k_n}, y) \equiv (y \doteq \overline{f(k_1, \dots, k_n)})).$$

- Funkcję $f : \omega^n \rightarrow \omega$ nazywamy **reprezentowalną** w PA wtedy i tylko wtedy, gdy istnieje formuła φ języka PA o $n + 1$ zmiennych wolnych taka, że φ reprezentuje f w PA.

Reprezentowalność

- Relacja identyczności = jest mocno reprezentowana w PA przez formułę $x_1 \doteq x_2$.
- Funkcja dodawania jest reprezentowana w PA przez formułę $x_1 \dot{+} x_2 \doteq x_3$.
- Funkcja mnożenia jest reprezentowana w PA przez formułę $x_1 \dot{\times} x_2 \doteq x_3$.
- Relacja mniejszości jest mocno reprezentowana w PA przez formułę $x_1 \leq x_2$.
- Relacja $R \subseteq \omega^n$ jest mocno reprezentowalna w PA wtedy i tylko wtedy, gdy jej funkcja charakterystyczna jest reprezentowalna w PA.

Twierdzenie o reprezentowalności

Dla dowolnej formuły φ języka PA i dowolnej liczby naturalnej n :
 $PA \vdash (\varphi(0) \wedge \varphi(\bar{1}) \wedge \dots \wedge \varphi(\overline{n-1}) \wedge x \leq \bar{n}) \rightarrow \varphi(x)$.

Dla dowolnej formuły φ języka PA i dowolnej liczby naturalnej n , jeżeli dla każdego $i < n$, $PA \vdash \neg\varphi(\bar{i})$ oraz $PA \vdash \varphi(\bar{n})$, to:
 $PA \vdash (\varphi(x) \wedge \forall y(y \leq x \rightarrow \neg\varphi(y))) \equiv (x \doteq \bar{n})$.

Twierdzenie o reprezentowalności.

- Każda funkcja rekurencyjna jest reprezentowalna w PA.
- Każda relacja rekurencyjna jest mocno reprezentowalna w PA.

Twierdzenie o reprezentowalności

Dowód twierdzenia o reprezentowalności jest dość łatwy, choć nieco żmudny. Dowodzi się mianowicie kolejno, że:

- funkcje proste są reprezentowalne,
- funkcja powstająca przez złożenie z funkcji reprezentowalnych jest reprezentowalna,
- funkcja powstająca przez schemat rekursji prostej z funkcji reprezentowalnych jest reprezentowalna,
- funkcja powstająca przez zastosowanie operacji minimum efektywnego do funkcji reprezentowalnej jest reprezentowalna.

W dowodzie wykorzystuje się (metasystemową) zasadę indukcji matematycznej.

Arytmetyzacja składni

Możliwość kodowania wyrażen języka PA przez liczby naturalne pozwala na „mówienie” o arytmetyce w niej samej, i to bez popadania w paradoksy pomieszania języka przedmiotowego i metajęzyka.

Dokładniej, owo „mówienie” w PA o PA umożliwia jest dodatkowo przez dwa fakty: to, że pojęciom metalogicznym odpowiadają funkcje i relacje rekurencyjne oraz to, że funkcje i relacje rekurencyjne są reprezentowalne w PA.

Są różne możliwości (jednoznacznego i efektywnego) kodowania wyrażen (skończonych ciągów symboli), np.:

- użycie rozkładu liczb naturalnych na czynniki pierwsze;
- zastosowanie funkcji kodującej Cantora;
- zastosowanie funkcji kodującej β Gödla;
- konkatencję w bazie dziesiętnej, itd.

Wybór funkcji kodującej ciągi

Lemat (funkcja β Gödla). *Istnieje 2-argumentowa funkcja rekurencyjna β taka, że:*

- dla dowolnych a oraz i : $\beta(a, i) \leq a \div 1$
 - dla dowolnych a_0, a_1, \dots, a_{n-1} istnieje a taka, że $\beta(a, i) = a_i$, dla $i < n$.
-
- Dla dowolnego ciągu liczb naturalnych a_1, \dots, a_n niech:
 $\langle a_1, \dots, a_n \rangle^\beta = \mu x (\beta(x, 0) = n \wedge \beta(x, 1) = a_1 \wedge \dots \wedge \beta(x, n) = a_n)$.
 - $lh^\beta(a) = \beta(a, 0)$.
 - $(a)_i^\beta = \beta(a, i + 1)$.
 - $seq^\beta(a) \equiv \forall x < a (lh^\beta(x) \neq lh^\beta(a) \vee \exists i < lh^\beta(a) (x)_i^\beta \neq (a)_i^\beta)$.
 - Dla każdej n , funkcja $(a_1, \dots, a_n) \mapsto \langle a_1, \dots, a_n \rangle^\beta$ jest rekurencyjna. Funkcje lh^β i $()_i^\beta$ są rekurencyjne. seq^β jest relacją rekurencyjną.

Wybór funkcji kodującej ciągi

Następujące funkcje są rekurencyjne (co widać z definicji):

- $ln^\beta(a, i) = \mu x (lh^\beta(x) = i \wedge \forall j < i (x)_j^\beta = (a)_j^\beta)$.
- $a *^\beta b = \mu x (lh^\beta(x) = lh^\beta(a) + lh^\beta(b) \wedge \forall i < lh^\beta(a) (x)_i^\beta = (a)_i^\beta \wedge \forall i < lh^\beta(b) (x)_{lh^\beta(a)+i}^\beta = (b)_i^\beta)$.

- Wcześniej poznaliśmy funkcje podobne do powyższych (kodujące i odkodowujące) – funkcję pary Cantora oraz funkcję kodowania wykorzystującą rozkład na czynniki pierwsze.
- W dalszym ciągu będziemy pomijać indeks górny β przy wprowadzonych wyżej symbolach.

Dowód lematu

Zarys dowodu lematu.

- Przypomnijmy, że relacja podzielności div jest rekurencyjna:
 $div(a, b) \equiv \exists z \leq a (a = b \cdot z)$.
- Rekurencyjna jest też symetryczna relacja rp zdefiniowana wzorem:
 $rp(a, b) \equiv a \neq 0 \wedge b \neq 0 \wedge \forall x \leq a \cdot b (div(a \cdot x, b) \rightarrow div(x, b))$,
 która zachodzi między a i b dokładnie wtedy, gdy a i b są względnie pierwsze.
- (*) Dla dowolnych różnych od 0 i od 1 liczb $a_1, \dots, a_n, b_1, \dots, b_m$ takich, że $rp(a_i, b_j)$ dla wszystkich i oraz j istnieje liczba c taka, że $div(c, a_i)$ dla $i = 1, \dots, n$, natomiast nie zachodzi $div(c, b_j)$ dla $j = 1, \dots, m$.
- Wprost z definicji relacji div oraz rp mamy:
 $(\dagger) (k \neq 0 \wedge z \neq 0 \wedge div(z, k)) \rightarrow rp(1 + (j + k) \cdot z, 1 + j \cdot z)$.

Dowód lematu

- Funkcja $op(a, b) = (a + b) \cdot (a + b) + a + 1$ jest rekurencyjna. Nadto:
 $(\dagger) \quad op(a, b) = op(c, d) \rightarrow a = c \wedge b = d.$
- Definiujemy funkcję β :

$$\beta(a, i) = \mu x < a \div 1 (\exists y < a \exists z < a (a = op(y, z) \wedge div(y, 1 + (op(x, i) + 1) \cdot z))).$$
- Z definicji wynika, że β jest rekurencyjna oraz że $\beta(a, i) \leq a \div 1$.
- Dla danych liczb a_0, a_1, \dots, a_{n-1} trzeba znaleźć liczbę a taką, że $\beta(a, i) = a_i$ dla $i < n$.
- Niech c będzie największą z liczb $op(a_i, i) + 1$ dla $i < n$, natomiast z liczbą podzielną przez wszystkie x takie, że $x < c$.
- Na mocy (\dagger) dla $k = r - j$ mamy: jeśli $j < r < c$, to $rp(1 + j \cdot z, 1 + r \cdot z)$.

Dowód lematu

- Na mocy (*) istnieje y taka, że dla $j < c$: $div(y, 1 + j \cdot z)$ dokładnie wtedy, gdy j jest postaci $op(a_i, i) + 1$.
- Niech $a = op(y, z)$. Wtedy $a_i < y < a$ oraz $z < a$.
- Dla dowodu, że $\beta(a, i) = a_i$ dla $i < n$ wystarczy pokazać, że:
 $a_i = \mu x \, div(y, 1 + (op(x, i) + 1) \cdot z)$.
- To wynika z faktu, że: jeśli $x < a_i$, to $op(x, i) < c$ oraz $op(x, i)$ nie jest postaci $op(a_j, j)$ dla $j < n$.
- Na mocy (‡) mamy jednak: $op(x, i) \leq op(a_i, i) < c$ oraz $op(x, i) \neq op(a_j, j)$, co kończy dowód.

Funkcja-pamięć

- Dla dowolnej funkcji $f : \omega^n \rightarrow \omega$ jej **ściągnięciem** nazywamy funkcję:

$$\lceil f \rceil(a) = f((a)_0, \dots, (a)_{n-1})$$
 - Dla dowolnej relacji $R \subseteq \omega^n$ jej **ściągnięciem** nazywamy relację:

$$\lceil R \rceil(a) \equiv R((a)_0, \dots, (a)_{n-1}).$$
 - Dla dowolnej funkcji $f : \omega^n \rightarrow \omega$ jej **funkcją-pamięcią** nazywamy funkcję:

$$\widehat{f}(a_1, \dots, a_{n-1}, b) = \langle f(a_1, \dots, a_{n-1}, 0), f(a_1, \dots, a_{n-1}, 1), \dots, f(a_1, \dots, a_{n-1}, b-1) \rangle.$$
-
- $f(a_1, \dots, a_n) = \lceil f \rceil(\langle a_1, \dots, a_n \rangle).$
 - $R(a_1, \dots, a_n) \equiv \lceil R \rceil(\langle a_1, \dots, a_n \rangle).$

Funkcja-pamięć może też zostać określona przy użyciu innych (pierwotnie rekurencyjnych) funkcji kodujących.

Funkcja-pamięć

- Funkcja f jest rekurencyjna dokładnie wtedy, gdy rekurencyjna jest jej funkcja-pamięć \hat{f} .

Dowód.

- Jeśli f rekurencyjna, to \hat{f} rekurencyjna, ponieważ:

$$\hat{f}(a_1, \dots, a_n, b) = \mu x (seq(x) \wedge lh(x) = b \wedge \forall i < b ((x)_i) = f(a_1, \dots, a_n, b)).$$
- Jeśli \hat{f} rekurencyjna, to f rekurencyjna, ponieważ:

$$f(a_1, \dots, a_n, b) = (\hat{f}(a_1, \dots, a_n, b + 1))_b.$$

Funkcja-pamięć pozwala zastąpić pewne definicje przez schematy rekursji definicjami używającymi tylko funkcji pierwotnie rekurencyjnych oraz efektywnego operatora minimum:

Funkcja-pamięć

Jeśli g oraz h są funkcjami rekurencyjnymi, to funkcja f określona następującym schematem rekursji:

- $f(a_1, \dots, a_n, b) = g(a_1, \dots, a_n)$, gdy $b = 0$,
- $f(a_1, \dots, a_n, b) = h(\widehat{f}(a_1, \dots, a_n, b), a_1, \dots, a_n, b)$, gdy $b > 0$

także jest rekurencyjna.

Mamy bowiem:

$$H(a_1, \dots, a_n, b) = \mu x (seq(x) \wedge lh(x) = b \wedge (x)_0 = g(a_1, \dots, a_n) \wedge \forall i < b (i > 0 \rightarrow (x)_i = h(ln(x, i), a_1, \dots, a_n, i))).$$

A zatem H jest rekurencyjna. Nadto, jest równa funkcji f . Rekurencyjność f jest teraz oczywista:

- dla $b = 0$ funkcja f jest równa rekurencyjnej funkcji g ,
- dla $b > 0$ mamy: $f(a_1, \dots, a_n, b) = h(H(a_1, \dots, a_n, b), a_1, \dots, a_n, b)$.

Funkcja-pamięć

W konsekwencji, jeśli g i h są rekurencyjne, to funkcja f określona warunkami:

- $f(a_1, \dots, a_n, 0) = g(a_1, \dots, a_n,)$
- $f(a_1, \dots, a_n, b + 1) = h(f(a_1, \dots, a_n, b), a_1, \dots, a_n, b)$

również jest rekurencyjna.

Wyrażenie schematów rekursji przez funkcję-pamięć pozwala lepiej zrozumieć działanie tych schematów.

Własności operacji ściągnięcia ukazują natomiast, że funkcje (i relacje) wieloargumentowe (na liczbach naturalnych) możemy zastąpić przez równe im funkcje (i relacje) jednoargumentowe.

Numery symboli

Każdemu symbolowi X języka PA przyporządkowujemy *numer* $sn(X)$, np. w taki sposób:

| | | | | | | |
|---------|--------|--------|----------|---------------|----------|-----------|
| X | \neg | \vee | \wedge | \rightarrow | \equiv | \exists |
| $sn(X)$ | 3 | 5 | 7 | 9 | 11 | 13 |

| | | | | | | |
|---------|-----------|-----------------|-----------------|----------------------|-----------------|-----------|
| X | \forall | \underline{s} | $\underline{+}$ | $\underline{\times}$ | $\underline{0}$ | $\dot{=}$ |
| $sn(X)$ | 15 | 17 | 19 | 21 | 23 | 25 |

Zmiennej x_i przyporządkowujemy liczbę $2i$: $sn(x_i) = 2i$.

Kodowanie wyrażeń

Zakładamy, że każdy term i każda formuła języka PA są zapisane w notacji **prefiksowej** (**polskiej**), czyli: funktor przed swoim argumentami. Tak więc, każdy term jest ciągiem o postaci $vv_1 \dots, v_n$, gdzie v jest symbolem funkcyjnym, a $v_1 \dots, v_n$ są termami. Każda formuła jest ciągiem o postaci $vv_1 \dots, v_n$, gdzie: albo v jest funktorem prawdziwościowym, a $v_1 \dots, v_n$ są formułami lub termami, albo v jest kwantyfikatorem, a $v_1 \dots, v_n$ są formułami lub termami. Notacja prefiksowa nie wymaga stosowania nawiasów.

- Każdemu termowi lub formule u o postaci $vv_1 \dots v_n$ przyporządkowujemy jego/jej **numer gödłowski** $\ulcorner u \urcorner$, zdefiniowany następująco:

$$\ulcorner u \urcorner = \langle sn(v), \ulcorner v_1 \urcorner, \dots, \ulcorner v_n \urcorner \rangle.$$
- Powyższa definicja może też być „rozpisana” w sposób wyraźny, dla każdego rodzaju termu lub formuły.

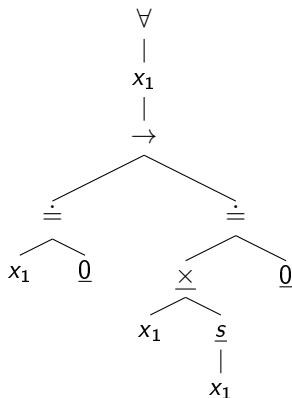
Kodowanie wyrażeń

Dla przykładu, numer gödłowski formuły $\forall x_1 (x_1 \doteq \underline{0} \rightarrow (x_1 \underline{\times} \underline{s}(x_1)) \doteq \underline{0})$ obliczamy następująco:

- Przekształcamy formułę do postaci prefiksowej:
 $\forall x_1 \rightarrow \doteq x_1 \underline{0} \doteq \underline{\times} x_1 \underline{s} x_1 \underline{0}$ (i w znany sposób odszukujemy jej podformuły).
- $\ulcorner \underline{s}(x_1) \urcorner = \langle 17, 2 \rangle$
- $\ulcorner x_1 \underline{\times} \underline{s}(x_1) \urcorner = \langle 21, 2, \langle 17, 2 \rangle \rangle$
- $\ulcorner (x_1 \underline{\times} \underline{s}(x_1) \doteq \underline{0}) \urcorner = \langle 25, \langle 21, 2, \langle 17, 2 \rangle \rangle, 23 \rangle$
- $\ulcorner x_1 \doteq \underline{0} \urcorner = \langle 25, 2, 23 \rangle$
- $\ulcorner x_1 \doteq \underline{0} \rightarrow (x_1 \underline{\times} \underline{s}(x_1)) \urcorner = \langle 9, \langle 25, 2, 23 \rangle, \langle 25, \langle 21, 2, \langle 17, 2 \rangle \rangle, 23 \rangle \rangle$
- $\ulcorner \forall x_1 (x_1 \doteq \underline{0} \rightarrow (x_1 \underline{\times} \underline{s}(x_1)) \doteq \underline{0}) \urcorner =$
 $\langle 15, 2, \langle 9, \langle 25, 2, 23 \rangle, \langle 25, \langle 21, 2, \langle 17, 2 \rangle \rangle, 23 \rangle \rangle \rangle$

Kodowanie wyrażeń

Ten sposób kodowania staje się jasny, gdy spojrzymy na drzewo syntaktyczne rozważanej formuły (w istocie kodujemy właśnie to drzewo):



Kodowanie wyrażeń

Omówione kodowanie wyrażeń jest:

- jednoznaczne (każdy term lub formuła otrzymuje dokładnie jeden numer gödłowski);
- efektywne (przypisanie numerów realizowane jest za pomocą funkcji rekurencyjnych).

Numery gödłowskie termów i formuł są oczywiście dość dużymi liczbami. W praktyce nie ma jednak żadnej potrzeby, aby je obliczać. Wystarczy możliwość ich jednoznacznego, efektywnego otrzymania.

Kodowanie wyrażeń

Ponadto, dla dowolnej liczby naturalnej a można w sposób efektywny rozstrzygnąć, czy jest ona numerem gödłowskim termu lub formuły. Wystarczy w tym celu:

- sprawdzić, czy zachodzi $seq(a)$;
- jeśli tak, to wyznaczyć $lh(a)$ oraz $(a)_i$ dla $i = 0, 1, \dots, lh(a)$;
- dla tak „rozłożonej” $a = \langle a_0, a_1, \dots, a_n \rangle$ sprawdzić, czy $a_0 = sn(X)$ dla jakiegoś symbolu X alfabetu języka PA;
- jeśli tak, to procedurę tę powtórzyć dla a_i , gdzie $i = 0, 1, \dots, lh(a)$;
- ponieważ $a_i < a$, więc po skończonej liczbie kroków procedura ta się zakończy (i uzyskamy odpowiedź czy a jest numerem gödłowskim termu lub formuły).

Kodowanie zmiennych

$$\text{Vble}(a) \equiv a = \langle (a)_0 \rangle \wedge \exists y \leq a ((a)_0 \dot{=} 2y)$$

$\text{Vble}(a)$ zachodzi, gdy a jest numerem gödłowskim pewnej zmiennej x_i , czyli gdy $a = \ulcorner x_i \urcorner$.

Rekurencyjność Vble wynika z faktu, że $(a)_0 < a$.

Zauważmy, że numer gödłowski zmiennej to nie to samo, co numer przypisany jej przez funkcję sn .

Kodowanie termów

Formuła $\text{Term}(a)$ jest równoważna:

| formule | jeśli |
|--|--|
| $0 = 0$ | $a = \langle sn(\underline{0}) \rangle$ |
| $\text{Term}((a)_1)$ | $a = \langle sn(\underline{s}), (a)_1 \rangle$ |
| $\text{Term}((a)_1) \wedge \text{Term}((a)_2)$ | $a = \langle sn(\underline{+}), (a)_1, (a)_2 \rangle$ |
| $\text{Term}((a)_1) \wedge \text{Term}((a)_2)$ | $a = \langle sn(\underline{\times}), (a)_1, (a)_2 \rangle$ |
| $\text{Vble}(a)$ | w p.p. |

$\text{Term}(a)$ czytamy: a jest numerem gödłowskim termu, czyli $a = \ulcorner t \urcorner$ dla pewnego termu t . Skrót „w p.p.” czytamy: „w pozostałych przypadkach.” Rekurencyjność Term dostajemy z: faktu, iż $(a)_1 < a$ oraz twierdzeń o definiowaniu warunkowym i definiowaniu przez schemat rekursji.

Kodowanie formuł

$\text{AtForm}(a) \equiv$

$$a = \langle (a)_0, (a)_1, (a)_2 \rangle \wedge (a)_0 = sn(\dot{=}) \wedge \text{Term}((a)_1) \wedge \text{Term}((a)_2).$$

Formuła $\text{Form}(a)$ jest równoważna:

| formule | jeśli |
|--|---|
| $\text{Form}((a)_1)$ | $a = \langle sn(\neg), (a)_1 \rangle$ |
| $\text{Form}((a)_1) \wedge \text{Form}((a)_2)$ | $a = \langle sn(\vee), (a)_1, (a)_2 \rangle$ |
| $\text{Form}((a)_1) \wedge \text{Form}((a)_2)$ | $a = \langle sn(\wedge), (a)_1, (a)_2 \rangle$ |
| $\text{Form}((a)_1) \wedge \text{Form}((a)_2)$ | $a = \langle sn(\rightarrow), (a)_1, (a)_2 \rangle$ |
| $\text{Form}((a)_1) \wedge \text{Form}((a)_2)$ | $a = \langle sn(\equiv), (a)_1, (a)_2 \rangle$ |
| $\text{Vble}((a)_1) \wedge \text{Form}((a)_2)$ | $a = \langle sn(\exists), (a)_1, (a)_2 \rangle$ |
| $\text{Vble}((a)_1) \wedge \text{Form}((a)_2)$ | $a = \langle sn(\forall), (a)_1, (a)_2 \rangle$ |
| $\text{AtForm}(a)$ | w p.p. |

$\text{Form}(a)$ zachodzi, gdy a jest numerem gödłowskim formuły języka PA.

Kodowanie operacji syntaktycznych

Funkcja $\text{Sub}(a, b, c)$:

| ma wartość | jeśli |
|---|---|
| c | $\forall b \text{le}(a) \wedge a = b$ |
| $\langle (a)_0, \text{Sub}((a)_1, b, c) \rangle$ | $a = \langle (a)_0, (a)_1 \rangle$ |
| $\langle (a)_0, \text{Sub}((a)_1, b, c), \text{Sub}((a)_2, b, c) \rangle$ | $a = \langle (a)_0, (a)_1, (a)_2 \rangle \wedge$ $(a)_0 \neq \text{sn}(\exists) \wedge (a)_0 \neq \text{sn}(\forall)$ |
| $\langle (a)_0, (a)_1, \text{Sub}((a)_2, b, c) \rangle$ | $(a = \langle \text{sn}(\exists), (a)_1, (a)_2 \rangle \vee$ $a = \langle \text{sn}(\exists), (a)_1, (a)_2 \rangle) \wedge (a)_1 \neq b$ |
| a | w p.p. |

Dla termów t, t_1, t_2 , zmiennej x oraz formuły ψ mamy: $\text{Sub}(\ulcorner t_1 \urcorner, \ulcorner x \urcorner, \ulcorner t_2 \urcorner) = \ulcorner t_1(x/t_2) \urcorner$ (podstawienie t_2 za x w t_1) $\text{Sub}(\ulcorner \psi \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner) = \ulcorner \psi(x/t) \urcorner$ (podstawienie t za x w ψ).

Kodowanie operacji syntaktycznych

Formuła $\text{Fr}(a, b)$ jest równoważna:

| formule | jeśli |
|--|--|
| $a = b$ | $\forall ble(a)$ |
| $\text{Fr}((a)_1, b)$ | $a = \langle (a)_0, a_1 \rangle$ |
| $\text{Fr}((a)_1, b) \vee \text{Fr}((a)_2, b)$ | $a = \langle (a)_0, a_1, a_2 \rangle \wedge$ $(a)_0 \neq sn(\exists) \wedge (a)_0 \neq sn(\forall)$ |
| $\text{Fr}((a)_2, b) \wedge (a)_1 \neq b$ | w p.p. |

Dla formuły ψ oraz zmiennej x zachodzi relacja $\text{Fr}(\ulcorner \psi \urcorner, \ulcorner x \urcorner)$ dokładnie wtedy, gdy x jest zmienną wolną w ψ .

Kodowanie operacji syntaktycznych

Formuła $\text{Subtl}(a, b, c)$ jest równoważna:

| formule | jeśli |
|---|--|
| $\text{Subtl}((a)_1, b, c)$ | $a = \langle (a)_0, (a)_1 \rangle$ |
| $\text{Subtl}((a)_1, b, c) \wedge \text{Subtl}((a)_2, b, c)$ | $a = \langle (a)_0, (a)_1, (a)_2 \rangle \wedge$ $(a)_0 \neq \text{sn}(\exists) \wedge (a)_0 \neq \text{sn}(\forall)$ |
| $\text{Subtl}((a)_2, b, c) \wedge$ $\wedge (\neg \text{Fr}((a)_2, b) \vee \neg \text{Fr}(c, (a)_1))$ | $(a = \langle \text{sn}(\exists), (a)_1, (a)_2 \rangle \vee$ $a = \langle \text{sn}(\forall), (a)_1, (a)_2 \rangle) \wedge$ $(a)_1 \neq b$ |
| $0 = 0$ | w p.p. |

$\text{Subtl}(\ulcorner \psi \urcorner, \ulcorner x \urcorner, \ulcorner t \urcorner)$ zachodzi dokładnie wtedy, gdy term t jest podstawialny za zmienną x do formuły ψ .

Kodowanie aksjomatów logicznych

Formuła $\text{LogAx}(a)$ jest alternatywą 14 warunków:

- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, x, \langle 9, y, x \rangle \rangle)$
- $\exists x < a \exists y < a \exists z < a (\text{Form}(x) \wedge \text{Form}(y) \wedge \text{Form}(z) \wedge$
 $a = \langle 9, \langle 9, x, \langle 9, y, z \rangle \rangle, \langle 9, \langle 9, x, y \rangle, \langle 9, x, z \rangle \rangle \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, \langle 9, x, y \rangle, \langle 9, \langle 3, y \rangle, \langle 3, x \rangle \rangle \rangle)$
- $\exists x < a (\text{Form}(x) \wedge a = \langle 9, \langle 3, \langle 3, x \rangle \rangle, x \rangle)$
- $\exists x < a (\text{Form}(x) \wedge a = \langle 9, x, \langle 3, \langle 3, x \rangle \rangle \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, \langle 7, x, y \rangle, x \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, \langle 7, x, y \rangle, y \rangle)$

Kodowanie aksjomatów logicznych

- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge \text{Form}(z) \wedge$
 $a = \langle 9, \langle 9, x, y \rangle, \langle 9, \langle 9, x, z \rangle, \langle 9, x, \langle 7, y, z \rangle \rangle \rangle \rangle$)
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, x, \langle 5, x, y \rangle \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, y, \langle 5, x, y \rangle \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge \text{Form}(z) \wedge$
 $a = \langle 9, \langle 9, x, z \rangle, \langle 9, \langle 9, y, z \rangle, \langle 9, \langle 5, x, y \rangle, z \rangle \rangle \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, \langle 11, x, y \rangle, \langle 9, x, y \rangle \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, \langle 11, x, y \rangle, \langle 9, y, x \rangle \rangle)$
- $\exists x < a \exists y < a (\text{Form}(x) \wedge \text{Form}(y) \wedge a = \langle 9, \langle 9, x, y \rangle, \langle 9, \langle 9, y, x \rangle, \langle 11, x, y \rangle \rangle \rangle)$

$\text{LogAx}(a)$ zachodzi dokładnie wtedy, gdy a jest numerem gödłowskim aksjomatu logicznego PA.

Kodowanie aksjomatów identyczności

Formuła $\text{EqAx}(a)$ jest alternatywą 8 warunków:

- $\exists x < a (a = \langle 25, 2x, 2x \rangle)$
- $\exists x < a \exists y < a (a = \langle 9, \langle 25, 2x, 2y \rangle, \langle 25, 2y, 2x \rangle \rangle)$
- $\exists x < a \exists y < a \exists z < a (a = \langle 9, \langle 7, \langle 25, 2x, 2y \rangle, \langle 25, 2y, 2z \rangle, \langle 25, 2x, 2z \rangle \rangle \rangle)$
- $\exists x < a \exists y < a \exists z < a (a = \langle 9, \langle 25, 2x, 2y \rangle, \langle 25, \langle 17, 2x \rangle, \langle 17, 2y \rangle \rangle \rangle)$
- $\exists x < a \exists y < a \exists z < a (a = \langle 9, \langle 25, 2x, 2y \rangle, \langle 25, \langle 19, 2x, 2z \rangle, \langle 19, 2y, 2z \rangle \rangle \rangle)$
- $\exists x < a \exists y < a \exists z < a (a = \langle 9, \langle 25, 2x, 2y \rangle, \langle 25, \langle 19, 2z, 2x \rangle, \langle 19, 2z, 2y \rangle \rangle \rangle)$
- $\exists x < a \exists y < a \exists z < a (a = \langle 9, \langle 25, 2x, 2y \rangle, \langle 25, \langle 21, 2x, 2z \rangle, \langle 21, 2y, 2z \rangle \rangle \rangle)$
- $\exists x < a \exists y < a \exists z < a (a = \langle 9, \langle 25, \langle 25, 2x, 2y \rangle, \langle 25, \langle 21, 2z, 2x \rangle, \langle 21, 2z, 2v \rangle \rangle \rangle \rangle)$

$\text{EqAx}(a)$ zachodzi dokładnie wtedy, gdy a jest numerem gödłowskim jednego z aksjomatów identyczności PA.

Kodowanie aksjomatów pozalogicznych

Formuła $NLAx(a)$ jest alternatywą następujących warunków:

- $\exists x < a \exists y < a ((Vble(x) \wedge Vble(y)) \wedge a = \langle 9, \langle 25, \langle 17, x \rangle, \langle 17, y \rangle \rangle, \langle 25, x, y \rangle \rangle)$
- $\exists x < a (Vble(x) \wedge a = \langle 3, \langle 25, 23, \langle 17, x \rangle \rangle \rangle)$
- $\exists x < a (Vble(x) \wedge a = \langle 25, \langle 19, x, 23 \rangle, x \rangle)$
- $\exists x < a \exists y < y ((Vble(x) \wedge Vble(y)) \wedge a = \langle 25, \langle 19, x, \langle 17, y \rangle \rangle, \langle 17, \langle 19, x, y \rangle \rangle \rangle)$
- $\exists x < a (Vble(x) \wedge a = \langle 25, \langle 21, x, 23 \rangle, 23 \rangle)$
- $\exists x < a \exists y < a ((Vble(x) \wedge Vble(y)) \wedge a = \langle 25, \langle 21, x, \langle 17, y \rangle \rangle, \langle 19, \langle 21, x, y \rangle, x \rangle \rangle)$
- $\exists x < a \exists y < a ((Form(x) \wedge Vble(y) \wedge Fr(x, y)) \wedge$
 $a = \langle 9, \langle 7, Sub(x, y, 23) \rangle, \langle 15, y, \langle 9, x, Sub(x, y, \langle 17, y \rangle \rangle) \rangle \rangle, \langle 15, y, x \rangle \rangle)$

$NLAx(a)$ zachodzi dokładnie wtedy, gdy a jest numerem gödłowskim aksjomatu pozalogicznego PA.

Zauważmy, że ostatni z powyższych warunków odpowiada nieskończonemu zbiorowi aksjomatów.

Kodowanie reguł wnioskowania

Relacja $\text{sub}(a, b)$ zachodzi dokładnie wtedy, gdy istnieją $x < b$ oraz $y < b$ takie, że:

- $\text{Term}(x)$
- $\text{Vble}(y)$
- $\text{Form}(a)$
- $\text{Form}(b)$
- $\text{Subtl}(a, y, x)$
- $b = \text{Sub}(a, y, x)$.

$\text{sub}(\ulcorner \varphi \urcorner, \ulcorner \psi \urcorner)$ zachodzi dokładnie wtedy, gdy formuła ψ powstaje z formuły φ przez podstawienie w φ pewnego termu za pewną zmienną.

Kodowanie reguł wnioskowania

Relacja $MP(a, b, c)$ zachodzi, gdy:

- $Form(a)$
- $Form(b)$
- $Form(c)$
- $a = \langle 9, (a)_1, (a)_2 \rangle$
- $b = (a)_1$
- $c = (a)_2$.

$MP(a, b, c)$ zachodzi dokładnie wtedy, gdy formuła o numerze gödłowskim c jest wnioskiem reguły odrywania, gdzie przesłankami są formuły o numerach gödłowskich a oraz b .

Kodowanie reguł wnioskowania

Relacja $GR(a, b)$ zachodzi, gdy istnieje $x < b$ taka, że:

- $Form(a)$
- $Form(b)$
- $\forall x$
- $(b)_0 = sn(\forall)$
- $(b)_1 = x$
- $(b)_2 = a$.

$GR(a, b)$ zachodzi dokładnie wtedy, gdy gdy formuła o numerze gödłowskim b jest wnioskiem reguły generalizacji, gdzie przesłanką jest formuła o numerze gödłowskim a .

Kodowanie reguł wnioskowania

Relacja $OA(a, b)$ zachodzi, gdy istnieją $x < a$, $y < a$ oraz $z < b$ takie, że:

- $Form(x)$
- $Form(y)$
- $Vble(z)$
- $a = \langle 9, x, \langle 13, z, y \rangle \rangle$
- $b = \langle 9, x, y \rangle$.

Relacja $OA(a, b)$ zachodzi dokładnie wtedy, gdy b jest numerem gödłowskim wniosku reguły opuszczania kwantyfikatora generalnego, gdzie przesłanka ma numer gödłowski a .

Kodowanie reguł wnioskowania

Relacja $DA(a, b)$ zachodzi, gdy istnieją $x < a$, $y < a$ oraz $z < b$ takie, że:

- $\text{Form}(x)$
- $\text{Form}(y)$
- $\text{Vble}(z)$
- $a = \langle 9, x, y \rangle$
- $b = \langle 9, x, \langle 13, z, y \rangle \rangle$
- $\neg \text{Fr}(x, z)$.

Relacja $DA(a, b)$ zachodzi dokładnie wtedy, gdy b jest numerem gödłowskim wniosku reguły dołączania kwantyfikatora generalnego, gdzie przesłanka ma numer gödłowski a .

Kodowanie reguł wnioskowania

Relacja $OE(a, b)$ zachodzi, gdy istnieją $x < a$, $y < a$ oraz $z < b$ takie, że:

- $Form(x)$
- $Form(y)$
- $Vble(z)$
- $a = \langle 9, \langle 13, z, x \rangle \rangle$
- $b = \langle 9, x, y \rangle$.

Relacja $OE(a, b)$ zachodzi dokładnie wtedy, gdy b jest numerem gödłowskim wniosku reguły opuszczania kwantyfikatora egzystencjalnego, gdzie przesłanka ma numer gödłowski a .

Kodowanie reguł wnioskowania

Relacja $DE(a, b)$ zachodzi, gdy istnieją $x < a$, $y < a$ oraz $z < b$ takie, że:

- $Form(x)$
- $Form(y)$
- $Vble(z)$
- $a = \langle 9, x, y \rangle$
- $b = \langle 9, \langle 13, z, x \rangle \rangle$
- $\neg Fr(y, z)$.

Relacja $DE(a, b)$ zachodzi dokładnie wtedy, gdy b jest numerem gödłowskim wniosku reguły dołączania kwantyfikatora egzystencjalnego, gdzie przesłanka ma numer gödłowski a .

Kodowanie dowodów

$Ax(a) \equiv \text{Log}Ax(a) \vee \text{Eq}Ax(a) \vee \text{NL}Ax(a)$. Relacja ta zachodzi dokładnie dla liczb będących numerami gödłowskimi aksjomatów PA.

Dowody są *ciągami* formuł. Nadto, jak pamiętamy, każdy dowód jest też *drzewem*: w korzeniu znajduje się dowodzona formuła, w liściach aksjomaty, a bezpośrednie poprzedniki każdego wierzchołka są przesłankami reguły wnioskowania, której wnioskiem jest właśnie ów wierzchołek.

Ciągi numerów formuł możemy kodować na tej samej zasadzie, na jakiej kodujemy ciągi symboli. Jeśli $\Delta = (\psi_1, \psi_2, \dots, \psi_n)$ jest ciągiem formuł, to przez *numer gödłowski* ciągu Δ rozumiemy (wyznaczoną jednoznacznie i efektywnie) liczbę:

$$\langle \ulcorner \psi_1 \urcorner, \ulcorner \psi_2 \urcorner, \dots, \ulcorner \psi_n \urcorner \rangle.$$

Kodowanie dowodów

Relację $\text{Dow}(a, b)$ definiujemy przez koniunkcję warunków:

$$\text{seq}(a) \wedge \text{lh}(a) \neq 0 \wedge (a)_{\text{lh}(a) \div 1} = b \wedge \forall i < \text{lh}(a) (\text{Form}((a)_i) \wedge \exists j < i \exists k < i (\text{sub}((a)_j, (a)_i) \vee \text{MP}((a)_j, (a)_k, (a)_i) \vee \text{GR}((a)_j, (a)_i) \vee \text{OA}((a)_j, (a)_i) \vee \text{DA}((a)_j, (a)_i) \vee \text{OE}((a)_j, (a)_i) \vee \text{DE}((a)_j, (a)_i))).$$

Relacja $\text{Dow}(a, b)$ zachodzi dokładnie wtedy, gdy a jest numerem gödłowskim dowodu formuły o numerze gödłowskim b .

Jak bowiem pamiętamy, dowód formuły ψ to ciąg formuł o ostatnim elemencie ψ taki, że każdy z elementów tego ciągu jest bądź aksjomatem, bądź wnioskiem którejś z reguł wnioskowania, której przesłankami są wcześniejsze (od tego wniosku) elementy owego ciągu.

Rekurencyjność kodowań

Twierdzenie.

Wszystkie zdefiniowane powyżej funkcje i relacje są rekurencyjne.

- Dowód tego twierdzenia wynika bezpośrednio z podanych wyraźnych rekurencyjnych definicji.
- Odwołujemy się przy tym także do faktów wspomnianych wyżej, przy kodowaniu termów.

Twierdzenia PA

Wszystkie dotąd rozważane funkcje i relacje odpowiadające pojęciom metalogicznym okazały się rekurencyjne. A co z relacją $PA \vdash \psi$? Gdy zachodzi $PA \vdash \psi$, to ψ jest twierdzeniem PA. Okazuje się, że relacja ta (dokładniej: relacja między numerami dowodów a numerami formuł) nie jest rekurencyjna, jest jedynie rekurencyjnie przeliczalna. Mamy bowiem: $PA \vdash \psi$ dokładnie wtedy, gdy *istnieje* dowód ψ w PA. W definicji pojęcia „być twierdzeniem PA” występuje zatem jeden nieograniczony kwantyfikator egzystencjalny.

- Definiujemy: $\text{Tw}(a) \equiv \exists x \text{Dow}(x, a)$.
- Relacja Tw jest rekurencyjnie przeliczalna.

Klasyfikacja złożoności pojęć

Jak wiemy, operacje używające kwantyfikatorów ograniczonych prowadzą od relacji rekurencyjnych do relacji rekurencyjnych. Kwantyfikatory nieograniczone już nie mają tej własności – istotnie zwiększają stopień skomplikowania pojęć. Można dokonać logicznej klasyfikacji pojęć uwzględniającej liczbę kwantyfikatorów nieograniczonych potrzebnych w ich definicjach.

Szczególnie istotne są dwie hierarchie, nazywane:

- **hierarchią arytmetyczną** (kwantyfikujemy tylko zmienne indywidualne);
- **hierarchią analityczną** (kwantyfikujemy zmienne przebiegające podzbiory uniwersum).

Hierarchia arytmetyczna

Definicja Hierarchii Arytmetycznej.

- $\Sigma_0^0 = \Pi_0^0 =$ zbiór relacji rekurencyjnych;
 - Relacja $R \subseteq \omega^k$ jest klasy Σ_{n+1}^0 wtedy i tylko wtedy, gdy istnieje relacja $Q \subseteq \omega^{k+1}$ klasy Π_n^0 taka, że $R(a_1, \dots, a_k) \equiv \exists x Q(a_1, \dots, a_k, x)$.
 - Relacja $R \subseteq \omega^k$ jest klasy Π_{n+1}^0 wtedy i tylko wtedy, gdy istnieje relacja $Q \subseteq \omega^{k+1}$ klasy Σ_n^0 taka, że $R(a_1, \dots, a_k) \equiv \forall x Q(a_1, \dots, a_k, x)$.
-
- Relacje klasy Σ_1^0 to dokładnie relacje rekurencyjnie przeliczalne.
 - Relacja R jest rekurencyjna wtedy i tylko wtedy, gdy R oraz $\neg R$ są rekurencyjnie przeliczalne.

Hierarchia arytmetyczna

- Jeżeli relacja R jest klasy Σ_n^0 (odpowiednio, Π_n^0) zaś f_1, \dots, f_k są funkcjami rekurencyjnymi, to relacja P określona wzorem:

$$P(\vec{a}) \equiv R(f_1(\vec{a}), \dots, f_k(\vec{a}))$$

jest również klasy Σ_n^0 (odpowiednio, Π_n^0).

- Każda klasa hierarchii arytmetycznej jest zamknięta ze względu na koniunkcję i alternatywę.

Tu (i dalej) \vec{a} oznacza ciąg argumentów o takiej długości, ile argumentów ma rozważana relacja lub funkcja.

Dla dowolnego zbioru X relacji przez zbiór **uzupełnień** relacji z X rozumiemy zbiór $\mathcal{C}X$ zdefiniowany następująco: $R \in \mathcal{C}X$ wtedy i tylko wtedy, gdy $\forall \vec{a} (R(\vec{a}) \equiv \neg P(\vec{a}))$ dla pewnej relacji $P \in X$.

Hierarchia arytmetyczna

- Klasa Σ_n^0 jest identyczna z klasą uzupełnień relacji z klasy Π_n^0 i *vice versa*.
- Operacja kwantyfikatora ogólnego nie wyprowadza poza klasę Π_n^0 (dla $n > 0$).
- Operacja kwantyfikatora egzystencjalnego nie wyprowadza poza klasę Σ_n^0 (dla $n > 0$).

Prawdziwe są następujące (właściwe) inkluzje:

- $\Pi_n^0 \subset \Sigma_{n+1}^0$,
- $\Sigma_n^0 \subset \Pi_{n+1}^0$,
- $\Pi_n^0 \subset \Pi_{n+1}^0$,
- $\Sigma_n^0 \subset \Sigma_{n+1}^0$.

Hierarchia arytmetyczna

- Dla każdej klasy \sum_n^0 (odpowiednio, \prod_n^0) ($n > 0$) istnieje w \sum_n^0 (odpowiednio, \prod_n^0) relacja uniwersalna dla wszystkich relacji tej klasy.
 - Dla każdego $n > 0$: $\prod_n^0 \neq \sum_n^0$.
 - Dla każdego n : $\sum_n^0 \neq \sum_{n+1}^0$ oraz $\prod_n^0 \neq \prod_{n+1}^0$.
-
- Dla $n > 0$ relacja uniwersalna dla klasy \sum_n^0 należy do \sum_n^0 , ale nie należy ani do \prod_n^0 ani do \sum_{n-1}^0 .
 - Dla $n > 0$ relacja uniwersalna dla klasy \prod_n^0 należy do \prod_n^0 , ale nie należy ani do \sum_n^0 ani do \prod_{n-1}^0 .
 - Jeżeli relacja uniwersalna dla relacji klasy X sama należy do X , to $CX \neq X$.

Hierarchia arytmetyczna

Przykład. Pojęcie **granicy ciągu** jest pojęciem klasy Π_3^0 (i nie jest pojęciem ani klasy Σ_3^0 ani Σ_2^0 ani Π_2^0):

$$a = \lim a_n \equiv \forall k \exists m \forall n (n > m \rightarrow |a_n - a| < \frac{1}{k+1}).$$

Przykład. Jak widzieliśmy, zbiór twierdzeń Arytmetyki Peana jest klasy Σ_1^0 , czyli jest rekurencyjnie przeliczalny (ale **nie** jest rekurencyjny!).

Przykład. Pojęcie **prawdy** nie może zostać scharakteryzowane na żadnym piętrze hierarchii arytmetycznej.

Przykład. Również definicja pojęcia **dobrego porządku** wykracza poza hierarchię arytmetyczną.

Inne funkcje kodujące

- Zamiast funkcji β Gödla można używać innych funkcji rekurencyjnych w arytmetyzacji składni. Przy tym, stosowane kodowanie może uwzględniać budowę składniową wyrażeń, bądź kodować po prostu ciągi symboli.
 - W wielu podręcznikach używa się kodowania wykorzystującego rozkład liczb na czynniki pierwsze.
-
- Dla wyrażenia (termu lub formuły) u o postaci: $vv_1v_2 \dots v_n$ jego numer gödłowski $gn(u)$ określamy indukcyjnie:

$$gn(u) = 2^{gn(v)} \cdot 3^{gn(v_1)} \cdot 5^{gn(v_2)} \cdot \dots \cdot p_n^{gn(v_n)}.$$
 - Tu p_n jest n -tą liczbą pierwszą ($p_0 = 2$, $p_1 = 3$, $p_2 = 5$, itd.)

Inne funkcje kodujące

Dla rozważanej wcześniej formuły $\forall x_1 (x_1 \doteq \underline{0} \rightarrow (x_1 \times \underline{s}(x_1)) \doteq \underline{0})$ mamy (przy ustalonej uprzednio funkcji gn):

- $gn(\underline{s}(x_1)) = 2^{17} \cdot 3^2$
- $gn(x_1 \times \underline{s}(x_1)) = 2^{21} \cdot 3^2 \cdot 5^{2^{17} \cdot 3^2}$
- $gn(x_1 \times \underline{s}(x_1)) \doteq \underline{0} = 2^{25} \cdot 3^{2^{21} \cdot 3^2 \cdot 5^{2^{17} \cdot 3^2}} \cdot 5^{23}$
- $gn(x_1 \doteq \underline{0}) = 2^{25} \cdot 3^2 \cdot 5^{23}$
- $gn(x_1 \doteq \underline{0} \rightarrow (x_1 \times \underline{s}(x_1)) \doteq \underline{0}) = 2^9 \cdot 3^{gn(x_1 \doteq \underline{0})} \cdot 5^{gn(x_1 \times \underline{s}(x_1)) \doteq \underline{0}}$
- $gn(\forall x_1 (x_1 \doteq \underline{0} \rightarrow (x_1 \times \underline{s}(x_1)) \doteq \underline{0})) = 2^{15} \cdot 3^2 \cdot 5^{gn(x_1 \doteq \underline{0} \rightarrow (x_1 \times \underline{s}(x_1)) \doteq \underline{0})}$.

Kodowanie jest jednoznaczne. Dla dowolnej liczby a można ustalić czy jest ona kodem jakiegoś wyrażenia.

Inne funkcje kodujące

- Można też kodować wyrażenia po prostu jako ciągi symboli. Niech funkcja σ numerująca symbole alfabetu przyjmuje wartości dodatnie. Jeśli $s_1 s_2 \dots s_n$ jest ciągiem symboli alfabetu, to za kod tego ciągu można wziąć liczbę $2^{\sigma(s_1)} \cdot 3^{\sigma(s_2)} \cdot \dots \cdot p_{n-1}^{\sigma(s_n)}$.
- Można kodować wyrażenia ustalając np., że symbole alfabetu kodujemy liczbami zaczynającymi się (w zapisie dziesiętnym) od cyfry 8, po której występuje pewna liczba cyfr 1 (inna dla każdego symbolu alfabetu).
- Można kodować wyrażenia używając funkcji Cantora lub jakiegokolwiek innej (rekurencyjnej) funkcji kodującej ciągi skończone.
- Wybór funkcji kodującej jest więc sprawą konwencji. Kodowanie musi być jedynie jednoznaczne i rekurencyjne. W rozważaniach metateoretycznych nigdy nie obliczamy wartości funkcji kodujących wyrażenia.

Wykorzystywana literatura

- Cori, R., Lascar, D. 2001. *Mathematical Logic. A Course with Exercises*. Oxford University Press, Oxford.
- Hinman, P.G. 2005. *Fundamentals of Mathematical Logic*. A K Peters, Wellesley.
- Kleene, S.C. 1952. *Introduction to Metamathematics*. Wolters-Noordhoff Publishing — Groningen, North-Holland Publishing Company — Amsterdam Oxford, American-Elsevier Publishing Company, Inc. — New York.
- Ławrow, I.A., Maksimowa, L.L. 2004. *Zadania z teorii mnogości, logiki matematycznej i teorii algorytmów*. Wydawnictwo Naukowe PWN, Warszawa.
- Murawski, R. 2000³. *Funkcje rekurencyjne i elementy metamatematyki. Problemy zupełności, rozstrzygalności, twierdzenia Gödla*. Wydawnictwo Naukowe UAM, Poznań.