Semantyka gier

Tomasz Maciejewski

IIUWr

21 stycznia 2009

Definicja semantyki gier

Semantyka gier to podejście do semantyki denotacyjnej, w której model obliczeń przedstawiony jest jako interakcja między systemem a środowiskiem. Program jest modelowany jako zbiór możliwych interakcji.

Zastosowanie

Gry to bardzo bogate uniwersum, które może modelować różne języki programowania:

- języki czysto funkcyjne
- ze zmienną pamięcią (ang. *mutable store*)
- z operatorami kontrolnymi (ang. *control operators*)
- z pamięcią wyższego rzędu (wskaźnikami)
- z niedeterminizmem

Historia

- Paul Lorenzen po raz pierwszy zaprezentował semantykę gier dla logiki w późnych **latach 50**. Jej rozwojem zajął się później Kuno Lorenz. Od tamtego czasu powstało wiele różnych semantyk gier.
- Około roku **1995** semantyka gier przeżyła pewien renesans za sprawą prac Johana van Benthema na temat analogii między logiką a teorią gier. Również wymyślona przez Jean-Yves Girarda logika liniowa przyczyniła się do wzrostu zainteresowania tematem gier.
- Wielu autorów prac, m. in.: **Samson Abramsky**, **Guy McCusker**, J. van Benthem, A. Blass, D. Gabbay, M. Hyland, W. Hodges, R. Jagadessan, G. Japaridze, E. Krabbe, L. Ong, H. Prakken, G. Sandu D. Walton, J. Woods.

Historia

- Paul Lorenzen po raz pierwszy zaprezentował semantykę gier dla logiki w późnych **latach 50**. Jej rozwojem zajął się później Kuno Lorenz. Od tamtego czasu powstało wiele różnych semantyk gier.
- Około roku **1995** semantyka gier przeżyła pewien renesans za sprawą prac Johana van Benthema na temat analogii między logiką a teorią gier. Również wymyślona przez Jean-Yves Girarda logika liniowa przyczyniła się do wzrostu zainteresowania tematem gier.
- Wielu autorów prac, m. in.: **Samson Abramsky**, **Guy McCusker**, J. van Benthem, A. Blass, D. Gabbay, M. Hyland, W. Hodges, R. Jagadessan, G. Japaridze, E. Krabbe, L. Ong, H. Prakken, G. Sandu D. Walton, J. Woods.

Historia

- Paul Lorenzen po raz pierwszy zaprezentował semantykę gier dla logiki w późnych **latach 50**. Jej rozwojem zajął się później Kuno Lorenz. Od tamtego czasu powstało wiele różnych semantyk gier.
- Około roku **1995** semantyka gier przeżyła pewien renesans za sprawą prac Johana van Benthema na temat analogii między logiką a teorią gier. Również wymyślona przez Jean-Yves Girarda logika liniowa przyczyniła się do wzrostu zainteresowania tematem gier.
- Wielu autorów prac, m. in.: **Samson Abramsky**, **Guy McCusker**, J. van Benthem, A. Blass, D. Gabbay, M. Hyland, W. Hodges, R. Jagadessan, G. Japaridze, E. Krabbe, L. Ong, H. Prakken, G. Sandu D. Walton, J. Woods.

Logika Lorenzena

Lorenzen przedstawił model dowodu twierdzenia jako dialog między entuzjastą (ang. *verifier*) V a sceptykiem (ang. *falsifier*) F.

Przykład

$$\forall x \in \mathbb{N}, \exists y \in \mathbb{N} \quad y > x \land y \text{ jest liczbą pierwszą}$$

- F próbuje obalić twierdzenie, wybierając jakieś x, które wg niego obala twierdzenie.
- V odpowiada, znajdując y, które spełnia $y > x \land y$ jest liczbą pierwszą.
- F wybiera jedną stronę koniunkcji, która wg niego wydaje się fałszywa.
- itd...

Logika Lorenzena

Lorenzen przedstawił model dowodu twierdzenia jako dialog między entuzjastą (ang. *verifier*) V a sceptykiem (ang. *falsifier*) F.

Przykład

$$\forall x \in \mathbb{N}, \exists y \in \mathbb{N} \quad y > x \land y \text{ jest liczbą pierwszą}$$

- F próbuje obalić twierdzenie, wybierając jakieś x, które wg niego obala twierdzenie.
- V odpowiada, znajdując y, które spełnia $y > x \land y$ jest liczbą pierwszą.
- F wybiera jedną stronę koniunkcji, która wg niego wydaje się fałszywa.
- itd...

Logika Lorenzena

Lorenzen przedstawił model dowodu twierdzenia jako dialog między entuzjastą (ang. *verifier*) V a sceptykiem (ang. *falsifier*) F.

Przykład

$$\forall x \in \mathbb{N}, \exists y \in \mathbb{N} \quad y > x \wedge y \text{ jest liczbą pierwszą}$$

- F próbuje obalić twierdzenie, wybierając jakieś x, które wg niego obala twierdzenie.
- V odpowiada, znajdując y, które spełnia $y > x \wedge y$ jest liczbą pierwszą.
- F wybiera jedną stronę koniunkcji, która wg niego wydaje się fałszywa.
- itd...

Logika Lorenzena

Lorenzen przedstawił model dowodu twierdzenia jako dialog między entuzjastą (ang. *verifier*) V a sceptykiem (ang. *falsifier*) F.

Przykład

$$\forall x \in \mathbb{N}, \exists y \in \mathbb{N} \quad y > x \land y \text{ jest liczbą pierwszą}$$

- F próbuje obalić twierdzenie, wybierając jakieś x, które wg niego obala twierdzenie.
- V odpowiada, znajdując y, które spełnia $y > x \land y$ jest liczbą pierwszą.
- F wybiera jedną stronę koniunkcji, która wg niego wydaje się fałszywa.
- itd...

Logika Lorenzena

Lorenzen przedstawił model dowodu twierdzenia jako dialog między entuzjastą (ang. *verifier*) V a sceptykiem (ang. *falsifier*) F.

Przykład

$$\forall x \in \mathbb{N}, \exists y \in \mathbb{N} \quad y > x \wedge y \text{ jest liczbą pierwszą}$$

- F próbuje obalić twierdzenie, wybierając jakieś x, które wg niego obala twierdzenie.
- V odpowiada, znajdując y, które spełnia $y > x \wedge y$ jest liczbą pierwszą.
- F wybiera jedną stronę koniunkcji, która wg niego wydaje się fałszywa.
- itd...

Logika Lorenzena

Lorenzen przedstawił model dowodu twierdzenia jako dialog między entuzjastą (ang. *verifier*) V a sceptykiem (ang. *falsifier*) F.

Przykład

$$\forall x \in \mathbb{N}, \exists y \in \mathbb{N} \quad y > x \land y \text{ jest liczbą pierwszą}$$

- F próbuje obalić twierdzenie, wybierając jakieś x, które wg niego obala twierdzenie.
- V odpowiada, znajdując y, które spełnia $y > x \land y$ jest liczbą pierwszą.
- F wybiera jedną stronę koniunkcji, która wg niego wydaje się fałszywa.
- itd...

Przykład interakcji – funkcja

Obliczenie to interakcja między graczem (P) a adwersarzem (O). Rolę gracza może pełnić program lub system, a adwersarza – środowisko, kontekst.

Przykład: funkcja pierwszego rzędu

Interakcja może wyglądać tak:

O: Jaki jest wynik funkcji?
P: Jaki jest argument funkcji?
O: Argument to 3.
P: Wynik to 4

Przykład interakcji – funkcja

Obliczenie to interakcja między graczem (P) a adwersarzem (O). Rolę gracza może pełnić program lub system, a adwersarza – środowisko, kontekst.

Przykład: funkcja pierwszego rzędu

Interakcja może wyglądać tak:

O: Jaki jest wynik funkcji?

P: Jaki jest argument funkcji?

O: Argument to 3.

P: Wynik to 4

Przykład interakcji – funkcja

Obliczenie to interakcja między graczem (P) a adwersarzem (O). Rolę gracza może pełnić program lub system, a adwersarza – środowisko, kontekst.

Przykład: funkcja pierwszego rzędu

Interakcja może wyglądać tak:

O: Jaki jest wynik funkcji?

P: Jaki jest argument funkcji?

O: Argument to 3.

P: Wynik to 4

Przykład interakcji – funkcja

Obliczenie to interakcja między graczem (P) a adwersarzem (O). Rolę gracza może pełnić program lub system, a adwersarza – środowisko, kontekst.

Przykład: funkcja pierwszego rzędu

Interakcja może wyglądać tak:

O: Jaki jest wynik funkcji?

P: Jaki jest argument funkcji?

O: Argument to 3.

P: Wynik to 4

Przykład interakcji – funkcja

Obliczenie to interakcja między graczem (P) a adwersarzem (O). Rolę gracza może pełnić program lub system, a adwersarza – środowisko, kontekst.

Przykład: funkcja pierwszego rzędu

Interakcja może wyglądać tak:

O: Jaki jest wynik funkcji?

P: Jaki jest argument funkcji?

O: Argument to 3.

P: Wynik to 4

Przykład interakcji – funkcja

Obliczenie to interakcja między graczem (P) a adwersarzem (O). Rolę gracza może pełnić program lub system, a adwersarza – środowisko, kontekst.

Przykład: funkcja pierwszego rzędu

Interakcja może wyglądać tak:

O: Jaki jest wynik funkcji?

P: Jaki jest argument funkcji?

O: Argument to 3.

P: Wynik to 4

Przykład interakcji – program

$$f : \mathbb{N} \to \mathbb{N} \vdash f(3) : \mathbb{N}$$

Interakcja może wyglądać tak:

O: Jaki jest wynik programu?

P: Jaka jest wartość funkcji?

O: Jaki jest argument funkcji?

P: Argumentem jest 3.

O: Wartością jest 4.

P: Wynikiem programu jest 4.

Przykład interakcji – program

$$f : \mathbb{N} \to \mathbb{N} \vdash f(3) : \mathbb{N}$$

Interakcja może wyglądać tak:

O: Jaki jest wynik programu?

P: Jaka jest wartość funkcji?

O: Jaki jest argument funkcji?

P: Argumentem jest 3.

O: Wartością jest 4.

P: Wynikiem programu jest 4.

Przykład interakcji – program

$$f : \mathbb{N} \to \mathbb{N} \vdash f(3) : \mathbb{N}$$

Interakcja może wyglądać tak:

O: Jaki jest wynik programu?

P: Jaka jest wartość funkcji?

O: Jaki jest argument funkcji?

P: Argumentem jest 3.

O: Wartością jest 4.

P: Wynikiem programu jest 4.

Przykład interakcji – program

$$f : \mathbb{N} \to \mathbb{N} \vdash f(3) : \mathbb{N}$$

Interakcja może wyglądać tak:

O: Jaki jest wynik programu?

P: Jaka jest wartość funkcji?

O: Jaki jest argument funkcji?

P: Argumentem jest 3.

O: Wartością jest 4.

P: Wynikiem programu jest 4.

Przykład interakcji – program

$$f : \mathbb{N} \to \mathbb{N} \vdash f(3) : \mathbb{N}$$

Interakcja może wyglądać tak:

O: Jaki jest wynik programu?

P: Jaka jest wartość funkcji?

O: Jaki jest argument funkcji?

P: Argumentem jest 3.

O: Wartością jest 4.

P: Wynikiem programu jest 4.

Przykład interakcji – program

$$f : \mathbb{N} \to \mathbb{N} \vdash f(3) : \mathbb{N}$$

Interakcja może wyglądać tak:

O: Jaki jest wynik programu?

P: Jaka jest wartość funkcji?

O: Jaki jest argument funkcji?

P: Argumentem jest 3.

O: Wartością jest 4.

P: Wynikiem programu jest 4.

Skrócona notacja

$$
\begin{array}{cccccc}
f: & \mathbb{N} & \rightarrow & \mathbb{N} & \vdash f(3): & \mathbb{N} \\
 & & & & & \mathsf{q} \\
 & & & \mathsf{q} & & \\
 & \mathsf{q} & & & & \\
 & 3 & & & & \\
 & & & 4 & & \\
 & & & & & 4
\end{array}
$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \to \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$
\begin{aligned}
M_G &= \{a_1, a_2, b_1, b_2, b_3\} \\
\lambda_G &: \quad a_1, a_2 \mapsto O, \quad b_1, b_2, b_3 \mapsto P \\
P_G &= \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\}
\end{aligned}
$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \to \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$\begin{aligned} M_G &= \{a_1, a_2, b_1, b_2, b_3\} \\ \lambda_G &: \quad a_1, a_2 \mapsto O, \qquad b_1, b_2, b_3 \mapsto P \\ P_G &= \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\} \end{aligned}$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \rightarrow \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$
\begin{array}{rcl}
M_G & = & \{a_1, a_2, b_1, b_2, b_3\} \\
\lambda_G & : & a_1, a_2 \mapsto O, \quad b_1, b_2, b_3 \mapsto P \\
P_G & = & \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\}
\end{array}
$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \to \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$
\begin{array}{rcl}
M_G & = & \{a_1, a_2, b_1, b_2, b_3\} \\
\lambda_G & : & a_1, a_2 \mapsto O, \quad b_1, b_2, b_3 \mapsto P \\
P_G & = & \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\}
\end{array}
$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \rightarrow \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$
\begin{aligned}
M_G &= \{a_1, a_2, b_1, b_2, b_3\} \\
\lambda_G &: a_1, a_2 \mapsto O, \quad b_1, b_2, b_3 \mapsto P \\
P_G &= \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\}
\end{aligned}
$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \to \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$\begin{aligned} M_G &= \{a_1, a_2, b_1, b_2, b_3\} \\ \lambda_G &: a_1, a_2 \mapsto O, \quad b_1, b_2, b_3 \mapsto P \\ P_G &= \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\} \end{aligned}$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \to \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$\begin{aligned} M_G &= \{a_1, a_2, b_1, b_2, b_3\} \\ \lambda_G &: \quad a_1, a_2 \mapsto O, \qquad b_1, b_2, b_3 \mapsto P \\ P_G &= \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\} \end{aligned}$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \to \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$\begin{aligned} M_G &= \{a_1, a_2, b_1, b_2, b_3\} \\ \lambda_G &: \quad a_1, a_2 \mapsto O, \quad b_1, b_2, b_3 \mapsto P \\ P_G &= \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\} \end{aligned}$$

Gra – bardziej formalnie

Gra G to trójka (M_G, λ_G, P_G), gdzie:

M_G to zbiór wszystkich możliwych ruchów,

λ_G to funkcja typu $M_G \rightarrow \{O, P\}$,

$P_G \subseteq M_G^\star$ taki, że:

- jest niepusty,
- jest zbiorem ciągów zawierających na przemian ruchy adwersarza i gracza (zaczynając od adwersarza),
- jest zamknięty ze względu na prefiksy.

Przykład

$$
\begin{array}{rcl}
M_G & = & \{a_1, a_2, b_1, b_2, b_3\} \\
\lambda_G & : & a_1, a_2 \mapsto O, \quad b_1, b_2, b_3 \mapsto P \\
P_G & = & \{\epsilon, a_1, a_1 b_1, a_2, a_2 b_2, a_2 b_3\}
\end{array}
$$

Strategia

Formalnie, deterministyczna strategia σ dla gry G, to taki niepusty podzbiór P_G, że:

- $\forall s \in \sigma \quad |s|$ jest parzysta,
- $\epsilon \in \sigma$,
- $\forall s \in P_G, \forall a, b \in M_G \quad sab \in \sigma \implies s \in \sigma$,
- $\forall s \in \sigma, \forall a, b, c \in M_G \quad sab \in \sigma \wedge sac \in \sigma \implies b = c$.

Strategia

Formalnie, deterministyczna strategia σ dla gry G, to taki niepusty podzbiór P_G, że:

- $\forall s \in \sigma \quad |s|$ jest parzysta,
- $\epsilon \in \sigma$,
- $\forall s \in P_G, \forall a, b \in M_G \quad sab \in \sigma \implies s \in \sigma$,
- $\forall s \in \sigma, \forall a, b, c \in M_G \quad sab \in \sigma \wedge sac \in \sigma \implies b = c$.

Strategia

Formalnie, deterministyczna strategia σ dla gry G, to taki niepusty podzbiór P_G, że:

- $\forall s \in \sigma \quad |s|$ jest parzysta,
- $\epsilon \in \sigma$,
- $\forall s \in P_G, \forall a, b \in M_G \quad sab \in \sigma \implies s \in \sigma$,
- $\forall s \in \sigma, \forall a, b, c \in M_G \quad sab \in \sigma \wedge sac \in \sigma \implies b = c$.

Strategia

Formalnie, deterministyczna strategia σ dla gry G, to taki niepusty podzbiór P_G, że:

- $\forall s \in \sigma \quad |s|$ jest parzysta,
- $\epsilon \in \sigma$,
- $\forall s \in P_G, \forall a, b \in M_G \quad sab \in \sigma \implies s \in \sigma$,
- $\forall s \in \sigma, \forall a, b, c \in M_G \quad sab \in \sigma \land sac \in \sigma \implies b = c$.

Gry dla podstawowych typów

Przeciwnik pyta o liczbę, a gracz może mu odpowiedzieć dowolną liczbą.

$$
\begin{aligned}
M_{\mathbb{N}} &= \{q, 0, 1, 2, \ldots\} \\
\lambda_{\mathbb{N}}(q) &= O \\
\lambda_{\mathbb{N}}(n) &= P \quad \text{dla } n \in \{0, 1, 2, \ldots\}
\end{aligned}
$$

Jedyne poprawne rozgrywki to te postaci $q \cdot n$.

Gry dla krotek

Dla gier A i B możemy zdefiniować grę $A \times B$:

$$\begin{aligned}
M_{A\times B} &= M_A + M_B \\
\lambda_{A\times B}(x) &= \begin{cases} \lambda_A(x) & \text{dla } x \in M_A \\ \lambda_B(x) & \text{dla } x \in M_B \end{cases} \\
P_{A\times B} &= \{s \in M_{A\times B} \mid s \restriction M_A \in P_A \wedge s \restriction M_B \in P_B\}
\end{aligned}$$

Dwie rozgrywki równoległe. Tylko przeciwnik może zmieniać grę.

Gry dla funkcji

Dla gier A i B możemy zdefiniować grę $A \Rightarrow B$:

$$
\begin{aligned}
M_{A \Rightarrow B} &= M_A + M_B \\
\lambda_{A \Rightarrow B}(x) &= \begin{cases} \overline{\lambda_A}(x) & \text{dla } x \in M_A \\ \lambda_B(x) & \text{dla } x \in M_B \end{cases} \\
P_{A \Rightarrow B} &= \{ s \in M_{A \times B} \mid s \restriction M_A \in \overline{P_A} \wedge s \restriction M_B \in P_B \}
\end{aligned}
$$

Dwie rozgrywki przeciwległe: w jednej jesteśmy graczem a w drugiej – przeciwnikiem. Tylko gracz może zmieniać grę.

Strategia odgapiania

Strategia odgapiania (ang. *copy-cat strategy*) dla gry $A \Rightarrow A$.

Ruch	1.	2.	Gracz
1		a_1	O
2	a_1		P
3	a_2		O
4		a_2	P
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Ta strategia gwarantuje, że w jednej grze na pewno wygramy.

Interpretacja programu

Program

$$x : A, y : B \vdash P : C$$

można zinterpretować jako strategię gry:

$$A \times B \Rightarrow C$$

Interpretacja programu – przykład

Przykład

$$x : \mathbb{N}, y : \mathbb{N} \vdash x + y : \mathbb{N}$$

$$\begin{array}{ccccc}
\mathbb{N} & \times & \mathbb{N} & \Rightarrow & \mathbb{N} \\
 & & & & q \\
q & & & & \\
x & & & & \\
 & & q & & \\
 & & y & & \\
 & & & & x + y
\end{array}$$

Obie wersje są identyczne!

Interpretacja programu – przykład

Przykład

$$x : \mathbb{N}, y : \mathbb{N} \vdash x + y : \mathbb{N}$$

$$\begin{array}{ccccc}
\mathbb{N} & \Rightarrow & (\mathbb{N} & \Rightarrow & \mathbb{N}) \\
 & & & & q \\
q & & & & \\
x & & & & \\
 & & q & & \\
 & & y & & \\
 & & & & x + y
\end{array}$$

Obie wersje są identyczne!

Interpretacja programu – przykład

Przykład

$$x : \mathbb{N}, y : \mathbb{N} \vdash x + y : \mathbb{N}$$

$$\begin{array}{ccccc}
\mathbb{N} & \Rightarrow & (\mathbb{N} & \Rightarrow & \mathbb{N}) \\
 & & & & q \\
q & & & & \\
x & & & & \\
 & & q & & \\
 & & y & & \\
 & & & & x + y
\end{array}$$

Obie wersje są identyczne!

A dalej...

Bardziej zaawansowane tematy:

- złożenie z ukryciem: dwie strategie $\sigma : A \Rightarrow B, \tau : B \Rightarrow C$ możemy złożyć w jedną $\sigma;\tau : A \Rightarrow C$,
- istnienie strategii wygrywających: w każdej sytuacji jesteśmy w stanie zrobić ruch,
- polimorfizm.

Do poczytania

S. Abramsky: *Semantics of Interactions* (http://web.comlab.ox.ac.uk/people/Samson.Abramsky/gsem/)

G. McCusker: *Game Semantics: an Overview*

Koniec

Dziękuję za uwagę.