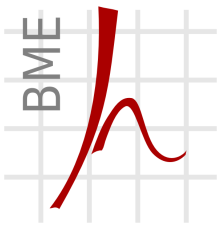
Programok értelmezése

Kód visszafejtés.



Híradástechnikai Tanszék

Izsó Tamás

2016. szeptember 22.

Section 1

Programok értelmezése

programok szemantika értelmezése

Formális szemantika segítségével precízen megadhatjuk, értelmezhetjük a program, hardware, stb viselkedését.

- felhasználhatjuk pecíz, egyértelmű, és mindenre kiterjedő dokumentumok készítéséhez.
- alapjául szolgálhat a programok, berendezések implementálásához, analíziséhez, a működés helyességének bizonyításához.

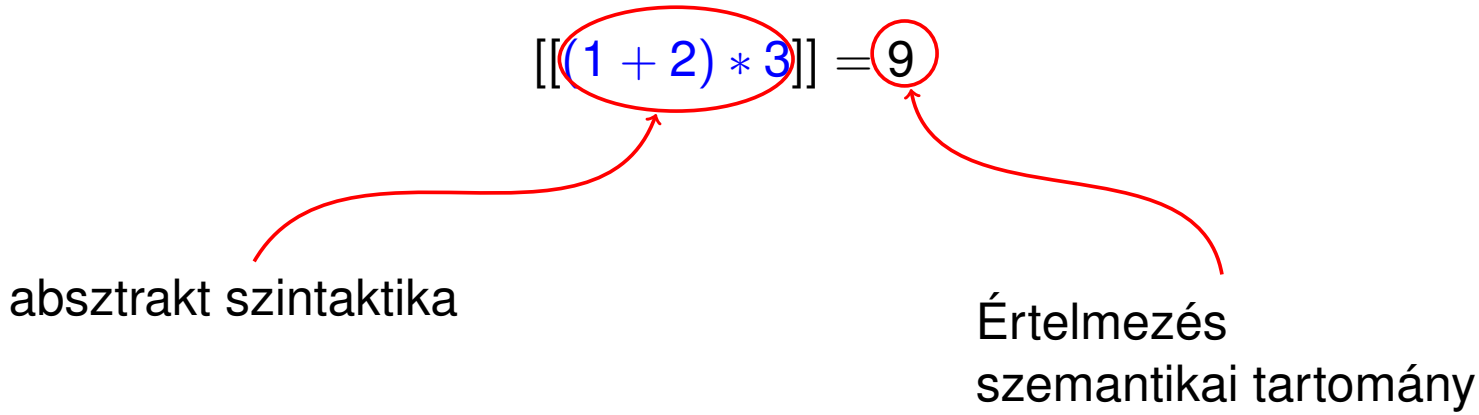
Szemantika és szintaktika fogalma:

- Szintaktika a programok nyelvi szerkezetével,
- szemantika a szintaktikailag korrekt program értelmezésével foglakozik.

Szemantikák

- *Operációs* (más néven műveleti) szemantika azzal foglalkozik, hogy az egyes utasítások hatására egy absztrakt gép belső állapota hogyan változik a program végrehajtása során.
- *Axiomatikus* szemantika az absztrakt gép állapotainak tulajdonságait vizsgálja a program végrehajtása során. A módszer a matematikai logikára támaszkodik, és a főbb felhasználási területe a programok helyességének a bizonyítása.
- *Denotációs* szemantika valamilyen szintaktikai szabály alapján helyesen leírt szöveg egy másik tartományon (ez általában a matematikai tartomány) történő értelmezésével foglalkozik.

Denotációs szemantika



$\mathcal{G}[\![.]\!], \mathcal{A}[\![.]\!], \mathcal{B}[\![.]\!], \mathcal{N}[\![.]\!], \ldots$ szemantikai függvények

Numerikus konstans kiértékelése

Szintaktikai tartomány

d: Digit – decimális számjegy karakter

n : Numeral – numerikus literál

Absztrakt származtatási szabályok

d ::= **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9**

n ::= *d* | *n d*

Szemantikai tartomány

Number = 0, 1, 2, 3, 4, . . . – természetes számok

Numerikus konstans kiértékelése

Szemantikai függvények

$\mathcal{N}$: Numeral $\rightarrow$ Number

Szemantikai egyenletek

$\mathcal{N}\ [\![\mathrm{n\ d}]\!] = 10 * \mathcal{N}[\![\mathrm{n}]\!] + \mathcal{N}[\![\mathrm{d}]\!]$

$\mathcal{N}[\![\mathbf{0}]\!] = 0\ \ \mathcal{N}[\![\mathbf{3}]\!] = 3\ \ \mathcal{N}[\![\mathbf{6}]\!] = 6\ \ \mathcal{N}[\![\mathbf{8}]\!] = 8$

$\mathcal{N}[\![\mathbf{1}]\!] = 1\ \ \mathcal{N}[\![\mathbf{4}]\!] = 4\ \ \mathcal{N}[\![\mathbf{7}]\!] = 7\ \ \mathcal{N}[\![\mathbf{9}]\!] = 9$

$\mathcal{N}[\![\mathbf{2}]\!] = 2\ \ \mathcal{N}[\![\mathbf{5}]\!] = 5$

Numerikus konstans kiértékelésre példa

$$
\begin{aligned}
\mathcal{N}[\![\mathbf{165}]\!] &= 10 * \mathcal{N}[\![\mathbf{16}]\!] + \mathcal{N}[\![\mathbf{5}]\!]) \\
&= 10 * \mathcal{N}[\![\mathbf{16}]\!] + 5 \\
&= 10 * (10 * \mathcal{N}[\![\mathbf{1}]\!] + \mathcal{N}[\![\mathbf{6}]\!]) + 5 \\
&= 10 * (10 * \mathcal{N}[\![\mathbf{1}]\!] + 6) + 5 \\
&= 10 * (10 * 1 + 6) + 5 \\
&= 10 * 16 + 5 \\
&= 160 + 5 \\
&= 165
\end{aligned}
$$

Mini imperatív While programozási nyelv

n : Numeral

x : Var – változók

a : Aexp – aritmetikai kifejezés

b : Bexp – logikai kifejezés

S : Stm – utasítás

Az egyes kategoriákba tartozó neveket indexeléssel különböztetjük meg, például n_1 , n_2 . . ..

While nyelv szintaxisa

$$
\begin{array}{lcl}
a & ::= & n \mid x \mid a_1 + a_2 \mid a_1 * a_2 \mid a_1 - a_2 \\
b & ::= & \textbf{true} \mid \textbf{false} \mid a_1 = a_2 \mid a_1 \leq a_2 \mid \neg b \mid b_1 \wedge b_2 \\
S & ::= & x := a \mid \textbf{skip} \mid S_1 \; ; \; S_2 \mid \textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2 \\
 & & \mid \textbf{while } b \textbf{ do } S
\end{array}
$$

Programállapot

Változó és memória (program állapot) megfeletetése.

Az *SV* a memóriában tárolható értékek halmaza:

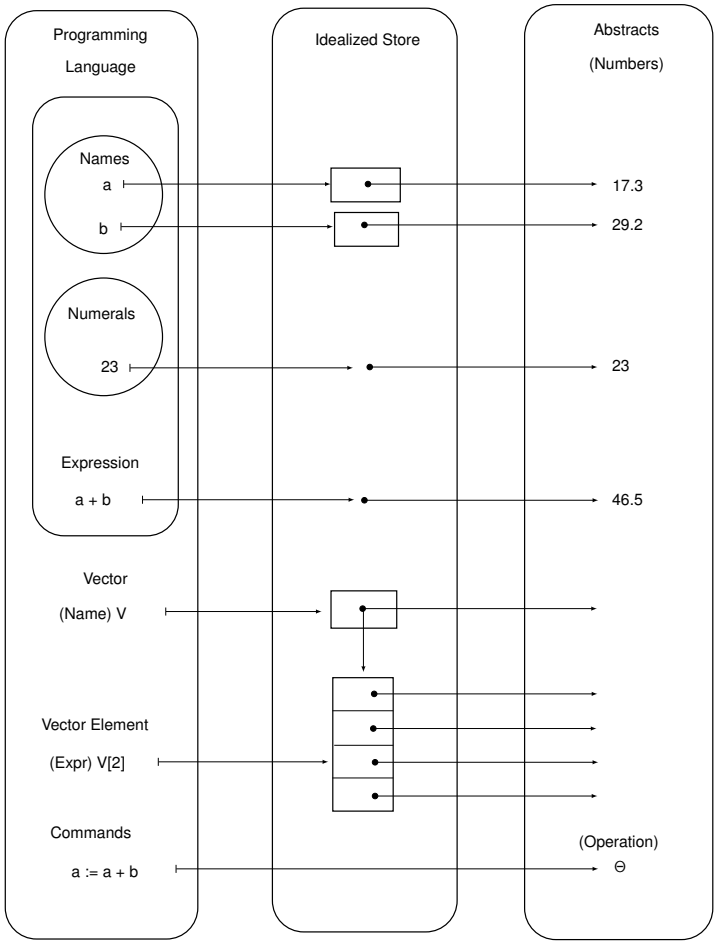
$$\mathrm{SV} = \mathrm{Number} \cup \mathrm{Boolean} \cup \{\mathrm{undefined}\}$$

Példa : s=$[x_1 \mapsto 5, x_2 \mapsto 9, x_3 \mapsto 110, x_{200} \mapsto \mathbf{tt}]$

s x_2 = 9

s x_8 = **undefined**

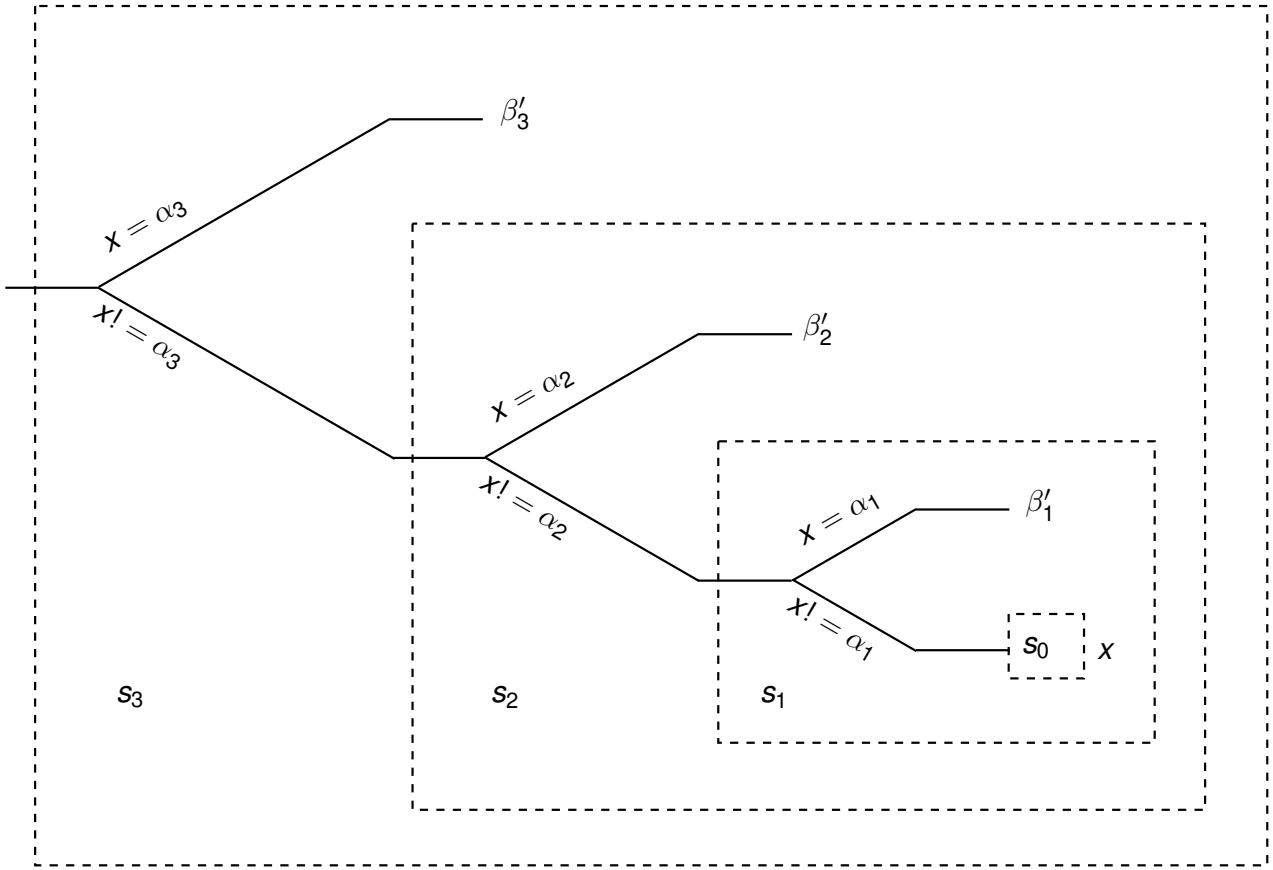
Absztrakt tártoló



Szemantikai függvények:

$Loc = Val$
$Env = Id \rightarrow \text{Loc}$
$Store = Loc \rightarrow Val$
$State = Store \times Env$

Programállapot kiértékelése



Aritmetikai kifejezések szemantikája

$$\mathcal{A} : \mathit{Aexp} \to (\mathit{State} \to \mathit{Number})$$

$$\begin{aligned}
\mathcal{A}[\![n]\!]s &= \mathcal{N}[\![n]\!] \\
\mathcal{A}[\![x]\!]s &= s\,x \\
\mathcal{A}[\![a_1 \mathbf{+} a_2]\!]s &= \mathcal{A}[\![a_1]\!]s + \mathcal{A}[\![a_2]\!]s \\
\mathcal{A}[\![a_1 \mathbf{-} a_2]\!]s &= \mathcal{A}[\![a_1]\!]s - \mathcal{A}[\![a_2]\!]s \\
\mathcal{A}[\![a_1 \mathbf{*} a_2]\!]s &= \mathcal{A}[\![a_1]\!]s * \mathcal{A}[\![a_2]\!]s
\end{aligned}$$

Példa: s=$[x_1 \mapsto 5, x_2 \mapsto 9, x_3 \mapsto 110, x_{200} \mapsto \mathbf{tt}]$

$$\begin{aligned}
\mathcal{A}[\![x_1 + 2]\!]s &= \mathcal{A}[\![x_1]\!]s + \mathcal{A}[\![2]\!]s \\
&= s\,x_1 + \mathcal{N}[\![2]\!] \\
&= 5 + 2 \;=\; 7
\end{aligned}$$

Logikai kifejezések szemantikája

$$\mathcal{B} : Bexp \to (State \to Boolean)$$

$$\mathcal{B}[\![\textbf{true}]\!]s = \textbf{tt}$$

$$\mathcal{B}[\![\textbf{false}]\!]s = \textbf{ff}$$

$$\mathcal{B}[\![a_1 = a_2]\!]s = \begin{cases} \textbf{tt} & \text{ha } \mathcal{A}[\![a_1]\!]s = \mathcal{A}[\![a_2]\!]s \\ \textbf{ff} & \text{ha } \mathcal{A}[\![a_1]\!]s \neq \mathcal{A}[\![a_2]\!]s \end{cases}$$

$$\mathcal{B}[\![a_1 \leq a_2]\!]s = \begin{cases} \textbf{tt} & \text{ha } \mathcal{A}[\![a_1]\!]s \leq \mathcal{A}[\![a_2]\!]s \\ \textbf{ff} & \text{ha } \mathcal{A}[\![a_1]\!]s > \mathcal{A}[\![a_2]\!]s \end{cases}$$

$$\mathcal{B}[\![\neg b]\!]s = \begin{cases} \textbf{tt} & \text{ha } \mathcal{B}[\![b]\!]s = \textbf{ff} \\ \textbf{ff} & \text{ha } \mathcal{B}[\![b]\!]s = \textbf{tt} \end{cases}$$

$$\mathcal{B}[\![b_1 \wedge b_2]\!]s = \begin{cases} \textbf{tt} & \text{ha } \mathcal{B}[\![b_1]\!]s = \textbf{tt} \text{ és } \mathcal{B}[\![b_2]\!]s = \textbf{tt} \\ \textbf{ff} & \text{ha } \mathcal{B}[\![b_1]\!]s = \textbf{ff} \text{ vagy } \mathcal{B}[\![b_2]\!]s = \textbf{ff} \end{cases}$$

Utasítások szemantikája

$$\mathcal{S} : Stm \to (State \to State)$$

$$\begin{aligned}
\mathcal{S}[\![x := a]\!]s &= s[x \mapsto \mathcal{A}[\![a]\!]s\,] \\
\mathcal{S}[\![\textbf{skip}]\!] &= \mathrm{id} \\
\mathcal{S}[\![S_1\ ;\ S_2]\!] &= \mathcal{S}[\![S_2]\!] \circ \mathcal{S}[\![S_1]\!] \\
\mathcal{S}[\![\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2]\!] &= cond(\mathcal{B}[\![b]\!], \mathcal{S}[\![S_1]\!], \mathcal{S}[\![S_2]\!]) \\
\mathcal{S}[\![\textbf{while } b \textbf{ do } S]\!]s &= \mathrm{Fix}F \\
\text{ahol } Fg &= cond(\mathcal{B}[\![b]\!], g \circ \mathcal{S}[\![S]\!], id)
\end{aligned}$$

Összetett utasítások szemantikája

$$
\begin{aligned}
\mathfrak{S}[\![S_1 \,;\, S_2]\!]s &= \\
&= \mathfrak{S}[\![S_2]\!] \circ \mathfrak{S}[\![S_1]\!] \\
&= \begin{cases}
s'' & \text{ha létezik az } s' \text{ állapot } \mathfrak{S}[\![S_1]\!]s = s' \\
 & \text{és } \mathfrak{S}[\![S_2]\!]s' = s'' \\
 & \\
\textbf{undef} & \text{ha } \mathfrak{S}[\![S_1]\!]s = \textbf{undef} \\
 & \text{vagy } s' \text{ létezik, de } \mathfrak{S}[\![S_2]\!]s' = \textbf{undef}
\end{cases}
\end{aligned}
$$

cond szemantikai függvény

$$cond : (State \to T) \times (State \to State) \times (State \to State) \to (State \to State)$$

$$\mathfrak{S}[\![\textbf{if } b \textbf{ then } S_1 \textbf{ else } S_2]\!]s = cond(\mathcal{B}[\![b]\!], \mathfrak{S}[\![S_1]\!], \mathfrak{S}[\![S_2]\!])s$$

$$= \begin{cases} s' & \text{ha } \mathcal{B}[\![b]\!]s = \textbf{tt} \text{ és } \mathfrak{S}[\![S_1]\!]s = s' \\ & \text{vagy } \mathcal{B}[\![b]\!]s = \textbf{ff} \text{ és } \mathfrak{S}[\![S_2]\!]s = s' \\ \textbf{undef} & \text{ha } \mathcal{B}[\![b]\!]s = \textbf{tt} \text{ és } \mathfrak{S}[\![S_1]\!]s = \textbf{undef} \\ & \text{vagy ha } \mathcal{B}[\![b]\!]s = \textbf{ff} \text{ és } \mathfrak{S}[\![S_2]\!]s = \textbf{undef} \end{cases}$$

while utasítás szemantikája

Operációs szemantika megadja, hogy egy adott állapotból melyik állapotba kerül a rendszer. Ez alapján mondhatnánk:

$$\mathfrak{S}[\![\textbf{while } b \textbf{ do } S]\!] = \mathfrak{S}[\![\textbf{if } b \textbf{ then}(\ S\ ;\textbf{while } b \textbf{ do } S)\ \textbf{else skip}]\!]$$

Denotációs szemantika esetén azonban ez nem adja meg helyesen a **while** utasítás jelentését. Ugyanakkor

$$\begin{aligned}
&\mathfrak{S}[\![\textbf{while } b \textbf{ do } S]\!] \\
&\quad = \mathfrak{S}[\![\textbf{if } b \textbf{ then}(\ S\ ;\textbf{while } b \textbf{ do } S)\ \textbf{else skip}]\!] \\
&\quad = \mathit{Cond}(\mathcal{B}[\![b]\!], \mathfrak{S}[\![\ S\ ;\textbf{while } b \textbf{ do } S]\!], \mathfrak{S}[\![\textbf{skip}]\!]) \\
&\quad = \mathit{Cond}(\mathcal{B}[\![b]\!], \mathfrak{S}[\![\textbf{while } b \textbf{ do } S]\!] \circ \mathfrak{S}[\![S]\!], \mathit{id}_\Sigma)
\end{aligned}$$

A következő jelölés bevezetésével, $g := \mathfrak{S}[\![\textbf{while } b \textbf{ do } S]\!]$ megkapjuk a $g = \mathit{Cond}(\mathcal{B}[\![b]\!], g \circ \mathfrak{S}[\![S]\!], \mathit{id}_\Sigma)$ rekurzív definícióval adott függvényt, amelynek a fix pontját kell meghatározni.

while utasítás szemantikája

$$\mathfrak{S}[\![\textbf{while } b \textbf{ do } S]\!] = \mathrm{Fix}\, F \; ahol \; F\, g = cond(\mathcal{B}[\![b]\!], g \circ \mathfrak{S}[\![S]\!], id)$$

$$\mathrm{Fix} : ((State \to State) \to (State \to State)) \to (State \to State)$$

Program:

while $\neg(x = 0)$ **do** $x := x - 1;$

Keressük azt a g függvényt, ami eleget tesz a következő feltételnek

$$(F\, g)s = \begin{cases} g(\mathfrak{S}[\![x := x - 1]\!]s) & \text{ha } s\,x \neq 0 \\ s & \text{ha } s\,x = 0 \end{cases}$$

Megoldás:

$$g(s) = \begin{cases} s[x \mapsto 0] & \text{ha } s\,x \geq 0 \\ \textbf{undefined} & \text{ha } s\,x < 0 \end{cases}$$

Mi az hogy fixpont

Matematikai Illusztráció!

$$F(x,y) \Leftarrow \text{ if } x = y \text{ then } \quad y+1$$
$$\text{else } F(x, F(x-1, y+1))$$

$f_1(x,y)$: if $x = y$ then $y+1$ else $x+1$

$f_2(x,y)$: if $x \geq y$ then $x+1$ else $y-1$

$f_3(x,y)$: if $x \geq y \wedge (x-y$ páros) then $x+1$ else undefined

Fixpont kiszámítása

$$g^0 \bot = \textbf{undefined}$$

$$g^1 \bot = \begin{cases} s[x \mapsto 0] & \text{ha } s\,x = 0 \\ g^0(\mathfrak{S}[\![x := x - 1]\!]s) & \text{ha } s\,x \neq 0 \end{cases}$$

$$= \begin{cases} s[x \mapsto 0] & \text{ha } s\,x = 0 \\ \textbf{undefined} & \text{ha } s\,x \neq 0 \end{cases}$$

$$g^2 \bot = \begin{cases} s[x \mapsto 0] & \text{ha } s\,x = 0 \\ g^1(\mathfrak{S}[\![x := x - 1]\!]s) & \text{ha } s\,x \neq 0 \end{cases}$$

$$= \begin{cases} s[x \mapsto 0] & \text{ha } s\,x = 0 \\ s[x \mapsto 0] & \text{ha } s\,x - 1 = 0 \\ \textbf{undefined} & \text{ha } s\,x < 0 \ \vee \ s\,x > 1 \end{cases}$$

Fixpont kiszámítása – folytatás

$$g^3\bot = \begin{cases} s[x \mapsto 0] & \text{ha } s\,x = 0 \\ g^2(\mathfrak{S}[\![x := x - 1]\!]s) & \text{ha } s\,x \neq 0 \end{cases}$$

$$= \begin{cases} s[x \mapsto 0] & \text{ha } s\,x = 0 \\ s[x \mapsto 0] & \text{ha } s\,x - 1 = 0 \\ s[x \mapsto 0] & \text{ha } s\,x - 2 = 0 \\ \textbf{undefined} & \text{ha } s\,x < 0 \ \vee \ s\,x > 2 \end{cases}$$

$$\vdots$$

$$\lim_{i\to\infty} g^i(s) = \begin{cases} s[x \mapsto 0] & \text{ha } s\,x \geq 0 \\ \textbf{undefined} & \text{ha } s\,x < 0 \end{cases}$$

2. példa (feladat)

Program:

$y := 1;\ \ \textbf{while}\ \ \neg(x = 1)\ \ \ \textbf{do}\ \ (y := y * x;\ \ x := x - 1);$

1. Az utasítások szemantikai értelmezése (16. dia) alapján írja fel a **while** szematikai értelmezését.
2. Számolja ki felírt függvény fixpontját.

2. példa (megoldás)

Program:

$\mathrm{y} := 1\,;\ \ \textbf{while}\ \ \neg(x = 1)\ \ \ \textbf{do}\ \ (\mathrm{y} := \mathrm{y} * \mathrm{x}\,;\ \ \mathrm{x} := \mathrm{x} - 1)\,;$

Keressük azt a g függvényt, ami eleget tesz a következő feltételnek

$$(F\,g)s = \begin{cases} g(\mathfrak{S}[\![y := y * x\,;\, x := x - 1]\!]s) & \text{ha } \mathcal{B}[\![\neg(x = 1)]\!]s = \textbf{tt} \\ s & \text{ha } \mathcal{B}[\![\neg(x = 1)]\!]s = \textbf{ff} \end{cases}$$

Vagy ezzel ekvivalens,

$$(F\,g)s = \begin{cases} g(s[y \to (s\,y) * (s\,x)][x \to (s\,x) - 1]) & \text{ha } s\,x \neq 1 \\ s & \text{ha } s\,x = 1 \end{cases}$$

2. példa Fixpont kiszámítása – folytatás I

$$g^0 \perp = \textbf{undefined}$$

$$g^1 \perp = \begin{cases} s & \text{ha } s\,x = 1 \\ \textbf{undefined} & \text{ha } s\,x \neq 1 \end{cases}$$

$$g^2 \perp = \begin{cases} s & \text{ha } s\,x = 1 \\ g^1(s[y \to (s\,y) * (s\,x)][x \to (s\,x) - 1]) & \text{ha } s\,x \neq 1 \end{cases}$$

$$= \begin{cases} s & \text{ha } s\,x = 1 \\ s[y \to (s\,y) * 2][x \to 1] & \text{ha } s\,x = 2 \\ \textbf{undefined} & \text{ha } s\,x \neq 1 \ \wedge \ s\,x \neq 2 \end{cases}$$

2. példa Fixpont kiszámítása – folytatás II

$$g^3\bot = \begin{cases} s & \text{ha } s\,x = 1 \\ g^2(s[y \to (s\,y) * (s\,x)][x \to (s\,x) - 1]) & \text{ha } s\,x \neq 1 \end{cases}$$

$$= \begin{cases} s & \text{ha } s\,x = 1 \\ s[y \to (s\,y) * 2][x \to 1] & \text{ha } s\,x = 2 \\ s[y \to (s\,y) * 2 * 3][x \to 1] & \text{ha } s\,x = 3 \\ \textbf{undefined} & \text{ha } s\,x \neq 1 \wedge s\,x \neq 2 \wedge \\ & \quad s\,x \neq 3 \end{cases}$$

$\ldots$

ahol

$$s = s_0\ [y \to 1]$$