Київський національний університет імені Тараса Шевченка
Факультет комп’ютерних наук та кібернетики

Розширення мови SIPL:
з випадковим вибором

Виконав:
магістр групи ШІ-1
Андрющенко Олексій

Київ - 2021

Синтаксис БНФ

Таблиця 1.

<table>
<tr><td>Ліва частина правила – метазмінна</td><td>Права частина правила</td><td>Ім'я правила</td></tr>
<tr><td><програма> ::=</td><td>begin <оператор> end</td><td>NP1</td></tr>
<tr><td><оператор> ::=</td><td><змінна> := <вираз> |
<оператор> ; <оператор> |
if <умова> then <оператор> else <оператор> |
while <умова> do <оператор> |
begin <оператор> end |
skip
Random (<оператор> | <оператор>)</td><td>NS1
NS2
NS3
NS4
NS5
NS6
NS7</td></tr>
<tr><td><вираз> ::=</td><td><число> |
<змінна> |
<вираз> + <вираз> |
<вираз> – <вираз> |
<вираз> * <вираз> |
(<вираз>)</td><td>NA1
NA2
NA3
NA4
NA5
NA6</td></tr>
<tr><td><умова> ::=</td><td><вираз> = <вираз> |
<вираз> > <вираз> |
<умова> ∨ <умова> |
<умова> ∧ <умова> |
¬ <умова> |
(<умова>)</td><td>NB1
NB2
NB3
NB4
NB5
NB6</td></tr>
<tr><td><змінна> ::=</td><td>. . . | M | N | . . .</td><td>NV...</td></tr>
<tr><td><число> ::=</td><td>. . . | –1 | 0 | 1 | . . .</td><td>NN...</td></tr>
</table>

Таблиця 2.

Метазмінна	Синтаксична категорія	Нова метазмінна
<программа>	Prog	P
<оператор>	Stm	S
<вираз>	Aexp	a
<умова>	Bexp	b
<змінна>	Var	x
<число>	Num	n

Синтаксис (метазмінні)

Таблиця 3.

<table>
<tr><td>Ліва частина правилі - метазмінна</td><td>Права частина правила</td><td>Ім’я правила</td></tr>
<tr><td>P ::=</td><td>begin S end</td><td>NP1</td></tr>
<tr><td>S ::=</td><td>x ::= n |
S1 ; S2 |
if b then S1 else S2 |
while b do S |
begin S end |
skip |
Random (S1| S2)</td><td>NS1
NS2
NS3
NS4
NS5
NS6
NS7</td></tr>
<tr><td>a ::=</td><td>n |
x |
a1 + a2 |
a1 - a2 |
a1 * a2 |
(a)</td><td>NA1
NA2
NA3
NA4
NA5
NA6</td></tr>
<tr><td>b ::=</td><td>a1 = a2 |
a1 > a2 |
b1 ∨ b2 |
b1 ∧ b2 |
¬ b |
(b)</td><td>NB1
NB2
NB3
NB4
NB5
NB6</td></tr>
<tr><td>x ::=</td><td>… | M | N | …</td><td>NV…</td></tr>
<tr><td>n ::=</td><td>… | -1 | 0 | 1 | …</td><td>NN…</td></tr>
</table>

Отримали багатоосновну алгебру даних мои $SIPL\_R$:

$$A\_Int\_Bool\_State = < Int, Bool, State; add, sub, mult, or, and, neg, eq, gr, \\ , \Rightarrow x, x \Rightarrow , \overline{n}, id, \nabla >$$

Композиційна семантика

Визначимо тепер класи функцій, які будуть задіяні при визначені семантики SIPL_R:

1. n-арні операції над базовими типами:

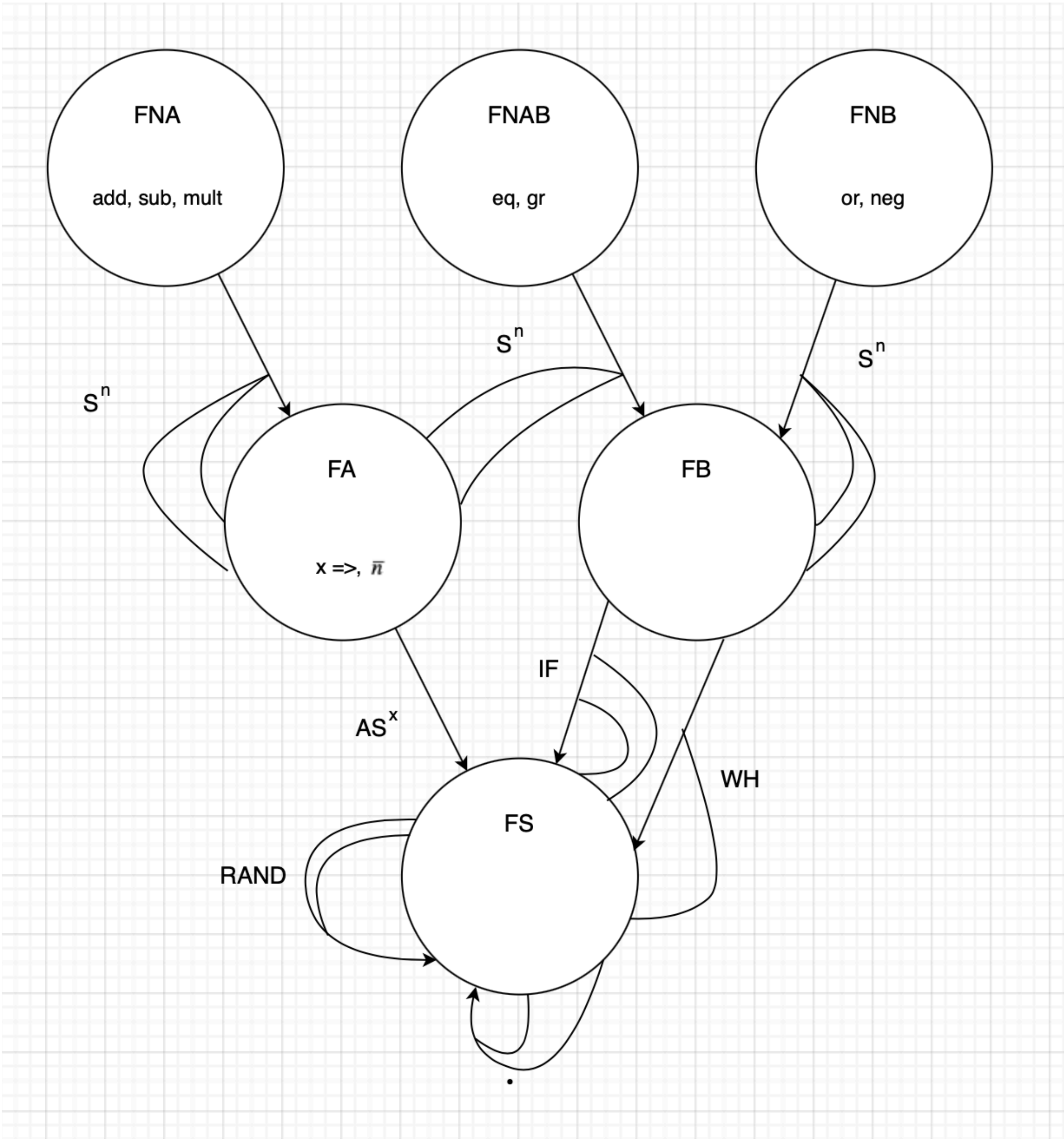
- $FNA = Int^n \to Int$ — n-арні арифметичні функції (операції);
- $FNB = Bool^n \to Bool$ — n-арні булеві функції (операції);
- $FNAB = Int^n -> Bool$ — n-арні функції (операції) порівняння над булевими типами даних;

2. Функції над станами змінних:

- $FA = State \to Int$ — номінативні арифметичні функції;
- $FB = State \to Bool$ — номінативні предикати;
- $FA = State \to State$ — біномативні функції-перетворювачі (трансформатори)

Отримали алгебру функцій (програмну алгебру):

$$A\_Prog\_R = < FNA, FNB, FNAB, FA, FB, FS; S^n, AS^x, \bullet, IF, WH, RAND, x \Rightarrow, id >$$



Формули для обчилення композицій і функцій алгебри A_Prog_R (fc — номінативна функція):

Таблиця 4.

Композиція	**Формула обчислення**	**Ім'я формули**
Суперпозиція	$(S^n(f, g_1, \ldots, g_n))(st) = f(g_1(st), \ldots, g_n(st))$	AF_S
Присвоювання	$AS^x(fa)(st) = st \nabla [x \mapsto fa(st)]$	AF_AS
Послідовне виконання	$fs_1 \bullet fs_2(st) = fs_2(fs_1(st))$	AF_SEQ
Випадковий вибір	$RAND(fs_1, fs_2) = \begin{cases} fs_1(st) \\ fs_2(st) \end{cases}$ — вибирається випадковим чином	AF_RAND
Умовний оператор	$IF(fb, fs_1, fs_2) = \begin{cases} fs_1(st), \text{якщо } fb(st) = true, \\ fs_2(st), \text{якщо } fb(st) = false\,. \end{cases}$	AF_IF
Цикл	$WH(fb, fs)(st) = st_n$, де $st_0 = st, st_1 = fs(st_0), st_2 = fs(st_1), st_n = fs(st_{n-1})$, причому $fb(st_0) = true, fb(st_1) = true, \ldots,$ $fb(st_{n-1}) = true, fb(st_n) = false$	AF_WH
Функція розіменування	$x \Rightarrow (st) = st(x)$	AF_DNM
Тотожна функція	$id(st) = st$	AF_ID

Програма мови $SIPL$ може бути перетворена на семантичний терм (терм програмної алгебри), який задає її семантику (семантичну функцію), перетвореннями такого типу:

- $sem\_P : Prog \to TFS$
- $sem\_S : Stm \to TFS$
- $sem\_A : Aexp \to TFA$
- $sem\_B : Bexp \to TFB$

де TFS, TFA, TFB, TF заддають відповідні множини термів.

Правила перетворення програми на семантичний терм:

Таблиця 5.

Правило заміни	Ім'я правила
$sem\_P : Prog \to TFS$ задається правилами:	
$sem\_P\ (begin\ S\ end) = sem\_S(S)$	NS_Prog
$sem\_S : Stm \to TFS$ задається правилами:	
$sem\_S\ (x ::= a) = AS^X(sem\_A(a))$	NS_Stm_AS
$sem\_S\ (S_1; S_2) = sem\_S(S_1) \bullet sem\_S(S_2)$	NS_Stm_Seq
$sem\_S\ (Random(S_1 \mid S_2)) = RAND(sem\_S(S_1), sem\_S(S_2))$	NS_Stm_Rand
$sem\_S\ (if\ b\ then\ S_1\ else\ S_2) =$ $= IF(sem\_B(b), sem\_S(S_1), sem\_S(S_2))$	NS_Stm_If
$sem\_S\ (while\ b\ do\ S) = WH(sem\_B(b), sem\_S(S))$	NS_Stm_Wh
$sem\_S\ (begin\ S\ end) = sem_S(S)$	NS_Stm_Be
$sem\_S\ (skip) = id$	NS_Stm_Skip
$sem\_S : Aexp \to TFA$ задається правилами:	
$sem\_A(n) = \overline{n}$	NS_A_Num
$sem\_A(x) = x \Rightarrow$	NS_A_Var
$sem\_A(a_1 + a_2) = S^2(add, sem\_A(a_1), sem\_A(a_2))$	NS_A_Add
$sem\_A(a_1 - a_2) = S^2(sub, sem\_A(a_1), sem\_A(a_2))$	NS_A_Sub

$sem\_A(a_1 * a_2) = S^2(mult, sem\_A(a_1), sem\_A(a_2))$	NS_A_Mult
$sem\_A((a)) = sem\_A(a)$	NS_A_Par
$sem\_B : Bexp \rightarrow TFB$ задається правилами:	
$sem\_B(a_1 = a_2) = S^2(eq, sem\_A(a_1), sem\_A(a_2))$	NS_B_eq
$sem\_B(a_1 > a_2) = S^2(gr, sem\_A(a_1), sem\_A(a_2))$	NS_B_gr
$sem\_B(b_1 \vee b_2) = S^2(or, sem\_B(b_1), sem\_B(b_2))$	NS_B_or
$sem\_B(\neg b) = S^1(neg, sem\_B(b))$	NS_B_neg
$sem\_B((b)) = sem\_B(b)$	NS_B_Par

Побудуємо семантичний терм програми COUNT_COINS(C, H, T):

(C — count, H — heads, T — tails)

```
sem_P(COUNT_COINS) = sem_P(
        begin
                H := 0
                T := 0
                N := 0
                while ¬N = C do
                        Random(
                                H := H + 1
                                |
                                T := T + 1
                        )
                        N := N + 1
        end
)
```

sem_S(H := 0;T := 0; N:= 0;while ¬N = C do Random(H := H + 1 | T := T + 1); n := N + 1;)

sem_S(H := 0) • sem_S(T := 0) • sem_S(N := 0) • sem_S(while ¬N = C do Random(H := H + 1 | T := T + 1); n := N + 1;)

$AS^H(0)$ • $AS^T(0)$ •$AS^N(0)$ • WH(sem_B(¬N = C), sem_S(Random(H := H + 1 | T := T + 1); n := N + 1;))

$AS^H(0)$ • $AS^T(0)$ •$AS^N(0)$ • WH(S^1(neg, sem_B(N = C)), sem_S(Random(H := H + 1 | T := T + 1)) • sem_S(n := N + 1)))

$AS^H(0)$ • $AS^T(0)$ •$AS^N(0)$ • WH(S^1(neg, S^2(eq, sem_A(N), sem_A(C)), RAND(sem_S(H := H + 1), sem_S(T := T + 1)) • sem_S(n := N + 1))

$AS^H(0)$ • $AS^T(0)$ •$AS^N(0)$ • WH(S^1(neg, S^2(eq, N⇒, C⇒), RAND(AS^H(sem_A(H + 1)), AS^T (sem_A(T + 1)) • AS^N(sem_A(N + 1)))

$AS^H(0)$ • $AS^T(0)$ •$AS^N(0)$ • WH(S^1(neg, S^2(eq, N⇒, C⇒), RAND(AS^H(S^2(add, sem_A(H), sem_A(1))), AS^T (S^2(add, sem_A(T), sem_A(1))) • AS^N(S^2(add, sem_A(N), sem_A(1))))

$AS^H(0)$ • $AS^T(0)$ •$AS^N(0)$ • WH(S^1(neg, S^2(eq, N⇒, C⇒), RAND(AS^H(S^2(add, H⇒, 1)), AS^T (S^2 (add, T⇒, 1)) • AS^N(S^2(add, N⇒, 1))).

Отже, результуючий терм має наступний вигляд:

$AS^H(0)$ •
$AS^T(0)$ •
$AS^N(0)$ •
WH(
    S^1(neg, S^2(eq, N⇒, C⇒),
    RAND(
        AS^H(S^2(add, H⇒, 1)),
        AS^T (S^2(add, T⇒, 1)
) •
    AS^N(S^2(add, N⇒, 1))
)

Операційна семантика

Таблиця 6.

Назва Правила	**Правило операційної семантики**
	Правила для програи та операторів
PR	$\dfrac{< S, st > \mapsto st'}{< begin\ S\ end, st > \mapsto st'}$
AS	$\dfrac{< a, st > \mapsto n}{< x := a, st > \mapsto st \triangledown [x \mapsto n]}$
SEQ	$\dfrac{< S_1, st > \mapsto st_1, \quad < S_2, st_1 > \mapsto st_2}{< S_1; S_2, st > \mapsto st_2}$
RAND1	$\dfrac{< S_1, st > \mapsto st_1,}{< Rand(S_1 \mid S_2), st > \mapsto st_1}$
RAND2	$\dfrac{< S_2, st > \mapsto st_2,}{< Rand(S_1 \mid S_2), st > \mapsto st_2}$
IFtrue	$\dfrac{< b, st > \mapsto true \quad < S_1, st > \mapsto st'}{< if\ b\ then\ S_1\ else\ S_2, st > \mapsto st'}$
IFfalse	$\dfrac{< b, st > \mapsto false \quad < S_2, st > \mapsto st'}{< if\ b\ then\ S_1\ else\ S_2, st > \mapsto st'}$
WHfalse	$\dfrac{< b, st > \mapsto false}{< while\ b\ do\ S, st > \mapsto st}$
WHtrue	$\dfrac{< b, st > \mapsto true \mid < S, st > \mapsto st'' \mid < while\ b\ do\ S, st'' > \mapsto st'}{< while\ b\ do\ S, st > \mapsto st}$
BEG	$\dfrac{< S, st > \mapsto st'}{< begin\ S\ end, st > \mapsto st'}$
skip	$< skip, st > \mapsto st$
	Правила для виразів
Num	$< n, st > \mapsto n$
Var	$< x, st > \mapsto st(x)$
A+	$\dfrac{< a_1, st > \mapsto n_1 \quad < a_2, st > \mapsto n_2}{< a_1 + a_2, st > \mapsto add(n_1, n_2)}$

A-	$\dfrac{< a_1, st > \mapsto n_1 \quad < a_2, st > \mapsto n_2}{< a_1 - a_2, st > \mapsto sub(n_1, n_2)}$
A*	$\dfrac{< a_1, st > \mapsto n_1 \quad < a_2, st > \mapsto n_2}{< a_1 * a_2, st > \mapsto mult(n_1, n_2)}$
A()	$\dfrac{< a, st > \mapsto n}{< (a), st > \mapsto n}$
Правила для умов	
B=	$\dfrac{< a_1, st > \mapsto r_1 \quad < a_2, st > \mapsto r_2}{< a_1 = a_2, st > \mapsto eq(r_1, r_2)}$
B>	$\dfrac{< a_1, st > \mapsto r_1 \quad < a_2, st > \mapsto r_2}{< a_1 > a_2, st > \mapsto gr(r_1, r_2)}$
B∨	$\dfrac{< a_1, st > \mapsto r_1 \quad < a_2, st > \mapsto r_2}{< a_1 \vee a_2, st > \mapsto or(r_1, r_2)}$
B∧	$\dfrac{< a_1, st > \mapsto r_1 \quad < a_2, st > \mapsto r_2}{< a_1 \wedge a_2, st > \mapsto and(r_1, r_2)}$

Операційну семантику для даного розширення мови SIPL позначатимемо $Sem\_P\_R_{OP}$.

Побудуємо в операційній семантиці дерево обчислення програми COUNT_COINS на стані $st = [H \mapsto 0, T \mapsto 0, N \mapsto 2]$.

(C — count, H — heads, T — tails)

```
sem_P(COUNT_COINS) = sem_P(
        begin
                N := 0
                while ¬N = C do
                        Random(
                                H := H + 1
                                |
                                T := T + 1
                        )
                        N := N + 1
        end
)
```

Нам потрібно побудувати виведення для формули $< COUNT\_COINS, st > \mapsto st'$

де стан $st = [H \mapsto 0, T \mapsto 0, N \mapsto 2]$ відомий, а st' — ні.

Позначимо:

$Body = N := 0; while \neg N = C\ do\ Random(H := H + 1 \,|\, T := T + 1); N := N + 1$

$Cond = \neg N = C$

$Wbody = Random(H := H + 1 \,|\, T := T + 1); N := N + 1$

$$\frac{< Body, st > \mapsto st'}{< begin\ Body\ end, st > \mapsto st'}$$

$$\frac{< N := 0 > \mapsto st_1, \quad < while\ Cond\ do\ Wbody, st_1 > \mapsto st'}{\frac{< Body, st > \mapsto st'}{< begin\ Body\ end, st > \mapsto st'}}$$

$st_1 = [H \mapsto 0, T \mapsto 0, N \mapsto 0, C \mapsto 0]$

Розглянемо $< while\ Cond\ do\ Wbody, st_1 > \mapsto st'$:

1) Умова $Cond : \neg N = C$ дає true: $\dfrac{< N = C, st_1 > \mapsto false}{\neg N = C, st_1 > \mapsto true}$

Виконаємо тіло:

$$\frac{< Random(H := H + 1 \,|\, T := T + 1), st_1 > \mapsto st_{1.1} \quad < N := N + 1, st_{1.1} > \mapsto st_2}{< Random(H := H + 1 \,|\, T := T + 1); N := N + 1, st_1 > \mapsto st_2}$$

Виберемо перший варіант:

$$\frac{< H := H + 1, st_1 > \mapsto st_{1.1}}{< Random(H := H + 1 \,|\, T := T + 1), st_1 > \mapsto st_{1.1}}$$

Тому:

$st_{1.1} = [H \to 1, T \to 0, N \to 2, C \to 0]$

$st_2 = [H \to 1, T \to 0, N \to 2, C \to 1]$

2) Умова $Cond : \neg N = C$ дає true: $\dfrac{< N = C, st_2 > \mapsto false}{\neg N = C, st_2 > \mapsto true}$

Виконаємо тіло:

$$\frac{< Random(H := H + 1 \,|\, T := T + 1), st_2 > \mapsto st_{2.1} \quad < N := N + 1, st_{2.1} > \mapsto st_3}{< Random(H := H + 1 \,|\, T := T + 1); N := N + 1, st_2 > \mapsto st_3}$$

Виберемо другий варіант:

$$\frac{< H := H + 1, st_2 > \mapsto st_{2.1}}{< Random(H := H + 1 \,|\, T := T + 1), st_2 > \mapsto st_3}$$

Тому:

$st_{2.1} = [H \to 1, T \to 1, N \to 2, C \to 1]$

$st_3 = [H \to 1, T \to 1, N \to 2, C \to 2]$

3) Умова $Cond : \neg N = C$ дає false: $\dfrac{< N = C, st_3 > \mapsto true}{\neg N = C, st_3 > \mapsto false}$

Тому, $st' = st_3 = [H \to 1, T \to 1, N \to 2, C \to 2]$.

Отже, в цій программі монету було підкинуто два рази, один раз випав орел, один раз решка.

(повне представлення задачі не було зроблене, оскільки воно виходило надто громіздким).

Доведення еквівалентності композиційної та операційної семантик

Теорема (про еківалентність композиційної та операційної семантик. Для довільної програми P мови $SIPL_R$ її композиційна семантика збігається з її операційною семантикою, тобто $sem\_P(P) = Sem\_P\_R_{OP}(P)$.

Доведення. Спочатку доведемо, що для довільного аріфметичного виразу a та довільної умови b маємо, що $sem\_A(a) = Sem\_P\_R_{OP}(a)$ та $sem\_A(b) = Sem\_P\_R_{OP}(b)$. Використовуємо індукцію за структурою a та b. Це твердження випливає з таблиці 5 та 6, які задають однакові значення для складових a та b. Далі доводимо, що $sem\_P(P) = Sem\_P\_R_{OP}(P)$ індукцією за структурою оператора S. Таблиці 5 та 6 задають однакові значення (крім циклу) для складових S. Що стосується циклу, то тут є два випадки:

1. Якщо тіло циклу не виконується жодного разу (умова хибна), то жодна функція не виконається і стани залишуться незмінними.

2. Якщо тіло цикло виконується, то треба застосувати індукцію. Якщо на кроці n стани програм є рівні, то на кроці n + 1 стани залишуться рівними, оскільки інші функції та оператори еквівалентні.

Отже, для довільної програми P мови $SIPL_R$ її композиційна семантика збігається з її операційною семантикою.

Аксіоматична семантика

Таблиця 7.

Правило виведення	**Позначення правила**
$\{P[x \mapsto a]\}\ x := a\ \{P\}$	AS
$\{P\}\ skip\ \{P\}$	skip
$\dfrac{\{P\}\ S_1\ \{Q\}\quad \{Q\}\ S_2\ \{R\}}{\{P\}\ S_1; S_2\ \{R\}}$	S
$\dfrac{\{P\}\ S_1\ \{R\} \mid \{P\}\ S_2\ \{R\}}{\{P\}\ Rand(S_1 \mid S_2)\ \{R\}}$	R
$\dfrac{\{b \wedge P\}\ S_1\ \{Q\} \mid \{\neg b \wedge P\}\ S_2\ \{Q\}}{\{P\}\ if\ b\ then\ S_1\ else\ S_2\ \{Q\}}$	IF
$\dfrac{\{b \wedge P\}\ S\ \{P\}}{\{P\}\ while\ b\ do\ S\ \{Q\}}$	WH
$\dfrac{\{P'\}S\{Q'\}}{\{P\}S\{Q\}}$, якщо $P \Rightarrow P', Q' \Rightarrow Q$	C
$\dfrac{\{P\}S\{Q\}}{\{P\}begin\ S\ end\{Q\}}$	BE

Доведемо в логіці Флойда-Хоара часткову коректність програми COUNT_COINS рахування кількості підкидань монет.

(C — count, H — heads, T — tails)

```
sem_P(COUNT_COINS) = sem_P(
        begin
                C := 0
                while ¬N = C do
                        Random(
                                H := H + 1
                                |
                                T := T + 1
                        )
                        C := C + 1
        end
)
```

Спочатку сформулюємо асерцію, яка задає коректнісь програми COUNT_COINS.

Передумовою є предикат $N \geq 0 \wedge N = n$, післялумовою $H + T = n$. Отже, треба побудувати виведення асерції:

$\{N \geq 0 \wedge N = n\}$ $COUNT\_COINS$ $\{H + T = n\}$

Доведемо, що предикат $H + T = C$ є інваріантом циклу.

Використаємо зворотний метод для поюудови передумов. Спочатку будуємо асерцію для оператора присвоювання $C := C + 1$:

$\{(H + T = C)[C \mapsto C + 1]\}C := C + 1\{H + T = C\}$.

Обчислюємо $(H + T = C)[C := C + 1]$. Отримуємо передумову $H + T = C + 1$.

Тепер розглянемо цю передумову як післяумову оператора рандома $RAND(H := H + 1 | T := T + 1)$. За правилом R виплиає:

$$\frac{\{H+T=C+1\}\ H:H+1\ \{H+T=C+1\}\ |\ \{H+T=C+1\}\ T:=T+1\ \{H+T=C+1\}}{\{H+T=C+1\}\ Rand(H:=H+1\,|\,T:=T+1)\ \{H+T=C+1\}}$$

Розглянемо, перший випадок:

$\{(H + T = C + 1)[H \mapsto H + 1]\}H := H + 1\{H + T = C + 1\}$.

Обчислюємо $(H + T = C + 1)[H := H + 1]$. Отримуємо передумову $H + 1 + T = C + 1$. Одержуємо предикат, який дорівнює $H + T = C$.

Розглянемо, другий випадок:

$\{(H + T = C + 1)T \mapsto T + 1]\}T := T + 1\{H + T = C + 1\}$.

Обчислюємо $(H + T = C + 1)[T := T + 1]$. Отримуємо передумову $H + T + 1 = C + 1$. Одержуємо предикат, який дорівнює $H + T = C$.

Застосувавши правило для оператора послідовного виконання, отримуємо асерцію

$$\{H+T=C\}RAND(H:=H+1\,|\,T:+T+1);C:=C+1\{H+T=C\}.$$

Щоб застосувати правило для циклу, посилимо передумову до $N>0\wedge H+T=C$ і за правилом наслідку виводимо таку асерцію:

$$\{N>0\wedge H+T=C\}RAND(H:=H+1\,|\,T:+T+1);C:=C+1\{H+T=C\}$$

Звідси випливає, що предикат $H+T=C$ дійсно є інваріантом циклу.

Застосувавши правило циклу маємо

$$\{H+T=C\}$$

$$while\,\neg N=C\ do\ RAND(H:H+1\,|\,T:+T+1);C:=C+1$$

$$\{\neg(\neg N=C)\wedge H+T=C\}$$

Ця формула еквівалентна такій формулі

$$N=C\wedge H+T=C\Rightarrow H+T=N.$$

Отже, $\{N\geq 0\wedge N=n\}\ COUNT\_COINS\ \{H+T=n\}$.