Przekazywanie parametrów przez nazwę

Przykład 13
W poniższym programie parametr przekazywany jest przez **nazwę**.

```
begin
        var x; var z;
        proc p(y); begin z:=x+y;   x:=y+1 end;
        x:=0; z:=0;
        begin
            var x;
            x:=1;                          z:=x+z+x; {z=1}
            call p(z+x); ⟶                 x:=z+x+1 {x=3}
        end;
        z:=z+x
    end
    {z=4}
```

Gdyby parametr był przekazywany przez zmienną, wywołanie **call p(z+x)** byłoby błędne.

Gdyby parametr był przekazywany przez wartość, wywołanie **call p(z+x)** przebiegałoby następująco:

```
begin
      var x; var z;
      proc p(y); begin z:=x+y;   x:=y+1 end;
      x:=0; z:=0;
      begin
          var x;
          x:=1;                          y:= z+x; {y=1}
          call p(z+x); ⟶                 z:= x+y; {z=1}
      end;                               x:= y+1; {x=2}
      z:=z+x
  end
  z=3
```

Formalizacja

Wprowadzamy pojęcie lewych i prawych wartości wyrażeń:

- $Lv = N + B + Loc$ – lewe wartości
- $Rv = N + B$ – prawe wartości

oraz domknięcia:

- $Cl = [S \to Lv + \{\bot_E\}]$

Definiujemy pozostałe dziedziny semantyczne:

- $Sv = N$
- $S = [Loc \to Sv + \{\ \underline{\text{wolna}}\ \}]$
- $Proc = [Cl \to [S \to S_\perp]]$
- $Dv = Loc + Proc + Cl + \{\ \underline{\text{puste}}\ \}$
- $Env = [Ide \to Dv]$

Dla skrócenia zapisów przyjmiemy

- $Exp = Aexp + Bexp$

Funkcje semantyczne

$$\mathcal{L} : Exp \to [Env \to [S \to Lv + \{\bot_E\}]]$$

$$\mathcal{R} : Exp \to [Env \to [S \to Rv + \{\bot_E\}]]$$

$$\mathcal{D} : Decl \to [Env \to [S \to Env \times S]]$$

$$\mathcal{C} : Com \to [Env \to [S \to S_\bot]]$$

Równania semantyczne

Exp

$$\mathcal{L}[[n]] = \lambda\rho \in Env.\lambda s \in S.n$$

$$\mathcal{R}[[n]] = \lambda\rho \in Env.\lambda s \in S.n$$

$$\mathcal{L}[[v]] = \lambda\rho \in Env.\lambda s \in S.\textbf{if } \rho v \in Loc \textbf{ then } \rho v \textbf{ else}$$
$$\quad \textbf{if } \rho v \in Cl \textbf{ then } (\rho v)s \textbf{ else } \perp_E$$

$$\mathcal{R}[[v]] = \lambda\rho \in Env.\lambda s \in S.\textbf{if } \rho v \in Loc \textbf{ then if } s(\rho v) \notin Rv$$
$$\textbf{then } \perp_E \textbf{ else } s(\rho v) \textbf{ else if } \rho v \in Cl \textbf{ then}$$
$$\textbf{if } (\rho v)s \in Rv \textbf{ then } (\rho v)s \textbf{ else if } (\rho v)s \in Loc \textbf{ then}$$
$$\textbf{if } s((\rho v)s) \in Rv \textbf{ then } s((\rho v)s) \textbf{ else } \perp_E \textbf{ else } \perp_E \textbf{ else } \perp_E$$

$$\mathcal{L}[[a_1 + a_2]] = \lambda\rho \in Env.\lambda s \in S.\textbf{if } \mathcal{R}[[a_1]]\rho s \notin N \textbf{ then } \perp_E \textbf{ else}$$
$$\textbf{if } \mathcal{R}[[a_2]]\rho s \notin N \textbf{ then } \perp_E \textbf{ else } \mathcal{R}[[a_1]]\rho s + \mathcal{R}[[a_2]]\rho s$$

Pozostałe wyrażenia definiujemy analogicznie.

Decl

$$\mathcal{D}[\![\ \underline{\text{proc}}\ P(v); c]\!] = \lambda\rho \in Env.\lambda s \in S.((\rho[f/P], s)$$

gdzie $f = \lambda\xi \in Cl.\lambda s' \in S.\mathcal{C}[\![c]\!](\rho[\xi/v])s'$

Com

$$\mathcal{C}[\![v := e]\!] = \lambda\rho \in Env.\lambda s \in S.\textbf{if } \mathcal{L}[\![v]\!]\rho s \notin Loc \textbf{ then } \perp_S$$
$$\textbf{else if } \mathcal{R}[\![e]\!]\rho s \notin Sv \textbf{ then } \perp_S \textbf{ else } s[\mathcal{R}[\![e]\!]\rho s/\mathcal{L}[\![v]\!]\rho s]$$

$$\mathcal{C}[\![\ \underline{\text{call}}\ P(e)]\!] = \lambda\rho \in Env.\lambda s \in S.\textbf{if } \rho P \notin Proc \textbf{ then } \perp_S$$
$$\textbf{else } (\rho P)(\mathcal{L}[\![e]\!]\rho)s.$$

Semantyka aksjomatyczna

Niech proc $P(\overline{v})$; c będzie deklaracją procedury, gdzie $\overline{v}$ jest listą parametrów formalnych.

8. **Reguła instrukcji wywołania procedury**

$$\frac{\{\alpha\}\ c\ \{\beta\}}{\forall \overline{v}(roz(\overline{v}) \Rightarrow \{\alpha\}P(\overline{v})\{\beta\})}$$

gdzie β nie może zawierać wolnego wystąpienia żadnego z parametrów z $\overline{v}$ przekazywanych przez wartość, a $roz(\overline{v})$ oznacza, że spełnione są następujące warunki:

(a) Wszystkie parametry aktualne z $\overline{v}$ przekazywane przez zmienną są różne i żaden z nich nie występuje wśród parametrów aktualnych przekazywanych przez wartość.

(b) Żadna z tych zmiennych nielokalnych w c, którym tam przypisuje się wartość, nie występuje w parametrach aktualnych z $\overline{v}$ przekazywanych przez wartość.

Dla konkretnego wywołania $P(\overline{a})$, w którym lista $\overline{a}$ spełnia powyższe warunki, regułę (8) stosujemy wraz z regułą

9.

$$\frac{roz(\overline{a}), \forall \overline{v}(roz(\overline{v}) \Rightarrow \{\alpha\}P(\overline{v})\{\beta\})}{\{\alpha[\overline{a}/\overline{v}]\}\ P(\overline{a})\ \{\beta[\overline{a}/\overline{v}]\}}$$

Przykład 14

Rozważmy procedurę:

proc $swap$(var x, y);
begin var z;
$z := x$; $x := y$; $y := z$
end

Ciało procedury c spełnia specyfikację

$$\{(x = X) \wedge (y = Y)\}\, c\, \{(x = Y) \wedge (y = X)\} \qquad (14)$$

X i Y są tu tzw. zmiennymi specyfikacyjnymi, oznaczającymi wartości początkowe zmiennych x i y. Muszą to być nowe identyfikatory, które nie występują w programie.

Rozważmy wywołanie $swap(a, b)$. Stosując reguły (8) i (9) otrzymujemy

$$\{((x = X) \wedge (y = Y))[a/x, b/y]\}$$

$$swap(a, b)$$

$$\{((x = Y) \wedge (y = X))[a/x, b/y]\}$$

co daje

$$\{(a = X) \wedge (b = Y)\}\ swap(a, b)\ \{(a = Y) \wedge (b = X)\}$$

Ale jeśli wywołanie $swap(a, b)$ wystąpi w sytuacji, gdy $a = 1$, $b = 2$, to chcielibyśmy móc wyprowadzić

$$\{(a = 1) \wedge (b = 2)\}\ swap(a, b)\ \{(a = 2) \wedge (b = 1)\}$$

Ponieważ X i Y reprezentują dowolne wartości początkowe, specyfikacja (14) jest równoważna

$$\forall X, Y.\{(x = X) \wedge (y = Y)\}\ c\ \{(x = Y) \wedge (y = X)\} \qquad (15)$$

Możemy teraz wziąć instancję (15) otrzymaną przez zastąpienie X przez 1 i Y przez 2.

Przykład 15
Pokażemy teraz, że warunek rozłączności parametrów aktualnych jest istotny. Rozważmy procedurę:

$$\begin{array}{ll} \underline{\text{proc}} & dwa(\ \underline{\text{var}}\ x, y); \\ & x := 1;\ \ y := 2 \end{array}$$

i jej wywołanie $dwa(a, a)$. Instrukcja c spełnia specyfikację

$$\{tt\}\ \ c\ \ \{(x = 1) \wedge (y = 2)\}$$

ale nie zachodzi

$$\{tt\}\ \ dwa(a, a)\ \ \{(a = 1) \wedge (a = 2)\}$$

co jest równoważne

$$\{tt\}\ \ dwa(a, a)\ \ \{ff\}$$