

## 講 座

## ALGOL N について

(II) 構文と *program* の静的構造 (つづき)

和田 英 一\*

## 2.6 Type

2.6.1 *primary* である, ある  $\langle \text{expression} \rangle$  を *type* とよぶ.

*type* を次のように再帰的に定義する.

- (1) **effect** は *type* である.
- (2) **real** は *type* である.
- (3) **bits** は *type* である.
- (4) **string** は *type* である.
- (5) **reference** は *type* である.
- (6)  $T$  を *type* とすると

**array** [ ]  $T$

は *type* であり, これを「**array-style**」の *type* とよぶ.

(7)  $n \geq 0$  の  $S_1, S_2, \dots, S_n$  を互いに異なる  $\langle \text{selector} \rangle$ ,  $T_1, T_2, \dots, T_n$  を *type* とすると

**structure** ( $S_1 T_1, \dots, S_n T_n$ )

は *type* であり, これを「**structure-style**」の *type* とよぶ.

(8)  $n \geq 0$  の  $T_1, T_2, \dots, T_n, T$  を *type* とすると

**procedure** ( $T_1, \dots, T_n$ )  $T$

は *type* であり, これを「**procedure-style**」の *type* とよぶ.

2.6.2 *type* のある集合に対して, 次のような表わし方がある.

$T$	すべての <i>type</i> の集合を表わす.
$T$ [ <b>array</b> ]	<b>array-style</b> のすべての <i>type</i> の集合を表わす.
$T$ [ <b>structure</b> ]	<b>structure-style</b> のすべての <i>type</i> の集合を表わす.
$T$ [ <b>procedure</b> ]	<b>procedure-style</b> のすべての <i>type</i> の集合を表わす.

## 2.7 Typed program

$P$  を *semi-legal program* とする.  $P$  のどの *direct constituent*  $\bar{E}$  にも *type* が割り付けられたとする.  $t(\bar{E})$  で  $\bar{E}$  に割り付けられた *type* を代表する. 次の条件 (1)~(9) がそこに述べられたどの *direct constituent*  $\bar{E}$  に対しても成り立ったとする.

このように *type* の割り付けられた *semi-legal program* を *typed program* とよぶ.

- (1)  $\bar{E}$  が  $\langle \text{go to statement} \rangle$  のとき,  $t(\bar{E})$  は **effect** である.
- (2)  $\bar{E}$  が  $\langle \text{effect notation} \rangle$  のとき,  $t(\bar{E})$  は **effect** である.
- (3)  $\bar{E}$  が  $\langle \text{real notation} \rangle$  のとき,  $t(\bar{E})$  は **real** である.
- (4)  $\bar{E}$  が  $\langle \text{bits notation} \rangle$  のとき,  $t(\bar{E})$  は **bits** である.
- (5)  $\bar{E}$  が  $\langle \text{string notation} \rangle$  のとき,  $t(\bar{E})$  は **string** である.
- (6)  $\bar{E}$  が  $\langle \text{reference notation} \rangle$  のとき,  $t(\bar{E})$  は **reference** である.
- (7)  $\bar{E}$  が  $\langle \text{array notation} \rangle$  のとき,  $t(\bar{E})$  は **array-style** である.
- (8)  $\bar{E}$  が  $\langle \text{structure notation} \rangle$  のとき,  $t(\bar{E})$  は **structure-style** である.
- (9)  $\bar{E}$  が  $\langle \text{procedure notation} \rangle$  のとき,  $t(\bar{E})$  は **procedure-style** である.

2.8 Variable declaration と *parameter-specification*

$\langle \text{variable declaration} \rangle$  と *parameter-specification* は *variable* に対し *type* と *quantity* を宣言する. ( $\rightarrow$ 3.1, 4.7)

2.8.1  $P$  を *typed program*,  $\bar{D}$  を  $P$  の *proper declaration* とする.

$D$  が  $\langle \text{variable declaration} \rangle$

“**let**  $\langle \text{identifier} \rangle V$  **be**  $\langle \text{expression} \rangle E$ ”

\* 東京大学工学部計数工学科

か  $D$  が *parameter-specification*

$\langle \bar{E}, \bar{V} \rangle$

(ここに  $E$  が  $\langle \text{expression} \rangle$ ,  $V$  が  $\langle \text{identifier} \rangle$ ) であるとき, *type*  $t(\bar{E})$  が *variable*  $V$  に対し  $\bar{D}$  「により宣言さ」れたという。

2.8.2 ある *variable* に対し *type* と *quantity* が 5. に記述した方法により「前以って」宣言されていることがある。どの *variable* に対しても多くともひとつの *type* と, 多くともひとつの *quantity* しか前以って宣言されていない。

2.8.3  $P$  を *typed program*,  $V$  を  $P$  の  $\langle \text{identifier} \rangle$  の形の *direct constituent* とする。  $V$  の *type* は次の場合 (1), (2) により宣言される。

場合 (1):  $P$  の *direct constituent* であって, 自分の *interior* に  $\bar{V}$  を含み, 自分の *proper interior* に *variable*  $V$  に対する  $P$  の *proper declaration* を含むような *assemblage* が存在する。この場合,  $\bar{E}$  をそのようなすべての *constituent assemblage* の間の包み方の順序で最小なものとする,  $\bar{V}$  の *type* は  $\bar{E}$  の *proper interior* における *proper declaration* により *variable*  $V$  に対して宣言された *type* である。

場合 (2): (1) におけるような *assemblage* はひとつもないが,  $V$  に対し *type* が前以って宣言されている。この場合,  $\bar{V}$  の *type* は  $V$  に対し前以って宣言された *type* である。

## 2.9 Formula declaration

2.9.1  $n \geq 0$  に対し

“( $T_1, \dots, T_n$ )”

の形の表現を,  $T_1, T_2, \dots, T_n$  が *type* であるとき, 「*type list*」とよぶ。

*frame*  $G$  と *type list*  $O$  との順序対  $\langle G, O \rangle$  を「*skeleton*」とよぶ。  $O$  が上にかかれた形であり,  $G$  が

“( )”

の形の *constituent* を正確に  $n$  個もつとき,  $\langle G, O \rangle$  を「*compatible skeleton*」とよび, さらに  $G$  と  $O$  とを互いに他に対して「*compatible*」であるという。

{上にのべたような  $G$  の *constituent* を  $G$  の「*argument place*」とよぶ。}

2.9.2  $P$  が *typed program*,  $\bar{D}$  が  $P$  の

“let  $\langle \text{frame} \rangle G$  represent  $\langle \text{expression} \rangle E$ ”

の形の *proper declaration* であるとする。

$t(\bar{E})$  が

“(*procedure* ( $T_1, \dots, T_n$ )  $T$ )”

(ここに  $T_1, T_2, \dots, T_n, T$  は *type*) の形であり, *skeleton*  $\langle G, (T_1, T_2, \dots, T_n) \rangle$  が *compatible* であるとき,  $\bar{D}$  は「*compatible*」であり, この *skeleton* に「対して *type*  $T$  を宣言する」という。このような  $\bar{D}$  を「*skeleton*  $\langle G, (T_1, T_2, \dots, T_n) \rangle$  に対する *declaration*」とよぶ。

2.9.3 ある *skeleton* に対し, *type* は 5. に記述した方法により「前以って」宣言されていることがある。どの *skeleton* に対しても多くともひとつの *type* しか前以って宣言されていない。

2.9.4  $P$  が *typed program*,  $\bar{E}$  が  $P$  の  $\langle \text{formula} \rangle$  の形の *direct constituent*,  $\bar{F}_1, \bar{F}_2, \dots, \bar{F}_n$  が  $\bar{E}$  のここにかいた順に現われる *immediate constituent* であるとする

“( $t(\bar{F}_1), \dots, t(\bar{F}_n)$ )”

の形の表現を  $\bar{E}$  の「*type-list*」とよぶ。  $G$  が  $\bar{E}$  の *frame*,  $\bar{O}$  が  $\bar{E}$  の *type-list* であるとき,  $\langle G, \bar{O} \rangle$  を  $\bar{E}$  「の *skeleton*」とよぶ。

2.9.5  $P$  が *typed program*,  $\bar{F}$  が  $P$  の  $\langle \text{formula} \rangle$  の形の *direct constituent* であるとする。  $\bar{F}$  の *type* は次の場合 (1), (2) により「宣言」される。

場合 (1):  $P$  が自分の *interior* に  $\bar{F}$  を含み, 自分の *proper interior* に  $\bar{F}$  の *skeleton* に対する *proper declaration* を含むような  $\langle \text{block} \rangle$  の形の *direct constituent* をもつ。この場合,  $\bar{E}$  を  $P$  のそのようなすべての *direct constituent* の間の包み方の順序で最小のものとする, 「 $\bar{F}$  の *type*」は  $\bar{F}$  の *skeleton* に対し  $\bar{E}$  の *proper interior* において *proper declaration*  $\bar{D}$  により宣言された *type* である。  $\bar{D}$  を「 $\bar{F}$  に対する *declaration*」とよぶ。

場合 (2):  $P$  が (1) におけるような *direct constituent* をひとつももたないが,  $\bar{F}$  の *skeleton* に対し *type*  $T$  が前以って宣言されている。この場合, 「 $\bar{F}$  の *type*」は *type*  $T$  である。

## 2.10 Labelling

2.10.1  $P$  が *typed program*,  $\bar{D}$  が  $P$  の *proper labelling*,  $D$  が

“ $L$  :”

の形であるとする, *label*  $L$  を  $D$  「により *label* さ」れたといい,  $\bar{D}$  を「 $L$  に対する *labelling*」とよぶ。

2.10.2  $P$  が *typed program*,  $F$  がその

“go to  $\langle \text{identifier} \rangle L$ ”

の形の *direct constituent* であるとする, 自分の

*interior* に  $\bar{F}$  を含み, 自分の *proper interior* に *label*  $L$  に対する *labelling* を含むような  $\langle \text{block} \rangle$  の形の *direct constituent* を  $P$  がもつときのみ,  $L$  を「*lable*」されたという。

{ $\bar{L}$  が *label* されており,  $\bar{E}$  が上の性質をもった最小の *direct constituent*,  $\bar{D}$  が  $\bar{E}$  の *proper interior* における  $L$  に対する *labelling* のひとつであるとするとき,  $\bar{D}$  を「 $\bar{L}$  に対する *labelling*」とよぶ。}

### 2.11 Legal program

*typed program*  $P$  がそのどの *direct constituent*  $E$  に対しても次の (L1)~(L37) の条件を満たしているときのみ,  $P$  を *legal program* とよぶ。

#### 2.11.1 $E$ が $\langle \text{identifier} \rangle$

“ $V$ ”

の場合,

(L1)  $\bar{V}$  の *type* が宣言されていて,

(L2)  $t(\bar{E})$  が  $\bar{V}$  の *type* である。

#### 2.11.2 $E$ が $\langle \text{go to statement} \rangle$

“*go to*  $\langle \text{identifier} \rangle L$ ”

の場合,

(L3)  $\bar{L}$  が *label* されている。

#### 2.11.3 $E$ が $\langle \text{block} \rangle$

“*begin*  $\langle \text{declaration} \rangle D_1$ ;

.....

$\langle \text{declaration} \rangle D_m$ ;

$\langle \text{identifier} \rangle L_1^1$ : .....  $\langle \text{identifier} \rangle L_{i_1}^1$ ;

$\langle \text{expression} \rangle E_1$ ;

.....

$\langle \text{identifier} \rangle L_1^n$ : .....  $\langle \text{identifier} \rangle L_{i_n}^n$ ;

$\langle \text{expression} \rangle E_n$  *end*”

(ここに  $m \geq 0, n \geq 1, i_1 \geq 0, i_2 \geq 0, \dots, i_n \geq 0$ ) の場合,

(L4)  $\bar{D}_j (1 \leq j \leq m)$  が *variable*  $V$  に対する *declaration* であるとき,  $\bar{D}_1, \bar{D}_2, \dots, \bar{D}_m$  の  $\bar{D}_j$  以外のものが  $V$  の *declaration* でなく,

(L5)  $\bar{D}_j (1 \leq j \leq m)$  が  $\langle \text{variable declaration} \rangle$  であるとき,  $\bar{D}_j$  が  $1 \leq k \leq j$  の  $\bar{D}_k$  のどの *explicit constituent* の *declaration* でなく,

(L6)  $\bar{D}_j (1 \leq j \leq m)$  が  $\langle \text{formula declaration} \rangle$  であるとき,  $\bar{D}_j$  が *compatible* であり,

(L7)  $\bar{D}_j (1 \leq j \leq m)$  が *skeleton*  $\langle G, O \rangle$  に対する *declaration* であるとき,  $\bar{D}_1, \bar{D}_2, \dots, \bar{D}_m$  の  $\bar{D}_j$  以外のものが  $\langle G, O \rangle$  に対する *declaration* でなく,

(L8)  $D_j (1 \leq j \leq m)$  が  $\langle \text{formula declaration} \rangle$  であ

るとき,  $\bar{D}_j$  が  $1 \leq k \leq j$  の  $\bar{D}_k$  の *explicit constituent* に対する *declaration* でなく,

(L9) *label*  $L_1^1, \dots, L_{i_1}^1, \dots, L_1^n, \dots, L_{i_n}^n$  が互いに異なり,

(L10)  $t(\bar{E})$  が  $t(\bar{E}_n)$  である。

#### 2.11.4 $E$ が $\langle \text{closed expression} \rangle$

“ $\langle \langle \text{expression} \rangle F \rangle$ ”

の場合,

(L11)  $t(\bar{E})$  が  $t(\bar{F})$  である。

#### 2.11.5 $E$ が $\langle \text{code} \rangle$

“*code* [ $\langle \text{selector} \rangle S_1 \langle \text{expression} \rangle E_1, \dots,$

$\langle \text{selector} \rangle S_n \langle \text{expression} \rangle E_n \langle \text{primary} \rangle T$

: $\langle \text{codebody} \rangle X$ ”

(ここに  $n \geq 0$ ) の場合,

(L12)  $S_1, S_2, \dots, S_n$  が互いに異なり,

(L13)  $t(\bar{E})$  が  $t(\bar{T})$  である。

#### 2.11.6 $E$ が

“*real* [ $\langle \text{expression} \rangle F \rangle \langle \text{real donor} \rangle J$ ”

の形の  $\langle \text{real notation} \rangle$  の場合,

(L14)  $t(\bar{F})$  が (**procedure (real) real**) である。

#### 2.11.7 $E$ が

“*bits* [ $\langle \text{expression} \rangle F \rangle \langle \text{bits donor} \rangle J$ ”

の形の  $\langle \text{bits notation} \rangle$  の場合,

(L15)  $t(\bar{F})$  が (**procedure (bits) bits**) である。

#### 2.11.8 $E$ が

“*string* [ $\langle \text{expression} \rangle F \rangle \langle \text{string donor} \rangle J$ ”

の形の  $\langle \text{string notation} \rangle$  の場合,

(L16)  $t(\bar{F})$  が (**procedure (string) string**) である。

#### 2.11.9 $E$ が $\langle \text{array notation} \rangle$

“*array* [ $\langle \text{array bound} \rangle Y \langle \text{expression} \rangle E_1, \dots,$   
 $\langle \text{expression} \rangle E_n$ ]

(ここに  $n \geq 1$ ) の場合,

(L17)  $Y$  が

“[ $\langle \text{expression} \rangle F_1$ :  $\langle \text{expression} \rangle F_2$ ]”

か

“[ $\langle \text{expression} \rangle F_1$ ::]”

か

“[ $\langle \text{expression} \rangle F_2$ ]”

の形であるとき,  $t(\bar{F}_1)$  と  $t(\bar{F}_2)$  が **real** であり,

(L18)  $t(\bar{E}_1), t(\bar{E}_2), \dots, t(\bar{E}_n)$  が同一であり,

(L19)  $t(\bar{E})$  が **array** [ ]  $t(\bar{E}_1)$  である。

#### 2.11.10 $E$ が $\langle \text{structure notation} \rangle$

“*structure* [ $\langle \text{selector} \rangle S_1 \langle \text{expression} \rangle E_1, \dots,$

⟨selector⟩  $S_n$  ⟨expression⟩  $E_n$ ”

(ここに  $n \geq 0$ ) の場合,

(L 20)  $S_1, S_2, \dots, S_n$  が互いに異なり,

(L 21)  $t(\bar{E})$  が **structure** ( $S_1 t(\bar{E}_1), \dots, S_n t(\bar{E}_n)$ )

である。

2.11.11  $E$  が ⟨procedure notation⟩

“**procedure** (⟨expression⟩  $T_1, \dots, \langle \text{expression} \rangle T_n$ )

⟨primary⟩  $T$  ⟨procedure donor⟩  $J$ ”

(ここに  $n \geq 0$ ) の場合,

(L 22)  $t(\bar{E})$  が (**procedure** ( $t(T_1), \dots, t(T_n)$ )  $t(T)$ )

である。

さらに  $J$  が

“: (⟨identifier⟩  $V_1, \dots, \langle \text{identifier} \rangle V_m$ )

⟨primary⟩  $F$ ”

(ここに  $m \geq 0$ ) の形であるとき,

(L 23)  $m=n$  であり,

(L 24)  $V_1, V_2, \dots, V_m$  が互いに異なり,

(L 25)  $t(\bar{F})$  が  $t(\bar{T})$  である。

2.11.12  $E$  が ⟨array element⟩

“⟨secondary⟩  $F$  [⟨expression⟩  $E'$ ]”

の場合,

(L 26)  $t(\bar{E}')$  が **real** であり,

(L 27)  $t(\bar{F})$  が  $T$  を *type* として “**array** [ ]  $T$ ”

の形であり,

(L 28)  $t(\bar{E})$  が  $T$  である。

2.11.13  $E$  が ⟨structure element⟩

“⟨secondary⟩  $F$  [⟨selector⟩  $S$ ]”

の場合,

(L 29)  $t(\bar{F})$  が

**structure** ( $S_1 T_1, \dots, S_n T_n$ )

(ここに  $n \geq 0$ ,  $S_1, S_2, \dots, S_n$  は ⟨selector⟩,  $T_1, T_2, \dots, T_n$  は *type*) の形であり,

(L 30)  $S$  が  $1 \leq i \leq n$  のある  $i$  に対して  $S_i$  であり,

(L 31)  $t(\bar{E})$  が  $T_i$  である。

2.11.14  $E$  が ⟨procedure call⟩

“⟨secondary⟩  $F$  (⟨expression⟩  $E_1, \dots, \langle \text{expression} \rangle E_n$ )”

(ここに  $n \geq 0$ ) の場合,

(L 32)  $t(\bar{F})$  が

“(**procedure** ( $T_1, \dots, T_m$ )  $T$ )”

(ここに  $m \geq 0$ ,  $T_1, T_2, \dots, T_m, T$  は *type*) の形であり,

(L 33)  $m=n$  であり,

(L 34)  $t(\bar{E}_1), \dots, t(\bar{E}_n)$  がそれぞれ  $T_1, \dots, T_n$  であり,

(L 35)  $t(\bar{E})$  が  $T$  である。

2.11.15  $E$  が ⟨formula⟩ の場合,

(L 36)  $\bar{E}$  の *type* が宣言されていて,

(L 37)  $t(\bar{E})$  が  $\bar{E}$  の *type* である。

(昭和46年7月2日受付)