Chapitre 8 : Contrôle de type

8.1 Les systèmes de typage

Exemple d'un contrôle de type simple

Syntaxe d'un langage rudimentaire :

P → *D* ; **begin** *I* **end**
D → *D* ; *D* | **id** : *T*
T → **char** | **int** | **boolean** | **array[nb] of** *T* | *-T*
| **function** *T* **to** *T*
E → **literal** | **nb** | *E* **=** *E* | *E***(***E***)** | *F*
F → **id** | *E***[***E***]** | *E-*
I → *F* **:=** *E* | **if** *E* **then** *I* **fi** | **while** *E* **do** *I* **od** | *I* ; *I*

- **Exemple de programme :**

clef : -int ; ok : boolean;
begin ok := clef - = 2000 ; if ok then clef- := 1 end

La grammaire est ambiguë. On peut utiliser des règles d'associativité. Le bloc **begin...end** permet de lever l'ambiguïté entre déclarations et instructions. La variable *F* désigne les expressions assignables.

- **Type des identificateurs :** Au moyen d'un attribut synthétisé *T.type.*

P → *D* ; **begin** *I* **end**
D → *D* ; *D*
D → **id** : *T* {*addtype*(**id**.*ent, T.type*)}
T → **char** {*T.type* := *char*}
T → **int** {*T.type* := *int*}
T → **boolean** {*T.type* := *boolean*}
T → **array[nb] of** *T* {*T.type* := *array*(*num.val*,*T*1.*type*)}
T → *-T* {*T.type* := *pointer*(*T*1.*type*)}
T → **function** *T* **to** *T* {*T.type* := *T*1.*type* → *T*2.*type*}

La procédure *addtype*(**id**.*ent, T.type*) ajoute l'information relative au type de l'identificateur dans la table des symboles.

- **Type des expressions :** Au moyen d'un attribut synthétisé.

E → **literal** {*E.type* := *char*}
E → **nb** {*E.type* := *int*}
E → *E* **=** *E* {*E.type* := **if** *E*1.*type* = *E*2.*type* <> *erreur*
then *boolean* **else** *erreur*}
E → *E***(***E***)** {*E.type* := **if** *E*2.*type* = *s* **and**

*E*1.*type* = *s* → *t* **then** *t* **else** *erreur*}
E → *F* {*E.type* := *F.type*}
F → **id** {*F.type* := *lookup*(**id**.*ent*)}
F → *E***[***E***]** {*F.type* := **if** *E*2.*type* = *int* **and**
*E*1.*type* = *array*(*s*, *t*) **then** *t* **else** *erreur*}
F → *E*- {*F.type* := **if** *E*1.*type* = *pointer*(*t*) **then** *t* **else** *erreur*}

La fonction *lookup*(**id**.*ent*) retourne le type de l'identificateur trouvé dans la table des symboles. On ne vérifie pas la validité de l'indice dans le tableau (contrôle dynamique).

- **Type des instructions :** au moyen d'un attribut synthétisé *I.type.*

I → *F* **:=** *E* {*I.type* := **if** *F.type* = *E.type*
then *void* **else** *erreur*}
I → **if** *E* **then** *I* **fi** {*I.type* := **if** *E.type* = *boolean*
then *I*1.*type* **else** *erreur*}
I → **while** *E* **do** *I* **od** {*I.type* := **if** *E.type* = *boolean*
then *I*1.*type* **else** *erreur*}
I → *I* ; *I* {*I.type* := **if** *I*1.*type* = *I*2.*type* = *void*
then *void* **else** *erreur*}

Les fonctions à plusieurs arguments peuvent être traitées au moyen des types produits.