Programmer en Exalgo

1 Introduction

Il existe un programme de traduction de Exalgo vers C que vous pouvez utiliser pour tester des programmes écrits en Exalgo.

Ce programme est disponible en forme C ou exécutable dans les fichiers `~jrobson/Exalgo/Exalgov3a.c` et `~jrobson/Exalgo/Exalgov3a.exe` (pour la version la plus récente version 3a). Si vous voulez compiler vous-même le fichier c, il faudra aussi copier les deux fichiers lex.yy.c et symtab.c.

Vous pouvez, par exemple, compiler et exécuter un fichier `prog.exo` par les commandes

```
~jrobson/Exalgo/Exalgo.exe <prog.exo >prog.c
gcc prog.c
a.out
```

Le traducteur ne marche que pour une version restreinte de Exalgo (détails plus bas) et il est mis à votre disposition sans garantie.

2 Restrictions/Clarifications sur le langage

- Chaque type dans une déclaration de variable ou définition d'un type de tableau ou pointeur ou spécification d'un paramètre ou champ d'un enregistrement doit être un identificateur, soit un type prédéfini soit un type défini dans le programme.
- Le nombre de paramètres d'une fonction ou procédure est limité à 31.
- Une déclaration de variables peut inclure plusieurs variables comme: `var i,j,k:entier;`
- Les spécifications des types de paramètres peuvent inclure plusieurs paramètres avec une syntaxe comme:

  ```
  procédure P(val x,y,z:réel; ref p,q:entier);
  ```

- La `fin` avant la `finfonction` ou `finprocedure` peut être omise.
- Tout identificateur utilisé doit être défini/déclaré avant utilisation ; donc, les fonctions/procédures mutuellement récursives ne sont pas traitées; seule exception : une déclaration d'un type `^X` suivie immédiatement par une déclaration du type `X`.
- dans une instruction

  ```
  pour i allant de exp1 à exp2 par pas de exp3 faire
  ```

la variable `i` est implicitement déclarée (de type `entier`) et les expressions `exp1 exp2 exp3` sont évaluées une seule fois avant la première itération; des instructions changeant la valeur de la variable de contrôle `i` sont fortement déconseillées et seront détectées comme des erreurs dans une prochaine version.

- Entrées et sorties sont effectuées par des appels des procédures `printf` et `scanf` qui ont la même syntaxe qu'en C sauf qu'il n'y a pas de `&` avant les variables dans `scanf`.

3 Avertissements

- Si le programme a des erreurs de syntaxe, seul le premier est signalé.
- Le traducteur n'analyse pas les types du programme; des éventuels erreurs de type seront trouvés ou non par `gcc`.
- Tout type de tableau défini comme `tableau inf..sup de type` sera traité comme si `inf` était nul.
- N'essayez pas de lire le C produit; il est écrit pour être lu par un ordinateur, pas par un humain !
- N'essayez pas de lire le programme `~jrobson/Exalgo/Exalgo.c` si vous n'avez pas suivi un cours de compilation et/ou lex/yacc !

4 Nouveautés

- La version v3 permet d'omettre pas mal de choses du programme où ça ne produit pas d'ambiguité (par exemple `fin si` avant `fin pour`, `alors`, `;` entre déclarations où instructions, `par pas de 1`) et donne un avertissement chaque fois.
- La version v3a fait la même chose mais sans avertissements.