TP n°4, I41 – Projet *Emmanuel Bruno*
L2 Sciences et Technologies
Université du Sud Toulon-Var
Année universitaire 2005 – 2006

La génération du code

Objectif
Produire au fur et à mesure de l'analyse syntaxique le code machine correspondant à l'expression analysée.

ATTENTION : Ce TP est réalisé en modifiant l'analyseur syntaxique. Pour réaliser la génération de code, vous utiliserez des appels à la fonction `put()` que vous avez implantée au premier TP comme nous l'avons vu en TD. Pour que cela fonctionne, il faut donc inclure algo_risc.h dans algo_syntaxique.c

1. Mise en place
Mettez en place les inclusions nécessaires et créez un programme principal test_algo.c qui initialise la machine RISC qui execute une analyse syntaxique (d'un facteur dans un premier temps, cf. question suivante). Vous devrez aussi pouvoir utiliser la fonction put() dans le programme algo_syntaxique.
Dans un premier temps nous n'analysons que des expressions arithmétiques, ajoutez donc après l'analyse une instruction qui affiche la valeur qui se trouve au sommet de la pile des registres et n'oubliez pas d'ajouter un HRS à la fin du programme.
Votre programme principal pourra ensuite afficher le code généré avec la fonction decode() et quand vous pensez que cela est correct vous pourrez utiliser la fonction load() pour le charger en mémoire puis la fonction execute() pour le lancer. **Pour pourvoir tester attendre la question suivante.**

2. Code d'un facteur
Pour commencer, vous compléterez la fonction facteur() pour qu'elle ajoute le code correspondant à chaque facteur lors de l'analyse. Vous pouvez alors analyser une expression arithmétique constante qui se trouve dans un fichier. Votre programme principal devra initialiser l'analyse lexicale en ouvrant le fichier facteur.txt et executera la fonction facteur().

3. Code d'un terme
Vous pouvez ensuite comme nous l'avons vu en TD mettre en place la génération du code lors de l'analyse d'un terme, ainsi vous pourrez générer et exécuter un programme qui évalue des produits de constantes. Votre programme principal devra initialiser l'analyse lexicale en ouvrant le fichier terme.txt et executera la fonction terme().

4. Code d'une expression simple
Procédez de la même manière pour l'analyse des expressions simples.

Pour aller plus loin (cette partie est facultative)

5. Code d'une expression simple avec variables
Pour générer le code d'un expressions avec variables, **vous devez avoir une table des symboles opérationnelle.**

Modifier votre analyseur syntaxique pour qu'il reconnaisse les suites d'instructions composées de
- demander_une_valeur_pour *identificateur*
- et de montrer_la_valeur_de *expression*

Le code correspondant à demander_une_valeur_pour comprend la saisie d'une valeur au clavier, cette valeur est mise en sommet de pile puis la valeur en sommet de pile est enregistrée à l'adresse de la variable (connue grâce à la table des symboles).

6. Code d'un algorithme complet
Le code correspondant à un algorithme complet demande l'analyse de toute la grammaire. La génération du code est faite en utilisant des sauts.