Modif projet Compilation

1. Concat function

Pas de changements côtés lexical/syntaxique -> on utilise le même ‘+’ que pour une addition (barith)

Pas de changements main.c/symbols.c/toy.h/utils.c/ast.c

Fichier analysis.c :

Fonction analysis\_expression, case 2 / barith

suppression de la condition op1 & op2 doivent être des nombres.

On vérifie maintenant si op1&op1 sont des nombres, pour attribuer au nœud le bon type, si ce n’est pas des nombres le type du nœud sera « string » si l’opérateur est un +.

Fichier toy-runtime.h : déclaration de la nouvelle fonction concat

\_toy\_string \_toy\_concat\_string(\_toy\_string str1, \_toy\_string str2);

Fichier toy\_runtime.c: corps de la fonction concat

Prends en paramètre deux toy\_string (char\*) et retourne un toy\_string

Cas string null : return NULL

Fichier prodcode.c :

Dans le cas où c’est un nœud de type string, arité 2 et qu’il y a un ‘+’, on génère la fonction de concaténation.

Problème :

fail-concat.toy : print(« 1 » + « 2 »)  faux ? Print(1+2)  faux ?

2- Ajout des opérateurs +=, -=, \*= et /=

Lexical :

+= PE

-= ME

\*= TE

Syntaxic :

%right '=' PE ME TE DE

expr :

| var '=' expr { $$ = make\_expression("=", assign, 2, $1,$3);}  
| var PE expr { $$ = make\_expression("+=", assign, 2, $1,$3);}  
| var ME expr { $$ = make\_expression("-=", assign, 2, $1,$3);}  
| var TE expr { $$ = make\_expression("\*=", assign, 2, $1,$3);}  
| var DE expr { $$ = make\_expression("/=", assign, 2, $1,$3);}

Analysis.c

RAS

prodcode.c

RAS