Pablo Rauzy <pr@up8.edu>
https://pablo.rauzy.name/teaching/ic

Interprétation et compilation

TP 6 : Démarrage projet

Dans ce TP :
— Écriture d'un compilateur complet très simple.

Exercice 0.

Nous allons commencer par écrire ensemble en Racket un petit compilateur qui prend en entrée un programme consistant en un unique entier et qui produit le code MIPS qui va afficher cet entier.

1. Le lexer (`lexer.rkt`).
2. Le parseur (`parser.rkt`) et les structures pour la syntaxe parsée (`ast.rkt`).
3. L'analyse sémantique (`analyzer.rkt`) et les structures pour l'AST (`ast.rkt`).
4. La compilation (`compiler.rkt`) et les structures pour la représentation intermédiaire de l'assembleur MIPS (`ast.rkt`).
5. L'écriture des instructions MIPS (`mips-printer.rkt`).

Exercice 1.

Nous allons ensuite (toujours ensemble) rajouter la possibilité d'écrire des additions d'entiers plutôt qu'un simple entier.

1. Mise à jour du lexer (`lexer.rkt`).
2. Mise à jour du parser (`parser.rkt`) et des structures pour la syntaxe parsée (`ast.rkt`).
3. Mise à jour de l'analyse sémantique (`analyzer.rkt`) et des structures pour la représentation intermédiaire de l'assembleur MIPS (`ast.rkt`).
4. Mise à jour de l'écriture des instructions MIPS (`mips-printer.rkt`).

Exercice 2.

À vous de continuer, par couches successives allant d'un bout à l'autre, à améliorer le compilateur.

1. D'autres opérations arithmétiques.
2. Des variables.
3. Un type booléen.
4. Des opérations de comparaisons.
5. Des opérations logiques.
6. Des fonctions.
7. Etc.

Ce TP n'est pas à rendre.