TD3 : An Introduction to Compiler Construction

Thibaut Benjamin[1] and Burkhart Wolff[2][0000−0002−9648−7663]

[1] LMF, Université Paris-Saclay, IRT SystemX Paris, 91120 Palaiseau, France
thibaut.benjamin@universite-paris-saclay.fr
[2] LMF, Université Paris-Saclay, Paris, France
wolff@lmf.cnrs.fr

Abstract. This TD gives an introduction to the concepts of attibute grammars and syntax-directed translation.
The current goal is to define a small type-checker for a C-like language fragment.
The accompanying material can be found on the the following site: [1]

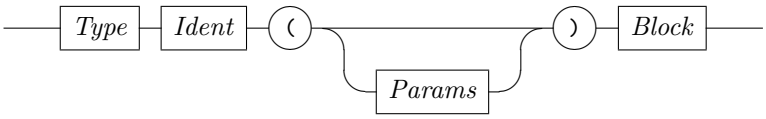
Keywords: Grammars, Parsers, Railroad Diagrams, Programming in SML

1 TD 3 : Defining a Context Analysis

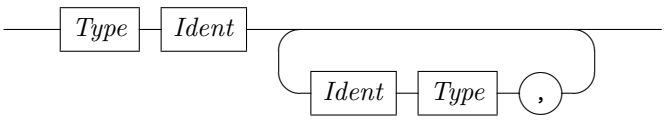
1.1 Exo 11

We define a small fragment of the C-language. It comprises only 3 types: `int`, `bool` and `void`. The language fragment provides function definitions consisting of the header and the block:

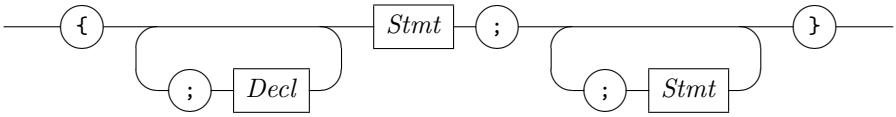
– *Fundecl*:



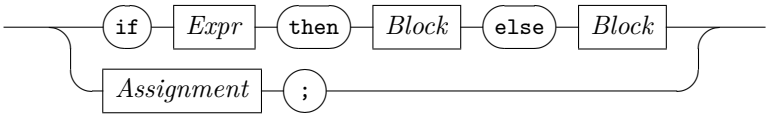
– *Params*:



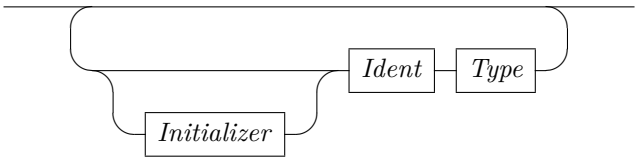
– *Block*:



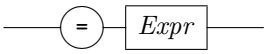
– *Stmt*:



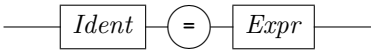
– *Decl*:



– *Initializer*:



– *Assignment*:



Tasks:

1. Construct an abstract syntax (in an SML-like notation)
2. Build an attributation that defines the
   – type environment
   – value environment
3. Note that Blocks can be nested and represent scopes of variables.

We assume the following basic data structures:

1. bool, int, sets, lists, strings with MOAL notations.
2. TEnv's which are (string, Type)Maps with the operations:
   (a) *update*: written $E\ [x \mapsto t] = E'$
   (b) *lookup*: written *lookup* $x\ E$
   (c) *is-definedness-test*: written $x\ (isdef)\ E$

References

1. Wolff, B.: Teaching Website: PolyTech Compiler Course. `https://usr.lmf.cnrs.fr/~wolff/teach-material/2025-2026/ET4-Compil/index.html` (2025), [Online; accessed 8-Dec-2025]