

```
1  (*
2      Chapitre 3
3      Graignic Guillaume
4      Cadoret olivier
5      main.ml
6  *)
7
8  open Grammar;;
9  open Lexer;;
10
11 (* Pour des petits tests *)
12 let lexbuf = s scanner (Lexing.from_string "begin int a ; a <- b + c end");;
13 (*
14 val lexbuf : Arbre.arbre_abstrait =
15   Arbre.Bloc ([Arbre.Declaration (Arbre.Int, "a")],
16     [Arbre.Affectation ("a", Arbre.Plus (Arbre.Ident "b", Arbre.Ident "c"))])
17 *)
18
19 (* Pour les plus grand tests, on lit le fichier "test" qui contient un code à
   analyser
20 Le fichier test contient "begin bool x ; begin int a ; a <- b + c end ; x <- (b
   and c) end" *)
21 let lexbuf2 = s scanner (Lexing.from_channel (open_in "test"));;
22 (*
23 val lexbuf2 : Arbre.arbre_abstrait =
24   Arbre.Bloc ([Arbre.Declaration (Arbre.Bool, "x")],
25     [Arbre.Bloc ([Arbre.Declaration (Arbre.Int, "a")],
26       [Arbre.Affectation ("a", Arbre.Plus (Arbre.Ident "b", Arbre.Ident "c"))]);
27     Arbre.Affectation ("x", Arbre.Et (Arbre.Ident "b", Arbre.Ident "c"))])
28 *)
29
30 (* TEST OK *)
```