

Cours de Compilation - Test de connaissances

Master d'Informatique M1 2011–2012

15 septembre 2011

Nom : Prénom :

Soit un langage d'expressions booléennes défini par :

| | |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code><prog></code> | <code>::= <inst> <prog> <inst></code> |
| <code><inst></code> | <code>::= set <i>ident</i> (<i>ident</i>) = <expr></code> |
| <code><expr></code> | <code>::= true false <i>ident</i> </code> <code> <expr> <binop> <expr> <i>ident</i> (<expr>)</code> <code> (<expr>)</code> |
| <code><binop></code> | <code>::= && =</code> |

Soit le programme:

```
set neg (x) = x = false
set f (x) = x || neg (x)
```

On introduit des tokens SET, TRUE, FALSE correspondant aux mots clés **set**, **true** et **false**, le token IDENT pour les identificateurs et les tokens suivants pour les symboles : LP ((), RP ()), AND (&&), OR (||) et EQ (=).

1. Quels sont les symboles non-terminaux de la grammaire?

Correction : `<prog>`, `<inst>`, `<expr>` et `<binop>`

2. La grammaire est-elle ambiguë?

Correction : oui, deux manières d'interpréter `true || false` ~~et~~ `true`

3. Donner la suite de tokens associée au programme.

Correction :

```
SET IDENT LP IDENT RP EQ IDENT EQ FALSE
SET IDENT LP IDENT RP EQ IDENT OR IDENT LP IDENT RP
```

4. Donner l'arbre de dérivation syntaxique correspondant au programme.

Correction :

