Milestone 1

Tormod Hellen and Guillaume Pansier

03.06.2014

1 BNF syntax

```
Prog ::= Def #
       | Def Prog
Def ::= DEF Lhs == Expr (Def)
Lhs ::= MAIN:Type                          (MAIN)
       | id(A):Type                        (Function_def)
Type ::= nat                               (Type)
       | bool                                      (Type)
A ::= id:Type(B)                           (Params)
       | e
B ::= ,id:Type B
       | id : Type B
       | e
Expr ::= number                 (Number)
       | true                              (bool)
       | false                                     (bool)
       | id C                              (Function, Var)
       | if Expr then Expr E fi                    (ITE)
C ::= (Expr D)
       | e
D ::= , Expr D
       | e
E ::= ELSE Expr
       | e
```

2 BNF syntax without left-recursion

```
Prog ::= Def Prog1 #
Prog1 ::= Def Prog1
       | e
Def ::= DEF Lhs == Expr (Def)
Lhs ::= MAIN : Type                                (MAIN)
       | id ( A ) : Type                           (Function_def)
```

```
A ::= id : Type B                                   (Params)
    | e
B ::= , id : Type B
    | e
Type ::= nat                                (Type)
    | bool                                  (Type)
Expr ::= number                 (Number)
    | true                                  (bool)
    | false                                         (bool)
    | id C                                  (Function, Var)
    | if Expr then Expr E fi                    (ITE)
C ::= (Expr D)
    | e
D ::= , Expr D
    | e
E ::= ELSE Expr
    | e
```

All test files in skelton