

101: Byg egen compiler

undertitel

Mathias

Mirza

Mads

Gruppemedlemmere:

Gruppe Opgave



Datalogisk Institut
Compilers 2015

Indhold

Litteratur

2

For at løse opgaver i task 1, kigger vi kun på `Lexer.lex`, `Parser.grm`, `Interpreter.sml`, `Typchecker.sml` og `codegen.sml`.

Hvis vi, tager det slavisk, som i den rækkefølge de skal løses i. Det vil sige, for at implementere `TRUE` og `FALSE`, så skal de først defineres i `Lexer.lex` under keywords, der indsættes

```
42      | "true"          => Parser.TRUE pos
43      | "false"         => Parser.FALSE pos
```

i `Parser.grm` er begge blevet implementeret som tokens og `expresseion`

```
13 %token <(int*int)> TRUE FALSE

68      | TRUE           { Constant (BoolVal true , $1) }
69      | FALSE          { Constant (BoolVal false , $1) }
```

Som vi gjorde med `TRUE` og `FALSE`, så skal der også defineres en rule token for `TIMES` og `DIVIDE` inde i `Lexer.lex`, og der laves det samme som med `TRUE`. Det samme gælder for selve token, hvor det er en `(int*int)`. Selve precedence level af `TIMES` og `DIVIDE` er den højste, dvs. at $4 + 2 \cdot 3$ betyder $4 + (2 \cdot 3)$

Dog i `TIMES`, `DIVIDE`, skal vi bruge 3 registre, som det ses i

```
1      | Exp TIMES Exp  { Times($1 , $3 , $2) }
2      | Exp DIVIDE Exp { Divide($1 , $3 , $2)}
```

Litteratur