

Intermediate Representation (IR) grammar

(N^* denotes 0, 1 or more repetitions of N)

```
IRProgram    → StringsBlock IRStm*
StringsBlock → strings { StringDef* }
               →
StringDef    → id : STRING
IRStm        → MOVE ( TEMP id , IRExp )
               → MOVE ( MEM ( IRExp ) , IRExp )
               → NOOP
               → JUMP ( IRExp )
               → CJUMP ( IRExp , op , IRExp , id , id )
               → EXP ( IRExp )
               → LABEL id
               → SEQ ( IRStm , IRStm )
               → PROLOGUE ( POSINT , POSINT )
               → EPILOGUE ( POSINT , POSINT )
IRExp        → BINOP ( IRExp , op , IRExp )
               → CALL ( IRExp Arg* )
               → CONST IntLit
               → MEM ( IRExp )
               → TEMP id
               → NAME id
               → ESEQ ( IRStm , IRExp )
IntLit       → POSINT
               → NEGINT
Arg          → , IRExp
```

Additional information about the terminal symbols:

STRING is a sequence of arbitrary characters enclosed in quotation marks, for example: "**Th&_\$\$is 12is a3@ STRING.**" [Note: there is no syntax for escaping characters, so a *STRING* cannot include the " character.]

op is one of the following: **LT LE EQ ADD SUB MUL DIV**

id is a non-empty sequence of letters, underscores, @ symbols, dots (the . character) and digits, starting with a non-digit

POSINT is a non-empty sequence of digits [Note: **0** is a *POSINT*!]

NEGINT is a minus-sign followed by a non-empty sequence of digits

There is no syntax for comments.