**Compiler Course Work Part II Report**

Task II-1 :

****

T-IF

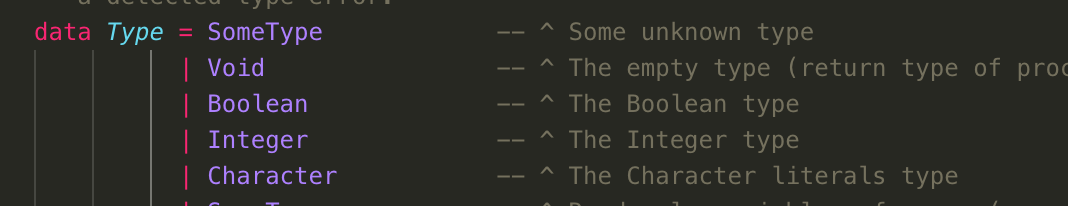
T-COND

T-REPEAT

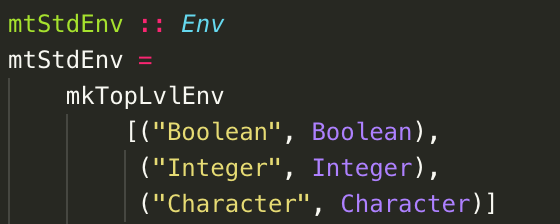
Task II-2:

Character:

Add Character type in Type.hs



Modify MTStdEnv.hs

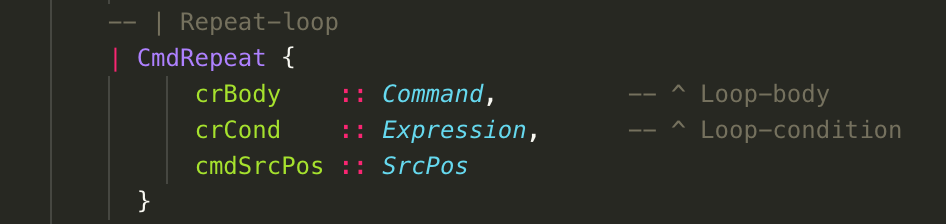


Modify TypeChecker.hs

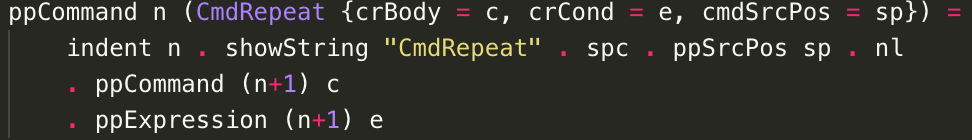


Repeat until loop:

Modify MTIR.hs

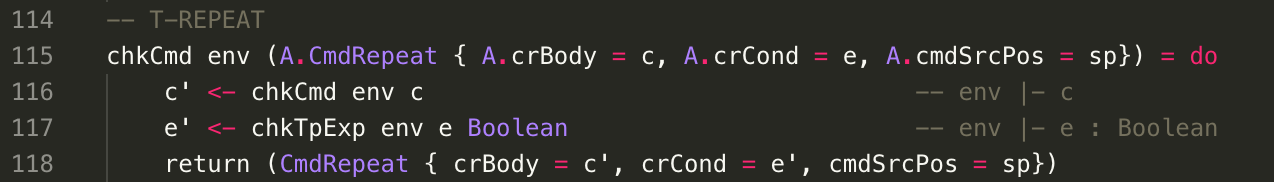


Modify PPMTIR.hs



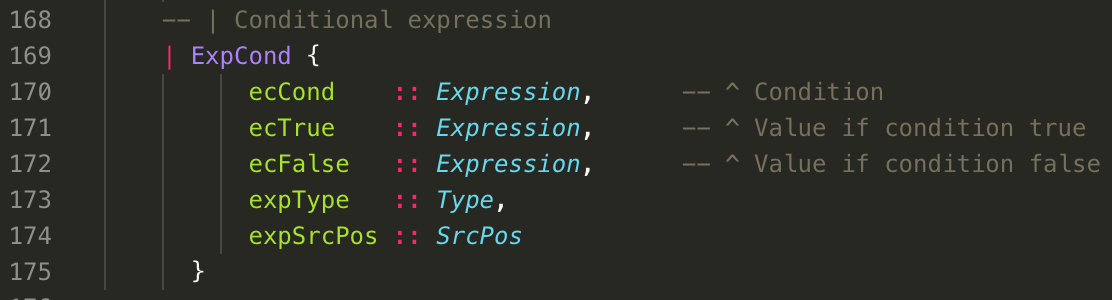
These parts is pretty similar to the AST.hs and PPAST.hs

Modify TypeChecker.hs add chkCmd



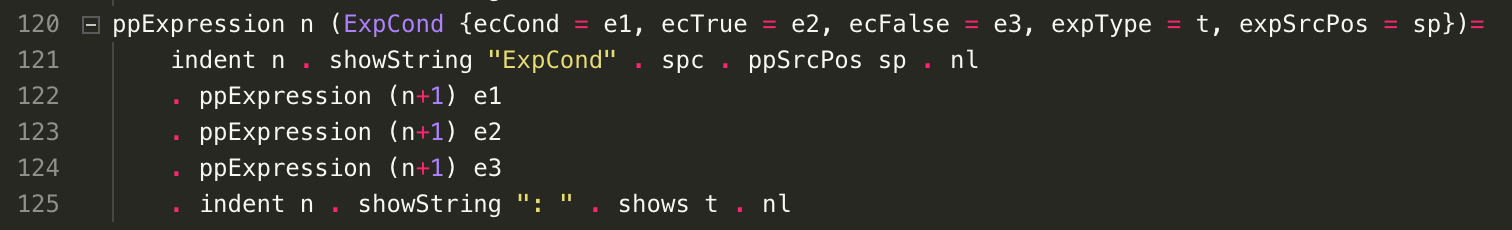
Cond Expression:

Modify MTIR.hs:

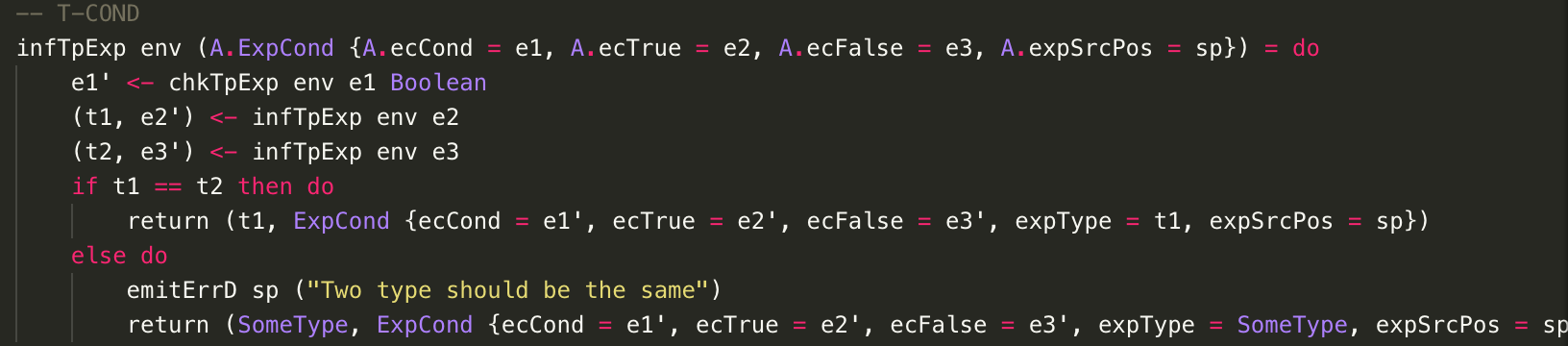


Kind of different from AST.hs Expression should have a type.

Modify PPMTIR.hs:

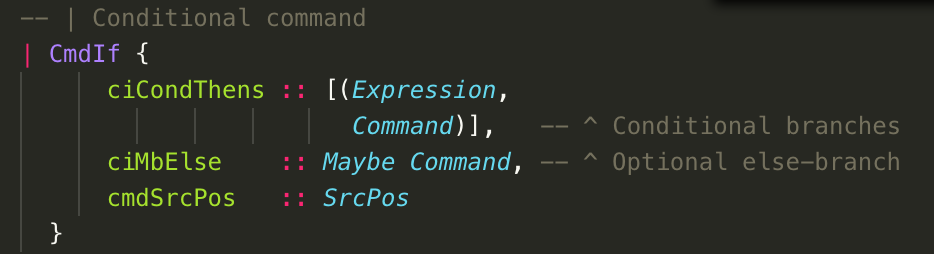


Modify TypeChecker.hs

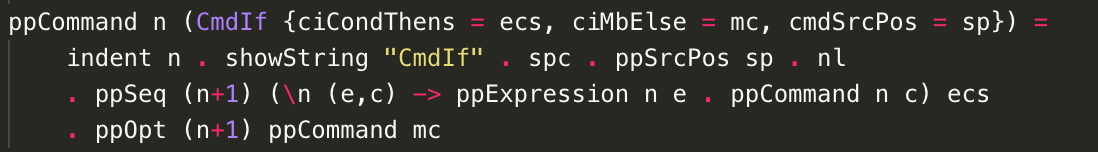


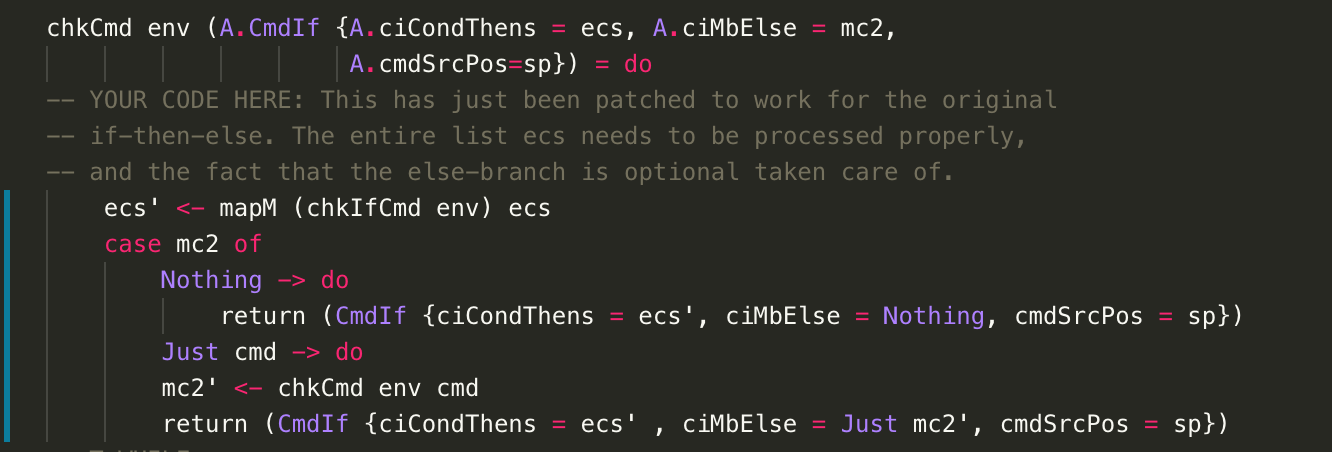
Extended if-then-else-commands:

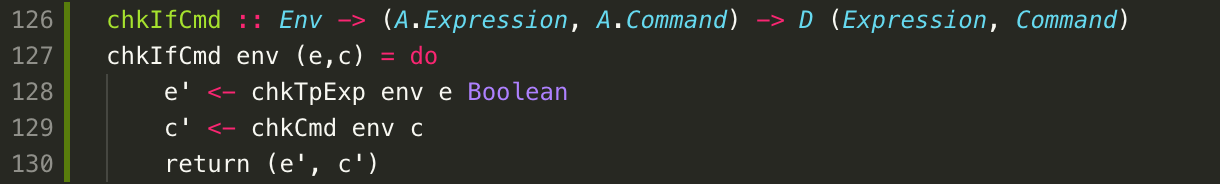
MTIR.hs



PPMTIR.hs:



TypeChecker.hs:



Create a function to check each pair of expression and command

Task II-3:

1. MyTAMCode-3a.tam

LOADL 1

GETINT

#loop:

LOADL 0

LOAD [SB + 1]

LSS

JUMPIFZ #halt

LOAD [SB + 0]

PUTINT

LOAD [SB + 0]

LOADL 1

ADD

STORE [SB + 0]

LOAD [SB + 1]

LOADL 1

SUB

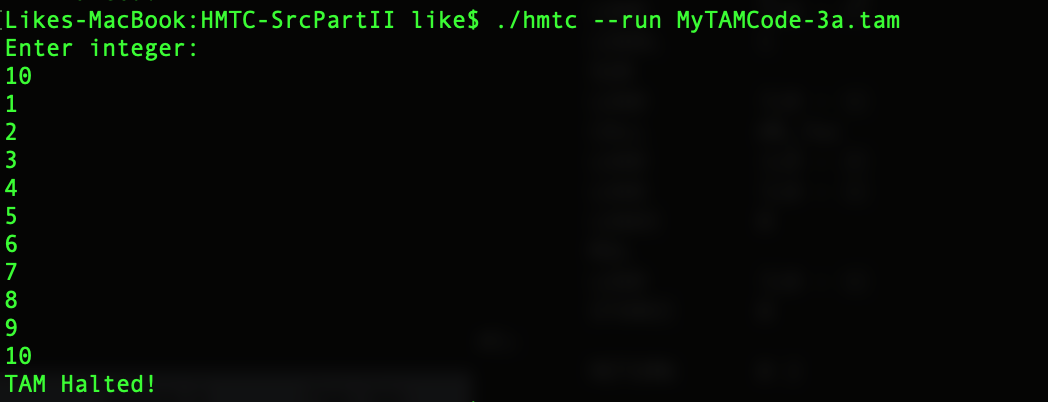
STORE [SB + 1]

JUMP #loop

#halt:

HALT

Result:



1. MyTAMCode-3b.tam

GETINT

CALL #fac

MUL

PUTINT

HALT

#fac:

LOAD [LB - 1]

LOADL 1

SUB

LOAD [LB + 3]

JUMPIFZ #basecase

CALL #fac

MUL

RETURN 1 0

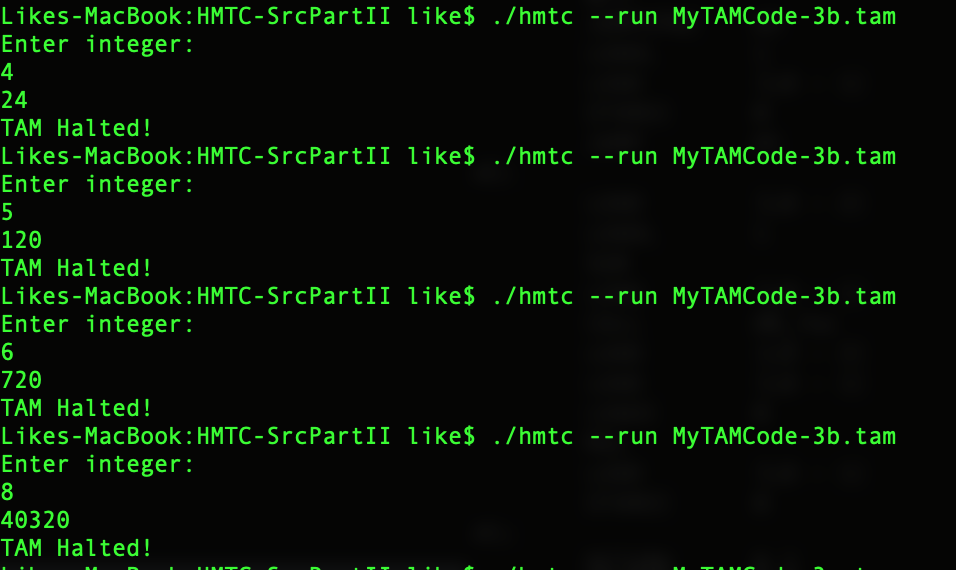
#basecase:

POP 0 1

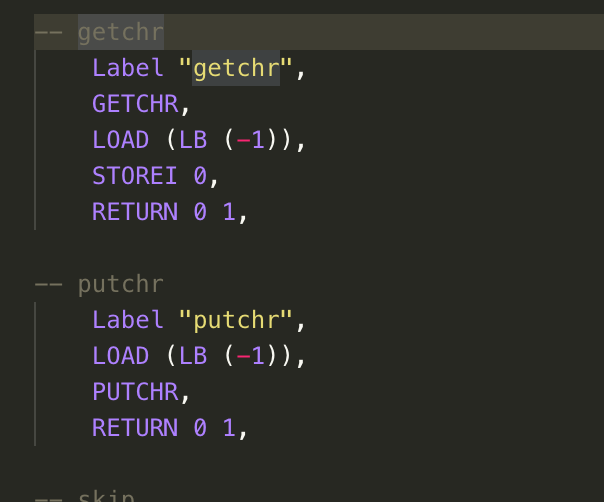
LOADL 1

RETURN 1 0

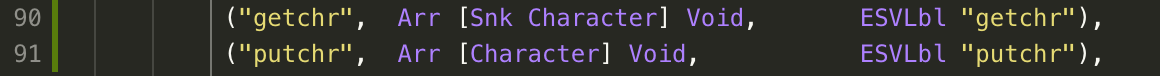
Result:



1. LibMT.hs:

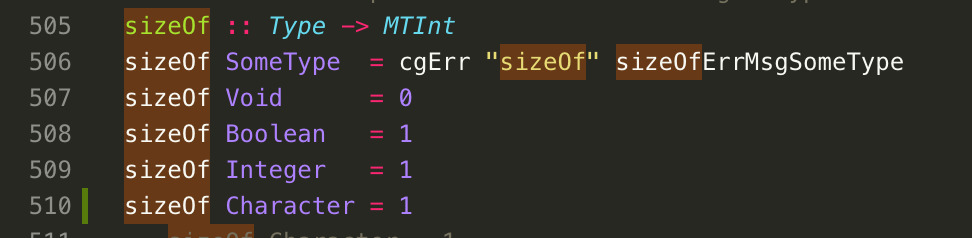


MTStdEnv.hs:

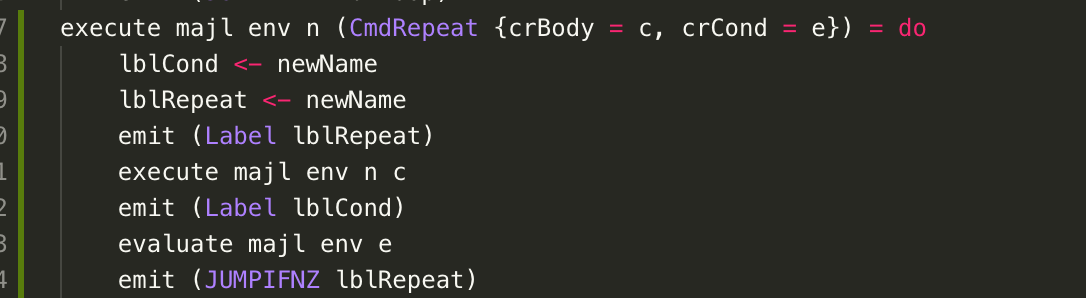


Task II-4: modify codeGenerator .hs

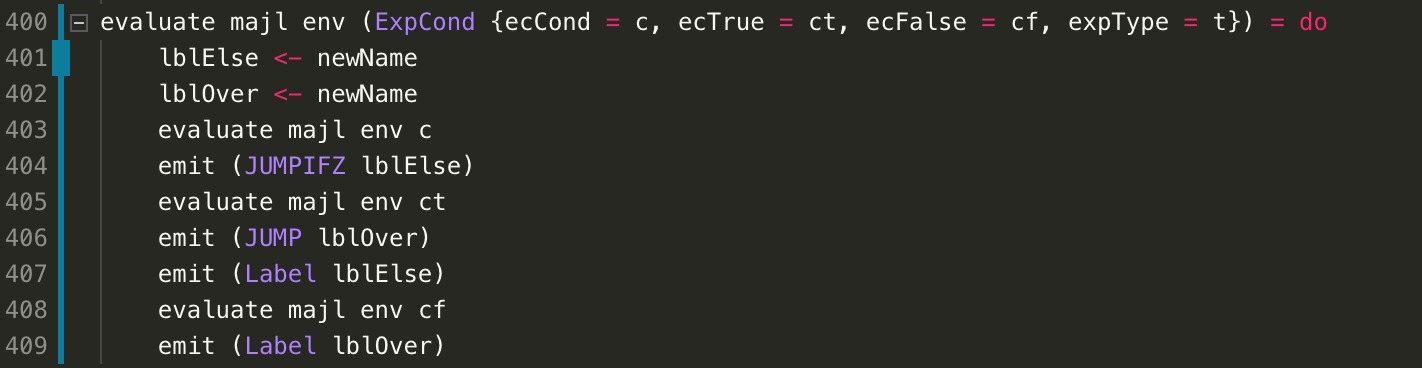
Character type:



Repeat until loop:



Condition expression:



Extend if:

Two part:

