Project Proposal

Group: tekcar

November 20, 2017

Features

1. `call/cc`
2. Tail Call Optimization
3. Exceptions

$$
\begin{array}{rcl}
cmp & ::= & \texttt{eq?} \mid \texttt{<} \mid \texttt{<=} \mid \texttt{>} \mid \texttt{>=} \\
exp & ::= & int \mid (\texttt{read}) \mid (\texttt{-}\ exp) \mid (\texttt{+}\ exp\ exp) \\
 & \mid & var \mid (\texttt{let}\ ([var\ exp])\ exp) \\
 & \mid & \texttt{\#t} \mid \texttt{\#f} \mid (\texttt{and}\ exp\ exp) \mid (\texttt{not}\ exp) \\
 & \mid & (cmp\ exp\ exp) \mid (\texttt{if}\ exp\ exp\ exp) \\
 & \mid & (\texttt{vector}\ exp^{+}) \mid (\texttt{vector-ref}\ exp\ exp) \\
 & \mid & (\texttt{vector-set!}\ exp\ exp\ exp) \mid (\texttt{void}) \\
 & \mid & (exp\ exp^{*}) \mid (\texttt{lambda}\ (var^{*})\ exp) \\
 & \mid & (\texttt{call/cc}\ exp) \mid (\texttt{with-handlers}\ ([var\ exp])\ exp) \\
 & \mid & (\texttt{raise}\ exp) \mid (\texttt{exn:fail?}\ exp) \\
def & ::= & (\texttt{define}\ (var\ var^{*})\ exp) \\
R_8 & ::= & (\texttt{program}\ def^{*}\ exp)
\end{array}
$$

Plan

1. A new pass named `convert-to-cps` to convert the input program to Continuation Passing Style.

   And `(call/cc f)` can be coverted to
   `((lambda (g cc) (g (lambda (v cc-skip) (cc v)) cc)) f' cc)`

2. Change the `callq` and `retq` to `jmp` for tail calls.

3. Add another argument for cps calls to deal with exceptions:
   `(app f args* cc)` $\rightarrow$ `(app f' args'* cc' handler)`.
   And use `call/cc` to jump to the `handler` when exceptions are raised.