Assignment 7

Out: Wednesday, 03 December 2003
Due: Friday, 12 December 2003

Total: 120 points

Exercise 1 (40 points) Let UFPL be the untyped functional programming language obtained from erasing types in TFPL. We extend UFPL with the following language constructs:

expressions  
\[ e ::= \ldots \cdot E \mid \text{callcc}(e) \mid \text{throw}(e_1, e_2) \]

values  
\[ v ::= \ldots \mid \cdot E \]

evaluation contexts  
\[ E ::= \ldots \mid \text{callcc}(E) \mid \text{throw}(E, e) \]

The evaluation rules for these language constructs are given below:

\[
E[\text{callcc}(v)] \rightarrow E[v(\cdot E)] \\
E[\text{throw}(\cdot E', e)] \rightarrow E'[e]
\]

Please implement a function \textit{wrap-fun-cont} which takes a function \( f \) and an evaluation context \( \cdot E \) and then returns the evaluation context \( \cdot E[f([\ ])] \). In other words, we have

\[
E_0[\text{wrap-fun-cont}(f)(\cdot E)] \rightarrow^* E_0[\cdot E[f([\ ])]]
\]

for any \( E_0 \).

Exercise 2 (80 points) Please implement an evaluator for the functional programming language TFPL by following the dynamic semantics defined via the use of evaluation contexts.