Compilation of Pattern Matching

April 28, 2017

1 Sequential Abstract Machine for MPL (SAMPL)

SAMPL acts as the target to which the sequntial MPL compiles and is subsequently run. *SAMPL* is inspired by *modern-SEC Machine* which in turn was inspired by Landin's SECD machine. However, there are a few differences between *SAMPL* and *modern-SEC Machine*, the most notable one of which are the following :-

- *SAMPL* is a more sophisticated machine as it has built in *data types* and *codata types* where as *modern-SEC Machine* lacks these constructs.
- *modern-SEC Machine* uses *strict evaluation* strategy for the execution of programs. In contrast, *SAMPL* uses a mixture of two strategies for program reduction, namely *lazy evaluation* when handling *records* of a *codata type* and *strict evaluation* strategy when reducing other MPL constructs

1.1 Evaluation Strategies

SAMPL is an

Recall that AMPL is represented as a four tuple $\{S, T, E, C\}$ of *Stack*, *Translation*, *Environment* and *Code*. However, *Translation* can be removed from the tuple while discussing *SAMPL*. There

The major differences between *modern-SECD Machine* and *SAMPL* is are the following :-

- *modern-SEC Machine* uses De-Bruijn's indexes to lookup values in an environment. However,
-