

Hi Grigore,

I have a feeling that we in fact disagree on nothing. Let me use a simple IMP program as an example to illustrate what I mean.

Remind that we have Maude module called IMP that builds all patterns.

```
fmod IMP is
  sorts Variable Trm Trm? Predicate Pattern ... .
  ...
  op ite : Pattern{Bool} Pattern{Pgm} Pattern{Pgm}
    -> Pattern{Pgm} [frozen(2 3)] .
  op while : Pattern{Bool} Pattern{Pgm}
    -> Pattern{Pgm} [frozen(1 2)] .
  ...
endfm
```

My goal is to write a Maude module called EXE that reduces configurations of IMP programs according to their semantics.

```
fmod EXE is
  vars P Q : Pattern{Pgm} . var B : Pattern{Bool} .
  eq ite(tt, P, Q) = P .
  eq ite(ff, P, Q) = Q .
  eq while(B, P) = ite(B, seq(P, while(B, P)), skip) .
  ...
endfm

reduce

#cfg(#k(seq(...)),
      #state(merge(binder(
```

Let us