-- Haskell interpreter for state machines

```
run :: State -> Accept -> Transitions -> [Label] -> Bool
run current accept transitions [] = current ‘elem‘ accept
run current accept transitions (l:ls) =
  case lookup l (fromJust (lookup current transitions)) of
   Nothing -> False
   Just next -> run next accept transitions ls
```

-- Desired output from partial evaluation

```
run1 ls = if null ls
            then False
            else if head ls == ’a’
                    then run2 (tail ls)
                    else False

run2 ls = if null ls
            then True
            else if head ls == ’b’
                    then run1 (tail ls)
                    else False
```