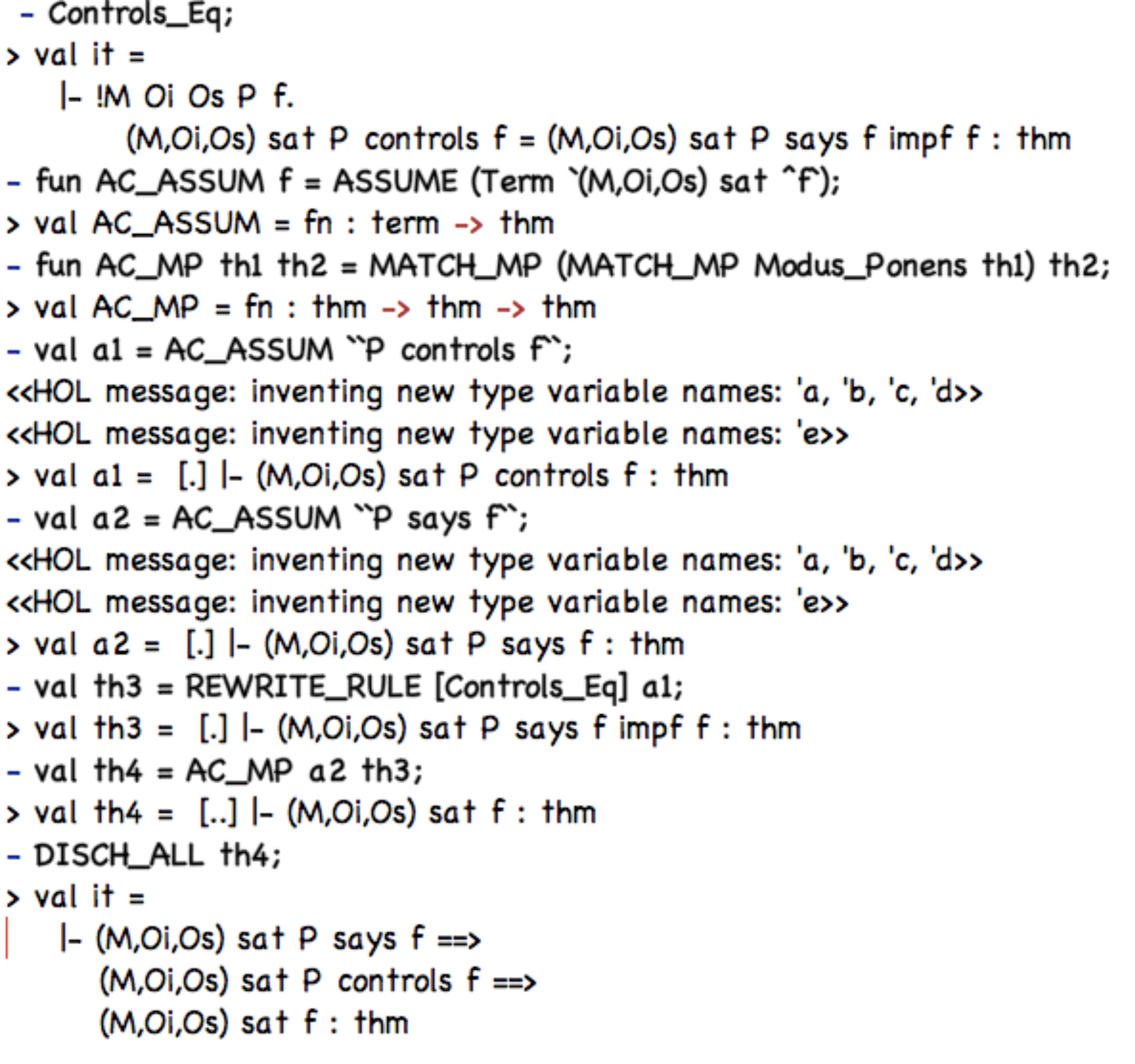

```
 - Controls_Eq;
> val it =
    |- !M Oi Os P f.
         (M,Oi,Os) sat P controls f = (M,Oi,Os) sat P says f impf f : thm
- fun AC_ASSUM f = ASSUME (Term `(M,Oi,Os) sat ^f`);
> val AC_ASSUM = fn : term -> thm
- fun AC_MP th1 th2 = MATCH_MP (MATCH_MP Modus_Ponens th1) th2;
> val AC_MP = fn : thm -> thm -> thm
- val a1 = AC_ASSUM ``P controls f``;
<<HOL message: inventing new type variable names: 'a, 'b, 'c, 'd>>
<<HOL message: inventing new type variable names: 'e>>
> val a1 =  [.] |- (M,Oi,Os) sat P controls f : thm
- val a2 = AC_ASSUM ``P says f``;
<<HOL message: inventing new type variable names: 'a, 'b, 'c, 'd>>
<<HOL message: inventing new type variable names: 'e>>
> val a2 =  [.] |- (M,Oi,Os) sat P says f : thm
- val th3 = REWRITE_RULE [Controls_Eq] a1;
> val th3 =  [.] |- (M,Oi,Os) sat P says f impf f : thm
- val th4 = AC_MP a2 th3;
> val th4 =  [..] |- (M,Oi,Os) sat f : thm
- DISCH_ALL th4;
> val it =
    |- (M,Oi,Os) sat P says f ==>
       (M,Oi,Os) sat P controls f ==>
       (M,Oi,Os) sat f : thm
```