

Oct 17, 16 7:32

## ImpStepStub.v

Page 1/2

```

Require Import List.
Require Import String.
Require Import ZArith.

Open Scope list_scope.
Open Scope string_scope.
Open Scope Z_scope.

Require Import ImpSyntax.
Require Import ImpCommon.
Require Import ImpEval.

Inductive step :
  store -> heap -> stmt ->
  store -> heap -> stmt -> Prop :=
| step_set :
  forall s h x e v,
  eval_e s h e v ->
  step
  s h (Sset x e)
  (update s x v) h Snop

(**
  TODO

  Please write the rules for Salloc and Swrite.

  You may want to use helpers from ImpCommon.v.
*)

| step_ifelse_t :
  forall s h e p1 p2,
  eval_e s h e (Vbool true) ->
  step
  s h (Sifelse e p1 p2)
  s h p1
| step_ifelse_f :
  forall s h e p1 p2,
  eval_e s h e (Vbool false) ->
  step
  s h (Sifelse e p1 p2)
  s h p2
| step_while_t :
  forall s h e p,
  eval_e s h e (Vbool true) ->
  step
  s h (Swhile e p)
  s h (Sseq p (Swhile e p))
| step_while_f :
  forall s h e p,
  eval_e s h e (Vbool false) ->
  step
  s h (Swhile e p)
  s h Snop
| step_seq_nop :
  forall s h p2,
  step
  s h (Sseq Snop p2)
  s h p2
| step_seq :
  forall s h p1 p2 s' h' p1',
  step
  s h p1
  s' h' p1' ->
  step
  s h (Sseq p1 p2)
  s' h' (Sseq p1' p2) .

Inductive step_star :

```

Oct 17, 16 7:32

## ImpStepStub.v

Page 2/2

```

  store -> heap -> stmt ->
  store -> heap -> stmt -> Prop :=
| step_star_refl :
  forall s h p,
  step_star
  s h p
  s h p
| step_star_l :
  forall s1 h1 p1 s2 h2 p2 s3 h3 p3,
  step
  s1 h1 p1
  s2 h2 p2 ->
  step_star
  s2 h2 p2
  s3 h3 p3 ->
  step_star
  s1 h1 p1
  s3 h3 p3.

```