

Oct 17, 16 6:50

ImpStep.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
| step_alloc :
  forall s h x e1 e2 i v,
  eval_e s h e1 (Vint i) ->
  eval_e s h e2 v ->
  0 <= i ->
  step
  s h (Salloc x e1 e2)
  (update s x (Vaddr (zlen h))) (alloc h i v) Snop
| step_write :
  forall s h x e1 e2 a l i v h',
  lkup s x = Some (Vaddr a) ->
  read h a = Some (Vint l) ->
  eval_e s h e1 (Vint i) ->
  eval_e s h e2 v ->
  0 <= i < l ->
  write h (Zsucc (a + i)) v = Some h' ->
  step
  s h (Swrite x e1 e2)
  s h' Snop
| 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

```

Oct 17, 16 6:50

ImpStep.v

Page 2/2

```

| 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 :
  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.

```