

Oct 17, 16 13:44	ImpTermination.v	Page 1/2
<pre> Require Import List. Require Import String. Require Import ZArith. Open Scope list_scope. Open Scope string_scope. Open Scope Z_scope. Require Import StructTactics. Require Import ImpSyntax. Require Import ImpCommon. Require Import ImpEval. Require Import ImpStep. Require Import ImpSemanticsFacts. Definition always_diverges (p : stmt) : Prop := forall s p' s', step_star s p s' p' -> p' <> Snop. Definition WTN : stmt := Swhile (Eval (Vbool true)) Snop. Lemma while_t_diverges : always_diverges WTN. Proof. unfold always_diverges, WTN; intros. prep_induction H. induction H; intros; subst. - congruence. - inv H. (** check IH, not inductive! *) Abort. Lemma while_t_sstar_cases : forall s s' p', step_star s WTN s' p' -> p' = Sseq Snop WTN \ / p' = WTN. Proof. unfold WTN; intros. prep_induction H. induction H; intros; subst. - right. reflexivity. - inv H. (** check IH, still not inductive! *) Abort. Definition WTN' : stmt := Sseq Snop (Swhile (Eval (Vbool true)) Snop). Lemma while_t_sstar_cases : forall s p s' p', step_star s p s' p' -> (p = WTN \ / p = WTN') -> p' = WTN \ / p' = WTN'. Proof. unfold WTN, WTN'; intros. prep_induction H. induction H; intros; subst. - inv H0. + left. reflexivity. + right. reflexivity. - apply IHstep_star. inv H1. + inv H. * auto. * inv H7. + inv H. </pre>		

Oct 17, 16 13:44	ImpTermination.v	Page 2/2
<pre> * auto. * inv H7. Qed. Lemma while_t_diverges : always_diverges WTN. Proof. unfold always_diverges, WTN; intros. apply while_t_sstar_cases in H. - unfold WTN, WTN' in *. inv H; congruence. - unfold WTN, WTN' in *; auto. Qed. </pre>		