

```

{-# OPTIONS --guardedness #-}
{-# OPTIONS --allow-unsolved-metas #-}

open import Language
open import Traces
open import Codata.Musical.Notation
open import Relation.Binary.PropositionalEquality using (≡_)
open import BigRel hiding (execDeterministic)
open exloop hiding (exloopincrementing)
open Trace₁

exloopincrementing : exec (Swhile (λ _ → 1) (Sassign 0 add1)) startState incrementingtrace
exloopincrementing = t startState
  where
    t : (st : State) → exec (Swhile (λ _ → 1) (Sassign 0 add1)) st (incrementingFrom st)
    t st = execWhileLoop
      (tcons st (# (tcons st (# (tnil (update 0 (add1 st) st)))))
        -
        ≡_.refl
        (execseqCons st _ (# execseqNil (execAssign (tcons (# tnil)))))
        (execseqCons st _ (# (execseqCons st _ (# (execseqNil (t (next st)))))))

execDeterministic : {s : Stmt} {st : State} {tr₁ tr₂ : Trace₁}
  → exec s st tr₁
  → exec s st tr₂
  → tr₁ ≈ tr₂

-- Shut agda up about lack of implementation
execDeterministic = {! !}

```