

module target where

-- Operator precedence and associativity

infix 4 $_ \leq_$

infixl 6 $_ \dot{-}_s _ -_s _$

open import lib

-- Stack descriptor: (frames, displacement)

record SD : Set where

constructor $_ \dot{-}_s _$

field

f : \mathbb{N}

d : \mathbb{N}

-- Stack descriptor operations

$_ +_s _ : SD \rightarrow \mathbb{N} \rightarrow SD$

f , d $\dot{+}_s$ n = f , d + n

$_ \dot{-}_s _ : SD \rightarrow \mathbb{N} \rightarrow SD$

f , d $\dot{-}_s$ n = f , d - n

-- $_ -_s _ : (sd : SD) \rightarrow (n : \mathbb{N}) \rightarrow n \leq SD.d \ sd \rightarrow SD$

-- (S_f , S_d $\dot{-}_s$ n) p = S_f , (S_d - n) p

$_ -_s _ : (sd : SD) \rightarrow (n : \mathbb{N}) \rightarrow (n \leq d : n \leq SD.d \ sd) \rightarrow SD$

(f , d $\dot{-}_s$ n) $n \leq d$ = f , (d - n) $n \leq d$

$\dot{-}_s \equiv : \forall \{f \ d \ d' \ n\} \rightarrow \{n \leq d' : n \leq d'\} \rightarrow (d' - n) \ n \leq d' \equiv d \rightarrow f , d \equiv ($

$\dot{-}_s \equiv p \text{ rewrite } p = \text{refl}$

-- Stack descriptor lexicographic ordering

data $_ \leq_s _ : SD \rightarrow SD \rightarrow Set$ where

$\leq\text{-}f : \forall \{f \ f' \ d \ d'\} \rightarrow f < f' \rightarrow f , d \leq_s f' , d'$

$\leq\text{-}d : \forall \{f \ d \ d'\} \rightarrow d \leq d' \rightarrow f , d \leq_s f , d'$

$\leq_s\text{-refl} : \forall \{sd : SD\} \rightarrow sd \leq_s sd$

$\leq_s\text{-refl} \{ f , d \} = \leq\text{-}d \leq\text{-refl}$

$\leq_s\text{-trans} : \forall \{sd \ sd' \ sd'' : SD\} \rightarrow sd \leq_s sd' \rightarrow sd' \leq_s sd'' \rightarrow sd \leq_s sd''$

$\leq_s\text{-trans} (\leq\text{-}f \ f < f') (\leq\text{-}d \ _) = \leq\text{-}f \ f < f'$

$\leq_s\text{-trans} (\leq\text{-}f \ f < f') (\leq\text{-}f \ f' < f'') = \leq\text{-}f (\leq\text{-trans} \ f < f' \ f' < f'')$

$\leq_s\text{-trans} (\leq\text{-}d \ _) (\leq\text{-}f \ f' < f'') = \leq\text{-}f \ f' < f''$

$\leq_s\text{-trans} (\leq\text{-}d \ d \leq d') (\leq\text{-}d \ d' \leq d'') = \leq\text{-}d (\leq\text{-trans} \ d \leq d' \ d' \leq d'')$

$\dot{+}_s \rightarrow \leq_s : \forall \{sd : SD\} \rightarrow \forall \{n : \mathbb{N}\} \rightarrow sd \leq_s sd \dot{+}_s n$

$\dot{+}_s \rightarrow \leq_s = \leq\text{-}d \ \dot{+} \rightarrow \leq$

-- Operator

data UnaryOp : Set where

UNeg : UnaryOp

data BinaryOp : Set where

BPlus : BinaryOp

BMinus : BinaryOp

BTimes : BinaryOp

data RelOp : Set where

RLeq : RelOp

RLt : RelOp

-- Nonterminals

-- Lefthand sides

data L (sd : SD) : Set where

l-var : ($sd^v : SD$) $\rightarrow sd^v \leq_s sd \rightarrow L \ sd$

l-sbrs : L sd

-- Simple righthand sides

data S (sd : SD) : Set where

s-l : L sd $\rightarrow S \ sd$

s-lit : $\mathbb{Z} \rightarrow S \ sd$

-- Righthand sides

data R (sd : SD) : Set where

r-s : S sd $\rightarrow R \ sd$

r-unary : UnaryOp $\rightarrow S \ sd \rightarrow R \ sd$

r-binary : S sd \rightarrow BinaryOp $\rightarrow S \ sd \rightarrow R \ sd$

-- Instruction sequences

data I (sd : SD) : Set where

stop : I sd

assign-inc : ($\delta : \mathbb{N}$) $\rightarrow L \ (sd \dot{+}_s \delta) \rightarrow R \ sd \rightarrow I \ (sd \dot{+}_s \delta) \rightarrow I \ sd$

assign-dec : ($\delta : \mathbb{N}$) $\rightarrow (\delta \leq d : \delta \leq SD.d \ sd) \rightarrow L \ ((sd \dot{-}_s \delta) \ \delta \leq d) \rightarrow R \ sd$

if-then-else-inc : ($\delta : \mathbb{N}$) $\rightarrow S \ sd \rightarrow RelOp \rightarrow S \ sd \rightarrow I \ (sd \dot{+}_s \delta) \rightarrow I \ (sd$

if-then-else-dec : ($\delta : \mathbb{N}$) $\rightarrow (\delta \leq d : \delta \leq SD.d \ sd) \rightarrow S \ sd \rightarrow RelOp \rightarrow S \ sd$

adjustdisp-inc : ($\delta : \mathbb{N}$) $\rightarrow I \ (sd \dot{+}_s \delta) \rightarrow I \ sd$

adjustdisp-dec : ($\delta : \mathbb{N}$) $\rightarrow (\delta \leq d : \delta \leq SD.d \ sd) \rightarrow I \ ((sd \dot{-}_s \delta) \ \delta \leq d) \rightarrow I \ sd$

popto : ($sd' : SD$) $\rightarrow sd' \leq_s sd \rightarrow I \ sd' \rightarrow I \ sd$