1 Program Semantics

theory *Semantics* = *Main*:Here we declare a framework for program semantics in form of a state transition relation. A state consists of a position and a memory component. Note that we use type variables, hence one can instantiate programs, positions and memories with arbitrary types.**consts** *ReachableFrom*:: $('a \times 'a)\ set \Rightarrow 'a\ set \Rightarrow 'a\ set$

inductive *ReachableFrom R S*

intros

init: $a \in S \Longrightarrow a \in ReachableFrom\ R\ S$

step: $a \in ReachableFrom\ R\ S \Longrightarrow (a,b) \in R \Longrightarrow b \in ReachableFrom\ R\ S$

consts *ReachableFromInv*:: $('a \times 'a)\ set \Rightarrow ('a\ set) \Rightarrow ('a\ set) \Rightarrow 'a\ set$

inductive *ReachableFromInv R S I*

intros

init: $a \in S \Longrightarrow a \in ReachableFromInv\ R\ S\ I$

step: $a \in ReachableFromInv\ R\ S\ I \Longrightarrow (a,b) \in R \Longrightarrow a \in I \Longrightarrow b \in I \Longrightarrow b \in ReachableFromInv\ R\ S\ I$

locale *Semantics* =

fixes *initS*:: *'prog* $\Rightarrow$ (*'pos* $\times$ *'mem*) *set*

fixes *effS*:: *'prog* $\Rightarrow$ ((*'pos* $\times$ *'mem*) $\times$ (*'pos* $\times$ *'mem*)) *set*

fixes *Reachables*:: *'prog* $\Rightarrow$ (*'pos* $\times$ *'mem*) *set*

defines

$Reachables \equiv \lambda\ \Pi.\ ReachableFrom\ (effS\ \Pi)\ (initS\ \Pi)$

lemma *ReachableFromInv-ReachableFrom*:

$s \in ReachableFromInv\ R\ S\ I \Longrightarrow s \in ReachableFrom\ R\ S$

lemma *ReachableFrom-conv*:

$s \in ReachableFrom\ R\ S = (\exists\ s0.\ s0 \in S \wedge (s0,s) \in R^*)$

lemma (**in** *Semantics*) *Reachables-conv*:

$Reachables\ \Pi = \{s.\ \exists\ s0.\ s0 \in initS\ \Pi \wedge (s0,s) \in (effS\ \Pi)^*\}$

lemma (**in** *Semantics*) *Reachables-induct* [*consumes 1*, *case-names init step*]:

$[\![\ x \in Reachables\ \Pi;\ \bigwedge s.\ s \in (initS\ \Pi) \Longrightarrow P\ s;\ \bigwedge s\ s'.\ [\![\ s \in Reachables\ \Pi;\ P\ s;\ (s,s') \in (effS\ \Pi)\ ]\!] \Longrightarrow P\ s'\ ]\!] \Longrightarrow P\ x$

end