1 Safety Logic

theory *SafetyLogic* = *Main*:The Safety Logic defines how to formulate and prove safety properties of programs. In order to use this framework one needs to come up with a formula language and judgements for provability and validity of formulas.**locale** *SafetyLogic* =

fixes *TrueF*:: *'form* ($\llcorner True \lrcorner$)

fixes *FalseF*:: *'form* ($\llcorner False \lrcorner$)

fixes *Conj*:: *'form list* $\Rightarrow$ *'form* ($\llcorner\bigwedge\lrcorner$- [70])

fixes *Impl*:: *'form* $\Rightarrow$ *'form* $\Rightarrow$ *'form* (- $\llcorner\Rightarrow\lrcorner$ - [61,60] 60)

fixes *valid*:: *'prog* $\Rightarrow$ (*'pos* $\times$ *'mem*) $\Rightarrow$ *'form* $\Rightarrow$ *bool* ((-,- $\models$ -) [61,61,60] 60)

fixes *provable* :: *'prog* $\Rightarrow$ *'form* $\Rightarrow$ *bool* ((- $\vdash$ -) [61,60] 60)

assumes *semConj*: $\Pi,s \models \llcorner\bigwedge\lrcorner\ L = (\forall f{\in}set\ L.\ \Pi,s \models f)$

assumes *semImpE*: $\Pi,s \models (f1 \llcorner\Rightarrow\lrcorner f2) \Longrightarrow (\Pi,s \models f1 \longrightarrow \Pi,s \models f2)$

assumes *semTrue*: $\Pi,s \models \llcorner True \lrcorner$

assumes *semFalse*: $\neg (\Pi,s \models \llcorner False \lrcorner)$

end