1 Safety Policy against arithmetic overflow

theory *JBC-SafetyPolicy* = *JBC-Semantics*:

constdefs
MAX :: *int*
MAX ≡ *2147483647*

MX :: *val*
MX ≡ *Intg MAX*

constdefs *safeF*::*jbc-prog* ⇒ *pos* ⇒ *expr*
safeF Π *p* ≡ (*case* (*cmd* Π *p*) *of None* ⇒ *FF*
| *Some c* ⇒ *case c*
of Load nat ⇒ *TT*
| *Store nat* ⇒ *TT*
| *Push val* ⇒ *TT*
| *New cname* ⇒ *TT*
| *Getfield vname cname* ⇒ *TT*
| *Putfield vname cname* ⇒ *TT*
| *Checkcast cname* ⇒ *TT*
| *Invoke mname nat* ⇒ *TT*
| *Return* ⇒ *TT*
| *Pop* ⇒ *TT*
| (*IBin no*) ⇒ *Rel* (*Num* (*St 1*) *no* (*St 0*)) *Leq* (*Cn MX*)
| *Goto int* ⇒ *TT*
| *CmpEq* ⇒ *TT*
| *IfIntCmp ro b* ⇒ *TT*
| *IfFalse int* ⇒ *TT*
| *Throw* ⇒ *TT*)

constdefs *sys-xcptns* :: *cname list*
sys-xcptns ≡ [*NullPointer*, *ClassCast*, *OutOfMemory*]

constdefs *initF*::*jbc-prog* ⇒ *expr*
initF Π == *And* ([*Pos* (*ipc* Π),*Eq* (*Rg 0*) (*Cn Null*),*Eq FrNr* (*Cn* (*Intg 1*))]@
(*map* (λ*C*. *Ty* (*Cn* (*Addr* (*addr-of-sys-xcpt C*))) (*Class C*)) *sys-xcptns*)@
(*let* (*C*,*M*,*pc*)=*ipc* Π;
(*D*,*Ts*,*T*,*mxs*,*mxl*,*bd*) = *method* (*fst* Π) *C M*
in map (λ*n*. *not-none* (*Rg n*)) (*upt 1* (*Suc mxl*))))

end