**CFG’s with semantics**

**<S>** <defs> public**{rAM=TS[i].VP}**

class **{T=”class”}** Main**{N=ID.VP}<**INH>**{P=rP insertCT(rAM,T,N,P,&ref)}**

**{** <CLASS\_BODY>{**&ref}** static**{TM=”static”}** public **{AM=”public”}**

void**{rT=”void”}** main**{N=ID.VP}**

(**{S=CreateScope()}){rT=rT+”->”+”void”}{insertCDT(N,”void**

**->void”,AM,TM)}**{<MST> }

<CLASS\_BODY>{&ref} } <defs>

**<INH>** extends ID**{P=lookupCT(ID.VP)**

**if(T==NULL){“Undeclared”}**

**else return ID}**| Eps

<defs> <CLASS\_DEF><defs>|Eps

<CLASS\_BODY> <o\_static>{**rTM}** <AM> **{rAM}**

<DT> **{rDT,rN}**<CLASS\_BODY’> |<AM> **{rAM}**<CLASS\_BODY’’>

**{rDT,rN}{insertCDT(rN,rDT,rAM,”-”,N)**}|<OBJ>

**{rN,rDT,rAM}{insertCDT(rN,rDT,rAM,”-”,CN)} <**CLASS\_BODY>|

<CLASS\_BODY2>**{rN,rDT,rAM}{insertCDT(rN,rDT,rAM,”rTM”,CN)}** |Eps

<CLASS\_BODY’> <M>**{insertCDT(rN,rDT,rAM,,rTM,CN)}**;<CLASS\_BODY>|

( **{S=CreateScope()}** <ARGS> **{rP}** )

**{insertCDT(rN,rDt+”>”rP,rAM,rTM,CN)}**{<MST>}<CLASS\_BODY>

<CLASS\_BODY’’> DT ID<CLASS\_BODY’> | ID <CLASS\_BODY’’’>

<CLASS\_BODY’’’> ID<CLASS\_BODY’>|(<ARGS>){<CONSTRUCTOR\_CALLING><MST>}

<CLASS\_DEF> <AM> class ID<INH>{<CLASS\_BODY>}

<MST> <SST><MST>|Eps

<SST> ID **{rN1=TS[i].VP**}

<SST’> **{rN}** | DT **{rT=TS[i].VP}** <SST”>| <For\_st>|<Until\_st> |<IF\_ST>|

<INC\_DEC>|this.<this1> | super.<super1>

<OBJ’> ID = new <DT> (<ARGS2>)

<ARRAY\_DEC’> [<E>]<opt> ID<opt2>

<SST’> (<ARGS2>**{rPL}** ) **{ RT=lookupFN(rN1,rPL,CN,,rAM,rTM)}** ;| ID

**{rN2=TS[i].VP lookupCT(rN1)}** <SST7>

**{insertST(rN2,RT”🡪”+rN1,S)}**;|[<SST9>; |

<P””> **{T}**=**{OP=TS[i].VP}**<Q>

**{T1 T=typecompatibility(T1,T2,OP)}**;|=<SST12>;|<INC\_DEC>;|

this.<this’>|super.<super’>

<SST”> [<SST11>;| ID <SST10>;| this.<this’>|super.<super’>

<SST7> ={OP=TS[i].VP} <SST12>

**{ RT=typeCompatibility=(rN1,rN2,rOP) }** |<M’>

<SST8> [] ID <M>|ID<M>

<SST9> <E>] <opt> <opt3>|]<SST8>

<SST10> =<SST12>|<M’>| <P””>=<Q> (<ARGS2>)

<SST11> ]<SST13>| <E>]<opt> <opt3>

<SST12> new <DT>**{rN2}** (<ARGS2>{rPL})

**{RT=lookupCDT(rN2,rPL,rN1,rAM,rTM)}** |<OE><M’>

<SST13> [ ] ID <SST12>|ID<SST7>

<opt3> ID<opt2>|=<Q>

<SST> <DEC>|<FUNC\_CALL>|<OBJ>>|<ARRAY\_DEC>|<FOR\_LOOP>

|<Until\_st>|<IF\_ST>|<ASSIGN\_ST>|<INC>|<DECR>

<OBJ> <DT> ID = new <DT> (<ARGS2>)

<CONSTRUCTOR> <AM> ID(<ARGS>){<CONSTRUCTOR\_CALLING><MST>}

<CONSTRUCTOR\_CALLING> super(<ARGS2>);|this(<ARGS2>);|Eps

<ASSIGN\_ST> <P>=<Q>

<P> ID**{rN2=TS[i].VP}**<P””>

<P””> <P’>| (<ARGS2>**{rPL}**) **{T=lookupCDT(rN2,rPL,rAM,rTM,CN)}** ;<P’>

|[<E>]<opt><P’>|,<Q>| (<ARGS2>)=<P’>| (<ARGS2>**{rPL}**),<Q>**{rPL2}**

**{T=lookupCDT(rN2,rPL+rPL2,rAM,rTM,CN)}**  | (<ARGS2>**{rPL}**).<P>

<Q> <P>**{rT1}**|<OE>**{rT1}**

<P’> . **T1=lookupST(rN1,S,CN)}** <P>**{rN2}**|Eps

**T1=lookupST(rN1,S,CN)}**

<DEC> <O\_Static><AM> <DT> ID<M>;

<M> = **{OP=TS[i].VP}** <OE> **{RT=typeCompatibility(rDT,T,OP)}** <M’>|,ID<M>|Eps

<M’> ,ID**{rN=TS[i].VP}{insertCDT(rN,rDT,rAM,,rTM,N)}** <M>|Eps

<O\_Static> static|Eps

<IF\_ST> if(<Cond>){<MST>} <O\_ELIF><O\_ELSE>

<O\_ELIF> elif<IF\_ST>|Eps

<O\_ELSE> else{<MST>}

<COND> <OE>|true|false

<FUNC\_CALL> ID (<ARGS2>);

<ARGS2> Eps|<Q><NEXT2>

<NEXT2> ,<OE><NEXT2>|Eps|,<OE>;|<MST>

<FUNC\_DEF> fun ID (<Argument>):<DT>{<MST>}

<ARGS> Eps|<DT><X1><NEXT>

<NEXT> ,<DT><X1><NEXT>|Eps

<X1> ID<Y1>

<Y1> Eps|[<OE>]<opt>

<ARRAY\_DEC> <DT2>[<E>]<opt> ID<opt2>;

<opt> [<OE>]|Eps

<opt2> ,ID<opt2>|Eps|=new <DT2>(<ARGS2>)

<until> until(<COND>){<MST>}

<do\_until> do{ <MST> } until (<COND>) ;

<OE> <AE><OE’>

<OE’> or <AE><OE’>|Eps

<AE> <RE><AE’>

<AE’> and <RE><AE’>|Eps

<RE> <E><RE’>

<RE’> ROP <E> <RE’>|Eps

<E> <T><X>

<X> PM <T> <X>|Eps

<T> <F> <T’>

<T’> MDM <F> <T’>|Eps

<F> ID<F’>|<CONST><X>|<INC\_DEC> ID <X> |(<OE>)|

not <F>| this.<this’>| super.<super’>

<this’> ID<this\_super’’>

<this\_super’’> <X>|(<ARGS2>);<X>

<super’> ID <this\_super’’>

<F’> <X><F’’>|(<ARGS2>)<X>|[<OE>]<F’’’>|.<P””>=<Q>;

<F’’> <INC\_DEC>|eps

<F’’’> <x>|[<OE>]<X>

<DT> DT<DT’>|ID<DT’>

<DT’> [ ]|[ ][ ]|eps

<DT2> DT|ID

<AM> public|private|protected