MOLIPROCI

GRAMMER DESIGN AND TEST CASE DERIVATIONS

AT

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

IN PARTIAL FULFILMENT OF:

ISF342 Compiler Design/ CSF363 Compiler Construction

SUBMITTED BY

	Group -1	Group -2
1.	Prateek Jain (2012C6PS392P)	Ritwik Sahani (2012C6PS686P)
2.	Vishal Goyal (2012A7PS071P)	Jashan Goyal (2012C6PS612P)

STUDENTS AT

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
(Februray, 2015)

LANGUAGE FEATURES

Mechanics is a branch of science that deals with the action of forces on the physical behavior of a body or bodies. Simulation is the imitation of the operation of a real-world process or system over time. Simulation in mechanics basically refers to modelling all domain values while changing variables in a specific amount.

We opt to develop a similar language which can be used to simulate data for mechanics problem. The main reason to pick this domain is deficiency of any modulated language in this domain which provide the user the power to code the descriptions of problem and through scenario description generate data for studying motion in mechanics. The need to have a language rather than using any software or libraries is that these libraries or softwares will always constraint the user to develop the scenario of the problem in specific directions which will narrow down the scope of exploration in simulating data in scenarios of mechanics. Whereas if there would be a language in this field, user would not have to remain constrained and can freely explore the field. The independence to think, interpret and finally code mechanics will give the users exciting feeling using the language.

One special thing about a language in mechanics is that the user just need to think about phenomena and scenario of the situation neglecting all complexity of calculation. The language can act as a helper which will take care about all exhaustive calculations and the user just need to care about describing the situation.

Mechanics is a wide field where every object interaction has its own interpretation. The problems tackled in this domain are related to the linear motion of an object. Linear motion is a motion of objects in a straight line with constant velocity known as Uniform motion or with variable velocity known as non uniform motion. We focus to provide a language where the user will describe the path for the linear motion and various scenario constraints acting on the objects. The language would be able to simulate data along the whole path of the motion. The user may not desire to view whole simulated data. Instead he may be interested in knowing values at particular instances. The language provides to make constructs where user will be able to produce values at desired points and at desired situations.

Language supports two primitive data types ‘int’ and ‘float’ and three derived data types ‘object’,’linear’ and ‘vector’. Language has basic operators such as ‘+’,’-’,’*’,’/’,’and’,’or’,’not’,’<’,’>’,’<=’,’>=’,’==’,’=>’ (for assignment) and complex operators such as ‘lin’ and ‘frict’. These complex operators are used with ‘linear’ data type only.

The structure of complex data types are:

Vector → {x **:** float, y **:** float}

Object → {velocity **:** vector; acceleration **:** vector; position **:** vector; mass **:** float}

Linear → {start **:** vector; end **:** vector;friction**:**float }

Local Scoping is used in the language. The language also supports nested functions and complex assignment operators used for data types such as ‘object’, ’vector’ and ‘linear’ whereas language supports simple assignment statements for ‘int’ and ‘float’ data types. Language has conditional statements, iterative statements, I/O operations, expressions, declaration and assignment statements. Arrays for all data types are allowed in the language. Derived data types are passed by reference whereas primitive data types are passed by value. Every function expect ‘_init_ ’ has to be of some return type. Every program needs to have one ‘_init_’ function.

LEXICAL UNITS

Pattern	Token	Purpose
=>	ASSIGN	Assignment operator
+	ADD	add operator
-	SUB	subtract operator
*	MUL	multiply operator
/	DIV	division operator
and	AND	logical and operator
or	OR	logical or operator
not	NOT	logical not operator
lin	LIN	linear path assignment
frict	FRICT	friction assignment
[1-9][0-9]*	INTCONST	Integer number
[0-9]+.[0-9]+	FLOATCONST	float number
([a-z] | [A-Z]) ([a-z][A-Z][0-9])*	ID	Identifier
,	COMMA	vector component seperator
;	SMC	value seperator in a list
)	RP	closing parentheses in function call
(	LP	opening parentheses in function call
}	RC	closing parentheses in object value assignment
{	LC	opening parentheses in object value assignment
]	RS	closing parentheses in array statements
[	LS	opening parentheses in array statements
at	AT	conditional statement for output statement
when	WHEN	conditional statement
.	DOT	accessing structure variables
loop	LOOP	loop beginning statement
init	INIT	main function identifier
==	EC	equality comparison
<	LT	less than comparison
>	GT	greator comparison
<=	LTE	less than or equal to comparison
>=	GTE	greator than or equal to comparison
<<	FS	function body starts
>>	FE	function body ends
xpos	XP	x keyword of vector structure
ypos	YP	y keyword of vector structure
pos	POS	position keyword
"!!	COM	Comment
vel	VEL	velocity keyword of object
acc	ACC	acceleration keyword of object
mass	MAS	mass keyword of object
startpos	SP	start position keyword of object
endpos	EP	end position keyword of object
int	INT	int datatype
float	FLOAT	float datatype
object	OBJECT	object datatype
vector	VECTOR	vector datatype
linear	LNR	linear datatype
out	OUT	output statement for one variable
out_all	OUTA	output statement for all variables
_([a-z]|[A-Z]|[0-9]) +	funcID	function name
([a-z][A-Z][0-9_][./\{}@#$%^&*():””<>.?|])*	STRING	String used in comments
t	TIME	time with respect to frame
EOL	SEP	seperator between each statement
E	E	Epsilon
return	RETURN	return keyword
in_inp	ININ	input keyword for integer values
flo_inp	INF	input keyword for float values

GRAMMAR

<SEPA> → SEP | E
<program> → <function><SEPA><program> | E
<function> → INIT LP RP <SEPA>FS<SEPA> <functionBody><SEPA> FE
<function> → <dataType>ID LP <paramList> RP<SEPA> FS <SEPA><functionBody><SEPA> FE
<paramList> → <dataType>ID<paramList2> | E
<paramList2> → SMC<dataType>ID<paramList2>|E
<paramListN> → ID<paramListN2>|E
<paramListN2> → SMC ID <paramList2>|E
<dataType> → INT
<dataType> → FLOAT
<dataType> → OBJECT
<dataType> → VECTOR
<dataType> → LINEAR
<functionBody> → <stnt>SEP<functionBody>|E
<stnt> → <declStnt><C>
<stnt> → <assignPlusFunct><C>
<stnt> → <condStnt><C>
<stnt> → <loop><C>
<stnt> → <returnCall><C>
<stnt> → <outStnt><C>
<stnt> → <inStnt><C>
<stnt> → <C>
<stnt> → <function>
<C> → <comment>|E
<comment> → COM STRING
<furID> → SMC ID<furID>|E
<declStnt> → OBJECT ID <o>
<o>→ <o2><objAssign>
<o> → <furID>
<o2> → LS <intVal> RS | E
<objAssign> → ASSIGN<objVal>
<objVal> → LC<vecAss>SMC<vecAss>SMC<vecAss>SMC<floatVal> RC
<objVal> → ID<objs>
<objs> → <funcCall>|E
<declStnt> → LINEAR ID <l>
<l>→ <l2><linAssign>
<l> → <furID>
<l2> → LS <intVal> RS | E
<linAssign> → ASSIGN<linVal>
<linVal> → <velVal>LIN<vecVal><furLin><Z>
<linVal> → ID <lins><Z>
<lins> → <funcCall>|E
<Z> → ADD<linVal>|E
<furLin> → FRICT LP <floatVal>RP|E
<declStnt> → VECTOR ID <v>
<v> → <v2><vectorAssign>
<v> → <furID>
<v2> → LS <intVal> RS | E
<vectorAssign> → ASSIGN <vecVal>
<vecVal> → LC <vecAss> RC
<vecVal> → ID<vecs>
<vecs> → <funcCall>|E
<vecAss> → <floatVal> COMMA <floatVal>
<declStnt> → FLOAT ID <f>
<f>→ <f2><floatAssign>
<f> → <furID>
<f2> → LS <intVal> RS | E
<floatAssign> → ASSIGN <floatVal>
<floatVal> → <floatexpr>
<declStnt> →INT ID <i>
<i>→ <i2><intAssign>
<i> → <furID>
<i2> → LS <intVal> RS | E
<intAssign> → ASSIGN <intVal>
<intVal> → <intexpr>
<assignPlusFunct> → <ID><rest>
<ID> → ID<ARR>
<rest> → ASSIGN <assign>
<rest> → LP<paramlistN>RP
<ARR> → LS<intVal>RS |E
<assign> → <vectorVal>
<assign> → <linVal>
<assign> → <objVal>
<assign> → <floatVal>
<assign> → <intVal>
<condStnt> → FS <functionBody> FE WHEN LP<condCheck>RP
<condCheck> → <not><woCheck>
<woCheck> → LP<furCond>RP
<woCheck> → <furCond>
<furCond> → <intVal><condOp><intVal><condCheck2> //check if LP RP after not
<furCond> → <floatVal><condOp><floatVal><condCheck2>

<condCheck2> → <condcon><woCheck2>
<woCheck2> → LP<condCheck>RP| E
<woCheck2> → <condCheck>|E
<not> → NOT | E
<condOp> → EC
<condOp> →LT
<condOp> →GT
<condOp> →LTE
<condOp> → GTE
<condcon> → AND
<condcon> → OR
<loop> → LOOP LP <condCheck>RP<SEPA>FS<SEPA><functionBody><SEPA>FE<SEPA>
<funcCall> → LP <paramlistN> RP
<returnCall> → RETURN <val>
<val> → <intVal>
<val> → <floatVal>
<val> → <objVal>
<val> → <linVal>
<val> → <vectorVal>
<outStnt> → <out>
<outStnt> → <outall>
<out> → OUT LP <varList> RP AT LP <condCheck> RP
<varList> → ID
<varList> → TIME
<outall> → OUTA LP <objectList> RP AT LP <condCheck> RP
<objectList> → <ObjVal>
<objectList> →<linVal>
<objectList> →<vecVal>
<inStnt> → <inFloat>LP ID RP
<inStnt> → <inInt>LP ID RP
<inFloat> → INF
<inInt> → ININ
<floatexpr> → <floatmulexpr><floatM>
<floatM> → <addOP> <floatexpr> | E
<addOP> → ADD
<addOP> → SUB
<floatmulexpr> → <floatfactor><floatF>
<floatF> → <mulOP><floatmulexpr> | E
<mulOP> → MUL
<mulOP> → DIV
<floatfactor> → LP <floatexpr> RP
<floatfactor> → ID <flos>
<flos> → E| <funcCall>
<floatfactor> → FLOATCONST
<floatfactor> → TIME
<intexpr> → <intmulexpr><intM>
<intM> → <addOP><intexpr> | E
<intmulexpr> → <intfactor><intF>
<intF> → <mulOP> <intmulexpr> | E
<intfactor> →LP <intexpr> RP
<intfactor> → <intos>
<intos> → E|<funcCall>
<intfactor> → <intConst>
<intfactor> → INTCONST

TEST CASES

CODE-1

```
_init_ ()
<<
object o1
o1 => {2.3,1.1;2.0,8.1;2.0,0.0;9.3}
int a => 10
loop(a<=13)<<
a=>a+1
>>
>>
```

CODE-2

```
_init_()
<<
object o1 => {1.0,2.0;1.0,2.0;1.0,2.0;2.1}
object o2 => o1
out_all(o1) at (t==1)
>>
```

CODE-3

```
_init_()
<<
object o1 => {1.0,2.0;1.0,2.0;1.0,2.0;2.1}
float m=>2.9
<< o1.mass=o1.mass+m
>> when(t==5)
out_all(o1) at (t==10)
>>
```

CODE-4

```
float rmc(int a,int b)
<<
float x => 23.9
return x
>>
_init_()<<
rmc(2;1)
>>
```

CODE-5

```
_init_()
<<
linear l1 => {0.0,0.0}lin{10.0,0.0}
out(l1.vel.xpos) at (pos.xpos==5)
>>
```

DERIVATION FOR TEST CASES

CODE-1

```
_init_ ()
<<
object o1
o1 => {2.3,1.1;2.0,8.1;2.0,0.0;9.3}
int a => 10
loop(a<=13)<<
a=>a+1
>>
>>
```

Derivation for code 1

<program> --> <function> <SEPA> <program>

<program> --> INIT LP RP <SEPA>FS<SEPA> <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS <SEPA> <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <stnt>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <declStnt><C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 <o> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP <stnt> SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP <assignplusfunct> SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP <ID> <rest> SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 <rest> SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN <assign> SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN <objval> SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC<vecAss>SMC<vecAss>SMC<vecAss>SMC<floatVal> RC SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC <floatVal> COMMA <floatVal> SMC <floatVal> COMMA <floatVal> SMC <floatVal> COMMA <floatVal> SMC<floatVal> RC SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP <declStnt> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID <i2><intAssign> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID E ASSIGN <intVal> SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID E ASSIGN <intexpr> SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID E ASSIGN 10 SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID E ASSIGN 10 SEP <stnt> SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID E ASSIGN 10 SEP <loop><C> SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID E ASSIGN 10 SEP LOOP LP <condCheck> RP <SEPA> FS <SEPA> <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT ID E ASSIGN 10 SEP LOOP LP <not> <woCheck> RP <SEPA> FS <SEPA> <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E <furCond> RP <SEPA> FS <SEPA> <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E <intVal> <condOp> <intVal> <condCheck2> RP <SEPA> FS <SEPA> <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E ID LTE INTCONST <condCheck2> RP <SEPA> FS <SEPA> <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> <stnt> SEP <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> ID ASSIGN <intexpr> SEP <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> a ASSIGN <intmulexpr><intM> SEP <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> a ASSIGN <intFactor> E <intM> SEP <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> a ASSIGN ID E <addOp> <intExpr> SEP <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> a ASSIGN a E ADD 10 SEP <functionBody><SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> a ASSIGN a E ADD 10 SEP E <SEPA>FE<SEPA> E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP <SEPA> FS <SEPA> a ASSIGN a E ADD 10 SEP E E FE E E SEP <functionBody> <SEPA> FE <SEPA>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a E ASSIGN 10 SEP LOOP LP E a LTE 13 E RP SEP FS SEP a ASSIGN a E ADD 1 SEP E E FE E E SEP E SEP FE E

<program> --> INIT LP RP SEP FS SEP OBJECT o1 SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a ASSIGN 10 SEP LOOP LP a LTE 13 RP SEP FS SEP a ASSIGN a ADD 1 SEP FE SEP SEP FE

<program> --> INIT LP RP SEP FS SEP OBJECT o1 SEP o1 ASSIGN LC 2.3 COMMA 1.1 SMC 2.0 COMMA 8.1 SMC 2.0 COMMA 0.0 SMC 9.3 RC SEP INT a ASSIGN 10 SEP LOOP LP a LTE 13 RP SEP FS SEP a ASSIGN a ADD 1 SEP FE SEP SEP FE

CODE-2

```
_init_()
<<
object o1 => {1.0,2.0;1.0,2.0;1.0,2.0;2.1}
object o2 => o1
out_all(o1) at (t==1)
>>
```

Derivation for code 2

<program> --> <function> <SEPA> <program>

<program> --> INIT LP RP <SEPA>FS<SEPA> <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS <SEPA> <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <stnt>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <declStnt><C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 <o> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 <o2><objASSIGN> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN <objVal> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC <floatVal> COMMA <floatVal> SMC <floatVal> COMMA <floatVal> SMC <floatVal> COMMA <floatVal> SMC <floatVal> RC <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP <stnt> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP <declStnt> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT ID <o> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 <o2><objAssign> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN <objVal> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN ID E SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN o1 E SEP <outStnt>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN o1 E SEP <outall> SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN o1 E SEP OUTA LP <objVal> RP AT LP <varList> EC <floatVal> RP SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN o1 E SEP OUTA LP ID RP AT LP TIME EC FLOATCONST RP SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN o1 E SEP OUTA LP o1 RP AT LP TIME EC 1.0 RP SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN o1 E SEP OUTA LP o1 RP AT LP TIME EC 1.0 RP SEP E E FE SEP <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP OBJECT o2 E ASSIGN o1 E SEP OUTA LP o1 RP AT LP TIME EC 1.0 RP SEP E E FE SEP E

<program> --> INIT LP RP SEP FS SEP OBJECT o1 ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC SEP OBJECT o2 ASSIGN o1 SEP OUTA LP o1 RP AT LP TIME EC 1.0 RP SEP FE SEP

CODE-3

```
_init_()
<<
object o1 => {1.0,2.0;1.0,2.0;1.0,2.0;2.1}
float m=>2.9
<< o1.mass=o1.mass+m
>> when(t==5)
out_all(o1) at (t==10)
>>
```

Derivation for Code 3

<program> --> <function> <SEPA> <program>

<program> --> INIT LP RP <SEPA>FS<SEPA> <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS <SEPA> <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <stnt>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP <declStnt><C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 <o> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 <o2><objASSIGN> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 <o2><objASSIGN> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN <objVal> <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC <floatVal> COMMA <floatVal> SMC <floatVal> COMMA <floatVal> SMC <floatVal> COMMA <floatVal> SMC <floatVal> RC   <C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP <stnt> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP <declStnt> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT ID <f> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m <f2><floatAssign> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN <floatexpr> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN <floatmulexpr><floatM> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN <floatfactor> <floatF> <floatM> <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E <C> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP <stnt> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP <condStnt> SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS <stnt>SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS <assignPlusFunct>SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT ID E ASSIGN FLOATCONST E E E SEP FS <ID><rest> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS ID <ARR > <rest> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS ID E ASSIGN <assign> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC   E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN <floatVal> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN <floatexpr> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN <floatmulexpr> <floatM> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN <floatfactor> <floatF> <floatM> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E <floatM> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E <addOP> <floatexpr> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD <floatmulexpr><floatM> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD ID <floatM> SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP <functionBody> FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP <condCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP <not><woCheck> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E<furCond> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E<intVal><condOp><floatVal><condCheck2> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME <condOp><floatVal><condCheck2> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC <floatVal><condCheck2> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST <condCheck2> RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP <functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP <stnt>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP <outStnt><C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP <outall><C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP OUTA LP <objectList> RP AT LP <varList> EC <floatVal> RP<C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP OUTA LP <objVal> RP AT LP <varList> EC <floatVal> RP<C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP OUTA LP ID RP AT LP TIME EC <floatVal> RP<C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP OUTA LP ID RP AT LP TIME EC FLOATCONST RP<C>SEP<functionBody> <SEPA> FE <SEPA> <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP OUTA LP ID RP AT LP TIME EC FLOATCONST RP E SEP E <SEPA> FE <SEPA> <program>

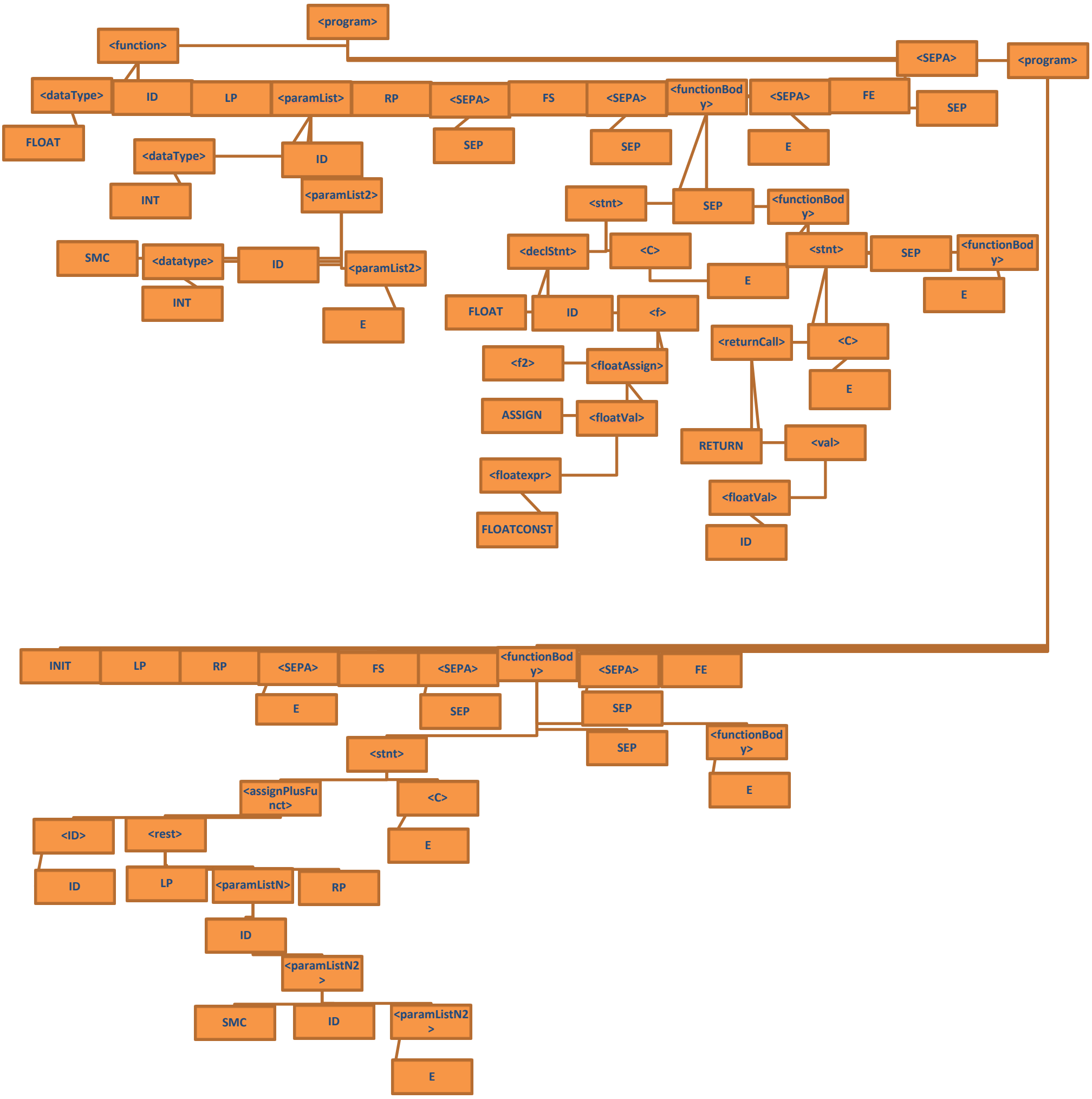
<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP OUTA LP ID RP AT LP TIME EC FLOATCONST RP E SEP E E FE E <program>

<program> --> INIT LP RP SEP FS SEP OBJECT o1 E ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC E SEP FLOAT m E ASSIGN FLOATCONST E E E SEP FS o1.mass E ASSIGN ID E ADD m E SEP E FE WHEN LP E TIME EC FLOATCONST E RP SEP OUTA LP ID RP AT LP TIME EC FLOATCONST RP E SEP E E FE E E

<program> --> INIT LP RP SEP FS SEP OBJECT o1 ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC SEP FLOAT m ASSIGN FLOATCONST SEP FS o1.mass ASSIGN ID ADD m SEP FE WHEN LP TIME EC FLOATCONST RP SEP OUTA LP ID RP AT LP TIME EC FLOATCONST RP SEP FE

<program> --> INIT LP RP SEP FS SEP OBJECT o1 ASSIGN LC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 1.0 COMMA 2.0 SMC 2.1 RC SEP FLOAT m ASSIGN FLOATCONST SEP FS o1.mass ASSIGN o1.mass ADD m SEP FE WHEN LP TIME EC FLOATCONST RP SEP OUTA LP o1 RP AT LP TIME EC FLOATCONST RP SEP FE

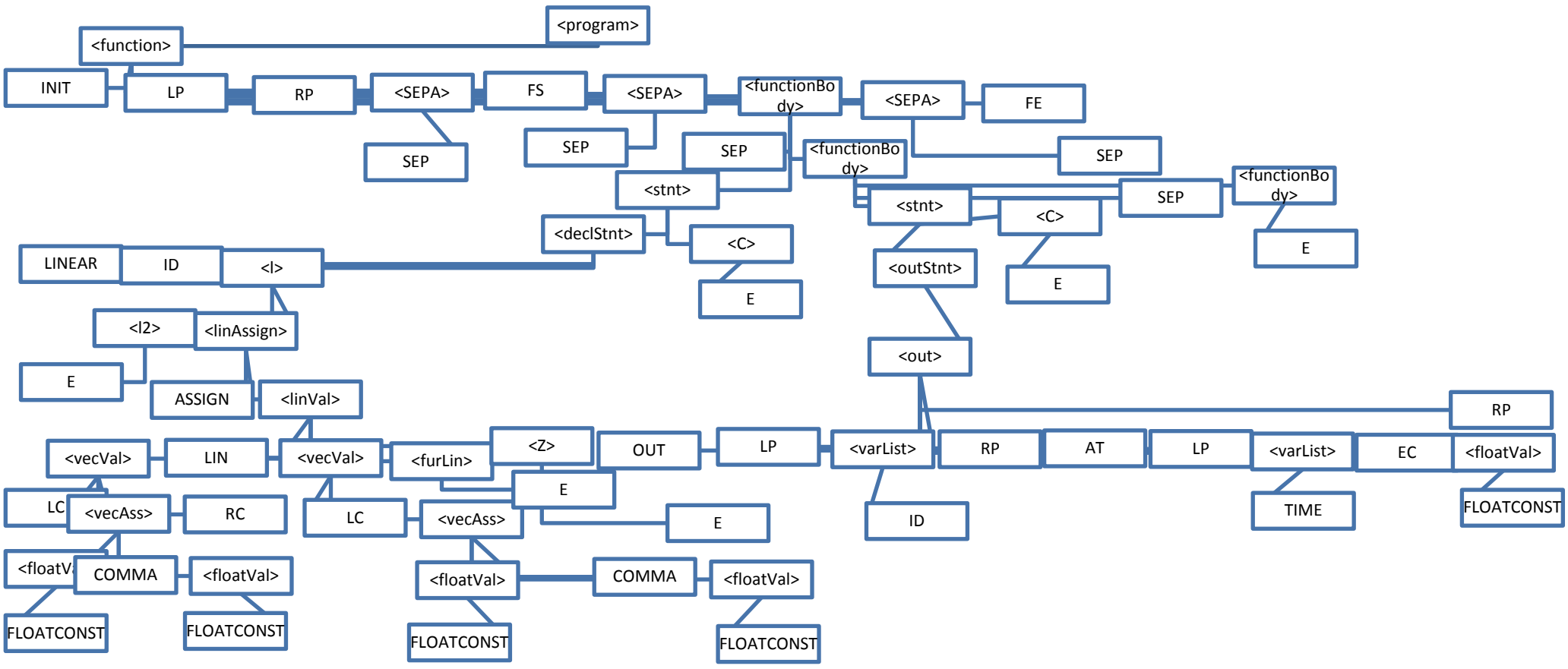
Derivation For Test Case 4



Code – 4

```
float rmc(int a,int b)
<<
float x => 23.9
return x
>>
_init_()<<
rmc(2;1)
>>
```

Derivation For Test Case 5



CODE-5

```
_init_()
<<
linear l1 => {0.0,0.0}lin{10.0,0.0}
out(l1.vel.xpos) at (pos.xpos==5)
>>
```