Running semantic checker on good.cl

./mysemant good.cl

#38

\_program

#38

\_class

Graph

Object

"good.cl"

(

#40

\_attr

vertices

VList

#40

\_new

VList

: VList

#41

\_attr

edges

EList

#41

\_new

EList

: EList

#43

\_method

add\_vertice

#43

\_formal

v

Vertice

Object

#43

\_block

#44

\_assign

edges

#44

\_dispatch

#44

\_dispatch

#44

\_object

v

: Vertice

outgoing

(

)

: EList

append

(

#44

\_object

edges

: EList

)

: EList

: EList

#45

\_assign

vertices

#45

\_dispatch

#45

\_object

vertices

: VList

cons

(

#45

\_object

v

: Vertice

)

: VList

: VList

: VList

#48

\_method

print\_E

Object

#48

\_dispatch

#48

\_object

edges

: EList

print

(

)

: Object

#49

\_method

print\_V

Object

#49

\_dispatch

#49

\_object

vertices

: VList

print

(

)

: Object

)

#53

\_class

Vertice

IO

"good.cl"

(

#55

\_attr

num

Int

#0

\_no\_expr

: \_no\_type

#56

\_attr

out

EList

#56

\_new

EList

: EList

#58

\_method

outgoing

EList

#58

\_object

out

: EList

#60

\_method

number

Int

#60

\_object

num

: Int

#62

\_method

init

#62

\_formal

n

Int

SELF\_TYPE

#63

\_block

#64

\_assign

num

#64

\_object

n

: Int

: Int

#65

\_object

self

: SELF\_TYPE

: SELF\_TYPE

#70

\_method

add\_out

#70

\_formal

s

Edge

SELF\_TYPE

#71

\_block

#72

\_assign

out

#72

\_dispatch

#72

\_object

out

: EList

cons

(

#72

\_object

s

: Edge

)

: EList

: EList

#73

\_object

self

: SELF\_TYPE

: SELF\_TYPE

#77

\_method

print

Object

#78

\_block

#79

\_dispatch

#79

\_object

self

: SELF\_TYPE

out\_int

(

#79

\_object

num

: Int

)

: SELF\_TYPE

#80

\_dispatch

#80

\_object

out

: EList

print

(

)

: Object

: Object

)

#86

\_class

Edge

IO

"good.cl"

(

#88

\_attr

from

Int

#0

\_no\_expr

: \_no\_type

#89

\_attr

to

Int

#0

\_no\_expr

: \_no\_type

#90

\_attr

weight

Int

#0

\_no\_expr

: \_no\_type

#92

\_method

init

#92

\_formal

f

Int

#92

\_formal

t

Int

#92

\_formal

w

Int

SELF\_TYPE

#93

\_block

#94

\_assign

from

#94

\_object

f

: Int

: Int

#95

\_assign

to

#95

\_object

t

: Int

: Int

#96

\_assign

weight

#96

\_object

w

: Int

: Int

#97

\_object

self

: SELF\_TYPE

: SELF\_TYPE

#101

\_method

print

Object

#102

\_block

#103

\_dispatch

#103

\_object

self

: SELF\_TYPE

out\_string

(

#103

\_string

" ("

: String

)

: SELF\_TYPE

#104

\_dispatch

#104

\_object

self

: SELF\_TYPE

out\_int

(

#104

\_object

from

: Int

)

: SELF\_TYPE

#105

\_dispatch

#105

\_object

self

: SELF\_TYPE

out\_string

(

#105

\_string

","

: String

)

: SELF\_TYPE

#106

\_dispatch

#106

\_object

self

: SELF\_TYPE

out\_int

(

#106

\_object

to

: Int

)

: SELF\_TYPE

#107

\_dispatch

#107

\_object

self

: SELF\_TYPE

out\_string

(

#107

\_string

")"

: String

)

: SELF\_TYPE

#108

\_dispatch

#108

\_object

self

: SELF\_TYPE

out\_int

(

#108

\_object

weight

: Int

)

: SELF\_TYPE

: SELF\_TYPE

)

#116

\_class

EList

IO

"good.cl"

(

#119

\_attr

car

Edge

#0

\_no\_expr

: \_no\_type

#121

\_method

isNil

Bool

#121

\_bool

1

: Bool

#123

\_method

head

Edge

#123

\_block

#123

\_dispatch

#123

\_object

self

: SELF\_TYPE

abort

(

)

: Object

#123

\_object

car

: Edge

: Edge

#125

\_method

tail

EList

#125

\_block

#125

\_dispatch

#125

\_object

self

: SELF\_TYPE

abort

(

)

: Object

#125

\_object

self

: SELF\_TYPE

: SELF\_TYPE

#134

\_method

cons

#134

\_formal

e

Edge

EList

#135

\_dispatch

#135

\_new

ECons

: ECons

init

(

#135

\_object

e

: Edge

#135

\_object

self

: SELF\_TYPE

)

: EList

#138

\_method

append

#138

\_formal

l

EList

EList

#139

\_cond

#139

\_dispatch

#139

\_object

self

: SELF\_TYPE

isNil

(

)

: Bool

#139

\_object

l

: EList

#140

\_dispatch

#140

\_dispatch

#140

\_dispatch

#140

\_object

self

: SELF\_TYPE

tail

(

)

: EList

append

(

#140

\_object

l

: EList

)

: EList

cons

(

#140

\_dispatch

#140

\_object

self

: SELF\_TYPE

head

(

)

: Edge

)

: EList

: EList

#144

\_method

print

Object

#145

\_dispatch

#145

\_object

self

: SELF\_TYPE

out\_string

(

#145

\_string

"\n"

: String

)

: SELF\_TYPE

)

#164

\_class

ECons

EList

"good.cl"

(

#166

\_attr

cdr

EList

#0

\_no\_expr

: \_no\_type

#168

\_method

isNil

Bool

#168

\_bool

0

: Bool

#170

\_method

head

Edge

#170

\_object

car

: Edge

#172

\_method

tail

EList

#172

\_object

cdr

: EList

#174

\_method

init

#174

\_formal

e

Edge

#174

\_formal

rest

EList

EList

#175

\_block

#176

\_assign

car

#176

\_object

e

: Edge

: Edge

#177

\_assign

cdr

#177

\_object

rest

: EList

: EList

#178

\_object

self

: SELF\_TYPE

: SELF\_TYPE

#182

\_method

print

Object

#183

\_block

#184

\_dispatch

#184

\_object

car

: Edge

print

(

)

: Object

#185

\_dispatch

#185

\_object

cdr

: EList

print

(

)

: Object

: Object

)

#194

\_class

VList

IO

"good.cl"

(

#197

\_attr

car

Vertice

#0

\_no\_expr

: \_no\_type

#199

\_method

isNil

Bool

#199

\_bool

1

: Bool

#201

\_method

head

Vertice

#201

\_block

#201

\_dispatch

#201

\_object

self

: SELF\_TYPE

abort

(

)

: Object

#201

\_object

car

: Vertice

: Vertice

#203

\_method

tail

VList

#203

\_block

#203

\_dispatch

#203

\_object

self

: SELF\_TYPE

abort

(

)

: Object

#203

\_object

self

: SELF\_TYPE

: SELF\_TYPE

#212

\_method

cons

#212

\_formal

v

Vertice

VList

#213

\_dispatch

#213

\_new

VCons

: VCons

init

(

#213

\_object

v

: Vertice

#213

\_object

self

: SELF\_TYPE

)

: VList

#216

\_method

print

Object

#216

\_dispatch

#216

\_object

self

: SELF\_TYPE

out\_string

(

#216

\_string

"\n"

: String

)

: SELF\_TYPE

)

#221

\_class

VCons

VList

"good.cl"

(

#223

\_attr

cdr

VList

#0

\_no\_expr

: \_no\_type

#225

\_method

isNil

Bool

#225

\_bool

0

: Bool

#227

\_method

head

Vertice

#227

\_object

car

: Vertice

#229

\_method

tail

VList

#229

\_object

cdr

: VList

#231

\_method

init

#231

\_formal

v

Vertice

#231

\_formal

rest

VList

VList

#232

\_block

#233

\_assign

car

#233

\_object

v

: Vertice

: Vertice

#234

\_assign

cdr

#234

\_object

rest

: VList

: VList

#235

\_object

self

: SELF\_TYPE

: SELF\_TYPE

#239

\_method

print

Object

#240

\_block

#241

\_dispatch

#241

\_object

car

: Vertice

print

(

)

: Object

#242

\_dispatch

#242

\_object

cdr

: VList

print

(

)

: Object

: Object

)

#249

\_class

Parse

IO

"good.cl"

(

#252

\_attr

boolop

BoolOp

#252

\_new

BoolOp

: BoolOp

#256

\_method

read\_input

Graph

#258

\_let

g

Graph

#258

\_new

Graph

: Graph

#258

\_block

#259

\_let

line

String

#259

\_dispatch

#259

\_object

self

: SELF\_TYPE

in\_string

(

)

: String

#260

\_loop

#260

\_dispatch

#260

\_object

boolop

: BoolOp

and

(

#260

\_comp

#260

\_eq

#260

\_object

line

: String

#260

\_string

"\n"

: String

: Bool

: Bool

#260

\_comp

#260

\_eq

#260

\_object

line

: String

#260

\_string

""

: String

: Bool

: Bool

)

: Bool

#260

\_block

#263

\_dispatch

#263

\_object

g

: Graph

add\_vertice

(

#263

\_dispatch

#263

\_object

self

: SELF\_TYPE

parse\_line

(

#263

\_object

line

: String

)

: Vertice

)

: Object

#264

\_assign

line

#264

\_dispatch

#264

\_object

self

: SELF\_TYPE

in\_string

(

)

: String

: String

: String

: Object

: Object

#267

\_object

g

: Graph

: Graph

: Graph

#272

\_method

parse\_line

#272

\_formal

s

String

Vertice

#273

\_let

v

Vertice

#273

\_dispatch

#273

\_new

Vertice

: Vertice

init

(

#273

\_dispatch

#273

\_object

self

: SELF\_TYPE

a2i

(

#273

\_object

s

: String

)

: Int

)

: Vertice

#273

\_block

#274

\_loop

#274

\_comp

#274

\_eq

#274

\_dispatch

#274

\_object

rest

: String

length

(

)

: Int

#274

\_int

0

: Int

: Bool

: Bool

#274

\_block

#277

\_let

succ

Int

#277

\_dispatch

#277

\_object

self

: SELF\_TYPE

a2i

(

#277

\_object

rest

: String

)

: Int

#278

\_let

weight

Int

#278

\_dispatch

#278

\_object

self

: SELF\_TYPE

a2i

(

#278

\_object

rest

: String

)

: Int

#280

\_dispatch

#280

\_object

v

: Vertice

add\_out

(

#280

\_dispatch

#280

\_new

Edge

: Edge

init

(

#280

\_dispatch

#280

\_object

v

: Vertice

number

(

)

: Int

#281

\_object

succ

: Int

#282

\_object

weight

: Int

)

: Edge

)

: Vertice

: Vertice

: Vertice

: Vertice

: Object

#285

\_object

v

: Vertice

: Vertice

: Vertice

#290

\_method

c2i

#290

\_formal

char

String

Int

#291

\_cond

#291

\_eq

#291

\_object

char

: String

#291

\_string

"0"

: String

: Bool

#291

\_int

0

: Int

#292

\_cond

#292

\_eq

#292

\_object

char

: String

#292

\_string

"1"

: String

: Bool

#292

\_int

1

: Int

#293

\_cond

#293

\_eq

#293

\_object

char

: String

#293

\_string

"2"

: String

: Bool

#293

\_int

2

: Int

#294

\_cond

#294

\_eq

#294

\_object

char

: String

#294

\_string

"3"

: String

: Bool

#294

\_int

3

: Int

#295

\_cond

#295

\_eq

#295

\_object

char

: String

#295

\_string

"4"

: String

: Bool

#295

\_int

4

: Int

#296

\_cond

#296

\_eq

#296

\_object

char

: String

#296

\_string

"5"

: String

: Bool

#296

\_int

5

: Int

#297

\_cond

#297

\_eq

#297

\_object

char

: String

#297

\_string

"6"

: String

: Bool

#297

\_int

6

: Int

#298

\_cond

#298

\_eq

#298

\_object

char

: String

#298

\_string

"7"

: String

: Bool

#298

\_int

7

: Int

#299

\_cond

#299

\_eq

#299

\_object

char

: String

#299

\_string

"8"

: String

: Bool

#299

\_int

8

: Int

#300

\_cond

#300

\_eq

#300

\_object

char

: String

#300

\_string

"9"

: String

: Bool

#300

\_int

9

: Int

#301

\_block

#301

\_dispatch

#301

\_object

self

: SELF\_TYPE

abort

(

)

: Object

#301

\_int

0

: Int

: Int

: Int

: Int

: Int

: Int

: Int

: Int

: Int

: Int

: Int

: Int

#305

\_attr

rest

String

#0

\_no\_expr

: \_no\_type

#307

\_method

a2i

#307

\_formal

s

String

Int

#308

\_cond

#308

\_eq

#308

\_dispatch

#308

\_object

s

: String

length

(

)

: Int

#308

\_int

0

: Int

: Bool

#308

\_int

0

: Int

#309

\_cond

#309

\_eq

#309

\_dispatch

#309

\_object

s

: String

substr

(

#309

\_int

0

: Int

#309

\_int

1

: Int

)

: String

#309

\_string

"-"

: String

: Bool

#309

\_neg

#309

\_dispatch

#309

\_object

self

: SELF\_TYPE

a2i\_aux

(

#309

\_dispatch

#309

\_object

s

: String

substr

(

#309

\_int

1

: Int

#309

\_sub

#309

\_dispatch

#309

\_object

s

: String

length

(

)

: Int

#309

\_int

1

: Int

: Int

)

: String

)

: Int

: Int

#310

\_cond

#310

\_eq

#310

\_dispatch

#310

\_object

s

: String

substr

(

#310

\_int

0

: Int

#310

\_int

1

: Int

)

: String

#310

\_string

" "

: String

: Bool

#310

\_dispatch

#310

\_object

self

: SELF\_TYPE

a2i

(

#310

\_dispatch

#310

\_object

s

: String

substr

(

#310

\_int

1

: Int

#310

\_sub

#310

\_dispatch

#310

\_object

s

: String

length

(

)

: Int

#310

\_int

1

: Int

: Int

)

: String

)

: Int

#311

\_dispatch

#311

\_object

self

: SELF\_TYPE

a2i\_aux

(

#311

\_object

s

: String

)

: Int

: Int

: Int

: Int

#321

\_method

a2i\_aux

#321

\_formal

s

String

Int

#322

\_let

int

Int

#322

\_int

0

: Int

#323

\_block

#324

\_let

j

Int

#324

\_dispatch

#324

\_object

s

: String

length

(

)

: Int

#325

\_let

i

Int

#325

\_int

0

: Int

#326

\_loop

#326

\_lt

#326

\_object

i

: Int

#326

\_object

j

: Int

: Bool

#327

\_let

c

String

#327

\_dispatch

#327

\_object

s

: String

substr

(

#327

\_object

i

: Int

#327

\_int

1

: Int

)

: String

#328

\_cond

#328

\_eq

#328

\_object

c

: String

#328

\_string

" "

: String

: Bool

#329

\_block

#330

\_assign

rest

#330

\_dispatch

#330

\_object

s

: String

substr

(

#330

\_plus

#330

\_object

i

: Int

#330

\_int

1

: Int

: Int

#330

\_sub

#330

\_sub

#330

\_dispatch

#330

\_object

s

: String

length

(

)

: Int

#330

\_object

i

: Int

: Int

#330

\_int

1

: Int

: Int

)

: String

: String

#331

\_assign

i

#331

\_object

j

: Int

: Int

: Int

#333

\_cond

#333

\_eq

#333

\_object

c

: String

#333

\_string

","

: String

: Bool

#334

\_block

#335

\_assign

rest

#335

\_dispatch

#335

\_object

s

: String

substr

(

#335

\_plus

#335

\_object

i

: Int

#335

\_int

1

: Int

: Int

#335

\_sub

#335

\_sub

#335

\_dispatch

#335

\_object

s

: String

length

(

)

: Int

#335

\_object

i

: Int

: Int

#335

\_int

1

: Int

: Int

)

: String

: String

#336

\_assign

i

#336

\_object

j

: Int

: Int

: Int

#339

\_block

#340

\_assign

int

#340

\_plus

#340

\_mul

#340

\_object

int

: Int

#340

\_int

10

: Int

: Int

#340

\_dispatch

#340

\_object

self

: SELF\_TYPE

c2i

(

#340

\_dispatch

#340

\_object

s

: String

substr

(

#340

\_object

i

: Int

#340

\_int

1

: Int

)

: String

)

: Int

: Int

: Int

#341

\_assign

i

#341

\_plus

#341

\_object

i

: Int

#341

\_int

1

: Int

: Int

: Int

#342

\_cond

#342

\_eq

#342

\_object

i

: Int

#342

\_object

j

: Int

: Bool

#342

\_assign

rest

#342

\_string

""

: String

: String

#342

\_string

""

: String

: String

: String

: Int

: Int

: Int

: Object

: Object

: Object

#349

\_object

int

: Int

: Int

: Int

)

#357

\_class

Main

Parse

"good.cl"

(

#359

\_attr

g

Graph

#359

\_dispatch

#359

\_object

self

: SELF\_TYPE

read\_input

(

)

: Graph

#361

\_method

main

Object

#362

\_block

#363

\_dispatch

#363

\_object

g

: Graph

print\_V

(

)

: Object

#364

\_dispatch

#364

\_object

g

: Graph

print\_E

(

)

: Object

: Object

)

#370

\_class

BoolOp

Object

"good.cl"

(

#372

\_method

and

#372

\_formal

b1

Bool

#372

\_formal

b2

Bool

Bool

#373

\_cond

#373

\_object

b1

: Bool

#373

\_object

b2

: Bool

#373

\_bool

0

: Bool

: Bool

#377

\_method

or

#377

\_formal

b1

Bool

#377

\_formal

b2

Bool

Bool

#378

\_cond

#378

\_object

b1

: Bool

#378

\_bool

1

: Bool

#378

\_object

b2

: Bool

: Bool

)

Running semantic checker on bad.cl

./mysemant bad.cl

Class Main is not defined.

bad.cl:41: Attribute a0 cant be overriden in parents class

bad.cl:41: Attribute a0 cant be overriden in parents class

bad.cl:41: Attribute a1 cant be overriden in parents class

bad.cl:41: Attribute a2 cant be overriden in parents class

bad.cl:58: In redefined method method0, parameter type String is different from original type Int.

bad.cl:59: In redefined method method1, parameter type String is different from original type Int.

bad.cl:88: Assignment to undeclared variable notattr.

bad.cl:90: Assignment to undeclared variable self.

bad.cl:94: Expression type String does not conform to declared static dispatch type Int.

bad.cl:96: Expression type String does not conform to declared static dispatch type SELF\_TYPE.

bad.cl:98: Dispatch to undefined method notMethod.

bad.cl:100: In call of method concat, type Int of parameter arg does not conform to declared type String.

bad.cl:100: Method concat called with wrong number of arguments.

bad.cl:102: Expression type Object does not conform to declared static dispatch type String.

bad.cl:107: Undeclared identifier notattr.

bad.cl:107: Dispatch to undefined method concat.

bad.cl:109: Dispatch to undefined method notMethod.

bad.cl:115: Predicate of 'if' does not have type Bool.

bad.cl:117: Undeclared identifier notattr.

bad.cl:122: Loop condition does not have type Bool.

bad.cl:124: Undeclared identifier notattr.

bad.cl:130: Undeclared identifier notattr.

bad.cl:136: non-Int arguments: String + Int

bad.cl:139: Inferred type Int of initialization of x does not conform to identifier's declared type String.

bad.cl:144: non-Int arguments: String + Int

bad.cl:154: Undeclared identifier notattr.

bad.cl:158: Undeclared identifier notattr.

bad.cl:158: non-Int arguments: Object + Int

bad.cl:160: non-Int arguments: Bool + Bool

bad.cl:162: non-Int arguments: String + Int

bad.cl:166: Undeclared identifier notattr.

bad.cl:166: Argument of 'not' has type Object instead of Bool.

bad.cl:170: Undeclared identifier noattr.

bad.cl:170: Undeclared identifier noattr.

bad.cl:170: Illegal comparison with a basic type.

bad.cl:172: Undeclared identifier noattr.

bad.cl:172: Illegal comparison with a basic type.

bad.cl:174: Illegal comparison with a basic type.

bad.cl:180: Undeclared identifier notattr.

bad.cl:180: Argument of 'not' has type Object instead of Bool.

bad.cl:182: Argument of 'not' has type String instead of Bool.

bad.cl:186: Undeclared identifier notattr.

Compilation halted due to static semantic errors.