صالح زارع زاده – ۶۱۰۳۹۶۱۰۹

Code generation فرایندی است که در ان با استفاده از اختصاص دهی حافظه و تبدبل کد منبع به یک فرم جدید میتوان ان را توسط ماشین اجرا کرد.مترجم ها معمولا چندین گذر بر روی فرم های میانی انجام میدهند زیرا در ابتدا نیاز است که کذر ها به گذر قبلی خود نیاز دارند و اینکه الگوریتم های مختلفی برای بهینه سازی وجود دارد. در مراحل پیش ما گرفتن ارور های مختلف یک کد و ساختن یک درخت برای چک کردن تایپ ها با استفاده از مراحل lexer , parser , semantic بود .

ما در اینجا accumulator, stack machine را داریم که برای ترجمه کد استفاده میشوند.

Accumulator : ارزیابی اخرین عبارت (expression) را نگه میدارد.

$s0 : یک کپی از self را نگه میدارد.

$sp : اولین حافظه خالی در استک را نگه میدارد.

$t0,….,$tn : رجیستر های موقت هستد که در حافظه ذخیره میشوند و برای انجام عملیات ها استفاده میشوند.

دستورعمل ها به صورت بازگشتی هندل میشوند و در واقع از دستورهای بدیهی شروع و سپس دستورهای بزرگتر و سپس کل دستور را ترجمه میکنند.برای مثال x = add(w,add(y,z)) ابتدا مقادیر w , add(y,z) را بدست میاورد و در واقع با استفاده از self,E,S که self همان s0 است و E یک map از متغییر ها به مکان ان ها در حافظه است و S هم یک map از مکان های حافظه به مقادیر موجود در ان ها است سپس با استفاده از این دو مقدار x را بدست میاورد و در صورت نیاز E,S را تغییر میدهد تا با مقادیر جدید مطابقت کنند.

در ابتدا program\_class::cgen است که دو عملیات در ان انجام میشود initialize\_constants و اضافه کردن ان ها به Idtable و ساختن یک شی از کلاس CgenClassTable که با ان برنامه ترجمه اغاز میشود.

در ابتدا تایپ های integer , string , Boolean را ترجمه (generate) میکنیم و سپس وارد یک scope میشویم و در اخر از ان خارج میشویم سپس ما install\_basic\_classes را اجرا میکنیم که کلاس های پایه را به cgenclasstable اضافه میکند و سپس build\_inheritance\_tree را احرا میکنیم که درخت ارث بری را میسازد که تا بتوانیم فرزندان هر کلاس را بدانیم و سپس ما code را اجرا میکنیم که در ان تابع های مختلفی را فراخوانی میکنیم که عبارت اند از:

code\_preprocessing : یک عدد یکتا برای هرکلاس با توجه به جایگاهش در درخت ارث بری در نظر میگیریم.

code\_global\_data : در این بخش قسمت .data را مقدار دهی میکنیم که برای استفاده در قسمت خافظه به کار میرود و استک ماشین را در اینجا قرار میدهیم .

code\_select\_gc : لیبلی درست میکند که به garbage\_collector اشاره میکند.

code\_constants : چند تا object ثابت را مقدار دهی میکنیم برای مثال : string table , int table , code bools

code\_nametab : برای هر درخت در درخت ارث بری اسم ان را ذخیره میکند.

code\_objTab : prototype object , object init را را برای کلاس ها تعیین میکند و هر دو لیبل را برای هر کلاس در درخت ارث بری در .word ذخیره میکنیم.

code\_dispatch : بر روی کلاس ها بر اساس تگ ان ها حرکت میکینم و به صورت بازگشتی dispatch table جد های ان را بدست میاوریم.

code\_prototype : براساس تگ هر کلاس و اندازه ان و یک pointer به dispatch table و همه متغییر های هر کلاس بر اسای متغییر های خود کلاس و متغییر های جد های ان است .

code\_global\_text : همه ی متغییر های global را در .text ذخیره میکنیم و این قسمت ثابت است.

code\_methods : ما با مقدار دهی کلاس ها و دخیره کردن FP , SELF , RA برای رفتن به جد های ان ها برای چک کردن تابع ها است درواقع در انجا ما از هر تابع از لیبل ان شروع میکنیم و سپس fp , s0 , ra را ذخیره میکنیم سپس قسمت بدنه تابع را به صورت بازگشتی ترجمه میکنیم و سپس fp,s0,ra را از استک پاپ میکنیم و سپس به استیت اولیه برمیگردیم و استک هم به استیت اولیه خود برمیگردد.

مثال در مراحل بازگشتی ترجمه کد :

Loop\_class : ابتدا مقدار شرط ان را ارزیابی میکنیم اگر false بود از ان خارج میشویم و مقدار در غیر این صورت مقدار بدنه ی حلقه را ارزیابی میکنیم و مقدار ان را برابر تهی میگذرایم و سپس مقادیر accumulator و تغییرات را در s اعمال میکنیم.

Let\_class : ابتدا وارد scope جدید میشویم و سپس قسمت Init را ارزیابی میکنیم و سپس یک خانه حافظه جدید برای این متغییر که در let تعریف شده در نظر میگیریم و اسم متغییر و حافظه مربوط به ان را در identifiers قرار میدهیم و سپس اگر init تهی بود(مقدار دهی نشده بود) برای هر کدام از انواع کلاس های string , int , bool تعیین میکنیم مقدار اولیه ان را در غیر این صورت مقدار ان را تهی در نظر میگیریم و اگر مقدار دهی شده بود مقدار ان را در خانه حافظه ان قرار میدهیم و شرایط جدید را ذخیره میکنیم و قسمت بدنه را به صورت بازگشتی ارزیابی میکنیم و در اخر از ان scope که وارد ان شدیم خارج میشویم.

یک کد اجرا شده در برنامه :

(\*

\* Cool program reading descriptions of weighted directed graphs

\* from stdin. It builds up a graph objects with a list of vertices

\* and a list of edges. Every vertice has a list of outgoing edges.

\*

\* INPUT FORMAT

\* Every line has the form vertice successor\*

\* Where vertice is an int, and successor is vertice,weight

\*

\* An empty line or EOF terminates the input.

\*

\* The list of vertices and the edge list is printed out by the Main

\* class.

\*

\* TEST

\* Once compiled, the file g1.graph can be fed to the program.

\* The output should look like this:

nautilus.CS.Berkeley.EDU 53# spim -file graph.s <g1.graph

SPIM Version 5.4 of Jan. 17, 1994

Copyright 1990-1994 by James R. Larus (larus@cs.wisc.edu).

All Rights Reserved.

See the file README a full copyright notice.

Loaded: /home/n/cs164/lib/trap.handler

5 (5,5)5 (5,4)4 (5,3)3 (5,2)2 (5,1)1

4 (4,5)100 (4,3)55

3 (3,2)10

2 (2,1)150 (2,3)200

1 (1,2)100

(5,5)5 (5,4)4 (5,3)3 (5,2)2 (5,1)1 (4,5)100 (4,3)55 (3,2)10 (2,1)150 (2,3)200 (1,2)100

COOL program successfully executed

\*)

class Graph {

vertices : VList <- new VList;

edges : EList <- new EList;

add\_vertice(v : Vertice) : Object { {

edges <- v.outgoing().append(edges);

vertices <- vertices.cons(v);

} };

print\_E() : Object { edges.print() };

print\_V() : Object { vertices.print() };

};

class Vertice inherits IO {

num : Int;

out : EList <- new EList;

outgoing() : EList { out };

number() : Int { num };

init(n : Int) : SELF\_TYPE {

{

num <- n;

self;

}

};

add\_out(s : Edge) : SELF\_TYPE {

{

out <- out.cons(s);

self;

}

};

print() : Object {

{

out\_int(num);

out.print();

}

};

};

class Edge inherits IO {

from : Int;

to : Int;

weight : Int;

init(f : Int, t : Int, w : Int) : SELF\_TYPE {

{

from <- f;

to <- t;

weight <- w;

self;

}

};

print() : Object {

{

out\_string(" (");

out\_int(from);

out\_string(",");

out\_int(to);

out\_string(")");

out\_int(weight);

}

};

};

class EList inherits IO {

-- Define operations on empty lists of Edges.

car : Edge;

isNil() : Bool { true };

head() : Edge { { abort(); car; } };

tail() : EList { { abort(); self; } };

-- When we cons and element onto the empty list we get a non-empty

-- list. The (new Cons) expression creates a new list cell of class

-- Cons, which is initialized by a dispatch to init().

-- The result of init() is an element of class Cons, but it

-- conforms to the return type List, because Cons is a subclass of

-- List.

cons(e : Edge) : EList {

(new ECons).init(e, self)

};

append(l : EList) : EList {

if self.isNil() then l

else tail().append(l).cons(head())

fi

};

print() : Object {

out\_string("\n")

};

};

(\*

\* Cons inherits all operations from List. We can reuse only the cons

\* method though, because adding an element to the front of an emtpy

\* list is the same as adding it to the front of a non empty

\* list. All other methods have to be redefined, since the behaviour

\* for them is different from the empty list.

\*

\* Cons needs an extra attribute to hold the rest of the list.

\*

\* The init() method is used by the cons() method to initialize the

\* cell.

\*)

class ECons inherits EList {

cdr : EList; -- The rest of the list

isNil() : Bool { false };

head() : Edge { car };

tail() : EList { cdr };

init(e : Edge, rest : EList) : EList {

{

car <- e;

cdr <- rest;

self;

}

};

print() : Object {

{

car.print();

cdr.print();

}

};

};

class VList inherits IO {

-- Define operations on empty lists of vertices.

car : Vertice;

isNil() : Bool { true };

head() : Vertice { { abort(); car; } };

tail() : VList { { abort(); self; } };

-- When we cons and element onto the empty list we get a non-empty

-- list. The (new Cons) expression creates a new list cell of class

-- ECons, which is initialized by a dispatch to init().

-- The result of init() is an element of class Cons, but it

-- conforms to the return type List, because Cons is a subclass of

-- List.

cons(v : Vertice) : VList {

(new VCons).init(v, self)

};

print() : Object { out\_string("\n") };

};

class VCons inherits VList {

cdr : VList; -- The rest of the list

isNil() : Bool { false };

head() : Vertice { car };

tail() : VList { cdr };

init(v : Vertice, rest : VList) : VList {

{

car <- v;

cdr <- rest;

self;

}

};

print() : Object {

{

car.print();

cdr.print();

}

};

};

class Parse inherits IO {

boolop : BoolOp <- new BoolOp;

-- Reads the input and parses the fields

read\_input() : Graph {

(let g : Graph <- new Graph in {

(let line : String <- in\_string() in

while (boolop.and(not line="\n", not line="")) loop {

-- out\_string(line);

-- out\_string("\n");

g.add\_vertice(parse\_line(line));

line <- in\_string();

} pool

);

g;

} )

};

parse\_line(s : String) : Vertice {

(let v : Vertice <- (new Vertice).init(a2i(s)) in {

while (not rest.length() = 0) loop {

-- out\_string(rest);

-- out\_string("\n");

(let succ : Int <- a2i(rest) in (let

weight : Int <- a2i(rest)

in

v.add\_out(new Edge.init(v.number(),

succ,

weight))

) );

} pool;

v;

}

)

};

c2i(char : String) : Int {

if char = "0" then 0 else

if char = "1" then 1 else

if char = "2" then 2 else

if char = "3" then 3 else

if char = "4" then 4 else

if char = "5" then 5 else

if char = "6" then 6 else

if char = "7" then 7 else

if char = "8" then 8 else

if char = "9" then 9 else

{ abort(); 0; } -- the 0 is needed to satisfy the typchecker

fi fi fi fi fi fi fi fi fi fi

};

rest : String;

a2i(s : String) : Int {

if s.length() = 0 then 0 else

if s.substr(0,1) = "-" then ~a2i\_aux(s.substr(1,s.length()-1)) else

if s.substr(0,1) = " " then a2i(s.substr(1,s.length()-1)) else

a2i\_aux(s)

fi fi fi

};

(\*

a2i\_aux converts the usigned portion of the string. As a programming

example, this method is written iteratively.

The conversion stops at a space or comma.

As a side effect, r is set to the remaining string (without the comma).

\*)

a2i\_aux(s : String) : Int {

(let int : Int <- 0 in

{

(let j : Int <- s.length() in

(let i : Int <- 0 in

while i < j loop

(let c : String <- s.substr(i,1) in

if (c = " ") then

{

rest <- s.substr(i+1,s.length()-i-1);

i <- j;

}

else if (c = ",") then

{

rest <- s.substr(i+1, s.length()-i-1);

i <- j;

}

else

{

int <- int \* 10 + c2i(s.substr(i,1));

i <- i + 1;

if i=j then rest <- "" else "" fi;

}

fi fi

)

pool

)

);

int;

}

)

};

};

class Main inherits Parse {

g : Graph <- read\_input();

main() : Object {

{

g.print\_V();

g.print\_E();

}

};

};

class BoolOp {

and(b1 : Bool, b2 : Bool) : Bool {

if b1 then b2 else false fi

};

or(b1 : Bool, b2 : Bool) : Bool {

if b1 then true else b2 fi

};

};

نتیجه اجرا زمانی که به کد ماشین ترجمه شده است :

# start of generated code

.data

.align 2

.globl class\_nameTab

.globl Main\_protObj

.globl Int\_protObj

.globl String\_protObj

.globl bool\_const0

.globl bool\_const1

.globl \_int\_tag

.globl \_bool\_tag

.globl \_string\_tag

\_int\_tag:

.word 10

\_bool\_tag:

.word 11

\_string\_tag:

.word 12

.globl \_MemMgr\_INITIALIZER

\_MemMgr\_INITIALIZER:

.word \_NoGC\_Init

.globl \_MemMgr\_COLLECTOR

\_MemMgr\_COLLECTOR:

.word \_NoGC\_Collect

.globl \_MemMgr\_TEST

\_MemMgr\_TEST:

.word 0

.word -1

str\_const36:

.word 12

.word 6

.word String\_dispTab

.word int\_const6

.ascii "BoolOp"

.byte 0

.align 2

.word -1

str\_const35:

.word 12

.word 6

.word String\_dispTab

.word int\_const4

.ascii "Main"

.byte 0

.align 2

.word -1

str\_const34:

.word 12

.word 6

.word String\_dispTab

.word int\_const5

.ascii "Parse"

.byte 0

.align 2

.word -1

str\_const33:

.word 12

.word 6

.word String\_dispTab

.word int\_const5

.ascii "VCons"

.byte 0

.align 2

.word -1

str\_const32:

.word 12

.word 6

.word String\_dispTab

.word int\_const5

.ascii "VList"

.byte 0

.align 2

.word -1

str\_const31:

.word 12

.word 6

.word String\_dispTab

.word int\_const5

.ascii "ECons"

.byte 0

.align 2

.word -1

str\_const30:

.word 12

.word 6

.word String\_dispTab

.word int\_const5

.ascii "EList"

.byte 0

.align 2

.word -1

str\_const29:

.word 12

.word 6

.word String\_dispTab

.word int\_const4

.ascii "Edge"

.byte 0

.align 2

.word -1

str\_const28:

.word 12

.word 6

.word String\_dispTab

.word int\_const7

.ascii "Vertice"

.byte 0

.align 2

.word -1

str\_const27:

.word 12

.word 6

.word String\_dispTab

.word int\_const5

.ascii "Graph"

.byte 0

.align 2

.word -1

str\_const26:

.word 12

.word 6

.word String\_dispTab

.word int\_const6

.ascii "String"

.byte 0

.align 2

.word -1

str\_const25:

.word 12

.word 6

.word String\_dispTab

.word int\_const4

.ascii "Bool"

.byte 0

.align 2

.word -1

str\_const24:

.word 12

.word 5

.word String\_dispTab

.word int\_const3

.ascii "Int"

.byte 0

.align 2

.word -1

str\_const23:

.word 12

.word 5

.word String\_dispTab

.word int\_const2

.ascii "IO"

.byte 0

.align 2

.word -1

str\_const22:

.word 12

.word 6

.word String\_dispTab

.word int\_const6

.ascii "Object"

.byte 0

.align 2

.word -1

str\_const21:

.word 12

.word 7

.word String\_dispTab

.word int\_const10

.ascii "\_prim\_slot"

.byte 0

.align 2

.word -1

str\_const20:

.word 12

.word 7

.word String\_dispTab

.word int\_const9

.ascii "SELF\_TYPE"

.byte 0

.align 2

.word -1

str\_const19:

.word 12

.word 7

.word String\_dispTab

.word int\_const9

.ascii "\_no\_class"

.byte 0

.align 2

.word -1

str\_const18:

.word 12

.word 8

.word String\_dispTab

.word int\_const11

.ascii "<basic class>"

.byte 0

.align 2

.word -1

str\_const17:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii " "

.byte 0

.align 2

.word -1

str\_const16:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "-"

.byte 0

.align 2

.word -1

str\_const15:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "9"

.byte 0

.align 2

.word -1

str\_const14:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "8"

.byte 0

.align 2

.word -1

str\_const13:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "7"

.byte 0

.align 2

.word -1

str\_const12:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "6"

.byte 0

.align 2

.word -1

str\_const11:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "5"

.byte 0

.align 2

.word -1

str\_const10:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "4"

.byte 0

.align 2

.word -1

str\_const9:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "3"

.byte 0

.align 2

.word -1

str\_const8:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "2"

.byte 0

.align 2

.word -1

str\_const7:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "1"

.byte 0

.align 2

.word -1

str\_const6:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "0"

.byte 0

.align 2

.word -1

str\_const5:

.word 12

.word 5

.word String\_dispTab

.word int\_const1

.byte 0

.align 2

.word -1

str\_const4:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii "\n"

.byte 0

.align 2

.word -1

str\_const3:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii ")"

.byte 0

.align 2

.word -1

str\_const2:

.word 12

.word 5

.word String\_dispTab

.word int\_const0

.ascii ","

.byte 0

.align 2

.word -1

str\_const1:

.word 12

.word 5

.word String\_dispTab

.word int\_const2

.ascii " ("

.byte 0

.align 2

.word -1

str\_const0:

.word 12

.word 7

.word String\_dispTab

.word int\_const10

.ascii "example.cl"

.byte 0

.align 2

.word -1

int\_const11:

.word 10

.word 4

.word Int\_dispTab

.word 13

.word -1

int\_const10:

.word 10

.word 4

.word Int\_dispTab

.word 10

.word -1

int\_const9:

.word 10

.word 4

.word Int\_dispTab

.word 9

.word -1

int\_const8:

.word 10

.word 4

.word Int\_dispTab

.word 8

.word -1

int\_const7:

.word 10

.word 4

.word Int\_dispTab

.word 7

.word -1

int\_const6:

.word 10

.word 4

.word Int\_dispTab

.word 6

.word -1

int\_const5:

.word 10

.word 4

.word Int\_dispTab

.word 5

.word -1

int\_const4:

.word 10

.word 4

.word Int\_dispTab

.word 4

.word -1

int\_const3:

.word 10

.word 4

.word Int\_dispTab

.word 3

.word -1

int\_const2:

.word 10

.word 4

.word Int\_dispTab

.word 2

.word -1

int\_const1:

.word 10

.word 4

.word Int\_dispTab

.word 0

.word -1

int\_const0:

.word 10

.word 4

.word Int\_dispTab

.word 1

.word -1

bool\_const0:

.word 11

.word 4

.word Bool\_dispTab

.word 0

.word -1

bool\_const1:

.word 11

.word 4

.word Bool\_dispTab

.word 1

class\_nameTab:

.word str\_const22

.word str\_const23

.word str\_const28

.word str\_const29

.word str\_const30

.word str\_const31

.word str\_const32

.word str\_const33

.word str\_const34

.word str\_const35

.word str\_const24

.word str\_const25

.word str\_const26

.word str\_const27

.word str\_const36

class\_objTab:

.word Object\_protObj

.word Object\_init

.word IO\_protObj

.word IO\_init

.word Vertice\_protObj

.word Vertice\_init

.word Edge\_protObj

.word Edge\_init

.word EList\_protObj

.word EList\_init

.word ECons\_protObj

.word ECons\_init

.word VList\_protObj

.word VList\_init

.word VCons\_protObj

.word VCons\_init

.word Parse\_protObj

.word Parse\_init

.word Main\_protObj

.word Main\_init

.word Int\_protObj

.word Int\_init

.word Bool\_protObj

.word Bool\_init

.word String\_protObj

.word String\_init

.word Graph\_protObj

.word Graph\_init

.word BoolOp\_protObj

.word BoolOp\_init

BoolOp\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word BoolOp.and

.word BoolOp.or

Main\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word Parse.read\_input

.word Parse.parse\_line

.word Parse.c2i

.word Parse.a2i

.word Parse.a2i\_aux

.word Main.main

Parse\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word Parse.read\_input

.word Parse.parse\_line

.word Parse.c2i

.word Parse.a2i

.word Parse.a2i\_aux

VCons\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word VCons.isNil

.word VCons.head

.word VCons.tail

.word VList.cons

.word VCons.print

.word VCons.init

VList\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word VList.isNil

.word VList.head

.word VList.tail

.word VList.cons

.word VList.print

ECons\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word ECons.isNil

.word ECons.head

.word ECons.tail

.word EList.cons

.word EList.append

.word ECons.print

.word ECons.init

EList\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word EList.isNil

.word EList.head

.word EList.tail

.word EList.cons

.word EList.append

.word EList.print

Edge\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word Edge.init

.word Edge.print

Vertice\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

.word Vertice.outgoing

.word Vertice.number

.word Vertice.init

.word Vertice.add\_out

.word Vertice.print

Graph\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word Graph.add\_vertice

.word Graph.print\_E

.word Graph.print\_V

String\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word String.length

.word String.concat

.word String.substr

Bool\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

Int\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

IO\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word IO.out\_string

.word IO.out\_int

.word IO.in\_string

.word IO.in\_int

Object\_dispTab:

.word Object.abort

.word Object.type\_name

.word Object.copy

.word -1

BoolOp\_protObj:

.word 14

.word 3

.word BoolOp\_dispTab

.word -1

Main\_protObj:

.word 9

.word 6

.word Main\_dispTab

.word 0

.word str\_const5

.word 0

.word -1

Parse\_protObj:

.word 8

.word 5

.word Parse\_dispTab

.word 0

.word str\_const5

.word -1

VCons\_protObj:

.word 7

.word 5

.word VCons\_dispTab

.word 0

.word 0

.word -1

VList\_protObj:

.word 6

.word 4

.word VList\_dispTab

.word 0

.word -1

ECons\_protObj:

.word 5

.word 5

.word ECons\_dispTab

.word 0

.word 0

.word -1

EList\_protObj:

.word 4

.word 4

.word EList\_dispTab

.word 0

.word -1

Edge\_protObj:

.word 3

.word 6

.word Edge\_dispTab

.word int\_const1

.word int\_const1

.word int\_const1

.word -1

Vertice\_protObj:

.word 2

.word 5

.word Vertice\_dispTab

.word int\_const1

.word 0

.word -1

Graph\_protObj:

.word 13

.word 5

.word Graph\_dispTab

.word 0

.word 0

.word -1

String\_protObj:

.word 12

.word 5

.word String\_dispTab

.word int\_const1

.word 0

.word -1

Bool\_protObj:

.word 11

.word 4

.word Bool\_dispTab

.word 0

.word -1

Int\_protObj:

.word 10

.word 4

.word Int\_dispTab

.word 0

.word -1

IO\_protObj:

.word 1

.word 3

.word IO\_dispTab

.word -1

Object\_protObj:

.word 0

.word 3

.word Object\_dispTab

.globl heap\_start

heap\_start:

.word 0

.text

.globl Main\_init

.globl Int\_init

.globl String\_init

.globl Bool\_init

.globl Main.main

BoolOp\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal Object\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

BoolOp.and:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 8($fp)

lw $a0 12($a0)

bne $a0 $zero label0

label1:

la $a0 bool\_const0

b label2

label0:

lw $a0 4($fp)

label2:

lw $ra 4($sp)

addiu $sp $sp 112

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

BoolOp.or:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 8($fp)

lw $a0 12($a0)

bne $a0 $zero label3

label4:

lw $a0 4($fp)

b label5

label3:

la $a0 bool\_const1

label5:

lw $ra 4($sp)

addiu $sp $sp 112

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Main\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $zero

sw $a0 12($s0)

move $a0 $zero

sw $a0 20($s0)

move $a0 $s0

jal Parse\_init

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label6

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label6:

move $a0 $s0

lw $t1 8($a0)

lw $t1 28($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 20($s0)

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Main.main:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 20($s0)

bne $a0 $zero label7

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label7:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 20($s0)

bne $a0 $zero label8

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label8:

lw $t1 8($a0)

lw $t1 16($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Parse\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $zero

sw $a0 12($s0)

move $a0 $s0

jal IO\_init

la $a0 BoolOp\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal BoolOp\_init

lw $a0 4($sp)

addiu $sp $sp 4

sw $a0 12($s0)

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Parse.read\_input:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

la $a0 Graph\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal Graph\_init

lw $a0 4($sp)

addiu $sp $sp 4

sw $a0 0($fp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label9

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label9:

move $a0 $s0

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 -4($fp)

label10:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

lw $a0 -4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const4

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label14

la $a1 bool\_const0

jal equality\_test

label14:

lw $t1 12($a0)

la $a0 bool\_const1

beqz $t1 label13

la $a0 bool\_const0

label13:

sw $a0 8($sp)

lw $a0 -4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const5

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label16

la $a1 bool\_const0

jal equality\_test

label16:

lw $t1 12($a0)

la $a0 bool\_const1

beqz $t1 label15

la $a0 bool\_const0

label15:

sw $a0 4($sp)

lw $a0 12($s0)

bne $a0 $zero label12

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label12:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

lw $a0 12($a0)

beqz $a0 label11

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 -4($fp)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label18

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label18:

move $a0 $s0

lw $t1 8($a0)

lw $t1 32($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 4($sp)

lw $a0 0($fp)

bne $a0 $zero label17

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label17:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label19

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label19:

move $a0 $s0

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 -4($fp)

b label10

label11:

move $a0 $zero

lw $a0 0($fp)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Parse.parse\_line:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 4($fp)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label21

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label21:

move $a0 $s0

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 4($sp)

la $a0 Vertice\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal Vertice\_init

lw $a0 4($sp)

addiu $sp $sp 4

bne $a0 $zero label20

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label20:

lw $t1 8($a0)

lw $t1 36($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 -8($fp)

label22:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 16($s0)

bne $a0 $zero label26

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label26:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const1

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label25

la $a1 bool\_const0

jal equality\_test

label25:

lw $t1 12($a0)

la $a0 bool\_const1

beqz $t1 label24

la $a0 bool\_const0

label24:

lw $a0 12($a0)

beqz $a0 label23

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 16($s0)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label27

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label27:

move $a0 $s0

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 -12($fp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 16($s0)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label28

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label28:

move $a0 $s0

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 -16($fp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -12

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 -8($fp)

bne $a0 $zero label31

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label31:

lw $t1 8($a0)

lw $t1 32($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 12($sp)

lw $a0 -12($fp)

sw $a0 8($sp)

lw $a0 -16($fp)

sw $a0 4($sp)

la $a0 Edge\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal Edge\_init

lw $a0 4($sp)

addiu $sp $sp 4

bne $a0 $zero label30

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label30:

lw $t1 8($a0)

lw $t1 28($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 4($sp)

lw $a0 -8($fp)

bne $a0 $zero label29

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label29:

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

b label22

label23:

move $a0 $zero

lw $a0 -8($fp)

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Parse.c2i:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const6

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label35

la $a1 bool\_const0

jal equality\_test

label35:

lw $a0 12($a0)

bne $a0 $zero label32

label33:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const7

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label39

la $a1 bool\_const0

jal equality\_test

label39:

lw $a0 12($a0)

bne $a0 $zero label36

label37:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const8

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label43

la $a1 bool\_const0

jal equality\_test

label43:

lw $a0 12($a0)

bne $a0 $zero label40

label41:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const9

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label47

la $a1 bool\_const0

jal equality\_test

label47:

lw $a0 12($a0)

bne $a0 $zero label44

label45:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const10

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label51

la $a1 bool\_const0

jal equality\_test

label51:

lw $a0 12($a0)

bne $a0 $zero label48

label49:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const11

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label55

la $a1 bool\_const0

jal equality\_test

label55:

lw $a0 12($a0)

bne $a0 $zero label52

label53:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const12

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label59

la $a1 bool\_const0

jal equality\_test

label59:

lw $a0 12($a0)

bne $a0 $zero label56

label57:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const13

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label63

la $a1 bool\_const0

jal equality\_test

label63:

lw $a0 12($a0)

bne $a0 $zero label60

label61:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const14

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label67

la $a1 bool\_const0

jal equality\_test

label67:

lw $a0 12($a0)

bne $a0 $zero label64

label65:

lw $a0 4($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const15

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label71

la $a1 bool\_const0

jal equality\_test

label71:

lw $a0 12($a0)

bne $a0 $zero label68

label69:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label72

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label72:

move $a0 $s0

lw $t1 8($a0)

lw $t1 0($t1)

jalr $t1

addiu $sp $sp 8

la $a0 int\_const1

b label70

label68:

la $a0 int\_const9

label70:

b label66

label64:

la $a0 int\_const8

label66:

b label62

label60:

la $a0 int\_const7

label62:

b label58

label56:

la $a0 int\_const6

label58:

b label54

label52:

la $a0 int\_const5

label54:

b label50

label48:

la $a0 int\_const4

label50:

b label46

label44:

la $a0 int\_const3

label46:

b label42

label40:

la $a0 int\_const2

label42:

b label38

label36:

la $a0 int\_const0

label38:

b label34

label32:

la $a0 int\_const1

label34:

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Parse.a2i:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 4($fp)

bne $a0 $zero label77

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label77:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const1

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label76

la $a1 bool\_const0

jal equality\_test

label76:

lw $a0 12($a0)

bne $a0 $zero label73

label74:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

la $a0 int\_const1

sw $a0 8($sp)

la $a0 int\_const0

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label82

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label82:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const16

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label81

la $a1 bool\_const0

jal equality\_test

label81:

lw $a0 12($a0)

bne $a0 $zero label78

label79:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

la $a0 int\_const1

sw $a0 8($sp)

la $a0 int\_const0

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label87

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label87:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const17

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label86

la $a1 bool\_const0

jal equality\_test

label86:

lw $a0 12($a0)

bne $a0 $zero label83

label84:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 4($fp)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label88

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label88:

move $a0 $s0

lw $t1 8($a0)

lw $t1 44($t1)

jalr $t1

addiu $sp $sp 8

b label85

label83:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

la $a0 int\_const0

sw $a0 8($sp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 4($fp)

bne $a0 $zero label91

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label91:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const0

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

sub $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label90

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label90:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label89

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label89:

move $a0 $s0

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

label85:

b label80

label78:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

la $a0 int\_const0

sw $a0 8($sp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 4($fp)

bne $a0 $zero label94

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label94:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const0

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

sub $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label93

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label93:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label92

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label92:

move $a0 $s0

lw $t1 8($a0)

lw $t1 44($t1)

jalr $t1

addiu $sp $sp 8

jal Object.copy

lw $t1 12($a0)

neg $t1 $t1

sw $t1 12($a0)

label80:

b label75

label73:

la $a0 int\_const1

label75:

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Parse.a2i\_aux:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

la $a0 int\_const1

sw $a0 -20($fp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 4($fp)

bne $a0 $zero label95

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label95:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 -24($fp)

la $a0 int\_const1

sw $a0 -28($fp)

label96:

lw $a0 -28($fp)

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 -24($fp)

move $t1 $a0

lw $t1 12($t1)

lw $a0 4($sp)

lw $a0 12($a0)

blt $a0 $t1 label98

la $a0 bool\_const0

b label99

label98:

la $a0 bool\_const1

label99:

addiu $sp $sp 4

lw $a0 12($a0)

beqz $a0 label97

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

lw $a0 -28($fp)

sw $a0 8($sp)

la $a0 int\_const0

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label100

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label100:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 -32($fp)

lw $a0 -32($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const17

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label104

la $a1 bool\_const0

jal equality\_test

label104:

lw $a0 12($a0)

bne $a0 $zero label101

label102:

lw $a0 -32($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 str\_const2

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label108

la $a1 bool\_const0

jal equality\_test

label108:

lw $a0 12($a0)

bne $a0 $zero label105

label106:

lw $a0 -20($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const10

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

mul $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

lw $a0 -28($fp)

sw $a0 8($sp)

la $a0 int\_const0

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label110

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label110:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label109

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label109:

move $a0 $s0

lw $t1 8($a0)

lw $t1 36($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

add $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 -20($fp)

lw $a0 -28($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const0

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

add $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 -28($fp)

lw $a0 -28($fp)

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 -24($fp)

lw $t1 4($sp)

addiu $sp $sp 4

move $t2 $a0

la $a0 bool\_const1

beq $t1 $t2 label114

la $a1 bool\_const0

jal equality\_test

label114:

lw $a0 12($a0)

bne $a0 $zero label111

label112:

la $a0 str\_const5

b label113

label111:

sw $s0 0($sp)

addiu $sp $sp -4

la $a0 str\_const5

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

label113:

b label107

label105:

sw $s0 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

lw $a0 -28($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const0

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

add $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 8($sp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 4($fp)

bne $a0 $zero label116

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label116:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 -28($fp)

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

sub $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const0

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

sub $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label115

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label115:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

lw $a0 -24($fp)

sw $a0 -28($fp)

label107:

b label103

label101:

sw $s0 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

lw $a0 -28($fp)

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const0

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

add $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 8($sp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 4($fp)

bne $a0 $zero label118

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label118:

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 -28($fp)

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

sub $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 0($sp)

addiu $sp $sp -4

la $a0 int\_const0

sw $a0 0($sp)

addiu $sp $sp -4

lw $a0 8($sp)

jal Object.copy

lw $t1 12($a0)

lw $t2 4($sp)

lw $t2 12($t2)

sub $t1 $t1 $t2

sw $t1 12($a0)

addiu $sp $sp 8

sw $a0 4($sp)

lw $a0 4($fp)

bne $a0 $zero label117

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label117:

lw $t1 8($a0)

lw $t1 20($t1)

jalr $t1

addiu $sp $sp 8

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

lw $a0 -24($fp)

sw $a0 -28($fp)

label103:

b label96

label97:

move $a0 $zero

lw $a0 -20($fp)

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VCons\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal VList\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

VCons.isNil:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

la $a0 bool\_const0

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VCons.head:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 12($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VCons.tail:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 16($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VCons.init:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 8($fp)

lw $s0 4($sp)

sw $a0 12($s0)

addiu $sp $sp 4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 4($fp)

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

move $a0 $s0

lw $ra 4($sp)

addiu $sp $sp 112

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VCons.print:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 12($s0)

bne $a0 $zero label119

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label119:

lw $t1 8($a0)

lw $t1 44($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 16($s0)

bne $a0 $zero label120

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label120:

lw $t1 8($a0)

lw $t1 44($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VList\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal IO\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

VList.isNil:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

la $a0 bool\_const1

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VList.head:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label121

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label121:

move $a0 $s0

lw $t1 8($a0)

lw $t1 0($t1)

jalr $t1

addiu $sp $sp 8

lw $a0 12($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VList.tail:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label122

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label122:

move $a0 $s0

lw $t1 8($a0)

lw $t1 0($t1)

jalr $t1

addiu $sp $sp 8

move $a0 $s0

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VList.cons:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

lw $a0 4($fp)

sw $a0 8($sp)

move $a0 $s0

sw $a0 4($sp)

la $a0 VCons\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal VCons\_init

lw $a0 4($sp)

addiu $sp $sp 4

bne $a0 $zero label123

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label123:

lw $t1 8($a0)

lw $t1 48($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

VList.print:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

la $a0 str\_const4

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label124

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label124:

move $a0 $s0

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

ECons\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal EList\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

ECons.isNil:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

la $a0 bool\_const0

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

ECons.head:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 12($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

ECons.tail:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 16($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

ECons.init:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 8($fp)

lw $s0 4($sp)

sw $a0 12($s0)

addiu $sp $sp 4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 4($fp)

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

move $a0 $s0

lw $ra 4($sp)

addiu $sp $sp 112

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

ECons.print:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 12($s0)

bne $a0 $zero label125

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label125:

lw $t1 8($a0)

lw $t1 32($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 16($s0)

bne $a0 $zero label126

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label126:

lw $t1 8($a0)

lw $t1 48($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

EList\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal IO\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

EList.isNil:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

la $a0 bool\_const1

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

EList.head:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label127

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label127:

move $a0 $s0

lw $t1 8($a0)

lw $t1 0($t1)

jalr $t1

addiu $sp $sp 8

lw $a0 12($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

EList.tail:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label128

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label128:

move $a0 $s0

lw $t1 8($a0)

lw $t1 0($t1)

jalr $t1

addiu $sp $sp 8

move $a0 $s0

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

EList.cons:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -8

lw $a0 4($fp)

sw $a0 8($sp)

move $a0 $s0

sw $a0 4($sp)

la $a0 ECons\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal ECons\_init

lw $a0 4($sp)

addiu $sp $sp 4

bne $a0 $zero label129

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label129:

lw $t1 8($a0)

lw $t1 52($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

EList.append:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label133

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label133:

move $a0 $s0

lw $t1 8($a0)

lw $t1 28($t1)

jalr $t1

addiu $sp $sp 8

lw $a0 12($a0)

bne $a0 $zero label130

label131:

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label135

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label135:

move $a0 $s0

lw $t1 8($a0)

lw $t1 32($t1)

jalr $t1

addiu $sp $sp 8

sw $a0 4($sp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 4($fp)

sw $a0 4($sp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

move $a0 $s0

bne $a0 $zero label137

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label137:

move $a0 $s0

lw $t1 8($a0)

lw $t1 36($t1)

jalr $t1

addiu $sp $sp 8

bne $a0 $zero label136

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label136:

lw $t1 8($a0)

lw $t1 44($t1)

jalr $t1

addiu $sp $sp 8

bne $a0 $zero label134

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label134:

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

b label132

label130:

lw $a0 4($fp)

label132:

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

EList.print:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

la $a0 str\_const4

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label138

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label138:

move $a0 $s0

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Edge\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal IO\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Edge.init:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 12($fp)

lw $s0 4($sp)

sw $a0 12($s0)

addiu $sp $sp 4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 8($fp)

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 4($fp)

lw $s0 4($sp)

sw $a0 20($s0)

addiu $sp $sp 4

move $a0 $s0

lw $ra 4($sp)

addiu $sp $sp 116

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Edge.print:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

la $a0 str\_const1

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label139

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label139:

move $a0 $s0

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 12($s0)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label140

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label140:

move $a0 $s0

lw $t1 8($a0)

lw $t1 16($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

la $a0 str\_const2

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label141

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label141:

move $a0 $s0

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 16($s0)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label142

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label142:

move $a0 $s0

lw $t1 8($a0)

lw $t1 16($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

la $a0 str\_const3

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label143

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label143:

move $a0 $s0

lw $t1 8($a0)

lw $t1 12($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 20($s0)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label144

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label144:

move $a0 $s0

lw $t1 8($a0)

lw $t1 16($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Vertice\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $zero

sw $a0 16($s0)

move $a0 $s0

jal IO\_init

la $a0 EList\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal EList\_init

lw $a0 4($sp)

addiu $sp $sp 4

sw $a0 16($s0)

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Vertice.outgoing:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 16($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Vertice.number:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

lw $a0 12($s0)

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Vertice.init:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

lw $a0 4($fp)

lw $s0 4($sp)

sw $a0 12($s0)

addiu $sp $sp 4

move $a0 $s0

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Vertice.add\_out:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 4($fp)

sw $a0 4($sp)

lw $a0 16($s0)

bne $a0 $zero label145

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label145:

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

move $a0 $s0

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Vertice.print:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 12($s0)

sw $a0 4($sp)

move $a0 $s0

bne $a0 $zero label146

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label146:

move $a0 $s0

lw $t1 8($a0)

lw $t1 16($t1)

jalr $t1

addiu $sp $sp 8

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 16($s0)

bne $a0 $zero label147

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label147:

lw $t1 8($a0)

lw $t1 48($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Graph\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $zero

sw $a0 12($s0)

move $a0 $zero

sw $a0 16($s0)

move $a0 $s0

jal Object\_init

la $a0 VList\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal VList\_init

lw $a0 4($sp)

addiu $sp $sp 4

sw $a0 12($s0)

la $a0 EList\_protObj

jal Object.copy

sw $a0 0($sp)

addiu $sp $sp -4

jal EList\_init

lw $a0 4($sp)

addiu $sp $sp 4

sw $a0 16($s0)

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Graph.add\_vertice:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 16($s0)

sw $a0 4($sp)

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 4($fp)

bne $a0 $zero label149

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label149:

lw $t1 8($a0)

lw $t1 28($t1)

jalr $t1

addiu $sp $sp 8

bne $a0 $zero label148

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label148:

lw $t1 8($a0)

lw $t1 44($t1)

jalr $t1

addiu $sp $sp 8

lw $s0 4($sp)

sw $a0 16($s0)

addiu $sp $sp 4

sw $s0 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp -4

lw $a0 4($fp)

sw $a0 4($sp)

lw $a0 12($s0)

bne $a0 $zero label150

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label150:

lw $t1 8($a0)

lw $t1 40($t1)

jalr $t1

addiu $sp $sp 8

lw $s0 4($sp)

sw $a0 12($s0)

addiu $sp $sp 4

lw $ra 4($sp)

addiu $sp $sp 108

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Graph.print\_E:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 16($s0)

bne $a0 $zero label151

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label151:

lw $t1 8($a0)

lw $t1 48($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

Graph.print\_V:

move $fp $sp

move $s0 $a0

addiu $sp $sp -100

sw $ra 0($sp)

addiu $sp $sp -4

sw $fp 0($sp)

addiu $sp $sp -4

sw $s0 0($sp)

addiu $sp $sp -4

addiu $sp $sp 0

lw $a0 12($s0)

bne $a0 $zero label152

la $a0 str\_const0

li $t1 1

jal \_dispatch\_abort

label152:

lw $t1 8($a0)

lw $t1 44($t1)

jalr $t1

addiu $sp $sp 8

lw $ra 4($sp)

addiu $sp $sp 104

lw $s0 4($sp)

lw $fp 8($sp)

jr $ra

String\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal Object\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Bool\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal Object\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Int\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal Object\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

IO\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

jal Object\_init

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

Object\_init:

addiu $sp $sp -12

sw $fp 12($sp)

sw $s0 8($sp)

sw $ra 4($sp)

move $fp $sp

move $s0 $a0

move $a0 $s0

move $a0 $s0

lw $ra 4($sp)

lw $s0 8($sp)

lw $fp 12($sp)

addiu $sp $sp 12

jr $ra

# end of generated code