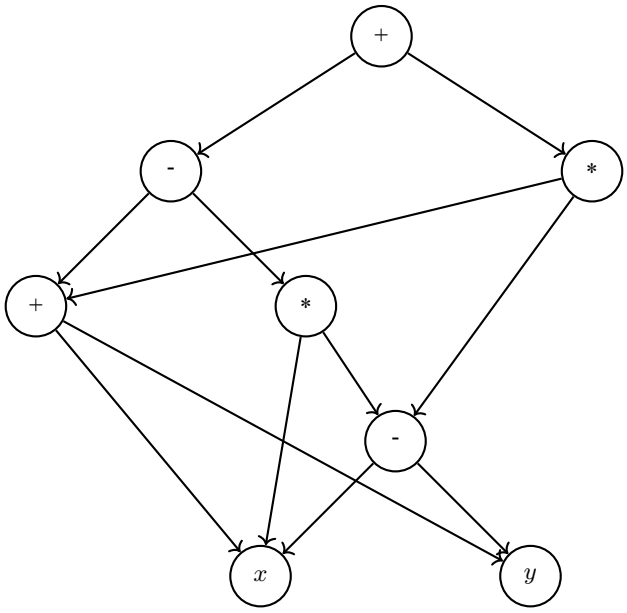
Compiler Principle: 中间代码生成 **Spring 2025**

Homework 8 — May 12

Lecturer: Feng Xiaobing *Completed by: 2022K8009929010 Zhang Jiawei*

8.1

画出 DAG 如下图所示：

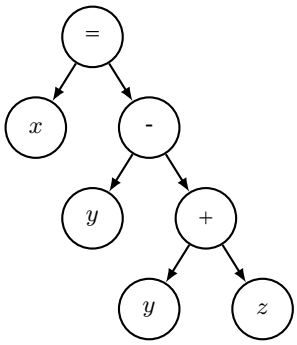


每个子表达式的值编码如下表所示：

1	**id**	x	
2	**id**	y	
3	+	1	2
4	−	1	2
5	*	1	4
6	−	3	5
7	*	3	4
8	+	5	7

8.2

1. 翻译出的抽象语法树如下图所示：



2. 先写成三地址代码,如下所示:

```
t1 = y + z
t2 = y - t1
x = t2
```

实现这个三地址代码的四元式序列如下表所示:

	op	arg_1	arg_2	$result$
0	+	y	z	t1
1	−	y	t1	t2
2	=	t2		x

3. 三元式序列如下所示:

	op	arg_1	arg_2
0	+	y	z
1	−	y	(0)
2	=	x	(1)

4. 间接三元式序列如下所示:

	$instruction$
35	(0)
36	(1)
37	(2)

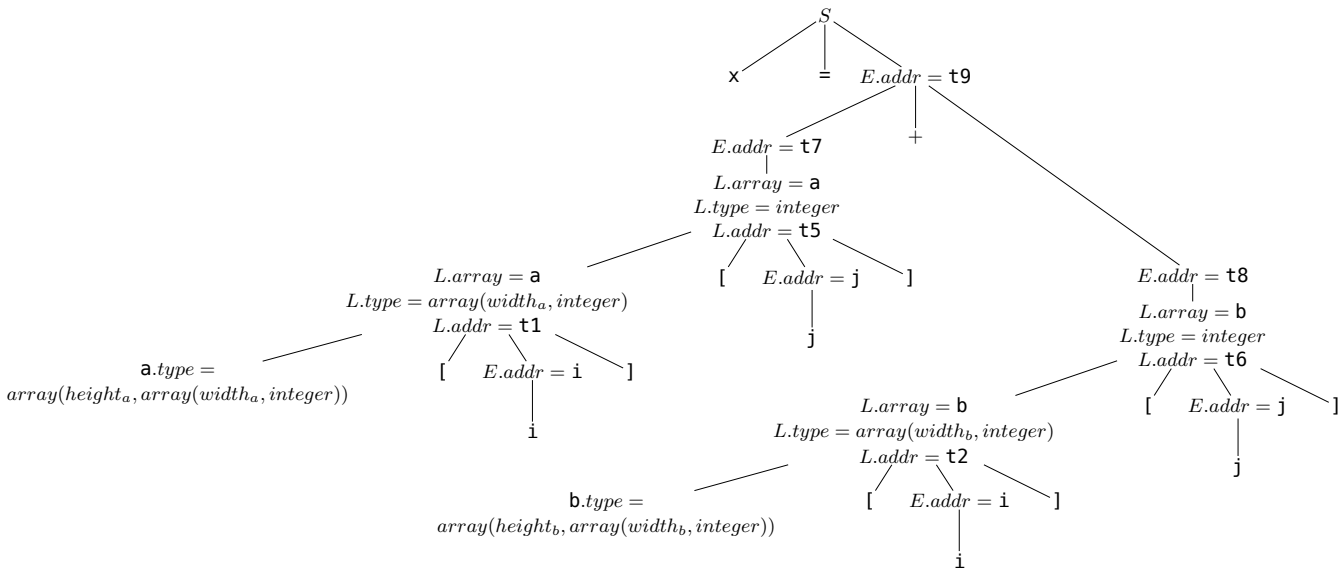
8.3

各个标识符的类型和相对地址如下表所示:

标识符	类型	大小	环境	相对地址
x	float	4	0	0
p.x	float	4	1	0
p.y	float	4	1	4
p	record	8	0	8
q.m.tag	int	4	2	0
q.m.float	float	4	2	4
q.m	record	8	1	0
q.n.idx	int	4	2	0
q.n.y	float	4	2	4
q.n	record	8	1	8
q	record	16	0	16

8.4

(1) 该表达式的注释语法分析树如下所示：

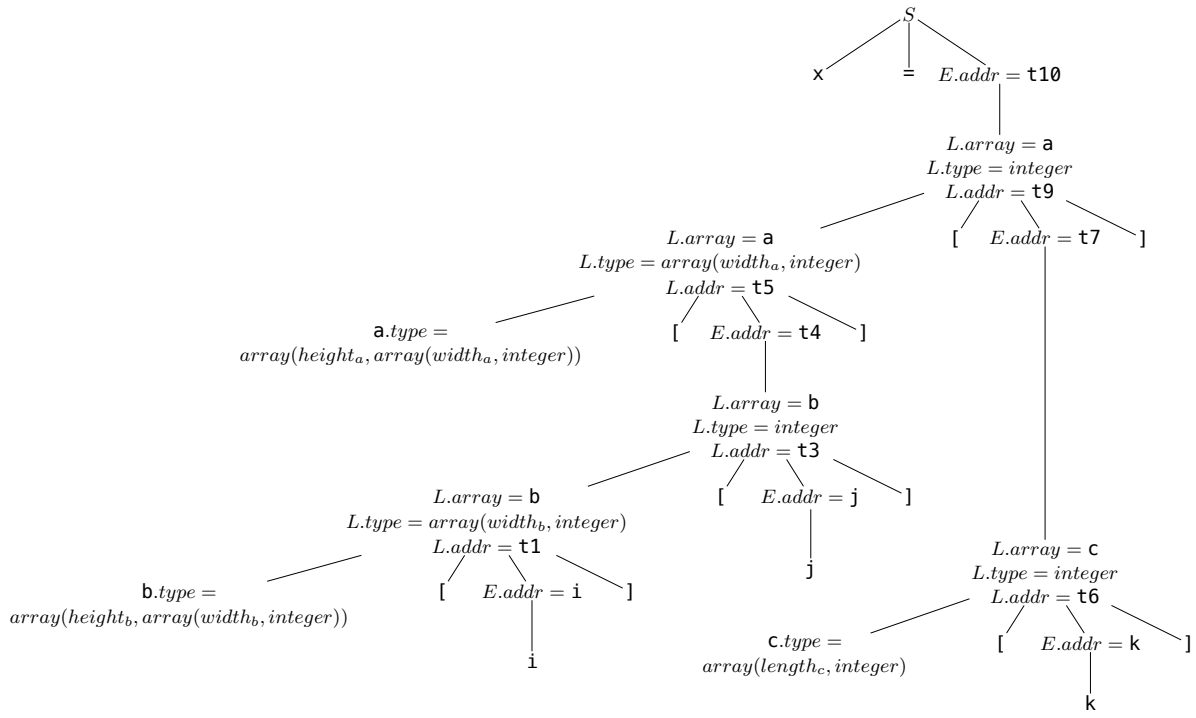


翻译成三地址代码如下所示：

```
t1 = i * 4width_a
t2 = i * 4width_b
t3 = j * 4
t4 = j * 4
t5 = t1 + t3
t6 = t2 + t4
t7 = a[t5]
```

```
8    t8 = b[t6]
9    t9 = t7 + t8
10   x = t9
```

(2) 该表达式的注释语法分析树如下所示：



翻译成三地址代码如下所示：

```
1  t1 = i * 4width_b
2  t2 = j * 4
3  t3 = t1 + t2
4  t4 = b[t3]
5  t5 = t4 * 4width_a
6  t6 = k * 4
7  t7 = c[t6]
8  t8 = t7 * 4
9  t9 = t5 + t8
10 t10 = a[t9]
11 x = t10
```