

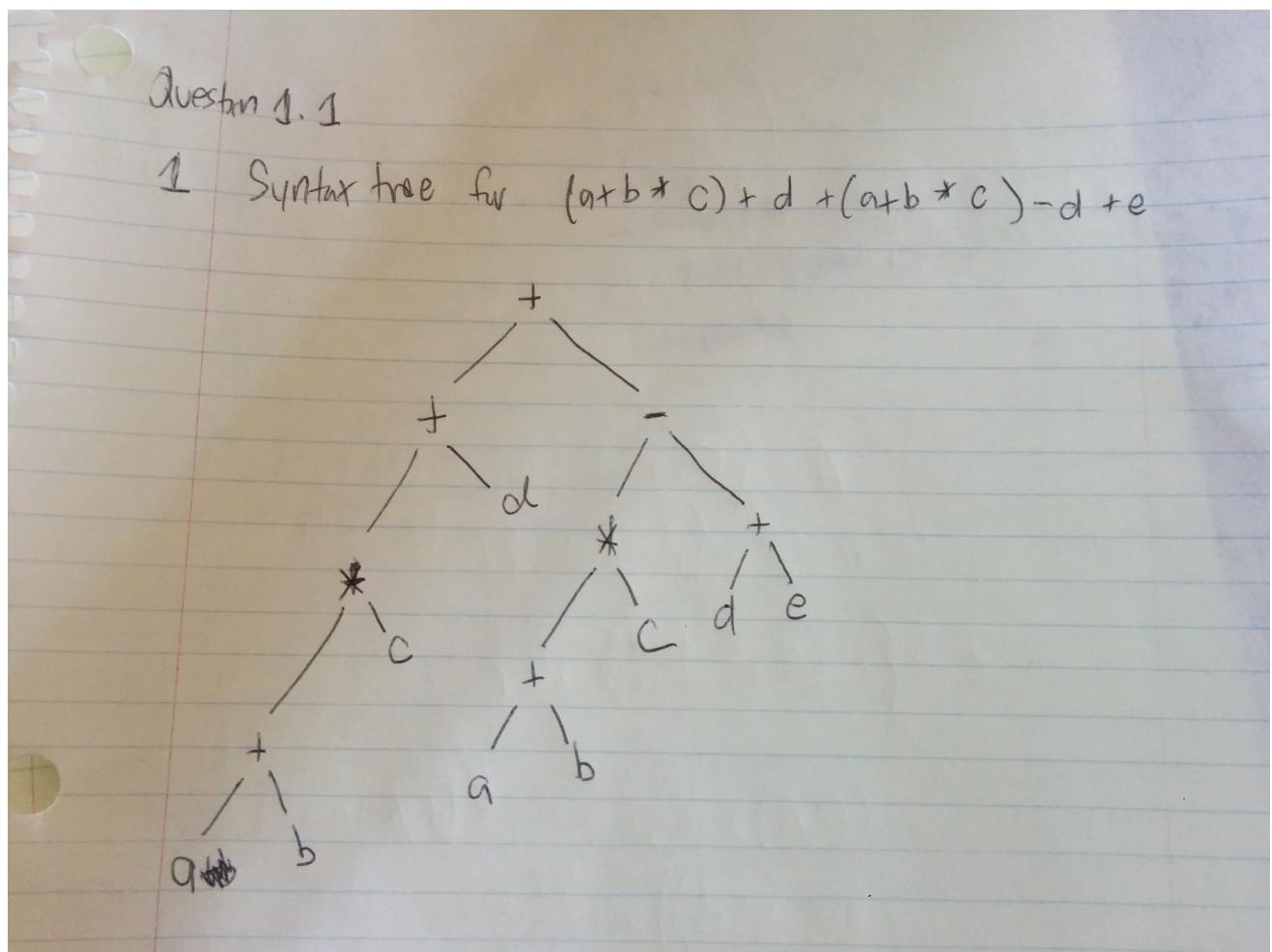
# Homework 7

Abhi Agarwal

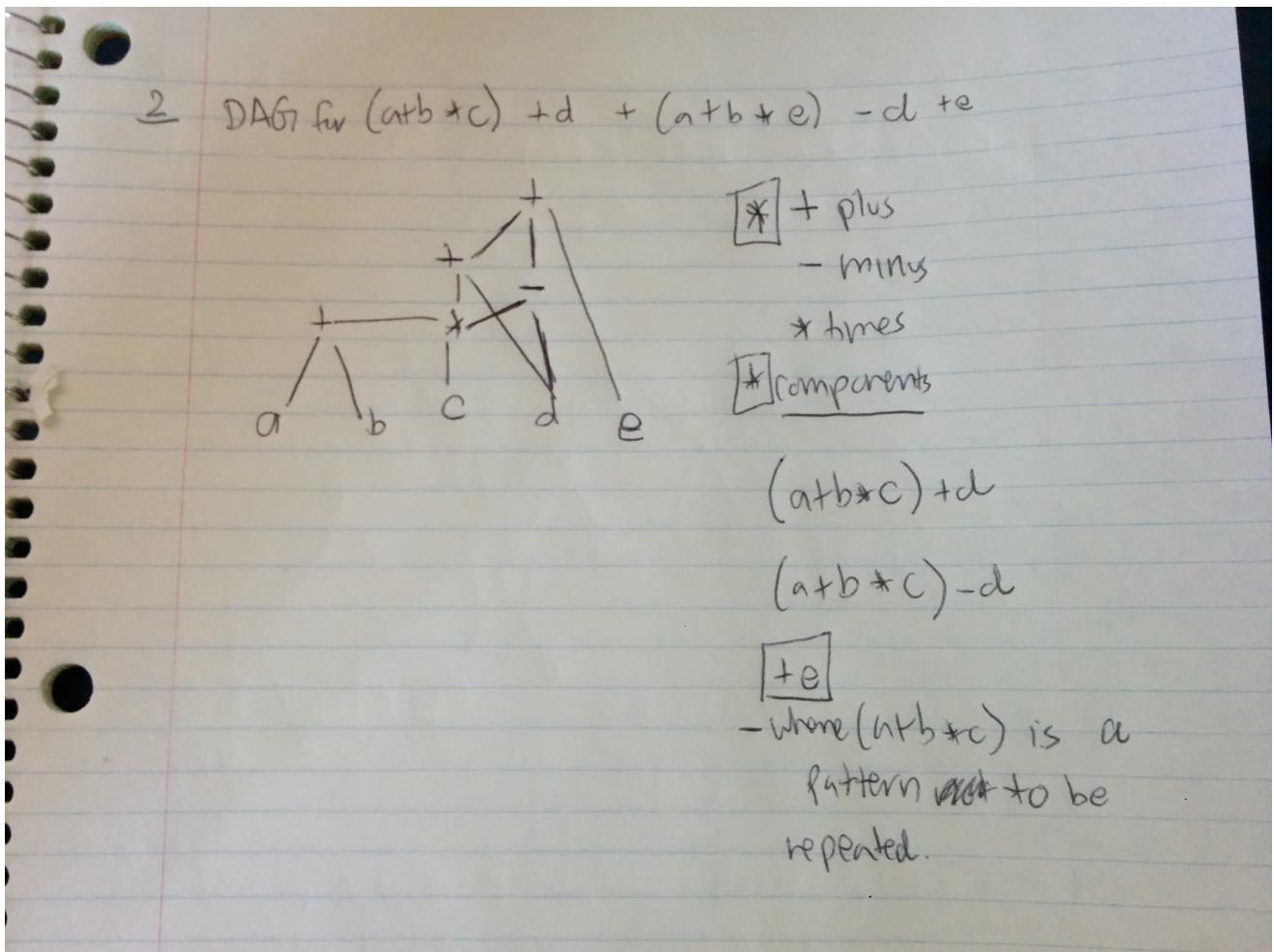
## 1 Representation

### 1.1 Question 1.1

#### 1.1.1 Syntax Tree



### 1.1.2 DAG (Directed Acyclic Graph) representation



### 1.1.3 Three-Address Code representation

- where  $(wkb+e)$  is a pattern ~~not~~ to be repeated.

3 Three - Address Code representation

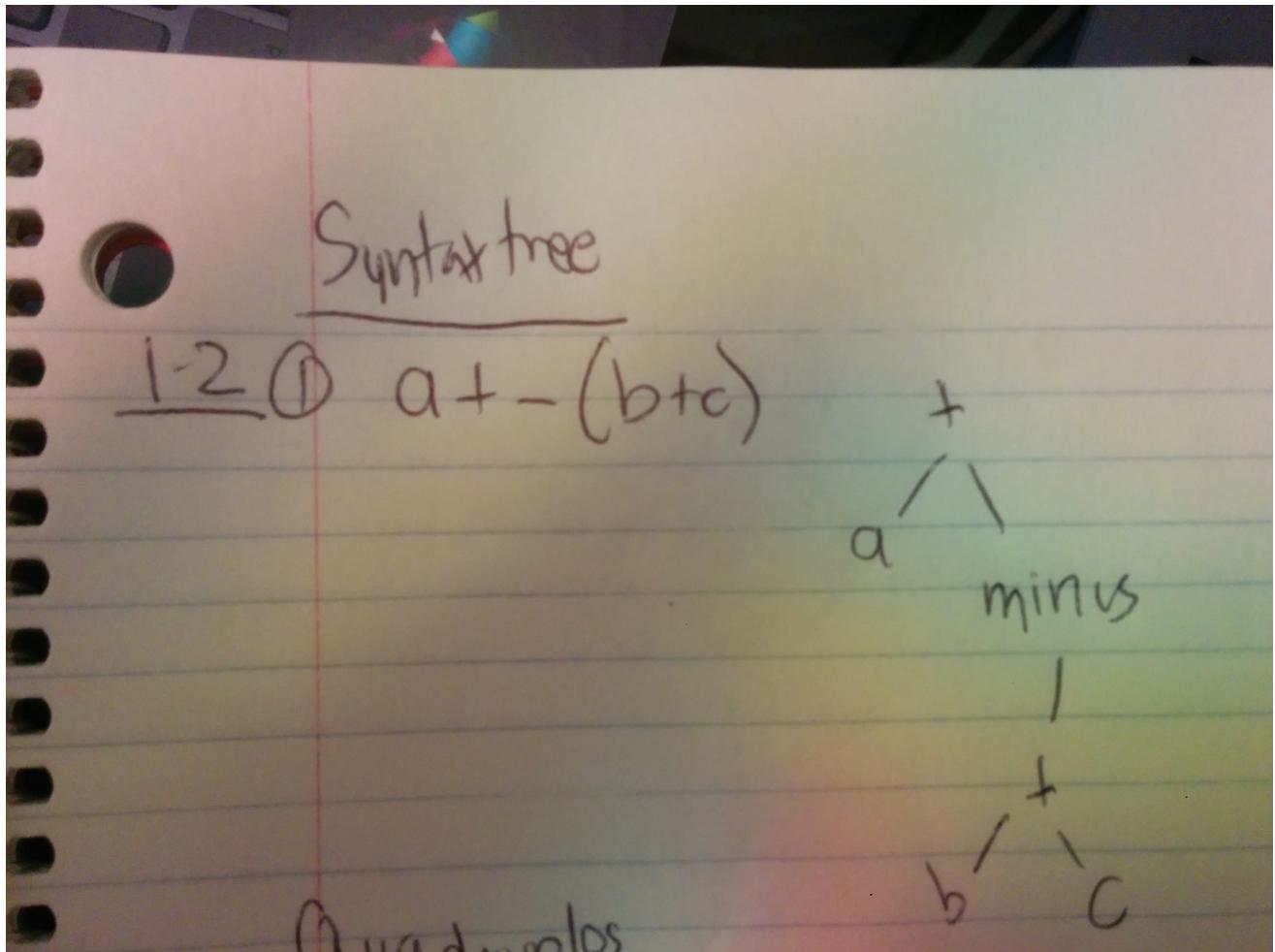
$$t_1 = a+b$$
$$t_2 = t_1 * c$$
$$t_3 = t_2 + d$$
$$t_4 = t_2 - d$$
$$t_5 = t_3 + t_4 + e$$

Where  $t_5$  is final answer

```
graph TD; t1[t1] --- t2[t2]; t2 --- t3[t3]; t2 --- t4[t4]; t3 --- t5[t5]; t4 --- t5; a --- t1; b --- t1; c --- t2; d --- t2; e --- t5;
```

## 1.2 Question 1.2

### 1.2.1 Syntax Tree



### 1.2.2 Quadruples

Quadruples

Op	Arg1	Arg2	Result
0	+ b	c	$t_1$
1	minus $t_1$		$t_2$
2	+ a	$t_2$	$t_3$

$t_1 = b + c$   
 $t_2 = -t_1$   
 $t_3 = a + t_2$

### 1.2.3 Triples

③ Triples

	OP	Arg1	Arg2	
0	+	b	c	
1	minus	(0)		
2	+	a	(1)	

Result referred to by position

```

    . + \
    a   \ minus
        | 
        + \
        b   \ c
  
```

#### 1.2.4 Indirect Triples

(4) Indirect triples

	instuch	0	op	arg 1	arg 2
35	(0)	+	b	c	
36	(1)	minus	(0)		
37	(2)	+	a	(1)	
.					
.					
.					

## 2 Translation

### 2.1 Question 2.1

2.1.1 Add a translation rule for the following expression production:  
 $E \rightarrow E_1 * E_2$

① Production	Rules
$S \Rightarrow id = E_1 ; S_2$   $\epsilon$	$S \cdot code = E_1 \cdot code \parallel [id' = 'E_1 \cdot addr] \parallel   S_2 \cdot code$ $S_2 \cdot code = []$
$E \Rightarrow E_1 + E_2$	$E_1 \cdot e = E \cdot addr = newTemp()$ $E \cdot code = E_1 \cdot code \parallel E_2 \cdot code \parallel [E \cdot addr = E_1 \cdot addr + E_2 \cdot addr]$
<del><math>E \Rightarrow E_1 * E_2</math></del>	$E_1 \cdot e = E \cdot addr = newTemp()$ $E \cdot code = E_1 \cdot code \parallel E_2 \cdot code \parallel [E \cdot addr = E_1 \cdot addr * E_2 \cdot addr]$
$  - E_1$ ⋮	...

## 2.2 Question 2.2

2.2.1 Explain how the code is expected to evaluate, assuming parameters are passed call-by-value.

## 2.3 Question 2.3

2.3.1  $S \rightarrow \text{for}(S_1; B; S_2) S_3$

Extending the code from the book

PRODUCTION	SEMANTIC RULES
$P \rightarrow S$	$S.\text{next} = \text{newlabel}()$ $P.\text{code} = S.\text{code} \parallel \text{label}(S.\text{next})$
$S \rightarrow \text{assign}$	$S.\text{code} = \text{assign}.\text{code}$
$S \rightarrow \text{if } (B) S_1$	$B.\text{true} = \text{newlabel}()$ $B.\text{false} = S_1.\text{next} = S.\text{next}$ $S.\text{code} = B.\text{code} \parallel \text{label}(B.\text{true}) \parallel S_1.\text{code}$
$S \rightarrow \text{if } (B) S_1 \text{ else } S_2$	$B.\text{true} = \text{newlabel}()$ $B.\text{false} = \text{newlabel}()$ $S_1.\text{next} = S_2.\text{next} = S.\text{next}$ $S.\text{code} = B.\text{code}$ $\quad \parallel \text{label}(B.\text{true}) \parallel S_1.\text{code}$ $\quad \parallel \text{gen('goto' } S.\text{next})$ $\quad \parallel \text{label}(B.\text{false}) \parallel S_2.\text{code}$
$S \rightarrow \text{while } (B) S_1$	$\text{begin} = \text{newlabel}()$ $B.\text{true} = \text{newlabel}()$ $B.\text{false} = S.\text{next}$ $S_1.\text{next} = \text{begin}$ $S.\text{code} = \text{label}(\text{begin}) \parallel B.\text{code}$ $\quad \parallel \text{label}(B.\text{true}) \parallel S_1.\text{code}$ $\quad \parallel \text{gen('goto' } \text{begin})$
$S \rightarrow S_1 S_2$	$S_1.\text{next} = \text{newlabel}()$ $S_2.\text{next} = S.\text{next}$ $S.\text{code} = S_1.\text{code} \parallel \text{label}(S_1.\text{next}) \parallel S_2.\text{code}$

Figure 6.36: Syntax-directed definition for flow-of-control statements

We assume that  $\text{newlabel}()$  creates a new label each time it is called, and  $\text{label}(L)$  attaches label  $L$  to the current node. 9

Question 2.3

Production	Semantic Rules
$S \rightarrow \text{for}(S_1; B; S_2) S_3$	$S_1.\text{next} = \text{newlabel}()$
	<del>B.true = newlabel()</del>
	$B.\text{false} = S.\text{next}$
	$S_2.\text{next} = S_1.\text{next}$
	$S_3.\text{next} = \text{newlabel}()$
	$S.\text{code} = S_1.\text{code}$
	$\quad \quad \quad    \text{label}(S_1.\text{next}) \quad    B.\text{code}$
	$\quad \quad \quad    \text{label}(B.\text{true}) \quad    S_3.\text{code}$
	$\quad \quad \quad    \text{label}(S_3.\text{next}) \quad    S_2.\text{code}$
	$\quad \quad \quad    \text{gen('goto' } S_1.\text{next})$
referencing page 402.	