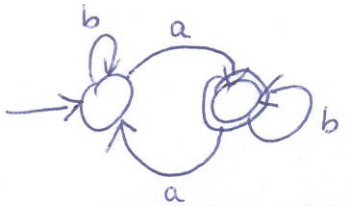
1.



2.

$$S \to bS \mid aT$$
$$T \to \varepsilon \mid bT \mid aS$$

Start state : S

3. No there is no type τ. A typing attempt fails in the rule for conditionals, since the two branches have types ref int and int, respectively:

$$\frac{\frac{\Gamma \vdash x : \text{int} \quad \Gamma \vdash 1 : \text{int}}{\Gamma \vdash x + 1 : \text{int}}}{\Gamma' \vdash \text{alloc}\,(x+1) : \text{ref int}} \qquad \text{and} \qquad \frac{}{\Gamma' \vdash 4 : \text{int}}$$

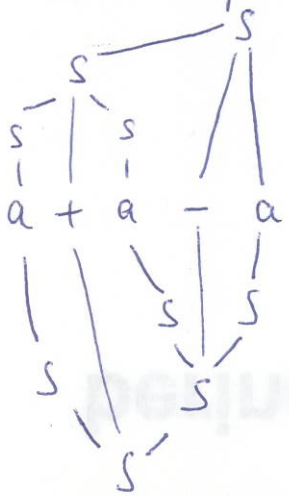
where $\Gamma' = \Gamma[x \mapsto \text{int}]$

The rule for conditionals requires the two branches to yield the same type.

4. An example grammar is

$$S \to S + S \mid S - S \mid a \quad , \text{ start symbol } S$$

String $a + a - a$ has two parse trees



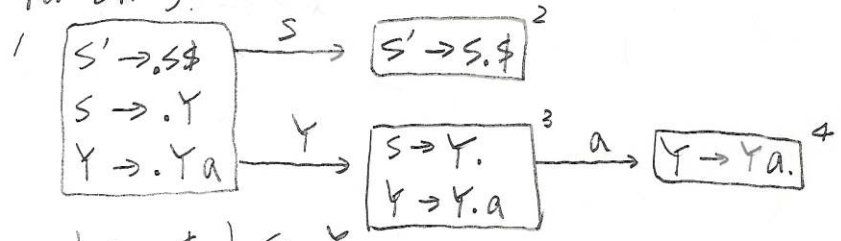
Problem 5.

1 $S' \to S\$$
2 $S \to Y$
3 $Y \to Ya$

5'

· need to prove this is SLR grammar, but isn't LR(0) and LL(k) grammar.

For LR(0).



	a	$	S	Y
1			g2	g3
2		a		
3	r2/s4	r2		
4	r3	r3		

5'

There is an duplicated entry, so it is not in LR(0)

With SLR, $a \notin$ Follow(S) then

	a	$	S	Y
1			g2	g3
2		a		
3	s4	r2		
4	r3	r3		

5'

no duplication, thus it is in SLR.

Finally, this gramma has left recursion. that cannot be LL(k) 5'

Problem 6

need to prove it is LL(0), i.e. no look ahead

$S' \to a\$$

two cases. input a: accept
others: reject ⇒ no look ahead

5'
5'