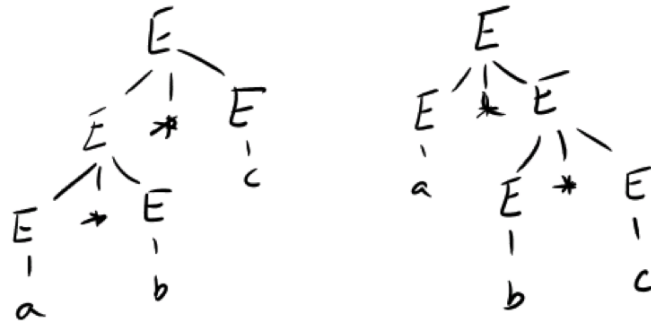


```

1. let rec cst_to_ast t = match t with
    | E1 (t',Minus,t'') -> Sub(cst_to_ast t', cst_to_ast t'')
    | E2 t' -> cst_to_ast t'
    | T1 (Ident s) -> Id s
    | T2 (t',Star,Ident s) -> Times(cst_to_ast t', Id s) ;;

```

2. (a)



(b)	Action	Stack	Input	Action	Stack	Input
	Sh		a*b*c	Sh		a*b*c
	R E→id	a	*b*c	R E→id	a	*b*c
	Sh	E	*b*c	Sh	E	*b*c
	Sh	E *	b*c	Sh	E *	b*c
	R E→id	E * b	*c	R E→id	E * b	*c
	R E→E*E	E * E	*c	Sh	E * E	*c
	Sh	E	*c	Sh	E * E *	c
	Sh	E *	c	R E→id	E * E * c	
	R E→id	E * c		R E→E*E	E * E * E	
	R E→E*E	E * E		R E→E*E	E * E	
	Acc	E		Acc	E	

(c) (i) Tree 1 is correct. (ii) The first parse is correct.

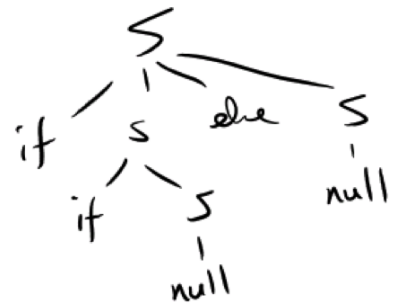
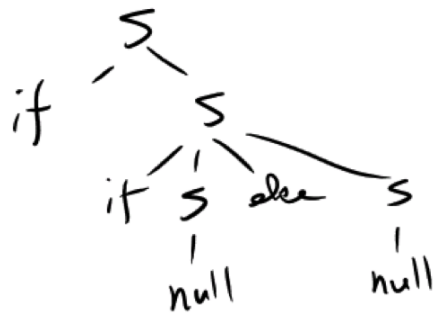
(d) %left Minus
%left Star

```

3. A → B = A | B
   B → B == E | E
   E → E - T | T
   T → id | T * id

```

4. (a)



(b)

Action	Stack	Input
Sh		if if null else null
Sh	if	if null else null
Sh	if if	null else null
Red	if if null	else null
Sh	if if S	else null
Sh	if if S else	null
Red	if if S else null	
Red	if if S else S	
Red	if S	
Acc	S	

Action	Stack	Input
Sh		if if null else null
Sh	if	if null else null
Sh	if if	null else null
Red	if if null	else null
Red	if if S	else null
Sh	if S	else null
Sh	if S else	null
Red	if S else null	
Red	if S else S	
Acc	S	

(c) %left If
 %left Else