Homework 2

Sunday, February 20, 2022 2:09 PM

Jalen Powell

1. a.) baab

S → AaBb

→ ba Bb (A=>b)

→ baab (B=>a)

- Valid Sentence

b.) bbbab

S → AaBb

→ AbaBb (A => Ab)

→ Abbabb (A => Ab)

→ bbbaBb (A => b)

- To get the desired outcome, this next step can't be executed. Invalid

c.) bbaaaaa

S → AaBb

→ AbaBb (A=>Ab)

→ bbaBb (A=> b)

→ bbaaBb (B=>aB)

- The last step is a b, so it won't reach desired outputs
- Invalid sentence

d.) bbaab

S → AaBb

→ AbaBb (A=>Ab)

→ bbaBb (A=>b)

→ bbaab (B=>a)

- Valid Sentence

2. Identify all of the tokens, and which lexemes they categorize. Put them in a table.

Tokens	Lexemes
Assignment Operator	=
Arithmetic Operators	+, *
Identifiers	A, B, C
Grouping Symbols	(,)

3. B = B + (C + (A + A))

<assign> → <id> = <expr>

→ B = <expr>

→ B = <id> + <expr>

→ B = B + (<expr>)

→ B = B + (<id> + <expr>)

→ B = B + (C + <expr>)

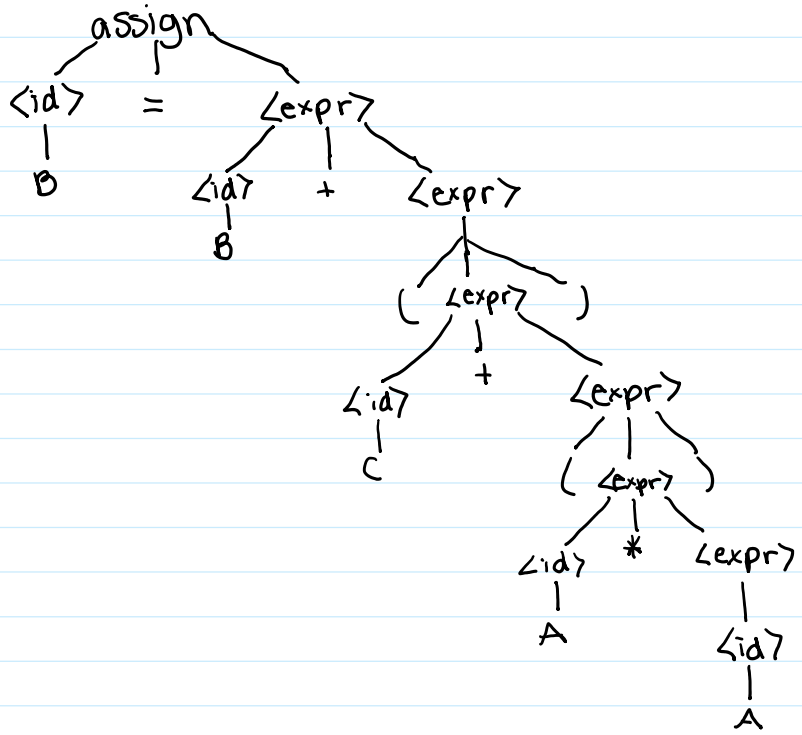
→ B = B + (C + (<expr>)

→ B = B + (C + (<id> + <expr>)

→ B = B + (C + (A * <expr>))

→ B = B + (C + (A * <id>)

→ B = B + (C + (A * A)



4. S → Aa | Bb

A → Aa | AbC | C

B → S | bb

C → C

- A → Aa | AbC
- A → C
- A → AA'

  => A' → a | bC

- S → Aa | Bb

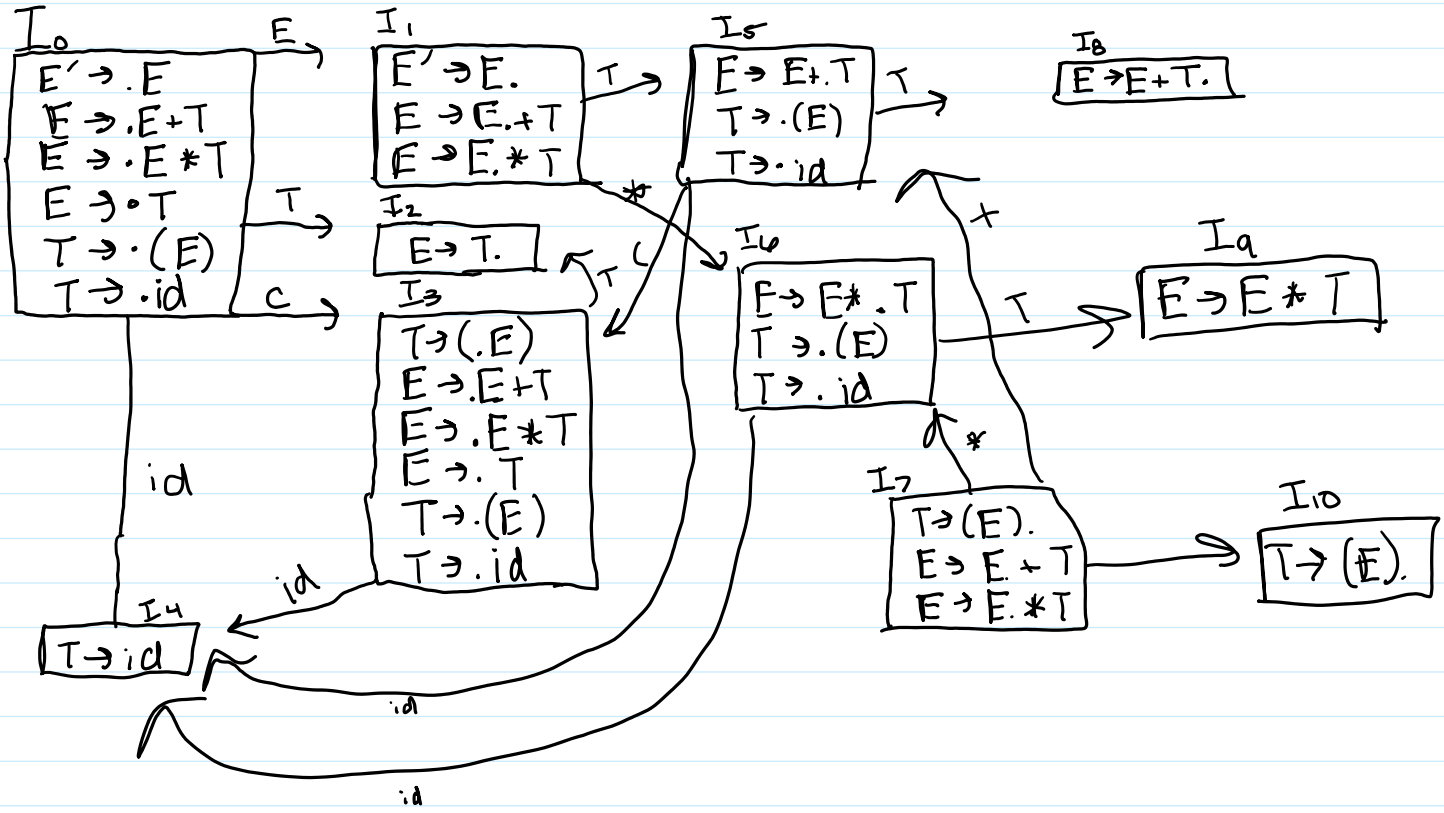
- $A \rightarrow AA'$
- $A' \rightarrow a \mid bC$
- $B \rightarrow C$
- $B \rightarrow S \mid bb$
- $C \rightarrow C$

5. $A \rightarrow aBc \mid ac \mid a$
   $B \rightarrow b \mid aB$

- $A \rightarrow aA'$
- $A' \rightarrow Bc \mid c \mid \epsilon$
- $B \rightarrow b \mid aB$

6. $E \rightarrow E+T \mid E*T \mid T$
   $T \rightarrow (E) \mid id$

- $E' \rightarrow E$
- $E \rightarrow E+T$
- $E \rightarrow E*T$
- $T \rightarrow (E)$
- $T \rightarrow id$



State	Action						Goto	
	+	*	(	)	id	$	E	T
0			s^3		s^4		1	2
1	s^5	s^6				accept		
2	r^3	r^3	r^3	r^3	r^3	r^3		
3			s^3		s^4		7	2
4	r^5	r^5	r^5	r^5	r^5	r^5		
5			s^3		s^4			8
6			s^3		s^4			9
7	s^5	s^6		s10				
8	r1	r1	r1	r1	r1	r^1		
9	r^2	r^2	r^2	r^2	r^2	r^2		
10	r^4	r^4	r^4	r^4	r^4	r^4		

1) E → E+T
2) E → E*T
3) E → T
4) T → (E)
5) T → id

7 (id+id) + id

Stack	Buffer	Action
$	(id+id)* id$	—
$	(id+id)* id$	Shift
$(	id+id)* id$	Shift
$id	+id)* id$	reduce T → id
$(T	+id)* id$	reduce E → T
$(E	+id)* id$	Shift
$(E+	id)* id$	Shift
$(E+id	)* id$	reduce T → id
$(E+T	)* id$	reduce E → E+T
$(E	)* id$	Shift
$(E)	* id$	reduce T → (E)
$ T	* id$	reduce E → T
$ E	* id$	Shift
$ E*	id$	Shift
$ E*id	$	reduce T → id
$ E*T	$	reduce E → E*T
$E	$	Accept

$ E	* id $	shift
$ E*	id $	shift
$ E* id	$	reduce T→id
$ E * T	$	reduce E→E*T
$ E	$	Accept

8. (id+id)+id

E→E* T
→E * id T→id
→T * id E→T
→(E) *id T→(E)
→(E+T)* id E→E+T
→(E+id) * id T→id
→(T+id) * id E→T
→(id+id) * id T→id