CSE-420, Mid-term exam, Fall 2019
Solution Sheet

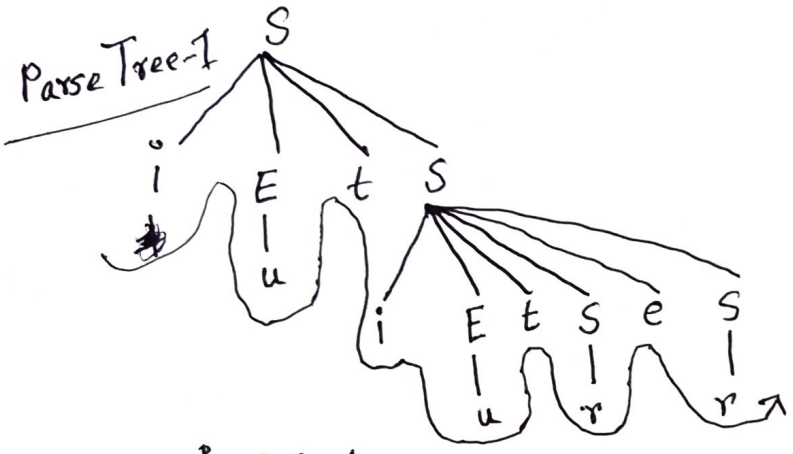
1. a) G1:

$S \rightarrow iEtS$
$S \rightarrow iEtSeS$
$S \rightarrow r$
$E \rightarrow (E)$
$E \rightarrow$ ~~true~~ u
$E \rightarrow$ ~~false~~ f

String: iutiutrer

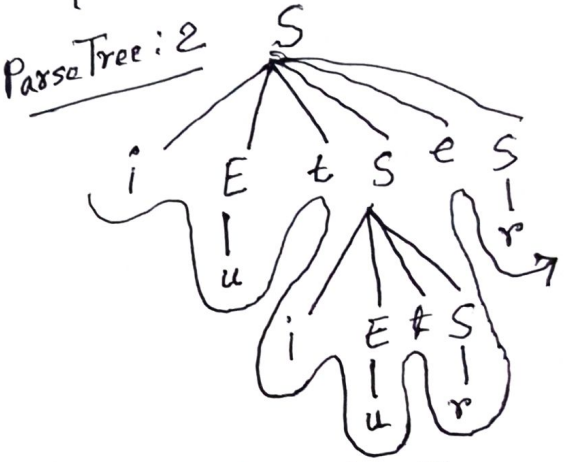
Statement : S
Expression : E
if : i
then : t
return : r
else : e
true : u
false : f

Parse Tree-1



iutiutrer

Parse Tree : 2



iutiutrer

There are two parse trees for the same string "iutiutrer".
Therefore, Grammar G1 is ambiguous.

b) Re-locatable machine code: A hypothetical machine code with logical addressing.
Absolute machine code: A real machine code or byte code with physical addressing.

Example:

```
MOV a, R1
ADD #2, R1
MOV R1, b
```

Assembly code

```
0001  01  00  00000000*
0011  01  10  00000010
0010  01  00  00000100*
```

Re-Locatable machine code

considering address of 'a' and 'b' are '0' and '4' respectively.

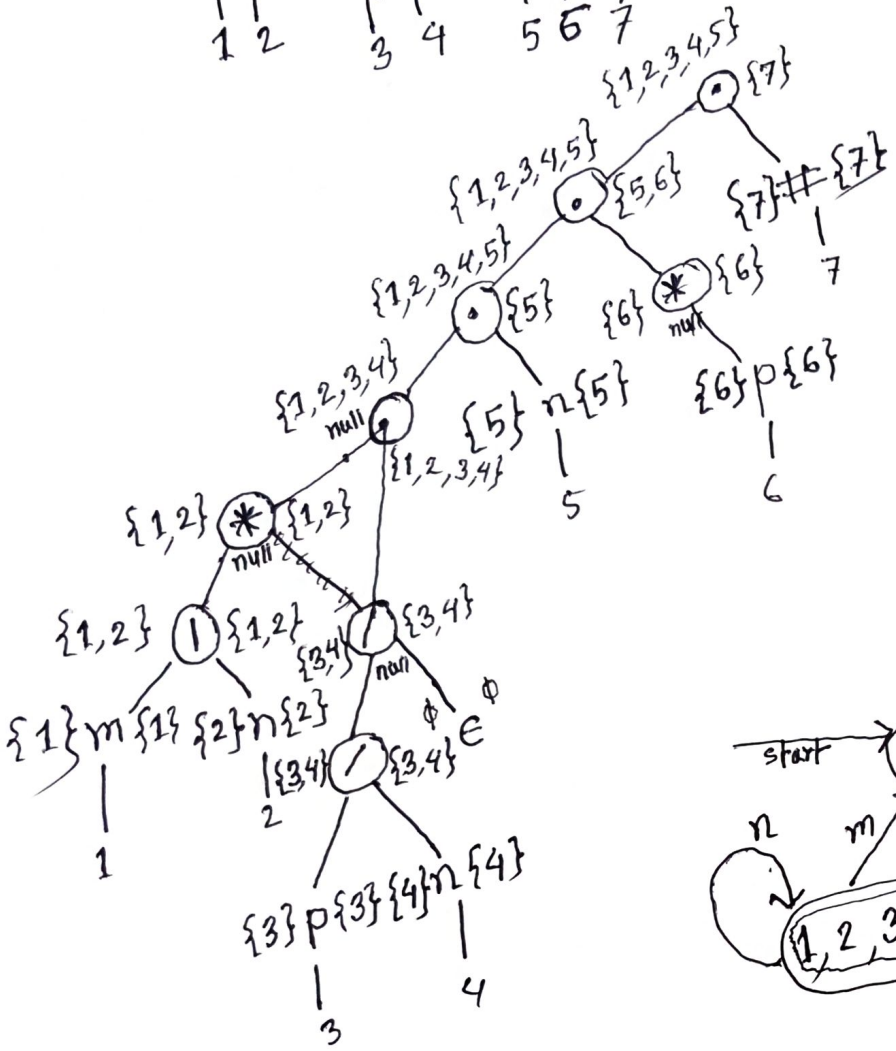
```
0001 01 00 00001111
0011 01 10 00000010
0010 01 00 00010011
```

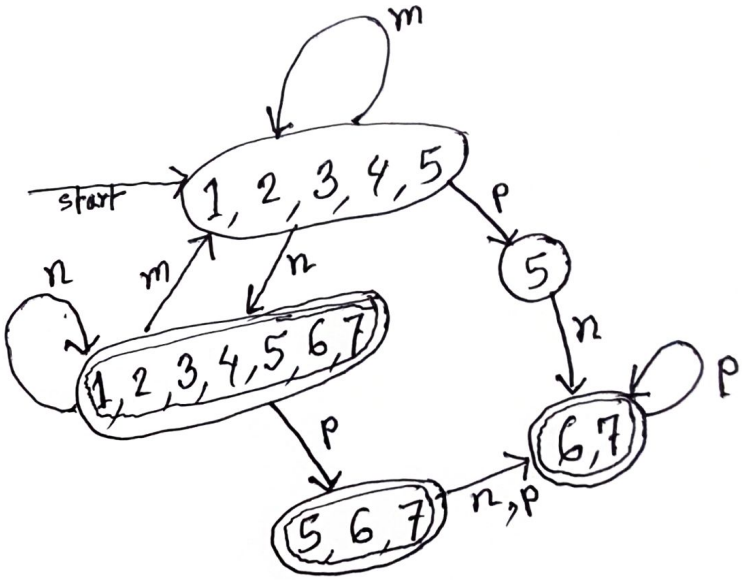
Absolute machine code considering offset value b = 00001111

2. RE: $(m|n)^*(p|n|\epsilon)np^*$

ARE: $(m|n)^*.(p|n|\epsilon).n.p^*.\#$

1 2 3 4 5 6 7





DFA

Symbol	Position Number	Followpos
m	1	1, 2, 3, 4, 5
n	2	1, 2, 3, 4, 5
p	3	5
n	4	5
n	5	6, 7
p	6	6, 7
#	7	—

3.(a)

G2:

$D \rightarrow TL$
$T \rightarrow int \mid float$
$L \rightarrow id, L \mid id$

⇓

$D \rightarrow TL$
$T \rightarrow int \mid float$
$L \rightarrow id\ L'$
$L' \rightarrow , L \mid \epsilon$

} Left Factored Grammar

Declaration : D
Type : T
VariableList : L
identifier : id

(b)

	FIRST	FOLLOW
$D \rightarrow TL$	{int, float}	{$}
$T \rightarrow int \mid float$	{int, float}	{id}
$L \rightarrow id\ L'$	{id}	{$}
$L' \rightarrow , L \mid \epsilon$	{, , ε}	{$}

$Fi(D) = Fi(TL)$
$= Fi(T)$
$= \{int, float\}$
$Fi(T) = \{int, float\}$

$Fi(L) = Fi(id\ L')$
$= \{id\}$
$Fi(L') = \{, , \epsilon\}$
$Fo(D) = \{\$\}$
$Fo(T) = Fi(L) = \{id\}$
$Fo(L) = Fo(D) \cup Fo(L')$
$= \{\$\} \cup Fo(L)$
$= \{\$\}$
$Fo(L') = Fo(L)$
$= \{\$\}$

(c)

	int	float	id	,	$
D	D→TL	D→TL			
T	T→int	T→float			
L			L→idL'		
L'				L'→,L	L'→ε

LL(1) parse tree

(d)

Stack	input	Output
$D	int id, id, id$	
$LT	int id, id, id$	D→TL
$L int	int id, id, id$	T→int
$L	id, id, id$	Drop int
$L'id	id, id, id$	L→idL'
$L'	, id, id$	Drop id
$L,	, id, id$	L'→,L
$L	id, id$	Drop ,
$L'id	id, id$	L→idL'
$L'	, id$	Drop id
$L,	, id$	L'→,L
$L	id$	Drop ,
$L'id	id$	L→idL'
$L'	$	Drop id
$	$	L'→ε
		Accept