AMOGH MISHRA NLP-HW#2 am15323

2. S → NP VP | PRP → they
NP → Adj NP | N → potatoes
NP → PRP | Adj → baking
NP → N | V → baking
VP → V NP | V → are
VP → AUX V NP | AUX → are

"they are baking potatoes"

Chart [0]

(a) S → . NP VP [0,0] PRP → . they [0,0] ✓ (scan)
NP → . Adj NP [0,0] N → . potatoes [0,0]
NP → . PRP [0,0]
NP → . N [0,0]
Adj → . baking [0,0]

chart[1]

PRP → they . [0,1] (now: complete will run)
NP → PRP . [0,1] (")
~~NPS~~ S → NP . VP [0,1] (predict will run)
VP → . V NP [1,1] "
VP → . AUX V NP [1,1] "
V → . baking [1,1] (~~complete~~ skip) ✗
V → . are [1,1] (~~complete~~ " scan) ✓
AUX → . are [1,1] (" scan) ✓

Chart [2]

V → are . [1,2] (complete)
AUX → are . [1,2] "

VP → V.NP [1,2] (predict)

VP → AUX . V NP [1,2] "

NP → . N [2,2] "

NP → . Adj NP [2,2] "

NP → . PRP [2,2] "

V → . baking [2,2] scan ✓

V → . are [2,2] skip x

N → . potatoes [2,2] skip "

Adj → . baking [2,2] ~~skip~~ scan ✓

PRP → . they [2,2]

chart [3]

V → baking. [2,3] (complete)

Adj → baking. [2,3] (complete)

VP → ~~V.NP~~ AUX V . NP [2,3] (predict)

NP → Adj . NP [2,3] (predict)

NP → . N [3,3] "

NP → . Adj NP [3,3] "

NP → . PRP [3,3] "

Adj → . Baking [3,3] "

PRP → . they [3,3] skip x

N → . potatoes [3,3] scan ✓

~~VP~~ .

chart [4]

N → potatoes . [3,4] complete

NP → N. [3,4]

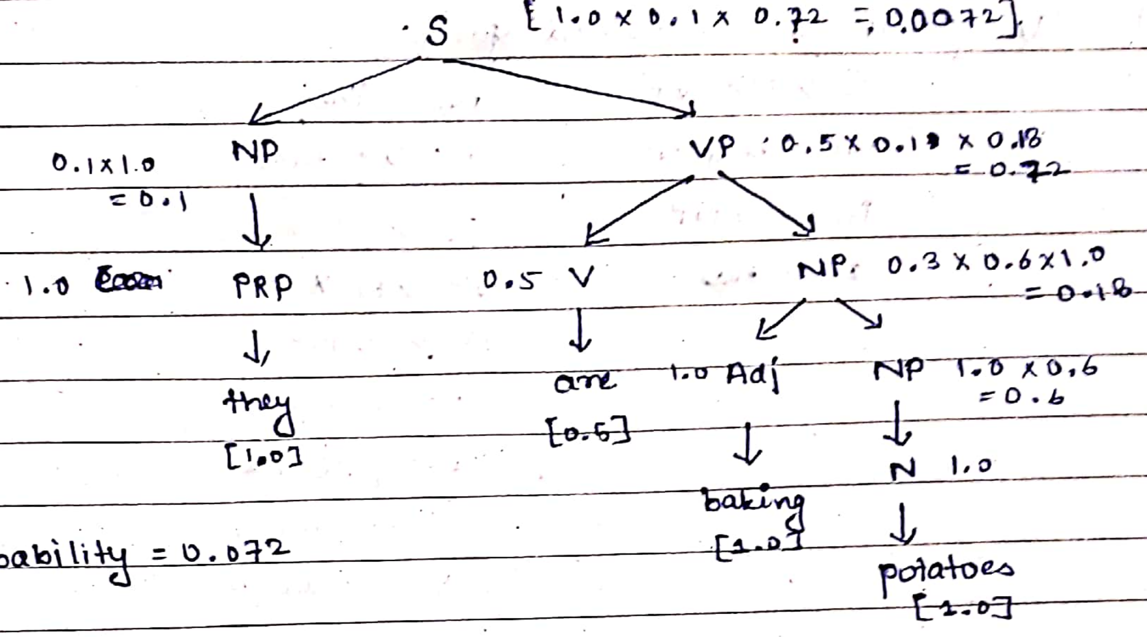
VP → AUX V NP. [1,4] → VP → V NP. [1,4]

NP → Adj NP. [2,4]

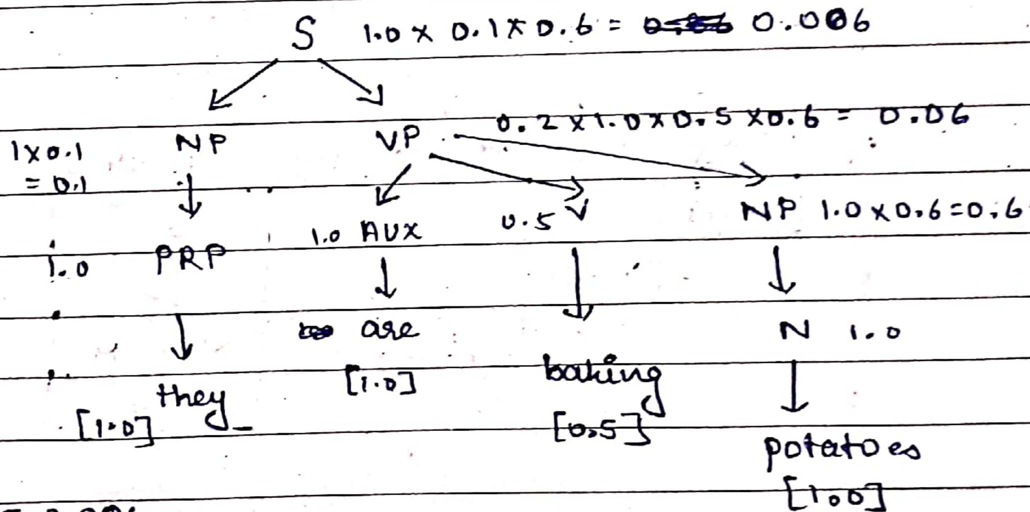
S → NP VP. [0,4]

~~VP → V NP.~~ E

2(b)



probability = 0.072



probability = 0.006

Ans. 3 (a) In CNF, there are 2 rules: A → a

or A → BC

B → b

C → c

∴ if A → B and B → b ⇒ A → b

if A → BCD we can convert this to CNF,

A → ED

E → BC

(b) Using CNF discussed, transform

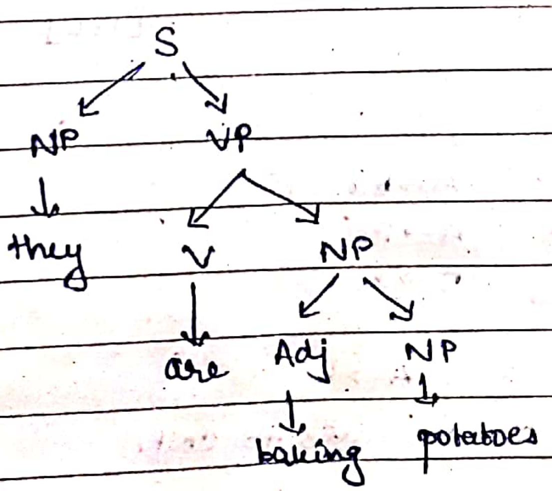
S → NP VP	NP → they
NP → Adj NP	NP → potatoes
NP → V NP	
~~VP → Aux~~	VP → AUX A
	A → V NP
	PRP → they
	N → potatoes
	Adj → baking
	V → baking
	V → are
	Aux → are

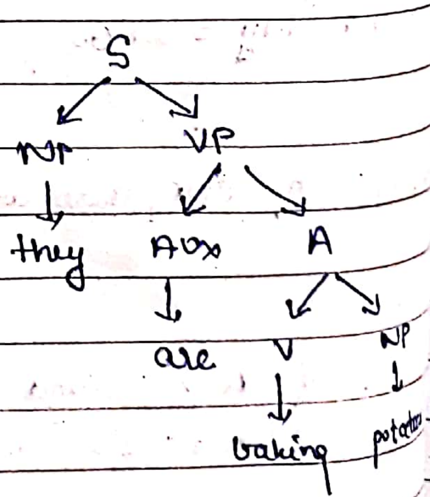
(b)

(i)

	they	are	~~booki~~ baking	potatoes
0	NP, PRP	-	-	S
1		V, AUX	-	VP
2			Adj, V	NP, VP, A
3				N, NP

(ii)





"2 possible parse trees"

Q4.

(i) A: { }

σ: root β: [he, sent, her, a, funny, meme, today]

(ii) A: { }

σ: he root (shift) β: [sent, her, a, funny, meme, today]

(iii) A: he sent (left-arc)

σ: root β: [sent, her, . . .]

(iv) A: he sent

(shift)

σ: sent root β: [her, a, funny . . .]

(v) A: he sent her

(right-arc)

σ: root β: [sent, a, funny, . . .]

(vi) A: he sent her

(shift)

σ: sent root β: [a, funny, . . .]

(vii) A: he sent her (Shift)

a
sent
σ: root

β: [funny, meme, today]

(viii) A: he sent her (Shift)

funny
a
sent
σ: root

β: [meme, today]

(ix) A: he sent her funny meme (left-arc)

a
sent
σ: root

β: [meme, today]

(x) A: he sent her a funny meme (left-arc)

sent
σ: root

β: [meme, today]

(xi) A: he sent her a funy meme (right-arc)

σ: root

β: [sent, today]

(xii) A: he sent her a funny meme (shift)

sent
σ: root

[today]

13) A: he sent her a funny meme today

σ : root β : [sent] (right-arc)

14. A: root he sent her a funny meme today. Ans

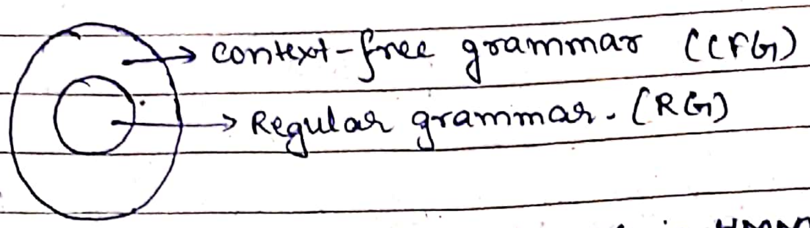
σ : [ ] β : [root] (right-arc)

15. A: → same as above (14)

(Shift)

σ : [root] β : [ ]

1(c) From the notes of Complexity class, it's seen that



i.e RG ⊂ CFG and example of RG is HMM as HMM are a type of Markov Chain which is effectively a finite state automata.

Whereas PCFG are CFG. Therefore, it is not possible to have all PCFG represented as HMM. For eg: $a^n b^m$ where $n = m$.

~~For~~ from lecture notes:→ "The set of all regular language is strictly smaller than set of CFG".

1(a) "they are baking potatoes"

PRP → V → Adj N — (i)

OR

PRP Aux V N —(ii)

Using probability → (transition and emission)

= P(PRP|start). P(V|PRP). P(Adj|V). P(N|Adj).
P(they|PRP). P(are|V). P(baking|Adj). P(potato|N)

= 0.1 × 0.8 × 0.3 × 0.6 × 1.0 × 0.5 × 1.0 × 1.0 = 0.072

Similarly for (ii)

P(tags, words) = P(PRP|start). P(AUX|PRP). P(V|AUX). P(N|V).
P(they|PRP). P(are|AUX). P(baking|V). P(potato|N)

= 0.1 × 0.2 × 1.0 × 0.6 × 1.0 × 1.0 × 0.5 × 1.0 = 0.006

Scanned by CamScanner

1(b) HMM model representing POS sequence: