4/20 Mathematical Linguistics

A CFG G is self-embedding

$\iff$ for some nonterminal A and $x, y \in \Sigma^*$,

$$A \overset{*}{\underset{G}{\Rightarrow}} xAy \ \wedge$$

$$\underline{x \neq \varepsilon \ \wedge \ y \neq \varepsilon}$$

Theorem (Chomsky) Let G be a CFG.

G is not self-embedding $\Rightarrow$ $L(G)$ is regular

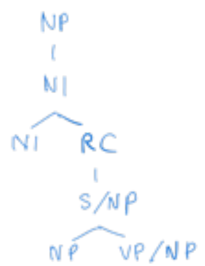
Problem 1.7

Let G consist of the following productions:

$S \to NP\ VP$	$RC \to S/NP$
$NP \to N1$	$S/NP \to NP\ VP/NP$
$N1 \to N$	$VP/NP \to Vt$
$N1 \to N1\ RC$	$N \to$ people
	$Vt \to$ see

Show that $\{ w \in \{\text{people}, \text{see}\}^* \mid NP \Rightarrow^* \text{people } w \}$ is generated by the grammar

$$S \to \varepsilon$$
$$S \to S \text{ people } S \text{ see}$$



X or no X

war or no war

$\{ xx \mid x \in \{a,b\}^* \}$ $\{ xx \mid x \in D_1^* \}$

Chomsky 2004:

. . . in all of this work of the late 50s and early 60s there was only one result that I know of that had any linguistic significance. ... That's the fact that there's a constructive procedure to map context-free grammars into a strongly equivalent — crucially — non-deterministic push-down storage automaton. ... It's not interesting mathematics. It's just a constructive procedure. In fact, it's what underlies every parser. That's why every parser is a non-deterministic push-down storage automaton. It doesn't contribute much, but it explains why that's what every parser is. In so far as language is more or less context-free, you can parse it that way. But apart from that, I don't know of any results that are interesting. I mean, some results are amusing, but not linguistically significant.

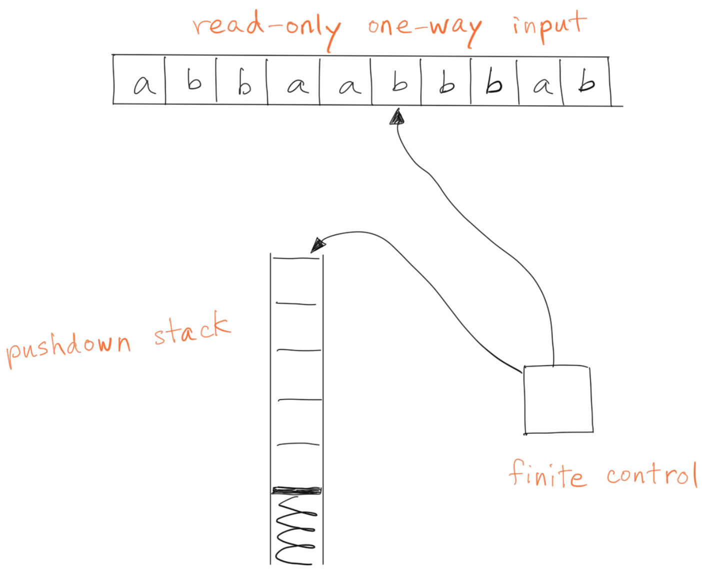
Recognizer

$$\text{input string } w \longrightarrow \begin{cases} \text{yes} & w \in L(G) \\ \text{no} & w \notin L(G) \end{cases}$$

Parser

$$\text{input string} \longrightarrow \begin{cases} \text{yes} \,/\, \text{parse trees for } w \\ \qquad\quad \text{derivation trees} \\ \text{no} \end{cases}$$

Pushdown Automaton



Top-down Recognizer

$$(S,\ \text{input}) \vdash^{*} (\varepsilon, \varepsilon)$$

(stack, remainder of input)

$$(X\alpha, x) \vdash (Y_1 \ldots Y_n\alpha, x) \qquad \text{predict } X \to Y_1 \ldots Y_n$$

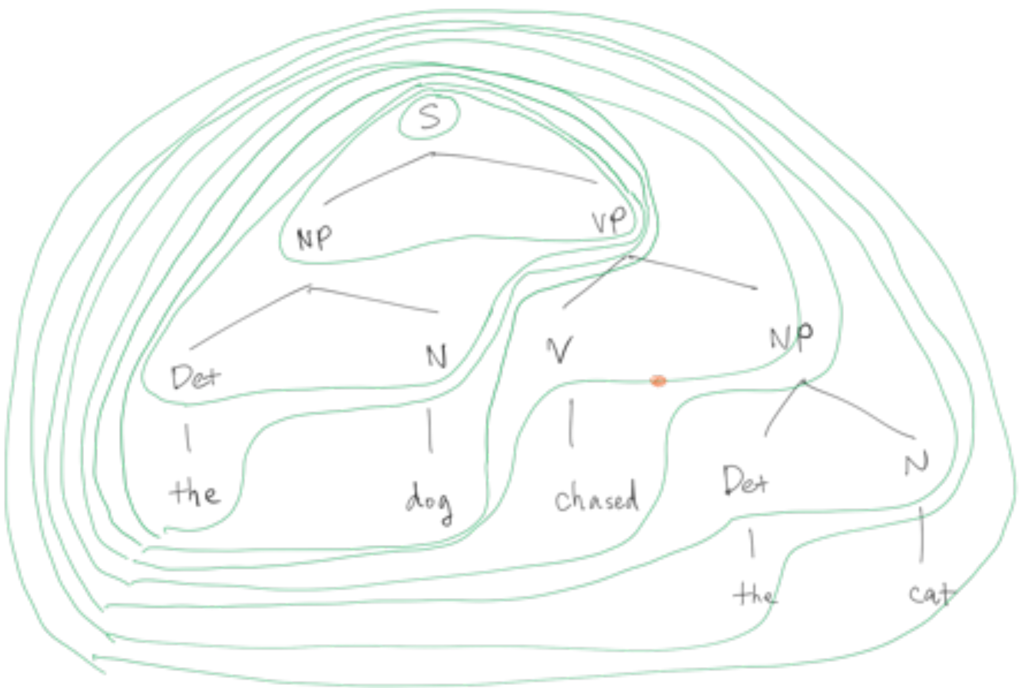
$$(a\alpha, ax) \vdash (\alpha, x) \qquad \text{match}$$

accepting computation ≈ leftmost derivation

$S \to NP\ VP$
$NP \to Det\ N$
$VP \to V\ NP$ $\quad VP \to V$
$Det \to \text{the}$
$N \to \text{dog}$
$N \to \text{cat}$
$V \to \text{chased}$

(S, the dog chased the cat) $\vdash$ (NP VP, the dog …) predict
$\vdash$ (Det N VP, the dog …) predict
$\vdash$ (the N VP, the dog …) predict
$\vdash$ (N VP, dog chased the cat)

$S \Rightarrow NP\ VP \Rightarrow Det\ N\ VP \Rightarrow \text{the}\ N\ VP$:
$\Rightarrow$



(V NP, chased the cat)

Bottom-up Recognizer

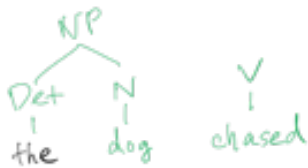
$$(\varepsilon, \text{input}) \vdash^* (S, \varepsilon)$$

$(\alpha Y_1 \dots Y_n, x) \vdash (\alpha X, x)$ reduce $X \to Y_1 \dots Y_n$

$(\alpha, ax) \vdash (\alpha a, x)$ shift

accepting computation $\approx$ rightmost derivation in reverse

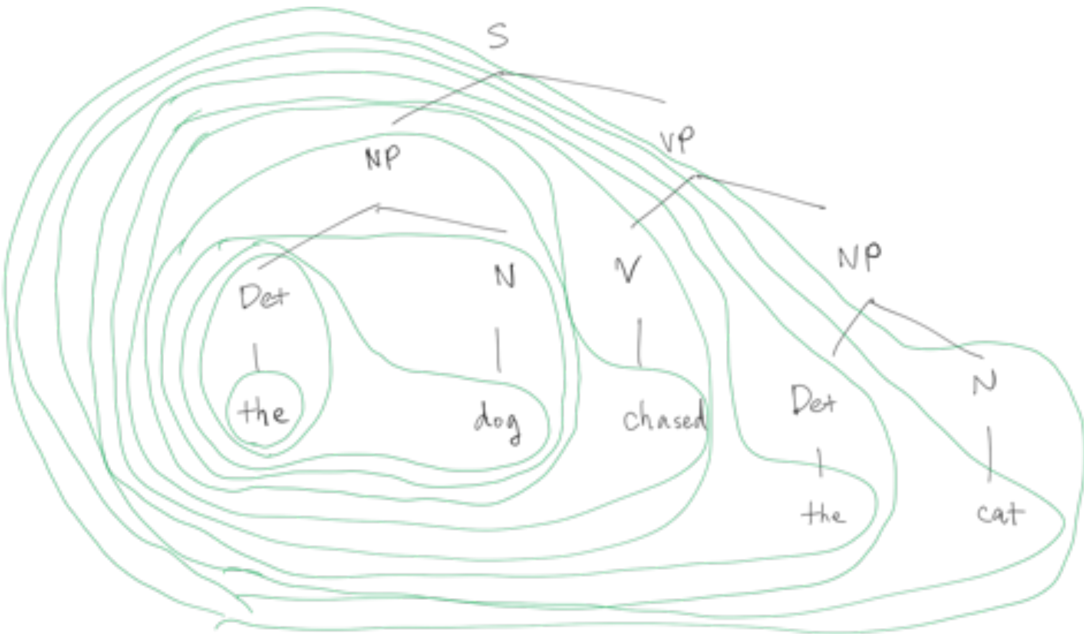
$S \to NP\ VP$
$NP \to Det\ N$
$VP \to V\ NP$ $\quad VP \to V$
$Det \to$ the
$N \to$ dog
$N \to$ cat
$V \to$ chased



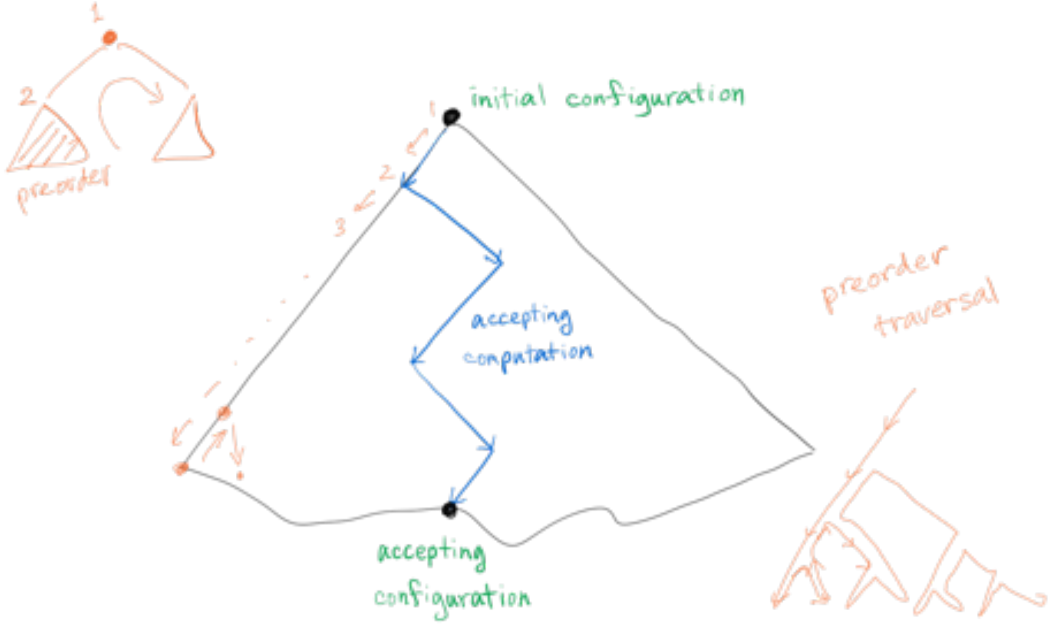
$(\varepsilon$, the dog chased the cat$)$ $\vdash$ (the, dog …) shift
$\vdash$ (Det, dog …) reduce
$\vdash$ (Det dog, chased the cat) shift
⋮

$S \Rightarrow NP\ VP \Rightarrow NP\ V\ NP \Rightarrow NP\ V\ Det\ N \Rightarrow \cdots$

$\vdash (NP\ V\ Det\ N, \varepsilon) \vdash (NP\ V\ NP, \varepsilon) \vdash (NP\ VP, \varepsilon)$
$\vdash (S, \varepsilon)$



Computation Tree



Left Recursion

$$(X\alpha, x)$$

$$\swarrow$$

$$(X\gamma\alpha, x)$$

$$\swarrow$$

$$(X\gamma\gamma\alpha, x)$$

$$\swarrow$$

$$\vdots$$

$$X \rightarrow X\gamma$$

left-recursive production

computation tree of top-down recognizer

When Bottom-up Recognizer Loops

$$(\alpha, x) \rightarrow (\alpha X, x) \rightarrow (\alpha X X, x) \rightarrow \cdots$$

$X \rightarrow \varepsilon$
epsilon production

$$(\alpha X, x) \rightarrow \cdots \rightarrow (\alpha X, x) \rightarrow \cdots \rightarrow (\alpha X, x) \rightarrow \cdots$$

$X \stackrel{+}{\Rightarrow} X$
cycle

(NP chased, the cat)
(NP V, the cat) (NP chased the, cat)
(NP V the, cat) (NP chased Det, cat) (NP chased Det cat, ε)
(NP V Det, cat) (NP V the cat, ε) (NP chased Det cat, ε) (NP chased Det N, ε)
(NP V Det cat, ε) (NP V the N, ε) (NP chased Det N, ε) (NP chased NP, ε)
(NP V Det N, ε) (NP chased NP, ε)
(NP V NP, ε)
(NP VP, ε)
(S, ε)

(Det N chased, the cat)

(Det N V, the cat) (Det N chased the, cat)

(Det N V the, cat) (Det N chased Det, cat) (Det N chased Det cat, ε)

(Det N V Det, cat) (Det N V the cat, ε) (Det N chased Det cat, ε) (Det N chased Det N, ε)

(Det N V Det cat, ε) (Det N V the N, ε) (Det N chased Det N, ε) (Det N chased NP, ε)

(Det N V Det N, ε) (Det N chased NP, ε)

(Det N V NP, ε)

(Det N VP, ε)

left-branching

center-embedded

right-branching

Left-Corner Recognizer

$$(\sim S,\ \text{input}) \vdash^{*} (\varepsilon, \varepsilon)$$

$(\alpha, ax) \vdash (a\alpha, x)$ shift

$(\sim a\alpha, ax) \vdash (\alpha, x)$ match

$(Y_1\alpha, x) \vdash (\sim Y_2 \ldots \sim Y_n X \alpha, x)$ reduce/predict

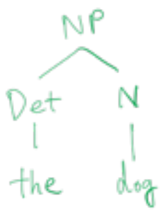
$(Y_1 \sim X \alpha, x) \vdash (\sim Y_2 \ldots \sim Y_n \alpha, x)$ reduce/predict/complete

stack alphabet

$N \cup \Sigma \cup \{\sim X \mid X \in N \cup \Sigma\}$

$X \to Y_1 \ldots Y_n$

S



$S \to NP\ VP$
$NP \to Det\ N$
$VP \to V\ NP$ $\quad VP \to V$
Det → the
N → dog
N → cat
V → chased

$\boxed{\sim S}$

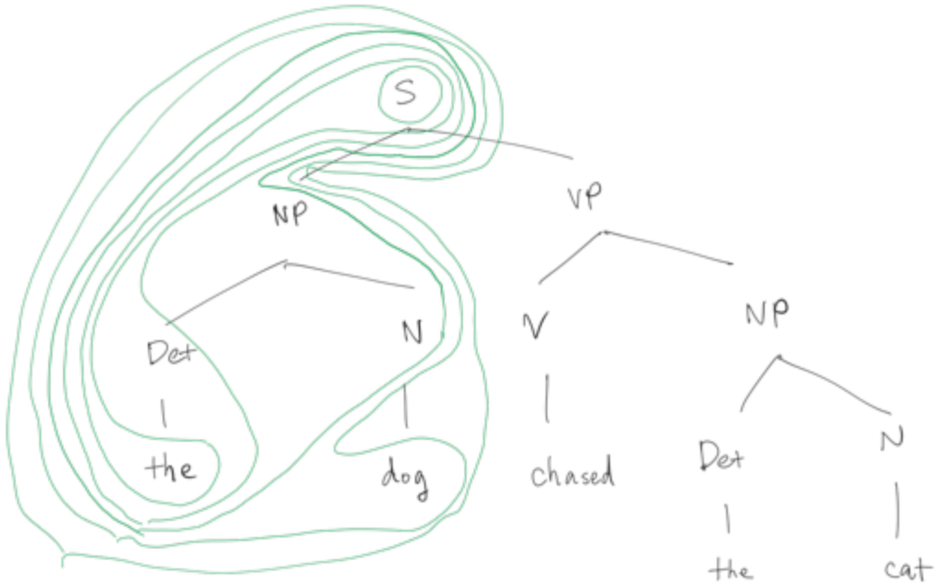
$\boxed{\text{the} \sim S}$

$\boxed{Det \sim S}$

$\boxed{\sim N\ NP \sim S}$

$\boxed{\text{dog} \sim N\ NP \sim S}$

$\boxed{NP \sim S}$


S
NP
VP
Det
N
V
NP
the
dog
chased
Det
N
the
cat