Context-free languages and grammars

September 19, 2019

CS/ECE 374 A

Ian Ludden

Reminders

- Midterm 1: Monday, Sep 30, 7-9 p.m.
  - DRES: reserve ASAP
  - Review session(s) next week
  - This is the last material that may be covered by the exam
- Homework 3 due next Tuesday

Learning Objectives

By the end of this lecture, you will be able to:

- Recall the definition of a context-free grammar/language (CFG/CFL).
- Give examples of CFGs/CFLs.
- Derive strings generated by CFGs using parse trees.
- Determine the CFL generated by a CFG.
- Compare/contrast CFLs with regular languages.
- Identify CFGs in Chomsky normal form.

Context-free = regular + recursion

Regular languages

- Sequencing (A · B)
- Branching (A + B)
- Repetition (A*)

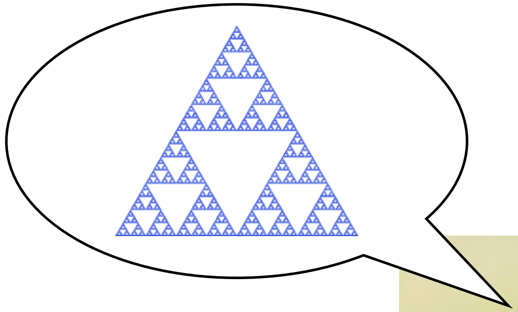
Context-free = regular + recursion

~~Regular~~ Context-free languages

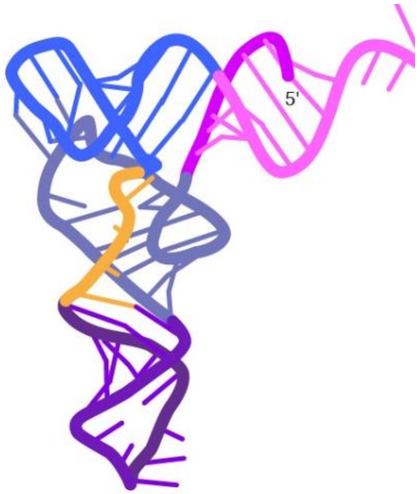
- Sequencing (A · B)
- Branching (A + B)
- Repetition (A*)
- Recursion

All regular languages are context-free.
(Proof in Section 5.5 of lecture notes.)

Motivation



- Not all languages are regular
  - $L = \{0^n 1^n \mid n \geq 0\}$
- Recursive languages occur in nature
  - Gentner, T. Q.; Fenn, K. M.; Margoliash, D.; Nusbaum, H. C. (2006). "Recursive syntactic pattern learning by songbirds." *Nature.* **440** (7088): 1204-1207.
- Natural Language Processing (NLP)
  - Charniak, E. (1997). "Statistical Parsing with a Context-Free Grammar and Word Statistics." *AAAI/IAAI.*
- Probabilistic modeling of RNA structures
  - Sakakibara Y.; Brown M.; Hughey R.; Mian I. S.; et al. (1994). "Stochastic context-free grammars for tRNA modelling." *Nucleic Acids Research.* **22** (23): 5112–5120.



Formal Definition

- A ***context-free grammar*** is a structure defined by:
  - A finite set Σ of *symbols* or ***terminals***
  - A finite set Γ of ***non-terminals*** (disjoint from Σ)
  - A finite set *R* of ***production rules*** of the form *A -> w*, where A is a non-terminal and w is a string of symbols and non-terminals
  - A starting non-terminal, typically *S*

$G = (\Sigma, \Gamma, R, S)$

Example

Context-free grammar for (a subset of) English sentences
Symbols are words, strings are sentences

$$\langle\text{sentence}\rangle \to \langle\text{noun phrase}\rangle\langle\text{verb phrase}\rangle\langle\text{noun phrase}\rangle$$

$$\langle\text{noun phrase}\rangle \to \langle\text{adjective phrase}\rangle\langle\text{noun}\rangle$$

$$\langle\text{adj. phrase}\rangle \to \langle\text{article}\rangle \mid \langle\text{possessive}\rangle \mid \langle\text{adjective phrase}\rangle\langle\text{adjective}\rangle$$

$$\langle\text{verb phrase}\rangle \to \langle\text{verb}\rangle \mid \langle\text{adverb}\rangle\langle\text{verb phrase}\rangle$$

$$\langle\text{noun}\rangle \to \text{dog} \mid \text{trousers} \mid \text{daughter} \mid \text{nose} \mid \text{homework} \mid \text{time lord} \mid \text{pony} \mid \cdots$$

$$\langle\text{article}\rangle \to \text{the} \mid \text{a} \mid \text{some} \mid \text{every} \mid \text{that} \mid \cdots$$

$$\langle\text{possessive}\rangle \to \langle\text{noun phrase}\rangle\text{'s} \mid \text{my} \mid \text{your} \mid \text{his} \mid \text{her} \mid \cdots$$

$$\langle\text{adjective}\rangle \to \text{friendly} \mid \text{furious} \mid \text{moist} \mid \text{green} \mid \text{severed} \mid \text{timey-wimey} \mid \text{little} \mid \cdots$$

$$\langle\text{verb}\rangle \to \text{ate} \mid \text{found} \mid \text{wrote} \mid \text{killed} \mid \text{mangled} \mid \text{saved} \mid \text{invented} \mid \text{broke} \mid \cdots$$

$$\langle\text{adverb}\rangle \to \text{squarely} \mid \text{incompetently} \mid \text{barely} \mid \text{sort of} \mid \text{awkwardly} \mid \text{totally} \mid \cdots$$

Example

$\Sigma = \{0, 1\}$ — Terminals

$\Gamma = \{S, A, B, C\}$ — Non-terminals

$S \to A \mid B$

$A \to 0A \mid 0C$

$B \to B1 \mid C1$

$C \to \varepsilon \mid 0C1$

Production rules

'|' means 'or'

$xAy \rightsquigarrow xwy$

(produces immediately)

$S \rightsquigarrow^* w$

(produces eventually)

Example

$\Sigma = \{0, 1\}$

$\Gamma = \{S, A, B, C\}$

$S \to A \mid B$

$A \to 0A \mid 0C$

$B \to B1 \mid C1$

$C \to \varepsilon \mid 0C1$

$S \to A$

$\to 0A$

$\to 00C$

$\to 000C1$

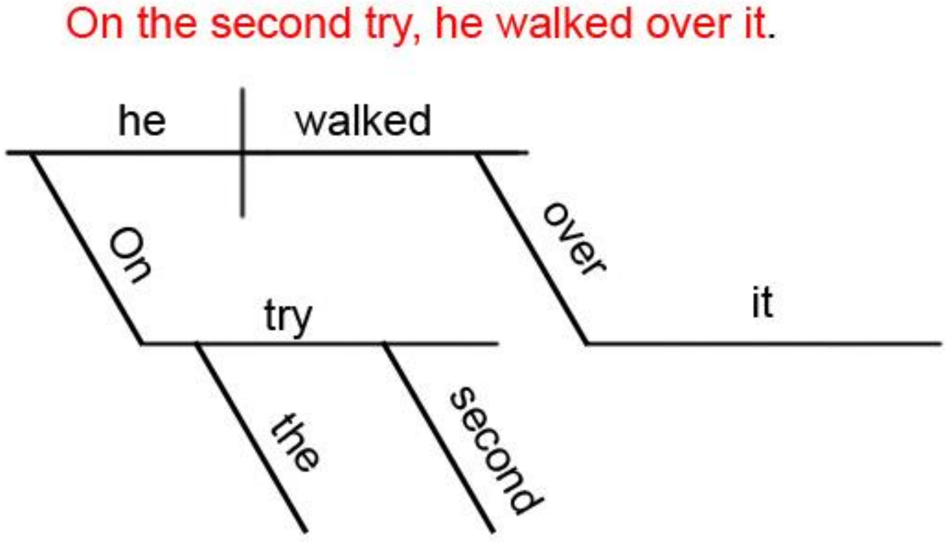
$\to 0000C11$

$\to 0000\varepsilon 11$

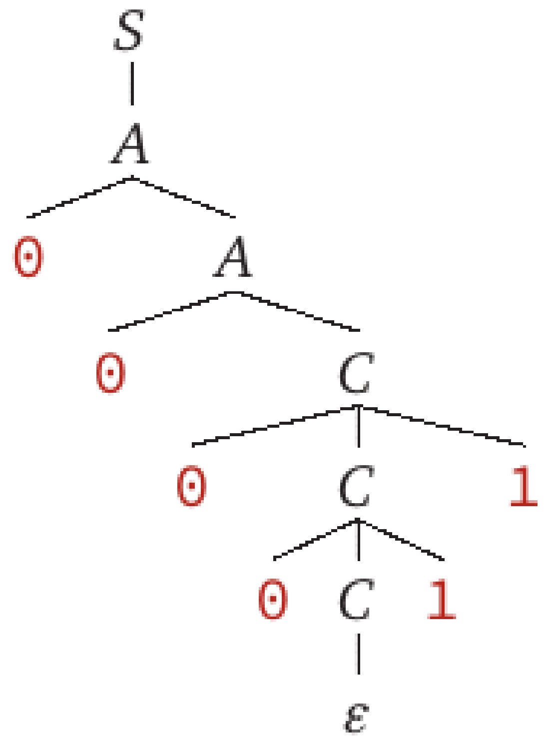
$\to 000011$

Surely there's a more descriptive way to write this derivation…

Old-school English class approach



Parse tree



Parse trees visualize string derivations.

$\Sigma = \{0, 1\}$

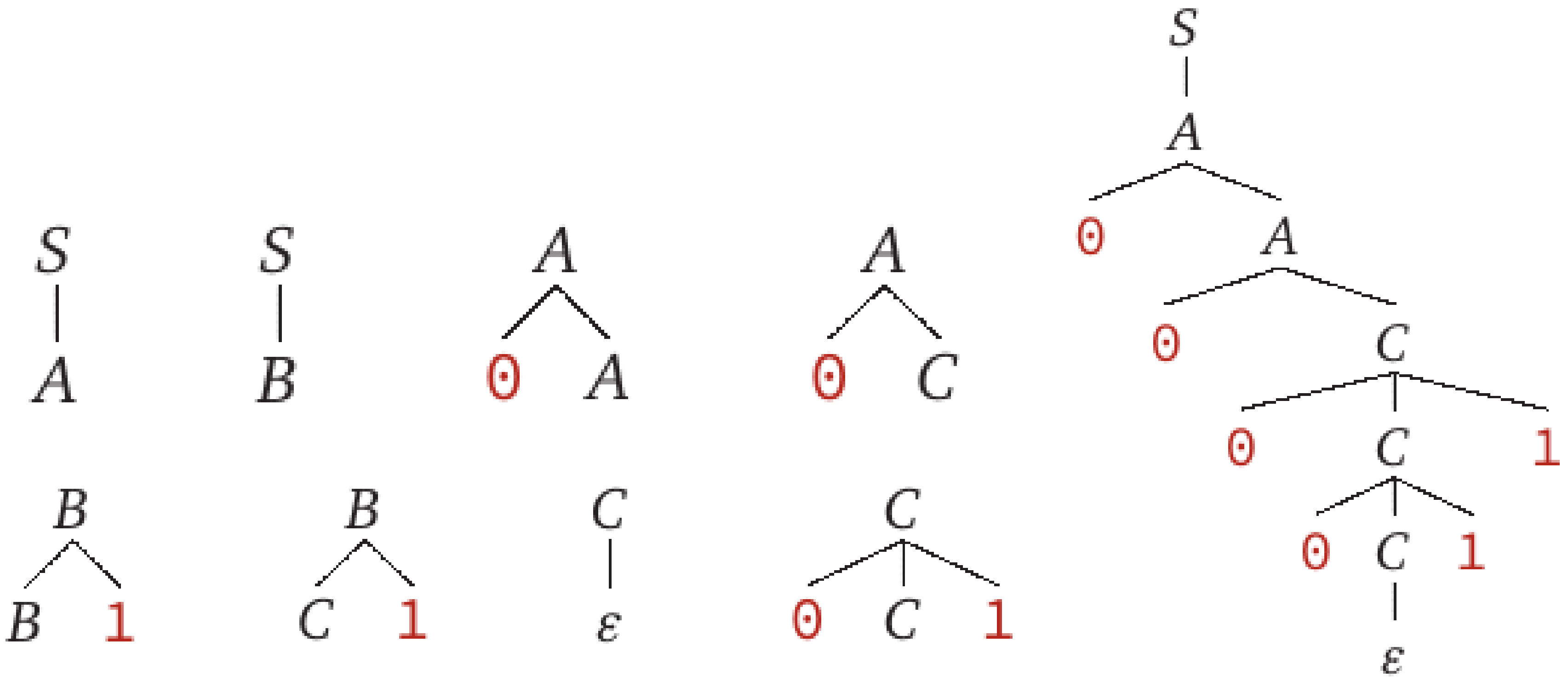
$\Gamma = \{S, A, B, C\}$

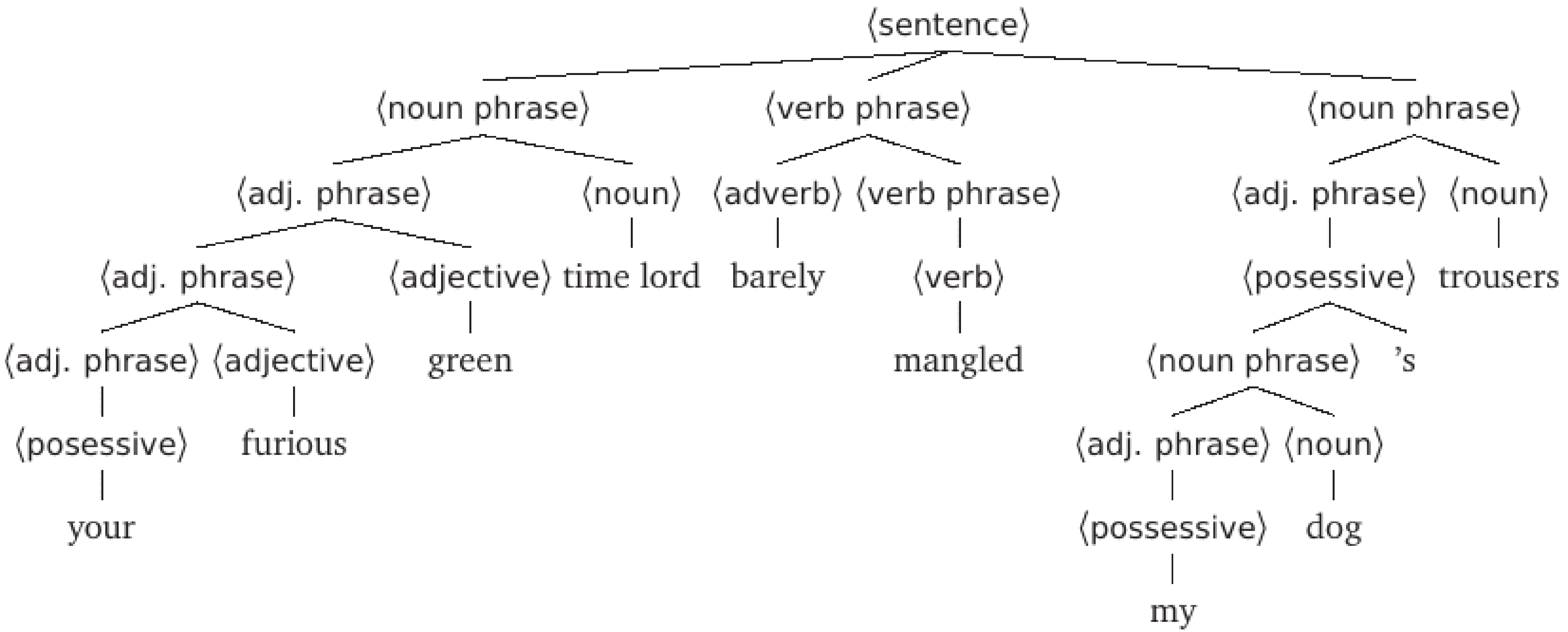
$S \rightarrow A \mid B$

$A \rightarrow 0A \mid 0C$

$B \rightarrow B1 \mid C1$

$C \rightarrow \varepsilon \mid 0C1$




⟨sentence⟩
⟨noun phrase⟩
⟨verb phrase⟩
⟨noun phrase⟩
⟨adj. phrase⟩
⟨noun⟩
⟨adverb⟩
⟨verb phrase⟩
⟨adj. phrase⟩
⟨noun⟩
⟨adj. phrase⟩
⟨adjective⟩
time lord
barely
⟨verb⟩
⟨posessive⟩
trousers
⟨adj. phrase⟩
⟨adjective⟩
green
mangled
⟨noun phrase⟩
’s
⟨posessive⟩
furious
⟨adj. phrase⟩
⟨noun⟩
your
⟨possessive⟩
dog
my

Exercise: Parse trees

$\Sigma = \{1, 2, +, x\}$

$\Gamma = \{S, A, M, C\}$

$S \rightarrow A \mid M \mid 1 \mid 2$

$A \rightarrow S + S$

$M \rightarrow S \; x \; S$

Activity (2 min.)

1. Derive 2 + 1 x 1 from this grammar using a parse tree.

2. Compare with neighbor(s).

Exercise: Parse trees

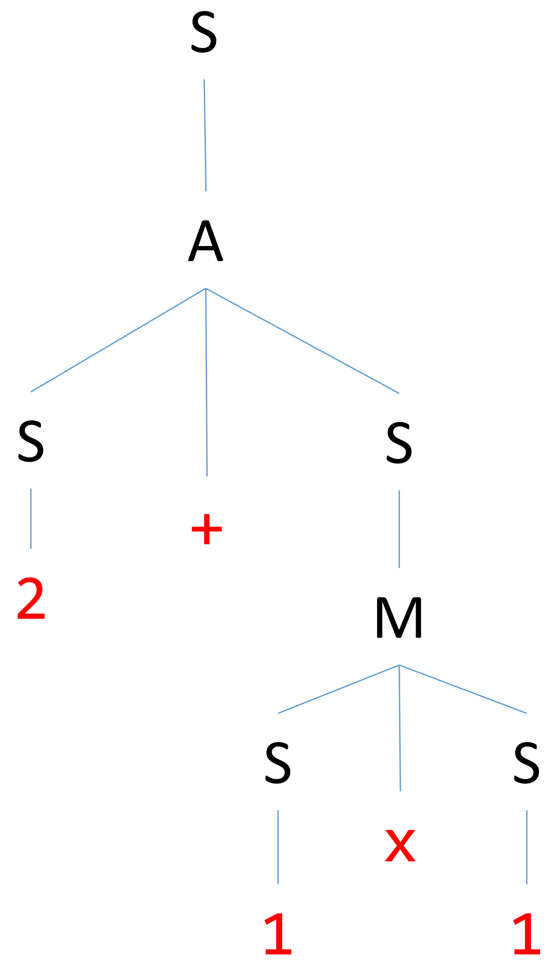
$\Sigma = \{1, 2, +, x\}$

$\Gamma = \{S, A, M, C\}$

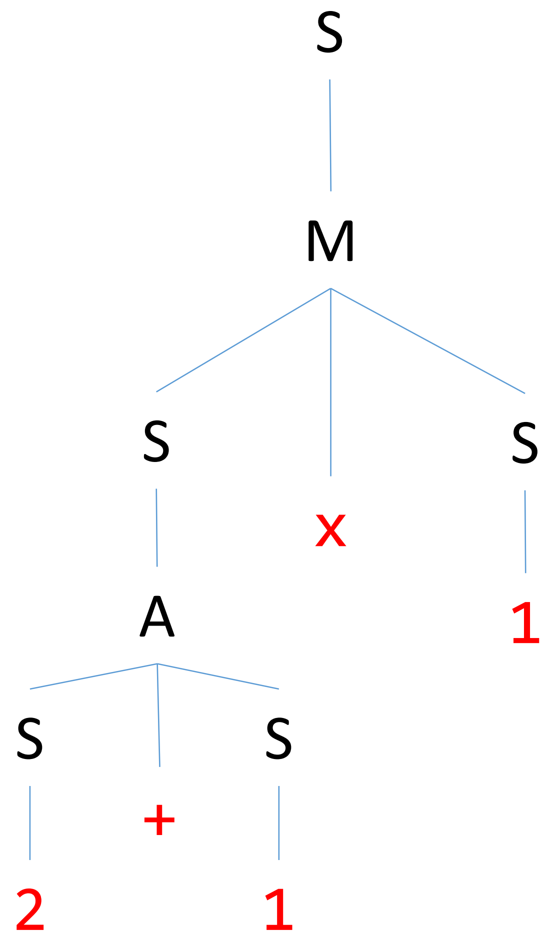
$S \rightarrow A \mid M \mid 1 \mid 2$

$A \rightarrow S + S$

$M \rightarrow S \; x \; S$



2 + (1 x 1) = 4



(2 + 1) x 1 = 3

Ambiguity

4 (disambiguation)

From Wikipedia, the free encyclopedia

- A string w is ***ambiguous*** with respect to a grammar if there is more than one parse tree for w.

- A grammar G is ***ambiguous*** if some string is ambiguous with respect to G.

- A context-free language L is ***inherently ambiguous*** if every context-free grammar that generates L is ambiguous.
  (Contrived examples)

Disambiguating

$\Sigma = \{1, 2, +, x\}$
$\Gamma = \{S, A, M, C\}$

$S \rightarrow A \mid M \mid 1 \mid 2$
$A \rightarrow S + S$
$M \rightarrow S \times S$

$\Sigma = \{1, 2, +, x, (,)\}$
$\Gamma = \{S, A, M, C\}$

$S \rightarrow A \mid M \mid 1 \mid 2$
$A \rightarrow (S + S)$
$M \rightarrow (S \times S)$

No longer ambiguous!

Arithmetic Expressions

- Arithmetic expressions, possibly with redundant parentheses, over the variables X and Y:

$$E \to E + T \mid T \qquad \text{(expressions)}$$
$$T \to T \times F \mid F \qquad \text{(terms)}$$
$$F \to (E) \mid \mathsf{X} \mid \mathsf{Y} \qquad \text{(factors)}$$

Every Eexpression is a sum of Terms, every Term is a product of Factors, and every Factor is either a variable or a parenthesized Eexpression.

Regular Expressions

- Regular expressions over the alphabet $\{0,1\}$ *without* redundant parentheses

$S \to T \mid T+S$	(Regular expressions)
$T \to F \mid FT$	(Terms = summable expressions)
$F \to \emptyset \mid W \mid (T+S) \mid X* \mid (Y)*$	(Factors = concatenable expressions)
$X \to \emptyset \mid \varepsilon \mid 0 \mid 1$	(Directly starrable expressions)
$Y \to T+S \mid F \bullet T \mid X* \mid (Y)* \mid ZZ$	(Starrable expressions needing parens)
$W \to \varepsilon \mid Z$	(Words = strings)
$Z \to 0 \mid 1 \mid ZZ$	(Non-empty strings)

From grammars to languages

- For non-terminal A, L(A) is the set of all strings generated by A.
- Given context-free grammar $G = (\Sigma, \Gamma, R, S)$, L($G$) = L($S$).
- A ***context-free language*** is the language generated by a context-free grammar.
- Often easiest to examine “later” rules first when determining L(G).

What language do you speak?

$\Sigma = \{0, 1\}$

$\Gamma = \{S, A, B, C\}$

$S \rightarrow A \mid B$

$A \rightarrow 0A \mid 0C$

$B \rightarrow B1 \mid C1$

$C \rightarrow \varepsilon \mid 0C1$

<u>Lemma</u>: $L(C) = \{0^m 1^n \mid m = n \geq 0\}$.

<u>Proof</u> ($\supseteq$):

Let n be an arbitrary non-negative integer.

Assume $C \rightsquigarrow^* 0^m 1^m$ for all $m < n$.

Two cases:

- $n = 0$, $0^n 1^n = \varepsilon$, $C \rightarrow \varepsilon$, done.
- $n > 1$, $C \rightarrow 0C1$. By I.H.,

$0C1 \rightsquigarrow^* 0(0^{n-1}1^{n-1})1 = 0^n 1^n$, done.

Thus $L(C) \supseteq \{0^m 1^n \mid m = n \geq 0\}$.

What language do you speak?

$\Sigma = \{0, 1\}$

$\Gamma = \{S, A, B, C\}$

$S \to A \mid B$

$A \to 0A \mid 0C$

$B \to B1 \mid C1$

$C \to \varepsilon \mid 0C1$

<u>Lemma</u>: $L(C) = \{0^m 1^n \mid m = n \geq 0\}$.

<u>Proof</u> ($\subseteq$):

Fix w in L(C).

Assume for all x in L(C) with $|x| < |w|$,

$x = 0^m 1^m$ for some $m \geq 0$.

Two cases (first production):

- $C \to \varepsilon$, $w = \varepsilon = 0^0 1^0$, done.
- $C \to 0C1$ So $w = 0x1$ for some x in L(C). By I.H., $x = 0^m 1^m$ for some $m \geq 0$.

So $w = 0(0^m 1^m)1 = 0^{m+1} 1^{m+1}$, done.

Thus $L(C) \subseteq \{0^m 1^n \mid m = n \geq 0\}$.

What language do you speak?

$\Sigma = \{0, 1\}$

$\Gamma = \{S, A, B, C\}$

$S \rightarrow A \mid B$

$A \rightarrow 0A \mid 0C$

$B \rightarrow B1 \mid C1$

$C \rightarrow \varepsilon \mid 0C1$

$L(C) = \{0^m 1^n \mid m = n \geq 0\}$.

$L(B) = ?$

$L(A) = ?$

$L(S) = ?$

What language do you speak?

$\Sigma = \{0, 1\}$

$\Gamma = \{S, A, B, C\}$

$S \rightarrow A \mid B$

$A \rightarrow 0A \mid 0C$

$B \rightarrow B1 \mid C1$

$C \rightarrow \varepsilon \mid 0C1$

$L(C) = \{0^m 1^n \mid m = n \geq 0\}$.

$L(B) = \{0^m 1^n \mid m < n \geq 0\}$.

$L(A) = ?$

$L(S) = ?$

What language do you speak?

$\Sigma = \{0, 1\}$

$\Gamma = \{S, A, B, C\}$

$S \rightarrow A \mid B$

$A \rightarrow 0A \mid 0C$

$B \rightarrow B1 \mid C1$

$C \rightarrow \varepsilon \mid 0C1$

$L(C) = \{0^m1^n \mid m = n \geq 0\}$.

$L(B) = \{0^m1^n \mid m < n \geq 0\}$.

$L(A) = \{0^m1^n \mid m > n \geq 0\}$.

$L(S) = ?$

What language do you speak?

$\Sigma = \{0, 1\}$

$\Gamma = \{S, A, B, C\}$

$S \rightarrow A \mid B$

$A \rightarrow 0A \mid 0C$

$B \rightarrow B1 \mid C1$

$C \rightarrow \varepsilon \mid 0C1$

$L(C) = \{0^m 1^n \mid m = n \geq 0\}$.

$L(B) = \{0^m 1^n \mid m < n \geq 0\}$.

$L(A) = \{0^m 1^n \mid m > n \geq 0\}$.

$L(S) = \{0^m 1^n \mid m \neq n \geq 0\}$.

The grammar that generates a CFL is not unique.

$\Sigma = \{0, 1\}$

How to generate 0^*1^*?

$S \rightarrow \varepsilon \mid 0S \mid S1$

vs.

$S \rightarrow AB$

$A \rightarrow \varepsilon \mid 0A$

$B \rightarrow \varepsilon \mid 1B$

More fun CFLs/CFGs

- Binary palindromes:

$S \to 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$

- Binary strings with same number of 0s and 1s:

$S \to 0S1 \mid 1S0 \mid SS \mid \varepsilon$ (This is HW 0.3)

or... $S \to 0S1S \mid 1S0S \mid \varepsilon$

- Balanced strings of parentheses:

$S \to (S) \mid SS \mid \varepsilon$ or $S \to (S)S \mid \varepsilon$

Are all languages context-free?

- **No**. Canonical example: L = $\{0^n1^n0^n \mid n \geq 0\}$ is **not** context-free.

(To get a feel for why, try to create a context-free grammar that generates L.)

- Counting argument: The set of possible CFGs over Σ is *countably* infinite, but the set of all languages over Σ is *uncountably* infinite.

- There are also techniques for proving a specific language is not context-free. If curious, search "pumping lemma."

Chomsky Normal Form (CNF)

- Developed by Noam Chomsky in 1959

- A context-free grammar is in CNF if every rule is one of:

$A \rightarrow BC$ (A can be S, but neither B nor C can be S)

$A \rightarrow a$

$S \rightarrow \varepsilon$ (only if ε is in L(G))

Why are CNF grammars nice/useful?

- Full binary parse trees
  - Easy brute-force check to see if a given string can be generated

- Simple structure makes proofs easier

- Assumed by popular parsing algorithms

Every CFG has a CNF equivalent.

- By “equivalent,” we mean “defines the same language.”
- Lecture notes: 5.9 CNF Conversion Algorithm
- No more than quadratic size increase

Recap: Learning Objectives

By the end of this lecture, you will be able to:

- Recall the definition of a context-free grammar/language (CFG/CFL).
- Give examples of CFGs/CFLs.
- Derive strings generated by CFGs using parse trees.
- Determine the CFL generated by a CFG.
- Compare/contrast CFLs with regular languages.
- Identify CFGs in Chomsky normal form.

Recap: Learning Objectives

By the end of this lecture, you will be able to:

- Recall the definition of a context-free grammar/language (CFG/CFL).
- Give examples of CFGs/CFLs.
- Derive strings generated by CFGs using parse trees.
- Determine the CFL generated by a CFG.
- Compare/contrast CFLs with regular languages.
- Identify CFGs in Chomsky normal form.

By the end of tomorrow's lab, you will be able to:

- Construct and describe CFGs that generate given languages.