Unrestricted Grammars

Homework

- Homework #7
  - Due today
- Homework #8 (due Tuesday or Thursday)
  - 9.5
  - 9.38 (only decode first 4 transitions)
  - 11.6
  - Give the topic presented in this class that your enjoyed the most.
  - Give the topic presented in this class that your enjoyed the least.

Plan for today

- Relating CFL to Recursive Languages
  - Unrestricted Grammars
- Computation and Unsolvability

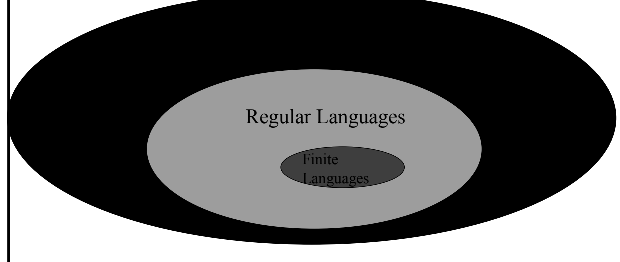
Before We Start

- Any questions?

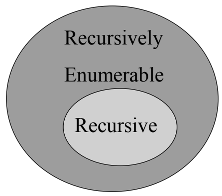
Languages

- You are the weakest link!
  - What is a language?
  - What is a class of languages?

Now we have 2 pictures…this one



And this one…



How do these 2 relate

Unrestricted grammars

- To answer this we'll have to take another look at grammars.

Context Free Grammars

- Let's formalize this a bit:
  - A context free grammar (CFG) is a 4-tuple: (V, Σ, S, P) where
    - V is a set of variables
    - Σ is a set of terminals
    - V and Σ are disjoint (I.e. $V \cap \Sigma = \emptyset$)
    - $S \in V$, is your start symbol

Context Free Grammars

- Let's formalize this a bit:
  - Production rules
    - Of the form $A \rightarrow \beta$ where
      - $A \in V$
      - $\beta \in (V \cup \Sigma)^*$ string with symbols from V and Σ
    - We say that γ can be derived from α in one step:
      - $A \rightarrow \beta$ is a rule
      - $\alpha = \alpha_1 A \alpha_2$
      - $\gamma = \alpha_1 \beta \alpha_2$
      - $\alpha \Rightarrow \gamma$

Context Free Grammars

- Let's formalize this a bit:
  - Production rules
    - We say that the grammar is context-free since this substitution can take place regardless of where A is.
    - We write $\alpha \Rightarrow^* \gamma$ if γ can be derived from α in zero or more steps.

Unrestricted Grammars

- With unrestricted grammars, there is no restriction on the length of the left hand side of a production.
- The only rule is that the left hand side must contain at least 1 varaible
  - Example:
    - $ABC \rightarrow aB$
    - $Ba \rightarrow ACA$
    - $aAa \rightarrow b$

Unrestricted grammars

- Let's formalize this a bit:
  - An unrestricted (or phase-structure) grammar is a 4-tuple: (V, Σ, S, P) where
    - V is a set of variables
    - Σ is a set of terminals
    - V and Σ are disjoint (I.e. $V \cap \Sigma = \emptyset$)
    - $S \in V$, is your start symbol

Unrestricted grammars

- Let's formalize this a bit:
  - Production rules
    - Of the form $\alpha \rightarrow \beta$ where
      - $\alpha, \beta \in (V \cup \Sigma)^*$ string with symbols from V and Σ
      - α contains at least 1 variable.
    - If $\alpha \rightarrow \beta$ is a rule , we say that γ can be derived from α in one step:
      - By replacing a occurrence of α on the right hand side with β

Unrestricted grammar

- Example
  - $L = \{ a^i b^i c^i \mid i \geq 1 \}$ note: this is not a CFL
  - $S \rightarrow A_1BCS_1 \mid A_1BC$ (1)
  - $S_1 \rightarrow ABCS_1 \mid ABC$ (2)
  - $BA \rightarrow AB$ (3) $CA \rightarrow AC$ (4)
  - $CB \rightarrow BC$ (5) $cC \rightarrow cc$ (6)
  - $bC \rightarrow bc$ (7) $bB \rightarrow bb$ (8)
  - $aB \rightarrow ab$ (9) $aA \rightarrow aa$ (10)
  - $A_1 \rightarrow a$ (11)

Unrestricted grammar

- Derive aabbcc
  - $S \rightarrow A_1BC\underline{S_1}$ (1)
  - $\rightarrow A_1BCABC\underline{S}$ (2)
  - $\rightarrow A_1BCABCABC$ (2)
  - $\rightarrow aB\underline{CA}BCABC$ (11)
  - $\rightarrow a\underline{BA}CBCABC$ (4)
  - $\rightarrow aABCB\underline{CA}BC$ (3)
  - $\rightarrow aABC\underline{BA}CBC$ (4)
  - $\rightarrow aAB\underline{CA}BCBC$ (3)
  - $\rightarrow aABACBCBC$ (4)

Unrestricted grammar

- Derive aabbcc
  - $\rightarrow aA\underline{BA}CBCBC$
  - $\rightarrow aAAB\underline{CB}CBC$ (3)
  - $\rightarrow aAABBC\underline{CB}C$ (5)
  - $\rightarrow aAABB\underline{CB}CC$ (5)
  - $\rightarrow \underline{aA}ABBBCCC$ (5)
  - $\rightarrow a\underline{aA}BBBCCC$ (10)
  - $\rightarrow aa\underline{aB}BBCCC$ (10)
  - $\rightarrow aaabBBCCC$ (9)

Unrestricted grammar

- Derive aabbcc
  - $\rightarrow aaa\underline{bB}BCCC$
  - $\rightarrow aaab\underline{bB}CCC$ (8)
  - $\rightarrow aaabb\underline{bC}CC$ (8)
  - $\rightarrow aaabbb\underline{cC}C$ (7)
  - $\rightarrow aaabbbc\underline{cC}$ (6)
  - $\rightarrow aaabbbccc$ (6)

- Questions?

Context Sensitive Grammar

- Context Sensitive Grammars
  - Productions
    - $\alpha \rightarrow \beta$ where α contains at least 1 variable
    - And $|\alpha| \leq |\beta|$
  - A variable can only be replaced in the <u>context</u> of other symbols
  - A language derived from a context sensitive grammar is a context sensitive language.
    - The last example was a context sensitive language

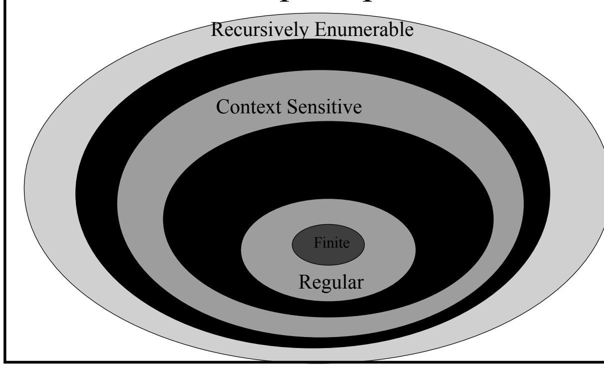
Context Sensitive Grammar

- Do Context Sensitive Languages have a corresponding machine ?
  - Of course, all language classes do.
  - Linear Bounded Automata
    - Like a TM except
      - Has two additional symbols 〈 and 〉
      - The LBA's starting configuration is (q_0, 〈 x 〉)
      - The machine cannot move left of the 〈 or right of the 〉
      - An LBA can only use n cells on the tape where n is the size of the input string.

It can be shown that:

- Every Context Free Language is Context Sensitive
  - By definition of the grammars
- Every Context Sensitive Language is Recursive
  - Minor modification to turn an LBA into a TM that always halts.
- There is a recursive language that is not Context Sensitive
  - One of those strange diagonal type languages.
  - Captain Kirk → Robot → BOOM.

Our complete picture:



It also can be shown:

- Every recursively enumerable language can be generated by an unrestricted grammar.
- In fact, Chomksy (the grammar guy), set out to define the four language classes:
  - Regular, CF, CS, Recursively Enumerable
  - By just using grammars.

Chomsky Hierarchy (1956, 1959)

Type	Languages (grammars)	Form of productions in grammar	Accepting device
3	Regular	$A \rightarrow aB, A \rightarrow a$ $(A, B \in V, a \in \Sigma)$	Finite automaton
2	Context-free	$A \rightarrow \alpha$ $(A \in V, \alpha \in (V \cup \Sigma)^*)$	Pushdown automaton
1	Context-sensitive	$\alpha \rightarrow \beta$ $(\alpha, \beta \in (V \cup \Sigma)^*, \|\beta\| \geq \|\alpha\|,$ α contains a variable)	Linear-bounded automaton
0	Recursively enumerable (unrestricted or phrase-structure)	$\alpha \rightarrow \beta$ $(\alpha, \beta \in (V \cup \Sigma)^*,$ α contains a variable)	Turing machine

Summary

- Unrestricted Grammars
- Context Sensitive Grammars
- Linear Bounded Automata
- Chomsky Hierarchy

- Questions?