Unrestricted Grammars

Reminder

- Final exam
  - The date for the Final has been decided:
  - Saturday, November 16th
  - 12:30pm – 2:30pm
  - 07 – 1420

Homework

- Homework #7
  - Help after lecture

Announcement

- October 31st (today) is halloween
  - Why isn't anyone dressed as Kleene?!?!
  - Refreshments at CS Offices
    - After this course
  - Remember…valuable prizes.

Plan for today

- Relating CFL to Recursive Languages
  - Unrestricted Grammars

Before We Start

- An example of computing a numerical function with a TM

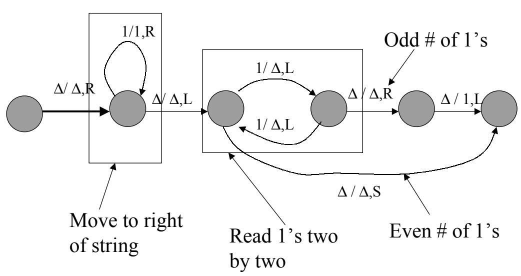
Computation with Turing Machines

- Computing functions with TMs
  - Formally,
    - Let T = (Q, Σ, Γ, q_0, δ) be a TM and let f be a partial function on Σ^*. We say T computes f if for every x ∈ Σ^* where f is defined:
      - $(q_0, \underline{\Delta}x) \mapsto^* (h, \underline{\Delta}f(x))$
    - And for every other x, T fails to halt on input x.

Computation with Turing Machines

- A TM that computes n mod 2
  - n is input on the tape as a sequence of n 1's
  - Solution will be the number of 1's remaining on the tape when the machine halts.
  - Basic idea:
    - Machine makes a pass from left to right to first blank.
    - Move left, "erasing" 1's two at a time
    - If n is odd, a single 1 will be left on the tape
    - Otherwise no 1's will be left on the tape.

Computation with Turing Machines



Computation with Turing Machines

- You know you want to see it…
  - A final JFLAP run

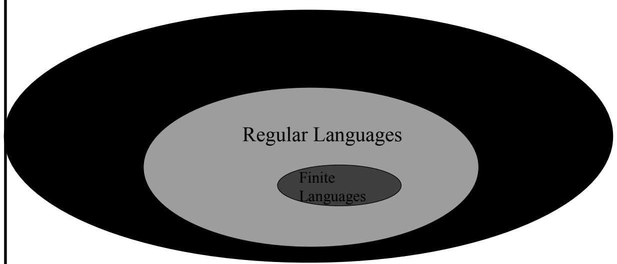
Grammars

- Now back to our regularly scheduled lecture.

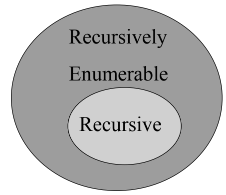
Languages

- 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 $A \to \beta$ 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:
      - $A \to \beta$ 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:
    - $ABC \to aB$
    - $Ba \to ACA$
    - $aAa \to b$

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 $\alpha \to \beta$ where
      - $\alpha , \beta \in (V \cup \Sigma)^*$ string with symbols from V and Σ
      - α contains at least 1 variable.
    - If $\alpha \to \beta$ 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
  - $S \to A_1BCS_1 \mid A_1BC$ (1)
  - $S_1 \to ABCS_1 \mid ABC$ (2)
  - $BA \to AB$ (3) $CA \to AC$ (4)
  - $CB \to BC$ (5) $cC \to cc$ (6)
  - $bC \to bc$ (7) $bB \to bb$ (8)
  - $aB \to ab$ (9) $aA \to aa$ (10)
  - $A_1 \to a$ (11)

Unrestricted grammar

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

Unrestricted grammar

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

Unrestricted grammar

- Derive aabbcc
  - → aaabBBCCC
  - → aaabbBCCC (8)
  - → aaabbbCCC (8)
  - → aaabbbcCC (7)
  - → aaabbbccC (6)
  - → 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 context of other symbols
  - A language derived fron a context sensitive grammar is a context sensitive language.
    - The last example was a context sensitive language

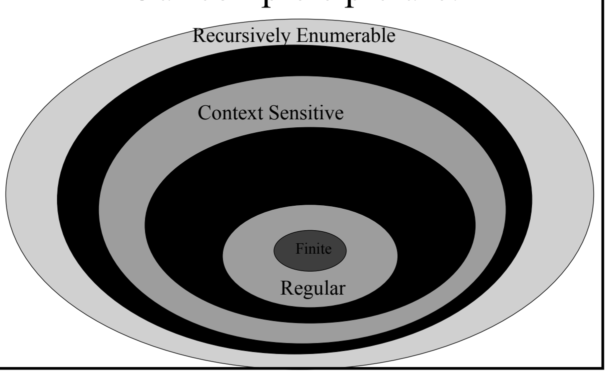
Context Sensitive Grammar

- 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:



Context Sensitive Languages

- Fun facts
  - Context Sensitive Languages are closed under
    - Union, Intersection, and Concatenation…
    - If L is context sensitive, so is L^+
    - Complement…Open question for quite some time
      - Turns out to be yes.
    - Still unknown if every CSL can be accepted by a DETERMINISTIC Linear Bounded Automata.

Context Sensitive Languages

- More fun facts
  - Most "practical" languages are context sensitive.
    - Programming languages
    - Spoken languages

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.

Theory Hall of Fame

- <u>Noam Chomsky</u>
  - <u>The Grammar Guy</u>
  - 1928 –
  - b. Philadelphia, PA
  - PhD – UPenn (1955)
    - Linguistics
  - Prof at MIT (Linguistics) (1955 - present)
  - Probably more famous for his leftist political views.

Chomsky Hierarchy (1956, 1959)

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

Summary

- Unrestricted Grammars
- Context Sensitive Grammars
- Linear Bounded Automata
- Chomsky Hierarchy
- Questions?
  - Next time…Computability