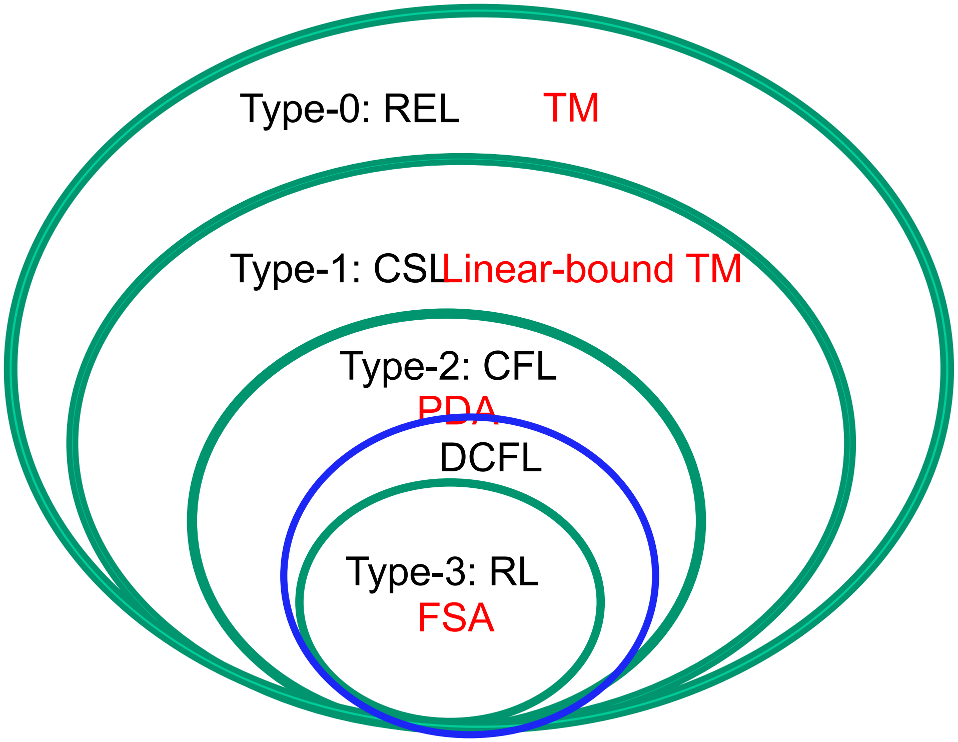
CS 3313
Foundations of Computing:

Properties of Context Free Languages

http://gw-cs3313-2021.github.io

Properties of Language Classes

- A *language class* is a set of languages.
  - Example: the regular languages.
  - Example: context free languages.
- Language classes have two important kinds of properties:
  1. Decision properties.
  2. Closure properties.



Chomsky Hierarchy

Closure Properties

- A *closure property* of a language class says that given languages in the class, an operation (e.g., union) produces another language in the same class.
- Example: the regular languages are closed under union, concatenation, and (Kleene) closure.

Decision Properties

- A *decision property* for a class of languages is an algorithm that takes a formal description of a language (e.g., a DFA or PDA or CFG) and tells whether or not some property holds.
- Example: Is language L empty?
- Example: Is L(M1) = L(M2) ?
- Example: Does L(M) halt on all inputs w ?

Properties of Languages

- Closure Properties: what happens when we “combine” two languages or perform set operations on them ?
  - Ex: Is Intersection of two regular languages still a regular language ?
  - Why is this important ?
    - Construct a larger set from smaller sets
    - Problem decomposition
- Decision Problems: can we provide procedures to determine properties of a language ?
  - Ex: are two machines equivalent? Does automaton accept an infinite set ?

Recall: Closure Properties of Regular Languages

- Theorem: states that if L_1 and L_2 are regular languages, so are the languages that result from the following operations:
  - $L_1 \cup L_2$
  - $L_1 \cap L_2$
  - L_1L_2
  - $\overline{L_1}$
  - L_1^*
  - L_1^R
- We can use these to show other properties
  - Ex: Set difference
  - We also provided constructive proofs, using product DFAs, which constructed a DFA to accept the language (such as $L_1 \cap L_2$)
- Regular languages are closed under homomorphism (substitutions)

Closure under Homomorphisms

- A *homomorphism* (or substitution) on an alphabet is a function that gives a string for each symbol in that alphabet.
- Example: $h(0) = ab$; $h(1) = \epsilon$.
- Extend to strings by $h(a_1 \ldots a_n) = h(a_1) \ldots h(a_n)$.
- Example: $h(01010) = ababab$.
- If L is a language, and h is a homomorphism on its alphabet, then $h(L) = \{h(w) \mid w \text{ is in } L\}$.

Decision Properties

- A *decision problem is decidable* for a class of languages if there is an algorithm that takes a formal description of a language (e.g., a DFA) and tells whether or not the property holds.
  - Also called a decision property; an algorithm always halts!
- Example: Is language L empty?
- Recall – the following properties about Regular languages are decidable:
  - Emptiness (Is L empty)
  - Finiteness (Is L finite)
  - Membership (does w belong to L)
  - Equivalence (is L1= L2)

So what are the properties of CFLs

- A language is CFL if it is generated by a CFG G or accepted by a PDA M
- A language is Deterministic Context Free (DCFL) if it is accepted by a Deterministic PDA (DPDA)
  - A DPDA has at most one choice of moves from any instantaneous description – i.e., for any state and any symbol on top of the stack, when reading an input symbol or empty string, the PDA has at most one choice (to next state and string to push onto the stack).

Closure Properties

- 1. Suppose we have two CFG grammars $G_1 = (V_1, T_1, P_1, S_1)$ and $G_2 = (V_2, T_2, P_2, S_2)$
  - Assume V_1 and V_2 disjoint.
- Consider the grammar $G = (V_1 \cup V_2 \cup \{S\}, T_1 \cup T_2, P_1 \cup P_2, S)$ and add the production $S \rightarrow S_1 \mid S_2$
- What does G generate ? If $S \Rightarrow^* w$,
- Then either $S \Rightarrow S_1 \Rightarrow^* w$ or $S \Rightarrow S_2 \Rightarrow^* w$
- Therefore $w \in L(G_1)$ or $w \in L(G_2)$

Closure Properties

- 2. Suppose we have two CFG grammars $G_1 = (V_1, T_1, P_1, S_1)$ and $G_2 = (V_2, T_2, P_2, S_2)$
  - Assume V_1 and V_2 disjoint.
- Consider the grammar $G = (V_1 \cup V_2 \cup \{S\}, T_1 \cup T_2, P_1 \cup P_2, S)$ and add the production $S \rightarrow S_1.S_2$
- What does G generate ? *If* $S =>^* w$,
- Then $S => S_1S_2 \ =>^* w_1S_2 =>^* w_1w_2$

  where $S_1 =>^* w_1 \ \ S_2 =>^* w_2$
- Therefore $w = w_1.w_2$ and $w_1 \epsilon L(G_1)$ and $w_2 \epsilon L(G_2)$

Closure Properties

- 3. Suppose we have a CFG $G_1 = (V_1, T_1, P_1, S_1)$
- Consider the grammar $G = (V_1 \cup \{S\}, T_1, P_1, S)$ and add the production $S \rightarrow S_1 S \mid \lambda$
- What does G generate ? *If* $S =>^* w$,
- Then $S => S_1 S =>^* S_1 S_1 S =>^* S_1 S_1 \ldots S_1 =>^* w_1 w_2 \ldots w_k$

  where $S_1 =>^* w_i$ *for all* $1 \leq i \leq k$
- Therefore $w = w_1 w_2 \ldots w_k$ and each $w_i \in L(G_1)$

Closure Properties

- 4. Suppose we have a CFG grammars $G_1 = (V_1, T_1, P_1, S_1)$
- Let h be a homomorphism on the terminal symbols in T_1
- Create new set of productions P_2 where

  for every production in P_1 replace every occurrence of $a \epsilon T_1$ with $h(a)$
- Let $G = (V_1, h(T_1), P_2, S_1)$
- What does G generate ? *If* $S \Rightarrow^* w$ *in* G_1 *using productions in P1*,
- Then $S \Rightarrow h(w)$ *using productions in* P_2

Closure Properties

- 5. Suppose we have a CFG $G_1 = (V_1, T_1, P_1, S_1)$
- Consider the grammar $G=(V_1 , T_1, P_2, S)$ and where P_2 is the set of productions obtained by reversing the body (RHS) of every production in P_1
  - For every $A \rightarrow a\alpha$ in P_1, we add $A \rightarrow \alpha^R a$ to P_2
- What does G generate ? *If* $S \Rightarrow^* w$ *using productions in* P_1
- Then $S \Rightarrow w^R$ *using productions in* P_2
- Therefore $w \in L(G)$ *iff* $w = x^R$ and $x \in L(G_1)$

Closure Properties of CFLs

- Theorem: CFLs are closed under
  - Union
  - Concatenation
  - Star Closure
  - Homomorphism
  - Reversal

- If L1 and L2 are CFL then so is their
  - Union, concatenation, star closure, homomorphism, reversal

What about Intersection ?

- $L_1 = \{ a^i b^i c^j \}$ – context free
- $L_2 = \{ a^i b^j c^j \}$ – context free
- $L_1 L_2 = \{ a^n b^n c^n \}$ – *not context free!*

- *CFLs* are not closed under intersection.
  - *If L_1 and L_2 are CFLs then their intersection **may not** be context free.*

- From above, we can prove (using DeMorgan's laws) that CFLs are not closed under complementation

Decision Properties of CFLs

- the following properties about context free languages are decidable:
  - Emptiness (Is L empty)
  - Finiteness (Is L finite)
  - Membership (does w belong to L)
- Proofs ????

Non-Decision Properties

- Many questions that can be decided for regular sets cannot be decided for CFL's.

1. Are two CFL's the same? L1= L2 ?
2. Are two CFL's disjoint? L1 ∩ L2 = ∅ ?
3. Is CFL L ambiguous ?

- Need theory of Turing machines and decidability to prove no algorithm exists.

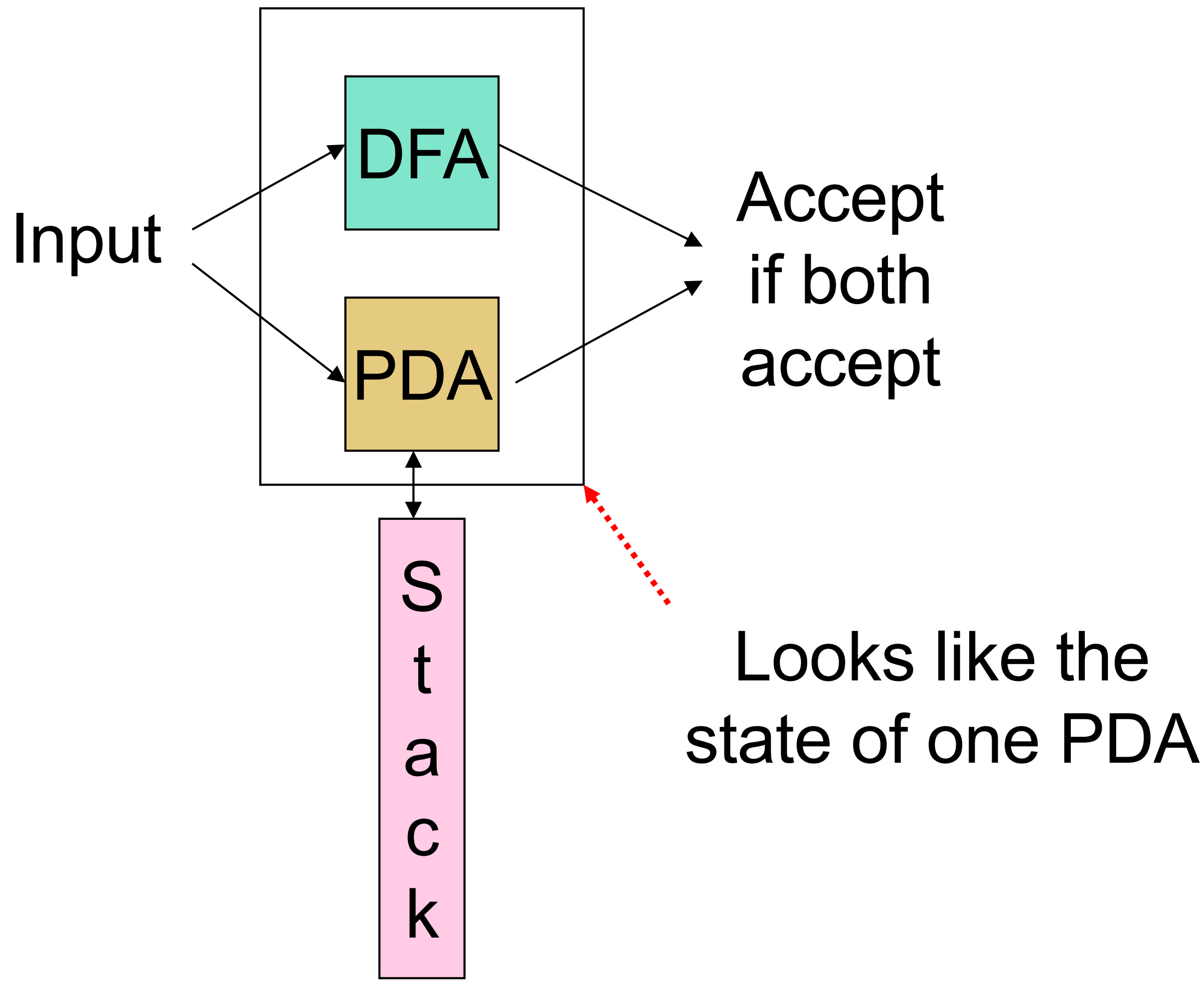
Set operations with Regular Languages

- Any regular language is also a context free language
  - Since we can provide a regular grammar for the regular language
    - Or a PDA that does not use the stack = NFA !
- If L_1 is CFL and L_2 is regular language then:
  - $L_1 \cup L_2$ is a CFL
  - L_1L_2 is a CFL
  - $L_1 \cap L_2$ is a CFL
- First two follow from CFL closure properties…
- So how do we prove intersection is closed ?

Intersection with a Regular Language

- Intersection of two CFL's need not be context free.
- But the intersection of a CFL with a regular language is always a CFL.
- Proof involves running a DFA in parallel with a PDA, and noting that the combination is a PDA.
  - PDA's accept by final state.

DFA and PDA in Parallel



Formal Construction

- Let the DFA A have transition function δ_A.
- Let the PDA P have transition function δ_P.
- States of combined PDA are $[q,p]$, where q is a state of A and p a state of P.
- $\delta([q,p], a, X)$ contains $([\delta_A(q,a),r], \alpha)$ *if* $\delta_P(p, a, X)$ contains (r, α).
  - Note a could be λ, in which case $\delta_A(q,a) = q$.

Formal Construction – (2)

- Final states of combined PDA are those $[q,p]$ such that q is a final state of A and p is an accepting state of P.
- Initial state is the pair $([q_0,p_0]$ consisting of the initial states of each.
- Easy induction: $([q_0,p_0], w, Z_0) \vdash^* ([q,p], \lambda, \alpha)$ if and only if $\delta_A(q_0,w) = q$ and in P: $(p_0, w, Z_0) \vdash^* (p, \lambda, \alpha)$.

Deterministic PDAs and Deterministic Context Free Languages

- Languages accepted by DPDAs are a subclass of CFLs…

  Deterministic Context Free Languags (DCFLs)

  - They have slightly different closure properties

- Why should we be interested in DCFLs (DPDA) ?
  - Practical applications…..parsers are deterministic
  - Syntax of programming languages can be specified using a subclass of DCFLs known as LR(k) (or LL(k)) grammars…
    - Tune in on Thursday for an overview of parsing and a construction of a compiler.

Summary: Closure Properties

- Regular Languages are closed under:
  - Union, Concatenation, Closure, Homomorphism, Reversal
  - Intersection, Complement, Difference
- CFLs are closed under
  - Union, Concatenation, Closure, Homomorphism, Reversal
- CFLs not closed under intersection and complementation
  - If L1 and L2 are CFL then their intersection (or complement) may not be CFL.
- CFLs closed under intersection with regular languages
  - If L1 is CFL and L2 is Regular, then intersection if CFL

- Application of closure properties can make our proofs more elegant and simpler….and prove some properties that may be difficult to do so otherwise.

Exercises/Discussions

- Apply closure properties to Prove or Disprove the questions

1. If $\{ww^R\}$ is a CFL and $\{ww\}$ is **not** a CFL then is $\{xww^Ry \mid x,y \text{ in } \{0,1\}^* \text{ and } w \text{ in } \{a,b\}^*\}$ a CFL ?
   - We did a direct proof during the lab exercise….can you derive the same result without applying pumping lemma
2. If $\{ww^R\}$ is a CFL, then is $L_1 = \{ ww^R \mid w \text{ has even length} \}$ a context free language?
3. Is $L_2 = \{ a^n b^m c^m d^n \mid n \text{ is not a multiple of } 3 \}$ a CFL ?
   - Cursory look seems to imply that we need keep track of two concurrent properties…..
4. If *h(L)* is a CFL, for a homomorphism h, then L must be a CFL ?