# CS340 plan

2010

## Week 1 (AUG 2): Introduction

- Scope and motivation for theory of computation
- Problems and effective procedures. Examples.
- Formalizing computing problems as testing membership in a set of words: the language of positive instances of the problem.
- Basics
  - Alphabets
  - Word
  - Language
- Examples of languages.
- Why studying languages? Two points of view:
  - The linguistic/applicative point of view:
    * For computers: compilers, interpreters
    * Biotechs (the 4 bases of DNA: ACGT, or the 20 amino acids used as building blocks for proteins)
    * Natural Language Processing
  - The computational point of view:
    * set membership as idealization of computing problems
    * distinguish languages by the computational power required to recognize them (complexity classes)
- Concatenation
  - The concatenation operation
  - The empty word
  - Noncommutativity: $xy = yx \implies \exists u : x = u^i, y = u^j$.
  - Free Monod
  - Power
  - Length
- Relation between words: prefix, suffix, factor, subword.

- Distance between words (SKIPPED):
  - Longest prefix length
  - Suffix, factor, subword
  - Editing distance
- Orders on words: lexicographic, military
- Calculability (informal)
  - recursively enumerable language
  - recursive language
- Operations on languages:
  - union
  - intersection
  - complement
  - concatenation
  - power
  - Kleene star
  - Left quotient ${}_{w\backslash}L = w^{-1}.L = \{v \in \Sigma^\star : wv \in L\}$
  - Right quotient $L_{/w} = L.w^{-1} = \{v \in \Sigma^\star : vw \in L\}$
- Regular languages

## Week 2 (AUG 9): Regular expressions & Introduction of Automata

- Regular expressions
- Equivalence between regular expressions and regular languages
- Equivalences between two regular expressions
- Non-regular languages: proof that all languages cannot be represented using regular expressions (using a counting argument: RegExp are countable while languages are not).

- Finite state automata as a model of effective procedure
- DFA
- Evaluation of a word on a DFA
- NFA as a generalization of DFA
- Evaluation of a word on an NFA
- eliminating $\epsilon$-transitions (with backward closure; I have not presented forward closure)
- determinization

## Week 3 (AUG 16): Operations on automata

- trimming (removing states that are neither accessible nor co-accessible)
- translating regular expressions to NFA: Thompson's algorithm.
- translating automata to RegExp: Brzozowski & McCluskey's algorithm.
- Equivalence of representations of regular languages seen so far:
  - Regular expressions
  - NFA
  - DFA
- Some other regular operations: complementation, transposition, intersection (using product, or using union and complementation)

## Week 4 (AUG 23): Minimization

- Two more regular operations that can be done on automata: right and left quotients on languages.
- equivalence between states
- the quotient automaton (for the equivalence relation)
- Myhill-Nerode Theorem
- Minimization of automata. Two algorithms:
  - Moore: partition refinements
  - Brzozowski: $Determize(Transpose(Determinize(Transpose(\mathcal{A}))))$
- An application of automata: pattern matching.

## Week 5 (AUG 30): Decidable problems for Regular languages, Regular Grammars, Pumping Lemma, PDA

- Important (decidable) problems for regular languages:
  - membership $w \in L$
  - emptiness $L = \emptyset$
  - universality $L = \Sigma^\star$
  - inclusion $L_1 \subseteq L_2$
  - equivalence $L_1 = L_2$
- Grammars as generative devices (while automata are accepting devices).
- Formal definition of a grammar
- The Chomsky hierarchy

- Regular Grammars
  Equivalence between NFA and regular grammars
- Representations of regular languages seen so far:
  - Regular expressions
  - NFA
  - DFA
  - Regular grammars (a.k.a. right linear grammars)
  - Left linear grammars
- The pumping lemma as a tool to prove that a language is not regular. Application to $\{a^n b^n \mid n \in \mathbb{N}\}$.
- Intuition why $a^n b^n$ cannot be represented with a FA (it requires infinite memory to count the $a$s and $b$s).
- (Nondeterministic) Pushdown automata: definition and examples

# Week 6 (SEP 6): Context-Free Grammars

- Definition of Context-Free Grammars
- Examples of Context-Free Grammars (including $a^n b^n$).
- Derivations. (Note: the order in which production rules are applied does not matter in CFG.)
- Ambiguities
- Equivalence between CFG and PDA
- Stability of context-free languages
  - Closed under: union, concatenation, Kleene star
  - but NOT under: intersection and complementation
  - If $L_R$ is regular and $L$ is context-free, then $L_R \cap L$ is context free.
- Counting argument that there are some languages that are not context-free.
- Parse tree / Derivation tree
- Pumping lemma for context-free languages. Application to $\{a^n b^n c^n \mid n \in \mathbb{N}\}$.
- Important problems for context-free languages:
  - decidable:
    * emptiness $L = \emptyset$
    * membership $w \in L$
    * does a CFG generate an infinite language?
  - undecidable:
    * universality $L = \Sigma^\star$ (not proved)
    * inclusion $L_1 \subseteq L_2$
    * equivalence $L_1 = L_2$
    * is context-free language $L$ regular?
    * is a CFG ambiguous?

## Week 7 (SEP 13): Parsing Context Free Grammars

This week has midterm exam on Monday, Tuedsay, and Wednesday. If the timetable is not changed, we will have two hours of CS340 on Thursday on Wednesday.

This week will we taken by Jean-Franois Perrot.

- Introduction do parsing CFG
  - A search in the (possibly infinite) derivation graph
  - Top-down parsing
- LALR

## Week 8 (SEP 20): Parsing Context Free Grammars

This week will we taken by Jean-Franois Perrot.

- Usage of Yacc & Lex
- Application to parsing the language developed in CS350.

## Weeks 9–15 (SEP 27): LBA, Turing machines, Complexity Classes...

Six full weeks remains, plus one hour on Oct 20:

- SEP 27
- OCT 4
- OCT 18 (second mid-term exams on Thursay–Saturday, so probably only one hour of lecture on Wednesday the 20th)
- OCT 25
- NOV 1
- NOV 8
- NOV 15