cse303
ELEMENTS OF THE THEORY OF COMPUTATION
Professor Anita Wasilewska
SMALL REVIEW FOR FINAL
SOME Y/N QUESTIONS

Q1 Given $\Sigma = \emptyset$, there is $L \neq \emptyset$ over $\Sigma$
Yes: $\emptyset^* = \{e\}$ and $L = \{e\} \subseteq \Sigma^*$

Q2 There are uncountably many languages over $\Sigma = \{a\}$
Yes: $|\{a\}^*| = \aleph_0$ and $|2^{\{a\}^*}| = C$ and any set of cardinality $C$ is uncountable

Q3 Let $RE$ be a set of regular expressions.
$L \subseteq \Sigma^*$ is regular iff $L = L(r)$, for some $r \in RE$
Yes: this is definition of regular language

Q4 $L^* = \{w \in \Sigma^* : \exists q \in F(s, w) \vdash^*_M (q, e)\}$
No: this is definition of $L(M)$, not of $L^*$
SOME Y/N QUESTIONS

Q5 \( L^* = L^+ - \{e\} \)
No: only when \( e \notin L \)

Q6 \( L^* = \{w_1 \ldots w_n : w_i \in L, i = 1, \ldots, n\} \)
No: only when \( i = 0, 1, \ldots, n \)

Q7 For any languages \( L_1, L_2 \subseteq \Sigma^* \),
if \( L_1 \subseteq L_2 \), then \( (L_1 \cup L_2)^* = L_2^* \)
Yes languages are sets, so \( (L_1 \cup L_2) = L_2^* \) when \( L_1 \subseteq L_2 \)

Q8 \( (\phi^* \cap a) \cup (\phi \cup b^*) \cap \phi^* \) represents a language \( L = \{e\} \)
Yes \( (\{e\} \cap \{a\}) \cup \{b\}^* \cap \{e\} = \{b\}^* \cap \{e\} = \{e\} \)
SOME Y/N QUESTIONS

Q9  \[ L(M) = \{ w \in \Sigma^* : (q, w) \vdash^*_M (s, e) \} \]
    No:  only when \( q \in F \)

Q10  \[ L(M_1) = L(M_2) \text{ iff } M_1 \text{ and } M_2 \text{ are finite automata} \]
     No:  take as \( M_1, M_2 \) any finite automata such that \( L(M_1) \neq \phi \) and \( M_2 \) such that \( L(M_2) = \phi \)

Q11  Any finite language is Context Free
     Yes:  any finite language is regular and we proved that \( RL \subset CFL \)

Q12  Intersection of any two regular languages is CF language
     Yes:  Regular languages are closed under intersection and \( RL \subset CFL \)
SOME Y/N QUESTIONS

Q13  Union of a regular and a CF language is a CF language
Yes:  \( RL \subseteq CFL \) and FCL are closed under union

Q14  If \( L \) is regular, there is a PDA \( M \) such that \( L = L(M) \)
Yes:  FA is a PDA operating on an empty stock

Q15  \( L = \{a^n b^n c^n : n \geq 0\} \) is CF
No:  \( L \) is not CF, as proved by Pumping Lemma for CF languages

Q16  Let \( \Sigma = \{a\} \), then for any \( w \in \Sigma^* \) we have that \( w^R = w \)
Yes:  \( a^R = a \) and hence \( w^R = w \) for \( w \in \{a\}^* \)
SOME Y/N QUESTIONS

Q17  \( A \rightarrow Ax, A \in V, \ x \in \Sigma^* \) is the only rule allowed in a regular grammar
\textbf{No:} not only, \( A \rightarrow xB \) for \( B \neq A \) is also a rule of a regular grammar

Q18  Let \( G = (\{S, (, )\}, \{(, )\}, R, S) \) for \( R = \{S \rightarrow SS \mid (S)\} \)
\( L(G) \) is regular
\textbf{Yes:} \( L(G) = \emptyset \) and hence regular

Q19  The grammar with rules \( S \rightarrow AB, B \rightarrow b \mid bB, A \rightarrow e \mid aAb \) generates a language \( L = \{a^k b^j : k < j\} \)
\textbf{Yes:} the rule \( A \rightarrow e \mid aAb \) produces the same amount of a’s and b’s, and the rule \( B \rightarrow bB \) adds only b’s
Q20  We can always show that $L$ is regular using **Pumping Lemma**
No: we use **Pumping Lemma** to prove (if possible) that $L$ is not regular

Q21  $((p, e, \beta), (q, \gamma)) \in \Delta$ means: read nothing, move from $p$ to $q$
No: must add: and replace $\beta$ by $\gamma$ on the top of the stack

Q22  $L = \{a^n b^m c^n : n, m \in \mathbb{N}\}$ is CF
No: when $n = m$ we get $L = \{a^n b^n c^n : n \in \mathbb{N}\}$ that is not CF

Q23  Every subset of a regular language is a regular language
No: $L = \{a^n b^n : n \geq 0\} \subseteq a^* b^*$ and $L$ is not regular
Q24  Class of context-free languages is closed under intersection

No:  \( L_1 = \{ a^n b^n c^m : n, m \geq 0 \} \) is CF, 
     \( L_1 = \{ a^m b^n c^n : n, m \geq 0 \} \) is CF, but 
     \( L_1 \cap L_2 = \{ a^n b^n c^n, n \geq 0 \} \) is not CF

Q25  A regular language is a CF language

Yes:  Regular grammar is a special case of a context-free grammar

Q26  Any regular language is accepted by some PD automaton

Yes:  Any regular language is accepted by a finite automaton, and a finite automaton is a PD automaton (that never operates on the stock)
SOME Y/N QUESTIONS

Q27  Turing Machines can read and write
Yes:  by definition

Q28  A configuration of a Turing machine $M = (K, \Sigma, \delta, s, H)$ is any element of a set $K \times \Sigma^* \times (\Sigma^*(\Sigma - \{\square\}) \cup \{e\})$, where $\square$ denotes a blank symbol
No:  a configuration is an element of a set $K \times \triangleright \Sigma^* \times (\Sigma^*(\Sigma - \{\square\}) \cup \{e\})$

Q29  A computation of a Turing machine can start at any position of $w \in \Sigma$
Yes:  by definition
SOME Y/N QUESTIONS

Q30  In Turing machines, words $w \in \Sigma^*$ can’t contain blank symbols
No: $\Sigma$ in Turing machine contains the blank symbol $\sqcup$

Q31  It is proved that everything computable (algorithm) is computable by a Turing Machine and vice versa
No: this is Church - Turing Hypothesis, not a theorem

Q32  A Turing machine $M$ decides a language $L \subseteq \Sigma^*$, if for any word $w \in \Sigma^*$ the following is true.
If $w \in L$, then then $M$ accepts $w$;
and if $w \notin L$ then $M$ rejects $w$
No: must say: any word $w \in \Sigma_0^*$, and $L \subseteq \Sigma_0^*$ for $\Sigma_0 = \Sigma - \{\sqcup\}$
SOME PROBLEMS

P1
Let $\Sigma$ be any alphabet, $L_1, L_2$ and $e \in L_2$ Show that

$$(L_1 \Sigma^* L_2)^* = \Sigma^*$$

Solution
By definition, $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Sigma^*$ and $\Sigma^* \subseteq \Sigma^*$. Hence

$$(L_1 \Sigma^* L_2)^* \subseteq \Sigma^*$$

We have to show that also

$$\Sigma^* \subseteq (L_1 \Sigma^* L_2)^*$$

Let $w \in \Sigma^*$. We have that also $w \in (L_1 \Sigma^* L_2)^*$ because $w = ewe$ and $e \in L_1$ and $e \in L_2$
P2
Use book or lecture definition (specify which are you using) to construct a non-deterministic finite automaton $M$, such that

$$L(M) = (ab)^*(ba)^*$$

**Draw a state diagram.** Do not specify all components. **Justify** your construction by listing some strings accepted by the state diagram

**Solution 1:** We use the *lecture definition*

Components of $M$ are:

- $\Sigma = \{a, b\}$,
- $K = \{q_0, q_1\}$,
- $s = q_0$,
- $F = \{q_0, q_1\}$,

$$\Delta = \{(q_0, ab, q_0), (q_0, e, q_1), (q_1, ba, q_1)\}$$

You must **draw the diagram** only!

Strings accepted are: $ab, abab, abba, ababba, ...$

You must **trace the computations** accepting these strings!
P2

Solution 2: We use the book definition

Components of $M$ are:
$
\Sigma = \{a, b\}, \ K = \{q_0, q_1, q_2, q_3\}, \ s = q_0, \ F = \{q_2\},
$

$\Delta = \{(q_0, a, q_1), (q_1, b, q_0), (q_0, e, q_2), (q_2, b, q_3), (q_3, a, q_2)\}$

You must **draw the diagram** only!

Strings accepted are: $ab, abab, abba, ababba, \ldots$

You must **trace the computations** accepting these strings!
SOME PROBLEMS

P3

1. DRAW a DIAGRAM of a PDA $M$, such that

$$L(M) = \{b^n a^{2n} : n \geq 0\}$$

Solution 1

Here are the components- you must draw a diagram!

$M = (K, \Sigma, \Gamma, \Delta, s, F)$ for

$$K = \{s, f\}, \Sigma = \{a, b\}, \Gamma = \{a\}, s, F = \{f\},$$

$$\Delta = \{((s, b, e), (s, aa)), ((s, e, e), (f, e)), ((f, a, a), (f, e))\}$$
P3
2. Explain the construction. Write motivation.

Solution

$M$ operates as follows:

- $\Delta$ pushes $aa$ on the top of the stack while $M$ is reading $b$,
- switches to $f$ (final state) non-deterministically;
- and pops $a$ while reading $a$ (all in final state)

$M$ puts on the stock two $a$’s for each $b$, and then remove all $a$’s from the stock comparing them with $a$’s in the word while in the final state
P3

3. **Trace** a transitions of $M$ that leads to the acceptance of the string $bbaaaa$

The accepting computation is:

$$(s, bbaaaa, e) \vdash_M (s, baaba, aa) \vdash_M (s, aaaa, aaaa)$$

$$\vdash_M (f, aaaa, aaaa) \vdash_M (f, aaa, aaaa) \vdash_M (f, aa, aa) \vdash_M (f, a, a) \vdash_M (f, e, e)$$

**Solution 2**

$M = (K, \Sigma, \Gamma, \Delta, s, F)$ for

$K = \{s, f\}, \Sigma = \{a, b\}, \Gamma = \{b\}, s, F = \{f\},$

$$\Delta = \{((s, b, e), (s, b)), ((s, e, e), (f, e)), ((f, aa, b), (f, e))\}$$
Given a Regular grammar  $G = (V, \Sigma, R, S)$, where

$V = \{a, b, S, A\}$,  $\Sigma = \{a, b\}$,

$R = \{S \rightarrow aS \mid A \mid e, \ A \rightarrow abA \mid a \mid b\}$

1. Use the construction in the proof of L-GTheorem: Language $L$ is regular if and only if there exists a regular grammar $G$ such that $L = L(G)$

to construct a finite automaton $M$, such that $L(G) = L(M)$

Draw a diagram of $M$
SOME PROBLEMS

P4
Solution

Given \( R = \{ S \rightarrow aS \mid A \mid e, \; A \rightarrow abA \mid a \mid b \} \)

we construct a **non-deterministic** finite automata

\[
M = (K, \Sigma, \Delta, s, F)
\]

as follows:

\[
K = (V - \Sigma) \cup \{ f \}, \; \Sigma = \Sigma, s = S, \; F = \{ f \},
\]

\[
\Delta = \{(S, a, S), (S, e, A), (S, e, f), (A, ab, A), (A, a, f), (A, b, f)\}
\]
2. Trace a transitions of $M$ that lead to the acceptance of the string $aaaababa$, and compare with a derivation of the same string in $G$

Solution

The accepting computation is:

$$(S, aaaababa) \vdash_M (S, aaababa) \vdash_M (S, aababa) \vdash_M (S, ababa)$$

$$\vdash_M (A, ababa) \vdash_M (A, aba) \vdash_M (A, a) \vdash_M (f, e)$$

$G$ derivation is:

$$S \Rightarrow aS \Rightarrow aaS \Rightarrow aaaS \Rightarrow aaaA \Rightarrow aaaaabA$$

$$\Rightarrow aaaaababA \Rightarrow aaaaababa$$
P5  
Prove that the Class of context-free languages is NOT closed under intersection  
Proof  
Assume that the context-free languages are are closed under intersection  
Observe that both languages  
\[ L_1 = \{a^n b^n c^m : m, n \geq 0\} \]  
and  
\[ L_2 = \{a^m b^n c^n : m, n \geq 0\} \]  
are context-free  
So the language  
\[ L_1 \cap L_2 \]  
must be context-free, but  
\[ L_1 \cap L_2 = \{a^n b^n c^n : n \geq 0\} \]  
and we have proved that  
\[ L = \{a^n b^n c^n : n \geq 0\} \]  
is not context-free  
Contradiction