CSS321: Theory of Computation

Final Mock Exam (Instructor I)

curated by The Peanuts

Name.............ID.............Section.............Seat No.............

Conditions: Open Book

Directions:

1. This exam has 6 pages (including this page).
2. Please verify you have all pages. If you find an ε-page, you may ignore it.
3. External notes, books, or oracles are allowed. But your brain must operate in offline mode.
4. Maintain calm. Panic transitions are not part of the allowed state diagram.
5. If the exam starts feeling undecidable, breathe deeply and attempt a smaller instance.

*For solution, **click here**.*

Question 1

Let $\mathbb{N}$ be the set of natural numbers and $\mathbb{E} = \{2n \mid n \in \mathbb{N}\}$ be the set of even natural numbers.

(a) Define a bijection $f : \mathbb{N} \to \mathbb{E}$ and prove that it is indeed a bijection by showing it is both one-to-one and onto.

(b) Let $\mathbb{P} = \mathbb{N} \times \mathbb{N}$ be the set of all ordered pairs of natural numbers. Prove that $\mathbb{P}$ is countably infinite by defining an explicit bijection from $\mathbb{N}$ to $\mathbb{P}$.

Question 2

Consider the language $L = \{a^{2+3n} \mid n \in \mathbb{N}\} = \{aa, aaaaa, aaaaaaaa, \ldots\}$ over the alphabet $\Sigma = \{a\}$.

(a) Design a deterministic finite automaton (DFA) that accepts L. Draw the state diagram clearly.

(b) Write a regular expression that describes the language L. Then, use your DFA to show that the strings aa and $aaaaa$ are accepted by tracing the computation paths.

Question 3

Consider the context-free grammar G with the following production rules:

$$S \to aSc \mid B$$
$$B \to bBc \mid bc$$

where S is the start symbol.

(a) Describe the language $L(G)$ in set notation. What patterns do the strings in this language follow?

(b) Give a **leftmost derivation** for the string $aabbcccc$.

(c) Draw the parse tree for the string $aabbcccc$.

Question 4

Consider a simplified context-free grammar G for variable declarations in a programming language:

$$
\begin{aligned}
D &\to T\ V\ ; \\
T &\to \text{int} \mid \text{bool} \mid \text{string} \\
V &\to \text{id} \mid \text{id}, V
\end{aligned}
$$

where D is the start symbol representing a declaration, T represents a type, and V represents a list of variable identifiers.

(a) Draw a parse tree for the declaration `int x, y, z;`

(b) Convert the grammar G into Chomsky Normal Form (CNF). You may introduce new non-terminal symbols as needed. Show all steps of your conversion.

Question 5

Design a one-tape Turing machine M that reverses a string over the alphabet $\{a, b\}$. For example:

- Input: $\triangleright\sqcup abb$ should produce Output: $\triangleright\sqcup bba$
- Input: $\triangleright\sqcup aab$ should produce Output: $\triangleright\sqcup baa$

(a) Describe your algorithm in pseudocode or clear English explanation. Explain the strategy for reversing the string on a single tape.

(b) Draw the state diagram of your Turing machine or describe the transitions formally. You may use helper machines or mnemonics if needed.

(c) Illustrate how your Turing machine works by showing the computation sequence for the input string ab, starting from $\triangleright\sqcup ab$.

CSS321: Theory of Computation

Final Mock Exam (Instructor II)

curated by The Peanuts

Name.............ID.............Section.............Seat No.............

Conditions: Open Book

Directions:

1. This exam has 6 pages (including this page).
2. Please verify you have all pages. If you find an ε-page, you may ignore it.
3. External notes, books, or oracles are allowed. But your brain must operate in offline mode.
4. Maintain calm. Panic transitions are not part of the allowed state diagram.
5. If the exam starts feeling undecidable, breathe deeply and attempt a smaller instance.

Question 1

Consider the language $L = \{a^{n-1}b^n c^{n+1} \mid n \in \mathbb{N}, n \geq 1\}$ over the alphabet $\Sigma = \{a, b, c\}$.

(a) Design a one-tape Turing machine M' that decides L. You may assume the existence of:

- A Turing machine M that decides $\{a^n b^n c^n \mid n \in \mathbb{N}\}$
- A Turing machine $S_{\rightarrow}$ that shifts an input string one step to the right
(i.e., $\triangleright \underline{\sqcup} \omega \vdash^*_{S_{\rightarrow}} \triangleright \underline{\sqcup} \sqcup \omega$)

Describe clearly how you compose these machines to decide L.

(b) Illustrate how your machine M' works on the input string $abbcc$ by showing the key configurations during the computation.

Question 2

Design a one-tape Turing machine M that accepts the language

$$L = \{a^m b^n c^{mn} \mid m, n \geq 1\}$$

(a) Explain your algorithm clearly. How do you verify that the number of c's is exactly $m \times n$? Describe the strategy step-by-step.

(b) Draw the state diagram of your Turing machine or provide a formal description. You may use the mnemonics from the following table or define your own helper machines:

Mnemonic	Description
L	Move cursor to left
R	Move cursor to right
$L_\sqcup$	Scan left until $\sqcup$ is found
$R_\sqcup$	Scan right until $\sqcup$ is found
S_L	Shift entire tape to left
S_R	Shift entire tape to right

Question 3

Consider the context-free grammar G with productions:

$$S \to aSbScS \mid aSSbS \mid baSS$$
$$S \to bcSaS \mid cSaSb \mid cbSaS \mid \varepsilon$$

(a) Does this grammar generate the language

$$L = \{w \in \{a, b, c\}^* \mid n_a(w) = n_b(w) = n_c(w)\}$$

where $n_x(w)$ denotes the number of occurrences of symbol x in string w?

Prove your answer by either:

- Showing by induction that every string generated by G has equal numbers of a's, b's, and c's, or
- Providing a counterexample that shows a string in $L(G)$ that is not in L, or vice versa.

(b) Give **leftmost derivations** for the strings abc and $aabbcc$ using this grammar.

Question 4

Consider the following grammar for list structures:

$$S \to a \mid \wedge(T)$$
$$T \to T, S \mid S$$

where S is the start symbol, a represents an atom, and $\wedge(...)$ represents a list structure.

(a) Find the **leftmost derivation** for the string $\wedge((\wedge(a, a), \wedge(a)), a)$.

(b) Find the **rightmost derivation** for the same string.

(c) Draw the parse tree for the string $\wedge((\wedge(a, a), \wedge(a)), a)$.

Question 5

Let A be the language containing only the single string s, where

$$s = \begin{cases} 0 & \text{if life never will be found on Mars} \\ 1 & \text{if life will be found on Mars someday} \end{cases}$$

For the purposes of this problem, assume that the question of whether life will be found on Mars has an unambiguous YES or NO answer.

(a) Is A decidable? Provide a rigorous argument for your answer. Consider what it means for a language to be decidable and whether we can construct a Turing machine that decides A.

(b) Is A Turing-recognizable? Is the complement $\overline{A}$ Turing-recognizable? Explain your reasoning carefully. What does this tell us about the relationship between decidability and recognizability?