NTIN071 A&G: Information about exams

Exam dates

Exam dates will be available in the Study Information System. The credit from the tutorial is required to register, with the exception of early exam dates (i.e., before the exam period starts, typically the last week of classes).

There will be at least six exam dates: at least 1 in May, 4 in June, and 1 in September. (The September exam date will be determined later in the summer and it is only recommended in the unlikely case of students who will have already failed twice.)

You can register for the exam, as well as cancel your registration, no later than 24 hours before it starts. Please check less than 24 hours before the exam to see if another student's cancellation has changed your allocated exam time. Please note that the times are tentative, some delay often occurs.

Last attempts: In case of third attempts during your second enrollment in the course, university regulations require that the exam be conducted in front of a committee (typically consisting of your instructor and another instructor). All committee members may ask questions, and the grade is based on their consensus. Other than that, the format is the same.

For such cases, a special exam date will be open in the SIS at a time when all committee members are available. That special exam date will have capacity set to zero: should this apply to you, please register for the waitlist and you will be added manually.

Exam rules

- You must bring your student ID and may bring a drinking bottle. Nothing else is allowed at your assigned desk. Pen and paper will be provided. (You may leave your personal belongings inside the classroom but not at your desk.)
- Any electronic devices, phones, smartwatches, wearables, headphones, etc. as well as regular watches are strictly prohibited.
- No talking or whispering to yourself during the preparation time.
- Bathroom breaks are not allowed during the exam.

Exam Requirements

The exam requirements correspond to the material covered in the lecture, with the exception of the following topics which will not be tested:

- Two-way automata
- Automata with output
- Converting a PDA to a CFG
- Deterministic PDA
- Dyck languages
- Ambiguity in grammars

Exam format

The exam is oral, with written preparation phase (15 minutes). During the exam, you can be asked about any of the requirements, but the core of the exam will consist of an assignment that you will draw prior to the preparation phase. The assignment will consists of two questions selected from the list below. The questions are of several types:

- **Definitions:** Give a detailed formal definition of the given concept, some non-trivial example (possibly also an example of an object that does not meet the given definition). Be prepared to answer questions about related properties.
- **Language classification:** The goal is to classify a given language within the Chomsky hierarchy. Typically, this requires constructing a grammar/automaton and/or applying a suitable theorem (such as the Pumping lemma or Myhill-Nerode theorem). You will not be required to show that a language is, or is not, context-sensitive.
- **Properties of languages:** This includes closure properties as well as other properties of individual languges or classes of languages. State and prove what you know.
- **Algorithms:** Formulate the algorithm/construction, you can prepare an example. Prove correctness or other properties from the lecture.
- **Theorems:** Formulate the given mathematical theorem and give its proof. If it is an immediate consequence of another theorem from the lecture, give the proof of that theorem as well. Be prepared to explain all the necessary notions from the formulation, and discuss the corollaries and applications of the theorem.

List of exam questions

Definitions

(D1) ϵ-NFA.

(D2) Extended transition function of a nondeterministic finite automaton.

(D3) Language recognized by a nondeterministic finite automaton.

(D4) Computation graph, computation tree.

(D5) Equivalence of finite automata.

(D6) Equivalence (indistinguishability) of states of a DFA.

(D7) Regular expression and its language.

(D8) String substitution, string homomorphism, inverse homomorphism.

(D9) Language generated by a (general) grammar.

(D10) Language generated by a context-free grammar.

(D11) Parse tree.

(D12) Chomsky hierarchy.

(D13) Chomsky normal form.

(D14) Pushdown automaton.

(D15) Turing machine.

(D16) Nondeterministic Turing Machine.

(D17) Recursively enumerable language.

(D18) Recursive language.

(D19) Universal Turing machine.

(D20) Universal language.

(D21) Diagonal language.

(D22) The complexity class P.

(D23) The complexity class NP, verifiers.

(D24) The complexity class co-NP.

(D25) Space complexity: PSPACE, NPSPACE.

(D26) Reduction between decision problems.

(D27) NP-completeness.

(D28) The boolean satisfiability problem: SAT, 3SAT.

Language classification

(L1-42) Classify the given language within the Chomsky hierarchy.

Properties of languages

(P1) Equivalent representations of regular languages and conversions between them.

(P2) Regular languages and closure under union, intersection, complement.

(P3) Context-free languages and closure under union, intersection, complement.

(P4) Classes of languages and closure under complement.

(P5) Classes of languages and closure under intersection.

(P6) Testing emptiness of regular and context-free languages.

(P7) The Universal language is recursively enumerable but not recursive.

(P8) The Halting problem is undecidable.

Theorems

(T1) Pumping lemma for regular languages.

(T2) Myhill-Nerode theorem.

(T3) Kleene's theorem.

(T4) Pumping lemma for context-free languages.

(T5) Post's theorem.

(T6) Cook-Levin's theorem (an overview of the proof is enough).

Algorithms

(A1) Algorithm to find equivalent (indistinguishable) states of a DFA.

(A2) Algorithm to find unreachable states of a finite automaton.

(A3) Algorithm to find the reduct of a DFA.

(A4) Algorithm to convert a nondeterministic finite automaton to a DFA.

(A5) Algorithm to convert a context-free grammar to Chomsky normal form.

(A6) Algorithm to convert a context-free grammar to a pushdown automaton.

(A7) The CYK algorithm.