COMP30026 Models of Computation
期末复习 #0

张三学长

November 2, 2020

1 Exam Format and Delivery

The exam will be delivered on the Grok platform. You will access it through the COMP30026 Canvas site, from the ”Assignments” tab in the subject menu. It is a timed 3-hour written exam, with 15 minutes reading time. The format will be familiar to you from the practice exam (see ”Exam Preparation”).

The exam is a closed-book exam. The only authorised materials, apart from the Grok interface, are blank paper, pens, pencils and eraser. Make sure you have writing materials ready, as they may be useful for sketching and rough work. You will not be required to scan and upload any files.

The user interface will be the Grok interface that you have worked within throughout semester II. There will be some small additions: You will see a count-down timer, and there will be an ”End Exam” button, in case you want to leave the exam early. You will be able to work on the questions in any order you choose, and you can re-submit. As always, for each answer you submit, you will need to press ”Mark”, followed by ”Submit” — what gets marked is what you have submitted (the latest submission). Should Grok get stuck on a page or fail to render a diagram or an equation properly, you should save the content and then use your browser’s facility for refreshing the page.

Regarding technical problems on the day, please consult the advice at students.unimelb.edu.au/your-course/manage-your-course/exams-assessments-and-results/exams/technical-supportLinks to an external site.

If we need to broadcast a message to all students during the exam, that will happen as an announcement in Canvas. If you have a question for the examiners, you can use a chat facility that will be available through an ”Exam Support” tab in the subject menu. As always in exams, we cannot give you advice on how to approach the exam questions, but we can clarify a question if you find it hard to understand.

2 Examinable Material

All material covered in lectures is examinable, except as indicated below. The references to lecture slides are to the version of slides published on the LMS after each lecture (sometimes they differ from the pre-lecture version). ’Makinson’ refers to ”Sets, Logic, and Maths for Computing”, available as an e-book through the University Library. ’Dowsing’ refers to the ”Reading on Predicate Logic” available under Readings Online on the LMS. Similarly, ’Sipser’ refers to ”Regular languages”, and ’Sudkamp’ refers to ”Context-free grammars”.

- **Basic Haskell programming:**

Function & Recursion
Lists
Types
Trees
List Comprehension

Content corresponding to the first five Grok Haskell modules is examinable. In worksheets and assignments we have utilised Grok's interactive Haskell environment to source your answers to assessment tasks in the form of Haskell expressions, and the exam will use the same approach. You may be asked to read or write simple Haskell code, typically to implement concepts from logic, discrete mathematics, automata theory, language theory, or computability theory. We may also use Haskell to enable features like multiple-answer questions. The Haskell required will be a subset of what was needed for the Grok worksheets and assignment questions. In addition to the syntax- and type checking offered by Haskell, the submission environment will offer further basic well-formedness checks, so that you can avoid certain trivial mistakes in submission. Functions from the Haskell Prelude can be used freely. Other libraries (such as Data.List) can be used only if that is explicitly stated in a question.

- **Propositional logic and propositional resolution:**

All material from the Week 2 lecture, and the first part of the Week 3 lecture, is examinable. Supporting reading, if needed, includes Makinson, parts of Chapter 8.

- **Predicate logic:**

All material from the second part of the Week 3 lecture, and the first part of the Week 4 lecture, is examinable.

- **Resolution for predicate logic:**

All material from the second part of the Week 4 lecture, and the first part of the Week 5 lecture, is examinable, including the unification algorithm. Dowsing covers the same material in greater depth, but note that they use a different unification algorithm. In Dowsing, the sections on ”Clausal form’ and ’Refutation by Resolution’ are relevant, as they expand on what we did in lectures. Material covered by Dowsing but not by us, such as the ’semantic tree’ and the many variants of resolution, is not examinable.

- **Mathematical proof (second part of the Week 5 lecture):**

You are expected to be able to provide properly structured proofs for simple theorems about languages and other discrete structures, including the use of mathematical and structural induction. If you need supporting material on induction principles, see, for example, Makinson Sections 4.1-4.6.

- **Sets, binary relations and functions:**

All material from the Week 6 lecture is examinable. Supporting reading, if needed, can be found in Makinson Chapters 1-3.

- **Regular languages:**

All material from the Week 7 lecture, and the first part of the Week 8 lecture, is examinable. This includes the pumping lemma for regular languages, although you will not be asked to repeat its proof, only (possibly) to apply it. The material on regular languages is covered in greater depth in Sipser.

- **Context-free languages:**

The material from the second part of the Week 8 lecture, and the Week 9 lecture, is examinable. This includes the pumping lemma for context-free languages, although you will not be asked for its proof, only (possibly) to apply it. The material on context-free languages is covered in greater depth in Sudkamp.

- **Well-foundedness and termination:**

  All material from the first part of the Week 10 lecture is examinable.

- **Turing machines, computability, and decidability:**

  The material from the second part of the Week 10 lecture, as well as the Week 11 lecture and the first part of the Week 12 lecture, is examinable. You will not be asked to reproduce any of the results, but you are expected to know about the decidability or otherwise of the problems discussed in those lectures, and (possibly) to use that knowledge to produce simple decidability/undecidability proofs by reduction.

- The additional material on busy beavers is not examinable.

3 Exam Preparation

A practice exam will be held on Tuesday 3 November, starting at 15:00, Melbourne time. We will try to run it under conditions as close as we can get to the actual exam. That is, you will have an opportunity to test the link to Grok, and you can see the (minor) changes to the Grok interface that is used in exam mode.

To best prepare for the exam, make sure you attend the practice exam. Revisit the tutorial exercises. If there were drill questions that you left behind, do them before the exam. For more exercises, many of the reading resources available from the LMS have relevant exercises, as have some old exam papers from a number of computing subjects, as we now explain.

Some of the logic content of COMP30026 was previously taught in the subject 433-295 Discrete Structures, and in 433-255 Logic and Computation. Some of the regular/context-free language content, as well as the computability content, was taught in 433-330 Theory of Computation and 433-325 Theoretical Computer Science. A large number of old exam papers from those four subjects are available online from the University Library. To find them, go to https://library.unimelb.edu.au/examination_papersLinks to an external site. and use the search feature.

The Piazza Discussion Forum will stay open and be monitored until the exam.

Make sure you know when the exam is held, and how to access it through Canvas. For more information, see students.unimelb.edu.au/your-course/manage-your-course/exams-assessments-and-results/examsLinks to an external site. You may also want to check out videos about online exams, via studentit.unimelb.edu.au/exam-technology-webinarsLinks to an external site.

4 Plan for this Course

# 1	Basic Haskell Programming Mathematical Proof Sets, Binary Relations & Functions	with a summary of final exam
# 2	Propositional Logic & Resolution Predicate Logic & Resolution	
# 3	Regular Language	
# 4	Context-Free Language	
# 5	Well-Foundedness & Termination Turing Machines, Computability & Decidability	with a short course on reduction

Nov. 14

① 讲义

② 补充材料

③ Tutorial Work sheet WK1~12