

A Glimpse Beyond



Some highlights of this module

- Regular Expressions
 - Used when defining **programming languages** and properties of systems.
 - Can be used in **searching for patterns in text**
- Context-free Grammars
 - **Grammars** describe the meaning of sentences in English, and the meaning of programs in Java.
 - We showed how **programming languages** are defined using grammars and regular expressions.

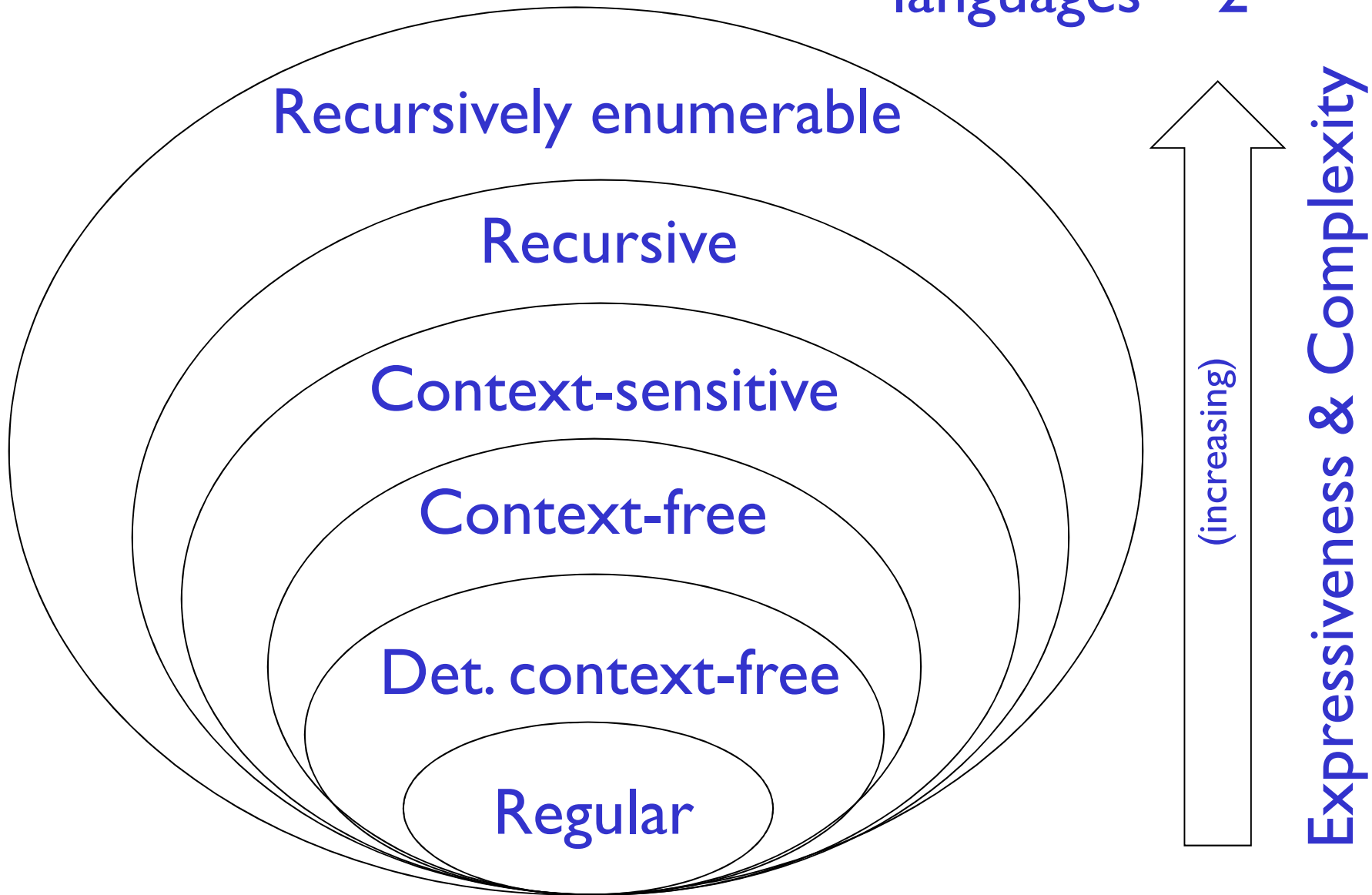
Some highlights of this module

- Turing Machines
 - This is a **general model of a computation**, more powerful than any computer that already exists or will ever be created.
 - By studying Turing machines, we can show that there are many things that **computers will never be able to do**, e.g.

Checking whether a given computer program terminates.

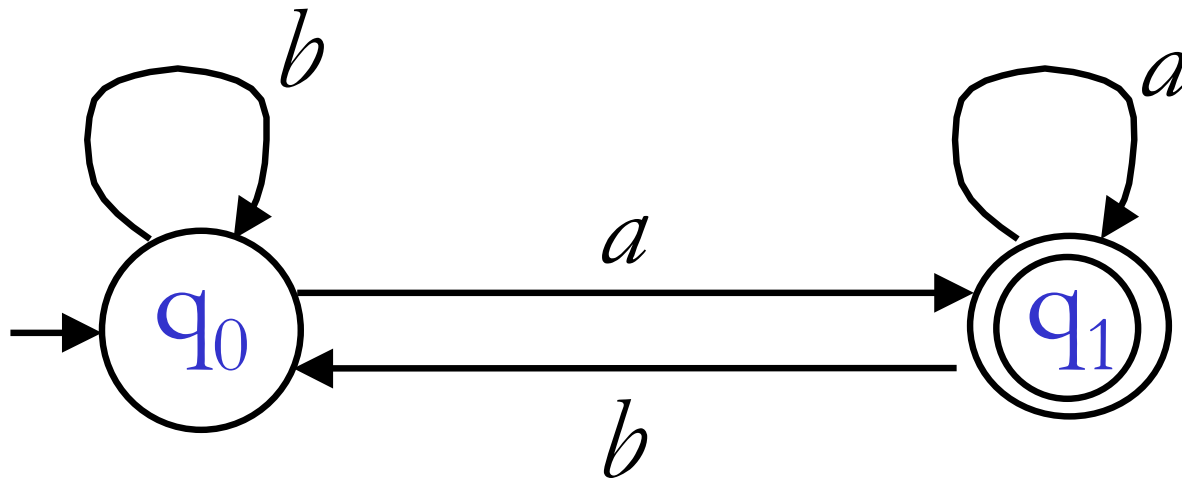
The Chomsky Hierarchy

All possible
languages = 2^{Σ^*}

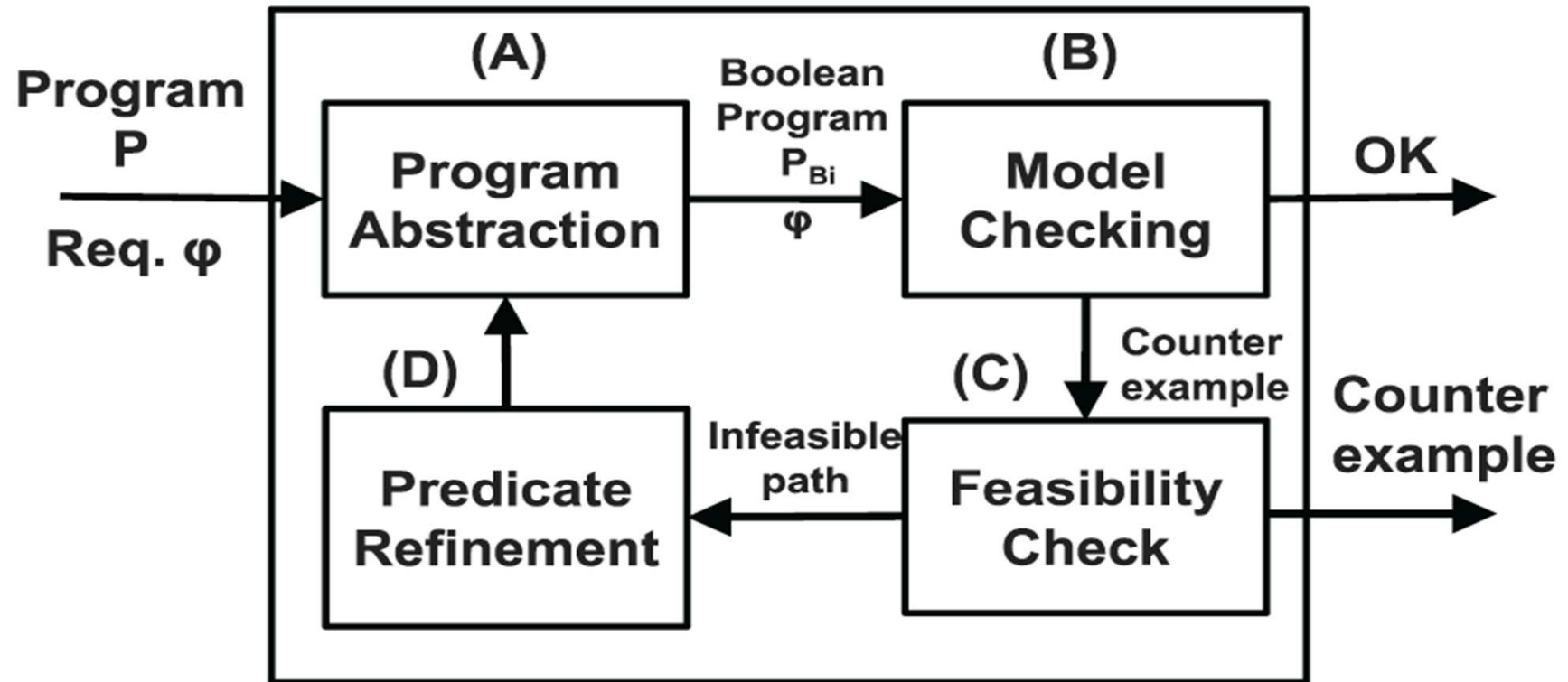


ω -automata / reactive models

$$\Sigma = \{a, b\}$$



Verification: CEGAR



Hybrid System

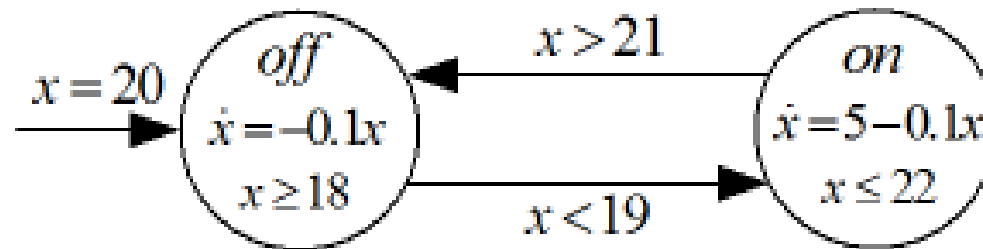
- Systems containing **both discrete and continuous** components
- Practical Examples:
 - Embedded System Controller
 - VLSI circuits
 - System Biology
- Safety Critical Area
- Formal Verification
 - Formal Model : Hybrid Automata



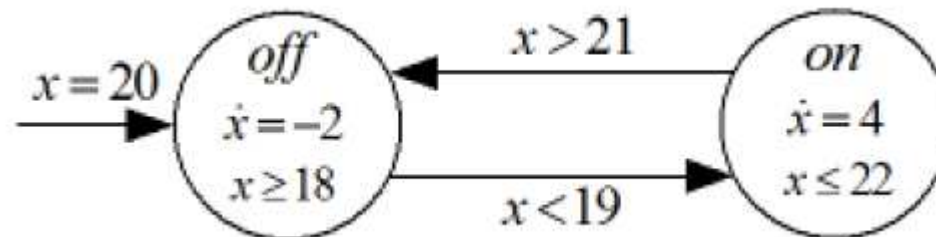
Hybrid Automata

- They consist of
 - A finite state transition system
 - Differential equations in each location

- Example



- Linear Hybrid Automata



Quo vadis?

- Model checking
 - COMP313: Formal Methods
 - COMP304: Knowledge Representation and Reasoning
- Reactive models:
 - COMP310: Multi-Agent Systems
 - COMP329: Robotics and Autonomous Systems
- Reductions:
 - COMP202: Complexity of Algorithms
 - COMP309: Efficient Sequential Algorithms

Final Exam Format

- 120 minutes, 40 multiple-choice questions (all different), 5 possible answers, only one is correct, 20% by guessing, 40% required to pass
- 3 minutes per question, almost no bookwork questions, i.e., you cannot just simply recall the correct answer
- past MCQ exams/class tests are not available, but questions similar to previous written exams and class tests
- check your calculations; it is easy to make a mistake

Final Exam Content

- **50%:** DFA, NFA, ϵ -NFA, GFA, RegEx, ~~RegEx~~ \rightarrow ϵ -NFA, ϵ -NFA \rightarrow NFA \rightarrow \rightarrow DFA, ϵ -NFA \rightarrow GFA \rightarrow RegEx, DFA minimisation (remember the all-reject state), non-regularity of a given language via pumping lemma or via closure properties + example regular languages from the lectures, to show regularity you just need a DFA/NFA/RegEx
- **30%:** CFG, PDA, ~~PDA~~ \rightarrow ~~CFG~~, CFG \rightarrow PDA, CNF, CYK, PCFG, the probability of parsing for a given PCFG, non-context-freeness via pumping lemma or via closure prop.+ examples
- **20%: single tape** Turing machines (simulation on configurations), which problems are recursive/recursively enumerable and Chomsky hierarchy (including context-sensitive + LR(k)), ~~LBA~~

Final Exam Preparation

- Questions concern almost all material, so read all the lecture slides, also look at all tutorial problems (solutions available on VITAL)
- Almost all lectures were recorded and are available on VITAL
- In Learning Resources on VITAL you can find:
 - Link to Automata Tutor
 - Link to Grammar Tutor
 - Two books with problems and their solutions
 - Online lectures by Jeff Ullman
 - Past written exams and class tests with solutions

Wolf, Goat, and Cabbage Puzzle



Hall of Fame

- Ben Clarke
- Will Dunnion
- Xiaofeng Jiang
- Shihao Liang
- Yiyi Pu
- Alice-Maria Toader
- Rufus Walkden
- Diou Yin
- Jingji Yu
- Yuxuan Zhang

End of module questionnaire

- Search for an email that contains words EvaSys and COMP218
- Without a good turnout we have no idea what needs to be improved and how
- If you did not like something say it, but if you liked something say it as well
- Feedback: questions in-class, discussion board, tutorials, automatatutor.com
- Already a lot of examples: if you would like to see more, which other ones should be taken out?

Merry Xmas!

