



Universiteit Utrecht

[Faculty of Science
Information and Computing Sciences]

Talen en Compilers

2022 - 2023, period 2

David van Balen

Department of Information and Computing Sciences
Utrecht University

2023-01-26

16. Summary



Repeat everything

The final exam covers the full contents of the course!



Repeat everything

The final exam covers the full contents of the course!

There may still be some questions on (or requiring knowledge of):

- ▶ The basic notions of languages, grammars, derivations and parse trees.
- ▶ Grammar transformations.
- ▶ Parser combinators.
- ▶ Semantic functions.
- ▶ Compositional functions (algebras, folds, ...).



Regular expressions & languages – tasks

- ▶ Given an informal description of a language, construct a regular expression for that language.
- ▶ Given a (non-)deterministic finite automaton or a regular grammar, construct a regular expression describing the same language.
- ▶ Given a regular expression, describe its language.



Pumping lemmas

The pumping lemmas establish properties that all regular (and all context-free) languages enjoy.



Pumping lemmas

The pumping lemmas establish properties that all regular (and all context-free) languages enjoy.

They can be used to prove that certain languages are not regular (or context-free).



Pumping lemmas

The pumping lemmas establish properties that all regular (and all context-free) languages enjoy.

They can be used to prove that certain languages are not regular (or context-free).

You should be able to prove that a given language is not regular or context-free, but you don't have to **memorize** the CF pumping lemma.



Pumping lemma – tasks

- ▶ Judge whether a given language is regular or not. If it is regular, prove it by providing a regular grammar (or automaton, or regular expression). If not, prove it using the pumping lemma.
- ▶ Recognize when a given language is context-free or not. If so, give a CF grammar (or PDA), and if not, prove it using the pumping lemma.

Some task along these lines will definitely be part of the exam. (Try the exercises in the lecture notes if you have not done so yet.)



Pumping lemma – tasks

- ▶ Judge whether a given language is regular or not. If it is regular, prove it by providing a regular grammar (or automaton, or regular expression). If not, prove it using the pumping lemma.
- ▶ Recognize when a given language is context-free or not. If so, give a CF grammar (or PDA), and if not, prove it using the pumping lemma.

Some task along these lines will definitely be part of the exam. (Try the exercises in the lecture notes if you have not done so yet.)

Remember that **finite** languages are always regular. Why?



LL and LR parsing

Parsing using stack machines. Linear time and space complexity.

- ▶ LL parsing starts with the start symbol on the stack. Actions are **expand** and **match**. We try to reach the empty stack on empty remaining input.
- ▶ LR parsing starts with the empty stack. Actions are **shift** and **reduce**. We try to reach the start symbol on the stack with empty remaining input.
- ▶ Both methods are nondeterministic unless further information about the grammars is being used.



LL parsing – tasks

- ▶ Given a word and a grammar, provide a step-by-step LL derivation of the word.
- ▶ Point out places where an LL derivation is nondeterministic.
- ▶ Compute the empty property, first and follow sets for the nonterminals of a given grammar, using an iterative process.
- ▶ Compute the lookahead sets for a given grammar.
- ▶ Decide whether a given grammar is LL(1) or not.



LR parsing – tasks

- ▶ Given a word and a grammar, provide a step-by-step LR derivation of the word.
- ▶ Point out places where an LR derivation is nondeterministic.
- ▶ Compute the LR(0) automaton of a given grammar.
- ▶ Detect shift-reduce and a reduce-reduce conflicts in an LR(0) automaton.
- ▶ Compute an action table based on the LR(0) automaton.
- ▶ Give a deterministic step-by-step LR derivation of a given word based on an action table.
- ▶ Try to resolve conflicts using the SLR(1) algorithm, taking follow sets into account.



What more?

If you like the topic:

- ▶ In the COSC master you can take courses on: domain specific languages, advanced functional programming, types and semantics, program verification and theorem proving.
- ▶ Many of the MSc students supervised by us perform tasks related to compilers, programming languages, program correctness, ...



Final remarks

- ▶ The exam:
 - ▶ Thursday, February 2nd, 2023
 - ▶ **Begins at 13:30**
 - ▶ Olympos
 - ▶ Lasts 3 hours
 - ▶ No materials (book, notes, etc.) allowed
 - ▶ Mainly applying, not remembering
- ▶ Good luck!
- ▶ Thanks for participating in the course.
- ▶ **Please fill out the online course evaluation!**

