Formal Languages and Compilers

Prof. Angelo Morzenti

Academic year 2023-2024

Introduction /1

- Formal language:
  - a language with precisely defined syntax and semantics
  - it is possibile to define procedures to
    - analyze the syntactic structure and check the grammatical correctness of the phrases
    - compute its meaning (for FLC semantics ≡ translation into a different language)
- The theory of formal languages:
  - studies the form, or ***syntax*** of the phrases
  - defines analysis methods and algorithms
- Compilation
  - applying theory of formal languages
    - to the compilation (and, at large, to the automatic processing)
      - of *programming* languages
      - in general, to all languages of informatics and to «technical languages»

Introduction /2

- Historical notes
  - 1950's – Noam Chomsky: mathematical model of grammars in connection with the study of natural languages
  - 1960's – (a lot of research in progr. lang.) ... ALGOL and CFG's
    - syntax-directed compilation, meta-compilers
    - deep link between ***formal language theory*** and ***automata theory***
    - grammars: regular, context free, context-sensitive
  - 1970's-1980's: Formal languages, automata and Compilation become fundamental disciplines of Computer science
  - Afterwards: these theories continuously foster innovative research and applications

Program/1

- Syntax
  - Formal language Theory
  - Regular expressions and languages
  - Context-free grammars (briefly: free grammars)
  - Grammars for regular languages
- Finite automata and regular language recognition
  - Deterministic finite automata
  - Nondeterministic finite automata
  - Relations between regular languages and finite automata

Program/2

- Phrase recognition and parsing
  - Recognizing free languages through pushdown automata
  - Deterministic syntax analysis (bottom-up / top-down)
- Semantic translation and static analysis
  - translation relation and function
  - regular translations
  - purely syntactic translations
  - semantic translations (attribute grammars)
  - static analysis of programs (based on finite automata and Data Flow equations)

Last year (2022-2023) some topics were eliminated wrt previous years

- Local languages
- Transform. Unilin.Gram $\Rightarrow$ E.R. by means of Language equations
- *LL* parser implementation with explicit stack
- Earley parsing
- Nivat's theorem
- Constant propagation, in the part on Static analysis

- You will find these topics in old exam exercises
  - **Please ignore them!**

Laboratory/design lessons

- Tools for compiler design and a «toy compiler»
  - **Lexical analysis (scanner)**
    - token identification, vocabulary search, attribute extraction
    - **flex** is a scanner generator
    - GNU General Public License (GPL)
  - **Syntax Analysis (parser)**
    - identify the phrase structure according to a given grammar
    - **bison** is a parser generator
    - GNU GPL
- where: in classroom
- When: mostly in the second part of the course
- Who: Dr. Michele Scuttari

Teaching Material

- Textbook
  - Stefano Crespi Reghizzi, Luca Breveglieri, Angelo Morzenti, Formal Languages and Compilation, Springer Verlag, Texts in Computer Science, 3rd Ed. 2019 or 2nd Ed. 2013
  - Italian Version (equivalent to the 2nd English one): Stefano Crespi Reghizzi, Luca Breveglieri, Angelo Morzenti, Linguaggi Formali e Compilazione, Esculapio Ed., 2015
- on the WeBeep platform
  - Lesson slides and recordings
  - Documentation for the laboratory part
  - Exam exercises with solutions

Teaching

- Lesson classes
  - «Theory» (~ 28 hrs) Prof. Morzenti
    - Includes many examples useful for written exam
  - Laboratory (~10 hrs) Dr. Scuttari
    - On the tools for compiler design
- Exercise classes Dr. Scuttari
  - «Theory» (~ 10 hrs)
  - Laboratory-Design (~ 2 hrs)

Exams

- Written Exams
  - No midterm test
  - Exams calls at course end according to School rules
  - Exam consists of two parts that can be taken separately:
    - Theory and Laboratory (modify a fragment of a given compiler)
    - Final mark: weighted average of the Theory (80%) and Practice (20%) parts (both must be ≥15)
    - Usually Laboratory in the initial part of the call (~ 60 min.), then Theory (~ 2hrs.)
  - Validity of a passed test: one year (5 calls, including the one where the mark was obtained, i.e., for the next 4 calls, also across different academic years)
  - If you redo a part you lose the corresponding previous mark, if you refuse an overall mark (2 partial sufficient marks) you must redo at least one part (with the usual constraint that any partial mark is valid for 4 calls after it is obtained); no sufficient mark can be «frozen» (if not accepted it is lost).
  - Some questions are «Optional», only needed to get the full mark; answering correctly all non-optional questions allows students to get a high grade
  - Do not assume that the marks of an exercise are linearly distributed among questions: if an error in one of many questions is really bad, you may get «insufficient» for the entire exercise
- Oral exam
  - oral exam only upon request by the teacher, tipically to clarify the written exam

References

- Prof. Angelo Morzenti
  - angelo.morzenti@polimi.it
  - Tel 02/23993634
  - Office: 3rd floor Ed. 22, Via Golgi 42
  - Meetings with students (in presence or in my WebEx room) upon appointment
- Dr. Michele Scuttari michele.scuttari@polimi.it