

Formal Languages & Compilers

Prof. Luca Breveglieri
a.y. 2007-08

What is a Formal Language ?

- Artificial languages / natural languages
 - communication with machines / with human beings
 - non-verbal / verbal
 - formal / non-formal (informal)
- Formal language:
 - a language is formal if its syntax (structure) and semantic (interpretation) are defined in a precise algorithmic way
 - it is possible to design a procedure that verifies the grammatical correctness of the phrases and determines their meaning
 - in a more restricted meaning, a formal language is:
 - a mathematical object built over an alphabet
 - by means of some axiomatic rules called *grammars*
 - or, alternatively, by means of certain mathematical tools called *automata*
- Theory of formal languages:
 - deals with the structure or syntax of the phrases
 - is the basis for designing *compilers*
 - uses grammars and automata to work with languages

Brief History

- Years '50 – Noam Chomsky proposes the mathematical model of a grammar (1956), for the purpose of studying natural languages
- Years '60 – (intense research) definition of the programming language ALGOL (first) and many others (later) – development of the theory and practice of compilation – discovery of the connection between the theory of formal languages and the theory of automata – invention of formal grammars: regular, free (context-free) and non-free (context-sensitive) – invention or extension of automata: finite, pushdown and others
- Years '70-'80 – formal language theory becomes a standard university discipline
- Later – both grammars and automata are used in basic and innovative applications

Lecture - Contents

- **Basics on languages and their operations**
- **Generative grammar:**
 - regular expression
 - context-free grammar
- **Recognition automaton:**
 - finite automaton
 - pushdown automaton
 - syntax analyser
- **Translation and semantic:**
 - syntactic translation
 - attribute grammar
 - static flow analysis

Laboratory - Contents

- **Software tools for designing compilers:**
 - lexical analysis – Flex
 - syntax analysis – Bison
 - design of a sample compiler
(for a pseudo-C language)
- Where: classroom
- How long: 10 hours (5 sessions)
- When: from december to january
- Teacher: dr. A. Barenghi
- Software tools (Flex, Bison and compiler-assembler) are optionally available for individual self-training.

Textbooks and Didactic Support

- Textbook
 - Stefano Crespi Reghizzi, *Linguaggi Formali e Compilazione*, Pitagora Editrice, Bologna, 2^{nda} edizione, 2007
 - Other texts suggested on the website corsi.metid.polimi.it
- Lecture slides (in english):
 - on the website corsi.metid.polimi.it
- Laboratory slides and other material (in english):
 - on the website corsi.metid.polimi.it - to come in due time
- Exam texts (in english) and solutions (in italian):
 - on the website corsi.metid.polimi.it
 - an english exercise book is in preparation – to come in due time
- Please request registration to the website !
 - click on the registration button and wait for authorization to be released in one or two days

Course & Exam Structure

- Lecture (40 hours)
 - theory of FLC including a few examples and exercises
- Laboratory (10 hours)
 - design of compilers by means of Flex and Bison
- Exam
 - written classwork
 - no middle calls
 - exam at the end of the course (5 calls per year)
 - exam is open-book (textbook and notes admitted)
 - two written independent parts (theory and laboratory)
 - parts can be done separately within one year
- No oral exam

Contacts

- Lecture:
 - Prof. Luca Breveglieri –
 - Email: luca.breveglieri@polimi.it
 - Tel: 02 2399 3653
 - Office: 1st floor building 18, Via Ponzio 34/A (DEI)
 - Contacts: office hours or email or phone call
- Practice: dr. Alessandro Barenghi (to follow)