Diskrete Mathematik

Übung 2

Daniele G. Spampinato

Information

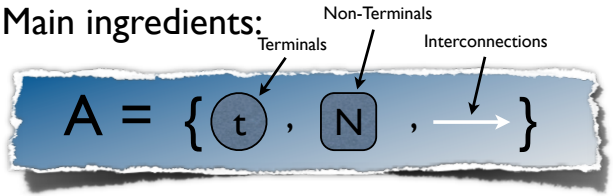
- Assignments normally due one week after handout
- 10/14 required for the Testat
- Groups allowed (no more than 2 persons)
- For any question: danieles@inf.ethz.ch

Recursion

- An entity is **recursive** if it uses itself to do part of its work
- A recursive process requires
  - a set of **base** or **terminal cases**
  - a set of **recursive** or **non-terminal steps**
- Recursive steps must allow reduction to a base case

Syntax Diagrams

- Can be used to define a formal language
- Main ingredients:



- A word belongs to the language if it's generated by a path in the main diagram

Mathematical Induction

http://www.youtube.com/watch?v=qybUFnY7Y8w

Mathematical Induction

- Applies to a variety of theorems expressible in terms of a **positive integer parameter**
- Relies on the **induction hypothesis**
- Similarities between recursion and induction:

Both are anchored on one or more **base cases** and rely on the ability to pass to smaller/bigger instances of the problem

Exercise 2

- 2.1 Induction
- 2.2 Recursion: parenthesis matching
- 2.3 Propositional logic
- 2.4 NAND
- 2.5 Whither goest thou?

Further readings

- http://en.wikipedia.org/wiki/Recursion
- http://en.wikipedia.org/wiki/Mathematical_induction
- C. A. Shaffer, ***A Practical Introduction to Data Structures and Algorithms Analysis***, Prentice Hall
- D. E. Knuth, **The Art of Computer Programming**, Addison-Wesley
- M. Aigner and G. M. Ziegler, **Proofs from THE BOOK**, Springer