Institut for Matematik og Datalogi
Syddansk Universitet

October 9, 2014
JBJ

# DM517 – Fall 2014 – Weekly Note 7

**Special weekly note**
As you will be working on the first obligatory assigment in week 43, this weekly note contains information on what we will do in week 44.

**Important correction from one of the lectures:**
During the first lecture on Turing machines I said something very wrong about PDAs with several stacks: It IS true that a PDA with 2-stacks is more powerfull that a normal PDA, BUT we do not gain any extra computational power by adding even more stacks! In fact, as you will prove at the exercises, **A PDA with 2 stacks can simulate any Turing machine** and hence their computational power equals that of a Turing machine.

**Key points**

- 2-PDAs are equivalent to Turing machines (see the exercises below).
- Many different variants of the TM have been defined, and most of them have the same computational power as a normal TM.
- A nondeterministic TM is not more powerful than a standard TM either. However, nondeterministic TM **may** be exponentially faster (but we don't know whether this is true). This open question is the well known P = NP question.
- A TM is said to **enumerate** a language $L$ if it, when started on an empty tape, prints all strings in $L$ to an attached printer (and no strings that are not in $L$).

**Lecture in week 41, 2014:**
We covered Sections 3.2 and 3.3.

**Lecture October 27, 2014:**

- We will finish Section 3.2 by taling about Turing machines as enumerators.
- We will also cover Section 4.1 on Decidability.
- We may start on Section 4.2

**Exercises October 29, 2014:**

- 3.22 (hint show how to use two stack to simulate a Turing machine. Let the first stack contain what is to the left of the tape head and the second the other part).
- Describe in words a 2-PDA for recognizing the language $\{a^n b^n c^n d^n | n \geq 0\}$.
- Show that every 2-PDA can be simulated by a 3-tape Turing machine.
- January 2003 Problem 4 (note that the notation is slightly different from what Sipser uses, but you should be able to figure it out).
- January 2008 problem 4 (Note that a Turing machine calculates a function $f$ if it, starting in configutation $q_0 w$ terminates in configuration $q_{accept} f(w)$ for every legal input $w$).
- January 2009 Problem 4.
- October 2011 Problem 3