Institut for Matematik og Datalogi
Syddansk Universitet

November 12, 2014
JBJ

DM517 – Fall 2014 – Weekly Note 10

Obligatory assignment 2
This is available from the homepage as well as BB and should be handed in no later than November 24.

No classes in Week 47
In order to give you time to concentrate on the obligatory assignments there are no lectures and no exercises in week 47. There will be two more lectures, namely November 24 and December 8 (there will be no lecture on December 1st).

Stuff covered in week 46

- The rest of Section 4.2.
- Section 5.1 pages 215-220.
- Notes on Undecidability from the coursepage.

Key points

- One way to prove that a language is undecidable is by reduction (via a Turing machine which performs the reduction) from another language which is known to be undecidable. Typical reductions are from the halting language, halting on the empty string, empty language TM (given $< M >$ is $L(M) = \emptyset$?).
- Similarly, if A can be reduced to B and B is Turing-recognizable, then A is Turing-recognizable. As a consequence of this: if A is **not** Turing-recognizable and we can reduce A to B, then also B is **not** Turing-recognizable.
- If a nontrivial language L concerns a single Turing machine and membership in the language is determined only by a Turing machines language, then Rices theorem says that L is undecidable, provided that some but not recognizable languages have the property described for L. More precisely: nontrivial properties of languages of Turing machines are undecidable. Here a property $\mathcal{P}$ is non-trivial if there exist two Turing machines M_1, M_2 so that $L(M_1)$ has the property while $L(M_2)$ does not. Examples of such properties that are undecidable by Rice's theorem are:
  1. The property that the language of M is regular. Here we can take M_1, M_2 such that $L(M_1) = \Sigma^*$ and $L(M_2) = \{a^n b^n | n \geq 0\}$.

2. The property that the language of M is empty. Here we can take M_1 such that $L(M_1) = \emptyset$ and M_2 such that $L(M_2) \neq \emptyset$. E.g. M_2 could be the TM that accepts every string.
3. The property that the language of M contains two strings of different lengths. Here we can use the TMs M_1, M_2 above to show that this is a non-trivial property of languages for TMs.

In each case one has to argue that the TMs M_1 and M_2 exist. Note that it has to be properties of $L(M)$ for a TM M and **not** a property saying what M does to its tape, states etc. Examples of such properties where Rice's theorem can **not** be used are:

1. The property that if TM M is started on the empty string it will eventually halt and have the string w on its tape. One such example is January 2000 Problem 4(b). Note that we also cannot use Rice's theorem directly on January 2000 Problem 4(a) since M may stop and reject the string 'dm17' so the question is NOT about the language of a TM. Note however that we may use Rice's theorem in the following way: Every TM M is equivalent (can be transformed into a TM with that property by an algorithm and hence by a TM) to a TM M' which halts on exactly those strings which it accepts: just let M' simulate M and loop if M wanted to reject the input. Now apply Rice's theorem to languages of Turing machines M' in the sense that now the property above (stopping on 'dm17') IS a property about the language of a Turing machine.
2. The property that if TM M is started on the empty string it will run through all of its states, except one (it cannot use both q_{accept} and q_{reject}) before eventually halting. This is undecidable, as we show in the notes on (un)decidability, but we cannot use Rice's theorem to prove it.
3. The property that if TMs M_1 and M_2 are started on w, then both will accept w (that is $w \in L(M_1) \cap L(M_2)$). Here the problem is that we are talking about the language of two Turing machines, not one, so we cannot use Rice's theorem to show that this problem is undecidable, which it is.