15-354: Midterm

October 13, 2009

Problem 1: Turing Machines (20 pts.)

Recall that a state p in a DFA is accessible if there is some input that takes the DFA to p (starting at the initial state). We can easily test whether state p is accessible using some graph exploration algorithm such as depth-first-search. Now define a state p of a Turing machine to be accessible if it has the analogous property: there is at least one input that takes us to p at some point during the computation.

- Is it decidable whether a state in a Turing machine is accessible?
  Justify your answer either by indicating what algorithm could be used to check accessibility, or by explaining why no such algorithm can exist.

Solution

Accessibility of a state is undecidable for TMs.

To see this, let's make the problem precise as follows: given a TM M and some state p, is there an input x such that during the computation of M on x state p is reached?

Let's use this to solve the Halting Problem: given a TM N and input z determine whether N halts on z. Construct a new TM M that, on input x, runs N for at most $|x|$ steps on z and goes into a special state p only if N halts on z. Then N halts on z if, and only if, state p is accessible in M.

Problem 2: The Confluence Problem (20 pts.)

Let N be the set of all 64-bit integers. Consider a function $f : N \to N$ that is easily computable; say, you have a piece of C code that implements f.

Recall the Confluence Problem: given two points $a, b \in N$ we need to test if there exist $n, m \geq 0$ such that

$$f^n(a) = f^m(b).$$

Clearly, one can solve the Confluence Problem by enumerating the orbits of a and b and checking if they overlap. Unfortunately, that would require some 2^{64} words of memory.

- Explain how to do this using just a few words of memory. Give a description of your algorithm (don't just scribble code, clearly explain the ideas behind your algorithm).

Solution

Run Floyd's algorithm to obtain points a' and b' on the limit cycles of a and b, respectively.

Then do a brute-force search for b' on the cycle of a' (stopping with NO if the search returns to a' without encountering b' and returning YES is b' is found).

Both parts clearly require $O(1)$ memory.

Problem 3: Square Roots of $\Sigma^\star$ (20 pts.)

Consider the alphabet $\Sigma = \{a, b\}$. The language $L = \varepsilon + a\Sigma^\star + \Sigma^\star b$ is a "square root" of $\Sigma^\star$ in the sense that $L \cdot L = \Sigma^\star$.

- Explain why every square root of $\Sigma^\star$ must contain ε.
- Prove that L is a square root of $\Sigma^\star$.

Solution

Otherwise every word in $L \cdot L$ would have length at least 2.

Clearly $L \cdot L$ contains ε, a and b, so we only have to worry about words of length at least 2. Moreover, since $a\Sigma^\star b \subseteq L \cdot L$ we may only consider words $x = bya$. But then x can be written as $x = zbaa \ldots a$ which shows that $x \in \Sigma^\star ba\Sigma^\star \subseteq L \cdot L$.

Problem 4: Odd Regular Expressions (20 pts.)

In class we have seen a regular expression for the language "even number of a's, even number of b's".

$$E = (aa + bb + (ab + ba)(aa + bb)^\star(ab + ba))^\star$$

- Give a regular expression A for odd/odd.
- Give a regular expression B for even/odd.

Don't use the messy conversion algorithm for this, try to express A and B in terms of E.

Solution

$$A = E(ab + ba)E \qquad B = EbE + EaEbEaE$$

There are lots of other possibilities. Note that one can check correctness by inspecting the diagram of the minimal automaton and exploiting symmetries (even/even input takes you from any state back to that state).

Problem 5: Wurzelbrunft and MSO (20 pts.)

Recall from class that we showed how to construct a formula $\Psi(X)$ in $\mathsf{MSO}[<]$ that expresses the assertion that the set of positions X has even cardinality. Inspired by this result, famous scholar Prof. Dr. Wurzelbrunft believes to have discovered a formula $\Phi(X,Y)$ in $\mathsf{MSO}[<]$ that expresses "X and Y have the same cardinality." Alas, the formula is some 50 pages long, so we can't show it here.

- Use $\Psi(X)$ to construct a $\mathsf{MSO}[<]$ formula for the even/even language.
- What well-reasoned professional advice can you give Wurzelbrunft concerning his formula $\Psi(X,Y)$? Justify your consulting fee.

Solution

$$\exists\, X, Y\, (\forall\, z\, (Q_a(z) \leftrightarrow X(z) \wedge Q_b(z) \leftrightarrow Y(z)) \wedge \Psi(X) \wedge \Psi(Y))$$

If Wurzelbrunft's formula existed we could write

$$\exists\, X, Y\, (\forall\, z\, (Q_a(z) \leftrightarrow X(z) \wedge Q_b(z) \leftrightarrow Y(z)) \wedge \Phi(X, Y))$$

But a word satisfies this formula if and only if it has the same number of a's and b's. This property is clearly not regular.