# “CS 374” Spring 2015 — Homework 9

Due Tuesday, April 21st, 2015 at 10am

---

## • • • Some important course policies • • •

---

- **You may work in groups of up to three people.** However, each problem should be submitted by exactly one person, and the beginning of the homework should clearly state the names and NetIDs of each person contributing.
- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use. See the academic integrity policies on the course web site for more details.
- **Submit your pdf solutions in Moodle.** See instructions on the course website and submit a separate pdf for each problem. Ideally, your solutions should be typeset in LaTex. If you hand write your homework make sure that the pdf scan is easy to read. Illegible scans will receive no points.
- **Avoid the Three Deadly Sins!** There are a few dangerous writing (and thinking) habits that will trigger an automatic zero on any homework or exam problem. Yes, we are completely serious.
  - Give complete solutions, not just examples.
  - Declare all your variables.
  - Never use weak induction.
- Unlike previous editions of this and other theory courses we are not using the “I don’t know” policy.

---

**See the course web site for more information.**

If you have any questions about these policies,
please don’t hesitate to ask in class, in office hours, or on Piazza.

---

1. Note: solutions to this problem are easily found on the web. However, if you don't think about and solve these yourself, you will not be prepared for a planned final exam problem. Most of these should be easy.

   (a) In lab you showed that the recursive languages are closed under a vareity of operations, such as union, intersection, complement, and concatenation. Prove that recursive languages are closed under Kleene *. That is, prove that if $L$ is recursive, then so is $L^*$.

   (b) Prove that recursively enumerable languages are closed under as many of these operations as you can: {union, intersection, complement, concatenation, Kleene *}. Again, finding the answer via googling will not help you later. *Hint: dovetailing*.

2. A *Word-Production Machine* ("WPM") is a TM $P$ that takes as input a natural number $n$, and (may) output some word $w$ (or may not halt). If a word $w$ is output on input $n$, then we say that $P(n) = w$. Otherwise, $P(n)$ is undefined. Let $\text{span}(P) = \{w \mid \text{for some } n, P(n) = w\}$. Prove that a language $L$ is accepted by some TM if and only if there is a WPM $P$ such that $\text{span}(P) = L$.

3. Many years ago, a popular storage medium for computers was *punched paper tape* (see http://en.wikipedia.org/wiki/Punched_tape).
   One of the problems with punched tape was that it could only be written once - since once the holes are punched to indicate a character, there was no way to erase the character. Thus, it was a reasonable permanent storage medium, but not very good for working memory. Nonetheless, as you will show in this problem, it is sufficient for any computation that a TM could otherwise do.

   Consider an arbitrary one-tape Turing machine $M$ with input alphabet $\{0, 1\}$ and tape alphabet $\{0, 1, B\}$. Describe how a punched-tape variant $M'$ of $M$ can simulate $M$, where the conventions for $M'$ are as follows.

   - $M'$ has a read-only input tape, with end markers \$, and with tape cells that are either unpunched (0), or punched (1). There cannot be more than one punch in any tape cell. The read head can move back and forth along the input tape, but cannot punch (write) on this tape.
   - $M'$ has a single work tape, initially all unpunched. It may move back and forth along the work tape, and punch a cell if it so chooses. Once punched, a cell may not become unpunched.
   - $M'$ can determine whether a cell has been punched or not.
   - $M'$ otherwise works as a normal TM would. That is, in a single move, based on its current state, whether or not the cells currently scanned on the input tape and work tape are punched, it may punch the work tape (or not), independently move the heads on the input and work tape left, right, or stay in the same cell, and change state.

   Describe how $M'$ can be used to simulate $M$. You need not fully describe the states and transitions of $M'$ in terms of those of $M$, but you should give sufficient detail that the states and transitions could be created from your description. Be sure to describe in detail how a single move of $M$ is simulated by $M'$. If $M$ on input $w$ takes $k$ steps before halting, roughly how many steps does your $M'$ take? (Asymptotic bounds are fine, e.g., $O(k), O(k \log k), O(k^2)$, etc.)