# Solution Set 4

**Due:** In class on Wednesday, March 3. Starred problems are optional.

**Problem 4-1.** Design a linear systolic array to recognize strings of the form $ww$ in real time.

**Solution:** We recognize strings of the form $ww$ in much the same way as we recognize palindromes. As in that problem, we first design a linear semisystolic array which is simple to understand, and then argue that it can be converted into a systolic array.

As in the palindrome recognizer, the semisystolic network has 1-register edges going down the array away from the host, and 0-regsiter edges coming back up to the host. Each processor will, as before, store two values and compare them to get the output. If the values are the same, and all the processors to the right also have values which are the same, then the processor will output a 1 to its left; otherwise it will output a zero. This output goes down the 0-register edges, so the answer will accumulate and return to the host in real time. To advance to the next tick, the "bottom" row values get passed one node to the right, and a new input from the host is read into the bottom row of the leftmost processor.

The difference from the palindrome recognizer lies in what happens to a value when it reaches the end of the bottom row, i.e. when it is passed to a processor which has no value in its top row. This value will be the last element in the first half of the next iteration of the string, so it needs to be compared to the next element received from the host, and thus needs to be passed all the way back to the leftmost processor. We can accomplish this by using the same back-edges which are used to propogate the result back in real time.

Note that there are now two general types of "rounds" of the array's execution. Half the time, the string inputted so far will have an odd length. In these rounds, the output will always be 0 (since $ww$ has an even length), so the leftmost processor can simply report that with no other input. These are also the rounds in which a topless processor will receive an input on its bottom, so the same back-edges can be safely co-opted for this purpose.

In the other rounds, room must be made for the element newly passed to the top of the leftmost processor, so both top and bottom rows shift over one. In these rounds, the back-edges return to their original purpose of carrying 0s or 1s back to indicate whether the string is of the right form; a processor outputs 1 to the left iff its two values match and it receives a 1 form the right.

This design uses the exact same network as the plaindrome-recognizer does. Importantly, if $G$ is its graph, then $2G-1$ has 1 register on each edge, and thus has no 0-weight cycles. So with a slowdown and retiming, the semisystolic design just described can be made systolic, as desired.

**Problem 4-2.** A ***punctuated palindrome*** is a string which, when spaces and punctuation marks are removed, is a palindrome. For example, "`A MAN, A PLAN, A CANAL: PANAMA!`" is a punctuated palindrome. Describe how to construct a linear systolic array that recognizes punctuated palindromes in real time. (*Hint:* At first glance, this problem seems resistant to efficient systolic conversion, because it requires both broadcast and accumulation, but think again.)

**Solution:** Take the standard systolic palindrome recognizer, add 0 weight forward rippling edges to broadcast whether or not the input is a letter, and convert to systolic in the standard way.

**Problem 4-3.** Show that the following two properties are equivalent for any semisystolic circuit $G$ and integer constant $c > 0$:

- The constraint graph $cG - 1$ contains no negative-weight cycles.
- There exists a retiming of $G$ whose clock period is at most $c$.

**Solution:** This proceeds more or less as the proof in the lecture notes, with a few twists. The only tricky part is the forward implication. Suppose $cG - 1$ has no negative cycles. Retime with $\lfloor r(u)/c \rfloor$, where $r(u)$ is the weight of the shortest path from $u$ to the host in $cG-1$, which exists by the supposition. For an edge $(u, v)$, by the triangle inequality, $r(u) \leq cw(u,v) - 1 + r(v) \Rightarrow w(u,v) - r(u)/c + r(v)/c \geq 1/c \Rightarrow w(u,v) - \lfloor r(u)/c \rfloor + \lfloor r(v)/c \rfloor \geq 1/c - 1 > -1$. Since all quantities in the last triple term expression are integers, $w(u,v) - \lfloor r(u)/c \rfloor + \lfloor r(v)/c \rfloor \geq 0$. Thus, this retiming is valid. Now we show that every path of length c+1 has weight at least 1, thus showing that the clock period is at most c.

For such a path $p$ from $u$ to $v$, by the triangle inequality, $r(u) \leq cw(p) - |p| + r(v) \Rightarrow w(p) - r(u)/c + r(v)/c \geq |p|/c \Rightarrow w(p) - \lfloor r(u)/c \rfloor + \lfloor r(v)/c \rfloor \geq |p|/c - 1 \geq (c+1)/c - 1 > 0$. Since all quantities in the last triple term expression are integers, $w(p) - \lfloor r(u)/c \rfloor + \lfloor r(v)/c \rfloor \geq 1$.

The backward implication is virtually identical to that given in the notes.