Correction for “Pebbles and Branching Programs for Tree Evaluation”, by Stephen Cook, Pierre McKenzie, Dustin Wehr, Mark Braverman, and Rahul Santhanam, ACM ToCT (3,2), 2012.

January, 2013

Theorem 5.15 in the above paper is correct, but the proof needs a small correction. The statement of the theorem gives a tight lower bound on the number of states in a nondeterministic thrifty branching program solving the tree evaluation problem for binary trees of height 4. Precisely

Theorem 1 *Every nondeterministic thrifty branching program solving $BT_2^4(k)$ has $\Omega(k^3)$ states.*

The proof considers YES inputs I in a certain set $E^{r,s}$ of inputs to a nondeterministic thrifty branching program B, and associates a thrifty accepting computation $\mathcal{C}(I)$ with each such I. The proof also associates a tuple

$$U(I) = (u, \gamma^I, \delta^I, x_1, x_2, x_3, x_4)$$

with I, where γ^I and δ^I are states in the computation $\mathcal{C}(I)$. We use (γ^I, δ^I) to denote the segment of $\mathcal{C}(I)$ between γ^I and δ^I.

The tag $u \in \{1, 2, 3\}$ in $U(I)$ specifies a partition of the middle nodes $\{v_2, v_3, v_4, v_5, v_6, v_7\}$ of the input tree into disjoint sets S_1 and S_2 with the following properties:

- Every middle node queried during the computation segment (γ^I, δ^I) is in S_1.
- S_1 has at most four nodes, and the values of all nodes in S_1 are specified by x_1, x_2, x_3, x_4 in $U(I)$.
- The parent of every node in S_2 is queried during (γ^I, δ^I).

Near the end of the proof is the following claim:

Claim: If $I, J \in E^{r,s}$ and $U(I) = U(J)$, then $I = J$.

The claim is correct, but the proof of the claim is wrong, since it states that if $U(I) = U(J)$ then input I is consistent with the segment (γ^J, δ^J) of the computation $\mathcal{C}(J)$. (A thrifty query for J might be nonthrifty for I, so the two answers could be different.)

To fix the proof, define a new input I' as follows: For each non-leaf node v_i, let $f_i^{I'}(x, y) = f_i^I(x, y)$ if x, y are the correct values for the children of node v_i in input I, and otherwise let $f_i^{I'}(x, y) = f_i^J(x, y)$. Let the values of the leaf nodes of I' be r or s, as in $E^{r,s}$.

Thus the node values for I and I' are the same, but some of the functions associated with I and I' are different. This input I' may not be in the set $E^{r,s}$, but this does not matter, because we assume that B is a nondeterministic thrifty branching program which runs correctly on all inputs.

The state sequence for the computation $\mathcal{C}(I)$ is also a possible state sequence for B on input I', because $\mathcal{C}(I)$ only makes thrifty queries. We construct a different accepting computation C' for the input I' as follows: C' coincides with $\mathcal{C}(I)$ until γ^I, then follows $\mathcal{C}(J)$ from γ^I to δ^I, and finally follows $\mathcal{C}(I)$ to the accept state. This is possible, because every query made by $\mathcal{C}(J)$ during the segment (γ^J, δ^J) is either thrifty for I as well as for J, (so the answer is specified by (x_1, x_2, x_3, x_4) in $U(I) = U(J)$, and is the same for all three inputs I, I', J), or it is not thrifty for I, so by construction of I' the answer is the same for inputs I' and J.

Suppose $I \neq J$. Given that both I and J are in $E^{r,s}$ and $U(I) = U(J)$, it follows that I and J (and hence I' and J) differ on the value of some middle node v in the set S_2 specified by the tag u. Thus by the stated property of S_2, the computation segment (γ^J, δ^J) queries the parent of v, and this query cannot be thrifty for both J and I'. But all accepting computations of a thrifty branching program must make only thrifty queries.

This proves the claim.

Acknowledgement: Thanks to David Liu and Toni Pitassi for pointing out the error and the correction.