A Problem due to J Moore
Jayadev Misra
3/28/01

1 The Problem

The following problem was posed by J Moore during the faculty lunch today. Let there be two machines –α and β– with two registers each, which can read/write a shared counter. Initially the counter holds the value 1 and all registers are empty. There are two atomic actions:

1. (read) A machine may read the counter value into one of its empty registers (and then the register becomes nonempty).
2. (write) A machine may write the sum of its register values —provided that both are nonempty— into the counter and then empty both registers.

We show that for any positive integer there is an execution in which the counter holds that value.

2 The Solution

The state of the system is a triple $\langle a, b, c\rangle$, where

a is the register values of α
b is the register values of β
c is the value in the counter.

If both registers of α are empty we write $-$ for a; otherwise, a is a single value or a pair of values. Similarly for b. Initial state of the system is $\langle -, -, 1\rangle$.

Theorem For any natural number k and positive integer x where $x \leq 2^k$, $\langle -, x, 2^k\rangle$ is reachable.
Proof: Proof is by induction on k.

Case $k = 0$: We have to show that $\langle -, 1, 1\rangle$ is reachable. In the initial state let β read the counter.

Case $k+1$, $k \geq 0$: We are given $x \leq 2^{k+1}$. We have to show that $\langle -, x, 2^{k+1}\rangle$ is reachable.

Subcase $x \leq 2^k$: Inductively, $\langle -, x, 2^k\rangle$ is reachable. The following sequence of steps interspersed with the states that are reached establishes the claim.

$\langle -, x, 2^k\rangle$
  α reads

$\langle 2^k, x, 2^k \rangle$
  α reads
$\langle (2^k, 2^k), x, 2^k \rangle$
  α writes
$\langle -, x, 2^{k+1} \rangle$

Subcase $x > 2^k$**:** Then, $x = 2^k + y$ for some y, $0 < y \leq 2^k$. Inductively, $\langle -, y, 2^k \rangle$ is reachable. The following sequence of steps interspersed with the states that are reached establishes the claim.

$\langle -, y, 2^k \rangle$
  α reads
$\langle 2^k, y, 2^k \rangle$
  α reads
$\langle (2^k, 2^k), y, 2^k \rangle$
  β reads
$\langle (2^k, 2^k), (y, 2^k), 2^k \rangle$
  β writes
$\langle (2^k, 2^k), -, 2^k + y \rangle$
  β reads
$\langle (2^k, 2^k), 2^k + y, 2^k + y \rangle$
  α writes
$\langle -, 2^k + y, 2^{k+1} \rangle$
$\langle -, x, 2^{k+1} \rangle$

Corollary: For each positive integer there is an execution such that the counter holds that value.

Proof: For any positive integer x, there is a k such that $x \leq 2^k$. From the theorem, $\langle -, x, 2^k \rangle$ is a reachable state. Since a register in β holds a value only by reading it from the counter, the counter value is x sometime during the computation.

3 Lower bound on the length of computation

Lemma 0: The number of steps taken by a machine in the shortest computation for any number is a multiple of 3.

Proof: In a shortest computation every read by a machine is followed by a subsequent write by that machine; otherwise the read could have been eliminated. Therefore, the last operation by each machine is a write, which can come only after two reads. Hence, the result. □

Corollary: All registers are empty at the end of a shortest computation.

Proof: The registers of a machine are left empty after each step by a machine.

Convention: Henceforth, a step means 3 substeps: read, read, write. The substeps of a machine may be interleaved with those of the other machine. Let $C(x)$ be the minimum number of steps to compute x.

Remark: Any computation that has $\langle -, -, p\rangle$ as an intermediate state has a multiple of p in the counter in all subsequent steps. It can be proved by induction on the number of substeps that every register and the counter hold a multiple of p. So, computation of a prime number never involves an intermediate state $\langle -, -, p\rangle$, where $p > 1$. □.

Lemma 1: $C(p \times q) \leq C(p) + C(q)$.

Proof: To compute $C(p \times q)$, first use a shortest schedule to compute $C(p)$. This leaves the machine in the state $\langle -, -, p\rangle$, from corollary to Lemma 0. Next, use the shortest schedule for computing q. Since we start with p in the counter, all subsequent steps have the effect of multiplying all values by p; so, $p \times q$ is in the counter at the end of the computation. □

Corollary: $C(2^{n+1}) \leq 1 + C(2^n)$.

$$
\begin{array}{ll}
 & C(2^{n+1}) \\
\leq & \{\text{from Lemma 1}\} \\
 & C(2) + C(2^n) \\
\leq & \{C(2) \leq 1, \text{ by construction of a schedule}\} \\
 & 1 + C(2^n)
\end{array}
$$
□

The computation described in the previous section is called the "standard computation", or "standard schedule". Let x have $\sigma(x)$ 1s in its binary representation, and $2^n \leq x < 2^{n+1}$. The standard computation involves $n + \sigma(x) - 1$ steps. Thus, the standard computation of 15 has $3 + 4 - 1 = 6$ steps. Now,

$$
\begin{array}{ll}
 & C(15) \\
\leq & \{\text{using Lemma 1}\} \\
 & C(3) + C(5) \\
\leq & \{\text{using standard computation for 3 and 5}\} \\
 & 2 + 3 \\
= & \{\text{arithmetic}\} \\
 & 5
\end{array}
$$

Hence the standard schedule is not necessarily the shortest.

Lemma 2: Let the shortest computation of z involve adding x and $z - x$. Then, $C(z) \geq 1 + C(x)$.

Proof: We take the shortest computation of z and show that by removing 3 substeps, we get a computation for x. From a schedule that computes z by adding x and $z - x$, remove the last step (of writing of z). The given schedule computes x at some point (because x and $z - x$ are in some machine's register). According to Lemma 0, this machine can discard two previous reads, and still

compute x. □

Lemma 3: For $k > 0$, $C(2^n) < C(2^n + k)$.
Proof: By induction on n.

Case $n = 0$**:** $C(2^0)$ is zero. Any number larger than 1 requires some computation.

Case $n + 1$**,** $n \geq 0$**:** We have to show $C(2^{n+1}) < C(2^{n+1} + k)$, for $k > 0$.
Let $z = 2^{n+1} + k$. Suppose z is computed by adding x to $z - x$. Assume that $x \geq z - x$. Then $x > 2^n$.

$$\begin{array}{ll} & C(z) \\ \geq & \{\text{from Lemma 2}\} \\ & 1 + C(x) \\ > & \{x > 2^n. \text{ Using induction, } C(x) > C(2^n)\} \\ & 1 + C(2^n) \\ \geq & \{\text{from corollary to Lemma 1}\} \\ & C(2^{n+1}) \end{array}$$

□

Lemma 4: $C(2^n) = n$.
Proof: By induction on n.

Case $n = 0$**:** $C(2^0)$ is zero.

Case $n+1$**,** $n \geq 0$**:** Since $2^{n+1} > 2^n$, $C(2^{n+1}) > C(2^n)$, from Lemma 3. From corollary to Lemma 1, $C(2^{n+1}) \leq 1 + C(2^n)$. Therefore, $C(2^{n+1}) = 1 + C(2^n)$. By induction hypothesis, $C(2^n) = n$. Therefore, $C(2^{n+1}) = 1 + n$. □

Lemma 5: $C(2^n) + 1 = n + 1$.
Proof:

$$\begin{array}{ll} & C(2^n + 1) \\ > & \{\text{from Lemma 3}\} \\ & C(2^n) \\ = & \{\text{from Lemma 4}\} \\ & n \end{array}$$

Now, the standard schedule to compute 2^n+1 has exactly $n+1$ steps. Hence, $C(2^n + 1) = n + 1$. □

Lemma 6: $C(2^m \times (2^n + 1)) = m + n + 1$.
Proof: We show that $C(2^m \times (2^n + 1)) \leq m + n + 1$ and $C(2^m \times (2^n + 1)) \geq m + n + 1$.

$$\begin{array}{ll} & C(2^m \times (2^n + 1)) \\ \leq & \{\text{from Lemma 1}\} \end{array}$$

$$
\begin{array}{ll}
 & C(2^m) + C(2^n + 1) \\
= & \{\text{from Lemma 4}\} \\
 & m + C(2^n + 1) \\
= & \{\text{from Lemma 5}\} \\
 & m + n + 1
\end{array}
$$

Next, we show $C(2^m \times (2^n + 1)) \geq m + n + 1$.

$$
\begin{array}{ll}
 & C(2^m \times (2^n + 1)) \\
= & \{\text{arithmetic}\} \\
 & C(2^{m+n} + 2^m) \\
\geq & \{2^{m+n} + 2^m > 2^{m+n}.\ \text{use Lemma 3}\} \\
 & 1 + C(2^{m+n}) \\
= & \{\text{from Lemma 4}\} \\
 & m + n + 1
\end{array}
$$

□

Lemma 7: Only the powers of 2 are computed during any shortest computation of 2^n.

Proof: We claim that in a shortest computation of 2^n, $n > 0$, 2^{n-1} and 2^{n-1} are added in the final step. The lemma follows from this result by induction.

Proof of the claim is by induction. It is easy to see for $n = 0$.

Case $n+1$, $n \geq 0$: Suppose x and y are added to form 2^{n+1}, where $x > 2^n$. Then,

$$
\begin{array}{ll}
 & C(2^{n+1}) \\
\geq & \{\text{from Lemma 2}\} \\
 & 1 + C(x) \\
> & \{x > 2^n.\ \text{use Lemma 3}\} \\
 & 1 + C(2^n)
\end{array}
$$

This contradicts Lemma 4: $C(2^{n+1}) = 1 + C(2^n)$. □

Lemma 8: Suppose $2^n + x$, where x is not a power of 2 is computed by adding 2^n and x. Then, $C(2^n + x) \geq 2 + n$.

Proof: A computation of 2^n does not compute x, from Lemma 7. So, it takes at least one substep to store x in the counter. Applying Lemma 2, at least one full step is required to add 2^n and x. □

4 Some Shortest Schedules

We compute the C values for some small integers.

n	$C(n)$	remarks
1	0	standard schedule
2	1	Lemma 4
3	2	Lemma 5
4	2	Lemma 4
5	3	Lemma 5
6	3	see below
7	4	see below
8	3	Lemma 4
9	4	Lemma 5
10	4	see below
11	≤ 5	standard schedule
12	4	see below

The computation of 7 involves adding either 1 and 6, 2 and 5, or 3 and 4. The first two computations involve at least $C(6) + 1$ and $C(5) + 1$ steps respectively, using Lemma 2. The last computation involves at least $C(4) + 2$ steps, using Lemma 8. Each of these equals 4 and the standard schedule has 4 steps.

The computation of 6: write 6 as $2^1 \times (2^1 + 1)$ and apply Lemma 6.

The computation of 10: write 10 as $2^1 \times (2^2 + 1)$ and apply Lemma 6.

The computation of 12: write 12 as $2^2 \times (2^1 + 1)$ and apply Lemma 6.