# Reversible Computation

S. Buss, 22 January 2013.

· Part I:

[Bennett '73; Bennett '89, Levine-Sherman '90].
[Also: Lecerf '63.]

Goal: Reversible simulation of deterministic computation.

"Poor man's" or "Global" reversibility.

→ Remember the input + count the number of steps.

→ Can reverse a computation step by returning to the initial configuration, and rerunning the entire computation up to the previous step.

Bennett '73 defined "reversible" more stringently:

No two configurations can lead to the same successor computation.

Slightly modified defn of multitape TM so that an instruction can either:

(a) Read a symbol + overwrite with a new symbol, or

(b) Ignore the read symbol (not even take into account) + move tape head -1, 0, or 1 squares right.

(Options (a) + (b) can be chosen differently on different tapes.)

Some things that can be done reversibly:

(1) Make a copy of a string (on top of blanks!)

(2) Erase (overwrite with blanks) a string if it is a copy of another string.

Bennett '73 construction

Let $M$ be a deterministic (multitape) TM, with $k$ tapes.
Reversibly simulate $M$ with a $(k+1)$ tape machine:

- The new tape holds a history of the transition rules used during the computation of $M$: ~~O(1)~~ one symbol per step of $M$:

Tape $k+1$: blank

Work tapes

Tapes 1-$k$: Input

simulates ↓ ↓ ↓ ← Run $M$ forward

Tape $k+1$: history

Tapes 1-$k$: Work tapes

output

↓↓↓ ← Copy output

Tape $k+1$: history

Tapes 1-$k$: Work tapes

Output | Output

↓ ↓ ↓ ← Run $M$ backwards

Tape $k+1$: blanks

Tapes 1-$k$: blank work

Input | Output

For machines that use space ~~O()~~ $o(n)$, the input is read only, not part of the modified tapes. We'll deal only w/ such machines that have a boolean output or short output.

The above simulation of $M$ uses

Time $O(T)$

Space $O(S+T)$

where $M \in TISP(T, S)$.

For poly-time, logspace machines this give a polyspace procedure.

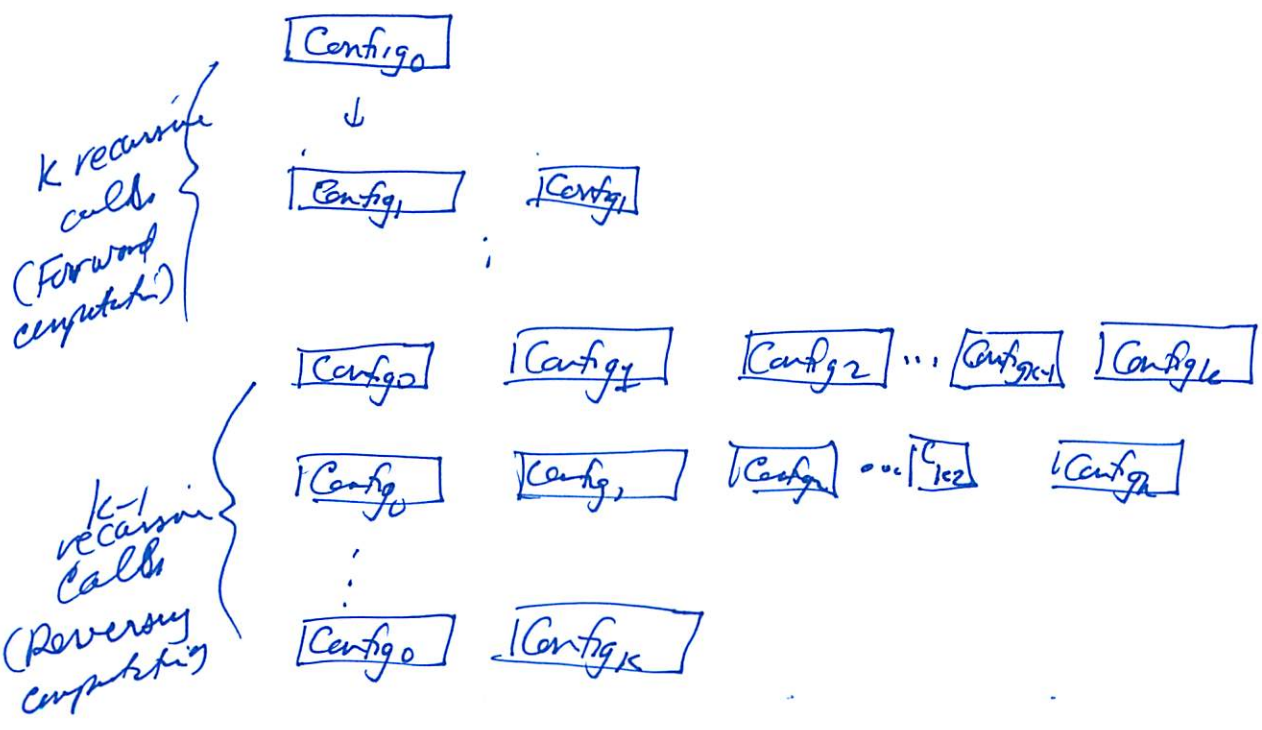
[Bennett '73]: Remarks you do a two level version of this and achieve Time $O(T)$ and Space $O(\sqrt{ST})$.

and that by using multiple nested levels can "perhaps" achieve TIME $O(T^2)$ and Space $O(S \log T)$.

This is carried out, and improved, in [Bennett '89 + Levine Sherman '90]

Parameters: $m$ = # of steps simulated at base level á la [Bennett '73.]

$k$ = # of blocks between recursive calls



At level $n$:

# of steps of $M$ simulated: $m \cdot k^n$.

Time used for one $\boxed{\text{Config}_0} \cdots \boxed{\text{Config}_k}$ computation at level $n$:

$$P_n = (2k-1)P_{n-1}$$

$$P_0 = m$$

So: $P_n = m(2k-1)^n$.

Space use is given by

$$S_n = \begin{cases} (k-1)m & \text{if } n=1 \\ (k-1)m + S_{n-1} & \text{if } n>1. \end{cases}$$

(Assuming $m \geq S$)

So $S_n = n \cdot m(k-1)$.

Take $m = S$ (for level $\emptyset$, history uses same space as T.M. M.)

Fix $k$.

Suppose $M$ ~~has~~ is in $TISP(T,S)$. $T = mk^n$ $S = m$

So $k^n = T/S$ ; $n = \frac{\log(T/S)}{\log k}$

[Levine-Sherman]:

The reversible computation uses

Space: $S' = S_n = S \cdot \log(T/S) \frac{k-1}{\log k}$.

and Time: $T' = P_n = T\left(\frac{T}{S}\right)^{\varepsilon}$

where $\varepsilon = \frac{\log(2 - 1/k)}{\log k}$.

and $\varepsilon \to 0$ as $k \to \infty$.

Pf: $T' = P_n = S \cdot (2k-1)^n = S \cdot k^n \cdot \left(\frac{2k-1}{k}\right)^n = T \cdot \left(\frac{2k-1}{k}\right)^{\log(T/S)/\log k}$

$= T \cdot \left(\frac{T}{S}\right)^{\varepsilon}$. qed

[Lange-McKenzie-Tapp '80] - A different space-optimal construction

Let $M$ be a deterministic, space $S$, Turing machine.

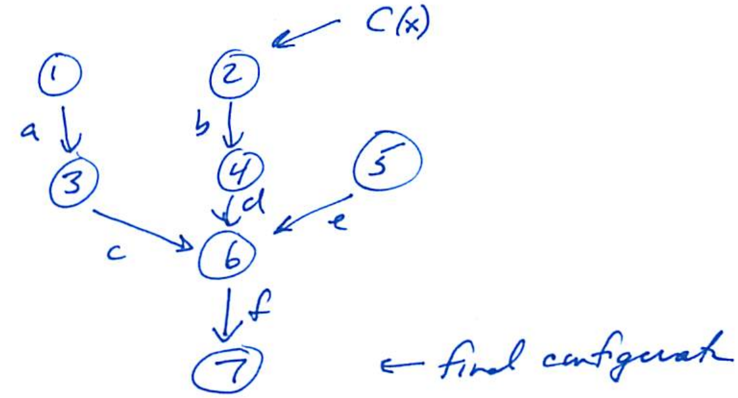
Assume initial configurations have no predecessor configurations and $M$ halts on all inputs.

<u>On input $x$:</u> Initial configuration is $C(x)$.

$C(x)$ determines a component in the <u>undirected</u> graph of one-step reachability among configurations

<u>Lemma</u> The connected component of $C(x)$ is treelike with a single root (sink node).

<u>Ex:</u>




From the directed graph $G$ of $M$'s configurations of length $S(|x|)$, form an "Eulerian tour" on the "edge ends" of $G$.

<u>Ex:</u> The edge ends there are $a1$, $a3$, $b2$, $b4$, $c3$, $c6$, $d4$, $d6$, $e5$, $e6$, $f6$, $f7$.

Each vertex of $G$ has finite fanin, corresponding to $M$ have only finitely many transition rules. Form an arbitrary cyclic permutation $\tau$ on the incident edge ends.

Also let $\pi$ swap the ends of ~~to~~ an edge.

<u>Ex:</u> $\pi(a1) = a3$ $\quad \pi(a3) = a1$, etc

$\tau(a3) = c3$ $\quad \tau(a1) = a1$,

$\tau(c6) = d6$ $\quad \tau(d6) = e6$, $\quad \tau(e6) = f6$ $\quad \tau(f6) = c6$.

Define • $\lambda = \pi\cancel{\sigma} = \pi\tau$

<u>Ex</u> $a1 \xrightarrow[\lambda]{} a3 \xrightarrow[\lambda]{} c6 \xrightarrow[\lambda]{} d4 \xrightarrow[\lambda]{} b2 \xrightarrow[\lambda]{} b4 \xrightarrow[\lambda]{} d6 \xrightarrow[\lambda]{} \dots$

Thus $\lambda$ is a cyclic permutation of the edge ends of the connected component of $C(x)$.

The reversible simulation just iterates $\lambda$ until reaching an (the) ~~f~~halting configuration.

It needs to remember only "the current edge-end"; hence uses only $O(S)$ space.

Hence ~~uses~~ time $2^{O(S)}$.

<u>Thm</u> $SPACE(S) \subseteq rev\text{-}SPACE(S)$.

[Buhrman-Tromp-Vitanyi '01; Williams, unpub.] gave some small improvements by combining Bennett's method with the LMT construction. Namely, the lower levels of Bennett's construction are replaced by a use of LMT.

<u>Williams conjecture</u> $TISP(T,S) \subseteq \cancel{rev\text{-}TISP(T^{1+\varepsilon}, 2}$
$rev\text{-}TISP(\max(T^{1+\varepsilon}, 2^{\varepsilon S}), S)$
should be possible. However current methods are not close to this yet.