Mechanized Undecidability of Halting Problems for Reversible Machines

Final Seminar Talk

Hizbullah A. A. Jabbar
Advisor: Andrej Dudenhefner
Supervisor: Prof. Gert Smolka

Saarland University

19/01/2022

Outline

Outline

Reversible Machines

- Reversible machines are those whose operations are injective.
- Reversibility is a dual to determinism.
- Interests in reversible machines stem from Landauer's Principe (Landauer, 1961)...
- ...where its converse is widely accepted to hold (Bennett, 2003).

Goal

The goal of this thesis is to mechanize in Coq the (un)-decidability of the halting problem for:

- Reversible 2-counters counter machine, and
- Reversible 2-dimensional cellular automata.

About FRACTRAN

- We showed that the halting problem for reversible FRACTRAN is decidable. . .
- . . . because it is difficult to store computation history in FRACTRAN.
- FRACTRAN is an example of a Turing-complete machine whose reversible counterpart has decidable halting problem.

Outline

Recap

Counter Machine

- Morita's Construction
- Graph representation
- Generic simulation lemmas

Cellular Automata

Future Work

Counter Machine

- Initially, we wanted to use existing counter machine formalizations in CLUP.
- The most notable are `MinskyMachine` (MM/MMA) and `CounterMachine` (CM) with two instructions: increment and decrement.
- However, it turns out that the halting problem for reversible MM and CM are decidable [1]...
- ...due to not having enough flexible control flow mechanism.
- Hence we use a different formalization: Morita's counter machine (Morita, 1996).

[1] https://github.com/uds-psl/coq-library-undecidability/pull/138

Morita's Counter Machine

- There are 5 instructions: increment, decrement, unconditional jump, zero test, and positive test.
- We model Morita's counter machine (MOR) as a list of *quadruples*.
- A quadruple (p, x, i, j) contains an instruction p, operates on the counter x, and relating the internal states i and j.
- A MOR is therefore a graph, where every quadruple (p, x, i, j) is an edge connecting state i to state j.
- We say that the quadruple (p, x, i, j) is *at* state i and *points to* state j.
- Translating a MMA/CM into a MOR is straightforward.

Morita's Counter Machine

- A MOR configuration is a pair $(i, \overrightarrow{v})$.
- Suppose we have the following MOR A: $[(\mathrm{INC}, 0, 0, 1), (\mathrm{DEC}, 0, 0, 1), (\mathrm{INC}, 0, 1, 0)]$.
- Starting from $(0, [0, 0])$, A could go to $(1, [1, 0]), (0, [2, 0]), (1, [1, 0]), \ldots$
- Alternatively, it could go to $(1, [1, 0]), (0, [2, 0]), (1, [3, 0]), \ldots$
- A MOR M *terminates* starting from a configuration s iff there exists a configuration t such that M takes some steps from s to t where it get stuck.
- A MOR M is extensionally reversible iff for all configurations s, t, and u: if M takes a step from s to u and it also takes a step from t to u then it must be that $s = t$.

Intensional Reversibility

$$A = [(\mathsf{INC}, 0, 0, 1), (\mathsf{DEC}, 0, 0, 1), (\mathsf{INC}, 0, 1, 0)]$$

- A is *not* extensionally reversible: The configuration $(1, [4, 0])$ can be reached in one step from $(0, [3, 0])$ or $(0, [5, 0])$.
- (Morita, 1996) proposed a syntactic criteria for reversibility via the so-called *range overlap*.
- Intuitively, two quadruples a and b overlap in range iff a and b point to the same state and they contain instructions that can change the value of the counters.
- A MOR is intensionally reversible iff none of its quadruples overlap in range.

Intensional Reversibility

$$A = [(\mathsf{INC}, 0, 0, 1), (\mathsf{DEC}, 0, 0, 1), (\mathsf{INC}, 0, 1, 0)]$$

- A is *not* intensionally reversible: $(\mathsf{INC}, 0, 0, 1)$ and $(\mathsf{DEC}, 0, 0, 1)$ overlap in range.
- In fact, intensional reversibility implies extensional reversibility.
- However, the converse is not true: there are MORs which are extensionally reversible but are not intensionally reversible.

Morita's Construction

- Morita's construction (Morita, 1996) provides a way to convert any k-MOR into an intensionally reversible 2-MOR.
- Undecidability of the halting problem of reversible MOR is then established via reduction from the halting problem of MMA.

Morita's Construction

Let M be a MOR with indegree n.

1. If $n \leq 1$ we are done. Otherwise, reduce the indegree to 2.
2. Add two extra counters to keep track of history, together with extra quadruples to work with them for every pair of quadruples that overlap in range.
3. Compress via prime exponentiation: instead of working with counters $\overrightarrow{v} = [v_1, v_2, v_3, v_4]$, use $[p_1^{v_1} p_2^{v_2} p_3^{v_3} p_4^{v_4}, 0]$ where p_1, p_2, p_3, p_4 are primes.

Graph representation

- It is more convenient to use a representation of MORs that has more structure.
- Specifically, since it is a graph, one can represent it as an adjacency list (= list of list), where each sublist contains quadruples that point to the same state.
- Furthermore, quadruples in different sublists point to different states.
- Computing indegree is trivial: the indegree of M is the length of the longest sublist in the graph representation of M.

Graph representation

The graph representation satisfies the following invariants:

uniformity
: For each sublists, every quadruples in the sublists point to the same state.

disjointness
: Quadruples in different sublists point to different states.

Lemma (Horizontal composability)

If every sublists is intensionally reversible, then the graph is intensionally reversible.

Generic simulation lemmas

- Morita's construction proceeds in stages where there are proof obligations to show that a stage simulates the preceeding one.
- As such, it is convenient to have a generic lemmas that, as long as those are satisfied, guarantee simulation with respect to termination, similar to e.g. (Leroy, 2009).
- Works for any machine whose termination is defined as "taking some steps and then get stuck".

Generic simulation lemmas

Let $\Rightarrow: X \to X \to \mathbb{P}$ and $\Rightarrow: Y \to Y \to \mathbb{P}$ be the step relations of two machines. There are two variants of the simulation lemmas so far:

- Lockstep simulation: For every step that $\Rightarrow$ takes, $\Rightarrow$ also takes a step.
- Many-step simulation: For every step that $\Rightarrow$ takes, $\Rightarrow$ takes *at least* one step.

Lockstep simulation

Let $\mathtt{sync} : X \to Y \to \mathbb{P}$ be a relation over configurations. If we have:

- For all s, t, and s', if given $s \Rightarrow t$ and $\mathtt{sync}\ s\ s'$, then we have $\exists t',\ s' \Rightarrow t' \wedge \mathtt{sync}\ t\ t'$.
- For all s', t', and s, if given $s' \Rightarrow t'$ and $\mathtt{sync}\ s\ s'$, then we have $\exists t,\ s \Rightarrow t \wedge \mathtt{sync}\ t\ t'$.

Then we have that for all s and s' that are in sync, $\Rightarrow$ terminates on s iff $\Rightarrow$ terminates on s'.

Many-step simulation

Let $\texttt{sync} : X \to Y \to \mathbb{P}$ be a relation over configurations such that for all s', either $\exists s,\ \texttt{sync}\ s\ s'$ or $\forall s,\ \neg\texttt{sync}\ s\ s'$. If we have

- For all s, t, and s', if given $s \Rightarrow t$ and $\texttt{sync}\ s\ s'$, then we have $\exists t',\ s' \Rightarrow^{+} t' \wedge \texttt{sync}\ t\ t'$.
- For all s', t', s, and t, if given $s' \Rightarrow^{+} t'$, $\texttt{sync}\ s\ s'$, and $\texttt{sync}\ t\ t'$, then we have $s \Rightarrow^{+} t$.

Then we have that for all s and s' that are in $\texttt{sync}$, $\Rightarrow$ terminates on s iff $\Rightarrow$ terminates on s'.

Outline

Recap

Counter Machine

- Morita's Construction
- Graph representation
- Generic simulation lemmas

Cellular Automata

Future Work

Cellular Automata

- A *one-dimensional* cellular automaton (CA) is a triple (Σ, r, f).
- Σ is a finite set of alphabet which is also called *states* in e.g. (Kari and Ollinger, 2008).
- $r \in \mathbb{N}$ is the neighborhood radius.
- $f : \Sigma^{2r+1} \to \Sigma$ is the local update rule.
- The configurations of a CA are elements of $\Sigma^{\mathbb{Z}}$ which changes through simultaneous applications of f:
  $c'(i) = f(c(i-r), c(i-r+1), \ldots, c(i+r-1), c(i+r))$.
  This is the *parallel map* of a CA.
- CAs represent massively parallel computations.

Reversible Cellular Automata

- Any d-dimensional CA can be simulated by a reversible $d+1$-dimensional CA (Toffoli, 1977).
- The undecidability of halting for a reversible 2-dimensional CA is established via undecidability of halting for 2-MOR.
- Need to construct a 1-dimensional CA that simulates a MOR.

Constructing 1-CA

- Instead of defining configurations as Σ^Z, we define a configuration $(l, a, r) : (\mathbb{N} \to \texttt{opt}\ \Sigma, \texttt{opt}\ \Sigma, N \to \texttt{opt}\ \Sigma)$.
- The parallel map is then defined accordingly e.g. $a' = f(l\,0, a, r\,0)$.
- A halting configuration is defined as

$$c_{halt} = \exists n,\ l\ n = \texttt{None} \land \exists n,\ r\ n = \texttt{None} \land a = \texttt{None}.$$

Outline

What has been done

- Full mechanization of reduction from MMA to MOR.
- Partial echanization of Morita's construction based on MOR.
- Mechanization of 1-CA and its halting problem.

Future work

Must-have goals:

- Finish mechanization of Morita's construction.
- Mechanize the reduction from 2-counter MOR to 1-dimensional CA.
- Mechanize the Toffoli construction (Toffoli, 1977).

Thank you for your attention!

MOR semantics

$$\frac{(\texttt{INC}, x, i, j) \in M \qquad v[x] = w}{M \vdash (i, \overrightarrow{v}) \Rightarrow (j, \overrightarrow{v}[w+1/x])} \qquad \frac{(\texttt{NOP}, x, i, j) \in M}{M \vdash (i, \overrightarrow{v}) \Rightarrow (j, \overrightarrow{v})}$$

$$\frac{(\texttt{DEC}, x, i, j) \in M \qquad v[x] = 1 + w}{M \vdash (i, \overrightarrow{v}) \Rightarrow (j, \overrightarrow{v}[w/x])}$$

$$\frac{(\texttt{ZER}, x, i, j) \in M \qquad v[x] = 0}{M \vdash (i, \overrightarrow{v}) \Rightarrow (j, \overrightarrow{v})}$$

$$\frac{(\texttt{POS}, x, i, j) \in M \qquad v[x] = 1 + w}{M \vdash (i, \overrightarrow{v}) \Rightarrow (j, \overrightarrow{v})}$$

Range Overlap

Let $a = (p_1, x_1, i_1, j_1)$ and $b = (p_2, x_2, i_2, j_2)$ be two quadruples and let $D = \{\texttt{INC}, \texttt{DEC}, \texttt{NOP}\}$.

Definition (Range overlap)

α and β overlap in range iff

$$j_1 = j_2 \wedge (x_1 \neq x_2 \vee p_1 = p_2 \vee p_1 \in D \vee p_2 \in D$$

Constructing Graph Representation

Let $f : X \to X \to \mathbb{P}$.

$$\begin{aligned}
\texttt{to_graph}\,[] &= [] \\
\texttt{to_graph}\,(h : t) &= (\texttt{filter}\,(f\ h)\ t) \\
&\quad :: \ \texttt{to_graph}\,(\texttt{filter}\,(\neg\,(f\ h)))
\end{aligned}$$

CA Parallel Map

Let A be a 1-dimensional CA. Let $f : (\Sigma, \Sigma, \Sigma) \to \Sigma$ be its local update rule and
$c = (l, a, r) : (\mathbb{N} \to \mathtt{opt}\ \Sigma, \mathtt{opt}\ \Sigma, N \to \mathtt{opt}\ \Sigma)$ be one of its configurations.

$$
\begin{aligned}
a' &= f(l\,0, a, r\,0) \\
l' &= \lambda n \to [0 \Rightarrow f(l\,1, l\,0, a)\,|S\ n' \Rightarrow f(l\,(S\ n), l\ n, l\ n')] \\
r' &= \lambda n \to [0 \Rightarrow f(l\,1, l\,0, a)\,|S\ n' \Rightarrow f(r\ n', r\ n, r\,(S\ n))]
\end{aligned}
$$