15-354: CDM **K. Sutner**

Assignment 7 Due: **November 10, 2023 24:00**.

1. Presburger and Skolem Arithmetic (50)

Background

Presburger arithmetic is the fragment of arithmetic that has addition but no multiplication; Skolem arithmetic is the fragment that has multiplication but no addition. Both turn out to have decidable first-order theory (in stark contrast to general arithmetic).

It is a folklore result that addition is rational and even synchronous. More precisely, there is a synchronous relation $\alpha(x, y, z)$ that expresses $x+y = z$ where the numbers are given in reverse binary. There is no corresponding synchronous (or even rational) relation $\mu(x, y, z)$ for multiplication. However, we can choose a different encoding from $\mathbb{N}_+$ to $\mathbf{2}^\star$ that makes multiplication rational. Alas, this encoding is slightly strange and mangles addition badly.

Task

A. Show that the addition relation α is rational for numbers in reverse binary.

B. Argue that α is actually synchronous.

C. Show that multiplication is not rational for numbers in reverse binary.

D. Concoct an encoding of the positive natural numbers that makes multiplication rational. Hint: think about primes.

E. Exlain why your encoding does not work for addition.

Comment

For part (A), you can either write a rational expression or draw a diagram. This is a bit of a pain; whatever you do, try to make it look nice and explain how and why it works. Don't try to enforce the no-trailing-zeros condition, it makes the machine too messy and requires 16 states. Allowing trailing zeros, 7 states are enough.

For (C), think of the numbers as as bit patterns and do some surgery, based on a loop the machine must necessarily enter. Then consider their numerical values.

For (E), no formal proof is necessary, just a reasonable explanation.

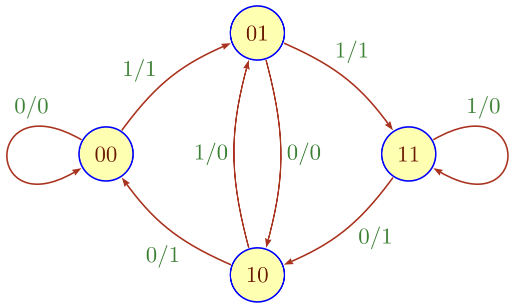
2. Biinfinite Cellular Automata (50)

Background

In class we showed how to use transducers operating on standard, finite words to model check elementary cellular automata over finite grids. In classical symbolic dynamics, one is more interested in spaces of the form

$$\mathcal{C} = \langle \mathbf{2}^{\mathbb{Z}}, \twoheadrightarrow \rangle$$

Perhaps surprisingly, it is relatively easy to construct a 2-track transducer that implements the global map for elementary cellular automata on $\mathbf{2}^{\mathbb{Z}}$. For ECA 90, the exclusive or of the left and right neighbor, the machine $\mathcal{A}_{\twoheadrightarrow}$ looks like so:



Acceptance here means: there is a biinfinite path that is labeled $x{:}y$. More precisely, given a biinfinite run $\pi = (p_i)_{i\in\mathbb{Z}}$ of an automaton, define the postively/negatively recurrent states to be

$$\mathsf{rec}_+(\pi) = \{\, p \mid \overset{\infty}{\exists}\, i \in \mathbb{N}\, (p_i = p) \,\}$$

$$\mathsf{rec}_-(\pi) = \{\, p \mid \overset{\infty}{\exists}\, i \in \mathbb{N}\, (p_{-i} = p) \,\}$$

If we think of $I = F = Q$ in the last example, acceptance means $\mathsf{rec}_-(\pi) \cap I \neq \emptyset$ and $\mathsf{rec}_+(\pi) \cap F \neq \emptyset$. By choosing the initial/final states I and F appropriately, we can control the behavior of a machine in more detail. For simplicity, let's call such a machine $\langle Q, \mathbf{2}, \delta; I, F\rangle$ operating on biinfinite words a $\mathbb{Z}$-automaton.

In class we also mentioned the almost-equal relation $\overset{*}{=}$ on $2^{\mathbb{Z}}$: x and y differ in at most finitely many places. More formally,

$$\exists\, n \in \mathbb{N}\, \forall\, i \in \mathbb{Z}\, (|i| \geq n \Rightarrow x_i = y_i)$$

This relation turn out to be useful, but we won't worry about applications.

Task

A. Eplain why we can safely assume that initial and final states in a $\mathbb{Z}$-automaton consist of non-trivial strongly connected components (possibly more than one).

B. Construct a $\mathbb{Z}$-automaton that implements the non-equal relation.

C. Construct a $\mathbb{Z}$-automaton that implements the almost-equal relation.

D. Explain how to check whether a biinfinite ECA has an injective global map.

E. Try to turn the construction form part (D) into a nice quadratic time algorithm that checks injectivity of ECA.

Comment For (D), it is easier to think about the negated property (which comes down to an existentially quantified formula that is easier to check).

For part (E) you need to think about the structure of the machine you built in (D) and figure out what graph theoretic properties are crucial here. A priori, quadratic time makes no sense for ECA (there are only 256, after all), but the construction should work directly for cellular automata using more than just 2 states per cell. So, if there are k possible states, the machine $\mathcal{A}_{\twoheadrightarrow}$ has k^2 and k^3 transitions; the algorithm should be quadratic in the size of this machine.