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

Assignment 7 Due: **April 9, 2021**.

1. Presburger and Skolem Arithmetic (30)

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 predicate 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.

For part (C) you may freely use pumping lemmings and the like.

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

2. MSO Descriptions (30)

Background

We have seen in class that the regular languages are precisely the ones that can be described by a $\mathsf{MSO}[<]$ formula φ:

$$L_\varphi = \{\, w \in \Sigma^\star \mid w \models \varphi \,\}.$$

Here one interprets a word w as a structure consisting of a the actual string together with positions ranging over $[|w|]$. Using multiple tracks it is straightforward to work with several words of the same length and use the formula to describe length-preserving relations rather than just plain languages. Let us write $Q_{a,w}(x)$ to mean that "word w has a letter a in position x," where w indicats one of the tracks.

Write $\mathsf{bin}(x)$ for the numerical value of a word $x \in \mathbf{2}^\star$ written in reverse binary; for example, $\mathsf{bin}(01101) = 22$.

Task

A. Construct a formula φ for the language $(a\{a,b\})^\star a$.

B. Construct a formula ψ such that $(u,v,w) \models \psi$ if, and only if, $\mathsf{bin}(u) + \mathsf{bin}(v) = \mathsf{bin}(w)$. Recall that the words are required to all have the same length, there is no overflow.

Comment

The last problem might be useful.

3. MSO and Regular Languages (40)

Background

According to Büchi's theorem, monadic second-order logic defines exactly the regular languages. However, we gave no proof for the direction "MSO implies regular." Here goes. We are trying to show that for every sentence φ in $\mathsf{MSO}[<]$ the associated language

$$\mathcal{L}(\varphi) = \{\, w \in \Sigma^\star \mid w \models \varphi \,\}$$

is in fact regular. The proof uses induction on the buildup of φ. Alas, there is the usual problem with semantics: the components of a sentence typically contain free variables, they are not sentences themselves. To deal with free variables it is best to consider augmented words over the alphabet

$$\Gamma = \Sigma \times (\mathbf{2})^k$$

where k is the number of variables, both first-order and second-order. In other words, we add an appropriate number of binary tracks to the original word that can be used to interpret the free variables in the quantifier-free part. There is no need for padding, the additional tracks have the same length as w. This is essentially the same machinery as for monadic second-order logic on infinite words discussed in class.

Also recall our formula $\mathsf{even}(X)$ for "X has even cardinality" from class.

Task

A. Prove by induction on the buildup of φ that $\mathcal{L}(\varphi)$ is regular. You may safely assume that the formula is in prenex normal form.

B. What can you say about the state-complexity of $\mathcal{L}(\varphi)$? Specifically, how large is the machine for the quantifier-free part and what happens when you deal with the quantifiers?

C. Translating $\mathsf{even}(X)$ into a FSM following your inductive definition and the formula from class would produce a large and ugly machine. Construct a small machine directly for this formula.

Comment

For part (B) don't try to get precise results, just a rough estimate of a how large these machines could be compared to the size of the formula. Part (C) is important: sometimes one can streamline machines quite a bit compared to the cookie-cutter result from the algorithm. In particular it may be helpful to enlarge the language by certain predicates with pre-defined machines.