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

Assignment 9 Due: **Nov. 26, 2024, 24:00**.

1. Arden's Lemma (30)

Background

Fix some alphabet Σ and let $\mathcal{L}(\Sigma)$ be the language semiring over Σ, including the Kleene star operation (a closed semiring with super idempotency). Given any two languages A and B over Σ we can solve the language equation

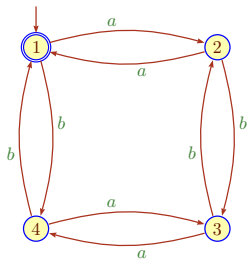
$$X = A \cdot X + B \tag{$\dagger$}$$

The claims below about solutions of this equation are known as Arden's lemma.

Let $\mathcal{A} = \langle Q, \Sigma, \tau; I, F\rangle$ be an NFA over Σ. For every state p, introduce a variable X_p and an equation

$$X_p = \Delta_p + \sum \big(a\, X_q \mid (p, a, q) \in \tau\big)$$

Here $\Delta_p = \varepsilon$ if $p \in F$ and $\emptyset$ otherwise. We get a system of equations $\mathcal{E}$ and the language of $\mathcal{A}$ has the form $\sum_{p\in I} X_p$.



One can use Arden's lemma to tackle this system of equation and ultimately winds up with a regular expression for the language of $\mathcal{A}$.

Task

A. Show that the equation ($\dagger$) always has a solution $X_0 = A^\star B$. In fact, X_0 is the smallest solution.

B. Moreover, if $\varepsilon \notin A$, then this solution is unique.

C. Construct $\mathcal{E}(\mathcal{A})$ for the even/even automaton from above.

D. Solve the system by repeated applications of Arden's lemma.

Comment For the last part, the right angle of attack is critical, otherwise you will wind up with horrific regular expression. In general, use common sense to simplify your regular expression, say, $1\alpha = \alpha$, $0 + \alpha = \alpha$, $0\alpha = 0$, $(1 + \alpha)^* = \alpha^\star$ and so on.

2. Functions versus Polynomials (30)

Background

For a domain A with good algebraic properties, functions $f : A^n \to A$ can sometimes be described in terms of polynomials: find some multivariate polynomial P such that $f = \widehat{P}$. We have written $\widehat{P}$ for the function $A^n \to A$ induced by the polynomial $P \in A[x_1, \ldots, x_n]$. In general there are not enough polynomials to describe all functions (just think about the reals), but for arithmetic modulo a prime everything works out nicely.

Task

A. Let p be an arbitrary prime. Show that any function $f : \mathbb{Z}_p^n \to \mathbb{Z}_p$ can be expressed in terms of a polynomial $P(x_1, \ldots, x_n)$.

B. For $p = 2$, what does this have to do with Boolean functions?

C. How about functions $f : \mathbb{Z}_m^n \to \mathbb{Z}_m$ for a general modulus m? Say, $m = 4$?

D. Testing whether a single polynomial equation $P(x_1, \ldots, x_n) = 0$ over the integers has a solution is undecidable. By contrast, show that over $\mathbb{Z}_2$ we can check in polynomial time whether there is a solution.

Comment

For the last part, we are not talking about a system of equations, just a single one.

3. Shrinking Dimension (40)

Background

As we have seen in class, there is a unique finite field $\mathbb{F}$ of size p^k for any prime p and $k \geq 1$. In one standard implementation we then think of $\mathbb{F}$ as a vector space of dimension k over $\mathbb{F}_p$, so the field elements are vectors of modular numbers. However, it is sometimes more convenient to deal with a lower-dimensional vector space over a larger ground field. More precisely, it may be better to build a tower of subfields

$$\mathbb{F}_p \subseteq \mathbb{K} \subseteq \mathbb{F}$$

and then to interpret $\mathbb{F}$ as a lower-dimensional vector space over $\mathbb{K}$. Alas, this only works under special circumstances which will be described in this problem.

Fix some prime characteristic p throughout.

Task

A. Show that the following are equivalent, where $1 \leq \ell \leq k$:

   (a) ℓ divides k

   (b) $p^\ell - 1$ divides $p^k - 1$

   (c) $x^\ell - 1$ divides $x^k - 1$ (in the polynomial ring $\mathbb{F}_p[x]$).

B. Show that if $\mathbb{K}$ is a subfield of $\mathbb{F}$ then $\mathbb{K}$ is (isomorphic to) $\mathbb{F}_{p^\ell}$ where ℓ divides k.

C. Show that if ℓ divides k then $\mathbb{F}_{p^\ell}$ is (isomorphic to) a subfield of $\mathbb{F}$.

Comment

The last item is the hardest; think splitting fields.

4. Building A Finite Field (40)

Background

As we have seen in class, there is a unique finite field of size p^k for any prime p and $k \geq 1$. Needless to say, the case $p = 2$ it is particularly interesting for actual implementations: the prime field can naturally be represented by bits and the arithmetic operations are given by xor (addition) and and (multiplication).

Building a finite field $\mathbb{F}_{2^k}$ requires a little more work.

Task

A. Show how to construct the finite field $\mathbb{F}$ of size 256. What data structures would you use, how would you implement arithmetic in this field?

B. How many primitive elements are there in this field?

C. What are all the subfields of $\mathbb{F}$? Why?

D. If we had constructed a field of size 32, what would the subfields be?

E. What is the main difficulty in doing a similar construction for the field of size 2^{1024}?

Comment Try to come up with a reasonable implementation without going overboard, industrial strength algorithms are actually quite tricky and complicated.