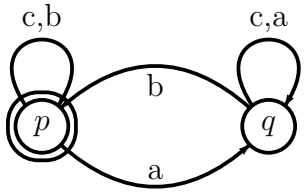
Lecture 1a: Monoids: An Example and an Application

Consider the language L of words over the alphabet $\{a, b, c\}$ consisting of all the words in which the last a (if any) does not occur to the right of all the b's. In other words, for every a there must a b somewhere to its right. Here is the minimal automaton for this language:



By the exercise on page 4 of Lecture 1, the syntactic monoid of L is the transition monoid of the above automaton. We now compute this transition monoid. Clearly δ_ϵ is the identity and as a matter of fact it equals δ_{c^i} for any $i \geq 0$. Further, δ_a is the constant function that returns q while δ_b is the constant function that returns p. Finally, it is easy to verify that $\delta_{wav} = \delta_a$ for $v \in c^*$ and $\delta_{wbv} = \delta_b$ for $v \in c^*$. Hence the syntactic monoid of L is the monoid $U_2 = (\{e, 1, 2\}, ., e)$ with the following multiplication table:

	e	1	2
e	e	1	2
1	1	1	2
2	2	1	2

The monoid U_2 is the monoid with two *reset* elements (we say y is a reset if $x.y = y$ for all x), and it plays an important role in the study of the algebraic theory of regular languages.

Exercise: Enumerate all the languages over $\{a, b\}$ whose syntactic monoid is U_2.

An Application: Alphabetic Languages

Given a monoid $(M, ., e)$, we say that $i \in M$ is an *idempotent* if $i.i = i$. A monoid is said to be idempotent if every element is an idempotent. The monoid U_2 is idempotent. A monoid $(M, ., e)$ is said to be commutative if $p.q = q.p$ for each $p, q \in M$. The monoid U_2 is not commutative. For example, the monoid $M = (\{e, p, q\}, ., e)$ with the following multiplication table is commutative (and idempotent).

	e	p	q
e	e	p	q
p	p	p	q
q	q	q	q

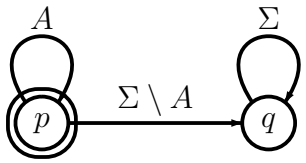
Consider the morphism $h : \{a, b, c\}^* \longrightarrow M$ that sends a, b to p and c to q. Observe that $h^{-1}(q)$ is all the words that contain at least one c and $h^{-1}(p)$ is all the words that contain at least one a or one b but no c's. As a matter of fact any two words which contain the same set of letters are mapped by h to the same element of M. For example $h(abbacbacaa) = h(abc) = q$. This is not peculiar to h or M, but true of any morphism into any idempotent commutative monoid, as stated below, where $\alpha(w)$ is the set of letters that occur in the word w.

Lemma 1 *Let $(M, ., e)$ be any commutative, idempotent monoid and let $h : (\Sigma^*, ., \epsilon) \longrightarrow (M, ., e)$ be any morphism. If w, w' are words over Σ such that $\alpha(w) = \alpha(w')$ then $h(w) = h(w')$. Consequently, if L is any language recognized by M then either both w and w' belong to L or neither of them belong to L.*

We say $w \equiv_\alpha w'$ iff $\alpha(w) = \alpha(w')$. This is a finite index congruence on Σ^*. For each $A \subseteq \Sigma$ there is exactly one equivalence class containing all the words w such that $\alpha(w) = A$. From the above lemma, $\equiv_\alpha$ saturates any L recognised by a commutative, idempotent monoid. Thus, such a language is just the union of some of the equivalence classes of $\equiv_\alpha$.

Notice that $\{w|\alpha(w) = A\}$ is exactly the same as the language $A^* \setminus \bigcup_{a \in A}(A \setminus a)^*$. Consequently, every language definable by a commutative, idempotent monoid can be defined from languages of the form A^*, $A \subseteq \Sigma$, using unions and complementation (i.e. they are *boolean combinations* of languages of the form A^*, $A \subseteq \Sigma$.)

Interestingly, the converse is true as well. As a first step, observe that if $A \subseteq \Sigma$ then the following is a minimal automaton for A^*.



We leave it as an easy exercise to verify that the transition monoid of this automaton is indeed commutative and idempotent. Thus, every language of the form A^* is recognised by a commutative, idempotent monoid. Further, the complement of such languages are recognised as well, as implied by the following proposition.

Proposition 2 *If L is recognised by a monoid M then so is $\overline{L}$.*

What about $L_1 \cup L_2$? The following lemma explains how to contruct a monoid recognising $L_1 \cup L_2$.

Proposition 3 *Let $M_1 = (M_1, ., e_1)$ and $M_2 = (M_2, ., e_2)$ recognize the languages L_1 and L_2 via the morphisms h_1 and h_2 and sets X_1 and X_2 respectively. Let $M_1 \times M_2$ be the monoid given by $(M_1 \times M_2, ., (e_1, e_2))$ where $(x_1, y_1).(x_2, y_2) = (x_1.x_2, y_1.y_2)$. Then $h : (\Sigma^*, ., \epsilon) \longrightarrow (M_1 \times M_2, ., (e_1, e_2))$, with $h(w) = (h_1(w), h_2(w)$ is a morphism and $L_1 \cup L_2 = h^{-1}(M_1 \times X_2 \cup X_1 \times M_2)$ and $L_1 \cap L_2 = h^{-1}(X_1 \times X_2)$.*

The monoid $M_1 \times M_2$ is called the *product* of M_1 and M_2. Here is an useful fact about commutative, idempotent monoids:

Proposition 4 *If M_1 and M_2 are commutative, idempotent monoids then so is $M_1 \times M_2$.*

Notice that this means that the class of languages recognized by commutative, idempotent monoids is closed under union and intersection. Thus, the boolean combination of languages of the form A^* are also recognized by commutative, idempotent monoids, giving the following theorem.

Theorem 5 *The class of languages recognised by commutative, idempotent monoids are exactly the class of languages that can be expressed as boolean combinations of languages of the form A^* ($A \subseteq \Sigma$).*

As we shall see later there are many more interesting examples of classes of languages that can be characterized using appropriately chosen classes of monoids. However, there remains the question: Why are were interested in such characterizations? To suggest an answer to this question, we establish one more property of the class of commutative, idempotent monoids.

Proposition 6 *The class of commutative, idempotent monoids is closed under submonoids and homomorphic images. That is, if $M \prec N$ and N is commutative and idempotent then so is M.*

Proof: Closure under submonoids is trivial. Suppose $h : (M, ., e) \longrightarrow (N, ., f)$ is a surjective morphism. Then, if $x, y \in N$ there is $p, q \in M$ such that $h(x) = p$ and $h(y) = q$. Thus,

$$p.q = h(x).h(y) = h(x.y) = h(y.x) = h(y).h(x) = q.p$$

Further, $p.p = h(x).h(x) = h(x.x) = h(x) = p$. Thus $(N, ., e)$ is also commutative and idempotent. ■

As an immediate corollary we see that

Corollary 7 *If L is recognised by a commutative, idempotent monoid then $\mathsf{Syn}(L)$ is commutative and idempotent.*

This result allows us to check if a given regular language can be expressed as the boolean combination of languages of the form A^* — compute its syntactic monoid and check that it is commutative and idempotent. As we shall see later for many interesting class of languages one can follow the same route as described above: characterize the class of monoids recognising them, show that the class is closed under division and check membership by checking properties on the syntactic monoid.

References

[1] V. Diekart, P. Gastin and M. Kufleitner: A Survey on Small Fragments of First-Order Logic over Finite Words, Int. J. Found. Comput. Sci., Vol 19, 2008.