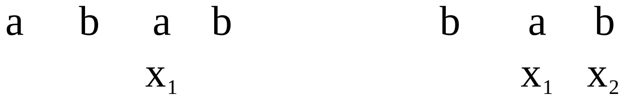
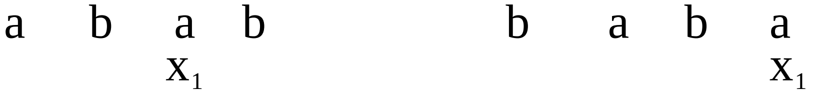
Lecture 4: EF games and first order definability

At the end of the last lecture we showed that a winning strategy for player 0 on the k round game played on words w and w' guarantees the existence of a formula with quantifier depth bounded by k that distinguishes w and w'. Do distinguishing formulas lead to winning strategies?

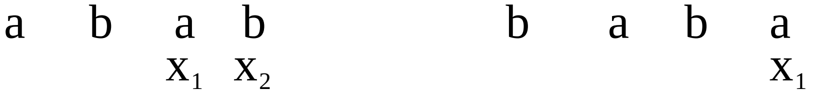
Consider the words $w = abab$ and $w' = baba$. One formula that distinguishes these two words is $\phi = \forall x.\ (a(x) \Rightarrow \exists y.\ (y > x))$. This formula is satisfied by w and not by w'. Here is how we synthesise a winning strategy for player 0 from this formula: Since w' does not satisfy ϕ, there is an i so that with $x = i$ the formula $a(x) \Rightarrow \exists y.\ (y > x)$ is not satisfied. Player 0 picks the word w' and picks this position i (4) giving:



Now, player 1 will place his x_1 against one of the two a's in w. Say, he picks the a at position 3 to give:



Now, w with $x = 3$ satisfies $\exists y.\ (y > x)$ while w' with $x = 4$ does not. Player 0 then picks the witness for this in word w, i.e. position 4 and places his x_2 there.



And now every move that player 1 makes will be losing.

The strategy construction using a distinguishing formula proceeds in the following way:

1. If the distinguishing formula is a quantifier free formula then clearly player 0 wins even the 0 round game.

2. If the distinguishing formula is of the form $\neg\phi$ then ϕ is also a distinguishing formula and we use that instead.

3. If the distinguishing formula is of the form $\phi_1 \wedge \phi_2$ then at least one of ϕ_1 or ϕ_2 is also a distinguishing formula and we use that instead.

4. If the distinguishing formula is $\exists x.\phi(x)$ then in one of the words w or w' there is a position i such that $\phi(x)$ is true when x is assigned the position i. Player 0 picks this word and the position i as his move. In the other word no matter which position we assign to x, $\phi(x)$ is not satisfied. Thus, no matter how player 1 responds to this move, resulting pair of words will be distinguished by $\phi(x)$, a formula of lower quantifier depth.

This is a winning strategy as player 0 has arranged things so that the words formed after each move are distinguished by formulas of lower quantifier depth. Thus after k rounds, where k is the quantifier depth of the formula distinguishing the original words, he is left with a quantifier free formula that distinguishes the two words. This gives us the following theorem.

Theorem 1 *Two V-words are distinguished by a formula of quantifier depth k if and only if player 0 has a winning strategy in the k round EF game associated with these words. In particular, if two words over Σ are distinguished by quantifier depth k sentences if and only if player 0 has a winning strategy in the k round EF game associated with these words.*

Proof: The direction from formulas to winning strategies is described above. The other direction was proved in the last class. ■

0.1 Evenness is not first-order definable

We now show that the words a^m and a^{m+1} are not distinguishable by quantifier depth k sentences if $m \geq 2^k$. We do this by showing that player 1 has a winning strategy in the k round game defined by these words. The proof proceeds by induction. When $k = 0$ we have the words a^i and a^{i+1}, $i \geq 1$ are clearly indistinguishable by atomic formulas (over the empty set of variables!).

Let us suppose that player 1 has a winning strategy in the r round game over (a^m, a^{m+1}), for all $r < k$ and $m \geq 2^r$. Consider the k round game over the words (a^m, a^{m+1}) with $m \geq 2^k$.

The move by player 1 would divide one of the words (the word that he picks) into three parts so that it looks like $a^s\ a\ a^t$ (where $s+1+t = m$ or $s+1+t = m+1$, depending on which word was picked). But note that either s or t is at least 2^{k-1}.

Suppose $t \geq s$ then player 1 picks position $s+1$ in the other word and plays that as his response. This ensures that w.r.t. the first variable placed on both words, the words to the left are identical and the words to the right are both of length $\geq 2^k$. After the first move the words look like $a^s(a,x)a^t$ and $a^s(a,x)a^{t'}$ where $|t - t'| = 1$, $t, t' \geq 2^{k-1}$. Thus, by the induction hypothesis, player 1 has a winning strategy on the $k-1$ round game on $(a^t, a^{t'})$. From now on, whenever player 0 picks a position among the initial $s+1$ positions in one word, player 1 simply duplicates the move in the other word. If player 0 picks a position in a^t (or $a^{t'}$) then player 1 responds using his strategy on the game $(a^t, a^{t'})$. It is not difficult to check that this is a winning strategy for player 1.

The construction of the strategy when $s \geq t$ proceeds similarly. Thus we have established that player 1 has a winning strategy in the k round EF game played on the words a^{2^k} and a^{2^k+1}.

We have just shown that player 1 has a winning strategy in this game and thus player 0 cannot have a winning strategy in this game. Thus, there is no quantifier depth k formula that can distinguish the words a^{2^k} and $a^{2^{k-1}}$.

Theorem 2 *The language $\{a^{2n} | n \geq 0\}$ is not definable in the first-order logic of words.*

Proof: Suppose ϕ is a formula that defines this language. Let the quantifier depth of ϕ be k. Then, either a^{2^k} and a^{2^k+1} are both in $L(\phi)$ or neither is in $L(\phi)$. This contradicts the definition of ϕ. ■

0.2 EF games for MSO

We can extend EF games to allow "second order" moves: In such a move, player 0 picks a subset of positions in one of the two words and labels them all by a second order variable X. In reponse player 1 must pick a subset of positions in the other word and label them all with X. As usual, after the k rounds are played, the winner is determined by whether the two resulting models satisfy the same set of quantifier free formulas or not (or equivalently by atomic formulas or not).

It is quite easy to establish that two words are distinguishable by MSO formulas of quantifier depth k if and only if player 0 has a winning strategy in the k round second-order EF game (where both first order and second order moves are allowed) played over these words.

An Application: We shall use this characterization to give an alternative proof of the fact that MSO formulas define regular languages: Let us write $w \sim_k w'$ to denote that player 1 has the winning strategy in the k round game on (w, w') (Equivalently, w and w' are indistinguishable via quantifier depth k formulas). Clealy $\sim_k$ is a equivalence relation on Σ^*. Moreover, a simple extension of theorem 1 from the previous lecture, shows that it is also of finite index.

We show that this relation is right invariant. Consider any pair of $w.z$ and $w'.z$. The winning strategy for player 1 is the following: when player 0 picks a position in z duplicate the move in the other word. Whenver player 0 picks positions in w or w' respond using the winning strategy on (w, w'). It is not difficult to check this is a winning strategy for player 1. Thus $w.z \sim_k w'.z$.

Finally observe that for any ϕ with quantifier depth k, $L(\phi) = \bigcup_{x \models \phi} [x]_{\sim_k}$. Hence, $\sim_k$ is a right congruence of finite index that saturates $L(\phi)$. Thus $L(\phi)$ is a regular language.

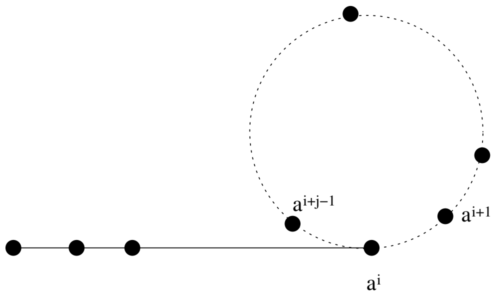
1 Aperiodic Monoids, Star-free sets and FO definability

We say that a monoid M is group-free if it does not contain a nontrivial (i.e. other than the 1 element group) subgroup (Note that we don't insist that the identity of the monoid appear in the group).

An element i in a monoid M is said to be an *idempotent* if $i.i = i$.

Lemma 3 *(Frobenius) Any finite cyclic semigroup contains idempotents.*

Proof: Consider a sequence $a, a^2, a^3, \ldots$ (where $a^i = a.a.a.\ldots.a(i \text{ times})$). There is some least i and least $j > i$ such that $a^i = a^{i+j} = a^ia^j$. Here, j called the *period* of the element a. Thus the sequence looks like the following lollipop:



Thus, $a^i = a^ia^j = a^ia^ja^j = \ldots = a^i(a^j)^n \ldots$. Our aim is to find a idempotent, i.e. an element of the form a^{i+k} such that $a^{i+k}.a^{i+k} = a^{i+k}$. This would clearly be true if $i + k$ is divisible by j (since $a^{i+k+i+k} = a^{i+mj+k} = a^{i+k}$). Thus, any $i + k$ with $i + k$ divisible by j is an idempotent. ■

This allows us to characterize group-free monoids as follows:

Lemma 4 *A monoid M is group-free if and only if there is an N such that for $a \in M$, $a^N = a^{N+1}$.*

Proof: Suppose, there is such a N and suppose G is a group contained in M. Let $a \in G$ and let a^{-1} be its inverse in G. Therefore $a^N = a^{N+1}$ implies $a^N.(a^{-1})^N = a.a^N.i(a^{-1})^N$. Thus $id_G = a$. Thus the group is trivial.

Suppose M is group-free. Pick any $a \in M$. By Lemma 3, $a^i = a^{i+j}$ for some j. If $j > 1$, then $\{a^i, a^{i+1}, \ldots, a^{i+j-1}\}$ forms a nontrivial subgroup of M. Thus $j = 1$. Thus $a^i = a^{i+1}$. Since M has only finitely many elements it follows that we can pick a N such that $a^N = a^{N+1}$ for all $a \in M$. ■

A monoid M is said to be *aperiodic* if there is a N such that $a^N = a^{N+1}$ for each $a \in M$. The above lemma shows that aperiodic monoids and group-free monoids are the same.

We say that a regular language is aperiodic if it is recognized by some aperiodic monoid. We next show that a regular language L is recognized by an aperiodic monoid if and only if its syntactic monoid is aperiodic.

Theorem 5 *Let L be recognized by a morphism $h : \Sigma^* \to M$ and suppose M is aperiodic. Then $\mathsf{Syn}(L)$ is aperiodic.*

Proof: Let N be such that $x^N = x^{N+1}$ for all $x \in M$. By Theorem 3 of lecture 1, η_L factors via h as follows:

$$\begin{tikzcd}
\Sigma^* \arrow[r, "h"] \arrow[dr, "\eta_L"'] & h(\Sigma)^* \arrow[r, hook] \arrow[d, dashed, "h_L"'] & M \\
& \mathsf{Syn}(L) &
\end{tikzcd}$$

η_L is a surjective map and thus h_L is also surjective. Thus if $y \in \mathsf{Syn}(L)$, $y = h_L(x)$. But $x^N = x^{N+1}$. Thus $h_L(x^N) = h_L(x^{N+1})$, giving $h_L(x)^N = h_L(x)^{N+1}$. ■

This shows that checking whether a language is aperiodic is decidable. We just need to check whether its syntactic monoid is aperiodic.

Exercise: Show that the syntactic monoid of the language $\{a^{2n} \mid n \geq 0\}$ is not aperiodic.

One of the corner stones of the study of regular languages is the result of M.P.Schutzenberger showing that languages recognized by aperiodic monoids are exactly the class of languages that are described by regular expressions using the operators + (union) . (concatenation) and - (complementation). McNaughton showed that the the class of languages that can be expressed using the first-order logic of words also coincides with this class. We shall establish these two results in this and the following lecture.

2 From Star-free languages to Aperiodic Monoids

We shall show that every regular language that can be described by some star-free regular expression is also recognized by an aperiodic monoid. This can be seen as follows:

1. It is trivial to specify aperiodic monoids recognizing the languages $\{a\}$, $\{\epsilon\}$ and $\emptyset$.
2. If L is accepted via the (aperiodic) monoid M, then so is $\overline{L}$.
3. If L_1 and L_2 are recognised using the morphisms h_1 and h_2 to the monoid M_1 and M_2 respectively so that $L_1 = h^{-1}(X_1)$ and $L_2 = h^{-1}(X_2)$ then $L_1 \cup L_2$ is recognized by the monoid $M_1 \times M_2$ via the morphism h with $h(a) = (h_1(a), h_2(a))$ as the pre-image of the set $X_1 \times M_2 \cup M_1 \times X_2$.
4. Finally, suppose L_1 and L_2 are aperiodic so that $x^N = x^{N+1}$ for each $x \in \mathsf{Syn}(L_1) \cup \mathsf{Syn}(L_2)$. Then, we claim that for any word $w \in \Sigma^*$, $uw^{2N}v \in L_1.L_2$ if and only if $uw^{2N+1}v \in L_1.L_2$ (Exercise). Thus, $[w^{2N}]_{L_1.L_2} = [w^{2N+1}]_{L_1.L_2}$. Therefore, $\mathsf{Syn}(L_1.L_2)$ is aperiodic.

Exercise: Show how to directly construct a monoid recognizing $L_1.L_2$ from monoids accepting L_1 and L_2.

Exercise: Show that every language that can be described via star-free regular expressions can also be expressed in the first-order logic of words.

Notes: Most of the results mentioned here are from [2] and [1].

References

[1] Nick Pippenger: *Theories of Computability*, Cambridge University Press, 1997.

[2] Howard Straubing: *Finite Automata, Formal Logic and Circuit Complexity*, Birkhäuser, 1994.