Lecture 6e: Ordered Monoids and languages definable in Σ_1 and Σ_2

Recall that we had characterized the class of languages definable in $FO^1(<)$, i.e., first-order logic sentences that use only one variable, and demonstrated an algorithm to check if a given regular language lies in this class. A different way to parametrize formulas of $FO(<)$ is via the number of alternations of quantifiers.

A formula is said to be in *prenex form* if it consists of a sequence of quantifiers followed by a quantifier-free formula (i.e. all the quantifiers occur before the other logical operators). A well-known fact in logic is that every FO formula can be rewritten into an equivalent formula in prenex form.

For eg. $\exists x.\exists y.((x < y) \wedge (\exists x.(y < x) \wedge (\exists x.(x < y))))$ can be equivalently expressed as $\exists x_1.\exists x_2.((x < x_2) \wedge (\exists x_3.(x_2 < x_3) \wedge (\exists x_4.(x_3 < x_4))))$ and then as $\exists x_1.\exists x_2.\exists x_3.\exists x_4.(x_1 < x_2) \wedge (x_2 < x_3) \wedge (x_3 < x_4)$ In general, by using a new variable for every quantifier and then *moving* all the quantifiers to the beginning of the formula we obtain such an equivalent prenex formula. We omit the details here.

Any formula in prenex form consists of alternating blocks of existential and universal quantifiers. The formula written above has a single block of existential quantifiers while the formula $\forall x.\forall y.\exists z.(x = y) \vee ((x < z) \wedge (z < y)) \vee ((y < z) \wedge (z < x))$ has two blocks, a universal block followed by an existential block.

Let $\Sigma_0 = \Pi_0$ be the set of quantifier free formula. The classes Σ_i and Π_i are defined inductively as follows: The class Σ_{i+1} consists of formulas obtained from formulas in Π_i by adding a block of existential quantifiers while Π_{i+1} consists of formulas obtained from formulas in Σ_i by adding a block of universal quantifiers.

The two example formulas listed above are in Σ_1 and Π_2 respectively. In general, Σ_i consists of formulas with i blocks of quantifiers beginning with an existential block while Π_i consists of formulas with i blocks beginning with a universal block. Quite clearly, if φ is a formula in Σ_i then $\neg\varphi$ can be transformed into an equivalent formula in Π_i by pushing the negation over the quantifiers. A similar result holds for formulas in Π_i.

We shall say that a language L is in Σ_i (respectively Π_i) if it is definable by a sentence in Σ_i (respectively Π_i). We will also use Σ_i to refer to the class of languages in Σ_i and soon.

Thus $L \in \Pi_i$ iff $\overline{L} \in \Sigma_i$. We note that $\Pi_i \subseteq \Sigma_{i+1}$ since we may always add an existential quantifier on an irrelevant variable at the beginning of the formula without altering its meaning. Similarly $\Sigma_i \subseteq \Pi_{i+1}$.

Unlike the case of the variable hierarchy, where we have $FO^3(<) = FO(<)$, the quantifier alternation hierarchy is known to be infinite, providing a finer stratification of first-order definable languages. But, as of now, our understanding of this hierarchy is limited. For instance, decidability of membership (of regular languages) in the levels Σ_1 and Σ_2 (and therefore for Π_1 and Π_2) is known, but the problem remains open for higher levels. We now describe monoid based characterizations for Σ_1 and Σ_2 which lead to a solution to their membership problems. The presentation here follows that in [3].

We hit a road block immediately! Observe that the class Σ_i is unlikely to be closed under complementation, unless $\Pi_i = \Sigma_i$, and it turns out it is not. However, the set of languages definable using a monoid (and hence any class of monoids) is closed under complementation and so there is no hope of obtaining Σ_i as the class of languages definable using a class of monoids.

J.E.Pin proposed a elegant solution to this problem using what are called *ordered monoids* which we study now. Recall that monoids were obtained naturally from the syntactic congruence $\Sigma^*/\equiv_L$ defined by any language L where $x \equiv_L y$ holds if and only if for all $u, v \in \Sigma^*$, $uxv \in L \iff uyv \in L$.

What if we refined this two sided requirement into just a one sided requirement? We say $x \leqslant_L y$ iff for all $u, v \in \Sigma^*$, $uyv \in L \implies uxv \in L$. This relation is a *pre-order*, i.e., it is reflexive and transitive but not anti-symmetric. As a matter of fact, its symmetric core (i.e) $\{(x, y) \mid x \leqslant_L y \;\wedge\; y \leqslant_L x\}$ is exactly the $\equiv_L$ relation. As with any pre-order, it induces a partial-order on the equivalence classes of its symmetric core. In what follows we write $[x]_L$ to denote the equivalence class of x w.r.t. $\equiv_L$.

Proposition 1 *The pair* $(\{[x]_L \mid x \in \Sigma^*\}, \leqslant_L)$ *is a partial order (where we lift the definition of* $\leqslant_L$ *by writing* $[x]_L \leqslant_L [u]_L$ *whenever* $x \leqslant_L y$*.)*

We observe that this relation is consistent with the multiplication in the monoid. That is, if $[x]_L \leqslant_L [y]_L$ then, for any u, v, $[u]_L[x]_L[v]_L \leqslant_L [v]_L$ (we leave this as an easy exercise to the reader). Further, we observe that for any x,y if $[x]_L \leqslant_L [y]_L$ and $[y]_L \subseteq L$ then $[x]_L \subseteq L$ (simply use $u = v = \epsilon$ in the definition of $\leqslant_L$).

Recall that the language L is recognized by the syntactic morphism η_L from $(\Sigma^*, ., \epsilon)$ to $(\{[x]_L \mid x \in \Sigma^*\}, ., [\epsilon]_L)$ and $L = \eta_L^{-1}(X)$ where $X = \{[x]_L \mid [x]_L \subseteq L\}$. The import of the previous paragraph is that this set X is a downward closed set w.r.t the order $\leqslant_L$. In any partial order a downward closed subset is called an *ideal*.

To summarize, we have equipped the syntactic monoid of L with a partial order, which is consistent with its multiplication and further observed that the subset recognizing L is an ideal. Most importantly, even though $\overline{L}$ is recognised by the same syntactic monoid its recognizing set is not an ideal (as a matter of fact it is an upward closed set, as the complement of any ideal is).

We shall refer to the structure $(\{[x]_L \mid x \in \Sigma^*\}, ., [\epsilon]_L, \leqslant_L)$ as the *ordered syntactic monoid* of L. We shall write $\mathsf{oSyn}(L)$ to denote this monoid. The ordered syntactic monoid of L and $\overline{L}$ are different as we would like them to be (as an aside note that the ordered syntactic monoid of $\overline{L}$ is $(\{[x]_L \mid x \in \Sigma^*\}, ., [\epsilon]_L, \leqslant_L^R)$). With this reasoning in mind we define ordered monoids and recognition via ordered monoids as follows.

Definition 2 *An* ordered monoid *is a tuple* $(M, ., 1, \leqslant)$ *where* $(M, ., 1)$ *is a monoid and* $\leqslant$ *is a partial order on* M *that is consistent, that is, if* $x \leqslant y$ *then* $uxv \leqslant uyv$ *for all* $u, v \in M$.

A morphism h *from an ordered monoid* $(M, ., 1_M, \leqslant_M)$ *to* $(N, ., 1_N, \leqslant_N)$ *is a monoid morphism from the monoid* $(M, ., 1_M)$ *to* $(N, ., 1_N)$ *which further satisfies* $x \leqslant_M y$ *implies* $h(x) \leqslant_N h(y)$ *(i.e. it is also monotone w.r.t. the orderings).*

We observe that any monoid can be turned into an ordered monoid by using $=$ (i.e. the identity relation) as the order. In particular, $(\Sigma^*, ., \epsilon, =)$, is an ordered monoid and is called the *free ordered monoid on* Σ. Another interesting class of examples of ordered monoids are the ordered syntactic monoids of languages as described above. The map $\eta_L :: x \mapsto [x]_L$ is an ordered monoid morphism from $(\Sigma^*, ., \epsilon, =)$ to $\mathsf{oSyn}(L)$.

Definition 3 *L is recognized by an ordered monoid morphism $h : (\Sigma^*, ., \epsilon, =) \longrightarrow (M, ., 1, \leqslant)$ if there is an ideal $X \subseteq M$ such that $L = h^{-1}(X)$.*

The following is just a consequence of the above discussion:

Proposition 4 *A language L over an alphabet Σ is regular if and only if it is recognized by a finite ordered monoid.*

Thus, recognition by ordered monoids provides an algebraic view of regular languages that allows us to examine classes that are not closed under complementation. The syntactic ordered monoid plays the role of the syntactic monoid in this setting:

Proposition 5 *Let $(M, ., 1, \leqslant)$ be a finite ordered monoid that recognizes a regular language L via the morphism h. Then, there is a surjective morphism h_L from the ordered submonoid $(h(\Sigma^*), ., 1, \leqslant)$ to the ordered syntactic monoid of L such that such that $\eta_L = h_L.h$.*

We leave the proof of this proposition as an exercise to the reader.

We still have to show how to compute the ordereed syntactic monoid of a regular language. The underlying monoid of $\mathsf{oSyn}(L)$ is $\mathsf{Syn}(L)$ which is computable. So, we just have to compute $\leqslant_L$. This is a consequence of the following lemma.

Lemma 6 *Let $X = \{[x]_L \mid x \in L\}$. Then, for any $x, y \in \Sigma^*$, $[x]_L \leqslant_L [y]_L$ iff for all $[u]_L, [v]_L \in \mathsf{Syn}(L)$, $[u]_L[x]_L[v]_L \in X$ whenever $[u]_L[y]_L[v]_L \in X$. Thus, the ordered syntactic monoid of any regular language is computable.*

Proof: Suppose $[x]_L \leqslant_L [y]_L$. By the definition of $\leqslant_L$, for all $u, v \in \Sigma^*$, $uxv \in L$ whenever $uyv \in L$. But, his means that for all $u, v \in \Sigma^*$, $[uxv]_L \in X$ whenever $[uyv]_L \in X$.

For the converse, suppose $uyv \in L$. Therefore $[u]_L[y]_L[v]_L \in X$ and by the hypothesis this means $[u]_L[x]_L[v]_L \in X$. This means $uxv \in \eta^{-1}(X)$ and thus $uxv \in L$. ■

We are now in a position to characterize the class of languages in Σ_1 and Σ_2.

The class Σ_1

Consider any formula $\varphi = \exists x_1.\exists x_2.\ldots\exists x_m.\varphi'$ in Σ_1 and a word $w = a_1 \ldots a_n$ such that $w \models \varphi$. Then, there is a valuation σ such that $w = w_1 a_{i_1} w_2 a_{i_2} \ldots w_r a_{i_r} w_{r+1}$, $r \leqslant m$ and the positions in the image of σ are in the set $\{i_1, \ldots i_r\}$.

Now, observe that the sequence $a_{i_1} a_{i_2} \ldots a_{i_r}$ also satisfies the formula φ using the valuation that sends x_i to j if $\sigma(x_i) = i_j$. Moreover, any word in $\Sigma^* a_{i_1} \Sigma^* a_{i_2} \ldots a_r \Sigma^*$ also satisfies φ. Thus, we have

Proposition 7 *Let φ be a formula in Σ_1 with m quantifiers. Then*

$$L(\varphi) \;=\; \bigcup\{\Sigma^* a_1 \Sigma^* a_2 \ldots a_r \Sigma^* \mid r \leqslant m, a_1 a_2 \ldots a_r \models \varphi\}$$

A language of the form $\Sigma^* a_1 \Sigma^* a_2 \ldots a_r \Sigma^*$ is called a *simple monomial* and finite unions of simple monomials is called a *simple polynomial.* Thus every Σ_1 language is a simple polynomial.

Proposition 8 *A language is in Σ_1 iff it is a simple polynomial.*

Proof: The remark above proves one direction. For the other direction note that the formula $\exists x_1.\exists x_2.\ldots\exists x_r.(x_1 < x_2) \wedge \ldots \wedge (x_{r-1} < x_r) \wedge a_1(x_1) \wedge \ldots \wedge a_r(x_r)$ describes the language $\Sigma^* a_1 \Sigma^* \ldots a_r \Sigma^*$. ■

A language is *upward closed* if $uav \in L$ whenever $uv \in L$. Clearly every simple monomial is upward closed and upward closed languages are closed under unions. Thus, as a corollary to the above proposition we also have

Corollary 9 *Every language in Σ_1 is upward closed.*

If a regular language L, with a FA on say n states, is upward closed then it may be expressed as the following simple polynomial

$$L \;=\; \{\Sigma^* a_1 \Sigma^* a_2 \ldots a_r \Sigma^* \mid r \leqslant n, a_1 a_2 \ldots a_r \in L\}$$

and hence it is also in Σ_1. Thus we have

Proposition 10 *A language is in Σ_1 iff it is upward closed.*

This also leads to an algebraic characterization of Σ_1 as follows:

Theorem 11 *A language L is in Σ_1 iff it is recognized by a morphism h to an ordered monoid $(M, ., 1)$ in which $1 \leqslant s$ holds for all $s \in M$.*

Proof: Note that if $1 \leqslant s$ then $h(uav) = h(u)h(a)h(v) \leqslant h(u)1h(v) = h(uv)$. Since the language is defined by an ideal it follows that if $uv \in L$ then $uav \in L$ and thus L is upward closed and hence in Σ_1.

For the converse, since the language is upward closed we have $uv \in L \implies uwv \in L$ and so $[u]_L.[\epsilon]_L.[v]_L \leqslant [u]_L.[w]_L.[v]_L$ for all $u, v, w \in \Sigma^*$ and thus $[\epsilon]_L \leqslant [w]_L$ for all $w \in \Sigma^*$. Thus $\mathsf{oSyn}(L)$ satisfies $1 \leqslant s$. ■

Thus one may check whether a regular language belongs to Σ_1 by computing its ordered syntactic monoid and verifying that it satisfies $1 \leqslant s$. It turns out that it is easy to check if the language of a finite automaton is upward closed (Exercise) and so the algebraic characterization is not essential. However, it is the technique that is being enunciated here and this works for Σ_2 as well (while the ad hoc automata based algorithm does not.)

Theorem 12 *Membership is decidable for Σ_1.*

Exercise: Prove that the class of finite ordered monoids satisfying $1 \leqslant s$ is closed under products and division.

The class Σ_2

A *monomial* is a language of the form $A_0^* a_1 A_1^* a_2 \ldots a_r A_r^*$ where $a_i \in \Sigma$ and $A_i \subseteq \Sigma$. A *polynomial* is simply a finite union of monomials. We shall show that the class Σ_2 is precisely the class of polynomials and obtain an effective algebraic characterization for this class.

One direction is easy. A monomial $A_0^* a_1 A_1^* a_2 \ldots a_r A_r^*$, can be expressed by the formula

$$\begin{aligned} \exists x_1 \exists x_2 \ldots \exists x_r . \forall y.\ & \bigwedge_{1 \leqslant i \leqslant r} a_i(x_i) \ \wedge \bigwedge_{1 \leqslant i \leqslant r-1} (x_i < x_{i+1}) \\ & \wedge\ (y < x_1) \implies A_0(y) \ \wedge\ (y > x_r) \implies A_r(y) \ \wedge \\ & \bigwedge_{1 \leqslant i < r} ((y > x_i) \wedge (y < x_{i+1}) \implies A_i(y) \end{aligned}$$

Proposition 13 *Every polynomial is in Σ_2.*

The other direction is non-trivial and uses the factorization forest theorem in its proof. We first establish some simple properties of polynomials.

Lemma 14 *The class of polynomials over Σ is the least class of languages closed under union and concatenation which contains all the finite languages as well as all languages of the form A^* for $A \subseteq \Sigma$.*

Proof: Clearly all polynomials are members of the least class containing finite languages and languages of the form A^*. For the converse, note that all finite languages are polynomials and so are languages of the form A^*. Thus, it suffices to prove that the class of polynomials is closed under union and concatenation. Closure under union follows from the definition of polynomials. Given two monomials $A_0^* a_1 \ldots a_r A_r^*$ and $B_0 b_1 \ldots b_m B_r^*$ their concatenation is the following polynomial:

$$A_0^* a_1 \ldots a_r A_r^* b_1 B_1^* \ldots b_m B_m^* \ \cup \ \bigcup_{b \in B_0} A_0^* a_1 \ldots a_r A_r^* b B_0^* b_1 B_1^* \ldots b_m B_m^*$$

Closure of polynomials under concatenation follows from the fact that $(L_1 \cup L_1').(L_2 \cup L_2') = L_1.L_2 \ \cup \ L_1.L_2' \ \cup \ L_1'.L_2 \ \cup \ L_1'.L_2'$. ∎

In what follows, we write $\alpha(w)$ for the set of letters that appear in the word w. Next, we show that the ordered syntactic monoids of languages in Σ_2 satisfy an useful property.

Lemma 15 *Let L be a language in Σ_2 and let m be the total number of quantifiers in a Σ_2 formula φ that defines L. For any word $u \in \Sigma^*$ and $v \in \alpha(u)^*$ we have $xu^N vu^N y \in L$ whenever $xu^N y \in L$ for all $N > m^2 + m$, $x, y \in \Sigma^*$. That is, $[u^N]_L [v]_L [u^N]_L \leqslant [u^N]_L$ in* $\mathsf{oSyn}(L)$ *for all $N > m^2 + m$.*

Proof: Let $\varphi = \exists x_1 x_2 \dots x_k \forall y_1 y_2 \dots y_l . \varphi'$ and let $xu^N y \models \varphi$. From a valuation σ assigning positions within $\rho = xu^N y$ for $x_1 \dots x_k$ which satisfies $\forall y_1 y_2 \dots y_l . \varphi'$ we manufacture a valuation σ' assigning positions in $\rho' = xu^N vu^N y$ satisfying $\forall y_1 y_2 \dots y_l . \varphi'$.

By our choice of N, we may write ρ as $u^{k_1} u^m u^{k_2}$ such that none of the positions within the middle u^m is in the range of σ. We now consider a break up of ρ' as $xu^{k_1} u^{m+k_2} vu^{m+k_1} u_2^k y$. The map σ' assigns all the x_is which are assigned a position within the prefix xu^{k_1} by rho the same position in the corresponding prefix of ρ'. Similarly any x_j assigned a position in the suffix $u^{k_2} y$ by σ is assigned the corresponding position within the same suffix of ρ' by σ'. Thus, all atomic formulas involving only $x_1, x_2 \dots x_k$ are either satisfied by both (ρ, σ) and (ρ', σ') or neither.

Suppose σ'_f is any extension of σ' which also assigns positions in ρ' for the variables $y_1, y_2, \dots y_l$. We show that there is an extension of σ_f of σ assigning positions in ρ for the variables $y_1, y_2, \dots y_l$ such that any atomic formula involving $x_1, x_2 \dots x_k, y_1, \dots y_l$ is either satisfied by both (ρ, σ_f) and (ρ', σ'_f) or neither.

If a variable y_j is assigned a position within the prefix xu^{k_1} or the suffix $u^{k_2} y$ by σ'_f then σ_f assigns the corresponding position in ρ for y_j. Let Y be the set of such variables. Then, it is easy check that both (ρ, σ_f) and (ρ, σ'_f) satisfy the same set of atomic formulas w.r.t. the variables $\{x_1, \dots, x_k\} \cup Y$.

Let $y_{j_1}, y_{j_2} \dots y_{j_r}$ be the variables assigned positions in the infix $u^{m+k_2} vu^{m+k_1}$ by σ'_f. Further suppose that the positions used are $p_1 < p_2 < \dots < p_{r'}$ with $r' \leqslant r$ (since multiple variables may be assigned the same position).

We pick one position, say q_i, from each of the first r' copies of t in the infix t^m of ρ such that the letter at position q_i in ρ is the letter at position p_i in ρ'. This is possible since $\alpha(uv) \subseteq \alpha(u)$. The map σ_f assigns position q_i to any variable that was assigned the position p_i by the valuation σ'_f.

Then (ρ, σ_f) satisfies any atomic formula over $\{x_1, \dots x_k, y_1, \dots, y_l\}$ iff it is also satisfied by (ρ', σ'_f). Hence (ρ, σ_f) satisfies φ' iff (ρ', σ'_f) satisfies φ'.

Thus, $\rho, \sigma \models \forall y_1 y_2 \dots y_k . \varphi'$ implies that $\rho', \sigma' \models \forall y_1 y_2 \dots y_k . \varphi'$, so that $\rho' \models \varphi$ whenever $\rho \models \varphi$ as required. ■

Let $(M, ., 1, \leqslant)$ be an ordered monoid. For any idempotent $e \in M$, we write S_e for the semigroup generated by the elements $\{s \mid e \leqslant_J s\}$.

Proposition 16 *If L is language in Σ_2 then its ordered syntactic monoid satisfies $ese \leqslant_L e$, for every idempotent e and every $s \in S_e$.*

Proof: Let $e \leqslant_J s$ in $\mathsf{oSyn}(L)$. Therefore $s = [v]_L$ for some v and since $e \leqslant_J [v]_L$ we may assume $e = [xvy]_L$ for some $x, y \in \Sigma^*$. In particular, if $e \leqslant_J s$ then we may assume that $e = [u]_L$ and $s = [v]_L$ and $\alpha(v) \subseteq \alpha(u)$.

Now, if $s \in S_e$ the $s = s_1 s_2 \dots s_k$, $e \leqslant_J s_i, 1 \leqslant i \leqslant k$. Let $s_i = [v_i]_L$. Therefore $e = [u_i]_L$ such that $\alpha(v_i) \subseteq \alpha(u_i)$. Let $u = u_1 u_2 \dots u_k$ and $v = v_1 v_2 \dots v_k$. Then $s = [v_1 \dots v_k]_L$. Further $[u]_L = e^k = e$. Thus, if $s \in S_e$ then we may assume $s = [v]_L$ and $e = [u]_L$ and $\alpha(v) \subseteq \alpha(u)$.

Then, by Lemma 15, $[u]_L^N[v]_L[u]_L^N \leqslant_L [u]_L^N$. But e is an idempotent and so we get $[u]_L[v]_L[u]_L \leqslant_L [u]_L$ as required. ■

Next we show that languages recognized by monoids satisfying $ese \leqslant e$ for $s \in S_e$ are always polynomials. This is a nontrivial fact and uses the factorization forest theorem in its proof.

Lemma 17 *Let $(M, ., 1, \leqslant)$ be an ordered monoid satisfying $ese \leqslant e$ whenever $s \in S_e$ and e is an idempotent. Let X be any ideal in M. Let $h : (\Sigma^*, ., \epsilon, =) \longrightarrow (M, ., 1, \leqslant)$ be a morphism. Then $h^{-1}(X)$ is a polynomial.*

Proof: The proof essentially follows the argument used to obtain well-typed regular expressions for $h^{-1}(s)$ for each s. Let

$$L_s^i = \{w \mid h(w) = s, w \text{ has a factorization tree of height} \leqslant i\}$$

. We construct a polynomial P_s^i for each $0 \leqslant i \leqslant 5|M|$ and $s \in M$ so that $L_s^i \subseteq P_s^i$ and $h(P_s^i) \subseteq s\downarrow$, where $s\downarrow$ is the ideal $\{t | t \leqslant s\}$.

We let

$$P_s^0 = \sum\{a \mid a \in \Sigma \cup \{\epsilon\}, h(a) = s\}$$

which is clearly a polynomial satisfying the required properties.

If s is not an idempotent then we set

$$P_s^{i+1} = P_s^i + \sum\{P_u^i.P_v^i \mid u.v = s\}$$

which is again a polynomial since polynomials are also closed under concatenation (and union). Also, $h(xy)$ with $x \in P_u^i$ and $y \in P_v^i$ is $h(x).h(y) \leqslant u.v \leqslant s$ as required. If $w \in L_s^{i+1} \backslash L_s^i$, since s is not an idempotent we must have u, v and $w = w_1.w_2$ such that $w_1 \in L_u^i$, $w_2 \in L_v^i$ and $s = uv$. Therefore $w \in P_{i+1}^s$.

Finally, if e is an idempotent then we set

$$P_e^{i+1} = P_e^i + \sum\{P_u^i.P_v^i \mid u.v = e\} + (P_e^i)A^*(P_e^i)$$

where A is the set of letters that appear in the words of the language $h^{-1}(e)$. Observe that $e \leqslant_J h(a)$ for each $a \in A$.

That P_e^{i+1} is a polynomial follows from closure under concatenation and union. If $x \in P_u^i$ and $y \in P_v^i$ with $uv = e$ then $h(xy) \leqslant e$ as above. If $x, y \in P_e^i$ and $v \in A^*$ then $h(xvy) = h(x)h(v)h(y) = eh(v)e$. But e is an idempotent and $h(v) \in S_e$ and therefore $eh(v)e \leqslant e$ as required. Finally, let $w \in L_e^{i+1} \backslash L_e^i$. We need to show that $w \in P_e^{i+1}$. If the root of the factorization tree for w has two children then the proof proceeds as in the previous case. If the root has at least 3 children then $w = w_1 w_2 \dots w_k$, $k \geqslant 3$ such that $w_i \in L_e^i$. Thus, by induction hypothesis $w_1, w_k \in P_e^i$. Further, by definition $\alpha(w_i) \subseteq A$ for each $1 \leqslant i \leqslant k$ and thus $w_2 w_3 \dots w_{k-1} \in A^*$. Thus $w \in P_e^{i+1}$.

The proof of the Lemma is now easy. Let $P_s = P_s^{5|M|}$. Clearly P_s is a polynomial such that $h^{-1}(s) \subseteq P_s$ and $h(P_s) \subseteq s\downarrow$. The polynomial $\bigcup_{s \in X} P_s$ is the language $h^{-1}(X)$ since $s\downarrow \subseteq X$ for each $s \in X$. ■

As a consequence of Propostions 13,16 and Lemma 17 we have the following theorem:

Theorem 18 *A language L is in Σ_2 iff it is a polynomial iff* $\mathsf{oSyn}(L)$ *satisfies $ese \leqslant_L e$ for every idempotent e and $s \in S_e$. Thus, the membership problem for Σ_2 is decidable.*

The proof of the above theorem also showed that any ordered monoid satisfying $ese \leqslant s$ for all $s \in S_e$ recognizes only languages in Σ_2.

Exercise: Show that the class of ordered monoids satisfying $ese \leqslant e$ for all idempotents e and $s \in S_e$ is closed under products and division.

References

[1] M. Bojanczyk: "Factorization Forests", *Proceedings of DLT 2009*, Springer LNCS 5583 (2009) 1-17.

[2] T. Colcombet: "Green's Relations and their Use in Automata Theory", *Proceedings of LATA 2011*, Spring LNCS 6638 (2011) 1-21.

[3] V Diekart, P Gastin and M Kufleitner: "A Survey on Small Fragments of First-Order Logic over Finite Words", International Journal of the Foundations of Computer Science **19(3)**, 2008.

[4] J.E.Pin: *Mathematical Foundations of Automata Theory*, MPRI Lecture Notes.