Uschi Heuter

First-order properties of trees, star-free expressions, and aperiodicity

FIRST-ORDER PROPERTIES OF TREES,
STAR-FREE EXPRESSIONS, AND APERIODICITY (*)

by Uschi HEUTER ([1])

Abstract. – *We characterize the first-order definable sets of finite trees in terms of certain star-free tree expressions and show that for regular sets of finite trees, first-order definability is a more restrictive notion than aperiodicity. These two theorems show how far the results of McNaughton and Schützenberger on star-free sets of words (stating the equivalence between first-order definability, star-freeness, and aperiodicity) can be transferred to the context of trees. Both results of the paper rely on the method of the Ehrenfeucht-Fraissé-game.*

Résumé. – *Nous caractérisons les ensembles d'arbres finis définissables au premier ordre en termes de certaines expressions sans-étoile et nous montrons que pour les ensembles réguliers, le fait d'être définissable au premier ordre est une notion plus restrictive que l'apériodicité. Ces deux théorèmes montrent jusqu'à quel point on peut étendre aux arbres les résultats de McNaughton et Schützenberger sur les ensembles de mots sans-étoile (qui établissent l'équivalence entre « être définissable au premier ordre », apériodicité et sans-étoile). Les deux résultats reposent sur la méthode des jeux de Ehrenfeucht-Fraissé.*

1. INTRODUCTION

McNaughton and Schützenberger showed in [8] and [11] that first-order definability, star-freeness and aperiodicity are equivalent notions for regular sets of words. Since the usual characterizations of regular word sets (in terms of regular expressions, monadic second-order logic, finite automata) have been carried over to sets of trees ([14, 2]), the question arises whether this transfer is also possible for the results of McNaughton and Schützenberger on star-free languages. In [13] first-order logic over trees, usual regular tree language expressions restricted to the star-free case, as well as aperiodicity

---

over trees were considered. It was observed there that these star-free expressions are strictly more expressive than first-order logic and also yield nonaperiodic sets. Concerning the relation between first-order definability and aperiodicity it is shown as in the corresponding proof for star-free word languages (see e. g. [9]) that first order definable sets are aperiodic; however, it remained open whether the converse also holds.

The present paper offers on one hand a restricted notion of star-freeness which exactly captures the strength of first-order logic over trees, and on the other hand it shows that there are aperiodic languages which are not first-order definable. This shows that the equivalence of the notions "first-order", "star-free" and "aperiodic" for regular word languages completely fails in the corresponding case of tree languages and that hence the analogy between regular sets of words and regular sets of trees does not extend to the important subclass of star-free sets.

Both main results of the paper rely on the method of the Ehrenfeucht-Fraissé-game over trees. In the characterization of first-order logic by certain star-free expressions the games are used to justify a decomposition of first-order formulas into conditions which speak only about certain parts of trees. In the construction of a non first-order definable but aperiodic set the games are applied to show indistinguishability of trees by first-order formulas.

The paper is structured as follows: After technical preliminaries (Section 2) we will formulate and prove in Section 3 the characterization of first-order logic over trees in terms of regular expressions. For this purpose we define "special trees", i. e. trees over an alphabet Σ which can be labeled at the frontier with extra symbols of a "concatenation alphabet" D. Each such symbol may occur at most once at the frontier. So the resulting concatenation of trees is a restriction of the usual one defined in [14, 2]; it corresponds to speaking about single nodes in first-order logic (as opposed to sets of nodes in monadic second-order logic used in [14, 2]). Our result will state that a tree language is first-order definable if and only if it is built up from finite sets of special trees using the operations union, complement and concatenation, all restricted to the class of special trees. The main part of the proof of this characterization will be a decomposition of first-order formulas. The proof of this decomposition lemma uses the Ehrenfeucht-Fraissé-game (Section 4). In Section 5 we will present an aperiodic language T which will be proved to be not first-order definable. To show this, we define a sequence of trees t_i, s_i with $t_i \in T$, $s_i \notin T$ such that for all i the trees t_i and s_i are indistinguishable by first-order formulas of quantifier-depth i. The indistinguishability will be shown again using the Ehrenfeucht-Fraissé-game.

A preliminary version of the paper has appeared in [5]. Some tedious but easy proofs left out here can be found in [6].

I would like to thank W. Thomas for introducing me to the subject and his advice and steady encouragement. I'm also obliged to Th. Hafer for helpful discussions.

2. TECHNICAL PRELIMINARIES

Let $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \ldots \cup \Sigma_n$ be a ranked alphabet where each Σ_i is a finite set of i-ary symbols (the sets Σ_i are not necessarily disjoint). Let furthermore D be an alphabet with 0-ary symbols, the "concatenation alphabet". A Σ, D-tree is a term built up from the symbols of $\Sigma \cup D$ in the usual way; by $T_{\Sigma, D}$ we denote the set of all Σ, D-trees. Instead of $T_{\Sigma, \varnothing}$ we just write T_Σ.

Let T, $T' \subset T_{\Sigma, D}$ and $d \in D$. Then $T \cdot^d T'$ is the set of all trees of $T_{\Sigma, D}$ which result from some $t \in T$ by substituting each occurrence of d in t by a tree of T'. The star-operation is defined by: $T^{*d} = \bigcup \{ T^{d, i} \mid i \geq 0 \}$, where $T^{d, 0} = \{d\}$ and $T^{d, i+1} = T^{d, i} \cdot^d (T \cup \{d\})$.

A tree language $T \subset T_\Sigma$ is called *regular* if there is a finite alphabet D, such that T can be constructed from finite subsets of $T_{\Sigma, D}$ by using union, the concatenation operations $\cdot^d$ and the star operations $*^d$ with $d \in D$.

The set $RE(\Sigma, D)$ of generalized regular Σ, D-expressions (*i.e.* with complement operation) is inductively defined by:

$$\varnothing \in RE(\Sigma, D), \qquad T_{\Sigma, D} \subset RE(\Sigma, D)$$

and if β_1, $\beta_2 \in RE(\Sigma, D)$ and $d \in D$ then also

$$(\beta_1 \vee \beta_2) \in RE(\Sigma, D), \qquad (\beta_1 \cdot^d \beta_2) \in RE(\Sigma, D), \qquad (\sim \beta_1) \in RE(\Sigma, D)$$

and

$$(\beta_1^{*d}) \in RE(\Sigma, D).$$

The tree language $T(\beta) \subset T_{\Sigma, D}$ defined by a regular expression $\beta \in RE(\Sigma, D)$ is defined according to the explained meaning of $\cdot^d$ and $*^d$; $\vee$ stands for union, and $\sim$ for the complement taken w.r.t. $T_{\Sigma, D}$.

A tree language $T \subset T_\Sigma$ is called *star-free* if there is a finite alphabet D and a regular expression $\beta \in RE(\Sigma, D)$ without star-operator such that $T(\beta) = T$. The set of all star-free expressions over $\Sigma \cup D$ is denoted by $SF(\Sigma, D)$.

To define aperiodic tree languages we refer to a restricted concatenation of trees, which takes place at only one leaf. We call a tree *special* over $\Sigma \cup \{c\}$ (where c is a 0-ary symbol not in Σ), if it has at most one occurrence of c. The set of all special trees over $\Sigma \cup \{c\}$ is denoted by S_Σ. If s, $s' \in S_\Sigma$, we simply write $s.s'$ instead of $s \cdot^c s'$. A tree language $T \subset T_\Sigma$ is called *aperiodic* (or *noncounting*) if

$$\exists\, n \geqq 0, \quad \forall s_0, s \in S_\Sigma, \quad \forall t \in T_\Sigma : \quad s_0.s^n.t \in T \quad \Leftrightarrow \quad s_0.s^{n+1}.t \in T.$$

This notion coincides with the notion of an aperiodic word language when words are considered as unary trees (*cf.* [8]). As for regular word languages, the property "aperiodic" is decidable also for regular tree languages. To show this, one has to verify that $(S_\Sigma, .)$ is a monoid (with identity c) and that each regular tree language $T \subset T_\Sigma$ induces a finite and effectively constructible quotient monoid $M(T)$ of $(S_\Sigma, .)$, defined by the equivalence relation

$$s_1 \equiv_T s_2 \quad \text{iff} \quad \forall s \in S_\Sigma, \quad \forall t \in T_\Sigma : \quad s.s_1.t \in T \quad \Leftrightarrow \quad s.s_2.t \in T.$$

The following result is obtained in exactly the same way as the corresponding theorem for word languages ([8]):

PROPOSITION 2.1 [13]: (*a*) *A regular tree language* $T \subset T_\Sigma$ *is aperiodic iff* $M(T)$ *contains only trivial groups.*

(*b*) *Aperiodicity is decidable for regular tree languages.* ●

We now turn to the description of tree languages in terms of mathematical logic. If Σ contains at most n-ary symbols, a tree $t \in T_\Sigma$ (resp. $t \in T_{\Sigma, D}$) is considered as a function $t: \mathrm{dom}(t) \to \Sigma$ [resp. $t: \mathrm{dom}(t) \to \Sigma \cup D$] where $\mathrm{dom}(t)$ is the set of nodes of t, represented by a finite prefix closed set of words over $\{1, \ldots, n\}^*$ with the following property:

(∗)   if $xi \in \mathrm{dom}(t)$ then also $xj \in \mathrm{dom}(t)$ for all $j < i$.

Denote the (partial) prefix ordering by $<$. For the treatment of subtrees it is convenient to admit sets $\mathrm{dom}(t)$ of the form $k \cdot P$ where $k \in \{1, \ldots, n\}^*$ and $P \subset \{1, \ldots, n\}^*$ is a finite prefix-closed set again with property (∗). In this case the node k is called *root* of t. The *frontier* $fr(t)$ is defined by $fr(t) := \{x \in \mathrm{dom}(t) \mid xi \notin \mathrm{dom}(t) \text{ for } i = 1, \ldots, n\}$. If t' is a tree with root k and t a tree with $k \in fr(t)$, then $t \cdot^k t'$ is the tree obtained by inserting t' at node k. A *cut* of a tree t is the frontier of a prefix-tree of t, which is a subtree with same root as t and a domain included in $\mathrm{dom}(t)$. Stated in different words, a cut of t is a maximal (w.r.t. set inclusion) set of nodes of $\mathrm{dom}(t)$ which are pairwise incomparable by $<$. For a cut S of t the word $w(S)$ is

given by the sequence of the letters at the nodes of the cut, read from left to right w. r. t. the lexicographical ordering of the nodes.

For a tree $t \in T_\Sigma$ let $S_1, \ldots, S_n$ be the successor relations (with $x S_i y \Leftrightarrow xi = y$) and let P_a be the subset of dom (t) with $k \in P_a \Leftrightarrow t(k) = a$ for $a \in \Sigma$. Then we will identify a tree $t \in T_\Sigma$ with the relational structure

$$t = (\text{dom}(t),\ <,\ S_1,\ \ldots,\ S_n,\ (P_a)_{a \in \Sigma}).$$

Now properties of trees can be formulated in terms of the corresponding first-order language $L_1(\Sigma)$. Formulas of this language are built up from variables $x, y, \ldots$ [ranging over nodes of dom (t)], the connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, the quantifiers $\exists$, $\forall$ and the symbols $<$, $=$, $S_1, \ldots, S_n$, P_a (for $a \in \Sigma$). A formula φ with the free variables $x_1, \ldots, x_r$ is denoted by $\varphi(x_1, \ldots, x_r)$, and a formula without free variables is called a *sentence*. The interpretation of a formula $\varphi(x_1, \ldots, x_r)$ in a tree t with specified nodes $k_1, \ldots, k_r$ is defined in the usual way; we write

$$(t, k_1, \ldots, k_r) \models \varphi(x_1, \ldots, x_r)$$

or just $(t, \bar{k}) \models \varphi(\bar{x})$, if φ is satisfied in t with k_i as interpretation for x_i. The set of all trees which satisfy a sentence φ is denoted by $T(\varphi)$. A set $T \subset T_\Sigma$ is *first-order* definable if there is a $\varphi \in L_1(\Sigma)$ with $T = T(\varphi)$. The *quantifier-depth* of a formula φ, denoted by $qd(\varphi)$, is the maximum number of nested quantifiers in φ.

In the sequel for simplicity of exposition we consider only binary trees, i.e. we deal with trees over an alphabet $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$ with $\Sigma_2 = \Sigma_0$ and $\Sigma_1 = \varnothing$.

3. FIRST-ORDER FORMULAS AND STAR-FREE EXPRESSIONS

Extending the notion of special tree to trees over $\Sigma \cup D$, we call a tree $t \in T_{\Sigma, D}$ *special* if there is at most one occurrence of each symbol $d \in D$. The set of all special Σ, D-trees is then denoted by $S_{\Sigma, D}$. The language $S_{\Sigma, D}$ is a regular and even star-free subset of $T_{\Sigma, D}$ [6]. The set of all special tree languages is closed under union and intersection, but not under concatenation and star-operation. It is closed under complement w. r. t. $S_{\Sigma, D}$.

Note that defining a language $T \subset T_\Sigma$ for a given alphabet Σ the concatenation alphabet D is not fixed; but in each $\alpha \in \text{SF}(\Sigma, D)$ defining T there are only finitely many symbols $d \in D$.

We now define an interpretation of the star-free expressions of $RE(\Sigma, D)$ by setting inductively $S(\varnothing) = \varnothing$, $S(t) = \{t\} \cap S_{\Sigma, D}$, $S(\alpha \vee \beta) = S(\alpha) \cup S(\beta)$, $S(\neg \alpha) = S_{\Sigma, D} - S(\alpha)$ and $S(\alpha \cdot^d \beta) = (S(\alpha) \cdot^d S(\beta)) \cap S_{\Sigma, D}$. We call a tree language *special star-free*, if it can be described by a star-free expression with this interpretation. Our first result (Theorem 3.1) states the equivalence between special star-free and first-order definable tree languages.

Another way of characterization is a syntactical one. Here we ensure already by the construction of a set of star-free expressions that only sets of special trees are obtained. We use a star-free expression $\alpha_{\Sigma, D}$ denoting $S_{\Sigma, D}$ (for its straightforward, but tedious definition, *see* [6]) and allow only expressions of the following form: $\varnothing$, $t \in S_{\Sigma, D}$, $\alpha_{\Sigma, D} - \alpha$, $\alpha_1 \vee \alpha_2$ and $(\alpha_1 \cdot^d \alpha_2) \wedge \alpha_{\Sigma, D}$. [Here $\alpha \wedge \beta$ abbreviates $\alpha_{\Sigma, D} - ((\alpha_{\Sigma, D} - \alpha) \vee (\alpha_{\Sigma, D} - \beta))$.] We denote by $SSF(\Sigma, D)$ the star-free expressions obtained in this way; for each expression $\alpha \in SSF(\Sigma, D)$ we may then use the standard interpretation $T(\alpha)$ instead of $S(\alpha)$.

Formally we have:

THEOREM 3.1: *Let* $T \subset T_\Sigma$. *Then:*

$T = T(\varphi)$ *for some* $\varphi \in L_1(\Sigma)$

iff there is an alphabet D *and a star-free expression* $\alpha \in SF(\Sigma, D)$ *such that* $T = S(\alpha)$;

iff there is an alphabet D *and an expression* $\alpha \in SSF(\Sigma, D)$ *such that* $T = T(\alpha)$. ●

To prove Theorem 3.1 we need some notations:

Let (t, λ) with $t \in T_\Sigma$, $\lambda \in \mathrm{dom}(t)$ and $k_0, k_1, \ldots, k_n \in \mathrm{dom}(t)$ be given such that:

(i) $k_0 < k_i$;

(ii) k_i, k_j are pairwise incomparable w.r.t. $<$ $(i, j = 1, \ldots, n)$;

(iii) $k_0 \leq \lambda$ and not $k_j < \lambda$ $(j = 1, \ldots, n)$.

Then the *fragment-tree* of (t, λ) given by the nodes $k_0, k_1, \ldots, k_n$ is the subtree of t which has root k_0 and which is obtained from t by restricting $\mathrm{dom}(t)$ to those nodes k with $k_0 \leq k$ and not $k_i < k$. We denote this fragment-tree by $(t, \lambda)^{[k_0, k_1 \cdots k_n]}$.
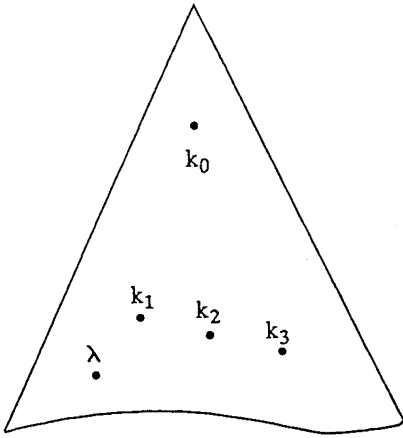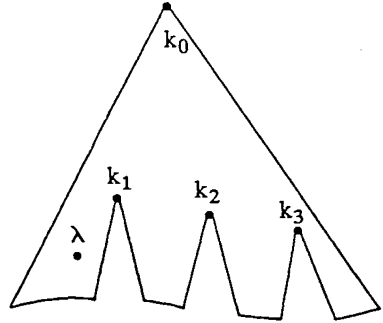
$(t, \lambda, k_0, \ldots, k_3)$ $(t, \lambda)^{[k_0, k_1 \cdots k_3]}$

We just write $(t, \lambda)^{[k_0, K]}$ if $K = \{k_1, \ldots, k_n\}$; instead of $(t, \lambda)^{[\epsilon, K]}$ resp. $(t, \lambda)^{[k_0, \varnothing]}$ we write $(t, \lambda)^{K]}$ resp. $(t, \lambda)^{[k_0}$. Fragment-trees of the form $t^{[k_0, K]}$ are defined in an analogous way as fragment-trees $(t, \lambda)^{[k_0, K]}$.

Immediately from the definition of a fragment-tree follows

LEMMA 3.2: *Let* $t^{[k_0, K]}$, $t^{[k, L]}$ *be two fragment-trees of* t *with* $k \in K$. *Then the tree*

$$t^{[k_0, K]} \cdot_k t^{[k, L]} = t^{[k_0, K - \{k\} \cup L]}$$

is also a fragment-tree (and well defined). ●

We introduce first-order formulas corresponding to fragment-trees:

DEFINITION 3.3: A first-order formula which is appropriate for the fragment-tree $(t, \lambda)^{[k_0, k_1, \ldots, k_n]}$ is a formula $\varphi(y, x_0, \ldots, x_n) \in L_1(\Sigma)$ where each quantified subformula is of the form

(1) $\exists z \left(x_0 \leqq z \wedge \bigwedge_{i=1}^{n} \neg (x_i < z) \wedge \psi(z, y, x_0, \ldots, x_n) \right)$

or

(2) $\forall z \left(\left(x_0 \leqq z \wedge \bigwedge_{i=1}^{n} \neg (x_i < z) \right) \rightarrow \psi(z, y, x_0, \ldots, x_n) \right).$

We denote such fragment-formulas by $\varphi(y)^{[x_0, x_1, \ldots, x_n]}$. ●

For $X = \{x_1, \ldots, x_n\}$ we write $\varphi(y)^{[x_0, X]}$ instead of $\varphi(y)^{[x_0, x_1, \ldots, x_n]}$. Formulas like $\varphi(y)^{[x_0}$, $\varphi(y)^{X]}$ and $\varphi^{[x_0, X]}$ are also used (corresponding to $(t, \lambda)^{[x_0}$, $(t, \lambda)^{X]}$ and $t^{[k_0, K]}$). The satisfaction relation is extended from trees and formulas to fragment-trees and fragment-formulas in the obvious way:

$(t, \lambda)^{[k_0, k_1, \ldots, k_n]} \models \varphi(y)^{[x_0, x_1, \ldots, x_n]}$:

$$\Leftrightarrow \quad (t, \lambda, k_0, k_1, \ldots, k_n) \models \varphi(y, x_0, x_1, \ldots, x_n).$$

Note that it is not necessary that a fragment-formula $\varphi(y)^{[x_0, x_1, \ldots, x_n]}$ contains the subformula $\bigwedge_{i=1}^{n} x_0 < x_i$; since the satisfaction relation is only defined for fragment-trees, $\bigwedge_{i=1}^{n} k_0 < k_i$ holds per definition.

Remark 3.4: Let $t^{[k_0, K]}$, $t^{[k, L]}$ with $k \in K$ be two fragment-trees and $\varphi_1^{[x_0, X]}$, $\varphi_2^{[x, Y]}$ with $x \in X$ two appropriate fragment-formulas. Then we have:

$$t^{[k_0, K]} \models \varphi_1^{[x_0, X]} \quad \text{and} \quad t^{[k, L]} \models \varphi_2^{[x, Y]}$$

iff

$$t^{[k_0, K - \{k\} \cup L]} \models \exists x (\varphi_1^{[x_0, X]} \wedge \varphi_2^{[x, Y]}). ●$$

Note that the formula $\exists x (\varphi_1^{[x_0, X]} \wedge \varphi_2^{[x, Y]})$ is equivalent to

$$\exists x \left(x_0 \leqq x \wedge \bigwedge_{i=1}^{m} \neg (x_i < x) \wedge \bigwedge_{i=1}^{m} \neg (y_i < x) \wedge \varphi_1^{[x_0, X \cup \{x\}]} \wedge \varphi_2^{[x, Y]} \right)$$

and hence to a fragment-formula of the form $\varphi^{[x_0, X - \{x\} \cup Y]}$. So fragment-trees are possible models, as desired.

Proof of Theorem 3.1: The easier direction (from right to left) is shown by induction on star-free expressions: For each $\alpha \in SF(\Sigma, D)$ with $S(\alpha) \subset T_\Sigma$ there is a sentence $\varphi_\alpha \in L_1(\Sigma)$ such that $S(\alpha) = T(\varphi_\alpha)$.

It is convenient to drop first the restriction $S(\alpha) \subset T_\Sigma$ and to show that for each $\alpha \in SF(\Sigma, D)$ there is a formula $\varphi_\alpha \in L_1(\Sigma)$ of the form:

$$\varphi_\alpha : \quad \bigvee_{i \in I} \varphi_i^{x_1 \cdots x_{n_i}]}$$

which describes the expression α in the following way:

$$\forall t \in T_{\Sigma, D}: \quad t \in S(\alpha) \quad \Leftrightarrow \quad \exists i \in I, \quad \exists t' \in T_\Sigma \quad \text{with} \quad k_1, \ldots, k_{n_i} \in \mathrm{dom}(t'):$$

$$t'^{k_1 \cdots k_{n_i}]} \models \varphi_i^{x_1 \cdots x_{n_i}]} \quad \text{and} \quad t' \cdot^{k_1} d_1 \cdot \ldots \cdot^{k_{n_i}} d_{n_i} = t.$$

Obviously $\emptyset$ is described by $\exists y \, (y < y)$. A tree $t \in T_{\Sigma, D} - S_{\Sigma, D}$ again is described by $\exists y \, (y < y)$, since in this case $S(t) = \emptyset$. Concerning trees $t \in S_{\Sigma, D}$ the construction will be inductively:

The tree $a \in \Sigma$ is described by

$$\exists y \, (\underbrace{\forall z \, (y \leq z)}_{y \text{ root}} \wedge \underbrace{\forall z \, (z \leq y)}_{y \text{ at the front}} \wedge \, y \in P_a)$$

and the tree $d_i \in D$ by

$$\forall z \, (\neg \, (x_i < z) \to (z = x_i)).$$

Note that the last formula is a fragment-formula of the form $\varphi^{x_i]}$.

Let $t = a(t_1, t_2)$ be given and the tree t_1 resp. t_2 be described by $\varphi_1^{x_1, \ldots, x_m]}$ resp. $\varphi_2^{x'_1, \ldots, x'_n]}$. The t describing formula is then given by $\varphi^{x_1 \cdots x_m, x'_1 \cdots x'_n]}$ where

$$\varphi: \ \exists y \, (\forall z \, (y \leq z) \wedge y \in P_a$$

$$\wedge \, \exists y_1 \, (y \, S_1 \, y_1 \wedge \varphi_1^{[y_1, x_1 \cdots x_m]}) \wedge \exists y_2 \, (y \, S_2 \, y_2 \wedge \varphi_2^{[y_2, x'_1 \cdots x'_n]})),$$

e. g. all quantified subformulas occurring in φ have to be "relativized" to the variables $x_1, \ldots, x_m, x'_1, \ldots, x'_n$ as in Definition 3.3.

Now, let $\alpha, \beta \in SF(\Sigma, D)$ be given. We assume we have formulas of the form

$$\varphi_\alpha: \ \bigvee_{i \in I} \varphi_i^{x_1 \cdots x_{m_i}]}, \qquad \varphi_\beta: \ \bigvee_{j \in J} \psi_j^{x'_1 \cdots x'_{m_j}]}$$

to describe α resp. β. Then a formula which is equivalent to $\alpha \cdot^d \beta$ is given by

$$\bigvee_{i \in I} \bigvee_{j \in J} (\varphi_i^{x_1 \cdots x_{m_i}]} \overset{d}{-} \psi_j^{x'_1 \cdots x'_{m_j}]}),$$

where $\varphi^{x_1 \cdots x_m]} \overset{d}{-} \psi^{x'_1 \cdots x'_n]} =$

$- \ \varphi^{x_1 \cdots x_m]} \quad$ if $\ d \notin \{d_1, \ldots, d_m\};$

— $\exists y\ (y < y)$ if there are i, j ($i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$), such that $d_i \neq d$ and $d_i = d'_j$

— $(\exists x\ \varphi\ (x_i/x)^{x_1 \cdots x_{i-1},\, x,\, x_{i+1} \cdots x_m]} \wedge \psi^{[x,\, x'_1 \cdots x'_n]})^{x_1 \cdots x_{i-1},\, x_{i+1} \cdots x_m,\, x'_1 \cdots x'_n]}$
else (*i. e.* especially $\exists i \in \{1, \ldots, m\}$ with $d_i = d$).

The second case ensures that only trees $t \in S_{\Sigma, D}$ satisfy the formula. In the third formula $\varphi\ (x_i/x)$ arises from φ by substituting x_i by x. This substitution is necessary since one of the variables $x'_1 \ldots x'_n$ may be equal to x_i (*i. e.* corresponding to d_i). This variable however must be excluded from the quantification $\exists x$, which causes that the former variable x_i is now bounded, *i. e.* the concatenation symbol d_i in α is vanished.

Concerning the boolean connectives, the construction of $\varphi_{\alpha \vee \beta}$ is trivial. Building the formula $\varphi_{\neg \alpha}$, the main steps are the construction of formulas $\varphi_{\neg \alpha\, i}$ and the conjunction of them. Given φ_α of the form $\varphi^{x_1 \cdots x_n]}$ the formula $\varphi_{\neg \alpha}$ expresses the disjunction of $\neg\, \varphi^{x_1 \cdots x_n]}$ with all formulas of the form $(\exists y\ y = y)^{x_1, \ldots, x_m]}$ for $m \neq n$, $m \leq |D|$. This step is tedious but not difficult and hence left out here (for more details *see* [6]). Using these results for the case of expressions α with $S(\alpha) \subset T_\Sigma$ we have formulas φ_α, which are disjunctions of sentences of $L_1 (\Sigma)$. This completes the proof of one direction of Theorem 3.1.

Conversely we show that to each $\varphi \in L_1 (\Sigma)$ there exist an alphabet D and an expression $\alpha_\varphi \in SF(\Sigma, D)$ such that $T(\varphi) = S(\alpha_\varphi)$. Therefore we have to interpret regular expressions in fragment-trees, and we have to state what is the counterpart of free variables within star-free expressions. Let $S_{\{d_1, \ldots, d_n\}}$ be the set of trees with exactly one occurrence of the symbols $d_1, \ldots, d_n$ from D and no occurrence of another symbol d from D. We write

$$t^{[k_0,\, k_1 \cdots k_n]} \models_{\{d_1 \ldots d_n\}} \alpha$$

if

$$S(\alpha) \subset S_{\{d_1, \ldots, d_n\}} \quad \text{and} \quad t^{[k_0,\, k_1 \cdots k_n]\,.k_1}\, d_1 \ldots \cdot^{k_n} d_n \in S(\alpha).$$

Corresponding to Lemma 3.4 follows.

LEMMA 3.5: *Let* $\alpha, \beta \in SF(\Sigma, D)$ *be given with* $S(\alpha) \subset S_{D'}$, $S(\beta) \subset S_{D''}$ *for* $D', D'' \subset D$, $D' \cap D'' \subset \{d_i\}$ *and* $d_i \in D'$. *For fragment-trees* $t^{[k_0, K]}$, $t^{[k_i, L]}$ *with* $k_i \in K$ *we have:*

$$t^{[k_0, K]} \models_{D'} \alpha \quad \text{and} \quad t^{[k_i, L]} \models_{D''} \beta$$

iff

$$t^{[k_0, K-\{k_i\} \cup L]} \models_{(D'-\{d_i\} \cup D'')} \alpha \cdot^{d_i} \beta. \quad \bullet$$

Note that this Lemma is (trivially) true, because there is only one concatenation point labeled d_i in α corresponding to the single "composition" node k_i in t.

It suffices to show the following

LEMMA 3.6: *For each set of variables* $X = \{x_1, \ldots, x_n\}$ *and any formula* $\varphi^{[x_0, X]}$ *there is an alphabet* $D \supset \{d_1, \ldots, d_n\}$ *and an expression* $\alpha_\varphi \in SF(\Sigma, D)$ *such that for all* $t \in T_\Sigma$:

(*) $t^{[k_0, K]} \models \varphi^{[x_0, X]}$ iff $t^{[k_0, K]} \models_{\{d_1, \ldots, d_n\}} \alpha_\varphi. \quad \bullet$

We write $D(X)$ for the subset $\{d_1, \ldots, d_n\}$ of D and say in the case of (*) that the expression α_φ is $D(X)$-*equivalent* to $\varphi^{[x_0, X]}$. (Note that $|X| = |D(X)|$.)

It suffices to show Lemma 3.6 in order to prove the Theorem, because with the Lemma also for each $\varphi^{[x_0, \varnothing]}$ there is an $\varnothing$-equivalent expression α_φ. Furthermore, each $\varphi \in L_1(\Sigma)$ is equivalent to $\bar{\varphi} = \exists x_0 \ (\neg \exists y (y < x_0) \wedge \varphi^{[x_0, \varnothing]})$, and we have for arbitrary $\varphi \in L_1(\Sigma)$ and $t \in T_\Sigma$ with root k_0:

$$t \models \varphi \quad \Leftrightarrow \quad t \models \bar{\varphi} \quad \Leftrightarrow \quad t^{[k_0, \varnothing]} \models \varphi^{[x_0, \varnothing]} \quad \Leftrightarrow \quad t^{[k_0, \varnothing]} \models_\varnothing \alpha_\varphi \quad \Leftrightarrow \quad t \in S(\alpha_\varphi).$$

Note that concerning the second equivalence it is trivial that t and $t^{[k_0, \varnothing]}$ are equivalent since k_0 is the root of t. The formula $\varphi^{[x_0, \varnothing]}$ arose from φ by "relativizing" each quantification to x_0. So with the interpretation of x_0 by the root k_0 the equivalence follows immediately.

To show Lemma 3.6 we need the following Decomposition Lemma which will be proved in Section 4, using the Ehrenfeucht-Fraïssé-game.

Decomposition Lemma 3.7

For each formula $\varphi(y)^{[x_0, X]}$ with quantifier-depth q there exist a finite number of formulas $\varphi_{a,i}^{[x_0, X_{a,i} \cup \{y\}]}$, $\varphi_{b,i}^{[y, X_{b,i}]}$ with quantifier-depth q and $X_{a,i} \cup X_{b,i} = X$, such that $\exists y \, \varphi(y)^{[x_0, X]}$ is equivalent to

$$\bigvee_{i \in I} \exists y \, (\varphi_{a,i}^{[x_0, X_{a,i} \cup \{y\}]} \wedge \varphi_{b,i}^{[y, X_{b,i}]}). \quad \bullet$$

Intuitively the second part of the conjunction only speaks about parts of trees "below" y and the first of parts "above" y (in the sense of "not below y"). Therefore the subformulas are labeled with "a", resp. "b".

We prove Lemma 3.6 by induction on the quantifier-depth q of the formulas $\varphi^{[x_0, X]}$. The most interesting step concerns quantification (the others are easy). We know that for each formula $\varphi^{[x_0, X]}$ with $qd(\varphi) \leq q$ there exists an alphabet D and $\alpha \in SF(\Sigma, D)$ such that $t^{[k_0, K]} \models \varphi^{[x_0, X]}$ iff $t^{[k_0, K]} \models_{D(X)} \alpha$, with appropriate $D(X) \subset D$.

Now let $\varphi(y)^{[x_0, X]}$ with $qd(\varphi) = q$ be given. By the Decomposition Lemma, $\exists y \varphi(y)^{[x_0, X]}$ is equivalent to

$$\bigvee_{i \in I} \exists y (\varphi_{a, i}^{[x_0, X_{a, i} \cup \{y\}]} \wedge \varphi_{b, i}^{[y, X_b, i]})$$

where the formulas $\varphi_{a, i}$, $\varphi_{b, i}$ are also of quantifier-depth q. Let $\alpha_{a, i}$ resp. $\alpha_{b, i}$ be the expressions which are $D(X_{a, i} \cup \{y\})$-equivalent to $\varphi_{a, i}^{[x_0, X_{a, i} \cup \{y\}]}$, resp. $D(X_{b, i})$-equivalent to $\varphi_{b, i}^{[y, X_b, i]}$. If we set $D(X_{a, i} \cup \{y\}) = D(X_{a, i}) \cup \{d_y\}$, we have by Remarks 3.4 and 3.5 that $\exists y (\varphi_{a, i}^{[x_0, X_{a, i} \cup \{y\}]} \wedge \varphi_{b, i}^{[y, X_b, i]})$ is $D(X_{a, i}) \cup D(X_{b, i})$-equivalent to $\alpha_{a, i} \cdot^{d_y} \alpha_{b, i}$. Hence $\exists y \varphi(y)^{[x_0, X]}$ is $D(X)$-equivalent to $\bigvee_{i \in I} \alpha_{a, i} \cdot^{d_y} \alpha_{b, i}$ as desired.

4. PROOF OF THE DECOMPOSITION LEMMA WITH THE EHRENFEUCHT-FRAISSE-GAME

In this section we prove the Decomposition Lemma by means of the Ehrenfeucht-Fraissé-game. (In some cases we state standard lemmas without proof; for details we refer the reader to [10].) We start introducing these games played on two fragment-trees of the form $(t, \lambda_0)^{[k_0, K]}$ and $(t', \lambda_0')^{[k_0', K']}$ with $|K| = |K'|$. A play of the Ehrenfeucht-Fraisse game $G_n((t, \lambda_0)^{[k_0, K]}, (t', \lambda_0')^{[k_0', K']})$ consists of a sequence of n moves. Within each move player I chooses an element of $dom(t^{[k_0, K]})$ or of $dom(t'^{[k_0', K']})$, and player II chooses an element of the other domain. The element which is chosen from $dom(t^{[k_0, K]})$ [resp. $dom(t'^{[k_0', K']})$] in the i-th move is called λ_i (resp. λ_i'). For simplicity of exposition we set $\lambda_{n+1+i} := k_i$ and $\lambda_{n+1+i}' := k_i'$ ($i = 0, \ldots, |K|$).

Player II wins the play of the game $G_n((t, \lambda_0)^{[k_0, K]}, (t', \lambda_0')^{[k_0', K']})$, if for all i, j with $0 \leq i, j \leq n+1+|K|$ we have:

(i) $\lambda_i < \lambda_j \Leftrightarrow \lambda_i' < \lambda_j'$;

(ii) $\lambda_i S_k \lambda_j \Leftrightarrow \lambda_i' S_k \lambda_j'$ ($k = 1, 2$);

(iii) $\lambda_i = \lambda_j \Leftrightarrow \lambda_i' = \lambda_j'$;

(iv) $\lambda_i \in P_a \Leftrightarrow \lambda_i' \in P_a$.

Thus player II has won if he respects the relations $<$, $=$, S_k, P_a in all his choices, that means if the sequences of nodes $(\lambda_0, \ldots, \lambda_n, k_0, \ldots, k_{|K|})$ and $(\lambda_0', \ldots, \lambda_n', k_0', \ldots, k_{|K'|}')$ define a partial isomorphism w.r.t. $<$, $=$, S_k and P_a $(a \in \Sigma)$.

Player II has a winning-strategy in the game $G_n((t, \lambda_0)^{[k_0, K]}, (t', \lambda_0')^{[k_0', K']})$, [denoted here by "II wins $G_n((t, \lambda_0)^{[k_0, K]}, (t', \lambda_0')^{[k_0', K']})'$"], if player II is able to respond to any move of player I such that II wins the resulting play.

We write:

$$(t, \lambda_0)^{[k_0, K]} \sim_n (t', \lambda_0')^{[k_0', K']}: \text{iff II wins } G_n((t, \lambda_0)^{[k_0, K]}, (t', \lambda_0')^{[k_0', K']}).$$

The logical counterpart of this relation is the following:

DEFINITION 4.1:

$(t, \lambda_0)^{[k_0, K]} \equiv_n (t', \lambda_0')^{[k_0', K']}:$ iff for all $\varphi(y)^{[x_0, X]}$ with $qd(\varphi) = n:$

$$((t, \lambda_0)^{[k_0, K]} \vDash \varphi(y)^{[x_0, X]} \Leftrightarrow (t', \lambda_0')^{[k_0', K']} \vDash \varphi(y)^{[x_0, X]}).$$

LEMMA 4.2 (cf. [10], Lemma 13.4, Lemma 13.10, Theorem 13.11): *Let $n > 0$ and $(t, \lambda_0)^{[k_0, K]}$, $(t', \lambda_0')^{[k_0', K']}$ be two fragment-trees. Then:*

(a) $t^{[k_0, K]} \sim_n t'^{[k_0', K']}$ *iff for all $\lambda \in \mathrm{dom}(t^{[k_0, K]})$ there is a $\lambda' \in \mathrm{dom}(t'^{[k_0', K']})$ such that:*

$$(t, \lambda)^{[k_0, K]} \sim_{n-1} (t', \lambda')^{[k_0', K']}$$

and for all $\lambda' \in \mathrm{dom}(t'^{[k_0', K']})$ there is a $\lambda \in \mathrm{dom}(t^{[k_0, K]})$ such that:

$$(t, \lambda)^{[k_0, K]} \sim_{n-1} (t', \lambda')^{[k_0', K']};$$

(b) $(t, \lambda_0)^{[k_0, K]} \sim_n (t', \lambda_0')^{[k_0', K']}$ *iff* $(t, \lambda_0)^{[k_0, K]} \equiv_n (t', \lambda_0')^{[k_0', K']};$

(c) *the relation $\sim_n$ has a finite number of equivalence classes (w.r.t. fragment-trees $(t, \lambda_0)^{[k_0, K]}$ with fixed $|K|$);*

(d) *each $\sim_n$-class π (also called n-type π) is definable by a formula φ_π (a "type description") with quantifier-depth n;*

(e) *each formula φ of the form $\varphi(y)^{[x_0, X]}$ is equivalent to a finite disjunction of type-descriptions φ_π, that means:*

$\varphi(y)^{[x_0, X]}$ *is equivalent to a formula of the form $\varphi_{\pi_i}(y)^{[x_0, X]}$* (over all appropriate fragment-trees). ●

Note that all definitions and lemmas remain true for fragment-trees of the form $t^{[x_0, X]}$, *i.e.* without a further designated node λ_0, and *a fortiori* for trees t without any parameters.

We now show a Composition Lemma for games with which we are able to prove the Decomposition Lemma for formulas. (A similar result for linear orderings can be found in [10], Theorem 6.6). We consider fragment-trees $(t, \lambda_0)^{[k_0, K]}$ and $(t', \lambda_0')^{[k_0', K']}$ with $|K| = |K'|$, which shall be decomposed at the node λ_0 (resp. λ_0') into fragment-trees $t^{[k_0, K_a \cup \{\lambda_0\}]}$ (resp. $t'^{[k_0', K_a' \cup \{\lambda_0'\}]}$) above (again in the sense of "not below") the node λ_0 (resp. λ_0') and $t^{[\lambda_0, K_b]}$ (resp. $t'^{[\lambda_0', K_b']}$) below the node λ_0 (resp. λ_0'). The set of nodes $K_a \subset K$ (resp. K_a') thereby consists of the nodes of K (resp. K') above the node λ_0 (resp. λ_0') and the set $K_b \subset K$ (resp. K_b') of the nodes of K (resp. K') below the node λ_0 (resp. λ_0'). The Composition Lemma shows that it is possible to obtain a winning-strategy for the game on the composed fragment-tree from the winning-strategies for the constituting fragment-trees. Formally:

Composition Lemma 4.3

Let $t, t' \in T_\Sigma$.

If $\quad t^{[k_0, K_a \cup \{\lambda_0\}]} \sim_n t'^{[k_0', K_a' \cup \{\lambda_0'\}]} \quad$ and $\quad t^{[\lambda_0, K_b]} \sim_n t'^{[\lambda_0', K_b']}, \quad$ then also $(t, \lambda_0)^{[k_0, K_a \cup K_b]} \sim_n (t', \lambda_0')^{[k_0', K_a' \cup K_b']}$.

Sketch of proof

Let G_a be a winning-strategy for player II in the game on the upper two fragment-trees of t, t' and G_b a winning-strategy for the lower fragment-trees.

One has to verify that the following strategy is a winning-strategy on the composed fragment-trees:

Player II selects for each node chosen by player I from $\mathrm{dom}\,(t^{[k_0, K_a \cup \{\lambda_0\}]}) - \{\lambda_0\}$ (resp. $\mathrm{dom}\,(t'^{[k_0', K_a' \cup \{\lambda_a'\}]}) - \{\lambda_0'\}$) the node as given by the strategy G_a, and for each node chosen by player I from $\mathrm{dom}\,(t^{[\lambda_0, K_b]})$ [resp. $\mathrm{dom}\,(t'^{[\lambda_0', K_b']})$] the corresponding node given by the strategy G_b.

For the inductive proof that a winning strategie results *see* [6]. ●

Now we are able to prove the Decomposition Lemma of the previous Section:

Proof of the decomposition Lemma 3.7: By Lemma 4.2(e), each formula $\varphi(y)^{[x_0, X]}$ with $qd(\varphi) = n$ is equivalent to a finite disjunction of n-type-descriptions of the form $\varphi_{n_i}(y)^{[x_0, X]}$. We show that each $\varphi_\pi(y)^{[x_0, X]}$ is decomposable in the desired way.

We take the set T of all pairs (π_a, π_b) such that:

if $t^{[k_0, K_a \cup \{\lambda\}]} \in \pi_a$ and $t^{[\lambda, K_b]} \in \pi_b \in \pi_b$ then $t^{[k_0, K_a \cup K_b]} \in \pi.$

This set is well defined by Lemma 4.2 (b) together with the Composition Lemma, and it is finite by Lemma 4.2 (c).

Let $\varphi_{\pi_a}^{[x_0, X_a \cup \{y\}]}$ be the formula defining π_a and $\varphi_{\pi_b}^{[y, X_b]}$ the one defining π_b, each of them of quantifier-depth n. [These formulas exist by Lemma 4.2 (d).] Hence it follows, as desired, that:

$$t^{[k_0, K]} \vDash \exists y\, \varphi_\pi (y)^{[x_0, X]}$$

if

$$t^{[k_0, K]} \vDash \bigvee_{(\pi_a, \pi_b) \in \pi} \exists y\, (\varphi_{\pi_a}^{[x_0, X_a \cup \{y\}]} \wedge \varphi_{\pi_b}^{[y, X_b]}). \quad \bullet$$

5. APERIODIC LANGUAGES AND THEIR RELATIONSHIP TO FIRST-ORDER DEFINABLE LANGUAGES

We show in this section that first-order definability is a more restrictive notion than aperiodicity; hence the well-known result of the theory of regular word languages on the equivalence between aperiodicity and first-order definability fails in the case of tree languages. That each first-order definable tree language is aperiodic, is shown immediately by induction (*see* [9] for a corresponding proof in the case of word languages).

THEOREM 5.1: *There is an aperiodic regular tree language which is not first-order definable.*

Proof: Let $\Sigma = \{a, b\}$. We define $T \subset T_\Sigma$ to be the set of trees where *for all cuts S with $|S| > 1$ the word $w(S)$ is in $\Sigma^* aa \Sigma^*$*. It is easy to see that T is a regular, aperiodic tree language: Note that the existence of a cut $\notin \Sigma^* aa \Sigma^*$ is directly expressible in monadic second order logic over finite trees (which implies by [14], [2] that the resulting tree language is regular). Concerning aperiodicity this existence claim for trees $s_0 . s^n . t$ does not depend on the choice of n provided $n > 1$.

We show that T is not first-order definable in the following way, using the equivalence $\equiv_n$ of Definition 4.1:

(∗)   For each $n > 0$ there are two trees $t_n, s_n \in T_\Sigma$ with $t_n \equiv_n s_n$,

$$\text{but } t_n \in T \text{ and } s_n \notin T.$$

With (∗) it follows that for any first-order sentence φ, say of quantifier depth n, there exist trees $t, s \in T_\Sigma$, which are on the one hand indistinguishable by φ (since $t \equiv_n s$) and on the other hand satisfy $t \in T$ and $s \notin T$. So, clearly, T will not be first-order definable.

We define the trees t_n, s_n by induction. The trees t_1, s_1 look as follows:
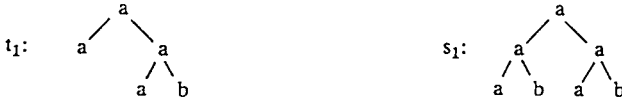
It is easy to see that the trees t_1 and s_1 are indistinguishable by any of the formulas with quantifier depth 1. [Note that these formulas are, up to logical equivalence, Boolean combinations of formulas $\exists x (x \in P_a)$.] Hence we have $t_1 \equiv_1 s_1$. Moreover holds $t_1 \in T$ and $s_1 \notin T$ (the cut $\{11, 12, 21, 22\}$ does not contain two consecutive letters a).

The main difficulty of the proof is the construction of t_{n+1}, s_{n+1}. Let trees t_n, s_n be given with $t_n \equiv_n s_n$, $t_n \in T$, $s_n \notin T$ and the following additional three properties:

(i) The root of both trees is labeled with a.

(ii) The leftmost path of s_n is labeled with a.

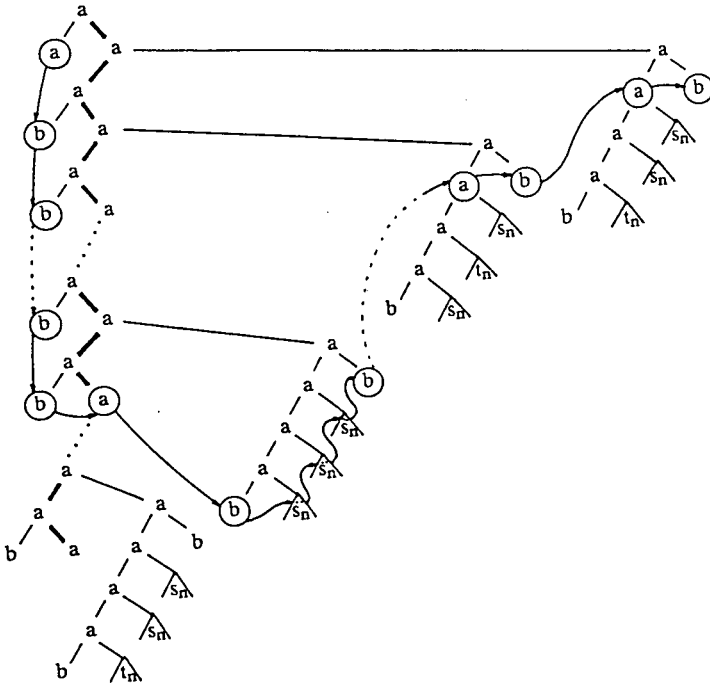(iii) There is a cut S in s_n with $w(S) \notin \Sigma^* aa \Sigma^*$ which ends with b.

Note that the trees t_1, s_1 realize these three properties. We will ensure them also for t_{n+1} and s_{n+1}.

To define t_{n+1} and s_{n+1} we consider for arbitrary $k \geqq 1$ a "zigzag tree" z_k and a fixed "path tree" p:

The trees z_k and p are trees over $\Sigma \cup \{c_1, \ldots, c_k, d_1, d_2, d_3\}$. Now t_{n+1} will be from the set of trees of the form:

(1)
$$z_k \overset{\cdot\, c_1}{} (p \overset{\cdot\, d_1}{} s_n^{\cdot d_{i\,2}} s_n^{\cdot d_{i\,3}} t_n) \\ \vdots \\ \overset{\cdot\, c_k}{} (p \overset{\cdot\, d_1}{} s_n^{\cdot d_{i\,2}} s_n^{\cdot d_{i\,3}} t_n),$$

with $\{d_{i\,2}, d_{i\,3}\} = \{d_2, d_3\}$, and s_{n+1} will be one of the following trees (differing from t_{n+1} at exactly one c_j):

(2)
$$z_k \overset{\cdot\, c_1}{} (p \overset{\cdot\, d_1}{} s_n^{\cdot d_{i\,2}} s_n^{\cdot d_{i\,3}} t_n) \\ \vdots \\ \overset{\cdot\, c_j}{} (p \overset{\cdot\, d_1}{} s_n^{\cdot d_2} s_n^{\cdot d_3} s_n) \\ \vdots \\ \overset{\cdot\, c_k}{} (p \overset{\cdot\, d_1}{} s_n^{\cdot d_{i\,2}} s_n^{\cdot d_{i\,3}} t_n),$$

also with $\{d_{i\,2}, d_{i\,3}\} = \{d_2, d_3\}$.

We shall explain that each tree of the above form (1) is a member of T [$i.\,e.$ that for all cuts S of such a tree we have $w(S) \in \Sigma^* \, aa \, \Sigma^*$], and for each tree of the above form (2) there is a cut S with $w(S) \notin \Sigma^* \, aa \, \Sigma^*$.

To show the first claim, consider t_{n+1} say of the form:

Then any cut which does not start by two consecutive letters a starts with abb^*. To pass through the whole tree on a cut one has to cross the "zig-zag line" and to pass one of the subtrees $p^{\cdot d_1} s_n^{\cdot d_{i2}} s_n^{\cdot d_{i3}} t_n$, and the only way to avoid two consecutive a is to choose here the leftmost node of the subtree, which is labeled with b. To reach now the rightmost node of this subtree

(which is necessary for the construction of a cut), one has to pass the three sub-subtrees (t_n or s_n) at least at the root. Thereby one has to choose a cut in the sub-subtree t_n (then the claim is shown, because in t_n all cuts are from $\Sigma^* aa \Sigma^*$) or otherwise to choose the root of t_n. Because of the three assumed properties for s_n, t_n above, this root is labeled with a and each node of the leftmost path of the tree s_n also, hence in all cases one obtains a cut which contains two consecutive letters a. (Note that if the tree t_n would be replaced only for d_1, the last argument would fail, because there would not follow a tree s_n, but only a node labeled with b.)

Now we consider the claim for a tree of form (2):

The marked cut clearly is a not in $\Sigma^* aa \Sigma^*$. Note that here the property is used that there is a cut in s_n without two consecutive letters a which ends with b (induction assumption).

Because of the finite index of $\equiv_n$ [Lemma 4.2 (c)] there are numbers k and $j_1 < j_2 < j_3$, such that:

$$z_k^{\cdot c_1} p(s_n) \ldots {}^{\cdot c_{j1}} (p^{\cdot d_1} s_n^{\cdot d_2} s_n) \ldots {}^{\cdot c_k} p(s_n)$$

$$\equiv_n z_k^{\cdot c_1} p(s_n) \ldots {}^{\cdot c_{j2}} (p^{\cdot d_1} s_n^{\cdot d_2} s_n) \ldots {}^{\cdot c_k} p(s_n)$$

$$\equiv_n z_k^{\cdot c_1} p(s_n) \ldots {}^{\cdot c_{j3}} (p^{\cdot d_1} s_n^{\cdot d_2} s_n) \ldots {}^{\cdot c_k} p(s_n),$$

where $p(s_n) := p^{\cdot d_1} s_n^{\cdot d_2} s_n^{\cdot d_3} s_n$. Note that in each of these trees one d_3 remains for later substitution.

Let r_1, r_2, r_3 be trees as given by such parameters j_1, j_2, j_3. These trees are also equivalent to each tree which arises from them by replacing arbitrary occurrences of s_n by t_n. This results from a (repeated) application of the Composition Lemma 4.3 and the induction assumption $s_n \equiv_n t_n$.

We will consider trees r_1', r_2' defined as follows:

r_1' arises from z_k such that the labels $\{c_1, \ldots, c_k\} - \{c_{j_1}, c_{j_2}\}$ of z_k are replaced by trees $p^{\cdot d_1} s_n^{\cdot d_2} t_n^{\cdot d_3} s_n$, the label c_{j_1} by $p^{\cdot d_1} s_n^{\cdot d_2} s_n$ and the label c_{j_2} by $p^{\cdot d_1} s_n^{\cdot d_2} s_n^{\cdot d_3} t_n$;

r_2' arises in a similar way from z_k, where the labels $\{c_1, \ldots, c_k\} - \{c_{j_1}, c_{j_2}\}$ are replaced as before, c_{j_1} by $p^{\cdot d_1} s_n^{\cdot d_2} s_n^{\cdot d_3} s_n$ and c_{j_2} by $p^{\cdot d_1} s_n^{\cdot d_2} s_n$.

Clearly $r_1' \equiv_n r_1$ and $r_2' \equiv_n r_2$ and hence $r_1' \equiv_n r_2'$ by the above remark.

Now we may define the desired trees t_{n+1} and s_{n+1}:

$$t_{n+1} := r_1'^{\cdot d_3} t_n \qquad \text{and} \qquad s_{n+1} := r_1'^{\cdot d_3} s_n.$$

Obviously the trees t_{n+1}, s_{n+1} are of the form (1), resp. (2), described above, hence $t_{n+1} \in T$, $s_{n+1} \notin T$ and they realize the three properties (i), (ii), (iii). So it remains to show $t_{n+1} \equiv_{n+1} s_{n+1}$. By Lemma 4.2 (a) it suffices to show:

(i) for all $k \in \text{dom}(t_{n+1})$ there is a $k' \in \text{dom}(s_{n+1})$ such that:

$$(t_{n+1}, k) \equiv_n (s_{n+1}, k')$$

(ii) for all $k' \in \text{dom}(s_{n+1})$ there is a $k \in \text{dom}(t_{n+1})$ such that:

$$(t_{n+1}, k) \equiv_n (s_{n+1}, k').$$

(i): There are two possibilities where the node k may occur in t_{n+1}. The first one is $k \in \text{dom}(r_1')$ but $r_1'(k) \neq d_3$ (note that $t_{n+1} = r_1'^{\cdot d_3} t_n$). In this case we choose $k' := k$ and it follows with $(r_1', k) \equiv_n (r_1', k')$ and $t_n \equiv_n s_n$ by the Composition Lemma that $(r_1'^{\cdot d_3} t_n, k) \equiv_n (r_1'^{\cdot d_3} s_n, k')$, or equivalently $(t_{n+1}, k) \equiv_n (s_{n+1}, k')$.

The second possibility is the occurrence of k in that subtree t_n of t_{n+1} which replaces d_3 of r_1'. Considering the construction of r_2' it is clear that $s_{n+1} = r_2'^{\cdot d_3} t_n$ (note that $s_{n+1} := r_1'^{\cdot d_3} s_n$). Now we let λ be the node with $r_1'(\lambda) = d_3$ and λ' the one with $r_2'(\lambda') = d_3$. Then the trees $t_{n+1}^{[\lambda}$ and $s_{n+1}^{[\lambda'}$ are both isomorphic to t_n (consider the constructions of t_{n+1} and s_{n+1}). Suppose k' of $\text{dom}(s_{n+1}^{[\lambda'})$ is the node corresponding to k of $\text{dom}(t_{n+1}^{[\lambda})$. Then clearly:

$$t_{n+1}^{[\lambda]} \equiv_n s_{n+1}^{[\lambda']} \text{ (or equivalently } r_1' \equiv_n r_2') \qquad \text{and} \qquad (t_{n+1}^{[\lambda}, k) \equiv_n (s_{n+1}^{[\lambda'}, k').$$

Hence with Composition Lemma we obtain $(t_{n+1}, k) \equiv_n (s_{n+1}, k')$ which was to be shown.

(ii) To show this direction of the proof one argues in a similar way with the tree r_3 instead of r_2.   ●

6. CONCLUSION

In this paper we have presented two results on first-order definable sets of finite trees: We have characterized first-order logic over finite trees in terms of a special class of star-free tree expressions, and we have clarified the relation between first-order logic and aperiodicity: For sets of finite trees, first-order definability is a more restrictive notion than aperiodicity.

It is well known that monadic second-order logic (first-order logic extended by quantification over sets) over trees characterizes the regular tree languages ([2, 14]). Thomas defined in [13] two restricted versions of monadic second-order logic related to tree structures: *chain-* and *antichain*-definability. A subset of dom(t) is a *chain* if it is linearly ordered by the partial tree ordering $<$. It is an *antichain* if any two distinct elements in it are incomparable w.r.t. $<$. A set $T \subset T_\Sigma$ is chain-(antichain-) definable if there is a $\varphi \in L_2(\Sigma)$ such that T is the set of all trees t in which φ is satisfied under the proviso that the set variables in φ range only over chains (antichains). Thomas showed the equivalence between the star-free languages and antichain definable languages; he also proved the equivalence of aperiodicity and first-order logic for chain definable sets of trees. Given arbitrary alphabets Σ (including $\Sigma_1 \neq \emptyset$) the classes of regular, star-free, special star-free, monadic second-order (m.s.o.) definable, antichain definable, chain definable, first-order (f.o.) definable and aperiodic tree languages are – as far as known – related as follows:

☐ m.s.o. definable (= regular)

▨ chain definable

▨ antichain definable (= star-free)

▥ aperiodic

▢ f.o. definable (= special star-free)
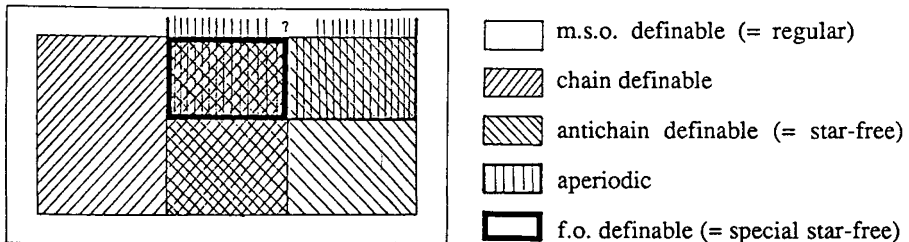
If we consider sets of strictly binary trees (*i.e.* over $\Sigma = \Sigma_0 = \Sigma_2$ with $\Sigma_1 = \emptyset$), it is open whether there are regular tree languages, which are not

star-free. A joint result with D. Niwinski states that a natural candidate for a non star-free set of trees, consisting of all binary trees over the alphabet $\{a, b\}$ with an even number of letters a, is in fact star-free [7]. Hence the intuition concerning sets of words that star-free languages "are not able to capture modulo counting" fails in the case of tree languages.

Further questions which remain open are:

- Is first-order definability decidable for regular sets of trees?
- Which subclass of regular tree expressions characterizes the chain definable tree languages?
- Is there an aperiodic tree language which is not antichain-definable (consider the question mark in the diagram)?
- Are there regular tree languages which are not (chain + antichain)-definable (in this case the set quantifiers range either over chains or antichains)?

REFERENCES

1. J. R. Büchi, Weak second-order arithmetic and finite automata, *Z. math. Logik Grundlagen Math*, 6, 1960, pp. 66-92.
2. J. Doner, Tree acceptors and some of their applications, *J. of Comp. and System Sci.*, 4, 1970, pp. 406-451.
3. C. C. Elgot, Decision problems of finite automata design and related arithmetics, *Trans. Amer. Math. Soc.*, 98, 1961, pp. 21-52.
4. F. Gecseg and M. Steinby, "Tree Automata", Akademiai Kiado, Budapest, 1984.
5. U. Heuter, First-order properties of finite trees, star-free expressions and aperiodicity, *Proc. 5th STACS, R. Cori and M. Wirsing*, Eds., *L.N.C.S.*, 294, 1988, pp. 136-149.
6. U. Heuter, Zur Klassifizierung regulärer Baumsprachen, Dissertation an der RWTH, Aachen, 1989.
7. U. Heuter and D. Niwinski, A note on starfree tree languages (to appear), 1989.
8. R. McNaughton and S. Papert, "Counter-free Automata", *M.I.T.-Press*, Cambridge, Mass., 1971.
9. A. R. Meyer, A note on star-free events, *J. Assoc. Comput. Mach.*, 16, 1969, pp. 220-225.
10. J. G. Rosenstein, "Linear Orderings", Academic Press, New York, 1982.
11. M. P. Schützenberger, On monoids having only nontrivial subgroups, *Inform. Contr.*, 8, 1965, pp. 190-194.
12. W. Thomas, Classifying regular events in symbolic logic, *J. of Comput. and System Sci*, 25, 1982, pp. 360-376.
13. W. Thomas, Logical aspects in the study of tree languages, Ninth colloquium on trees in algebra and programming, *B. Courcelle* Ed., Cambridge Univ. Press, 1984, pp. 31-51.
14. J. W. Thatcher and J. B. Wright, Generalized finite automata with an application to a decision problem of second-order logic, *Math. Syst. Theory*, 2, 1968, pp. 57-82.