Ordering Constraints over Feature Trees Expressed in Second-order Monadic Logic

Martin Müller Joachim Niehren
Programming Systems Lab, Universität des Saarlandes
66041 Saarbrücken, Germany
{mmueller,niehren}@ps.uni-sb.de

August 6, 1998

Abstract

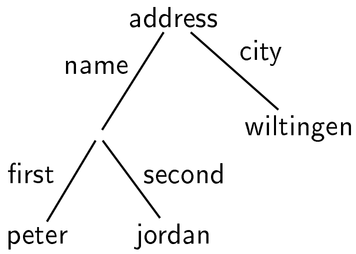
The system $FT_{\leq}$ of ordering constraints over feature trees has been introduced as an extension of the system FT of equality constraints over feature trees. While the first-order theory of FT is well understood, only few complexity and decidability results are known for fragments of the first-order theory of $FT_{\leq}$. We introduce a new handle for such decidability questions by showing how to express ordering constraints over feature trees in second-order monadic logic (S2S or WS2S). Our relationship implies a new decidability result for feature logics, namely that the entailment problem of $FT_{\leq}$ with existential quantifiers $\varphi_1 \models \exists x_1 \ldots \exists x_n \varphi_2$ is decidable. We also show that this problem is PSPACE-hard even though the quantifier-free case can be solved in cubic time. To our knowledge, this is the first time that a non-trivial decidability result of feature logic is reduced to Rabins famous tree theorem.
Keywords Feature logic, tree orderings, entailment, decidability, complexity, second-order monadic logic.

1 Introduction

Feature logic is a formalism to describe objects by the values of their attributes or *features*. It has its origin in the three areas of knowledge representation, with *concept descriptions*, *frames*, or ψ-terms (Brachman & Levesque, 1984; Aït-Kaci, 1986; Nebel, 1990; Nebel & Smolka, 1990), natural language processing, in particular approaches based on *unification grammars* (Kay, 1979; Kaplan & Bresnan, 1982; Shieber, Uszkoreit, Pereira, Robinson, & Tyson, 1983; Shieber, 1986; Pollard & Sag, 1994; Rounds, 1997), and constraint programming languages with record structures (Aït-Kaci & Nasr, 1986; Mukai, 1988; Aït-Kaci & Podelski, 1993; Smolka, 1995).

In the literature, two main approaches to feature logics can be distinguished. They both rely on similar first-order languages based on some kind of feature constraints, but differ significantly in the choice of semantics. In the mainstream approach followed in computational linguists (Pollard & Sag, 1994; Carpenter, 1992), feature constraints are typically interpreted over so-called feature structures. We follow Smolka's approach (Aït-Kaci, Podelski, & Smolka, 1994; Backofen & Smolka, 1995; Backofen, 1994, 1995) where feature constraints are interpreted in the structure of feature trees.

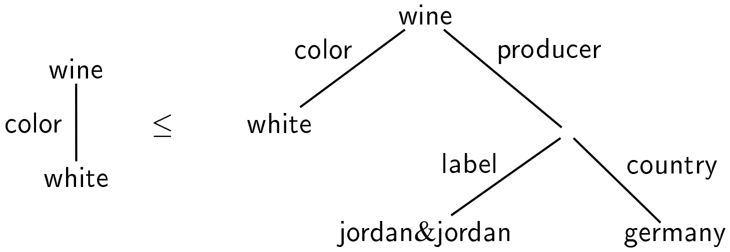
Feature Trees. A feature tree is a tree with unordered edges labeled by features and with optionally labeled nodes. Features are functional in that all features labeling edges that depart from the same node are pairwise distinct. As an example, consider the following feature tree which records a part of the address of Peter Jordan in Wiltingen:



More formally, a feature tree is defined by a tree domain, which is a prefix-closed non-empty set of words over features, and a partial labeling function from its tree domain to a set of labels. For instance, the tree domain of the above example tree is $\{\varepsilon, \mathsf{name},\ \mathsf{name\ first}, \mathsf{name\ second},\ \mathsf{city}\}$. Note also, that the node at path name is unlabeled, *i.e.*, the labeling function is undefined for this element of the tree domain.

Weak Subsumption Ordering. The feature tree semantics leads naturally to the weak subsumption ordering (Dörre, 1991; Müller, Niehren, & Podelski, 1997). A related ordering relation called strong subsumption (Dörre & Rounds, 1992) may, in contrast, appear more natural for feature structures; see the paragraph on feature structures below. In this paper, our focus is on weak subsumption.

Intuitively, a feature tree τ_1 is smaller than a feature tree τ_2 with respect to the weak subsumption ordering if τ_1 can be obtained from τ_2 by removing edges and node labels. More precisely, this means that the tree domain of τ_1 is a subset of the tree domain of τ_2, and that the (partial) labeling function of τ_1 is contained in the labeling function of τ_2. In this case we write $\tau_1 \leq \tau_2$. An example is given in the picture.



Ordering Constraints over Feature Trees. We investigate the system $FT_\leq$ of ordering constraints over feature trees (Müller et al., 1997; Müller, to appear; Müller & Niehren, 1998; Müller, Niehren, & Treinen, 1998). The constraints φ of $FT_\leq$ are defined from variables x, x', features f and labels a according to the following abstract syntax:

$$\varphi \ ::= \ x \leq x' \ \mid \ x[f]x' \ \mid \ a(x) \ \mid \ \varphi \wedge \varphi'$$

The semantics of $FT_\leq$ is given by the interpretation in the structure of feature trees where the symbol $\leq$ is interpreted as weak subsumption ordering. The semantics of feature selection $x[f]y$ and labeling constraints $a(x)$ is defined as usual. For instance, both trees in the picture above are possible denotations for x in solutions of the constraint $\mathsf{wine}(x) \wedge x[\mathsf{color}]x' \wedge \mathsf{white}(x')$. We consider two cases: Either we interpret constraints in the structure of possibly infinite feature trees or in the structure of finite trees. The particular choice will be made explicit whenever it matters.

The system $\mathrm{FT}_{\leq}$ is an extension of the system FT of equality constraints over feature trees (Aït-Kaci et al., 1994; Backofen & Smolka, 1995). The syntax of FT coincides with the syntax of $\mathrm{FT}_{\leq}$ except that FT provides for equalities $x{=}y$ instead of ordering constraints $x{\leq}y$. The semantics of feature selection and labeling constraints in FT are the same as in $\mathrm{FT}_{\leq}$. Equalities are expressible in $\mathrm{FT}_{\leq}$ since the equivalence $x{=}y \leftrightarrow x{\leq}y \wedge y{\leq}x$ is valid in $\mathrm{FT}_{\leq}$.

Decidability and Complexity. The first-order theory of FT is well-known to be decidable (Backofen & Smolka, 1995) but to have non-elementary complexity (Vorobyov, 1996). Several of its fragments can be decided in quasi-linear time (Smolka & Treinen, 1994), including the satisfiability problem for FT and the entailment problem with existential quantification $\varphi \models \exists x_1 \ldots \exists x_n \varphi'$. Much less is known on the first-order theory of $\mathrm{FT}_{\leq}$. Previously, the entailment problem $\varphi \models \varphi'$ of $\mathrm{FT}_{\leq}$ was shown to have cubic time complexity (Müller et al., 1997), but the decidability question for more expressive fragments of the first-order theory of $\mathrm{FT}_{\leq}$ was left open.

We consider the entailment problem $\varphi \models \exists x_1 \ldots \exists x_n \varphi'$ for $\mathrm{FT}_{\leq}$ with existential quantifiers. The proof technique used in (Müller et al., 1997) does not generalize to the case with existential quantifiers, since it is based on the independence property (Colmerauer, 1984) of $\mathrm{FT}_{\leq}$ constraints (if $\varphi \models_{\mathrm{FT}_{\leq}} \bigvee_{i=1}^{n} \varphi_i$ then there exists i, $1 \leq i \leq n$, such that $\varphi \models_{\mathrm{FT}_{\leq}} \varphi_i$). When existential quantifiers are admitted in disjunctions, independence fails. To see this, notice that the entailment judgement

$$x{\leq}y \wedge a(y) \;\models\; a(x) \vee \exists z(x{\leq}z \wedge b(z))$$

is valid in $\mathrm{FT}_{\leq}$ for distinct labels a and b, while the constraint $x{\leq}y \wedge a(y)$ does not entail one of the disjuncts $a(x)$ or $\exists z(x{\leq}z \wedge b(z))$ separately.

We show that the entailment problem $\varphi \models \exists x_1 \ldots \exists x_n \varphi'$ for $\mathrm{FT}_{\leq}$ is PSPACE-hard by reduction to the inclusion problem of non-deterministic finite automata. The inclusion problem for non-deterministic finite automata $\mathcal{A}$ and $\mathcal{A}'$ is the problem whether $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$ holds for the languages recognized by $\mathcal{A}$ and $\mathcal{A}'$. Our reduction applies to both, the finite tree case and the infinite tree case. For the purpose of this introduction, however, we discuss a simpler reduction which only applies to the case of possibly infinite feature trees.

For reducing the inclusion problem of non-deterministic finite automata to the entailment problem of $\mathrm{FT}_{\leq}$ with existential quantifiers, we represent a given finite automaton $\mathcal{A}$ through an existential $\mathrm{FT}_{\leq}$ formula $\Phi_{\mathcal{A}}(z)$ with a single free variable z such that the following equivalence holds for all finite automata $\mathcal{A}$ and $\mathcal{A}'$:

$$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}') \qquad \text{iff} \qquad \Phi_{\mathcal{A}'}(z) \models \Phi_{\mathcal{A}}(z)$$

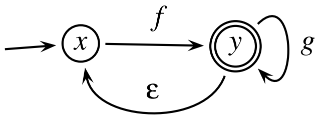
In the case of infinite trees, a possible definition for $\Phi_{\mathcal{A}}$ can be given as follows: We fix a label a and define the feature tree $\tau_{\mathcal{A}}$ to be the least feature tree whose labeling function $\mathcal{L}_{\tau_{\mathcal{A}}}$ satisfies $\mathcal{L}_{\tau_{\mathcal{A}}}(\pi) = a$ for exactly the paths π in the language recognized by

$\mathcal{A}$. Note that $\tau_{\mathcal{A}}$ is infinite if $\mathcal{L}(\mathcal{A})$ is infinite. Let $\mathcal{P}_{\mathcal{A}}$ be the unary predicate on feature trees such that for all τ:

$$\mathcal{P}_{\mathcal{A}}(\tau) \qquad \text{iff} \qquad \tau_{\mathcal{A}} \leq \tau$$

The essential idea is now to define the formula $\Phi_{\mathcal{A}}(z)$ such that it expresses the predicate $\mathcal{P}_{\mathcal{A}}$ in the sense of (Backofen, 1994), *i.e.*, such that $\mathcal{P}_{\mathcal{A}}(\tau)$ holds exactly if τ is the denotation of z in some solution of $\Phi_{\mathcal{A}}(z)$. We illustrate the definition of $\Phi_{\mathcal{A}}(z)$ for the following automaton $\mathcal{A}$ with alphabet $\{f, g\}$ and states x, y:

$\mathcal{A}$:



We identify the states of the automaton $\mathcal{A}$ with existentially bound variables of $\Phi_{\mathcal{A}}(z)$ which we define as follows:

$$\begin{array}{rcll} \Phi_{\mathcal{A}}(z) & = & \exists x \exists y (& z \leq x \wedge \\ & & & \exists x' \, (x'[f]y \wedge x' \leq x) \wedge \\ & & & \exists y' \, (y'[g]y \wedge y' \leq y) \wedge \\ & & & x \leq y \ \wedge \ a(y) \ ) \end{array}$$

The formula $\Phi_{\mathcal{A}}(z)$ reflects the fact that state x in $\mathcal{A}$ is initial by the ordering constraint $x \leq z$, and the fact that y is final by the labeling constraint $a(y)$. The transition rule $x \stackrel{f}{\to} y$ is represented through $\exists x' \, (x'[f]y \wedge x' \leq x)$, the rule $y \stackrel{g}{\to} y$ by $\exists y' \, (y'[f]y \wedge y' \leq y)$, and the rule $y \stackrel{\varepsilon}{\to} x$ by the constraint $x \leq y$.

Relationship to Second-Order Monadic Logic. The main goal of this paper is to investigate decidable fragments of the first-order theory of $\mathrm{FT}_{\leq}$. To this end, we pursue a general approach: We show how to express ordering constraints over feature trees in second-order monadic logic, WS2S or S2S, which are both known to be decidable. These results follow from the decidability of the emptiness problems or tree automata (Thatcher & Wright, 1968; Doner, 1970; Gecseg & Steinby, 1984; Comon et al., 1998) and Rabin automata for infinite trees (Rabin, 1969; Thomas, 1990, 1997), respectively.

We prove the decidability of the entailment problem of $\mathrm{FT}_{\leq}$ with existential quantifiers $\varphi \models \exists x_1 \ldots \exists x_n \varphi'$ under the assumption of a countably infinite set of features and a finite set of node labels. In the case of finite trees, we give a reduction to the weak second-order monadic logic with countably many successors WSωS which is expressible in WS2S (Thatcher & Wright, 1968), the weak second-order monadic with two successor. In the case of possibly infinite trees we reduce to the full second-order monadic logic with countably many successors SωS which, in analogy, is expressible in S2S (Rabin, 1969), the full second-order monadic logic with two successors.

We express feature constraints in second-order monadic logic according to the well-known idea to identify a tree with a set of words; for instance, an unlabeled feature can be identified with of its tree domain which is a prefix closed set. This idea has already been used for encoding other tree logics in (W)S2S (for instance, see (Comon, 1995)). To our knowledge however, there exists no proof of a non-trivial decidability result in feature logics by reduction into (W)S2S. In particular, no reduction of the first-order theory of FT to (W)S2S is known. It also seems difficult to express the simple first-order equations like $x = f(y, f(y, y))$ in terms of automata (Comon et al., 1998). When considering trees as sets of words, the problem is that it seems difficult if not impossible to express simultaneously prefix-closedness (of tree domains) and feature selection $\tau[f]\tau'$. Prefix-closedness can be expressed in a logic that provides concatenation πf of words with letters to the right, whereas feature selection needs concatenation $f\pi$ of words with letters to the left. The logic (W)S2S provides only one of the two forms of concatenation operations and becomes undecidable when extended with the missing one (see for instance (Thomas, 1990), Theorem 11.6).

The new observation underlying this paper is that there exists a language of feature constraints which can be encoded into (W)S2S without expressing prefix-closedness: We call this constraint system $FT_{\leq}^{-}$. In fact, we show how to encode the first-order theory of $FT_{\leq}^{-}$ in linear time into (W)S2S, and vice versa.[1] The syntax of $FT_{\leq}^{-}$ coincides with that of $FT_{\leq}$ and its semantics is obtained by restricting its semantic domain to so-called *sufficiently labeled* feature trees. We call a feature tree sufficiently labeled if its tree domain equals the prefix-closure of the set of its labeled nodes. Thus, a finite tree is sufficiently labeled if and only if all its leaves are labeled. In case of infinite trees, for instance, the tree with domain f^* and labeling function $\{(f^n, a) \mid n \text{ even}\}$ is sufficiently labeled. Most importantly, this tree uniquely corresponds to the set of its a-labeled nodes $\{f^n \mid n \text{ even}\}$, provided sufficient labeling is assumed.

This discussion shows that we can prove a fragment of the first-order theory of $FT_{\leq}$ decidable by reducing it to the first-order theory of $FT_{\leq}^{-}$. We present such an encoding for the entailment problem of $FT_{\leq}$ with existential quantifiers (which is essentially the identity function). The correctness proof for our encoding is the technically most involved part of this article.

Recent Developments. Most of the presented results have been first published in the earlier conference version of this article (Müller & Niehren, 1998). Compared to this conference version, full proofs and two new results have been added.

1. The PSPACE-hardness result for the entailment problem of $FT_{\leq}$ with existential quantifiers in the case of finite trees. In (Müller & Niehren, 1998), we could only prove coNP-hardness; PSPACE-hardness has been first shown in (Müller et al., 1998) but the encoding given here is slightly more direct in that it uses non-deterministic finite automata instead of regular expressions.

2. The result that (W)S2S can be encoded into the first-order theory of $FT_{\leq}$ or $FT_{\leq}^{-}$.

The presented work leaves at least two questions open which have meanwhile been answered in a follow-up paper (Müller et al., 1998).

3. Is the first-order theory of $FT_{\leq}$ decidable or even expressible in (W)S2S?

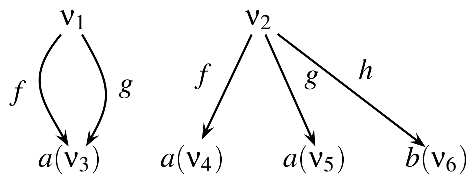
4. What is the precise complexity of the entailment problem of $FT_{\leq}$ with existential quantifiers?

The answer of question 3 is: no, the first-order theory of $FT_{\leq}$ is undecidable! The answer to question 4 is that the entailment problem of $FT_{\leq}$ with existential quantifiers is PSPACE-complete.

[1] The inverse encoding has been found in collaboration with Ralf Treinen.

Feature Structures versus Feature Trees. In the mainstream approach followed in computational linguists (Pollard & Sag, 1994; Carpenter, 1992), feature constraints are typically interpreted over so-called feature structures. In this context, two ordering relations have been introduced (Dörre, 1993): weak subsumption (Dörre, 1991) and strong subsumption (Dörre & Rounds, 1992). For sake of comparison, we illustrate the notion of feature structures and their orderings.

A feature structure is a graph-like logical structure with edges labeled by features and labeled nodes. For instance, consider the following feature structure $\mathcal{N}$ with nodes $\nu_1, \ldots, \nu_6$:



For every node ν of $\mathcal{N}$ let $\mathtt{tree}_{\mathcal{N}}(\nu)$ be the feature tree obtained by unfolding the graph $\mathcal{N}$ starting at ν and $\mathtt{graph}_{\mathcal{N}}(\nu)$ be the subgraph of $\mathcal{N}$ reachable from ν.

We can now give definitions for weak and strong subsumption on $\mathcal{N}$ which are equivalent to the usual definition in the literature. For all nodes ν, ν' of $\mathcal{N}$ we say that $\nu \leq_{\text{weak}} \nu'$ iff $\mathtt{tree}_{\mathcal{N}}(\nu) \leq \mathtt{tree}_{\mathcal{N}}(\nu')$ holds with respect to the ordering on feature trees considered in this paper. For instance, $\nu_1 \leq_{\text{weak}} \nu_2$ holds in $\mathcal{N}$. The strong subsumption ordering $\nu \leq_{\text{strong}} \nu'$ holds if and only if $\mathtt{graph}_{\mathcal{N}}(\nu)$ can be homomorphically embedded into $\mathtt{graph}_{\mathcal{N}}(\nu')$. For instance, $\nu_3 \leq_{\text{strong}} \nu_4$ but not $\nu_1 \leq_{\text{strong}} \nu_2$.

Plan of the Paper. Section 2 introduces the syntax and semantics of the constraint system $FT_{\leq}$, and Section 3 studies its expressiveness. Section 4 reviews the satisfiability test from (Müller et al., 1997), and Section 5 considers simple forms of entailment. Section 6 gives the lower bound complexity result for entailment with existential quantifiers. Section 7 defines second-order monadic logic and shows how to embed it into the first-order theories of $FT_{\leq}$. Section 8 gives our reduction of entailment in $FT_{\leq}$ to validity in SωS *resp.* WSωS and proves it correct. Section 9 summarizes.

2 Syntax and Semantics of $FT_{\leq}$

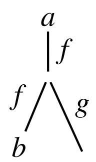
The constraint system $FT_{\leq}$ is a fragment of a first-order language. Its syntax is given by a set of ordering constraints and its semantics by interpretation in a structure of feature trees. We consider two cases, either the structure of possibly infinite feature trees or the structure of finite feature trees.

We assume an infinite set of *variables* ranged over by x, y, z, a countably infinite sets $\mathcal{F}$ of *features* ranged over by f, g, and a non-empty finite set $\mathcal{L}$ of *labels* ranged over by a, b.

2.1 Feature Trees

A *path* π is a word of features, *i.e.*$\pi \in \mathcal{F}$. The *empty path* is denoted by ε and the free-monoid concatenation of paths π and π' as $\pi\pi'$; we have $\varepsilon\pi = \pi\varepsilon = \pi$. Given paths π and π', π' is called a *prefix of* π if $\pi = \pi'\pi''$ for some path π''. A *tree domain* is a non-empty prefix closed set of paths. A *feature tree* τ is a pair (D, L) consisting of a tree domain D and a partial function $L : D \rightharpoonup \mathcal{L}$ that we call *labeling function* of τ. Given a feature tree τ, we write D_τ for its tree domain and L_τ for its labeling function. A feature tree is *finite* if its tree domain is finite, and *infinite* otherwise. A *node of* τ is an element of D_τ. The *root* of τ is the node ε. A *leaf of* τ is a maximal node of τ with respect to the prefix ordering. A node π of τ is *labeled with a* if $(\pi, a) \in L_\tau$. A node of τ is unlabeled if it is not labeled by any label a.

For instance, $\tau_0 = (\{\varepsilon, f, fg, ff\}, \{(\varepsilon, a), (ff, b)\})$ is a finite feature tree with domain $D_{\tau_0} = \{\varepsilon, f, fg, ff\}$ and labeling function $L_{\tau_0} = \{(\varepsilon, a), (ff, b)\}$.



The tree τ_0 has two leaves ff and fg. The root of τ_0 is labeled with a and its nodes f and fg are unlabeled.

The set of features occuring in some feature tree τ is denoted by $\mathcal{F}(\tau)$, i.e. $\mathcal{F}(\tau) = \{f \mid \pi f \pi' \in D_\tau\}$. If $\pi \in D_\tau$ we write as $\tau[\pi]$ the subtree of τ at path π which is formally defined by $D_{\tau[\pi]} = \{\pi' \mid \pi\pi' \in D_\tau\}$ and $L_{\tau[\pi]} = \{(\pi', a) \mid (\pi\pi', a) \in L_\tau\}$.

2.2 The Structures $\mathrm{FT}_\leq$ and $\mathrm{FT}_\leq^{fin}$

We consider two case, the structure of possibly infinite feature trees $\mathrm{FT}_\leq$ (overloading the names of the structure $\mathrm{FT}_\leq$ and the constraint system $\mathrm{FT}_\leq$) and the structure of finite feature trees $\mathrm{FT}_\leq^{fin}$. The domain of the structure $\mathrm{FT}_\leq$ is the set of feature trees built from features in $\mathcal{F}$ and labels in $\mathcal{L}$. Its signature consists of the set of binary relation symbols $\{[f] \mid f \in \mathcal{F}\} \cup \{\leq\}$ and the set of unary relation symbols $\mathcal{L}$. These relation symbols are interpretation as the following relations between feature trees. For all τ, τ_1, τ_2, we define:

$$\begin{array}{rll} \tau_1 \leq \tau_2 & \text{iff} & D_{\tau_1} \subseteq D_{\tau_2} \text{ and } L_{\tau_1} \subseteq L_{\tau_2} \\ \tau_1[f]\tau_2 & \text{iff} & D_{\tau_2} = \{\pi \mid f\pi \in D_{\tau_1}\} \text{ and } L_{\tau_2} = \{(\pi, a) \mid (f\pi, a) \in L_{\tau_1}\} \\ a(\tau) & \text{iff} & (\varepsilon, a) \in L_\tau \end{array}$$

The structure $\mathrm{FT}_\leq^{fin}$ is the restriction of the structure $\mathrm{FT}_\leq$ to the domain of finite feature trees.

2.3 Ordering Constraints and First-Order Formulas

An *ordering constraint* φ of $\mathrm{FT}_\leq$ is defined by the following abstract syntax:

$$\varphi \quad ::= \quad x \leq y \mid a(x) \mid x[f]y \mid \varphi_1 \wedge \varphi_2$$

An ordering constraint is a conjunction of *basic constraints* which are either *basic ordering constraints* $x \leq y$, *labeling constraints* $a(x)$, or *selection constraints* $x[f]y$. We write $\bar{x}$ for a possibly empty word of variables $x_1 \ldots x_n$ and $\exists \bar{x} \varphi$ instead of $\exists x_1 \ldots \exists x_n \varphi$.

We denote first-order formula built from ordering constraints plus the usual first-order connectives with Φ.

A *variable assignment* into the structure $\mathrm{FT}_{\leq}$ (*resp.*, $\mathrm{FT}_{\leq}^{fin}$) is a function α mapping variables to feature trees (*resp.*, finite feature trees). The truth value of a first-order formula under a variable assignment α into $\mathrm{FT}_{\leq}$ (*resp.*, $\mathrm{FT}_{\leq}^{fin}$) is defined as usual for first-order languages. A *solution* of Φ is a variable assignment into $\mathrm{FT}_{\leq}$ (*resp.*, $\mathrm{FT}_{\leq}^{fin}$) which makes Φ true. We call Φ *satisfiable* if there exists a solution of Φ and *valid* if every variable assignment is a solution of Φ. We say that Φ *entails* Φ', written $\Phi \models \Phi'$, if $\Phi \to \Phi'$ is valid, and that Φ is *equivalent* to Φ' if $\Phi_1 \leftrightarrow \Phi_2$ is valid. We denote with $\mathcal{V}(\Phi)$ the set of variables occurring free in Φ and with $\mathcal{F}(\Phi)$ the set of features occurring in Φ.

2.4 Alternative Definitions of Feature Trees

In the literature, there are two alternative definitions of feature trees distinct from ours (Aït-Kaci et al., 1994; Backofen, 1995). According to (Aït-Kaci et al., 1994), every node must be labeled, and (Backofen, 1995) requires exactly the leaves to be labeled. In contrast, we follow previous work of ours and *allow* labels at all nodes but do *not require* any (Müller et al., 1997).

For equality constraints as in FT, the particular definition of feature trees does not matter. The reason is that the first-order theory of FT is completely axiomatizable (Backofen & Smolka, 1995). Each definition of feature trees yields a model of the axiomatization of FT. All these models are distinct but their first-order theories coincide due to complete axiomatization.

With respect to ordering constraints as in $\mathrm{FT}_{\leq}$ the particular definition of feature trees does matter. For example, let us consider the formulas Φ_1 and Φ_2 with two labels $a \neq b$ (in case where at least that two distinct labels exist):

$$\begin{array}{rcl} \Phi_1 & = & \exists y(x{\leq}y \wedge a(y)) \wedge \exists z(x{\leq}z \wedge b(z)) \\ \Phi_2 & = & \exists x \forall y\, x{\leq}y \end{array}$$

The formula Φ_1 says that the root of the denotation of x is unlabeled since $a \neq b$. Thus, Φ_1 is satisfiable in $\mathrm{FT}_{\leq}$, but not in a structure of feature trees where every node must be labeled. The formula Φ_2 says that there exists a least feature tree with respect to the weak subsumption ordering. Such a tree exists in $\mathrm{FT}_{\leq}$, namely $(\{\varepsilon\}, \emptyset)$. However, there is no smallest tree in structures of feature trees that require all nodes (or all leaves) to be labeled. Thus, Φ_2 distinguishes the structure $\mathrm{FT}_{\leq}$ from those proposed for FT in (Aït-Kaci et al., 1994; Backofen, 1995; Backofen & Smolka, 1995).

The failure of independence for $\mathrm{FT}_{\leq}$ with existential quantifiers does not depend on which particular structure of feature trees is chosen. However, the counter example against independence given in the introduction

$$x{\leq}y \wedge a(y) \;\models\; a(x) \vee \exists z(x{\leq}z \wedge b(z))$$

does not work for the structures of feature trees that are completely labeled or labaled exactly at the leaves. Although in both cases the entailment holds, this does not contradict independence since the stronger judgement

$$x{\leq}y \wedge a(y) \;\models\; a(x)$$

is also valid. For completely labeled feature trees this is clear, and for feature trees with labeled leaves notice that $x \leq y \wedge a(y)$ implies that x is a leaf and, hence, labeled with a .

However, the following counter example works in both cases, as we leave for the reader to check. In this example, we use $x[f]a$ as an abbreviation for the existential formula $\exists x' \, (x[f]x' \wedge a(x'))$.

$$x \leq y \wedge y[f]a \ \models \ x[f]a \vee \exists z(x \leq z \wedge z[f]b)$$

3 Expressiveness

We present several predicates on feature trees and show how to express them in the first-order theory of $\mathrm{FT}_{\leq}$. This section is meant to illustrate the first-order theory of $\mathrm{FT}_{\leq}$, but most of the predicates discussed here are needed in the proofs to come.

An *n-ary predicate* $\mathcal{P}$ is an n-ary relation between (finite) feature trees. We write $\mathcal{P}(\tau_1, \ldots, \tau_n)$ if $(\tau_1, \ldots, \tau_n) \in \mathcal{P}$. We denote a formula Φ with free variables $x_1, \ldots, x_n$ with $\Phi(x_1, \ldots, x_n)$ whereby an ordering on the variables of Φ is fixed. We recall the definition of a formula expressing a predicate from (Backofen, 1994).

Definition 3.1 *An n-ary predicate $\mathcal{P}$* is expressed by a formula $\Phi(x_1, \ldots, x_n)$ *with free variables $x_1, \ldots, x_n$ if for all (finite) feature trees $\tau_1, \ldots, \tau_n$:*

$$\mathcal{P}(\tau_1, \ldots, \tau_n) \ \textit{ iff exists a solution } \alpha \textit{ of } \Phi(x_1, \ldots, x_n) \textit{ with } \alpha(x_1) = \tau_1, \ldots, \alpha(x_n) = \tau_n$$

Of course, this definition depends on the choice of one of the structures $\mathrm{FT}_{\leq}$ or $\mathrm{FT}^{fin}_{\leq}$.

3.1 Compatibility

A predicate that is central for the satisfiability problem of $\mathrm{FT}_{\leq}$ is the label compatibility predicate $\tau \sim a$ which holds if the root of τ is unlabeled or labeled with a. This predicate can be expressed through the formula $x \sim a$ defined as follows:

$$x \sim a \ = \ \exists y(x \leq y \wedge a(y))$$

Our satisfiability test also uses a binary compatibility predicate $\tau \sim \tau'$ which holds if for all path π and labels a, b, $(\pi, a) \in \mathcal{L}(\tau)$ and $(\pi, b) \in \mathcal{L}(\tau')$ imply $a = b$. We express this predicate by the formula $x \sim y$ defined by:

$$x \sim y \ = \ \exists z(x \leq z \wedge y \leq z)$$

3.2 Path Predicates and Path Constraints

Throughout the paper, we will need a collection of predicates that describe feature trees by their paths. These predicates can be expressed by positive existential formulas of $\mathrm{FT}_{\leq}$ that we call *path constraints*.

Let $\pi \in \mathcal{F}^*$ be a path and a a label. The predicate $\tau[\pi]{\downarrow}$ holds iff $\pi \in D_\tau$. The predicate $a(\tau[\pi])$ is valid iff $(\pi, a) \in L_\tau$, and the predicate $\tau \leq \tau'[\pi]$ holds if $\pi \in \mathcal{D}_{\tau'}$ and τ is smaller then $\tau'[\pi]$.

For the construction of a least solutions we will use path constraints $x[\pi]\downarrow$, $a(x[\pi])$, and $y \leq x[\pi]$ which express the predicates $\tau[\pi]\downarrow$, $a(\tau[\pi])$ and $\tau' \leq \tau[\pi]$. For their definition, we introduce the following auxiliary predicate $\tau[\pi]\tau'$ which holds iff $\pi \in \mathcal{D}_\tau$ and $\tau[\pi] = \tau'$. We express the predicate $\tau[\pi]\tau'$ by the following formula $x[\pi]y$ which generalizes the selection constraint $x[f]y$ from a single feature to arbitrary paths.

$$
\begin{array}{rcl}
x[\varepsilon]y & = & x \leq y \wedge y \leq x \\
x[\pi\pi']y & = & \exists z(x[\pi]z \wedge z[\pi']y)
\end{array}
$$

We can now define the formulas expressing $\tau[\pi]\downarrow$, $a(\tau[\pi])$, and $\tau \leq \tau'[\pi]$ by means of existential quantification:

$$
\begin{array}{rcl}
x[\pi]\downarrow & = & \exists y(x[\pi]y) \\
a(x[\pi]) & = & \exists y(x[\pi]y \wedge a(y)) \\
x \leq y[\pi] & = & \exists z(x \leq z \wedge y[\pi]z)
\end{array}
$$

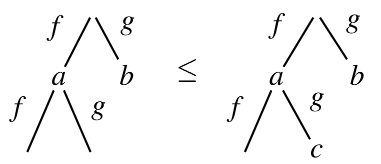
In our PSPACE-hardness proof, we will talk about another predicate $\tau?[\pi] \leq \tau'$, which holds if either $\pi \notin \mathcal{D}_\tau$ or if the subtree $\tau[\pi]$ is smaller than τ'. This predicate can be expressed by the following path constraint $x?[\pi] \leq y$:

$$
x?[\pi] \leq y \;=\; \exists z(x \leq z \wedge z[\pi]y)
$$

We will also need a predicate $\textsf{all-compat}_{F^*}(\tau)$ for some finite set of features $F \subseteq \mathcal{F}$ which holds if, for all $\pi, \pi' \in F^*$, either $\pi \notin \mathcal{D}_\tau$ or $\pi' \notin \mathcal{D}_{\tau'}$ or $\tau[\pi] \sim x[\pi']$. We express this predicate through the following formula:

$$
\textsf{all-compat}_{F^*}(x) \;= \exists y(\; x \leq y \;\wedge\; \exists z(y \leq z \wedge \bigwedge_{f \in F} z[f]y))
$$

For illustration consider the following feature trees τ_1 on the left and τ_2 on the right



and notice that $\textsf{all-compat}_{f^*}(\tau_1)$ holds while $\textsf{all-compat}_{f^*}(\tau_2)$ does not hold.

Notice also that $\textsf{all-compat}_{F^*}(\tau)$ implies that the set $\{a \mid (\pi, a) \in \mathcal{L}_\tau,\ \pi \in F^*\}$ is either empty or a singleton. To see this, assume the contrary, namely that there exists $\pi, \pi' \in F^*$ and $a \neq b$ such that $(\pi, a), (\pi, b) \in L_\tau$. Obviously, this implies the incompatibility of $\tau[\pi]$ and $\tau'[\pi']$ and therefore $\neg\textsf{all-compat}_{F^*}(\tau)$. In contrast, $\textsf{all-compat}_{F^*}(\tau)$ does *not* say that the subtrees $\tau[\pi]$ at paths $\pi \in F^*$ are uniformly labeled, as the example above shows.

Another interesting observation is that the following equivalence holds:

$$
\textsf{all-compat}_{F^*}(x) \;\models\!\!\!\!\!\;\dashv\; \exists z(\bigwedge_{\pi \in F^*} x?[\pi] \leq z)
$$

The direction from right to left is trivial. The direction from left to right can be proved by a contradiction argument similar to the one above.

3.3 Sufficiently Labeled Trees

As argued in the introduction, the restriction of $FT_{\leq}$ to so-called sufficiently labeled feature trees plays a crucial rôle in the relation between feature logics and second-order monadic logic with two successors.

Definition 3.2 *We call a feature tree* τ sufficiently labeled *if for every* $\pi \in D_\tau$ *there exists a path* π' *and a label a such* $(\pi\pi', a) \in L_\tau$.

A feature tree τ is sufficiently labeled if and only if its tree domain coincides with the prefix closure of the set of its labeled nodes, *i.e.*:

$$\mathcal{D}_\tau = \{\pi \mid \text{exists } \pi' \text{ and } a \text{ such that } \pi \text{ is a prefix of } \pi' \text{ and } (\pi', a) \in \mathcal{L}_\tau\}$$

Note that a finite feature tree is sufficiently labeled if and only if all its leaves are labeled. In case of infinite feature trees this does not necessarily hold. For instance the tree $(f^*, \emptyset)$ is not sufficiently labeled even though all its leaves (*i.e.*, none) are labeled.

Notice also that a sufficiently labeled feature tree (finite or infinite) contains at least one label.

Let the predicate $\mathsf{suff\text{-}lab}(\tau)$ hold if τ is sufficiently labeled. Even this predicate can be expressed in the first-order theory of $FT_{\leq}$, for instance by the following formula $\mathsf{suff\text{-}lab}(x)$.

$$\mathsf{suff\text{-}lab}(x) \quad = \quad \forall y\ (\forall z\ (x{\sim}z \leftrightarrow y{\sim}z) \rightarrow x{\leq}y)$$

This formula requires the denotation τ_x of x to be smaller than all trees which are compatible with the same trees as τ_x.

4 Satisfiability and Least Solutions

In (Müller et al., 1997), we have given a satisfiability test for $FT_{\leq}$ and $FT_{\leq}^{fin}$. This test also computes the least solution of a satisfiable constraint. The form of these least solutions can be interpreted as a criterion for entailment.

We call a constraint φ $\mathsf{F1}$-$\mathsf{F2}$-closed if it satisfies the following properties:

F1.1	$x{\leq}x$ *in* φ	if	$x \in \mathcal{V}(\varphi)$
F1.2	$x{\leq}z$ *in* φ	if	$x{\leq}y$ *in* φ and $y{\leq}z$ *in* φ
F2	$x'{\leq}y'$ *in* φ	if	$x[f]x'$ *in* φ, $x{\leq}y$ *in* φ and $y[f]y'$ *in* φ

Theorem 4.1 (Satisfiability) *There exists a cubic time algorithm that, given a constraint* φ*, either detects the unsatisfiability of* φ *or returns a satisfiable constraint* $\tilde{\varphi}$ *equivalent to* φ*, which is* $\mathsf{F1}$-$\mathsf{F2}$*-closed in addition and is a conjunction of* φ *with some basic ordering constraints* $x{\leq}y$ *where* $x, y \in \mathcal{V}(\varphi)$.

Proof. We first consider an extended set of rules that defines a satisfiability test. Let *false* and $x{\sim}y$ be auxiliary constraints where *false* denotes inconsistency and the semantics of $x{\sim}y$ is given by the equivalence $x_1{\sim}x_2 \leftrightarrow \exists x (x_1{\leq}x \wedge x_2{\leq}x)$. In the infinite tree case, we call a constraint φ F-*closed* if it satisfies the properties $\mathsf{F1}$–$\mathsf{F5}$ below. In the finite tree case, a constraint φ is called F-*closed* if it satisfies the properties $\mathsf{F1}$–$\mathsf{F6}$.

F3.1	$x \sim y$ *in* φ	if	$x \leq y$ *in* φ
F3.2	$x \sim z$ *in* φ	if	$x \leq y$ *in* φ and $y \sim z$ *in* φ
F3.3	$x \sim y$ *in* φ	if	$y \sim x$
F4	$x' \sim y'$ *in* φ	if	$x[f]x'$ *in* φ, $x \sim y$ *in* φ and $y[f]y'$ *in* φ
F5	*false in* φ	if	$a(x) \wedge x \sim y \wedge a'(y)$ *in* φ and $a \neq a'$
F6	*false in* φ	if	$x_1 \leq x_1' \wedge x_2[f]x_1' \wedge x_2 \leq x_2' \wedge x_3[f]x_2' \wedge \ldots \wedge x_n \leq x_1$ *in* φ

For every constraint φ an equivalent F-closed constraint φ' can be computed in cubic time by applying the rules in F exhaustively. Furthermore, φ is unsatisfiable if φ' contains *false*. Otherwise, φ is satisfiable as proved in (Müller et al., 1997). Deleting the auxiliary constraints $x_1 \sim x_2$ from φ' transforms φ' into an equivalent $\mathsf{F1}$-$\mathsf{F2}$-closed constraint. □

In order to describe the form of least solutions of satisfiable constraint, we recall an inference system also presented in (Müller et al., 1997) which allows to derive *syntactic entailment judgements* of the form $\varphi \vdash \tau[\pi]\!\downarrow$ and $\varphi \vdash a(x[\pi])$. We first present inference rules for auxiliary judgements $\varphi \vdash y \leq x[\pi]$:

$$\begin{array}{rll} \varphi \vdash y \leq x[\varepsilon] & \text{if} & y \leq x \textit{ in } \varphi \\ \varphi \vdash y \leq x[f] & \text{if} & x[f]y \textit{ in } \varphi \\ \varphi \vdash y \leq x[\pi_1\pi_2] & \text{if} & \text{exists } z : \varphi \vdash y \leq z[\pi_2] \text{ and } \varphi \vdash z \leq x[\pi_1] \end{array}$$

We lift these rules to an inference system for the judgements $\varphi \vdash a(x[\pi])$ and $\varphi \vdash x[\pi]\!\downarrow$:

$$\begin{array}{rll} \varphi \vdash x[\pi]\!\downarrow & \text{if} & \text{exists } z : \varphi \vdash z \leq x[\pi] \\ \varphi \vdash a(x[\pi]) & \text{if} & \text{exists } z : \varphi \vdash z \leq x[\pi] \text{ and } a(z) \textit{ in } \varphi \end{array}$$

Proposition 4.2 (Least Solutions) *If φ is satisfiable and* $\mathsf{F1}$-$\mathsf{F2}$*-closed then φ admits a least solution* least_φ *of the following form:*

$$\begin{array}{rcl} D_{\mathsf{least}_\varphi(x)} & = & \{\pi \mid \varphi \vdash x[\pi]\!\downarrow\} \\ L_{\mathsf{least}_\varphi(x)} & = & \{(\pi, a) \mid \varphi \vdash a(x[\pi])\} \end{array}$$

The satisfiability assumption for φ can also be weakened to that least_φ *maps every variable to a feature tree.*

Proof. We first note that syntactic entailment is correct with respect to semantic entailment. We can show by induction on derivations of the above inference rules that the following properties hold for all φ, x, y, π, a.

1. $\varphi \vdash x \leq y[\pi]$ implies $\varphi \models x \leq y[\pi]$.
2. $\varphi \vdash x[\pi]\!\downarrow$ implies $\varphi \models x[\pi]\!\downarrow$.
3. $\varphi \vdash a(x[\pi])$ implies $\varphi \models a(x[\pi])$.

Now consider a particular constraint φ which is satisfiable and $\mathsf{F1}$-$\mathsf{F2}$-closed. Since syntactic entailment is a correct and φ is satisfiable, least_φ maps every variable to a feature tree. Under this condition, least_φ is a solution of an $\mathsf{F1}$-$\mathsf{F2}$-closed constraint as proved in Lemma 25 of (Müller et al., 1997). Finally, the correctness of syntactic entailment also implies that least_φ is smaller than every other solution of φ. □

5 Simple Forms of Entailment

The proofs of our results on complexity and decidability for entailment of $FT_{\leq}$ with existential quantifiers will be based on results about simpler forms of entailment. All these results also follow from the more general follow-up paper (Müller et al., 1998). The purpose of this section collection is reformulate entailment results from (Müller et al., 1997; Müller & Niehren, 1998) such that they can be applied without reference to (Müller et al., 1998).

5.1 Least Solutions

The syntactic description of least solutions given in Proposition 4.2 implies the following useful criterion for entailment of path constraints of the form $x[\pi]\downarrow$ and $a(x[\pi])$.

Proposition 5.1 *Let φ be satisfiable and* F1*-*F2*-closed. For every variable $x \in \mathcal{V}(\varphi)$, and all a, π, z the following two equivalences hold:*

$$\begin{array}{rcl} \varphi \models x[\pi]\downarrow & \textit{iff} & \varphi \vdash x[\pi]\downarrow \\ \varphi \models a(x[\pi]) & \textit{iff} & \varphi \vdash a(x[\pi]) \end{array}$$

Proof. The implications from the right to the left follow from the correctness of syntactic entailment (compare proof of Proposition 4.2). The converse implication follows from the representation of least solutions stated in Proposition 4.2: Assume for instance $\varphi \models x[\pi]\downarrow$. Since φ is F1-F2-closed and satisfiable, the variable assignment least_φ is a solution of φ. Thus, $\mathsf{least}_\varphi(x)[\pi]\downarrow$ holds, *i.e.*, $\varphi \vdash x[\pi]\downarrow$. □

5.2 Entailment without Existential Quantification

Proposition 5.1 shows how to decide two forms of entailment for path constraints with some implicit existential quantification. In (Müller et al., 1997), we have shown how to decide entailment without quantification in general. This result has an interesting consequence that we will need later on.

Theorem 5.2 ((Müller et al., 1997)) *The entailment problem $\varphi \models \varphi'$ can be tested in cubic time. In particular, if φ is satisfiable and* $\mathsf{F1}-F2$*-closed, then:*

$$\varphi \models x{\leq}y \ \textit{ if and only if } \ x{\leq}y \text{ in } \varphi.$$

For the proof, we refer to (Müller et al., 1997).[2] We now generalize Theorem 5.2 to the more complex class of path constraints of the form $x?[\pi] \leq y$. We therefore extend the definition of *syntactic entailment* by judgements of the form $\varphi \vdash x?[\pi] \leq y$ and $\varphi \vdash \mathsf{all\text{-}compat}_{F^*}(x)$ as follows.

$$\begin{array}{rll} \varphi \vdash x?[\varepsilon] \leq y & \text{if} & x{\leq}y \textit{ in } \varphi \\ \varphi \vdash x?[f] \leq y & \text{if} & x[f]y \textit{ in } \varphi \\ \varphi \vdash x?[\pi_1\pi_2] \leq y & \text{if} & \text{exists } z \text{ such that } \varphi \vdash x?[\pi_1] \leq z \text{ and } \ \varphi \vdash z?[\pi_2] \leq y \end{array}$$

[2] In (Müller et al., 1997), we prove a slightly more general result that also deals with basic constraints of the form $a(x)$, $x[f]y$, and $x{\sim}y$.

Proposition 5.3 *Let* φ *be satisfiable and* F1-F2*-closed. Then the following equivalence holds:*

$$\varphi \models x?[\pi] \leq y \ \textit{ if and only if } \ \varphi \vdash x?[\pi] \leq y.$$

The proof of this proposition constitutes the rest of this section. It uses Theorem 5.2, which Proposition 5.3 actually generalizes due to the equivalence of $x?[\varepsilon] \leq y$ and $x{\leq}y$. The proof setup is the same as for the proof of the corresponding characterization statement in the full version of (Müller et al., 1998). We provide the argument here nonetheless for completeness' sake, and because the proof setup may be interesting in its own right. The proof is based on the notion of a constraint extension which serves for contradiction non-entailed constraints.

Definition 5.4 (Constraint Extension) *For all constraints* φ *and features* f *fix a fresh variable* x_f *and define the* extension φ^x_f of φ at x and f *as follows:*

$$\begin{array}{rcl}\varphi^x_f & = & \varphi \wedge x[f]x_f \wedge x_f{\leq}x_f \wedge \\ & & \bigwedge\{x_f{\leq}y \mid \varphi \vdash x?[f] \leq y\} \wedge \\ & & \bigwedge\{y{\leq}x_f \mid \varphi \vdash y \leq x[f]\}\end{array}$$

Lemma 5.5 *If* φ *is* F1-F2*-closed, then* φ^x_f *is* F1-F2*-closed for all* x *and* f.

The following two lemmas state technical properties of the constraint extension: Lemma 5.6 focuses on syntax, and Lemma 5.7 on semantics.

Lemma 5.6 *If* φ *is* F1-F2*-closed,* $x, y \in \mathcal{V}(\varphi)$*, and* $f \in \mathcal{F}(\varphi)$*. Then* $\varphi^x_f \vdash x_f?[\pi] \leq y$ *implies* $\varphi \vdash x?[f\pi] \leq y$.

Proof. By induction over π:

$\pi = \varepsilon$**:** If $\varphi^x_f \vdash x_f?[\varepsilon] \leq y$, then by definition of syntactic entailment and the F1-F2-closedness of φ^x_f there must exist a variable $z \neq x_f$ such that $x_f{\leq}z$ *in* φ^x_f and $z{\leq}y$ *in* φ. By definition of φ^x_f this implies $\varphi \vdash x?[f] \leq z$, and therefore $\varphi \vdash x?[f] \leq y$.

$\pi = g\pi'$**:** If $\varphi^x_f \vdash x_f?[\varepsilon] \leq y$, then by definition of syntactic entailment and the F1-F2-closedness of φ^x_f there must exist variables z, z' such that $x_f{\leq}z, z[g]z'$ *in* φ^x_f and $\varphi^x_f \vdash z'?[\pi'] \leq y$. Be definition of φ^x_f we know that $z, z' \neq x_f$ and hence that $z[f]z'$ *in* φ, and that $\varphi \vdash x?[f] \leq z$. Furthermore, the induction assumption implies that $\varphi \vdash z'?[\pi'] \leq y$. In combination this yields $\varphi \vdash x?[fg\pi'] \leq y$, *i.e.*, $\varphi \vdash x?[f\pi] \leq y$.

□

Lemma 5.7 $\varphi \models x?[f\pi] \leq y$ *implies* $\varphi^x_f \models x_f?[\pi] \leq y$.

Proof. $\varphi \models x?[f\pi] \leq y$ implies $\varphi \models \forall x_f(x[f]x_f \rightarrow x_f?[\pi] \leq y$, and therefore $\varphi^x_f \models x[f]x_f \wedge x_f?[\pi] \leq y$. □

Proof of Theorem 5.3. The direction from right to left follows from correctness of syntactic entailment for $x?[\pi] \leq y$. One shows by induction over π that for all x, y, φ,

- $\varphi \vdash x?[\pi] \leq y$ implies $\varphi \models x?[\pi] \leq y$.

For the inverse direction assume $\varphi \not\vdash x?[\pi] \leq y$. We show by induction over π that $\varphi \not\models x?[\pi] \leq y$.

$\pi = \varepsilon$: $\varphi \not\vdash x?[\varepsilon] \leq y$ implies that $x{\leq}y$ *not in* φ because φ is F1-F2-closed. Hence, by Proposition 5.2, $\varphi \not\models x{\leq}y$ and therefore, equivalently, $\varphi \not\models x?[\varepsilon] \leq y$.

$\pi = f\pi'$: If $\varphi \not\vdash x?[f\pi'] \leq y$ then $\varphi^x_f \not\vdash x_f?[\pi'] \leq y$ by Lemma 5.6. By induction assumption this implies that $\varphi^x_f \not\models x_f?[\pi'] \leq y$, and Lemma 5.7 implies $\varphi \not\models x?[f\pi'] \leq y$.

□

6 PSPACE-Hardness of Entailment with Existential Quantifiers

We consider the expressiveness of the entailment problem of $FT_{\leq}$ with existential quantification $\varphi \models \exists\bar{x}\varphi'$.[3]

Theorem 6.1 *The entailment problems with existential quantification $\varphi \models \exists\bar{x}\varphi'$ for $FT_{\leq}$ and of $FT^{fin}_{\leq}$ are PSPACE-hard.*

The remainder of this section develops the proof. In Section 6.1 we discuss the main idea, and in Section 6.2 we present a linear time reduction of the inclusion problem of non-deterministic finite automata to the entailment problem $\varphi \models \exists\bar{x}\varphi'$. Since the inclusion problem between regular word languages is PSPACE-complete (Hopcroft & Ullman, 1979; Seidl, 1990), we obtain PSPACE-hardness of entailment.

6.1 Main Ideas

For reducing the inclusion problem of non-deterministic finite automata to the entailment problem of $FT_{\leq}$ with existential quantifiers, we represent a given finite automaton $\mathcal{A}$ through an existential $FT_{\leq}$ formula $\Phi_{\mathcal{A}}(\bar{z})$ with a sequence of free variables $\bar{z}$ such that the following equivalence holds for all finite automata $\mathcal{A}$ and $\mathcal{A}'$:

$$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}') \qquad \text{iff} \qquad \Phi_{\mathcal{A}'}(\bar{z}) \models \Phi_{\mathcal{A}}(\bar{z})$$

In the case of infinite trees, a possible definition of $\Phi_{\mathcal{A}}$ has been proposed in the introduction. Given a label a, recall that we may define $\Phi_{\mathcal{A}}(z)$ for a single free variable z such that it expresses the predicate $\mathcal{P}$ which satisfies for all τ:

$$\mathcal{P}(\tau) \ \text{ iff } \ L_\tau \supseteq \{(\pi, a) \mid \pi \in \mathcal{L}(\mathcal{A})\}$$

This idea was sketched in the introduction and is proved in the conference version of this article (Müller & Niehren, 1998). In case of finite trees, the same reduction can

[3]Quantifiers on the left-hand side of entailment judgements can be used since $\exists\bar{y}\varphi \models \exists\bar{x}\varphi'$ is equivalent to $\varphi \models \exists\bar{x}\varphi'$ if $\{\bar{y}\} \cap \mathcal{V}(\exists\bar{x}\varphi') = \emptyset$. This can be ensured by α-conversion.

still be used for encoding the inclusion problem of finite automata which recognize a finite language. This problem is coNP-complete (Garey & Johnson, 1979).[4]

A PSPACE-hardness proof for the case of finite trees has been presented for a first time in (Müller et al., 1998). Here, we present a slight variation of this proof which relies on finite automata instead of regular expression. The main idea of both proofs is the same. It consists in defining a formula $\Phi^1_{\mathcal{A}}(z,z')$ which expresses the binary predicate $\mathcal{P}^1_{\mathcal{A}}$ such that for all τ and τ':

$$\mathcal{P}^1_{\mathcal{A}}(\tau,\tau') \quad \text{iff} \quad \text{for all } \pi \in \mathcal{L}(\mathcal{A}) : \tau?[\pi] \leq \tau'$$

In principle, this idea is fine except that we do not know how to express the predicate $\mathcal{P}^1_{\mathcal{A}}$ by a positive existential formula. We will therefore pick a slightly different predicate $\mathcal{P}^2_{\mathcal{A}}$ which satisfies for all τ and τ':

$$\mathcal{P}^2_{\mathcal{A}}(\tau,\tau') \quad \text{iff} \quad \mathcal{P}^1_{\mathcal{A}}(\tau,\tau') \text{ and } \mathsf{all\text{-}compat}_{F^*}(\tau)$$

where $F \subset \mathcal{F}$ is the finite alphabet of $\mathcal{A}$. As we will see, this predicate can be expressed.

6.2 Automata as Constraints

With every non-deterministic finite automaton $\mathcal{A}$ over F we now associate an existential formula $\Phi_{\mathcal{A}}$ as follows.

An automaton $\mathcal{A} = (Q, q_s, q_f, \delta)$ over F is given by a set Q of states, the initial and the final states q_s and q_f, respectively, and the transition relation δ of $\mathcal{A}$. We write $\mathcal{A} \vdash q \xrightarrow{\pi} q'$ if $\mathcal{A}$ allows for a transition from q to q' reading π (*i.e.*, if (q,π,q') is in the transitive closure δ). Furthermore we write as $\mathcal{L}(\mathcal{A})$ the language accepted by $\mathcal{A}$, *i.e.*, $\mathcal{L}(\mathcal{A}) = \{\pi \mid \mathcal{A} \vdash q_s \xrightarrow{\pi} q_f\}$.

Definition 6.2 (Automata as Constraints) *For a given automaton $\mathcal{A} = (Q, q_s, q_f, \delta)$, and arbitrary distinct variables z and z', we define the following existential formula $\Phi_{\mathcal{A}}(z,z')$ which uses the states as existentially quantified variables:*

$$\Phi_{\mathcal{A}}(z,z') \;=\; \exists Q (z \leq q_s \;\wedge\; q_f \leq z' \;\wedge\; \bigwedge_{(q,f,q')\in\delta} q?[f] \leq q' \;\wedge\; \mathsf{all\text{-}compat}_{F^*}(z))$$

We write as $\varphi_{\mathcal{A}}(z,z')$ the quantifier-free matrix of $\Phi_{\mathcal{A}}(z,z')$.

Lemma 6.3 *The formula $\Phi_{\mathcal{A}}(z,z')$ has size linear in the size of the automaton $\mathcal{A}$ and the size of the alphabet F.*

Notice that $\Phi_{\mathcal{A}}(z,z')$ is satisfiable over finite and infinite trees. To see this, observe that the valuation mapping z and z' to the smallest feature tree (τ with $D_\tau = \{\varepsilon\}$ and $L_\tau = \emptyset$) is the least solution of $\Phi_{\mathcal{A}}(z,z')$.

Proposition 6.4 *Let $\mathcal{A} = (Q, q_s, q_f, \delta)$ be a nondeterministic automaton. Then the following entailment judgement holds for both $FT_{\leq}$ and $FT^{fin}_{\leq}$:*

$$\Phi_{\mathcal{A}}(z,z') \;\models\!\!\!\mid\; \bigwedge_{\pi\in\mathcal{L}(\mathcal{A})} z?[\pi] \leq z' \wedge \mathsf{all\text{-}compat}_{F^*}(z)$$

[4]Thanks to Franz Baader for pointing this out.

Proof. For the direction from left to right one shows by induction over π that

$$\varphi_{\mathcal{A}}(z,z') \models \bigwedge_{\pi\in\mathcal{L}(\mathcal{A})} q_s?[\pi] \leq q_f$$

Since also $\varphi_{\mathcal{A}}(z)$ implies $z{\leq}q_s \wedge q_f{\leq}z'$ we directly obtain

$$\varphi_{\mathcal{A}}(z) \models \bigwedge_{\pi\in\mathcal{L}(\mathcal{A})} z?[\pi] \leq z' \wedge \text{all-compat}_{F^*}(z)$$

Since the right hand side has only a single free variable z also $\Phi_{\mathcal{A}}(z)$ implies it.

For the inverse direction, we assume that α is an arbitrary solution of the right hand side and show that α is also a solution of $\Phi_{\mathcal{A}}$. We now extend α to a valuation α' on the local variables in $\Phi_{\mathcal{A}}$ (*i.e.*, to the variables in $\varphi_{\mathcal{A}}$). We define

$$\begin{array}{lcllcl}
\alpha'(z) & = & \alpha(z) & & & \\
\alpha'(z') & = & \alpha(z') & & & \\
\alpha'(q) & = & \tau_q & \text{where} \quad D_{\tau_q} & = & \bigcup\{D_{\alpha(z)[\pi]} \mid \pi\in D_{\alpha(z)} \text{ and } \mathcal{A}\vdash q_s \xrightarrow{\pi} q\} \\
 & & & \text{and} \quad L_{\tau_q} & = & \bigcup\{L_{\alpha(z)[\pi]} \mid \pi\in D_{\alpha(z)} \text{ and } \mathcal{A}\vdash q_s \xrightarrow{\pi} q\}
\end{array}$$

The valuation α' is well-defined since $\alpha \models \text{all-compat}_{F^*}(z)$. Notice also that α' is a valuation into finite trees if α is one. It remains to show that α' is a solution of $\varphi_{\mathcal{A}}(z,z')$.

- $\alpha' \models z{\leq}q_s$ is clear since

  $$D_{\alpha'(z)} = D_{\alpha(z)[\varepsilon]} \subseteq D_{\alpha'(q_s)} \quad \text{and} \quad L_{\alpha'(z)} = L_{\alpha(z)[\varepsilon]} \subseteq L_{\alpha'(q_s)}$$

  $\alpha' \models q_f{\leq}z_f$ holds because $\alpha \models \bigwedge_{\pi\in\mathcal{L}(\mathcal{A})} z?[\pi] \leq z_f$, so that for all $\pi \in \mathcal{L}(\mathcal{A}) \cap D_{\alpha(z)} : \alpha(z)[\pi]{\leq}\alpha(z_f)$. Hence,

  $$\begin{array}{lcl}
  D_{\alpha'(q_f)} = \bigcup_{\pi\in\mathcal{L}(\mathcal{A})\cap D_{\alpha(z)}} D_{\alpha(z)[\pi]} & \subseteq & D_{\alpha(z_f)} \\
  L_{\alpha'(q_f)} = \bigcup_{\pi\in\mathcal{L}(\mathcal{A})\cap L_{\alpha(z)}} L_{\alpha(z)[\pi]} & \subseteq & L_{\alpha(z_f)}
  \end{array}$$

- Assume $(q,f,q')\in\delta$. We have to show that $\alpha' \models q?[f] \leq q'$, *i.e.*, that $f\in D_{\alpha'(q)}$ implies $\alpha'(q)[f]{\leq}\alpha'(q')$. We check that $D_{\alpha'(q)[f]} \subseteq D_{\alpha'(q')}$, the argument for the labeling function is analogous.

  If $f\in D_{\alpha'(q)}$, then the definition of α' implies that there exists a path π such that $\pi f \in D_{\alpha(z)}$ and $\mathcal{A}\vdash q_s \xrightarrow{\pi} q$. Furthermore, we know by $(q,f,q')\in\delta$ that, for all $\pi\in F^*$, $\mathcal{A}\vdash q_s \xrightarrow{\pi} q$ if and only if $\mathcal{A}\vdash q_s \xrightarrow{\pi f} q'$. Hence

  $$\begin{array}{lcl}
  D_{\alpha'(q)[f]} & = & \bigcup\{D_{\alpha(z)[\pi f]} \mid \pi\in D_{\alpha(z)} \text{ and } \mathcal{A}\vdash q_s \xrightarrow{\pi} q\} \\
  & = & \bigcup\{D_{\alpha(z)[\pi f]} \mid \pi f\in D_{\alpha(z)} \text{ and } \mathcal{A}\vdash q_s \xrightarrow{\pi f} q'\} \\
  & = & D_{\alpha'(q')}
  \end{array}$$

- $\alpha' \models \text{all-compat}_{F^*}(z)$ holds since $\alpha \models \text{all-compat}_{F^*}(z)$ and $\alpha(z) = \alpha'(z)$.

□

Proposition 6.5 (Contradiction) *Let $\mathcal{A}$ be a non-deterministic automaton. Then $\pi \notin \mathcal{L}(\mathcal{A})$ implies $\Phi_A(z,z') \not\models z?[\pi] \leq z'$.*

Proof. By Proposition 6.4, it is sufficient to show that there exists a solution of

$$\bigwedge_{\pi \in \mathcal{L}(\mathcal{A})} z?[\pi] \leq z' \wedge \textsf{all-compat}_{F^*}(z) \tag{1}$$

that contradicts $z?[\pi] \leq z'$. Let α be a solution of (1). By the form of least solutions, Proposition 4.2, we can assume without loss of generality that $D_{\alpha(z)}, D_{\alpha(z')} \subseteq F^*$. Now let $g \notin F$ be a fresh feature, and set $\alpha'(z) = \tau$ where $D_\tau = D_{\alpha(z)} \cup \{\pi g\}$ and $L_\tau = L_{\alpha(z)}$. Apparently, α' is still a solution (1) since this formula does not mention a path constraint for path πg, and α' contradicts $z?[\pi] \leq z'$ since $g \in D_{\alpha'(z)[\pi]} \not\subseteq D_{\alpha'(z)}$. □

Proposition 6.6 (Reduction) *For two non-deterministic automata $\mathcal{A}_1$ and $\mathcal{A}_2$ defined on the same set of states, it holds that*

$$\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2) \quad \textit{iff} \quad \Phi_{\mathcal{A}_2} \models_{\mathrm{FT}_\leq} \Phi_{\mathcal{A}_1} \quad \textit{iff} \quad \Phi_{\mathcal{A}_2} \models_{\mathrm{FT}_\leq^{fin}} \Phi_{\mathcal{A}_1}$$

Proof. The direction from left to right ($\mathcal{L}(\mathcal{A}_1) \subseteq \mathcal{L}(\mathcal{A}_2)$ implies both $\Phi_{\mathcal{A}_2} \models_{\mathrm{FT}_\leq} \Phi_{\mathcal{A}_1}$ and $\Phi_{\mathcal{A}_2} \models_{\mathrm{FT}_\leq^{fin}} \Phi_{\mathcal{A}_1}$) follows immediately from Proposition 6.4.

The inverse direction is shown by contraposition. If $\mathcal{L}(\mathcal{A}_1) \not\subseteq \mathcal{L}(\mathcal{A}_2)$ then there exists a path $\pi \in \mathcal{L}(\mathcal{A}_1)$ such that $\pi \notin \mathcal{L}(\mathcal{A}_2)$. Since $\pi \in \mathcal{L}(\mathcal{A}_1)$, Proposition 6.4 yields that $\Phi_{\mathcal{A}_1}(z,z') \models z?[\pi] \leq z'$. Since $\pi \notin \mathcal{L}(\mathcal{A}_2)$, Proposition 6.5 implies $\Phi_{\mathcal{A}_2}(z,z') \not\models z?[\pi] \leq z'$. Hence $\Phi_{\mathcal{A}_2} \not\models \Phi_{\mathcal{A}_1}$. This argument holds for both $\mathrm{FT}_\leq$ and $\mathrm{FT}_\leq^{fin}$. □

By way of remark, notice that the naïve approach to transform $\Phi_{\mathcal{A}}(z,z')$ to a unary predicate by setting $\Phi'_{\mathcal{A}}(z) = \exists z' \Phi_{\mathcal{A}}(z,z')$ does not work due to the equivalence of $\textsf{all-compat}_{F^*}(x)$ and $\exists z (\bigwedge_{\pi \in F^*} x?[\pi] \leq z)$ mentioned at the end of Section 3.2.

7 Second-order Monadic Logic

We recall the definitions of *second-order monadic logic with countably many successors* SωS (Rabin, 1969) and of *weak second-order monadic logic with countably many successors* WSωS (Thatcher & Wright, 1968). Syntactically, SωS and WSωS coincide. We assume an additional infinite set of *path variables* denoted by p that is disjoint from the variables denoted by x. Formulas ψ of SωS and WSωS are built from variables x and p and features f.

$$\begin{array}{rcl} w & ::= & p \mid \varepsilon \mid fw \\ \psi & ::= & p{\in}x \mid p{=}w \mid \psi \wedge \psi' \mid \neg\psi \mid \forall p \psi \mid \forall x \psi \end{array}$$

The semantics of SωS is defined as follows. A path variable p is interpreted as a path (a word over features) and a variable x as a set of words over features. The denotation of ε is the empty path and the denotation of fw denotes the path obtained by concatenation f in front of the denotation of w. The membership constraint $p{\in}x$ holds if the denotation of p is a member of the denotation of x. The equality constraint

$p{=}w$ holds if the denotations of p and w are equal. The semantics of WSωS coincides with the semantics of SωS except that in WSωS a variable x denotes a *finite* set of paths.

As derived forms we will use the following formulas with their usual semantics:

$$\exists p\psi,\ \exists x\psi,\ \psi \to \psi'\ \psi \leftrightarrow \psi'$$

Theorem 7.1 ((Rabin, 1969; Thatcher & Wright, 1968; Doner, 1970)) *The satisfiability problems of WSωS and SωS are decidable.*

The second-order monadic logic with two successors, S2S and WS2S, are obtained as the restrictions of SωS and WSωS to a finite set of features $\{1,2\}$.

7.1 The Structures $\mathrm{FT}^{-}_{\leq}$ and $\mathrm{FT}^{-fin}_{\leq}$

Second-order monadic logic can be understood as a feature logic closely related to $\mathrm{FT}_{\leq}$. For this purpose, it is sufficient to restrict the structures $\mathrm{FT}_{\leq}$ and $\mathrm{FT}^{fin}_{\leq}$ to sufficiently labeled feature trees.

Definition 7.2 *The structure $FT^{-}_{\leq}$ is the restriction of the structure $FT_{\leq}$ to the domain of sufficiently labeled feature trees. The structure $FT^{-fin}_{\leq}$ is the restriction of the structure $FT^{fin}_{\leq}$ to the domain of sufficiently labeled feature trees.*

The first-order theories of $\mathrm{FT}_{\leq}$ and $\mathrm{FT}^{-}_{\leq}$ differ. One example is the formula Φ_2 discussed in Section 2.4. Another example is the following existential formula Φ_3:

$$\Phi_3 = \exists x(x{\leq}x_1 \wedge x{\leq}x_2)$$

The formula Φ_3 requires for all τ_1 and τ_2 that there exists τ such that $\tau{\leq}\tau_1$ and $\tau{\leq}\tau_2$. Φ_3 is valid over $\mathrm{FT}_{\leq}$ but not valid over $\mathrm{FT}^{-}_{\leq}$. In $\mathrm{FT}_{\leq}$ one may always choose $\tau = (\{\varepsilon\}, \emptyset)$. This particular choice is impossible in $\mathrm{FT}^{-}_{\leq}$ since $(\{\varepsilon\}, \emptyset)$ is not sufficiently labeled. Worse, if $\tau_1 = (\{\varepsilon\}, \{(\varepsilon, a_1)\})$, $\tau_2 = (\{\varepsilon\}, \{(\varepsilon, a_2)\})$, and $a_1 \neq a_2$ then there exists no appropriate tree τ in $\mathrm{FT}^{-}_{\leq}$ at all.

Proposition 7.3 *The first-order theory of $FT^{-}_{\leq}$ can be embedded in linear time into the first-order theory of $FT_{\leq}$.*

Proof. This follows from the fact that the predicate $\mathsf{suff\text{-}lab}$ can be expressed in $\mathrm{FT}_{\leq}$. In order to encode a closed $\mathrm{FT}^{-}_{\leq}$ formula Φ one simply restricts all quantification within Φ to the domain of sufficiently labeled feature trees by replacing all subformulas $\exists x\Phi'$ recursively by $\exists x(\mathsf{suff\text{-}lab}(x) \wedge \Phi')$. □

In the remainder of this section we show that second-order monadic logic and the theory of weak subsumption for sufficiently labeled feature trees are interreducible. For our main result given in Section 8, namely the decision procedure for entailment $\varphi \models \exists \bar{x}\varphi'$, it is sufficient to embed the theory of $\mathrm{FT}^{-}_{\leq}$ (*resp.*, $\mathrm{FT}^{-fin}_{\leq}$) into S2S (*resp.*, WS2S). This direction is shown in Section 7.3 below.

The inverse direction, namely the embedding of (W)S2S into $FT^{-}_{\leq}$ is meant as an additional illustration. This embedding did not appear in the conference version (Müller & Niehren, 1998), but was found in collaboration with Ralf Treinen and is contained in the full version of (Müller et al., 1998).

As an immediate consequence, Proposition 7.3 implies that second order monadic logic is expressible in the first-order theory of $FT_{\leq}$.

7.2 Encoding (W)S2S in $FT^{-}_{\leq}$

We show how to encode second-order monadic logic into the first-order theory of the structure of sufficiently labeled feature trees. We give a single embedding that is correct both as an embedding for WS2S into the first-order theory of $FT^{-fin}_{\leq}$ and for S2S into the first-order theory of $FT^{-}_{\leq}$.

The key to this embedding are two predicates which express in $FT^{-}_{\leq}$ and $FT^{-fin}_{\leq}$ that a tree denotes a set (*resp.*, a finite set) of words over $\{1,2\}$, and that a tree denotes a singleton set. We define these predicates first.

Given a finite set of features F, let $\mathsf{label\text{-}arity}_{a(F)}(\tau)$ be the unary predicate which holds for a feature tree τ if and only if τ is labeled with a at the root and if τ has exactly the features in F at the root: That is, $\mathsf{label\text{-}arity}_{a(F)}(\tau)$ holds if and only if $a(\tau)$ and $F = \mathcal{D}_\tau \cap \mathcal{F}$. If $F = \{f_1, \ldots, f_n\}$ then we express $\mathsf{label\text{-}arity}_{a(F)}$ by the following first-order formula $\mathsf{label\text{-}arity}_{a(F)}(x)$ in $FT^{-}_{\leq}$:

$$\begin{array}{lcll} \mathsf{label\text{-}arity}_{a(\{f_1,\ldots,f_n\})}(x) & = & \exists x_1 \ldots \exists x_n \, (& a(x) \wedge \bigwedge_{i=1}^{n} x[f_i]x_i \wedge \\ & & & \forall y((a(y) \wedge \bigwedge_{i=1}^{n} y[f_i]x_i) \rightarrow x \leq y)) \end{array}$$

We fix three features $0,1,2$ and a label a. Based on the previous formula, we can express the predicate $\mathsf{bin\text{-}tree}_a(\tau)$ which holds if τ is a binary tree over features $\{1,2\}$ such that all its nodes labeled with a unless they are unlabeled, *i.e.* with $D_\tau \subseteq \{1,2\}^*$ and $L_\tau \subseteq D_\tau \times \{a\}$. We express $\mathsf{bin\text{-}tree}_a(\tau)$ by the following formula $\mathsf{bin\text{-}tree}_a(x)$:

$$\mathsf{bin\text{-}tree}_a(x) \; = \; \exists y(x \leq y \wedge \mathsf{label\text{-}arity}_{a(\{1,2\})}(y) \wedge y?[1] \leq y \wedge y?[2] \leq y)$$

Note that if $\mathsf{bin\text{-}tree}_a(\tau)$ holds for some τ then $\{\pi \mid L_\tau(\pi) = a\} \neq \emptyset$ since every sufficiently labeled feature tree has at least one label. In other words, we can only represent empty sets with the predicate $\mathsf{bin\text{-}tree}_a$. We express possibly empty sets by the predicate $\mathsf{set}(\tau)$ which satisfies $D_\tau \subseteq 0(1 \cup 2)^*$ and $\{(0,a)\} \subseteq L_\tau \subseteq D_\tau \times \{a\}$. We express this predicate as follows:

$$\mathsf{set}(x) \quad = \quad \mathsf{label\text{-}arity}_{a(\{0\})}(x) \wedge \exists y(x[0]y \wedge \mathsf{bin\text{-}tree}_a(y))$$

The empty set is encoded by the tree $(\{0\}, \{(0,a)\}$ which is the least sufficiently labeled tree satisfying the predicate set.

$$\begin{array}{rcl} \mathsf{empty\text{-}set}(x) & = & \mathsf{set}(x) \wedge \forall y(\mathsf{set}(y) \rightarrow x \leq y) \\ \mathsf{non\text{-}empty\text{-}set}(x) & = & \mathsf{set}(x) \wedge \neg \mathsf{empty\text{-}set}(x) \end{array}$$

Singleton sets are minimal non-empty sets:

$$\mathsf{singleton}(x) \quad = \quad \mathsf{non\text{-}empty\text{-}set}(x) \wedge \neg \exists y(\mathsf{non\text{-}empty\text{-}set}(y) \wedge y \leq x \wedge \neg x \leq y)$$

Now we define the embedding of (W)S2S into $FT^{-}_{\leq}$ (*resp.*$FT^{-fin}_{\leq}$) as a mapping from (W)S2S formulas into the first-order theory of $FT^{-}_{\leq}$ (*resp.*$FT^{-fin}_{\leq}$).

$$
\begin{array}{lcl}
[\![p = p']\!] & = & p = p' \\
[\![p = \varepsilon]\!] & = & \textsf{label-arity}_{a(\emptyset)}(p) \\
[\![p = 1w]\!] & = & \exists p (p[1]p' \wedge \textsf{singleton}(p) \wedge [\![p' = w]\!]) \\
[\![p = 2w]\!] & = & \exists p (p[2]p' \wedge \textsf{singleton}(p) \wedge [\![p' = w]\!]) \\
[\![\psi \wedge \psi']\!] & = & [\![\psi]\!] \wedge [\![\psi']\!] \\
[\![\neg \psi]\!] & = & \neg [\![\psi]\!] \\
[\![\exists x \psi]\!] & = & \exists x (\textsf{set}(x) \wedge [\![\psi]\!]) \\
[\![\exists p \psi]\!] & = & \exists p (\textsf{singleton}(p) \wedge [\![\psi]\!])
\end{array}
$$

Proposition 7.4 *An S2S (resp., WS2S) formula ψ is valid if and only if the formula $[\![\psi]\!]$ is valid over $FT^{-}_{\leq}$ (resp., $FT^{-fin}_{\leq}$).*

Proof. One shows by structural induction over the formula ψ that the solutions of ψ and $[\![\psi]\!]$ are in 1-1 correspondence through the encoding of sets as trees as discussed above. □

7.3 Encoding $FT^{-}_{\leq}$ in (W)S2S

We do not give the reduction of $FT^{-}_{\leq}$ in (W)S2S directly but by detour through the second-order monadic logic with countably many successors (W)SωS. WSωS (*resp.*, SωS) is known to be expressible in WS2S (*resp.*, S2S) (Thatcher & Wright, 1968; Rabin, 1969).

Trees as Sets. Every sufficiently labeled feature tree can be identified with a unique n-tuple of pairwise disjoint non-empty sets of paths with non-empty union, and vice versa, if n is the number of labels in $\mathcal{L}$. For every label a we define a function γ_a from feature trees to non-empty sets of paths:

$$\gamma_a(\tau) = \{\pi \mid L_\tau(\pi) = a\}$$

For $\mathcal{L}=\{a_1, \ldots, a_n\}$ we define $\gamma(\tau)$ as the following n-tuple of sets of paths:

$$\gamma(\tau) = (\gamma_{a_1}(\tau), \ldots, \gamma_{a_n}(\tau))$$

Proposition 7.5 *The mapping γ from sufficiently labeled feature trees to n-tuples of pairwise disjoint sets of words with non-empty union is one-to-one and onto. Furthermore, τ is a finite tree if and only if every component of $\gamma(\tau)$ is finite.*

Proof. Let τ be a sufficiently labeled feature tree. Since $\varepsilon \in D_\tau$ there exists a path π and a label a such that $(\varepsilon\pi, a) \in L_\tau$. Hence $\cup_{i=1}^{n} \gamma_{a_i}(\tau)$ is nonempty. The sets $\gamma_{a_i}(\tau)$ are pairwise disjoint since L_τ has to be a partial function. It is also clear that $\cup_{i=1}^{n} \gamma_{a_i}(\tau)$ is finite if τ is finite. The converse follows from the fact that a sufficiently labeled infinite tree has infinitely many labeled nodes.

In order to prove that γ is one-to-one and onto, we define the inverse mapping of γ as follows. Let $(\Pi_1,\dots,\Pi_n)$ be pairwise disjoint sets of words over features that have a nonempty union. We define $\gamma^{-1}(\Pi_1,\dots,\Pi_n)$ as follows:

$$\begin{array}{rcl} D_{\gamma^{-1}(\Pi_1,\dots,\Pi_n)} & = & \{\pi \mid \pi \text{ is a prefix of some word in } \cup_{i=1}^n \Pi_i\} \\ L_{\gamma^{-1}(\Pi_1,\dots,\Pi_n)} & = & \{(\pi, a_i) \mid 1 \le i \le n,\ \pi \in \Pi_i\} \end{array}$$

Since $\bigcup_{i=1}^n \Pi_i$ is assumed non-empty, we have $\varepsilon \in D_{\gamma^{-1}(\Pi_1,\dots,\Pi_n)}$, which is also prefix closed by construction. The relation $L_{\gamma^{-1}(\Pi_1,\dots,\Pi_n)}$ is a partial function since all Π_i are assumed pairwise disjoint. Hence $\gamma^{-1}(\Pi_1,\dots,\Pi_n)$ is a feature tree, which clearly is sufficiently labeled.

It is quite obvious that γ^{-1} is in fact the inverse function of γ, *i.e.*, that $\gamma^{-1}(\gamma(\tau)) = \tau$ for all sufficiently complete τ and that $\gamma(\gamma^{-1}(\Pi_1,\dots,\Pi_n)) = (\Pi_1,\dots,\Pi_n)$ for all $\Pi_1,\dots,\Pi_n$ that are pairwise disjoint and have a non-empty union. $\square$

Note that we need not require prefix closedness for the sets in the domain of γ, since the domain of a sufficiently labeled feature tree τ is uniquely determined by its labeling function L_τ. This observation is crucial for our reduction to second-order monadic logic. Note also that the notion of sufficient labeling does not make sense for the alternative notions of feature trees mentioned above (Aït-Kaci et al., 1994; Backofen, 1995).

Definition 7.6 *The structure $FT^-_\le$ is the restriction of the structure $FT_\le$ to the domain of sufficiently labeled feature trees.*

We may interpret $\mathrm{FT}^-_\le$ either over finite trees or over possibly infinite trees. Whenever this choice matters, we will make it explicit.

Reduction to SωS or WSωS. We next define a mapping from first-order formulas over ordering constraints (interpreted over $\mathrm{FT}^-_\le$) to formulas of second-order monadic logic with countably many successors. We will make use of the following abbreviations:

$$x{\cap}y{=}\emptyset \quad = \quad \neg\exists p(p{\in}x \wedge p{\in}y) \quad \text{and} \quad x \subseteq y \quad = \quad \forall p(p{\in}x \rightarrow p{\in}y)$$

For every variable x and label a let x_a be a fresh variable. Suppose that $\mathcal{L} = \{a_1,\dots,a_n\}$. Here comes the definition of the mapping $[\![\ \_\ ]\!]$:

$$\begin{array}{lcl} [\![a(x)]\!] & = & \varepsilon{\in}x_a \\ [\![x[f]y]\!] & = & \bigwedge_{i=1}^n \forall p(fp{\in}x_{a_i} \leftrightarrow p{\in}y_{a_i}) \\ [\![x{\le}y]\!] & = & \bigwedge_{i=1}^n x_{a_i}{\subseteq}y_{a_i} \\ [\![\varphi\wedge\varphi']\!] & = & [\![\varphi]\!]\wedge[\![\varphi']\!] \\ [\![\neg\varphi]\!] & = & \neg[\![\varphi]\!] \\ [\![\exists x\varphi]\!] & = & \exists x_{a_1}\dots\exists x_{a_n}((\bigwedge_{\substack{i,j=1\\ i\ne j}}^n x_{a_i}{\cap}x_{a_j}{=}\emptyset)\wedge\exists p(\bigvee_{i=1}^n p{\in}x_{a_i})\ \wedge [\![\varphi]\!]) \end{array}$$

Proposition 7.7 *A first-order formula Φ is valid over $FT^-_\le$ interpreted over finite (resp. infinite) trees if and only if its translation $[\![\varphi]\!]$ is valid over WSωS (resp. SωS).*

Proof. If α is a solution of Φ then α' with $\alpha'(x_{a_i}) = \gamma_{a_i}(\alpha(x))$ for all i, $1 \le i \le n$, is a solution of $[\![\,\Phi\,]\!]$. If β is a solution of $[\![\,\Phi\,]\!]$ then the mapping β' with $\beta'(x) = \gamma^{-1}(\beta(x_{a_1}),\dots,\beta(x_{a_n}))$ is a solution of Φ. The existence of the inverse mapping γ^{-1} of γ is proved by Proposition 7.5. $\square$

An immediate corollary of this proposition is the following one:

Corollary 7.8 *The first-order theory of $FT^-_\le$ is decidable.*

In (Müller et al., 1998) we show that the first-order theory of $\mathrm{FT}_\le$ is undecidable. Hence, the inverse of Proposition 7.3 does not hold: $\mathrm{FT}_\le$ cannot be embedded into $\mathrm{FT}^-_\le$.

8 Deciding Entailment with Existential Quantifiers

In this section we prove the decidability of the entailment problem of $\mathrm{FT}_\le$ by reduction to Rabin's decidability result for second-order monadic logic. Our proof is based on a model change from (possibly labeled) feature trees to sufficiently labeled feature trees.

Theorem 8.1 *The entailment problem of $FT_\le$ with existential quantification $\varphi \models \exists\bar{x}\varphi'$ is decidable, both when interpreted $FT_\le$ or $FT^{fin}_\le$.*

The proof is developed in this section. It consists in reducing the entailment problem of $\mathrm{FT}_\le$ to the entailment problem of $\mathrm{FT}^-_\le$ which is proved decidable in Corollary 7.8. All our proofs apply without difference to the case of finite trees and to the case of possibly infinite feature trees.

We need distinct notations for entailment with respect to $\mathrm{FT}_\le$ (*resp.*, $\mathrm{FT}^{fin}_\le$) and $\mathrm{FT}^-_\le$ (*resp.*, $\mathrm{FT}^{-fin}_\le$). For this purpose, we write $\Phi \models_{\mathrm{FT}_\le} \Phi'$ and $\Phi \models_{\mathrm{FT}^-_\le} \Phi'$ ignoring the finiteness annotation.

Definition 8.2 *Throughout the section, we fix a label $b \in \mathcal{L}$ and a feature $g \in \mathcal{F}$. We define the following function $\eta_{g,b}$ which maps constraints to first-order formula over constraints:*

$$\eta_{g,b}(\varphi) = \varphi \wedge \bigwedge_{y \in \mathcal{V}(\varphi)} b(g[y])$$

Note that $\varphi \vdash y[\pi]\!\downarrow$ implies $\eta_{g,b}(\varphi) \vdash b(y[\pi g])$ for all π and $y \in \mathcal{V}(\varphi)$. This has the following consequence which illustrates the usefulness of the formula $\eta_{g,b}(\varphi)$

Lemma 8.3 *A constraint φ with $g \notin \mathcal{F}(\varphi)$ is satisfiable in $FT_\le$ if and only if $\eta_{g,b}(\varphi)$ is satisfiable in $FT^-_\le$, both in the case of finite and in the case of infinite trees.*

Proof. If $\eta_{g,b}(\varphi)$ is satisfiable of $\mathrm{FT}^-_\le$ then clearly φ is satisfiable over $\mathrm{FT}_\le$. Conversely, if φ is satisfiable then we can assume without loss of generality that φ is F1-F2-closed. Otherwise, we can replace φ by its F1-F2-closure which is provided by to Theorem 4.1. Since $g \notin \mathcal{F}(\varphi)$, the function least_φ defined in Proposition 4.2 maps every variable to a feature tree. Hence, this Proposition implies that least_φ is a solution of $\eta_{g,b}(\varphi)$ over $FT_{\leq}$. The note preceding the lemma implies that $\mathsf{least}_{\varphi}(x)$ is sufficiently labeled for all $x \in \mathcal{V}(\varphi)$, *i.e.* $\eta_{g,b}(\varphi)$ is satisfiable over $FT^{-}_{\leq}$. □

We now generalize Lemma 8.3 to entailment with existential quantifiers.

Proposition 8.4 (Model Change) *Let $\bar{x}$ a sequence of variables and φ and φ' constraints such that $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$. If $g \notin \mathcal{F}(\varphi \wedge \varphi')$ then $\varphi \models_{FT_{\leq}} \exists\bar{x}\varphi'$ if and only if $\eta_{g,b}(\varphi) \models_{FT^{-}_{\leq}} \exists\bar{x}\varphi'$, both in the case of finite and in the case of infinite trees.*

Notice that the assumption $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$ does not restrict generality, since we can always assume that φ contains trivial inequalities $y{\leq}y$ for all $y \in \mathcal{V}(\exists\bar{x}\varphi')$.

Proof. This proof of this proposition is technically involved. It is given in Sections 8.1 and 8.2 and follows from the Propositions 8.10 and 8.15. □

Note that Proposition 8.4 fails when $\eta_{g,b}(\varphi)$ is replaced by φ. This can be illustrated by the formula $\Phi_3 = \exists x(x{\leq}x_1 \wedge x{\leq}x_2)$ which is valid over $FT_{\leq}$ but not over $FT^{-}_{\leq}$since it requires for each pair of trees the existence of a smaller tree. In order to relate this fact to Proposition 8.4, let φ_3 be the tautological constraint $x_1{\leq}x_1 \wedge x_2{\leq}x_2$ which satisfies $\mathcal{V}(\Phi_3) \subseteq \mathcal{V}(\varphi_3)$. Now, $\varphi_3 \models_{FT_{\leq}} \Phi_3$ but $\varphi_3 \not\models_{FT^{-}_{\leq}} \Phi_3$. However, $\eta_{g,b}(\varphi_3) \models_{FT^{-}_{\leq}} \Phi_3$ where $\eta_{g,b}(\varphi_3) = \varphi_3 \wedge b(x_1[g]) \wedge b(x_2[g])$.

Proof of Theorem 8.1. We wish to decide an entailment problem of the form $\varphi \models_{FT_{\leq}} \exists\bar{x}\varphi'$. We choose a feature g not occurring in φ or φ' (this exists since the set of all features $\mathcal{F}$ is infinite). By Lemma 8.4 it is sufficient to decide the entailment propositions $\eta_{g,b}(\varphi) \models_{FT^{-}_{\leq}} \exists\bar{x}\varphi'$ over $FT^{-}_{\leq}$. By Proposition 7.7, $\eta_{g,b}(\varphi) \models_{FT^{-}_{\leq}} \exists\bar{x}\varphi'$ holds if and only if the translation $[\![\eta_{g,b}(\varphi) \rightarrow \exists\bar{x}\varphi']\!]$ is a valid formula of WSωS in the case of finite trees and of SωS in the case of infinite trees. The validity of these formulas is decidable by Rabin's Theorem 7.1. □

8.1 Entailment in $FT^{-}_{\leq}$ Implies Entailment in $FT_{\leq}$

In this section, we proof one of the two implications in Proposition 7.7. We consider the entailment judgement $\varphi \models \exists\bar{x}\varphi'$ under the assumption that $g \notin \mathcal{F}(\varphi \wedge \varphi')$. Recall that the feature g and the label b have been fixed globally.

Adding Labels. We define a mapping $\delta_{g,b}$ from feature trees to feature trees. Intuitively, $\delta_{g,b}(\tau)$ is obtained by adding a leaf $(\pi g, b)$ to every node π of τ. Formally, we assume a feature tree τ such that $g \notin \mathcal{F}(\tau)$.

$$\begin{array}{rcl} D_{\delta_{g,b}(\tau)} & = & D_\tau \cup \{\pi g \mid \pi \in D_\tau\} \\ L_{\delta_{g,b}(\tau)} & = & L_\tau \cup \{(\pi g, b) \mid \pi \in D_\tau\} \end{array}$$

Lemma 8.5 *If τ is a feature tree such that $g \notin \mathcal{F}(\tau)$ for all π then $\delta_{g,b}(\tau)$ is a feature tree that is sufficiently labeled.*

Proof. The assumption $g \notin \mathcal{F}(\tau)$ implies that $L_{\delta_{g,b}(\tau)}$ is a partial function such that $\delta_{g,b}(\tau)$ is indeed a feature tree. □

Lemma 8.6 *Assume $g \notin \mathcal{F}(\alpha(x))$ for x and $g \notin \mathcal{F}(\varphi)$. If α is a solution of φ in $FT_{\leq}$ then $\delta_{g,b} \circ \alpha$ is a solution of φ in $FT_{\leq}^{-}$.*

Proof. We have to show that every basic constraint in φ is satisfied by $\delta_{g,b} \circ \alpha$.

1. Case $x[f]y$ *in* φ where $f \neq g$ due to $g \notin \mathcal{F}(\varphi)$. We have to verify for all π that $f\pi \in D_{\delta_{g,b}(\alpha(x))}$ is equivalent to $\pi \in D_{\delta_{g,b} \circ \alpha(y)}$. This is proved by the following sequence of equivalences:

$$f\pi \in D_{\delta_{g,b}(\alpha(x))} \quad \text{iff} \quad f\pi \in D_{\alpha(x)} \cup \{\pi' g \mid \pi' \in D_{\alpha(x)}\}$$

   Note that $f\pi = \pi' g$ and $f \neq g$ implies the existence of π'' such that $\pi = \pi'' g$ and $f\pi'' = \pi'$. Hence

$$\begin{array}{lll} f\pi \in D_{\delta_{g,b}(\alpha(x))} & \text{iff} & f\pi \in D_{\alpha(x)} \cup \{f\pi'' g \mid f\pi'' \in D_{\alpha(x)}\} \\ & \text{iff} & \pi \in D_{\alpha(y)} \cup \{\pi'' g \mid \pi'' \in D_{\alpha(y)}\} \\ & \text{iff} & \pi \in D_{\delta_{g,b}(\alpha(y))} \end{array}$$

   The reasoning for the labeling function is similar.

2. Case $x \leq y$ *in* φ. We have to verify the domain inclusion $D_{\delta_{g,b}(\alpha(x))} \subseteq D_{\delta_{g,b}(\alpha(y))}$.

$$\begin{array}{lll} D_{\delta_{g,b}(\alpha(x))} & = & D_{\alpha(x)} \cup \{\pi' g \mid \pi' \in D_{\alpha(x)}\} \\ & \subseteq & D_{\alpha(y)} \cup \{\pi' g \mid \pi' \in D_{\alpha(y)}\} \\ & = & D_{\delta_{g,b}(\alpha(y))} \end{array}$$

   The reasoning for the labeling function is again similar.

3. The case $a(x)$ *in* φ is simple, since no label is deleted from $L_{\alpha(x)}$.

□

Deleting Labels. There exists a left-inverse $\delta_{g,b}^{-1}$ of the function $\delta_{g,b}$. For arbitrary τ, we define a feature tree $\delta_{g,b}^{-1}(\tau)$ as follows:

$$\begin{array}{lll} D_{\delta_{g,b}^{-1}(\tau)} & = & D_\tau \setminus \{\pi g \pi' \mid \pi, \pi' \in \mathcal{F}^*\} \\ L_{\delta_{g,b}^{-1}(\tau)} & = & L_\tau \setminus \{(\pi, b) \mid \pi = \pi' g \pi'',\ b \in \mathcal{L}\} \end{array}$$

The left-inverse removes at paths with feature g, together with their labels if a label exists.

Lemma 8.7 *If $g \notin \mathcal{F}(\tau)$ then $\delta_{g,b}^{-1}(\delta_{g,b}(\tau)) = \tau$.*

Proof. Since $g \notin \mathcal{F}(\tau)$, the tree $\delta_{g,b}(\tau)$ is well-defined and hence $\delta_{g,b}^{-1}(\delta_{g,b}(\tau))$ is.

$$\begin{array}{lll} D_{\delta_{g,b}^{-1}(\delta_{g,b}(\tau))} & = & D_{\delta_{g,b}} \setminus \{\pi g \pi' \mid \pi, \pi' \in \mathcal{F}^*\} \\ & = & (D_\tau \cup \{\pi g \mid \pi \in D_\tau\}) \setminus \{\pi g \pi' \mid \pi, \pi' \in \mathcal{F}^*\} \\ & = & D_\tau \end{array}$$

The last equality holds, since we have require $g \notin \mathcal{F}(\tau)$. The argument for the labeling function is symmetric.

□

Lemma 8.8 *Let $g \notin \mathcal{F}(\varphi)$. If α is a solution of φ in $FT_{\leq}^{-}$ then $\delta_{g,b}^{-1} \circ \alpha$ is a solution of φ in $FT_{\leq}$.*

Proof. We have to show that every basic constraint in φ is satisfied by $\delta_{g,b}^{-1} \circ \alpha$.

1. Case $x[f]y$ *in* φ where $f \neq g$ due to $g \notin \mathcal{F}(\varphi)$. We have to verify for all π that $f\pi \in D_{\delta_{g,b}^{-1}(\alpha(x))}$ is equivalent to $\pi \in D_{\delta_{g,b}^{-1}(\alpha(y))}$. This is proved by the following equivalences:

$$\begin{array}{lll} f\pi \in D_{\delta_{g,b}^{-1}(\alpha(x))} & \text{iff} & f\pi \in D_{\alpha(x)} \setminus \{f\pi g\pi' \mid \pi,\pi' \in \mathcal{F}^*\} \\ & \text{iff} & \pi \in D_{\alpha(y)} \setminus \{\pi g\pi' \mid \pi,\pi' \in \mathcal{F}^*\} \\ & \text{iff} & \pi \in D_{\delta_{g,b}^{-1}(\alpha(y))} \end{array}$$

   The reasoning for the labeling function is similar.

2. Case $x{\leq}y$ *in* φ. We have to verify the inclusions $D_{\delta_{g,b}^{-1}(\alpha(x))} \subseteq D_{\delta_{g,b}^{-1}(\alpha(y))}$ and $L_{\delta_{g,b}^{-1}(\alpha(x))} \subseteq L_{\delta_{g,b}^{-1}(\alpha(y))}$, which is both obvious.

3. The case $a(x)$ *in* φ is simple, since no label is deleted at the root of some tree $\alpha(x)$.

□

Lemma 8.9 *Let φ be a constraint with $g \notin \mathcal{F}(\varphi)$, $\bar{x}$ a sequence of variables, and α a variable assignment such that $g \notin \mathcal{F}(\alpha(y))$ for all $y \in \mathcal{V}(\exists\bar{x}\varphi)$. Then α is a solution of $\exists\bar{x}\varphi$ over $FT_{\leq}$ if and only if $\delta_{g,b} \circ \alpha$ is a solution of $\exists\bar{x}\varphi$ over $FT_{\leq}^{-}$.*

Proof. Let α be a solution of $\exists\bar{x}\varphi$ over $\mathrm{FT}_{\leq}$. There exists a sequence of trees $\bar{\tau}$ such that $\alpha[\bar{\tau}/\bar{x}]$ is a solution of φ over $\mathrm{FT}_{\leq}$. (Here, $\alpha[\bar{\tau}/\bar{x}]$ denotes the valuation that maps $\bar{x}$ pointwise to $\bar{\tau}$ and coincides with α everywhere else.) Since $g \notin \mathcal{F}(\varphi)$, the mapping $\delta_{g,b}^{-1} \circ (\alpha[\bar{\tau}/\bar{x}])$ is also a solution of φ by Lemma 8.8. The latter variable assignment coincides with $\alpha[\delta_{g,b}^{-1}(\bar{\tau})/\bar{x}]$ since we have assumed $g \notin \mathcal{F}(\alpha(y))$ for all y. Thus $\delta_g \circ (\alpha[\delta_{g,b}^{-1}(\bar{\tau})/\bar{x}])$ is a solution of φ over $\mathrm{FT}_{\leq}^{-}$ (Lemma 8.6), which implies that $\delta_g \circ \alpha$ is a solution of $\exists\bar{x}\varphi$ over $\mathrm{FT}_{\leq}^{-}$.

For the converse, assume that $\delta_{g,b} \circ \alpha$ is a solution of $\exists\bar{x}\varphi$ over $\mathrm{FT}_{\leq}^{-}$. There exists a sequence of trees $\bar{\tau}$ such that $(\delta_{g,b} \circ \alpha)[\bar{\tau}/\bar{x}]$ is a solution of φ over $\mathrm{FT}_{\leq}^{-}$. Hence, $\delta_{g,b}^{-1} \circ ((\delta_{g,b} \circ \alpha)[\bar{\tau}/\bar{x}])$ is a solution of φ over $\mathrm{FT}_{\leq}$(Lemma 8.8). Also, $\delta_{g,b}^{-1} \circ \delta_{g,b} \circ \alpha = \alpha$ due to Lemma 8.7 and $g \notin \alpha(y)$ for all y. Thus:

$$\delta_{g,b}^{-1} \circ ((\delta_{g,b} \circ \alpha)[\bar{\tau}/\bar{x}]) = (\delta_{g,b}^{-1} \circ \delta_{g,b} \circ \alpha)[\delta_{g,b}^{-1}(\bar{\tau})/\bar{x}] = \alpha[\delta_{g,b}^{-1}(\bar{\tau})/\bar{x}]$$

This proves that α is a solution of $\exists\bar{x}\varphi$ over $\mathrm{FT}_{\leq}$. □

Proposition 8.10 *Let $g \notin \mathcal{F}(\varphi \wedge \varphi')$ and $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$. If $\eta_{g,b}(\varphi) \models_{\mathrm{FT}_{\leq}^{-}} \exists\bar{x}\varphi'$ then $\varphi \models_{\mathrm{FT}_{\leq}} \exists\bar{x}\varphi'$.*

Proof. Let $g \notin \mathcal{F}(\varphi \wedge \varphi')$, $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$, and $\eta_{g,b}(\varphi) \models_{\mathrm{FT}^-_\leq} \exists\bar{x}\varphi'$. We have to show that every solution α of φ in $\mathrm{FT}_\leq$ is also a solution of $\exists\bar{x}\varphi'$. Since the number of features in $\mathcal{F}$ is infinite and $g \notin \mathcal{F}(\varphi \wedge \varphi')$, we only need to consider solutions α of φ such that $g \notin \mathcal{F}(\alpha(x))$ for all $x \in \mathcal{V}(\varphi)$. This fact is proved as Lemma 8.13 below.

Since α is a solution of φ, and $g \notin \mathcal{F}(\alpha(x))$ for all $x \in \mathcal{V}(\varphi \wedge \exists\bar{x}\varphi')$, $\delta_{g,b} \circ \alpha$ is a solution of φ (Lemma 8.9). From the definition of $\delta_{g,b}$ it follows that $\delta_{g,b} \circ \alpha$ is also a solution of $\bigwedge_{y \in \mathcal{V}(\varphi)} b(y[g])$, *i.e.*, $\delta_{g,b} \circ \alpha$ is a solution of $\eta_{g,b}(\varphi)$ over $\mathrm{FT}^-_\leq$. Entailment as assumed implies that $\delta_{g,b} \circ \alpha$ is a solution of $\exists\bar{x}\varphi'$ over $\mathrm{FT}^-_\leq$. Thus α is also a solution of $\exists\bar{x}\varphi'$ over $\mathrm{FT}_\leq$ (Lemma 8.9). □

Fresh Features. The remainder of this section completes the proof of Proposition 8.10. We show that for deciding entailment $\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi'$ we may restrict consideration to solutions of φ that do not mention a particular feature that does not occur in φ or φ'.

Lemma 8.11 *Let $g \in \mathcal{F}$ and a finite set $F \subseteq \mathcal{F}$ such that $g \notin F$. Then there exists a mapping $\theta : \mathcal{F} \to \mathcal{F}$, which is one-to-one, does not map onto g, and is such that θ restricted to F is the identity function on F.*

Proof. Since $\mathcal{F}$ is countably infinite there exists an enumeration of $\mathcal{F}$ say $\mathcal{F} = \{f_i \mid i \geq 1\}$ is such an enumeration. Let n be the maximal index of a feature in $F \cup \{g\}$ in this enumeration, *i.e.*, $n = \max\{i \mid f_i \in F \cup \{g\}\}$, which exists since F is finite. We define θ by the following equation:

$$\theta(f) = \begin{cases} f_{n+1} & \text{if } f = g \\ f & \text{if } f = f_i,\ 1 \leq i \leq n, \text{ and } f \neq g. \\ f_{n+i+1} & \text{if } f = f_i \text{ and } i \geq n+1 \end{cases}$$

The function θ is well defined because $g \notin F$ and because $\mathcal{F}$ is infinite. It is obvious that θ is one-to-one, does not map onto g, and leaves F invariant. □

Lemma 8.12 *Let Φ be a first-order formula over ordering constraints, α a variable assignment into feature trees and $\theta : \mathcal{F} \to \mathcal{F}$ a function that is one to one and leaves $\mathcal{F}(\Phi)$ invariant. Then α is a solution of Φ if and only if $\theta \circ \alpha$ is a solution of Φ.*

Proof. It is obvious that α is a solution of Φ if and only if $\theta \circ \alpha$ is a solution of $\theta(\Phi)$. Since θ leaves the features in Φ invariant we have $\theta(\Phi) = \Phi$. □

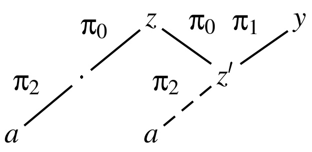
Lemma 8.13 (Fresh Features) *Let $g \notin \mathcal{F}(\varphi \wedge \varphi')$ and $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$. If every solution α of φ with $g \notin \mathcal{F}(\alpha(x))$ for all $x \in \mathcal{V}(\varphi)$ is a solution of $\exists\bar{x}\varphi$ then $\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi'$ holds.*

Proof. Let $\theta : \mathcal{F} \to \mathcal{F}$ be a mapping that is one-to-one, does not map onto g, and leaves $\mathcal{F}(\varphi)$ invariant (Lemma 8.11). Suppose that α is a solution of φ over $\mathrm{FT}_\leq$. Hence, $\theta \circ \alpha$ is a solution of φ over $\mathrm{FT}_\leq$ (Lemma 8.12) that satisfies $g \notin \mathcal{F}(\alpha(x))$ for all $x \in \mathcal{V}(\varphi)$. By assumption, $\theta \circ \alpha$ is a solution of $\exists\bar{x}\varphi'$ over $\mathrm{FT}^-_\leq$. Thus, α is a solution of $\exists\bar{x}\varphi'$ over $\mathrm{FT}_\leq$ (Lemma 8.12). □

8.2 Entailment in $\mathrm{FT}_{\leq}$ Implies Entailment in $\mathrm{FT}_{\leq}^{-}$

We now proof the remaining implication of Proposition 7.7.

Lemma 8.14 (Mountain Chains) *Let α be a solution of φ and assume variables y, z, z', paths π_0, π_1, π_2 and a label a. If $\varphi \models \exists z' (z?[\pi_0] \leq z' \wedge z' \leq z[\pi_1])$ and $(\pi_0\pi_2, a) \in L_{\alpha(z)}$ then $(\pi_1\pi_2, a) \in L_{\alpha(y)}$.*



Proof. By induction on the length of π_0 one shows that $(\pi_2, a) \in L_{\alpha(z')}$, hence $(\pi_1\pi_2, a) \in L_{\alpha(y)}$. (The picture on the right illustrates the situation. The vertical dimension (top to bottom) corresponds to feature selection, the horizontal dimension (left to right) to the ordering $\leq$.) $\square$

Proposition 8.15 *Suppose $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$ and $g \notin \mathcal{F}(\varphi \wedge \varphi')$. If $\varphi \models_{\mathrm{FT}_{\leq}} \exists\bar{x}\varphi'$ then $\eta_{g,b}(\varphi) \models_{\mathrm{FT}_{\leq}^{-}} \exists\bar{x}\varphi'$.*

Proof. We assume $\varphi \models_{\mathrm{FT}_{\leq}} \exists\bar{x}\varphi'$ and show that every solution of $\eta_{g,b}(\varphi)$ in $\mathrm{FT}_{\leq}^{-}$ is a solution of $\exists\bar{x}\varphi'$ in $\mathrm{FT}_{\leq}^{-}$. Lets consider a α of $\eta_{g,b}(\varphi)$ in $\mathrm{FT}_{\leq}^{-}$. Thus, φ and φ' are satisfiable and can be assume to be F1-F2-closed without loss of generality because of Theorem 4.1. We also assume $\mathcal{V}(\varphi) \cap \{\bar{x}\} = \emptyset$ without loss of generality such that $\mathcal{V}(\varphi') \subseteq \{\bar{x}\} \cup \mathcal{V}(\varphi)$

We have to construct a solution α' of φ' over $\mathrm{FT}_{\leq}^{-}$ which coincides with α on the variables in $\mathcal{V}(\varphi)$. Since $\alpha'(y)$ must be sufficiently labeled, it suffices to define its labeling function for $y \in \{\bar{x}\}$:

L1	$(\pi, a) \in L_{\alpha'(y)}$	if	$\varphi' \vdash a(y[\pi])$
L2	$(\pi g, b) \in L_{\alpha'(y)}$	if	$\varphi' \vdash y[\pi]\downarrow$
L3	$(\pi_1\pi_2, a) \in L_{\alpha'(y)}$	if	$\begin{cases} \text{exists } z \in \mathcal{V}(\varphi) \text{ and } z' \text{ such that} \\ \varphi' \vdash z' \leq y[\pi_1],\ \varphi' \vdash z?[\pi_0] \leq z' \text{ and } (\pi_0\pi_2, a) \in L_{\alpha(z)} \end{cases}$

Compare condition L3 with the mountain chain situation depicted above. For all $y \in \{\bar{x}\}$, we define $D_{\alpha'(y)} = \{\pi \mid \pi \text{ is a prefix of } \pi' \text{ and } (\pi', a) \in L_{\alpha'(y)} \text{ for some } a\}$. For all $y \notin \{\bar{x}\}$ we set $\alpha'(y) = \alpha(y)$. It is clear that $\alpha'(y)$ is sufficiently labeled because of L2. It remains to show that α' is a solution of φ', *i.e.*, that α' satisfies all basic constraints in φ'.

1. Case $x \leq y$ *in* φ'. We have to show that $(\pi, a) \in L_{\alpha'(x)}$ implies $(\pi, a) \in L_{\alpha'(y)}$.

   (a) Case $x, y \in \{\bar{x}\}$.

   L1 If $(\pi, a) \in L_{\alpha'(x)}$ has been added because of $\varphi' \vdash a(x[\pi])$ then $\varphi' \vdash a(y[\pi])$ such that $(\pi, a) \in L_{\alpha'(y)}$.

L2 If $(\pi, a) \in L_{\alpha'(x)}$ has been added because of $\pi = \pi' g$, $a = b$, $\varphi' \vdash x[\pi']\downarrow$ then also $\varphi' \vdash y[\pi']\downarrow$ such that $(\pi' g, b) = (\pi, a) \in L_{\alpha'(y)}$.

L3 Let $(\pi, a) \in L_{\alpha'(x)}$ because there exists $z \in \mathcal{V}(\varphi)$, z and π_0, π_1, π_2 such that $\pi = \pi_1\pi_2$, $\varphi' \vdash z' \leq x[\pi_1]$, $\varphi' \vdash z?[\pi_0] \leq z'$, and $(\pi_0\pi_2, a) \in L_{\alpha(z)}$. Since $x \leq y$ *in* φ' we also have $\varphi' \vdash z' \leq y[\pi_1]$ and hence $(\pi_1\pi_2, a) \in L_{\alpha'(y)}$.

(b) Case $x \in \{\bar{x}\}$ and $y \in \mathcal{V}(\varphi)$.

L1 Let $(\pi, a) \in \alpha'(x)$ because $\varphi' \vdash a(x[\pi])$. In this case, $\varphi' \vdash a(y[\pi])$. The correctness of this relation implies $\varphi' \models_{\mathrm{FT}_\leq} a(y[\pi])$. Our assumption of entailment with respect to $\mathrm{FT}_\leq$ and $y \in \mathcal{V}(\varphi)$ yields:

$$\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi' \models_{\mathrm{FT}_\leq} a(y[\pi])$$

Since α is a solution of $\eta_{g,b}(\varphi)$ it is also a solution of φ such that $(\pi, a) \in L_{\alpha(y)} = L_{\alpha'(y)}$.

L2 Let $(\pi g, b) \in \alpha'(x)$ because $\varphi' \vdash x[\pi]\downarrow$. Symmetrically to the previous case we can show that $\varphi \models_{\mathrm{FT}_\leq} y[\pi]\downarrow$. Proposition 5.1 implies $\varphi \vdash y[\pi]\downarrow$ and thus $\eta_{g,b}(\varphi) \models b(y[\pi g])$. Finally, α is a solution of $\eta_{g,b}(\varphi)$ such that $(\pi g, b) \in L_{\alpha(y)} = L_{\alpha'(y)}$.

L3 Let $(\pi, a) \in L_{\alpha'(x)}$ because there exist $z \in \mathcal{V}(\varphi)$, z', π_0, π_1, π_2 such that $\pi = \pi_1\pi_2$, $\varphi' \vdash z' \leq x[\pi_1]$, $\varphi' \vdash z?[\pi_0] \leq z'$, and $(\pi_0\pi_2, a) \in L_{\alpha(z)}$. In this case, $\varphi' \vdash z \leq y[\pi_1]$ and thus $\varphi' \models_{\mathrm{FT}_\leq} \exists z' (z?[\pi_0] \leq z' \wedge z' \leq y[\pi_1])$. Since $y, z \in \mathcal{V}(\varphi)$, we have:

$$\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi' \models_{\mathrm{FT}_\leq} \exists z' (z?[\pi_0] \leq z' \wedge z' \leq y[\pi_1])$$

Lemma 8.14, assumption $(\pi_0\pi_2, a) \in L_{\alpha(z)}$, and that α is a solution of φ imply $(\pi_1\pi_2, a) \in L_{\alpha(y)}$, *i.e.*, $(\pi, a) = L_{\alpha'(y)}$.

(c) Case $x \in \mathcal{V}(\varphi)$ and $y \in \{\bar{x}\}$. If $(\pi, a) \in L_{\alpha'(x)}$ then $(\pi, a) \in L_{\alpha(x)}$. Since $\varphi' \vdash x \leq y[\varepsilon]$, $\varphi' \vdash x?[\varepsilon] \leq x$ and $x \in \mathcal{V}(\varphi)$ L3 implies $(\varepsilon\pi, a) \in L_{\alpha'(y)}$.

(d) If $x, y \in \mathcal{V}(\varphi)$ then $\varphi \models_{\mathrm{FT}_\leq} x \leq y$ such that $L_{\alpha(x)} \subseteq L_{\alpha(y)}$ and hence $L_{\alpha'(x)} \subseteq L_{\alpha'(y)}$.

2. Case $x[f]y$ *in* φ'. We have to show that $(f\pi, a) \in L_{\alpha'(x)}$ if and only if $(\pi, a) \in L_{\alpha'(y)}$. We first assume $(f\pi, a) \in L_{\alpha'(x)}$ and prove $(\pi, a) \in L_{\alpha'(y)}$.

(a) Case $x, y \in \{\bar{x}\}$.

L1 Let $(\pi, a) \in L_{\alpha'(y)}$ because $\varphi' \vdash a(y[\pi'])$. Hence, $\varphi' \vdash a(x[f\pi])$ such that $(f\pi, a) \in \alpha'(x)$.

L2 Let $(\pi, a) \in L_{\alpha'(y)}$ because $a = b$ and $\pi = \pi' g$ for some π' with $\varphi' \vdash y[\pi']\downarrow$. Hence, $\varphi' \vdash x[f\pi']\downarrow$ such that $(f\pi' g, b) \in L_{\alpha'(x)}$, *i.e.*, $(f\pi, a) \in L_{\alpha'(x)}$.

L3 Let $(\pi, a) \in L_{\alpha'(y)}$ because there exists $z \in \mathcal{V}(\varphi)$, z', π_0, π_1, and π_2 such that $\varphi' \vdash z' \leq y[\pi_1]$, $\varphi' \vdash z?[\pi_0] \leq z'$, $(\pi_1\pi_2, a) \in L_{\alpha(z)}$, and $\pi = \pi_1\pi_2$. Since $x[f]y$ we also have $\varphi' \vdash x?[f\pi_1] \leq z'$ which yields $(f\pi_1\pi_2, a) \in L_{\alpha'(x)}$, *i.e.*, $(f\pi, a) \in L_{\alpha'(x)}$.

(b) Case $x \in \{\bar{x}\}$ and $y \in \mathcal{V}(\varphi)$. If $(\pi, a) \in L_{\alpha'(y)}$ then $(\pi, a) \in L_{\alpha(y)}$. By applying L3 with $z = z' = y$, $\pi_0 = \varepsilon$, $\pi_1 = f$, and $\pi_2 = \pi$, we obtain $(f\pi, a) \in L_{\alpha'(x)}$.

(c) Case $x \in \mathcal{V}(\varphi)$ and $y \in \{\bar{x}\}$. Let $(\pi, a) \in L_{\alpha'(y)}$.

L1 If $\varphi' \vdash a(y[\pi])$ then $\varphi' \vdash a(x[f\pi])$. Since $x \in \mathcal{V}(\varphi)$ this yields $\varphi \models_{\mathrm{FT}_\leq} a(x[\pi])$ such that $(f\pi, a) \in L_{\alpha(x)}$, *i.e.*, $(f\pi, a) \in L_{\alpha'(x)}$.

L2 Let $(\pi, a) \in L_{\alpha'(y)}$ because $a = b$ and $\pi = \pi' g$ for some π' with $\varphi' \vdash y[\pi']\downarrow$, Hence $\varphi' \vdash x[f\pi']\downarrow$ such that $\varphi \models_{\mathrm{FT}_\leq} x[f\pi']\downarrow$ since $x \in \mathcal{V}(\varphi)$. Proposition 5.1 yields $\varphi \vdash x[f\pi]\downarrow$ and thus $\eta_{g,b}(\varphi) \vdash b(x[f\pi g])$. Since α is a solution of $\eta_{g,b}(\varphi)$, we have $(f\pi' g, b) \in L_{\alpha(x)}$, *i.e.*, $(f\pi, a) \in L_{\alpha'(x)}$.

L3 Let $(\pi, a) \in L_{\alpha'(y)}$ because there exists $z \in \mathcal{V}(\varphi)$, z', π_0, π_1, and π_2 such that $\varphi' \vdash z' \leq y[\pi_1]$, $\varphi' \vdash z?[\pi_0] \leq z'$, $(\pi_0\pi_2, a) \in L_{\alpha(z)}$, and $\pi = \pi_1\pi_2$. Since $x[f]y$ we also have $\varphi' \vdash z' \leq x[f\pi_1]$. Since $x, z \in \mathcal{V}(\varphi)$ this implies $\varphi \models_{\mathrm{FT}_\leq} \exists z' (z?[\pi_0] \leq z' \wedge z' \leq x[f\pi_1])$. Now, Lemma 8.14 and $(\pi_0\pi_2, a) \in L_{\alpha(z)}$ imply $(f\pi_1\pi_2, a) \in L_{\alpha(x)}$, *i.e.*, $(f\pi, a) \in L_{\alpha'(x)}$.

(d) Case $x, y \in \mathcal{V}(\varphi)$. In this case $\varphi \models_{\mathrm{FT}_\leq} x[f]y$ such that if $(\pi, a) \in L_{\alpha(y)}$ then $(f\pi, a) \in L_{\alpha(x)}$.

For the converse of the case $x[f]y \in \varphi'$, we assume $(f\pi, a) \in L_{\alpha'(x)}$ and prove $(\pi, a) \in L_{\alpha'(y)}$.

(a) Case $x, y \in \{\bar{x}\}$.

L1 Let $(f\pi, a) \in L_{\alpha'(x)}$ because and $\varphi' \vdash a(x[f\pi'])$. Since φ' is F1-F2-closed and satisfiable, this implies $\varphi' \vdash a(y[\pi])$ such that $(\pi, a) \in \alpha'(y)$.

L2 Let $(f\pi, a) \in L_{\alpha'(x)}$ because $f\pi = \pi' g$, $a = b$ and $\varphi' \vdash x[\pi']\downarrow$. Our assumption $g \notin \mathcal{F}(\varphi')$ implies $f \neq g$ such that there exists π'' with $\pi = \pi'' g$ and $\pi' = f\pi''$. Since φ' is closed under F2, the assumptions $\varphi' \vdash x[f\pi'']\downarrow$ and $x[f]y$ *in* φ' imply $\varphi' \vdash y[\pi'']\downarrow$. Hence, $(\pi'' g, b) \in L_{\alpha'(y)}$, *i.e.*, $(\pi, a) \in L_{\alpha'(y)}$.

L3 Let $(f\pi, a) \in L_{\alpha'(x)}$ because there exist $z \in \mathcal{V}(\varphi)$, z', and π_0, π_1, π_2 such that $f\pi = \pi_1\pi_2$, $\varphi' \vdash z \leq x[\pi_1]$, $\varphi' \vdash z?[\pi_1] \leq z'$, and $(\pi_0\pi_2, a) \in L_{\alpha(z)}$.

A. If $\pi_1 = \varepsilon$ then $f\pi = \pi_2$ and $z' \leq x$ *in* φ'. Hence $\varphi' \vdash y \leq z[\pi_0 f]$ and $\varphi' \vdash a(z[\pi_0 f\pi])$. Trivially $\varphi' \vdash y?[\varepsilon] \leq y$, since $y \in \mathcal{V}(\varphi') \subseteq \mathcal{V}(\varphi)$ such that L3 implies $(\varepsilon\pi, a) \in L_{\alpha(y)}$.

B. Otherwise, $\pi_1 = f\pi_1'$ and $\pi = \pi_1'\pi_2$ for some π'. The F1-F2 closedness and satisfiability of φ' implies that $\varphi' \vdash y?[\pi_1'] \leq z'$. Hence, L3 yields $(\pi_1'\pi_2, a) \in L_{\alpha'(y)}$, *i.e.*, $(\pi, a) \in L_{\alpha'(y)}$.

(b) Case $x \in \{\bar{x}\}$ and $y \in \mathcal{V}(\varphi)$.

L1 Let $(f\pi, a) \in L_{\alpha'(x)}$ because $\varphi' \vdash a(x[f\pi])$. The F1-F2 closedness of φ implies $\varphi' \vdash a(y[\pi])$ and hence $\varphi \models_{\mathrm{FT}_\leq} a(y[\pi])$ since $y \in \mathcal{V}(\varphi)$. For α being a solution of φ this implies $(\pi, a) \in L_{\alpha(y)}$ which is equivalent to $(\pi, a) \in L_{\alpha'(y)}$.

L2 Let $(f\pi, a) \in L_{\alpha'(x)}$ because $f\pi = \pi' g$, $a = b$ and $\varphi' \vdash x[\pi']\downarrow$. Since $f \neq g$ there exists there exists π'' such that $\pi = \pi'' g$ and $f\pi'' = \pi'$. By F2-closedness and $x[f]y \in \varphi$ we obtain $\varphi' \vdash y[\pi'']\downarrow$. The definition of $\eta_{g,b}(\varphi)$ yields $\varphi \vdash b(y[\pi' g])$. Hence, $(\pi'' g, b) \in L_{\alpha(y)}$ which is equivalent to $(\pi, a) \in L_{\alpha'(y)}$.

L3 Let $(f\pi, a) \in L_{\alpha'(x)}$ because there exist $z \in \mathcal{V}(\varphi)$, z' and π_0, π_1, π_2 such that $f\pi = \pi_1\pi_2$, $\varphi' \vdash z' \leq x[\pi_1]$, $\varphi' \vdash z?[\pi_0] \leq z'$, and $(\pi_0\pi_2, a) \in L_{\alpha(z)}$.

A. Case $\pi_1 = \varepsilon$. Hence $f\pi = \pi_2$ and $z' \leq x$ *in* φ'. In this case $\varphi' \vdash z?[\pi_0 f] \leq y$ such that $z, y \in \mathcal{V}(\varphi)$ implies

$$\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi' \models_{\mathrm{FT}_\leq} z?[\pi_0 f] \leq y$$

Since α is a solution of φ and $(\pi_0 f\pi, a) \in L_{\alpha(z)}$. Lemma 8.14 implies $(\pi, a) \in L_{\alpha(y)}$, *i.e.*, $(\pi, a) \in L_{\alpha'(y)}$.

B. Otherwise there exists π_1' such that $\pi_1 = f\pi_1'$ and $\pi = \pi_1'\pi_2$. The F1-F2 closedness of φ' and $\varphi' \vdash z' \leq x[f\pi_1']$ imply $\varphi' \vdash z' \leq y[\pi_1']$. Since $y, z \in \mathcal{V}(\varphi)$ we know that

$$\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi' \models_{\mathrm{FT}_\leq} \exists z' \, (z' \leq z[\pi_0] \wedge y?[\pi_1'] \leq z')$$

Hence, our assumption $(\pi_0\pi_2, a) \in L_{\alpha(z)}$ and Lemma 8.14 imply $(\pi_1'\pi_2, a) \in \alpha(y)$, *i.e.*, $(\pi, a) \in \alpha'(y)$.

(c) Case $x \in \mathcal{V}(\varphi)$ and $y \in \{\bar{x}\}$. If $(f\pi, a) \in L_{\alpha'(x)}$ then $(f\pi, a) \in L_{\alpha(x)}$. L3, $y \notin \mathcal{V}(\varphi)$, and $x \in \mathcal{V}(\varphi)$ imply $(\pi, a) \in L_{\alpha'(y)}$.

(d) Case $x, y \in \mathcal{V}(\varphi)$. In this case $\varphi \models_{\mathrm{FT}_\leq} x[f]y$ such that if $(f\pi, a) \in L_{\alpha(x)}$ then $(\pi, a) \in L_{\alpha(y)}$.

3. Case $a(x)$ *in* φ'.

(a) If $x \in \{\bar{x}\}$ then $\varphi' \vdash a(x[\varepsilon])$ such that $(\varepsilon, a) \in L_{\alpha'(x)}$ (condition L1).

(b) If $x \in \mathcal{V}(\varphi)$ then $\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi' \models_{\mathrm{FT}_\leq} a(x)$. Hence $(\varepsilon, a) \in \alpha(x)$, *i.e.*, $(\varepsilon, a) \in \alpha'(x)$.

□

9 Conclusion and Future Work

We have investigated decidability and complexity questions for fragments of the first-order theory of ordering constraints over feature trees ($\mathrm{FT}_\leq$). We have proved that the entailment problem of $\mathrm{FT}_\leq$ with existential quantifiers, $\mathrm{Ent}_\exists(\mathrm{FT}_\leq)$, is PSPACE-hard, both for the case of finite and infinite trees, and also decidable in both cases.

Our main contribution in this paper is that we have related $\mathrm{FT}_\leq$ to the monadic second-order logic S2S, that enabled us to solve the first non-trivial decision problem over feature trees by reduction to Rabin's famous decidability result for (W)SωS. As the main handle on the proof we distinguished a constraint system $\mathrm{FT}_\leq^-$ whose first-order theory, $\mathrm{FO}(\mathrm{FT}_\leq^-)$, is equivalent to S2S and whose entailment problem $\mathrm{Ent}_\exists(\mathrm{FT}_\leq^-)$

coincides with the corresponding one of $\mathrm{FT}_{\leq}$. In summary, we have completed the following picture.

$$\mathrm{Ent}_{\exists}(\mathrm{FT}_{\leq}) \;\leftrightarrow\; \mathrm{Ent}_{\exists}(\mathrm{FT}_{\leq}^{-}) \;\subseteq\; FO(\mathrm{FT}_{\leq}^{-}) \;=\; \mathrm{S2S} \;\hookrightarrow\; FO(\mathrm{FT}_{\leq})$$

In more recent work (Müller et al., 1998) we have shown that the first-order theory of $\mathrm{FT}_{\leq}$, $\mathrm{FO}(\mathrm{FT}_{\leq})$, is undecidable in contrast to the first-order theory of $\mathrm{FT}_{\leq}^{-}$. Hence $\mathrm{FO}(\mathrm{FT}_{\leq})$ cannot be embedded into S2S.

It remains open to find a more direct relation between FO(FT) and S2S.

$$FO(\mathrm{FT}_{\leq}) \hookleftarrow FO(FT) \overset{?}{\sim} \mathrm{S2S}$$

Another open question is to find larger decidable fragments of the first-order theory of $\mathrm{FT}_{\leq}$ for which entailment is decidable. this also includes the decidability question for the corresponding entailment problem of CFT which can be expressed in the first-order theory of $\mathrm{FT}_{\leq}$ (Müller et al., 1998).

Acknowledgments. We thank Jean-Marc Talbot, Sophie Tison, and Marc Tommasi for inspiring discussions. We thank Ralf Treinen for allowing us to present the encoding of second-order monadic logic into $\mathrm{FT}_{\leq}$ in this paper. Furthermore, we thank Ralf Treinen and the anonymous referees for careful reading and invaluable remarks on this paper. The research reported here has been supported by the the Esprit Working Group CCL II (EP 22457) and the DFG through SFB 378 at the Universität des Saarlandes.

References

Aït-Kaci, H., & Podelski, A. (1993). Towards a meaning of life. *The Journal of Logic Programming*, *16*(3 – 4), 195-234.

Aït-Kaci, H., Podelski, A., & Smolka, G. (1994). A feature-based constraint system for logic programming with entailment. *Theoretical Computer Science*, *122*(1–2), 263–283.

Aït-Kaci, H. (1986). An algebraic semantics approach to the effective resolution of type equations. *Theoretical Computer Science*, *45*, 293–351.

Aït-Kaci, H., & Nasr, R. (1986). LOGIN: A logic programming language with built-in inheritance. *The Journal of Logic Programming*, *3*(3), 185-215.

Backofen, R. (1994). *Expressivity and decidability of first-order languages over feature trees.* Doctoral Dissertation. Universität des Saarlandes, Technische Fakultät, D–66041 Saarbrücken.

Backofen, R. (1995). A complete axiomatization of a theory with feature and arity constraints. *The Journal of Logic Programming*, *24*(1 – 2), 37–71. (Special Issue onComputational Linguistics and Logic Programming)

Backofen, R., & Smolka, G. (1995). A complete and recursive feature theory. *Theoretical Computer Science*, *146*(1–2), 243–268.

Brachman, R. J., & Levesque, H. J. (1984). The tractability of subsumption in frame-based description languages. In *Proceedings of the national conference on artificial intelligence* (pp. 34–37).

Carpenter, B. (1992). *The logic of typed feature structures* (Vol. 32). Cambridge, UK: Cambridge University Press.

Colmerauer, A. (1984). Equations and inequations on finite and infinite trees. In ICOT (Ed.), *Proceedings of the 2nd international conference on fifth generation computer systems* (pp. 85–99). Omsha Ltd., Tokyo and North-Holland, Amsterdam.

Comon, H. (1995). Sequentiality, second-order monadic logic, and tree automata. In *Proceedings of the 10th IEEE symposium on logic in computer science* (pp. 508–517). San Diego, California: IEEE Computer Society Press.

Comon, H., Dauchet, M., Gilleron, R., Lugiez, D., Tison, S., & Tommasi, M. (1998). *Tree automata: Techniques and applications.* Lille, France. (Book Draft)

Doner, J. (1970). Tree acceptors and some of their applications. *Journal of Computing Systems Science*, *4*, 406–451.

Dörre, J. (1991). Feature logics with weak subsumption constraints. In *Annual meeting of the association of computational linguistics* (p. 256-263).

Dörre, J. (1993). *Feature-Logik und Semiunfikation.* Doctoral dissertation, Philosophische Fakultät der Universität Stuttgart. (In German)

Dörre, J., & Rounds, W. C. (1992). On subsumption and semi-unification in feature algebras. *Journal of Symbolic Computation*, *13*, 441–461.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness.* W.H. Freeman and Company, New York.

Gecseg, F., & Steinby, M. (1984). *Tree automata.* Akad. Kiado, Budapest.

Hopcroft, J. D., & Ullman, J. D. (1979). *Introduction to automata theory, languages and computation.* Addison-Wesley, Reading, MA.

Kaplan, R. M., & Bresnan, J. (1982). Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), *The mental representation of grammatical relations* (p. 173-281). The MIT Press, Cambridge, MA.

Kay, M. (1979). Functional grammar. In C. Chiarello et al. (Ed.), *Proceedings of the 5th annual meeting of the Berkeley linguistics society* (pp. 142–158).

Mukai, K. (1988). Partially specified terms in logic programming for linguistic analysis. In *Proceedings of the 6th international conference on fifth generation computer systems.* Tokyo, Japan: ICOT.

Müller, M. (LNAI, to appear). Ordering constraints over feature trees with ordered sorts. In P. Lopez, S. Manandhar, & W. Nutt (Eds.), *Computational logic and natural language understanding.* Springer-Verlag, Berlin. (Available at `http://www.ps.uni-sb.de/~mmueller/papers/clnlp.ps.Z`)

Müller, M., & Niehren, J. (1998). Ordering constraints over feature trees expressed in second-order monadic logic. In T. Nipkow (Ed.), *International conference on rewriting techniques and applications* (LNCS, Vol. 1379, pp. 196–210). Tsukuba, Japan: Springer-Verlag, Berlin.

Müller, M., Niehren, J., & Podelski, A. (1997). Ordering constraints over feature trees. In G. Smolka (Ed.), *Proceedings of the 3^rd^ international conference on principles and practice of constraint programming* (LNCS, Vol. 1330, pp. 297–311). Schloß Hagenberg, Linz, Austria: Springer-Verlag, Berlin. (Full version submitted to special journal issue of CP'97)

Müller, M., Niehren, J., & Treinen, R. (1998). The first-order theory of ordering constraints over feature trees. In *Proceedings of the 13^th^ IEEE symposium on logic in computer science* (pp. 432–443). IEEE Computer Society Press.

Nebel, B. (1990). *Reasoning and revision in hybrid representation systems* (Vol. 422). Springer-Verlag, Berlin.

Nebel, B., & Smolka, G. (1990). Representation and reasoning with attributive descriptions. In K. Bläsius, U.Hedtstück, & C.-R. Rollinger (Eds.), *Sorts and types in artificial intelligence* (Vol. 418, pp. 112–139). Springer-Verlag, Berlin.

Pollard, C., & Sag, I. (1994). *Head-driven phrase structure grammar.* Cambridge University Press, Cambridge, England.

Rabin, M. O. (1969). Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, *141*, 1–35.

Rounds, W. C. (1997). Feature logics. In J. v. Benthem & A. ter Meulen (Eds.), *Handbook of logic and language* (pp. 475–533). Elsevier Science Publishers B.V. (North Holland). (Part 2: General Topics)

Seidl, H. (1990). Deciding equivalence of finite tree automata. *SIAM Journal of Computing*, *19*(3), 424–437.

Shieber, S. (1986). *An introduction to unification-based approaches to grammar.* Center for the Study of Language and Information.

Shieber, S., Uszkoreit, H., Pereira, F., Robinson, J. A., & Tyson, M. (1983). The formalism and implementation of PATR-II. In J. Bresnan (Ed.), *Research on interactive acquisition and use of knowledge.* Menlo Park, California: SRI International.

Smolka, G. (1995). The Oz Programming Model. In J. van Leeuwen (Ed.), *Computer science today* (LNCS, Vol. 1000, pp. 324–343). Springer-Verlag, Berlin.

Smolka, G., & Treinen, R. (1994). Records for logic programming. *The Journal of Logic Programming*, *18*(3), 229–258.

Thatcher, J. W., & Wright, J. B. (1968). Generalized finite automata theory with an application to a decision problem of second-order logic. *Mathematical Systems Theory*, *2*(1), 57–81.

Thomas, W. (1990). Automata on infinite objects. In J. van Leeuwen (Ed.), *Handbook of theoretical computer science* (Vol. B (Formal Models and Semantics), pp. 133–191). The MIT Press, Cambridge, MA.

Thomas, W. (1997). Languages, automata, and logic. In G. Rozenberg & A. Salomaa (Eds.), *Handbook of formal languages* (Vol. 3). Springer-Verlag, Berlin.

Vorobyov, S. (1996). An improved lower bound for the elementary theories of trees. In *Proceedings of the conference on automated deduction* (LNCS, Vol. 1104, p. 275-287). Rutgers, NJ: Springer-Verlag, Berlin.