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

September 24, 1999

Abstract

The language $FT_{\leq}$ 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 decidability results are known for the first-order theory of $FT_{\leq}$. We introduce a new method for proving the decidability of fragments of the first-order theory of $FT_{\leq}$. This method is based on reduction to second order monadic logic that is decidable according to Rabin's famous tree theorem. The method applies to any fragment of the first-order theory of $FT_{\leq}$ for which one can change the model towards sufficiently labeled feature trees – a class of trees that we introduce. As we show, the first order-theory of ordering constraints over sufficiently labeled feature trees is equivalent to second-order monadic logic (S2S for infinite and WS2S for finite feature trees). We apply our method for proving that entailment of $FT_{\leq}$ with existential quantifiers $\varphi_1 \models \exists x_1 \ldots \exists x_n \varphi_2$ is decidable. Previous results were restricted to entailment without existential quantifiers which can be solved in cubic time. Meanwhile, entailment with existential quantifiers has been shown PSPACE-complete (for finite and infinite feature trees respectively).

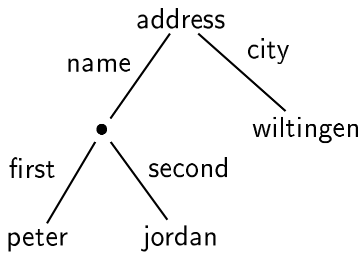
Keywords Feature logic, tree orderings, entailment, decidability, complexity, second-order monadic logic.

1 Introduction

Feature logic is a formalism to describe objects by the values of their attributes or *features*. It has its roots 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, especially approaches based on *unification grammars* [Kay, 1979, Kaplan & Bresnan, 1982, Shieber *et al.*, 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].

Two main approaches to feature logics can be distinguished according to the underlying logical structures. Both approaches rely on quite similar syntactic constructions called *feature constraints* but differ significantly in their semantics. In the HPSG-community in computational linguistics [Pollard & Sag, 1994, Carpenter, 1992], feature constraints are typically interpreted over so-called *feature structures* (see below). In programming language research [Aït-Kaci *et al.*, 1994, Backofen & Smolka, 1995, Backofen, 1994, Backofen, 1995], feature constraints are usually interpreted in a single structure, the structure of *feature trees*. A feature tree can be seen as a *record* and a feature constraint as a record description. In this article, we follow the approach where feature constraints are interpreted over feature trees.

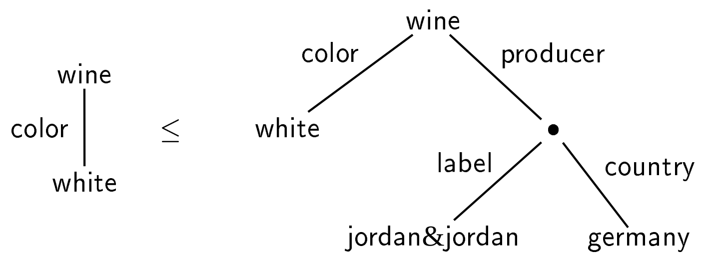
Feature Trees. We assume a set of *features* and a set of *node labels*. An (optionally labeled) *feature tree* is a tree with unordered edges each of which is labeled by a *feature* and with nodes which may or may not be labeled by a node label. Features are functional in that all features labeling 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 Mr. Peter Jordan in Wiltingen:



A feature tree is defined by a *tree domain* specifying its *nodes* and a *labeling function*. The idea is that a *node* in a tree is identified with the *path* by which it can be accessed from the root of the tree. Given this, a tree domain becomes a *prefixed closed* set of words over features. A labeling function specifies the subset all labeled nodes and a node label for each of these nodes. Thus, the *labeling function* of a tree becomes a partial function from its tree domain to the set of node labels.
The tree domain of the above tree is the set $\{\varepsilon, \mathsf{name}, \mathsf{name\,first}, \mathsf{name\,second}, \mathsf{city}\}$. Note that the node (addressed by the word) name is unlabeled, *i.e.*, the labeling function is undefined for this element of the tree domain.

Information Ordering. Feature trees and feature structures can be ordered in an natural way by comparing the amount of the information they carry. On feature trees, this leads to a partial ordering that is called *information ordering* [Müller *et al.*, 2000] or equivalently *weak subsumption* [Dörre, 1991]. On feature structures, another partial ordering relation is obtained which is called *strong subsumption* [Dörre & Rounds, 1992]. It is also possible to define strong subsumption for feature trees and weak subsumption for feature structures, even though this seems to be less natural. In this article, we focus on feature trees with weak subsumption. A closer comparison to strong subsumption is given in the paragraph on feature structures below.
Intuitively, a feature tree τ_1 is smaller than a feature tree τ_2 with respect to the information 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 below.



Ordering Constraints over Feature Trees. We investigate the system $FT_{\leq}$ of ordering constraints over feature trees [Müller *et al.*, 2000, Müller, to appear, Müller & Niehren, 1998, Müller *et al.*, 1998]. The feature constraints provided by $FT_{\leq}$ are constructed from variables ranged over by x,y, features f and node labels a. The abstract syntax of *ordering constraints* φ in the language $FT_{\leq}$ is defined as follows:

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

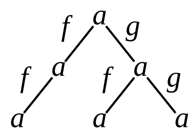
The semantics of $FT_{\leq}$ is given by the interpretation in the structure of feature trees where the symbol $\leq$ is interpreted as the information ordering. The semantics of feature selection $x[f]y$ and labeling constraints $a(x)$ is defined as usual. For instance, both trees in the picture above are possible denotations for x in solutions of the constraint $\mathsf{wine}(x) \wedge x[\mathsf{color}]y \wedge \mathsf{white}(y)$. 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 $FT_{\leq}$ 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 $FT_{\leq}$ 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 $FT_{\leq}$. Equalities are expressible in $FT_{\leq}$ since the equivalence $x{=}y \leftrightarrow x{\leq}y \wedge y{\leq}x$ is valid in $FT_{\leq}$.

Decidability and Complexity. The first-order theory of equality constraints 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 its 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 ordering constraints in $FT_{\leq}$. Previously, the entailment problem $\varphi \models \varphi'$ of $FT_{\leq}$ was shown to have cubic time complexity [Müller *et al.*, 2000] but decidability for more expressive fragments of the first-order theory of $FT_{\leq}$ was left open.

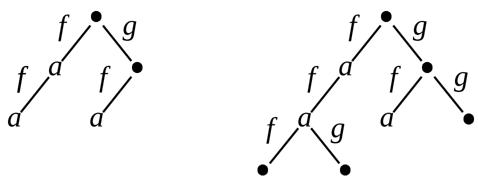
Relationship to Second-Order Monadic Logic. In this article, we present a new method for proving the decidability of fragments of the first-order theory of $FT_{\leq}$. To this end, we pursue a general approach which is based on reduction to second order monadic logics, WS2S for finite feature trees and S2S for infinite feature trees. The decidability of WS2S is well-known and follows from a classical reduction to the emptiness problem of tree automata [Thatcher & Wright, 1968, Doner, 1970, Gecseg & Steinby, 1984, Comon *et al.*, 1998]. The decidability of S2S is a classical consequence of Rabin's famous theorem on automata for infinite trees [Rabin, 1969, Thomas, 1990, Thomas, 1997].

We express feature constraints in second-order monadic logic according to a well-known idea: we identify a feature tree with a set of words and express feature constraints by formulas of (W)S2S. The same idea for constructor trees (ground terms) can be found in [Comon, 1995]. Let us assume for simplicity that the set of labels $\mathcal{L}$ is the singleton $\mathcal{L} = \{a\}$. Under this assumption, a *completely labeled* feature tree (whose labeling function is total) can be identified with its tree domain, i.e with a *prefixed closed* set of words. For instance:

$\{\varepsilon, f, g, gf, gg\}$ $\Leftrightarrow$



An optionally labeled feature tree (as considered in this article) can represent an arbitrary set of words, but several trees may correspond to the same set. For instance, the set $\{f, ff, gf\}$ can be represented by the set of labeled nodes in either of the following two trees:



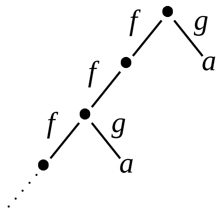
If we seek for a unique feature tree to represent the set $\{f, ff, gf\}$ then it might seem appropriate to choose the smallest feature tree whose set of a-labeled nodes equals $\{f, ff, gf\}$, *i.e.*, the tree which above is depicted to the left. This tree can also be distinguished by the property of being *sufficiently labeled*, *i.e.*, its tree domain is the prefixed-closure of the set of its labeled nodes. Thus, we can represent a sets uniquely by an optionally labeled feature tree if we require prefixed closedness for the set of labeled nodes.

To conclude, it seems that we have to require prefixed-closedness in order to encode sets of words as feature trees, independently of whether we work with completely or optionally labeled feature trees. There raises an intriguing problem with our approach of expressing feature constraints in (W)S2S. It seems difficult, if not impossible, to express prefixed-closedness and feature selection $\tau[f]\tau'$ in (W)SωS simultaneously. To express prefixed-closedness in (W)SωS, one needs concatenation πf of words with letters to the right, whereas feature selection requires concatenation $f\pi$ of words with letters to the left. The logics (W)S2S provides at most one of the two forms of concatenation and becomes undecidable when extended with the omitted one (see for instance [Thomas, 1990], Theorem 11.6). As a consequence, nobody has so far been able to use (W)SωS for deriving decidability results for feature logics. In particular, the first-order theory of FT could not be embedded into (W)S2S. In fact, it seems already difficult to express first-order equations like $x = a(y, a(y, y))$ in terms of W(S2S) or tree automata [Comon *et al.*, 1998].

In this article, we propose an work around to the above problem. We define the language $\mathrm{FT}^{-}_{\leq}$ of *ordering constraint over sufficiently labeled feature trees*. The syntax of $\mathrm{FT}^{-}_{\leq}$ coincides with the syntax of $\mathrm{FT}_{\leq}$ but its semantics is based on the restricted structure of feature trees. The new observation underlying the approach to be presented is that prefixed-closedness has not to be expressed when encoding feature constraints over $\mathrm{FT}^{-}_{\leq}$ into (W)S2S (in contrast to $\mathrm{FT}_{\leq}$). For finite feature trees, first-order formulas over $\mathrm{FT}^{-}_{\leq}$ can be translated into WS2S; for possibly infinite feature trees, the full power of S2S is needed.

A finite tree is sufficiently labeled if and only if all its leaves are labeled; inner nodes may or may not be labeled. The analogous characterization is not valid for infinite feature trees since these need not have leaves at all. As a counter example, consider the following infinite feature tree which sufficiently labeled since it has sufficiently many labeled leaves, namely the paths $f^{2n}g$ for all $n \geq 0$.

$$\{ f^{2n}g \mid n \geq 0 \} \qquad \Leftrightarrow$$



Most importantly, a sufficiently labeled tree is uniquely determined by its labeling function. For instance, the above tree is the unique sufficiently labeled tree whose set of a-labeled nodes is $\{ f^{2n}g \mid n \geq 0 \}$ provided that the set of labels $\mathcal{L}$ is the singleton $\mathcal{L} = \{a\}$.

In this article, we will show how to encode the first-order theory of $\mathrm{FT}^{-}_{\leq}$ into (W)S2S without expressing prefixed-closedness, and vice versa (see Theorem 4.2). Let $k = |\mathcal{F}|$ be denote the cardinality of set of features $\mathcal{F}$. Our reductions apply if $k \geq 2$ and $k \leq \omega$, *i.e.*, if the set of features $\mathcal{F}$ is at most countably infinite. For finite trees, we reduce the first-order theory of $\mathrm{FT}^{-}_{\leq}$ to WSkS, the weak second-order monadic logic with k successors which can in turn be expressed in WS2S, the weak second-order monadic with two successors [Thatcher & Wright, 1968]. In the case of possibly infinite trees we reduce to SkS, the full second-order monadic logic with k successors which, in analogy, is expressible in S2S, the full second-order monadic logic with two successors [Rabin, 1969].

The first-order theory of $\mathrm{FT}^{-}_{\leq}$ can be embedded into the first-order theory of $\mathrm{FT}_{\leq}$ since the latter can express sufficient labeling (see Proposition 3.3). We thereby obtain the following relationships (where FO stands for first-order theory):

$$(W)\mathrm{S2S} \quad = \quad FO(\mathrm{FT}^{-}_{\leq}) \quad \subseteq \quad FO(\mathrm{FT}_{\leq})$$

These relations suggest a method for proving the decidability of a fragment of the first-order theory of $\mathrm{FT}_{\leq}$: simply encode the fragment of $\mathrm{FT}_{\leq}$ into the corresponding fragment of $\mathrm{FT}^{-}_{\leq}$. This induces an encoding into (W)S2S which is decidable.

Entailment of $\mathrm{FT}_{\leq}$ with Existential Quantifiers. We consider the entailment problem $\varphi \models \exists x_1 \ldots \exists x_n \varphi'$ for $\mathrm{FT}_{\leq}$ with existential quantifiers. Without existential quantifiers the problem can be solved in cubic time [Müller *et al.*, 2000]. But with existential quantifiers, entailment becomes surprisingly hard. We will illustrate how to prove PSPACE hardness in case of infinite feature trees and coNP hardness for finite trees.

The new difficulty can be illustrated by considering the independence property [Colmerauer, 1984]: if $\varphi \models_{\mathrm{FT}_\leq} \bigvee_{i=1}^{n} \varphi_i$ then there exists i, $1 \leq i \leq n$, such that $\varphi \models_{\mathrm{FT}_\leq} \varphi_i$. Independence holds for the language of ordering constraints in $\mathrm{FT}_\leq$ [Müller *et al.*, 2000] but fails when existential quantifiers are admitted. To see this, notice that the entailment judgment

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

is valid in $\mathrm{FT}_\leq$ while the left hand side $x{\leq}y \wedge a(y)$ does neither entails $a(x)$ nor $\exists z(x{\leq}z \wedge b(z))$ on the right hand side (provided that $a \neq b$).

In this article, 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. We apply the method sketched above and show that entailment for $\mathrm{FT}_\leq^-$ with existential quantifiers can be reduced to the corresponding problem for $\mathrm{FT}_\leq^-$ but in a non-trivial way (see Proposition 6.8). In the case of finite trees, we obtain a reduction to WS2S and for infinite trees into S2S. Proving the correctness of Proposition 6.8 is involved. The problem is that entailment with existential quantifiers differs for the structures $\mathrm{FT}_\leq$ and $\mathrm{FT}_\leq^-$. For instance, true $\models \exists x(x{\leq}x_1 \wedge x{\leq}x_2)$ holds for $\mathrm{FT}_\leq$ but *not* over $\mathrm{FT}_\leq^-$. We will explain this example in Section 6.2 and also how the problem can be overcome.

Recent Developments. The presented article has emerged from an earlier conference paper [Müller & Niehren, 1998]. In comparison, the article is extended by complete proofs and the following new result:

1. (W)S2S can be encoded into the first-order theory of $\mathrm{FT}_\leq$ or $\mathrm{FT}_\leq^-$.

The present article leaves two open questions which have meanwhile been answered:

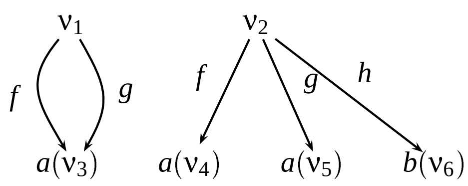
2. Is the first-order theory of $\mathrm{FT}_\leq$ decidable or even expressible in (W)S2S?

3. What is the precise complexity of entailment of $\mathrm{FT}_\leq$ with existential quantifiers?

In what concerns question 2, it is shown in [Müller *et al.*, 1998] that the first-order theory of $\mathrm{FT}_\leq$ is in fact undecidable. Thus, it cannot be expressed in (W)S2S. In the same paper the answer to question 3 is also given: Entailment of $\mathrm{FT}_\leq$ with existential quantifiers is PSPACE-complete, both in the case of finite trees and for possibly infinite trees. A lot of additional machinery and ideas is needed for the latter result. Proving PSPACE-completeness for entailment with existential quantifiers requires a direct algorithm rather than an encoding into second order monadic logic. Also the PSPACE-hardness proof for entailment over finite feature trees is somewhat tedious.

Feature Structures versus Feature Trees. In the HPSG-driven approach pursued in computational linguists [Pollard & Sag, 1994, Carpenter, 1992], feature constraints are typically interpreted over so-called feature structures.

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



In this context, two partial orderings have been introduced [Dörre, 1993]: weak subsumption [Dörre, 1991] and strong subsumption [Dörre & Rounds, 1992]. In order to illustrate the difference, we give definitions for both partial orderings which are equivalent to those in the literature.

For every node ν of $\mathcal{N}$ let $\mathsf{graph}_{\mathcal{N}}(\nu)$ be the subgraph of $\mathcal{N}$ reachable from ν and let $\mathsf{tree}_{\mathcal{N}}(\nu)$ be the feature tree obtained from unfolding $\mathsf{graph}_{\mathcal{N}}(\nu)$. For all nodes ν, ν' of $\mathcal{N}$ we say that *weak subsumption* $\nu \leq_{\text{weak}} \nu'$ holds iff $\mathsf{tree}_{\mathcal{N}}(\nu) \leq \mathsf{tree}_{\mathcal{N}}(\nu')$ is valid with respect to the information ordering on feature trees. 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 $\mathsf{graph}_{\mathcal{N}}(\nu)$ can be homomorphically embedded into $\mathsf{graph}_{\mathcal{N}}(\nu')$. For instance, $\nu_3 \leq_{\text{strong}} \nu_4$ but not $\nu_1 \leq_{\text{strong}} \nu_2$.

Plan of the Paper. Section 2 recalls the definition of ordering constraints over feature trees and gives lower complexity bounds for entailment with existential quantifies. Section 3 investigates alternative structures of feature trees in some more detail. Section 4 presents a collection of results on the relationship between the first-order theories of $\mathrm{FT}_{\leq}$ and $\mathrm{FT}_{\leq}^{-}$, and second-order monadic logics. Section 5 recalls some relevant results on satisfiability from [Müller *et al.*, 2000]. Sections 6 and 7 present and prove correct our reduction of entailment with existential quantifiers in $\mathrm{FT}_{\leq}$ to the corresponding problem in $\mathrm{FT}_{\leq}^{-}$. In Section 8 we complete some less exiting proofs. Section 9 summarizes and concludes.

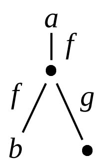
2 Syntax and Semantics

We assume an infinite set of *variables* ranged over by x, y, z, a countable sets $\mathcal{F}$ with at least two elements that are called *features* and ranged over by f, g, and a finite set $\mathcal{L}$ with at least two elements called *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 prefixed-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. We freely consider a labeling function $L : D \rightharpoonup \mathcal{L}$ as a binary relation $L \subseteq D \times \mathcal{L}$ whenever this is more convenient. 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 $L_\tau(\pi) = a$. A node of τ is unlabeled if it is not labeled by any label $a \in \mathcal{L}$.
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 occurring in some feature tree τ is denoted by $\mathcal{F}(\tau)$, *i.e.*, $\mathcal{F}(\tau) = \{f \mid \pi f \pi' \in D_\tau\}$. Given a function $\alpha : \mathcal{V} \to \mathrm{FT}_\leq$ and a set of variables $V \subseteq \mathcal{V}$ we define $\mathcal{F}_V(\alpha)$ by $\bigcup_{x \in V} \mathcal{F}(\alpha(x))$.

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

We consider two cases, the structure of possibly infinite feature trees $\mathrm{FT}_\leq$ 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 interpreted as the following relations between feature trees. For all τ, τ_1, τ_2, we define:

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

The structure $\mathrm{FT}_\leq^{fin}$ is the restriction of the structure $\mathrm{FT}_\leq$ to the domain of finite feature trees.
If τ is a tree and $f \in D_\tau$ a feature in its tree domain there we write $\tau[f]$ for the *subtree* of τ at feature f, i.e. $\tau' = \tau[f]$ is the unique freature tree satisfying $\tau[f]\tau'$.

2.3 Ordering Constraints and First-Order Formulas

An *ordering constraint* φ of the constraint languages $\mathrm{FT}_\leq$ and $\mathrm{FT}_\leq^{fin}$ (we freely overload names of the structure and constraint language) 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 with Φ a first-order formula built from ordering constraints plus the usual first-order connectives. 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.

A *variable assignment* into a logical structure $\mathcal{A}$ (such as $FT_{\leq}$ or $FT_{\leq}^{fin}$) is a function α mapping variables to elements of the domain of $\mathcal{A}$. The truth value of a first-order formula Φ with the same signature than $\mathcal{A}$ under a variable assignment α into $\mathcal{A}$ is defined as usual for first-order languages. A *solution* α of Φ over $\mathcal{A}$ is a variable assignment into $\mathcal{A}$ that makes Φ true. We write $\alpha \models_{\mathcal{A}} \Phi$ if α is a solution of Φ over $\mathcal{A}$. We call Φ *satisfiable* in $\mathcal{A}$ if there exists a solution of Φ in $\mathcal{A}$ and *valid* in $\mathcal{A}$ if every variable assignment into $\mathcal{A}$ is a solution of Φ. We say that Φ *entails* Φ' over $\mathcal{A}$ and write $\Phi \models_{\mathcal{A}} \Phi'$ if every solution of Φ over $\mathcal{A}$ is a solution of Φ, *i.e.*, if the implication $\Phi \rightarrow \Phi'$ is valid over $\mathcal{A}$. We call Φ and Φ' *equivalent* over $\mathcal{A}$ if $\Phi_1 \leftrightarrow \Phi_2$ is valid.

An *n-ary predicate* $\mathcal{P}$ over a structure $\mathcal{A}$ is an n-ary relation between elements of the domain of $\mathcal{A}$. We write $\mathcal{P}(\tau_1, \dots, \tau_n)$ if $(\tau_1, \dots, \tau_n) \in \mathcal{P}$ for some element $\tau_1, \dots, \tau_n$ of the domain of $\mathcal{A}$. We denote a formula Φ with free variables $x_1, \dots, x_n$ with $\Phi(x_1, \dots, x_n)$ whereby an ordering on the variables of Φ is fixed.

Definition 2.1 *An n-ary predicate $\mathcal{P}$ over a structure $\mathcal{A}$ is* expressed *by a formula $\Phi(x_1, \dots, x_n)$ with over the signature of $\mathcal{A}$ if:*

$$\mathcal{P} = \{(\alpha(x_1), \dots, \alpha(x_n)) \mid \alpha \models_{\mathcal{A}} \Phi(x_1, \dots, x_n)\}$$

For the structure FT of feature trees with equality, this definition was investigated in depth by [Backofen, 1994].

2.4 Complexity of Entailment with Existential Quantifiers

Entailment between ordering constraints with existential quantifiers $\varphi \models \exists x_1 \dots \exists x_n \varphi'$ is a surprisingly hard problem. This problem is proved PSPACE complete in follow-up work [Müller *et al.*, 1998], both for finite and possibly infinite feature trees.

A simple proof for both PSPACE hardness in case of infinite trees was first given in [Müller & Niehren, 1998]. In case of finite trees, the analogous proof yields coNP hardness only. We illustrate the idea behind this proof in order to give an example for the expressiveness of entailment with existential quantifiers. Proving PSPACE hardness for finite trees is less obvious [Müller *et al.*, 2000].

We first consider the case of infinite trees and show how to express universality of finite automata by entailment with existential quantifiers. We fix a label a. For every finite automaton $\mathcal{A}$ we show how to express the following predicate $\mathcal{P}_{\mathcal{A}}$ of possibly infinite feature trees by an positive existential formula.

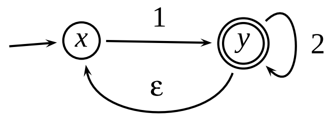
$$\mathcal{P}_{\mathcal{A}} =_{\text{def}} \{\tau \mid \text{for all } \pi \in \mathcal{L}(\mathcal{A}) : L_\tau(\pi) = a\}$$

Note that all trees in $\mathcal{P}_{\mathcal{A}}$ are infinite if $\mathcal{L}(\mathcal{A})$ is infinite. Our goal is to express $\mathcal{P}_{\mathcal{A}}$ by an ordering constraint with existential quantifiers $\Phi_{\mathcal{A}}(z)$ and a single global variable z. Given this property it follows for arbitrary finite automata $\mathcal{A}$ and $\mathcal{A}'$ that:

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

We illustrate the definition of $\Phi_{\mathcal{A}}(z)$ for the following automaton $\mathcal{A}$ with alphabet $\{1,2\}$ and states x,y:

$\mathcal{A}$:



We assume that the alphabet $\{1,2\}$ is included in the feature set and define $\Phi_{\mathcal{A}}(z)$ by:

$$\Phi_{\mathcal{A}}(z) \quad =_{\text{def}} \quad \exists x \exists y (z{\geq}x \wedge x[1]{\geq}y \wedge y[2]{\geq}y \wedge y{\geq}x \wedge a(y))$$

Here, we use the following abbreviation: for arbitrary u,v,f we write $u[f]{\geq}v$ instead of $\exists w(u[f]w \wedge v{\leq}w)$. The states x,y of $\mathcal{A}$ become existentially bound variables of $\Phi_{\mathcal{A}}(z)$. The constraint $z{\geq}x$ expresses that x is the initial state of $\mathcal{A}$ and the constraint $a(y)$ that y is the finial state of $\mathcal{A}$. Furthermore, for each transition of the form $u \xrightarrow{f} v$ in $\mathcal{A}$ there is a corresponding formula $u[f]{\geq}v$. An epsilon transition $u \xrightarrow{\varepsilon} v$ of $\mathcal{A}$ simply corresponds to an ordering constraint $u{\geq}v$. Proving the correctness of this encoding is not difficult.
For finite trees, the above reduction allows to encode those finite automata that recognize a finite language. Universality for automata with finite languages is coNP complete and thus entailment of with existential quantifiers coNP hard.

3 Related Structures of Feature Trees

Beside of optionally labeled and sufficiently labeled feature trees, two alternative notions of feature trees can be found in the literature. We recall these notions and compare them with respect to their first-order theories, independence, and their relationship to second order monadic logic.

3.1 Alternative Definitions of Feature Trees

Optionally labeled feature trees as considered in this article and previously [Müller *et al.*, 2000] *allow* labels at all nodes but do *not require* any. In [Aït-Kaci *et al.*, 1994], *completely labeled* feature trees are considered, i.e. feature trees with a total labeling function. Finally, [Backofen, 1995] considers *leaf-labeled* feature trees, i.e feature trees where exactly the leaves are labeled.

Sets of Words as Trees. As argued in the introduction, we can relate a language of ordering constraints over some structure of features trees to (W)S2S if the features trees provided by the structure allow to represent arbitrary sets of words (over features) in a unique way. Whether this is possible or not depends on the notion of feature trees used. Given a label $a \in \mathcal{L}$, a feature tree can be understood as a representation for the set of its a-labeled nodes, a subset of $\mathcal{F}^*$.
A completely labeled tree can represent a prefixed closed set of words, whereas a leaf-labeled tree can represent the frontier of a prefixed closed set. Optionally labeled trees represent arbitrary sets of words but in general each set corresponds to multiple trees. Only sufficiently labeled tree can uniquely represent arbitrary non-empty sets of words.

First-Order Theories. 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 has a complete axiomatization [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 the complete axiomatization.
With respect to ordering constraints the choice of the specific definition of feature trees matters. For example, let us consider the formulas $\Phi_1(x)$ and Φ_2 with labels $a \neq b$:

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

The formula $\Phi_1(x)$ says that the root of the denotation of x is unlabeled since $a \neq b$. Thus, $\Phi_1(x)$ is satisfiable in $\mathrm{FT}_{\leq}$ and $\mathrm{FT}_{\leq}^{fin}$ 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 information ordering. Such a tree exists in $\mathrm{FT}_{\leq}^{fin}$ namely $(\{\varepsilon\}, \emptyset)$. However, there is no least tree in structures of feature trees where all nodes (or all leaves) are required to be labeled. Thus, Φ_2 distinguishes the first-order theory $\mathrm{FT}_{\leq}$ from those of the alternative structures in [Aït-Kaci *et al.*, 1994, Backofen, 1995, Backofen & Smolka, 1995].

Independence. The failure of independence for ordering constraints with existential quantifiers does not depend on the structure of feature trees chosen (optionally labeled, completely labeled, leaf-labeled, or sufficiently labeled). The counter example against independence given in the introduction, however, does not apply for the structures of feature trees that are completely labeled or leaf-labeled. The judgment $x{\leq}y \wedge a(y) \models a(x) \vee \exists z(x{\leq}z \wedge b(z))$ holds in all these structures but this does not contradict independence. In fact, for both completely labeled or leaf-labeled trees the first disjunct is entailed, but not for optionally labeled or sufficiently labeled trees:

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

This judgment holds for completely labeled trees since x must be labeled and cannot bear any other label than a. It also holds for feature trees with labeled leaves where $x{\leq}y \wedge a(y)$ implies that y and thus x denotes a leaf and hence must be labeled.
The following counter example for independence applies to any of the structures discussed so far. We write $a(x[f])$ as an abbreviation for the existential formula $\exists x'(x[f]x' \wedge a(x'))$. For any of the structures considered, the following entailment judgment is valid if $a \neq b$ but neither of the two disjuncts is entailed:

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

3.2 Sufficiently Labeled Feature Trees

So called sufficiently labeled feature trees play a crucial rôle in the relation between feature logics and second-order monadic logic.

Definition 3.1 *We call a feature tree* τ sufficiently labeled *if for every* $\pi \in D_\tau$ *there exists a path* π' *and a label* $a \in \mathcal{L}$ *such* $L_\tau(\pi\pi') = a$.

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 labeled node.

Lemma 3.2 *Let $\emptyset \neq L \subseteq \mathcal{F}^* \times \mathcal{L}$ be a partial function. Then there exists a unique sufficiently labeled feature tree with labeling function L and this tree is the least tree with labeling function L.*

Proof. Since $\{\pi \mid L(\pi)$ is defined $\} \neq \emptyset$ we can define a tree domain D as follows:

$$D = \{\pi \mid \text{ exists a suffix } \pi' \text{ of } \pi \text{ such that } L_\tau(\pi') \text{ is defined}\}$$

The feature tree τ with $D_\tau = D$ and $L_\tau = L$ is sufficiently labeled and smaller than all other trees with labeling function L. □

Let the predicate $\mathsf{suff\text{-}lab}(\tau)$ hold if τ is sufficiently labeled. For expressing this we first express the compatibility predicate $\tau{\sim}\tau'$ which holds for two trees τ and τ' if for all path π and labels a, b, $L_\tau(\pi) = a$ and $L_{\tau'}(\pi) = b$ imply $a = b$. We can express the compatibility predicate for $\mathrm{FT}_\leq$ and $\mathrm{FT}_\leq^{\mathit{fin}}$ by the following existential formula $x{\sim}y$ over ordering constraints:

$$x{\sim}y \ =_{\mathrm{def}} \ \exists z (x{\leq}z \wedge y{\leq}z)$$

Proposition 3.3 *If $\mathcal{L}$ contains at least two labels then the predicate $\mathsf{suff\text{-}lab}$ can be expressed by a first-order formula over ordering constraints in $FT_\leq$ or $FT_\leq^{fin}$.*

Proof. We express the predicate $\mathsf{suff\text{-}lab}$ by the following formula $\mathsf{suff\text{-}lab}(x)$:

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

This formula requires for $\alpha \models_{\mathrm{FT}_\leq} \mathsf{suff\text{-}lab}(x)$ that $\alpha(x)$ is smaller than all trees which are compatible with the same trees than $\alpha(x)$, and that $\alpha(x)$ is not the least tree $(\{\varepsilon\}, \emptyset)$. The correctness of the formula $\mathsf{suff\text{-}lab}(x)$ is not obvious. First note that for all $\alpha : \mathcal{V} \rightarrow \mathrm{FT}_\leq$ it holds that:

$$\alpha \models_{\mathrm{FT}_\leq} \forall z\ (x{\sim}z \leftrightarrow y{\sim}z) \rightarrow x{\leq}y \quad \text{iff} \quad L_{\alpha(x)} = L_{\alpha(y)}$$

For the one direction, let $\alpha \models_{\mathrm{FT}_\leq} \forall z\ (x{\sim}z \leftrightarrow y{\sim}z) \rightarrow x{\leq}y$ and suppose that $L_{\alpha(x)}(\pi)$ is defined for some path π. Let $b \neq L_{\alpha(x)}(\pi)$ be a label (which exists since $\mathcal{L}$ contains at least two elements) and let τ_π^b be the smallest tree with $L_\tau(\pi) = b$. Hence, $\alpha(x) \not\sim \tau_\pi^b$ and thus $\alpha(y) \not\sim \tau_\pi^b$. Hence $L_{\alpha(y)}$ is defined and $L_{\alpha(y)}(\pi) \neq b$ for all $b \neq L_{\alpha(x)}(\pi)$. Hence, $L_{\alpha(y)}(\pi) = L_{\alpha(x)}(\pi)$. The other direction is straightforward.
Suppose that $\alpha \models_{\mathrm{FT}_\leq} \mathsf{suff\text{-}lab}(x)$ is valid. We have seen so far that $\alpha(x)$ is the least tree with labeling function $L_{\alpha(x)}$. Note next that $\{\pi \mid L_{\alpha(x)}(\pi)$ is defined $\} \neq \emptyset$. Otherwise, $\alpha(x)$ would be the least tree $(\{\varepsilon\}, \emptyset)$ which contradicts our assumption that $\alpha \models \neg\forall z\ (x{\leq}z)$. According to Lemma 3.2 there exists a unique sufficiently labeled feature tree τ with labeling function $L_{\alpha(x)}$ and this tree is the least tree with labeling function $L_{\alpha(x)}$. Thus, τ is equal to $\alpha(x)$ which in turn has to be sufficiently labeled. □

3.3 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 3.4 *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 $\exists\forall$ formula Φ_2 discussed in Section 3.1. Another example is the following existential formula $\Phi_3(x_1,x_2)$:

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

The formula $\Phi_3(x_1,x_2)$ requires for all solutions α that there exists τ such that $\tau{\leq}\alpha(x_1)$ and $\tau{\leq}\alpha(x_2)$. Formula $\Phi_3(x_1,x_2)$ is valid over $\mathrm{FT}_{\leq}$ but not valid over $\mathrm{FT}^{-}_{\leq}$. In $\mathrm{FT}_{\leq}$ one may choose $\tau = (\{\varepsilon\},\emptyset)$ for all α. This particular choice is impossible in $\mathrm{FT}^{-}_{\leq}$ since $(\{\varepsilon\},\emptyset)$ is not sufficiently labeled. Even worse, if $\alpha(x_1) = (\{\varepsilon\},\{(\varepsilon,a_1)\})$ and $\alpha(x_2) = (\{\varepsilon\},\{(\varepsilon,a_2)\})$ for $a_1 \neq a_2$ then there exists no appropriate tree τ in $\mathrm{FT}^{-}_{\leq}$ at all.

Proposition 3.5 *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 $\mathrm{FT}_{\leq}$ (Proposition 3.3). 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 $\exists x(\mathsf{suff\text{-}lab}(x) \wedge \Phi')$. □

4 Second-order Monadic Logic

Let $k = |\mathcal{F}|$ be the cardinality of the set of features $\mathcal{F}$ i.e. $2 \leq k \leq \omega$. We recall the definitions of *second-order monadic logic with k successors* SkS [Rabin, 1969] and of *weak second-order monadic logic with k successors* WSkS [Thatcher & Wright, 1968].

Syntactically, SkS and WSkS 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 SkS and WSkS are built from variables x and p and features f.

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

The semantics of SkS 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 WSkS coincides with the semantics of SkS except that in WSkS a variable x denotes a *finite* set of paths. As derived forms we will use the following formulas with their usual semantics:

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

The second-order monadic logic with two successors, S2S and WS2S, are obtained for $k = 2$. It is well known that (W)SkS can be expressed in (W)S2S for all $2 \leq k \leq \omega$ [Thatcher & Wright, 1968, Rabin, 1969].

Theorem 4.1 [Rabin, 1969, Thatcher & Wright, 1968, Doner, 1970] *The satisfiability problems of WS2S and S2S are decidable.*

4.1 Relation to Feature Logics

Theorem 4.2 *The first-order theories of $FT^{-}_{\leq}$ and $FT^{-fin}_{\leq}$ can be embedded in linear time into S2S and WS2S respectively, and vice versa.*

In other words, second-order monadic logic and the first-order theory of the information ordering for sufficiently labeled feature trees are interreducible.
An embedding of the theory of $FT^{-}_{\leq}$ (*resp.*, $FT^{-fin}_{\leq}$) into SkS (*resp.*, WSkS) is shown in Section 4.3 below. This yields reductions into S2S (*resp.*, WS2S). An inverse embedding of (W)S2S into $FT^{-}_{\leq}$ was found in collaboration with Ralf Treinen. It did not appear in the conference version of this article [Müller & Niehren, 1998] and is given in Section 4.2.

Corollary 4.3 *Second order monadic logic S2S and WS2S can be expressed in the first order theory $FT_{\leq}$ and $FT^{fin}_{\leq}$ resp.*

Proof. Theorem 4.2 shows that (W)S2S can be expressed in the first-order theory of $FT^{-}_{\leq}$ (reps. $FT^{-fin}_{\leq}$) which in turn can be expressed in the first-order theory of $FT_{\leq}$ (resp. $FT^{fin}_{\leq}$) according to Proposition 3.5. □

Corollary 4.4 *The first-order theories of $FT^{-}_{\leq}$ and $FT^{-fin}_{\leq}$ are decidable.*

Proof. This is an immediate consequence from Theorems 4.2 and 4.1. □

Despite of Corollary 4.4, the converse of Corollary 4.3 does not hold. Otherwise, the first-order theories of $FT_{\leq}$ and $FT^{fin}_{\leq}$ would be decidable, in contradiction to a result obtained in a follow-up paper of this article [Müller *et al.*, 1998]. This failure illustrates a surprising difference between $FT_{\leq}$ and $FT^{-}_{\leq}$ which shows that the restriction to sufficiently labeled feature trees has an important consequence for the expressiveness of ordering constraints.
Corollary 4.4 incorporates a new strategy for deciding a fragment of the first-order theories of $FT_{\leq}$ and $FT^{fin}_{\leq}$. It is sufficient to encode the fragment of $FT_{\leq}$ into the corresponding fragment of $FT^{-}_{\leq}$. Of course, this method fails for the full first-order theory of $FT_{\leq}$ because of its undecidability. Nevertheless, this method can be use for solving difficult problems such as entailment of $FT_{\leq}$ with existential quantifiers.

Proposition 4.5 *If the number of features in $\mathcal{F}$ is countably infinite then the entailment problem with existential quantifiers for $FT_{\leq}$ (resp. $FT^{fin}_{\leq}$) can be reduced in linear time to the entailment problem for $FT^{-}_{\leq}$(resp. $FT^{-fin}_{\leq}$).*

Proof. Proposition 4.5 is a corollary to Proposition 6.8 to be presented. The proof of the latter proposition is quite involved. It requires some preparations collected in Section 5. Note that the result depends on the existence of an infinite number of features. □

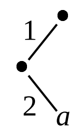
Theorem 4.6 *The entailment problem with existential quantifiers $\varphi \models \exists x_1 \ldots \exists x_n\ \varphi'$ for $FT_{\leq}$ (resp. $FT^{fin}_{\leq}$) is decidable.*

Proof. This is an immediate corollary of Proposition 4.5 and Corollary 4.4. □

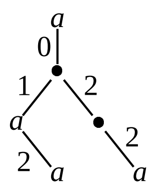
4.2 Encoding (W)S2S in $\mathrm{FT}^{-}_{\leq}$

We show how to encode second-order monadic logic into the first-order theory 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 $\mathrm{FT}^{-fin}_{\leq}$ and for S2S into the first-order theory of $\mathrm{FT}^{-}_{\leq}$.

We first explain how we encode words and sets of words as sufficiently labeled feature trees. Let $a \in \mathcal{L}$ be a label and $0, 1, 2 \in \mathcal{F}$ features. A word $\pi \in \{1,2\}^*$ is encoded by the sufficiently labeled tree $[\![\pi]\!]$ with $L_{[\![\pi]\!]} = \{(\pi, a)\}$. For instance:

$[\![\{12\}]\!] \quad =$ 

We encode a set $\Pi \subseteq \{1,2\}^*$ of words by the sufficiently labeled tree $[\![\Pi]\!]$ which satisfies: $L_{[\![\Pi]\!]} = \{(\varepsilon, a)\} \cup \{0\pi \mid \pi \in \Pi\} \times \{a\}$. In particular, the empty set $\emptyset$ is represented by the tree $(\{\varepsilon\}, \{(\varepsilon, a)\})$ which is sufficiently labeled. For example:

$[\![\{1, 12, 22\}]\!] \quad =$ 

For all words π and sets $\Pi \in \{1,2\}^*$, membership $\pi \in \Pi$ holds if $[\![\pi]\!]$ carries less information than the subtree of the translation of Π at feature 0, i.e. if $[\![\pi]\!] \leq [\![\Pi]\!][0]$.

The key to the embedding of (W)S2S is to express two predicates in $\mathrm{FT}^{-}_{\leq}$ and $\mathrm{FT}^{-fin}_{\leq}$ which require that a tree encodes a word or a (finite) set of words, respectively.

Given features $f_1 \ldots, f_n$, let $a\{f_1, \ldots, f_n\}$ be the unary predicate which holds for a feature tree τ if and only if the root of τ is labeled by a and edges labeled by exactly the features in $\{f_1, \ldots, f_n\}$: That is, $a\{f_1, \ldots, f_n\}(\tau)$ holds if and only if $a(\tau)$ and

$\{f_1,\ldots,f_n\} = \mathcal{D}_\tau \cap \mathcal{F}$. We express the predicate $a\{f_1,\ldots,f_n\}$ in $FT^-_\leq$ by the following first-order formula with free variable x:

$$a\{f_1,\ldots,f_n\}(x) \quad =_{\text{def}} \quad \exists x_1 \ldots \exists x_n \; (\quad a(x) \wedge \bigwedge_{i=1}^{n} x[f_i]x_i \wedge \\ \forall y((a(y) \wedge \bigwedge_{i=1}^{n} y[f_i]x_i) \rightarrow x \leq y))$$

Based on the formula $a\{f_1,\ldots,f_n\}(x)$, we can express the predicate $\mathsf{bin\text{-}tree}_a$ which holds if τ is a binary tree over features $\{1,2\}$ whose nodes are unlabeled or labeled with a, *i.e.*, if $D_\tau \subseteq \{1,2\}^*$ and $\emptyset \neq L_\tau \subseteq D_\tau \times \{a\}$. We express $\mathsf{bin\text{-}tree}_a(\tau)$ in $FT^-_\leq$ and $FT^{-fin}_\leq$ by the following formula $\mathsf{bin\text{-}tree}_a(x)$:

$$\mathsf{bin\text{-}tree}_a(x) \;=_{\text{def}}\; \exists y(x \leq y \wedge a\{1,2\}(y) \wedge \bigwedge_{f\in\{1,2\}} \exists z(y \leq z \wedge z[f]y))$$

Note that if $\mathsf{bin\text{-}tree}_a(\tau)$ holds for a sufficiently labeled tree τ then $\{\pi \mid L_\tau(\pi) = a\} \neq \emptyset$ since every sufficiently labeled feature tree has at least one label. The predicate word holds for a tree τ which represents a word of feature in that there exists $\pi \in \mathcal{F}^*$ such that $[\![\pi]\!] = \tau$. Hence, $\mathsf{word}(\tau)$ holds if τ is a minimal binary a-tree.

$$\mathsf{word}(x) \;=_{\text{def}}\; \mathsf{bin\text{-}tree}_a(x) \wedge \neg\exists y(\mathsf{bin\text{-}tree}_a(y) \wedge y \leq x \wedge \neg x \leq y)$$

It is interesting to see why this encoding works: If α solves $\mathsf{word}(x)$ over $FT^-_\leq$ then $\alpha(x)$ must have exactly one leaf that is labeled by a: At least one anyway since $\alpha(x)$ is sufficiently labeled, and not more than one because otherwise some leaves could be dropped to find a smaller sufficiently labeled solution. Since α solves $\mathsf{bin\text{-}tree}_a(x)$, the unique leaf of $\alpha(x)$ must be a-labeled but none of it inner nodes may be since otherwise inner labels may be dropped.
We express possibly empty sets of words by the predicate set such that $\mathsf{set}(\tau)$ holds for τ iff $\tau = [\![\Pi]\!]$ for some set $\Pi \subseteq \{1,2\}^*$. We express this predicate as follows:

$$\begin{aligned} \mathsf{empty\text{-}set}(x) &=_{\text{def}} a\{\,\}(x) \\ \mathsf{non\text{-}empty\text{-}set}(x) &=_{\text{def}} a\{0\}(x) \wedge \exists y(x[0]y \wedge \mathsf{bin\text{-}tree}_a(y)) \\ \mathsf{set}(x) &=_{\text{def}} \mathsf{empty\text{-}set}(x) \vee \mathsf{non\text{-}empty\text{-}set}(x) \end{aligned}$$

In Figure 1, the reduction of (W)S2S into the first-order theory of $FT^-_\leq$ (*resp.*, $FT^{-fin}_\leq$) is given. The encoding applies to closed formulas only. It treats both path variables p and set variables x as tree variables. The translation of quantifiers requires $\mathsf{word}(p)$ for all path variables p and $\mathsf{set}(x)$ for all set variables x. The translation of membership $[\![\, p \in x \,]\!]$ requires for its solutions α that $[\![\, \alpha(p) \,]\!]$ is smaller than the subtree of $\alpha(x)$ at path 0 as explained above. The translation of terms $[\![\, p = w \,]\!]$ uses an auxiliary translation $[\![\, w \,]\!]_p$ which is defined along the recursive definition of w.

Proposition 4.7 *A closed S2S (resp., WS2S) formula ψ is valid if and only if its translation $[\![\psi]\!]$ 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. More precisely, we can translate an assignment α from paths variables to words

$$
\begin{array}{lcllcl}
[\![\varepsilon]\!]_x & = & a\{\,\}(x) & [\![p{=}w]\!] & = & [\![w]\!]_p \\
[\![1w]\!]_x & = & \exists y(x[1]y \wedge [\![w]\!]_y) & & & \\
[\![2w]\!]_x & = & \exists y(x[2]y \wedge [\![w]\!]_y) & [\![p \in x]\!] & = & \exists y(x[0]y \wedge p{\leq}y) \\
[\![p]\!]_x & = & p{\leq}x \wedge x{\leq}p & & & \\
[\![\exists p\psi]\!] & = & \exists p(\mathsf{word}(p) \wedge [\![\psi]\!]) & [\![\psi \wedge \psi']\!] & = & [\![\psi]\!] \wedge [\![\psi']\!] \\
[\![\exists x\psi]\!] & = & \exists x(\mathsf{set}(x) \wedge [\![\psi]\!]) & [\![\neg\psi]\!] & = & \neg[\![\psi]\!]
\end{array}
$$

Figure 1: Encoding (W)S2S in the first order theory of $FT^-_{\leq}$ (*resp.*, $FT^{-fin}_{\leq}$)

in $\mathcal{F}^*$ and set variables to sets in $P(\mathcal{F}^*)$ to an assignment $[\![\alpha]\!] : \mathcal{V} \to FT^-_{\leq}$ of tree variables to sufficiently labeled trees such that $[\![\alpha]\!](p) = [\![\alpha(p)]\!]$ and $[\![\alpha]\!](x) = [\![\alpha(x)]\!]$ for all x, p. Let ψ be an formula of (W)S2S with path variables $P(\psi)$ and set variables $S(\psi)$. We can show for all α that α is a solution of ψ if and only if $[\![\alpha]\!]$ is a solution of $\psi \wedge \bigwedge_{p\in P(\psi)} \mathsf{word}(p) \wedge \bigwedge_{x\in S(\psi)} \mathsf{set}(x)$. $\square$

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

We give the reduction of $FT^-_{\leq}$ or $FT^{-fin}_{\leq}$ into (W)S2S by detour through the second-order monadic logic with $k = |\mathcal{F}|$ successors (W)SkS.

Trees as Sets of Words. Let $1 \leq n < \infty$ be the number of labels in $\mathcal{L}$. Every sufficiently labeled feature tree can be identified with a unique n-tuple of pairwise disjoint sets of paths with non-empty union, and vice versa. For every label $a \in \mathcal{L}$ 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 4.8 *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, every sufficiently labeled tree τ is finite 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 is 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 $\Pi_1, \ldots, \Pi_n$ be pairwise disjoint sets of words over features that have a nonempty union. We define $\gamma^{-1}(\Pi_1, \ldots, \Pi_n)$ as follows:

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

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

Figure 2: Encoding the first-order theory of $FT_{\leq}$ or $FT_{\leq}^{fin}$ into (W)SkS where $k = |\mathcal{F}|$.

Since $\bigcup_{i=1}^{n} \Pi_i$ is assumed to be non-empty, we have $\varepsilon \in D_{\gamma^{-1}(\Pi_1,\ldots,\Pi_n)}$ which is prefix-closed by construction. The relation $L_{\gamma^{-1}(\Pi_1,\ldots,\Pi_n)}$ is a partial function since all Π_i are assumed to be pairwise disjoint. Hence $\gamma^{-1}(\Pi_1,\ldots,\Pi_n)$ 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 labeled τ and that $\gamma(\gamma^{-1}(\Pi_1,\ldots,\Pi_n)) = (\Pi_1,\ldots,\Pi_n)$ for all $\Pi_1,\ldots,\Pi_n$ that are pairwise disjoint and have a non-empty union. □

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.

Reduction to (W)SkS. We next define a mapping from first-order formulas over ordering constraints (interpreted over $FT_{\leq}^{-}$ or $FT_{\leq}^{-fin}$) to formulas of second-order monadic logic with k successors. We will make use of the following abbreviations:

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

For every variable x and label a let x_a be a fresh variable. Suppose that $\mathcal{L} = \{a_1,\ldots,a_n\}$. In Figure 2, the definition of the mapping $[\![\_]\!]$ is given.

Proposition 4.9 *A formula Φ whose bound variables are renamed apart is valid over $FT_{\leq}^{-}$ (resp.. $FT_{\leq}^{-fin}$) if and only if its translation $[\![\Phi]\!]$ is valid over WSkS (resp. SkS).*

Proof. If α is a solution of Φ then α' with $\alpha'(x_{a_i}) = \gamma_{a_i}(\alpha(x))$ for all i, $1 \leq i \leq n$, is a solution of $[\![\Phi]\!]$. If β is a solution of $[\![\Phi]\!]$ then the mapping β' with $\beta'(x) = \gamma^{-1}(\beta(x_{a_1}),\ldots,\beta(x_{a_n}))$ is a solution of Φ. The existence of the inverse mapping γ^{-1} of γ is proved by Proposition 4.8. □

5 Satisfiability and Entailment of Simple Path Constraints

We now prepare the reduction of the entailment problem with existential quantifiers for $FT_{\leq}$ or $FT_{\leq}^{fin}$ to the corresponding problem for $FT_{\leq}^{-}$ or $FT_{\leq}^{-fin}$, respectively. For this purpose, we recall results on satisfiability and least solutions from [Müller *et al.*, 2000] and then formulate a corollary about entailment of simple path constraints.

5.1 Simple Path Constraints

We will use a collection of predicates based on the subtree relation at a fixed path, as well as appropriate formulas to express these predicates that we call path constraints. We distinguish simple path constraints that *require* the existence of a path and conditional path constraints which impose a restriction under the condition that a path exists. In this section, we restrict ourselves to simple path constraints. Conditional path constraints will be introduced in Section 7.1 where entailment with existential quantifiers is considered.

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 $L_{\tau[\pi]} = \{(\pi', a) \mid L_\tau(\pi\pi') = a\}$. Let $\pi \in \mathcal{F}^*$ be a path. The *subtree predicate* $\tau[\pi]\tau'$ holds for two trees τ and τ' iff $\pi \in \mathcal{D}_\tau$ and $\tau[\pi] = \tau'$. We express the subtree predicate by the following formula $x[\pi]y$ which generalizes $x[f]y$ from a single feature f to an arbitrary path π.

$$
\begin{aligned}
x[\varepsilon]y &= x{\le}y \land y{\le}x \\
x[\pi\pi']y &= \exists z (x[\pi]z \land z[\pi']y)
\end{aligned}
$$

We need three further predicates: $\tau[\pi]{\downarrow}$ holds iff $\pi \in D_\tau$, $a(\tau[\pi])$ is valid iff $L_\tau(\pi) = a$, and $\tau{\le}\tau'[\pi]$ holds if $\pi \in \mathcal{D}_{\tau'}$ and τ is smaller than $\tau'[\pi]$. We define *simple path constraints* as formulas to express these predicates by means of existential quantification:

$$
\begin{aligned}
x[\pi]{\downarrow} &= \exists y (x[\pi]y) \\
a(x[\pi]) &= \exists y (x[\pi]y \land a(y)) \\
x{\le}y[\pi] &= \exists z (x{\le}z \land y[\pi]z)
\end{aligned}
$$

Lemma 5.1 *For all variables x,y,z and path π it holds that:*

1. *The constraint $x{\le}z[\pi] \land y{\le}z[\pi]$ entails $x{\sim}y$ for $FT_{\le}$ and $FT_{\le}^{fin}$.*
2. *If $\pi \neq \varepsilon$ then $x{\le}x[\pi]$ is unsatisfiable over $FT_{\le}^{fin}$ but satisfiable over $FT_{\le}$.*

Proof. The first property follows directly the definitions. The second property justifies the well-known occurs check which holds for finite trees but not for infinite ones. □

We recall the notion of *syntactic support* from [Müller *et al.*, 2000] which verifies entailment judgments by purely syntactical means. We consider three forms of judgments $\varphi \vdash y{\le}x[\pi]$, $\varphi \vdash x[\pi]{\downarrow}$, and $\varphi \vdash a(x[\pi])$.

$$
\begin{aligned}
\varphi \vdash y{\le}x[\varepsilon] \quad &\text{if} \quad y{\le}x \textit{ in } \varphi \\
\varphi \vdash y{\le}x[f] \quad &\text{if} \quad x[f]y \textit{ in } \varphi \\
\varphi \vdash y{\le}x[\pi_1\pi_2] \quad &\text{if} \quad \text{exists } z : \varphi \vdash y{\le}z[\pi_2] \text{ and } \varphi \vdash z{\le}x[\pi_1] \\
\\
\varphi \vdash x[\pi]{\downarrow} \quad &\text{if} \quad \text{exists } z : \varphi \vdash z{\le}x[\pi] \\
\varphi \vdash a(x[\pi]) \quad &\text{if} \quad \text{exists } z : \varphi \vdash z{\le}x[\pi] \text{ and } a(z) \textit{ in } \varphi
\end{aligned}
$$

Lemma 5.2 (Correctness) *For all φ, x, y, π, and a it is valid over $FT_{\le}$ and $FT_{\le}^{fin}$ that:*

1. *if $\varphi \vdash x{\le}y[\pi]$ then $\varphi \models x{\le}y[\pi]$ holds.*
2. *if $\varphi \vdash x[\pi]{\downarrow}$ then $\varphi \models x[\pi]{\downarrow}$ holds.*
3. *if $\varphi \vdash a(x[\pi])$ then $\varphi \models a(x[\pi])$ holds.*

Proof. Straightforward induction on the rules defining syntactic support. □

5.2 Satisfiability and Least Solutions

In [Müller *et al.*, 2000], an algorithm is given that tests a constraint φ for satisfiability over $\mathrm{FT}_{\leq}$ and $\mathrm{FT}_{\leq}^{fin}$, respectively. It also computes the least solution of a satisfiable constraint. We recall these results since they are essential for this article.
We call a constraint φ *closed* (under reflexivity F1.1, transitivity F1.2 and decomposition F2) 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

We define the *closure* of a constraint φ to be the smallest closed constraint which contains φ. Note that the closure of a constraint is independent of the structure chosen, $\mathrm{FT}_{\leq}$ or $\mathrm{FT}_{\leq}^{fin}$. For every constraint there exists a unique closure. The closure of a constraint φ is a conjunction of φ with some basic constraints $x{\leq}y$ where $x,y \in \mathcal{V}(\varphi)$.

Theorem 5.3 (Satisfiability and Least Solutions) *There exists a cubic time algorithm which computes the closure of a constraint and decides its satisfiability both over $FT_{\leq}$ and $FT_{\leq}^{fin}$. A satisfiable constraint* φ *has a least solution* least_φ *which, if* φ *is closed, satisfies for all* $x \in \mathcal{V}(\varphi)$*:*

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

Proof. We only sketch the proof given in [Müller *et al.*, 2000]. The central idea is to consider an extended constraint language which provides atomic compatibility constraints of the form $x{\sim}y$. We call a constraint of the extended language *clash-free for $FT_{\leq}$* if it satisfies F3-F5 and *clash-free for $FT_{\leq}^{fin}$* if it satisfies F3-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	not	$a(x) \wedge x{\sim}y \wedge b(y)$ *in* φ and $a \neq b$	
F6	not	$\varphi \vdash x{\leq}x[\pi]$ and $\pi \neq \varepsilon$	

For every constraint φ one can compute its saturation with respect to F1-F4 in cubic time and then check whether it is clash-free by inspection of F5-F6. If not then φ is unsatisfiable and otherwise satisfiable. The latter can be shown by proving that least_φ solves φ if φ is closed and clash-free (see Proposition 4 and Lemma 5 of [Müller *et al.*, 2000]).
Finally, suppose that φ is closed and satisfiable (it does not matter whether φ contains compatibility constraints or not). Hence the saturation φ' of φ with respect to F1-F4 must be clash-free. Hence, $\mathsf{least}_{\varphi'}$ is the least solution of φ'. Since φ is a closed constraint, it coincides with φ' up to additional compatibility constraints. Hence least_φ is equal to $\mathsf{least}_{\varphi'}$ and solves φ' and thus φ. □

5.3 Entailment of Simple Path Constraints

The syntactic description of least solutions given in Theorem 5.3 implies a criterion for entailment of simple path constraints.

Corollary 5.4 (Simple Path Constraints) *Let* φ *be satisfiable and 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. Syntactic support implies entailment due to Lemma 5.2. The converse follows from Theorem 5.3 on least solutions: Let φ be closed and satisfiable and least_φ its least solution. If $\varphi \models x[\pi]{\downarrow}$ then $\mathsf{least}_\varphi(x)[\pi]{\downarrow}$ holds, *i.e.*, $\varphi \vdash x[\pi]{\downarrow}$. If $\varphi \models a(x[\pi])$ then $a(\mathsf{least}_\varphi(x)[\pi])$ holds, *i.e.*, $\varphi \vdash a(x[\pi])$. □

Note that Corollary 5.4 does not cover entailment of all kinds of simple path constraints. For instance, it does not determine when $\varphi \models x{\leq}\pi[y]$ holds. For $\pi = \varepsilon$ the latter kind of entailment can be decided due to a result in [Müller *et al.*, 2000]. We recall this result for sake of completeness but do not use it in the sequel.

Theorem 5.5 [Müller *et al.*, 2000] *The entailment problem* $\varphi \models \varphi'$ *can be tested in cubic time both for* $FT_\leq$ *and* $FT_\leq^{fin}$*. For both structures it holds that if* φ *is satisfiable and closed, and* $x, y \in \mathcal{V}(\varphi)$ *then:*

$$\varphi \models x{\leq}y \ \ \textit{iff} \ \ x{\leq}y \text{ in } \varphi.$$

Proof. This result is non-trivial since it is no longer sufficient to consider least solutions. For the proof we refer to [Müller *et al.*, 2000]. □

6 Deciding Entailment with Existential Quantifiers

We now reduce entailment with existential quantifiers for $\mathrm{FT}_\leq$ (resp. $\mathrm{FT}_\leq^{fin}$) to the corresponding problem for $\mathrm{FT}_\leq^-$ (resp. $\mathrm{FT}_\leq^{-fin}$) under the assumption that the set of feature $\mathcal{F}$ is countably infinite. We thereby prove Proposition 4.5 as a corollary to Proposition 6.8 below. Recall that Proposition 4.5 covers the main step for proving the reduction of the entailment problem with existential quantifiers to second-order monadic logic.

> **Caveat:** We need distinct notations for entailment with respect to $\mathrm{FT}_\leq$ (*resp.*, $\mathrm{FT}_\leq^{fin}$) and $\mathrm{FT}_\leq^-$ (*resp.*, $\mathrm{FT}_\leq^{-fin}$). From now on, we write $\Phi \models_{\mathrm{FT}_\leq} \Phi'$ and $\Phi \models_{\mathrm{FT}_\leq^-} \Phi'$ and always ignore potential finiteness annotations.

6.1 Model Change for Quantifier-free Entailment

We show that changing the model for $FT_\leq$ to $FT^-_\leq$ does not affect satisfiability or quantifier-free entailment. Even though this model-invariance does not hold for entailment with existential quantifiers, these considerations will shed some light on the more general case.

Definition 6.1 *Let $b \in \mathcal{L}$ and $g \in \mathcal{F}$. For all trees τ with $g \notin \mathcal{F}(\tau)$, we define a sufficiently labeled feature tree $\mathsf{ex}^b_g(\tau)$ by adding sufficiently many labels as follows:*

$$\begin{aligned} D_{\mathsf{ex}^b_g(\tau)} &= D_\tau \cup \{\pi g \mid \pi \in D_\tau\} \\ L_{\mathsf{ex}^b_g(\tau)} &= L_\tau \cup \{(\pi g, b) \mid \pi \in L_\tau\} \end{aligned}$$

Proposition 6.2 (Adding Labels) *Let $b \in \mathcal{L}$, $g \in \mathcal{F}$, φ a constraint, and $\bar{x}$ a sequence of variables. If $\alpha : \mathcal{V} \to FT_\leq$ is a variable assignment with $g \notin \mathcal{F}(\varphi) \cup \mathcal{F}_{\mathcal{V}(\exists\bar{x}\varphi)}(\alpha)$ then:*

$$\alpha \models_{FT_\leq} \exists\bar{x}\varphi \quad \textit{iff} \quad \mathsf{ex}^b_g \circ \alpha \models_{FT^-_\leq} \exists\bar{x}\varphi$$

Proof. The proof is rather lengthy but simple. It can be found in Section 8.2. □

As a first illustration of the importance of Proposition 6.2, we show how to encode the satisfiability problem of $FT_\leq$ into the corresponding problem of $FT^-_\leq$.

Lemma 6.3 (Model change and satisfiability) *Let φ be closed. If φ is satisfiable over $FT_\leq$ then φ is satisfiable over $FT^-_\leq$, and vice versa.*

Proof. If φ is satisfiable over $FT_\leq$ then it has a least solution (Theorem 5.3). Since φ is closed, its least solution is equal to least_φ as defined in Theorem 5.3. Let $g \notin \mathcal{F}(\varphi)$ be some feature (which exists since $\mathcal{F}$ is infinite) and $b \in \mathcal{L}$ a label. Hence, $g \notin \mathcal{F}_{\mathcal{V}(\varphi)}(\mathsf{least}_\varphi)$ also in case of infinite trees and Proposition 6.2 implies that $\mathsf{ex}^b_g \circ \mathsf{least}_\varphi$ is a solution of φ. □

Our next goal is to lift Lemma 6.3 from satisfiability to quantifier-free entailment. This step is quite simple provided the following Lemma is given.

Lemma 6.4 (Fresh Features) *Let Φ be a first-order formula over ordering constraints, and g a fresh feature $g \notin \mathcal{F}(\Phi)$. Then Φ is valid in $FT_\leq$ if for all $\alpha : \mathcal{V} \to FT_\leq$ with $g \notin \mathcal{F}_{\mathcal{V}(\Phi)}(\alpha)$ it holds that $\alpha \models_{FT_\leq} \Phi$.*

Proof. The proof relies on the fact that the set of feature $\mathcal{F}$ is infinite. It is again not difficult and thus given in Section 8.1. □

Lemma 6.5 (Model change and quantifier-free entailment) *Let φ and φ' be constraints. Then:*

$$\varphi \models_{FT_\leq} \varphi' \quad \textit{iff} \quad \varphi \models_{FT^-_\leq} \varphi'$$

Notice that Lemma 6.5 does not require closedness for φ (nor φ') even though it is more general than Lemma 6.3 (which requires prefix closedness). The reason is that the proof given for Lemma 6.5 uses least solutions, in contrast to the following proof of the more general Lemma 6.3.

Proof. The implication from the right to the left is somewhat tedious: We assume $\varphi \models_{\mathrm{FT}^-_\leq} \varphi'$ and show $\varphi \models_{\mathrm{FT}_\leq} \varphi'$. We fix a label $b \in \mathcal{L}$ and a feature $g \notin \mathcal{F}(\varphi \to \varphi')$ which exists since $\mathcal{F}$ is infinite. Since $g \notin \mathcal{F}(\varphi \to \varphi')$ we can show the validity of $\varphi \to \varphi'$ by applying Lemma 6.4. Let $\alpha : \mathcal{V} \to \mathrm{FT}_\leq$ be a mapping with $g \notin \mathcal{F}_{\mathcal{V}(\varphi\to\varphi')}(\alpha)$. In order to prove $\alpha \models_{\mathrm{FT}_\leq} \varphi \to \varphi'$ we suppose that $\alpha \models_{\mathrm{FT}_\leq} \varphi$. Given the freshness condition for g, Proposition 6.2 implies $\mathsf{ex}^b_g \circ \alpha \models_{\mathrm{FT}^-_\leq} \varphi$. Entailment $\varphi \models_{\mathrm{FT}^-_\leq} \varphi'$ yields $\mathsf{ex}^b_g \circ \alpha \models_{\mathrm{FT}^-_\leq} \varphi'$ such that the converse of Proposition 6.2 implies $\alpha \models_{\mathrm{FT}_\leq} \varphi'$.
The converse is straightforward: Suppose $\varphi \models_{\mathrm{FT}_\leq} \varphi'$ and let $\alpha : \mathcal{V} \to \mathrm{FT}^-_\leq$ be a variable assignment. We assume $\alpha \models_{\mathrm{FT}^-_\leq} \varphi$ and show that $\alpha \models_{\mathrm{FT}^-_\leq} \varphi'$. Our assumption yields $\alpha \models_{\mathrm{FT}_\leq} \varphi$ such that entailment implies $\alpha \models_{\mathrm{FT}_\leq} \varphi'$. Since α is a mapping into $\mathrm{FT}^-_\leq$ it follows that $\alpha \models_{\mathrm{FT}^-_\leq} \varphi'$ as required. $\square$

6.2 Entailment with Existential Quantifiers

Unfortunately, Lemma 6.5 does not generalize to entailment with existential quantification. This means that $\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi'$ does not imply $\varphi \models_{\mathrm{FT}^-_\leq} \exists\bar{x}\varphi'$ in general. The problem can be illustrated by the counter example in (1) and (2):

$$\mathrm{true} \models_{\mathrm{FT}_\leq} \exists x(x \leq x_1 \wedge x \leq x_2) \tag{1}$$

$$\mathrm{true} \not\models_{\mathrm{FT}^-_\leq} \exists x(x \leq x_1 \wedge x \leq x_2) \tag{2}$$

The formula $\exists x(x \leq x_1 \wedge x \leq x_2)$ on the right hand side in (1) and (2) is valid over $\mathrm{FT}_\leq$ but not over $\mathrm{FT}^-_\leq$. It requires for every pair of trees (the values of x_1, x_2) that there exists a third tree (for x) which is smaller than each of them. In the case of $\mathrm{FT}_\leq$, x may denote the tree $(\{\varepsilon\}, \emptyset)$ independently of the choices for x_1 and x_2. In fact, the tree $(\{\varepsilon\}, \emptyset)$ is the only possible choice for x if $a \neq b$, x_1 denotes $(\{\varepsilon\}, \{(\varepsilon, a\})$, and x_2 denotes $(\{\varepsilon\}, \{(\varepsilon, a\})$. Since the tree $(\{\varepsilon\}, \emptyset)$ is not sufficiently labeled, we cannot chose any sufficiently labeled tree for x in $\mathrm{FT}^-_\leq$ given the above values for x_1 and x_2 (which, in fact, are sufficiently labeled).
The first idea for resolving the trouble is to require sufficiently many labels in a syntactic manner, *i.e.*, by additional labeling constraints for all global variables. Of course, labeling constraints cannot be added arbitrarily without affecting the set of solutions in an uncontrolled way. So a refined idea is to require $b(x[g])$ for all global variables x, a fixed fresh feature g, and a fixed label b. In this way the problem in example (2) can be solved since the following entailment judgment holds:

$$b(x_1[g]) \wedge b(x_2[g]) \models_{\mathrm{FT}^-_\leq} \exists x(x \leq x_1 \wedge x \leq x_2) \tag{3}$$

In contrast to (2), the variable assignment α which maps x_1 to $(\{\varepsilon\}, \{(\varepsilon, a\})$ and x_2 to $(\{\varepsilon\}, \{(\varepsilon, a\})$ does no longer solve the left-hand side of (3). The "extended" variable assignment $\mathsf{ex}^b_g \circ \alpha$ solves the right hand side of (3) since the existentially quantified variable x can be mapped to $\mathsf{ex}^b_g((\{\varepsilon\}, \emptyset))$.

Definition 6.6 (Formula Extension) *Let $b \in \mathcal{L}$ and $g \in \mathcal{F}$. For every first-order formula Φ over ordering constraints with $g \notin \mathcal{F}(\Phi)$ we define a first-order formula $\mathsf{ex}_g^b(\Phi)$ by:*

$$\mathsf{ex}_g^b(\Phi) = \Phi \wedge \bigwedge_{y \in \mathcal{V}(\Phi)} b(y[g])$$

The idea behind this formula is that $\mathsf{ex}_g^b(\varphi)$ syntactically enforces sufficiently many labels for the least solution of $\mathsf{ex}_g^b(\varphi)$. Of course, for entailment over $\mathrm{FT}_\leq$ one must also consider variable assignments into feature trees which are not necessarily sufficiently labeled. Therefore, the precise rôle of the formula $\mathsf{ex}_g^b(\varphi)$ for entailment remains unclear at first sight. It will soon be clarified (see Proposition 6.8 below).
The good news is that Definition 6.6 can be used for resolving counter examples such as (1) in a systematic way: In a first step, the left-hand side of (1) has to be replaced by an equivalent constraint that contains at least the variables of its right-hand side. For instance, we may replace true by $x_1 \leq x_1 \wedge x_2 \leq x_2$. In a second step, we apply Definition 6.6 and obtain a constraint equivalent to $b(x_1[g]) \wedge b(x_2[g])$ which is precisely the left-hand side of (3).

Lemma 6.7 (The Trouble) *Let φ be a closed constraint, g a fresh feature $g \notin \mathcal{F}(\varphi)$, $\bar{x}$ a sequence of variables, and b a label. A variable assignment $\alpha : \mathcal{V} \to FT_\leq^-$ with $\alpha \models_{\mathrm{FT}_\leq} \mathsf{ex}_g^b(\exists\bar{x}\varphi)$ satisfies $\alpha \models_{\mathrm{FT}_\leq^-} \exists\bar{x}\varphi$ under the following precondition:*

(PC-Tr) *for all $x \in \mathcal{V}(\exists\bar{x}\varphi)$ and all paths π: if $\varphi \vdash x[\pi]\downarrow$ then $\alpha \models_{\mathrm{FT}_\leq} b(x[\pi g])$*

Proof. This proof is non-trivial. It is given in Section 7. □

Note that we can always find a fresh feature g for any constraint φ since the set of all features $\mathcal{F}$ is infinite whereas $\mathcal{F}(\varphi)$ is finite (*cf.*, Lemma 6.4).

Proposition 6.8 (Model change and entailment with existential quantifiers) *Let $\bar{x}$ be a sequence of variables, φ and φ' closed constraints such that $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$, and b a label. If g is a fresh feature $g \notin \mathcal{F}(\varphi \to \exists\bar{x}\varphi')$ then:*

$$\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi' \quad \textit{iff} \quad \mathsf{ex}_g^b(\varphi) \models_{\mathrm{FT}_\leq^-} \exists\bar{x}\varphi'$$

Proof. Let φ, φ' be closed constraints, $g \notin \mathcal{F}(\varphi \to \exists\bar{x}\varphi')$, and $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$.
The implication from the right to the left remains as tedious as for Lemma 6.5: We assume $\mathsf{ex}_g^b(\varphi) \models_{\mathrm{FT}_\leq^-} \exists\bar{x}\varphi'$. For proving the validity of $\varphi \to \exists\bar{x}\varphi'$ over $\mathrm{FT}_\leq$ we apply Lemma 6.4. Since $g \notin \mathcal{F}(\varphi \to \exists\bar{x}\varphi')$ is assumed, it is sufficient to fix a variable assignment $\alpha : \mathcal{V} \to \mathrm{FT}_\leq$ with $g \notin \mathcal{F}_{\mathcal{V}(\varphi \to \exists\bar{x}\varphi')}(\alpha)$ and $\alpha \models_{\mathrm{FT}_\leq} \varphi$ and to show that $\alpha \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi'$. If $\alpha \models_{\mathrm{FT}_\leq} \varphi$, then Proposition 6.2 together with the above freshness condition for g implies $\mathsf{ex}_g^b \circ \alpha \models_{\mathrm{FT}_\leq^-} \varphi$. From the definition of ex_g^b it follows that $\mathsf{ex}_g^b \circ \alpha \models_{\mathrm{FT}_\leq^-} \bigwedge_{y \in \mathcal{V}(\varphi)} b(y[g])$ and thus $\mathsf{ex}_g^b \circ \alpha \models_{\mathrm{FT}_\leq^-} \mathsf{ex}_g^b(\varphi)$. Entailment $\mathsf{ex}_g^b(\varphi) \models_{\mathrm{FT}_\leq^-} \exists\bar{x}\varphi'$ implies that $\mathsf{ex}_g^b \circ \alpha \models_{\mathrm{FT}_\leq^-} \exists\bar{x}\varphi'$. Thus $\alpha \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi'$ follows from Proposition 6.2.
The converse implication (which was straightforward for Lemma 6.5) now becomes rather difficult. We assume $\varphi \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi'$ and $\alpha \models_{\mathrm{FT}_\leq^-} \mathsf{ex}_g^b(\varphi)$. We show that $\alpha \models_{\mathrm{FT}_\leq^-}$

$\exists\bar{x}\varphi'$. From $\alpha \models_{FT^-_\leq} \mathrm{ex}^b_g(\varphi)$ it follows that $\alpha \models_{FT_\leq} \mathrm{ex}^b_g(\varphi)$ and hence $\alpha \models_{FT_\leq} \mathrm{ex}^b_g(\exists\bar{x}\varphi')$ since $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$. It remains to show that $\alpha \models_{FT^-_\leq} \exists\bar{x}\varphi'$. This can be done by an application of the Trouble Lemma 6.7 (here the closedness of φ' is used). It remains to verify the precondition **(PC-Tr)** of the Trouble Lemma. We let $x \in \mathcal{V}(\exists\bar{x}\varphi')$ and π satisfying $\varphi' \vdash x[\pi]\downarrow$ and show that $\alpha \models_{FT_\leq} b(x[\pi g])$. From $\varphi' \vdash x[\pi]\downarrow$ it follows that $\exists\bar{x}\varphi' \models_{FT_\leq} x[\pi]\downarrow$ such that entailment yields $\varphi \models_{FT_\leq} x[\pi]\downarrow$. The latter and the closedness of φ imply $\varphi \vdash x[\pi]\downarrow$ as shown by Corollary 5.4. By definition of syntactic support there exists a variable $y \in \mathcal{V}(\varphi)$ such that $\varphi \vdash y \leq x[\pi]$ and thus $\varphi \models_{FT_\leq} y \leq x[\pi]$. The definition of ex^b_g yields $\mathrm{ex}^b_g(\varphi) \models_{FT_\leq} b(x[\pi g])$, *i.e.* $\alpha \models_{FT_\leq} b(x[\pi g])$. □

Note that the assumption $\mathcal{V}(\exists\bar{x}\varphi') \subseteq \mathcal{V}(\varphi)$ is essential for Proposition 6.8. Otherwise, the extension $\mathrm{ex}^b_g(\varphi)$ would not enforce sufficiently many labels (see Example (1)). On the other hand side, this assumption does not restrict generality. If it is not satisfied then we can simply add tautologies $y \leq y$ for all variables $y \in \mathcal{V}(\exists\bar{x}\varphi')$ to φ.
Note also the Proposition 6.8 insists on the closedness assumption for both φ and φ'. Both assumptions are necessary. The closedness of φ' is required by the Trouble Lemma 6.7 whose precondition **(PC-Tr)** follows from Corollary 5.4 which requires the closedness of φ.

7 Resolving the Trouble

In the previous section we reduced the entailment problem with existential quantifiers in $FT_\leq$ to the corresponding problem in $FT^-_\leq$ provided that the Trouble Lemma 6.7 holds. In order to prove this lemma, we first introduce conditional path constraints.

7.1 Conditional Path Constraints

The predicate $\tau?[\pi] \leq \tau'$ holds if either $\pi \notin D_\tau$ or $\pi \in \tau$ and $\tau[\pi] \leq \tau'$. We express this predicate by the following formula that we call a *conditional path constraint*:

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

Lemma 7.1 (Path Constraints and Satisfiability) *For all x, y, z, paths π_1, π_2, and labels a, b: If $a \neq b$ then the formula $b(x[\pi_1\pi_2]) \wedge x?[\pi_1] \leq y \wedge y \sim z \wedge a(z[\pi_2])$ is unsatisfiable.*

Figure 3 illustrates the situation of Lemma 7.1.
We extend the definition of *syntactic support* to judgments $\varphi \vdash x?[\pi] \leq y$ with conditional path constraints:

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

Lemma 7.2 (Correctness) *For all x, y, π, φ if $\varphi \vdash x?[\pi] \leq y$ then $\varphi \models x?[\pi] \leq y$.*

Lemma 7.3 (Cancellation of Mountains) *For closed φ and all x, y, π:*

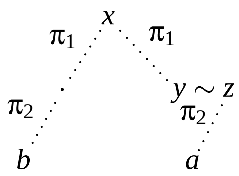


Figure 3: Path constraints and satisfiability: $b(x[\pi_1\pi_2]) \wedge x?[\pi_1] \leq y \wedge y{\sim}z \wedge a(z[\pi_2])$ is unsatisfiable if $a \neq b$. The vertical dimension (top to bottom) corresponds to feature selection, the horizontal dimension (left to right) to the ordering $\leq$.

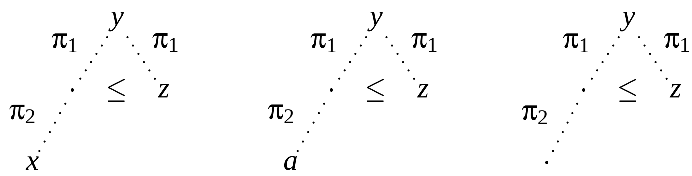


Figure 4: Cancellation of Mountains

1. *If $\varphi \vdash x{\leq}y[\pi_1\pi_2]$ and $\varphi \vdash y?[\pi_1] \leq z$ then $\varphi \vdash x{\leq}z[\pi_2]$.*
2. *If $\varphi \vdash a(y[\pi_1\pi_2])$ and $\varphi \vdash y?[\pi_1] \leq z$ then $\varphi \vdash a(z[\pi_2])$.*
3. *If $\varphi \vdash y[\pi_1\pi_2]{\downarrow}$ and $\varphi \vdash y?[\pi_1] \leq z$ then $\varphi \vdash z[\pi_2]{\downarrow}$.*

Proof. The situation is depicted in Figure 4. The latter two cases follow from the first one. For the first case, we assume $\varphi \vdash x{\leq}y[\pi_1\pi_2]$ and $\varphi \vdash y?[\pi_1] \leq z$. Then there exists a variable $z' \in \mathcal{V}(\varphi)$ such that $\varphi \vdash z'{\leq}y[\pi_1]$ and $\varphi \vdash x{\leq}z'[\pi_2]$. We can show by induction on π_1 that $z{\leq}z'$ *in* φ. Hence $\varphi \vdash x{\leq}z'[\pi_2]$. □

Lemma 7.4 (Mountain chains) *Let y,z,z' be variables, π_0,π_1,π_2 paths and $a \in \mathcal{L}$ a label. If $\alpha \models_{\mathrm{FT}_\leq} \exists z'(z?[\pi_0] \leq z' \wedge z'{\leq}y[\pi_1])$ and $L_{\alpha(z)}(\pi_0\pi_2) = a$ then $L_{\alpha(y)}(\pi_1\pi_2) = a$.*

Proof. Straightforward. The situation of the Lemma is illustrated in Figure 5. □

7.2 Proving the Trouble Lemma

Lemma 6.7 (The Trouble) *Let φ be a closed constraint, $g \notin \mathcal{F}(\varphi)$ a feature, $\bar{x}$ a sequence of variables, and b a label. A variable assignment $\alpha : \mathcal{V} \to FT_\leq^-$ with $\alpha \models_{\mathrm{FT}_\leq} \mathrm{ex}_g^b(\exists\bar{x}\varphi)$ satisfies $\alpha \models_{\mathrm{FT}_\leq^-} \exists\bar{x}\varphi$ under the following precondition:*

(PC-Tr) *for all $x \in \mathcal{V}(\exists\bar{x}\varphi)$ and all paths π: if $\varphi \vdash x[\pi]{\downarrow}$ then $\alpha \models_{\mathrm{FT}_\leq} b(x[\pi g])$*

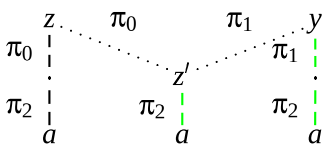
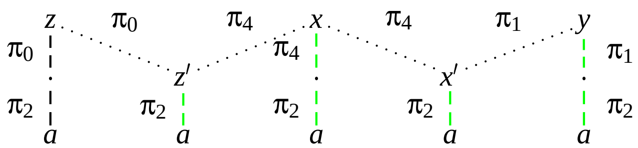


Figure 5: Mountain chain

Proof. Let φ be a closed constraint and α a solution of φ that satisfies $\alpha \models_{FT_\leq} \mathsf{ex}_g^b(\exists\bar{x}\varphi)$. It is sufficient to construct a variable assignment $\beta : \mathcal{V} \to FT_\leq^-$ which solves φ and coincides with α on the variables in $\mathcal{V}(\varphi) \cup \{\bar{x}\}$. We call a variable y *global* if $y \in \mathcal{V}(\exists\bar{x}\varphi)$ and *local* if $y \in \{\bar{x}\}$.
For every global variable y we define a sufficiently labeled feature tree $\beta(y)$ by $\beta(y) = \alpha(y)$. For all local variable $y \in \{\bar{x}\}$ we define a sufficiently labeled feature tree $\beta(y)$ by specifying its labeling function which in turn determines its tree domain (Lemma 3.2):

$$\begin{array}{llll}
\mathsf{L1} & L_{\beta(y)}(\pi) = a & \text{if} & \varphi \vdash a(y[\pi]) \\
\mathsf{L2} & L_{\beta(y)}(\pi g) = b & \text{if} & \varphi \vdash y[\pi]\!\downarrow \\
\mathsf{L3} & L_{\beta(y)}(\pi_1\pi_2) = a & \text{if} & \left\{ \begin{array}{l} \text{exists } z \in \mathcal{V}(\exists\bar{x}\varphi) \text{ and } z' \in \mathcal{V}(\varphi) \text{ such that} \\ \varphi \vdash z?[\pi_0] \leq z',\ \varphi \vdash z' \leq y[\pi_1], \text{ and } L_{\alpha(z)}(\pi_0\pi_2) = a \end{array} \right.
\end{array}$$

Compare condition $\mathsf{L3}$ with the mountain chain situation depicted in Figure 5. The value of the local variable y depends on the value of the global variable z in such a way that $L_{\beta(y)}(\pi_1\pi_2) = a$ has to be required according to the Mountain Chain Lemma 7.4. The reader might wonder, why mountain chains with only two mountains (see Figure 5) are sufficient to characterize the relationships between global and local variables. Why, for instance is the following more general situation covered?



The reason is that φ is assumed to be closed such that inner mountains can be canceled out. In the above example for instance, Lemma 7.3 implies $z' \leq x'$ *in* φ such that the more specific situation in Figure 5 applies as well.
We have to verify that $\beta(y)$ *is well-defined for all local variables* y. First note that $D_{\beta(y)} \neq \emptyset$: Because of $\mathsf{L2}$ the value $L_{\beta(y)}(g) = b$ is defined such that the tree domain $D_{\beta(y)} = \{\pi \mid \text{ exists suffix } \pi' \text{ of } \pi \text{ such that} L_{\beta(y)}(\pi') \text{ is defined}\}$ is non-empty. We next show that $L_{\beta(y)}$ is a partial function, *i.e.*, that $L_{\beta(y)}(\pi)$ is uniquely defined for all path π where $L_{\beta(y)}(\pi)$ is defined. Since $g \notin \mathcal{F}(\varphi)$, case $\mathsf{L2}$ cannot overlap with either $\mathsf{L1}$ nor $\mathsf{L3}$. Suppose that y a local variable, π a path and a, b labels such that $L_{\beta(y)}(\pi) = a$ by $\mathsf{L1}$ and $L_{\beta(y)}(\pi) = b$ by $\mathsf{L3}$. According to $\mathsf{L3}$, it holds that $\pi = \pi_1\pi_2$ for some $\pi_1\pi_2$ and there are global variable z and $z' \in \mathcal{V}(\varphi)$ such that:

$$\varphi \vdash z' \leq y[\pi_1],\ \varphi \vdash z?[\pi_0] \leq z',\ L_{\alpha(z)}(\pi_0\pi_2) = b$$

From L1 it follows that $\varphi \vdash a(y[\pi_1\pi_2])$. Thus, there is a variable $y' \in \mathcal{V}(\varphi)$ such that $\varphi \vdash y' \leq y[\pi_1]$ and $\varphi \vdash a(y'[\pi_2])$. From the correctness of syntactic support (Lemma 5.2) and Lemma 5.1 we obtain:

$$\varphi \models_{\mathrm{FT}_\leq} z' \leq y[\pi_1] \wedge y' \leq y[\pi_1] \models_{\mathrm{FT}_\leq} z' \sim y'$$

Since $\alpha \models_{\mathrm{FT}_\leq} \exists \bar{x}\varphi$ and z is global the following constraint is satisfiable over $\mathrm{FT}_\leq$:

$$b(z[\pi_0\pi_2]) \wedge z?[\pi_0] \leq z' \wedge z' \sim y' \wedge a(y'[\pi_2])$$

This is precisely the situation of Lemma 7.1 which proves that $a = b$. Hence, $L_{\beta(y)}$ is indeed a partial function and $\beta(y)$ a sufficiently labeled feature tree.
It remains to show that β *solves* φ*, i.e., that* β *satisfies all basic constraints in* φ.

1. Case $x[f]y$ *in* φ. We have to show that $L_{\beta(x)}(f\pi) = a$ if and only if $L_{\beta(y)}(\pi) = a$.
   We first show that $L_{\beta(x)}(f\pi) = a$ *implies* $L_{\beta(y)}(\pi) = a$.
   (a) Case x, y are both local.
      L1 Suppose that $L_{\beta(x)}(f\pi) = a$ because of $\varphi \vdash a(x[f\pi'])$. Since φ is closed it follows from Mountain Removal Lemma 7.3 (part 2) that $\varphi \vdash a(y[\pi])$ and hence $\beta(y)(\pi) = a$.
      L2 Let $L_{\beta(x)}(f\pi) = a$ because of $f\pi = \pi' g$, $a = b$ 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, the Mountain Removal Lemma 7.3 (part 3) and $x[f]y$ *in* φ yield $\varphi \vdash y[\pi'']\downarrow$. Hence, $L_{\beta(y)}(\pi'' g) = b$, *i.e.*, $L_{\beta(y)}(\pi) = b$.
      L3 Assume that $L_{\beta(x)}(f\pi) = a$ since there exist a global variable z, a variable $z' \in \mathcal{V}(\varphi)$, and paths π_0, π_1, π_2 such that $f\pi = \pi_1\pi_2$, $\varphi \vdash z?[\pi_1] \leq z'$, $\varphi \vdash z' \leq x[\pi_1]$, and $L_{\alpha(z)}(\pi_0\pi_2) = a$.
         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 $\varphi \vdash a(z[\pi_0 f\pi])$. Since φ is closed under reflexivity (F1.1) it follows that $\varphi \vdash y?[\varepsilon] \leq y$. Thus and since y is global, L3 yields $L_{\alpha(y)}(\varepsilon\pi) = a$.
         B. Otherwise, $\pi_1 = f\pi_1'$ and $\pi = \pi_1'\pi_2$ for some π'. Since φ is closed and $x[f]y$ *in* φ we can apply the Mountain Removal Lemma 7.3 (part 1) and obtain $\varphi \vdash z'?[\pi_1'] \leq y$. Hence, L3 yields $L_{\beta(y)}(\pi_1'\pi_2) = a$, *i.e.*, $L_{\beta(y)}(\pi) = a$.
   (b) Case x local and y global:
      L1 Let $L_{\beta(x)}(f\pi) = a$ since $\varphi \vdash a(x[f\pi])$. Since φ is closed and $x[f]y$ *in* φ the Mountain Removal Lemma yields $\varphi \vdash a(y[\pi])$. Since y is global the correctness of syntactic support (Lemma 5.2) implies $\alpha \models_{\mathrm{FT}_\leq} \exists \bar{x}\varphi \models_{\mathrm{FT}_\leq} a(y[\pi])$. Thus $L_{\beta(y)}(\pi) = L_{\alpha(y)}(\pi) = a$.
      L2 **A rather interesting case:** Let $L_{\beta(x)}(f\pi) = a$ 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 $f\pi'' = \pi'$. Since φ is closed and $x[f]y$ *in* φ the Mountain Removal Lemma 7.3 yields $\varphi \vdash y[\pi'']\downarrow$. Thus, the precondition **(PC-Tr)** implies $\alpha \models_{\mathrm{FT}_\leq} b(y[\pi' g])$. Hence, $L_{\alpha(y)}(\pi'' g) = b$ which is equivalent to $L_{\beta(y)}(\pi) = a$.

L3 Let $L_{\beta(x)}(f\pi) = a$ since there exist a global variable z, a variable $z' \in \mathcal{V}(\varphi)$, and paths 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 $L_{\alpha(z)}(\pi_0\pi_2)a$.

A. Case $\pi_1 = \varepsilon$. Hence $f\pi = \pi_2$ and $z' \leq x$ *in* φ since φ is closed. In this case $\varphi \vdash z?[\pi_0 f] \leq y$ such that the globality of z, y implies:

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

Hence, $L_{\alpha(z)}(\pi_0 f\pi) = a$. The Mountain Chain Lemma 7.4 yields $L_{\beta(y)}(\pi) = L_{\alpha(y)}(\pi) = a$.

B. Otherwise there exists a path π'_1 such that $\pi_1 = f\pi'_1$ and $\pi = \pi'_1\pi_2$. Since φ is closed, $x[f]y$ *in* φ, and $\varphi \vdash z' \leq x[f\pi'_1]$, the Mountain Removal Lemma 7.3 implies $\varphi \vdash z' \leq y[\pi'_1]$. Since y, z are global, we deduce:

$$\alpha \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi \models_{\mathrm{FT}_\leq} \exists z' (z' \leq z[\pi_0] \wedge y?[\pi'_1] \leq z')$$

Our assumption $L_{\alpha(z)}(\pi_0\pi_2) = a$ and the Mountain Chain Lemma 7.4 imply $L_{\alpha(y)}(\pi'_1\pi_2) = a$, *i.e.*, $L_{\beta(y)}(\pi) = a$.

(c) Case x global and y local: If $L_{\beta(x)}(f\pi) = a$ then $L_{\alpha(x)}(f\pi) = a$. Rule L3 implies $L_{\beta(y)}(\pi) = a$.

(d) Case x, y are both global: In this case $\alpha \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi \models_{\mathrm{FT}_\leq} x[f]y$ and if $L_{\alpha(x)}(f\pi) = a$ then $L_{\alpha(y)}(\pi) = a$.

For the converse implication of the case $x[f]y$ in φ *we show that* $L_{\beta(y)}(\pi) = a$ *implies* $L_{\beta(x)}(f\pi) = a$.

(a) Case x local and y global: If $L_{\beta(y)}(\pi) = a$ then $L_{\alpha(y)}(\pi) = a$. By applying L3 with $z = z' = y$, $\pi_0 = \varepsilon$, $\pi_1 = f$, and $\pi_2 = \pi$, we obtain $L_{\beta(x)}(f\pi) = a$.

(b) Case x, y are both local:

L1 Assume $L_{\beta(y)}(\pi) = a$ because of $\varphi \vdash a(y[\pi'])$. Hence, $\varphi \vdash a(x[f\pi])$ such that $L_{\beta(x)}(f\pi) = a$.

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

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

(c) Case x global and y local. Let $L_{\beta(y)}(\pi) = a$.

L1 If $\varphi \vdash a(y[\pi])$ then $\varphi \vdash a(x[f\pi])$. Since x is global, we obtain $\alpha \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi \models_{\mathrm{FT}_\leq} a(x[\pi])$ such that $L_{\alpha(x)}(f\pi) = a$, *i.e.*, $L_{\beta(x)}(f\pi) = a$.

L2 Assume $L_{\beta(y)}(\pi) = a$ because of $a = b$ and $\pi = \pi' g$ for some π' with $\varphi \vdash y[\pi']\downarrow$, Hence $\varphi \vdash x[f\pi']\downarrow$ such that the precondition **(PC-Tr)** yields $\alpha \models_{\mathrm{FT}_\leq} b(x[f\pi g])$. *i.e.*, $L_{\beta(x)}(f\pi) = a$.

L3 Let $(\pi, a) \in L_{\beta(y)}$ since there exists a global variable z, a variable $z' \in \mathcal{V}(\varphi)$, and path π_0, π_1, π_2 such that $\varphi \vdash z?[\pi_0] \le z'$, $\varphi \vdash z' {\le} y[\pi_1]$, $(\pi_0\pi_2, a) \in L_{\alpha(z)}$, and $\pi = \pi_1\pi_2$. Since $x[f]y$ *in* φ we also have $\varphi \vdash z' {\le} x[f\pi_1]$. The globality of x,z implies:

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

Hence, the Mountain Chain Lemma 7.4 and $L_{\alpha(z)}(\pi_0\pi_2) = a$ imply $L_{\alpha(x)}(f\pi_1\pi_2) = a$, *i.e.*, $L_{\beta(x)}(f\pi) = a$.

(d) Case x,y are both global: In this case $\alpha \models_{\mathrm{FT}_{\le}} \exists\bar{x}\varphi \models_{\mathrm{FT}_{\le}} x[f]y$ such that $L_{\alpha(y)}(\pi) = a$ implies $L_{\beta(x)}(f\pi) = L_{\alpha(x)}(f\pi) = a$.

2. Case $x {\le} y$ *in* φ. We have to show that $L_{\beta(x)}(\pi) = a$ implies $L_{\beta(y)}(\pi) = a$.

   (a) Case x is local and y global:

   L1 Let $L_{\beta(x)}(\pi) = a$ since $\varphi \vdash a(x[\pi])$. Hence, $\varphi \vdash a(y[\pi])$ such that correctness of syntactic support yields $\varphi \models_{\mathrm{FT}_{\le}} a(y[\pi])$. The globality of y yields $\alpha \models_{\mathrm{FT}_{\le}} \exists\bar{x}\varphi \models_{\mathrm{FT}_{\le}} a(y[\pi])$. Thus, $L_{\beta(y)}(\pi) = L_{\alpha(y)}(\pi) = a$.

   L2 Let $L_{\beta(x)}(\pi g) = b$ since $\varphi \vdash x[\pi]{\downarrow}$. As in the previous case, it follows that $\alpha \models_{\mathrm{FT}_{\le}} y[\pi]{\downarrow}$. Since y is global, the precondition **(PC-Tr)** yields $\alpha \models b(y[\pi g])$. Hence, $L_{\beta(y)}(\pi g) = L_{\alpha(y)}(\pi g) = b$.

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

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

   The Mountain Chain Lemma 7.4 together with $L_{\alpha(z)}(\pi_0\pi_2) = a$ implies $L_{\alpha(y)}(\pi_1\pi_2) = a$, *i.e.*, $L_{\beta(y)}(\pi) = a$.

   (b) Case x,y are local:

   L1 If $L_{\beta(x)}(\pi) = a$ because of $\varphi \vdash a(x[\pi])$ then $\varphi \vdash a(y[\pi])$ such that $L_{\beta(y)}(\pi) = a$.

   L2 If $L_{\beta(x)}(\pi) = a$ since $\pi = \pi' g$, $a = b$, and $\varphi \vdash x[\pi']{\downarrow}$ then $\varphi \vdash y[\pi']{\downarrow}$ such that $L_{\beta(y)}(\pi' g) = L_{\beta(y)}(\pi) = a$.

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

   (c) Case x global and y local: If $L_{\beta(x)}(\pi) = a$ then $L_{\alpha(x)}(\pi) = a$. Since $\varphi \vdash x {\le} y[\varepsilon]$, $\varphi \vdash x?[\varepsilon] \le x$ and x global, case L3 implies $L_{\beta(y)}(\varepsilon\pi) = a$.

   (d) If x,y be global then $\alpha \models_{\mathrm{FT}_{\le}} \exists\bar{x}\varphi \models_{\mathrm{FT}_{\le}} x {\le} y$ such that $L_{\alpha(x)} \subseteq L_{\alpha(y)}$ and hence $L_{\beta(x)} \subseteq L_{\beta(y)}$.

3. Case $a(x)$ *in* φ.

   (a) If x is local then $\varphi \vdash a(x[\varepsilon])$ such that $L_{\beta(x)}(\varepsilon) = a$ according to rule L1.

   (b) If x is global then $\alpha \models_{\mathrm{FT}_{\le}} \exists\bar{x}\varphi \models_{\mathrm{FT}_{\le}} a(x)$. Hence $L_{\beta(x)}(\varepsilon) = L_{\alpha(x)}(\varepsilon) = a$.

□

8 More Details of the Proofs

We give the proofs of Lemma 6.4 and Proposition 6.2 that were omitted in the core of this article. Both proofs are rather straightforward even though they are quite long. We add them here for sake of completeness.

8.1 Fresh Features

We first prove Lemma 6.4 which shows that we may restrict ourselves to variable assignments that do not contain a fixed fresh feature, since the set of all features $\mathcal{F}$ is infinite.

Lemma 8.1 *Let $g \in \mathcal{F}$ and a finite set $F \subseteq \mathcal{F}$ such that $g \notin F$. Under these assumptions 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 8.2 *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. Under these assumption it holds that $\alpha \models_{\mathrm{FT}_\leq} \Phi$ if and only if $\theta \circ \alpha \models_{\mathrm{FT}_\leq} \Phi$.*

Proof. It is obvious that $\alpha \models_{\mathrm{FT}_\leq} \Phi$ if and only if $\theta \circ \alpha \models_{\mathrm{FT}_\leq} \theta(\Phi)$. Since θ leaves the features in Φ invariant we have $\theta(\Phi) = \Phi$. □

Lemma 6.4 (Fresh Features) *Let Φ be a first-order formula over ordering constraints, and g a fresh feature $g \notin \mathcal{F}(\Phi)$. A formula Φ is valid in $FT_\leq$ if for all $\alpha : \mathcal{V} \to FT_\leq$ with $g \notin \mathcal{F}_{\mathcal{V}(\Phi)}(\alpha)$ it holds that $\alpha \models_{\mathrm{FT}_\leq} \Phi$.*

Proof. We have to show $\beta \models_{\mathrm{FT}_\leq} \Phi$ for an arbitrary variable assignment $\beta : \mathcal{V} \to \mathrm{FT}_\leq$. We fix $\beta : \mathcal{V} \to \mathrm{FT}_\leq$ and a function $\theta : \mathcal{F} \to \mathcal{F} \setminus \{g\}$ that is one-to-one and invariant on $\mathcal{F}(\Phi)$ (θ exists according to Lemma 8.1). The variable assignment $\theta \circ \beta$ satisfies $g \notin \mathcal{F}_{\mathcal{V}(\Phi)}(\theta \circ \beta)$ such that the assumption of the lemma yields $\theta \circ \alpha \models_{\mathrm{FT}_\leq} \Phi$. We can now apply Lemma 8.2 in order to obtain $\alpha \models_{\mathrm{FT}_\leq} \Phi$. □

8.2 Adding Labels

Adding labels is an important procedures for our proofs on entailment. Recall that we defined a mapping ex_g^b such that a sufficiently labeled feature tree $\mathsf{ex}_g^b(\tau)$ is obtained by adding a leaves $(\pi g, b)$ to every node π of τ. It remains to show that ex_g^b satisfies Proposition 6.2.

Proposition 6.2 (Adding Labels) *Let g be an arbitrary label, φ be a constraint, $\bar{x}$ a sequence of variables, and $\alpha : \mathcal{V} \to FT_\leq$ a variable assignment satisfying $g \notin \mathcal{F}(\varphi) \cup \mathcal{F}_{\mathcal{V}(\exists\bar{x}\varphi)}(\alpha)$. Then it holds that:*

$$\alpha \models_{\mathrm{FT}_\leq} \exists\bar{x}\varphi \quad \textit{iff} \quad \mathsf{ex}_g^b \circ \alpha \models_{\mathrm{FT}_\leq^-} \exists\bar{x}\varphi$$

Lemma 8.3 *If τ is a feature tree, g a fresh feature $g \notin \mathcal{F}(\tau)$, and b a label then $\mathsf{ex}_g^b(\tau)$ is sufficiently labeled feature tree.*

Proof. The assumption $g \notin \mathcal{F}(\tau)$ implies that $L_{\mathsf{ex}_g^b(\tau)}$ is a partial function such that $\mathsf{ex}_g^b(\tau)$ is indeed a feature tree. All leaves of $\mathsf{ex}_g^b(\tau)$ (the maximal paths of its domain) are of the form πg for some π and thus labeled by b. Hence $\mathsf{ex}_g^b(\tau)$ is sufficiently labeled. □

Lemma 8.4 *Assume $g \notin \mathcal{F}_{\mathcal{V}(\varphi)}(\alpha)$ and $g \notin \mathcal{F}(\varphi)$. If α is a solution of φ in $FT_\leq$ then $\mathsf{ex}_g^b \circ \alpha$ is a solution of φ in $FT_\leq^-$.*

Proof. We have to show that every basic constraint in φ is satisfied by $\mathsf{ex}_g^b \circ \alpha$.

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 $f\pi \in D_{\mathsf{ex}_g^b(\alpha(x))}$ is equivalent to $\pi \in D_{\mathsf{ex}_g^b \circ \alpha(y)}$. This is proved by the following sequence of equivalences:

   $$f\pi \in D_{\mathsf{ex}_g^b(\alpha(x))} \quad \text{iff} \quad f\pi \in D_{\alpha(x)} \cup \{\pi' g \mid \pi' \in D_{\alpha(x)}\}$$

   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

   $$\begin{array}{rll} f\pi \in D_{\mathsf{ex}_g^b(\alpha(x))} & \text{iff} & f\pi \in D_{\alpha(x)} \cup \{f\pi'' g \mid f\pi'' \in D_{\alpha(x)}\} \\ & \text{iff} & \pi \in D_{\alpha(y)} \cup \{\pi'' g \mid \pi'' \in D_{\alpha(y)}\} \\ & \text{iff} & \pi \in D_{\mathsf{ex}_g^b(\alpha(y))} \end{array}$$

   The reasoning for the labeling function is similar.

2. Case $x{\leq}y$ *in* φ. We have to verify the domain inclusion $D_{\mathsf{ex}_g^b(\alpha(x))} \subseteq D_{\mathsf{ex}_g^b(\alpha(y))}$.

   $$\begin{array}{rcl} D_{\mathsf{ex}_g^b(\alpha(x))} & = & D_{\alpha(x)} \cup \{\pi' g \mid \pi' \in D_{\alpha(x)}\} \\ & \subseteq & D_{\alpha(y)} \cup \{\pi' g \mid \pi' \in D_{\alpha(y)}\} \\ & = & D_{\mathsf{ex}_g^b(\alpha(y))} \end{array}$$

   The reasoning for the labeling function is again similar.

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

□

In order to prove the converse of Lemma 8.4 it is useful to consider the deletion of labels, *i.e.*, a left-inverse function $(\mathsf{ex}_g^b)^{-1}$ to ex_g^b. For arbitrary τ, we define a feature tree $(\mathsf{ex}_g^b)^{-1}(\tau)$ as follows:

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

The left-inverse removes all paths with feature g together with its labels if it exists.

Lemma 8.5 *For all* g, τ, b *if* $g \notin \mathcal{F}(\tau)$ *then* $(\mathsf{ex}_g^b)^{-1}(\mathsf{ex}_g^b(\tau)) = \tau$.

Proof. Since $g \notin \mathcal{F}(\tau)$, the tree $\mathsf{ex}_g^b(\tau)$ is well-defined and hence $(\mathsf{ex}_g^b)^{-1}(\mathsf{ex}_g^b(\tau))$ is also well-defined. Furthermore, its tree domain satisfies:

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

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

Lemma 8.6 *For all* φ*,* b*,* $g \notin \mathcal{F}(\varphi)$ *and* $\alpha : \mathcal{V} \to FT_{\leq}^-$*: if* $\alpha \models_{\mathrm{FT}_{\leq}^-} \varphi$ *then* $(\mathsf{ex}_g^b)^{-1} \circ \alpha \models_{\mathrm{FT}_{\leq}} \varphi$.

Proof. We have to show that every basic constraint in φ is satisfied by $(\mathsf{ex}_g^b)^{-1} \circ \alpha$.

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 $f\pi \in D_{(\mathsf{ex}_g^b)^{-1}(\alpha(x))}$ is equivalent to $\pi \in D_{(\mathsf{ex}_g^b)^{-1}(\alpha(y))}$. This is proved by the following equivalences:

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

   The reasoning for the labeling function is similar.

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

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

□

Proof of Proposition 6.2. Let $\alpha : \mathcal{V} \to \mathrm{FT}_{\leq}$ be a solution of $\exists\bar{x}\varphi$, $\alpha \models_{\mathrm{FT}_{\leq}} \exists\bar{x}\varphi$. There exists a sequence of trees $\bar{\tau}$ such that $\alpha[\bar{\tau}/\bar{x}] \models_{\mathrm{FT}_{\leq}} \varphi$ where $\alpha[\bar{\tau}/\bar{x}]$ denotes the valuation that maps $\bar{x}$ pointwise to $\bar{\tau}$ and coincides with α everywhere else. Since $g \notin \mathcal{F}(\varphi)$: $(\mathsf{ex}_g^b)^{-1} \circ \alpha[\bar{\tau}/\bar{x}] \models_{\mathrm{FT}_{\leq}^-} \varphi$ by Lemma 8.6. The latter variable assignment coincides with $\alpha[(\mathsf{ex}_g^b)^{-1}(\bar{\tau})/\bar{x}]$ since we have assumed $g \notin \mathcal{F}_{\mathcal{V}(\varphi)}(\alpha(y))$. Thus $\mathsf{ex}_g^b \circ (\alpha[(\mathsf{ex}_g^b)^{-1}(\bar{\tau})/\bar{x}]) \models_{\mathrm{FT}_{\leq}^-} \varphi$ by Lemma 8.4 and this implies $\mathsf{ex}_g^b \circ \alpha \models_{\mathrm{FT}_{\leq}^-} \exists\bar{x}\varphi$.
For the converse, we assume that $\mathsf{ex}_g^b \circ \alpha \models_{\mathrm{FT}_{\leq}^-} \exists\bar{x}\varphi$. There exists a sequence of sufficiently labeled feature trees $\bar{\tau}$ such that $(\mathsf{ex}_g^b \circ \alpha)[\bar{\tau}/\bar{x}] \models_{\mathrm{FT}_{\leq}^-} \varphi$. It follows from Lemma 8.6 that $(\mathsf{ex}_g^b)^{-1} \circ ((\mathsf{ex}_g^b \circ \alpha)[\bar{\tau}/\bar{x}]) \models_{\mathrm{FT}_{\leq}} \varphi$. Also, $(\mathsf{ex}_g^b)^{-1} \circ \mathsf{ex}_g^b \circ \alpha = \alpha$ due to Lemma 8.5 and $g \notin \mathcal{F}_{\mathcal{V}(\varphi)}(\alpha)$. Thus, the following equation hold and prove $\alpha \models_{\mathrm{FT}_{\leq}} \exists\bar{x}\varphi$:

$$(\mathsf{ex}_g^b)^{-1} \circ ((\mathsf{ex}_g^b \circ \alpha)[\bar{\tau}/\bar{x}]) = ((\mathsf{ex}_g^b)^{-1} \circ \mathsf{ex}_g^b \circ \alpha)[(\mathsf{ex}_g^b)^{-1}(\bar{\tau})/\bar{x}] = \alpha[(\mathsf{ex}_g^b)^{-1}(\bar{\tau})/\bar{x}]$$

9 Conclusion and Future Work

We have investigated the decidability of fragments of the first-order theory of ordering constraints over feature trees ($\mathrm{FT}_{\leq}$ and $\mathrm{FT}_{\leq}^{fin}$). The approach chosen was to relate $\mathrm{FT}_{\leq}$ and $\mathrm{FT}_{\leq}^{fin}$ to the second-order monadic logic (W)S2S. We obtained a new method for proving the decidability of a fragments of the first-order of $\mathrm{FT}_{\leq}$and $\mathrm{FT}_{\leq}^{fin}$ which makes essential use of Rabin's famous tree theorem. We have proved that the entailment problem for $\mathrm{FT}_{\leq}$ with existential quantifiers $\mathrm{Ent}_{\exists}(\mathrm{FT}_{\leq})$ is decidable for both the case of finite tree and for infinite trees.
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 which, in analogy, also holds for finite trees.

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

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 the first-order theory of equality constraints over feature trees FO(FT) and S2S. Since FO(FT) is decidable, it might still be equivalent to S2S even though $\mathrm{FO}(\mathrm{FT}_{\leq})$ is not.

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

Another open question is to find larger decidable fragments of the first-order theory of $\mathrm{FT}_{\leq}$ for which entailment is decidable. This question also includes the decidability question of entailment with existential quantifiers for CFT which can be expressed in the first-order theory of $\mathrm{FT}_{\leq}$.

Acknowledgments. We thank Ralf Treinen for allowing us to present the encoding of S2S into $FT_{\leq}^{-}$ in this paper. Furthermore, we thank Ralf Treinen and the anonymous referees for invaluable remarks on this paper. We thank Jean-Marc Talbot, Sophie Tison, and Marc Tommasi for inspiring discussions. 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 & Nasr, 1986] Hassan Aït-Kaci & Roger Nasr. LOGIN: A logic programming language with built-in inheritance. *The Journal of Logic Programming*, 3(3):185–215, 1986.

[Aït-Kaci & Podelski, 1993] Hassan Aït-Kaci & Andreas Podelski. Towards a meaning of life. *The Journal of Logic Programming*, 16(3 – 4):195–234, July, August 1993.

[Aït-Kaci *et al.*, 1994] Hassan Aït-Kaci, Andreas Podelski, & Gert Smolka. A feature-based constraint system for logic programming with entailment. *Theoretical Computer Science*, 122(1–2):263–283, January 1994.

[Aït-Kaci, 1986] Hassan Aït-Kaci. An algebraic semantics approach to the effective resolution of type equations. *Theoretical Computer Science*, 45:293–351, 1986.

[Backofen & Smolka, 1995] Rolf Backofen & Gert Smolka. A complete and recursive feature theory. *Theoretical Computer Science*, 146(1–2):243–268, July 1995.

[Backofen, 1994] Rolf Backofen. *Expressivity and Decidability of First-order Languages over Feature Trees*. Doctoral Dissertation. Universität des Saarlandes, Technische Fakultät, D–66041 Saarbrücken, 1994.

[Backofen, 1995] Rolf Backofen. A complete axiomatization of a theory with feature and arity constraints. *The Journal of Logic Programming*, 24(1 – 2):37–71, 1995. Special Issue on Computational Linguistics and Logic Programming.

[Brachman & Levesque, 1984] Ronald J. Brachman & Hector J. Levesque. The tractability of subsumption in frame-based description languages. In *Proceedings of the National Conference on Artificial Intelligence*, pages 34–37, August 1984.

[Carpenter, 1992] Bob Carpenter. *The Logic of Typed Feature Structures*, volume 32 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1992.

[Colmerauer, 1984] Alain Colmerauer. Equations and inequations on finite and infinite trees. In ICOT, editor, *Proceedings of the 2nd International Conference on Fifth Generation Computer Systems*, pages 85–99. Omsha Ltd., Tokyo and North-Holland, Amsterdam, 1984.

[Comon *et al.*, 1998] Hubert Comon, Max Dauchet, Remi Gilleron, Denis Lugiez, Sophie Tison, & Marc Tommasi. Tree automata: Techniques and applications, 1998. Book Draft, `http://www.grappa.univ-lille3.fr/tata`.

[Comon, 1995] Hubert Comon. Sequentiality, second-order monadic logic, and tree automata. In *Proceedings of the 10th IEEE Symposium on Logic in Computer Science*, pages 508–517, San Diego, California, June 1995. IEEE Computer Society Press.

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

[Dörre & Rounds, 1992] Jochen Dörre & William C. Rounds. On subsumption and semi-unification in feature algebras. *Journal of Symbolic Computation*, 13:441–461, 1992.

[Dörre, 1991] Jochen Dörre. Feature logics with weak subsumption constraints. In *Annual Meeting of the Association of Computational Linguistics*, pages 256–263, 1991.

[Dörre, 1993] Jochen Dörre. *Feature-Logik und Semiunfikation*. PhD thesis, Philosophische Fakultät der Universität Stuttgart, July 1993. In German.

[Gecseg & Steinby, 1984] F. Gecseg & M. Steinby. *Tree Automata*. Akad. Kiado, Budapest, 1984.

[Kaplan & Bresnan, 1982] Ronald M. Kaplan & Joan Bresnan. Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA, 1982.

[Kay, 1979] Martin Kay. Functional grammar. In C. Chiarello et al., editor, *Proceedings of the 5th Annual Meeting of the Berkeley Linguistics Society*, pages 142–158, 1979.

[Müller *et al.*, 2000] Martin Müller, Joachim Niehren, & Andreas Podelski. Ordering constraints over feature trees. *Constraints, an International Journal*, 5(1–2), January 2000. Special Issue on CP'97, Linz, Austria.

[Mukai, 1988] Kuniaki Mukai. Partially specified terms in logic programming for linguistic analysis. In *Proceedings of the 6th International Conference on Fifth Generation Computer Systems*, Tokyo, Japan, 1988. ICOT.

[Müller & Niehren, 1998] Martin Müller & Joachim Niehren. Ordering constraints over feature trees expressed in second-order monadic logic. In Tobias Nipkow, editor, *International Conference on Rewriting Techniques and Applications*, volume 1379 of *Lecture Notes in Computer Science*, pages 196–210, Tsukuba, Japan, 1998. Springer-Verlag, Berlin.

[Müller *et al.*, 1998] Martin Müller, Joachim Niehren, & Ralf Treinen. The first-order theory of ordering constraints over feature trees. In *Proceedings of the 13th IEEE Symposium on Logic in Computer Science*, pages 432–443. IEEE Computer Society Press, 1998.

[Müller *et al.*, 2000] Martin Müller, Joachim Niehren, & Ralf Treinen. The first-order theory of ordering constraints over feature trees, 2000. Extended version of [Müller *et al.*, 1998]. Submitted.

[Müller, to appear] Martin Müller. Ordering constraints over feature trees with ordered sorts. In P. Lopez, Suresh Manandhar, & Werner Nutt, editors, *Computational Logic and Natural Language Understanding*, Lecture Notes in Artificial Intelligence. Springer-Verlag, Berlin, to appear. Available at `http://www.ps.uni-sb.de/~mmueller/papers/clnlp.ps.Z`.

[Nebel & Smolka, 1990] Bernhard Nebel & Gert Smolka. Representation and reasoning with attributive descriptions. In K.H. Bläsius, U.Hedtstück, & C.-R. Rollinger, editors, *Sorts and Types in Artificial Intelligence*, volume 418 of *Lecture Notes in Artificial Intelligence*, pages 112–139. Springer-Verlag, Berlin, 1990.

[Nebel, 1990] Bernhard Nebel. *Reasoning and Revision in Hybrid Representation Systems*, volume 422 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin, 1990.

[Pollard & Sag, 1994] Carl Pollard & Ivan Sag. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. Cambridge University Press, Cambridge, England, 1994.

[Rabin, 1969] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[Rounds, 1997] William C. Rounds. Feature logics. In Johan van Benthem & Alice ter Meulen, editors, *Handbook of Logic and Language*, pages 475–533. Elsevier Science Publishers B.V. (North Holland), 1997. Part 2: General Topics.

[Shieber *et al.*, 1983] Steward Shieber, Hans Uszkoreit, Fernando Pereira, J. Alan Robinson, & M. Tyson. The formalism and implementation of PATR-II. In Joan Bresnan, editor, *Research on Interactive Acquisition and Use of Knowledge*. SRI International, Menlo Park, California, 1983.

[Shieber, 1986] Steward Shieber. *An Introduction to Unification-based Approaches to Grammar*. CSLI Lecture Notes No. 4. Center for the Study of Language and Information, 1986.

[Smolka & Treinen, 1994] Gert Smolka & Ralf Treinen. Records for logic programming. *The Journal of Logic Programming*, 18(3):229–258, April 1994.

[Smolka, 1995] Gert Smolka. The Oz Programming Model. In Jan van Leeuwen, editor, *Computer Science Today*, volume 1000 of *Lecture Notes in Computer Science*, pages 324–343. Springer-Verlag, Berlin, 1995.

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

[Thomas, 1990] Wolfgang Thomas. Automata on infinite objects. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B (Formal Models and Semantics), chapter 4, pages 133–191. The MIT Press, Cambridge, MA, 1990.

[Thomas, 1997] Wolfgang Thomas. Languages, automata, and logic. In Grzegorz Rozenberg & Arto Salomaa, editors, *Handbook of Formal Languages*, volume 3. Springer-Verlag, Berlin, 1997.

[Vorobyov, 1996] Sergei Vorobyov. An improved lower bound for the elementary theories of trees. In *Proceedings of the Conference on Automated Deduction*, volume 1104 of *Lecture Notes in Computer Science*, pages 275–287, Rutgers, NJ, 1996. Springer-Verlag, Berlin.