A COMPLETE AND RECURSIVE FEATURE THEORY*

Rolf Backofen and Gert Smolka
German Research Center for Artificial Intelligence (DFKI)
W-6600 Saarbrücken, Germany
{backofen,smolka}@dfki.uni-sb.de

Abstract

Various feature descriptions are being employed in constrained-based grammar formalisms. The common notational primitive of these descriptions are functional attributes called features. The descriptions considered in this paper are the possibly quantified first-order formulae obtained from a signature of features and sorts. We establish a complete first-order theory FT by means of three axiom schemes and construct three elementarily equivalent models.

One of the models consists of so-called feature graphs, a data structure common in computational linguistics. The other two models consist of so-called feature trees, a record-like data structure generalizing the trees corresponding to first-order terms.

Our completeness proof exhibits a terminating simplification system deciding validity and satisfiability of possibly quantified feature descriptions.

1 Introduction

Feature descriptions provide for the typically partial description of abstract objects by means of functional attributes called features. They originated in the late seventies with so-called unification grammars [14], a by now popular family of declarative grammar formalisms for the description and processing of natural language. More recently, the use of feature descriptions in logic programming has been advocated and studied [2, 3, 4, 17, 16]. Essentially, feature descriptions provide a logical version of records, a data structure found in many programming languages.

Feature descriptions have been proposed in various forms with various formalizations [1, 13, 9, 15, 5, 10]. We will follow the logical approach pioneered by [15], which accommodates feature descriptions as standard first-order formulae interpreted in first-order structures. In this approach, a semantics for feature descriptions can be given by means of a feature theory (i.e., a set of closed feature descriptions having at least one model). There are two complementary ways of specifying a feature theory: either by explicitly constructing a standard model and taking all sentences valid in it, or by stating axioms and proving their consistency. Both possibilities are exemplified in [15]: the feature graph algebra $\mathcal{F}$ is given as a standard model, and the class of feature algebras is obtained by means of an axiomatization.

Both approaches to fixing a feature theory have their advantages. The construction of a standard model provides for a clear intuition and yields a complete feature theory (i.e., if ϕ is a closed feature description, then either ϕ or $\neg\phi$ is valid). The presentation of a recursively enumerable axiomatization has the advantage that we inherit from predicate logic a sound and complete deduction system for valid feature descriptions.

The ideal case then is to specify a feature theory by both a standard model and a corresponding recursively enumerable axiomatization. The existence of such a double characterization, however, is by no means obvious since it implies that the feature theory is decidable. In fact, so far no decidable, consistent and complete feature theory has been known.

In this paper we will establish a complete and decidable feature theory FT by means of three axiom schemes. We will also construct three models of FT, two consisting of so-called feature trees, and one consisting of so-called feature graphs. Since FT is complete, all three models are elementarily equivalent (i.e., satisfy exactly the same first-order formulae). While the feature graph model captures intuitions common in linguistically motivated investigations, the feature tree model provides the connection to the tree constraint systems [8, 11, 12] employed in logic programming.

Our proof of FT's completeness will exhibit a simplification algorithm that computes for every feature description an equivalent solved form from which the solutions of the description can be read of easily. For a closed feature description the solved form is either $\top$ (which means that the description is valid) or $\bot$ (which means that the description is invalid). For

*We appreciate discussions with Joachim Niehren and Ralf Treinen who read a draft version of this paper. The research reported in this paper has been supported by the Bundesminister für Forschung und Technologie under contracts ITW 90002 0 (DISCO) and ITW 9105 (Hydra). For space limitations proofs are omitted; they can be found in the complete paper [6].

a feature description with free variables the solved form is $\bot$ if and only if the description is unsatisfiable.

1.1 Feature Descriptions

Feature descriptions are first-order formulae built over an alphabet of binary predicate symbols, called *features*, and an alphabet of unary predicate symbols, called *sorts*. There are no function symbols. In admissible interpretations features must be functional relations, and distinct sorts must be disjoint sets. This is stated by the first and second axiom scheme of FT:

(Ax1) $\forall x \forall y \forall z (f(x,y) \wedge f(x,z) \rightarrow y \doteq z)$ (for every feature f)

(Ax2) $\forall x (A(x) \wedge B(x) \rightarrow \bot)$ (for every two distinct sorts A and B).

A typical feature description written in matrix notation is

$$x \;:\; \exists y \begin{bmatrix} \mathit{woman} & & \\ \mathit{father} & : & \begin{bmatrix} \mathit{engineer} \\ \mathit{age} : y \end{bmatrix} \\ \mathit{husband} & : & \begin{bmatrix} \mathit{painter} \\ \mathit{age} : y \end{bmatrix} \end{bmatrix}.$$

It may be read as saying that x is a woman whose father is an engineer, whose husband is a painter, and whose father and husband are both of the same age. Written in plain first-order syntax we obtain the less suggestive formula

$$\begin{aligned} \exists y, F, H\ (\ & \mathit{woman}(X) \wedge \\ & \mathit{father}(x,F) \wedge \mathit{engineer}(F) \wedge \mathit{age}(F,y) \wedge \\ & \mathit{husband}(x,H) \wedge \mathit{painter}(H) \wedge \mathit{age}(H,y)\ ). \end{aligned}$$

The axiom schemes (Ax1) and (Ax2) still admit trivial models where all features and sorts are empty. The third and final axiom scheme of FT states that certain "consistent" descriptions have solutions. Three Examples of instances of FT's third axiom scheme are

$$\begin{gathered} \exists x, y, z\ (f(x,y) \wedge A(y) \wedge g(x,z) \wedge B(z)) \\ \forall u, z\ \exists x, y\ (f(x,y) \wedge g(y,u) \wedge h(y,z) \wedge yf\uparrow) \\ \forall z\ \exists x, y\ (f(x,y) \wedge g(y,x) \wedge h(y,z) \wedge yf\uparrow), \end{gathered}$$

where $yf\uparrow$ abbreviates $\neg\exists z(f(y,z))$. Note that the third description

$$f(x,y) \wedge g(y,x) \wedge h(y,z) \wedge fy\uparrow$$

is "cyclic" with respect to the variables x and y.

1.2 Feature Trees

A feature tree (examples are shown in Figure 1) is a tree whose edges are labeled with features, and whose nodes are labeled with sorts. As one would expect, the labeling with features must be deterministic, that is, the direct subtrees of a feature tree must be uniquely identified by the features of the

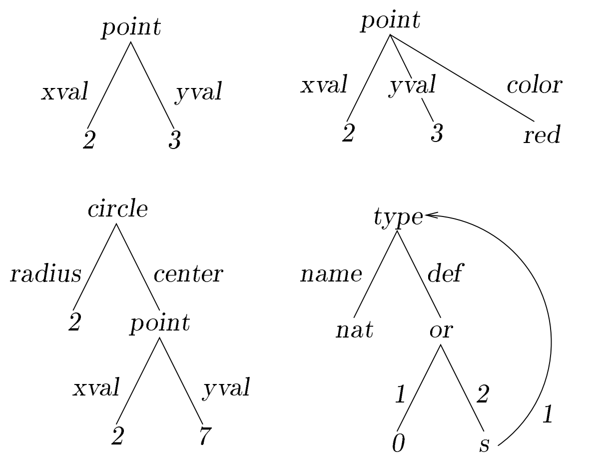

Figure 1: Examples of Feature Trees.

edges leading to them. Feature trees can be seen as a mathematical model of records in programming languages. Feature trees without subtrees model atomic values (e.g., numbers). Feature trees may be finite or infinite, where infinite feature trees provide for the convenient representation of cyclic data structures. The last example in Figure 1 gives a finite graph representation of an infinite feature tree, which may arise as the representation of the recursive type equation $\mathit{nat} = \mathit{0} + s(\mathit{nat})$.

Feature descriptions are interpreted over feature trees as one would expect:

- Every sort symbol A is taken as a unary predicate, where a *sort constraint* $A(x)$ holds if and only if the root of the tree x is labeled with A.

- Every feature symbol f is taken as a binary predicate, where a *feature constraint* $f(x,y)$ holds if and only if the tree x has the direct subtree y at feature f.

The theory of the corresponding first-order structure (i.e., the set of all closed formulae valid in this structure) is called FT. We will show that FT is in fact exactly the theory specified by the three axiom schemes outlined above, provided the alphabets of sorts and features are both taken to be infinite. Hence FT is complete (since it is the theory of the feature tree structure) and decidable (since it is complete and specified by a recursive set of axioms).

Another, elementarily equivalent, model of FT is the substructure of the feature tree structure obtained by admitting only rational feature trees (i.e., finitely branching trees having only finitely many subtrees). Yet another model of FT can be obtained from so-called feature graphs, which are finite, directed, possibly cyclic graphs labelled with sorts and features similar to feature trees. In contrast to feature trees, nodes of feature graphs may or may not be labelled with sorts. Feature graphs correspond to the so-called feature structures commonly found in linguistically motivated investigations [14, 7].

1.3 Organization of the Paper

Section 2 recalls the necessary notions and notations from Predicate Logic. Section 3 defines the theory FT by means of three axiom schemes. Section 4 establishes the overall structure of the completeness proof by means of a lemma. Section 5 studies quantifier-free conjunctive formulae, gives a solved form, and introduces path constraints. Section 6 defines feature trees and graphs and establishes the respective models of FT. Section 7 studies the properties of so-called prime formulae, which are the basic building stones of the solved form for general feature constraints. Section 8 presents the quantifier elimination lemmas and completes the completeness proof.

2 Preliminaries

Throughout this paper we assume a signature SOR$\uplus$FEA consisting of an infinite set SOR of unary predicate symbols called **sorts** and an infinite set FEA of binary predicate symbols called **features**. For the completeness of our axiomatization it is essential that there are both infinitely many sorts and infinitely many features.The letters A, B, C will always denote sorts, and the letters f, g, h will always denote features.

A **path** is a word (i.e., a finite, possibly empty sequence) over the set of all features. The symbol ε denotes the empty path, which satisfies $\varepsilon p = p = p\varepsilon$ for every path p. A path p is called a **prefix** of a path q, if there exists a path p' such that $pp' = q$.

We also assume an infinite alphabet of variables and adopt the convention that x, y, z always denote variables, and X, Y always denote finite, possibly empty sets of variables. Under our signature SOR $\uplus$ FEA, every term is a variable, and an atomic formula is either a **feature constraint** xfy ($f(x,y)$ in standard notation), a **sort constraint** Ax ($A(x)$ in standard notation), an equation $x \doteq y$, $\perp$ ("false"), or $\top$ ("true"). Compound formulae are obtained as usual with the connectives $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\neg$ and the quantifiers $\exists$ and $\forall$. We use $\tilde{\exists}\phi$ [$\tilde{\forall}\phi$] to denote the existential [universal] closure of a formula ϕ. Moreover, $\mathcal{V}(\phi)$ is taken to denote the set of all variables that occur free in a formula ϕ. The letters ϕ and ψ will always denote formulae.

We assume that the conjunction of formulae is an associative and commutative operation that has $\top$ as neutral element. This means that we identify $\phi \wedge (\psi \wedge \theta)$ with $\theta \wedge (\psi \wedge \phi)$, and $\phi \wedge \top$ with ϕ (but not, for example, $xfy \wedge xfy$ with xfy). A conjunction of atomic formulae can thus be seen as the finite multiset of these formulae, where conjunction is multiset union, and $\top$ (the "empty conjunction") is the empty multiset. We will write $\psi \subseteq \phi$ (or $\psi \in \phi$, if ψ is an atomic formula) if there exists a formula ψ' such that $\psi \wedge \psi' = \phi$.

Moreover, we identify $\exists x \exists y \phi$ with $\exists y \exists x \phi$. If $X = \{x_1, \ldots, x_n\}$, we write $\exists X \phi$ for $\exists x_1 \ldots \exists x_n \phi$. If $X = \emptyset$, then $\exists X \phi$ stands for ϕ.

Structures and satisfaction of formulae are defined as usual. A valuation into a structure $\mathcal{A}$ is a total function from the set of all variables into the universe $|\mathcal{A}|$ of $\mathcal{A}$. A valuation α' into $\mathcal{A}$ is called an **x-update** [**X-update**] of a valuation α into $\mathcal{A}$ if α' and α agree everywhere but possibly on x [X]. We use $\phi^{\mathcal{A}}$ to denote the set of all valuations α such that $\mathcal{A}, \alpha \models \phi$. We write $\phi \models \psi$ ("ϕ entails ψ") if $\phi^{\mathcal{A}} \subseteq \psi^{\mathcal{A}}$ for all structures $\mathcal{A}$, and $\phi \models\mkern-12mu\models \psi$ ("ϕ is equivalent to ψ") if $\phi^{\mathcal{A}} = \psi^{\mathcal{A}}$ for all structures $\mathcal{A}$.

A **theory** is a set of closed formulae. A **model** of a theory is a structure that satisfies every formulae of the theory. A formula ϕ is a **consequence of a theory** T ($T \models \phi$) if $\tilde{\forall}\phi$ is valid in every model of T. A formula ϕ **entails** a formula ψ in a theory T ($\phi \models_T \psi$) if $\phi^{\mathcal{A}} \subseteq \psi^{\mathcal{A}}$ for every model $\mathcal{A}$ of T. Two formulae ϕ, ψ are **equivalent** in a theory T ($\phi \models\mkern-12mu\models_T \psi$) if $\phi^{\mathcal{A}} = \psi^{\mathcal{A}}$ for every model $\mathcal{A}$ of T.

A theory T is **complete** if for every closed formula ϕ either ϕ or $\neg\phi$ is a consequence of T. A theory is **decidable** if the set of its consequences is decidable. Since the consequences of a recursively enumerable theory are recursively enumerable (completeness of first-order deduction), a complete theory is decidable if and only if it is recursively enumerable.

Two first-order structures $\mathcal{A}$, $\mathcal{B}$ are **elementarily equivalent** if, for every first-order formula ϕ, ϕ is valid in $\mathcal{A}$ if and only if ϕ is valid in $\mathcal{B}$. Note that all models of a complete theory are elementarily equivalent.

3 The Axioms

The first axiom scheme says that features are functional:

(Ax1) $\forall x \forall y \forall z (xfy \wedge xfz \rightarrow y \doteq z)$ (for every feature f).

The second scheme says that sorts are mutually disjoint:

(Ax2) $\forall x (Ax \wedge Bx \rightarrow \perp)$ (for every two distinct sorts A and B).

The third and final axiom scheme will say that certain "consistent feature descriptions" are satisfiable. For its formulation we need the important notion of a solved clause.

An **exclusion constraint** is an additional atomic formula of the form $xf\uparrow$ ("f undefined on x") taken to be equivalent to $\neg\exists y\,(xfy)$ (for some variable $y \neq x$).

A **solved clause** is a possibly empty conjunction ϕ of atomic formulae of the form xfy, Ax and $xf\uparrow$ such that the following conditions are satisfied:

1. no atomic formula occurs twice in ϕ
2. if $Ax \in \phi$ and $Bx \in \phi$, then $A = B$
3. if $xfy \in \phi$ and $xfz \in \phi$, then $y = z$
4. if $xfy \in \phi$, then $xf\uparrow \notin \phi$.

Figure 2 gives a graph representation of the solved clause

$$xfu \wedge xgv \wedge xh\uparrow \wedge$$

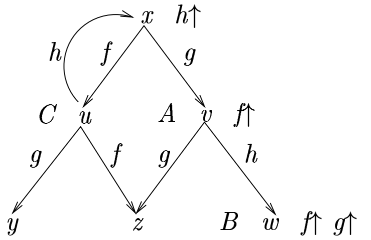

Figure 2: A graph representation of a solved clause.

$$\begin{aligned} & Cu \wedge uhx \wedge ugy \wedge ufz \wedge \\ & Av \wedge vgz \wedge vhw \wedge vf\uparrow \wedge \\ & Bw \wedge wf\uparrow \wedge wg\uparrow \,. \end{aligned}$$

As in the example, a solved clause can always be seen as the graph whose nodes are the variables appearing in the clause and whose arcs are given by the feature constraints xfy. The constraints Ax, $xf\uparrow$ appear as labels of the node x.

A variable x is **constrained** in a solved clause ϕ if ϕ contains a constraint of the form Ax, xfy or $xf\uparrow$. We use $\mathcal{CV}(\phi)$ to denote the set of all variables that are constrained in ϕ. The variables in $\mathcal{V}(\phi) - \mathcal{CV}(\phi)$ are called the **parameters** of a solved clause ϕ. In the graph representation of a solved clause the parameters appear as leaves that are not not labeled with a sort or a feature exclusion. The parameters of the solved clause in Figure 2 are y and z.

We can now state the third axiom scheme. It says that the constrained variables of a solved clause have solutions for all values of the parameters:

(Ax3) $\tilde{\forall}\exists X\phi$ (for every solved clause ϕ and $X = \mathcal{CV}(\phi)$).

The **theory** FT is the set of all sentences that can be obtained as instances of the axiom schemes (Ax1), (Ax2) and (Ax3). The **theory** FT_0 is the set of all sentences that can be obtained as instances of the first two axiom schemes.

As the main result of this paper we will show that FT is a complete and decidable theory.

By using an adaption of the proof of Theorem 8.3 in [15] one can show that FT_0 is undecidable.

4 Outline of the Completeness Proof

The completeness of FT will be shown by exhibiting a simplification algorithm for FT. The following lemma gives the overall structure of the algorithm, which is the same as in Maher's [12] completeness proof for the theory of constructor trees.

Lemma 4.1 *Suppose there exists a set of so-called prime formulae such that:*

1. *every sort constraint Ax, every feature constraint xfy, and every equation $x \doteq y$ such that $x \neq y$ is a prime formula*
2. *$\top$ is a prime formula, and there is no other closed prime formula*
3. *for every two prime formulae β and β' one can compute a formula δ that is either prime or $\bot$ and satisfies*
$$\beta \wedge \beta' \models\!\!\!\mid_{FT} \delta \quad \text{and} \quad \mathcal{V}(\delta) \subseteq \mathcal{V}(\beta \wedge \beta')$$
4. *for every prime formula β and every variable x one can compute a prime formula β' such that*
$$\exists x\beta \models\!\!\!\mid_{FT} \beta' \quad \text{and} \quad \mathcal{V}(\beta') \subseteq \mathcal{V}(\exists x\beta)$$
5. *if β, $\beta_1, \ldots, \beta_n$ are prime formulae, then*
$$\exists x(\beta \wedge \bigwedge_{i=1}^{n} \neg\beta_i) \models\!\!\!\mid_{FT} \bigwedge_{i=1}^{n} \exists x(\beta \wedge \neg\beta_i)$$
6. *for every two prime formulae β, β' and every variable x one can compute a Boolean combination δ of prime formulae such that*
$$\exists x(\beta \wedge \neg\beta') \models\!\!\!\mid_{FT} \delta \quad \text{and} \quad \mathcal{V}(\delta) \subseteq \mathcal{V}(\exists x(\beta \wedge \neg\beta')).$$

Then one can compute for every formula ϕ a Boolean combination δ of prime formulae such that $\phi \models\!\!\!\mid_{FT} \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\phi)$.

Proof. Suppose a set of prime formulae as required exists. Let ϕ be a formula. We show by induction on the structure of ϕ how to compute a Boolean combination δ of prime formulae such that $\phi \models\!\!\!\mid_{FT} \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\phi)$.

If ϕ is an atomic formula Ax, xfy or $x \doteq y$, then ϕ is either a prime formula, or ϕ is a trivial equation $x \doteq x$, in which case it is equivalent to the prime formula $\top$.

If ϕ is $\neg\psi$, $\psi \wedge \psi'$ or $\psi \vee \psi'$, then the claim follows immediately with the induction hypothesis.

It remains to show the claim for $\phi = \exists x\psi$. By the induction hypothesis we know that we can compute a Boolean combination δ of prime formulae such that $\delta \models\!\!\!\mid_{FT} \psi$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\psi)$. Now δ can be transformed to a disjunctive normal form where prime formulae play the role of atomic formulae; that is, δ is equivalent to $\sigma_1 \vee \ldots \vee \sigma_n$, where every "clause" σ_i is a conjunction of prime and negated prime formulae. Hence

$$\exists x\phi \models\!\!\!\mid \exists x(\sigma_1 \vee \ldots \vee \sigma_n) \models\!\!\!\mid \exists x\sigma_1 \vee \ldots \vee \exists x\sigma_n,$$

where all three formulae have exactly the same free variables. It remains to show that one can compute for every clause σ a Boolean combination δ of prime formulae such that $\exists x\sigma \models\!\!\!\mid_{FT} \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\exists x\sigma)$. We distinguish the following cases.

(i) $\sigma = \beta$ for some basic formula β. Then the claim follows by assumption (4).

(ii) $\sigma = \beta \wedge \bigwedge_{i=1}^{n} \neg\beta_i$, $n > 0$. Then the claim follows with assumptions (5) and (6).

(iii) $\sigma = \bigwedge_{i=1}^{n} \neg\beta_i$, $n > 0$. Then $\sigma \models\!\!\!\mid_{FT} \top \wedge \bigwedge_{i=1}^{n} \neg\beta_i$ and the claim follows with case (ii) since $\top$ is a prime formula by assumption (2).

(iv) $\sigma = \beta_1 \wedge \ldots \wedge \beta_k \wedge \neg\beta'_1 \wedge \ldots \wedge \beta'_n$, $k > 1$, $n \geq 0$. Then we know by assumption (3) that either $\beta_1 \wedge \ldots \wedge \beta_k \models\!\!\!\mid_{FT} \bot$ or $\beta_1 \wedge \ldots \wedge \beta_k \models\!\!\!\mid_{FT} \beta$ for some prime formula β. In the former case we choose $\delta = \neg\top$, and in the latter case the claim follows with case (i) or (ii). $\square$

Note that, provided a set of prime formulae with the required properties exists, the preceding lemma yields the completeness of FT since every closed formula can be simplified to $\top$ or $\neg\top$ (since $\top$ is the only closed prime formula).

In the following we will establish a set of prime formula as required.

5 Solved Formulae

In this section we introduce a solved form for conjunctions of atomic formulae.

A **basic formula** is either $\perp$ or a possibly empty conjunction of atomic formulae of the form Ax, xfy, and $x \doteq y$. Note that $\top$ is a basic formula since $\top$ is the empty conjunction.

Every basic formula $\phi \neq \perp$ has a unique decomposition $\phi = \phi_N \wedge \phi_G$ into a possibly empty conjunction ϕ_N of equations "$x \doteq y$" and a possibly empty conjunction ϕ_G of sort constraints "Ax" and feature constraints "xfy". We call ϕ_N the **normalizer** and and ϕ_G the **graph** of ϕ.

We say that a basic formula ϕ **binds** x **to** y if $x \doteq y \in \phi$ and x occurs only once in ϕ. Here it is important to note that we consider equations as directed, that is, assume that $x \doteq y$ is different from $y \doteq x$ if $x \neq y$. We say that ϕ **eliminates** x if ϕ binds x to some variable y.

A **solved formula** is a basic formula $\gamma \neq \perp$ such that the following conditions are satisfied:

1. an equation $x \doteq y$ appears in γ if and only if γ eliminates x
2. the graph of γ is a solved clause.

Note that a solved clause not containing exclusion constraints is a solved formula, and that a solved formula not containing equations is a solved clause. The letter γ will always denote a solved formula.

We will see that every basic formula is equivalent in FT_0 to either $\perp$ or a solved formula.

Figure 3 shows the so-called **basic simplification rules**. With $\phi[x \leftarrow y]$ we denote the formula that is obtained from ϕ by replacing every occurrence of x with y. We say that a formula ϕ **simplifies to** a formula ψ by a simplification rule ρ if $\frac{\phi}{\psi}$ is an instance of ρ. We say that a basic formula ϕ **simplifies to** a basic formula ψ if either $\phi = \psi$ or ϕ simplifies to ψ in finitely many steps each licensed by one of basic simplification rules in Figure 3.

Note that the basic simplification rules (1) and (2) correspond to the first and second axiom scheme, respectively. Thus they are equivalence transformation with respect to FT_0. The remaining three simplification rules are equivalence transformations in general.

Proposition 5.1 *The basic simplification rules are terminating and perform equivalence transformations with respect to FT_0. Moreover, a basic formula $\phi \neq \perp$ is solved if and only if no basic simplification rule applies to it.*

Proposition 5.2 *Let ϕ be a formula built from atomic formulae with conjunction. Then one can compute a formula δ that is either solved or $\perp$ such that $\phi \models\!\!\mid_{FT_0} \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\phi)$.*

1. $$\frac{xfy \wedge xfz \wedge \phi}{xfz \wedge y \doteq z \wedge \phi}$$
2. $$\frac{Ax \wedge Bx \wedge \phi}{\perp} \quad A \neq B$$
3. $$\frac{Ax \wedge Ax \wedge \phi}{Ax \wedge \phi}$$
4. $$\frac{x \doteq y \wedge \phi}{x \doteq y \wedge \phi[x \leftarrow y]} \quad x \in \mathcal{V}(\phi) \text{ and } x \neq y$$
5. $$\frac{x \doteq x \wedge \phi}{\phi}$$

Figure 3: The basic simplification rules.

In the quantifier elimination proofs to come it will be convenient to use so-called path constraints, which provide a flexible syntax for atomic formulae closed under conjunction and existential quantification. We start by defining the denotation of a path.

The interpretations $f^{\mathcal{A}}$, $g^{\mathcal{A}}$ of two features f, g in a structure $\mathcal{A}$ are binary relations on the universe $|\mathcal{A}|$ of $\mathcal{A}$; hence their composition $f^A \circ g^{\mathcal{A}}$ is again a binary relation on $|\mathcal{A}|$ satisfying

$$a(f^{\mathcal{A}} \circ g^{\mathcal{A}})b \quad \Longleftrightarrow \quad \exists c \in |\mathcal{A}|\colon\ af^{\mathcal{A}}c \ \wedge\ cf^{\mathcal{A}}b$$

for all $a, b \in |\mathcal{A}|$. Consequently we define the **denotation** $p^{\mathcal{A}}$ **of a path** $p = f_1 \cdots f_n$ in a structure $\mathcal{A}$ as the composition

$$(f_1 \cdots f_n)^{\mathcal{A}} \quad := \quad f_1^{\mathcal{A}} \circ \cdots \circ f_n^{\mathcal{A}},$$

where the empty path ε is taken to denote the identity relation. If $\mathcal{A}$ is a model of the theory FT_0, then every paths denotes a unary partial function on the universe of $\mathcal{A}$. Given an element $a \in |\mathcal{A}|$, $p^{\mathcal{A}}$ is thus either undefined on a or leads from a to exactly one $b \in |\mathcal{A}|$.

Let p, q be paths, x, y be variables, and A be a sort. Then **path constraints** are defined as follows:

$$\begin{aligned}
\mathcal{A}, \alpha \models xpy \quad &:\Longleftrightarrow \alpha(x)\, p^{\mathcal{A}}\, \alpha(y) \\
\mathcal{A}, \alpha \models xp{\downarrow}yq \ &:\Longleftrightarrow \exists a \in |\mathcal{A}|\colon\ \alpha(x)\, p^{\mathcal{A}}\, a \wedge \alpha(y)\, q^{\mathcal{A}}\, a \\
\mathcal{A}, \alpha \models Axp \quad &:\Longleftrightarrow \exists a \in |\mathcal{A}|\colon\ \alpha(x)\, p^{\mathcal{A}}\, a \wedge a \in A^{\mathcal{A}}.
\end{aligned}$$

Note that path constraints xpy generalize feature constraints xfy.

A **proper path constraint** is a path constraint of the form "Axp" or "$xp{\downarrow}yq$".

Every path constraint can be expressed with the already existing formulae, as can be seen from the following equivalences:

$$\begin{aligned}
x\varepsilon y \quad &\models\!\!\mid \quad x \doteq y \\
xfpy \quad &\models\!\!\mid \quad \exists z(xfz \wedge zpy) \qquad (z \neq x, y) \\
xp{\downarrow}yq \quad &\models\!\!\mid \quad \exists z(xpz \wedge yqz) \qquad (z \neq x, y) \\
Axp \quad &\models\!\!\mid \quad \exists y(xpy \wedge Ay) \qquad (y \neq x).
\end{aligned}$$

The **closure** $[\gamma]$ **of a solved formula** γ is the closure of the atomic formulae occurring in γ with respect to the following deduction rules:

$$\frac{}{x\varepsilon x} \quad \frac{x \doteq y}{x\varepsilon y} \quad \frac{xpy \quad yfz}{xpfz} \quad \frac{xpz \quad yqz}{xp{\downarrow}yq} \quad \frac{Ay \quad xpy}{Axp}.$$

Recall that we assume that equations $x \doteq y$ are directed, that is, are ordered pairs of variables. Hence, $x\epsilon y \in [\gamma]$ and $y\epsilon x \notin [\gamma]$ if $x \doteq y \in \gamma$.

The **closure of a solved clause** δ is defined analogously.

Proposition 5.3 *Let γ be a solved formula. Then:*

1. *if $\pi \in [\gamma]$, then $\gamma \models \pi$*
2. *$x\varepsilon y \in [\gamma]$ iff $x = y$ or $x \doteq y \in \gamma$*
3. *$xfy \in [\gamma]$ iff $xfy \in \gamma$ or $\exists z$: $x \doteq z \in \gamma$ and $zfy \in \gamma$*
4. *$xpfy \in [\gamma]$ iff $\exists z$: $xpz \in [\gamma]$ and $zfy \in \gamma$*
5. *if $p \neq \varepsilon$ and $xpy, xpz \in [\gamma]$, then $y = z$*
6. *it is decidable whether a path constraint is in $[\gamma]$.*

6 Feature Trees and Feature Graphs

In this section we establish three models of FT consisting of either feature trees or feature graphs. Since we will show that FT is a complete theory, all three models are in fact elementarily equivalent.

A **tree domain** is a nonempty set $D \subseteq \mathrm{FEA}^\star$ of paths that is **prefix-closed**, that is, if $pq \in D$, then $p \in D$. Note that every tree domain contains the empty path.

A **feature tree** is a partial function $\sigma\colon \mathrm{FEA}^\star \to \mathrm{SOR}$ whose domain is a tree domain. The paths in the domain of a feature tree represent the nodes of the tree; the empty path represents its root. We use D_σ to denote the domain of a feature tree σ. A feature tree is called **finite** [**infinite**] if its domain is finite [infinite]. The letters σ and τ will always denote feature trees.

The **subtree** $p\sigma$ of a feature tree σ at a path $p \in D_\sigma$ is the feature tree defined by (in relational notation)

$$p\sigma \quad := \quad \{(q, A) \mid (pq, A) \in \sigma\}.$$

A feature tree σ is called a **subtree** of a feature tree τ if σ is a subtree of τ at some path $p \in D_\tau$, and a **direct subtree** if $p = f$ for some feature f.

A feature tree σ is called **rational** if (1) σ has only finitely many subtrees and (2) σ is finitely branching (i.e., for every $p \in D_\sigma$, the set $\{pf \in D_\sigma \mid f \in \mathrm{FEA}\}$ is finite). Note that for every rational feature tree σ there exist finitely many features $f_1, \ldots, f_n$ such that $D_\sigma \subseteq \{f_1, \ldots, f_n\}^\star$.

The **feature tree structure** $\mathcal{T}$ is the $\mathrm{SOR}\uplus\mathrm{FEA}$-structure defined as follows:

- the universe of $\mathcal{T}$ is the set of all feature trees
- $\sigma \in A^\mathcal{T}$ iff $\sigma(\varepsilon) = A$ (i.e., σ's root is labeled with A)
- $(\sigma, \tau) \in f^\mathcal{T}$ iff $f \in D_\sigma$ and $\tau = f\sigma$ (i.e., τ is the subtree of σ at f).

The **rational feature tree structure** $\mathcal{R}$ is the substructure of $\mathcal{T}$ consisting only of the rational feature trees.

Theorem 6.1 *The feature tree structures $\mathcal{T}$ and $\mathcal{R}$ are models of the theory FT.*

A **feature pregraph** is a pair (x, γ) consisting of a variable x (called the **root**) and a solved clause γ not containing exclusion constraints such that, for every variable y occurring in γ, there exists a path p satisfying $xpy \in [\gamma]$. If one deletes the exclusion constraints in Figure 2, one obtains the graphical representation of a feature pregraph whose root is x.

A feature pregraph (x, γ) is called a **subpregraph** of a feature pregraph (y, δ) if $\gamma \subseteq \delta$ and $x = y$ or $x \in \mathcal{V}(\delta)$. Note that a feature pregraph has only finitely many subpregraphs.

We say that two feature pregraphs are **equivalent** if they are equal up to consistent variable renaming. For instance, $(x,\ xfy \wedge ygx)$ and $(u,\ ufx \wedge xgu)$ are equivalent feature pregraphs.

A **feature graph** is an element of the quotient of the set of all feature pregraphs with respect to equivalence as defined above. We use $\overline{(x, \gamma)}$ to denote the feature graph obtained as the equivalence class of the feature pregraph (x, γ).

In contrast to feature trees, not every node of a feature graph must carry a sort.

The **feature graph structure** $\mathcal{G}$ is the $\mathrm{SOR} \uplus \mathrm{FEA}$-structure defined as follows:

- the universe of $\mathcal{G}$ is the set of all feature graphs
- $\overline{(x, \gamma)} \in A^\mathcal{G}$ iff $Ax \in \gamma$
- $(\overline{(x, \gamma)}, \sigma) \in f^\mathcal{G}$ iff there exists a maximal feature subpregraph (y, δ) of (x, γ) such that $xfy \in \gamma$ and $\sigma = \overline{(y, \delta)}$.

Theorem 6.2 *The feature graph structure $\mathcal{G}$ is a model of the theory FT.*

Let $\mathcal{F}$ be the structure whose domain consists of all feature pregraphs and that is otherwise defined analogous to $\mathcal{G}$. Note that $\mathcal{G}$ is in fact the quotient of $\mathcal{F}$ with respect to equivalence of feature pregraphs.

Proposition 6.3 *The feature pregraph structure $\mathcal{F}$ is a model of FT_0 but not of FT.*

7 Prime Formulae

We now define a class of prime formulae having the properties required by Lemma 4.1. The prime formulae will turn out to be solved forms for formulae built from atomic formulae with conjunction and existential quantification.

A **prime formula** is a formula $\exists X \gamma$ such that

1. γ is a solved formula
2. X has no variable in common with the normalizer of γ
3. every $x \in X$ can be reached from a free variable, that is, there exists a path constraint $ypx \in [\gamma]$ such that $y \notin X$.

The letter β will always denote a prime formula.

Note that $\top$ is the only closed prime formula, and that $\exists X\gamma$ is a prime formula if $\exists x\exists X\gamma$ is a prime formula. Moreover, every solved formula is a prime formula, and every quantifier-free prime formula is a solved formula.

The definition of prime formulae certainly fulfills the requirements (1) and (2) of Lemma 4.1. The fulfillment of the requirements (3) and (4) will be shown in this section, and the fulfillment of the requirements (5) and (6) will be shown in the next section.

Proposition 7.1 *Let $\exists X\gamma$ be a prime formula, $\mathcal{A}$ be a model of FT, and $\mathcal{A},\alpha \models \exists X\gamma$. Then there exists one and only one X-update α' of α such that $\mathcal{A},\alpha' \models \gamma$.*

The next proposition establishes that prime formulae are closed under existential quantification (property (4) of Lemma 4.1).

Proposition 7.2 *For every prime formula β and every variable x one can compute a prime formula β' such that $\exists x\beta \models\!\!\!|_{FT}\ \beta'$ and $\mathcal{V}(\beta') \subseteq \mathcal{V}(\exists x\beta)$.*

Proposition 7.3 *If β is a prime formula, then $FT \models \tilde{\exists}\beta$.*

The next proposition establishes that prime formulae are closed under consistent conjunction (property (3) of Lemma 4.1).

Proposition 7.4 *For every two prime formulae β and β' one can compute a formula δ that is either prime or $\bot$ and satisfies*

$$\beta \wedge \beta' \models\!\!\!|_{FT}\ \delta \quad \text{and} \quad \mathcal{V}(\delta) \subseteq \mathcal{V}(\beta \wedge \beta').$$

Proposition 7.5 *Let ϕ be a formula that is built from atomic formulae with conjunction and existential quantification. Then one can compute a formula δ that is either prime or $\bot$ such that $\phi \models\!\!\!|_{FT}\ \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\phi)$.*

The **closure of a prime formula** $\exists X\gamma$ is defined as follows:

$$\begin{aligned}[\exists X\gamma] \;\; := \;\; \{\, \pi \in [\gamma] \mid \mathcal{V}(\pi) \cap X = \emptyset \;\text{ or }\; \pi = x\varepsilon x \\ \text{or }\; \pi = x\varepsilon{\downarrow}x\varepsilon \,\}.\end{aligned}$$

The **proper closure of a prime formula** β is defined as follows:

$$[\beta]^\star \;\; := \;\; \{\pi \in [\beta] \mid \pi \text{ is a proper path constraint}\}.$$

Proposition 7.6 *If β is a prime formula and $\pi \in [\beta]$, then $\beta \models \pi$ (and hence $\neg\pi \models \neg\beta$).*

We now know that the closure $[\beta]$, taken as an infinite conjunction, is entailed by β. We are going to show that, conversely, β is entailed by certain finite subsets of its closure $[\beta]$.

An **access function** for a prime formula $\beta = \exists X\gamma$ is a function that maps every $x \in \mathcal{V}(\gamma) - X$ to the rooted path $x\varepsilon$, and every $x \in X$ to a rooted path $x'p$ such that $x'px \in [\gamma]$ and $x' \notin X$. Note that every prime formula has at least one access function, and that the access function of a prime formula is injective on $\mathcal{V}(\gamma)$ (follows from Proposition 5.3 (5)).

The **projection** of a prime formula $\beta = \exists X\gamma$ with respect to an access function @ for β is the conjunction of the following proper path constraints:

$$\begin{aligned}&\{x\varepsilon{\downarrow}y\varepsilon \mid x \doteq y \in \gamma\} \cup \\ &\{Ax'p \mid Ax \in \gamma,\; x'p = @x\} \cup \\ &\{x'pf{\downarrow}y'q \mid xfy \in \gamma,\; x'p = @x,\; y'q = @y\}.\end{aligned}$$

Obviously, one can compute for every prime formula an access function and hence a projection. Furthermore, if λ is a projection of a prime formula β, then λ taken as a set is a finite subset of the closure $[\beta]$.

Proposition 7.7 *Let λ be a projection of a prime formula β. Then $\lambda \subseteq [\beta]^\star$ and $\lambda \models\!\!\!|_{FT}\ \beta$.*

As a consequence of this proposition one can compute for every prime formula an equivalent quantifier-free conjunction of proper path constraints.

We close this section with a few propositions stating interesting properties of closures of prime formulae. These propositions will not be used in the proofs to come.

Proposition 7.8 *If β is a prime formula, then $\beta \models\!\!\!|_{FT}\ [\beta]^\star$.*

Proposition 7.9 *If β is a prime formula, and π is a proper path constraint, then*

$$\pi \in [\beta]^\star \quad \Longleftrightarrow \quad \beta \models_{FT} \pi.$$

Proposition 7.10 *Let β, β' be prime formulae. Then $\beta \models_{FT} \beta' \iff [\beta]^\star \supseteq [\beta']^\star$.*

Proposition 7.11 *Let β, β' be prime formulae, and let λ' be a projection of β'. Then $\beta \models_{FT} \beta' \iff [\beta]^\star \supseteq \lambda'$.*

Proposition 7.11 gives us a decision procedure for "$\beta \models_{FT} \beta'$" since membership in $[\beta]^\star$ is decidable, λ' is finite, and λ' can be computed from β'.

8 Quantifier Elimination

In this section we show that our prime formulae satisfy the requirements (5) and (6) of Lemma 4.1 and thus obtain the completeness of *FT*. We start with the definition of the central notion of a joker.

A **rooted path** xp consists of a variable x and a path p. A rooted path xp is called **unfree** in a prime formula β if

$$\exists \text{ prefix } p' \text{ of } p \;\exists\, yq\!: \;\; x \neq y \text{ and } xp'{\downarrow}yq \in [\beta].$$

A rooted path is called **free** in a prime formula β if it is not unfree in β.

Proposition 8.1 *Let $\beta = \exists X\gamma$ be a prime formula. Then:*

1. *if xp is free in β, then x does not occur in the normalizer of γ*
2. *if $x \notin \mathcal{V}(\beta)$, then xp is free in β for every path p.*

A proper path constraint π is called an **x-joker** for a prime formula β if $\pi \notin [\beta]$ and one of the following conditions is satisfied:

1. $\pi = Axp$ and xp is free in β
2. $\pi = xp{\downarrow}yq$ and xp is free in β
3. $\pi = yp{\downarrow}xq$ and xq is free in β.

Proposition 8.2 *It is decidable whether a rooted path is free in a prime formula, and whether a path constraint is an x-joker for a prime formula.*

Lemma 8.3 *Let β be a prime formula, x be a variable, π be a proper path constraint that is not an x-joker for β, $\mathcal{A}$ be a model of FT, $\mathcal{A}, \alpha \models \beta$, $\mathcal{A}, \alpha' \models \beta$, and α' be an x-update of α. Then $\mathcal{A}, \alpha \models \pi$ if and only if $\mathcal{A}, \alpha' \models \pi$.*

Lemma 8.4 *Let β be a prime formula and $\pi_1, \ldots, \pi_n$ be x-jokers for β. Then*

$$\exists x \beta \quad \models_{FT} \quad \exists x(\beta \wedge \bigwedge_{i=1}^{n} \neg \pi_i).$$

The proof of this lemma uses the third axiom scheme, the existence of infinitely many features, and the existence of infinitely many sorts.

Lemma 8.5 *Let β, β' be prime formulae and α be a valuation into a model $\mathcal{A}$ of FT such that*

$$\mathcal{A}, \alpha \models \exists x(\beta \wedge \beta') \quad \textit{and} \quad \mathcal{A}, \alpha \models \exists x(\beta \wedge \neg \beta').$$

Then every projection of β' contains an x-joker for β.

Lemma 8.6 *If β, $\beta_1, \ldots, \beta_n$ are prime formulae, then*

$$\exists x(\beta \wedge \bigwedge_{i=1}^{n} \neg \beta_i) \quad \models\mid_{FT} \quad \bigwedge_{i=1}^{n} \exists x(\beta \wedge \neg \beta_i).$$

Lemma 8.7 *For every two prime formulae β, β' and every variable x one can compute a Boolean combination δ of prime formulae such that*

$$\exists x(\beta \wedge \neg \beta') \models\mid_{FT} \delta \quad \text{and} \quad \mathcal{V}(\delta) \subseteq \mathcal{V}(\exists x(\beta \wedge \neg \beta')).$$

Theorem 8.8 *For every formula ϕ one can compute a Boolean combination δ of prime formulae such that $\phi \models\mid_{FT} \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\beta)$.*

Corollary 8.9 *FT is a complete and decidable theory.*

References

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

[2] H. Aït-Kaci and R. Nasr. LOGIN: A logic programming language with built-in inheritance. *The Journal of Logic Programming*, 3:185–215, 1986.

[3] H. Aït-Kaci and A. Podelski. Towards a meaning of LIFE. In *Proc. of the PLILP'91*, Springer LNCS vol. 528, pages 255–274. Springer-Verlag, 1991.

[4] H. Aït-Kaci, A. Podelski, and G. Smolka. A feature-based constraint system for logic programming with entailment. In *Fifth Generation Computer Systems 1992*, pages 1012–1021, Tokyo, Japan, June 1992. Institute for New Generation Computer Technology.

[5] F. Baader, H.-J. Bürckert, B. Nebel, W. Nutt, and G. Smolka. On the expressivity of feature logics with negation, functional uncertainty, and sort equations. Research Report RR-91-01, German Research Center for Artificial Intelligence (DFKI), January 1991.

[6] R. Backofen and G. Smolka. A complete and recursive feature theory. Research Report RR-92-30, German Research Center for Artificial Intelligence (DFKI), July 1992.

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

[8] A. Colmerauer. Equations and inequations on finite and infinite trees. In *Proceedings of the 2nd International Conference on Fifth Generation Computer Systems*, pages 85–99, 1984.

[9] M. Johnson. *Attribute-Value Logic and the Theory of Grammar*. CSLI Lecture Notes 16. Center for the Study of Language and Information, Stanford University, CA, 1988.

[10] M. Johnson. Logic and feature structures. In *Proceedings of IJCAI-91*, Sydney, Australia, 1991.

[11] J.-L. Lassez, M. Maher, and K. Marriot. Unification revisited. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann, Los Altos, CA, 1988.

[12] M. J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite trees. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science*, pages 348–457, Edinburgh, Scotland, July 1988.

[13] W. C. Rounds and R. T. Kasper. A complete logical calculus for record structures representing linguistic information. In *Proceedings of the 1st IEEE Symposium on Logic in Computer Science*, pages 38–43, Boston, MA, 1986.

[14] S. M. Shieber. *An Introduction to Unification-Based Approaches to Grammar*, volume 4 of *CSLI Lecture Notes*. Center for the Study of Language and Information, Stanford University, CA, 1986.

[15] G. Smolka. Feature constraint logics for unification grammars. *Journal of Logic Programming*, 12:51–87, 1992.

[16] G. Smolka, M. Henz, and J. Würtz. Object-oriented concurrent constraint programming in oz. Research Report RR-93-16, German Research Center for Artificial Intelligence (DFKI), Apr. 1993.

[17] G. Smolka and R. Treinen. Records for logic programming. In *Proceedings of the 1992 Joint International Conference and Symposium on Logic Programming*, pages 240–254, Washington, DC, 1992.