Feature Constraints with First-Class Features

Ralf Treinen*

German Research Center for Artificial Intelligence (DFKI), Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany, email: treinen@dfki.uni-sb.de

Abstract. Feature Constraint Systems have been proposed as a logical data structure for constraint (logic) programming. They provide a record-like view to trees by identifying subtrees by keyword rather than by position. Their atomic constraints are finer grained than in the constructor-based approach. The recently proposed *CFT* [15] in fact generalizes the rational tree system of Prolog II.
We propose a new feature constraint system *EF* which extends *CFT* by considering features as first class values. As a consequence, *EF* contains constraints like $x[v]w$ where v is a variable ranging over features, while *CFT* restricts v to be a fixed feature symbol.
We show that the satisfiability of conjunctions of atomic *EF*-constraints is NP-complete. Satisfiability of quantifier-free *EF*-constraints is shown to be decidable, while the $\exists^*\forall^*\exists^*$ fragment of the first order theory is undecidable.

1 Introduction

Feature constraints provide records as logical data structure for constraint (logic) programming. Their origins are the feature descriptions from computational linguistics (see [13] for references) and Aït-Kacis's ψ-terms [1] which have been employed in the logic programming language Login [2]. Smolka [13] gives a unified logical view of most earlier feature formalisms and presents an expressive feature logic.

The predicate logic view to feature constraints, which has been pioneered by [13], laid the ground for the development of the constraint systems *FT* [3, 5] and *CFT* [15]. The latter constraint system subsumes Colmerauer's classical rational tree constraint system [6], but provides for finer grained and more expressive constraints. An efficient implementation of tests for satisfiability and entailment in *CFT* has been given in [16]. In fact, satisfiability of *CFT*-constraints can be tested in at most quadratic time, and for a mildly restricted case in quasi-linear time. *CFT* is the theoretical base for the constraint system of Oz [14].

CFT's standard model consists of so-called *feature trees*, that is possibly infinite trees where the nodes are labeled with *label symbols* and the edges are labeled with *feature symbols*. The labels of the edges departing from a node,

* Supported by the Bundesminister für Forschung und Technologie (contract ITW 9105), the Esprit Basic Research Project ACCLAIM (contract EP 7195) and the Esprit Working Group CCL (contract EP 6028).

In Andrzej M. Borzyszkowski and Stefan Sokołowski, eds., Mathematical Foundations of Computer Science, Gdańsk, Poland, 30 August–3 September, LNCS vol. 711, pp 734–743

called the *features of* that node, are pairwise distinct. The atomic constraints of CFT are equations, *label constraints* Ax ("x has label A"), *feature constraints* $x[f]y$ ("y is a child of x via feature f") and *arity constraints* $x\{f_1, \dots, f_n\}$ ("x has exactly the features $f_1, \dots, f_n$"). A rational tree constraint $x \doteq K(y_1, \dots, y_n)$ can now be expressed in CFT as

$$Kx \wedge x\{1, \dots, n\} \wedge x[1]y_1 \wedge \dots x[n]y_n \ .$$

Note that in CFT we can express the fact that x has the child y at feature f by $x[f]y$, which is inconvenient to express in the rational tree constraint system if the signature is finite and impossible if the signature is infinite. CFT's atomic constraints are finer grained and hence lead to an elegant and powerful tree constraint system for logic programming. A complete axiomatization of FT (that is CFT without arity constraints) has been given in [5], while the question of complete axiomatizability of CFT is still open.

In this paper we are concerned with an extension of CFT which is desirable for logic programming and which also leads to a more basic view of feature constraints. Instead of considering a family of binary feature constraints $x[f]y$, indexed by feature symbols f, we consider features to be first-class values and introduce *one* ternary variable feature constraint $x[v]y$, where v ranges over a distinguished sort of feature symbols. The interesting point is that we now get quantification over features for free from predicate logic, which leads to a dramatic gain in expressiveness. In contrast to [15], we can now for instance express the fact that y is a direct subtree of x by $\exists v\, x[v]y$, and an arity constraint $x\{f_1, \dots, f_n\}$ can now be seen as a mere abbreviation for

$$\forall v(\exists y\ x[v]y \leftrightarrow \bigvee_{i=1}^{n} v \doteq f_i) \ .$$

It turns out that in certain (intended) models the subtree relation between feature trees is expressible (see Section 7).

Feature descriptions with first class features have already been considered by Johnson [10]. In contrast to our work, Johnson was not concerned with quantifiers or with arity constraints.

After fixing the constraint system EF in Section 2, we will address the problem of satisfiability of positive constraints, that is of conjunctions of atomic constraints, in Section 4. Although redundant for the first order theory, we keep the arity constraint since adding arity constraints to the fragment of *quantifier-free* formulae still leads to a gain in expressiveness. For the same reason, we add a constraint $x[t]\!\uparrow$ standing for $\forall y(\neg x[t]y)$. We present a nondeterministic algorithm with polynomial complexity. Note that, as an easy corollary of [4], satisfiability of constraints *without* arity constraints is decidable using the algorithm of [15]. In Section 5 we show the problem to be NP-hard, which results in positive constraint satisfaction to be NP-complete.

Section 6 extends the solvability result to conjunctions of positive and negative atomic constraints, which yields decidability of the $\exists^*$ fragment of the first order theory. We finally show in Section 7 that the canonical models of EF have undecidable first order theories. The proofs missing in this paper can be found in [18].

2 F- and EF-Constraints

In this section we define the constraint system F and its extension EF. We assume a fixed set FEA of *feature symbols*, ranged over by f, g, and a fixed set LAB of *label symbols*[2], ranged over by A, B. The language of the constraint system F has two sorts, *feat* and *tree*, and an infinite supply of variables of each sort. We use letters x, y, z to denote *tree* variables and letters v, w for *feat* variables. The constant symbols of sort *feat* are the elements of FEA, there are no further constant or function symbols. The predicate symbols are, besides equality $\doteq$:

- A unary predicate symbol L of type *tree* for each $L \in LAB$. We use prefix notation Lx for the so-called *label constraints*.
- A ternary predicate symbol $\cdot[\cdot]\cdot$ of sort *tree* $\times$ *feat* $\times$ *tree*. We use mixfix notation $x[v]y$ for the so-called *feature constraints*.

A *constraint* μ is a possibly empty conjunction of literals, where we identify as usual a finite multiset with the conjunction of its members. A constraint ϕ is a *clause* if ϕ contains no equation. A constraint γ (resp. clause ψ) which contains no negated atom is called *positive*. We write $\tilde{\forall}\mu$ $(\tilde{\exists}\mu)$ for the universal (existential) closure of μ.

Now we describe three F-structures which are candidates for being "natural" models. All three structures are in fact models of the axiom system presented in Section 3.

The structure $\mathfrak{I}$ consists of all finite and infinite feature trees (see Section 1). $\mathfrak{F}$ consists of all finitely branching (but probably infinite) feature trees. The structure $\mathfrak{R}$ consists of all finitely branching (probably infinite) feature trees which have only finitely many subtrees (these trees are called *rational*). In the three models, a constraint Ax holds if and only if the root of x is labeled with label symbol A, and $x[v]y$ holds if y is a child of x via feature v. Hence, we have the substructure relationship $\mathfrak{R} \subset \mathfrak{F} \subset \mathfrak{I}$.

A first indication of the great expressivity of F is the fact that these models are not elementarily equivalent, in contrast to the situation of constructor trees where the model of all infinite trees and the model of rational trees cannot be distinguished by a single first order logic sentence [12]. This can be seen as follows: We take $y \prec x$ as an abbreviation for $\exists v\; x[v]y$ (read: y is a child of x). Now the formula

$$fs(x) \quad := \quad \exists y\Big(x \prec y \wedge \forall z_1, z_2\,(z_2 \prec z_1 \wedge z_1 \prec y \rightarrow z_2 \prec y)\Big)$$

[2] Labels have been called *sorts* in earlier publications on feature constraint systems (e.g. [15]). We changed this name here in order to avoid confusion with the sorts of predicate logic.

expresses in some sense that x can be "flattened". The reader will easily verify that $\forall x\, fs(x)$ is valid in $\mathfrak{R}$, but neither in $\mathfrak{F}$ nor in $\mathfrak{I}$. In fact, $fs(x)$ holds in $\mathfrak{F}$ iff x is rational, and holds in $\mathfrak{I}$ iff x has at most $cardinality(FEA)$ many subtrees.

The following formula expresses that there is a feature tree x with an infinite sequence of children which have themselves a strictly increasing number of children. This formula holds in $\mathfrak{I}$ but not in $\mathfrak{F}$.

$$\begin{aligned}\exists x(\exists y\, y \prec x \wedge \forall y_1(y_1 \prec x \rightarrow \exists y_2(y_2 \prec x \wedge \forall z(z \prec y_1 \rightarrow z \prec y_2)\wedge \\ \exists z(\neg z \prec y_1 \wedge z \prec y_2))))\end{aligned}$$

Note that the formula $\exists x \forall y\, y \prec x$ does not hold in $\mathfrak{I}$, since the cardinality of $\mathfrak{I}$ is strictly greater than the cardinality of FEA.

We now extend the constraint system F to the system EF by adding the following predicates symbols:

- A unary predicate symbol F for each finite subset F of FEA. We use postfix notation xF for the so-called *arity constraints*.
- A binary predicate symbol $\cdot[\cdot]\uparrow$ of sort $tree \times feat$.

The additional predicate symbols of EF have explicit definitions in F:

$$\begin{aligned} x\{f_1,\ldots,f_n\} &\leftrightarrow \forall v(\exists y\, x[v]y \leftrightarrow \textstyle\bigvee_{i=1}^n v \doteq f_i) && x \neq y \\ x[v]\uparrow &\leftrightarrow \neg\exists y\, x[v]y && x \neq y \end{aligned}$$

Hence, we will consider $\mathfrak{I}$, $\mathfrak{F}$ and $\mathfrak{R}$ to be EF-structures as well as F-structures.

3 An Axiomatization

In this section we give a system EF of axioms which describe the "intended" structures of EF. We begin with five straightforward axiom schemes.

(D) $\neg f \doteq g$	$f, g \in FEA$; $f \neq g$
(L) $\forall x\, \neg(Ax \wedge Bx)$	$A \neq B$
(F) $\forall x, y, z, v(x[v]y \wedge x[v]z \rightarrow y \doteq z)$	
(A) $\forall x(x\{f_1,\ldots,f_n\} \leftrightarrow \forall v(\exists y\, x[v]y \leftrightarrow \bigvee_{i=1}^n v \doteq f_i))$	$x \neq y$
(U) $\forall x, v(x[v]\uparrow \leftrightarrow \neg\exists y\, x[v]y)$	$x \neq y$

One possible model of these axioms interprets all relation symbols as the empty relation. We wish to exclude those models by requiring that certain clauses like

$$Ax \wedge x\{f,g\} \wedge x[f]x \wedge x[g]y \wedge y[v]\uparrow \wedge y[w]z \tag{1}$$

have a solution. In order to state the axiom scheme for the satisfiability of clauses we need some more definitions. A *solved positive clause* is a positive clause ϕ which satisfies:

1. if $Ax \wedge Bx \subseteq \phi$ then $A = B$;
2. if $x[t]y \wedge x[t]z \subseteq \phi$ then $y = z$;
3. if $x[t]y \wedge x[s]\uparrow \subseteq \phi$ then $t \neq s$;
4. if $xF \wedge xG \subseteq \phi$ then $F = G$;
5. if $xF \wedge x[t]y \subseteq \phi$ then $t \in F$;
6. if $xF \wedge x[t]\uparrow \subseteq \phi$ then t is a variable.

For example (1) is a solved positive clause. A variable x is *constrained* in a clause ϕ if ϕ contains a constraint of the form Ax, xF, $x[t]y$ or $x[t]\uparrow$. We say that ϕ *constrains x at t* if ϕ contains $x[t]y$, $x[t]\uparrow$ or xF with $t \in F$. We use $\mathcal{C}\phi$ to denote the set of constrained variables of ϕ. For a clause ϕ we define

$$\Delta\phi := \{\neg(v \doteq t) \mid \phi \text{ constrains some } x \text{ at } v \text{ and at } t,\ v \neq t,\ v \text{ is a variable}\}$$

Now the axiom scheme stating satisfiability of solved positive clauses reads

$$\boxed{(Con)\quad \tilde{\forall}(\Delta\phi \rightarrow \exists\mathcal{C}\phi\ \phi) \qquad \text{if } \phi \text{ is a solved positive clause}}$$

Taking the solved positive clause (1), we obtain the axiom

$$\forall z, v, w\Big(\neg v \doteq w \rightarrow \exists x, y(Ax \wedge x\{f,g\} \wedge x[f]x \wedge x[g]y \wedge y[v]\uparrow \wedge y[w]z)\Big)$$

Note that $\neg v \doteq w$ is satisfiable in every model of the axioms, hence (1) is satisfiable in every model of the axioms.

Taking the clause (1) we know that in the three structures of Section 2 the solution to x is unique if y and z are fixed. This is what the last axiom scheme expresses. We write $\hat{\exists}x\Psi$ (read: "there is at most one x such that Ψ") as an abbreviation for

$$\forall y_1, y_2\Big(\Psi[x \mapsto y_1] \wedge \Psi[x \mapsto y_2] \rightarrow y_1 \doteq y_2\Big)$$

and accordingly for sets of variables. This quantifier has the important property that for all formulas Φ, Ψ

$$\tilde{\forall}\exists X(\Phi \wedge \Psi) \wedge \tilde{\forall}\hat{\exists}X\Psi \models \tilde{\forall}(\Psi \rightarrow \Phi)\ .$$

A variable x is *determined* in a clause ϕ if ϕ contains a label constraint Ax, an arity constraint xF and for each $f \in F$ a feature constraint of the form $x[f]y$. We use $\mathcal{D}\phi$ to denote the set of all determined variables in ϕ. If for instance ϕ is the clause from (1) then $\mathcal{D}\phi = \{x\}$. The axiom scheme on the uniqueness of solutions reads

$$\boxed{(Det)\quad \tilde{\forall}\hat{\exists}\mathcal{D}\phi\ \phi \qquad \text{if } \phi \text{ is a solved positive clause}}$$

Taking for ϕ the clause from (1) we get the following instance of (Det):

$$\forall y, z, v, w\ \hat{\exists}x(Ax \wedge x\{f,g\} \wedge x[f]x \wedge x[g]y \wedge y[v]\uparrow \wedge y[w]z)$$

Lemma 1. *The structures $\mathfrak{I}$, $\mathfrak{F}$ and $\mathfrak{R}$ are models of EF.*

4 Satisfiability of Positive EF-Constraints is in NP

In this section, we present a nondeterministic algorithm which decides satisfiability of positive constraints in the models of *EF*. The algorithm consists of a rewrite relation $\Rightarrow_P$ such that in all models of *EF* the constraint γ is equivalent to the disjunction of its $\Rightarrow_P$-normal forms. Every $\Rightarrow_P$-irreducible form is either $\bot$ or of the form $\delta \wedge \phi$, where δ is an idempotent substitution, ϕ is solved positive constraint and $\delta\phi = \phi$. In the following, χ ranges over variables of sort *feat* or *tree*, and t ranges over arbitrary terms (that is, variables or feature symbols). The first set of rules ensures that the equational part of an irreducible constraint (if different from $\bot$) is an idempotent substitution which is applied to the remainder.

(P1)	$\dfrac{t \doteq t \wedge \phi}{\phi}$		*(P2)*	$\dfrac{\chi \doteq t \wedge \phi}{\chi \doteq t \wedge \phi[\chi \leftarrow t]}$	$\chi \in \mathcal{V}\phi, \chi \neq t$
(P3)	$\dfrac{f \doteq v \wedge \phi}{v \doteq f \wedge \phi}$		*(P4)*	$\dfrac{f \doteq g \wedge \phi}{\bot}$	$f \neq g$

The rules of the second set coincide with the conditions of the definition of solved positive clauses. *(P5)* guarantees condition 1, *(P6)* condition 2, *(P7)* condition 3, *(P8)* condition 4, *(P9)* and *(P10)* condition 5 and *(P11)* and *(P12)* condition 6.

(P5)	$\dfrac{Ax \wedge Bx \wedge \phi}{\bot}$	$A \neq B$	*(P6)*	$\dfrac{x[t]y \wedge x[t]z \wedge \phi}{y \doteq z \wedge x[t]z \wedge \phi}$	
(P7)	$\dfrac{x[t]y \wedge x[t]\uparrow \wedge \phi}{\bot}$				
(P8)	$\dfrac{xF \wedge xG \wedge \phi}{\bot}$	$F \neq G$	*(P9)*	$\dfrac{xF \wedge x[f]y \wedge \phi}{\bot}$	$f \notin F$
(P11)	$\dfrac{xF \wedge x[f]\uparrow \wedge \phi}{\bot}$	$f \in F$	*(P12)*	$\dfrac{xF \wedge x[f]\uparrow \wedge \phi}{xF \wedge \phi}$	$f \notin F$
(P10)	$\dfrac{xF \wedge x[v]y \wedge \phi}{v \doteq f \wedge xF \wedge x[f]y \wedge \phi[v \leftarrow f]}$	$f \in F$			

Note that only the rule (P10) is indeterministic by allowing for an arbitrary choice of v among the members of F. The rewriting relation $\Rightarrow_P$ defined by the above rewrite system is terminating, as the reader easily verifies. A theory T is *satisfaction complete* [9] if for every positive constraint γ either $T \models \tilde{\exists}\gamma$ or $T \models \neg\tilde{\exists}\gamma$ holds. Hence we obtain

Theorem 2. *EF is satisfaction complete. A positive constraint γ is satisfiable in EF iff there is an $\Rightarrow_P$-irreducible form of γ different from $\bot$.*

Note that the length of a rewriting sequence starting from γ is polynomial in the size of γ.

Corollary 3. *Satisfiability of positive EF-constraints is decidable in NP time.*

5 Satisfiability of Positive *EF*-Constraints is NP-hard

We employ a reduction of the Minimum Cover Problem [8] to the satisfiability problem of positive *EF*-constraints. Since the Minimum Cover Problem is known to be NP-complete and since our reduction is polynomial, this will prove satisfiability to be NP-hard.

The Minimum Cover Problem reads as follows:

> Given a collection $S_1, \ldots, S_n$ of finite sets and a natural number $k \leq n$.
> Is there a subset $I \subseteq \{1, \ldots, n\}$ with $\mathit{cardinality}(I) \leq k$ such that
>
> $$\bigcup_{j \in I} S_j = \bigcup_{i=1}^{n} S_i \quad ?$$

Let an instance $(S_1, \ldots, S_n; k)$ of the Minimum Cover Problem be given. We define $U := \bigcup_{i=1}^{n} S_i$ and for any $u \in U$: $\delta_u := \{j \mid i \in S_j\}$. Without loss of generality, we assume that $1, \ldots, n \in \mathit{FEA}$. We construct a constraint $\Psi := \Psi_1 \wedge \Psi_2 \wedge \Psi_3$ that is satisfiable if and only if the instance of the minimum cover problem has a solution. We use variables x_u $(u \in U)$ for the elements of U and variables $z_1, \ldots, z_n$ to denote the sets $S_1, \ldots, S_n$. The first formula Ψ_1 requires that z_j is a direct subtree of x_u if and only if $u \in S_j$:

$$\bigwedge_{u \in U} x_u \delta_u \quad \wedge \quad \bigwedge_{u \in U} \bigwedge_{j \in \delta_i} x_u[j] z_j \ .$$

The choice of an appropriate set I is now expressed as an assignment of labels to the variables z_i. The idea is to assign the label IN to the variable z_i if $i \in I$, and OUT otherwise. The formula Ψ_2 expresses the fact that at least $n - k$ of the z_i have the label IN . It is defined as

$$\exists x \left(x\{1, \ldots, n\} \quad \wedge \quad \bigwedge_{i=1}^{n} x[i] z_i \quad \wedge \quad \bigwedge_{i=1}^{n-k} \exists v, y (x[v] y \wedge \textsc{out}\ y \wedge y\{i\}) \right) .$$

The arity constraints for y forces for each i a different choice of y. The formula Ψ_3 expresses the fact that each x_i has an immediate subtree with label IN , which according to the definition of Ψ_1 must be one of the z_i.

$$\bigwedge_{i \in U} \exists v, z (x_i[v] z \wedge \textsc{in}\ z) \ .$$

The length of the formula Ψ is in fact linear in the size of the representation of the minimum cover problem. Hence, together with Corollary 3 we obtain

Theorem 4. *Satisfiability of positive EF-constraints is NP-complete.*

6 Satisfiability of Constraints

In this section, we extend the results of Section 4 to conjunction of positive and negative atomic constraints. This extension is complicated by the fact that the independence of constraints[6] does not hold in our case, in contrast to the constraint system *CFT* of [15]. A counter example to the Independence Property[3] is

$$x\{f,g\} \wedge\ x[f]y \wedge x[g]z \wedge Ay \wedge Bz \wedge x[v]x' \models_{EF} Ax' \vee Bx'$$

but the left hand side does not imply any of the two disjuncts alone.

We define a rewrite system $\Rightarrow_N$ by the rules of Section 4 plus the following ones:

(N1)	$\dfrac{Ax \wedge \neg Ax \wedge \phi}{\perp}$		*(N2)*	$\dfrac{Ax \wedge \neg Bx \wedge \phi}{Ax \wedge \phi}$	$A \neq B$
(N3)	$\dfrac{xF \wedge \neg xF \wedge \phi}{\perp}$		*(N4)*	$\dfrac{xF \wedge \neg xG \wedge \phi}{xF \wedge \phi}$	$F \neq G$
(N5)	$\dfrac{\neg x[t]y \wedge \phi}{x[t]\uparrow \wedge \phi}$		*(N6)*	$\dfrac{\neg x[t]y \wedge \phi}{\exists z(x[t]z \wedge \neg z \doteq y) \wedge \phi}$	z new
(N7)	$\dfrac{\neg x[t]\uparrow \wedge \phi}{\exists z\ x[t]z \wedge \phi}$	z new			

A clause ψ is a *solved clause*, if its positive part is a solved positive clause, if ψ does not contain constraints of the form $\neg x[t]\uparrow$ or $\neg x[t]y$, and if ψ contains a negative label (resp. arity) constraint for x, then it does not contain a positive label (resp. arity) constraint for x. Note that every $\Rightarrow_N$-normal form of a constraint is either $\perp$ or of the form $\delta \wedge \phi \wedge \tau$, where δ is an idempotent substitution with $\delta(\phi \wedge \tau) = \phi \wedge \tau$ (hence, δ is not relevant for satisfiability), ϕ is a solved clause and τ is a conjunction of negated equations.

Lemma 5. *Every solved clause is satisfiable in every model of EF.*

We still need a criterion whether some solved clause together with some inequations is satisfiable. We say that a set η of equations is *complete wrt.* ψ, if $\eta \models x \doteq y$ and $x[t]x' \wedge y[t]y' \subseteq \psi$ imply that $\eta \models x' \doteq y'$. Given ψ and η, it is easy to compute a set η' of equations such that $\psi \wedge \eta \models\!\!\!\dashv_{EF}\ \psi \wedge \eta'$. ($\eta'$ can be seen as the equational part of the congruence closure of $\eta \wedge \psi$, see [15].)

Lemma 6. *Let ψ be a solved clause, and η be complete wrt. ψ. If $\psi \wedge \eta$ is satisfiable in EF and if $\mathcal{V}\eta \subseteq \mathcal{D}\psi$, then $\psi \models_{EF} \eta$.*

A clause ψ is called *saturated* if $xF \in \psi$ and $f \in F$ imply that $x[f]y \in \psi$ for some y. Every solved clause can be transformed into an equivalent saturated clause, with existentially quantifiers for the new variables.

[3] A constraint system is *independent* [6] if: $\phi \wedge \neg\phi_1 \wedge \ldots \wedge \neg\phi_2$ is satisfiable iff $\phi \wedge \neg\phi_i$ is satisfiable for every i This equivalent to: $\phi \models \phi_1 \vee \ldots \phi_n$ iff $\phi \models \phi_i$ for some i.

Lemma 7. *Let ψ be a solved and saturated clause and let $\eta_1, \ldots, \eta_n$ be conjunctions of equations such that for every i: $\mathcal{V}\eta_i \not\subseteq \mathcal{D}\psi$. Then.*

$$\models_{EF} \tilde{\exists}\Big(\psi \wedge \neg\eta_1 \wedge \ldots \wedge \neg\eta_n\Big) .$$

Theorem 8. *EF is complete for Σ_1, that is for every quantifier-free formula w, either $\models_{EF} \tilde{\exists} w$ or $\models_{EF} \neg\tilde{\exists} w$.*

It is decidable whether for an quantifier-free w: $\models_{EF} \tilde{\exists} w$.

Proof. We transform a given quantifier-free formula w into disjunctive normal form and test every disjunction (i.e., constraint) μ for satisfiability as follows: We compute all $\Rightarrow_N$-normal forms of μ. (Note that $\Rightarrow_N$ is terminating.) μ is satisfiable iff one of its normal forms ν is. If $\nu = \bot$, then ν is of course not satisfiable in any model of *EF*. Otherwise we extend ν to an (modulo new variables) equivalent saturated clause ν'. If there is an inequation $\neg x \doteq y$ in ν' such that all variables of the completion of $x \doteq y$ wrt. ν' are determined in ν', then ν' is by Lemma 5 and Lemma 6 not satisfiable in any model of *EF*. Otherwise, by Lemma 7, ν' is satisfiable in every model of *EF*.

7 Undecidability of the First Order Theory

In this section we will just give the key argument why the first order theories of the $\mathfrak{I}$, $\mathfrak{F}$ and $\mathfrak{R}$ are undecidable. For complete proofs we refer to [18]. Venkataraman [19] has shown that the first order theory of constructor trees with the subterm relation is undecidable (see also [17]). Since feature constraints are in fact even more expressive than constructor tree constraints, it suffices to show that we can express the subterm relation between feature trees as a first order logic formula. To be more specific, we don't have to code the subterm relation in its full generality. It is sufficient that for each structure under consideration there is a set *Rep* of feature trees that contains at least the rational feature trees such that $s(x, y)$ holds iff $x \in Rep$ und y is a subtree of x:

$$\begin{aligned} s(x,y) := \exists z \quad & \big(x \prec z \wedge \forall x_1, x_2\,(x_2 \prec x_1 \wedge x_1 \prec z \rightarrow x_2 \prec z)\big) \wedge \\ \forall z\ & \big(\big(x \prec z \wedge \forall x_1, x_2\,(x_2 \prec x_1 \wedge x_1 \prec z \rightarrow x_2 \prec z)\big) \\ & \rightarrow y \prec z\big) \end{aligned}$$

With a direct coding of the Post Correspondence Problem into the three theories along the technique given in [17] we can show (see [18]):

Theorem 9. *The $\exists^*\forall^*\exists^*$ fragment of the first order theories of the structures $\mathfrak{R}$, $\mathfrak{I}$ and $\mathfrak{F}$ are undecidable.*

I am grateful to Gert Smolka for discussions on an earlier version of this paper and to an anonymous referee for useful comments.

References

1. H. Aït-Kaci. An algebraic semantics approach to the effective resolution of type equations. *Theoretical Comput. Sci.*, 45:293–351, 1986.
2. H. Aït-Kaci and R. Nasr. LOGIN: A logic programming language with built-in inheritance. *Journal of Logic Programming*, 3:185–215, 1986.
3. H. Aït-Kaci, A. Podelski, and G. Smolka. A feature-based constraint system for logic programming with entailment. In *Int. Conf. on 5th Generation Computer Systems*, pages 1012–1021, 1992.
4. R. Backofen. On the decidability of functional uncertainty. In *Rewriting Techniques and Applications*, LNCS, 1993. Springer-Verlag.
5. R. Backofen and G. Smolka. A complete and recursive feature theory. In *Proc. of the 31 th ACL*, Columbus, Ohio, 1993. Complete version as DFKI Research Report RR-92-30.
6. A. Colmerauer. Equations and inequations on finite and infinite trees. In *2nd Int. Conf. on 5th Generation Computer Systems*, pages 85–99, 1984.
7. H. Comon. Unification et disunification. Théorie et applications, 1988. Doctoral Thesis, Institut National Polytechnique de Grenoble.
8. M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, 1979.
9. J. Jaffar and J.-L. Lassez. Constraint logic programming. In *14th POPL*, pages 111–119, Munich, Germany, Jan. 1987. ACM.
10. 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.
11. E. Kounalies, D. Lugiez, and L. Potier. A solution of the complement problem in associative-commutative theories. In *MFCS 1991*, LNAI, vol. 520, pages 287–297, Springer-Verlag.
12. M. J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite trees. In *Third LICS*, pages 348–357. 1988.
13. G. Smolka. Feature constraint logics for unification grammars. *Journal of Logic Programming*, 12:51–87, 1992.
14. G. Smolka, M. Henz, and J. Würtz. Object-oriented concurrent constraint programming in Oz. Research Report RR-93-16, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-W-6600 Saarbrücken, Germany, Apr. 1993.
15. G. Smolka and R. Treinen. Records for logic programming. In K. Apt, editor, *Proceedings of the Joint International Conference and Symposium on Logic Programming*, pages 240–254, 1992.
16. G. Smolka and R. Treinen. Records for logic programming. Research Report RR-92-23, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-W-6600 Saarbrücken, Germany, Aug. 1992.
17. R. Treinen. A new method for undecidability proofs of first order theories. *Journal of Symbolic Computation*, 14(5):437–457, Nov. 1992.
18. R. Treinen. On feature constraints with variable feature access. Research Report RR-93-21, Deutsches Forschungszentrum für Künstliche Intelligenz, Stuhlsatzenhausweg 3, D-W-6600 Saarbrücken, Germany, 1993.
19. K. N. Venkataraman. Decidability of the purely existential fragment of the theory of term algebra. *J. ACM*, 34(2):492–510, Apr. 1987.