How to Win a Game with Features

Rolf Backofen[*]
Institut für Informatik
Ludwig-Maximilian-Universität München
Oettingenstraße 67, D-80538 München, Germany
email: backofen@informatik.uni-muenchen.de
http://www.informatik.uni-muenchen.de/~backofen

Ralf Treinen[†]
Laboratoire de Recherche en Informatique (LRI)
Bât. 490, Université de Paris-Sud, F-91405 Orsay, France
email: treinen@lri.fr
http://www.lri.fr/~treinen

May 15, 1996

Abstract

We employ the model-theoretic method of Ehrenfeucht-Fraïssé Games to prove the completeness of the theory *CFT*, which has been introduced in [22] for describing rational trees in a language of selector functions. The comparison to other techniques used in this field shows that Ehrenfeucht–Fraïssé Games lead to simpler proofs.

1 Introduction

Trees are the prevailing data structure in symbolic computation since they provide for a mathematical model of hierarchically structured data. In the area of symbolic computation, trees come in two flavors: *constructor trees* and *feature trees* [4]. In both kinds of trees the nodes are decorated with so-called *labels*. In the case of constructor trees, the outgoing edges of a node are ordered, that is they can be seen as consecutively numbered. In case of feature trees, the outgoing edges of a node are unordered but decorated with different symbols, called *feature symbols*. Finite

[*] This work was done while the first author was with the German Research Center for Artificial Intelligence (DFKI), Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany

[†] Major parts of this work have been done while the second author was with DFKI.

To appear in: *Information and Computation*, special issue on the 1994 conference on Constraints in Computational Logics, edited by Jean-Pierre Jouannaud.

constructor trees are often identified with finite ground terms, and they may come with additional restrictions: There may be an *arity function* associating the number of outgoing edges with the label of a node, or they may be additional sort restrictions which are conveniently expressed with tree automata [6].

In this paper, we are not only interested in finite trees but also in rational trees [9], that is infinite trees with only finitely many different subtrees. *Rational trees* represent cyclic data structures (see the last example in Figure 1 where a cyclic graph is used to depict an infinite rational feature tree). Cyclic graphs could also be used to represent cyclic data structures, but in this case special care has to be taken to avoid multiple representations of the same data item, as it is for instance the case with different finite automata describing the same regular language. In fact, rational trees could be identified with equivalence classes (modulo renaming of states) of a certain kind of *minimal* finite automata. However, generalizing finite trees to rational trees leads to a simpler model than generalizing finite trees to arbitrary finite graphs. In the following, "trees" are always rational trees.

Furthermore, we are interested in the situation that there is an infinite supply of symbols. An instance of a symbolic computation system (e.g. a program) can use only a finite number of symbols as long as there are no operations generating new symbols (this is the case for the systems considered in this paper, see [23] for an approach incorporating operations on symbols). Hence, from a finite program the finite set of symbols used in this program could always be computed. However, a closed world assumption (the knowledge about a finite supply of symbols) is often not appropriate for symbolic computation systems since the set of data represented in a system may grow, and since in these systems incremental algorithms which deal with incomplete data specifications are of great importance. Hence, the formal theories described below will be based on infinite supplies of symbols.

Feature trees are more convenient to use then constructor trees since they allow to chose symbolic names for discriminating the outgoing edges of a node (i.e., they are nested *records*). In contrast, constructor trees can be seen as nested *arrays*, they require the user to keep track of what he had mind with the ith subtree of a tree. This difference might appear not very important at a first glance, since a compiler can easily translate a given record scheme into an array scheme. In fact, the real difference lies only in the respective description languages associated with the two kinds of trees. In particular, the description languages of feature trees allows to express properties of trees without fixing a record scheme. Record descriptions have a long history in knowledge representation and in particular in computational linguistics, see [21] for a survey.

The most important description language in the field of symbolic computation is *first-order logic*. An important paradigm in the field of symbolic computation, popularized by *Constraint Logic Programming* [15], is the use of (a restricted sub-language of) first-order logic in combination with non-declarative formalisms. In this context, the first-order formulae are usually called *constraints* since their role is to restrict the possible values of variables, which are shared among the constraints and the non-declarative formalism, by imposing some conditions. In this paper we use the term *constraint system* as synonym for a first-order structure. For the user of a symbolic computation system the most intuitive way to understand the constraint processing is to have this structure in mind, but for the system itself it is only the *theory* of the structure, that is the set of all its valid sentences, that counts. The constraint processing procedures like the tests for entailment and disentailment (see [22]) can be formulated as decision problems for fragments of the theory of a constraint system. In most of the cases it is not necessary to have a decision procedure for the complete theory. Decision procedures for complex formulae are however needed for deciding properties *of* constraint systems, see for instance the motivating example of [8].

When proving the decidability of the theory of a first-order structure one often shows the completeness of some *axiomatization* of the theory. A complete axiomatization of a theory T is a decidable subset of T such that every sentence of T can be derived from it. In almost all of the cases, a complete axiomatization is described by a finite set of syntactically simple formula schemata. A complete axiomatization T of the theory of a structure $\mathfrak{A}$ serves two purposes: First, by using any complete deduction method of first order logic we obtain a decision method for the theory of $\mathfrak{A}$, since for any sentence w, either w or its negation $\neg w$ is a consequence T which will eventually be detected if we run two deduction machines in parallel. Second, T describes all the structures which are elementarily equivalent to $\mathfrak{A}$, that is which have the same theory, since by the completeness of T a structure $\mathfrak{B}$ is elementarily equivalent to $\mathfrak{A}$ iff it is a model of T.

The constraint system RT [5] of rational constructor trees is parameterized by a finite or infinite functional (i.e., containing only function symbols) signature Σ. The universe consists of all rational constructor trees with labels from Σ, subject to arity restrictions. A function symbol f of arity n is interpreted as the function that maps trees $t_1, \ldots, t_n$ to the new tree with root labeled f and edges from the root to $t_1, \ldots, t_n$. Besides these functions, RT contains only the equality predicate $\doteq$. Complete axiomatizations of RT have been given independently in [7] for the case of a finite signature, and in [17] for both the case of a finite and an infinite signature.

The most basic feature tree constraint system is the system FT [1]. For given infinite sets of labels and features, its universe consists of the set of all rational feature trees with node and edge decorations taken from the respective sets. The only predicates are equality, a unary predicate Ax for every label symbol A, which holds if the root of x has the label A, and a binary predicate xfy for every feature f, which holds if there is an edge decorated with f from the root of x to the root of y. A complete axiomatization of FT has been given in [4].

A comparison of the expressive power of RT and FT makes no sense since their universes are different. We therefore fix (only for the purpose of a comparison of the systems) a functional signature Σ containing infinitely many functional symbols for every arity, define the set of labels to be Σ and choose the set of features to be the set of natural numbers. In this setup, the constructor trees are a proper subset of the feature trees (the edges of a constructor tree can be seen as consecutively numbered). Finally, we extend RT to a new system RT^+ which has as universe the set of all feature trees, and where the functions are defined as in RT but may take arbitrary feature trees as input.

The two constraint systems RT^+ and FT are not comparable in power. The FT-constraint Ax, where the arity of A is n, can be expressed in RT^+ by $\exists y_1, \ldots, y_n\, x \doteq A(y_1, \ldots, y_n)$. The FT-constraint xny, however, cannot be described in RT^+ since it requires an infinite disjunction:

$$\bigvee_{arity(A) \geq n} \exists y_1, \ldots, y_{arity(A)}\, x \doteq A(y_1, \ldots, y_{n-1}, y, y_{n+1}, \ldots, y_{arity(A)})$$

On the other hand, a RT^+-constraint like $x \doteq f(y_1, \ldots, y_n)$ can not be expressed in FT. Note that

$$Ax \wedge x1y_1 \wedge \ldots \wedge xny_n$$

is not sufficient since it allows x to have additional features. In fact [2] shows that there is no FT constraint denoting exactly one tree.

The constraint system CFT [22] extends FT by a unary predicate $x\{f_1, \ldots, f_n\}$ for every finite set $\{f_1, \ldots, f_n\}$ of features. Note that a RT^+-constraint $x \doteq f(y_1, \ldots, y_n$ can now be expressed in CFT by

$$Ax \wedge x1y_1 \wedge \ldots \wedge xny_n \wedge x\{1, \ldots, n\}$$

A axiomatization of CFT has been given in [22] and first proven complete in [3].

A feature constraint system F with first-class features, that is allowing quantification over features, has been investigated in [24]. F is a proper extension of CFT but has an undecidable theory. A proper extension FTX of CFT which permits only a limited quantification over features and which enjoys a decidable theory has however been presented in [23].

The above mentioned completeness results for axiomatizations of feature constraint structures have been obtained by quantifier elimination: The proofs for FT [4] and CFT [3] use a similar structure as [17], while similar ideas as in [7] have been used for FTX [23]. In this paper we give an alternative completeness proof for the axiomatization of CFT. Our completeness proof uses Fraïssé's theorem and its game-theoretic formulation due to Ehrenfeucht. This method employs an argument concerning chains of relations between elements in a structure. Feature logic is well suited for such an argument, since chains of relations are in a natural way expressed as so-called path constraints. Path constraints, like $x(f_1 \cdots f_n)y$, can be defined in FT by

$$x(f_1 \cdots f_n)y \Leftrightarrow \exists x_1, \ldots, x_{n-1}\, (xf_1x_1 \wedge x_1f_2x_2 \wedge \ldots \wedge x_{n-1}f_ny)$$

In the field of term rewriting systems (see [10] for a survey), the notion of an *occurrence* in a term is well established. In the context of feature logic, there is no need for introducing such a meta-notation, since we can use the path constraints which are an immediate offspring of the base language. In the context of finite constructor trees, Hodges [14] observes that the use of selector functions simplifies the completeness proof of an axiomatization. His completeness proof for an axiomatization of trees in the language of RT is by quantifier elimination.

Another well-known model-theoretic method for proving the completeness of a theory is the method of model completeness, due to Abraham Robinson [20, 16]. Recently, this method has been used to show the completeness of the theory of finite trees over a finite constructor signature [25].

Both methods for proving the completeness of CFT have their merits. The quantifier elimination used in [3] serves for a concrete decision algorithm, whereas the proof presented here is much simpler. Thus, we think our paper describes a method for proving completeness which can be more easily adapted to other variants of feature logic than the method of quantifier elimination. We will come back to a comparison of the different methods in Section 7.

After summarizing some background material in the next section, Section 3 briefly reviews the theory *CFT* from [22] and some of its basic properties. Section 4 reviews the method of Fraïssé [12] and Ehrenfeucht [11]. In Section 5, we discuss the path constraints we need for the formulation of the strategy. The core of the paper is Section 6, where we prove the completeness of CFT with the method of Section 4. We conclude with a brief comparison to other methods.

2 Preliminaries

We assume infinite sets *Lab* of *label symbols* and *Fea* of *feature symbols*. From this, we define the following first-order signature:

- a unary *label* predicate for every $A \in Lab$, written as Ax,
- a binary *feature* predicate for every $f \in Fea$, written as xfy,
- a unary *arity* predicate for every finite set $F \subseteq Fea$, written as xF,
- the equality predicate, written as $x \doteq y$.

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 $\epsilon p = p = p\epsilon$ 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. Under our signature, every term is a variable, and an atomic formula is either a feature constraint xfy, a label constraint Ax, an arity constraint xF or an equation $x \doteq y$. Compound formulae are obtained as usual. We use $\tilde{\exists}\phi$ [$\tilde{\forall}\phi$] to denote the existential [universal] closure of a formula ϕ. Moreover, $\mathit{var}(\phi)$ is taken to denote the set of all variables that occur free in a formula ϕ.

Structures and satisfaction of formulae are defined as usual. A valuation α into a structure $\mathfrak{A}$ is a total function from the set of all variables into $\mathfrak{A}$. If α is a valuation into $\mathfrak{A}$, x a variable and $a \in \mathfrak{A}$, then $\alpha[x \mapsto a]$ denotes the valuation which maps x to a and coincides with α for all other variables. We use $\phi^{\mathfrak{A}}$ to denote the set of all valuations α such that $\mathfrak{A}, \alpha \models \phi$. 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 $\phi^{\mathfrak{A}} \subseteq \psi^{\mathfrak{A}}$ for every model $\mathfrak{A}$ of T.

A theory T is *complete* if for every closed formula ϕ either ϕ or $\neg\phi$ is a consequence of T. By the well-known completeness theorem of the predicate calculus, the set of consequences of a recursively enumerable theory is again recursively enumerable. A standard argument of recursion theory yields that for any complete and recursively enumerable theory T its set of consequences is decidable: To check whether a closed formula ϕ is a consequence of T, enumerate the set of consequences of T. Since T is complete, either ϕ or $\neg\phi$ shows up in the enumeration. In the former case ϕ is a consequence of T, in the latter case it is not.

The *theory of* a first-order structure $\mathfrak{A}$ is the set of closed first-order formulae which are valid in $\mathfrak{A}$. Two first-order structures $\mathfrak{A}$, $\mathfrak{B}$ are *elementarily equivalent* if they have the same theory. Note that a theory is complete iff all its models are elementarily equivalent. Furthermore, if T is a complete theory and $\mathfrak{A}$ is a model of T, then the theory of $\mathfrak{A}$ coincides with the set of consequences of T. If in addition T is recursively enumerable, then the theory of $\mathfrak{A}$ is decidable.

3 The Theory *CFT*

3.1 Models

We consider two structures of the signature introduced in the last section. The universe of the structure $\mathfrak{I}$ consists of all feature trees. A *feature tree* is a partial function $t : \mathit{Fea}^\star \to \mathit{Lab}$ whose domain is *prefix-closed*, i.e., if $pq \in \mathrm{dom}(t)$, then $p \in \mathrm{dom}(t)$. The *subtree* $p^{-1}t$ of a feature tree t at a path $p \in \mathrm{dom}(t)$ is the feature tree defined by (in relational notation)

$$p^{-1}t \quad := \quad \{(q, A) \mid (pq, A) \in t\}.$$

A feature tree t is called a *subtree* of a feature tree r if t is a subtree of r at some path $p \in \mathrm{dom}(r)$.

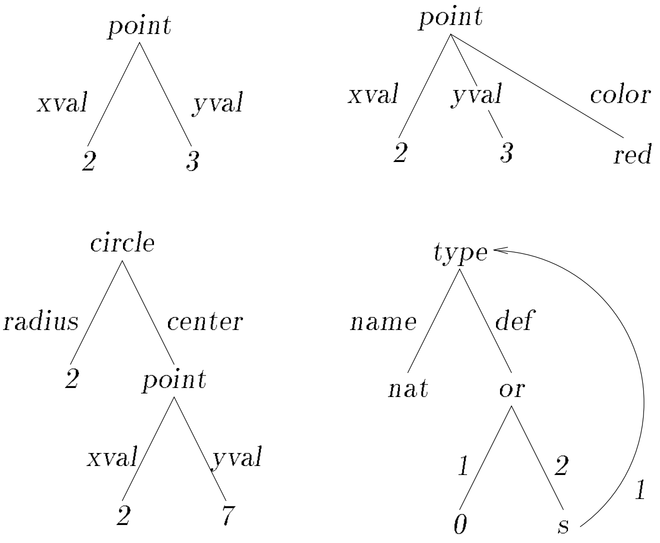

Figure 1: Examples of (in fact rational) feature trees.

The universe of the structure $\mathfrak{R}$ consists of all rational feature trees. A feature tree t is called *rational* if (1) t has only finitely many subtrees and (2) t is finitely branching (i.e., for every $p \in \mathrm{dom}(t)$, the set $\{pf \in \mathrm{dom}(t) \mid f \in \mathit{Fea}\}$ is finite).

The relational symbols are interpreted in $\mathfrak{I}$ as follows:

- $\mathfrak{I}, \alpha \models Ax$ iff $\alpha(x)$ has root label A,
- $\mathfrak{I}, \alpha \models xfy$ iff $f \in \mathrm{dom}(\alpha(x))$ and $\alpha(y) = f^{-1}\alpha(x)$ (i.e., $\alpha(y)$ is the subtree of $\alpha(x)$ at f), and
- $\mathfrak{I}, \alpha \models x\{f_1, \ldots, f_n\}$ if $\mathrm{dom}(\alpha(x)) \cap \mathit{Fea} = \{f_1, \ldots, f_n\}$ (i.e., $\alpha(x)$ has exactly the features $f_1, \ldots, f_n$ at its root).

The interpretation of the relational symbols in $\mathfrak{R}$ is the restriction of the interpretation in $\mathfrak{I}$ to the set of rational feature trees.

3.2 Axioms

The theory *CFT* consists of five axiom schemes. The first set of axioms expresses that labels are disjoint, that features are functional and that an arity constraint fixes the set of features at the root of a feature tree

(S)	$\forall x\,(Ax \wedge Bx \rightarrow \bot)$	$A \neq B$
(F)	$\forall x, y, z\,(xfy \wedge xfz \rightarrow y \doteq z)$	
(A1)	$\forall x, y\,(xF \wedge xfy \rightarrow \bot)$	$f \notin F$
(A2)	$\forall x\,(xF \rightarrow \exists y\, xfy)$	x different from y, $f \in F$

For the last axiom scheme (D) we need the following notion:

Definition 1 (Determinant) *A* simple determinant *is a conjunction of formulae*

$$Ax \wedge x\{f_1, \ldots, f_n\} \wedge xf_1y_1 \wedge \ldots \wedge xf_ny_n$$

where the variables $x, y_1, \ldots, y_n$ are not required to be distinct. We define $det(d) := \{x\}$ for a simple determinant as above. A determinant *δ is a conjunction of simple determinants $d_1 \wedge \ldots \wedge d_n$ such that the $det(d_i) \cap det(d_j) = \emptyset$ for $i \neq j$. We define $det(\delta) := det(d_1) \cup \ldots \cup det(d_n)$ to be the set of variables* determined *by δ.*

Using the quantifier $\exists!\bar{x}\ \Psi$ with the meaning "there exists exactly one tuple $\bar{x}$ such that Ψ", we can formulate the last axiom scheme:

(D) $\forall(var(\delta) \perp det(\delta))\ \exists! det(\delta)\ \delta$ $\quad$ δ is a determinant

An instance of axiom scheme (D) is

$$\begin{array}{ll} \forall z\ \exists! x, y\ (& Ax \wedge x\{f, g\} \wedge xfy \wedge xgz \wedge \\ & By \wedge y\{f, g, h\} \wedge yfz \wedge ygy \wedge yhx) \end{array}$$

Proposition 1 *Both $\mathfrak{I}$ and $\mathfrak{R}$ are models of CFT.*

3.3 Solved Forms

Definition 2 (Solved form) *A subformula of a determinant is called a* solved form. *A variable x is called* constrained *in a solved form S if S contains a constraint of the form Ax, xF or xfy. The set of variables constrained by S is denoted as $con(S)$.*

Hence, for a determinant δ, $con(\delta) = det(\delta)$. Given a solved form S, we denote with $det(S)$ the set $det(\delta_S)$, where δ_S is the largest subset of S that is a determinant. In the following, we use the letters $R, S, T \ldots$ to denote solved forms. The proof of the following lemma is straightforward (see also [22]):

Lemma 2 *Let S be an equation-free conjunction of atomic formulae. Then S is a solved form iff*

1. *S is clash-free, that is it contains no subformula of the form $Ay \wedge By$ (where $A \neq B$), $yF \wedge yG$ (where $F \neq G$) or $yF \wedge yfz$ (where $f \notin F$) and*

2. *$xfy, xfz \in S$ implies that y equals z.*

Proposition 3 *For every solved form S we have*

$$\forall(var(S) \perp con(S))\ \exists con(S)\ S$$

Note that the existence is no longer unique in case of a solved form.

Definition 3 (Reachability) *Given a solved form S, a variable $x \in \mathrm{var}(S)$ and a path $p \in \mathrm{Fea}^*$, we define $|xp|_S$ inductively as*

$$\begin{array}{rcl} |x\epsilon|_S & := & x \\ |xpf|_S & := & \left\{ \begin{array}{ll} \mathit{undefined} & \mathit{if}\ |xp|_S\ \mathit{is\ undefined,\ or\ if}\ |xp|_S = y \\ & \quad \mathit{and}\ S\ \mathit{contains\ no\ constraint\ of\ the\ form}\ yfz \\ z & \mathit{if}\ |xp|_S = y\ \mathit{and}\ yfz \in S \end{array} \right. \end{array}$$

A variable $y \in \mathrm{var}(S)$ is reachable from some *$x \in \mathrm{var}(S)$ if there is a path p such that $|xp|_S = y$.*

Definition 4 (Rooted solved form) *A* rooted solved form *S_x is a solved form S with a distinguished variable $x \in \mathrm{var}(S)$, such that for every $y \in \mathrm{var}(S)$ there is a path p with $|xp|_S = y$. A path p is called* acyclic *in a rooted solved form S_x if for all prefixes $q_1 \neq q_2$ of p we have $|xq_1|_S \neq |xq_2|_S$. For a rooted solved form S_x, the* set of paths to *a variable $y \in \mathrm{var}(S)$ is*

$$[y]_{S_x} := \{p \in \mathrm{Fea}^* \mid |xp|_S = y \ \mathit{and}\ p\ \mathit{is\ acyclic\ in}\ S_x\}$$

The length of a minimal path in $[y]_{S_x}$ is called the depth *of y in the rooted solved form S_x.*

Note that $[y]_{S_x}$ is always non-empty and finite, and that the length of the paths in $[y]_{S_x}$ is bounded by the number of different variables occurring in S.

3.4 Solved Forms and Inequations

Definition 5 (Clash) *An equation $y_1 \doteq y_2$* clashes *with a solved form S, where $y_1, y_2 \in \mathrm{var}(S)$, if there is a path $p \in \mathrm{Fea}^*$, $|y_1p|_S = z_1$, $|y_2p|_S = z_2$, and*

- *$A_1z_1 \wedge A_2z_2 \subseteq S$ with $A_1 \neq A_2$,*
- *or $z_1F_1 \wedge z_2F_2 \subseteq S$ with $F_1 \neq F_2$.*

Proposition 4 *If $y \doteq y'$ clashes with S, then $CFT \models S \rightarrow y \not\doteq y'$.*

Lemma 5 *Let S be a solved form, $y_1, y_2 \in \mathrm{var}(S)$, and let*

$$\begin{array}{llll} Fr(y_1,y_2) = \{(z_1,z_2) \mid \mathit{ex.}\ p \in \mathrm{Fea}^* : (& |y_1p|_S = z_1 & \wedge & |y_2p|_S = z_2 \\ \wedge & z_1 \neq z_2 & \wedge & \{z_1,z_2\} \not\subseteq det(S) \quad)\} \end{array}$$

If $y_1 \doteq y_2$ does not clash with S, then

$$CFT \models \tilde{\forall}\Big(S \rightarrow \Big(y_1 \not\doteq y_2 \leftrightarrow \bigvee_{(z_1,z_2)\in Fr(y_1,y_2)} z_1 \not\doteq z_2\Big)\Big)$$

Proof: This follows immediately from axiom scheme (D). □

In the unification theory for finite terms, an analogous concept is known. There, a satisfiable equation is equivalent to its *frontier*, that is the conjunction of equations obtained by maximal decomposition [19].

As an example of Lemma 5, consider

$$\begin{array}{rcl} S & := & Ax \wedge x\{f,g\} \wedge xfy \wedge xgz \wedge Bz \wedge z\{h\} \wedge yhx \\ & \wedge & Ax' \wedge x'\{f,g\} \wedge x'fy' \wedge x'gz' \wedge By' \wedge y'\{h\} \wedge y'hz' \end{array}$$

Note that $x \doteq x'$ does not clash with S. By Lemma 5,

$$CFT \models \tilde{\forall}\Big(S \rightarrow \Big(x \not\doteq x' \leftrightarrow (y \not\doteq y' \vee z \not\doteq z' \vee x \not\doteq z')\Big)\Big)$$

As another example, consider

$$S' = Ax \wedge x\{f\} \wedge xfx \wedge Ax' \wedge x'\{f\} \wedge x'fx'$$

Since $Fr(x, x')$ is empty, we get $CFT \models (S' \rightarrow (x \not\doteq x' \leftrightarrow \mathit{false}))$. If we replace in S' however Ax by Bx for some $B \neq A$, then $x \doteq x'$ clashes with S' and the lemma does not apply.

4 Ehrenfeucht-Fraïssé Games

Fraïssé [12] gives a definition of elementary equivalence in terms of mappings between structures. In this section we just summarize this method, more detailed expositions can be found e.g. in [13, 18].

Any two isomorphic structures are elementarily equivalent, but there are of course elementarily equivalent structures which are not isomorphic. Hence, to characterize elementary equivalence algebraically we have to weaken the notion of isomorphism. Let $\mathfrak{A}$ and $\mathfrak{B}$ be two structures of a signature σ which consists of (possibly infinitely many) relation symbols only[1], and let τ be a subsignature of σ. A finite sequence $(a_i, b_i)_{1 \leq i \leq n}$ in $(\mathfrak{A} \times \mathfrak{B})^*$ is a *partial τ-isomorphism* if for every $\mathfrak{A}$-valuation α with $\alpha(x_i) = a_i$, every $\mathfrak{B}$-valuation β with $\beta(x_i) = b_i$ and every atomic τ-formula w with $var(w) \subseteq \{x_1, \ldots, x_n\}$ we have $\mathfrak{A}, \alpha \models w \Leftrightarrow \mathfrak{B}, \beta \models w$. Note that, in the context of predicate logic with equality, w might be an equation. In this case, a partial isomorphism is always injective.

Instead of Fraïssé's original theorem we use here the game-theoretic reformulation due to Ehrenfeucht [11]. The game is played on two structures $\mathfrak{A}$ and $\mathfrak{B}$ by two players, the *Spoiler* and the *Duplicator*. In the beginning, the Spoiler chooses a finite

[1] We take this assumption just for simplicity, the definition extends to arbitrary signatures.

subsignature[2] $\tau \subseteq \sigma$ and the number n of rounds to play. The aim of the Duplicator is to build a partial τ-isomorphism of length n. In round i, the Spoiler chooses one of the two structures together with an element a_i, resp. b_i. Then, the Duplicator chooses an element b_i, resp. a_i in the other structure. Both players always know the present state of the game. The Duplicator wins if at the end the sequence $(a_i, b_i)_{1 \leq i \leq n}$ is a partial τ-isomorphism, otherwise the Spoiler wins.

Theorem 6 ([Ehrenfeucht, 1961]) *$\mathfrak{A}$ and $\mathfrak{B}$ are elementarily equivalent iff the Duplicator has a winning strategy for the Ehrenfeucht-Fraïssé game on $\mathfrak{A}, \mathfrak{B}$.*

As an example, take the structure $\mathfrak{I}$ from Section 3 and the structure $\mathfrak{F}$, which is the restriction of $\mathfrak{I}$ to those feature trees which have a finite domain. Note that $\mathfrak{F}$ is not a model of *CFT* since axiom scheme (D) is violated. The Spoiler can play the Ehrenfeucht-Fraïssé game on $\mathfrak{I}, \mathfrak{F}$ in such a way that the Duplicator looses. First, she chooses the finite subsignature consisting of the features f, g only (no label or arity predicates) and fixes the number of rounds to 2. In the first round, she chooses the element a_1 from $\mathfrak{I}$ to be the infinite tree with domain $(fg)^* \cup (fg)^* f$ which maps every node to the label A (note that it does not matter that A is not in the finite subsignature). No matter what the choice of the Duplicator from $\mathfrak{F}$ for b_1 is, the Spoiler will choose a_2 to be the infinite tree with domain $(gf)^* \cup (gf)^* g$, also mapping every node to A. Now we have for $\alpha(x_1) = a_1, \alpha(x_2) = a_2$ that $\mathfrak{I}, \alpha \models x_1 f x_2 \wedge x_2 g x_1$, but there is no $\mathfrak{B}$-valuation β with $\beta(x_1) = b_1$, such that $\mathfrak{F}, \beta \models x_1 f x_2 \wedge x_2 g x_1$. Hence, the Duplicator is looses.

With the structures $\mathfrak{I}$ and $\mathfrak{R}$, on the other hand, the Duplicator has a winning strategy. This strategy will be subject of the next sections.

5 Path Constraints

5.1 Motivation and Definition

For the rest of the paper, we assume two fixed structures $\mathfrak{A}$ and $\mathfrak{B}$ of *CFT*.

How can we find a winning strategy for the Duplicator? Suppose, the Spoiler has fixed n and the finite subsignature. We may assume that the arity predicates of the subsignature are exactly the sets of features in the subsignature, that is the finite subsignature is given as $(\sigma, \phi) \subseteq (Lab, Fea)$. At every stage of the game, the sequence constructed so far must of course be a partial (σ, ϕ)-isomorphism (otherwise, the Duplicator looses immediately), but this is not sufficient, since the Duplicator has to take into account all possible future moves of the Spoiler. A clever move of the Spoiler is to choose an element of a structure which is in relation to *many* elements

[2] Having the Spoiler choose the finite subsignature simplifies the formulation in the case of an infinite signature. This idea is due to Gert Smolka.

which are already in the game. Hence, the Duplicator has to watch for *chains* of relations between the chosen elements that may occur in the future moves. She may, however, exploit the knowledge of n and (σ, ϕ) to restrict the set of relevant chains.

In the context of *CFT*, there is a special class of chains of relations that are expressed as *path constraints* [4]. These are existentially quantified solved forms of a restricted format. As will be explained later, the existentially quantified variables represent in some sense the possible moves of the spoiler.

Definition 6 (Path Constraints) *Path constraints are additional atomic formulae of the forms xpy, Axp, xpF or $xp\downarrow xq$. Here, x, y are variables, $p, q \in Fea^*$, A is a label and F is an arity. The validity of a path constraint π under a valuation α in $\mathfrak{A}$ is inductively given by*

$$\begin{aligned} \mathfrak{A}, \alpha \models x\epsilon y &\Leftrightarrow \mathfrak{A}, \alpha \models x \doteq y \\ \mathfrak{A}, \alpha \models x(pf)y &\Leftrightarrow \mathfrak{A}, \alpha \models \exists z\,(xpz \wedge zfy) \\ \mathfrak{A}, \alpha \models Axp &\Leftrightarrow \mathfrak{A}, \alpha \models \exists z\,(xpz \wedge Az) \\ \mathfrak{A}, \alpha \models xpF &\Leftrightarrow \mathfrak{A}, \alpha \models \exists z\,(xpz \wedge zF) \\ \mathfrak{A}, \alpha \models xp\downarrow yq &\Leftrightarrow \mathfrak{A}, \alpha \models \exists z\,(xpz \wedge yqz) \end{aligned}$$

The path constraints of the form $xp\downarrow yq$ are called a *co-reference constraint.* We identify $xp\downarrow yq$ with $yq\downarrow xp$. A *trivial co-reference constraint* $xp\downarrow xp$ is abbreviated as $xp\downarrow$, it expresses that x has a path p. By the definition of the validity of path constraints, the additional syntax introduced with path constraints is just syntactic sugar for specific existentially quantified solved forms. In the following, we deliberately confuse a path constraint π with an arbitrary existentially quantified solved form that is equivalent to π by Definition 6.

We can also give a direct interpretation for path constraints. The interpretations $f^{\mathfrak{A}}$, $g^{\mathfrak{A}}$ of two features f, g in a structure $\mathfrak{A}$ satisfying the axioms *CFT* are binary relations on $\mathfrak{A}$. Hence, their composition $f^{\mathfrak{A}} \circ g^{\mathfrak{A}}$ is again a binary relation on $\mathfrak{A}$ satisfying

$$a(f^{\mathfrak{A}} \circ g^{\mathfrak{A}})b \iff \exists c \in \mathfrak{A}\colon\ af^{\mathfrak{A}}c \ \wedge\ cf^{\mathfrak{A}}b$$

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

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

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

Let p, q be paths, x, y be variables, and A be a label. Then the interpretation of path constraints is given as follows:

$$\begin{aligned}
\mathfrak{A}, \alpha \models xp \downarrow xq \quad &:\Longleftrightarrow \quad \exists a \in \mathfrak{A}\colon\ \alpha(x)\, p^{\mathfrak{A}}\, a \ \wedge\ \alpha(x)\, q^{\mathfrak{A}}\, a \\
\mathfrak{A}, \alpha \models Axp \quad &:\Longleftrightarrow \quad \exists a \in \mathfrak{A}\colon\ \alpha(x)\, p^{\mathfrak{A}}\, a \ \wedge\ a \in A^{\mathfrak{A}}. \\
\mathfrak{A}, \alpha \models xpF \quad &:\Longleftrightarrow \quad \exists a \in \mathfrak{A}\colon\ \alpha(x)\, p^{\mathfrak{A}}\, a \ \wedge\ a \in F^{\mathfrak{A}}.
\end{aligned}$$

5.2 True Sequences

We can now define, for any $l \geq 1$ and set X of variables, the set of path constraints within the subsignature (σ, ϕ), where the paths are restricted to length at most l and where only the variables from X are used:

$$P_{l,X}^{\sigma,\phi} := \{Axp, xpF, xp \downarrow yq \mid A \in \sigma, F \subseteq \phi, x, y \in X, p, q \in \phi^{\leq l}\}.$$

Here, $\phi^{\leq l}$ is the set of all strings from ϕ^* with length at most l. When σ, ϕ are known from the context, we will simply write $P_{l,X}$ instead of $P_{l,X}^{\sigma,\phi}$. We also write $P_{l,n}^{\sigma,\phi}$ for $P_{l,\{x_1,\ldots,x_n\}}^{\sigma,\phi}$.

Definition 7 *A sequence* $(a_i, b_i)_{1 \leq i \leq n} \in (\mathfrak{A} \times \mathfrak{B})^*$ *is* (σ, ϕ)-true up to l *if for all* $w \in P_{l,n}^{\sigma,\phi}$ *we have: if* $\alpha(x_i) = a_i$ *and* $\beta(x_i) = b_i$ *for all* $1 \leq i \leq n$*, then*

$$\mathfrak{A}, \alpha \models w \Leftrightarrow \mathfrak{B}, \beta \models w\,.$$

Proposition 7 *Every* (σ, ϕ)*-true sequence up to* 1 *is a partial* (σ, ϕ)*-isomorphism.*

Proof: This follows from the definitions, since $CFT \models \forall x, y\ (x \doteq y \leftrightarrow x\epsilon \downarrow y\epsilon)$, $CFT \models \forall x\ (xfy \leftrightarrow xf \downarrow y\epsilon)$, $CFT \models \forall x\ (Ax \leftrightarrow Ax\epsilon)$ and $CFT \models \forall x\ (xF \leftrightarrow x\epsilon F)$. □

Hence, the aim of the Duplicator can be described as constructing a (σ, ϕ)-true sequence up to 1. From the above discussion, it is clear that the Duplicator must always ensure that the sequence constructed so far is (σ, ϕ)-true up to some sufficiently large bound $l = \psi(m)$, which depends on the number of rounds m still to play. If $\psi(m)$ is chosen in the right way, then the Duplicator can extend every (σ, ϕ)-true sequence up to $\psi(m)$ to a (σ, ϕ)-true sequence up to 1 in the remaining m rounds, no matter how the Spoiler plays. The question is of course how an appropriate bound $\psi(m)$ can be determined.

A first guess could be $\psi(m) := m$, since the Spoiler can choose m elements in m rounds. The following example shows that this is not sufficient. Assume that the Spoiler has chosen elements $a_1 \in \mathfrak{A}$ and $a_2 \in \mathfrak{A}$ such that

$$a_1 (ffff)^{\mathfrak{A}} a_2,$$

that the Duplicator has chosen an element $b_1 \in \mathfrak{B}$, and that there are still 2 rounds to play. Assume that the Duplicator selects in this round an element $b_2 \in \mathfrak{B}$ such that

$$b_1(fff)^{\mathfrak{B}}b_2$$

If we define $\psi(m) := m$, then $((a_1, b_1), (a_2, b_2))$ would be a (σ, ϕ)-true sequence up to 2. In this case, the Duplicator will loose if the Spoiler selects a_3 such that $a_1(ff)^{\mathfrak{A}}a_3$ (i.e., the element "in the middle" of a_1 and a_2): if the Duplicator chooses b_3 such that $b_1(ff)^{\mathfrak{B}}b_3$ does not hold, then the Spoiler chooses as a_4 with $a_1 f^{\mathfrak{A}} a_4$ and $a_4 f^{\mathfrak{A}} a_3$ and the Duplicator looses; if the Duplicator selects b_3 such that $b_1(ff)^{\mathfrak{B}}b_3$, then she looses immediately.

Hence, the next guess could be $\psi(m) := 2^m$, since the Spoiler can with one move choose an element "in the middle" of a chain of relations between elements which are already in the sequence. This strategy of the Spoiler would cause the Duplicator, if the number of moves is increased by 1, to duplicate the bound for the first move, which results in the recursion equation $\psi(m+1) = 2 * \psi(m)$. In fact, it can be shown [13] that this bound is sufficient for simple theories like the theory of one successor function. In our case, where $\mathfrak{A}$ and $\mathfrak{B}$ are models of *CFT*, this is not sufficient as can be seen with the following example:

Suppose, the sequence constructed so far is $(a_1, b_1), \ldots, (a_n, b_n)$. The Spoiler chooses an $a \in \mathfrak{A}$ in such a way that for the valuation α with $\alpha(x_i) = a_i$, $\alpha(x_{n+1}) = a$ we have

$$\left.\begin{array}{l} \mathfrak{A}, \alpha \models x_1 r_1 \downarrow x_{n+1} p_1 \\ \mathfrak{A}, \alpha \models x_{n+1} p_1 q_1 \downarrow x_{n+1} p_2 \\ \mathfrak{A}, \alpha \models x_{n+1} p_2 q_2 \downarrow x_{n+1} p_3 \\ \qquad\quad \vdots \\ \mathfrak{A}, \alpha \models x_{n+1} p_k q_k \downarrow x_2 r_2 \end{array}\right\} \tag{1}$$

where all these constraints are in $P^{\sigma,\phi}_{\psi(m),n+1}$ (see Figure 2). Hence, the Duplicator has to find an element $b \in \mathfrak{B}$, such that for the variable valuation β with $\beta(x_i) = b_i$ and $\beta(x_{n+1}) = b$ the same formulae hold in $\mathfrak{B}, \beta$. The problem is that the conjunction of these constraints implies, in every model of *CFT*,

$$x_1 r_1 q_1 \cdots q_k \downarrow x_2 r_2 \tag{2}$$

Hence, in order to satisfy (1) in $\mathfrak{B}, \beta$, (2) has to be satisfied in $\mathfrak{B}, \beta$. But the length of $r_1 q_1 \cdots q_k$ may be much greater than $2 * \psi(m)$. The only thing we can say is that we don't have to care about "cycles" in (1), that is we may assume that every $p_i q_i \neq p_j q_j$ if $i \neq j$. Since there are less than $\mathit{cardinality}(\phi)^{\psi(m)+1}$ many different ϕ-paths of length at most $\psi(m)$, the length of $r_1 q_1 \cdots q_k$ is certainly smaller than $\psi(m) + \psi(m) * \mathit{cardinality}(\phi)^{\psi(m)+1}$. Since a co-reference $x_1 r_1 q_1 \cdots q_k \downarrow x_2 r_2$ entails for every path $r \in \mathit{Fea}^*$ the co-reference $x_1 r_1 q_1 \cdots q_k r \downarrow x_2 r_2 r$ and we want to consider

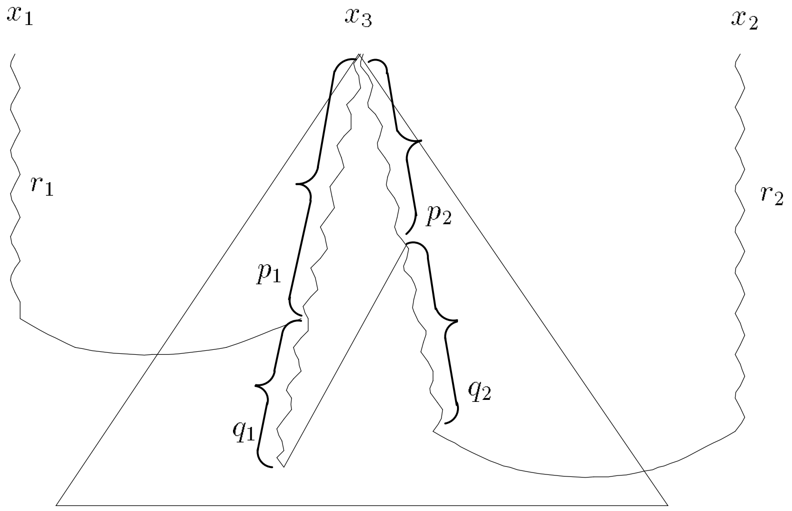

Figure 2: Example of an induced co-reference constraints for $n = 2$. The induced constraint is $x_1 r_1 q_1 q_2 \downarrow x_2 r_2$.

extensions r of length less than $\psi(m)$, we take this recursion equation in order to define ψ:

$$\begin{aligned} \psi(0) &:= 1 \\ \psi(m+1) &:= 2 \times \psi(m) + \psi(m) * \mathit{cardinality}(\phi)^{\psi(m)+1} + 1 \end{aligned}$$

Hence, we can formulate the following requirement for the Duplicator:

> If there are still m rounds to play after completion of this move, make sure that the sequence is true up to $\psi(m)$.

Since $\psi(0) = 1$, and since a 1-true sequence is a partial isomorphism, this will guarantee that the Duplicator wins.

5.3 Path Constraints and Solved Forms

The following lemma gives the connection between satisfiable sets of path constraints and solved forms.

Lemma 8 *Let $P \subseteq P^{\sigma,\phi}_{l,\{x\}}$ be a set of path constraints such that $\exists x\, P$ is satisfiable in CFT. Then there is a rooted solved form $S_x(x, \bar{y})$ with*

1. *CFT $\models \forall x\, (P \leftrightarrow \exists \bar{y}\, S)$*

2. *for every $y \in \bar{y}$ there is a $p \in \phi^{\leq l}$ such that $|xp|_S = y$;*

3. *$Axp \in P$, $xpF \in P$ or $xp \downarrow xq \in P$ implies $A|xp|_S \in S$, $|xp|_S F \in S$ or $|xp|_S = |xq|_S$, respectively.*

Proof: Considering P as a conjunction of existentially quantified solved forms, we first move all quantifier to the outside while renaming variables to avoid capture. We obtain an equivalent formula Q of the form $\exists \bar{v}\, M$, where M is a conjunction of atomic formulae. Then we rewrite Q with the following rule until we obtain a normal form:

$$\frac{\exists \bar{v}, v\ (yfv \wedge yfz \wedge w)}{\exists \bar{v}\ (yfz \wedge w[z/v])} \tag{3}$$

where $v \notin \bar{v}$,and where $w[z/v]$ is the result of replacing every occurrence of v in w by z. The rewriting is obviously terminating since the size of the formula is reduced in every step. Both operations are equivalence transformations in *CFT* that do not change the set of free variables.

Let N be the normal form of Q, and suppose that N is not a solved form. By Lemma 2, N either contains a clash or a subformula $yfz_1 \wedge yfz_2$ where $z_1 \neq z_2$. The existence of a clash contradicts the satisfiability of P. In the second case, since $\mathit{var}(N) = \{x\}$, at least one of z_1 and z_2 must be existentially quantified, hence the rewriting rule (3) applies and N cannot be in normal form.

For the second claim, note that $|xp|_S = y$ is equivalent to $\mathit{CFT} \models S \rightarrow xpy$. Hence, we have to show that for every $y \in \bar{y}$ there is a $p \in \phi^{\leq l}$ such that $\mathit{CFT} \models S \rightarrow xpy$. This claim holds trivially for the initial formula M. Since the claim is conserved during the application of the rewrite rule (3), it holds also for S.

□

For example, the set of path constraints $\{xff \downarrow xg, Axgg\}$ is equivalent to

$$\exists y_1, y_2, y_3\ (xfy_1 \wedge xgy_2 \wedge y_1fy_2 \wedge y_2gy_3 \wedge Ay_3)$$

We have $y_1 = |xf|_S$, $y_2 = |xg|_S$, and $y_3 = |xgg|_S$.

Proposition 9 *Let S_x be a rooted solved form, such that all variables in $\mathrm{var}(S_x)$ have a depth smaller than l, and let $\bar{v} = \mathrm{var}(S) \perp \{x\}$. Then there is a set $P \subseteq P_{l+1,\{x\}}$, such that $\mathit{CFT} \models \forall x\, (\exists \bar{v}\, S \leftrightarrow P)$.*

Proof: We choose for every $y \in \mathit{var}(S_x)$ a path $p_y \in [y]_{S_x}$ of minimal length, and define

$$P = \{xp_yf \downarrow xp_z \mid yfz \in S_x\} \cup \{Axp_y \mid Ay \in S_x\} \cup \{xp_yF \mid yF \in S_x\} \qquad \square$$

6 Completeness of *CFT*

Theorem 10 *The theory CFT is complete.*

To simplify notation, we write α for some valuation in $\mathfrak{A}$ with $\alpha(x_i) = a_i$ for $1 \leq i \leq n$ and $\alpha(x_{n+1}) = a$, and β' for some valuation in $\mathfrak{B}$ with $\beta'(x_i) = b_i$. In the following, we take the variable x instead of x_{n+1}. Hence, α and β' represent the sequence constructed so far plus the choice of the Spoiler. It is now the Duplicator's task to find a β extending β' to x.

Since $(a_1, b_1), \ldots, (a_n, b_n)$ is true up to $\psi(m+1)$, we know for any $w \in P_{\psi(m),n}$ that $\mathfrak{A}, \alpha \models w$ iff $\mathfrak{B}, \beta' \models w$. Hence, in order to find an element $b \in \mathfrak{B}$ as required, we have only to care for the constraints which involve x. We distinguish between those path constraints which involve x only (the internal constraints), and those which link x with some other variable x_i (the external constraints).

$$\begin{aligned}
I^+ &:= \{w(x) \in P_{\psi(m),n+1} \mid \mathfrak{A}, \alpha \models w\} \\
I^- &:= \{\neg w(x) \in P_{\psi(m),n+1} \mid \mathfrak{A}, \alpha \models \neg w\} \\
E^+_= &:= \{w(x, x_i) \in P_{\psi(m),n+1} \mid \mathfrak{A}, \alpha \models w\} \\
E^-_= &:= \{\neg w(x, x_i) \in P_{\psi(m),n+1} \mid \mathfrak{A}, \alpha \models \neg w\}
\end{aligned}$$

Note that $E^+_=$ ($E^-_=$) consists of (negated) co-reference constraints only, we use the subscript "=" to emphasize this. We have to find some $b \in \mathfrak{B}$ such that for $\beta := \beta'[x \mapsto b]$ we have

$$\mathfrak{B}, \beta \models I^+ \wedge E^+_= \wedge I^- \wedge E^-_=$$

Theorem 10 is a consequence of the following lemma, which we will prove in the next subsection:

Lemma 11 *Let* $(a_1, b_1), \ldots, (a_n, b_n)$ *be* (σ, ϕ)*-true up to* $\psi(m+1)$ *and* $a \in \mathfrak{A}$*. Then there exists a formula* $\Delta(x, x_1, \ldots, x_n)$*, such that*

$$\begin{aligned}
\mathfrak{B}, \beta' &\models \exists x\, \Delta && (4) \\
\mathfrak{B}, \beta' &\models \forall x\, (\Delta \rightarrow (I^+ \wedge E^+_= \wedge I^- \wedge E^-_=)) && (5)
\end{aligned}$$

Proof of Theorem 10: By Lemma 11, the Duplicator has a strategy that guarantees the constructed sequence to be true up to $\psi(m)$ if there are still m rounds to play. This is, by Proposition 7, a winning strategy. □

6.1 Proof of Lemma 11

6.1.1 Induced Co-references

By Lemma 8, there is a solved form $R(x, \bar{v})$ with $CFT \models I^+ \leftrightarrow \exists \bar{v}\, R$. Note that some of the variables of R are already completely determined by the valuation of the x_i's in combination with the external co-references in $E^{\pm}_{=}$. These variables are at least those $z \in var(R)$ with the property that $|xp|_R = z$ and $xp \downarrow x_jq \in E^{\pm}_{=}$ for some x_j, q. As the discussion on induced co-references in Section 5.2 (see (1) and Figure 2) shows, these are not the only variables uniquely determined by the valuation of the x_i's.

In this section we therefore define the notion of an induced co-reference, and we show that the induced co-references can be reduced to co-references in $P_{\psi(m+1),n}$, which are satisfied by $\mathfrak{A}, \alpha$ and henceforth are also satisfied by $\mathfrak{B}, \beta'$.

Definition 8 *Let I^+ and $E^{\pm}_{=}$ be given as described. A* co-reference sequence *is a sequence of path constraints of the form*

$$\left.\begin{array}{lll} x_ir_1 \downarrow xp_1 & \in & E^{\pm}_{=} \\ xp_1q_1 \downarrow xp_2 & \in & I^+ \\ xp_2q_2 \downarrow xp_3 & \in & I^+ \\ \vdots & & \\ xp_kq_k \downarrow xr_2 & \in & I^+ \end{array}\right\} \qquad (6)$$

A co-reference sequence is called cycle-free *if $p_lq_l \neq p_{l'}q_{l'}$ for every $1 \leq l < l' \leq k$. The external co-reference* induced by (6) *is*

$$x_ir_1q_1 \ldots q_k \downarrow xr_2.$$

Proposition 12 *Let $x_ir_1s{\downarrow}xr_2$ be an external co-reference induced by a co-reference sequence. Then there exists an external co-reference $x_ir_1s' \downarrow xr_2$ that is induced by a cycle-free co-reference sequence.*

Proof: Let

$$\mathrm{Seq} = (\mathrm{Seq}_0, \ldots, \mathrm{Seq}_k) = (x_ir_1 \downarrow xp_1,\ xp_1q_1 \downarrow xp_2,\ xp_2q_2 \downarrow xp_3,\ \ldots\ ,\ xp_kq_k \downarrow xr_2)$$

be a co-reference sequence of minimal length that induces $x_ir_1s{\downarrow}xr_2$ for some s, that is $s = q_1, \ldots, q_k$, and assume that Seq is not cycle-free. Hence, there are $l < l'$ such that $p_lq_l = p_{l'}q_{l'}$. Then eliminating the elements $\mathrm{Seq}_l, \ldots, \mathrm{Seq}_{l'-1}$ from Seq results in a shorter co-reference sequence that induces $x_ir_1q_1, \ldots, q_l, q_{l'+1}, \ldots, q_k \downarrow xr_2$, in contradiction to the minimality of Seq. □

Proposition 13 *Let $x_ir_1q_1 \ldots q_k{\downarrow}xr_2$ be an external co-reference induced by a cycle-free co-reference sequence. Then $r_2 \in \phi^{\leq\psi(m)}$. Furthermore, we have $r_1q_1 \ldots q_k \in \phi^{\leq\psi(m+1)-(\psi(m)+1)}$.*

Proof: Let $\pi = x_ir_1q_1 \ldots q_k{\downarrow}xr_2$ be given as described, and let $\mathrm{Seq} = (\mathrm{Seq}_0, \ldots, \mathrm{Seq}_k)$ be a cycle-free co-reference sequence that induces π.

Since the final element $xp_kq_k \downarrow xr_2$ of Seq is in I^+, we obtain immediately that $r_2 \in \phi^{\leq \psi(m)}$.

By definition of a co-reference sequence, $r_1 \in \phi^{\leq \psi(m)}$ and $q_l \in \phi^{\leq \psi(m)}$ for every $l = 1, \ldots, k$. There are less than $\mathit{cardinality}(\phi)^{\psi(m)+1}$ many different ϕ-paths of length at most $\psi(m)$. Since Seq is cycle-free, this implies that the length k of Seq is smaller than $\mathit{cardinality}(\phi)^{\psi(m)+1}$. Hence, the length of $r_1q_1 \ldots q_k$ is smaller than $\psi(m) + \psi(m) * (\mathit{cardinality}(\phi)^{\psi(m)+1}) = \psi(m+1) \perp (\psi(m)+1)$. □

Now we define

$$\begin{array}{rcl} IC & := & \{x_iq_i \downarrow xp \mid x_iq_i \downarrow xp \text{ is induced by a cycle-free co-reference sequence}\} \\ C & := & \{x_iq_i \downarrow z\epsilon \mid z = |xp|_R \text{ and } x_iq_i \downarrow xp \in IC\} \end{array}$$

Obviously, all variables in $\mathit{var}(C)\backslash\{x_1, \ldots, x_n\}$ are variables of R that are uniquely determined in $\exists \bar{v}\; R \wedge E^{\pm}_{=}$ by the valuation of the x_i's.

Proposition 14 *Let $xp \downarrow xq \in I^+$ be a path constraint such that there is some prefix p' of p with $|xp'|_R \in \mathit{var}(C)$. Then $|xq|_R \in \mathit{var}(C)$.*

Proof: Let $xp \downarrow xq \in I^+$ and let $p = p'p''$ such that $|xp'|_R \in \mathit{var}(C)$. Hence, there are some x_i, q_i such that $x_iq_i \downarrow xp'$ is in IC. Let Seq be a co-reference sequence that induces $x_iq_i \downarrow xp'$. Then appending $xp \downarrow xq = xp'p'' \downarrow xq$ to Seq produces a co-reference sequence that induces $x_iq_ip'' \downarrow xq$. Proposition 12 shows that there is an external co-reference $x_iq' \downarrow xq$ that is induced by an cycle-free co-reference sequence. Hence, $|xq|_R \in \mathit{var}(C)$. □

6.1.2 Definition of $\Delta(x)$ and Proof of Lemma 11 (4)

We could now already show a weaker version of Lemma 11, where only $I^+ \wedge E^{\pm}_{=}$ are considered, by defining $\Delta(x) = \exists \bar{v}\; (R \wedge C)$. We will not prove this but move on to the definition of a Δ which also entails $I^- \wedge E^{-}_{=}$.

To illustrate the idea, assume that $\neg Axf \in I^-$, where $|xf|_R = y$. If R does not contain a label constraint for y, then we can extend R by a label constraint By where $B \notin \sigma$. The fact that we have introduced a new label constraint which (possibly) does not hold in $\mathfrak{A}, \alpha$ does not hurt at all, since we only care for the labels in the finite subset σ. The point is that, since by axiom scheme (S) different label constraints are pairwise incompatible, any label constraint Ay with $A \in \sigma$ is now disentailed. In this way, we can use positive constraints to enforce some negative constraints.

In the first step, we extend R to a solved form S such that every variable from $\mathit{var}(S) \perp \mathit{var}(C)$ carries an arity constraint. Let h be some feature not contained in ϕ, and let $Y = \mathit{var}(R) \perp \mathit{var}(C)$. For every $y \in Y$ let

$$F_y \;\; := \;\; \{f \mid yfv \in R \text{ for some } v \in \mathit{var}(R)\}$$

be the set of features defined on y in R. Now we define

$$\begin{aligned} Y_{\mathrm{na}} &:= \{y \in Y \mid R \text{ contains no arity constraint for } y\} \\ S &:= R \wedge \bigwedge_{y \in Y_{\mathrm{na}}} y(F_y \cup \{h\}) \end{aligned}$$

In the next step, we extend S to S' such that for all $y \in Y$, if $yF \in S'$, then for every $f \in F$ there is a variable z such that $yfz \in S'$, and such that every variable from $var(S') \perp var(C)$ carries an arity constraint.

$$\begin{aligned} M &:= \{(y,f) \mid y \in Y,\ y\{\ldots,f,\ldots\} \in R \text{ and for all } v \in var(R) : yfv \notin S\} \\ S' &:= S \wedge \bigwedge_{(y,f) \in M, v \text{ new}} (yfv \wedge v\{\}) \end{aligned}$$

Let $V = var(S') \perp var(C)$. In the last step we extend S' to a determinant T, such that $var(T) \perp var(C) \subseteq det(T)$. We choose for every variable $y \in V$ a label $A_y \notin \sigma$ such that for all $y \in V$:

- $A_y \neq A_z$ for all $z \in V \perp \{y\}$,
- for all $p \in (\phi \cup \{h\})^{\leq \psi(m+1)+1}$ and $1 \leq i \leq n$: $\mathfrak{B}, \beta' \models \neg A_y x_i p$

This is possible since we assume an infinite supply of labels and features. We define

$$\begin{aligned} Y_{ns} &:= \{y \in V \mid S' \text{ contains no label constraint for } y\} \\ T &:= S' \wedge \bigwedge_{y \in Y_{ns}} A_y y \end{aligned}$$

Finally, we define $\bar{y} = var(T) \perp \{x\}$ and

$$\Delta := \exists \bar{y} (T \wedge C)$$

Proposition 15 $\mathfrak{B}, \beta' \models \exists x\ \Delta$.

Proof: Let T_{det} be the greatest subformula of T with $var(T_{\mathrm{det}}) \subseteq var(C)$, and let T_{indet} be the rest of T. By definition of T and by Proposition 14, the formula T_{indet} is a determinant with $con(T_{\mathrm{indet}}) \cap var(C \wedge T_{\mathrm{det}}) = \emptyset$. By axiom scheme (D), we know that

$$CFT \models \exists \bar{y}\ (T_{\mathrm{det}} \wedge C) \rightarrow \exists \bar{y}\ (T_{\mathrm{indet}} \wedge T_{\mathrm{det}} \wedge C)$$

We show that $\mathfrak{B}, \beta' \models \exists \bar{z}\ (C \wedge T_{\mathrm{det}})$, where $\bar{z} = var(T_{\mathrm{det}}) \subseteq \bar{y}$. Let C_{def} be a subset of C containing for every $z \in \bar{z}$ exactly one constraint $z\epsilon \downarrow x_i p_i$. Let β_{def} be the modification of β' on $\bar{z}$ with the property that $\mathfrak{B}, \beta_{\mathrm{def}} \models C_{\mathrm{def}}$.

We claim that $\mathfrak{B}, \beta_{\mathrm{def}} \models T_{\mathrm{det}} \wedge C$. For the path constraints $z\epsilon \downarrow x_i q_i \in C \perp C_{\mathrm{def}}$, $\mathfrak{B}, \beta_{\mathrm{def}} \models z\epsilon \downarrow x_i q_i$ if and only if $\mathfrak{B}, \beta' \models x_i q_i \downarrow x_j p_j$, where $z\epsilon \downarrow x_j p_j \in C_{\mathrm{def}}$ is the path constraint defining the valuation of z. Now we know $\mathfrak{A}, \alpha \models x_j p_j \downarrow x_i q_i$, since both $z\epsilon{\downarrow}x_i q_i$ and $z\epsilon{\downarrow}x_j p_j$ are induced external co-reference constraints. By Proposition 13, we know that $x_i p_i \downarrow x_j q_j \in P_{\psi(m+1),n}$, which implies $\mathfrak{B}, \beta' \models x_i p_i \downarrow x_j q_j$.

The proof for the constraints in T_{det} is analogous. □

6.1.3 Proof of Lemma 11 (5)

We split the proof into several propositions, according to the kind of constraints that are to be entailed. First we look at the easy ones: positive constraints (Proposition 16), negated path constraints where the path itself is not defined (Proposition 17), negated path constraints where the path (or both in case of a co-reference) lead to a variable in $var(C)$ (Proposition 18) and negative label and arity constraints (Proposition 19, if none of the two previous propositions applies).

The difficult case is the one of negated co-reference constraints. We first show, in Lemma 20, that we did not by accident introduce external co-references in the construction of T. Using this proposition, we can finally show that the negated external (Proposition 21) and internal (Proposition 22) are implied.

Proposition 16 $\mathfrak{B}, \beta' \models \forall x\, (\Delta \to (I^+ \wedge E^{\pm}_{=}))$.

Proof: For a constraint $xp \downarrow x_i q_i$ the claim follows since $(xp \downarrow x_i q_i)$ is a cycle-free induced external co-reference sequence, and henceforth contained in IC. For the constraints in I^+ this follows from the definition of R and from $R \subseteq T$. □

Proposition 17 *Let $\pi \in I^- \cup E^-_=$ contain xp, where $\neg xp{\downarrow} \in I^-$. Then $\mathfrak{B}, \beta' \models \forall x\, (\Delta \to \pi)$.*

Proof: Let $\neg xp{\downarrow} \in I^-$. Let qf be the unique prefix of p such that $xq{\downarrow} \in I^+$ (q might be ϵ), and $\neg x(qf){\downarrow} \in I^-$.

If R contains an arity constraint yF for $y = |xq|_R = |xq|_T$, then $f \notin F$ since $\neg xqf{\downarrow} \in I^-$. Since by construction $yF \in T$, this implies $CFT \models \forall x\, (\Delta \to \neg xqf{\downarrow})$ and henceforth $CFT \models \forall x\, (\Delta \to \pi)$.

If R contains no arity constraint yF for $y = |xq|_R = |xq|_T$, then we have added in T an arity constraint $y(F_y \cup \{h\})$ with h different from f. Now $F_y = \{g \mid \exists y' : ygy' \in R\}$ cannot contain f since $\neg xqf{\downarrow} \in I^-$. Hence, we have again $CFT \models \forall x\, (\Delta \to \neg xqf{\downarrow})$ and therefore $CFT \models \forall x\, (\Delta \to \pi)$. □

Proposition 18 *Let $\pi \in I^- \cup E^-_=$ such that for every p, if xp occurs in π then $|xp|_R \in var(C)$. Then $\mathfrak{B}, \beta' \models \forall x\, (\Delta \to \pi)$.*

Proof: Let $\pi = \neg xp \downarrow xq \in I^-$ such that such that both $|xp|_R \in var(C)$ and $|xq|_R \in var(C)$. Let $xp \downarrow x_i p_i \in IC$ and $xq \downarrow x_j p_j \in IC$ be two external co-references for xp and xq, respectively.

Since $\mathfrak{A}, \alpha \models \neg xp \downarrow xq$ and $\mathfrak{A}, \alpha \models IC$, we get $\mathfrak{A}, \alpha \models \neg x_i p_i \downarrow x_j q_j$. By Proposition 13, $x_i p_i \downarrow x_j q_j \in P_{\psi(m+1),n}$, hence $\mathfrak{B}, \beta' \models \neg x_i p_i \downarrow x_j q_j$. Since $CFT \models \forall x\ (\Delta \rightarrow (xp \downarrow x_i p_i \wedge xq \downarrow x_j p_j))$, we obtain $\mathfrak{B}, \beta' \models \forall x\ (\Delta \rightarrow \neg xp \downarrow xq)$.

The proof for the other kinds of constraints is analogous. □

Proposition 19 *Let $\pi \in I^-$ be of the form $\neg Axp$ or $\neg xpF$ such that $xp\downarrow \in I^+$ and $|xp|_R \notin \mathrm{var}(C)$. Then $\mathfrak{B} \models \forall x\ (\Delta \rightarrow \pi)$.*

Proof: Let $\neg Axp \in I^-$ such that $xp\downarrow \in I^+$ (the proof for arity constraints is analogous). Hence, $|xp|_R$ is defined.

Since $\neg Axp \in I^-$, we know that R contains no label constraint Ay for $y = |xp|_R = |xp|_T$. Hence, either R contains a label-constraint By with $B \neq A$, which implies that T contains By, or we have added a By in T with $B \neq A$. In any case, this implies $CFT \models \forall x\ (\Delta \rightarrow \neg Axp)$. □

Lemma 20 *Let $y \in \mathrm{var}(T) \perp \mathrm{var}(C)$. Then for every $x_i q_i \downarrow xq \in IC$ we have $\mathfrak{B}, \beta' \models \forall \mathrm{var}(T)\ (T \wedge C \rightarrow \neg y\epsilon \downarrow x_i q_i)$.*

Proof: Note that $x_i q_i \downarrow xq \in IC$ implies by Proposition 13 that $q_i \in \phi^{\leq \psi(m+1)-(\psi(m)+1)}$. Let δ be the greatest rooted solved form which is rooted by y and contained in T, and let $|xp|_R = y$. Furthermore, let for some new variable y', $\alpha_{y'} := \alpha[y' \mapsto \alpha(x_i) q_i^{\mathfrak{A}}]$ (hence, $\mathfrak{A}, \alpha_{y'} \models y'\epsilon \downarrow x_i q_i$). We have the following cases:

1. $\delta \not\subseteq R$. By the way T was constructed, we cannot have added an arity constraint or a feature constraint in T without adding a label constraint. Hence, $\delta \not\subseteq R$ implies that we have added a label constraint Az in T for some $z \notin var(C)$ such that
$$\text{for all } r' \in (\phi \cup \{h\})^{\leq \psi(m+1)+1} : \quad \mathfrak{B}, \beta' \models \neg A x_i r'. \tag{7}$$
Now, Az in δ implies
$$\mathfrak{B}, \beta' \models \forall x\ (\Delta \rightarrow Axpr)$$
for some $r \in (\phi \cup \{h\})^{\leq \psi(m)+1}$ with $|yr|_\delta = z$. Since $q_i r \in (\phi \cup \{h\})^{\leq \psi(m+1)+1}$, (7) implies
$$\mathfrak{B}, \beta' \models \neg A x_i q_i r,$$
which implies $\mathfrak{B}, \beta' \models \forall x(\Delta \rightarrow \neg xpr \downarrow x_i q_i r)$, hence $\mathfrak{B}, \beta' \models \forall x(\Delta \rightarrow \neg xp \downarrow x_i q_i)$.

2. $\delta \subseteq R$. Let $\bar{v} = var(\delta) \perp \{y\}$. This case is divided into the following cases:

(a) $\mathfrak{A}, \alpha_{y'} \not\models \exists y(y' \doteq y \wedge \exists\bar{v}\delta)$. Since $\delta \subseteq R$, $CFT \models \forall x(\exists\bar{u}R \leftrightarrow I^+)$ with $\bar{u} = var(R) \perp \{x\}$ and $I^+ \subseteq P_{\psi(m),\{x\}}$, there is by Lemma 8 and Proposition 9 a finite set of path constraints $P \subseteq P_{\psi(m)+1,\{y\}}$ such that

$$CFT \models \forall y\ (\exists\bar{v}\ \delta \leftrightarrow P)$$

Since $\mathfrak{A}, \alpha_{y'} \not\models \exists y(y' \doteq y \wedge \exists\bar{v}\delta)$, we know that there is a path constraint $\pi \in P$ such that $\mathfrak{A}, \alpha_{y'} \not\models \exists y(y' \doteq y \wedge \pi)$.

The first case is that π is of the form Ayr. Then

$$CFT \models \forall x\ (\Delta \rightarrow Axpr).$$

Since $\alpha_{y'}$ was the unique modification of α satisfying $\mathfrak{A}, \alpha_{y'} \models y'\epsilon \downarrow x_i q_i$, we get

$$\mathfrak{A}, \alpha \models \neg A x_i q_i r.$$

Since $q_i \in \phi^{\leq\psi(m+1)-(\psi(m)+1)}$ by Proposition 13 and $r \in \phi^{\leq\psi(m)+1}$, we know that $Ax_iq_ir \in P_{\psi(m+1),n}$. Hence, $\mathfrak{B}, \beta' \models \neg Ax_iq_ir$, which implies $\mathfrak{B}, \beta' \models \forall x(\Delta \rightarrow \neg xpr \downarrow x_iq_ir)$.

The proof for the other kinds of path constraints is analogous.

(b) $\mathfrak{A}, \alpha_{y'} \models \exists y(y' \doteq y \wedge \exists\bar{v}\delta)$. Let $\exists\bar{v}'\delta'$ be a fresh copy of $\exists\bar{v}\delta$ such that y is renamed to y'. Then $y \doteq y'$ does not clash with $\delta \wedge \delta'$ and $\mathfrak{A}, \alpha_{y'} \models \exists\bar{v}'\delta'$. Since $y \notin var(C)$, we have $\mathfrak{A}, \alpha \models \neg xp \downarrow x_iq_i$. Hence, $\delta \subseteq R$ implies

$$\mathfrak{A}, \alpha \models \forall y, y'\ (xp \downarrow y\epsilon \wedge x_iq_i \downarrow y'\epsilon \rightarrow \neg y \doteq y' \wedge \exists\bar{v}\delta \wedge \exists\bar{v}'\delta') \qquad (8)$$

Now (8) implies by Lemma 5 that there is a path $r \in \phi^{\leq\psi(m)}$ such that $|yr|_\delta \notin det(\delta)$ and

$$\mathfrak{A}, \alpha \models \forall y, y'\ (xp \downarrow y\epsilon \wedge x_iq_i \downarrow y'\epsilon \rightarrow \neg yr \downarrow y'r),$$

Since $z = |yr|_\delta$ is an undetermined variable in T, and all undetermined variables in T are contained in $var(C)$, we know that there is a $x_jp_j \downarrow z\epsilon \in C$. Hence,

$$\mathfrak{A}, \alpha \models \neg x_jp_j \downarrow x_iq_ir$$

and

$$CFT \models \forall x\ (\Delta \rightarrow (xpr \downarrow x_iq_ir \leftrightarrow x_jp_j \downarrow x_iq_ir)).$$

Now Proposition 13 shows $p_j \in \phi^{\leq\psi(m+1)}$ and $q_ir \in \phi^{\leq\psi(m+1)}$. Hence, $\mathfrak{B}, \beta' \models \neg x_jp_j \downarrow x_iq_ir$, hence $\mathfrak{B}, \beta' \models \forall x\ (\Delta \rightarrow \neg xpr \downarrow xq_ir)$. □

Proposition 21 *Let $\pi \in E_{\equiv}^{-}$ be of the form $\neg xp \downarrow x_iq_i$. Then $\mathfrak{B}, \beta' \models \forall x\ (\Delta \rightarrow \pi)$.*

Proof: If $\mathfrak{A}, \alpha \models \neg x_i q_i{\downarrow}$, then the claim follows since $(a_i, b_i)_{i=1,\ldots,n}$ is true up to $\psi(m+1)$. Otherwise, let y' be a new variable, and assume wlog. that $\mathfrak{A}, \alpha \models y'\epsilon{\downarrow}x_i q_i$.

If $\mathfrak{A}, \alpha \models \neg xp{\downarrow}$, the claim follows by Proposition 17. Otherwise, let $y = |xp|_T$. If $y \in \mathit{var}(C)$, then there is a $xp{\downarrow}x_j q' \in \mathit{IC}$. Hence, $\mathfrak{A}, \alpha \models \neg x_i q_i{\downarrow}x_j q'$, and consequently $\mathfrak{B}, \beta' \models \neg x_i q_i \downarrow x_j q'$. Hence, $\mathfrak{B}, \beta' \models \forall x\, (\Delta \rightarrow \neg x_i q_i \downarrow xp)$. If $y \notin \mathit{var}(C)$, then the proof follows from Lemma 20. □

Proposition 22 *Let $\pi \in I^-$ be of the form $\neg xp \downarrow xq$. Then $\mathfrak{B}, \beta' \models \forall x\, (\Delta \rightarrow \pi)$.*

Proof: If $\neg xp{\downarrow} \in I^-$ or $\neg xq{\downarrow} \in I^-$, then the claim follows from Proposition 17.

Otherwise, let $y = |xp|_R$ and $y' = |xq|_R$. If $y \doteq y'$ clashes with R, then the claim follows immediately from Proposition 4.

Otherwise, let R' be R extended by all feature constraints $vfv' \in T$ with $vfv' \notin R$. Note that for every $y \in \mathit{var}(R')$, $y \notin \mathit{det}(R')$ implies that either there is no label constraint Ay for y in R', or there is no arity constraint yF for y in R'. Now $y \doteq y'$ does not clash with R'. Hence, there is by Lemma 5 a path r, $z = |yr|_{R'}$, $z' = |y'r|_{R'}$, $z \neq z'$, such that one of z and z' is not in $\mathit{det}(R')$, and $\mathfrak{A}, \alpha \models \forall \bar{y}\, (R' \rightarrow z \not\doteq z')$. Hence, $\mathit{CFT} \models \forall \bar{y}\, (R' \wedge z \not\doteq z' \rightarrow \neg xp \downarrow xq)$.

Let $p' \in \phi^{\leq \psi(m)+1}$ and $q' \in \phi^{\leq \psi(m)+1}$ be minimal paths with $z = |xp'|_{R'}$ and $z' = |xq'|_{R'}$, respectively. Note that if $z \in \mathit{var}(R)$ (resp. $z' \in \mathit{var}(R)$),then $\mathrm{len}(p') \leq \psi(m)$ (resp. $\mathrm{len}(q') \leq \psi(m)$).

We have the following cases:

1. $z, z' \in \mathit{var}(C)$. Then the claim follows from Proposition 18

2. $z \in \mathit{var}(C)$, $z' \notin \mathit{var}(C)$. Hence, there is a $x_i q_i{\downarrow}xp' \in \mathit{IC}$, and $\mathfrak{A}, \alpha \models \neg x_i q_i{\downarrow}xq'$. By Lemma 20, $\mathfrak{B}, \beta' \models \forall x\, (\Delta \rightarrow \neg x_i q_i \downarrow xq')$, and by construction of Δ we have $\mathfrak{B}, \beta' \models \forall x\, (\Delta \rightarrow x_i q_i \downarrow xp')$. Hence, $\mathfrak{B}, \beta' \models \forall x\, (\Delta \rightarrow \neg xp \downarrow xq)$.

3. $z, z' \notin \mathit{var}(C)$. Let wlog. $z \notin \mathit{det}(R')$. If R contains no label constraint for z, then we have added a label A for z in T which is different from the label for z' in T. Hence, $\mathit{CFT} \models \forall x\, (\Delta \rightarrow (Axp' \wedge \neg Axq'))$, which implies $\mathfrak{B}, \beta \models \forall x\, (\Delta \rightarrow \neg xp' \downarrow xq')$.

   A similar analysis applies if R contains no arity constraint for z. □

Now, (4) is Proposition 15, and (5) follows from Proposition 16, 17, 18, 19, 22 and 21.

7 Conclusion

We have proven the completeness of the feature theory *CFT*, which unifies the completeness results for *FT* [4] and for rational constructor trees [7, 17]. We feel that the use of features and path constraints significantly simplifies the logic of trees. The same proof idea could be applied to *FT* (where we can always, by lack of arity predicates, add predicates which enforce the inequality of all involved variables). We are confident, that also in the case of *FT* the technique of Ehrenfeucht-Fraïssé Games yields a simpler proof than the quantifier elimination given in [4].

We conclude with a comparison to other techniques, which have been recently employed for proving the completeness of tree axioms: Model Completeness, for the case of finite trees over a finite constructor signature [25], and quantifier elimination for *CFT* [2, 3].

The proof technique using model completeness is due to Abraham Robinson [20]. A theory T is called model complete, if on the class of models of T, the substructure relation coincides with the *elementary* substructure relation (which means that the elements of a substructure $\mathfrak{A}$ of $\mathfrak{B}$ have in both structures the same first-order properties). Model completeness alone is independent of completeness, but if in addition the theory T has an algebraic prime model, then model completeness implies completeness. For the completeness proof of Clarks Equality Theory, that is the axioms of finite trees over a finite constructor signature, it is fairly obvious that the tree structure itself is algebraically prime. To prove the model completeness of the theory, the most convenient way is to show that if $\mathfrak{A} \subseteq \mathfrak{B}$ are models of the theory, then any existential sentence in $\mathfrak{B}_\mathfrak{A}$ is valid in $\mathfrak{A}_\mathfrak{A}$ (the index $\mathfrak{A}$ indicates, that we consider all elements of $\mathfrak{A}$ as additional constants).

Hence, there is a similarity to the technique of Ehrenfeucht-Fraïssé Games, where the additional constants from $\mathfrak{A}$, which occur in an existential formula, correspond to the given sequence $(a_i, b_i)_i$ in the game, and the existential quantifiers correspond to the Duplicators quest for an element. Nevertheless, it seems to be more difficult to prove that arbitrary existential sentences are maintained, since we may have several existential quantifiers, and since we cannot exploit an upper bound on the length of "interesting chains", as we did when playing the game. On the other hand, if we can prove model completeness, we obtain additional insight about the theory.

Now let's turn to the comparison of our proof with the quantifier elimination proof done in [2, 3], which uses an overall structure similar than [17]. Clearly, we cannot fully eliminate quantifiers. Hence, this is a quantifier elimination relative to a set of formulae (called prime formulae), i.e., every *CFT*-formula ϕ can be transformed into a Boolean combination of prime formulae.

The set of prime formulae consists of all existential quantified solved forms which are rooted (i.e., all variables are reachable from the free variables). For the quantifier elimination one has to show that the set of prime formulae satisfies certain properties. It must contain all atomic formulae, and must be closed under conjunction and existential quantification. Furthermore, one has to show that for all prime formulae

$\psi, \psi_1, \ldots, \psi_n$

$$\exists x(\psi \wedge \bigwedge_{i=1}^{n} \neg\psi_i) \mathrel{\models\!\!\!\mid}_{CFT} \bigwedge_{i=1}^{n} \exists x(\psi \wedge \neg\psi_i), \tag{9}$$

and that for all prime formulae ψ, ψ' there exists a Boolean combination of prime formulae δ such that

$$\exists x(\psi \wedge \neg\psi') \mathrel{\models\!\!\!\mid}_{CFT} \delta \tag{10}$$

(9) and (10) together allow for the elimination of one existential quantifier. A universal quantifier is eliminated by transforming $\forall x\phi$ into $\neg\exists x\neg\phi$.

The most difficult part is to prove (9), i.e., to show that

$$\bigwedge_{i=1}^{n} \exists x(\psi \wedge \neg\psi_i) \models_{CFT} \exists x(\psi \wedge \bigwedge_{i=1}^{n} \neg\psi_i). \tag{11}$$

To show this implication, for every β_i a finite set of path constraints Π_i is calculated such that $\Pi_i \mathrel{\models\!\!\!\mid}_{CFT} \beta_i$. In second step, ψ is extended to a prime formula ψ_{ext} such that

$$\psi_{ext} \models \bigwedge_{i=1}^{n} \neg\Pi_i.$$

The construction of ψ_{ext} is similar to the construction of T in the proof of Lemma 11.

By and large, we can say that our proof contains the kernel of the quantifier elimination in [2, 3] (i.e., the construction of ψ_{ext} for handling negative information), but has a simpler overall structure since it avoids additional ballast. Examples are the proof of the closure properties of prime formulae under conjunction and existential quantification (which are not difficult but somewhat tedious) and the calculation of a finite set of path constraints describing negative information (in general, there may be an infinite set of path constraints entailed by a prime formula). The use of path constraints is a technical tool in [2, 3], whereas their use in the proof described here corresponds in a natural way to chains of relations. On the other hand, the quantifier elimination in [2, 3] serves for a concrete decision method.

We want to thank Lawrence Moss, who initiated this research by suggesting the use of Ehrenfeucht-Fraïssé Games for proving completeness of the theory FT. We acknowledge discussions with Hubert Comon, Hans Leiss, Andreas Podelski, and Gert Smolka.

The first author has been supported by the Bundesminister für Bildung, Wissenschaft, Forschung und Technologie (VERBMOBIL, 01 IV 101 K/1). The second author has been supported by the Bundesminister für Bildung, Wissenschaft, Forschung und Technologie (Hydra, ITW 9105), the Esprit Working Group CCL (EP 6028), and the Esprit Basic Research Project ACCLAIM (EP 6195).

References

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

[2] Rolf Backofen. *Expressivity and Decidability of First-order Languages over Feature Trees.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany, 1994.

[3] Rolf Backofen. A complete axiomatization of a theory with feature and arity constraints. *The Journal of Logic Programming*, 24(1&2):37–71, July/August 1995.

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

[5] Alain 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.

[6] Hubert Comon and Catherine Delor. Equational formulae with membership constraints. *Information and Computation*, 112(2):167–216, August 1994.

[7] Hubert Comon and Pierre Lescanne. Equational problems and disunification. *Journal of Symbolic Computation*, 7(3,4):371–425, 1989.

[8] Hubert Comon and Ralf Treinen. The first order theory of lexicographic path orderings is undecidable. *Theoretical Computer Science*, 176, April 1997. To Appear.

[9] Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25(2):95–169, 1983.

[10] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B - Formal Models and Semantics, chapter 6, pages 243–320. Elsevier Science Publishers and The MIT Press, 1990.

[11] Andrzej Ehrenfeucht. An application of games to the completeness problem for formalized theories. *Fundamenta Mathematicae*, 49:129–141, 1961.

[12] Roland Fraïssé. Sur les classifications des systems de relations. *Publications Sci. de l'Université d'Alger*, I:35–182, June 1954.

[13] W. Thomas H.D. Ebbinghaus, J. Flum. *Mathematical Logic.* Springer-Verlag, 1984.

[14] Wilfrid Hodges. *Model Theory.* Encyclopedia of Mathematics and its Applications 42. Cambridge University Press, 1993.

[15] Joxan Jaffar and Jean-Louis Lassez. Constraint logic programming. In *Proceedings of the 14th ACM Conference on Principles of Programming Languages*, pages 111–119, Munich, Germany, January 1987. ACM.

[16] Angus Macintyre. Model completeness. In Jon Barwise, editor, *Handbook of Mathematical Logic*, Studies in Logic and the Foundations of Mathematics, vol. 90, chapter A.4, pages 139–180. North-Holland, 1977.

[17] Michael J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite trees. In *Proceedings of the Third Annual Symposium on Logic in Computer Science*, pages 348–357. IEEE Computer Society, 1988.

[18] J. A. Makowsky. Model theory and computer science: An appetizer. In S. Abramsky, Dov M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic on Computer Science*, volume 1, pages 763–814. Clarendon Press, Oxford, 1992.

[19] Alberto Martelli and Ugo Montanari. An efficient unification algorithm. *ACM Transactions on Programming Languages and Systems*, 4(2):258–282, April 1982.

[20] Abraham Robinson. *Introduction to Model Theory and to the Metamathematics of Algebra.* Studies in Logic and the Foundations of Mathematics. North-Holland, 2nd edition, 1974.

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

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

[23] Ralf Treinen. Feature trees over arbitrary structures. In Patrick Blackburn and Maarten de Rijke, editors, *Specifying Syntactic Structures*, Studies in Logic, Language and Information. CSLI Publications. To appear. Collection of selected papers from the Workshop on Logic, Structures and Syntax, Amsterdam, September 1994.

[24] Ralf Treinen. Feature constraints with first-class features. In Andrzej M. Borzyszkowski and Stefan Sokołowski, editors, *Mathematical Foundations of Computer Science 1993*, Lecture Notes in Computer Science, vol. 711, pages 734–743. Springer-Verlag, 30 August–3 September 1993.

[25] Sergei Vorobyov. Theory of finite trees revisited: Application of model-theoretic algebra. Report CRIN 94-R-135, CRIN, Nancy, France, October 1994.