A Complete Axiomatization of a Theory with Feature and Arity Constraints *

ROLF BACKOFEN

▷ CFT is a recent constraint system providing records as a logical data structure for logic programming and for natural language processing. It combines the rational tree system as defined for logic programming with the feature tree system as used in natural language processing.
The formulae considered in this paper are all first-order-logic formulae over a signature of binary and unary predicates called features and arities, respectively. We establish the theory CFT by means of seven axiom schemes and show its completeness.
Our completeness proof exhibits a terminating simplification system deciding validity and satisfiability of possibly quantified record descriptions. ◁

*The research reported in this paper has been supported by the Bundesminister für Forschung und Technologie under contract 01 IV 101 K/1 (VERBMOBIL).

Address correspondence to Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany. E-mail: backofen@dfki.uni-sb.de

THE JOURNAL OF LOGIC PROGRAMMING

655 Avenue of the Americas, New York, NY 10010 0743-1066/93/$3.50

1. Introduction

Records are an important data structure in programming languages. They appeared first with imperative languages such as ALGOL 68 and Pascal, but are now also present in modern functional languages such as SML. Variants of records have also been employed in logic programming and in computational linguistics, but in different ways.

In logic programming, first order terms are used as restricted means for describing a special kind of records. There, records have fixed arities, and attributes are identified by their position in the first order term. The functions used in the first order terms are extensional, i.e., two terms $f(t_1, \ldots, t_n)$ and $g(t'_1, \ldots, t'_m)$ are assumed to be equal if and only if $f = g$, $n = m$ and $t_i = t'_i$ for every $i \in 1 \ldots n$. Hence, two records are identical if they have the same set of attributes and identical values under each attribute.

Another way of describing records is by feature descriptions, which are common in the area of computational linguistics. Here, the attributes of a record are modeled by functional, binary relations called features. This implies that the attributes are identified by name instead of by position. Furthermore, feature descriptions do not fix an arity for the record they describe. Thus, additional attributes can always be added to a feature description without making it inconsistent. This allows for great flexibility, since it is possible to describe only some selected attributes of a record without specifying the others (and even without specifying which other attributes must exists). This is not the case if we model records by first order terms. On the other hand, it is impossible (under an infinite signature, which we consider here) to state in a feature descriptions which features are missing, since the arity is not fixed.

In this paper, we consider the first order theory CFT, which was introduced in [ST94a]. CFT combines feature descriptions with the expressivity of first order terms. In this theory, record descriptions are first order formulae interpreted over first order structures.

There are two complementary ways of specifying a theory for records: either by explicitly constructing a standard model and taking all sentences valid in it, or by stating axioms and proving their consistency. Both approaches to fixing a theory for records have their advantages. The construction of a standard model provides for a clear intuition and yields a complete theory (i.e., if ϕ is a closed record description, then either ϕ or $\neg\phi$ is a consequence of the theory). The presentation of a recursively enumerable axiomatization has the advantage that we inherit from predicate logic a sound and complete deduction system for valid record descriptions. Note that all models of a complete theory are elementarily equivalent.

The ideal is to specify a theory for records by both a standard model and a corresponding recursively enumerable axiomatization. The existence of such a double characterization, however, is by no means obvious, since it implies that the theory is decidable. In [ST94a], both approaches have been exemplified. A standard model, namely the model of feature trees, has been presented, together with a first order theory CFT based on a set of axioms. It has been conjectured that CFT is a complete theory. We will show in this paper that CFT is in fact a complete axiomatization of the algebra of feature trees. Furthermore, it has been shown in [ST94a] that the theory is complete for $\exists^*$-sentences.

Our proof of CFT's completeness will exhibit a simplification algorithm that computes for every feature description an equivalent solved form from which the solutions of the description can be read off easily. For a closed feature description the solved form is either $\top$ (which means that the description is valid) or $\bot$ (which means that the description is invalid). For a feature description with free variables the solved form is $\bot$ if and only if the description is unsatisfiable. We do not know whether our simplification algorithm can be made feasible, nor do we know its worst-case complexity.

Note that the notion of completeness considered in this paper is different from the notion of completeness considered in related work by Kasper and Rounds [KC90] and Moss [Mos92]. These authors study logical equivalence for rooted and quantifier-free feature descriptions (called feature terms in [Smo92, BBN$^+$93]) and give complete equational axiomatizations of the respective congruence relations. In contrast, we are concerned with a much larger class of possibly quantified feature descriptions. Moreover, exploiting the power of predicate logic, we are not committed to any particular model or any particular deductive system, but instead prove a result that implies that any complete proof system for Predicate Logic will be complete for proving equivalence of feature descriptions with respect to any model of our feature theory.

1.1. Records as Feature Trees

Records are described in CFT in the tradition of feature descriptions, which have a long history. They originated in the late seventies in the framework of so-called unification grammars [Kay79, KB82], a by now very popular family of declarative grammar formalisms for the description and processing of natural language. Feature descriptions have been proposed in various forms with various formalizations [AK84, AK86, KR86, RK86, KC90, Joh88, Joh91, Smo92, BBN$^+$93, Bac94b, Car92, Mos92]. More recently, the use of feature descriptions in logic programming has been advocated and studied [AKN86, AKN89, AKP93, AKPS94, ST94a].

The work presented here follows the logical approach as introduced by [Smo92], where feature descriptions are first order formulae. Consider a typical feature description written in matrix notation

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

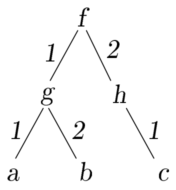
This may be read as saying that x is a woman whose father is an engineer, whose husband is a painter, and whose father and husband are of the same age. Written in plain first-order syntax we obtain the less transparent formula

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

As descriptional primitives, the feature description contains the atomic formulae $f(x, y)$ for feature selection (which we will henceforth write in infix notation) and $A(x)$ for sort membership. In addition, CFT offers for every finite set of features F a unary predicate xF (written in postfix notation) stating that the only features defined on x are those listed in F (see section 1.2).

In the standard model of CFT, records are modeled by feature trees. A feature tree (see Figure 1) is a (possibly infinite) tree whose edges are labeled with features, and whose nodes are labeled with sorts. As one would expect, the labeling with features must be functional, that is, the direct subtrees of a feature tree must be uniquely determined by the features of the edges leading to them. Feature trees without subtrees model atomic values (e.g., numbers). Feature trees may be finite or infinite, where infinite feature trees provide for the convenient representation of cyclic data structures. The last example in Figure 1 gives a finite graph representation of an infinite feature tree, which may arise as the syntactic representation of the recursive type equation $\mathit{nat} = \mathit{0} + s(\mathit{nat})$.

A first-order ground term, say $f(g(a, b), h(c))$, can be seen as a feature tree whose nodes are labeled with function symbols and whose arcs are labeled with numbers:



Thus the trees corresponding to first-order terms are in fact feature trees observing certain restrictions (e.g., the features departing from a node must be consecutive positive integers).

The standard model of CFT is the first order structure $\mathfrak{I}$, whose universe is the set of all feature trees (over a given signature), and whose interpretation of the atomic formulae is defined as follows:

- Every sort symbol A is taken as a unary predicate, where a *sort constraint* $A(x)$ holds if and only if the root of the tree x is labeled with A.
- Every feature symbol f is taken as a binary predicate, where a *feature constraint* xfy holds if and only if the tree x has the direct subtree y at feature f.
- Every finite set F of features is taken as a unary predicate, where an *arity constraint* xF holds if and only if the tree x has direct subtrees exactly at the features appearing in F.

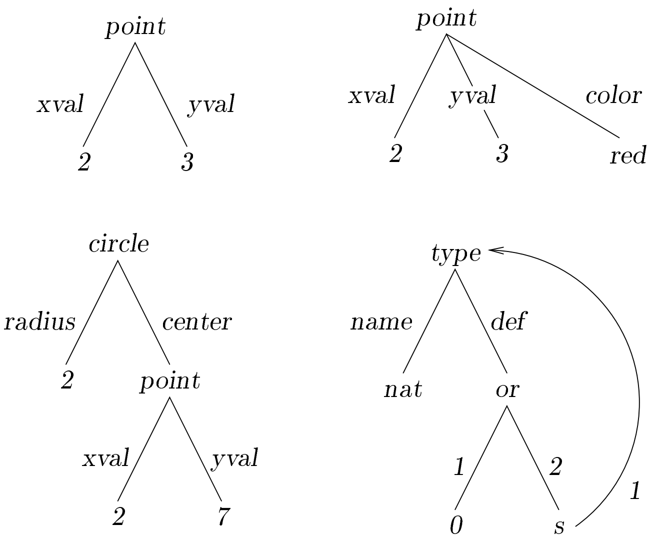


Figure 1. Examples of Feature Trees.

The descriptions or constraints of CFT are now exactly the first order formulae obtained from the primitive forms specified above, where we include equations $x \doteq y$ between variables.

1.2. Expressivity of CFT

CFT can be seen as the minimal combination of Colmerauer's rational tree system RT [Col82, Col84] with the feature tree system FT [BS93]. For this reason, CFT is a promising constraint system for use both in logic programming and computational linguistics. Note that we are assuming an infinite signature for CFT. In the case of finite signatures, all three languages have the same expressivity.

The difference between FT and CFT is that CFT additionally has arity constraints. This implies that every FT formula is also a CFT formula. However, we have to extend the completeness proof for FT (see [BS93]) in a non-trivial way, since we have to handle additional equations imposed by the arity constraints. E.g.

$$Ax \wedge Ay \wedge x\{f\} \wedge xfx \wedge y\{f\} \wedge yfy \models_{\mathrm{CFT}} x \doteq y$$

holds in CFT stating that there is only one solution for the formula $Ax \wedge x\{f\} \wedge xfx$. In FT it is not possible to identify one element of the domain by a formula. Thus, CFT requires records to be extensional (i.e., two records are identical if the have the same sort, the same set of attributes and the identical values under the corresponding attributes). Note that this property could not be guaranteed using the language of FT (i.e., FT has non-extensional models). The integration of extensionality into feature descriptions was considered in [Car92]. Since the underlying feature logic was too weak to express extensionality, the notion of extensional types was introduced. But as Carpenter observed, the conditions stated for extensionality are too weak to identify feature descriptions which describe infinite trees. In CFT, those descriptions can be identified (as the above example shows). Rounds introduced in [Rou92] different concepts of extensionality. Using his classification, the standard model of CFT is strongly extensional.

For the comparison of RT and CFT, [ST94a] have presented a translation of RT-formulae into CFT-formulae that preserves validity, i.e., there exists for every formula σ in RT a corresponding formula in γ in CFT such that σ is valid in the standard model of RT if and only if γ is valid in the standard model of CFT. The following examples are taken from [ST94a]. Given an RT formula σ

$$x = point(y, z),$$

translating σ yields the CFT formula

$$x : point \wedge x\{1, 2\} \wedge x \; 1 \; y \wedge x \; 2 \; z.$$

But again, CFT has more expressive power than RT. It is possible to express within CFT that a record has some feature without saying anything about other features. A description of the form

$$x \; color \; y$$

just tells that x has a color feature, but it does not disallow other features such as shape, size, position or anything else. In the case of a finite signature, this could be defined by a disjunction of the form

$$x = circle(\ldots, y, \ldots) \vee x = triangle(\ldots, y, \ldots) \vee \ldots$$

enumerating all constructors for which a *color* attribute is appropriate. But the computational behavior of this disjunction is much worse than that of the single constraint $x \; color \; y$. In the case of an infinite signature (which we consider here), such a single feature constraint is not definable in RT (since it would correspond to an infinite disjunction).

1.3. Quantifier Elimination

The completeness proof uses a version of the standard method of quantifier elimination which was introduced by [Mah88]. For this method, it is necessary to find a class of formulae (here called prime formulae) satisfying certain properties. Quantifier elimination is then performed with respect to this class of formulae, i.e., every formula ϕ can be transformed into an equivalent Boolean combination of prime formulae. In our case, the set of prime formulae is the set of existentially quantified solved formulae. As defined in [ST94a], a solved formula is a normal form of conjunction of atomic formulae having certain desirable properties. In particular, it is always satisfiable.

The first property we need for prime formulae is that every closed prime formula is valid in CFT, which is a trivial consequence of the axioms. The second property is that the class of prime formulae is closed under conjunction and existential quantification. Again, this is easy to show in our case.

The third (and difficult to prove) property is that the following two equivalences are valid in CFT: (1) Given prime formulae $\beta, \beta_1, \ldots, \beta_n$, then

$$\exists X(\beta \wedge \bigwedge_{i=1}^{n} \neg\beta_i) \models\!\!\!\dashv \bigwedge_{i=1}^{n} \exists X(\beta \wedge \neg\beta_i), \tag{1}$$

and (2) there exists for all prime formulae β, β' a Boolean combination of prime formulae δ such that

$$\exists X(\beta \wedge \neg\beta') \models\!\!\!\dashv \delta, \tag{2}$$

where X is a set of variables. These schemes can now be used for a system transforming every formula in the language of CFT into a Boolean combination of prime formulae. If the input formula is closed, the result will also be closed. Since every closed prime formula is valid in CFT, we know that the result of transforming a closed formula ϕ reduces either to $\top$ or to $\bot$. In the first case, ϕ is valid in CFT. Otherwise, $\neg\phi$ is valid in CFT.

The transformation works as follows. An invariant of the transformation is that both the input and output formulae of a single transformation step are of the form

$$Q_1 \ldots Q_n \gamma$$

where $Q_1 \ldots Q_n$ are quantifiers and γ is a Boolean combination of prime formulae. A single transformation step now eliminates the innermost quantifier.

If the innermost quantifier Q_n is an existential one, then we first transform γ into disjunctive normal form, treating the prime formulae as atoms. Then we can distribute the existential quantifier over the disjuncts, yielding a disjunction of formulae of the form

$$\exists x(\bigwedge_{i=1}^{n} \beta_i \wedge \bigwedge_{j=1}^{k} \neg\beta'_j)$$

where all β_i and β'_j are prime formulae. Since prime formulae are closed under conjunction, we can assume that the disjuncts are of the form

$$\exists x(\beta \wedge \bigwedge_{j=1}^{k} \neg\beta'_j.)$$

Now we can apply scheme 1, transforming each disjunct into a conjunction of the form

$$\bigwedge_{j=1}^{k} \exists x(\beta \wedge \neg\beta'_j),$$

which can be transformed into a Boolean combination of prime formulae δ by scheme 2. All together, we have eliminated the innermost existential quantifier.

If, on the other hand, the innermost quantifier is a universal one, we substitute $\neg\exists x \neg\gamma$ for $\forall x\gamma$. Then we put $\neg\gamma$ into its negation normal form γ', treating the prime formulae as atoms. Now applying the elimination method as described for existential quantification on $\exists x\gamma'$ yields a Boolean combination of prime formulae δ. Now putting $\neg\delta$ into negation normal form again (treating prime formulae as atoms) yields a Boolean combination of prime formulae that is equivalent to $\forall x\gamma$.

We have described the elimination of a single quantifier. But as the schemes 1 and 2 use an existential quantification over a whole set of variables X, the elimination methods apply also to a whole set of quantifiers of the same type (i.e., if we start with a formula $Q_1 \dots Q_k \dots Q_{k+n}\gamma$ where $Q_k \dots Q_{k+n}$ are either of the form $\exists x_k \dots \exists x_{k+n}$ or of the form $\forall x_k \dots \forall x_{k+n}$, then we can eliminate $Q_k \dots Q_{k+n}$ in one step.

1.4. Related Work

A complete axiomatization for RT over an infinite signature is given in [Mah88], and for FT in [BS93]. All proofs have the same overall structure using a quantifier elimination method as described in the last section, but differ in the way the schemes 1 and 2 are proved. Maher's proof heavily depends on the structure of first order terms in using substitutions. This is not applicable in our case since we are using a purely relational language. A complete axiomatization for RT over a finite signature is given in [Mah88, CL89]. An extension of the language of CFT, where features are first class values, was considered in [Tre93]. There it was shown that the full first order theory of the feature tree model over this language is undecidable.

When comparing the completeness proofs for FT and CFT, additional problems arise in CFT in the handling of inequations. Manipulation of inequations is needed for the proofs of the schemes in 1 and in 2. To give a concrete example, consider the FT-formula $\exists x(\beta \wedge \neg\beta')$ with

$$\begin{aligned} \beta &:= \exists x_1, x_2(xfx_1 \wedge xgx_2 \wedge Ax_1 \wedge Ax_2) \\ \beta' &:= \exists y(xfy \wedge xgy), \end{aligned}$$

which is an instance of left hand side of scheme 2. In the standard model of FT (which is the same as for CFT), there always exists a valuation for x satisfying β such that the values under the features f and g are different. This implies that the equivalence

$$\exists x(\beta \wedge \neg\beta') \models\!\!\!\!\!\!\!\dashv \exists x\beta \tag{3}$$

is valid in FT. Hence, $\exists x\beta$ is the Boolean combination of prime formulae as required by scheme 2. Roughly speaking, this equivalence is proven by extending β to a prime formula β_{ext} which makes x_1 and x_2 different, e.g. the prime formula

$$\exists x_1, x_2, x_1', x_2' \left(\begin{array}{c} xfx_1 \wedge xgx_2 \wedge Ax_1 \wedge Ax_2 \wedge \\ x_1fx_1' \wedge Bx_1' \wedge x_2fx_2' \wedge B'x_2' \end{array} \right)$$

with B, B' being two different sort symbols. Clearly, $\exists x \beta_{ext}$ is satisfiable in FT. Hence, there exists in every model of FT a valuation for x satisfying β_{ext}. Since this valuation must also satisfy β and cannot satisfy β', this shows the equivalence in 3.

Therefore, it is necessary in the proof to characterize the variables for which such additional constraints must be added. In the case of FT this is easy; they are exactly the variables where an additional equation is added when applying the solved form algorithm on

$$\beta \wedge \beta' \quad = \quad \exists x_1, x_2, y(xfx_1 \wedge xgx_2 \wedge Ax_1 \wedge Ax_2 \wedge xfy \wedge xgy).$$

But in the case of CFT, it can be more complex, since variables can be determined using the arity constraints. Consider the following two formulae β_1 and β_2:

$$\begin{array}{rcl}\beta_1 & = & \exists x_1, x_2, x_3, x_4 \left(\begin{array}{l} xfx_1 \wedge xgx_2 \wedge \\ Ax_1 \wedge x_1\{f\} \wedge x_1fx_3 \wedge \\ Ax_2 \wedge x_2\{f\} \wedge x_2fx_4 \end{array}\right) \\ \beta_2 & = & \exists x_1, x_2 \left(\begin{array}{l} xfx_1 \wedge xgx_2 \wedge \\ Ax_1 \wedge x_1\{f\} \wedge x_1fx_1 \wedge \\ Ax_2 \wedge x_2\{f\} \wedge x_2fx_2 \end{array}\right)\end{array}$$

We let β' again be $\exists y(xfy \wedge xgy)$. Although in both cases an additional equation $x_1 \doteq x_2$ is added when solving $\beta_1 \wedge \beta'$ or $\beta_2 \wedge \beta'$, the equivalence $\exists x(\beta_1 \wedge \neg\beta') \models\!\!\!\!\!\!\dashv \exists x \beta_1$ is valid in CFT, whereas the equivalence $\exists x(\beta_2 \wedge \neg\beta') \models\!\!\!\!\!\!\dashv \exists x \beta_2$ is not.

The work done in this paper can be seen as an extension of [ST94a]. There, two decision procedures for fragments of CFT are presented. The first procedure tests satisfiability, which is the same as testing validity of the positive existential fragment of CFT. This is used in our proof for calculating a solved form for the conjunction of prime formulae. The second algorithm checks entailment or disentailment of one prime formula by another. A formula γ entails a formula γ' in CFT (written $\gamma \models_{\mathrm{CFT}} \gamma'$) iff in every model $\mathcal{A}$ of CFT and for every every valuation α in $\mathcal{A}$, $\mathcal{A}, \alpha \models \gamma$ implies $\mathcal{A}, \alpha \models \gamma'$. Since γ entails γ' if and only if

$$\mathrm{CFT} \models \forall X(\gamma \rightarrow \gamma')$$

where X is the set of free variables of γ and γ', one can check entailment of arbitrary formulae in CFT. The algorithm presented in [ST94a] applies only to existentially quantified conjunctions of atomic constraints. Thus, the quantifier elimination is a real extension of the work done there, since it applies to arbitrary CFT-formulae. One of the simplest examples that is not covered by [ST94a] is to test the validity of the entailment

$$\exists x_1, x_2(xfx_1 \wedge Ax_1 \wedge xgx_2 \wedge Bx_2) \models_{\mathrm{CFT}} \exists y_1, y_2(xfy_1 \wedge xgy_2 \wedge y_1 \neq y_2).$$

The use of such negated equations has e.g. been considered in [Car92]. Note that for testing entailment of existential quantified conjunction of atomic constraints, the algorithm as described in [ST94a] is more useful than using quantifier elimination, since it is optimized for this purpose.

Another completeness proof for CFT is presented in [BT94], where Ehrenfeucht-Fraisse games are used. This method is based on semantics in showing that all models of CFT are elementarily equivalent (i.e., make the same sentences valid), which immediately implies that CFT is complete. This yields a trivial decision method for CFT-sentences by enumerating all consequences of CFT. Given an arbitrary sentence ϕ, the enumeration will produce either ϕ or $\neg\phi$ since CFT is complete. On the other hand, this paper employs a proof theoretic method in showing explicitly that for every sentence ϕ, either ϕ or $\neg\phi$ is valid in CFT. Both methods have their merits. The proof in [BT94] is shorter (though similar problems arise in handling inequations), while the proof in this paper presents a decision method for validity.

1.5. Organization of the Paper

Section 2 recalls the necessary notions and notations from Predicate Logic. Section 3 defines the standard model for CFT. Section 4 defines the theory CFT by means of seven axiom schemes. Section 5 establishes the overall structure of the completeness proof by means of a lemma. Section 6 studies quantifier-free conjunctive formulae, gives a solved form, and introduces path constraints. Section 7 defines congruences and normalizers. Section 8 studies the properties of so-called prime formulae, which are the basic building blocks of the solved form for general feature constraints. Section 9 presents the quantifier elimination lemmas and completes the proof of completeness. We present in this section a concrete example for testing validity of some formula. Furthermore, we prove that FT is really less expressive than CFT.

Technical Note Although we have introduced CFT as a constraint language that allows for sort constraints of the form Ax, we will for the sake of flexibility replace these constraints by a new kind of constraints. In order to build the new constraints, we must introduce certain constants (or atoms). The intended meaning of constants is that they represent distinct elements of the domain that have no features defined on them. We can now easily simulate sort constraints using constants; we use a constant symbol for every sort symbol and add a feature *sort* to hold it. A sort constraint Ax can than be represented by the constraint

$$x \ sort \ A.$$

For the sake of clarity, we refer in the following to the new language as CFT′, and to the language originally introduced [ST94a] as CFT.

The additional flexibility can be seen through the following example. We can express the notion that two objects x and y have the same sort by the formula

$$\exists z(x \ sort \ z \wedge y \ sort \ z),$$

which is impossible when using sort constraints. Clearly, the completeness proof for CFT′ can easily be adopted for the original language CFT.

2. Preliminaries

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

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

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

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

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

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

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

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

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

3. The Feature Tree Structure

In this section we establish the standard model for CFT$'$.

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

A *feature tree* is a pair $\sigma = (D, \lambda)$, where D is a tree domain and λ is a partial function $\lambda\colon \mathrm{FEA}^\star \to \mathrm{CON}$ satisfying

- $\mathrm{dom}(\lambda) \subseteq D$,
- if $\lambda(p)$ is defined for some $p \in D$, then $pq \notin D$ for every nonempty path q.

The paths in D represent the nodes of the tree; the empty path represents its root; λ represents the leaves of the tree that are constants. A feature tree $\sigma = (D, \lambda)$ is called *finite [infinite]* if its domain D is finite [infinite]. The letters σ and τ will always denote feature trees.

The *subtree* $p\sigma$ of a feature tree $\sigma = (D, \lambda)$ at a path $p \in D$ is the feature tree (D', λ') defined by (in relational notation)

$$D' = \{q \mid pq \in D\} \quad \text{and} \quad \lambda' = \{(q, a) \mid (pq, a) \in \lambda\}.$$

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

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

The *feature tree structure* $\mathfrak{I}$ is the $\mathrm{CON} \uplus \mathrm{FEA}$-structure defined as follows:

- the universe of $\mathfrak{I}$ is the set of all feature trees
- $a^{\mathfrak{I}} = (\{\epsilon\}, \{(\epsilon, a)\})$ for every constant symbol $a \in \mathrm{CON}$.
- $(\sigma, \tau) \in f^{\mathfrak{I}}$ iff $\tau = f\sigma$ (i.e., τ is the subtree of σ at f),
- $\sigma = (D, \lambda) \in F^{\mathfrak{I}}$ iff $\lambda(\epsilon)$ is undefined and $D \cap \mathrm{FEA} = F$ (i.e. σ is not the interpretation of a constant and has exactly the features in F defined)

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

4. The Axioms

The first six axiom schemes of the theory CFT′ are:

(Ax1)	$\tilde{\forall}(xfy \wedge xfz \rightarrow y \doteq z)$	for every feature f.
(Ax2)	$\tilde{\forall}(cfx \rightarrow \perp)$	for all constants c.
(Ax3)	$c_1 \not\doteq c_2$	if c_1 and c_s are different constants
(Ax4)	$\tilde{\forall}(xF \wedge xfy \rightarrow \perp)$	if $f \notin F$.
(Ax5)	$cF \rightarrow \perp$	for every constant c and arity F.
(Ax6)	$\tilde{\forall}(xF \rightarrow \exists y(xfy))$	if $f \in F$ and $x \neq y$.

The last three axiom schemes handle the arity constraints. They guarantee that if x has arity F, then exactly the features $f \in F$ are defined on x.

In order to achieve a complete theory, we must add an axiom scheme that is similar to axiom (Ax3) of the theory FT as presented in [BS93]. In contrast to FT, it is not enough to guarantee that solved forms are consistent in the intended models. Consider the formula

$$x\{f\} \wedge xfx.$$

Then there exists exactly one element of $\mathfrak{R}$ and $\mathfrak{I}$ that satisfies this description. The uniqueness of the solution of such descriptions must also be expressed in the axioms. Note that it is not possible to fix one element of the domain in the theory FT since we cannot restrict the arities of the variables in FT. The axiom scheme that guarantees both the existence and under certain conditions also the uniqueness of solutions of solved forms was first introduced by [ST94a]. They also introduced a complete axiomatization for CFT in this paper without actually proving completeness. Before stating the required axiom scheme, we will recall the important notion of a determinant as presented in [ST94a].

Definition 4.1. [Basic Constraint] A *basic constraint* is either $\perp$ or a possibly empty conjunction of atomic formulae.

Note that $\top$ is a basic constraint since $\top$ is the empty conjunction.

Definition 4.2. [Determinant] A *determinant for x* is a formula of the form

$$x\{f_1, \ldots, f_n\} \wedge xf_1t_1 \wedge \ldots \wedge xf_nt_n,$$

where each t_i is a variable or constant. We will write the above formula for convenience as

$$x \doteq (f_1 : t_1, \ldots, f_n : t_n).$$

Given a basic constraint ϕ, we say that x is *determined in* ϕ if ϕ contains a determinant for x. A *determinant for* pairwise distinct variables $x_1, \ldots, x_n$ is a conjunction

$$x_1 \doteq D_1 \wedge \ldots \wedge x_n \doteq D_n,$$

where $D_1, \ldots, D_n$ are determinants for $x_1, \ldots, x_n$. For a basic constraint ϕ we define $\mathcal{D}(\phi)$ to be the set of variables that are determined in ϕ.

The variables in $\mathcal{V}(\delta) \backslash \mathcal{D}(\delta)$ are called the *parameters* of δ.

For the remaining axiom scheme we must introduce a new existential quantifier $\exists! x\phi$. This quantifier is an abbreviation for

$$\exists x\phi \wedge \forall x, y(\phi \wedge \phi[x \leftarrow y] \rightarrow x \doteq y).$$

For a set of variables X the quantifier $\exists! X\phi$ is defined as usual. Now we can define the last axiom scheme as introduced by [ST94a], which states that for every valuation of the parameters of a determinant δ there is exactly one valuation for the variables determined by δ:

(Ax7) $\quad \tilde{\forall}(\exists! D(\delta)\delta) \quad$ if δ is a determinant.

An example of an instance of scheme (Ax7) is

$$\forall y, z, w \exists! x, u, v \left(\begin{array}{ccl} x & \doteq & (f\!: u \;\; g\!: v) \\ u & \doteq & (h\!: x \;\; g\!: y \;\; f\!: z) \\ v & \doteq & (g\!: z \;\; h\!: w) \end{array} \right)$$

The theory CFT$'$ consists of the axiom schemes $(Ax1)$–$(Ax7)$.

Proposition 4.1. The structures $\mathfrak{I}$ and $\mathfrak{R}$ are models of CFT$'$.

PROOF. That the first six axioms schemes are satisfied is obvious. To show that $\mathfrak{I}$ and $\mathfrak{R}$ satisfies the last axiom scheme, one assumes arbitrary feature trees for the universally quantified variables and constructs feature trees for the existentially quantified variables. $\square$

5. Outline of the Completeness Proof

The completeness of CFT$'$ will be shown by exhibiting a simplification algorithm for CFT$'$. The following lemma gives the overall structure of the algorithms, which is the same as in Maher's [Mah88] completeness proof for the theory of constructor trees. The same structure was used in the completeness proof for FT (see [BS93]).

Lemma 5.1. Suppose there exists a set of prime formulae such that:

1. *every arity constraint xF, every feature constraint xft, and every equation $t \doteq t'$ with $t \neq t'$ is a prime formula*
2. *$\top$ is a prime formula, and there is no other closed prime formula*
3. *for every two prime formulae β and β' one can compute a formula δ that is either prime or $\perp$ and satisfies*

$$\beta \wedge \beta' \models\!\!\!\mid_{\mathrm{CFT}'} \delta \quad \text{and} \quad \mathcal{V}(\delta) \subseteq \mathcal{V}(\beta \wedge \beta')$$

4. *for every prime formula β and every variable x one can compute a prime formula β' such that*

$$\exists x\beta \models\!\!\!\mid_{\mathrm{CFT}'} \beta' \quad \text{and} \quad \mathcal{V}(\beta') \subseteq \mathcal{V}(\exists x\beta)$$

5. *if β, $\beta_1, \ldots, \beta_n$ are prime formulae, then*

$$\exists x(\beta \wedge \bigwedge_{i=1}^{n} \neg\beta_i) \models\!\!\!\mid_{\mathrm{CFT}'} \bigwedge_{i=1}^{n} \exists x(\beta \wedge \neg\beta_i)$$

6. *for every two prime formulae β, β' and every variable x one can compute a Boolean combination δ of prime formulae such that*

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

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

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

If ϕ is an atomic formula tF or tft', then ϕ is equivalent to $\perp$. If ϕ is an atomic formula xF, xft or $t \doteq t'$, then ϕ is either a prime formula, or ϕ is a trivial equation $t \doteq t$, in which case it is equivalent to the prime formula $\top$.

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

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

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

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

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

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

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

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

□

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

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

6. Solved Formulae and Path Constraints

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

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

Definition 6.1. [Solved Formula] A basic constraint γ is a *solved formula* if

1. no atomic formula occurs twice in γ;
2. an equation $x \doteq t$ appears in γ if and only if γ eliminates x;
3. if $xft \in \gamma$ and $xft' \in \gamma$, then $t = t'$;
4. if $xF, xG \in \gamma$, then $F = G$;
5. if $xF \in \gamma$ and $f \notin F$, then $xfy \notin \gamma$;
6. γ does not contain an atomic formula of the form $c \doteq t$, cF or cft.

Every solved form γ has a unique decomposition $\gamma = \gamma_N \wedge \gamma_G$ into a possibly empty conjunction γ_N of equations "$x \doteq y$" and a possibly empty conjunction γ_G of constraints "xF" and feature constraints "xfy". We call γ_N the *normalizer* and γ_G the *graph* of γ.

Proposition 6.1. Let γ be the graph of a solved formula. A variable x is said to be constrained *in γ if γ contains a constraint xft or xF. Let $\mathcal{C}(\gamma)$ be the set of all variables constrained in γ. Then*

$$\mathrm{CFT}' \models \tilde{\forall} \exists \mathcal{C}(\gamma)\gamma$$

PROOF. We will extend γ to a determinant δ with $\mathcal{D}(\delta) = \mathcal{C}(\gamma)$.

For every $x \in \mathcal{C}(\gamma)$ and $x \notin \mathcal{D}(\gamma)$ let F_x be a set of features such that F_x contains exactly the features f with $xfy \in \gamma$ and let δ be defined as

$$\delta = \gamma \cup \{xF_x \mid x \in \mathcal{C}(\gamma)\}$$

By definition, δ is a determinant. By axiom $(Ax7)$ we know that

$$\mathrm{CFT}' \models \tilde{\forall} \exists \mathcal{D}(\delta)\delta$$

which proves $\mathrm{CFT}' \models \tilde{\forall} \exists \mathcal{C}(\gamma)\gamma$. □

The letter γ always denotes a solved form. We will see that every basic constraint is equivalent in CFT′ to either $\perp$ or a solved formula.

Figure 1 shows the so-called *basic simplification rules*. By $\phi[x \leftarrow y]$ we denote the formula that is obtained from ϕ by replacing every occurrence of x with y. We say that a formula ϕ ***simplifies to*** a formula ψ by a simplification rule ρ if $\frac{\phi}{\psi}$ is an instance of ρ. We say that a basic constraint ϕ ***simplifies to*** a basic constraint ψ if either $\phi = \psi$ or ϕ simplifies to ψ in finitely many steps each licensed by one of the basic simplification rules in Figure 1.

Note that the basic simplification rules (Cong), (CFCl), (CCl), (FArCl) and (CArCl) correspond to the axioms schemes (Ax1), (Ax2), (Ax3), (Ax4) and (Ax5), respectively. The rule (ArCl) follows from (Ax4) and (Ax6). Thus, they are equivalence transformation with respect to CFT′. The remaining simplification rules are equivalence transformations in general.

Proposition 6.2. The basic simplification rules are terminating and perform equivalence transformations with respect to CFT′*. Moreover, a basic constraint $\phi \neq \perp$ is solved if and only if no basic simplification rule applies to it.*

PROOF. To see that the basic simplification rules are terminating, observe that no rule adds a new variable and that every rule preserves eliminated variables. Since rule (Elim) increases the number of eliminated variables, and the remaining rules obviously terminate, the entire system must terminate. The other claims are easy to verify. □

(Cong) $$\frac{xft_1 \wedge xft_2 \wedge \phi}{xft_1 \wedge t_1 \doteq t_2 \wedge \phi}$$

(Elim) $$\frac{x \doteq t \wedge \phi}{x \doteq t \wedge \phi[x \leftarrow t]} \qquad x \in \mathcal{V}(\phi) \text{ and } x \neq y$$

(Triv) $$\frac{t \doteq t \wedge \phi}{\phi}$$

(Orient) $$\frac{c \doteq x}{x \doteq c}$$

(CFCl) $$\frac{cft}{\perp}$$

(CCl) $$\frac{c_1 \doteq c_2}{\perp} \qquad c_1 \neq c_1$$

(ArCl) $$\frac{xF \wedge xG \wedge \phi}{\perp} \qquad F \neq G$$

(FArCl) $$\frac{xfy \wedge xF \wedge \phi}{\perp} \qquad f \notin F$$

(CArCl) $$\frac{cF \wedge \phi}{\perp}$$

Figure 1. The basic simplification rules.

Proposition 6.3. Let ϕ be a basic constraint. Then one can compute a formula δ that is either solved or $\bot$ such that $\phi \models\!\!\!\dashv_{\mathrm{CFT}'} \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\phi)$.

PROOF. Follows from the preceding proposition and the fact that the basic simplification rules do not introduce new variables. □

We say that a basic constraint *clashes* if it can be reduced to $\bot$ with one of the clash rules (i.e., rules CFCl–CArCl), and we call a basic constraint *clash-free* if it does not clash.

In the quantifier elimination proofs to come it will be convenient to use so-called path constraints, which provide a flexible syntax for atomic formulae closed under conjunction and existential quantification. The notion of path constraints was introduced in [BS93]. We start by defining the denotation of a path.

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

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

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

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

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

Definition 6.2. [Path Constraints] Let p, q be paths, x, y be variables, F be an arity, and c be a constant symbol. Then *path constraints* are defined as follows:

$$\begin{array}{rcl}
\mathcal{A}, \alpha \models xpc & :\Longleftrightarrow & \alpha(x)\, p^{\mathcal{A}}\, c^{\mathcal{A}} \\
\mathcal{A}, \alpha \models xpy & :\Longleftrightarrow & \alpha(x)\, p^{\mathcal{A}}\, \alpha(y) \\
\mathcal{A}, \alpha \models xp\!\downarrow\! yq & :\Longleftrightarrow & \exists a \in |\mathcal{A}|\colon \;\; \alpha(x)\, p^{\mathcal{A}}\, a \;\wedge\; \alpha(y)\, q^{\mathcal{A}}\, a \\
\mathcal{A}, \alpha \models xpF & :\Longleftrightarrow & \exists a \in |\mathcal{A}|\colon \;\; \alpha(x)\, p^{\mathcal{A}}\, a \;\wedge\; a \in F^{\mathcal{A}}
\end{array}$$

A *proper path constraint* is a path constraint of the form "xpc", "xpF" or "$xp\!\downarrow\! yq$".

Note that path constraints xpy generalize feature constraints xfy. We use $xp\!\downarrow$ as an abbreviation for $xp\!\downarrow\! xp$. By definition, $xp\!\downarrow$ is satisfied by some valuation α into some structure $\mathcal{A}$ iff the path $p^{\mathcal{A}}$ is defined on $\alpha(x)$.

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

$$
\begin{array}{rcll}
x\varepsilon t & \models\!\!\!\dashv & x \doteq t & \\
xfpt & \models\!\!\!\dashv & \exists z(xfz \wedge zpt) & (z \neq x, t) \\
xp\downarrow yq & \models\!\!\!\dashv & \exists z(xpz \wedge yqz) & (z \neq x, y) \\
xpF & \models\!\!\!\dashv & \exists y(xpy \wedge yF) & (y \neq x).
\end{array}
$$

Definition 6.3. [Closure] The *closure* $[\gamma]$ *of a solved formula* γ is the closure of the atomic formulae occurring in γ with respect to the following deduction rules:

$$
\frac{}{x\varepsilon x} \qquad \frac{x \doteq t}{x\varepsilon t} \qquad \frac{xpy \quad yft}{xpft} \qquad \frac{xpt \quad yqt}{xp\downarrow yq} \qquad \frac{yF \quad xpy}{xpF}.
$$

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

Proposition 6.4. Let γ be a solved formula. Then:

if $\pi \in [\gamma]$, then $\gamma \models_{\mathrm{CFT}'} \pi$
2. *$x\varepsilon t \in [\gamma]$ iff $x = t$ or $x \doteq t \in \gamma$*
3. *$xft \in [\gamma]$ iff $xft \in \gamma$ or $\exists z$: $x \doteq z \in \gamma$ and $zft \in \gamma$*
4. *$xpft \in [\gamma]$ iff $\exists z$: $xpz \in [\gamma]$ and $zft \in \gamma$*
5. *if $p \neq \varepsilon$ and $xpt, xpt' \in [\gamma]$, then $t = t'$*
6. *it is decidable whether a path constraint is in $[\gamma]$.*

Proof. For the first claim one verifies the soundness of the deduction rules for path constraints. The verification of the other claims is straightforward. □

7. Congruences and Normalizers

Until now we have defined the notion of normalizer as being the set of equations attached to a solved formula. But for the completeness proof we need a more detailed definition of a normalizer. To this end we use the notion of congruence of a basic constraint. The definitions of congruence and normalizers are taken from [ST94a], where they were defined and used for the first time.

A *congruence of a basic constraint* ϕ is an equivalence relation $\approx$ between variables satisfying the following:

- $x \doteq y \in \phi$ implies $x \approx y$
- $xfy, x'fy' \in \phi$ and $x \approx x'$ implies $y \approx y'$.

It is easy to see that the set of congruences of a basic constraint is closed under intersection. Since the equivalence relation identifying all variables is a congruence for every basic constraint, we know that every basic constraint has a least congruence.

It will be convenient to represent congruences as idempotent substitutions.

Definition 7.1. A *normalizer of a congruence* $\approx$ is an idempotent substitution θ that maps variables to constants or variables and satisfies

$$\forall x, y : (\theta(x) = \theta(y) \Leftrightarrow x \approx y).$$

We say that substitution θ is finite if there are only finitely many variables x with $\theta(x) \neq x$. A finite substitution can be represented as

$$\bigwedge \{x \doteq \theta(y) \mid x \neq \theta(y)\}.$$

For convenience, we will simply use θ to denote this formula. Clearly, for every basic constraint ϕ and every substitution θ we have

$$\theta \wedge \phi \models\!\!\!\!\!\dashv \theta \wedge \theta\phi.$$

Definition 7.2. [Normalizer] A *normalizer* of a basic constraint ϕ is a normalizer of the least congruence of ϕ.

We will now recall some properties of normalizers that have been proven in [ST94a]. A *graph constraint* is a basic constraint that contains no equations. A graph constraint is called a *graph* if it is a solved formula.

Proposition 7.1. Let $\mathcal{A}$ be a model of CFT$'$, *ϕ a basic constraint and θ a normalizer of ϕ. Then ϕ is unsatisfiable in $\mathcal{A}$ if and only if $\theta\phi_G$ clashes, where ϕ_G is a graph constraint containing all constraints of ϕ of the form xF and xft.*

Proposition 7.2. Let $\gamma = \gamma_G \wedge \gamma_N$ be the normal form of a basic constraint ϕ that is normal with respect to the rules (Triv), (Cong), (Orient) and (Elim). Then $\theta = \gamma_N$ is a normalizer of ϕ satisfying $\gamma_G = \theta\gamma_G$ and $\mathcal{V}(\theta) \subseteq \mathcal{V}(\phi)$.

This proposition allows us to calculate normalizers. Note that this also implies that for a solved formula the two notions of normalizer as defined in Definition 1 and in Definition 2 agree.

A basic constraint ϕ is called *saturated* if for every arity constraint $xF \in \phi$ and every feature $f \in F$ there exists a feature constraint $xfy \in \phi$.

Lemma 7.1. Let γ be a saturated graph constraint and θ be a normalizer of some congruence of γ. If $\theta\gamma$ is clash-free and if $\mathcal{V}(\theta) \subseteq \mathcal{D}(\gamma)$, then

$$\gamma \models_{\mathrm{CFT}'} \theta.$$

For our purposes, we need two additional propositions.

Proposition 7.3. A substitution θ is a normalizer of some congruence of a graph constraint ϕ if and only if $\theta\phi$ is a graph.

Proposition 7.4. Let θ be the normalizer of some congruence of graph γ and let $\theta = \theta' \cup \theta''$ be a partition of θ. If θ' is a normalizer of some congruence of γ, then θ'' is a normalizer of some congruence of $\theta'\gamma$.

PROOF. Let γ, θ, θ', and θ'' be given as described. If θ' is a normalizer of some congruence of γ, we have to show that θ'' is a normalizer of some congruence of $\theta'\gamma$. Clearly, θ'' is an idempotent substitution. The congruence property follows from the last proposition together with the fact that $\theta''(\theta'(x)) = \theta(x)$. □

8. Prime Formulae

We now define a class of prime formula for the theory CFT$'$ that have the properties as required by lemma 5.1.

Definition 8.1. [Prime Formula] Let ϕ be a basic constraint. A formula $\beta = \exists X \phi$ is called *prime* if it satisfies the following conditions:

1. ϕ is solved and saturated;
2. X has no variable in common with the normalizer of ϕ;
3. for every $x \in X$ there is a variable $y \in \mathcal{V}(\beta)$ and a path p such that $ypx \in [\phi]$.

The letter β will always denote a prime formula. Note that $\top$ is the only closed prime formula.

Next we will show that every existentially quantified basic constraint can be transformed into $\perp$ or a prime formula. To do this we need the notion of decided variables, which are variables that are reachable from the free variables of a formula. We will show that every existentially quantified formula is equivalent to the set of constraints on the decided variables. For convenience, we will use a slightly generalized notion of decidedness which is more appropriate for the proofs to come.

Definition 8.2. [Decided Variables] Let γ be some solved formula, and let $\psi = \exists X \gamma$. A variable $x \in \mathcal{V}(\gamma)$ is said to be *explicitly decided in ψ* if there is a variable y free in ψ and a path p such that

$$ypx \in [\gamma].$$

A variable $x \in \mathcal{V}(\gamma)$ is called *implicitly decided in* ψ if γ contains a determinant D for x where each parameter of D is explicitly decided in ψ. We say that $x \in \mathcal{V}(\gamma)$ is *decided in* ψ if there is a z with $x \varepsilon z \in [\gamma]$ and z is explicitly or implicitly decided in ψ.

We say that a variable is undecided if it is not decided. The set of decided variables of a formula ψ will be denoted by $\mathcal{D}ec(\psi)$. The set of explicitly decided variables is denoted by $\mathcal{D}ec_e(\psi)$. Note that if $\exists X \gamma$ is a prime formula, then every variable in $\mathcal{V}(\gamma)$ is explicitly decided. For the formula

$$\psi = \exists x, x_1, x_2(xfy \wedge x_1\{f,g\} \wedge x_1fy \wedge x_1gx_2 \wedge zhx_2)$$

we get $\mathcal{D}ec_e(\psi) = \{y, z, x_2\}$ and $\mathcal{D}ec(\psi) = \mathcal{D}ec_e(\psi) \cup \{x_1\}$. The variable x is the only one which is undecided in ψ.

Proposition 8.1. Let γ be a solved formula, $\psi = \exists X \gamma$, and Y be the subset of X containing all variables that are decided in ψ. Then for every valuation α into a CFT′ *model $\mathcal{A}$ with $\mathcal{A}, \alpha \models \psi$ there exists a unique Y-update α' of α such that*

$$\mathcal{A}, \alpha' \models \exists X \backslash Y \phi.$$

Proposition 8.2. Let γ be a solved formula and X be a set of variables. If x is a variable that is decided in $\exists X \gamma$ and γ contains a constraint xfy, then y is also decided in $\exists X \gamma$.

The following lemmas and propositions will show that we can transform every existentially quantified basic constraint into a prime formula. A constraint c is called a *constraint for* x if c is of the form xft, xF or $x \doteq t$. We will say that two formulae $\psi = \exists X \gamma$ and $\psi' = \exists X' \gamma'$ *differ only on the undecided variables* if $\mathcal{V}(\psi) = \mathcal{V}(\psi')$, $\mathcal{D}ec_e(\psi) = \mathcal{D}ec_e(\psi')$, and ψ and ψ' contain exactly the same constraints for the explicitly decided variables.

Lemma 8.1. Let ϕ, ϕ' be graphs and let $\psi = \exists X \phi$ and $\psi' = \exists X' \phi'$ be formulae that differ only on the undecided variables. Then

$$\psi \models\!\!\!\mid_{\mathrm{CFT}'} \psi'.$$

Proof. Let ψ, ψ' be given as described and let $Z = \mathcal{D}ec_e(\psi) \cap X = \mathcal{D}ec_e(\psi') \cap X'$. As ψ and ψ' contains the same constraints for the decided variables, we can write ψ and ψ' as

$$\psi = \exists Z \exists Y(\gamma \wedge \xi) \quad \text{and} \quad \psi' = \exists Z \exists Y'(\gamma' \wedge \xi),$$

where $Y = X \backslash Z$, $Y' = X' \backslash Z$ and ξ contains all constraints for the variables in Z. Note that all variables of ξ are decided in ψ. Hence $\mathcal{V}(\xi) \cap Y = \emptyset$ and $\mathcal{V}(\xi) \cap Y' = \emptyset$. This implies that

$$\psi \models\!\!\!\mid \exists Z(\exists Y \gamma \wedge \xi) \quad \text{and} \quad \psi' \models\!\!\!\mid \exists Z(\exists Y' \gamma' \wedge \xi),$$

Now γ and γ' are graphs. The variables which are free in $\exists Y\gamma$ and $\exists Y'\gamma'$ are decided or free in ψ and ψ'. This implies that γ and γ' contain no constraints for the free variables in $\exists Y\gamma$ and $\exists Y'\gamma'$. Hence,

$$\text{CFT}' \models \tilde{\forall}\exists Y\gamma \quad \text{and} \quad \text{CFT}' \models \tilde{\forall}\exists Y'\gamma'$$

by proposition 6.1. This shows $\psi \models\!\!\!\dashv \exists Z\xi$ and $\psi' \models\!\!\!\dashv \exists Z\xi$. $\square$

Lemma 8.2 (Garbage Collection). Let $\psi = \exists X\gamma$ and $\psi' = \exists X'\gamma'$ be existentially quantified solved formulae that differ only on the undecided variables. Then

$$\psi \models\!\!\!\dashv_{\text{CFT}'} \psi'.$$

PROOF. Let $Y = \mathcal{D}ec_e(\psi) \cap X = \mathcal{D}ec_e(\psi') \cap X'$, $Z = X \backslash Y$ and $Z' = X' \backslash Y$. Y contains the existentially quantified, explicitly decided variables, whereas Z and Z' contain the variables that are not explicitly decided in ψ and ψ', respectively. We will show that there is a possible empty conjunction of equations ϕ such that

$$\exists X\gamma \models\!\!\!\dashv_{\text{CFT}'} \exists Y(\phi \wedge \exists Z\gamma_G) \quad \text{and} \quad \exists X\gamma \models\!\!\!\dashv_{\text{CFT}'} \exists Y(\phi \wedge \exists Z'\gamma'_G). \tag{1}$$

Once we have shown this, the lemma can be proven as follows. Since $\mathcal{V}(\exists Z\gamma_G) \subseteq \mathcal{D}ec_e(\psi)$, we know that every variable explicitly decided in $\exists Z\gamma_G$ must also be explicitly decided in ψ: a variable x is explicitly decided in $\exists Z\gamma_G$ if there is a variable $y \in \mathcal{V}(\exists Z\gamma_G)$ with $ypx \in [\gamma_G]$. Since $y \in \mathcal{D}ec_e(\psi)$, we know that there is variable $z \in \mathcal{V}(\psi)$ with $zqy \in [\gamma]$ for some path q. Hence, $zpqx \in [\gamma]$, which implies that z is explicitly decided in ψ.

Similarly we can show that $\mathcal{D}ec_e(\exists Z'\gamma'_G) \subseteq \mathcal{D}ec_e(\psi)$. This implies that $(\exists Z\gamma_G)$ and $(\exists Z'\gamma'_G)$ are graphs that do not differ on the decided variables. Then the previous lemma shows that

$$\exists Z\gamma_G \models\!\!\!\dashv_{\text{CFT}'} \exists Z'\gamma'_G,$$

which proves $\psi \models\!\!\!\dashv_{\text{CFT}'} \psi'$.

For the proof of (1) let ϕ be the subset of equations $x \doteq t$ in $\gamma_N \cap \gamma'_N$ with $\mathcal{V}(x \doteq t) \subseteq \mathcal{D}ec_e(\psi)$. Then all variables occuring on the left side of an equation in $\gamma_N \backslash \phi$ (resp. $\gamma'_N \backslash \phi$) cannot be explicitly decided in ψ (resp. ψ'). Since γ and γ' eliminate the variables on the left side of the equations, we get

$$\exists X\gamma \models\!\!\!\dashv \exists X(\phi \wedge \gamma_G) \quad \text{and} \quad \exists X\gamma' \models\!\!\!\dashv \exists X(\phi \wedge \gamma'_G)$$

Now (1) follows from the fact that $\mathcal{V}(\phi) \cap Z = \emptyset$ and $\mathcal{V}(\phi) \cap Z' = \emptyset$. $\square$

Proposition 8.3. For every prime formula β and every set of variables X one can compute a prime formula β' such that

$$\exists X\beta \models\!\mid_{\mathrm{CFT}'} \beta' \quad \text{and} \quad \mathcal{V}(\beta') \subseteq \mathcal{V}(\exists X\beta).$$

PROOF. We will proof that we can compute a formula β' as required by the lemma for the special case $X = \{x\}$. For arbitrary sets X we can compute a β' by iterative application of the method for this special case.

Let $\beta = \exists Y\gamma$ be a prime formula. and x be a variable. We construct a prime formula β' such that $\exists x\beta \models\!\mid_{\mathrm{CFT}'} \beta'$ and $\mathcal{V}(\beta') \subseteq \mathcal{V}(\exists x\beta)$. We distinguish the following cases.

1. $x \notin \mathcal{V}(\beta)$. Then $\beta' := \beta$ does the job.

2. $\gamma = (x \doteq t \ \wedge \gamma')$. Then $\beta' := \exists Y\gamma'$ does the job.

3. $\gamma = (y \doteq x \wedge \gamma')$. Then $\beta' := \exists Y(\gamma'[x \leftarrow y])$ does the job since $\gamma \models\!\mid x \doteq y \wedge \gamma'[x \leftarrow y]$.

4. $x \notin Y$ *and* x *occurs in the graph but not in the normalizer of* γ. Then $\exists x\exists Y\gamma \models\!\mid \gamma_N \wedge \exists x\exists Y\gamma_G$. Let γ'_G contain all the constraints for the variable that are decided in $\exists x\exists Y\gamma_G$. Then $\exists x\exists Y\gamma_G$ and $\exists x\exists Y\gamma'_G$ have the same set of decided variables and contain the same constraints for the decided variables. Since γ_G and γ'_G contain no equations, they are solved clauses. Hence, by proposition 8.2

$$\exists x\exists Y\gamma_G \models\!\mid_{\mathrm{CFT}'} \exists x\exists Y\gamma'_G$$

This implies that $\beta' = \gamma_N \wedge \exists x\exists Y\gamma'_G$ is a prime formula with $\beta \models\!\mid_{\mathrm{CFT}'} \beta'$ and $\mathcal{V}(\beta') \subseteq \mathcal{V}(\beta)$. □

Proposition 8.4. For every two prime formulae β and β' one can compute a formula δ that is either prime or $\perp$ and satisfies

$$\beta \wedge \beta' \models\!\mid_{\mathrm{CFT}'} \delta \quad \text{and} \quad \mathcal{V}(\delta) \subseteq \mathcal{V}(\beta \wedge \beta').$$

PROOF. Let $\beta = \exists X\gamma$ and $\beta' = \exists X'\gamma'$ be prime formulae. Without loss of generality we can assume that X and X' are disjoint. Hence

$$\beta \wedge \beta' \quad \models\!\mid \quad \exists X\exists X'(\gamma \wedge \gamma').$$

Since $\gamma \wedge \gamma'$ is a basic constraint, Proposition 6.3 tells us that we can compute a formula ϕ that is either solved or $\perp$, and satisfies $\gamma \wedge \gamma' \models\!\mid_{\mathrm{CFT}'} \phi$ and $\mathcal{V}(\phi) \subseteq \mathcal{V}(\gamma \wedge \gamma')$. If $\phi = \perp$, then $\delta := \perp$ does the job. Otherwise, ϕ is solved. Since

$$\beta \wedge \beta' \quad \models\!\mid_{\mathrm{CFT}'} \quad \exists X\exists X'\phi,$$

we know by Proposition 8.3 how to compute a prime formula β'' such that $\beta \wedge \beta' \models\!\!\!\dashv_{\mathrm{CFT}'} \beta''$. From the construction of β'' one can verify easily that $\mathcal{V}(\beta'') \subseteq \mathcal{V}(\beta \wedge \beta')$. □

Now we extend the notion of a closure as defined for solved formulae to prime formulae.

Definition 8.3. [Closure of Prime Formulae] The *closure of a prime formula* $\beta = \exists X \gamma$ is defined as follows:

$$[\exists X\gamma] \quad := \quad \{\pi \in [\gamma] \mid \ \pi = x\varepsilon{\downarrow} \text{ or } \pi \text{ proper path constraint with } \mathcal{V}(\pi) \cap X = \emptyset\ \}.$$

Proposition 8.5. If β is a prime formula and $\pi \in [\beta]$, then $\beta \models \pi$ (and hence $\neg\pi \models \neg\beta$).

PROOF. Let $\beta = \exists X\gamma$ be a prime formula, $\mathcal{A}, \alpha \models \beta$, and $\pi \in [\beta]$. Let α' be an arbitrary X-update of α such that $\mathcal{A}, \alpha' \models \gamma$. Since $[\beta] \subseteq [\gamma]$, we have $\pi \in [\gamma]$ and thus $\mathcal{A}, \alpha' \models \pi$. If π has no variable in common with X, then $\mathcal{A}, \alpha \models \pi$. Otherwise, π has the form "$x\varepsilon{\downarrow}$" and hence $\mathcal{A}, \alpha \models \pi$ holds trivially. □

We now know that the closure $[\beta]$, taken as an infinite conjunction, is entailed by β. We will show that, conversely, β is entailed by certain finite subsets of its closure $[\beta]$. For this we first need the definition of a rooted path.

Definition 8.4. [Rooted Path] A *rooted path* xp consists of a variable x and a path p. The *value* $|xp|_\gamma$ of a rooted path xp in some solved formula γ is defined as follows:

$$|xp|_\gamma := \begin{cases} x & \text{iff } p = \varepsilon \wedge x \doteq t \notin \gamma \\ t & \text{iff } p = \varepsilon \wedge x \doteq t \in \gamma \\ t & \text{iff } xpt \in [\gamma] \\ \text{undefined} & \text{otherwise.} \end{cases}$$

A rooted path xp is said to be *realized* in a solved formula γ iff $|xp|_\gamma$ is defined. A rooted path xp is *realized* in a prime formula $\beta = \exists X\gamma$ if either $p = \varepsilon$ or $x \in \mathcal{V}(\beta)$ and xp is realized in γ.

We say that a proper path constraint π *contains* a rooted path xp if $\pi = xp{\downarrow}$, $\pi = xpc$, $\pi = xp \downarrow yq$ or $\pi = yq \downarrow xp$

Proposition 8.6. $|\cdot|_\gamma$ is a partial function for every solved formula γ.

PROOF. Follows from proposition 6.4 (5). □

Proposition 8.7. Let xp be a rooted path with $p \neq \epsilon$. If xp is realized in some solved formula γ, then $|xp|_\gamma$ is either a constant or a variable z with $z \in \mathcal{V}(\gamma_G)$.

Proposition 8.8. Let $\beta = \exists X\gamma$ be a prime formula and $\pi = xp\downarrow yq$ be a proper path constraint with $\mathcal{V}(\pi) \cap X = \emptyset$. If both xp and yq are realized in β, then

$$\beta \wedge \pi \models\!\!\!\!\dashv_{\mathrm{CFT}'} \exists X(\gamma \wedge |xp|_\gamma \doteq |yq|_\gamma).$$

Definition 8.5. [Access Function] An *access function* for a prime formula $\beta = \exists X\gamma$ is a function that maps every $x \in \mathcal{V}(\gamma) - X$ to the rooted path $x\varepsilon$, and every $x \in X$ to a rooted path $x'p$ such that $x'px \in [\gamma]$ and $x' \notin X$.

Proposition 8.9. For every prime formula $\beta = \exists X\gamma$ and every access function @ of β,

$$|@x|_\gamma = x.$$

Thus, $|\cdot|_\gamma$ is the left inverse of @. But the converse is not true. Given the prime formula $\beta = \exists z\gamma$ with

$$\gamma = xfz \wedge ygz$$

and the access function with $@z = xf$, we have $@|yg|_\gamma = xf$.

Note that every prime formula has at least one access function, and that the access function of a prime formula is injective on $\mathcal{V}(\gamma)$ (follows from Proposition 6.4 (5)).

Definition 8.6. [Projection] The *projection* of a prime formula $\beta = \exists X\gamma$ with respect to an access function @ for β is the conjunction of the following proper path constraints:

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

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

Proposition 8.10. Let λ be a projection of a prime formula β. Then $\lambda \subseteq [\beta]$ and $\lambda \models\!\!\!\mid_{\mathrm{CFT}'} \beta$.

Proof. Let λ be the projection of a prime formula $\beta = \exists X\gamma$ with respect to an access function @.

Since every path constraint $\pi \in \lambda$ is in $[\beta]$ and thus satisfies $\beta \models \pi$, we have $\beta \models \lambda$.

To show the other direction, suppose $\mathcal{A}, \alpha \models \lambda$, where $\mathcal{A}$ is a model of CFT$'$. Then $\mathcal{A}, \alpha' \models x'px$ for every $x \in X$ with $@x = x'p$ defines a unique X-update α' of α. From the definition of a projection it is clear that $\mathcal{A}, \alpha' \models \gamma$. Hence $\mathcal{A}, \alpha \models \beta$. $\square$

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

9. Proof of the Main Lemmas

In this section we will show that our prime formulae for CFT$'$ satisfy requirements (5) and (6) of lemma 5.1. We will define the central notion of an X-joker, where X is a set of variables. This is the main device for proving the equivalences as required by the conditions 5 and 6 of the Lemma 5.1. Roughly speaking, a path constraint p is an X-joker for a prime formula β if it is not a consequence of β and contains a rooted path whose value is a variable which is both undecided and undetermined in $\exists X\beta$. To given an example, consider the path constraint $xf \downarrow xg$, which is a projection of the formula

$$\beta' \quad := \quad \exists y(xfy \wedge xgy)$$

that we used in the introductory example on page 44. Then $xf \downarrow xg$ is an $\{x\}$-joker for the formula

$$\beta \quad := \quad \exists x_1, x_2(xfx_1 \wedge xgx_2),$$

since the values of xf and xg in β are both undetermined and undecided in $\exists x\beta$. On the other hand, $xf \downarrow xg$ is not an $\{x\}$-joker for the formulae

$$\begin{aligned}
\beta_1 \quad &= \quad \exists x_1, x_2, x_3, x_4\,(xfx_1 \wedge xgx_2 \wedge x_1\{f\} \wedge x_1fx_3 \wedge x_2\{f\} \wedge x_2fx_4) \\
\beta_2 \quad &= \quad \exists x_1, x_2\,(xfx_1 \wedge xgx_2 \wedge x_1\{f\} \wedge x_1fx_1 \wedge x_2\{f\} \wedge x_2fx_2)\,.
\end{aligned}$$

But in the case of β_1 we can calculate an $\{x\}$-joker for β_1 which is a consequence of $xf \downarrow xg$, namely the path constraint $xff \downarrow xgf$. In the case of β_2 this is not possible.

Definition 9.1. A rooted path xp is said to be *determined* in $\beta = \exists X\gamma$ if $|xp|_\gamma$ is defined and $|xp|_\gamma \in \mathcal{D}(\gamma)$.

Proposition 9.1. Let $\beta = \exists X\phi$ be some prime formula and $x \in \mathcal{D}(\gamma)$ be a variable that is undecided in β. Then there is a variable y and path p such that $xpy \in [\gamma]$, $y \notin \mathcal{D}(\gamma)$ and y is undecided in β.

PROOF. Since x is in $\mathcal{D}(\gamma)$ *and* is undecided, every determinant $\delta \subseteq \gamma$ with $x \in \mathcal{D}(\delta)$ must contain an undecided parameter. Now let δ be the largest determinant such that $\delta \subseteq \gamma$, $x \in \mathcal{D}(\delta)$ and for every $z \in \mathcal{V}(\delta)$ there is a path p with

$$xpz \in [\gamma].$$

Such a determinant must exists since β is saturated. Now let y be one parameter of δ that is undecided; y cannot be determined in γ. If γ contained a determinant D for y, then $\delta' = \delta \wedge y \doteq D$ would be a determinant that is larger than δ and satisfies $\delta' \subseteq \gamma$, $x \in \mathcal{D}(\delta')$ and $\forall z \in \mathcal{V}(\delta')\exists p : xpz \in [\gamma]$. Hence, y is the variable we sought. □

Definition 9.2. Let $\beta = \exists Y\gamma$ be a prime formula and X be a set of variables. A rooted path xp is said to be *decided in β wrt. X* if either $x \notin X$ or there is some prefix p' such that xp' is realized and $|xp'|_\gamma$ is either constant or a variable that is decided in $\exists X\beta$.

Proposition 9.2. If π is a proper path constraint such that all rooted paths contained in π are decided in β wrt. X, then either $\mathcal{A}, \alpha \models \forall X(\beta \rightarrow \pi)$ or $\mathcal{A}, \alpha \models \forall X(\beta \rightarrow \neg\pi)$.

Definition 9.3. [X-Joker] Let $\beta = \exists Y\phi$ be a prime formula and X be a set of variables. We say that a rooted path xp is *free* in β wrt. X if xp is neither determined in β nor decided in β wrt. X. A proper path constraint is called an *X-joker for β* if $\pi \notin [\beta]$ and one of the following conditions is satisfied:

- $\pi = xp\downarrow$ and xp is free in β wrt. X,
- $\pi = xpc$ and xp is free in β wrt. X,
- $\pi = xp \downarrow yq$ and xp is free in β wrt. X,
- $\pi = yq \downarrow xp$ and xp is free in β wrt. X.

Proposition 9.3. It is decidable whether a rooted path is free in a prime formula wrt. a set of variables, and whether a path constraint is an X-joker for a prime formula.

PROOF. Follows from proposition 6.4. □

Lemma 9.1. Let $\beta = \exists Y \gamma$ be a prime formula and π be a proper path constraint. Then either we can calculate an X-joker π' for β with

$$\beta \wedge \pi \models \pi'$$

or for every CFT$'$ *model $\mathcal{A}$ and every valuation α we have*

$$\mathcal{A}, \alpha \models \forall X(\beta \to \pi) \quad or \quad \mathcal{A}, \alpha \models \forall X(\beta \to \neg\pi).$$

PROOF. Without loss of generality we can assume that $\mathcal{V}(\pi) \cap Y = \emptyset$. If π is an element of $[\beta]$, then $\beta \models_{\mathrm{CFT}'} \pi$ by proposition 8.5. If the normal form of $\beta \wedge \pi$ is $\perp$, then $\beta \models \neg\pi$. If both fail, then we distinguish the cases listed below. We will say that a rooted path xp is decided when xp is decided in β wrt. X, and we will use the term undecided correspondingly. Analogously, we will say that a variable is (un-)decided if it is (un-)decided in $\exists X\beta$.

The possible cases are as follows:

1. *all rooted paths contained in π are decided.* Then proposition 9.2 shows that for every CFT$'$ model $\mathcal{A}$ and every α either $\mathcal{A}, \alpha \models \forall X(\beta \to \pi)$ or $\mathcal{A}, \alpha \models \forall X(\beta \to \neg\pi)$.
2. *π contains a rooted path xp that is undecided and not realized in β.* Then $xp\!\downarrow$ is an X-joker since γ is saturated.
3. *π contains at least one undecided rooted path, and the undecided rooted paths contained in π are realized in β.* We will subdivide this case as follows:

3.a $\pi = xp\!\downarrow$. Then π is in $[\beta]$.

3.b *$\pi = xpc$ and xp is undecided but realized in β.* By our assumption we can assume that xp is not determined in β since this would imply $\beta \wedge \pi \models\!\!\!\!\dashv_{\mathrm{CFT}'} \perp$. Hence, π must be an X-joker.

3.c $\pi = xpF$. Analogous to case (3.b).

3.d *$\pi = xp\!\downarrow\! yq$ and xp is decided and yq is undecided.* Then yq is realized. If yq is undetermined in β, then π is an X-joker.
Otherwise let $z = |yq|_\gamma$ with $z \in \mathcal{D}(\gamma)$. Since z is undecided, proposition 9.1 shows that there is a variable $u \notin \mathcal{D}(\gamma)$ that is undecided and a path r such that $zru \in [\gamma]$. Then yqr is a rooted path that is both undecided and not determined in β.
Now $|xpr|_\gamma$ must be either undefined or a variable z' with $z' \neq z$, since otherwise u would be a decided variable. Hence, $\pi' = xpr\!\downarrow\! yqr$ is not in $[\beta]$. This shows that π' is an X-joker with $\pi \models_{\mathrm{CFT}'} \pi'$.

3.e *$\pi = xp\!\downarrow\! yq$ and both xp and yq are undecided.* Then xp and yq are realized in β.
If $|xp|_\gamma$ is not an element of $\mathcal{V}(\gamma_G)$, then xp is not determined in β, which implies that π is an X-joker, and similar for yq.
Otherwise, let @ be some access function of β and θ be a normalizer of $\gamma_G \wedge |xp|_\gamma \doteq |yq|_\gamma$. Note that

$$\beta \wedge \pi \models\!\!\!\!\dashv_{\mathrm{CFT}'} \exists Y(\gamma_N \wedge \gamma_G \wedge |xp|_\gamma \doteq |yq|_\gamma)$$

and
$$\gamma_G \wedge |xp|_\gamma \doteq |yq|_\gamma \;\; \models\!\!\!\!\dashv_{\mathrm{CFT}'} \;\; \gamma_G \wedge \theta.$$
Since $\mathcal{V}(|xp|_\gamma \doteq |yq|_\gamma) \subseteq \mathcal{V}(\gamma_G)$, we can assume by proposition 7.2 that $\mathcal{V}(\theta) \subseteq \mathcal{V}\gamma_G$. Since γ_N eliminates the variable on the left side of the equations, this implies

$$\gamma_N \wedge \gamma_G \wedge |xp|_\gamma \doteq |yq|_\gamma \models\!\!\!\!\dashv_{\mathrm{CFT}'} \gamma_N \wedge \gamma_G \wedge \theta.$$

Furthermore, we can assume without loss of generality that θ contains no trivial equations of form $z \doteq z$. Hence, $@z_1 \downarrow @z_2 \notin [\beta]$ for every equation $z_1 \doteq z_2$ in θ. Since we have assumed $\beta \wedge \pi \not\models_{\mathrm{CFT}'} \bot$, we know that $\theta\gamma_G$ is clash-free.
If θ contains an equation $z \doteq c$ where z is undecided, then $z \in \mathcal{V}(\gamma_G)$. Now z cannot be determined in γ_G as $\theta\gamma_G$ is clash-free. Hence, $@zc$ is an X-joker π' with $\beta \wedge \pi \models_{\mathrm{CFT}'} \pi'$.
If θ contains an equation $z_1 \doteq z_1$ or $z_2 \doteq z_1$ where z_1 is undecided and z_2 is decided, then $\pi' = @z_1 \downarrow @z_2$ is a proper path constraint with $\beta \wedge \pi \models \pi'$. Furthermore, we can apply case (3.d) to π' yielding an X-joker π'' with $\beta \wedge \pi \models \pi''$.
If θ contains an equation $z_1 \doteq z_1$ or $z_2 \doteq z_1$ where z_1 and z_2 are undecided and z_1 is not determined in γ_G, then $\pi' = @z_1 \downarrow @z_2$ is an X-joker with $\beta \wedge \pi \models \pi'$.
The remaining case is that θ contains only equations of the form $z \doteq c$ with z decided or equations of the form $z_1 \doteq z_2$ where either both variables are decided or both variables are undecided but determined in γ_G. We will show that in this case $\mathcal{A}, \alpha \models \exists X(\beta \wedge \pi)$ implies $\mathcal{A}, \alpha \models \forall X(\beta \rightarrow \pi)$.
Now assume that $\mathcal{A}, \alpha \models \exists X(\beta \wedge \pi)$. We will show that then

$$\mathcal{A}, \alpha \models \forall X \forall Y(\gamma_G \rightarrow \theta). \tag{1}$$

This implies that $\mathcal{A}, \alpha \models \forall X \forall Y(\gamma_N \wedge \gamma_G \rightarrow |xp|_\gamma \doteq |yq|_\gamma)$, which is equivalent to $\mathcal{A}, \alpha \models \forall X(\exists Y \gamma \rightarrow \pi)$.
Let θ' be the subset of θ containing all equations among decided variables, and let θ'' be the rest of θ. θ'' contains only equations between variables that are determined in γ_G. It is easy to check that θ' is a normalizer of some congruence of γ_G. This implies by proposition 7.3 that $\theta'\gamma_G$ is a solved graph.
Let α' be the unique extension of α to the variables that are decided, and let $Z \subseteq X \cup Y$ be the set of undecided variables. Clearly, $\mathcal{A}, \alpha' \models \theta'$. Furthermore, $\mathcal{V}(\theta') \cap Z = \emptyset$. This implies

$$\mathcal{A}, \alpha' \models \forall Z(\gamma_G \leftrightarrow \theta'\gamma_G).$$

Since θ'' is a normalizer of some congruence of $\theta'\gamma_G$ by proposition 7.4, $\theta'\gamma_G$ is a solved graph and $\mathcal{V}(\theta'') \subseteq \mathcal{D}(\theta'\gamma_G)$, we know by lemma 7.1 that

$$\theta'\gamma_G \models \theta''.$$

Hence, $\mathcal{A}, \alpha' \models \forall Z(\gamma_G \rightarrow \theta' \wedge \theta'')$, which implies $\mathcal{A}, \alpha' \models \forall Z(\gamma_G \rightarrow \pi)$. From this follows (1) as α' was the unique update of α to $\mathcal{D}ec(\exists X \beta)$.

□

Corollary 9.1. Let β be a prime formula and π be a proper path constraint. If there is a CFT′ *model $\mathcal{A}$ and a valuation α into $\mathcal{A}$ with*

$$\mathcal{A}, \alpha \models \exists X(\beta \wedge \pi) \quad \textit{and} \quad \mathcal{A}, \alpha \models \exists X(\beta \wedge \neg\pi),$$

then we can calculate an X-joker for β with $\beta \wedge \pi \models \pi'$.

Lemma 9.2. Let $\beta = \exists Y\gamma$ be a prime formula and $\pi_1, \ldots, \pi_n$ be X-jokers for β. Then

$$\exists X\beta \models_{\mathrm{CFT}'} \exists X(\beta \wedge \bigwedge_{i=1}^{n} \neg\pi_i)$$

PROOF. Let $\beta = \exists Y\gamma$ be a prime formula, $\pi_1, \ldots, \pi_n$ $(n > 0)$ be X-jokers for β, $\mathcal{A}$ be some model of CFT′, and α be some valuation into $\mathcal{A}$ with $\mathcal{A}, \alpha \models \exists X\beta$. We have to show that $\mathcal{A}, \alpha \models \exists X(\beta \wedge \bigwedge_{i=1}^{n} \neg\pi)$. We will define a prime formula β' satisfying the following:

- $\beta' \models \beta$,
- $\exists X\beta \models\!\!\!\!\dashv_{\mathrm{CFT}'} \exists X\beta'$,
- $\mathcal{A}, \alpha \models \forall X(\beta' \to \neg\pi_i)$ for all $i = 1..n$

Once we have defined a β' satisfying these conditions, we can prove the claim using the following argument. Since $\exists X\beta \models\!\!\!\!\dashv_{\mathrm{CFT}'} \exists X\beta'$ and $\mathcal{A}, \alpha \models \exists X\beta$, there must be an X-update α' of α such that $\mathcal{A}, \alpha' \models \beta'$. But as $\beta' \models \beta$ and for all $i = 1..n$ $\mathcal{A}, \alpha \models \forall X(\beta' \to \neg\pi_i)$, we know that $\mathcal{A}, \alpha' \models \beta \wedge \bigwedge_{i=1}^{n} \neg\pi_i$. This shows $\mathcal{A}, \alpha \models \exists X(\beta \wedge \bigwedge_{i=1}^{n} \neg\pi_i)$.

For the construction of β' let R_d denote the set of all rooted path that occur in some π_i and that are decided in β wrt. X. In the following, we will just say that a rooted path xp is decided when xp is decided in β wrt. X, and we will use undecided similarly. Let $Z \subseteq \mathcal{V}(\gamma_G)$ be the set of all variables of γ_G that are undecided and not determined in γ_G. For each $z \in Z$ we fix a non-empty set of features F_z with the following properties:

1. $F_z = \{f \mid zfy \in \gamma\} \cup \{h\}$, where h is a new feature.
2. $\mathcal{A}, \alpha \models \forall X(\beta \to \neg yqF_z)$ for all $yq \in R_d$

It is understood that $F_z \neq F_{z'}$ for $z \neq z'$.

We can find such sets F_z satisfying the above properties if for every $yq \in R_d$ there are infinitely many sets F with $\mathcal{A}, \alpha \models \forall X(\beta \to \neg yqF)$. For this it is sufficient to prove that for every $yq \in R_d$ there is at most one set of features F with $\mathcal{A}, \alpha \models \exists X(\beta \wedge yqF)$. Assume that $\mathcal{A}, \alpha \models \exists X(\beta \wedge yqF_1)$ and $\mathcal{A}, \alpha \models \exists X(\beta \wedge yqF_2)$ with $F_1 \neq F_2$. By proposition 9.2 we can conclude that in this case $\mathcal{A}, \alpha \models \forall X(\beta \to yqF_1)$ and $\mathcal{A}, \alpha \models \forall X(\beta \to yqF_2)$. This would imply

$$\mathcal{A}, \alpha \models \forall X(\beta \to (yqF_1 \wedge yqF_2)).$$

Since $yqF_1 \wedge yqF_2 \models\!\mid_{\mathrm{CFT}'} \perp$, this is contradictory to our assumption that $\mathcal{A}, \alpha \models \exists X\beta$.

The formula $\beta' = \exists X\gamma'$ is now defined by

$$\gamma' = \gamma \wedge \bigwedge_{z \in Z} zF_z$$

Clearly, $\beta' \models_{\mathrm{CFT}'} \beta$. Furthermore, $\exists X\beta \models\!\mid \exists X\beta'$ by proposition 8.2.

It remains to show that $\mathcal{A}, \alpha \models \forall X(\beta' \rightarrow \bigwedge_{i=1}^{n} \neg\pi_i)$ for all $i = 1..n$. We distinguish the following cases for π_i:

1. *π_i contains a rooted path xp that is undecided and not realized in β*: Let p' be the longest path such that xp' is realized in β (such a path must exists since at least $x\epsilon$ is realized), and let $p = p'fq$. Note that xp' is not determined in β as β is saturated. Since xp is undecided, we know that $|xp'|_\gamma$ is a variable z with $z \in Z$, which implies that γ' contains an arity constraint zF_z. As p' is the longest subpath of p' with xp' realized in β we know that $zft \notin \gamma$ and therefore $f \notin F_z$. Hence, $\beta \models \neg xp{\downarrow}$.
2. *every undecided rooted path contained in π_i is realized in β.* Note that in this case π_i cannot be of the form $xp{\downarrow}$, since xp realized implies that $xp{\downarrow} \in [\beta]$. We will split this case up as follows:
   (a) $\pi_i = xpc$. Then xp must be undecided as π_i is an X-joker. Since xp is realized in β, we know that $|xp|_\gamma$ is a variable z with $z \in Z$. Then either γ contains an arity constraint zF or we have added an arity constraint zF_z in γ'. In each case we get $\beta' \models \neg\pi_i$.
   (b) $\pi_i = xpF$. Analogous to case (2a)
   (c) *$\pi_i = xp{\downarrow}yq$ or $\pi_i = yq{\downarrow}xp$ where xp is undecided and not determined in β.* By the above cases we can assume that xp is realized in β. Again we get $|xp|_\gamma = z \in Z$. This implies that we have added a feature constraint zF_z in γ'.
   If yq is undecided, we can assume without loss of generality that yq is also realized in β. $|yq|_\gamma$ must be variable since yq is undecided. Let $z' = |yq|_\gamma$. If yq is determined in β, then γ contains an arity constraint $z'F$ with $F \neq F_z$ as F_z contains a feature h which is new. If yq is not determined in β, then $z' \in Z$. This implies that we have added an arity constraint $z'F_{z'}$ in γ'. In both cases we get $\beta' \models_{\mathrm{CFT}'} \neg\pi$.
   If yq is decided, then $\mathcal{A}, \alpha \models \forall X(\beta \rightarrow \neg yqF_z)$ by the definition of F_z. As $\beta' \models \beta$, this shows $A, \alpha \models \forall X(\beta' \rightarrow \neg xp \downarrow yq)$.

□

Lemma 9.3. If β, $\beta_1, \ldots, \beta_n$ are prime formulae, then

$$\exists X(\beta \wedge \bigwedge_{i=1}^{n} \neg\beta_i) \quad \models\!\mid_{\mathrm{CFT}'} \quad \bigwedge_{i=1}^{n} \exists X(\beta \wedge \neg\beta_i).$$

Proof. Let $\beta, \beta_1, \ldots, \beta_n$ be prime formulae. Then $\exists X(\beta \wedge \bigwedge_{i=1}^n \neg\beta_i) \models \bigwedge_{i=1}^n \exists X(\beta \wedge \neg\beta_i)$ is trivial. To see the other direction, suppose that $\mathcal{A}$ is a model of CFT$'$ and $\mathcal{A}, \alpha \models \bigwedge_{i=1}^n \exists X(\beta \wedge \neg\beta_i)$. We must exhibit some X-update α' of α such that $\mathcal{A}, \alpha' \models \beta$ and $\mathcal{A}, \alpha' \models \neg\beta_i$ for $i = 1, \ldots, n$.

Without loss of generality we can assume that $\mathcal{A}, \alpha' \models \exists X(\beta \wedge \beta_i)$ for $i = 1, \ldots, m$ and $\mathcal{A}, \alpha' \models \neg\exists X(\beta \wedge \beta_i)$ for $i = m+1, \ldots, n$. For every $i = 1, \ldots, m$ let λ_i be a projection of β_i.

Since for every $i = 1, \ldots, m$

$$\lambda_i \models\!\!\!\!\!\dashv_{\mathrm{CFT}'} \beta_i,$$

we know that there is a proper path constraint π with

$$\mathcal{A}, \alpha \models \exists X(\beta \wedge \pi) \quad \text{and} \quad \mathcal{A}, \alpha \models \exists X(\beta \wedge \neg\pi),$$

This implies by corollary 9.1 that we can calculate, for every $i = 1, \ldots, m$, an X-joker π_i' for β with $\beta \wedge \pi_i \models_{\mathrm{CFT}'} \pi_i'$. By Lemma 9.2 we have

$$\exists X\beta \models \exists X(\beta \wedge \bigwedge_{i=1}^{m} \neg\pi_i').$$

from which

$$\exists X\beta \models \exists X(\beta \wedge \bigwedge_{i=1}^{m} \neg\pi_i).$$

follows.

Since $\neg\pi_i \models \neg\beta_i$ by Proposition 8.5, we have

$$\exists X\beta \models \exists X(\beta \wedge \bigwedge_{i=1}^{m} \neg\beta_i).$$

Hence we know that there exists an X-update α' of α such that $\mathcal{A}, \alpha' \models \beta$ and $\mathcal{A}, \alpha' \models \neg\beta_i$ for $i = 1, \ldots, m$. Since we know that $\mathcal{A}, \alpha \models \neg\exists X(\beta \wedge \beta_i)$ for $i = m+1, \ldots, n$, we have $\mathcal{A}, \alpha' \models \neg\beta_i$ for $i = m+1, \ldots, n$. □

Lemma 9.4. For every two prime formulae β, β' and every set of variables X one can compute a Boolean combination δ of prime formulae such that

$$\exists X(\beta \wedge \neg\beta') \models\!\!\!\!\!\dashv_{\mathrm{CFT}'} \delta \quad \textit{and} \quad \mathcal{V}(\delta) \subseteq \mathcal{V}(\exists X(\beta \wedge \neg\beta'))$$

Proof. Let λ be a projection of β' and $\mathcal{A}$ be model of CFT$'$. We distinguish the following cases:

1. *There exists a $\pi \in \lambda$ such that we can derive an X-joker π' with $\beta \wedge \pi \models_{\mathrm{CFT}'} \pi'$ using lemma 9.1.* Then $\exists X \beta \models_{\mathrm{CFT}'} \exists X(\beta \wedge \neg\pi')$ by lemma 9.2. Since $\beta \wedge \neg\pi' \models_{\mathrm{CFT}'} \neg\pi$, we get

$$\exists X \beta \models_{\mathrm{CFT}'} \exists X(\beta \wedge \neg\pi).$$

Since $\beta' \models_{\mathrm{CFT}'} \lambda \models \pi$, we know that $\neg\pi \models_{\mathrm{CFT}'} \neg\beta'$ and hence $\exists X\beta \models_{\mathrm{CFT}'} \exists X(\beta \wedge \neg\beta')$. Thus

$$\exists X(\beta \wedge \neg\beta') \mathrel{\models\!\!\!\dashv}_{\mathrm{CFT}'} \exists X\beta$$

The rest follows from proposition 8.3.

2. *For every $\pi \in \lambda$ lemma 9.1 does not produce an X-joker π' with $\beta \wedge \pi \models_{\mathrm{CFT}'} \pi'$.* Then for every valuation α into $\mathcal{A}$ and every $\pi \in \lambda$ either $\mathcal{A}, \alpha \models \forall X(\beta \to \pi)$ or $\mathcal{A}, \alpha \models \forall X(\beta \to \neg\pi)$. This implies that either

$$\mathcal{A}, \alpha \models \forall X(\beta \to \bigwedge_{\pi\in\lambda} \pi)$$

or

$$\mathcal{A}, \alpha \models \forall X(\beta \to \neg(\bigwedge_{\pi\in\lambda} \pi)).$$

Since $\bigwedge_{\pi\in\lambda} \pi \mathrel{\models\!\!\!\dashv} \lambda \mathrel{\models\!\!\!\dashv}_{\mathrm{CFT}'} \beta'$, this implies that there is no valuation α with

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

Hence

$$\exists X(\beta \wedge \neg\beta') \mathrel{\models\!\!\!\dashv}_{\mathrm{CFT}'} \exists X\beta \wedge \neg\exists X(\beta \wedge \beta').$$

The rest follows from propositions 8.3 and 8.4.

□

Theorem 9.1. For every formula ϕ one can compute a Boolean combination δ of prime formulae such that $\phi \mathrel{\models\!\!\!\dashv}_{\mathrm{CFT}'} \delta$ and $\mathcal{V}(\delta) \subseteq \mathcal{V}(\beta)$.

PROOF. Follows from Lemma 5.1, Propositions 8.4 and 8.3, and Lemmas 9.3 and 9.4. □

Corollary 9.2. CFT$'$ *is a complete and decidable theory.*

PROOF. The completeness of CFT$'$ follows from the preceding theorem and the fact that $\top$ is the only closed prime formula. The decidability follows from the completeness and the fact that CFT$'$ is given by a recursive set of sentences. $\square$

For the special case $\mathcal{V}(\beta) \cup \mathcal{V}(\beta') = X$, the algorithm which can be extracted from the lemma 9.4 is closely related to the entailment test as described in [ST94a]. If case 1 of lemma 9.4 applies, then β does not entail β'.[1] Coincidently, the same approach is used in the proof of lemma 9.4 and in the algorithm described in [ST94a]. Roughly speaking, both test whether the unification of β and β' further constrains some variables of β. On the other hand, we have also to consider the case where $X \subsetneq \mathcal{V}(\beta) \cup \mathcal{V}(\beta')$. This case is outside the scope of [ST94a].

Now we want to give a concrete example of how the quantifier elimination works. Suppose we shall prove that if c_1 and c_2 are two different constant symbols, then

$$\mathrm{CFT}' \models \forall x[(xfc_1 \wedge xgc_2) \rightarrow \exists y_1, y_2(xfy_1 \wedge xgy_2 \wedge \neg(y_1 = y_2))]. \tag{2}$$

This is the same as showing that $xfc_1 \wedge xgc_2$ entails $\exists y_1, y_2(xfy_1 \wedge xgy_2 \wedge y_1 \neq y_2)$. In the following, we will abbreviate $xfc_1 \wedge xgc_2$ by β, and $xfy_1 \wedge xgy_2$ by β'. Note that both β, β' are prime formulae. The first step is to eliminate the quantifiers $\exists y_1 \exists y_2$. A projection for $y_1 = y_2$ is $y_1\epsilon \downarrow y_2\epsilon$. Since both $y_1\epsilon$ and $y_2\epsilon$ are decided in β' wrt. $\{y_1, y_2\}$, we know that $y_1\epsilon \downarrow y_2\epsilon$ is no $\{y_1, y_2\}$-joker for β'. Hence, we can apply case 2 of lemma 9.4:

$$\begin{array}{c} \forall x[\neg\beta \ \vee \ \exists y_1, y_2(\beta' \wedge \neg(y_1 = y_2))] \\ \Downarrow \text{ case 2 of lemma 9.4} \\ \forall x[\neg\beta \ \vee \ (\exists y_1, y_2\beta' \wedge \neg\exists y_1, y_2(\beta' \wedge y_1 = y_2))]. \end{array}$$

Now $\exists y_1, y_2(\beta' \wedge y_1 = y_2)$ is no prime formula. An equivalent prime formula is $\beta'' = \exists y(xfy \wedge xgy)$. Now we have to eliminate the out-most quantifier $\forall x$, for which purpose we have first to apply some first-order equivalence transformation:

$$\begin{array}{c} \forall x[\neg\beta \ \vee \ (\exists y_1, y_2\beta' \wedge \neg\beta'')]. \\ \Downarrow \\ \neg\exists x[\beta \ \wedge \ (\neg\exists y_1, y_2\beta' \vee \beta'')]. \\ \Downarrow \\ \neg[\exists x(\beta \wedge \neg\exists y_1, y_2\beta') \ \vee \ \exists x(\beta \wedge \beta'')]. \end{array}$$

Since $c_1 \neq c_2$, we get $\exists x(\beta \wedge \beta'') = \exists x(\beta \wedge \exists y(xfy \wedge xgy)) \models\!\!\!\!\!\dashv_{\mathrm{CFT}'} \bot$. Hence, we have to consider only $\neg\exists x(\beta \wedge \neg\exists y_1, y_2\beta')$. Now a projection λ for $\exists y_1, y_2\beta'$ is $\{xf\downarrow, xg\downarrow\}$. Since $\lambda \subseteq [\beta]$, we can again apply case 2 of lemma 9.4, yielding

$$\neg[\exists x\beta \wedge \neg\exists x(\beta \wedge \exists y_1, y_2\beta')].$$

[1] Note that under our assumption $\mathcal{V}(\beta) \cup \mathcal{V}(\beta') = X$, the disentailment of β' by β implies that there can be an X-joker for β calculated in case 1. Hence, the instance of case 2 where β disentails β' is not used under the assumption $\mathcal{V}(\beta) \cup \mathcal{V}(\beta') = X$.

But $\exists x\beta \models\!\!\!\!\dashv_{\mathrm{CFT}'} \top$ and $\exists x(\beta \wedge \exists y_1, y_2\beta') \models\!\!\!\!\dashv_{\mathrm{CFT}'} \top$, which implies that we get $\neg[\top \wedge \neg\top]$, which is the same as $\neg\bot$ or $\top$. This proves 2.

Finally, we want to show that CFT is less expressive than FT, which is established by the existence of a quantifier elimination for FT as proven in [BS93]. We have claimed that (1) in FT it is impossible to identify a unique element of the domain, and that (2) the arity predicate cannot be defined within FT. These claims are a trivial consequence of the following lemma. We show the result for the original language CFT as defined in [ST94a] and its subsignature FT as defined in [BS93], since we can use some proposition and lemmas proven in [BS93].[2] Anyway, the same method applies for CFT$'$ and its subsignature FT$'$ not containing arity constraints (for a quantifier elimination of FT$'$ see [Bac94a]).

Lemma 9.5. Let $\phi(x)$ be any first-order FT*-formula with one free variable x such that* FT $\models \exists x\phi(x)$*. Then there is a feature f such that for all sort symbols A,*

$$\mathrm{FT} \models \exists x(\exists y(xfy \wedge Ay) \wedge \phi(x)).$$

Proof. Note that we assume the definitions of [BS93] for the different notions used in this proof. Roughly speaking, these notions are just the restrictions of the corresponding notions as defined in this papers to the signature of FT.

Let $\phi(x)$ be a formula with one free variable, and let $\gamma(x)$ be the corresponding Boolean combination of prime formulae equivalent to $\phi(x)$ which is the result of quantifier elimination. Note that x is the only free variable in $\gamma(x)$ by the definition of the quantifier elimination. Without loss of generality, we can assume that $\gamma(x)$ is in disjunctive normal form. Since prime formulae are closed under conjunction, we can furthermore assume that every disjunct of $\gamma(x)$ is of the form $\beta(x) \wedge \bigwedge_{j=1}^{k} \neg\beta_j(x)$, where $\beta(x), \beta_1(x), \ldots \beta_k(x)$ are prime formulae such that x is the only variable free in $\beta(x), \beta_1(x), \ldots \beta_k(x)$.[3] Furthermore, we can assume that every disjunct of $\gamma(x)$ is satisfiable in FT.

Now we choose some disjunct $\beta \wedge \bigwedge_{j=1}^{k} \neg\beta_j$ of $\gamma(x)$. Let f be an arbitrary feature that is not used in $\beta, \beta_1, \ldots \beta_k$, A be some arbitrary sort symbol and β' be the prime formula equivalent to the conjunction of $\exists y(xfy \wedge Ay)$ and β. Since we have infinitely many feature and sort symbols in FT, for the claim of the lemma it is sufficient to prove

$$\mathrm{FT} \quad \models \quad \exists x(\beta' \wedge \bigwedge_{j=1}^{k} \neg\beta_j). \tag{3}$$

Since $\beta \wedge \bigwedge_{j=1}^{k} \neg\beta_j$ is satisfiable and the closure of a prime formula is equivalent (seen as an infinite conjunction) to the prime formula itself (see [BS93], Proposition

[2] Recall that CFT and FT use sort constraints Ax, which are unary, disjoint predicates, but do not have constant symbols.

[3] Recall that every closed prime formula is valid in FT and hence equivalent to $\top$. This implies that we can assume without loss of generality that $\beta(x), \beta_1(x), \ldots \beta_k(x)$ have x as a free variable.

7.8), we know that there for every $j \in 1 \dots k$ there is a path constraint π_j contained in the closure $[\beta_j]$ of β_j such that

$$\mathrm{FT} \models \exists x(\beta \wedge \neg \pi_i).$$

This implies that π_j is not contained in the closure of β. Since x is the only free variable, we know that for every $j \in 1 \dots k$, π_j is an x-joker for β. But this must also hold for β' since the feature f is not used in β_j for every $j \in 1 \dots k$, which implies that f is not used in the π_j's. Now Lemma 8.4 of [BS93] (which is similar to Lemma 9.2 in this paper) shows that

$$\exists x \beta' \models\!\mid_{\mathrm{FT}} \exists x(\beta' \wedge \bigwedge_{j=1}^{n} \neg \pi_j).$$

Since β is satisfiable, $\exists x \beta \models_{\mathrm{FT}} \exists x \beta'$ and $\neg \pi_j \models_{\mathrm{FT}} \neg \beta_j$ for every $j \in 1 \dots k$, this immediately proves (3). $\square$

10. Conclusion

We have proven the completeness of the theory CFT$'$ given by seven axioms. Our completeness proof exhibits a terminating simplification system deciding validity and satisfiability of arbitrary CFT$'$-formulae. The simplification computes for every formula ϕ an equivalent normal form consisting of a Boolean combination of existential quantified solved formulae, from which the solutions of ϕ can be easily read of.

One can think of different extensions of the work in this paper. First, one can consider finite trees instead of infinite trees. For this purpose, the axiomatization of CFT$'$ has to be changed in order to exclude cyclic feature descriptions. We conjecture that it is sufficient to modify axiom scheme (Ax7) and to add one new axiom scheme, thus resulting in

(Ax7)	$\tilde{\forall}(\exists! D(\delta)\delta)$	if δ is a determinant that contains no cycle
(Ax8)	$\neg \exists x(xpx)$	for every path $p \in \mathrm{FEA}^*$.

These axioms have to be considered when testing satisfiability of quantifier-free formulae. Clearly, this has also effects on the simplification system for CFT$'$-formulae (since such a satisfiability test is integrated). But on the other hand, the restriction to finite trees should have no effects on the completeness proof itself. To take an example, one of the parts of the completeness proof is to show that

$$\bigwedge_{i=1}^{n} \exists X(\beta \wedge \neg \beta_i) \quad \models_{\mathrm{CFT}'} \quad \exists X(\beta \wedge \bigwedge_{i=1}^{n} \neg \beta_i), \tag{1}$$

where $\beta, \beta_1, \ldots, \beta_n$ are prime formulae (i.e., existential quantified solved formulae). This implication is proven by constructing a prime formulae β_{ext} with the property that $\beta_{ext} \models_{\mathrm{CFT}'} \beta$ and for every $i = 1 \ldots n$,

$$\mathrm{CFT}' \models \forall X(\beta_{ext} \rightarrow \neg\beta_i).$$

Clearly, the existence of an X-solution for β_{ext} for every valuation that satisfies $\exists X\beta$ proves (1). This is guaranteed by the construction of β_{ext} which is performed in such a way that β_{ext} contains no cycles if β contains no cycles. Hence, the argumentation in this proof (as well as in the other proofs of this paper) carries over to finite trees. To summarize, considering finite trees as a standard model would change the behavior of the simplification system presented in this paper, but the proof of correctness of the simplification should remain unchanged.

The second and more interesting extension is to add new predicates to the language of CFT' and to see whether the theory of the feature tree model over the extended language remains completely axiomatizable. An interesting predicate is *AjoinAt*, which has been introduced in [SHW93, Smo94b] in the context of the Oz-system (for the description of the Oz-system and the underlying concepts see also [Smo94a, ST94b]). $AjoinAt(\sigma, f, \sigma', \tau)$ holds if τ is a feature tree which has the same subtrees as σ except at the feature f, where τ has σ' as a subtree. If the feature f is defined on σ, then τ has the same arity as σ, otherwise the arity of τ is the arity of σ extended by the feature f.

Finally, it would be interesting to determine the complexity of the quantifier elimination for CFT' and to compare it with the complexity of Rabin's algorithm for deciding SnS. This is of interest since there exist approaches to translate the theory of constructor trees for restricted signatures into SnS [CP94]. But to our knowledge it is still an open problem whether one can translate also CFT' into SnS.

Acknowledgements

I am grateful to Gert Smolka and Ralf Treinen for discussions on an earlier version of this paper, and to Stephen Spackman for reading draft versions of this paper. Furthermore, I would like to thank two anonymous referees for their valuable comments.

REFERENCES

[AK84.] Hassan Aït-Kaci. *A Lattice-Theoretic Approach to Computation Based on a Calculus of Partially Ordered Type Structures.* PhD thesis, University of Pennsylvenia, Philadelphia, PA, 1984.

[AK86.] Hassan Aït-Kaci. An algebraic semantics approach to the effective resolution of type equations. *Theoretical Computer Science*, 45:293–351, 1986.

[AKN86.] Hassan Aït-Kaci and Roger Nasr. LOGIN: A logic programming language with built-in inheritance. *The Journal of Logic Programming*, 3:185–215, 1986.

[AKN89.] Hassan Aït-Kaci and Roger Nasr. Integrating logic and functional programming. *Lisp and Symbolic Computation*, 2:51–89, 1989.

[AKP93.] Hassan Aït-Kaci and Andreas Podelski. Towards a meaning of LIFE. *The Journal of Logic Programming*, 16:195–234, 1993.

[AKPS94.] 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.

[Bac94a.] Rolf Backofen. *Expressivity and Decidability of First-order Languages over Feature Trees*. PhD thesis, Universität des Saarlandes, 1994. to appear.

[Bac94b.] Rolf Backofen. Regular path expressions in feature logic. *Journal of Symbolic Computation*, 17:412–455, 1994.

[BBN+93.] Franz Baader, Hans-Jürgen Bürckert, Bernhard Nebel, Werner Nutt, and Gert Smolka. On the expressivity of feature logics with negation, functional uncertainty, and sort equations. *Journal of Logic, Language and Information*, 2:1–18, 1993.

[BS93.] Rolf Backofen and Gert Smolka. A complete and recursive feature theory. In *Proc. of the 31st ACL*, pages 193–200, Columbus, Ohio, 1993. acl. Full version has appeared as Research Report RR-92-30, DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany, and will appear in Theoretical Computer Science.

[BT94.] Rolf Backofen and Ralf Treinen. How to win a game with features. In Jean-Pierre Jouannaud, editor, *1st International Conference on Constraints in Computational Logics*, Lecture Notes in Computer Science, vol. 845, pages 320–335, München, Germany, 7–9 September 1994. Springer-Verlag.

[Car92.] Bob Carpenter. *The Logic of Typed Feature Structures*, volume 32 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, UK, 1992.

[CL89.] Hubert Comon and Pierre Lescanne. Equational problems and disunification. *Journal of Symbolic Computation*, 7:371–425, 1989.

[Col82.] Alain Colmerauer. PROLOG and infinite trees. In K.L. Clark and S.-A. Tärnlund, editors, *Logic Programming*, pages 153–172. Academic Press, 1982.

[Col84.] 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.

[CP94.] Hubert Comon and Andreas Podelski. Private communication. 1994.

[Joh88.] Mark 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.

[Joh91.] M. Johnson. Logic and feature structures. In *Proceedings of IJCAI-91*, Sydney, Australia, 1991.

[Kay79.] Martin Kay. Functional grammar. In *Proceedings of the Fifth Annual Meeting of the Berkeley Linguistics Society*, Berkeley, CA, 1979. Berkeley Linguistics Society.

[KB82.] Ronald M. Kaplan and Joan Bresnan. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–381. The MIT Press, Cambridge, MA, 1982.

[KC90.] Robert T. Kasper and Rounds William C. The logic of unification in grammar. *Linguistics and Philosopy*, 13:35–58, 1990.

[KR86.] Robert T. Kasper and William C. Rounds. A logical semantics for feature structures. In *Proceedings of the 24th Annual Meeting of the ACL, Columbia University*, pages 257–265, New York, N.Y., 1986.

[Mah88.] Michael J. Maher. Complete axiomatizations of the algebras of finite, rational and infinite trees. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science*, pages 348–457, Edinburgh, Scotland, July 1988.

[Mos92.] Lawrence S. Moss. Completeness theorems for logics of feature structures. In Y. N. Moschovakis, editor, *Logic from Computer Science*, pages 387–403, Berlin, Heidelberg, New York, 1992. Springer-Verlag.

[RK86.] William C. Rounds and Robert T. Kasper. A complete logical calculus for record structures representing linguistic information. In *Proceedings of the 1st IEEE Symposium on Logic in Computer Science*, pages 38–43, Boston, MA, 1986.

[Rou92.] Willian C. Rounds. On extensionality in feature algebras. unpublished manuscript, EECS Department, University of Michigan, Ann Arbor, Michigan 48109 USA, 1992.

[SHW93.] Gert Smolka, Martin Henz, and Jörg Würtz. Object-oriented concurrent constraint programming in Oz. In Jaume Augustí and Pere Garcia, editors, *Programación Declarativa*, pages 5–20, Blanes, Spain, 30 September–1 October 1993. To be published in *Principles and Practice of Constraint Programming*, [SHW94].

[SHW94.] Gert Smolka, Martin Henz, and Jörg Würtz. Object-oriented concurrent constraint programming in Oz. In P. van Hentenryck and V. Saraswat, editors, *Principles and Practice of Constraint Programming*. The MIT Press, 1994. To appear.

[Smo92.] Gert Smolka. Feature constraint logics for unification grammars. *Journal of Logic Programming*, 12:51–87, 1992.

[Smo94a.] Gert Smolka. A calculus for higher-order concurrent constraint programming with deep guards. Research Report RR-94-03, DFKI, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, February 1994.

[Smo94b.] Gert Smolka. The definition of kernel Oz. Research report, DFKI, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, 1994. To appear.

[ST94a.] Gert Smolka and Ralf Treinen. Records for logic programming. *The Journal of Logic Programming*, 18(3):229–258, April 1994.

[ST94b.] Gert Smolka and Ralf Treinen. A survey of Oz. Research report, DFKI, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, 1994. To appear.

[Tre93.] Ralf Treinen. Feature constraints with first-class features. In Andrzej M. Borzyszkowski and Stefan Sokołowski, editors, *Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science, vol. 711, pages 734–743, Gdańsk, Poland, 30 August–3 September 1993. Springer-Verlag.