HH

KMU-NTZ 93-27

su 9404

UNIVERSITÄT LEIPZIG

CERN LIBRARIES, GENEVA

P00020332

Naturwissenschaftlich-Theoretisches Zentrum

Completeness of the pool calculus

with an open built-in theory

by

Uwe Petermann*

NTZ der Universität Leipzig

*permanent address:

HS f. Technik, Wirtschaft, Kultur Leipzig

FB Informatik/Math./Na.-Wi.

Postfach 66, D-04251 Leipzig (F.R.G.)

Preprint-Nr.24/1993

Abstract

We present an approach¹ for building-in theories into theorem provers which are based on the connection method. The approach may be applied to a wide class of theories, among them taxonomic, constraint and equational theories. We generalize the concepts of a spanning mating and of a unifier of a connection towards the theory case. We introduce the notion of the completeness of a set of theory connections with respect to a given query language. This completeness means that for a given theory there are "enough" theory connections in order to refute all theory unsatisfiable matrices which belong to the considered query language. Moreover we define the solvability of the theory unification problem in a set of theory connections. This property means that a complete set of theory unifiers may be constructed for each given theory connection. In order to prove the completeness of a prover with a built-in theory it is sufficient to construct a complete set of theory connections such that the theory unification problem in that set is solvable. This report is the extended and revised version of a paper presented at the 3rd Kurt Gödel Colloquium 1993 in Brno (Czech Republic).

Key words: Theorem proving, connection method, built-in theories.

¹This research has been supported by a grant of the Deutsche Forschungsgemeinschaft within the research program "Deduction".

1 Introduction

The aim of the present paper is to study an approach for proving the completeness of theorem proving calculi with built-in theories which are based on the connection method [Bib82]. A reasoning system with a built-in theory consists of a so called foreground reasoner which realizes the base calculus and a so called background reasoner which performs the inferences within the built-in theory. The background reasoner is triggered whenever the foreground reasoner detects a situation where the special capabilities of the background reasoner should be used. From the viewpoint of the entire system sequences of inferences within the built-in theory are compressed into complex steps. The considered base calculus relies on a certain algorithmic realization of the following necessary and sufficient criterion for the unsatisfiability of a set of clauses M :

Criterion: There exist a set M' of copies of clauses from M and a substitution σ such that each disjunct of the disjunctive normal form of M' contains a pair of literals L and $\neg L'$ such that σ unifies the atoms L and L' [And81, Bib82]

Pairs consisting of occurrences of complementary literals L and $\neg L'$ in different clause copies are called connections. By some abuse also the literal pairs themselves will be called connections. The conjunctions $\sigma(L) \wedge \sigma(\neg L')$ play the role of elementary contradictions. Altogether they are sufficient to refute the considered set of clauses. In the case of a built-in theory $\mathcal{T}$ we have to consider more general sets of literals as connections. They will be called $\mathcal{T}$ -connections or theory connections if it is clear which theory is in mind. In particular theory connections may consist of more or less than two literals. Let us discuss a simple example (modified from [BB87]).

Example 1.1 Let us consider a signature with unary predicates *politician*, *company*, *gives*, a binary predicate *likes* and constant *any* and *some*. Let $\mathcal{T}_1$ be the theory with this signature and the following axioms:

$$\neg \text{likes}(X, Y) \vee \text{gives}(Y)$$

$$\neg \text{politician}(X) \vee \neg \text{likes}(X, Y) \vee \text{company}(Y)$$

The following matrix is a $\mathcal{T}_1$ -theorem. The predicate and constant names will be abbreviated by their initial letters in order to achieve a concise notation.

Every disjunct of the disjunctive normal form of this matrix contains a $\mathcal{T}_1$ -connection. We indicate theory connections by (multiple) edges and enumerate them by the label of the edges.

The indicated connections form a so called spanning mating. Below they are written as columns from left to right according to their number.

$$\left\{ \begin{array}{c} p(a) \\ l(X, s) \\ \neg c(Y) \end{array} \right\} \quad \left\{ \begin{array}{c} l(X, s) \\ \neg g(Y) \end{array} \right\} \quad \left\{ \begin{array}{c} p(a) \\ \neg p(X) \end{array} \right\} \quad (2)$$

Those connections become $\mathcal{T}_1$ -unsatisfiable if the substitution $\{X \mapsto a, Y \mapsto s\}$ is applied to the matrix. The following picture illustrates the cooperation between background and fore-

ground reasoner when treating the $\mathcal{T}_1$ -connection $\left\{ \begin{array}{c} p(a) \\ l(X, s) \\ \neg c(Y) \end{array} \right\}$. The foreground reasoner detects this literal set as a candidate for a $\mathcal{T}_1$ -connection and asks the background reasoner to find a substitution making it $\mathcal{T}_1$ -unsatisfiable. The background reasoner returns the substitution $\{X \mapsto a, Y \mapsto s\}$.

In order to built-in a theory $\mathcal{T}$ one has to characterize for $\mathcal{T}$ a set $\mathcal{U}$ which contains "enough" $\mathcal{T}$ -connections and to make sure that the appropriate $\mathcal{T}$ -unifiers of these connections may be found. This idea will be formalized by the notions *complete set $\mathcal{U}$ of $\mathcal{T}$ -connections* (cf. definition 3.2) and *$\mathcal{T}$ -unification problem in a set $\mathcal{U}$ of $\mathcal{T}$ -connections* (cf. definition 3.4). In some cases it is suitable to restrict the set of admissible substitutions $\mathcal{S}$ which may be returned by the background reasoner in order to meet soundness requirements. This is the case for constraint theories which may be treated by A. Frisch's so called substitutional framework. In the present paper we will not treat this detail formally and refer the reader to a forthcoming paper. The sets $\mathcal{U}$ and $\mathcal{S}$ describe the interface of the background reasoner to the foreground reasoner sufficiently. The algorithmic solution of the $\mathcal{T}$ -unification problem describes an abstract implementation of the background reasoner. Finally we need a decidable superset $\mathcal{C}$ of $\mathcal{U}$ which contains candidates for $\mathcal{T}$ -connections which may be detected by the foreground reasoner and cause it to call the service of the background reasoner. This distinction between the sets $\mathcal{C}$ of candidates and $\mathcal{U}$ of $\mathcal{T}$ -connections is necessary for practical reasons. The set $\mathcal{U}$ might be quit

complex - for example NP-complete for the theory of equality. Thus, to decide whether a set of literals is in $\mathcal{U}$ is the task of the background reasoner. The set $\mathcal{C}$ should be decidable with low complexity.

For the empty theory the set of all complementary pairs is complete and a most general unifier may be computed as a uniform representative of all unifiers. Similar results may be proved for a number of interesting theories. Taxonomic, constraint and equational theories will be discussed in this paper. In some cases certain restrictions of the query language are essential. For the discussed examples the completeness of a connection calculus with built-in theory may be proved applying our completeness theorem (5.1). We demonstrate that approach for an adaption of G. Neugebauer's and T. Schaub's pool calculus as foreground reasoner. The pool calculus is one of the representations of the connection method. We give a rigorous proof for the completeness of the pool calculus with a model elimination like strategy, i.e. with a quite restrictive inference rule. This way we obtain not only the extension of the connection method to theory reasoning but also a completeness result for a version with a lower branching factor.

Related work. Theory connections are a generalization of W. Bibel's eq-connections (cf. [Bib82]). The extension of resolution to theory resolution is due to M. Stickel. Many improvements of resolution have been shown as special kinds of theory resolution in [Sti85]. There are no completeness results for the first-order case in [Sti85]. A technique for completeness proofs of first-order connection based theory reasoning calculi has been developed by the present author in [Pet90, Pet92, Pet93]. An independent approach has been studied by P. Baumgartner in the framework of model elimination tableaux, c.f. [Bau92] for a concise presentation. For a newer overview concerning theory reasoning see [BFP92]. An application of theory resolution, the knowledge representation system KRYPTON with the concept language KL-ONE, has been described by R. J. Brachman, V. P. Gilbert and H. J. Levesque [BFL83]. N. V. Murray and E. Rosenthal consider theory links for their path dissolution calculus [MR87]. The connection graph procedure with built-in theories is due to M. Stickel [Sti85] and has been implemented in the Markgraf Karl prover [OS91] too. J. Gallier et al. [GNPS92] proved the NP-completeness of the so called rigid E-unification. Their claim that the the mating method [And81] together with their procedure for the simultaneous rigid E-unification problem is complete suffers from a gap in the proof. A recent result of B. Beckert [Bec93] concerning rigid E-unification allows a rigorous completeness proof of a connection calculus with equality. Another approach considering theories given by classes of models has been presented by H.-J. Brckert [Br90]. Our approach carries over to that case if one considers a complete set for theory connections of each model of the considered class.

2 Preliminaries

2.1 Clause logic

We assume that the reader is familiar with the basic notions of first-order logic in clausal form (cf. [Lov78]).

A *clause* is a multiset of literals. A *matrix* is a multiset of clauses. Multisets will be denoted as sequences of their elements. A (partial) *path (in) through* a matrix M contains (at most) exactly one literal from each clause of M . We will distinguish paths and clauses with respect to their intended meaning. We consider a *clause* $L_1, \dots, L_n$ to be the abbreviation of the universal closure $\bar{\forall}(L_1 \vee \dots \vee L_n)$ of the disjunction of its elements. A *path* $L_1, \dots, L_n$ abbreviates the existential closure $\bar{\exists}(L_1 \wedge \dots \wedge L_n)$ of the conjunction of its elements. Clauses will be abbreviated also by Γ, Δ, Λ or Θ . Γ, Δ means the union $\Gamma \cup \Delta$ and Γ, L means $\Gamma \cup \{L\}$. Paths will be denoted also by p or q . A set of partial paths in a matrix M is called a *mating* in M . The meaning of a matrix $\{C_1, \dots, C_n\}$ is the conjunction $\bar{\forall}C_1 \wedge \dots \wedge \bar{\forall}C_n$. The meaning of a mating $\{p_1, \dots, p_n\}$ is the disjunction $\bar{\exists}p_1 \vee \dots \vee \bar{\exists}p_n$. A partial path u in matrix M is *spanning* a path p through M if $u \subseteq p$. A mating U in a matrix M is *spanning* if for every path p through M exists an element $u \in U$ which is spanning p . Clearly, if a mating U is unsatisfiable in a theory $\mathcal{T}$ and U is spanning in a matrix M then M is $\mathcal{T}$ -unsatisfiable too.

If L is a positive literal then $\bar{L}$ denotes the literal $\neg L$. If L has the form $\neg K$ then $\bar{L}$ denotes the literal K . If p is the path $L_1, \dots, L_n$ then $\bar{p}$ denotes the clause $\bar{L}_1, \dots, \bar{L}_n$. And, vice versa, if Γ is the clause $L_1, \dots, L_n$ then $\bar{\Gamma}$ denotes the path $\bar{L}_1, \dots, \bar{L}_n$.

The set of variables occurring in a term t , literal L , clause Γ or path p will be denoted by $Var(t)$, $Var(L)$, $Var(\Gamma)$ or $Var(p)$ respectively.

A *substitution* is a mapping from the set of variables into the set of terms which is almost everywhere equal to the identity. The *domain* of a substitution σ is the set $D(\sigma) = \{X \mid \sigma(X) \neq X\}$. The set of *variables introduced by* σ is the set $I(\sigma) = \bigcup_{x \in D(\sigma)} Var(\sigma(X))$. A substitution σ may be extended canonically to a mapping from the set of terms into the set of terms. This extension will be denoted by σ too. The *composition* $\sigma\theta$ of substitutions σ and θ is the substitution which assigns to every variable X the term $\theta(\sigma(X))$. A substitution σ is called *idempotent* if $\sigma = \sigma\sigma$. A substitution σ is idempotent iff $D(\sigma) \cap I(\sigma) = \emptyset$. Substitutions σ and λ are identical with respect to a set of variables V iff for each $X \in V$ holds $\sigma(X) = \lambda(X)$. This will be denoted by $\sigma =_V \lambda$. We say that σ is *more general than* λ if there exists a substitution δ such that $\sigma\delta =_{D(\lambda)} \lambda$.

A set of matrices which is closed w.r.t. the application of substitutions, forming submatrices and paths will be called a *query language*.

2.2 Interpretations

An interpretation I consists of a non-empty domain D and maps

- an element of D to each constant symbol,
- an n – ary function to each n – ary function symbol and
- an n – ary relation to each n – ary predicate symbol.

Let Σ be a signature. Then the *Herbrand universe* for Σ is the set of variable-free terms that can be generated using the constant and function symbols occurring in Σ . If there is no constant symbol in Σ then we use a single constant symbol a . An *Herbrand interpretation* over Σ is any interpretation over the Herbrand universe for Σ taken as domain which maps constants to itself and any function symbol f to a mapping $\lambda t_1, \dots, t_n. f(t_1, \dots, t_n)$.

The *atome set* or *Herbrand base* of Σ is the set of atoms $p(t_1, \dots, t_n)$ where p is an n -place predicate symbol occurring in Σ and the t_j are elements of the Herbrand universe of Σ . Any Herbrand interpretation H over Σ can be represented by the set of all ground atoms $p(t_1, \dots, t_n)$ such that $(t_1, \dots, t_n)$ is element of the relation which H assigns to p . Below we will identify Herbrand interpretations with their representation.

Let H be an Herbrand interpretation over Σ . H satisfies

- a ground atom A if $A \in H$,
- a ground literal $\neg A$ if $A \notin H$.
- a disjunction of ground or closed formulas $F_1 \vee \dots \vee F_n$ if H satisfies F_i for some $i = 1, \dots, n$,
- a conjunction of ground or closed formulas $F_1 \wedge \dots \wedge F_n$ if H satisfies F_i for all $i = 1, \dots, n$,
- a universal closure $\forall F$ if H satisfies all ground substitutions of F ,
- an existential closure $\exists F$ if H satisfies some ground substitution of F .
- a clause $\{L_1, \dots, L_n\}$ if H satisfies the universal closure $\bar{\forall}(L_1 \vee \dots \vee L_n)$,
- a matrix $\{\Gamma_1, \dots, \Gamma_n\}$ if H satisfies the conjunction $\bar{\forall}\Gamma_1 \wedge \dots \wedge \bar{\forall}\Gamma_n$.
- a path $\{L_1, \dots, L_n\}$ if H satisfies the existential closure $\bar{\exists}(L_1 \wedge \dots \wedge L_n)$,
- a mating $\{p_1, \dots, p_n\}$ if H satisfies the disjunction $\bar{\exists}p_1 \vee \dots \vee \bar{\exists}p_n$.

A syntactic entity S of the mentioned above kind is *satisfiable* if there is a Herbrand interpretation satisfying S and *unsatisfiable* else.

Let $\mathcal{A}$ be a set of clauses. Then the triple $\mathcal{T} = (\mathcal{L}, \models, \mathcal{A})$ will be called a *theory* with language $\mathcal{L}$ and entailment operator $\models$. The elements of $\mathcal{A}$ are called the *axioms* of the theory. A $\mathcal{T}$ -model is a Herbrand interpretation satisfying $\mathcal{T}$. A syntactic entity S of the mentioned above kind is $\mathcal{T}$ -satisfiable if there is a $\mathcal{T}$ -model satisfying S . Otherwise S is called $\mathcal{T}$ -unsatisfiable.

Let $\mathcal{E}$ be the theory of equality, i.e. the theory with a set of axioms consisting of clauses expressing reflexivity, symmetry, transitivity and functional and predicative substitutivity of the equality sign $=$. The $\mathcal{E}$ -models in a given signature are those Herbrand interpretations which interpret the equality sign by a congruence relation.

3 Complete sets of theory connections

In the present section we generalize the concepts of a connection and of a unifier of a connection towards theory reasoning. This generalization follows the treatment of equality in [Bib82]. We formalize what it means to have for a given theory "enough" theory connections in order to refute all theory unsatisfiable matrices which belong to a given query language. We formulate a Herbrand theorem by use of this notion. The notion of a complete set of unifiers for a theory connection generalizes the notion of complete set of theory unifiers of a pair of terms. The notion of solvable unification problem describes that for each element of a given complete set of theory connections may be constructed a complete set of theory unifiers. Let $\mathcal{T}$ be a theory given as a set of clauses.

Definition 3.1 ($\mathcal{T}$ -complementary)

A path u is called $\mathcal{T}$ -complementary iff $\exists (\bigwedge_{L \in u} L)$, i.e. the existential closure of the conjunction of the elements of u , is $\mathcal{T}$ -unsatisfiable. $\square$

Lemma 3.1 (Alternative characterization of $\mathcal{T}$ -complementary)

Let u be a path. By Δ denote the set of all atoms in u and by Γ the set of atoms L such that $\neg L \in u$. Then u is $\mathcal{T}$ -complementary iff $\mathcal{T} \models \forall (\Delta \rightarrow \Gamma)$.

The notion of complementarity is defined via the $\mathcal{T}$ -unsatisfiability according to the negative representation in the present paper. It is the advantage of the notion of complementarity that the definition of other notions of the calculus may rely on the $\mathcal{T}$ -complementarity notion and become independent on the chosen (positive or negative) representation.

Definition 3.2 ($\mathcal{T}$ -unifier, $\mathcal{T}$ -connection)

- (1) A substitution σ will be called a $\mathcal{T}$ -unifier of a set of literals u iff $\sigma(u)$ is $\mathcal{T}$ -complementary.
- (2) A substitution σ will be called a *simultaneous* $\mathcal{T}$ -unifier of a set U of sets of literals iff for every $u \in U$ $\sigma(u)$ is $\mathcal{T}$ -complementary.
- (3) A set of literals u is called a $\mathcal{T}$ -connection if there exists a $\mathcal{T}$ -unifier for u .
- (4) A partial path p in a matrix M which is a representative² of a $\mathcal{T}$ -connection will be called a $\mathcal{T}$ -connection in M .

$\square$

Now we define the notion of a complete set of $\mathcal{T}$ -connections $\mathcal{U}$. The completeness of $\mathcal{U}$ will be relative to a given query language $\mathcal{Q}$.

Definition 3.3 ($\mathcal{U}$ -connection in a matrix, complete set of $\mathcal{T}$ -connections $\mathcal{U}$, spanning $\mathcal{U}$ -mating)

Let $\mathcal{Q}$ be a query language. Let $\mathcal{U}$ be a decidable set of $\mathcal{T}$ -connections such that $\mathcal{U} \subseteq \mathcal{Q}^3$. Moreover, $\mathcal{U}$ is closed w.r.t. instantiation by $\mathcal{T}$ -unifiers.

²Multisets are equivalence classes. Paths may be considered as representatives of multisets. We neglect the distinction between multisets and their representatives.

³We always can consider a set as a multiset. In this sense $\mathcal{U}$ may be a subset of $\mathcal{Q}$.

- (1) A $\mathcal{U}$ -connection in a matrix $M \in \mathcal{Q}$ is a partial path p in M which is a representative of an element $u \in \mathcal{U}$.
- (2) The set $\mathcal{U}$ will be called *complete w.r.t. the query language $\mathcal{Q}$* if
 - (a) for each $\mathcal{T}$ -complementary ground path $p \in \mathcal{Q}$ exists $u \in \mathcal{U}$ such that $u \subseteq p$.
 - (b) for each $\mathcal{T}$ -complementary ground path of the form $\sigma(u) \in \mathcal{Q}$ such that $u \in \mathcal{Q}$ holds $u \in \mathcal{U}$.
- (3) Any set of $\mathcal{U}$ -connections in a matrix M is called a $\mathcal{U}$ -mating in M . A $\mathcal{U}$ -mating U is said to *span a set of partial paths $\mathcal{P}$* in a matrix M if for every $p \in \mathcal{P}$ contains a $\mathcal{U}$ -connection. U is said to *span a matrix M* if U spans the set of all paths through M .

□

Remark: The notion of a $\mathcal{T}$ -connection appeals to the semantical property to have a $\mathcal{T}$ -unifier. To be a $\mathcal{U}$ -connection should be a decidable property because we expect the background reasoner to return a definite negative answer in case of lack of $\mathcal{T}$ -unifiers for a given pattern. The $\mathcal{T}$ -completeness of a set of $\mathcal{T}$ -connections $\mathcal{U}$ formalizes the idea that there are enough $\mathcal{T}$ -connections in order to prove all theorems within a query language by the help of the "British Museum" procedure, i.e. by the exhaustive construction of all ground instantiations and checking them for the existence of a spanning $\mathcal{U}$ -mating. The formal expression of this claim is the Herbrand theorem.

Theorem 3.1 (*Herbrand theorem*)

Let $\mathcal{Q}$ be a query language, $\mathcal{U}$ a set of $\mathcal{T}$ -connections which is complete w.r.t. to $\mathcal{Q}$. Then for every $\mathcal{T}$ -unsatisfiable matrix $M \in \mathcal{Q}$ there exists a set M' of copies of clauses from M , a $\mathcal{U}$ -mating U which is spanning M' and a substitution σ such that $\sigma(u)$ is $\mathcal{T}$ -complementary for each $u \in U$.

Proof. For every $\mathcal{T}$ -unsatisfiable matrix $M \in \mathcal{Q}$ there exists a set M' of ground instances of clauses from M which is $\mathcal{T}$ -unsatisfiable. The dual version of this claim for positive representation has been proved in [RS70] (for formulas in prenex form) and in [Pet92] for arbitrary formulas. M' has the form $\sigma(M'')$ where M'' is a set of copies of clauses from M . Moreover, $M', M'' \in \mathcal{Q}$ because the query language $\mathcal{Q}$ is closed w.r.t. instantiation. Because

- (1) M' is ground and $\mathcal{T}$ -unsatisfiable,
- (2) $\mathcal{Q}$ is closed w.r.t. forming paths and
- (3) $\mathcal{U}$ is complete w.r.t. $\mathcal{Q}$

every path through M' contains a ground $\mathcal{T}$ -connection which is element of $\mathcal{U}$ (condition 2a of definition 3.2). Let $M' = \sigma(M'')$. Then every of those connections has the form $\sigma(u)$ where $u \in \mathcal{Q}$ because $\mathcal{Q}$ is closed w.r.t. forming paths and instantiation. Therefore by condition 2b of definition 3.2 we have $u \in \mathcal{U}$. Thus, there is a $\mathcal{U}$ -mating U spanning M'' . Moreover, for each $u \in U$ $\sigma(u)$ is $\mathcal{T}$ -complementary. **q.e.d.**

The elements of the spanning $\mathcal{U}$ -mating and the substitution σ which exist according to theorem 3.1 should be found incrementally by use of a kind of unification. Now we introduce the notions for a formal treatment of this kind of unification.

Definition 3.4 (more general T -unifier, T -unification problem in $\mathcal{U}$)

- (1) Let ϱ and σ be T -unifiers of a path u such that $D(\varrho), D(\sigma) \subseteq \text{Var}(u)$. Then ϱ is called *more general than* σ if there exists η such that $\varrho\eta =_{D(\sigma)} \sigma$. This will be denoted by $\varrho \leq \sigma$.
- (2) A set S of T -unifiers of a multiset $u \in \mathcal{U}$ will be called *complete* if for each T -unifier σ exists a substitution $\varrho \in S$ such that $\varrho \leq \sigma$.
- (3) Let $\mathcal{U}$ be a set of multisets of literals. We say that the *T -unification problem in $\mathcal{U}$ is solvable* if for every $u \in \mathcal{U}$ there exists an enumerable complete set of T -unifiers for u .

□

Proposition 3.1 Suppose that the T -unification problem is solvable for a set of T -connections $\mathcal{U}$ which is subset of a query language $\mathcal{Q}$. Then for every simultaneous T -unifier θ of a non-empty set of T -connections $U \subseteq \mathcal{U}$ a simultaneous T -unifier σ of U and a substitution σ' satisfying $\theta = \sigma \circ \sigma'$ may be found incrementally.

Proof. Let S_u denote the complete set of T -unifiers for each $u \in \mathcal{U}$. Moreover, let θ be the simultaneous T -unifier of a finite, non-empty set of T -connections $U \subseteq \mathcal{U}$.

Claim: For each enumeration $u_1, \dots, u_n$ of the elements of a non-empty set of T -connections $U \subseteq \mathcal{U}$ may be constructed sequences $\{\sigma_i\}_{i=1}^n, \{\eta_i\}_{i=0}^n, \{\varrho_i\}_{i=0}^n$ such that

- (1) $\eta_0 = \theta$ and $\varrho_0 = \{ \}$ and
- (2) for every $i, 1 \leq i \leq n$
 - (a) $\sigma_i \in S_{\varrho_{i-1}(u_i)}$,
 - (b) $\sigma_i \eta_i = \eta_{i-1}$ and
 - (c) $\varrho_i = \varrho_{i-1} \sigma_i$.
 - (d) $\theta = \varrho_i \eta_i$.

This claim may be proved by induction on the cardinality n of U .

Base case: For $n = 1$ any T -unifier of u_1 is a simultaneous T -unifier of $U = \{u_1\}$. The substitutions η_1 and $\sigma_1 \in S_{u_1}$ which satisfy condition 2b exist because of the solvability of the unification problem w.r.t. $\mathcal{U}$. ϱ_1 may be constructed according to condition 2c.

Step case: Suppose the claim has been proved for all simultaneously T -unifiable sets of T -connections with cardinality n and let $U' = U \cup \{u_{n+1}\}$. Let $\{\sigma_i\}_{i=1}^n, \{\eta_i\}_{i=0}^n, \{\varrho_i\}_{i=0}^n$ be the sequences which exist for $U = \{u_1, \dots, u_n\}$ by the induction hypothesis. We have $\theta = \varrho_n \eta_n$ from condition 2d. Then $\varrho_n(u_{n+1}) \in \mathcal{Q}$ and $\eta_n(\varrho_n(u_{n+1})) \in \mathcal{Q}$ because $u_{n+1} \in \mathcal{U} \subseteq \mathcal{Q}$ and $\mathcal{Q}$ is closed w.r.t. instantiation. Then from condition 2b of definition 3.3 follows $\varrho_n(u_{n+1}) \in \mathcal{U}$. Because η_n is a T -unifier of $\varrho_n(u_{n+1})$ there exist substitutions $\sigma_{n+1} \in S_{\varrho_n(u_{n+1})}$ and η_{n+1} such that $\eta_n = \varrho_{n+1} \eta_{n+1}$. Setting $\varrho_{n+1} = \varrho_n \sigma_{n+1}$ we obtain $\theta = \varrho_{n+1} \eta_{n+1}$.

q.e.d.

In the following subsections we illustrate the introduced notions by a number of representative examples.

3.1 Taxonomic reasoning

Knowledge representation languages allow to represent taxonomic knowledge and to draw inferences concerning the relations between several notions. A general reasoner may be combined with such a knowledge representation system which then plays the role of the background reasoner. A classical example is KRYPTON [BFL83]. KRYPTON represents taxonomies in its terminological component called the T-box. The T-box deals with the formalizations of concepts like “animal which eats meat”. If *animal* and *eatsmeat* are given concepts then the T-box may be told that a concept, say *wolf*, means an animal which eats meat. This knowledge may be expressed by a theory $\mathcal{T}_2$ consisting of the following clauses⁴:

$$\begin{aligned} &\neg wolf(X), animal(X) \\ &\neg wolf(X), eatsmeat(X) \end{aligned} .$$

A suitable set of $\mathcal{T}_2$ -connection candidates is the set of literal sets of the following three forms:

$$\left\{ \begin{array}{c} p(s) \\ \neg p(t) \end{array} \right\} \quad \left\{ \begin{array}{c} wolf(s) \\ \neg eatsmeat(t) \end{array} \right\} \quad \left\{ \begin{array}{c} wolf(s) \\ \neg animal(t) \end{array} \right\} \quad (3)$$

where p is an arbitrary predicate and s and t are arbitrary terms. The set $\mathcal{U}_2$ of all candidates such that the terms s and t are unifiable forms a complete set of theory connections for $\mathcal{T}_2$. The unification problem in $\mathcal{U}_2$ is solvable. It simply reduces to the universal unification of the terms s and t occurring in (3). It is interesting that in KRYPTON there may be defined theories which are not open. Such a theory must be Skolemized in order to achieve the completeness of the prover with built-in theory. For a more detailed discussion of this topic see [Pet92].

3.2 Reasoning in constraint theories

In constraint theories the range of a quantifier may be restricted by a certain formula. Those so called constraints are built from predicate symbols which are forbidden anywhere else in a formula. The built-in theory is formulated in terms of the constraint predicates.

Example 3.1 Let us consider a signature with unary predicates *animal*, *eatsmeat*, *wolf*, *wild* and *dangerous* and a constant *rob*. The predicates *animal*, *eatsmeat* and *wolf* are the constraint predicates. Let $\mathcal{T}_3$ be the theory with this signature and the following clauses as axioms:

$$\begin{aligned} &wolf(rob) \\ &\neg wolf(X), animal(X) \\ &\neg wolf(X), eatsmeat(X) \end{aligned}$$

We examine the $\mathcal{T}_3$ -theorem which says that if all meat eating beings are dangerous then there are animals which are dangerous. The negation of that theorem may be formulated by use of restricted quantifiers

$$\forall_{X:eatsmeat} dangerous(X) \rightarrow \exists_{Y:animal} dangerous(Y). \quad (4)$$

⁴For the notations of KRYPTON see [BFL83].

If we translate the subformula $\forall_{X:\text{eatsmeat}} \text{dangerous}(X)$ by $\forall X(\text{eatsmeat}(X) \rightarrow \text{dangerous}(X))$ and $\exists_{Y:\text{animal}} \text{dangerous}(Y)$ by $\exists Y(\text{animal}(Y) \wedge \text{dangerous}(Y))$. With this translation we obtain a clause form of the query (4) which consists of two clauses which have been written as rows of the following matrix.

$$\left[\begin{array}{c} \textcircled{1} \\ \neg \text{eatsmeat}(X) \text{ dangerous}(X) \\ \textcircled{3} \quad \quad \quad \textcircled{2} \\ \neg \text{animal}(Y) \neg \text{dangerous}(Y) \end{array} \right] \quad (5)$$

A variable bound by a restricted quantifier ranges over the set of individuals which satisfy the constraint. This information may be expressed syntactically by attaching the constraint to the variable (cf. [Fri91]). For example $\forall_{X:\text{eatsmeat}}$ ranges over individuals which satisfy $\text{eatsmeat}(X)$. Formula (4) has then the following matrix form with variables with constraints:

$$\left[\begin{array}{c} \neg \text{dangerous}(Y : \text{animal}) \\ \text{dangerous}(X : \text{eatsmeat}) \end{array} \right] \quad (6)$$

The matrix size and the size of a minimal spanning mating in (6) is smaller than in (5). Only one connection

$$\left\{ \begin{array}{c} \text{dangerous}(X : \text{eatsmeat}) \\ \neg \text{dangerous}(Y : \text{animal}) \end{array} \right\} \quad (7)$$

is needed for a spanning mating. However, unification needs more effort now. The unifier for connection (7) has the form $\sigma = \{X, Y \mapsto (Z : \text{wolf})\}$. It has the remarkable property that there exists a substitution $\sigma' = \{Z \mapsto \text{rob}\}$ such that

$$\mathcal{T}_3 \models \sigma'(\sigma(\text{eatsmeat}(X))) \text{ and } \mathcal{T}_3 \models \sigma'(\sigma(\text{animal}(Y))).$$

This property is called $\mathcal{T}_3$ -admissibility.

Let $\mathcal{Q}$ be a query language which consists of all such matrices which where all occurring constraints are satisfiable. If the constraint theory has a least model then set of literal sets of the form 7 is a suitable set of candidates. Moreover, the set $\mathcal{U}_3$ of all those $\mathcal{T}_3$ -connections of the form 7 which have a $\mathcal{T}_3$ -admissible unifier is complete w.r.t. $\mathcal{Q}$. Unfortunately, the transformation of arbitrary first-order formulas into clause form needs Skolemization which might include a transformation of the underlying theory. Due to this transformation it may lose its good properties. In [Fri91] may be found a sufficient criterion for the solvability of the unification problem.

3.3 Reasoning with Equality: at first a restricted case

We apply now our approach to a restricted form of equational reasoning. This application is motivated by the translation of certain modal logics into first-order logic following P. Enjalbert

et al. [DEL90]. For example let us consider a first-order modal logic S4 with modal operators $\Box$ and $\Diamond$. Then the modal language may be translated into a sorted first-order language with additional sorts **W**, for worlds, and **A**, for actions performing the transliteration from one world to another. For example the formula⁵

$$\Box p(X) \rightarrow \Box \Box p(Y)$$

will be translated into

$$\forall A p(A, X) \rightarrow \forall b \forall c p((\epsilon!b)!c, Y). \quad (8)$$

Here ϵ means a constant of sort **A** and $!$ a binary operation which has a first argument of sort **A**, a second argument of sort **W** and returns a result of sort **W**. Moreover each predicate symbol obtains an additional argument (the first one) of sort **W**. The logic S4 is characterized by an equational theory $\mathcal{T}_4$ with operation symbols $*$ and $!$ of sort **A** and $!$ described above. Sort **A** with the operations $*$ and $!$ forms a monoid which is operating by the operation $!$ on sort **W**.

The formula 8 gives after negation and transformation into clause form the following matrix

$$\left[\begin{array}{c} p(A, X) \\ \searrow \\ \neg p((\epsilon!b)!c, Y) \end{array} \right]$$

In the considered equational theory holds $\epsilon!(b * c) = (\epsilon!b)!c$. Therefore, the indicated connection has the $\mathcal{T}_4$ -unifier

$$\{A \mapsto b * c, X \mapsto Y\}.$$

The set of formulas $\mathcal{Q}_4$ which are obtained by translation form a query language which satisfies the following restrictions. For every query $M \in \mathcal{Q}_4$ holds

- (1) the equality symbol does not occur in M and
- (2) if M' has been obtained from M by exhaustive application of the rewrite rule

$$X!(Y * Z) \sim (X!Y)!Z$$

and $t!X$ and $s!X$ are subterms of M' then s and t are equal.

Due to the first restriction set of $\mathcal{T}_4$ connections of the form $\left\{ \begin{array}{c} P(\underline{s}) \\ \neg P(\underline{t}) \end{array} \right\}$ is complete w.r.t. $\mathcal{Q}_4$. The unification problem reduces to the unification of sequences of terms $\underline{s}$ and $\underline{t}$ modulo the equational theory $\mathcal{T}_4$ which characterizes the modal system. The second restriction causes that the unification problem becomes solvable [DEL90]. Moreover, for every $\mathcal{T}_4$ -connection exists a finite complete set of unifiers.

⁵Uppercase latin letters denote variables which become universally quantified after negating and transforming the considered formula into negation normal. Lowercase latin letters denote constants or variables which become existentially quantified after that transformation.

3.4 Reasoning with Equality: the general case

The theory of equality $\mathcal{E}$ considered with arbitrary queries seems to be more difficult than the examples considered above. All $\mathcal{E}$ -connections have the form $l_1 \doteq r_1, \dots, l_m \doteq r_m, \neg s \doteq t$. The set of all such literal sets can be chosen as a set of candidates for $\mathcal{E}$ -connections. The set of all $\mathcal{E}$ -connections is decidable. There are two procedures one of J. Gallier et al. [GNPS92] and the other of B. Beckert [Bec93] which decide whether for given literal set $l_1 \doteq r_1, \dots, l_m \doteq r_m, \neg s \doteq t$ exists an $\mathcal{E}$ -unifier⁶. Both allow to compute for a given $\mathcal{E}$ -unifier θ of an $\mathcal{E}$ -connection $l_1 \doteq r_1, \dots, l_m \doteq r_m, \neg s \doteq t$ an $\mathcal{E}$ -unifier σ such that there exists a substitution η such that $\sigma\eta$ and θ are equal modulo the equations $\theta(l)_1 \doteq \theta(r)_1, \dots, \theta(l)_m \doteq \theta(r)_m$.

This property is weaker than that defined in definition 3.4. In particular it is too weak for the construction in proposition 3.1 because the set of equations may be different in each of the elements of mating. J. Gallier et al. claim that their method can be extended in order to treat also the simultaneous unification problem [GNPS92]. Unfortunately their proof of the completeness of that extended procedure in [GNPS92] relies on a minimality constraint imposed on $\mathcal{E}$ -unifiers which is in general not satisfied. Therefore, the claim of J. Gallier et al. remains unproved.

Yet another approach is that of J. Goubault [Gou93]. It relies on the computation of congruence closures. Again the result is weaker than the property which is needed for the construction in proposition 3.1. J. Goubault's method allows to compute for a given $\mathcal{E}$ -unifier θ of an $\mathcal{E}$ -connection $l_1 \doteq r_1, \dots, l_m \doteq r_m, \neg s \doteq t$ an $\mathcal{E}$ -unifier σ such that there exists a substitution η such that $\sigma\eta$ and θ to the equations $l_1 \doteq r_1, \dots, l_m \doteq r_m$ give the same set of equational consequences.

B. Beckert proposes a solution which allows to compute a complete set of $\mathcal{E}$ -unifiers in the sense of definition 3.4 in two steps. In the **first step** for a given $\mathcal{E}$ -unifier θ of an $\mathcal{E}$ -connection $l_1 \doteq r_1, \dots, l_m \doteq r_m, \neg s \doteq t$ $\mathcal{E}$ -unifiers σ for which exist substitutions η such that $\sigma\eta$ and θ are equal modulo the equations $\theta(l)_1 \doteq \theta(r)_1, \dots, \theta(l)_m \doteq \theta(r)_m$. This step is completion based. In a **second step** $\mathcal{E}$ -unifiers σ' for which exist substitutions η such that $\sigma\eta$ and θ are equal. Unfortunately the second step does not take advantage of term orderings in order to narrow the search space. Thus, combining unifiers (actually this takes place when simultaneous unifiers are computed from given unifiers of single $\mathcal{E}$ -connections) will be less efficient than computing single $\mathcal{E}$ -unifiers.

For this reason it remains an interesting question whether there exists a modification of the procedure for the construction $\mathcal{E}$ -unifiers for single connections may be extended in order to compute simultaneous $\mathcal{E}$ -unifiers directly. Perhaps this is not contradictory to the existence of counter example to the claim of Gallier et al. which has been mentioned above. Actually, though there are examples of spanning $\mathcal{E}$ -matings in matrices such that there is no simultaneous $\mathcal{E}$ -unifier with the minimality property which is sufficient to use the method of J. Gallier et al. in all known to us examples it is possible to construct another mating perhaps in another instantiation which has a simultaneous $\mathcal{E}$ -unifier which might be constructed by the method of Gallier et al.

⁶In [GNPS92] J. Gallier et al. introduced the notion rigid E-unifier of s and t for an $\mathcal{E}$ -unifier of the $\mathcal{E}$ -connection $l_1 \doteq r_1, \dots, l_m \doteq r_m, \neg s \doteq t$ where E denotes the set of equations $\{l_1 \doteq r_1, \dots, l_m \doteq r_m\}$. Earlier the notion eq-connection and eq-unifier has been used by W. Bibel [Bib82].

4 A total theory connection calculus

In the present section we introduce a generalization of the pool calculus [NS91] towards theory reasoning. In order to develop this connection calculus we take the Herbrand theorem as start point. We explain this calculus by the help of the sample matrix (1).

Example 4.1 Let us consider a signature with unary predicates *politician*, *company*, *gives*, a binary predicate *likes* and constants *any* and *some*. Let $\mathcal{T}_3$ be the theory with this signature and the following axioms:

$$\neg \text{likes}(X, Y), \text{ gives}(Y)$$

$$\neg \text{politician}(X), \neg \text{likes}(X, Y), \text{ company}(Y)$$

The following matrix is a $\mathcal{T}_3$ -theorem. In order to achieve a more concise notation we indicate theory connections by (multiple) edges and enumerate them by the label of the edges. Moreover the predicate and constant names will be abbreviated by there initial letters. The indicated connections form a spanning mating. These connections become $\mathcal{T}_3$ -unsatisfiable if the substitution $\{X \mapsto a, Y \mapsto s\}$ is applied to the matrix.

The existence of the spanning mating in (9) and of the unifier may be proved in a three step derivation. This derivation is displayed below by a sequence of structured matrices. Each of these matrices represents a derivation state. In every state a set of paths remains to be examined in subsequent derivation steps. In the initial matrix (the leftmost) all paths which continue the empty path via the literal which is pointed by the arrow have to be considered. The pointed literal will be called the *actual goal*. In the second matrix the first theory connection has been found. It remains to consider those paths which continue either the path $p(b), l(X, a)$ (dashed boxed literals) via the literal $\neg g(Y)$ (pointed by the arrow) or the path $p(b)$ (dashed boxed literal) via the literal $\neg p(X)$. All paths of the former group contain the second theory connection which is found in the second step. In the third matrix it remains to consider the paths of the latter group. All of them contain the third connection. The proof is completed if this connection has been found. Consequently no more actual goal appears in the rightmost matrix.

$$\begin{array}{ccc}
\left[\begin{array}{cc} p(a) \\ l(X, s), & \neg p(X) \\ \neg c(Y) & \neg g(Y) \end{array} \right] & \vdash_{\text{extension}} & \left[\begin{array}{cc} \textcircled{1} p(a) \\ l(X, s), & \neg p(X) \\ \neg c(Y) & \neg g(Y) \end{array} \right] \\
\vdash_{\text{reduction}} & & \vdash_{\text{reduction}} \\
\left[\begin{array}{cc} \textcircled{1} p(a) \\ l(X, s), & \neg p(X) \\ \neg c(Y) & \textcircled{2} \neg g(Y) \end{array} \right] & \vdash_{\text{reduction}} & \left[\begin{array}{cc} \textcircled{1} p(a) \\ l(X, s), & \textcircled{3} \neg p(X) \\ \neg c(Y) & \textcircled{2} \neg g(Y) \end{array} \right]
\end{array} \tag{10}$$

The crucial data structure of the pool calculus that maintains the information which paths remain to be examined is a *pool* of so called *hooks*.

Definition 4.1 (Pools, hooks)

A *hook* for a matrix M is a pair (p, Γ) where p is a partial path in a set of instances M' of M and Γ is a subclause of a clause $\Gamma' \in M'$ such that $p \cap \Gamma = \emptyset$. The hook (p, Γ) will be denoted by $(p \perp \Gamma)$. We use the sign $\perp$ in order to indicate that the partial path p and the clause Γ are in a sense orthogonal. The partial path p is called the *actual path*. The elements of Γ are called *goals*. The set of paths

$$Paths_{(p \perp \Gamma)}^{M'} = \{p' \mid \exists L(p \cup \{L\} \subseteq p', p' \cap \Gamma = \{L\}), p' \text{ is a path through } M'\}$$

will be called the *set of paths through M' represented by the hook $(p \perp \Gamma)$* .

Remark: We assume that the representation of paths and clauses as multisets of literals allows to determine the set of clauses M' which serve as upper index in $Paths_{(p \perp \Gamma)}^{M'}$. Thus, this index will be omitted in the following.

Any hook of the form $(p \perp \emptyset)$ will be called a *solved hook*. Any hook of the form $(\emptyset \perp \Gamma)$ for some clause copy Γ is called an *initial hook*.

A *pool* for a matrix M is a set of hooks for the matrix M . The set *set of paths represented by a pool P* will be the set

$$Paths_P = \bigcup_{h \in P} Paths_h.$$

□

Example 4.2 Below we display the same derivation as before as a sequence of pools. Each pool represents the same sets of paths like the corresponding matrix in the derivation (10). In every inference step a hook (called the chosen hook) is removed from the pool and some new hooks are produced. The chosen hook is underlined.

$$\begin{array}{c}
 \left\{ \underline{(\perp \ p(a))} \right\} \quad \vdash_{extension} \quad \left\{ \begin{array}{l} (\perp), \\ (p(a) \perp \neg p(a)), \\ \underline{(p(a), l(a, s) \perp \neg g(s))} \end{array} \right\} \\
 \vdash_{reduction} \quad \left\{ \begin{array}{l} (\perp), \\ \underline{(p(a) \perp \neg p(a))}, \\ (p(a), l(a, s) \perp) \end{array} \right\} \quad \vdash_{reduction} \quad \left\{ \begin{array}{l} (\perp), \\ (p(a) \perp), \\ (p(a), l(a, s) \perp) \end{array} \right\}
 \end{array} \tag{11}$$

Definition 4.2 ($\mathcal{T}$ -connection inference)

Let $\mathcal{U}$ be a complete set of $\mathcal{T}$ -connections and M a matrix. A $\mathcal{T}$ -connection inference is an inference rule of the form

$$\frac{(p \perp \Gamma_0, L_0) \quad \Gamma_1 \cup \{L_1\}, \dots, \Gamma_n \cup \{L_n\}}{(p \perp \Gamma_0), (p, L_0 \perp \Gamma_1), \dots, (p, L_0, \dots, L_{n-1} \perp \Gamma_n)} \sigma$$

where

- (1) $(p \perp \Gamma_0, L_0)$ is a hook, called the *chosen hook*,
- (2) if $0 < n$ then $\Gamma_1 \cup \{L_1\}, \dots, \Gamma_n \cup \{L_n\}$ are copies of clauses from M , called the *extension clauses*,
- (3) σ is a idempotent substitution.
- (4) $(p \perp \Gamma_0), (p, L_0 \perp \Gamma_1), \dots, (p, L_0, \dots, L_{n-1} \perp \Gamma_n)$ are hooks, called *new hooks*, and
- (5) there exists a subpath q of p such that $u \in \mathcal{U}$ and $\sigma(u)$ is $\mathcal{T}$ -complementary for the partial path $u = q \cup \{L_0, \dots, L_n\}$.

A $\mathcal{T}$ -connection inference is called an *extension step* if $n \neq 0$ and a *reduction step* else. $\square$

The calculus is described by rules which determine how from a chosen hook may be constructed new hooks. In the inference steps in derivation (11) also solved hooks have been adjoined to the pool resulting pool in order to illustrate the derivation. In the following definition we will prefer to avoid that kind of redundancy and adjoin only unsolved hooks to the new pool which has been obtained by an inference.

Definition 4.3 (Rule application)

A rule of the form $\frac{h \quad \Gamma}{H} \sigma$ may be applied to a pool P if $h \in P$. The new pool P' is obtained from P by first removing h , then adjoining those hooks from H which are not solved and finally applying the substitution σ to the resulting pool. We will write $P \Rightarrow P'$. $\square$

Definition 4.4 (Derivation)

An *initial pool* for a matrix M consists of a single initial hook for that matrix. A *derivation* for a matrix M is a sequence of pools $\{P_i\}_{i=0}^n$ such that

- (1) P_0 is an initial pool for that matrix and
- (2) for every $i < n$ $P_i \Rightarrow P_{i+1}$ is a $\mathcal{T}$ -connection inference for M and the variables occurring in clause copies which occur in the antecedent of that $\mathcal{T}$ -connection inference must not occur in a clause copy which has been used in an earlier inference of that derivation.

A derivation is called *ground* if the unifier in every $\mathcal{T}$ -connection step is empty. A derivation is *successful* if its last element is the empty pool. $\square$

Proposition 4.1 (Soundness)

Let M be a matrix and $\{P_i\}_{i < \lambda}$ be a successful $\mathcal{T}$ -connection derivation for M . Then M is $\mathcal{T}$ -complementary.

Proof. Let $\{P_i\}_{i < \lambda}$ be a derivation. We define sequences $\{M_i\}_{i < \lambda}$ of matrices and $\{U_i\}_{i < \lambda}$ of sets of connections as follows.

- (1) $U_0 = \emptyset$ and $M_0 = \{\Gamma\}$ where $P_0 = \{(\perp \Gamma)\}$.
- (2) For all $i < \lambda$
 - (a) U_{i+1} is obtained from U_i by adjoining the connection and applying the unifier which have been found in the $(i+1)$ -th inference.
 - (b) M_{i+1} is obtained from M_i by adjoining the clause copies which occur in the antecedent of the and $(i+1)$ -th inference and applying the unifier which has been found in that inference.

It suffices to prove that following invariants are satisfied.

- $U_0 = \text{Paths}_{M_0} \setminus \text{Paths}_{P_0} = \emptyset$
- U_{i+1} is spanning $\text{Paths}_{M_{i+1}} \setminus \text{Paths}_{P_{i+1}}$ for all $i < \lambda$.

The invariants are satisfied because in every $\mathcal{T}$ -connection inference (cf. definition 4.2) with chosen hook h_0 and new hooks $h_1, \dots, h_n$ all paths in the difference

$$\sigma(\text{Paths}_{h_0}) \setminus \sigma\left(\bigcup_{i=1}^n \text{Paths}_{h_i}\right)$$

contain the $\mathcal{T}$ -connection established in that inference.

Therefore, if $\lambda = k + 1$ for some natural number k and $P_k = \emptyset$ then U_k is spanning M_k . Moreover, every element of U_k is $\mathcal{T}$ -complementary. Thus M_k is a $\mathcal{T}$ -unsatisfiable instantiation of M which is therefore $\mathcal{T}$ -unsatisfiable too. **q.e.d.**

5 Completeness of the theory connection pool calculus

The completeness proof consists traditionally of the following steps: Herbrand theorem, proof of the ground completeness and lifting lemma. The Herbrand theorem (3.1) relies the completeness of a given set of $\mathcal{T}$ -connections $\mathcal{U}$, on the solvability of the $\mathcal{T}$ -unification problem in $\mathcal{U}$. The proof of the ground completeness relies on a number of interesting properties of minimal spanning matings which will be proved in the lemmas 5.1, 5.2 and 5.3.

Lemma 5.1 *Let U be a spanning mating in a matrix M . Let $U' \subseteq U$ be a such that no proper subset of U' is spanning in M . Then there exists U'' with the following properties:*

- (1) $U' \subseteq U'' \subseteq U$ and
- (2) U'' is a minimal spanning mating in matrix M .

Proof. Let us consider all descending sequences of sets $\{U_i\}_{i < k}$ such that

- (1) $U_0 = U$,
- (2) for each $i < k$: $U' \subseteq U_i$,
- (3) for each $i < k$: U_i is spanning in M and
- (4) for each $i < k$: $\text{card}(U_i) = \text{card}(U_{i+1}) + 1$.

Let $\{U_j\}_{j < l}$ be that sequence of this kind which has maximal length. Then U_{l-1} has no proper subset U''' which is spanning. Otherwise $U_{l-1} \neq U'$ and therefore the considered sequence could be extended by some set U^{iv} with $U''' \subseteq U^{iv} \subseteq U_{l-1}$. This contradicts the assumption about the chosen sequence. Thus, U_{l-1} has the required properties. q.e.d.

Definition 5.1 Let U be a mating in a matrix M . A path p through M is said to support an element u of U if p contains no other element of u as subpath. □

Lemma 5.2 *Let U be a minimal spanning mating in a matrix M . Let Γ be a clause in M such that there exist $L \in \Gamma$, $u \in U$ which satisfy $L \in u$.*

Then for every $L' \in \Gamma$ exists $u' \in U$ such that $L' \in u'$.

Proof. Let U, M, Γ, L and u satisfy the assumptions of the lemma. Suppose that there exists $L' \in \Gamma$ such that there is no $u' \in U$ which satisfies $L' \in u'$. Because U is minimal spanning in M there exists a path p which supports u . Then for the path $p' = (p \setminus \{L\}) \cup \{L'\}$ must exist some $u' \in U$ such that $u' \subseteq p'$. According to our assumption about L' holds $L' \notin u'$. Therefore $u' \subseteq p$. From $L \in u$ and $u' \subseteq (p \setminus \{L\}) \cup \{L'\}$ follows $u \neq u'$. This contradicts the assumption that p supports u . q.e.d.

Definition 5.2 Let $h = (p \perp \Delta)$ be an unsolved hook for a ground matrix M . By $M(h)$ let us denote the matrix which consists of the following clauses:

- (1) for each $L \in p$ the unit clause L ,
- (2) the clause Δ and
- (3) every clause $\Gamma \in M$ such that $p \cap \Gamma = \emptyset$ and $\Delta \cap \Gamma = \emptyset$.

□

Lemma 5.3 *Let $h = (p \perp \Gamma_0, L_0)$ be a hook in a ground matrix M , U a minimal spanning mating in $M(h)$ such that there is an application of a total $\mathcal{T}$ -connection inference according to definition 4.2*

$$\frac{(p \perp \Gamma_0, L_0) \quad \Gamma_1 \cup \{L_1\}, \dots, \Gamma_n \cup \{L_n\}}{(p \perp \Gamma_0), (p, L_0 \perp \Gamma_1), \dots, (p, L_0, \dots, L_{n-1} \perp \Gamma_n)} \quad \{ \} \quad (12)$$

where there exists a subpath q of p such that $u = q \cup \{L_0, \dots, L_n\} \in U$. Then for every $l \leq n$ such that $\Gamma_l \neq \emptyset$ there exists a subset U_l of U such that

- (1) U_l is minimal spanning in $M(h_l)$ for $h_l = (p, L_0, \dots, L_{l-1} \perp \Gamma_l)$ and
- (2) for each $L \in \Gamma_l$ exists $u' \in U_l$ such that $L \in u'$.

Proof.

If the assumptions of the lemma are satisfied then for every $l \leq n$ such that $\Gamma_l \neq \emptyset$ the following construction is possible. Let

- $U' = U \setminus \{u\}$ and
- for each l with $0 \leq l \leq n$: $U_l'' = \{u' \in U \mid L_k \in u' \text{ for } 0 \leq k \leq l, L' \in u' \text{ for some } L' \in \Gamma_l\}$.

Let us observe the following three facts:

- U' is spanning $M(h_l)$.
This is true because u is not a path in $M(h_l)$, $U' = U \setminus \{u\}$ and $M(h_l)$ is submatrix of $M(h)$ and U is spanning $M(h)$.
- $U_l'' \subseteq U'$.
It suffices to observe $u \notin U_l''$. This holds because $L \in u$ and Γ_l, L_l is a clause.
and
- No proper subset U''' of U_l'' is spanning in $M(h_l)$.
In order to prove this claim let U''' be a proper subset of U_l'' . We consider an element $u' \in U_l'' \setminus U'''$. The connection u' has a supporting path p' through $M(h)$ because $u' \in U_l'' \subseteq U' \subseteq U$ and U is minimal spanning in $M(h)$. Since $U''' \subseteq U_l'' \subseteq U$ there is no other $\mathcal{U}$ -connection in U''' which is subpath of p' . But p' is also a path through $M(h_l)$. Therefore U''' cannot be spanning in $M(h_l)$ because p' does not contain any connection which belongs to U''' .

Thus, $M(h_l)$, U'' and U' satisfy the assumptions of lemma 5.1. Therefore there exists a minimal spanning mating U^{iv} in $M(h_l)$ such that $U'' \subseteq U^{iv} \subseteq U'$. U^{iv} satisfies also condition 2 because $U_l'' \subseteq U^{iv}$.

q.e.d.

Proposition 5.1 (*Ground completeness*)

Let U be a minimal spanning mating in a matrix M . Then there exists a ground derivation which detects U .

Proof. (Ground completeness)

Let U and M satisfy the assumptions of the lemma. For every hook h in M let $\mu(h)$ be the number of paths through $M(h)$. We define the noetherian preorder $\succ$ in the set of hooks by the condition $h \succ h'$ iff $\mu(h) > \mu(h')$. The relation $\succ$ induces a noetherian multiset ordering on the set of pools. We construct a derivation $\{P_i\}_{i < \lambda}$, a sequence $\{U_i\}_{i < \lambda}$ of matings in M and sequence $\{\psi_i\}_{i < \lambda}$ of mappings from P_i into the powerset of U such that for every $i < \lambda$ the following conditions hold.

- (1) $U_i \subseteq U$ is spanning $Paths_M \setminus Paths_{P_i}$.
- (2) For every $h \in P_i$
 - (a) $\psi_i(h)$ is a minimal spanning mating in $M(h)$ and
 - (b) if $h = (p \perp \Delta)$ then for every $L \in \Delta$ exists $u \in \psi_i(h)$ such that $L \in u$.
- (3) $P_i \succ P_{i+1}$ and
- (4) if $P_i \neq \emptyset$ then there exists a successor P_{i+1} .

The derivation $\{P_i\}_{i < \lambda}$ cannot be infinite because $\succ$ is noetherian. Therefore condition (4) implies that it is a successful derivation and $\lambda = k + 1$ for some natural number k . Therefore U_k is spanning M and $U = U_k$ because $U_k \subseteq U$ and U is a minimal spanning mating in M . Thus, the existence of $\{P_i\}_{i < \lambda}$, $\{U_i\}_{i < \lambda}$ and $\{\psi_i\}_{i < \lambda}$ completes the proof.

Construction of P_0 , U_0 and ψ_0 : The spanning mating U is not empty. We choose an arbitrary element $u \in U$ and an arbitrary literal $L \in u$. Let Γ be the clause in M such that $L \in \Gamma$. Let $P_0 = \{(\emptyset \perp \Gamma)\}$, $U_0 = \emptyset$ and $\psi_0(h) = U$. Conditions (1) and (2) are satisfied.

Construction of P_{i+1} , U_{i+1} and ψ_{i+1} : Suppose a derivation $\{P_j\}_{j=0}^i$ and sequences $\{U_j\}_{j=0}^i$ and $\{\psi_j\}_{j=0}^i$ have been constructed such that

- U_j satisfies the condition (1) for every $j \leq i$,
- P_j and ψ_j satisfy the conditions (2), (3) and (4) for every $j < i$,
- P_i and ψ_i satisfy the condition (2) and
- $P_i \neq \emptyset$.

Let $h = (p \perp \Gamma_0, L_0)$, $h \in P_i$ be an arbitrary hook and let $U' = \psi_i(h)$. According to condition (2b) there exists a $\mathcal{U}$ -connection $u \in U'$ such that $L_0 \in u$. We define $U_{i+1} = U_i \cup \{u\}$. Let us consider a total $\mathcal{T}$ -connection inference

$$\frac{(p \perp \Gamma_0, L_0) \quad \Gamma_1 \cup \{L_1\}, \dots, \Gamma_n \cup \{L_n\}}{(p \perp \Gamma_0), (p, L_0 \perp \Gamma_1), \dots, (p, L_0, \dots, L_{n-1} \perp \Gamma_n)} \quad \{ \} \quad (13)$$

where there exists a subpath q of p such that $u = (q \cup \{L_0, \dots, L_n\})$. Let P_{i+1} be such that $P_i \Rightarrow P_{i+1}$ holds according to definition 4.3. Then P_{i+1} and U_{i+1} satisfy condition (1).

It remains to define ψ_{i+1} . For $l = 0, 1, \dots, n$ let $h_l = (p, L_0, \dots, L_{l-1} \perp \Gamma_l)$.

- For each $h' \in P_i \setminus \{h\}$ we define $\psi_{i+1}(h') = \psi_i(h')$. The conditions (2a) and (2b) hold also for $\psi_{i+1}(h')$.
- For each $l \leq n$ such that $\Gamma_l \neq \emptyset$ the matrix $M(h_l)$ and the mating U' satisfy the assumptions of lemma 5.3. Let $\psi_{i+1}(h_l)$ be the a subset U_l of U which exists according to that lemma.

Then the mapping ψ_{i+1} defined this way satisfies condition (2).

In order to prove condition (3) we may observe that $P_{i+1} \subseteq (P_i \setminus \{h\}) \cup \bigcup_{l=0}^n \{h_l\}$ and any path through one of the new hooks is a path through $M(h)$. Thus, $\bigcup_{l=0}^n \text{Paths}_{h_l} \subseteq \text{Paths}_h$. On the other hand, no path through $M(h)$ which consists L_0 is a path through h_l and hence $\mu(h_l) < \mu(h_0)$.

This finishes the construction of the sequences $\{P_i\}_{i < \lambda}$, $\{U_i\}_{i < \lambda}$ and $\{\psi_i\}_{i < \lambda}$ and the proof. **q.e.d.**

Lemma 5.4 (Lifting lemma)

Let $\mathcal{U}$ be a set of $\mathcal{T}$ -connections with a solvable unification problem. Suppose that for some matrix M there exist a set M' of copies of clauses from M , a $\mathcal{U}$ -mating U which is spanning in M' and a simultaneous $\mathcal{T}$ -unifier θ of U . Then any ground derivation in $\theta(M')$ may be lifted to a derivation for M .

Proof.

Let a matrix M , a set M' of copies of clauses from M , a $\mathcal{U}$ -mating U and a substitution θ satisfy the assumptions of the proposition. Let $\{P_i^0\}_{i=1}^n$ be any ground derivation in $\sigma(M')$ which detects the mating $U^0 = \theta(U)$. For each $i < n$ let h_i^0 be the chosen hook in the inference $P_i^0 \Rightarrow P_{i+1}^0$ and u_{i+1}^0 the connection which has been found in that inference.

The $\mathcal{T}$ -unification problem is solvable for $\mathcal{U}$. Let S_u denote the complete set of $\mathcal{T}$ -unifiers for each $u \in \mathcal{U}$. Let $\{u_1, \dots, u_n\}$ be a enumeration of U such that $\theta(u_i) = u_i^0$ for each $i < n$. θ is the simultaneous $\mathcal{T}$ -unifier of the mating $U \subseteq \mathcal{U}$. According to proposition 3.1 there exist sequences $\{\sigma_i\}_{i=1}^n$, $\{\eta_i\}_{i=0}^n$, $\{\varrho_i\}_{i=0}^n$ such that

(1) $\eta_0 = \theta$ and $\varrho_0 = \{ \}$ and

(2) for every $i, 1 \leq i \leq n$

(a) $\sigma_i \in S_{\varrho_{i-1}(u_i)}$,

- (b) $\sigma_i \eta_i = \eta_{i-1}$ and
- (c) $\varrho_i = \varrho_{i-1} \sigma_i$.
- (d) $\theta = \varrho_i \eta_i$.

By induction on n may be proved that a successful derivation $\{P_i\}_{i=0}^n$ may be constructed such that for each $i \leq n$ the chosen hook h_i satisfies $h_i^0 = \eta_i(h_i)$ and the found connection satisfies $u_i^0 = \eta_i(u_i)$. Moreover for each $i \leq n$ P_i may be mapped 1-1 into P_i^0 .

q.e.d.

Theorem 5.1 (*Completeness theorem*)

Suppose that for a theory T and a query language Q there is given a decidable set $\mathcal{U}$ of T -connections which is T -complete w.r.t. Q and the T -unification problem in $\mathcal{U}$ is solvable. Then for every T -unsatisfiable query from Q exists a clause Γ and a successful derivation starting from the initial pool $\{(\perp \vdash \Gamma)\}$ such that in each inference according to definition 4.2 for the chosen connection u holds $u \in \mathcal{U}$ and the chosen T -unifier σ is an element of the complete set of T -unifiers S_u which exists for u .

Proof. From the Herbrand theorem 3.1 follows that for a T -unsatisfiable matrix $M \in Q$ there exists a set M' of copies of clauses from M , a $\mathcal{U}$ -mating U which is spanning in M' and a substitution σ such that $\sigma(u)$ is T -complementary for each $u \in U$. From the ground completeness 5.1 follows that for $\sigma(M')$ exists a ground derivation which will find the spanning mating $\sigma(U)$. Lemma 5.4 allows to lift this derivation to a successful derivation for M . **q.e.d.**

Remark: This completeness result is stronger than that in [Bib82] in the following sense. In our formulation of the inference the actual goal is required to be an element of the chosen theory connection. In the calculus in [Bib82] is less restricted. A similar restriction of the inference rule appears in the theory model elimination tableau in [Bau92].

6 Towards an implementation

The work reported in this paper aims in the construction of an experimental tool for theory reasoning. The pool calculus with the restricted form of equational reasoning described in subsection 3.3 has been implemented by the author. A unification algorithm due to F. Debart [DEL90] has been used. The mentioned implementation is in fact an interpreter of the pool calculus written in Quintus-Prolog. Tests showed that this interpreter suffers from a rather high overhead. Actually we are going to adapt a Prolog technology prover (cf. [Neu92]) written in Quintus-Prolog.

7 Conclusion

We introduced a framework which allows to build-in a large class of theories into provers which are based on the connection method. The aim of this work is to obtain a generic completeness theorem for a total theory connection calculus. For this purpose a formal description of the interface between foreground and background reasoner is necessary. The interface is defined by a complete set $\mathcal{U}$ of theory connections and a solution for the T -unification problem for that set of theory connections. Given both ingredients for a theory the completeness of the total theory connection calculus follows from the mentioned generic completeness theorem. It is well known [Ede89] that other proof calculi like resolution and tableaux may simulate the connection method. These result may be carried over to theory reasoning [BFP92]. Therefore our approach may be used for other proof calculi too.

Acknowledgements

The author wishes to thank to W. Bibel and his Intellectics Group, to P. Enjalbert and the L.A.I.A.C. and to U. Furbach and P. Baumgartner for fruitful discussions concerning the topic of this paper are for their hospitality during several stays.

References

- [And81] P. Andrews. Theorem Proving via General Matings. *J.ACM*, 28(2):193–214, 1981.
- [Bau92] P. Baumgartner. Theory Model Elimination. In H. J. Ohlbach, editor, *Proc. GWAI 92*, 1992. MP-I-Inf.
- [BB87] K.H. Bläsius and H.-J. Bürkert. *Deduktionssysteme*. Oldenburg, 1987.
- [Bec93] Bernhardt Beckert. Ein vervollständigungs-basiertes Verfahren zur Behandlung von Gleichheit im Tableauealkül mit freien Variablen. Master's thesis, Universität Karlsruhe (TH), 1993.
- [BFL83] R. Brachmann, R. Fikes, and H. Levesque. KRYPTON: a functional approach to knowledge representation. *IEEE Computer*, 16(10):67–73, October 1983.
- [BFP92] P. Baumgartner, U. Furbach, and U. Petermann. A Unified Approach to Theory Reasoning. Forschungsbericht 15/92, University of Koblenz, 1992.

- [Bib82] W. Bibel. *Automated Theorem Proving*. Vieweg, 1st edition, 1982.
- [Bür90] H.J. Bürckert. A Resolution Principle for Clauses with Constraints. In M. E. Stickel, editor, *Proc CADE 10*, pages 178–192. Springer, 1990. LNCS/LNAI 449.
- [DEL90] F. Debart, P. Enjalbert, and M. Lescot. Multi modal logic programming using equational and order-sorted logic. In M. Okada and S. Kaplan, editors, *Proc. 2nd Conf. on Conditional and Typed Rewriting Systems*. Springer, 1990. LNCS.
- [Ede89] Elmar Eder. Relative complexities of first-order calculi. Habilitationsschrift, Uni. Dortmund, 1989.
- [Fri91] A. M. Frisch. The Substitutional Framework for Sorted Deduction: Fundamental Results on Hybrid Reasoning. *Artificial Intelligence*, 1991.
- [GNPS92] J. Gallier, P. Narendran, D. Plaisted, and W. Snyder. Theorem Proving with Equational Matings and Rigid E-unification. *J. of ACM*, 1992.
- [Gou93] Jean Goubault. A Rule-Based Algorithm for Rigid E-Unification. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *3rd Kurt Gödel Colloquium '93*, volume 713 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.
- [Lov78] D. Loveland. *Automated Theorem Proving - A Logical Basis*. North Holland, 1978.
- [MR87] N. Murray and E. Rosenthal. Theory Links: Applications to Automated Theorem Proving. *J. of Symbolic Computation*, (4):173–190, 1987.
- [Neu92] Gerd Neugebauer. Compiling first order logic: A prolog implementation. Technical report, FG Intellektik, Technisch Hochschule Darmstadt., 1992.
- [NS91] Gerd Neugebauer and Torsten Schaub. A pool-based connection calculus. Technical Report AIDA-91-02, FG Intellektik, Technisch Hochschule Darmstadt., 1991.
- [OS91] H.J. Ohlbach and J. Siekmann. The Markgraf Karl Refutation Procedure. In J.L. Lassez and G. Plotkin, editors, *Computational Logic — Essays in Honor of Alan Robinson*, pages 41–112. MIT Press, 1991.
- [Pet90] U. Petermann. Towards a connection procedure with built in theories. In *JELIA 90*. European Workshop on Logic in AI, Springer, LNCS 478, 1990.
- [Pet92] U. Petermann. How to build in an open theory into connection calculi. *J. on Computer and Artificial Intelligence*, 11(2):105–142, 1992.
- [Pet93] U. Petermann. Completeness of the pool calculus with an open built in theory. In Georg Gottlob, Alexander Leitsch, and Daniele Mundici, editors, *3rd Kurt Gödel Colloquium '93*, number 713 in *Lecture Notes in Computer Science*, pages 264–277. Springer-Verlag, 1993.
- [RS70] H. Rasiowa and R. Sikorski. *The Mathematics of Metamathematics*, volume 41 of *Monografie Matematyczne*. Polish Scientific Publishers, Warsaw, 1970.
- [Sti85] M.E. Stickel. Automated deduction by theory resolution. *J. of Automated Reasoning*, 4(1):333–356, 1985.

Contents

1	Introduction	3
2	Preliminaries	6
2.1	Clause logic	6
2.2	Interpretations	6
3	Complete sets of theory connections	8
3.1	Taxonomic reasoning	11
3.2	Reasoning in constraint theories	11
3.3	Reasoning with Equality: at first a restricted case	12
3.4	Reasoning with Equality: the general case	14
4	A total theory connection calculus	15
5	Completeness of the theory connection pool calculus	19
6	Towards an implementation	24
7	Conclusion	24

Uwe Petermann
FB IMN, HTWK Leipzig
Postfach 66, D-04251 Leipzig (F.R.G.)
Net: uwe@imn.th-leipzig.de