A Non-clausal Connection Calculus

Jens Otten

Institut für Informatik, University of Potsdam
August-Bebel-Str. 89, 14482 Potsdam-Babelsberg, Germany
jeotten@cs.uni-potsdam.de

Abstract. A non-clausal connection calculus for classical first-order logic is presented that does not require the translation of input formulae into any clausal form. The definition of clauses is generalized, which may now also contain (sub-) matrices. Copying of appropriate (sub-)clauses in a dynamic way, i.e. during the actual proof search, is realized by a generalized extension rule. Thus, the calculus combines the advantage of a non-clausal proof search in tableau calculi with the more efficient goal-oriented proof search of clausal connection calculi. Soundness, completeness, and (relative) complexity results are presented as well as some optimization techniques.

1 Introduction

Connection calculi are a well-known basis to automate formal reasoning in classical first-order logic. Among these calculi are the connection method [3, 4], the connection tableau calculus [9] and the model elimination calculus [10]. The main idea of connection calculi is to connect two atomic formulae P and $\neg P$ with the same predicate symbol but different polarity. The set $\{P, \neg P\}$ is called a *connection* and corresponds to a closed branch in the tableau framework [6] or an axiom in the sequent calculus [5]. As the proof search is guided by connections it is more goal-oriented compared to the proof search in sequent calculi or (standard) analytic tableau calculi.

The *clausal* connection calculus works for first-order formulae in disjunctive normal form or *clausal form*. Formulae that are not in this form have to be translated into clausal form. The standard transformation translates a first-order formula F into clausal form by applying the distributivity laws. In the worst case the size of the resulting formula grows exponentially with respect to the size of the original formula F. This increases the search space significantly when searching for a proof of F in the connection calculus.

A *structure-preserving* translation into clausal form, e.g. [14], introduces definitions for subformulae. Tests show that even such an optimized translation introduces a significant overhead for the proof search [12] as additional formulae are introduced. Both clausal form translations modify the structure of the original formula, making it more difficult to translate a found proof back into a more human-oriented form, e.g. [5]. For some logics, e.g. intuitionistic logic, these translations do *not* preserve logical validity.

A *non-clausal* connection calculus that works directly on the structure of the original formula does not have these disadvantages. There already exist a few descriptions of non-clausal connection calculi [1, 4, 7, 8]. But the cores of these calculi do not add any copies of quantified subformulae to the original formulae, i.e. they are only complete for ground formulae. To deal with first-order logic, e.g., copies of subformulae need to be added iteratively [8]. But this introduces a large amount of redundancy as copies of subformulae that are not required for a proof are still used during the proof search. Implementations of this approach, e.g. [16], show a rather modest performance. For a more effective non-clausal proof search, clauses have to be added carefully and dynamically during the proof search, in a way similar to the approach used for copying clauses in clausal connection calculi. To this end, the existing clausal connection calculus has to be generalized and its rules have to be carefully extended.

The rest of the paper is structured as follows. In Section 2 the standard clausal connection calculus is presented. Section 3 introduces the main ideas of a non-clausal proof search before the actual non-clausal calculus is described in Section 4. Section 5 contains correctness, completeness and complexity results. Some optimizations and extensions are presented in Section 6, before Section 7 concludes with a short summary.

2 The Clausal Connection Calculus

The reader is assumed to be familiar with the language of classical first-order logic, see, e.g., [4]. In this paper the letters P, Q, R are used to denote predicate symbols, f to denote function symbols, a, b, c to denote constants and x to denote variables. Terms are denoted by t and are built from functions, constants and variables. *Atomic formulae*, denoted by A, are built from predicate symbols and terms. The connectives $\neg$, $\wedge$, $\vee$, $\Rightarrow$ denote negation, conjunction, disjunction and implication, respectively. A *(first-order) formula*, denoted by F, G, H, consists of atomic formulae, the connectives and the existential and universal quantifiers, denoted by $\forall$ and $\exists$, respectively. A *literal*, denoted by L, has the form A or $\neg A$. The complement $\overline{L}$ of a literal L is A if L is of the form $\neg A$, and $\neg A$ otherwise. A *clause*, denoted by C, is of the form $L_1 \wedge \ldots \wedge L_n$ where L_i is a literal. A formula in *disjunctive normal form* or *clausal form* has the form $\exists x_1 \ldots \exists x_n (C_1 \vee \ldots \vee C_n)$ where each C_i is a clause. For classical logic every formula F can be translated into a validity-preserving formula F' in clausal form. A clause can be written as a set of literals $\{L_1, \ldots, L_n\}$. A formula in clausal form can be written as a set of clauses $\{C_1, \ldots, C_n\}$ and is called a *matrix*, denoted by M. In the graphical representation of a matrix, its clauses are arranged horizontally, while the literals of each clause are arranged vertically. A *polarity* is used to represent negation in a matrix, i.e. literals of the form A and $\neg A$ are represented by A^0 and A^1, respectively.

Example 1 (Matrix in Clausal Form). Let F_1 be the formula
$(\forall x((\neg P(x) \vee Q(f(x))) \Rightarrow (Q(x) \wedge (Q(a) \Rightarrow R(b)) \wedge \neg R(x))) \wedge Q(f(b))) \Rightarrow P(a)$.
The matrix M_1 of the formula F_1 is
$$\{ \{P^0(a)\}, \{P^1(x), Q^1(x)\}, \{P^1(x), Q^0(a), R^1(b)\}, \{P^1(x), R^0(x)\},$$
$$\{Q^0(f(x)), Q^1(x)\}, \{Q^0(f(x)), Q^0(a), R^1(b)\}, \{Q^0(f(x)), R^0(x)\}, \{Q^1(f(b))\} \} .$$
The graphical representation of M_1 (with some variables renamed) is shown in Figure 1.

A *connection* is a set of the form $\{A^0, A^1\}$. A *path* through $M = \{C_1, \ldots, C_n\}$ is a set of literals that contains one literal from each clause $C_i \in M$, i.e. $\cup_{i=1}^{n} \{L_i'\}$ with $L_i' \in C_i$. A *term substitution* σ is a mapping from the set of variables to the set of terms. In $\sigma(L)$ all variables of the literal L are substituted according to their mapping in σ. $A[x \backslash t]$ denotes the formula in which all free occurrences of x in A are replaced by t.

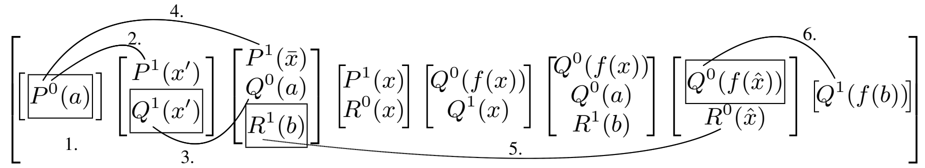


Fig. 1. Proof search in the connection calculus using the graphical matrix representation

Example 2 (Connection, Path, Term Substitution). Consider the matrix M_1 of Example 1. $\{P^0(a), P^1(x)\}$ and $\{R^0(x), R^1(b)\}$ are connections, $\{P^0(a), P^1(x), Q^0(f(x)), R^0(x), Q^1(f(b))\}$ is a path through M_1, and $\sigma(x) = a$ is a term substitution.

The matrix characterization [4] of validity is the underlying basis of the connection calculus and used for the proof in Section 5.1. The notion of *multiplicity* is used to encode the number of clause copies used in a connection proof. It is a function $\mu : \mathcal{C} \rightarrow I\!N$, where $\mathcal{C}$ is the set of clauses in M, that assigns each clause in M a natural number specifying how many copies of this clause are considered in a proof. In the *copy of a clause* C all variables in C are replaced by new variables. M^μ is the matrix that includes these clause copies. A connection $\{L_1, L_2\}$ with $\sigma(L_1) = \sigma(\overline{L_2})$ is called *σ-complementary*.

Theorem 1 (Matrix Characterization). *A matrix M is classically valid iff there exist a multiplicity μ, a term substitution σ and a set of connections $\mathcal{S}$, such that every path through M^μ contains a σ-complementary connection $\{L_1, L_2\} \in \mathcal{S}$.*

See [4] for a proof of Theorem 1. The connection calculus uses a *connection-driven* search strategy in order to calculate an appropriate set of connections $\mathcal{S}$. Proof search in the connection calculus starts by selecting a start clause. Afterwards connections are successively identified in order to make sure that all paths through the matrix contain a σ-complementary connection for some term substitution σ. This process is guided by an *active path*, a subset of a path through M.

Example 3 (Proof Search in the Clausal Connection Calculus). Consider the matrix M_1 of Example 1. The six steps required for a proof in the connection calculus for M_1, using the graphical matrix representation, are depicted in Figure 1. The literals of each connection are connected with a line. The literals of the active path are boxed. In the *start* step the first clause $\{P^0(a)\}$ is selected as start clause (step 1). While the *extension* step connects to a literal in a copy of a clause (steps 2, 3, 5 and 6), the *reduction* step connects to a literal of the active path (step 4). x', $\bar{x}$ and $\hat{x}$ are fresh variables. With the term substitution $\sigma(x') = a$, $\sigma(\bar{x}) = a$ and $\sigma(\hat{x}) = b$ all paths through the matrix M_1 contain a σ-complementary connection from the set $\{\{P^0(a), P^1(x')\}, \{Q^1(x'), Q^0(a)\}, \{P^1(\bar{x}), P^0(a)\}, \{R^1(b), R^0(\hat{x})\}, \{Q^0(f(\hat{x})), Q^1(f(b))\}\}$. Therefore, M_1 and F_1 are valid.

The proof search is now specified more precisely by a formal calculus [4, 12, 13].

Definition 1 (Clausal Connection Calculus). *The axiom and the rules of the* clausal connection calculus *are given in Figure 2. The words of the calculus are tuples of the form "$C, M, Path$", where M is a matrix, C and $Path$ are sets of literals or ε. C is called the* subgoal clause *and $Path$ is called the* active path. *C_1 and C_2 are clauses, σ is a term substitution, and $\{L_1, L_2\}$ is a σ-complementary connection. The substitution σ is global (or* rigid*), i.e. it is applied to the whole derivation.*

Axiom (A)	$\dfrac{}{\{\},M,Path}$	
Start (S)	$\dfrac{C_2,M,\{\}}{\varepsilon,\ M,\ \varepsilon}$	and C_2 is copy of $C_1{\in}M$
Reduction (R)	$\dfrac{C,M,Path{\cup}\{L_2\}}{C{\cup}\{L_1\},M,Path{\cup}\{L_2\}}$	with $\sigma(L_1){=}\sigma(\overline{L_2})$
Extension (E)	$\dfrac{C_2{\setminus}\{L_2\},M,Path{\cup}\{L_1\} \qquad C,M,Path}{C{\cup}\{L_1\},M,Path}$	and C_2 is a copy of $C_1{\in}M$ and $L_2{\in}C_2$ with $\sigma(L_1){=}\sigma(\overline{L_2})$

Fig. 2. The clausal connection calculus

An application of the start, reduction or extension rule is called a *start*, *reduction*, or *extension step*, respectively. A derivation for $C,M,Path$ with the term substitution σ, in which all leaves are axioms, is called a *clausal connection proof* for $C,M,Path$. A *clausal connection proof* for the matrix M is a clausal connection proof for $\varepsilon,M,\varepsilon$.

Theorem 2 (Correctness and Completeness). *A matrix M is valid in classical logic iff there is a clausal connection proof for M.*

The proof is based on the matrix characterization and can be found in [4]. Proof search in the clausal connection calculus is carried out by applying the rules of the calculus in an analytic way, i.e. from bottom to top. During the proof search backtracking might be required, i.e. alternative rules need to be considered if the chosen rule does not lead to a proof. Alternative applications of rules occur whenever more than one rule or more than one instance of a rule can be applied, e.g. when choosing the clause C_1 in the start and extension rule or the literal L_2 in the reduction and extension rule. No backtracking is required when choosing the literal L_1 in the reduction or extension rule as all literals in C are considered in subsequent proof steps anyway. The term substitution σ is calculated, step by step, by one of the well-known algorithms for term unification, e.g. [15], whenever a reduction or extension rule is applied.

Example 4 (Clausal Connection Calculus). Consider the matrix M_1 of Example 1. A derivation for M_1 in the clausal connection calculus with $\sigma(x')=a$, $\sigma(\bar{x})=a$ and $\sigma(\hat{x})=b$ is given in Figure 3 (some parentheses are omitted). Since all leaves are axioms it represents a clausal connection proof and therefore M_1 and F_1 are valid.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\},M_1,\{P^0a,Q^1x',R^1b,Q^0f\hat{x}\}}\,A \qquad \dfrac{}{\{\},M_1,\{P^0a,Q^1x',R^1b\}}\,A}{\{Q^0f\hat{x}\},M_1,\{P^0a,Q^1x',R^1b\}}\,E \qquad \dfrac{}{\{\},M_1,\{P^0a,Q^1x'\}}\,A}{\{R^1b\},M_1,\{P^0a,Q^1x'\}}\,E}{\{P^1\bar{x},R^1b\},M_1,\{P^0a,Q^1x'\}}\,R \qquad \dfrac{}{\{\},M_1,\{P^0a\}}\,A}{\{Q^1x'\},M_1,\{P^0a\}}\,E \qquad \dfrac{}{\{\},M_1,\{\}}\,A}{\dfrac{\{P^0a\},M_1,\{\}}{\varepsilon,M_1,\varepsilon}\,S}\,E
$$

Fig. 3. A proof in the clausal connection calculus

type	F^p	$M(F^p)$	type	F^p	$M(F^p)$
atomic	A^0	$\{\{A^0\}\}$	β	$(G \wedge H)^0$	$\{\{M(G^0), M(H^0)\}\}$
	A^1	$\{\{A^1\}\}$		$(G \vee H)^1$	$\{\{M(G^1), M(H^1)\}\}$
α	$(\neg G)^0$	$M(G^1)$		$(G \Rightarrow H)^1$	$\{\{M(G^0), M(H^1)\}\}$
	$(\neg G)^1$	$M(G^0)$	γ	$(\forall x G)^1$	$M(G[x\backslash x^*]^1)$
	$(G \wedge H)^1$	$\{\{M(G^1)\}, \{M(H^1)\}\}$		$(\exists x G)^0$	$M(G[x\backslash x^*]^0)$
	$(G \vee H)^0$	$\{\{M(G^0)\}, \{M(H^0)\}\}$	δ	$(\forall x G)^0$	$M(G[x\backslash t^*]^0)$
	$(G \Rightarrow H)^0$	$\{\{M(G^1)\}, \{M(H^0)\}\}$		$(\exists x G)^1$	$M(G[x\backslash t^*]^1)$

Table 1. Matrix of a formula F^p

3 Non-Clausal Proof Search

In this section the definitions of matrices and paths are generalized and the main ideas of the non-clausal connection calculus are illustrated with an introductory example.

3.1 Non-Clausal Matrices

First of all, the definition of matrices is generalized to arbitrary first-order formulae.

Definition 2 (Matrix). *A (non-clausal) matrix is a set of clauses, in which a* clause *is a set of literals and matrices. Let F be a formula and p be a polarity. The* matrix *of F^p, denoted by $M(F^p)$, is defined inductively according to Table 1. The* matrix *of F is the matrix $M(F^0)$. x^* is a new variable, t^* is the Skolem term $f^*(x_1, \ldots, x_n)$ in which f^* is a new function symbol and $x_1, \ldots, x_n$ are the free variables in $\forall x G$ or $\exists x G$.*

In the *graphical representation* of a matrix, its clauses are arranged horizontally, while the literals and (sub-)matrices of each clause are arranged vertically. A matrix M can be simplified by replacing matrices and clauses of the form $\{\{X_1, ..., X_n\}\}$ within M by $X_1, ..., X_n$. Whereas the definition of paths needs to be generalized to non-clausal matrices, all other concepts used for clausal matrices, e.g. the definitions of connections and term substitutions and the matrix characterization, remain unchanged.

Definition 3 (Path). *A* path *through a matrix M (or a clause C) is inductively defined as follows. The (only) path through a literal L is $\{L\}$. If $p_1, \ldots, p_n$ are paths through the clauses $C_1, \ldots, C_n$, respectively, then $p_1 \cup \ldots \cup p_n$ is a path through the matrix $M = \{C_1, \ldots, C_n\}$. If $p_1, \ldots, p_n$ are paths through the matrices/literals $M_1, \ldots, M_n$, respectively, then $p_1, \ldots, p_n$ are also paths through the clause $C = \{M_1, \ldots, M_n\}$.*

Example 5 (Matrix, Path). Consider the formula F_1 of Example 1. The simplified (non-clausal) matrix M_1^* of F_1 is $\{\{P^0(a)\}, \{\{\{P^1(x)\}, \{Q^0(f(x))\}\}, \{\{Q^1(x)\}, \{Q^0(a), R^1(b)\}, \{R^0(x)\}\}\}, \{Q^1(f(b))\}\}$. Its graphical representation is shown in Figure 4. $\{P^0(a), Q^1(x), Q^0(a), R^0(x), Q^1(f(b))\}$ is one of the three paths through M_1^*.

3.2 An Introductory Example

In order to carry out proof search for non-clausal matrices the rules of the clausal connection calculus need to be adapted. The following example illustrates the main ideas.

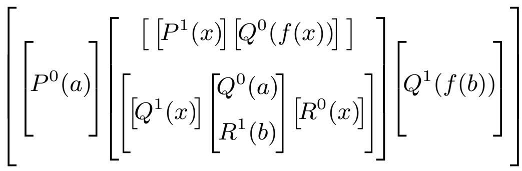


Fig. 4. Graphical representation of M_1^*

Example 6 (Proof Search Using Non-clausal Matrices). Consider the first-order formula F_1 of Example 1 and its graphical matrix representation in Figure 4. The non-clausal connection proof is depicted in Figure 5. Again, the proof search is guided by an active path, whose literals are boxed. Literals of connections are connected with a line. In the first step the start clause $\{P^0(a)\}$ is selected. The second proof step is an extension step and connects $P^0(a)$ with $P^1(x')$ applying the substitution $\sigma(x')=a$. Variables of the clause used for the extension step are always renamed. Next, all remaining paths through the second matrix of the second clause have to be investigated. To this end, the next extension step connects $Q^1(x')$ with $Q^0(a)$ of the clause $\{Q^0(a), R^1(b)\}$. Now, all paths containing the literals $P^0(a)$, $Q^1(x')$, and $R^1(b)$ still need to be investigated. The fourth proof step connects $R^1(b)$ with $R^0(\hat{x})$ contained in a copy of the second clause, as a connection to $R^0(x')$ with $\sigma(x')=a$ is not possible. The clauses in the copied clause occurring next to $R^0(\hat{x})$, i.e. $\{Q^1(\hat{x})\}$ and $\{Q^0(a), R^1(b)\}$, are deleted, as all paths through these two clauses contain the σ-complementary connection $\{R^1(b), R^0(\hat{x})\}$ with $\sigma(\hat{x})=b$ as well. The fifth and last (extension) step connects $Q^0(f(\hat{x}))$ with $Q^1(f(b))$. This concludes the proof and every path through the shown matrix contains a σ-complementary connection. Hence, F_1 is valid. The proof uses only four connections compared to five connections required in the clausal connection proof.

The study of the previous example suggests the following. The axiom as well as the start and reduction rules are the same for the clausal and the non-clausal calculus. The extension rule connects a literal L_1 of the subgoal clause to a literal L_2 occurring in a copy C_2 of an *extension clause* C_1. $\{L_1, L_2\}$ need to be σ-complementary for some σ and L_1 is added to the active path. Either C_1 contains a literal of the active path (fourth step in Example 6); or there is a path that contains the active path and L_2, and if C_1 has a *parent clause*, it contains a literal of the active path (third step). All literals of the new subgoal clause of C_2 that occur besides L_2, i.e. in a common path with L_2, are deleted from the new subgoal clause (fourth step). Finally, if a subgoal clause contains a matrix M, the proof search continues with a clause $C \in M$ (third step). Alternative clauses $C' \in M$ have to be considered on backtracking.

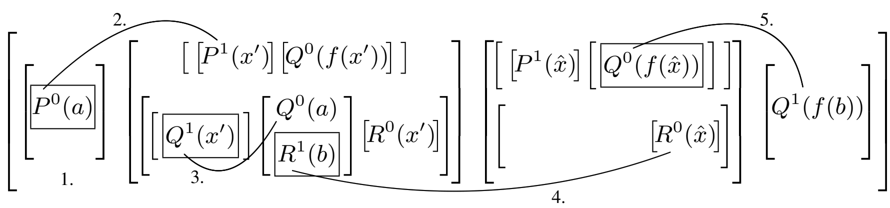


Fig. 5. Proof search in the *non-clausal* connection calculus using the graphical representation

4 The Non-Clausal Connection Calculus

This section introduces basic concepts and the formal non-clausal connection calculus.

4.1 Basic Concepts

The term α-related is used to express the fact that a clause occurs besides a literal in a matrix. Furthermore, the definitions of free variables and clause copies have to be generalized to cover non-clausal matrices.

Definition 4 (α-Related Clause). *Let C be a clause in a matrix M and L be a literal in M. C is* α-related *to L, iff M contains (or is equal to) a matrix $\{C_1, \ldots, C_n\}$ such that $C = C_i$ or C_i contains C, and C_j contains L for some $1 \le i, j \le n$ with $i \ne j$. C is* α-related *to a set of literals $\mathcal{L}$, iff C is α-related to all literals $L \in \mathcal{L}$.*

Definition 5 (Free Variables). *Let M be a matrix and C be a clause in M. The* free variables *of C are all variables that do only occur in C and (possibly) in literals L such that C is α-related to L.*

In the non-clausal calculus copies of clauses could be simply added to the matrix. As this would widen the search space, clauses are replaced by their copies instead.

Definition 6 (Copy of Clause). *Let M be a matrix and C be a clause in M. In the* copy of the clause *C, all free variables in C are replaced by new variables. $M[C_1 \backslash C_2]$ denotes the matrix M, in which the clause C_1 is replaced by the clause C_2.*

As explained in Section 3.2 an appropriate clause C_1 has to be used when the extension rule is applied. Either C_1 has to contain an element of the active path, then it was already used before, or C_1 needs to be α-related to all literals of the active path and it has to contain all subgoals that still need to be investigated. This is the case if C_1 has no parent clause or its parent clause contains a literal of the active path. The parent clause of a clause C is the smallest clause that contains C.

Definition 7 (Parent Clause). *Let M be a matrix and C be a clause in M. The clause $C' = \{M_1, \ldots, M_n\}$ in M is called the* parent clause *of C iff $C \in M_i$ for some $1 \le i \le n$.*

Definition 8 (Extension Clause). *Let M be a matrix and $Path$ be a set of literals. Then the clause C in M is an* extension clause *of M with respect to $Path$, iff either C contains a literal of $Path$, or C is α-related to all literals of $Path$ occurring in M and if C has a parent clause, it contains a literal of $Path$.*

In the extension rule of the clausal connection calculus (see Section 2) the new subgoal clause is $C_2 \setminus \{L_2\}$. In the non-clausal connection calculus the extension clause C_2 might contain clauses that are α-related to L_2 and do not need to be considered for the new subgoal clause. Hence, these clauses can be deleted from the subgoal clause. The resulting clause is called the β-clause of C_2 with respect to L_2.

Definition 9 (β-Clause). *Let $C = \{M_1, \ldots, M_n\}$ be a clause and L be a literal in C. The* β-clause *of C with respect to L, denoted by $\beta\text{-clause}_L(C)$, is inductively defined:*

$$\beta\text{-clause}_L(C) := \begin{cases} C \setminus \{L\} & \textit{if } L \in C, \\ \{M_1, \ldots, M_{i-1}, \{C^\beta\}, M_{i+1}, \ldots, M_n\} & \textit{otherwise,} \end{cases}$$

where $C' \in M_i$ contains L and $C^\beta := \beta\text{-clause}_L(C')$.

4.2 The Calculus

The non-clausal connection calculus has the same axiom, start rule, and reduction rule as the clausal connection calculus. The extension rule is slightly modified and a decomposition rule is added that splits subgoal clauses into their subclauses.

Definition 10 (Non-Clausal Connection Calculus). *The axiom and the rules of the* non-clausal connection calculus *are given in Figure 6. The words of the calculus are tuples "$C, M, Path$", where M is a matrix, C is a clause or ε and $Path$ is a set of literals or ε. C is a called the* subgoal clause. *C_1, C_2, and C_3 are clauses, σ is a term substitution, and $\{L_1, L_2\}$ is a σ-complementary connection. The substitution σ is rigid, i.e. it is applied to the whole derivation.*

Axiom (A) $\dfrac{}{\{\}, M, Path}$

Start (S) $\dfrac{C_2, M, \{\}}{\varepsilon,\ M,\ \varepsilon}$ and C_2 is copy of $C_1 \in M$

Reduction (R) $\dfrac{C, M, Path \cup \{L_2\}}{C \cup \{L_1\}, M, Path \cup \{L_2\}}$ with $\sigma(L_1) = \sigma(\overline{L_2})$

Extension (E) $\dfrac{C_3, M[C_1 \backslash C_2], Path \cup \{L_1\} \qquad C, M, Path}{C \cup \{L_1\}, M, Path}$

and $C_3 := \beta\text{-}clause_{L_2}(C_2)$, C_2 is copy of C_1, C_1 is an extension clause of M wrt. $Path \cup \{L_1\}$, C_2 contains L_2 with $\sigma(L_1) = \sigma(\overline{L_2})$

Decomposition (D) $\dfrac{C \cup C_1, M, Path}{C \cup \{M_1\}, M, Path}$ with $C_1 \in M_1$

Fig. 6. The non-clausal connection calculus

An application of the start, reduction, extension or decomposition rule is called *start*, *reduction*, *extension*, or *decomposition step*, respectively. Observe that the non-clausal calculus reduces to the clausal calculus for matrices that are in clausal form.

Definition 11 (Non-Clausal Connection Proof). *Let M be a matrix, C be a clause, and $Path$ be a set of literals. A derivation for $C, M, Path$ with the term substitution σ in the non-clausal connection calculus, in which all leaves are axioms, is called a* (non-clausal) connection proof *for $C, M, Path$. A* (non-clausal) connection proof *for M is a non-clausal connection proof for $\varepsilon, M, \varepsilon$.*

Proof search in the non-clausal connection calculus is carried out in the same way as in the clausal connection calculus (see Section 2), i.e. the rules of the calculus are applied in an analytic way. Additional backtracking might be required when choosing the clause C_1 in the decomposition rule. No backtracking is required when choosing the matrix M_1 in the decomposition rule as all matrices (and literals) in C are considered in subsequent proof steps anyway. The term substitution σ is calculated by one of the well-known algorithms for term unification, e.g. [15].

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\{\}, \hat{M}, \{P^0a, Q^1x', R^1b, Q^0f\hat{x}\}}\ A \quad \overline{\{\}, \hat{M}, \{P^0a, Q^1x', R^1b\}}\ A}{\{Q^0f\hat{x}\}, \hat{M}, \{P^0a, Q^1x', R^1b\}}\ E}{\{\{\{P^1\hat{x}\}, \{Q^0f\hat{x}\}\}\}, \hat{M}, \{P^0a, Q^1x', R^1b\}}\ D \quad \overline{\{\}, M', \{P^0a, Q^1x'\}}\ A}{\{R^1b\}, M', \{P^0a, Q^1x'\}}\ E \quad \overline{\{\}, M', \{P^0a\}}\ A}{\dfrac{\{Q^1x'\}, M', \{P^0a\}}{\{\{\{Q^1x'\}, \{Q^0a, R^1b\}, \{R^0x'\}\}\}, M', \{P^0a\}}\ D \quad \overline{\{\}, M_1^*, \{\}}\ A}\ E}{\dfrac{\{P^0a\}, M_1^*, \{\}}{\varepsilon, M_1^*, \varepsilon}\ S}$$

Fig. 7. A proof in the non-clausal connection calculus

Example 7 (Non-Clausal Connection Calculus). Consider the matrix M_1^* of Example 5. A derivation for M_1^* in the non-clausal connection calculus with $\sigma(x') = a$, $\sigma(\hat{x}) = b$, $M' = \{\,\{P^0(a)\}, \{\{\{P^1(x')\}, \{Q^0(f(x'))\}\}, \{\{Q^1(x')\}, \{Q^0(a), R^1(b)\}, \{R^0(x')\}\}\}, \{Q^1(f(b))\}\,\}$ and $\hat{M} = \{\,\{P^0(a)\}, \{\{\{P^1(\hat{x})\}, \{Q^0(f(\hat{x}))\}\}, \{\{Q^1(\hat{x})\}, \{Q^0(a), R^1(b)\}, \{R^0(\hat{x})\}\}\}, \{Q^1(f(b))\}\,\}$ is given in Figure 7. Since all leaves are axioms it represents a (non-clausal) connection proof and, therefore, M_1^* and F_1 are valid. This proof corresponds to the graphical proof representation given in Figure 5.

5 Correctness, Completeness and Complexity

In this section it is shown that the non-clausal connection calculus is sound and complete. Furthermore, its complexity is compared to the clausal connection calculus.

5.1 Correctness

Definition 12 (Superset Path through Clause). *Let p be a set of literals. p is a* superset path through C*, denoted by* $C \sqsubseteq p$*, iff there is path* p' *through* $\{C\}$ *with* $p' \subseteq p$.

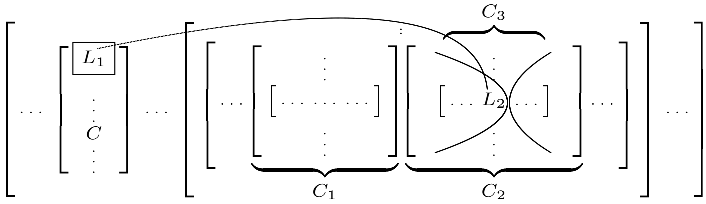
Lemma 1 (Correctness of the Non-Start Rules). *If there is a connection proof for* $C, M, Path$ *with the term substitution* σ*, then there is a multiplicity* μ *such that every path p through* M^μ *with* $Path \subseteq p$ *and* $C \sqsubseteq p$ *contains a* σ*-complementary connection.*

Proof. The proof is by structural induction on the construction of connection proofs. *Induction hypothesis (IH)*: If $Proof$ is a connection proof for $C, M, Path$ with σ, then there is a μ such that every path p through M^μ with $Path \subseteq p$ and $C \sqsubseteq p$ contains a σ-complementary connection.

1. Axiom: Let $\overline{\{\}, M, Path}$ be a connection proof. Let $\mu \equiv 1$ and $\sigma(x) = x$ for all x. Then $\{\} \sqsubseteq p$ holds for no path p through M^μ. Thus, *IH* follows.

2. Reduction: Let $\dfrac{Proof}{C, M, Path \cup \{L_2\}}$ be a connection proof for $C, M, Path \cup \{L_2\}$ for some σ. According to *IH* there is a μ such that every p through M^μ with $Path \cup \{L_2\} \subseteq p$ and $C \sqsubseteq p$ contains a σ-complementary connection. Then the derivation $\dfrac{\dfrac{Proof}{C, M, Path \cup \{L_2\}}}{C \cup \{L_1\}, M, Path \cup \{L_2\}}\ R$ with $\tau(\sigma(L_1)) = \tau(\sigma(\overline{L_2}))$ for some term substitution τ is a connection proof for $C \cup \{L_1\}, M, Path \cup \{L_2\}$. Let $\mu' := \mu$ and $\sigma' := \tau \circ \sigma$. Every path p' through $M^{\mu'}$ with $Path \cup \{L_2\} \subseteq p'$ and $C \cup \{L_1\} \sqsubseteq p'$ contains a σ'-complementary connection as well, since $\sigma'(L_1) = \sigma'(\overline{L_2})$.

3. Extension: Let $\frac{Proof_1}{C_3,\,M[C_1\backslash C_2],\,Path\cup\{L_1\}}$ and $\frac{Proof_2}{C,\,M,\,Path}$ be connection proofs for C_3, $M[C_1\backslash C_2], Path\cup\{L_1\}$ and $C, M, Path$, respectively, for some σ, with $C_3{:=}\beta\text{-}clause_{L_2}(C_2)$, C_2 is a copy of C_1, C_1 is an extension clause of M wrt. $Path \cup \{L_1\}$, and C_2 contains the literal L_2 with $\tau(\sigma(L_1)) = \tau(\sigma(\overline{L_2}))$ for some substitution τ. According to *IH* there is a μ_1 such that every path p through $(M[C_1\backslash C2])^{\mu_1}$ with $Path \cup \{L_1\} \subseteq p$ and $C_3 \sqsubseteq p$ contains a σ-complementary connection, and there is a μ_2 such that every p through M^{μ_2} with $Path \subseteq p$ and $C \sqsubseteq p$ contains a σ-complementary connection. Then $\dfrac{\dfrac{Proof_1}{C_3,\,M[C_1\backslash C_2],\,Path\cup\{L_1\}} \quad \dfrac{Proof_2}{C,\,M,\,Path}}{C\cup\{L_1\},\,M,\,Path}\;E$ is a connection proof for $C\cup\{L_1\}, M, Path$. This last extension step is illustrated below. It has to be shown that there is a multiplicity μ' and a substitution σ' such that every path p' through $M^{\mu'}$ with $Path \subseteq p'$ and $C \cup \{L_1\} \sqsubseteq p'$ contains a σ'-complementary connection. Let M' be the matrix M in which the (sub-)matrix $\{\ldots, C_1, \ldots\}$ that contains C_1 is replaced by the matrix $\{\ldots, C_1, C_2, \ldots\}$, i.e. the clause C_2 is added to M as shown below.



According to Def. 8 the following cases for the extension clause C_1 need to be considered:

1. If the extension clause C_1 contains a literal of $Path$, then every path p' through M' with $Path \cup \{L_1\} \subseteq p'$ is a superset path through C_2 as well, i.e. $C_2 \sqsubseteq p'$.
2. Otherwise, C_1 is α-related to all literals in $Path \cup \{L_1\}$ occurring in M'.
   (a) If C_1 has no parent clause, then for every p' through M' $C_1 \sqsubseteq p'$ holds. Then for every p' through M' with $Path \cup \{L_1\} \subseteq p'$ it is $C_1 \sqsubseteq p'$ and hence $C_2 \sqsubseteq p'$.
   (b) Otherwise, the parent clause of C_1 contains an element of $Path \cup \{L_1\}$. As C_1 is α-related to $Path\cup\{L_1\}$, there is a $\hat{C}$ containing a literal of $Path\cup\{L_1\}$ with $C_1, \hat{C} \in \hat{M}$. Therefore, for every p' through M' with $Path \cup \{L_1\} \subseteq p'$ it is $C_1 \sqsubseteq p'$ and thus $C_2 \sqsubseteq p'$.

C_3 is the β-clause of C_2 with respect to L_2, i.e. L_2 and all clauses that are α-related to L_2 are deleted from C_2. According to Def. 9 the only element deleted from a *clause* is the literal L_2. Therefore, for all p' through M' with $Path \cup \{L_1\} \subseteq p'$ it is $C_3 \sqsubseteq p'$ or $\{L_2\} \sqsubseteq p'$. The same holds if copies of clauses are added to M'. Let μ' be the multiplicity where all copies according to μ_1 and μ_2 as well as the clause copy C_2 are considered. Let $\sigma' := \tau \circ \sigma$. Then $C_3 \sqsubseteq p'$ or $L_2 \in p'$ holds for every p' through $M^{\mu'}$ with $Path \cup \{L_1\} \subseteq p'$. As there is a proof for $C_3, M[C_1\backslash C_2], Path \cup \{L_1\}$ with σ, every p through $(M[C_1\backslash C_2])^{\mu_1}$, and hence through $M^{\mu'}$, with $Path \cup \{L_1\} \subseteq p$ and $C_3 \sqsubseteq p$ contains a σ-complementary, and hence a σ'-complementary connection. Furthermore, every p' with $Path \cup \{L_1\} \subseteq p'$ that includes L_2 contains a σ'-complementary connection as $\sigma'(L_1) = \sigma'(\overline{L_2})$. Therefore, every p' through $M^{\mu'}$ with $Path \cup \{L_1\} \subseteq p'$ contains a σ'-complementary connection. Then every p' through $M^{\mu'}$, with $Path \subseteq p'$ and $\{L_1\} \sqsubseteq p'$ contains a σ'-complementary connection. As there is a proof for $C, M, Path$ with σ, every p through M^{μ_2}, and thus through $M^{\mu'}$, with $Path \subseteq p$ and $C \sqsubseteq p$ contains a σ-complementary, and hence a σ'-complementary, connection. Then every p' through $M^{\mu'}$, with $Path \subseteq p'$ and $C \cup \{L_1\} \sqsubseteq p'$ contains a σ'-complementary connection.

4. Decomposition: Let $\frac{Proof}{C\,\cup\,C_1,\,M,\,Path}$ be a connection proof for $C \cup C_1, M, Path$ for some σ. According to *IH* there is a μ such that every p through M^{μ} with $Path \subseteq p$ and $C \cup C_1 \sqsubseteq p$

contains a σ-complementary connection. Then $\dfrac{\dfrac{Proof}{C \cup C_1, M, Path}}{C \cup \{M_1\}, M, Path}\; D$ with $C_1 \in M_1$ is a connection proof for $C \cup \{M_1\}, M, Path$. Let $\mu' := \mu$ and $\sigma' := \sigma$. Every p' through $M^{\mu'}$ with $Path \subseteq p'$ and $C \cup \{M_1\} \sqsubseteq p'$ contains a σ'-complementary connection as well, since $C_1 \in M_1$ and thus for all p' the following holds: if $C \cup \{M_1\} \sqsubseteq p'$ then $C \cup C_1 \sqsubseteq p'$. □

Theorem 3 (Correctness of the Non-Clausal Connection Calculus). *A formula F is valid in classical logic, if there is a non-clausal connection proof for its matrix M.*

Proof. Let M be the matrix of F. If there is a non-clausal connection proof for $\varepsilon, M, \varepsilon$, it has the form $\dfrac{\dfrac{\vdots}{C_2, M, \{\}}}{\varepsilon, M, \varepsilon}\; S$ in which the clause C_2 is a copy of $C_1 \in M$. There has to be a proof for $C_2, M, \{\}$ for some σ. According to Lemma 1 there is a μ such that every path p through M^μ with $\{\} \subseteq p$ and $C_2 \sqsubseteq p$ contains a σ-complementary connection. As $\{\} \subseteq p$ and $C_2 \sqsubseteq p$ hold for all p through M^μ, every path through M^μ contains a σ-complementary connection. According to Theorem 1 the formula F is valid. □

5.2 Completeness

Definition 13 (Vertical Path Through Clause). *Let X be a matrix, clause, or literal. A* vertical path *p through X, denoted by $p \| X$, is a set of literals of X and inductively defined as follows: $\{L\} \| L$ for literal L; $p \| M$ for matrix M where $p \| C$ for some $C \in M$; $p \| C$ for clause $C \neq \{\}$ where $p = \bigcup_{M_i \in C} p_i$ and $p_i \| M_i$; $\{\} \| C$ for clause $C = \{\}$.*

Lemma 2 (Clauses and Vertical Paths). *Let M be the (non-clausal) matrix of a first-order formula F and M' be the matrix of the standard translation of F into clausal form (see Example 1). Then for every clause $C' \in M'$ there is an "original" clause $C \in M$ with $C' \| C$. If there is a connection proof for $C', M, Path$, then there is also a connection proof for $C, M, Path$. This also holds if the clauses $D' := C' \setminus \{L\}$ and $D := \beta\text{-}clause_L(C)$ are used, for some literal L, instead of C' and C, respectively.*

Proof. The existence of a clause $C \in M$ for every $C' \in M'$ with $C' \| C$ follows from Def. 2 and Def. 13. $C, M, Path$ can be derived from $C', M, Path$ by several decomposition steps. In D the literal L and all clauses that are α-related to L, i.e. that do not contain literals of D', are deleted from C. Therefore, it is $D' \| D$ and $D, M, Path$ can be derived from $D', M, Path$ as well. □

Lemma 3 (Completeness of the Non-Start Rules). *Let M be the matrix of a first-order formula F and M' be the matrix of the standard translation of F into clausal form. If there is a clausal connection proof for $C, M', Path$ with the term substitution σ, then there exists a non-clausal connection proof for $C, M, Path$ with σ.*

Proof. The idea is to translate a clausal connection proof for M' into a non-clausal connection proof for M. The proof is by structural induction on the construction of a clausal connection proof. *Induction hypothesis*: If there is a clausal connection proof for $C, M', Path$ with the term substitution σ, then there exists a non-clausal connection proof for $C, M, Path$ with the substitution σ. For the *induction start* the (only) axiom of the calculus is considered. For the *induction step* reduction and extension rules are considered. The axiom and reduction rule are essentially identical for the clausal and non-clausal calculus. For the extension rule Lemma 2 has to be applied. The details of the (straightforward) proof are left to the interested reader. □

Theorem 4 (Completeness of the Non-Clausal Connection Calculus). *If a formula F is valid in classical logic, there is a non-clausal connection proof for its matrix M.*

Proof. Let F be a valid formula, M be the non-clausal matrix of F and M' its matrix in clausal form. According to Theorem 2 there exists a clausal connection proof for M' and for $\varepsilon, M', \varepsilon$. Let C_2', M', ε be the premise of the start step in which C_2' is a copy of $C_1' \in M'$. According to Lemma 3 there is a non-clausal proof for C_2', M, ε. Let C_1 be the original clause of C_1' in M. According to Lemma 2 there is a non-clausal proof for C_2, M, ε where C_2 is a copy of C_1. Thus, there is a proof for $\varepsilon, M, \varepsilon$ and for M. □

5.3 Complexity

Definition 14 (Size of Connection Proof). *The* size *of a (clausal or non-clausal) connection proof is the number of proof steps in the connection proof.*

Theorem 5 (Linear Simulation of Clausal Calculus). *Let M be the matrix of a formula F and M' be the matrix of the standard translation of F into clausal form. Furthermore, let n be the size of a clausal connection proof for M' and m be the size of its largest subgoal clause. Then there is a non-clausal proof for M with size $\mathcal{O}(m \cdot n)$.*

Proof. The same technique used for the proof of Lemma 3 can be used to translate a clausal proof for M' into a non-clausal proof for M. Reduction steps can be directly translated without any modifications. Every start or extension step is translated into one start or extension step and a number of decomposition steps in the non-clausal proof, respectively. The number of decomposition steps is limited by twice the size of the subgoal clause in the clausal proof. □

Theorem 6 (No Polynomial Simulation of Non-Clausal Calculus). *Let M be the matrix of a formula F and M' be the matrix of the standard translation of F into clausal form. There is a class of formulae for which there is no clausal proof for M' with size $\mathcal{O}(n^k)$ for a fixed $k \in I\!N$, where n is the size of a non-clausal proof for M.*

Proof. Consider the (valid) formula class $((\forall x_1 P_1 x_1) \Rightarrow P_1 c_1)) \wedge \ldots \wedge ((\forall x_m P_m x_m) \Rightarrow P_m c_m))$ for some $m \in I\!N$ and the graphical representation of its matrix M shown below. The non-clausal connection proof for M with the term substitution $\sigma(x_i) = c_i$, for $1 \le i \le m$, has the following (simplified) graphical matrix representation

$$\left[\left[\begin{array}{c}\left[\,[P_1^1 x_1]\ [P_1^0 c_1]\,\right]\\ \vdots\\ \left[\,[P_m^1 x_m]\ [P_m^0 c_m]\,\right]\end{array}\right]\right]$$

and consists of one start step, m decomposition steps, and m extension steps. Therefore, the size of the non-clausal connection proof is $n = 2m+1$. The clausal matrix M' has the form $\{\{P_1^1 x_1, \ldots, P_m^1 x_m\}, \{P_1^1 x_1, \ldots, P_m^0 c_m\}, \ldots, \{P_1^0 c_1, \ldots, P_m^0 c_m\}, \{P_1^0 c_1, \ldots, P_m^0 c_m\}\}$ and consists of $m \cdot 2^m$ literals. In a clausal proof for M' every clause of M' has to be considered, i.e. every literal of M' has to be an element of at least one connection. Therefore, $m \cdot 2^{m-1}$ is the minimal number of connections and the minimal size of every clausal connection proof for M'. Hence, there is no clausal connection proof for M' with size $\mathcal{O}(n^k)$ for some fixed $k \in I\!N$. □

Theorem 6 holds for the structure-preserving translation into clausal form introduced in [12] as well. An example for an appropriate problem class is given in [2].

6 Optimizations and Extensions

In this section the connection calculus is further simplified. Some optimizations techniques and an extension of the calculus to some non-classical logics are described.

6.1 A Simplified Connection Calculus

If the matrices that are used in the non-clausal connection calculus are slightly modified, the start and reduction rule are subsumed by the decomposition and extension rule.

Definition 15 (Simplified Connection Calculus). *The* simplified connection calculus *consists only of the axiom, the extension rule and the decomposition rule of the non-clausal connection calculus (see Definition 10). It is shown in Figure 8.*

Axiom (A)	$\overline{\{\}, M, Path}$
Extension (E)	$\dfrac{C_3, M[C_1 \backslash C_2], Path {\cup} \{L_1\} \qquad C, M, Path}{C {\cup} \{L_1\}, M, Path}$ and $C_3{:=}\beta\text{-}clause_{L_2}(C_2)$, C_2 is copy of C_1, C_1 is an extension clause of M wrt. $Path \cup \{L_1\}$, C_2 contains L_2 with $\sigma(L_1) = \sigma(\overline{L_2})$
Decomposition (D)	$\dfrac{C \cup C_1, M, Path}{C {\cup} \{M_1\}, M, Path}$ with $C_1 {\in} M_1$

Fig. 8. The simplified connection calculus

Theorem 7 (Correctness & Completeness of the Simplified Connection Calculus). *Let F be a first-order formula and M its matrix. Let M^* be the matrix M, in which all literals $L \in C$ with $|C| > 1$ in M are replaced by the matrix $\{\{L\}\}$. Then F is valid in classical logic iff there is a simplified connection proof for $\{M^*\}, M^*, \{\}$.*

Proof. The start rule and the reduction rule are subsumed by the decomposition rule and the extensions rule, respectively. As the remaining rules of the calculus are not modified, the simplified connection calculus is correct and complete. □

6.2 Optimizations

Positive Start Clause. As for the clausal connection calculus, the start clause of the non-clausal connection calculus can be restricted to positive clauses.

Definition 16 (Positive Clause). *A clause C is a* positive clause *iff there is a vertical path p through C, i.e. $C \| p$, that contains only literals with polarity 0.*

Lemma 4 (Positive Start Clause). *The non-clausal connection calculus remains correct and complete, if the clause C_1 of the start rule is restricted to positive clauses and all clauses in C_2 that are not positive are deleted from C_2.*

Proof. Correctness is preserved. Completeness follows from the fact that every connection proof for M has to use all literals (within connections) from a vertical path p through some clause $C \in M$ such that p contains only literals with polarity 0. □

Regularity. Regularity is an effective technique for pruning the search space in clausal connection calculi [9]. It can be used for the non-clausal calculus as well.

Definition 17 (Regularity). *A connection proof is* regular *iff there are no two literals* L_1, L_2 *in the active path with* $\sigma(L_1) = \sigma(L_2)$.

The regularity condition is integrated into the calculus of Figure 6 by adding a restriction to the reduction and extension rule: $\forall L' \in C \cup \{L_1\} : \sigma(L') \notin \sigma(Path)$.

Lemma 5 (Regularity). *M is valid iff there is a regular connection proof for M.*

Proof. Regularity preserves correctness. Completeness follows from the fact that the clausal connection calculus with regularity is complete [9] and that it can be simulated by the non-clausal calculus with regularity (see Section 5.2). □

Restricted Backtracking. Proof search in the (clausal and non-clausal) connection calculus is *not* confluent, i.e. it might end up in dead ends. To achieve completeness *backtracking* is required (see remarks in Section 2 and 4.2), i.e. alternative rules or rule instances need to be considered. The main idea of *restricted backtracking* [12] is to cut off any so-called *non-essential backtracking* that occurs after a literal is *solved*. Even though this strategy is incomplete, it is very effective [12]. It can be used straight away to prune the search space in the non-clausal connection calculus as well.

6.3 Non-Classical Logics

The matrix characterization of classical validity (see Theorem 1) can be extended to some non-classical logics, such as modal and intuitionistic logic [18]. To this end a *prefix*, i.e. a string consisting of variables and constants, which essentially encodes the *Kripke world* semantics, is assigned to each literal. For a σ-complementary connection $\{L_1 : p_1, L_2 : p_2\}$ not only the terms of both literals need to unify under a term substitution σ, i.e. $\sigma(L_1) = \sigma(\overline{L_2})$, but also the corresponding prefixes p_1 and p_2 are required to unify under a prefix substitution σ', i.e. $\sigma'(p_1) = \sigma'(p_2)$. Therefore, by adding prefixes to the presented non-clausal connection calculus, it can be used for some non-classical logics as well [8]. For the proof *search* an additional *prefix unification* algorithm [11, 17] is required that unifies the prefixes of the literals in every connection.

7 Conclusion

A *formal* non-clausal connection calculus has been introduced that can be used for proof search in classical and some non-classical first-order logics. It does not require the translation of the input formula into any clausal form but preserves its structure. The calculus generalizes the clausal connection calculus by modifying the extension rule and adding a decomposition rule. Copying of clauses is done in a dynamic way and significant redundancy is removed by considering only β-clauses for new subgoal clauses. Thus, the calculus combines the advantages of a non-clausal proof search in tableau calculi [6] with the more goal-oriented search of clausal connection calculi [4].

In [7] a technique similar to β-clauses is used to prove completeness for non-clausal connection tableaux. But this technique is not used explicitly within the tableau calculus itself. Furthermore, only ground formulae are considered and a more general regularity condition is used, which is not restricted to literals of the active path.

This paper provides the formal basis for a planned *competitive* implementation of a non-clausal connection calculus in an elegant and compact style [12, 13]. Future work includes the development of further *non-clausal* optimization techniques.

Acknowledgements. The author would like to thank Wolfgang Bibel for his helpful comments on a preliminary version of this paper.

References

1. Andrews, P.B.: Theorem Proving via General Matings. *Journal of the ACM* 28, 193–214 (1981).
2. Antonsen, R., Waaler, A.: Liberalized Variable Splitting. *Journal of Automated Reasoning* 38, 3–30 (2007).
3. Bibel, W.: Matings in Matrices. *Communications of the ACM* 26, 844–852 (1983).
4. Bibel, W.: *Automated Theorem Proving*. Vieweg, Wiesbaden (1987).
5. Gentzen, G.: Untersuchungen über das logische Schließen. *Mathematische Zeitschrift* 39. 176–210, 405–431 (1935).
6. Hähnle, R.: Tableaux and Related Methods. In: Robinson, A., Voronkov, A. (eds.) *Handbook of Automated Reasoning*, pp. 100–178. Elsevier, Amsterdam (2001).
7. Hähnle, R., Murray, N.V., Rosenthal, E.: Linearity and Regularity with Negation Normal Form. *Theoretical Computer Science* 328, 325–354 (2004).
8. Kreitz, C., Otten, J.: Connection-based Theorem Proving in Classical and Non-classical Logics. *Journal of Universal Computer Science* 5, 88–112 (1999).
9. Letz, R., Stenz, G.: Model Elimination and Connection Tableau Procedures. In: Robinson, A., Voronkov, A. (eds.) *Handbook of Automated Reasoning*, pp. 2015–2114. Elsevier, Amsterdam (2001).
10. Loveland, D.: Mechanical Theorem-Proving by Model Elimination. *Journal of the ACM* 15, 236–251 (1968).
11. Otten, J.: Clausal Connection-Based Theorem Proving in Intuitionistic First-Order Logic. In: Beckert, B. (ed.) *TABLEAUX 2005*. LNAI, vol. 3702, pp. 245–261. Springer, Heidelberg (2005).
12. Otten, J.: Restricting Backtracking in Connection Calculi. *AI Communications* 23, 159–182 (2010).
13. Otten, J., Bibel, W.: leanCoP: Lean Connection-based Theorem Proving. *Journal of Symbolic Computation* 36, 139–161 (2003).
14. Plaisted, D., Greenbaum, S.: A Structure-preserving Clause Form Translation. *Journal of Symbolic Computation* 2, 293–304 (1986).
15. Robinson, A.: A Machine-oriented Logic Based on the Resolution Principle. *Journal of the ACM* 12, 23–41 (1965).
16. Schmitt, S., Lorigo, L., Kreitz, C., Nogin, A.: JProver: Integrating Connection-based Theorem Proving into Interactive Proof Assistants. In: Goré, R., Leitsch, A., Nipkow, T. (eds.) *IJCAR 2001*. LNAI, vol. 2083, pp. 421–426. Springer, Heidelberg (2001).
17. Waaler, A.: Connections in Nonclassical Logics. In: Robinson, A., Voronkov, A. (eds.) *Handbook of Automated Reasoning*, pp. 1487–1578. Elsevier, Amsterdam (2001).
18. Wallen, L.: *Automated Deduction in Nonclassical Logics*. MIT Press, Cambridge (1990).