Formal Derivations and Natural Proofs (Higher Set Theory)

by Peter Koepke

University of Bonn

Wintersemester 2011/12

Abstract

Remark: This interdisciplinary course is centered around many aspects of the topic of "mathematical proof", involving formal logic, linguistics, computer science, and even philosophy of mathematics. It is recommended as a sequel to a standard Mathematical Logic course as read at the University of Bonn, or as a general advanced logic course. Due to its broad character it does not squarely fit into the classifications of Bachelor or Master courses. For bureaucratic reasons the course is classified as the Master module Higher Set Theory. The course includes a certain amount of set theory since set theory is a standard background theory for mathematical proofs. It is understood that the course can also be used for credits in the Bachelor of Mathematics programme at the University of Bonn.

Contents: Formal logic models argumentative methods (in idealised domains). Apparently, formal derivations are very different from ordinary mathematical proofs. This lecture course examines both perspectives in some detail, with the intention of bridging the gap. On the formal side we present calculi and proof algorithms designed for naturality and efficiency. We also present various systems of formal mathematics. On the other hand we study the common language of mathematics and show how methods of formal linguistics may be used to restrict to a controlled natural language with a definite formal interpretation. The lecture course provides theoretical foundations for the Naproche system (**Na**tural language **pro**of **che**cking) which is being developed at Bonn (see `http://www.naproche.net`). The course also introduces the theoretical basis of logic programming and presents and applies the `Prolog` programming language.

Topics include

- review of standard first-order calculi
- a "natural language" first-order calculus
- analysing and processing simple natural language
- the resolution calculus and logic programming
- `Prolog` and natural language processing
- formal mathematics with a discussion of the underlying formal systems
- methods of automatic theorem proving
- ...
- the Naproche system

Table of contents

1 Formal languages and grammars

Definition 1. *An* alphabet *is a set* $\Sigma \neq \emptyset$*. Let*

$$\Sigma^* = \Sigma^{<\omega} = \{w | w \text{ is a finite sequence from } \Sigma\}.$$

A formal language *over* Σ *is a set* $L \subseteq \Sigma^*$*.*

Formal languages can be defined in various ways. One of the ways to define a language by finitary means are grammars.

Definition 2. *A* grammar *is a 4-tupel* $G=(V,\Sigma,P,S)$ *with*

- V *is a finite set of* variables *(non-terminals)*
- Σ *is a finite alphabet with* $\Sigma\cap V=\emptyset$
- $P\subseteq(V\cup\Sigma)^*\times(V\cap\Sigma)^*$ *is a finite set of* rules *or* productions
- $S\in V$ *is the* start variable

For $u,v\in(V\cup\Sigma)^*$ *define* $u\Rightarrow_G v$ *iff there are* $x,y\in(V\cup\Sigma)^*$ *and a rule* $(y,y')\in P$ *such that* $u=xyz$ *and* $v=xy'z$. *We also write* $y\rightarrow y'$. *Let* $\Rightarrow_G^*$ *be the reflexive and transitive hull of* $\Rightarrow_G$. *Then the language represented, generated, or defined by* G *is*

$$L(G)=\{w\in\Sigma^*|S\Rightarrow_G^* w\}.$$

If $w\in L(G)$ *and* $S\Rightarrow_G w_1\Rightarrow_G w_2\ldots\ldots\Rightarrow_G w_n=w$ *then* $(S,w_1,\ldots,w_n)$ *is a* derivation *of* w_n.

Examples are: some natural language sentences

- *sentence* $\rightarrow$ *nounphrase verbphrase*
- *nounphrase* $\rightarrow$ *article noun*
- *article* $\rightarrow$ the
- *article* $\rightarrow$ a
- *nown* $\rightarrow$ dog
- *verbphrase* $\rightarrow$ barks

First order formulas

- *formula* $\rightarrow$ $\neg$ *formula* etc

Well-formed bracketed formulas

-

We then follow the textbook
Uwe Schöning, Theoretische Informatik kurz gefasst
BI Wissenschaftsverlag, chapters 1.1.1 (grammars), 1.1.2 (Chomsky-hierarchy), 1.1.5. (Backus-Naur notation), 1.4 (Turing machines and Type 0 languages).

2 Some first-order calculi

2.1 A Hilbert style calculus from the course by Geschke

The logic course of Stefan Geschke introduced the following first-order calculus:

This calculus has the following features:

- every formula in a derivation of $T \vdash \varphi$ is a consequence of T
- propositional tautologies are immediately available, without derivation
- there is no hypothetical reasoning like proof by contradiction, or case distinction
- there can be no local assumptions in a proof like "Assume φ", "Take x such that ..."

This makes proofs somewhat unnatural:

A (trivial) standard proof of this property would look like:

Lemma 3. *A a group, $\forall x (\forall y\colon y \cdot x = y \rightarrow x = e)$.*

Proof. Consider x and assume $\forall y\colon y \cdot x = y$. Then

$$x = e \cdot x = e.$$

□

The "bureaucratic" bookkeeping of tautologies makes the Geschke proof difficult to read and obscures what a human reader may take as the real content of the argument. Maybe the bookkeeping can be handed over to a computer?

2.2 A sequent calculus

The following was introduced in my course 2 years ago.

Definition 4. *A finite sequence* $(\varphi_0, \ldots, \varphi_{n-1}, \varphi_n)$ *is called a* sequent*. The initial segment* $\Gamma = (\varphi_0, \ldots, \varphi_{n-1})$ *is the* antecedent *and* φ_n *is the* succedent *of the sequent. We usually write* $\varphi_0 \ldots \varphi_{n-1}\, \varphi_n$ *or* $\Gamma\, \varphi_n$ *instead of* $(\varphi_0, \ldots, \varphi_{n-1}, \varphi_n)$*. To emphasize the last element of the antecedent we may also denote the sequent by* $\Gamma'\, \varphi_{n-1}\, \varphi_n$ *with* $\Gamma' = (\varphi_0, \ldots, \varphi_{n-2})$*.*

A sequent $\varphi_0 \ldots \varphi_{n-1}\, \varphi$ *is* correct *if* $\{\varphi_0 \ldots \varphi_{n-1}\} \vDash \varphi$*.*

Definition 5. *The* sequent calculus *consists of the following (sequent-)rules:*

- *monotonicity* (MR) $\dfrac{\Gamma \quad\quad \varphi}{\Gamma \quad \psi \quad \varphi}$

- *assumption* (AR) $\dfrac{}{\Gamma \quad \varphi \quad \varphi}$

- $\rightarrow$*-introduction* $(\rightarrow I)$ $\dfrac{\Gamma \quad \varphi \quad \psi}{\Gamma \quad\quad \varphi \rightarrow \psi}$

- $\rightarrow$*-elimination* $(\rightarrow E)$ $\dfrac{\begin{array}{l}\Gamma \quad \varphi \\ \Gamma \quad \varphi \rightarrow \psi\end{array}}{\Gamma \quad \psi}$

- $\perp$*-introduction* $(\perp I)$ $\dfrac{\begin{array}{l}\Gamma \quad \varphi \\ \Gamma \quad \neg\varphi\end{array}}{\Gamma \quad \perp}$

- $\perp$*-elimination* $(\perp E)$ $\dfrac{\Gamma \quad \neg\varphi \quad \perp}{\Gamma \quad\quad \varphi}$

- $\forall$*-introduction* $(\forall I)$ $\dfrac{\Gamma \quad \varphi\frac{y}{x}}{\Gamma \quad \forall x \varphi}$ *, if* $y \notin \mathrm{free}(\Gamma \cup \{\forall x \varphi\})$

- $\forall$*-elimination* $(\forall E)$ $\dfrac{\Gamma \quad \forall x \varphi}{\Gamma \quad \varphi\frac{t}{x}}$ *, if* $t \in T^S$

- $\equiv$*-introduction* $(\equiv I)$ $\dfrac{}{\Gamma \quad t \equiv t}$ *, if* $t \in T^S$

- $\equiv$*-elimination* $(\equiv E)$ $\dfrac{\begin{array}{l}\Gamma \quad \varphi\frac{t}{x} \\ \Gamma \quad t \equiv t'\end{array}}{\Gamma \quad \varphi\frac{t'}{x}}$

The deduction relation *is the smallest subset* $\vdash \subseteq \mathrm{Seq}(S)$ *of the set of sequents which is closed under these rules. We write* $\varphi_0 \ldots \varphi_{n-1} \vdash \varphi$ *instead of* $\varphi_0 \ldots \varphi_{n-1}\, \varphi \in \vdash$*. For* Φ *an arbitrary set of formulas define* $\Phi \vdash \varphi$ *iff there are* $\varphi_0, \ldots, \varphi_{n-1} \in \Phi$ *such that* $\varphi_0 \ldots \varphi_{n-1} \vdash \varphi$*. We say that* φ *can be* deduced *or* derived *from* $\varphi_0 \ldots \varphi_{n-1}$ *or* Φ*, resp. We also write* $\vdash \varphi$ *instead of* $\emptyset \vdash \varphi$ *and say that* φ *is a* tautology*.*

In the sequent calculus, tautologies have to be derived since they are no axioms. Let us first give an ordinary proof of the tautology

Lemma 6. $\neg\neg\varphi \rightarrow \varphi$*.*

Proof. Assume $\neg\neg\varphi$. Assume for a contradiction that $\neg\varphi$. This is a contradiction to the assumption. Thus φ holds. □

We model that argument as follows:

1.	$\neg\neg\varphi$	$\neg\neg\varphi$	"Assume $\neg\neg\varphi$"
2.	$\neg\neg\varphi$ $\neg\varphi$	$\neg\neg\varphi$	
3.	$\neg\neg\varphi$ $\neg\varphi$	$\neg\varphi$	"Assume for a contradiction that $\neg\varphi$"
4.	$\neg\neg\varphi$ $\neg\varphi$	$\perp$	"This is a contradiction to the assumption"
5.	$\neg\neg\varphi$	φ	"Thus φ holds"
6.		$\neg\neg\varphi \rightarrow \varphi$	"**Lemma 4**"

The sequent calculus has the following features:

- there can be hypothetical reasoning
- all local assumptions have to be carried along in the antecedens of the sequent

Sequents can be reduced to formulas in a Hilbert-style proof: the sequent $\varphi_1 ... \varphi_n \psi$ corresponds to $(\varphi_1 \wedge ... \wedge \varphi_n) \rightarrow \psi$.

2.3 A "natural language" calculus

We consider a fragment of (mathematical) English formed by the words "contradiction", "not", "implies", "for all", "holds", "assume", and "thus":

Definition 7. *The collection of (natural language) formulas is defined by:*

- *every* relational formula $R(x_1, ..., x_n)$ *is a formula; for specific relations like "odd" or "="* *one may also write as usual "x is odd" or "$x < y$" instead of* $R(x)$ *or* $R(x, y)$*;*
- *"contradiction" is a formula;*
- *if A is a formula then "not A" is a formula;*
- *if A and B are formulas then "(A implies B)" is a formula;*
- *if A is a formula then "for all x (holds) A" is a formula.*

Brackets in formulas may be omitted according to the usual conventions.

Consider a calculus for such formulas:

Definition 8. *The* basic proof rules *are given by the rules of*

a) *contradiction* $\dfrac{A \qquad \mathit{not}\ A}{\mathit{contradiction}}$

b) *proof by contradiction* $\dfrac{\mathit{not}\ A\ \mathit{implies}\ \mathit{contradiction}}{A}$

c) *modus ponens* $\dfrac{A\ \mathit{implies}\ B \quad A}{B}$

d) *instantiation* $\dfrac{\mathit{for\ all}\ x\ \mathit{holds}\ A(x)}{A(y)}$

e) *generalization* $\dfrac{A(y)}{\mathit{for\ all}\ x\ \mathit{holds}\ A(x)}$

Note that the first four rules are *correct* in the following sense: if the assumptions of the rule hold in some structure then the conclusion also holds in the structure. The situation is more complex for the generalization rule which can only be applied in certain proof situations.

Definition 9. *A* (mathematical) text *is a sequence* $T = S_1 ... S_l$ *of* statements *where each statement is of the form* $S_k =$ *"Assume* A_k*."*, $S_k =$ *"*A_k*."*, *or* $S_k =$ *"Thus" for some formula* A_k*.*

A text is a *proof* if every line within the text is formally justified, e.g., that it can be generated by a proof rule from previous lines which are "visible" to the present line. Visibility can be calculated via *indentation depths*: a previous line is visible if it is not "blocked" by some "Thus" which has the same indentation level as that previous line. This is formalized by the following definitions.

Definition 10. *Let $T = S_1 ... S_l$ be a mathematical text. Then define:*

a) *For $k < l$ let*

$$\mathrm{ind}_T(k) = |\{j \leqslant k \,|\, S_j \text{ starts with "Assume"}\}| - |\{j < k \,|\, S_j \text{ starts with "Thus"}\}|$$

be the (indentation) depth *of S_k in T, it is given by the difference between the numbers of previous assumptions ("Assume") and the previous conclusions ("Thus").*

b) *The text T is* properly indented *if $\mathrm{ind}_T(k) \geqslant 0$ for all $k < l$, i.e., we cannot have more conclusions than assumptions.*

c) *For $i < k < l$ the line number i is* visible *from line number k if there is no j, $i \leqslant j < k$ such that $S_j =$ "Thus" and $\mathrm{ind}_T(i) = \mathrm{ind}_T(j)$. In case i is visible from k we also say that the formula A_i and the free variables of A_i are visible from k.*

Definition 11. *Let $T = S_1 ... S_l$ be a mathematical text. Let Φ be a set of formulas.*

a) *T is a* (formal) proof *from Φ if T is properly indented, and for all $k < l$ one of the following holds:*

   i. *$S_k =$ "Assume A_k .", or $S_k =$ "Thus"; this means that we can introduce an assumption or try to conclude a subargument at any place in a proof;*

   ii. *$S_k =$ "A_k ." where $A_k \in \Phi$ or $A_k = A_i$, $i<k$ where the line S_i is* visible *by S_k ; this means that the "axioms" contained in Φ or visible statements established previously can be used freely;*

   iii. *$S_k =$ "A_k ." where A_k can be produced by one of the basic proof rules from formulas which are elements of Φ or which are visible from k; moreover, if A_k is of the form $A_k =$ "for all x holds $A(x)$" and is produced by the rule of generalization from the formula $A(y)$, we also require that the variable $y \notin \mathrm{free}(\Phi)$ and that y is not visible from k as a free variable; so the generalization from $A(y)$ to "for all x holds $A(x)$" is possible if y was a "general" variable without further specifications in Φ or previous relevant formulas;*

   iv. *$S_k =$ "A_i implies A_{k-2}" where $S_{k-1} =$ "Thus" and $i \leqslant k - 2$ is the minimal line number which is visible from $k - 1$; we say that S_k is produced by the* rule of implication*; the result of a subargument from the assumption A_i to the conclusion A_{k-2} is the implication "A_i implies A_{k-2}".*

b) *T is a* (formal) proof of *A from Φ if $A = A_l$ and $\mathrm{ind}_T(l) = 0$; the latter means that all subarguments have been concluded.*

c) *A is* (formally) provable *from Φ if there exists a proof of A from Φ.*

d) *A is* (formally) provable *if it is provable from the empty set $\emptyset$, i.e., without further hypothesis.*

We can then prove

Lemma 12. *Not not φ implies φ.*

as follows:

Proof. Assume not not φ. Assume not φ. Contradiction. Thus not φ implies contradiction. φ. Thus not not φ implies φ. □

This calculus has the following features:

– the statements in the text approximate legitimate sentences of natural language

– the calculus does not contain equality and does not allow terms

3 A completeness theorem

We prove a completeness theorem for the "natural language" calculus. We begin by proving some *derived rules* which may also be used conveniently in further proofs.

Proposition 13. *Let A, B be formulas. Then A is provable from "B implies A", "(not B) implies A". This justifies the use of the* derived rule *of* case distinction*:*

$$\frac{B \text{ implies } A \quad \text{not } B \text{ implies } A}{A}\ .$$

Proof. The following is a proof of A from $F_1 =$"B implies A" and $F_2 =$"not B implies A". We also state the rules which are applied and the local depths and hypotheses.

k	Statement	Rule ... with hypothesis ...	$\mathrm{ind}_T(k)$	visible lines
1	Assume not A.	-	1	-
2	Assume not B.	-	2	1
3	A.	*modus ponens* with 2, F_2	2	1,2
4	Contradiction.	*contradiction* with 1, 3	2	1,2,3
5	Thus	-	2	1,2,3,4
6	not B implies a contradiction.	*implication*	1	1
7	B.	*proof by contradiction* with 6	1	1,6
8	A.	*modus ponens* with 7, F_1	1	1,6,7
9	Contradiction.	*contradiction* with 1, 8	1	1,6,7,8
10	Thus	-	1	1,6,7,8,9
11	not A implies a contradiction.	*implication*	0	-
12	A.	*proof by contradiction* with 11	0	11

□

Proposition 14. *Let A be a formula. Then A is provable from "contradiction". This justifies the use of the* derived rule *of* ex falsum libenter*:*

$$\frac{\textit{contradiction}}{A}\ .$$

Proof. The following is a formal proof of A from $F =$"contradiction".

k	Statement	Rule ... with hypothesis ...	$\mathrm{ind}_T(k)$	visible lines
1	Assume not A.	-	1	-
2	Contradiction.	Copying F	1	1
3	Thus	-	1	2
4	not A implies a contradiction.	*implication*	0	-
5	A.	*proof by contradiction* with 4	0	4

□

A formal proof as defined in Definition 11, though formulated in a "poor" vocabulary and grammar, can be read as a proof in the ordinary mathematical sense. Since mathematical proofs prove universally valid statements, we obtain the *correctness theorem*:

Theorem 15. *If a formula A is provable then it is universally valid.*

Our proof of GÖDEL's *completeness theorem* uses the approach by L. HENKIN [?]. Given a formula which is not provable build a HENKIN set of formulas (denoted by $\mathcal{H}$ in the subsequent proof) which describes a structure in which A fails. Then build such a structure $\mathcal{S}$ out of the terms of the language.

Theorem 16. *If a formula A is universally valid it is provable.*

Proof. Assume that A is not provable. It suffices to show that A is not universally valid by constructing a structure $\mathcal{S}$ in which A does not hold.

We shall recursively define a sequence $A_1, A_2, A_3, \ldots$ of formulas which describe the structure $\mathcal{S}$. Along the recursion we maintain that A is not provable from $A_1, \ldots, A_n$. To extend the sequence, we postulate two extension properties: by (1), every formula can be decided positively or negatively in the construction; by (2), we can add a counterexample to every universal formula which is not valid.

(1) Assume that A is not provable from $A_1, \ldots, A_n$ and let B be a formula. Then A is not provable from $A_1, \ldots, A_n, B$, *or* A is not provable from $A_1, \ldots, A_n$, "not B".

Proof. Assume not. Assume that the mathematical text Proof1, A is a proof of A from $A_1, \ldots, A_n$, B *and* that Proof2, A is a proof of A from $A_1, \ldots, A_n$, "not B". Then the following combined text is a proof of A from $A_1, \ldots, A_n$:

k	Statement	Rule ... with hypothesis ...
1	Assume B.	-
2	Proof1	given
3	A	given
4	Thus	-
5	B implies A	*implication*
6	Assume not B	-
7	Proof2	given
8	A	given
9	Thus	-
10	not B implies A	*implication*
11	A	*case distinction* with 5, 10

This contradicts the initial assumption. *qed*(1)

(2) Assume that A is not provable from

$$A_1, \ldots, A_n, \text{“not for all } x \text{ holds } B(x)\text{”}$$

and that y is a variable which does not occur in $A_1, \ldots, A_n$, "not for all x holds $B(x)$". Then A is not provable from

$$A_1, \ldots, A_n, \text{“not for all } x \text{ holds } B(x)\text{”}, \text{“not } B(y)\text{”}.$$

Proof. Assume not and assume that the text Proof1,A is a proof of A from

$$A_1, \ldots, A_n, \text{“not for all } x \text{ holds } B(x)\text{”}, \text{“not } B(y)\text{”}.$$

Then the following combined text is a proof of A from $A_1, \ldots, A_n$, "not for all x holds $B(x)$":

k	Statement	Rule ... with hypothesis ...
1	Assume $B(y)$.	-
2	For all x holds $B(x)$.	*generalization* with 1
3	Contradiction.	*contradiction* with 2 and "not for all x holds $B(x)$"
4	A	*ex falsum libenter* with 3
5	Thus	-
6	$B(y)$ implies A.	*implication*
7	Assume not $B(y)$.	-
8	Proof1	given
9	A.	given
10	Thus	-
11	not $B(y)$ implies A.	*implication*
12	A.	*case distinction* with 6, 11

This contradicts the initial assumption. *qed*(2)

The collection of formulas is countable since every formula is basically a finite sequence of symbols taken from a countable or even finite alphabet. Let F_1, F_2, ... be an enumeration of all formulas.

Define a sequence A_1, A_2, ... of formulas by recursion. At odd stages $1, 3, \ldots$, we ensure that every formula is decided by the sequence; at even stages $2, 4, 6, \ldots$, we care about quantifiers. So let $2m-1$ be an odd number, where $m \geqslant 1$, and assume that $A_1, \ldots, A_{2m-2}$ are defined. We shall define A_{2m-1} and A_{2m}.

If A is not provable from $\{A_1, \ldots, A_{2m-2}, F_m\}$, set $A_{2m-1} = F_m$; otherwise set $A_{2m-1} =$ "not F_m". Thereafter, if A_{2m-1} is of the form "not for all x holds $B(x)$", choose a variable y which does not occur in $\{A_0, \ldots, A_{2m-1}\}$ and set $A_{2m} =$ "not $B(y)$"; otherwise set $A_{2m} = A_{2m-1}$.

We prove several claims about the set of formulas $\mathcal{H} = \{A_1, A_2, \ldots\}$ which will correspond to the fact that the sequence describes a certain structure $\mathcal{S}$ as desired.

(3) For all n, A is not provable from $\{A_1, \ldots, A_n\}$.

Proof. This follows immediately from the construction and properties (1) and (2). *qed*(3)

(4) For every formula B, "not B" $\in \mathcal{H}$ iff $B \notin \mathcal{H}$.

Proof. Consider $B = F_m$. Assume that "not B" $\in \mathcal{H}$. Assume for a contradiction that also $B \in \mathcal{H}$. Choose a natural number n that B, "not B" $\in \{A_1, \ldots, A_n\}$. Then A is immediately provable from $\{A_1, \ldots, A_n\}$ by the rules of *contradiction* and *ex falsum libenter*. But this contradicts (3).

Conversely assume that "not B" $\notin \mathcal{H}$. Then by construction of $\mathcal{H}$, $A_{2m-1} = F_m = B \in \mathcal{H}$. *qed*(4)

(5) Let B be provable from $\mathcal{H}$. Then $B \in \mathcal{H}$.

Proof. Let Proof1, B be a proof of B from $\mathcal{H}$. Assume $B \notin \mathcal{H}$. By (4), "not B" $\in \mathcal{H}$. Then the following text is a proof of A from $\mathcal{H}$:

k	Statement	Rule ... with hypothesis ...
1	Proof1	given
2	B.	given
3	not B.	copying "not B" out of $\mathcal{H}$
4	Contradiction.	*contradiction* with 2, 3
5	A.	*ex falsum libenter* with 4

This contradicts (3). *qed*(5).

(6) "not A" $\in \mathcal{H}$.

Proof. By (3), $A \notin \mathcal{H}$. The claim follows by (4). *qed*(6)

(7) "contradiction" $\notin \mathcal{H}$.

Proof. If "contradiction" $\in \mathcal{H}$, say "contradiction" $= A_n$ then A is provable from $\{A_1, \ldots, A_n\}$ by the *ex falsum libenter* rule, which contradicts (3). *qed*(7)

(8) For all formulas B and C, we have "B implies C" $\in \mathcal{H}$ iff ($B \in \mathcal{H}$ implies $C \in \mathcal{H}$).

Proof. Assume "B implies C" $\in \mathcal{H}$ and assume that $B \in \mathcal{H}$. Then C is provable from $\mathcal{H}$. By (5), $C \in \mathcal{H}$, and thus $B \in \mathcal{H}$ implies $C \in \mathcal{H}$.

Conversely assume that "B implies C" $\notin \mathcal{H}$. By (4), "not (B implies C)" $\in \mathcal{H}$. From "not (B implies C)" one can prove B and "not C". By (5), $B \in \mathcal{H}$ and $C \notin \mathcal{H}$. Hence $B \in \mathcal{H}$ does not imply $C \in \mathcal{H}$. *qed*(8)

(9) For all formulas $B(x)$ we have: "for all x holds $B(x)$" $\in \mathcal{H}$ iff for all variables y holds $B(y) \in \mathcal{H}$.

Proof. Assume that "for all x holds $B(x)$" $\in \mathcal{H}$. Then for all variables y, $B(y)$ is provable from $\mathcal{H}$ by the rule of *instantiation*. By (5), $B(y) \in \mathcal{H}$.

Conversely assume that "for all x holds $B(x)$" $\notin \mathcal{H}$. By (4), "not for all x holds $B(x)$" $\in \mathcal{H}$. Choose an index m such that $F_m =$ "not for all x holds $B(x)$". By construction, $A_{2m-1} = F_m$ and $A_{2m} =$ "not $B(y)$" $\in \mathcal{H}$ for some variable y. By (4), $B(y) \notin \mathcal{H}$. *qed*(8)

Now define the structure $\mathcal{S} = (S, \ldots)$ as follows. Let S be the set of all variables occuring in $A_0, A_1, \ldots$. For every n-ary relation symbol R occuring in $A_0, A_1, \ldots$ define an n-ary relation R^S on S by

$$R^S(x_1, \ldots, x_n) \text{ iff } R(x_1, \ldots, x_n) \in \mathcal{H}.$$

(10) Let F be a formula. Then F holds in $\mathcal{S}$ iff $F \in \mathcal{H}$.
Proof. We prove the claim by induction on the length of F as a sequence of symbols. So assume that the claim holds for all shorter F'.
Case 1. F is a relational formula of the form $F = R(x_1, \ldots, x_n)$.
Then by definition of the structure $\mathcal{S}$, F holds in $\mathcal{S}$ iff $R^S(x_1, \ldots, x_n)$ iff $R(x_1, \ldots, x_n) \in \mathcal{H}$.
Case 2. $F =$ "contradiction".
Then F does *not* hold in $\mathcal{S}$. Also, by (7), $F \notin \mathcal{H}$.
Case 3. $F =$ "not B".
Then by the inductive assumption and (4),
F holds in $\mathcal{S}$
iff B does not hold in $\mathcal{S}$
iff $B \notin \mathcal{H}$
iff "not B" $\in \mathcal{H}$.
Case 4. $F =$ "B implies C".
Then by the inductive assumption and (8),
"B implies C" holds in $\mathcal{S}$
iff B holds in $\mathcal{S}$ implies C holds in $\mathcal{S}$
iff $B \in \mathcal{H}$ implies $C \in \mathcal{H}$
iff "B implies C" $\in \mathcal{H}$.
Case 5. $F =$ "for all x holds $B(x)$".
Then by the inductive assumption and (9),
"for all x holds $B(x)$" holds in $\mathcal{S}$
iff for all variables $y \in S$, $B(y)$ holds in $\mathcal{S}$
iff for all variables $y \in S$, $B(y) \in \mathcal{H}$
iff "for all x holds $B(x)$" $\in \mathcal{H}$. *qed*(9)

By (10) and (6), the initial formula A does not hold in $\mathcal{S}$. Thus A is not universally valid. □

4 Normal forms

There are many motivations to transform formulas into equivalent *normal forms*. The motivation here will be that normal forms are important for *automated theorem proving* and for *logic programming*.

We are particularly interested in transforming formulas ψ into formulas ψ' such that ψ is consistent iff ψ' is consistent. This relates to provability as follows: $\Phi \vdash \varphi$ iff $\Phi \cup \{\neg\varphi\}$ is not satisfiable/inconsistent. So a check for provability can be based on inconsistency checks.

Work in some fixed language S.

Definition 17.

a) *An S-formula is a* literal *if it is atomic or the negation of an atomic formula.*

b) *Define the* dual *of the literal L as*

$$\bar{L} = \begin{cases} \neg L, \text{ if } L \text{ is an atomic formula;} \\ K, \text{ if } L \text{ is of the form } \neg K. \end{cases}$$

c) *A formula φ is in* disjunctive normal form *if it is of the form*

$$\varphi = \bigvee_{i<m} \Big(\bigwedge_{j<n_i} L_{ij} \Big)$$

where each L_{ij} is a literal.

d) A formula φ is in conjunctive normal form *if it is of the form*

$$\varphi = \bigwedge_{i<m} \left(\bigvee_{j<n_i} L_{ij} \right)$$

where each L_{ij} is a literal. Sometimes a disjunctive normal form is also written in set notion as

$$\varphi = \{\{L_{00}, \ldots, L_{0n_0-1}\}, \ldots, \{L_{m-1,0}, \ldots, L_{m-1,n_{m-1}}\}\}.$$

Theorem 18. *Let φ be a formula without quantifiers. Then φ is equivalent to some φ' in disjunctive normal form and to some φ'' in conjunctive normal form.*

Proof. By induction on the complexity of φ. Clear for φ atomic. The $\neg$ step follows from the de Morgan laws:

$$\begin{aligned} \neg \bigvee_{i<m} \left(\bigwedge_{j<n_i} L_{ij} \right) &\leftrightarrow \bigwedge_{i<m} \neg \left(\bigwedge_{j<n_i} L_{ij} \right) \\ &\leftrightarrow \bigwedge_{i<m} \left(\bigvee_{j<n_i} \neg L_{ij} \right). \end{aligned}$$

The $\wedge$-step is clear for conjunctive normal forms. For disjunctive normal forms the associativity rules yield

$$\bigvee_{i<m} \left(\bigwedge_{j<n_i} L_{ij} \right) \wedge \bigvee_{i<m'} \left(\bigwedge_{j<n'_i} L'_{ij} \right) \leftrightarrow \bigvee_{i<m, i'<m'} \left(\bigwedge_{j<n_i} L_{ij} \wedge \bigwedge_{j<n'_i} L'_{ij} \right)$$

which is also in conjunctive normal form. □

Definition 19. *A formula φ is in* prenex normal form *if it is of the form*

$$\varphi = Q_0 x_0 \, Q_1 x_1 \ldots Q_{m-1} x_{m-1} \, \psi$$

where each Q_i is either the quantifier $\forall$ or $\exists$ and ψ is quantifier-free. Then the quantifier string $Q_0 x_0 \, Q_1 x_1 \ldots Q_{m-1} x_{m-1}$ is called the prefix *of φ and the formula ψ is the* matrix *of φ.*

Theorem 20. *Let φ be a formula. Then φ is equivalent to a formula φ' in prenex normal form.*

Proof. By induction on the complexity of φ. Clear for atomic formulas. If

$$\varphi \leftrightarrow Q_0 x_0 \, Q_1 x_1 \ldots Q_{m-1} x_{m-1} \, \psi$$

with quantifier-free ψ then by the de Morgan laws for quantifiers

$$\neg\varphi \leftrightarrow \bar{Q}_0 x_0 \, \bar{Q}_1 x_1 \ldots \bar{Q}_{m-1} x_{m-1} \, \neg\psi$$

where the dual quantifier $\bar{Q}$ is defined by $\bar{\exists} = \forall$ and $\bar{\forall} = \exists$.

For the $\wedge$-operation consider another formula

$$\varphi' \leftrightarrow Q'_0 x'_0 \, Q'_1 x'_1 \ldots Q'_{m'-1} x'_{m'-1} \, \psi'.$$

We may assume that the sets of bound variables of the prenex normal forms are disjoint. Then

$$\varphi \wedge \varphi' \leftrightarrow Q_0 x_0 \, Q_1 x_1 \ldots Q_{m-1} x_{m-1} Q'_0 x'_0 \, Q'_1 x'_1 \ldots Q'_{m'-1} x'_{m'-1} \, (\psi \wedge \psi').$$

(semantic argument). □

Definition 21. *A formula φ is* universal *if it is of the form*

$$\varphi = \forall x_0 \, \forall x_1 \ldots \forall x_{m-1} \, \psi$$

where ψ is quantifier-free. A formula φ is existential *if it is of the form*

$$\varphi = \exists x_0 \, \exists x_1 \ldots \exists x_{m-1} \, \psi$$

where ψ is quantifier-free.

We show a quasi-equivalence with respect to universal (and existential) formulas which is not a logical equivalence but concerns the consistency or satisfiability of formulas.

Theorem 22. *Let φ be an S-formula. Then there is a canonical extension S^* of the language S and a canonical universal $\varphi^* \in L^{S^*}$ such that*

$$\varphi \text{ is consistent iff } \varphi^* \text{ is consistent.}$$

The formula φ^* is called the SKOLEM normal form *of φ.*

Proof. By a previous theorem we may assume that φ is in prenex normal form. We prove the theorem by induction on the number of existential quantifiers in φ. If φ does not contain an existential quantifier we are done. Otherwise let

$$\varphi = \forall x_1 ... \forall x_m \exists y \psi$$

where $m < \omega$ may also be 0. Introduce a new m-ary function symbol f (or a constant symbol in case $m = 0$) and let

$$\varphi' = \forall x_1 ... \forall x_m \, \psi \frac{f x_1 ... x_m}{y}.$$

By induction it suffices to show that φ is consistent iff φ' is consistent.
(1) $\varphi' \rightarrow \varphi$.
Proof. Assume φ'. Consider $x_1, ..., x_m$. Then $\psi \frac{f x_1 ... x_m}{y}$. Then $\exists y \psi$. Thus $\forall x_1 ... \forall x_m \exists y \psi$. *qed*(1)
(2) If φ' is consistent then φ is consistent.
Proof. If $\varphi \rightarrow \bot$ then by (1) $\varphi' \rightarrow \bot$. *qed*(2)
(3) If φ is consistent then φ' is consistent.
Proof. Let φ be consistent and let $\mathcal{M} = (M, ...) \vDash \varphi$. Then

$$\forall a_1 \in M ... \forall a_m \in M \exists b \in M \mathcal{M} \frac{\vec{a} \; b}{\vec{x} \; y} \vDash \psi.$$

Using the axiom of choice there is a function $h: M^m \rightarrow M$ such that

$$\forall a_1 \in M ... \forall a_m \in M \mathcal{M} \frac{\vec{a} \; h(a_1, ..., a_m)}{\vec{x} \; y} \vDash \psi.$$

Expand the structure $\mathcal{M}$ to $\mathcal{M}' = \mathcal{M} \cup \{(f, h)\}$ where the symbol f is interpreted by the function h. Then $h(a_1, ..., a_m) = \mathcal{M}' \frac{\vec{a}}{\vec{x}} (f x_1 ... x_m)$ and

$$\forall a_1 \in M ... \forall a_m \in M \mathcal{M}' \frac{\vec{a} \; \mathcal{M}' \frac{\vec{a}}{\vec{x}} (f x_1 ... x_m)}{\vec{x} \; y} = \mathcal{M}' \frac{\vec{a}}{\vec{x}} \frac{\mathcal{M}' \frac{\vec{a}}{\vec{x}} (f x_1 ... x_m)}{y} \vDash \psi.$$

By the substitution theorem this is equivalent to

$$\forall a_1 \in M ... \forall a_m \in M \, \mathcal{M}' \frac{\vec{a}}{\vec{x}} \vDash \psi \frac{f x_1 ... x_m}{y}.$$

Hence

$$\mathcal{M}' \vDash \forall x_1 ... \forall x_m \, \psi \frac{f x_1 ... x_m}{y} = \varphi'.$$

Thus φ' is consistent. □

5 HERBRAND's theorem

By the previous chapter we can reduce the question whether a given finite set of formulas is inconsistent to the question whether some universal formula is inconsistent. By the following theorem this can be answered rather concretely.

Theorem 23. *Let S be a language which contains at least one constant symbol. Let*

$$\varphi = \forall x_0 \forall x_1 ... \forall x_{m-1} \, \psi$$

be a universal S-sentence with quantifier-free matrix ψ. Then φ is inconsistent if there are variable-free S-terms ("constant terms")

$$t_0^0, \ldots, t_{m-1}^0, \ldots, t_0^{N-1}, \ldots, t_{m-1}^{N-1}$$

such that

$$\varphi' = \bigwedge_{i<N} \psi\frac{t_0^i, \ldots, t_{m-1}^i}{x_0, \ldots, x_{m-1}} = \psi\frac{t_0^0, \ldots, t_{m-1}^0}{x_0, \ldots, x_{m-1}} \wedge \ldots \wedge \psi\frac{t_0^{N-1}, \ldots, t_{m-1}^{N-1}}{x_0, \ldots, x_{m-1}}$$

is inconsistent.

Proof. All sentences φ', for various choices of constant terms, are logical consequences of φ. So φ is consistent, all φ' are consistent.

Conversely assume that all φ' are consistent. Then by the compactness theorem

$$\Phi = \{\psi\frac{t_0, \ldots, t_{m-1}}{x_0, \ldots, x_{m-1}} \,|\, t_0, \ldots, t_{m-1} \text{ are constant } S\text{-terms}\}$$

is consistent. Let $\mathcal{M} = (M, \ldots) \vDash \Phi$. Let

$$H = \{\mathcal{M}(t) \,|\, t \text{ is a constant } S\text{-term}\}.$$

Then $H \neq \emptyset$ since S contains a constant symbol. By definition, H is closed under the functions of $\mathcal{M}$. So we let $\mathcal{H} = (H, \ldots) \subseteq \mathcal{M}$ be the substructure of $\mathcal{M}$ with domain H.

(1) $\mathcal{H} \vDash \varphi$.

Proof. Let $\mathcal{M}(t_0), \ldots, \mathcal{M}(t_{m-1}) \in H$ where $t_0, \ldots, t_{m-1}$ are constant S-terms. Then $\psi\frac{t_0, \ldots, t_{m-1}}{x_0, \ldots, x_{m-1}} \in \Phi$, $\mathcal{M} \vDash \psi\frac{t_0, \ldots, t_{m-1}}{x_0, \ldots, x_{m-1}}$, and by the substitution theorem

$$\mathcal{M}\frac{\mathcal{M}(t_0), \ldots, \mathcal{M}(t_{m-1})}{x_0, \ldots, x_{m-1}} \vDash \psi.$$

Since ψ is quantifier-free this transfers to $\mathcal{H}$:

$$\mathcal{H}\frac{\mathcal{M}(t_0), \ldots, \mathcal{M}(t_{m-1})}{x_0, \ldots, x_{m-1}} \vDash \psi.$$

Thus

$$\mathcal{H} \vDash \forall x_0 \forall x_1 \ldots \forall x_{m-1} \psi = \varphi.$$

qed(1)

Thus φ is consistent. □

In case that the formula ψ does not contain the equality sign $\equiv$ checking for inconsistency of

$$\varphi' = \bigwedge_{i<N} \psi\frac{t_0^i, \ldots, t_{m-1}^i}{x_0, \ldots, x_{m-1}} = \psi\frac{t_0^0, \ldots, t_{m-1}^0}{x_0, \ldots, x_{m-1}} \wedge \ldots \wedge \psi\frac{t_0^{N-1}, \ldots, t_{m-1}^{N-1}}{x_0, \ldots, x_{m-1}}$$

is in principle a straightforward finitary problem. φ' contains finitely many constant S-terms. φ' is consistent iff the relation symbols can be interpreted on appropriate tuples of the occuring S-terms to make φ' true. There are finitely many possibilities for the assignments of truth values of relations. This leads to the following (theoretical) algorithm for automatic proving for formulas without $\equiv$:

Let $\Omega \subseteq L^S$ be finite and $\chi \in L^S$. To check whether $\Omega \vdash \chi$:

1. Form $\Phi = \Omega \cup \{\neg\chi\}$ and let $\varphi = \forall(\bigwedge \Phi)$ be the universal closure of $\bigwedge \Phi$. Then $\Omega \vdash \chi$ iff $\Phi = \Omega \cup \{\neg\chi\}$ is inconsistent iff $(\bigwedge \Phi) \vdash \bot$ iff $\forall(\bigwedge \Phi) \vdash \bot$.
2. Transform φ into universal form $\varphi^{\forall} = \forall x_0 \forall x_1 \ldots \forall x_{m-1} \psi$ (SKOLEMization).
3. Systematically search for constant S-terms

   $$t_0^0, \ldots, t_{m-1}^0, \ldots, t_0^{N-1}, \ldots, t_{m-1}^{N-1}$$

   such that

   $$\varphi' = \bigwedge_{i<N} \psi\frac{t_0^i, \ldots, t_{m-1}^i}{x_0, \ldots, x_{m-1}} = \psi\frac{t_0^0, \ldots, t_{m-1}^0}{x_0, \ldots, x_{m-1}} \wedge \ldots \wedge \psi\frac{t_0^{N-1}, \ldots, t_{m-1}^{N-1}}{x_0, \ldots, x_{m-1}}$$

is inconsistent.

4. If an inconsistent φ' is found, output "yes", otherwise carry on.

Obviously, if "yes" is output then $\Omega \vdash \chi$. This is the *correctness* of the algorithm. On the other hand, HERBRAND's theorem ensures that if $\Omega \vdash \chi$ then an appropriate φ' will be found, and "yes" will be output, i.e., the algorithm is *complete*.

Let us assume from now on, that the formulas considered do not contain the symbol $\equiv$.

We shall see that the search for those S-terms and the inconsistency-check can be further systematized. We can assume that the quantifier-free formula ψ is in conjunctive normal form, i.e., a conjunction of clauses $\psi = c_0 \wedge c_1 \wedge \ldots \wedge c_{l-1}$. Then $\forall x_0 \forall x_1 \ldots \forall x_{m-1} \psi$ is inconsistent iff the set

$$\{c_i \frac{t_0, \ldots, t_{m-1}}{x_0, \ldots, x_{m-1}} | t_0, \ldots, t_{m-1} \text{ are constant } S\text{-terms}\}$$

is inconsistent.

The method of *resolution* gives an efficient method for showing the inconsistency of sets of clauses.

Definition 24. *Let $c^+ = \{K_0, \ldots, K_{k-1}\}$ and $c^- = \{L_0, \ldots, L_{l-1}\}$ be clauses with literals K_i and L_j. Note that $\{K_0, \ldots, K_{k-1}\}$ stands for the disjunction $K_0 \vee \ldots \vee K_{k-1}$. Assume that K_0 and L_0 are* dual*, i.e., $L_0 = \neg K_0$. Then the disjunction*

$$\{K_1, \ldots, K_{k-1}\} \cup \{L_1, \ldots, L_{l-1}\}$$

is a resolution of c^+ and c^-.

Resolution is related to the application of modus ponens: $\varphi \rightarrow \psi$ and φ correspond to the clauses $\{\neg\varphi, \psi\}$ and $\{\varphi\}$. $\{\psi\}$ is a resolution of $\{\neg\varphi, \psi\}$ and $\{\varphi\}$.

Theorem 25. *Let C be a set of clauses and let c be a resolution of two clauses $c^+, c^- \in C$. Then if $C \cup \{c\}$ is inconsistent then C is inconsistent.*

Proof. Let $c^+ = \{K_0, \ldots, K_{k-1}\}$, $c^- = \{\neg K_0, L_1 \ldots, L_{l-1}\}$, and $c = \{K_1, \ldots, K_{k-1}\} \cup \{L_1, \ldots, L_{l-1}\}$. Assume that $\mathcal{M} \vDash C$ is a model of C.
Case 1. $\mathcal{M} \vDash K_0$. Then $\mathcal{M} \vDash c^-$, $\mathcal{M} \vDash \{L_1 \ldots, L_{l-1}\}$, and

$$\mathcal{M} \vDash \{K_1, \ldots, K_{k-1}\} \cup \{L_1, \ldots, L_{l-1}\} = c.$$

Case 2. $\mathcal{M} \vDash \neg K_0$. Then $\mathcal{M} \vDash c^+$, $\mathcal{M} \vDash \{K_1 \ldots, K_{k-1}\}$, and

$$\mathcal{M} \vDash \{K_1, \ldots, K_{k-1}\} \cup \{L_1, \ldots, L_{l-1}\} = c.$$

Thus $\mathcal{M} \vDash C \cup \{c\}$. □

Theorem 26. *Let C be a set of clauses closed under resolution. Then C is inconsistent iff $\emptyset \in C$. Note that the empty clause $\{\} \leftrightarrow \bot$.*

Proof. If $\emptyset \in C$ then C is clearly inconsistent.

Conversely assume that C is inconsistent. By the compactness theorem there is a finite set of atomic formulas $\{\varphi_0, \ldots, \varphi_{n-1}\}$ such that

$$C' = \{c \in C | \text{for every literal } L \text{ in } c \text{ there exists } i < n \text{ such that } L = \varphi_i \text{ or } L = \neg\varphi_i\},$$

the restriction of C to $\{\varphi_0, \ldots, \varphi_{n-1}\}$ is inconsistent. Assume that the number n of atomic formulas with that property is chosen minimally.
Case 1. $n = 0$. Since the empty set of clauses is consistent, $C' \neq \emptyset$. On the other hand the only clause built from no atomic formulas is the clause $\{\} = \emptyset$. Thus $\emptyset \in C' \subseteq C$.
Case 2. $n = m + 1 > 0$. Assume for a contradiction that $\emptyset \notin C$. Let

$$C^+ = \{c \in C' | \neg\varphi_0 \notin c\}, C^- = \{c \in C' | \varphi_0 \notin c\}$$

and

$$C_0^+ = \{c \setminus \{\varphi_0\} | c \in C^+\},\ C_0^- = \{c \setminus \{\neg\varphi_0\} | c \in C^-\}.$$

(1) C_0^+ and C_0^- are closed under resolution.
Proof. Let d'' be a resolution of $d, d' \in C_0^+$. Let $d = c \setminus \{\varphi_0\}$ and $d' = c' \setminus \{\varphi_0\}$ with $c, c' \in C^+$. The resolution d'' was based on some atomic formula $\varphi_i \neq \varphi_0$. Then we can also resolve c, c' by the same atomic formula φ_i. Let c'' be that resolution of c, c'. Since C is closed under resolution, $c'' \in C$, $c'' \in C^+$, and $d'' = c'' \setminus \{\varphi_0\} \in C_0^+$. *qed*(1)
(2) $\emptyset \notin C_0^+$ or $\emptyset \notin C_0^-$.
Proof. If $\emptyset \in C_0^+$ and $\emptyset \in C_0^-$, and since $\emptyset \notin C$ we have $\{\varphi_0\} \in C^+$ and $\{\neg\varphi_0\} \in C^-$. But then the resolution $\emptyset$ of $\{\varphi_0\}$ and $\{\neg\varphi_0\}$ would be in C, contradiction. *qed*(2)
Case 1. $\emptyset \notin C_0^+$. By the minimality of n and by (1), C_0^+ is consistent. Let $\mathcal{M} \vDash C_0^+$. Let the atomic formula φ_0 be of the form $r t_0 \ldots t_{s-1}$ where r is an n-ary relation symbol and $t_0, \ldots, t_{s-1} \in T^S$. Since the formula $r t_0 \ldots t_{s-1}$ does not occur within C_0^+, we can modify the model $\mathcal{M}$ to a model $\mathcal{M}'$ by only modifying the interpretation $\mathcal{M}(r)$ exactly at $(\mathcal{M}(t_0), \ldots, \mathcal{M}(t_{s-1}))$. So let $\mathcal{M}'(\mathcal{M}(t_0), \ldots, \mathcal{M}(t_{s-1}))$ be *false*. Then $\mathcal{M}' \vDash \neg\varphi_0$. We show that $\mathcal{M}' \vDash C'$.

Let $c \in C'$. If $\neg\varphi_0 \in c$ then $\mathcal{M}' \vDash c$. So assume that $\neg\varphi_0 \notin c$. Then $c \in C^+$ and $c \setminus \{\varphi_0\} \in C_0^+$. Then $\mathcal{M} \vDash c \setminus \{\varphi_0\}$, $\mathcal{M}' \vDash c \setminus \{\varphi_0\}$, and $\mathcal{M}' \vDash c$. But then C' is consistent, contradiction.
Case 2. $\emptyset \notin C_0^-$. We can then proceed analogously to case 1, arranging that $\mathcal{M}'(\mathcal{M}(t_0), \ldots, \mathcal{M}(t_{s-1}))$ be *true*. So we get a contradiction again. $\square$

This means that the inconsistency check in the automatic proving algorithm can be carried out even more systematically: produce all relevant resolution instances until the empty clause is generated. Again we have correctness and completeness for the algorithm with resolution.

Example 27. Already with a view towards logical applications of `Prolog` let us consider a theory about the recursive definition of formulas. Let

$$\begin{array}{l} \mathrm{fm}(\mathrm{psi}) \\ \mathrm{fm}(\mathrm{phi}) \\ \forall X, Y (\mathrm{fm}(X) \wedge \mathrm{fm}(Y) \rightarrow \mathrm{fm}(\mathrm{and}(X, Y))) \end{array}$$

be a small axiom system concerning the formation of formulas; here "psi" and "phi" are constant symbols, "and" is a binary function symbol, and "formula" is a unary relation symbol. To show that $\psi \wedge (\psi \wedge \psi)$ is a formula one has to derive

$$\mathrm{fm}(\mathrm{and}(\mathrm{psi}, \mathrm{and}(\mathrm{psi}, \mathrm{psi})))$$

from the axioms. This is equivalent to showing that

$$\begin{array}{l} \mathrm{fm}(\mathrm{psi}) \\ \mathrm{fm}(\mathrm{phi}) \\ \forall X, Y (\mathrm{fm}(X) \wedge \mathrm{fm}(Y) \rightarrow \mathrm{fm}(\mathrm{and}(X, Y))) \\ \neg\mathrm{fm}(\mathrm{and}(\mathrm{psi}, \mathrm{and}(\mathrm{psi}, \mathrm{psi}))) \end{array}$$

is inconsistent. We can write the matrix of the conjunction of these formulas in conjunctive normal form as

$$C = \{\{\mathrm{fm}(\mathrm{psi})\}, \{\mathrm{fm}(\mathrm{psi})\}, \{\neg\mathrm{fm}(X), \neg\mathrm{fm}(Y), \mathrm{fm}(\mathrm{and}(X, Y))\}, \{\neg\mathrm{fm}(\mathrm{and}(\mathrm{psi}, \mathrm{and}(\mathrm{psi}, \mathrm{psi})))\}\}.$$

Obviously the universally quantified clause $\{\neg\mathrm{fm}(X), \neg\mathrm{fm}(Y), \mathrm{fm}(\mathrm{and}(X, Y))\}$ implies all its instantiations by constant terms. So we close the set C under such instantiations and under resolution. Deriving the empty clause $\{\}$ shows the desired inconsistency. We write the sequence of derived clauses in the format of a formal proof:

1	fm(psi)	assumption
2	fm(phi)	assumption
3	$\neg$fm(X), $\neg$fm(Y), fm(and(X, Y))	assumption
4	$\neg$fm(and(psi, and(psi, psi)))	assumption
5	$\neg$fm(psi), $\neg$fm(and(psi, psi)), fm(and(psi, and(psi, psi)))	instance of 3
6	$\neg$fm(psi), $\neg$fm(and(psi, psi))	resolution of 4, 5
7	$\neg$fm(and(psi, psi))	resolution of 1, 6
8	$\neg$fm(psi), fm(and(psi, psi))	instance of 3
9	$\neg$fm(psi)	resolution of 7, 8
10	{}	resolution of 1, 9

The choice of instances of the universal clause $\{\neg\mathrm{fm}(X),\ \neg\mathrm{fm}(Y),\ \mathrm{fm}(\mathrm{and}(X,\ Y))\}$ was directed by the desire to resolve certain clauses along the derivation. It is possible to find "fitting" instances by the method of *unification* which will be explained in the next chapter.