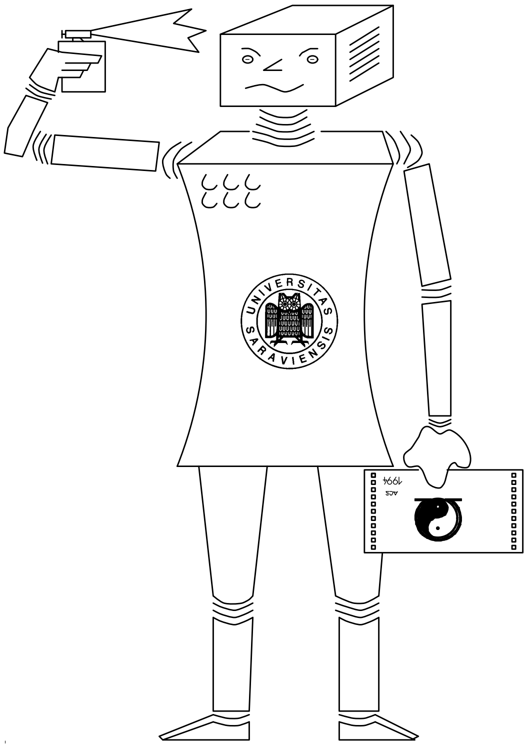
UNIVERSITÄT DES SAARLANDES
FACHBEREICH INFORMATIK
D-66041 SAARBRÜCKEN
GERMANY
WWW: http://js-sfbsun.cs.uni-sb.de/pub/www/



On the Representation of Mathematical Knowledge in Frames and its Consistency

Manfred Kerber

Published as: In Michel De Glas and Dov Gabbay, eds., *WOCFAI '91 – Proceedings of the First World Conference on the Fundamentals of Artificial Intelligence*, Paris, France, p.293–301, Angkor, Paris, France, 1991.

On the Representation of Mathematical Knowledge in Frames and its Consistency

Manfred Kerber
Fachbereich Informatik, Universität Kaiserslautern
D-6750 Kaiserslautern, Germany

Das wird nächstens schon besser gehen,
Wenn Ihr lernt alles reduzieren
Und gehörig klassifizieren.
J.W.Goethe, Faust I

Abstract: We show how to build up mathematical knowledge bases using frames. We distinguish three different types of knowledge: axioms, definitions (for introducing concepts like "set" or "group") and theorems (for relating the concepts). The consistency of such knowledge bases cannot be proved in general, but we can restrict the possibilities where inconsistencies may be imported to very few cases, namely to the occurrence of axioms. Definitions and theorems should not lead to any inconsistencies because definitions form conservative extensions and theorems are proved to be consequences.

Key words: frames, consistency, mathematical concept, conservative extension

1 Introduction

Many different forms of representing knowledge have been developed in artificial intelligence. Frames, first introduced by Marvin Minsky in [14], are very popular, because of the clarity and the expressive power in principle. Another advantage is the possibility to fix the slots and the types of the admissible slot fillers. Of course the *primitives* – in accordance to the ideas of Ronald J. Brachman [4] – have to be found. The power of such a representational approach can be seen in the CYC project where different kinds of knowledge are represented in a frame-like manner [13].

We shall use the frame approach of knowledge representation and an appropriate object language to represent conceptual mathematical knowledge like that of a mathematical dictionary. Concepts (like "group" or "field") are introduced as axioms or definitions and related by theorems. Our representation is an extension of standard logic. The knowledge about concepts consists of definitions, equivalent formulations and consequences, examples and counter-examples; the relations between *different* concepts are described by theorems. Each axiom, definition, and theorem is represented in a frame.

The language in which the mathematical factual knowledge is expressed plays a major rôle. Because first-order predicate logic is too limited to conveniently express mathematical knowledge we use higher order logic with sorts. We give no formal syntax here, but introduce it only in so far as it is necessary for the purpose of showing how such a knowledge base can be built up. The details can be found in [11]. How the knowledge can be translated into first order logic (in order to use first order theorem provers) can be found in [10].

When we represent mathematical concepts we have to show what a concept is and how to represent it. To this end, we take some introductory examples from mathematical textbooks. In [9] the conceptual knowledge of elementary axiomatic set theory [7] and the first part of [2] is represented.

When proving theorems by an automated theorem prover (for instance, a resolution based one), one has to give to the

Definition:	**associative**		property
parameters:	C	$:(\iota \rightarrow o)$	
	f	$:(C \times C \rightarrow C)$	
definition:	$\forall x, y, z : C \;\; f(f(x,y),z) = f(x, f(y,z))$		
context:	Basic Algebraic Definitions		

Figure 1: Definition of "associative"

prover the definitions, axioms, and theorems which are necessary for the proof. One can enter them for each proof anew, but aside the fact that this is very boring, each time again one can write something faulty. By a knowledge base we can avoid that we use two different definitions of the same concept, which seem to be the same but are not (*ignoratio elenchi*). If we have made an error we can repair it once for ever after detecting it. On the other hand it is very important to have a *consistent* knowledge base, because otherwise anything can be derived of it. Because of KURT GÖDEL'S second incompleteness theorem [6], the consistency of sufficiently expressive systems cannot be proved without using even stronger methods. But we can try to restrict the importation of contradictions to a few cases: the notion of axioms and these should be used relatively rarely. Of course such an approach cannot guarantee that the definitions of concepts are *correct*, that is, in accordance with the general use of them. We can define the concept "group" as something quite different from the general use, but because we cannot import contradictions by a definition other parts of the knowledge base not using this concept cannot be concerned of such a non-standard definition. (The importance of this fact for automated theorem proving is already noted in [3, p.13].)

Here we develop the ideas for a mathematical knowledge base, but the results are transferable to any one based on logic.

2 Examples

In this section we give an example, namely the concept "group". In order to define this concept, we must define the concept "associative". This can be done immediately in a sorted higher order logic $\mathcal{L}$ by

$$\forall C : (\iota \rightarrow o) \;\; \forall f : (C \times C \rightarrow C)$$
$$associative(C, f) \iff$$
$$\forall x, y, z : C \;\; f(f(x,y),z) = f(x, f(y,z)).$$

$(\iota \rightarrow o)$ denotes the type of all unary object predicates. We adopt the notion of types from ALONZO CHURCH [5]: ι is the type of individuals and o the type of truth values. The type itself is also the top sort of the corresponding type. We use sorts as abbreviations for corresponding predicates, that means the formula $\forall n : nat \;\; P(n)$ is an abbreviation for $\forall n : \iota \;\; nat(n) \Longrightarrow P(n)$. Analogously we abbreviate by $\forall f : (C \times C \rightarrow C) \;\; \varphi(f)$ the formula $\forall f : (\iota \times \iota \rightarrow \iota) \;\; function(f, C \times C, C) \Longrightarrow \varphi(f)$. We represent this knowledge in frames as above and we shall explain the newly introduced features of a frame, when they occur for the first time. With the different slots we get a description of what a concept is. Not only the predicate logic definition, but also other slots have to be filled in order to describe it in a detailed way. In the case of associativity we get the frame of Figure 1.

Here a certain definition is given. This is indicated by the keyword "**Definition:**" in the upper left corner. After the colon follows the name of the introduced concept. In this case it is "**associative**". Then we give the type of the concept. The entry "property" means that the whole concept models a property, similarly as predicate symbols model properties. Therefore a standard translation of the definition of a "property concept" into predicate logic can be done by a predicate symbol, but also other trans-

Axiom: **set**	
axioms:	$\forall x\!:\!Set\ \ Class(x)$
	$\forall X,Y\!:\!Class\ \ X \in Y \Longrightarrow Set(X)$
	$\forall X,Y\!:\!Class\ \ (\forall u\!:\!Set\ \ u \in X \iff u \in Y) \Longrightarrow X = Y$
	$\forall x,y\!:\!Set\ \ \exists z\!:\!Set\ \ (\forall u\!:\!Set\ \ u \in z \iff (u = x \vee u = y))\ \ (z = \{x,y\})$
consequences:	$\forall x,y\!:\!Set\ \ y \in \{x,x\} \iff x = y$ <proof-set-cons-1>
signature_ext:	$Class \sqsubseteq (\iota \to o), Set \sqsubseteq Class, \in : (Set \times Class \to o), \{.,.\} : (Set \times Set \to Set)$
context:	Set theory

Figure 2: Some axioms of set-theory

Definition: **ex_left_neutral_element**			property
parameters:	C	$:(\iota \to o)$	
	f	$:(C \times C \to C)$	
(optional)	0	$:C$	(called *neutral_element*)
definition:	$\forall x\!:\!C\ \ f(0,x) = x$		
context:	Basic Algebraic Definitions		

Figure 3: Definition of left neutral

lations are possible. A "property concept" represents the relationship between its parameters. But such a relationship consists not only of its definition, but also of consequences, examples, and so on. The other type of a concept is that of a "mapping concept". In this case a new object is created, for instance the concept "ordered_pair" is of that type.

The argument of the binary property **associative** is given in the slot "parameters". The number of parameters corresponds here to the arity of the defining predicate symbol. In this case we have an analogy between our parameters and the parameters in a computer language. In the slot "parameters" the *formal* parameters and their types are written. When using the defined object elsewhere they are bound to the *actual* parameters.

In the slot "definition" we find a (higher order) predicate logical definition of the concept. The slot "context" is provided for structuring the concepts in knowledge units. Here a more detailed information may become necessary, as for example the information what may be used by another concept and what not.

As example for axioms we choose the first axioms of GÖDEL'S set axioms [7] (see Figure 2).

In the slot "axioms" the corresponding axioms are written. The slot "consequences" contains lemmata (including a pointer to a proof) that belong immediately to that concept. In the slot "signature_ext" we state the axiomatized constants. In this case that are the predicates $Class$, Set, $\in$, and the function $\{.,.\}$.

In the next concept (see Figure 3) we have an "optional" parameter.

There are two different assertions using the concept $ex\_left\_neutral\_element$. "$ex\_left\_neutral\_element(C,f)$" means that there exists an element (let us call it "n") so that we have the formula $\forall x : C \quad f(n,x) = x$. But if we say "$ex\_left\_neutral\_element(C,f,0)$", we want to express that there exists an element (say "n") so that $\forall x\!:\!C\ \ f(n,x) = x$ and that this element n is equal to 0. The two expressions are closely related, but

Definition:	**group**		property
parameters:	G	$:(\iota \to o)$	(called *carrier*)
	$+$	$:(G \times G \to G)$	(called *operation*)
(optional)	0	$:G$	(called *neutral_element*)
(optional)	$-$	$:(G \to G)$	(called *inverse*)
definition:	$TRUE$		
equivalences:	1) $associative(G,+) \wedge$		
	$ex\_left\_neutral\_element(G,+,0) \wedge$		
	$ex\_left\_inverse(G,+,0,-)$		<proof-group-equ-1>
	2) $associative(G,+) \wedge$		
	$ex\_right\_neutral\_element(G,+,0) \wedge$		
	$ex\_right\_inverse(G,+,0,-)$		<proof-group-equ-2>
examples:	1) $(\mathbb{Z},+,0,-)$	model Integers	<proof-group-ex-1>
	2) $(\mathbb{Q},+,0,-)$	model Rationals	<proof-group-ex-2>
	3) $(\mathbb{Q}\backslash\{0\},\cdot,1,^{-1})$	model Rationals	<proof-group-ex-3>
superconcepts:	1) $associative(G,+)$		
	2) $ex\_neutral\_element(G,+,0)$		
	3) $ex\_inverse(G,+,0,-)$		
context:	Basic Algebraic Definitions		

Figure 4: Definition of “group”

nevertheless different.

Together with the analogously definable concept “ex_left_inverse” it is now possible to introduce the concept “group” (see Figure 4).

The $TRUE$ in the definition slot means that the concept is fully defined by the conjunction of its superconcepts.

The slot “equivalences” contains logical expressions that are necessary and sufficient to define the introduced concept. They are logically equivalent to the definition of the concept. They belong immediately to that concept and can be used as an alternative definition of the concept. For example if one wants to prove that a certain object is a group, it is easier to use one of the two equivalent formulations, because then one has less to prove. The definition itself is preferable if it is used as a premis, because it is stronger. This rule could be formulated as a fact in a meta-system. Here we see an important difference between our epistemological term “property” and the logical term “predicate”: The conceptual representation makes it possible to make some assertions *about* the concepts (as for instance to use a certain variant of the definition in some situation). This would not be possible if we had mere predicates. Whether and how this can be used for actually guiding a theorem prover, will of course be the ultimate test. In the slot “examples” we can find a reference to a model of the corresponding concept. Of course the equivalent formulations as well as consequences, examples, and counter-examples have to be *proved* before they can be inserted in the frame.

By the entries of the slot “superconcepts” we get a transitive network of concepts with inheritance. Every concept inherits all definitions, consequences, equivalences, and counter-examples of its superconcepts.

In the definition and axiom frames we represent certain theorems which belong immediately to that concept. But the most part of all theorems cannot be attached to one single concept (introduced as axiom or definition). So we need another kind of frames to represent these theorems. We represent them in the third type of frames, the theorem frames. For an example see Figure 5.

Theorem:	Name	Theorem
theorem:	$\forall s:(\iota \to o)\ \neg\exists g:(\iota \to (\iota \to o))\ \forall f:(\iota \to o)\ \ f \subseteq s \Longrightarrow (\exists j:\iota\ \ s(j) \wedge g(j) = f)^*$	
status:	"proved"	
proof:	<proof-Cantor>	
context:	Set theory	

Figure 5: Cantor's theorem

3 The Frame Language

In this section we briefly describe the notion of our knowledge units, the frames. There are three different types: definition frames, axiom frames, and theorem frames.

1. Definition: A *definition frame* (written ϑ) is a list consisting of the following parts:

- "parameters" is of a list of variable_symbols with sorts and optionally selector names.
- "definition" is either an $\mathcal{L}$-formula or an $\mathcal{L}$-term corresponding to the type of the frame (property or mapping).
- "main_property" is an $\mathcal{L}$-formula.
- "consequences", "equivalences", and "preconditions" are (possibly empty) lists of $\mathcal{L}$-formulae.
- "examples" and "counter_examples" are lists of $\mathcal{L}$-structures.
- "superconcepts" is a list of atomic $\mathcal{L}$-formulae.
- "used_in" is a list of axioms, definitions, and theorems.
- "context" is a name of a theory for modularization.

The slot "consequences" contains theorems that follow from the concept. For instance if the union of sets is defined, we should write into this slot, that the union is associative, commutative, and idempotent. The slot "superconcepts" is used for inheritance. For instance "associative" is a superconcept of "group". So every consequence in the "associative"-frame is also a consequence for the "group"-frame. On the other hand "group" is consequently a sub-concept for "associative". So every example for a group is in particular an example for an associative structure. For the slot fillers there exist some further constraints: for example if the frame type is "mapping" then a slot "sort" must be filled.

The semantics of the frames can be given by translating all parts but the examples into the underlying logic.** Examples must be models of the concept and counter-examples must not. The signature of a frame ϑ is the set of all extra-logical symbols in the terms and formulae in ϑ.

Analogously one can define axiom frames and theorem frames. All frames have a "context" slot and can have a "consequences" slot.

2. Definition: A *frame-extension* ϑ' of a frame ϑ is a frame with same name, classification, parameter list, definition or list of axioms or theorem and proof, list of superconcepts, preconditions, and context; the lists of consequences, equivalences, examples, counter_examples, and used_in list of ϑ are sublists of the corresponding lists of ϑ'; if the slot main_property of ϑ

*In plain text, the theorem states that the power set of a set has greater cardinality than the set itself. This formulation is almost that of PETER B. ANDREWS [1, p.184].

**That is of course only for having the theoretical notion of semantics. If one translated all knowledge into the underlying logic one would loose in practice all structural advantages of the frame representation.

is not filled, then the main_property of ϑ' can contain a formula or can be unfilled too.

In the next section we discuss the requirements that a set of such frames forms a knowledge base and when such a knowledge base is consistent.

4 Building up a Knowledge Base

When we build up a knowledge base we want it to be contradictory free. Unfortunately that cannot be proved in general. This is a consequence of GÖDEL'S second incompleteness result [6]: For every contradictory free formula set that entails PEANO arithmetic it cannot be proved (within the system) that it is contradictory free. This means if we restrict the entries to those which have to be shown contradictory free, we prune the expressive power of the representation facilities to an extent that is not usable for mathematical purpose.

What we can do, is to isolate the cases where contradictions can be brought in. So we distinguish between "axioms", where every logical formula is writable and hence contradictions can be written in the knowledge base, between "definitions" which have to form conservative extensions of the preceding knowledge base and so cannot lead to contradictions, and "theorems", which have be proved.

For instance the *axioms* ZFC of set theory (ZERMELO-FRAENKEL with axiom of choice) or those of VON NEUMANN-GÖDEL-BERNAYS cannot be proved to be contradictory free. If we add them to the knowledge base we might import contradictions. On the other hand the *definition* of the concept "group" cannot lead to contradictions (if it is really a definition). This definition can be regarded as an abbreviation for the conjunction of the concepts "associative", "ex_neutral_element", and "ex_inverse". It is important to see that the main part of mathematical activities is not introducing new *axioms*, but defining new concepts and proving theorems about them. In [7] GÖDEL uses 18 axioms to axiomatize set theory; almost all mathematics can be built up on these axioms. According to our calculus we need further *logical* axioms as the comprehension axioms in order to have a complete calculus (with respect to HENKIN'S general models semantics [8]).

By strict bookkeeping what has been used for a proof of a theorem one can minimize the possibilities that the proof is based on a faulty assumption and hence the theorem cannot be taken for sure. For instance if for the proof no use is made of ZFC or any theorem that has been proved with the help of it, the correctness of the theorem does not depend on the contradiction freeness of ZFC. Only axioms can import contradictions. All other entries (at least theoretically) cannot. Theorems have to be proved. Definitions should form conservative extensions.

Now we define what a knowledge base is. In such a knowledge base there are the three different kinds of knowledge: axioms, definitions, and theorems.

3. Definition (Knowledge Base): A knowledge base Δ over $\mathcal{L}$ is defined inductively:

- as the empty knowledge base $\Delta_\emptyset$, or
- as $\mathtt{cons}(\vartheta, \Delta')$ with definition frame ϑ relative to the knowledge base Δ', or
- as $\mathtt{cons}(\vartheta, \Delta')$ with axiom frame ϑ relative to the knowledge base Δ', or
- as $\mathtt{cons}(\vartheta, \Delta')$ with theorem frame ϑ relative to the knowledge base Δ', or
- it is equal to a knowledge base Δ' for all but one entry and this entry is a frame-extension of the other. Formally $\Delta\backslash\vartheta = \Delta'\backslash\vartheta'$ and ϑ is a frame-extension of ϑ'.

In all cases (but the first) Δ is called an *immediate extension* of Δ'. The transitive closure of this relation is called *extension*. The *signature of a knowledge base* is the union of all signatures of the containing frames.

A frame ϑ is a *definition frame relatively to a knowledge base* Δ' if it is a definition frame, the defined concept name is not in the signature of Δ, let the "parameters" be $(x_1, \ldots, x_m)$, then the "definition" slot is filled by a formula $\varphi(x_1, \ldots, x_m)$, where at most the variables $x_1, \ldots, x_m$ occur free and all other symbols are in the signature of Δ'. The signature of $\Delta = \texttt{cons}(\vartheta, \Delta')$ is the signature of Δ' plus the conceptname of ϑ as m-ary constant symbol. By $\mathcal{L}(\Delta)$ we denote the logic with the signature of Δ. The logical translation of the proper definition part of a definition frame is

$$\forall x_1, \ldots, x_m \ \mathit{conceptname}(x_1, \ldots, x_m) \iff \varphi(x_1, \ldots, x_m).$$

A frame ϑ is an *axiom frame relatively to a knowledge base* Δ if it is an axiom frame, the symbols in "signature_ext" are not in the signature of Δ, and no other symbols than those of the signature of Δ and those of "signature_ext" occur in ϑ. The signature of $\texttt{cons}(\vartheta, \Delta)$ is the union of these.

Analogously one can define the theorem frame relatively to a knowledge base.

4. Remark: For the following we need the notion of derivability. On the logical level we will assume that we have a sound calculus by which we can form a derivation operator $\vdash$. For the following it is not of importance how this calculus is defined. We only require it to be sound. Especially in an extended calculus the reasoning with examples and counter-examples could be included. Analogously to the derivability in AM [12] we can extend the derivability by the following: Let Δ be a knowledge base, φ be a formula then $\Delta \Vdash \varphi \iff$

1. $\Delta \vdash \varphi$,
2. $\varphi = \exists x_1, \ldots, x_m \psi(x_1, \ldots, x_m)$ and there is a model entry in a frame $\vartheta \in \Delta$ for $\psi(x_1, \ldots, x_m)$, or
3. $\varphi = \exists x_1, \ldots, x_m \neg\psi(x_1, \ldots, x_m)$ and there is a counter-example entry in a frame $\vartheta \in \Delta$ for $\psi(x_1, \ldots, x_m)$.

This form of reasoning is really used in mathematics. Examples are important for humans to understand a concept, but normally they are also given in order to assure the existence of certain objects (like "group"). In first approximation one can neglect examples and counter-examples and replace $\Vdash$ by $\vdash$ in the following.

5. Example: If we have defined the concepts "group" and "abelian_group" and we have a verified entry for examples in the frame "abelian_group", namely that $(\mathbb{Z}, +, 0, -)$ is an abelian group, then we can conclude that there are abelian groups. In addition – if we have represented "abelian_group" via superconcept "group" – we can conclude that there are groups. If we have an example for groups (like the symmetric group over $\{1, 2, 3, 4, 5\}$) which is a counter-example for commutative groups, then we can conclude that there is a non-abelian group. Or if we define an exponentiation function $\uparrow$ over $\mathbb{N}$ and we claim that this function is non-commutative we can prove that by giving one example, for instance $2 \uparrow 3 \neq 3 \uparrow 2$.

6. Definition: An $\mathcal{L}$-formula φ is called a theorem of a knowledge base Δ, if the symbols occurring in φ are in the signature of Δ and $\Delta \Vdash \varphi$.

7. Definition: A knowledge base Δ is called *contradictory free* iff there is no formula φ so that $\Delta \Vdash \varphi$ and $\Delta \Vdash \neg\varphi$.

8. Definition: An extension Δ of a knowledge base Δ' is called *conservative* iff for all formulae φ holds: If φ is in $\mathcal{L}(\Delta')$ then $\Delta' \Vdash \varphi$ iff $\Delta \Vdash \varphi$

9. Remark: In particular by a conservative extension we cannot import any contradiction. If the knowledge base Δ' is contradictory-free and Δ is a conservative extension of Δ', then Δ is also contradictory-free, because otherwise we could deduce a formula φ and its negation $\neg\varphi$ and by this any formula in $\mathcal{L}(\Delta)$, and hence any in $\mathcal{L}(\Delta')$.

10. Remark: Now we expect the lemma that definitions as those of the definition frames form conservative extensions. For first order logic that is true; unfortunately not for higher order logic underlying a very weak semantics as will show the next example.

11. Example: Let Δ consist of following axioms:

- $a : \iota$, $b : \iota$, $R : (\iota \to o)$
- $\forall P : (\iota \times \iota \to o)\ \forall x, y : \iota$ $P(x, y) \iff P(y, x)$
- $R(a) \land \neg R(b)$

These axioms are contradictory free since we can give a weak model: $\mathcal{D}_\iota = \{1, 2\}$, $\mathcal{D}_{(\iota\times\iota\to o)}$ consists only of the binary relations that map everything to $\mathtt{T}$. $\mathcal{J}(a) = 1$, $\mathcal{J}(b) = 2$, $\mathcal{J}(R)(1) = \mathtt{T}$, $\mathcal{J}(R)(2) = \mathtt{F}$. This is of course no longer a model when we "define" a new binary predicate Q by $\forall x, y\ Q(x, y) :\iff R(x) \land \neg R(y)$ and add this to our knowledge base. We have $Q(a, b)$ since $R(a)$ and $\neg R(b)$. On the other hand by the commutativity axiom we get $Q(b, a)$, hence $\neg R(a)$ and $R(b)$. That is, now we have a contradiction in our knowledge base. Hence this higher order logic with weak semantics is not *definition conservative*. This cannot happen when we have all comprehension axioms in the knowledge base. That are the following formulae [1, p.156]:
For every formula φ of which the free variables are exactly the following different variables $x_1, \ldots, x_m, y_1, \ldots, y_k$ of type $\tau_1, \ldots, \tau_m, \sigma_1, \ldots, \sigma_k$ the following formula is a comprehension axiom:

$$\forall y_1 \ldots \forall y_k\ \exists u_{(\tau_1\times\cdots\times\tau_m\to o)}\ \forall x_1 \ldots \forall x_m\ (u(x_1, \ldots, x_m) \iff \varphi).$$

12. Lemma: Concept definitions form conservative extensions, if the language $\mathcal{L}$ is first order, or the comprehension axioms are included.

Proof: Let Δ' be a knowledge base and ϑ a definition frame with conceptname *name*. In order to build $\Delta = \mathtt{cons}(\vartheta, \Delta')$, logically we have simply to add the formula $\forall x_1, \ldots, x_m\ name(x_1, \ldots, x_m) \iff \varphi$ where φ is the formula in the definition slot of the frame. But this formula is an instance of a comprehension axiom with u called *name*. Hence if a formula φ is in $\mathcal{L}(\Delta')$, it is deducible form Δ iff it is deducible from Δ'. Consequently the extension of Δ' by ϑ is conservative. We do not need such axioms if the language is first order (since here strong and weak semantics coincide) or if in the higher order logic λ-expressions are allowed (since these expressions guarantee the existence of sufficiently many objects).
For definitions of the type "mapping" we can proceed analogously. ■

13. Lemma: The empty knowledge $\Delta_\emptyset$ is consistent.

Proof: Since for all formulae φ with $\Delta_\emptyset \models \varphi$, φ is a tautology and φ and $\neg\varphi$ cannot be tautologies at once, $\Delta_\emptyset$ must be consistent. ■

14. Theorem: If Δ is a consistent knowledge base containing the comprehension axioms* and ϑ is a concept definition with respect to Δ then $\mathtt{cons}(\vartheta, \Delta)$ is consistent.

Proof: This follows immediately of the fact that concept definitions form conservative extensions. ■

15. Theorem: If Δ is a consistent knowledge base and ϑ a theorem of Δ, then $\mathtt{cons}(\vartheta, \Delta)$ is consistent.

Proof: Because the deductive closures of Δ and $\mathtt{cons}(\vartheta, \Delta)$ are the same, the theorem holds trivially. ■

5 Open Problems, Outlook

We have shown how to build up a knowledge base so that the possibilities to import inconsistencies are minimized. The critical situation is that, where axioms are introduced: they must not import contradictions and must be rich enough. Then the consistency of the knowledge base is not endangered by definitions and theorems. Of course the definition facilities must be stronger than the proposed ones: implicit definitions, partial definitions, and inductive definitions are necessary. But this should not lead to a new situation in principle.

In order to use such a system it will be useful to have the possibility to withdraw definitions in order to change them. The question is how to do that with minimal restructuring efforts.

*As above: or the language is first order or λ-expressions are allowed.

Acknowledgement

I like to thank my colleague Axel Präcklein for many discussions and thorough reading of a draft and my supervisor Jörg Siekmann for his encouragement and for his advice that resulted in numerous improvements.

References

[1] Peter B. Andrews. *An Introduction to Mathematical Logic and Type Theory: To Truth through Proof.* Academic Press, Orlando, Florida, USA, 1986.

[2] Emil Artin. *Galois Theory*, volume 2 of *Notre Dame Mathematical Lectures.* University of Notre Dame Press, Notre Dame, London; second edition, 1942.

[3] Robert S. Boyer and J Strother Moore. *A Computational Logic.* Academic Press, New York, USA, 1979.

[4] Ronald J. Brachman. On the epistemological status of semantic networks. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, chapter 10, pages 191–215. Morgan Kaufmann, 1985, Los Altos, California, USA, 1979. also in: Associative Networks: Representation and Use of Knowledge by Computers, 3–50, N. V. Findler, Edt., New York, USA, Academic Press.

[5] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, **5**:56–68, 1940.

[6] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, **38**:173–198, 1931.

[7] Kurt Gödel. *The Consistency of the Axiom of Choice and of the Generalized Continuum-Hypothesis with the Axioms of Set Theory*, volume 3 of *Annals of Mathematics Studies.* Princeton University Press, Princeton, New Jersey; eighth printing 1970, 1940.

[8] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, **15**:81–91, 1950.

[9] Manfred Kerber. A frame based approach to representing mathematical concepts. SEKI Report SR-89-20, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1989.

[10] Manfred Kerber. How to prove higher order theorems in first order logic. SEKI Report SR-90-19, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1990. Short version forthcoming in the proceedings of the 12th IJCAI, 1991.

[11] Manfred Kerber. *On the Representation of Mathematical Concepts and their Translation into First Order Logic.* PhD thesis, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1991, forthcoming.

[12] Douglas B. Lenat. *AM: An Artificial Intelligence Approach to Discovery in Mathematics as Heuristic Search.* PhD thesis, AI Lab, Stanford University, Stanford, California, USA, 1976. AIM-286, STAN-CS-76-570, and Heuristic Programming Project Report HPP-76-8.

[13] Douglas B. Lenat and Ramanathan V. Guha. *Building Large Knowledge-Based Systems – Representation and Inference in the CYC Project.* Addison-Wesley Publishing Company, Readings, Massachusetts, USA, 1990.

[14] Marvin Minsky. A framework for representing knowledge. In Ronald J. Brachman and Hector J. Levesque, editors, *Readings in Knowledge Representation*, chapter 12, pages 245–262. Morgan Kaufmann, 1985, Los Altos, California, USA, 1981. also in: Mind Design, 95–128, J. Haugeland, editor, Cambridge, Massachusetts, USA, MIT-Press.