In: MFDBS'89 (2nd Symp. on Mathematical Fundamentals of Database Systems)
(J. Demetrovics/B. Thalheim, ed.), Springer-Verlag, LNCS 364, Berlin, 1989, pp. 68–84

Specifying Closed World Assumptions for Logic Databases

Stefan Brass Udo W. Lipeck

FB Informatik VI, Universität Dortmund
Postfach 500500, D-4600 Dortmund 50, Fed. Rep. Germany

Abstract

"Closed world assumptions" (CWAs) are an important class of implicit completions for logic databases. We present a new general definition of CWA; it is parameterized, so that known and new versions of CWAs can be derived as special cases. Our CWA, in turn, instantiates the more basic notion of "database completion" and satisfies natural properties. It can even be characterized by the property of determining maximal completions without generating too much new information. We study syntactic as well as semantic definitions and prove them to be equivalent. By discussing several instances of CWAs we demonstrate the applicability of our framework to database specification.

1 Introduction

A logic database stores formulae which describe facts corresponding to conventional database information, rules for deducing further information, and indefinite information. Thus, a database state is a set of formulae and answers to queries should be logical consequences of such a state. The formulae are usually expressed in first order predicate logic or in a subset thereof like clausal logic.

Most logic databases, however, use some sort of implicit completion allowing to conclude formulae which are not consequences of the pure state. Typically, negative answers, i.e. negated formulae, can often not be obtained explicitly. A practically important class of database completions are "closed world assumptions" (CWAs), which originated with [Rei78] and were further developed by [Min82] and others (e.g. [GP86], [GPP86], [YH85]) into different versions.

The variety of completions proposed in the literature gives rise to the question which is the right one. Since Reiter's CWA lead to inconsistencies in the case of databases with indefinite information, other authors invented new CWAs which were similar but not explicitly related to each other. The approaches were driven by, e.g., the kind of formulae stored in the database or the kind of additional information to be concluded.

It is our point of view that a database completion cannot be chosen in advance for arbitrary applications, but must be tailored for specific ones. To this end, we present a new general definition of closed world assumption which includes a parameter allowing to derive the other CWA versions. It is this parameter which should be part of a database specification.

Intuitively, it describes the superset of "possible assumptions", usually given by a certain type of formulae. A CWA determines the completion of each state by a maximal part of these assumptions such that no new negations (negated conjunctions) of them can be concluded. We show that CWAs can even be characterized by this natural property of being maximal without generating too much new information.

Our paper is structured as follows: Section 2 compiles basic notions needed in the sequel. There we make one further step of abstraction and introduce general database completions together with their properties, having in mind that other kinds of completions have been proposed apart from closed world assumptions like, e.g., (first order) circumscription, minimal implication (second order circumscription), and Clark's completed databases (CDB). Surveys can be found in, e.g., [GMN84] and [She88]. By showing later that CWAs satisfy those general properties, we can unify and extend some results from the literature.

Section 3 will briefly outline the original notions of closed world assumptions as introduced by Reiter and Minker. Starting with Minker's work, CWAs have been defined not only in a syntactic way based on deducibility, but also in a semantic way based on minimal Herbrand models which formalize the idea of "closed worlds". In section 4, we also define our parameterized notion of CWA in these two ways, and we give a necessary and sufficient condition for their equivalence. This condition can be established for a large class of parameters by defining corresponding order relations on Herbrand models. Using the syntactic definition, we present the maximality characterization of CWA mentioned above.

Section 5 discusses typical (known and new) instances of CWAs by motivating the respective choice of the parameter. Thus, the applicability of our framework to the specification of logic databases is demonstrated.

This paper is based on a detailed elaboration in [Bra88].

2 Basic Notions and Properties

2.1 Formulae and Models

Let B be a set ("base") of non-logical symbols for constants, functions and relations (predicates). A B-structure σ is a corresponding interpretation of those symbols in an arbitrarily chosen carrier set; we write $\sigma[s]$ for that interpretation of a symbol s. We assume that the reader is familiar with the construction of formulae in first order predicate logic over a base as well as with the interpretation of such formulae in structures. Validity of a formula φ in a structure σ (for arbitrary assignments of values to free variables) is denoted by $\sigma \models \varphi$; this notation is used by analogy for formula sets Φ instead of single formulae φ. We write $\mathcal{F}_B$ for the entire set of formulae over B. The reader can find more detailed explanations in, e.g., [End72] and [Men78].

Since there are consistent and complete deduction systems for predicate logic, we need not make a difference between deducible and semantic implication, usually denoted by $\vdash$ and $\models$, respectively. So we can utilize the following relationship:

Lemma 1 *Let φ be a formula and let Φ be a set of formulae over B.*

$$\Phi \vdash \varphi \iff \Phi \models \varphi, \text{ i.e. } (\sigma \models \Phi \text{ implies } \sigma \models \varphi \text{ for all } B\text{-structures } \sigma).$$

Definition 2 *Let Φ be a set of formulae over B.*

1. *A model of Φ is a B-structure σ which satisfies all formulae in Φ, i.e. $\sigma \models \Phi$.*
2. *A B-structure is called Herbrand B-structure iff its carrier set consists of all ground terms over the base B (containing at least one constant symbol) and the function symbols are interpreted by usual composition of terms. (Thus, a Herbrand structure is completely specified by the interpretation of all relation symbols.)*
3. *A Herbrand model of Φ is a Herbrand structure which is a model of Φ.*

Concerning syntax, we only mention some notions of clausal sublogic explicitly.

Definition 3

1. *Atomic formulae are of the form $r(t_1, \ldots, t_n)$ with a relation symbol r and terms $t_1, \ldots, t_n$. They are called positive literals, their negations are called negative literals.*

2. *A clause is a disjunction* $\lambda_1 \vee \ldots \vee \lambda_m$ *of literals* $\lambda_1, \ldots, \lambda_m$*, which may be empty (false).*
3. *A Horn clause is a clause with at most one positive literal, a definite Horn clause has exactly one positive literal. Thus a definite Horn clause has the form* $\lambda_0 \vee \neg\lambda_1 \vee \ldots \vee \neg\lambda_m$ *with positive literals* $\lambda_0, \lambda_1, \ldots, \lambda_m$*, which is equivalent to* $\lambda_0 \leftarrow \lambda_1 \wedge \ldots \wedge \lambda_m$
4. *A positive clause consists of positive literals only.*
5. *Literals and clauses without variables are called ground literals and ground clauses, respectively. Positive ground literals are usually called facts. For a given subset* R *of predicate symbols,* R^+ *and* R^- *denote the sets of all positive resp. negative ground literals with predicate symbols only from* R*.*
6. *A clausal formula is a formula which is equivalent to a set of clauses.*

Obviously, any Herbrand structure σ may be identified with the set of facts valid in σ. In that sense, the inclusion $\sigma \subseteq \tau$ between Herbrand structures σ and τ denotes an inclusion between sets of facts.

The importance of clauses and Herbrand structures is emphasized by the next lemma:

Lemma 4 *For every set* Φ *of clausal formulae, the following holds:* Φ *has a Herbrand model if and only if it has an arbitrary model.*

The following well-known theorem from first-order logic will be used in later proofs:

Lemma 5 (Compactness Theorem) *If a set* Φ *of formulae is inconsistent, then there is a finite subset* $\{\varphi_1, \ldots, \varphi_n\} \subseteq \Phi$ *which is still inconsistent.*

We will also need the following simple relationship between logical implication and inconsistency:

Lemma 6 *Let* Φ *be a set of formulae and* $\psi_1, \ldots, \psi_n$ *be closed formulae.* $\Phi \cup \{\psi_1, \ldots, \psi_n\}$ *is inconsistent iff* $\Phi \vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$*.*

For order relations $<$ defined on Herbrand structures we will utilize two particular notions:

Definition 7

1. *A Herbrand model* σ *of* Φ *is called* $<$*-minimal iff there is no Herbrand model* τ *of* Φ *with* $\tau < \sigma$*.*
2. *The relation* $<$ *is well founded iff for every Herbrand model* σ *of an arbitrary set of clauses* Φ *a* $<$*-minimal Herbrand model* σ_0 *of* Φ *exists s.t.* $\sigma_0 \leq \sigma$*.*[1]

2.2 Completions of Logic Databases

Definition 8

- *A logic database is determined by a base* B*, a language* L *which is a set of formulae over* B *and a completion as defined below.*
- *A state of a logic database is a subset* Φ *of* L*. Let* $\mathcal{S}_L$ *denote the set of all states over* L*.*

The base B corresponds to the enumeration of relations and attributes in a relational database. The language L specifies the formulae which can be stored in the database. So it is possible to allow Horn clauses only or to put other requirements on the type of formulae. That the completion should be part of the database specification is a main issue of this paper.

Database states will normally be expected to be consistent and finite.

[1] This usage of the term "well founded" stems from papers on circumscription, but note that it does not necessarily exclude infinitely descending chains of structures, so it is not fully compatible with the "normal" meaning of this term.

Definition 9 *A B-completion on L is a mapping*

$$C : \mathcal{S}_L \to \mathcal{P}(\mathcal{F}_B)$$

such that $\Phi \subseteq C(\Phi)$ *for every state* Φ.

So a completion is simply a mapping which takes a database state and produces a "completed database state" which must be a set of formulae that contains the original database state.

A boolean query is any closed formula (over the base B of the database). The boolean query ω will be answered with "yes" in state Φ if $C(\Phi) \vdash \omega$. If $C(\Phi) \vdash \neg\omega$ holds instead, the answer will be "no". If neither the formula nor its negation follows from the completed database state, the answer should be "unknown".

It is interesting to note that a completion C can equivalently be described by a completion $\vdash_C$ of the $\vdash$-relation. Thus database completions such as CWAs which are specified corresponding to the definition above can be compared with completions like "minimal implication" which are formulated in this way:

Theorem 10

1. *Let C be a B-completion on L. Then the relation* $\vdash_C \subseteq \mathcal{S}_L \times \mathcal{F}_B$ *defined by*

$$\Phi \vdash_C \omega \iff C(\Phi) \vdash \omega$$

has following properties:

   - *If* $\Phi \vdash \omega$ *then* $\Phi \vdash_C \omega$ *("no loss of information").*
   - *If* $\Phi \vdash_C \omega_1, \ \ldots, \ \Phi \vdash_C \omega_n$ *and* $\{\omega_1, \ldots, \omega_n\} \vdash \omega$ *then* $\Phi \vdash_C \omega$ *("closure under logical consequences").*

2. *If a relation* $\vdash_C \subseteq \mathcal{S}_L \times \mathcal{F}_B$ *satisfies these two requirements for all states* Φ, *the mapping C defined by*

$$C(\Phi) = \{\omega | \omega \text{ is a formula over } B \text{ with } \Phi \vdash_C \omega\}$$

is a completion with

$$\Phi \vdash_C \omega \iff C(\Phi) \vdash \omega.$$

Now we formulate some properties of database completions in this general setting which makes them applicable to any completion. Most of them have been used in the literature, but for special completions only. The first three definitions list natural properties which should hold for any reasonable completion.

Definition 11

1. *A completion C preserves consistency iff* $C(\Phi)$ *is consistent for every consistent state* Φ.
2. *A completion C preserves equivalence iff* $C(\Phi_1)$ *and* $C(\Phi_2)$ *are equivalent for every pair of equivalent states* Φ_1 *and* Φ_2.
3. *A completion C is idempotent iff for all states* $\Phi_1, \Phi_2 \in \mathcal{S}_L$ *with* $\Phi_1 \subseteq \Phi_2$ *and* $C(\Phi_1) \vdash \Phi_2$ *the completed database states* $C(\Phi_1)$ *and* $C(\Phi_2)$ *are equivalent.*

Note that the property of idempotence implies that $C(C(\Phi))$ is equivalent to $C(\Phi)$; this simpler property, however, is not applicable to every completion since it requires $C(\Phi) \in \mathcal{S}_L$. Moreover our definition is stronger and implies other natural stability properties. It formalizes the expectation that a completion which assumes some fact does not change if we know that this fact really holds.

The following properties should be desired, too, but depend on certain parameters. They deal with the extend to which new information is generated to complete a database state.

Definition 12

1. *A completion C does not generate new information from Ω_0 iff for all $\omega \in \Omega_0$ and all states Φ holds:*
$$C(\Phi) \vdash \omega \iff \Phi \vdash \omega.$$
2. *A completion C is Ω_1-maximal wrt not generating new information from Ω_0 iff for each Φ and $\omega_1 \in \Omega_1$ with $C(\Phi) \not\vdash \omega_1$ there is a $\omega_0 \in \Omega_0$ with*
$$\Phi \not\vdash \omega_0 \text{ and } C(\Phi) \cup \{\omega_1\} \vdash \omega_0.$$
3. *A completion C is complete for a set of closed formulae Ω_1 iff for each Φ and $\omega_1 \in \Omega_1$ holds: $C(\Phi) \vdash \omega_1$ or $C(\Phi) \vdash \neg\omega_1$.*

The first property formalizes that we do not want arbitrary new information to be concluded from the completed, but not from the original database state. Therefore it is interesting to specify what sort of new information must not be generated by a specific completion.

On the other hand we do not want information to miss at random. The second property deals with the problem how much information may be assumed maximally without generating undesired new information. It means that you cannot add any new formula from Ω_1 to the completed state without generating new information from Ω_0.

Now the user probably wishes to get the answer either "yes" or "no" on any question, but not the answer "unknown". This is not always possible, because sometimes incomplete information is intentionally included into the database state. But in most cases the completion is used to achieve this sort of "completeness" formalized by the third property. Note that completeness means maximality for $\Omega_0 = \{false\}$ (if the completion is concistency preserving).

3 The Original Closed World Assumption

Normally, we only store positive information in a database and assume that it is complete, so that negative information can be derived by default. This is what any relational database does: It stores the tuples of several relations and it assumes that any tuple not stored is not contained in the relation. This means: if our database state Φ consists of facts only we assume a negative fact $\neg\lambda$ iff $\lambda \notin \Phi$. If we additionally have rules in the database then all we have to do is to substitute set containment by logical derivability. So we assume a negative fact $\neg\lambda$ iff $\Phi \not\vdash \lambda$. This is the definition of Reiter's CWA [Rei78]:
$$\text{CWA}(\Phi) = \Phi \cup \{\neg\lambda | \lambda \text{ is a positive ground literal and } \Phi \not\vdash \lambda\}.$$
It works similar to the "negation as failure"-rule in logic programming: To prove that $\neg\lambda$ is true one tries to prove λ, and if the proof for λ fails $\neg\lambda$ is assumed to be true. The difference is that here a proof is considered as failed also in the case when the proof procedure runs into an infinite loop. From the logical point of view, computability problems are ignored.

This works fine as long as Φ consists of Horn clauses only. But consider states consisting of arbitrary clauses, e.g. $\Phi = \{p(a) \vee p(b)\}$: Since $\Phi \not\vdash p(a)$ and $\Phi \not\vdash p(b)$ we would assume both of $\neg p(a)$ and $\neg p(b)$, which, however, leads to an inconsistent completed state. Here we must be more careful in selecting positive ground literals λ for negation; $\Phi \not\vdash \lambda$ does not suffice. Instead we have to check that there is no positive ground clause $\lambda \vee \lambda_1 \vee \ldots \vee \lambda_n$ that necessarily contains λ and follows from Φ. This is the definition of Minker's Generalized CWA [Min82][2]:

[2] In the original definition it is not explicitly required that $\lambda_1, \ldots, \lambda_n$ are ground. But we cannot leave this requirement out because otherwise the original semantic definition would not be equivalent. This shows the example $\Phi = \{q(a) \vee p(x), p(a)\}$ where a is the only constant: $\neg q(a)$ is true in every minimal Herbrand model, but $\Phi \vdash q(a) \vee p(x)$ and $\Phi \not\vdash p(x)$.

$$\begin{aligned}\mathrm{GCWA}(\Phi) \;=\; & \Phi \cup \{\neg\lambda | \lambda \text{ is a positive ground literal and} \\ & \quad \text{there is no empty or positive ground clause } \lambda_1 \vee \ldots \vee \lambda_n \\ & \quad \text{such that } \Phi \vdash \lambda \vee \lambda_1 \vee \ldots \vee \lambda_n \text{ and } \Phi \not\vdash \lambda_1 \vee \ldots \vee \lambda_n\}.\end{aligned}$$

This definition implies GCWA(Φ) = CWA(Φ) if CWA(Φ) is consistent.

Minker gave a second definition of his GCWA which he called "semantic", because it is based on minimal Herbrand models. The above definition, however, uses the "deducibility"-relation and is therefore a "syntactic" definition. The order on Herbrand models used here is just defined by set inclusion for the interpretation of relation symbols (predicates): $\sigma < \tau$ iff every fact true in σ is true in τ and at least one fact true in τ is not true in σ. So in a minimal Herbrand model as few facts hold as possible (Note that there may be more than one such model, though; e.g. $\Phi = \{p(a) \vee p(b)\}$ has the two minimal Herbrand models $\{p(a)\}$ and $\{p(b)\}$.). For completion we may safely assume only such negative ground literals that are true in any minimal Herbrand model:

$$\begin{aligned}\mathrm{GCWA}(\Phi) \;=\; & \{\neg\lambda | \lambda \text{ is a positive ground literal and} \\ & \quad \sigma \models \neg\lambda \text{ for all minimal Herbrand models } \sigma \text{ of } \Phi\}.\end{aligned}$$

4 A Unifying Framework for Closed World Assumptions

In this section, we unify and generalize the notions of CWAs previously proposed in the literature. We start with a new parameterized syntactic definition, give a natural characterization, and discuss the relationship between syntactic and semantic definition. The subsequent section 5 will show how to instantiate the parameter in typical cases.

4.1 The Definition

So we now want to define the CWA-completed database state $C(\Phi)$ of each database state Φ. We specify a set of "actual" assumptions $A(\Phi)$ for a state Φ and then define $C(\Phi) = \Phi \cup A(\Phi)$.

To this end, we start with a superset of "possible" assumptions Ψ. This will be the parameter in our definition of closed world assumption. Minker's GCWA, for example, can be obtained by choosing Ψ to be the set of negative ground literals (negative facts). We, however, allow any set of closed formulae to be used as a set of possible assumptions.

If we assume every formula from Ψ, we would get an inconsistent completion; e.g., in the database state $\{p(a)\}$ we cannot assume $\neg p(a)$. A minimal requirement on $C(\Phi)$, however, is consistency. On the other hand, a kind of completeness seems to be reasonable, too: we want to assume as much from Ψ as possible while preserving consistency. This leads to the definition of "maximal extension", which is a first trail for the desired subset $A_\Psi(\Phi) \subseteq \Psi$:

Definition 13 *Let Φ be a database state and Ψ be a set of possible assumptions. A subset $\Psi' \subseteq \Psi$ is a maximal Ψ-extension of Φ iff*

- $\Phi \cup \Psi'$ *is consistent and*
- $\Phi \cup \Psi' \cup \{\psi\}$ *is inconsistent for every* $\psi \in \Psi \setminus \Psi'$.

There remains the problem that more than one maximal Ψ-extension of Φ may exist. Consider the following example: Let $\Phi = \{p(a) \vee p(b)\}$ and $\Psi = \{\neg p(a), \neg p(b)\}$. Either of $\Psi'_1 = \{\neg p(a)\}$ and $\Psi'_2 = \{\neg p(b)\}$ is a maximal Ψ-extension of Φ. Because we cannot decide which of $\neg p(a)$ and $\neg p(b)$ to assume, we solve this problem in quite a cautious way: we assume none of them. Thus, in the general case we only take the intersection of all maximal Ψ-extensions:

Definition 14 ("Syntactic Definition") *Let Φ be a database state, and Ψ be a set of possible assumptions and $\overline{\Psi}(\Phi)$ be the set of all maximal Ψ-extensions of Φ. Then $A_\Psi(\Phi)$ (the set of "actual assumptions" from Ψ in state Φ) is defined as*

$$A_\Psi(\Phi) = \bigcap_{\Psi' \in \overline{\Psi}(\Phi)} \Psi',$$

and the Ψ-completion of Φ is $C_\Psi(\Phi) = \Phi \cup A_\Psi(\Phi)$.

The definition is called "syntactic" by analogy to MINKER's diction, and it in fact relies on the deducibility relation $\vdash$ (the consistency requirements of definition 13 may be read as "$\not\vdash$ false").

Note that there are always maximal Ψ-extensions of a consistent state Φ:

Lemma 15 *For each $\Psi_0 \subseteq \Psi$ s.t. $\Phi \cup \Psi_0$ is consistent, there is a maximal Ψ-extension Ψ' of Φ with $\Psi_0 \subseteq \Psi'$.*

Proof Let $\psi_1, \psi_2, \ldots$ be an enumeration of Ψ. Define

$$\Psi_i = \Psi_{i-1} \cup \begin{cases} \{\psi_i\} & \text{if } \Phi \cup \Psi_{i-1} \cup \{\psi_i\} \text{ is consistent} \\ \emptyset & \text{otherwise} \end{cases}$$

for $i = 1, 2, \ldots$. Now let

$$\Psi' = \bigcup_{i=0}^{\infty} \Psi_i.$$

$\Phi \cup \Psi'$ is consistent according to the compactness theorem and $\Phi \cup \Psi' \cup \{\psi\}$ is inconsistent for every $\psi \in \Psi \setminus \Psi'$ by construction. □

In order to confirm that our definition contains other CWA versions as special cases, we now give an equivalent formulation which corresponds to definitions found in the literature. The next lemma gives an intermediate description of the actual assumptions $A_\Psi(\Phi)$:

Lemma 16 *For Φ and Ψ given as above, the following holds:*

$$A_\Psi(\Phi) = \{\psi \in \Psi | \textit{for all finite subsets } \Psi_f \subseteq \Psi : \\ \Phi \cup \Psi_f \textit{ consistent implies } \Phi \cup \Psi_f \cup \{\psi\} \textit{ consistent}\}.$$

Proof Let $A'_\Psi(\Phi)$ be the set on the right hand side.

$A'_\Psi(\Phi) \subseteq A_\Psi(\Phi)$: For a formula $\psi \notin A_\Psi(\Phi)$, there must be a maximal Ψ-extension Ψ' of Φ with $\psi \notin \Psi'$. This means that $\Phi \cup \Psi' \cup \{\psi\}$ is inconsistent (because of the maximality of Ψ'). With the compactness theorem we can conclude that there is a finite subset $\Psi_f \subseteq \Psi'$ such that $\Phi \cup \Psi_f \cup \{\psi\}$ is inconsistent. But $\Phi \cup \Psi_f$ is consistent since $\Phi \cup \Psi'$ is consistent (by definition of maximal extension), so that $\psi \notin A'_\Psi(\Phi)$.

$A_\Psi(\Phi) \subseteq A'_\Psi(\Phi)$: Let $\psi \notin A'_\Psi(\Phi)$ and $\Psi_f \subseteq \Psi$ with $\Phi \cup \Psi_f$ consistent and $\Phi \cup \Psi_f \cup \{\psi\}$ inconsistent. According to the previous lemma there is a maximal Ψ-extension Ψ' of Φ with $\Psi_f \subseteq \Psi'$. Of course $\Phi \cup \Psi' \cup \{\psi\}$ must be inconsistent, too, but $\Phi \cup \Psi'$ must be consistent. This means that $\psi \notin \Psi'$ and $\psi \notin A_\Psi(\Phi)$. □

The following theorem now obviously shows the generalization of MINKER's GCWA definition.

Theorem 17 *Let Φ be a database state and Ψ a set of possible assumptions. Then we have:*

$$A_\Psi(\Phi) = \{\psi \in \Psi | \textit{there are no } \{\psi_1, \ldots, \psi_n\} \subseteq \Psi,\ n \geq 0, \\ \textit{such that } \Phi \vdash \neg\psi \vee \neg\psi_1 \vee \ldots \vee \neg\psi_n \textit{ and } \Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n\}.$$

Proof Using lemma 16 and lemma 6, we have

$$A_\Psi(\Phi) = \{\psi \in \Psi | \textit{for all finite subsets } \{\psi_1, \ldots, \psi_n\} \subseteq \Psi : \\ \Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n \textit{ implies } \Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n \vee \neg\psi\}.$$

Now the theorem follows by a simple logical equivalence. □

The rest of this subsection contains technical modifications of this theorem to formally prove that some other proposed versions of the CWA are really special cases. You may want to skip it.

The following lemma is an immediate corrollary of the theorem. It shows that some assumptions are only needed to block others.

Lemma 18 *Let* Φ *be a database state and* Ψ *be a set of possible assumptions. Let* $\Psi_0 \subseteq \Psi$ *such that for each* $\psi \in \Psi_0$ *with* $\Phi \not\vdash \psi$ *there are* $\psi_1, \ldots, \psi_n \in \Psi$ *with* $\Phi \vdash \neg\psi \vee \neg\psi_1 \vee \ldots \vee \neg\psi_n$ *and* $\Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$*. Then* $C_\Psi(\Phi) = \Phi \cup A_\Psi(\Phi)$ *is equivalent to* $\Phi \cup (A_\Psi(\Phi) \setminus \Psi_0)$*.*

An important special case is the following: Assume that there is a $\Psi_1 \subseteq \Psi$ such that $\Psi_0 = \Psi_1 \cup \{\neg\psi | \psi \in \Psi_1\}$ is a subset of Ψ. Then the conditions of the previous lemma are automatically satisfied and we can conclude that these assumptions are not needed in the completed database state. They are needed in the completion process, though, to block other assumptions. This was utilized in the CWA of [GP86].

If the possible assumptions are disjunctions of "atomic assumptions" Ψ_0 then it suffices to restrict $\psi_1, \ldots, \psi_n$ in theorem 17 to such atomic assumptions (This was utilized in the CWA of [YH85]):

Lemma 19 *Let* Ψ *be a set of possible assumptions and* $\Psi_0 \subseteq \Psi$ *such that every* $\psi \in \Psi$ *has the form* $\psi_1 \vee \ldots \vee \psi_n$ *with* $\psi_1, \ldots, \psi_n \in \Psi_0$*. Then the following holds:*

$$A_\Psi(\Phi) = \{\psi \in \Psi | \textit{there are no } \{\psi_1, \ldots, \psi_n\} \subseteq \Psi_0,\ n \geq 0, \\ \textit{such that } \Phi \vdash \neg\psi \vee \neg\psi_1 \vee \ldots \vee \neg\psi_n \textit{ and } \Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n\}.$$

Proof Let $A'_\Psi(\Phi)$ be the set on the right hand side.

$A_\Psi(\Phi) \subseteq A'_\Psi(\Phi)$: For a possible assumption $\psi \notin A'_\Psi(\Phi)$, there are $\psi_1, \ldots, \psi_n \in \Psi_0$ with $\Phi \vdash \neg\psi \vee \neg\psi_1 \vee \ldots \vee \neg\psi_n$ and $\Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$. Now theorem 17 and $\Psi_0 \subseteq \Psi$ imply $\psi \notin A_\Psi(\Phi)$.

$A'_\Psi(\Phi) \subseteq A_\Psi(\Phi)$: For a possible assumption $\psi \notin A_\Psi(\Phi)$, there are $\psi_1, \ldots, \psi_n \in \Psi$ with $\Phi \vdash \neg\psi \vee \neg\psi_1 \vee \ldots \vee \neg\psi_n$ and $\Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$ (by theorem 17). By the requirement on Ψ there are $\psi_{i,j} \in \Psi_0$, $i = 1, \ldots, n$, $j = 1, \ldots, m_i$ with $\psi_i = \psi_{i,1} \vee \ldots \vee \psi_{i,m_i}$. Now

$$\Phi \not\vdash \neg(\psi_{1,1} \vee \ldots \vee \psi_{1,m_1}) \vee \ldots \vee \neg(\psi_{n,1} \vee \ldots \vee \psi_{n,m_n})$$

means that there is a model σ of Φ and there are $j_1, \ldots, j_n$ with $1 \leq j_i \leq m_i$ such that $\sigma \not\models \neg\psi_{i,j_i}$ for $i = 1, \ldots, n$. Thus, $\Phi \not\vdash \neg\psi_{1,j_1} \vee \ldots \vee \neg\psi_{n,j_n}$. On the other hand,

$$\Phi \vdash \neg\psi \vee (\neg\psi_{1,1} \wedge \ldots \wedge \neg\psi_{1,m_1}) \vee \ldots \vee (\neg\psi_{n,1} \wedge \ldots \wedge \neg\psi_{n,m_n}).$$

So for each model σ of Φ with $\sigma \not\models \neg\psi$ there is an $i \in \{1, \ldots, n\}$ with $\sigma \models \neg\psi_{i,j}$ for $j = 1, \ldots, m_j$. Thus, $\Phi \vdash \neg\psi \vee \neg\psi_{1,j_1} \vee \ldots \vee \neg\psi_{n,j_n}$. Therefore $\psi \notin A'_\Psi(\Phi)$. □

4.2 Properties and Characterization

Our CWA indeed has the basic properties required for completions in subsection 2.2.

Theorem 20 C_Ψ *preserves consistency, is idempotent, and preserves equivalence.*

Proof The preservation of consistency and equivalence directly follows from the definition.

For the proof that C_Ψ is idempotent let states Φ_1 and Φ_2 with $\Phi_1 \subseteq \Phi_2$ and $C_\Psi(\Phi_1) \vdash \Phi_2$ be given. It suffices to show that $A_\Psi(\Phi_1) = A_\Psi(\Phi_2)$.

Let ψ be a possible assumption with $\psi \notin A_\Psi(\Phi_2)$. So there must be a maximal extension Ψ'_2 of Φ_2 with $\psi \notin \Psi'_2$. Now $\Phi_2 \cup \Psi'_2$ is consistent, therefore its subset $\Phi_1 \cup \Psi'_2$ is also consistent and by lemma 15 there is a maximal extension Ψ'_1 of Φ_1 with $\Psi'_2 \subseteq \Psi'_1$. Any model of $\Phi_1 \cup \Psi'_1 \cup \{\psi\}$ would be a model of $\Phi_2 \cup \Psi'_2 \cup \{\psi\}$ (since $\Phi_1 \cup \Psi'_1 \supseteq \Phi_1 \cup A_\Psi(\Phi_1) = C_\Psi(\Phi_1) \vdash \Phi_2$ and $\Psi'_1 \supseteq \Psi'_2$). But $\Phi_2 \cup \Psi'_2 \cup \{\psi\}$ is inconsistent, so that $\Phi_1 \cup \Psi'_1 \cup \{\psi\}$ is inconsistent, i.e. $\psi \notin \Psi'_1$ and $\psi \notin A_\Psi(\Phi_1)$.

For the other direction let $\psi \notin A_\Psi(\Phi_1)$. So there is a maximal extension Ψ'_1 of Φ_1 with $\psi \notin \Psi'_1$. Now $\Phi_1 \cup \Psi'_1$ is consistent, so let σ be a model. $\Phi_1 \cup \Psi'_1 \supseteq \Phi_1 \cup A_\Psi(\Phi_1) = C_\Psi(\Phi_1)$ and $C_\Psi(\Phi_1) \vdash \Phi_2$ imply that σ is a model of $\Phi_2 \cup \Psi'_1$. By lemma 15 there is a maximal extension Ψ'_2 of Φ_2 with $\Psi'_2 \supseteq \Psi'_1$. Since $\psi \notin \Psi'_1$ we have that $\Phi_1 \cup \Psi'_1 \cup \{\psi\}$ is inconsistent and therefore the superset $\Phi_2 \cup \Psi'_2 \cup \{\psi\}$ is inconsistent, too. Since $\Phi_2 \cup \Psi'_2$ is consistent, $\psi \notin \Psi'_2$ and $\psi \notin A_\Psi(\Phi_2)$. □

The properties 1 and 2 of definition 12 hold for the parameters $\Omega_1 = \Psi$ and $\Omega_0 = \{\neg\psi_1 \vee \ldots \vee \neg\psi_n \mid \psi_i \in \Psi\}$, and they even lead to a natural characterization of our CWA. Up to equivalence, the latter set contains negated conjunctions of the possible assumptions Ψ.

Theorem 21 *Let Ψ be a set of possible assumptions. C_Ψ does not generate new information from $\Omega_0 = \{\neg\psi_1 \vee \ldots \vee \neg\psi_n \mid \psi_i \in \Psi\}$, it is Ψ-maximal wrt to not generating new information from Ω_0. Moreover, it is the least completion (up to equivalence) that has these properties.*

Proof At first we have to prove

$$C_\Psi(\Phi) \vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n \iff \Phi \vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n.$$

The direction $\Longleftarrow$ trivially follows from $\Phi \subseteq C_\Psi(\Phi)$. The proof for the direction $\Longrightarrow$ is indirect: Assume that

$$C_\Psi(\Phi) \vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n \text{ and } \Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$$

for some $\psi_1, \ldots, \psi_n \in \Psi$. So $\Phi \cup A_\Psi(\Phi) \cup \{\psi_1, \ldots, \psi_n\}$ is inconsistent, but $\Phi \cup \{\psi_1, \ldots, \psi_n\}$ is consistent. Now the compactness theorem says that there are $\{\psi'_1, \ldots, \psi'_m\} \subseteq A_\Psi(\Phi)$ such that $\Phi \cup \{\psi'_1, \ldots, \psi'_m\} \cup \{\psi_1, \ldots, \psi_n\}$ still is inconsistent. Then lemma 15 says that there is a Ψ-maximal extension Ψ' of Φ with $\{\psi_1, \ldots, \psi_n\} \subseteq \Psi'$, which must be consistent with Φ, so that $\{\psi'_1, \ldots, \psi'_m\} \not\subseteq \Psi'$ and therefore $\{\psi'_1, \ldots, \psi'_m\} \not\subseteq A_\Psi(\Phi)$. Thus we have got a contradiction.

For the second part of the theorem we have to show that for each $\psi \in \Psi$ with $C_\Psi(\Phi) \not\vdash \psi$ there are $\psi_1, \ldots, \psi_n \in \Psi$ such that

$$\Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n \quad \text{and} \quad C_\Psi(\Phi) \cup \{\psi\} \vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n.$$

Since $C_\Psi(\Phi) \not\vdash \psi$ and therefore $\psi \notin C_\Psi(\Phi)$ there is a Ψ-maximal extension Ψ' of Φ with $\psi \notin \Psi'$. By definition $\Phi \cup \Psi' \cup \{\psi\}$ must be inconsistent and by the compactness theorem there are $\psi_1, \ldots, \psi_n \in \Psi' \subseteq \Psi$ with $\Phi \cup \{\psi_1, \ldots, \psi_n\} \cup \{\psi\}$ inconsistent, i.e.

$$C_\Psi(\Phi) \cup \{\psi\} \supseteq \Phi \cup \{\psi\} \vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$$

and therefore $C_\Psi(\Phi) \cup \{\psi\} \vdash \neg\psi_1 \vee \ldots \neg\psi_n$. By definition of maximal extension, $\Phi \cup \{\psi_1, \ldots, \psi_n\}$ is consistent and therefore $\Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$.

For the third part of the theorem let $D(\Phi)$ be any completion of Φ, i.e. $\Phi \subseteq D(\Phi)$, which has the two properties shown above for $C_\Psi(\Phi)$. Now we want to prove that $D(\Phi) \vdash C_\Psi(\Phi)$. Assume that this is not the case, so there is some $\psi \in C_\Psi(\Phi)$ with $D(\Phi) \not\vdash \psi$, i.e. ψ must belong to $C_\Psi(\Phi) \setminus \Phi \subseteq \Psi$. Since $D(\Phi)$ is Ψ-maximal wrt not generating new information from Ω_0, there are $\psi_1, \ldots, \psi_n \in \Psi$ with

$$\Phi \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n \text{ and } D(\Phi) \cup \{\psi\} \vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n,$$

i.e. $D(\Phi) \vdash \neg\psi \vee \neg\psi_1 \vee \ldots \vee \neg\psi_n$. From $\Phi \not\vdash \neg\psi_1 \vee \ldots \neg\psi_n$ we know according to the first part of this theorem that $C_\Psi(\Phi) \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$. Since $\Phi \cup \{\psi\} \subseteq C_\Psi(\Phi)$, this implies $\Phi \cup \{\psi\} \not\vdash \neg\psi_1 \vee \ldots \vee \neg\psi_n$, i.e. $\Phi \not\vdash \neg\psi \vee \neg\psi_1 \vee \ldots \vee \neg\psi_n$. Now we have the contradiction that $D(\Phi)$ generates new information from Ω_0. □

4.3 Syntactic and Semantic Definition

Starting with [Min82], closed world assumptions have been characterized "semantically" by means of minimal Herbrand models. Such a characterization can here be given in a parameterized form as in the following definition. In this subsection we consider only such sets of formulae where each formula is equivalent to a set of clauses. This requirement is needed because otherwise it would not be sufficient to consider only Herbrand models.

Definition 22 ("Semantic Definition") *Let Φ be a clausal database state, Ψ a clausal set of possible assumptions and $<$ an order relation on Herbrand structures. Then let*

$$B^<_\Psi(\Phi) = \{\psi \in \Psi | \text{ for all } <\text{-minimal Herbrand models } \sigma \text{ of } \Phi\text{: } \sigma \models \psi\},$$

and the Ψ-completion of Φ is $C^<_\Psi(\Phi) = \Phi \cup B^<_\Psi(\Phi)$.

Each version of the CWA (i.e. each value of the parameter Ψ) needs its own order relation. We now define a notion of "compatibility" between Ψ and $<$, and then we show in the following two theorems that it is a necessary and sufficient condition for the equivalence of syntactic and semantic definition. But first let us define a useful notation:

Definition 23 *For a Herbrand structure σ let $\Psi_\sigma = \{\psi \in \Psi | \sigma \models \psi\}$.*

Definition 24 *Let Ψ and $<$ be given as above. $<$ is called compatible with Ψ iff*

$$\sigma \leq \tau \implies \Psi_\sigma \supseteq \Psi_\tau \qquad \text{and} \qquad \Psi_\sigma \supset \Psi_\tau \implies \sigma < \tau.$$

Theorem 25 *Let $<$ be an order relation on Herbrand structures which is well founded. If $<$ is compatible with Ψ, then $B^<_\Psi(\Phi) = A_\Psi(\Phi)$ holds for all sets Φ of clausal formulae*[3].

Proof First we show by contraposition that $B^<_\Psi(\Phi) \subseteq A_\Psi(\Phi)$: Let $\psi \in \Psi$ but $\psi \notin A_\Psi(\Phi)$. Then there is a maximal Ψ-extension Ψ' of Φ with $\psi \notin \Psi'$. $\Phi \cup \Psi'$ is consistent, so there is a Herbrand model σ of $\Phi \cup \Psi'$. Let σ_0 be a $<$-minimal Herbrand model of Φ with $\sigma_0 \leq \sigma$ (such a model exists because $<$ is well founded). With the first condition of compatibility we can conclude $\Psi_{\sigma_0} \supseteq \Psi_\sigma \supseteq \Psi'$, so that $\sigma_0 \models \Phi \cup \Psi'$. Since $\Phi \cup \Psi' \cup \{\psi\}$ is inconsistent, we, however, have $\sigma_0 \not\models \psi$, so that $\psi \notin B^<_\Psi(\Phi)$.

Now we show the other direction $B^<_\Psi(\Phi) \supseteq A_\Psi(\Phi)$: Let $\psi \in \Psi$ but $\psi \notin B^<_\Psi(\Phi)$. Then there is a $<$-minimal Herbrand model σ of Φ with $\sigma \not\models \psi$. Now Ψ_σ is a maximal Ψ-extension of Φ:

- $\Phi \cup \Psi_\sigma$ is consistent since $\sigma \models \Phi \cup \Psi_\sigma$.
- If there were a $\psi' \in \Psi$ with $\psi' \notin \Psi_\sigma$ and $\Phi \cup \Psi_\sigma \cup \{\psi'\}$ consistent, then $\Phi \cup \Psi_\sigma \cup \{\psi'\}$ would have a Herbrand model σ' with $\Psi_{\sigma'} \supset \Psi_\sigma$. The second condition of compatibility, however, implies $\sigma' < \sigma$, so that a contradiction to the assumed minimality of σ arises.

Thus Ψ_σ is a maximal Ψ-extension of Φ with $\psi \notin \Psi_\sigma$, i.e. $\psi \notin A_\Psi(\Phi)$. □

Theorem 26 *Let $<$ again be an order relation which is well founded. If $B^<_\Psi(\Phi) = A_\Psi(\Phi)$ holds for all sets Φ as in theorem 25, then $<$ is compatible with Ψ.*

[3] $B^<_\Psi(\Phi)$ and $A_\Psi(\Phi)$ have been defined general enough to allow this kind of "database states". Especially infinite states are needed in the other direction (theorem 26).

For the proof we need the following lemma:

Lemma 27 *Let σ and τ be two Herbrand structures. Then there is a set of clauses Φ with σ and τ as only Herbrand models.*

Proof Let Φ_σ and Φ_τ be the set of (negative or positive) ground literals true in σ and τ respectively. Now define

$$\Phi = \{\varphi_\sigma \vee \varphi_\tau | \varphi_\sigma \in \Phi_\sigma, \varphi_\tau \in \Phi_\tau\}.$$

Then $\sigma \models \Phi$ and $\tau \models \Phi$ and for any other Herbrand structure ρ there are (negative or positive) ground literals φ_σ and φ_τ with $\sigma \models \varphi_\sigma$, $\rho \not\models \varphi_\sigma$ and $\tau \models \varphi_\tau$, $\rho \not\models \varphi_\tau$. Now we have $\varphi_\sigma \vee \varphi_\tau \in \Phi$ but $\rho \not\models \varphi_\sigma \vee \varphi_\tau$. □

Proof (of theorem 26) So we have to prove the two conditions of compatibility.

- Let us begin with $\sigma \leq \tau \implies \Psi_\sigma \supseteq \Psi_\tau$: If this would not hold then there would be Herbrand structures σ and τ with $\sigma \leq \tau$ but $\Psi_\sigma \not\supseteq \Psi_\tau$. Let $\psi_0 \in \Psi_\tau \setminus \Psi_\sigma$ and Φ be the set of clauses with only σ and τ as Herbrand models (according to the previous lemma). σ is the only minimal model of Φ, so $B^{<}_{\Psi}(\Phi) = \Psi_\sigma$. We now have to consider the following two cases and show that $A_\Psi(\Phi) \neq B^{<}_{\Psi}(\Phi)$:
  - $\Psi_\sigma \subseteq \Psi_\tau$: $\Phi \cup \Psi_\tau$ is consistent (τ is a model) and for every $\psi \in \Psi \setminus \Psi_\tau$ we have that $\Phi \cup \{\psi\}$ is inconsistent ($\tau \not\models \psi$ and therefore $\sigma \not\models \psi$ and there are no other Herbrand models of Φ). This means that Ψ_τ is a maximal Ψ-extension of Φ and the only one ($\Phi \cup \{\psi\}$ inconsistent for any $\psi \in \Psi \setminus \Psi_\tau$ implies that any other maximal Ψ-extension would have to be a proper subset of Ψ_τ). Now we have a contradiction because $\psi_0 \in A_\Psi(\Phi)$ and $\psi_0 \notin B^{<}_{\Psi}(\Phi)$.
  - $\Psi_\sigma \not\subseteq \Psi_\tau$, i.e. there is a $\psi_1 \in \Psi_\sigma \setminus \Psi_\tau$. $\Phi \cup \Psi_\tau \cup \{\psi\}$ is inconsistent for every $\psi \in \Psi \setminus \Psi_\tau$ because $\sigma \not\models \psi_0 \in \Psi_\tau$ and $\tau \not\models \psi$ and there are no other models of Φ, so Ψ_τ is a maximal Ψ-extension of Φ. $\psi_1 \notin \Psi_\tau$ implies $\psi_1 \notin A_\Psi(\Phi)$, but $\psi_1 \in B^{<}_{\Psi}(\Phi)$.
- Now we have to show the other condition of compatibility, namely $\Psi_\sigma \supset \Psi_\tau \implies \sigma < \tau$. If we assume the contrary, there are Herbrand structures σ and τ with $\Psi_\sigma \supset \Psi_\tau$ but $\sigma \not< \tau$. Let $\psi_0 \in \Psi_\sigma \setminus \Psi_\tau$ and Φ be a set of clauses with only Herbrand models σ and τ. Since $\sigma \not< \tau$, τ is a $<$-minimal model of Φ. $\tau \not\models \psi_0$, so $\psi_0 \notin B^{<}_{\Psi}(\Phi)$. On the other side $\Phi \cup \{\psi\}$ is inconsistent for every $\psi \in \Psi \setminus \Psi_\sigma$ since $\sigma \not\models \psi$ and therefore $\tau \not\models \psi$. This means that Ψ_σ is a maximal Ψ-extension of Φ and the only one. So we have $\psi_0 \in A_\Psi(\Phi)$, which is a contradiction to $A_\Psi(\Phi) = B^{<}_{\Psi}(\Phi)$.□

These two theorems show that compatibility is a necessary and sufficient condition for the equivalence of syntactic and semantic definition. The stronger symmetric versions of compatibility

$$\sigma \leq \tau \iff \Psi_\sigma \supseteq \Psi_\tau \qquad \text{and} \qquad \sigma < \tau \iff \Psi_\sigma \supset \Psi_\tau$$

are not necessary, as can be seen using $\Psi = \emptyset$ and a nontrivial order relation. But they are of course sufficient and the interesting point is that the right equivalence defines an order relation which can be shown to be well founded. Thus for every choice of the parameter Ψ there is a corresponding order relation such that $A_\Psi(\Phi) = B^{<}_{\Psi}(\Phi)$.

Theorem 28 *Let Ψ be as above and $<$ be the relation on Herbrand structures defined by*

$$\sigma < \tau \iff \Psi_\sigma \supset \Psi_\tau.$$

This relation is an order relation which is well founded.

Proof Since $\supset$ is an order relation, $<$ is an order relation, too.

So we have to show that for every Herbrand model σ_1 of an arbitrary set of clauses Φ there is a $<$-minimal Herbrand model σ_0 of Φ with $\sigma_0 \leq \sigma_1$. Let Σ be the set of all Herbrand models σ of Φ with $\sigma \leq \sigma_1$. We have to prove that Σ contains a minimal element. Using ZORN's lemma it suffices to show that every chain $\Sigma' \subseteq \Sigma$ is lower bounded. If Σ' contains its lower bound, nothing remains to be proven. Otherwise, define

$$\Psi' \;=\; \{\psi \in \Psi |\text{ there is a } \sigma \in \Sigma' \text{ with } \sigma \models \psi\} \;=\; \bigcup_{\sigma \in \Sigma'} \Psi_\sigma.$$

First we show that $\Phi \cup \Psi'$ is consistent: Otherwise there would be a finite subset $\{\psi_1, \ldots, \psi_n\} \subseteq \Psi'$ such that $\Phi \cup \{\psi_1, \ldots \psi_n\}$ still is inconsistent (by the compactness theorem). Now let $\sigma'_i \in \Sigma'$ with $\sigma'_i \models \psi_i$ for $i = 1, \ldots, n$. Since Σ' is a chain, $\{\sigma'_1, \ldots, \sigma'_n\}$ contains a minimal element σ'. $\sigma' \models \psi'_i$ $(i = 1, \ldots, n)$ holds by definition of $<$. So $\sigma' \models \Phi \cup \{\psi'_1, \ldots, \psi'_n\}$ follows which is a contradiction. Since now $\Phi \cup \Psi'$ is known to be consistent, it has a Herbrand model σ''. By definition of Ψ', $\Psi_{\sigma''} \supseteq \Psi_\sigma$ for each $\sigma \in \Sigma'$. But since we have excluded the trivial case that Σ' contains a lower bound of itself, we even have $\Psi_{\sigma''} \supset \Psi_\sigma$. (Any $\tau \in \Sigma'$ with $\Psi_\tau = \Psi_{\sigma''}$ would be a lower bound of Σ' since it is a chain.) Now by definition of $<$, σ'' indeed is a lower bound of Σ'. □

5 Instances of the CWA

5.1 Horn Clauses

Originally, the problem of database completion arose when viewing states of a relational database as sets of of positive ground literals (facts). For example, the table

orders	customer	article
	Smith	milk
	Jones	cookies

can be translated into the set of formulae

$$\Phi = \{\mathrm{order(Smith, milk), order(Jones, cookies)}\}.$$

Relational database systems assume that tuples not mentioned in the table are not contained in the relation. Negative facts like $\neg$order(Smith, cookies), however, are not a logical consequence of Φ. This difficulty can be overcome with REITER's original CWA (and MINKER's GCWA which is the same in the case of Horn clauses).

In order to derive this version of a CWA from our general definition, we have to choose the set of negative ground literals as the parameter Ψ. The corresponding order relation is

$$\sigma < \tau \iff \sigma \subset \tau.$$

Then we can conclude that any tuple not contained in the database state is not true.

The more general case of Horn clauses is really no problem, either. With this choice for Ψ we assume the negation of any fact that is not implied by the database state. A typical application of Horn clauses is the computation of the transitive closure of a relation:

$$\begin{array}{ll}\{ & \mathrm{direct\_part(spoke, wheel)}, \\ & \mathrm{direct\_part(wheel, bicycle)}, \\ & \mathrm{part(x, y)} \leftarrow \mathrm{direct\_part(x, y)}, \\ & \mathrm{part(x, z)} \leftarrow \mathrm{part(x, y)} \wedge \mathrm{direct\_part(y, z)} \quad \}.\end{array}$$

The reason why the situation is so simple in this case is that only definite information is given. So there is only one $<$-minimal Herbrand model and only one maximal Ψ-extension of a database state.

5.2 Disjunctive Information

Reiter's original CWA becomes inconsistent if disjunctive information is present. This is why Minker defined his GCWA. One such example is the following: Suppose that we know that a particular computer is faulty, since its power supply or its CPU is faulty, i.e. the following formula is stored in the database state Φ:

$$\text{faulty(power\_supply)} \vee \text{faulty(CPU)} \in \Phi.$$

Of course we cannot exclude that both parts are faulty, so that we have a real disjunction. We still need some sort of CWA because we want to conclude that parts not mentioned in this disjunction are not faulty. The GCWA correctly handles this case, so we again can select Ψ to be the set of negative ground literals.

There are other sorts of disjunctions which the GCWA does not handle correctly, e.g.:

$$\text{bloodtype(John, A)} \vee \text{bloodtype(John, 0)} \in \Phi.$$

Here we know that John can have only one of the two bloodtypes but not both. So we really mean an exclusive or, i.e. we want to assume implicitly:

$$\neg\text{bloodtype(John, A)} \vee \neg\text{bloodtype(John, 0)} \in \Psi.$$

This suggests to choose Ψ to be the set of negative ground clauses; then the CWA of Yahya and Henschen [YH85] results. It is a "stronger" CWA than the GCWA discussed above in the sense that any ground literal assumed by the GCWA will be assumed by this CWA, too. It can use the same order relation, though.

The above situation commonly arises when we want to store properties (e.g. bloodtype) of objects (e.g. persons) and we do not know the exact value but only some finite set containing the value. This situation is expressed in the database state Φ by a disjunction like

$$\text{property(object, value}_1) \vee \ldots \vee \text{property(object, value}_n).$$

5.3 Default Rules

In the context of incomplete information default rules are of great inportance, since they allow conclusions despite the fact that not every exceptional case can be excluded because of missing information. A typical example of this kind is the rule "Birds normally can fly". Now suppose all we know about Tweety is that it is a bird, but we do not know for sure that it is not an ostrich or another abnormal bird. This example can be formalized by using a relation "abnormal" to describe all exceptions to the rule:

$$\Phi = \{\text{flies(x)} \leftarrow \text{bird(x)} \wedge \neg\text{abnormal(x)}, \text{bird(Tweety)}\}.$$

This database state does not logically imply "flies(Tweety)" because Tweety could be an abnormal bird. The semantics of "abnormal", however, is that normally birds are normal, and this semantics needs to be included by means of implicit assumptions. So we want to conclude from a completed database that Tweety flies. On the contrary, the CWAs considered so far do not allow to generate any new positive ground information like flies(Tweety). The reason is that the formula may be written equivalently as

$$\text{flies(x)} \vee \text{abnormal(x)} \leftarrow \text{bird(x)}.$$

So we have written a disjunction, but we do not mean usual disjunctive information. We obtain two maximal extensions, one which assumes that Tweety is not abnormal and another which assumes that Tweety cannot fly, but these two assumptions exclude each other, so that none of them is taken into the actual completion. The first extension, however, would have been the right one; this suggests to delete ¬flies(Tweety) from Ψ.

In the general case we mark some relation symbols (such as "flies" in the previous example) as "variables" in the completion process[4] of the other relation symbols (such as "abnormal"), i.e. we want to minimize the set of abnormal birds, but we do not want to minimize the set of flying birds. If the set of variable relation symbols is called Z and the other relation symbols are called P, we choose $\Psi = P^-$ (negative ground literals with relation symbols from P only), and we do not assume anything special about the relation symbols from Z. In the example above, ¬abnormal(Tweety) is a possible assumption and is included into the completion (since there is no contradicting possible assumption or database information); thus we get flies(Tweety) as a logical consequence.

Note that there is always a tradeoff between completion and database state. Instead of putting this default rule into the state we could have used possible assumptions of the form flies(c) ← bird(c).

5.4 Open Relations

As our running example in this subsection and the next we will use a database about the place of residence for some people:

$$\Phi = \{\text{residence}(\text{Udo}, \text{Dortmund}), \text{residence}(\text{Stefan}, \text{Braunschweig}), \ldots\}.$$

In this subsection this list should be so incomplete that we cannot assume anything about the place of residence of somebody not mentioned in this list. The most simple way to handle such incomplete information is to use no completion at all! This case is of course included in our definition ($\Psi = \emptyset$). It is the "open world assumption": Do not assume anything but only use logical consequences of the database state.

A problem arises when we want to use the completion only for some relations, but the relations are interconnected through rules:

$$\text{drinks\_beer}(\text{x}) \leftarrow \text{residence}(\text{x}, \text{Dortmund}) \;\in\; \Phi.$$

If we minimize, e.g., the set of people drinking beer, i.e. we include ¬drinks_beer(p) into Ψ for any person p, but we do not know the residence of a person, say Peter, we can conclude ¬drinks_beer(Peter), because there is no evidence to the contrary, and thus ¬residence(Peter, Dortmund). This is probably not desired.

It helps to look at theorem 21 once more: We can avoid any not wanted new information by considering its negation a possible assumption. So if we do not want to get any new ground literal information about a set Q of relation symbols like, e.g., "residence" we have to include $Q^+ \cup Q^-$ into Ψ. Intuitively, the set Ψ does not only contain possible assumptions which are meant to become an actual assumption in some completed state, but also assumptions about possible states of the world (e.g., Peter possibly lives in Dortmund). They must be considered in order to discover possible inconsistencies.

At this point we have rediscovered the CWA-versions from [GP86] and [GPP86] called "careful closure" and "Extended CWA". These are CWAs which allow the relation symbols to be partitioned into "variables" Z, "normal (to be minimized)" relation symbols P and "open world" relation symbols Q. We get the first version [GP86] using

$$\Psi = P^- \cup Q^+ \cup Q^-$$

and the second stronger version [GPP86] using ground clauses consisting of these ground literals (again making "or" to "exclusive or")[5]. The corresponding order relation is for both cases:

$$\sigma < \tau \iff \sigma \cap P^+ \subset \tau \cap P^+ \text{ and } \sigma \cap Q^+ = \tau \cap Q^+.$$

[4]i.e. the process of completing relations by negative information

[5]To be precise, our approach does not lead to the same, but to equivalently completed database states.

But there is more to discover about an "open world"-relation such as "residence": Probably we would like to conclude that Stefan does not live in Dortmund, because we know that he lives in Braunschweig and it is unlikely that a person has two places of residence. This can be achieved using possible assumptions of the form

$$\neg\text{residence}(p, c_1) \vee \neg\text{residence}(p, c_2) \in \Psi$$

for all persons p and all pairs c_1, c_2 of different cities. Note that we will not get a contradiction if we allow some people to have two places of residence.

5.5 Exceptions and Null Values

So let us now assume that our list of residences is nearly complete, i.e. there are only few persons left whose residence is unknown. We want to assume that any person p does not live in city c as long as the opposite is not deducible from the database, with the exception of those few people.

One means for handling such a form of incomplete information is the protected Circumscription defined by Minker and Perlis ([MP84], [MP85]). They specify exceptional arguments for a relation r like "residence" in another database relation Er and then exclude the assumptions about r for such arguments. So the possible assumptions in our example must be formulae of the following three forms:

$$\neg\text{residence}(p, c)$$
$$\neg E\text{residence}(p, c)$$
$$\text{residence}(p, c) \leftarrow E\text{residence}(p, c)$$

for any person p and any city c. If we do not know that a pair (p, c) of arguments is exceptional in a database state, then all three assumptions become actual so that $\neg$residence(p, c) can be concluded. But if we do know (p, c) to be exceptional, then the precondition of the third formula is satisfied and the consequence contradicts the first formula, so that they exclude each other. This means that we end up in assuming nothing about (p, c) as desired.

A corresponding order relation can be defined as follows[6]:

$$\sigma < \tau \iff \sigma \subset \tau \text{ and } \sigma[r] \cap \sigma[Er] = \tau[r] \cap \sigma[Er] \text{ for all relation symbols } r.$$

The preceeding example can also be seen as an application of null values; if we read

$$E\text{residence}(p, \text{x}) \iff \text{residence}(p, \text{—}).$$

They have been interpreted in such a way that we could not conclude that one of the exceptions has a residence, i.e. not only their place of residence is unknown, but also the question whether they have a place of residence or not. So if we would define another relation by

$$\text{has\_residence}(\text{x}) \leftarrow \text{residence}(\text{x}, \text{y})$$

we will not get an answer "yes" or "no" for those persons, i.e. "—" stands for "unknown or inapplicable".

This is not acceptable if we require that each person has at least one residence. The most direct way to handle such null values is to create a new constant $—_i$ $(i = 1, 2, \ldots)$ for any occurence of a null value. To denote that Peter's place of residence is unknown, we will write

$$\text{residence}(\text{Peter}, —_1).$$

[6]Note that we intersect with $\sigma[Er]$ on both sides of the equation. Our order relation allows the minimization of the relation Er. This is done in the answer algorithms in [MP85], too, but is a slight difference to the definition of the order relation in [MP84].

To exclude that the closed world assumption generates formulae like $\neg\text{residence}(\text{Peter}, c)$ we again have to use some assumptions about possible worlds to cancel the unwanted assumptions; in our example:

$$\neg\text{residence}(p, c)$$
$$\text{residence}(p, c) \leftarrow \text{residence}(p, —_i)$$

for any person p and any (not null) city c.

A more direct choice for Ψ would use equalities of the form $c = —_i$ instead, but this would require to incorporate an equality theory into the database states.

6 Conclusions

This paper has presented a unifying framework for closed world assumptions. Our parameterized definition allows to derive known proposals from the literature as special cases and to adapt the "closed world" idea to other kinds of information to be stored in a logic database. Of course, practical applications will not show the typical patterns discussed in section 5 in a pure form, so that the suggested choices for the parameter have to be combined. Thus there cannot be any standard implicit database completion, but at least the CWA parameter should still be an explicit part of database specifications.

Although closed world assumptions seem to be very natural and have been widely accepted in the database field, they are not a necessary choice of database completion, either. Usually they are restricted to formulae equivalent to sets of clauses (on the syntactic side) and to Herbrand models (on the semantic side). The latter restriction prevents a satisfactory treatment of arbitrary queries with variables if no additional axioms are assumed. These are the "unique name axiom" (UNA: different names denote different objects) and the "domain closure axiom" (DCA: all objects are denoted by constants). Whereas the first one is quite reasonable, the other one forbids function symbols in the formulae. Other authors have already discussed several different completions based on arbitrary minimal models; if the axioms above hold, some of them correspond to certain CWA versions. Minimal model completions seem a promising way to overcome the mentioned limits, but they need a unification perhaps comparable to the CWA parameterization, too. This will be subject of our further research.

Another kind of database completion is Reiter's Default Reasoning: [Rei80]. In the special case of CWA-defaults there are some connections to instances of our definition which should be further investigated. In paticular our definition seems to simplify a characterization of this special form of default reasoning which was given in [BH86].

In current work the class of stratified databases has attracted considerable interest (see part I of [Min88]). We expect that such approaches can be treated within a further generalization of our framework by considering ordered sets of assumptions.

Acknowledgement

We would like to thank Bernhard Convent for helpful comments on an earlier version of this paper and Hans-Dieter Ehrich for supervising [Bra88].

References

[BH86] H. Bidoit and R. Hull. Positivism vs. minimalism in deductive databases. In *Proc. of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS)*, 123–132, 1986.

[Bra88] S. Brass. Vervollständigungen für Logikdatenbanken (completions of logic databases). Diploma Thesis, Informatics, Techn. Univ. Braunschweig, 1988. In German.

[End72] H. B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York, 1972.

[GMN84] H. Gallaire, J. Minker, and J.-M. Nicolas. Logic and databases: A deductive approach. *Computing Surveys (16)*, 153–185, 1984.

[GP86] M. Gelfond and H. Przymusinska. Negation as failure: Careful closure procedure. *Artificial Intelligence (30)*, 273–287, 1986.

[GPP86] M. Gelfond, H. Przymusinska, and T. Przymusinski. The extended closed world assumption and its relationship to parallel circumscription. In *Proc. of the Fifth ACM SIGACT-SIGMOD Symposium on Principles of Database Systems (PODS)*, 153–185, 1986.

[Men78] E. Mendelson. *Introduction to Mathematical Logic*. Van Nostrand-Reinhold, New York, second edition, 1978.

[Min82] J. Minker. On indefinite databases and the closed world assumption. In D. W. Loveland, editor, *6th Conference on Automated Deduction*, 292–308, Springer-Verlag, Berlin, 1982. Lecture Notes in Computer Science No.138.

[Min88] J. Minker, editor. *Foundations of Deductive Databases and Logic Programming*, Morgan Kaufmann Publishers, Los-Altos (Calif.), 1988.

[MP84] J. Minker and D. Perlis. Applications of protected circumscription. In R. E. Shostak, editor, *7th International Conference on Automated Deduction*, 414–425, Springer-Verlag, Berlin, 1984. Lecture Notes in Computer Science No.170.

[MP85] J. Minker and D. Perlis. Computing protected circumscription. *The Journal of Logic Programming (2)*, 235–249, 1985.

[Rei78] R. Reiter. On closed world data bases. In H. Gallaire and J. Minker, editors, *Logic and Data Bases*, 55–76, Plenum, New York, 1978.

[Rei80] R. Reiter. A logic for default reasoning. *Artificial Intelligence (13)*, 81–132, 1980.

[She88] J. C. Sheperdson. Negation in logic programming. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, 19–88, Morgan Kaufmann Publishers, Los-Altos (Calif.), 1988.

[YH85] A. Yahya and L. J. Henschen. Deduction in non-horn databases. *Journal of Automated Reasoning (1)*, 141–160, 1985.