TECHNISCHE UNIVERSITÄT DRESDEN

Faculty of Computer Science | Institute of Artificial Intelligence | Knowledge Representation and Reasoning

Modeling the Suppression Task under Weak Completion and Well-Founded Semantics

Emmanuelle-Anna Dietz Steffen Hölldobler Christoph Wernhard.

KRR Report 13-02

Mail to
Technische Universität Dresden
01062 Dresden

Bulk mail to
Technische Universität Dresden
Helmholtzstr. 10
01069 Dresden

Office
Room 2006
Nöthnitzer Straße 46
01187 Dresden

Internet
http://www.wv.inf.tu-dresden.de

Modeling the Suppression Task under Weak Completion and Well-Founded Semantics

Emmanuelle-Anna Dietz,* Steffen Hölldobler,** Christoph Wernhard***

International Center for Computational Logic, Technische Universität Dresden, D-01062 Dresden, Germany

Abstract: Formal approaches that aim at representing human reasoning should be evaluated based on how humans actually reason. One way in doing so, is to investigate whether psychological findings of human reasoning patterns are represented in the theoretical model. The computational logic approach discussed here is the so called weak completion semantics which is based on the three-valued Łukasiewicz logic. We explain how this approach adequately models Byrne's suppression task, a psychological study where the experimental results show that participants' conclusions systematically deviate from the classical logically correct answers. As weak completion semantics is a novel technique in the field of Computational Logic, it is important to examine how it corresponds to other already established non-monotonic approaches. For this purpose we investigate the relation of weak completion with respect to completion and three-valued stable model semantics. In particular, we show that well-founded semantics, a widely accepted approach in the field of non-monotonic reasoning, corresponds to weak completion semantics for a specific class of modified programs.

1. Introduction

Byrne's *suppression task* (Byrne, 1989) is a psychological study showing that people with no previous exposure to formal logic suppress previously drawn conclusions when additional information becomes available. Consider the following example: *If she has an essay to write, then she will study late in the library* and *She has an essay to write.* Most subjects (96%) conclude: *She will study late in the library.* If subjects, however, receive an additional conditional: *If the library stays open, she will study late in the library*, then only a minority (38%) concludes: *She will study late in the library.* This shows that conclusions which are correct with respect to classical logic can be suppressed in human reasoning by the presence of an additional conditional, and thus provides an excellent example for the human capability to draw *non-monotonic* inferences. In the complete experiment of Byrne, participants received the following three conditionals:

Simple:	*If she has an essay to write, then she will study late in the library.*
Alternative:	*If she has a textbook to read, then she will study late in the library.*
Additional:	*If the library stays open, then she will study late in the library.*

*dietz@iccl.tu-dresden.de
**sh@iccl.tu-dresden.de
***christoph.wernhard@tu-dresden.de

Conclusion	Given Fact	Group I	Group II	Group III
l	e	96%	96%	38%
$\neg l$	$\neg e$	46%	4%	63%
e	l	53%	16%	55%
$\neg e$	$\neg l$	69%	69%	44%

Table 1. Empirical results about suppression obtained by Byrne (1989).

The participants were divided into three groups: Group I received the SIMPLE conditional, group II the SIMPLE and the ALTERNATIVE conditional, and group III the SIMPLE and the ADDITIONAL conditional. In addition, the participants received a positive or negative fact, and were asked whether they conclude from the given conditionals and a given fact a further given fact. The positive and negative facts involved in the experiments are as follows:

e: *She has an essay to write.*
$\neg e$: *She does not have an essay to write.*
l: *She will study late in the library.*
$\neg l$: *She will not study late in the library.*

Table 1 gives an overview on the empirical results by Byrne (1989) about the suppression task. Percentages indicate the proportion of subjects in each of the groups that have drawn the respective conclusion from the indicated given fact and the conditionals. Where suppression took effect, the propositions are highlighted in bold. Similar results have been obtained by other researchers, see for example Dieussaert et al. (2000).

In investigations into human reasoning over the past decades, classical (propositional) logic has often played the role of a normative concept. However, empirical research suggests that humans systematically deviate from the classically correct answers, which is sometimes used as an argument against the usefulness of logic in the area of human reasoning. We do not follow this argument, but strive to model human reasoning – including its systematic deviations from "classical correctness" – with techniques from the field of Computational Logic, in particular, non-monotonic reasoning and three-valued semantics.

Just modeling is not satisfying: Strube (1992) argues that knowledge engineering should also aim at being *cognitively adequate*. Accordingly, when evaluating computational approaches which try to explain human reasoning we insist on assessing their cognitive adequacy. Strube distinguishes between *weak* and *strong* cognitive adequacy: Weak cognitive adequacy requires the system to be ergonomic and user-friendly, whereas strong cognitive adequacy involves an exact model of human knowledge and reasoning mechanisms that follows the relevant human cognitive processes. Knauff et al. (1997, 1995) define cognitive adequacy in the setting of qualitative spatial reasoning, where the authors distinguish between *conceptual* cognitive adequacy and *inferential* cognitive adequacy: Degrees of conceptual cognitive adequacy reflects to which extent a system corresponds to human conceptual knowledge. Inferential cognitive adequacy focuses on the procedural part and indicates whether the reasoning process of a system is structured similarly to the way humans reason. There seems to be a correspondence between these definitions and the proposition made by Stenning & van Lambalgen (2005, 2008) to model human reasoning by a two step process: Firstly, human reasoning should be modeled by setting up an appropriate representation (conceptual cognitive adequacy) and, secondly, the *reasoning process* should be modeled with respect to this representation (inferential cognitive adequacy).

It is straightforward to see that classical logic cannot model the suppression task adequately. At least some kind of non-monotonicity is needed. As appropriate representation to model the suppression task, Stenning & van Lambalgen (2005, 2008) propose logic programs under completion semantics based on the three-valued logic used by Fitting (1985), which itself is based on the three-valued logic by Kleene (1952). Unfortunately, some technical claims made by Stenning & van Lambalgen are wrong. Hölldobler & Kencana Ramli (2009a,b) have shown that the three-valued logic proposed by Stenning & van Lambalgen is inadequate for the suppression task, but that the suppression task can be adequately modeled if the three-valued logic by Łukasiewicz (1920) is used instead. The computational logic approach in (Dietz et al., 2012; Hölldobler & Kencana Ramli, 2009b) models the suppression task by means of logic programs under the so-called *weak completion semantics*, a variation of Clark's completion. They show that the conclusions drawn with respect to least models correspond to the findings by Byrne (1989) and conclude that the derived logic programs under Łukasiewicz logic are inferentially cognitively adequate for the suppression task. Wernhard (2011, 2012) discusses the application of different logic programming semantics to model human reasoning tasks according to the approach by Stenning & van Lambalgen and the roles of three-valuedness in this context in a different technical framework based on circumscription.

In this paper we focus on how weak completion semantics relates to other well-established non-monotonic logic approaches. As often described in the literature, most approaches differ in how logic programs behave with respect to cycles. A program is said to contain a cycle when at least one atom depends on itself, in the following sense: For all clauses of the form $p \leftarrow q_1 \wedge \ldots \wedge q_m \wedge \neg r_1 \wedge \ldots \wedge \neg r_n$ occurring in a program, the head atom p *depends* on all atoms occurring in the body, that is, on $q_1, \ldots, q_m, r_1, \ldots, r_n$. In addition, the *depends* is transitive. Consider the following example, adapted from Przymusinski (1994):

$$\begin{aligned} \mathcal{P}_{\mathit{fly}} &= \{\mathit{fly} \leftarrow \mathit{bird} \wedge \neg \mathit{abnormal},\ \mathit{bird}\}, \\ \mathcal{P}_{\mathit{cycle}} &= \{\mathit{abnormal} \leftarrow \mathit{irregular},\ \mathit{irregular} \leftarrow \mathit{abnormal}\}. \end{aligned}$$

The program $\mathcal{P}_{\mathit{cycle}}$ contains two cycles because *abnormal* and *irregular* depend on themselves. Przymusinski (1994) shows that programs with cycles might not reflect intuitive interpretations. For instance, under Clark's completion semantics (Clark, 1978) we can conclude *fly* from $\mathcal{P}_{\mathit{fly}}$. However, if we extend $\mathcal{P}_{\mathit{fly}}$ with $\mathcal{P}_{\mathit{cycle}}$, we cannot conclude *fly* anymore. This seems to be counterintuitive. Moreover, under the completion semantics as well as the stable model semantics (Gelfond & Lifschitz, 1988), cycles established through an odd number of negated atom occurrences can lead to inconsistency, that is, to programs which do not have a model: A program containing a clause $p \leftarrow \neg p$ does not have a stable model and the completion of this clause, $p \leftrightarrow \neg p$, is inconsistent.

A solution to these problems is to consider three-valued interpretations instead of total (two-valued) interpretations. Przymusinski (1990) proposed three-valued stable model semantics, also known as partial stable model semantics, an extension of stable model semantics. Under three-valued stable model semantics the program $\{p \leftarrow \neg p\}$ has a unique three-valued model in which p is *unknown*. If we extend $\mathcal{P}_{\mathit{fly}}$ with

$$\mathcal{P}_{\mathit{neg\text{-}cycle}} = \{\mathit{abnormal} \leftarrow \neg \mathit{regular},\ \mathit{regular} \leftarrow \neg \mathit{abnormal}\},$$

then we do not obtain just a single unique three-valued stable model but three three-valued stable models: one model where *fly*, *bird* and *regular* are true whereas *abnormal* is false, another one where *bird* and *abnormal* are true whereas *fly* and *regular* are false, and finally one where *bird* is true and all other atoms are unknown. The challenge is to find the model that corresponds most likely to the model a human would generate in a certain commonsense setting, rather than the perfect model in a purely logical context.

Well-founded semantics introduced by Van Gelder et al. (1991) is a widely accepted approach in the field of non-monotonic reasoning and is one step towards this direction. Compared to Clark's completion or stable model semantics, well-founded semantics is considered to be more accurate for programs with positive or negative cycles (Przymusinski, 1994). For instance, in the well-founded model of $\mathcal{P}_{neg\text{-}cycle}$ *abnormal* and *regular* are unknown and in the well-founded model of $\mathcal{P}_{cycle}$ *abnormal* and *irregular* are false. Under completion semantics there does not even exist a model of $\mathcal{P}_{neg\text{-}cycle}$. Atoms involved in positive cycles are false whereas atoms involved in negative cycles stay unknown. The idea behind this distinction is that the negation of *abnormal* or *irregular* shall not support the truth of any other element in the program. For instance, if *abnormal* or *regular* would be false in the well-founded model of $\mathcal{P}_{neg\text{-}cycle}$, they would be misleading for further positive conclusions and generate inconsistency. The least three-valued stable model coincides with the well-founded model (Przymusinski, 1990).

As well-founded semantics is a well-established approach in the literature, our main question of this paper is how does it relate to weak completion semantics? What are the similarities and where do they differ? Can both approaches adequately represent the suppression task?

The rest of this paper is structured as follows: In the following Section 2, we provide the necessary definitions about logic programs, interpretations and models under weak completion semantics. After that, we briefly review three-valued logics. Section 3 introduces the logic programs representing the suppression task as modeled by Stenning & van Lambalgen. We explain how the least models of the weak completion are computed and outline how abduction can be applied to model instances of the suppression task that involve backward reasoning. In Section 4 we first recapitulate other three-valued approaches to logic programming from the literature, the three-valued stable model semantics and the well-founded semantics, and discuss specific restrictions of programs with respect to circular dependency. We proceed by developing technical tools to compare three-valued logic programming semantics by adapting notions that are known for two-valued logic programming semantics such as supportedness and well-supportedness to three-valued settings. Section 5 presents the main technical results of this paper and shows how three-valued stable model semantics and weak completion semantics relate to each other. Moreover, we show that there is a strong correspondence to well-founded semantics. Section 6 reviews the suppression task in the light of the different investigated semantics. We conclude in Section 7 with sketching further experiments that seem suited to compare logic programming semantics with respect to their adequacy to model human reasoning.

2. Preliminaries

We define the necessary notations we will use throughout this paper and restrict ourselves to propositional logic as this is sufficient for the purpose of this paper. We assume a fixed set of *atoms*, denoted by ATOMS, that is nonempty and finite. *Formulas* are constructed from atoms, the truth-value constants $\top$, $\bot$ and U for *true*, *false* and *unknown*, the unary operator $\neg$ for negation, the binary connectives $\wedge, \vee$ for conjunction and disjunction, as well as the binary connectives $\leftarrow, \leftrightarrow$ for implication and equivalence. As meta-level notation we use n-ary versions of conjunction and disjunction. If A is an atom, then A and $\neg A$ are *literals*, the *positive literal* and *negative literal*, respectively literals with atom A. We call an implication whose left side is an atom a *clause*. A *program* is a finite set of clauses that have one of the following two particular forms:

$$A \leftarrow L_1 \wedge \ldots \wedge L_n, \text{ where } n \geq 0, \tag{1}$$

$$A \leftarrow \bot, \tag{2}$$

where A is an atom and the L_i with $1 \leq i \leq n$ are literals. Notice that in case $n = 0$ the right side of a clause of form (1) is $\top$. The atom A is called the *head* of the clause and the subformula to the right of the implication sign is called the *body* of the clause. Clauses of the form $A \leftarrow \top$ are called *positive facts*, whereas clauses of the form $A \leftarrow \bot$ are called *negative facts*.[1] To refer to the positive and negative part of a body, we introduce the following notation: If F is a conjunction of literals, then $\mathsf{pos}(F)$ ($\mathsf{neg}(F)$, resp.) denotes the conjunction of all positive (negative, resp.) literals in F. To let pos and neg also be applicable to bodies of negative facts, we define additionally $\mathsf{pos}(\bot) := \mathsf{neg}(\bot) := \top$.

If $\mathcal{P}$ is a program, then $\mathsf{atoms}(\mathcal{P})$ denotes the set of all atoms occurring in $\mathcal{P}$. The set of all clauses with head A in $\mathcal{P}$ is called the *definition* of A in $\mathcal{P}$. If this set is nonempty, the atom A is called *defined* in $\mathcal{P}$, otherwise A is called *undefined* in $\mathcal{P}$. The set of all atoms that are defined in $\mathcal{P}$ is denoted by $\mathsf{def}(\mathcal{P})$. The set of all atoms that are undefined in $\mathcal{P}$, that is, $\mathsf{ATOMS} \setminus \mathsf{def}(\mathcal{P})$, is denoted by $\mathsf{undef}(\mathcal{P})$.

A *normal* program – in the standard sense used in the literature on logic programming – is a program that does not contain negative facts, that is, a program whose clauses are all of the form (1). If $\mathcal{P}$ is a program, then $\mathcal{P}^+$ denotes the normal program obtained from $\mathcal{P}$ by deleting all negative facts. Obviously, for normal programs $\mathcal{P}$ it holds that $\mathcal{P} = \mathcal{P}^+$.

2.1 *Interpretations and Models*

An *interpretation* I is a mapping from the set of formulas to the set of truth values $\{\top, \bot, \mathsf{U}\}$, where $\top$ means *true*, $\bot$ means *false* and U means *unknown*. The truth value of a given formula under a given interpretation is determined according to the corresponding logic, as presented in the following section. We represent an interpretation as a pair $I = \langle I^\top, I^\bot \rangle$ of disjoint sets of atoms where $I^\top$ is the set of all atoms that are mapped by I to $\top$ and $I^\bot$ is the set of all atoms that are mapped by I to $\bot$. Atoms which do not occur in $I^\top \cup I^\bot$ are mapped to U.

There are two common ways to order three-valued interpretations, which, following Ruiz & Minker (1995), we call *truth-ordering* ($\preceq_t$) and *knowledge-ordering* ($\preceq_k$): For interpretations I and J we define $I \preceq_t J$ if and only if $I^\top \subseteq J^\top$ and $J^\bot \subseteq I^\bot$, whereas $I \preceq_k J$ if and only if $I^\top \subseteq J^\top$ and $I^\bot \subseteq J^\bot$.

A *model* of a formula F is an interpretation I such that $I(F) = \top$. A *model* of a set of formulas is an interpretation that is a model of each formula in the set. Models that are minimal with respect to the truth- or knowledge-ordering are called *truth-* or *knowledge-minimal models*, respectively. Likewise, models which are least with respect to the truth- or knowledge-ordering are called *truth-* or *knowledge-least models*.

2.2 *The Weak Completion of a Program*

When mechanisms of non-monotonic reasoning are applied to model human reasoning, it seems essential that only certain atoms are subjected to the closed world assumption, while others are considered to follow the open world assumption. *Weak completion* is

[1] This notion of falsehood appears to be counterintuitive at first sight, but programs will be interpreted under (weak) completion semantics where the implication sign is replaced by an equivalence sign.

F	$\neg F$
$\top$	$\bot$
$\bot$	$\top$
U	U

$\wedge$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	$\bot$
$\bot$	$\bot$	$\bot$	$\bot$

$\leftarrow_Ł$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	$\top$	$\top$
$\bot$	$\bot$	U	$\top$

$\leftarrow_S$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$	$\top$

$\leftarrow_K$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	U	U	$\top$
$\bot$	$\bot$	U	$\top$

$\vee$	$\top$	U	$\bot$
$\top$	$\top$	$\top$	$\top$
U	$\top$	U	U
$\bot$	$\top$	U	$\bot$

$\leftrightarrow_Ł$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	$\top$	U
$\bot$	$\bot$	U	$\top$

$\leftrightarrow_S$	$\top$	U	$\bot$
$\top$	$\top$	$\bot$	$\bot$
U	$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$	$\top$

$\leftrightarrow_K$	$\top$	U	$\bot$
$\top$	$\top$	U	$\bot$
U	U	U	U
$\bot$	$\bot$	U	$\top$

Table 2. Truth tables for three-valued logics. The $\top$'s highlighted in gray indicate that formulas of the form $A \leftarrow B$ which are true under $\leftarrow_Ł$ are true under $\leftarrow_S$, and vice versa.

a technique that allows both types of predicates to interact within a logic program. Consider the following transformation for a given program $\mathcal{P}$:

(1) Replace all clauses with the same head $A \leftarrow body_1,\ A \leftarrow body_2,\ \ldots,\ A \leftarrow body_n$, where $n \geq 1$, by $A \leftarrow body_1 \vee body_2 \vee \ldots \vee body_n$.
(2) For all $A \in \mathsf{undef}(\mathcal{P})$ add $A \leftarrow \bot$.
(3) Replace all occurrences of $\leftarrow$ by $\leftrightarrow$.

The resulting set of equivalences is the well-known Clark's *completion* (Clark, 1978) of $\mathcal{P}$, denoted by $\mathsf{c}\,\mathcal{P}$. If step 2 is omitted, then the resulting set is the *weak completion* of $\mathcal{P}$, denoted by $\mathsf{wc}\,\mathcal{P}$ (Hölldobler & Kencana Ramli, 2009b). Consider, for example, the program $\mathcal{P} = \{p \leftarrow q,\ p \leftarrow r,\ q \leftarrow \bot\}$. We have $\mathsf{def}(\mathcal{P}) = \{p, q\}$ and $r \in \mathsf{undef}(\mathcal{P}) = \mathsf{ATOMS} \setminus \{p, q\}$. Then $\mathsf{c}\,\mathcal{P} = \{p \leftrightarrow q \vee r,\ q \leftrightarrow \bot\} \cup \{A \leftrightarrow \bot \mid A \in \mathsf{undef}(\mathcal{P})\}$, where all atoms adhere to the closed world assumption. On the other hand, $\mathsf{wc}\,\mathcal{P} = \{p \leftrightarrow q \vee r,\ \ q \leftrightarrow \bot\}$, where only the defined atoms p and q adhere to the closed world assumption.

2.3 *Three-Valued Logics*

Since the first modern three-valued logic has been invented by Łukasiewicz (1920), various different interpretations of the three-valued connectives were proposed. Table 2 gives some quite common truth tables for negation, conjunction and disjunction. For implication and equivalence it shows different versions: Kleene (1952) introduced the implication ($\leftarrow_K$), whose truth table is identical to Łukasiewicz implication ($\leftarrow_Ł$) except in the case where precondition and conclusion are both mapped to U: In this case, the value of $\leftarrow_K$ is U, whereas the value of $\leftarrow_Ł$ is $\top$. The further common variant $\leftarrow_S$ of three-valued implication is called seq_3 by Gottwald (2001). The displayed versions of equivalence ($\leftrightarrow_Ł$, $\leftrightarrow_S$, $\leftrightarrow_K$) are derived by conjoining the respective implications with flipped arguments. If we understand operators in a formula with the meaning specified in Table 2 for $\{\neg, \wedge, \vee, \leftarrow_Ł, \leftrightarrow_Ł\}$, we say that we consider the formula *under Ł-semantics*. Fitting (1985) combined the truth tables for $\neg$, $\vee$, $\wedge$ from Łukasiewicz with the equivalence $\leftrightarrow_S$ for investigations within logic programming. The set of connectives Fitting used is $\{\neg, \wedge, \vee, \leftrightarrow_S\}$.

One should observe that in contrast to two-valued logic, $A \leftarrow B$ and $A \vee \neg B$ are not always semantically equivalent, neither for $\leftarrow_Ł$ nor for $\leftarrow_S$. Consider, for instance, an interpretation I such that $I(A) = I(B) = \mathsf{U}$. Then, $I(A \vee \neg B) = \mathsf{U}$ whereas $I(A \leftarrow_Ł B) = I(A \leftarrow_S B) = \top$. However, for the $\leftarrow_K$ implication we have that both $I(A \vee \neg B) = \mathsf{U}$ and $I(A \leftarrow_K B) = \mathsf{U}$.

Semantics		Set of Connectives				
Fitting	(F)	$\neg$	$\wedge$	$\vee$		$\leftrightarrow_S$
Kleene	(K)	$\neg$	$\wedge$	$\vee$	$\leftarrow_K$	
Ł-semantics	(Ł)	$\neg$	$\wedge$	$\vee$	$\leftarrow_Ł$	$\leftrightarrow_Ł$
S-semantics	(S)	$\neg$	$\wedge$	$\vee$	$\leftarrow_S$	$\leftrightarrow_S$
SvL-semantics	(SvL)	$\neg$	$\wedge$	$\vee$	$\leftarrow_K$	$\leftrightarrow_S$

Table 3. Overview of the three-valued semantics with corresponding set of connectives.

Stenning & van Lambalgen (2008) suggested to model the suppression task by extending the logic used by Fitting with $\leftarrow_K$. If we understand operators in this way, that is, with the meanings of $\{\neg, \wedge, \vee, \leftarrow_K, \leftrightarrow_S\}$, we call this *SvL-semantics.* Hölldobler & Kencana Ramli (2009b) showed that SvL-semantics leads to technical errors. They proposed to use Ł-semantics (cf. 2), which corrects these and allows to adequately model the suppression task. The erroneous effects of the original suggestion by Stenning & van Lambalgen (2008) will be demonstrated by two examples in Section 3.2. Under well-founded semantics, which we will discuss later, the interpretation of the implication can be modeled by $\leftarrow_S$ (Przymusinski, 1989), which corresponds to the three-valued logic S_3 (Rescher, 1969), that is, $\{\neg, \wedge, \vee, \leftarrow_S, \leftrightarrow_S\}$. If we understand operators in a formula with these meanings, we say that we consider *S-semantics.*

As indicated by the highlighted $\top$ signs in Table 2, whenever a formula is true under $\leftarrow_S$ then it is true under $\leftarrow_Ł$, and vice versa. Similarly, the cases where $\leftrightarrow_Ł$ and $\leftrightarrow_S$ have the value $\top$ coincide. From this follows that the models of a program or a set of equivalences obtained by completing a program are under S-semantics exactly the same as under Ł-semantics. Table 3 gives an overview of the three-valued semantics with the corresponding semantics of the set of connectives.

3. Modeling the Suppression Task

To model the suppression task, we follow the two-step approach by Stenning & van Lambalgen (2005, 2008). In this section, we discuss these steps, together with abduction, which we apply to model human reasoning in the "backward direction", as in those experiments by Byrne where it is investigated whether e (*She has an essay to write*) or $\neg e$ (*She does not have an essay to write*) is concluded.

3.1 *Reasoning Towards an Appropriate Logical Form*

In the model of Stenning & van Lambalgen (2005, 2008), the first step of human reasoning is reasoning towards an appropriate representation. *Conceptual* cognitive adequacy is the goal of the model with respect to this step. In particular, Stenning & van Lambalgen argue that conditionals shall not be encoded by inferences straight away, but rather by *licenses* for inference. For example, the SIMPLE conditional *If she has an essay to write, then she will study late in the library* should be encoded by the clause $l \leftarrow e \wedge \neg ab_1$, where ab_1 is an *abnormality predicate* expressing that something abnormal is known. In other words, l holds if e is true and nothing abnormal is known.

Table 4 shows the representational form of the first part of the suppression task as modeled by Stenning & van Lambalgen. In the first three cases, in addition to the conditionals, the participants had to draw conclusions based on the fact that *She has an essay to write* ($e \leftarrow \top$). In the last three cases they had to draw conclusions based on the fact that *She does not have an essay to write* ($e \leftarrow \bot$). The predicates ab_1, ab_2 and ab_3 represent different kinds of abnormality. For instance, each of the programs $\mathcal{P}_{e+Alt}$

		Conditionals				Facts
$\mathcal{P}_e$	$=$	$\{l \leftarrow e \wedge \neg ab_1,$		$ab_1 \leftarrow \bot,$		$e \leftarrow \top\}$
$\mathcal{P}_{e+Alt}$	$=$	$\{l \leftarrow e \wedge \neg ab_1,$	$l \leftarrow t \wedge \neg ab_2,$	$ab_1 \leftarrow \bot,$	$ab_2 \leftarrow \bot,$	$e \leftarrow \top\}$
$\mathcal{P}_{e+Add}$	$=$	$\{l \leftarrow e \wedge \neg ab_1,$	$l \leftarrow o \wedge \neg ab_3,$	$ab_1 \leftarrow \neg o,$	$ab_3 \leftarrow \neg e,$	$e \leftarrow \top\}$
$\mathcal{P}_{\neg e}$	$=$	$\{l \leftarrow e \wedge \neg ab_1,$		$ab_1 \leftarrow \bot,$		$e \leftarrow \bot\}$
$\mathcal{P}_{\neg e+Alt}$	$=$	$\{l \leftarrow e \wedge \neg ab_1,$	$l \leftarrow t \wedge \neg ab_2,$	$ab_1 \leftarrow \bot,$	$ab_2 \leftarrow \bot,$	$e \leftarrow \bot\}$
$\mathcal{P}_{\neg e+Add}$	$=$	$\{l \leftarrow e \wedge \neg ab_1,$	$l \leftarrow o \wedge \neg ab_3,$	$ab_1 \leftarrow \neg o,$	$ab_3 \leftarrow e,$	$e \leftarrow \bot\}$

Table 4. Representational form of the "forward reasoning" instances of the suppression task according to Stenning & van Lambalgen (2008).

and $\mathcal{P}_{e+Add}$ contains two clauses with the conclusion l. The programs differ in that in $\mathcal{P}_{e+Alt}$ the premise of the second clause is an *alternative* to the first clause, whereas in $\mathcal{P}_{e+Add}$ the premise of the second clause is an *additional* to the first clause. That the second clause in $\mathcal{P}_{e+Add}$ ($l \leftarrow o \wedge \neg ab_3$) takes effect as an additional precondition for l is achieved by the clause stating that ab_1 is true when *The library does not stay open* ($ab_1 \leftarrow \neg o$) and the clause that states that ab_3 is true when *She does not have an essay to write* ($ab_3 \leftarrow \neg e$).

Adopting the programs obtained by Stenning & van Lambalgen as result of the first step of reasoning towards an appropriate representation, we will now focus on the second step, the inferential aspects.

3.2 *Reasoning with Respect to Least Models*

Under Ł-semantics, the weak completion of a logic program can have several models. Consider for example $\mathcal{P}_{e+Add}$ from Table 4. Its weak completion is:

$$\mathsf{wc}\,\mathcal{P}_{e+Add} = \{l \leftrightarrow (e \wedge \neg ab_1) \vee (o \wedge \neg ab_3),\ ab_1 \leftrightarrow \neg o,\ ab_3 \leftrightarrow \neg e,\ e \leftrightarrow \top\}.$$

The interpretations $\langle \{e, o\}, \{ab_1, ab_3, l\} \rangle$ and $\langle \{e\}, \{ab_3\} \rangle$ are both models of $\mathsf{wc}\,\mathcal{P}_{e+Add}$ under Ł-semantics. How to know which model is the intended one? In logic programming and computational logic the intended models are often least models, if they exist. Following Apt & van Emden (1982), least models of logic programs can often be specified as least fixed points of appropriate semantic operators.

As shown by Hölldobler & Kencana Ramli (2009b), the model intersection property holds for logic programs under Ł-semantics, i.e., $\bigcap\{I \mid I(\mathcal{P}) = \top\}(\mathcal{P}) = \top$, where the *intersection* $I \cap J$ of two interpretations $I = \langle I^\top, I^\bot \rangle$ and $J = \langle J^\top, J^\bot \rangle$ is defined as $\langle I^\top \cap J^\top, I^\bot \cap J^\bot \rangle$. Moreover, the model intersection property also holds for the weak completion of logic programs. This guarantees that each logic program has a least model. Additionally, the least model of the weak completion of a program $\mathcal{P}$ under Ł-semantics ($\mathsf{lm}_L\mathsf{wc}\,\mathcal{P}$) is identical to the least fixed point of the following semantic operator, Φ_{SvL}, which was introduced by Stenning & van Lambalgen (2008): Let J be the result of the application of Φ_{SvL} to an interpretation I and a logic program $\mathcal{P}$, denoted by $\Phi_{SvL,\mathcal{P}}(I)$. Then we define J as follows:

$$\begin{aligned} J^\top &= \{A \mid \text{there exists a clause } A \leftarrow \mathit{Body} \in \mathcal{P} \text{ with } I(\mathit{Body}) = \top\} \text{ and} \\ J^\bot &= \{A \mid \text{there exists a clause } A \leftarrow \mathit{Body} \in \mathcal{P} \text{ and} \\ &\qquad \text{for all clauses } A \leftarrow \mathit{Body} \in \mathcal{P} \text{ we find } I(\mathit{Body}) = \bot\}. \end{aligned}$$

Starting with the empty interpretation $I = \langle \emptyset, \emptyset \rangle$, $\mathsf{lm}_L\mathsf{wc}\,\mathcal{P}$ can be computed by iterating $\Phi_{SvL,\mathcal{P}}$. To illustrate this result consider $\mathcal{P}_{e+Add}$ and let $I_0 = \langle \emptyset, \emptyset \rangle$ in:

$$\begin{aligned} I_1 &= \Phi_{SvL,\mathcal{P}_{e+Add}}(I_0) = \langle \{e\}, \emptyset \rangle \\ I_2 &= \Phi_{SvL,\mathcal{P}_{e+Add}}(I_1) = \langle \{e\}, \{ab_3\} \rangle = \Phi_{SvL,\mathcal{P}_{e+Add}}(I_2) \end{aligned}$$

One should observe that $\langle \{e\}, \{ab_3\} \rangle$ is not a model of $\mathcal{P}_{e+Add}$ under SvL-semantics because the clause $l \leftarrow o \wedge ab_3 \in \mathcal{P}_{e+Add}$ is mapped to U under SvL-semantics and not to $\top$ as under Ł-semantics. This is a counterexample to Lemma 4 (1.) in (Stenning & van Lambalgen, 2008, p. 194f), which states that the least fixed point of the $\Phi_{SvL,\mathcal{P}}$ operator under SvL-semantics is the (knowledge-) minimal model of $\mathcal{P}$. Furthermore, Stenning & van Lambalgen (2008) claim in Lemma 4 (3.) that all models of $\mathsf{c}\,\mathcal{P}$ are fixed points of $\Phi_{SvL,\mathcal{P}}$ and every fixed point is a model. Consider the completion of $\mathcal{P}_{\neg e+Alt}$, i.e.,

$$\{l \leftrightarrow (e \wedge \neg ab_1) \vee (t \wedge \neg ab_2),\ ab_1 \leftrightarrow \bot,\ ab_2 \leftrightarrow \bot,\ e \leftrightarrow \bot,\ t \leftrightarrow \bot\}.$$

Then t and e are mapped to $\bot$ and, consequently, l is mapped to $\bot$ as well. However, the least fixed point of $\Phi_{SvL,\mathcal{P}_{\neg e+Alt}}$ is $\langle \emptyset, \{e, ab_1, ab_2\} \rangle$, where t and l are undefined. This example also shows that reasoning under SvL-semantics with respect to the completion of a program is not adequate, since, as shown in Table 1, only 4% of the subjects conclude $\neg l$ in this case.

Notice that the operator defined by Stenning & van Lambalgen (2008) differs in a subtle way from the well-known Fitting operator Φ_F, introduced in (Fitting, 1985): The definition of Φ_F is like that of Φ_{SvL}, except that in the specification of $J^\bot$ the first line *"there exists a clause* $A \leftarrow$ Body $\in \mathcal{P}$ *and"* is dropped. The least fixed point of $\Phi_{F,\mathcal{P}}$ corresponds to the least model of the completion of $\mathcal{P}$ under S-semantics, or equivalently under Ł-semantics, whereas the least fixed point of $\Phi_{SvL,\mathcal{P}}$ corresponds to the least model of the *weak* completion of $\mathcal{P}$ under these semantics. If an atom A is undefined in the program $\mathcal{P}$, then, for arbitrary interpretations I it holds that $A \in J^\bot$ in $\Phi_{F,\mathcal{P}}(I) = \langle J^\top, J^\bot \rangle$, whereas, if Φ_{SvL} is applied instead of Φ_F, this does not hold for any interpretation I.

3.3 *Backward Reasoning with Abduction*

In order to adequately model the "backward reasoning" instances of the suppression task, corresponding to the last two rows in Table 1, we need to introduce abduction. The objective of abduction is, given a knowledge base and an observation, to compute an explanation which, combined with the knowledge base, allows to infer the observation. Following the approach of Kakas et al. (1993), we consider as an *abductive framework* a triple $\langle \mathcal{P}, \mathcal{A}, \models_L^{\mathsf{lmwc}} \rangle$ consisting of a program $\mathcal{P}$, called the *knowledge base*, a set $\mathcal{A}$ of *abducibles* consisting of the (positive and negative) facts for each undefined atom in $\mathcal{P}$, and the consequence relation $\models_L^{\mathsf{lmwc}}$, defined for all formulas F as $\mathcal{P} \models_L^{\mathsf{lmwc}} F$ if and only if $\mathsf{lm}_L\mathsf{wc}\,\mathcal{P}(F) = \top$. As observations $\mathcal{O}$ we consider sets of literals. A set of facts $\mathcal{E} \subseteq \mathcal{A}$ is called an *explanation* for $\mathcal{O}$ if $\mathcal{P} \cup \mathcal{E}$ is satisfiable and $\mathcal{P} \cup \mathcal{E} \models_L^{\mathsf{lmwc}} L$ holds for each $L \in \mathcal{O}$. An explanation $\mathcal{E}$ is said to be *minimal* if there is no other explanation $\mathcal{E}' \subset \mathcal{E}$ of $\mathcal{O}$. A formula F is said to *follow skeptically by abduction* from $\mathcal{P}$ and $\mathcal{O}$ if there exists an explanation of $\mathcal{O}$ and for all minimal explanations $\mathcal{E}$ for $\mathcal{O}$ it holds that $\mathcal{P} \cup \mathcal{E} \models_L^{\mathsf{lmwc}} F$. This notion of abductive consequence with respect to least models of the weak completion has been elaborated in (Hölldobler et al., 2011) to model the backward reasoning cases of the suppression task.

Table 5 shows the representational form of these instances, including the observations and the respective minimal explanations. In the first three cases, additionally to the

	Conditionals		$\mathcal{O}$	Minimal $\mathcal{E}$s
$\mathcal{P}_l$	$= \{l \leftarrow e \wedge \neg ab_1,$	$ab_1 \leftarrow \bot\}$	$\{l\}$	$\{e \leftarrow \top\}$
$\mathcal{P}_{l+Alt}$	$= \{l \leftarrow e \wedge \neg ab_1, l \leftarrow t \wedge \neg ab_2,$	$ab_1 \leftarrow \bot,\ ab_2 \leftarrow \bot\}$	$\{l\}$	$\{e \leftarrow \top\}, \{t \leftarrow \top\}$
$\mathcal{P}_{l+Add}$	$= \{l \leftarrow e \wedge \neg ab_1, l \leftarrow o \wedge \neg ab_3,$	$ab_1 \leftarrow \neg o, ab_3 \leftarrow \neg e\}$	$\{l\}$	$\{e \leftarrow \top,\ o \leftarrow \top\}$
$\mathcal{P}_{\neg l}$	$= \{l \leftarrow e \wedge \neg ab_1,$	$ab_1 \leftarrow \bot\}$	$\{\neg l\}$	$\{e \leftarrow \bot\}$
$\mathcal{P}_{\neg l+Alt}$	$= \{l \leftarrow e \wedge \neg ab_1, l \leftarrow t \wedge \neg ab_2,$	$ab_1 \leftarrow \bot,\ ab_2 \leftarrow \bot\}$	$\{\neg l\}$	$\{e \leftarrow \bot,\ t \leftarrow \bot\}$
$\mathcal{P}_{\neg l+Add}$	$= \{l \leftarrow e \wedge \neg ab_1, l \leftarrow o \wedge \neg ab_3,$	$ab_1 \leftarrow \neg o\ \ ab_3 \leftarrow e\}$	$\{\neg l\}$	$\{e \leftarrow \bot\}, \{o \leftarrow \bot\}$

Table 5. Representational form of the "backward reasoning" instances of the suppression task according to Stenning & van Lambalgen (2008) and Hölldobler et al. (2011).

conditionals, the participants had to draw conclusions based on the fact that *She goes to the library.* In the last three cases they had to draw conclusions based on the fact that *She does not go to the library.* For instance, in the case of $\mathcal{P}_{l+Alt}$ we know that *She goes to the library*, thus $\mathcal{O} = \{l\}$. There are two independent explanations for this observation: either e is true (*She has an essay to write*) or t is true (*She has a textbook to read*). For $\mathcal{P}_{l+Add}$, $\mathcal{O} = \{l\}$ is still the case, but now both e and o have to be true to explain the observation.

4. Well-Founded and Related Semantics

In order to show the correspondence between weak completion and well-founded semantics, we will now review the latter and the related three-valued stable model semantics. We proceed by giving definitions of certain relevant program classes, which constrain the allowed possibilities of circular dependency in programs. On this basis, we then develop three-valued generalizations of the concepts of supported and well-supported models, which have been originally specified just for two-valued semantics. In this section, unless specified otherwise, we quietly consider the S-semantics as the used three-valued semantics.

4.1 *Three-Valued Stable Model and Well-Founded Semantics*

Stable models have been originally defined by Gelfond & Lifschitz (1988) in terms of a program transformation that is often called *Gelfond-Lifschitz transformation.* Przymusinski (1990) extended their approach to three-valued models and used this as basis to show the relationship to the well-founded semantics. The *reduct of a normal program* $\mathcal{P}$ with respect to an interpretation I, denoted by $\mathcal{P}|_I$, is obtained from $\mathcal{P}$ by replacing in the bodies of all clauses $\mathcal{P}$ each negative literal $\neg A$ by $I(\neg A)$, that is, with the truth value constant corresponding to the value of $\neg A$ under I. Notice that a reduct is still a set of clauses, although, because truth value constants may now occur in bodies, it is possibly not a program according to our specification in Sect. 2. An interpretation I is a *three-valued stable model* of $\mathcal{P}$ if and only if I is a truth-minimal model of $\mathcal{P}|_I$.

In analogy to the well-known $T_\mathcal{P}$ operator for two-valued interpretations (Van Emden & Kowalski, 1976), Przymusinski (1990) introduced an operator $\Psi_\mathcal{P}$ for three-valued interpretations: Suppose that $\mathcal{P}$ is a normal logic program and I is a three-valued interpretation of $\mathcal{P}$: Define $\Psi_\mathcal{P}(I) = J$ to be the interpretation given by

(*i*) $J(A) = \top$ if there exists a clause $A \leftarrow \mathit{Body} \in \mathcal{P}$ such that $I(\mathit{Body}) = \top$;

(*ii*) $J(A) = \mathsf{U}$ if $J(A) \neq \top$ and there exists a clause $A \leftarrow Body \in \mathcal{P}$ such that $I(Body) = \mathsf{U}$;
(*iii*) $J(A) = \bot$, otherwise.

This operator can be applied to the sets of implications obtained as reduct $\mathcal{P}|_I$. As shown by Przymusinski (1990), the least fixed point of $\Psi_{\mathcal{P}|_I}$ is the truth-least model of $\mathcal{P}|_I$. It has been further shown by Przymusinski (1990) that each normal program has a knowledge-least three-valued stable model, which coincides with the *well-founded model*.

Example: We assume $\mathsf{ATOMS} = \{p, q\}$ and consider as a first example $\mathcal{P}_1 = \{p \leftarrow q\}$. As $\mathcal{P}_1$ does not contain an occurrence of a negative literal in the body of a clause, we get the reduct $\mathcal{P}_1|_I = \mathcal{P}_1$ for any interpretation I. The models of $\mathcal{P}_1$ are:

$$\begin{array}{lll} I_1 = \langle \emptyset, \{p, q\} \rangle, & I_2 = \langle \{p, q\}, \emptyset \rangle, & I_3 = \langle \emptyset, \emptyset \rangle, \\ I_4 = \langle \{p\}, \{q\} \rangle, & I_5 = \langle \{p\}, \emptyset \rangle, & I_6 = \langle \emptyset, \{q\} \rangle. \end{array}$$

The only three-valued stable model is I_1 because $I_1 \preceq_t I_j$ for all $j \in [2, 6]$.
Now let $\mathcal{P}_2 = \{p \leftarrow \neg q,\ q \leftarrow \neg p\}$ and consider the following interpretations:

$$I_1 = \langle \{p\}, \{q\} \rangle, \quad I_2 = \langle \{q\}, \{p\} \rangle, \quad I_3 = \langle \emptyset, \emptyset \rangle.$$

The reducts of $\mathcal{P}_2$ with respect to these interpretations are:

$$\mathcal{P}_2|_{I_1} = \{p \leftarrow \top,\ q \leftarrow \bot\}, \quad \mathcal{P}_2|_{I_2} = \{p \leftarrow \bot,\ q \leftarrow \top\}, \quad \mathcal{P}_2|_{I_3} = \{p \leftarrow \mathsf{U},\ q \leftarrow \mathsf{U}\}.$$

All three interpretations I_1, I_2 and I_3 are truth-minimal models of the corresponding reducts and, hence, they are three-valued stable models of $\mathcal{P}_2$. It is easy to check that they are the only three-valued stable models of $\mathcal{P}_2$. As $I_3 \preceq_k I_j$ for $j \in \{1, 2\}$, I_3 is the knowledge-least three-valued stable model of $\mathcal{P}$.

Well-founded semantics (Van Gelder et al., 1991) has been defined as follows: A set of atoms $U \subseteq \mathsf{atoms}(\mathcal{P})$ is said to be an *unfounded set* of $\mathcal{P}$ with respect to interpretation I if each atom $A \in U$ satisfies the following condition: For each clause $A \leftarrow Body \in \mathcal{P}$, at least one of the following holds:

(1) $I(Body) = \bot$.
(2) There is a literal L in $\mathsf{pos}(Body)$ with $L \in U$.

Given I and $\mathcal{P}$, the transformations $T_\mathcal{P}$, $\mathcal{U}_\mathcal{P}$, and $W_\mathcal{P}$ are defined as follows:

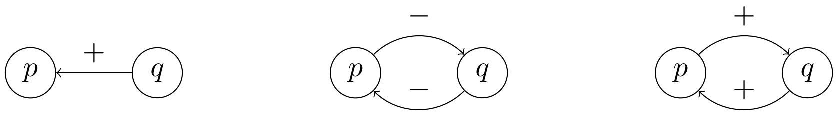
- $T_\mathcal{P}(I) = \{A \mid$ there exists a clause $A \leftarrow Body \in \mathcal{P}$ with $I(Body) = \top\}$,
- $\mathcal{U}_\mathcal{P}(I)$ is the *greatest unfounded set* of $\mathcal{P}$ with respect to I,
- $W_\mathcal{P}(I) = T_\mathcal{P}(I) \cup \neg\, \mathcal{U}_\mathcal{P}(I)$,

where the greatest unfounded set $\mathcal{U}_\mathcal{P}(I)$ of $\mathcal{P}$ with respect to I is the union of all unfounded sets of $\mathcal{P}$ with respect to I and $\neg\, \mathcal{U} := \{\neg A \mid A \in \mathcal{U}\}$.

$T_\mathcal{P}$, $\mathcal{U}_\mathcal{P}$ and $W_\mathcal{P}$ are monotonic transformations. The least fixed point of $W_\mathcal{P}(I)$ can be recursively defined as follows: Let α range over all countable ordinals. The sets I_α and I^∞ are defined recursively by starting with $I_0 = \langle \emptyset, \emptyset \rangle$:

(1) For limit ordinal α, $I_\alpha = \bigcup_{\beta < \alpha} I_\beta$.
(2) For successor ordinal $\alpha = \gamma + 1$, $I_{\gamma+1} = W_\mathcal{P}(I_\gamma)$.
(3) Finally, define $I^\infty = \bigcup_\alpha I_\alpha$.

I_α is the least fixed point of $W_\mathcal{P}$ where $I_\alpha = W_\mathcal{P}(I_\alpha)$. The least fixed point of $W_\mathcal{P}(I)$



$\mathcal{P}_1 = \{p \leftarrow q\}$ $\mathcal{P}_2 = \{p \leftarrow \neg q,\ p \leftarrow \neg q\}$ $\mathcal{P}_3 = \{p \leftarrow q,\ q \leftarrow p\}$

Figure 1. Program examples represented as graphs..

is the *well-founded model* of $\mathcal{P}$ ($\mathsf{wfm_S}$). A constructive definition of the well-founded semantics can be found in (Van Gelder, 1989).

4.2 *Program Classes with Respect to Cycles*

Let $\mathcal{P}$ be a program and $A, B \in \mathsf{atoms}(\mathcal{P})$. *A depends negatively on B* if and only if $\mathcal{P}$ contains a clause of the form $A \leftarrow \mathit{Body}$ and $\neg B$ is in $\mathsf{neg}(\mathit{Body})$. *A depends positively on B* if and only if A does not depend negatively on B and $\mathcal{P}$ contains a clause of the form $A \leftarrow \mathit{Body}$ and B is in $\mathsf{pos}(\mathit{Body})$. *A depends on B* if and only if A depends positively or negatively on B. In addition, dependency is transitive, thus, if A depends on B and B depends on C, then A depends on C, where one negative dependency is enough to define the whole dependency as negative. As an example consider the three programs in Figure 1 and their representations as graphs, where the nodes represent the atoms and the arcs represent the dependencies: An arc labeled "+" represents a positive dependency and an arc labeled "−" a negative dependency. The program $\mathcal{P}$ contains a *cycle* if at least one atom occurring in $\mathcal{P}$ depends on itself. In Figure 1 the programs $\mathcal{P}_2$ and $\mathcal{P}_3$ contain cycles.

Different program classes with respect to the occurrence of cycles are often defined through level mapping characterizations. A *level mapping* for a program $\mathcal{P}$ is a function $l : \mathsf{ATOMS} \mapsto \mathrm{N}$. We extend the definition to literals by setting $l(\neg A) := l(A)$. A program $\mathcal{P}$ is *acyclic with respect to a level mapping* l if and only if for every clause $A \leftarrow \mathit{Body} \in \mathcal{P}$ and for all literals L in *Body* we find that $l(A) > l(L)$. A program $\mathcal{P}$ is *acyclic* if and only if it is acyclic with respect to some level mapping. Consider again $\mathcal{P}_1$ in Figure 1. With $l(p) = 1$ and $l(q) = 0$ we find that $\mathcal{P}_1$ is acyclic, whereas $\mathcal{P}_2$ and $\mathcal{P}_3$ are not acyclic.

Stratified logic programs have been investigated in (Apt et al., 1988; Przymusinski, 1988). A level mapping characterization of this class of programs can be given as follows (Hitzler & Wendt, 2005): A program $\mathcal{P}$ is *stratified with respect to a level mapping* l if and only if for every clause $A \leftarrow \mathit{Body} \in \mathcal{P}$ we find that $l(A) \geq l(L)$ for all literals L in $\mathsf{pos}(\mathit{Body})$, and $l(A) > l(L)$ for all literals L in $\mathsf{neg}(\mathit{Body})$. A program $\mathcal{P}$ is *stratified* if and only if it is stratified with respect to some level mapping. Programs which only contain positive cycles are stratified. In our example $\mathcal{P}_3$ is stratified, but $\mathcal{P}_2$ is not.

Fages (1994) introduced the term *positive-order-consistent* to define programs that do not contain positive cycles. Nowadays, the term *tight* is often used for this property (Erdem & Lifschitz, 2003). A level mapping characterization for this class of programs is defined as follows: A program $\mathcal{P}$ is *tight with respect to a level mapping* l if and only if for every clause $A \leftarrow \mathit{Body} \in \mathcal{P}$ we find that $l(A) > l(L)$ for all literals L in $\mathsf{pos}(\mathit{Body})$. A program $\mathcal{P}$ is *tight* if and only if it is tight with respect to some level mapping. Programs which only contain negative cycles, are tight programs. In our example $\mathcal{P}_2$ is tight, but $\mathcal{P}_3$ is not.

Under two-valued semantics, negative *odd* cycles lead to inconsistency. Consider the

Program $\mathcal{P}$	Three-valued Stable Models of $\mathcal{P}$	Models of the Completion of $\mathcal{P}$	Models of the weak Completion of $\mathcal{P}$
$\mathcal{P}_1 = \{p \leftarrow q\}$	$\langle\emptyset, \{p, q\}\rangle$	$\langle\emptyset, \{p, q\}\rangle$	$\langle\emptyset, \{p, q\}\rangle$ $\langle\emptyset, \emptyset\rangle$ $\langle\{p, q\}, \emptyset\rangle$
$\mathcal{P}_2 = \{p \leftarrow \neg q,\ q \leftarrow \neg p\}$	$\langle\{p\}, \{q\}\rangle$ $\langle\emptyset, \emptyset\rangle$ $\langle\{q\}, \{p\}\rangle$	$\langle\{p\}, \{q\}\rangle$ $\langle\emptyset, \emptyset\rangle$ $\langle\{q\}, \{p\}\rangle$	$\langle\{p\}, \{q\}\rangle$ $\langle\emptyset, \emptyset\rangle$ $\langle\{q\}, \{p\}\rangle$
$\mathcal{P}_3 = \{p \leftarrow q,\ q \leftarrow p\}$	$\langle\emptyset, \{p, q\}\rangle$	$\langle\emptyset, \{p, q\}\rangle$ $\langle\emptyset, \emptyset\rangle$ $\langle\{p, q\}, \emptyset\rangle$	$\langle\emptyset, \{p, q\}\rangle$ $\langle\emptyset, \emptyset\rangle$ $\langle\{p, q\}, \emptyset\rangle$

Table 6. Program examples and the corresponding three-valued stable models, models of the completion and models of the weak completion, under the assumption $\mathsf{ATOMS} = \{p, q\}$.

following example:

$$\mathcal{P}_{\textit{neg-odd}} = \{p \leftarrow \neg q,\ q \leftarrow \neg r,\ r \leftarrow \neg p\}.$$

There is no two-valued stable model of $\mathcal{P}_{\textit{neg-odd}}$ and the completion of $\mathcal{P}_{\textit{neg-odd}}$ is inconsistent. Under three-valued stable model semantics atoms stay unknown when they are involved in negative cycles. Table 6 shows the three-valued stable models, the models of the completion and the models of the weak completion of our three example programs P_1, P_2 and P_3.

4.3 *Three-Valued Notions of Supported and Well-Supported Models*

In two-valued logic, the notion of *supported model* provides an alternate characterization of the models of Clark's completion (Apt et al., 1988). We adapt this characterization to three-valued logics. Our considerations apply to both Ł-semantics and S-semantics, since for the relevant classes of formulas both semantics lead to the same model relationship.

Definition 1. An interpretation I is *supported* with respect to a set of clauses $\mathcal{P}$ if and only if for all atoms A with $I(A) \neq \bot$ there exists a clause $A \leftarrow \textit{Body} \in \mathcal{P}$ such that $I(\textit{Body}) = I(A)$.

We say that I is a *supported model* of $\mathcal{P}$ if and only if I is a model of $\mathcal{P}$ and is supported with respect to $\mathcal{P}$. Analogously to the two-valued case, completion and supported models coincide for three-valued logics:

Lemma 2. *For any program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is a model of the completion of $\mathcal{P}$.
(2) I is a supported model of $\mathcal{P}$.

Proof. Easy to see from the definition of completion and the truth tables of the three-valued operators. □

In order to deal with positive cycles, some approaches propose to eliminate cyclic support for atoms and leave their truth value either unknown or map them to false. For two-valued logics, this is captured for example by the notions of *grounded models* (Elkan,

1990) and *well-supported models* (Fages, 1991), that is, models which are supported and assign $\top$ only to atoms that are not involved in positive cycles. Well-supported models in this sense are exactly the two-valued stable models. We now extend this concept to three-valued logics:

Definition 3. An interpretation I is *well-supported* with respect to a level mapping l and a finite set of clauses $\mathcal{P}$ if and only if for all atoms A with $I(A) \neq \bot$ there exists a clause $A \leftarrow Body \in \mathcal{P}$ such that $I(Body) = I(A)$ and for all literals L in $\mathsf{pos}(Body)$ it holds that $l(L) < l(A)$.

We call a clause $A \leftarrow Body$ that meets the requirement of Definition 3 a *supporting justification of* A. We say that I is a *well-supported model* of $\mathcal{P}$ if and only if I is a model of $\mathcal{P}$ and is well-supported with respect to $\mathcal{P}$ and some level mapping. The following lemma follows immediately from the definitions of supported and well-supported models:

Lemma 4. *Any well-supported model of a program* $\mathcal{P}$ *is a supported model of* $\mathcal{P}$*.*

For two-valued logics, the correspondence between stable models and well-supported models has been developed by Elkan (1990) in a stepwise way. We now adapt these steps to our three-valued setting in the following Lemmas 5–9. As in the case of completion and supported models, these propositions apply to both Ł-semantics and S-semantics.

Lemma 5. *Any model* I *of a normal program* $\mathcal{P}$ *is also a model of* $\mathcal{P}|_I$*.*

Proof. Immediate from the definition of $\mathcal{P}|_I$: We obtain $\mathcal{P}|_I$ from $\mathcal{P}$ by replacing subformulas with truth value constants corresponding to their value under I. □

Lemma 6. *Any well-supported model* I *of a normal program* $\mathcal{P}$ *is also a well-supported model of* $\mathcal{P}|_I$*.*

Proof. Let $A \leftarrow Body$ be a supporting justification of A in $\mathcal{P}$ with respect to I and a level mapping l such that $l(A) < l(L)$ for each L in $\mathsf{pos}(Body)$. Let $\neg B_1 \land \ldots \land \neg B_n$, where $n \geq 0$, be $\mathsf{neg}(Body)$ and let $Body'$ be $\mathsf{pos}(Body) \land I(\neg B_1) \land \ldots \land I(\neg B_n)$. The clause $A \leftarrow Body'$ is then a supporting justification in $\mathcal{P}|_I$: It is a member of $\mathcal{P}|_I$, it holds that $l(A) < l(L)$ for each L in $\mathsf{pos}(Body') = \mathsf{pos}(Body)$ and the semantic requirements are met since $I(Body') = I(Body)$. □

Lemma 7. *Any well-supported model is truth-minimal.*

Proof. By contradiction: Let $\mathcal{P}$ be a finite set of clauses and let I be a well-supported model of $\mathcal{P}$ with respect to a level mapping l. Assume that I is not truth-minimal, that is, there exists a model J of $\mathcal{P}$ such that $J \preceq_t I$ and $J \neq I$. Let $I^U := \mathsf{ATOMS} \setminus (I^\top \cup I^\bot)$ and analogously $J^U := \mathsf{ATOMS} \setminus (J^\top \cup J^\bot)$. The condition $J \preceq_t I$ and $J \neq I$ can then be equivalently expressed as $(J^\top \cup J^U) \subset (I^\top \cup I^U)$. The set of atoms $\Delta = (I^\top \cup I^U) \setminus (J^\top \cup J^U)$ then must be nonempty. Observe that for all $D \in \Delta$ it holds that $J(D) = \bot$ and $I(D) \neq \bot$. Now let D be one of those members of Δ whose value of the level mapping l is least among the values of l of the members of Δ. Let $D \leftarrow Body \in \mathcal{P}$ be a supporting justification of D with respect to I. Then it holds that $I(Body) \neq \bot$ and that $l(L) < l(D)$ for each literal L in $\mathsf{pos}(Body)$. Since J is a model of $\mathcal{P}$ and we have $J(D) = \bot$, it follows that $J(Body) = \bot$. Thus, there must be a literal L in $Body$ with $J(L) = \bot$ and $I(L) \neq \bot$. In the case that L is a negative literal $\neg B$ it must hold that $J(B) = \top$. Since $J^\top \subseteq I^\top$ it would follow that $I(B) = \top$ and thus $I(L) = \bot$, in contradiction to $I(L) \neq \bot$. In the case that L is a positive literal, we have that $L \in \Delta$ and $l(L) < l(D)$, in contradiction to the fact that $l(D)$ is a least level mapping value among the members of Δ. □

Lemma 8. *For any normal program $\mathcal{P}$ and interpretation I, the truth-minimal model of $\mathcal{P}|_I$ is well-supported.*

Proof. This follows from the fixed point construction of the truth-minimal model of $\mathcal{P}|_I$ by the operator Ψ introduced in (Przymusinski, 1990) (see Section 4.1). Well-supportedness is assured by any level mapping, where an atom is assigned level i if its value is determined in the ith iteration of the application of Ψ. $\square$

Lemma 9. *For any normal program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is a three-valued stable model of $\mathcal{P}$.
(2) I is a well-supported model of $\mathcal{P}$.

Proof. (1) $\rightarrow$ (2) Let I be a three-valued stable model of $\mathcal{P}$. It follow that I is a model of $\mathcal{P}$ and it is a truth-minimal model of $\mathcal{P}|_I$. By Lemma 8 it holds that I is well-supported with respect to $\mathcal{P}|_I$ and some level mapping l. Let $A \leftarrow Body$ be a justification of atom A with respect to $\mathcal{P}|_I$. It then holds that $I(A) = I(Body) \neq \bot$ and $l(A) > l(L)$ for all literals L in $Body$. In $\mathcal{P}$ there must be a clause $A \leftarrow Body'$ from which $A \leftarrow Body$ has been obtained in forming the reduct. From the construction of $\mathcal{P}|_I$ it follows that $\mathsf{pos}(Body') = \mathsf{pos}(Body)$ and that I is a model of $\mathsf{neg}(Body')$ and thus $A \leftarrow Body'$ is a justification of A with respect to $\mathcal{P}$.

(2) $\rightarrow$ (1) By Lemma 6 and 7 any well-supported model I of $\mathcal{P}$ is a truth-minimal model of $\mathcal{P}|_I$, and thus a stable model of $\mathcal{P}$. $\square$

Lemma 10. *Any three-valued stable model I of $\mathcal{P}$ is a model of the completion of $\mathcal{P}$.*

Proof. This follows immediately from Lemma 2, 4 and 9. $\square$

Figure 2 on page 16 summarizes the correspondences between several two- and three-valued semantics, including results reported in the literature so far. For instance, Fages showed that for tight logic programs under two-valued semantics, the stable models coincide with the models of the completion. Pzymusinski showed that the knowledge-least three-valued stable model coincides with the well-founded model.

5. Weak Completion and Well-Founded Semantics

With Theorem 11 below we now show that the knowledge-least model of the weak completion is identical to the well-founded model of the program, after a transformation that essentially effects simulation of the treatment of undefined atoms under weak completion. This transformation is specified as follows: We assume that ATOMS is the union of disjoint sets ATOMS' and $\mathsf{AUXATOMS} := \{\mathsf{n\_}A \mid A \in \mathsf{ATOMS}'\}$. Only members of ATOMS' are allowed to occur in input programs. For such programs $\mathcal{P}$ we define

$$\mathcal{P}^{\mathsf{mod}} := \mathcal{P}^{+} \cup \bigcup_{A \in \mathsf{undef}(\mathcal{P})} \{A \leftarrow \neg \mathsf{n\_}A,\ \mathsf{n\_}A \leftarrow \neg A\}.$$

We assume that atoms in $\mathsf{AUXATOMS}$ only occur in programs $\mathcal{P}^{\mathsf{mod}}$ resulting from the indicated transformation. As before in Sect. 4.3, our considerations in this section apply to both Ł-semantics and S-semantics, where for the investigated classes of formulas both semantics lead to the same model relationship. The coincidence of weak completion and well-founded semantics can now be stated as follows:

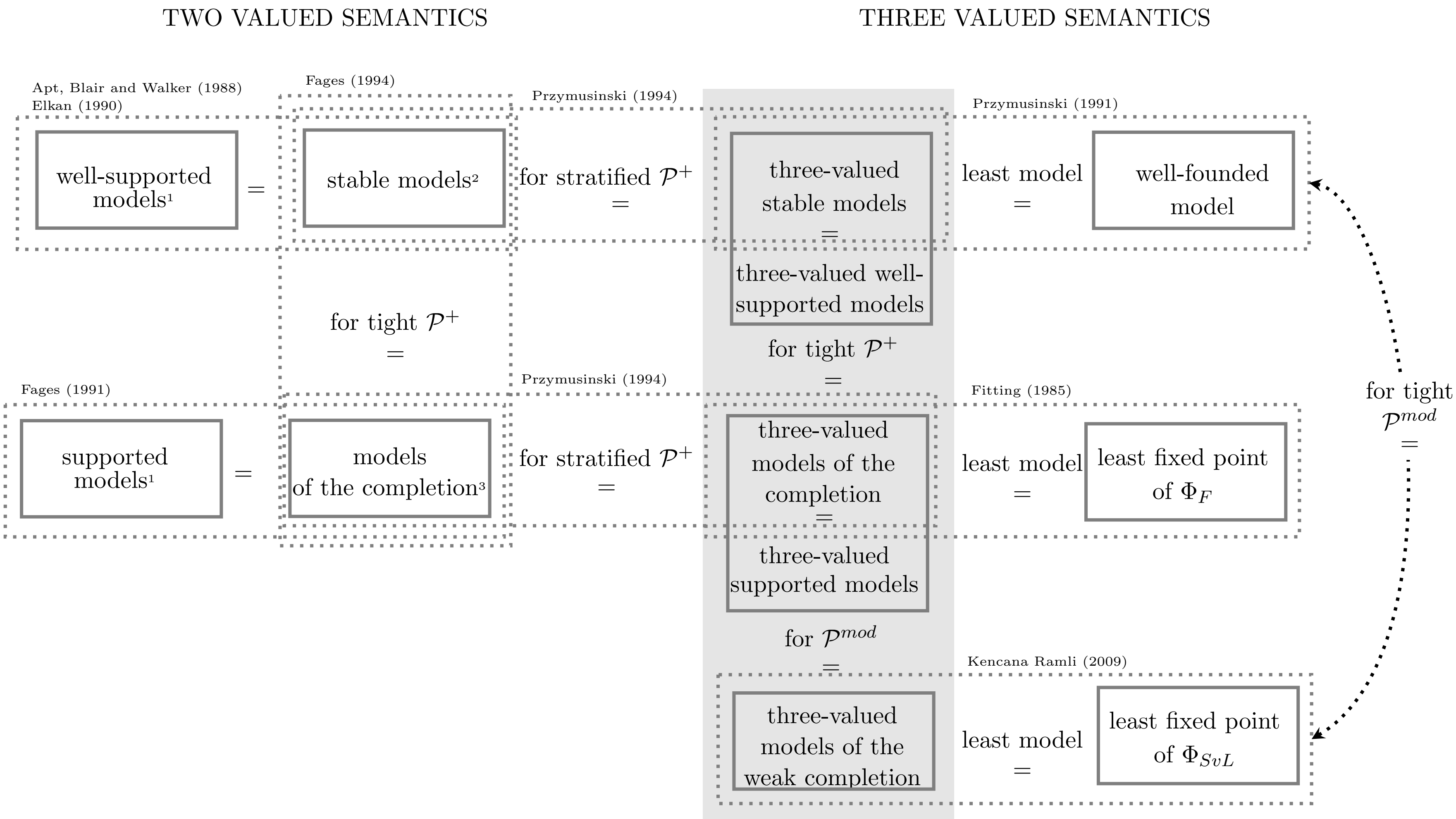


Figure 2. Overview of several two- and three-valued semantics. We show the correspondences in the gray box. $\mathcal{P}^{\mathsf{mod}}$ is defined as $\mathcal{P}^+ \cup \bigcup_{A \in \mathsf{undef}(\mathcal{P})} \{A \leftarrow \neg \mathsf{n\_}A,\ \mathsf{n\_}A \leftarrow \neg A\}$ where $\mathcal{P}^+$ is $\mathcal{P}$ without negative facts.

[1] Supported and well-supported models are discussed in Section 4.3.
[2] Stable models are discussed in Section 4.1.
[3] Models of the completion are discussed in Section 2.2.

Theorem 11. *For any tight program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is the knowledge-least model of the weak completion of $\mathcal{P}$.
(2) I is the well-founded model of $\mathcal{P}^{\mathsf{mod}}$.

In the rest of this section we develop the proof of Theorem 11, which involves further auxiliary definitions and some intermediate results, in particular about the correspondence between the three-valued completion semantics and three-valued stable model semantics [1]. We first note some properties of $\mathcal{P}^{\mathsf{mod}}$, which follow easily from its definition:

Lemma 12. (i) *If a program $\mathcal{P}$ is tight, then $\mathcal{P}^{\mathsf{mod}}$ is also tight.*
(ii) *For any program $\mathcal{P}$ it holds that $\mathcal{P}^{\mathsf{mod}}$ is a normal program.*

If we consider not just knowledge-least models, we have to map between interpretations that assign to the members of $\mathsf{AUXATOMS}$ values as required by $\mathcal{P}^{\mathsf{mod}}$ and interpretations where the value of members of $\mathsf{AUXATOMS}$ is always unknown. To this end, we define the following two conversions for interpretations I and sets of atoms S:

$$I_S^{\mathsf{mod}} := \langle I^\top \cup \{\mathsf{n\_}A \mid A \in S \cap I^\bot\},\ I^\bot \cup \{\mathsf{n\_}A \mid A \in S \cap I^\top\}\rangle.$$
$$I^{\mathsf{invmod}} := \langle I^\top \setminus \mathsf{AUXATOMS},\ I^\bot \setminus \mathsf{AUXATOMS}\rangle.$$

Notice that for all sets of atoms $S \subseteq \mathsf{ATOMS}'$, whenever an interpretation I is a model of $\{\mathsf{n\_}A \leftrightarrow \neg A \mid A \in S\}$, then

$$I = (I^{\mathsf{invmod}})_S^{\mathsf{mod}}.$$

We can thus conclude from $I \models \mathcal{P}^{\mathsf{mod}}$ that $(I^{\mathsf{invmod}})^{\mathsf{mod}}_{\mathsf{undef}(\mathcal{P})} \models \mathcal{P}^{\mathsf{mod}}$, and that for all interpretations I such that $I \models \{\mathsf{n\_}A \leftrightarrow \neg A \mid A \in \mathsf{undef}(\mathcal{P})\}$ the statements $I \models \mathcal{P}^{\mathsf{mod}}$ and $(I^{\mathsf{invmod}})^{\mathsf{mod}}_{\mathsf{undef}(\mathcal{P})} \models \mathcal{P}^{\mathsf{mod}}$ are equivalent. We can now state a correspondence between the weak completion and the completion:

Lemma 13. *For any program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is a model of the weak completion of $\mathcal{P}$.
(2) $I^{\mathsf{mod}}_{\mathsf{undef}(\mathcal{P})}$ is a model of the completion of $\mathcal{P}^{\mathsf{mod}}$.

Proof. Let I be a model of $\mathsf{c}\,\mathcal{P}$. Based on the second step of the definition of $\mathsf{c}\,\mathcal{P}$, for every $A \in \mathsf{undef}(\mathcal{P})$ we find $A \in I^\bot$. By adding the clauses $A \leftarrow \neg\mathsf{n\_}A$, $\mathsf{n\_}A \leftarrow \neg A$ for every $A \in \mathsf{undef}(\mathcal{P})$ in the construction of $\mathcal{P}^{\mathsf{mod}}$, the second step in the definition of completion doesn't apply anymore, and thus A can be either true, false or unknown. This corresponds to the definition of the weak completion of $\mathcal{P}$. With this observation the Lemma follows immediately from the fact that the definitions of all other atoms in $\mathcal{P}$ and $\mathcal{P}^{\mathsf{mod}}$ are identical. □

The relationship between I_S^{mod} and I^{invmod} indicated above allows to express Lemma 13 equivalently also with respect to interpretations I and I^{invmod}:

[1]Pereira et al. (1991) showed the correspondence between contradiction free extended stable model semantics and extended stable model semantics, an extension of well-founded semantics by introducing a similar transformation as for $\mathcal{P}^{\mathsf{mod}}$ where the transformed program is extended with the following clauses: $A \leftarrow \neg A'$, $A' \leftarrow \neg A$ and $A' \leftarrow \neg A'$. A further early documented use of the pattern $A \leftarrow \neg A'$, $A' \leftarrow \neg A$ was presented in the context of abduction (Satoh & Iwayama, 1991).

Lemma 14. *For any program $\mathcal{P}$ and interpretation I such that $I \models \{\mathsf{n\_}A \leftrightarrow \neg A \mid A \in \mathsf{undef}(\mathcal{P})\}$ the following two statements are equivalent:*

(1) I^{invmod} is a model of the weak completion of $\mathcal{P}$.
(2) I is a model of the completion of $\mathcal{P}^{\mathsf{mod}}$.

We now transfer Lemma 13 to knowledge-least models:

Lemma 15. *For any program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is the knowledge-least model of the weak completion of $\mathcal{P}$.
(2) I is the knowledge-least model of the completion of $\mathcal{P}^{\mathsf{mod}}$.

Proof. By (Hölldobler & Kencana Ramli, 2009b), there must exist a knowledge-least model of the weak completion of $\mathcal{P}$. From Lemma 13 we can thus conclude that the statement (1) is equivalent to

(3) $\quad I^{\mathsf{mod}}_{\mathsf{undef}(\mathcal{P})}$ is the knowledge-least model of the completion of $\mathcal{P}^{\mathsf{mod}}$.

By results from (Hölldobler & Kencana Ramli, 2009b) it follows also that if I is the knowledge-least model of the weak completion of $\mathcal{P}$, then for all atoms $A \in \mathsf{undef}(\mathcal{P})$ it holds that $I(A) = \mathsf{U}$. Thus, if I satisfies (1), then $I = I^{\mathsf{mod}}_{\mathsf{undef}(\mathcal{P})}$. Hence statement (1) implies (2). That also statement (2) implies (1) can be shown as follows: If I is a model of the completion of P^{mod}, then it must hold that $I = (I^{\mathsf{invmod}})^{\mathsf{mod}}_{\mathsf{undef}(\mathcal{P})}$, and thus, by the equivalence of (3) and (1), I^{invmod} is the knowledge-least model of the weak completion of $\mathcal{P}$. We then have $I^{\mathsf{invmod}} = (I^{\mathsf{invmod}})^{\mathsf{mod}}_{\mathsf{undef}(\mathcal{P})}$, which implies $I^{\mathsf{invmod}} = I$. □

Fages (1994) showed that under two-valued semantics the models of the completion of a normal logic program $\mathcal{P}$ coincide with the stable models of $\mathcal{P}$ if $\mathcal{P}$ is tight. In the following lemma, we transfer this result, which is sometimes called *Fages' theorem*, to three-valued semantics.

Lemma 16. *For any tight normal program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is a model of the completion of $\mathcal{P}$.
(2) I is a three-valued stable model of $\mathcal{P}$.

Proof. *(1) → (2)* This follows immediately from Lemma 10.
(2) → (1) By contradiction: Assume that $\mathcal{P}$ is tight and that I is a model of the completion of $\mathcal{P}$, but not a three-valued stable model. By Lemma 2 and 9, interpretation I is supported but not well-supported. Then for all level mappings l there exists an atom $A \notin I^{\perp}$ such that for all clauses $A \leftarrow Body \in \mathcal{P}$ with L in $\mathsf{pos}(Body)$ such that $l(L) < l(A)$ does not hold. Because I is a model of the completion of $\mathcal{P}$ such a clause must indeed exist. But then there is a positive cycle in the program, in contradiction to the precondition that $\mathcal{P}$ is tight. □

In the following two corollaries we instantiate Lemma 16 with $\mathcal{P}^{\mathsf{mod}}$ and restrict the considered interpretations to knowledge-least models.

Corollary 17. *For any tight program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is a model of the completion of $\mathcal{P}^{\mathsf{mod}}$.
(2) I is a three-valued stable model of $\mathcal{P}^{\mathsf{mod}}$.

Proof. By Lemma 12 it holds for all tight programs P that $\mathcal{P}^{\mathsf{mod}}$ is normal and tight. The corollary then is an immediate consequence of Lemma 16. □

Corollary 18. *For any tight program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is the knowledge-least model of the completion of $\mathcal{P}^{\mathsf{mod}}$.
(2) I is the knowledge-least three-valued stable model of $\mathcal{P}^{\mathsf{mod}}$.

Proof. By Lemma 15 the completion of $\mathcal{P}^{\mathsf{mod}}$ admits a knowledge-least model. From Corollary 17 follows that the set of three-valued stable models of $\mathcal{P}^{\mathsf{mod}}$ and the set of models of the completion of $\mathcal{P}^{\mathsf{mod}}$ are the same. Therefore, $\mathcal{P}^{\mathsf{mod}}$ must also have a knowledge-least three-valued stable model, which must be identical to the knowledge-least model of the completion of $\mathcal{P}$. □

Przymusinski (1990) has shown that knowledge-least three-valued stable models coincide with well-founded models:

Lemma 19. *For any normal program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is the knowledge-least three-valued stable model of $\mathcal{P}$.
(2) I is the well-founded model of $\mathcal{P}$.

In the following corollary we instantiate this result by Przymusinski with $\mathcal{P}^{\mathsf{mod}}$.

Corollary 20. *For any program $\mathcal{P}$ and interpretation I the following two statements are equivalent:*

(1) I is the knowledge-least three-valued stable model of $\mathcal{P}^{\mathsf{mod}}$.
(2) I is the well-founded model of $\mathcal{P}^{\mathsf{mod}}$.

Proof. Follows as corollary from Lemma 19 and Corollary 12.ii. □

Finally we combine the material developed in this section to prove Theorem 11:

Proof of Theorem 11. Let $\mathcal{P}$ be a tight program and let I be an interpretation. Then the following four statements are equivalent:

(1) I is the knowledge-least model of $\mathsf{wc}\,\mathcal{P}$.
(2) I is the knowledge-least model of $\mathsf{c}\,\mathcal{P}^{\mathsf{mod}}$ (by Lemma 15).
(3) I is the knowledge-least three-valued stable model of $\mathcal{P}^{\mathsf{mod}}$ (by Corollary 18 and Lemma 12.i).
(4) I is the well-founded model of $\mathcal{P}^{\mathsf{mod}}$ (by Corollary 20).

□

In the appendix we show the correspondence between the knowledge-least model of the weak completion and the well-founded model with another proof technique, where level mapping characterizations of both semantics are directly compared. While this applies only to knowledge-least models, with the techniques applied in this section, we have been able to prove results that apply to three-valued models in general, in particular Lemma 13 and 16.

6. Modeling the Suppression Task with Different Three-Valued Semantics

We now return to the suppression task and show the results obtained with the different discussed semantics for the program representations presented in Table 4 and

$\mathcal{P}$	$\mathsf{lm}_L\mathsf{wc}\,\mathcal{P}/\mathsf{wfm}_S\,\mathcal{P}^{\mathsf{mod}}$	$\mathsf{wfm}_S\,\mathcal{P}^+$	Byrne
$\mathcal{P}_e$	$\langle\{e,l\},\{ab_1\}\rangle$	$\langle\{e,l\},\{o,t,ab_1,ab_2,ab_3\}\rangle$	96% l
$\mathcal{P}_{e+Alt}$	$\langle\{e,l\},\{ab_1,ab_2\}\rangle$	$\langle\{e,l\},\{o,t,ab_1,ab_2,ab_3\}\rangle$	96% l
$\mathcal{P}_{e+Add}$	$\langle\{e\},\{ab_3\}\rangle$	$\langle\{e,ab_1\},\{l,o,t,ab_2,ab_3\}\rangle$	38% l
$\mathcal{P}_{\neg e}$	$\langle\emptyset,\{e,l,ab_1\}\rangle$	$\langle\emptyset,\{e,l,o,t,ab_1,ab_2,ab_3\}\rangle$	46% $\neg l$
$\mathcal{P}_{\neg e+Alt}$	$\langle\emptyset,\{e,ab_1,ab_2\}\rangle$	$\langle\emptyset,\{e,l,o,t,ab_1,ab_2,ab_3\}\rangle$	4% $\neg l$
$\mathcal{P}_{\neg e+Add}$	$\langle\{ab_3\},\{e,l\}\rangle$	$\langle\{ab_3\},\{e,l,o,t,ab_1,ab_2\}\rangle$	63% $\neg l$

Table 7. The results of the first part of the suppression task. The highlighted conclusions show the differences between the conclusions of the least models of the weak completion and the well-founded models.

$\mathcal{P}$	$\mathcal{O}$	$\mathcal{E}$	$\mathsf{lm}_L\mathsf{wc}\,(\mathcal{P}\cup\mathcal{E})/$ $\mathsf{wfm}_S\,((\mathcal{P}\cup\mathcal{E})^{\mathsf{mod}})$	$\mathsf{wfm}_S\,\mathcal{P}^+$	Byrne
$\mathcal{P}_l$	l	$e\leftarrow\top$	$\langle\{e,l\},\{ab_1\}\rangle$	$\langle\{e,l\},\{o,t,ab_1,ab_2,ab_3\}\rangle$	53% e
$\mathcal{P}_{l+Alt}$	l	$e\leftarrow\top$	$\langle\{e,l\},\{ab_1,ab_2\}\rangle$	$\langle\{e,l\},\{o,t,ab_1,ab_2,ab_3\}\rangle$	16% e
		$t\leftarrow\top$	$\langle\{l,t\},\{ab_1,ab_2\}\rangle$	$\langle\{l,t\},\{e,o,ab_1,ab_2,ab_3\}\rangle$	
$\mathcal{P}_{l+Add}$	l	$e\leftarrow\top$,	$\langle\{e,l,o\},\{ab_1,ab_3\}\rangle$	$\langle\{e,l,o\},\{t,ab_1,ab_2,ab_3\}\rangle$	55% e
		$o\leftarrow\top$			
$\mathcal{P}_{\neg l}$	$\neg l$	$e\leftarrow\bot$	$\langle\emptyset,\{e,l,ab_1\}\rangle$	$\langle\emptyset,\{e,l,o,t,ab_1,ab_2,ab_3\}\rangle$	69% $\neg e$
$\mathcal{P}_{\neg l+Alt}$	$\neg l$	$e\leftarrow\bot$,	$\langle\emptyset,\{e,l,t,ab_1,ab_2\}\rangle$	$\langle\emptyset,\{e,l,o,t,ab_1,ab_2,ab_3\}\rangle$	69% $\neg e$
		$t\leftarrow\bot$			
$\mathcal{P}_{\neg l+Add}$	$\neg l$	$e\leftarrow\bot$	$\langle\{ab_3\},\{e,l\}\rangle$	$\langle\{ab_1,ab_3\},\{e,l,o,t,ab_2\}\rangle$	44% $\neg e$
		$o\leftarrow\bot$	$\langle\{ab_1\},\{l,o\}\rangle$	$\langle\{ab_1,ab_3\},\{e,l,o,t,ab_2\}\rangle$	

Table 8. The results of the second part of the suppression task. The highlighted conclusions show the differences between the conclusions of the least models of the weak completion and the well-founded models.

Table 5. We define $\mathsf{ATOMS}' = \{e,l,o,t,ab_1,ab_2,ab_3\}$ and for the well-founded models $\mathcal{P}^+$ we assume the models w.r.t. $\mathsf{ATOMS} = \mathsf{ATOMS}'$. For the least models of the weak completion of $\mathcal{P}$ and the well-founded models of $\mathcal{P}^{mod}$ we assume the models w.r.t. $\mathsf{ATOMS} = \mathsf{ATOMS}' \cup \bigcup_{A\in\mathsf{undef}(\mathcal{P})}\{A \leftarrow \neg\mathsf{n\_}A,\ \mathsf{n\_}A \leftarrow \neg A\}$. Table 7 shows the least models of the weak completion and the well-founded models from the first part of the suppression task. Notice that for the well-founded model only normal logic programs ($\mathcal{P}^+$) are considered. Obviously there are differences between both semantics with respect to the least models. For instance, for $\mathcal{P}_{e+Add}$ and $\mathcal{P}_{\neg e+Alt}$, under weak completion semantics, l is neither in $I^\top$ or in $I^\bot$, whereas in the well-founded model $l \in I^\bot$ in both $\mathcal{P}^+_{e+Add}$ and $\mathcal{P}^+_{\neg e+Alt}$. This is due to the fact, that undefined atoms such as o in $\mathcal{P}^+_{e+Add}$ and t in $\mathcal{P}^+_{\neg e+Alt}$ are mapped to false in the well-founded model. Considering Byrne's results, well-founded semantics does not represent the participants' conclusions of suppressing information, whereas weak completion semantics does.

Table 8 shows the results from the second part of the suppression task where abduction is required. In the first three cases, both semantics have the same conclusions about e. In the case of $\mathcal{P}_{l+Alt}$ two explanations are possible ($e \leftarrow \top$ or $t \leftarrow \top$) with two different least models. With skeptical reasoning nothing can be concluded about e, which seems to adequately represent Byrne's findings. For $\mathcal{P}_{\neg l+Add}$ with skeptical reasoning nothing can be concluded about e under weak completion, whereas e is true under well-founded semantics. Considering Byrne's results, that 44% of the participants concluded $\neg e$, it is not clear which model would adequately represent these results.

7. Conclusion

Taking the least model of the weak completion of logic programs as representing the suppression task results in a modeling of human reasoning that corresponds to the empirical results obtained by Byrne (1989). We have shown here, how the well-founded semantics can be applied to achieve the same correspondence. In order to do this, we extended the two-valued characterization for supported and well-supported models to three-valued logics and examined quite generally the properties of weak completion, completion and three-valued stable model semantics. When we restrict ourselves to tight logic programs and apply some technical modifications, weak completion semantics and three-valued stable model semantics, which underlies the well-founded semantics, yield the same results.

This gives us insights into the behavior of the considered semantics. Undefined atoms are always false under the three-valued stable model semantics. The same holds for atoms that can only be justified through positive cycles. If the only possibility for justification available is through a cycle that involves negation, atoms are unknown in the well-founded model.

In this context it is interesting to examine whether these technical properties are somehow reflected by human behavior. How do people reason with cycles? Do they ignore tautological conditionals and how do they extract their knowledge from contradictory information? For this purpose, a psychological experiment has been carried out by Dietz et al. (2013), where participants were presented with problems consisting of conditionals that involved circular dependencies in 1, 2, or 3 steps. The participants had to choose whether the premises or the conclusions of the conditionals were true, false or unknown. The empirical results obtained so far indicate that in presence of circular dependency people actually tend to infer undefinedness in contrast to falsehood, in spirit of the weak completion semantics in contrast to what is suggested by the direct application of the well-founded or three-valued stable model semantics.

8. Acknowledgments

Many thanks to Jim Delgrande, Luís Moniz Pereira and Alexandre Miguel Pinto for inspiring and productive discussions. The authors are grateful to Bertram Fronhöfer who provided useful suggestions to improve the presentation. The work on the results of this report has in part been funded by DAAD's IPID program, financed by the German Federal Ministry of Education and Research (BMBF).

References

Apt, K. R., Blair, H. A., & Walker, A. (1988). Towards a theory of declarative knowledge. In J. Minker (Ed.), *Foundations of Deductive Databases and Logic Programming* (pp. 89–148). San Francisco, CA: Morgan Kaufmann.

Apt, K. R. & van Emden, M. H. (1982). Contributions to the theory of logic programming. *Journal of the ACM*, 29(3), 841–862.

Byrne, R. M. J. (1989). Suppressing valid inferences with conditionals. *Cognition*, 31, 61–83.

Clark, K. L. (1978). Negation as failure. In H. Gallaire & J. Minker (Eds.), *Logic and Data Bases*, volume 1 (pp. 293–322). New York, NY: Plenum Press.

Dietz, E.-A., Hölldobler, S., & Ragni, M. (2012). A computational logic approach to the suppression task. In N. Miyake, D. Peebles, & R. P. Cooper (Eds.), *Proceedings of the 34th Annual Conference of the Cognitive Science Society, CogSci 2013* (pp. 1500–1505).: Austin, TX: Cognitive Science Society.

Dietz, E.-A., Hölldobler, S., & Ragni, M. (2013). A computational logic approach to the abstract and the social case of the selection task. In *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning, COMMONSENSE 2013* Aeya Nappa, Cyprus.

Dieussaert, K., Schaeken, W., Schroyens, W., & D'Ydewalle, G. (2000). Strategies during complex conditional inferences. *Thinking and Reasoning*, 6(2), 125–161.

Elkan, C. (1990). A rational reconstruction of nonmonotonic truth maintenance systems (research note). *Artificial Intelligence*, 43(2), 219–234.

Erdem, E. & Lifschitz, V. (2003). Tight logic programs. *Theory and Practice of Logic Programming*, 3(4), 499–518.

Fages, F. (1991). A new fixpoint semantics for general logic programs compared with the well-founded and the stable model semantics. *New Generation Computing*, 9(3/4), 425–444.

Fages, F. (1994). Consistency of Clark's completion and existence of stable models. *Journal of Methods of Logic in Computer Science*, 1(1), 51–60.

Fitting, M. (1985). A Kripke-Kleene semantics for logic programs. *Journal of Logic Programming*, 2(4), 295–312.

Gelfond, M. & Lifschitz, V. (1988). The stable model semantics for logic programming. In R. A. Kowalski & K. A. Bowen (Eds.), *Proceedings of the International Logic Programming Conference and Symposium, 5th ICLP/SLP* (pp. 1070–1080). Cambridge, MA: MIT Press.

Gottwald, S. (2001). *A Treatise on Many-Valued Logics*, volume 9 of *Studies in Logic and Computation*. Baldock: Research Studies Press.

Hitzler, P. & Wendt, M. (2005). A uniform approach to logic programming semantics. *Theory and Practice of Logic Programming*, 5(1-2), 93–121.

Hölldobler, S. & Kencana Ramli, C. D. (2009a). Logic programs under three-valued Łukasiewicz semantics. In P. M. Hill & D. S. Warren (Eds.), *Logic Programming, 25th International Conference, ICLP 2009*, volume 5649 of *Lecture Notes in Computer Science* (pp. 464–478). Heidelberg: Springer.

Hölldobler, S. & Kencana Ramli, C. D. (2009b). Logics and networks for human reasoning. In C. Alippi, M. M. Polycarpou, C. G. Panayiotou, & G. Ellinas (Eds.), *International Conference on Artificial Neural Networks, ICANN 2009, Part II*, volume 5769 of *Lecture Notes in Computer Science* (pp. 85–94). Heidelberg: Springer.

Hölldobler, S., Philipp, T., & Wernhard, C. (2011). An abductive model for human reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the AAAI 2011 Spring Symposium*, AAAI Spring Symposium Series Technical Reports (pp. 135–138). Cambridge, MA: AAAI Press.

Kakas, A. C., Kowalski, R. A., & Toni, F. (1993). Abductive logic programming. *Journal of Logic and Computation*, 2(6), 719–770.

Kencana Ramli, C. D. (2009). Logic programs and three-valued consequence operators. Master's thesis, Institute for Artificial Intelligence, Department of Computer Science, Technische Universität Dresden, Dresden.

Kleene, S. C. (1952). *Introduction to Metamathematics*. Amsterdam: North-Holland.

Knauff, M., Rauh, R., & Renz, J. (1997). A cognitive assessment of topological spatial relations: Results from an empirical investigation. In *Proceedings of the 3rd International Conference on Spatial Information Theory, COSIT'97*, volume 1329 of *Lecture Notes in Computer Science* (pp. 193–206). Heidelberg: Springer.

Knauff, M., Rauh, R., & Schlieder, C. (1995). Preferred mental models in qualitative spatial reasoning: A cognitive assessment of Allen's calculus. In *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society, CogSci 1995* (pp. 200–205). Hillsdale, NJ: Lawrence Erlbaum.

Łukasiewicz, J. (1920). O logice trójwartościowej. *Ruch Filozoficzny*, 5, 169–171. English translation: On three-valued logic. In: Łukasiewicz J. and Borkowski L. (ed.). (1990). *Selected Works*, Amsterdam: North Holland, pp. 87–88.

Pereira, L. M., Aparício, J. N., & Alferes, J. J. (1991). The extended stable models of contradiction removal semantics. In P. Barahona, L. M. Pereira, & A. Porto (Eds.), *5th Portuguese AI International Conference (EPIA 91)*, volume 541 of *Lecture Notes in Computer Science* (pp. 105–119). Heidelberg: Springer.

Przymusinski, T. (1988). On the declarative semantics of deductive databases and logic programs. In J. Minker (Ed.), *Foundations of Deductive Databases and Logic Programming* (pp. 193–216). San Francisco, CA: Morgan Kaufmann.

Przymusinski, T. (1989). Every logic program has a natural stratification and an iterated least fixed point model. In A. Silberschatz (Ed.), *Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '89* (pp. 11–21). New York, NY: ACM.

Przymusinski, T. (1990). Well-founded semantics coincides with three-valued stable semantics. *Fundamenta Informaticae*, 13(4), 445–463.

Przymusinski, T. (1994). Well founded and stationary models of logic programs. *Annals of Mathematics and Artificial Intelligence*, 12(3-4), 141–187.

Rescher, N. (1969). *Many-valued logic*. New York, NY: McGraw-Hill.

Ruiz, C. & Minker, J. (1995). Computing stable and partial stable models of extended disjunctive logic programs. In J. Dix, L. Pereira, & T. C. Przymusinski (Eds.), *Non-Monotonic Extensions of Logic Programming*, volume 927 of *Lecture Notes in Computer Science* (pp. 205–229). Heidelberg: Springer.

Satoh, K. & Iwayama, N. (1991). Computing abduction using the TMS. In K. Furukawa (Ed.), *Proceedings of the Eigth International Conference on Logic Programming, ICLP 1991* (pp. 505–518). Cambridge, MA: MIT Press.

Stenning, K. & van Lambalgen, M. (2005). Semantic interpretation as computation in nonmonotonic logic: The real meaning of the suppression task. *Cognitive Science*, 6(29), 916–960.

Stenning, K. & van Lambalgen, M. (2008). *Human Reasoning and Cognitive Science*. A Bradford Book. Cambridge, MA: MIT Press.

Strube, G. (1992). The role of cognitive science in knowledge engineering. In F. Schmalhofer, G. Strube, & T. Wetter (Eds.), *Contemporary Knowledge Engineering and Cognition, First Joint Workshop*, volume 622 of *Lecture Notes in Computer Science* (pp. 159–174). Heidelberg: Springer.

Van Emden, M. H. & Kowalski, R. A. (1976). The semantics of predicate logic as a programming language. *ACM*, 23(4), 733–742.

Van Gelder, A. (1989). The alternating fixpoint of logic programs with negation. In A. Silberschatz (Ed.), *Proceedings of the Eighth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS '89* (pp. 1–10). New York, NY: ACM.

Van Gelder, A., Ross, K. A., & Schlipf, J. S. (1991). The well-founded semantics for general logic programs. *Journal of the ACM*, 38(3), 619–649.

Wernhard, C. (2011). *Forward Human Reasoning Modeled by Logic Programming Modeled by Classical Logic with Circumscription and Projection*. Technical Report Knowledge Representation and Reasoning 11-07, Technische Universität Dresden, Dresden.

Wernhard, C. (2012). Towards a declarative approach to model human reasoning with nonmonotonic logics. In T. Barkowsky, M. Ragni, & F. Stolzenburg (Eds.), *Human Reasoning and Automated Deduction: KI 2012 Workshop Proceedings*, volume SFB/TR 8 Report 032-09/2012 of *Report Series of the Transregional Collaborative Research Center SFB/TR 8 Spatial Cognition* (pp. 41–48). Bremen: Universität Bremen / Universität Freiburg.

Appendix A. Level Mapping Characterization for Weak Completion Semantics and Well-Founded Semantics

We compare weak completion and well-founded semantics by their level mapping characterizations. For this purpose we need to define a *three-valued level mapping* for $\mathcal{P}$ which is a level mapping that may be undefined for some atoms. An *I-three-valued level mapping* l_I for $\mathcal{P}$ is a three-valued level mapping for an interpretation where the domain of l_I is $dom(l_I) = I^\top \cup I^\bot$ and l_I is a function $l_I : I^\top \cup I^\bot \to \mathrm{N}$. All atoms which are unknown under I are not mapped to a number by l_I.

Hitzler & Wendt (2005) characterize well-founded semantics for normal logic programs as follows: Let $\mathcal{P}$ be a normal program, $I = \langle I^\top, I^\bot \rangle$ be a model of $\mathcal{P}$ and l_I be an I-three-valued level mapping of $\mathcal{P}$. *$\mathcal{P}$ is said to satisfy (WF) w.r.t. I and l_I* if for every $A \in dom(l_I)$ one of the following conditions is satisfied:

(WFi) $A \in I^\top$ and there exists a clause $A \leftarrow Body$ in $\mathcal{P}$ such that it holds for all literals L in *Body*: $L \in I^\top$ and $l_I(A) > l_I(L)$.

(WFii) $A \in I^\bot$ and for all clauses $A \leftarrow Body$ in $\mathcal{P}$, one of the following conditions holds:
- **(WFiia)** there exists a literal L in $\mathsf{pos}(Body)$ such that $L \in I^\bot$ and $l_I(A) \geq l_I(L)$,
- **(WFiib)** there exists a literal L in $\mathsf{neg}(Body)$ such that $L \in I^\top$ and $l_I(A) > l_I(L)$.

If $A \in dom(l_I)$ satisfies (WFi), then we say that A *satisfies* (WFi) *with respect to I and l_I*, and similarly if $A \in dom(l_I)$ satisfies (WFii).

Theorem 21. *Let $\mathcal{P}$ be a normal program with the well-founded model M. Then M is the greatest model among all models I for which there exists an I-three-valued level mapping l_I for $\mathcal{P}$ such that $\mathcal{P}$ satisfies (WF) w.r.t. I and l_I.*

Intuitively, a level mapping that satisfies (WF) w.r.t. to all $A \in dom(l_I)$ is acyclic w.r.t. $\langle I^\top, \emptyset \rangle$ and stratified w.r.t. $\langle \emptyset, I^\bot \rangle$.

Kencana Ramli (2009) gives the following level mapping characterization for the least model of the weak completion semantics:

Let $\mathcal{P}$ be a logic program, $I = \langle I^\top, I^\bot \rangle$ be a model of $\mathcal{P}$ and l_I be an I-three-valued level mapping for $\mathcal{P}$. *$\mathcal{P}$ is said to satisfy (L) w.r.t. I and l_I* if for every $A \in dom(l_I)$ one of the following conditions is satisfied:

(WCi) $A \in I^\top$ and there exists a clause $A \leftarrow Body$ in $\mathcal{P}$ such that it holds for all literals L in *Body*: $L \in I^\top$ and $l_I(A) > l_I(L)$.

(WCii) $A \in I^\bot$ and there exists a clause $A \leftarrow Body$ in $\mathcal{P}$ and for all such clauses, one of the following conditions holds:
- **(WCiia)** there exists a literal L in $\mathsf{pos}(Body)$ such that $L \in I^\bot$ and $l_I(A) > l_I(L)$,
- **(WCiib)** there exists a literal L in $\mathsf{neg}(Body)$ such that $L \in I^\top$ and $l_I(A) > l_I(L)$.

If $A \in dom(l_I)$ satisfies (WCi), then we say that A satisfies (WCi) w.r.t. I and l_I, and similarly if $A \in dom(l_I)$ satisfies (WCii).

Theorem 22. *Let $\mathcal{P}$ be a logic program with M, the least model of the weak completion. Then M is the greatest model among all models I for which there exists an I-three-valued level mapping l_I of $\mathcal{P}$ such that $\mathcal{P}$ satisfies (WC) w.r.t. I and l_I.*

Intuitively, the level mapping that satisfies (WC) w.r.t. to all $A \in dom(l_I)$ is acyclic w.r.t. $\langle I^\top, \emptyset \rangle$ and w.r.t. $\langle \emptyset, I^\bot \rangle$.

Both characterizations differ in two conditions: First, consider the conditions (WFii) and (WCii):

(WFii) $A \in I^\bot$ and for all clauses $A \leftarrow Body$ in $\mathcal{P}$, one of the following conditions holds: [...]

(WCii) $A \in I^{\perp}$ and there exists a clause $A \leftarrow \mathit{Body}$ in $\mathcal{P}$ and for all such clauses, one of the following conditions holds: [...]

By condition (WFii) all undefined atoms are in $I^{\perp}$ in the well-founded model whereas under weak completion semantics, they stay unknown. Furthermore, they differ in conditions (WFiia) and (WCiia):

(WFiia) there exists a literal L in $\mathsf{pos}(\mathit{Body})$ such that $L \in I^{\perp}$ and $l(A) \geq l(L)$,
(WCiia) there exists a literal L in $\mathsf{pos}(\mathit{Body})$ such that $L \in I^{\perp}$ and $l(A) > l(L)$,

In a well-founded model, all atoms which are part of a positive cycle are in $I^{\perp}$, whereas under weak completion these atoms stay unknown. Considering Theorem 11 again, we made one restriction and two adaptations:

(1) We restrict the correspondence to tight logic programs because of the difference between condition (WFiia) and condition (WCiia).
(2) Under well-founded semantics we consider $\mathcal{P}^{+}$ instead of $\mathcal{P}$ because well-founded semantics is not defined for programs with negative facts.
(3) For all atoms $A \in \mathsf{undef}(\mathcal{P})$ we introduce an auxiliary atom $\mathsf{n\_}A$ and add the following two clauses $A \leftarrow \neg \mathsf{n\_}A$ and $\mathsf{n\_}A \leftarrow \neg A$, so condition (WFii) does not apply for undefined atoms anymore and A stays undefined.