Computing Science

On the Equivalence between Assumption-Based Argumentation and Logic Programming

Martin Caminada
Department of Computing Science
University of Aberdeen, UK
martin.caminada@abdn.ac.uk

Claudia Schulz
Department of Computing
Imperial College London, UK
claudia.schulz@imperial.ac.uk

Technical Report Series **ABDN–CS–15-01**

Department of Computing Science **June 2015**
University of Aberdeen
King's College
Aberdeen AB24 3UE

On the Equivalence between Assumption-Based Argumentation and Logic Programming

Martin Caminada
Department of Computing Science
University of Aberdeen, UK
martin.caminada@abdn.ac.uk

Claudia Schulz
Department of Computing
Imperial College London, UK
claudia.schulz@imperial.ac.uk

Technical Report ABDN–CS–15-01
Department of Computing Science
University of Aberdeen

Abstract: In the current paper, we re-examine the relationship between Assumption-Based Argumentation (ABA) and logic programming (LP). For this, we specify a procedure that, given a flat ABA frameworks with unique non-assumption contraries, yields an associated logic program such that the 3-valued stable (resp. well-founded, regular, (2-valued) stable, and ideal) models of the logic program coincide with the complete (resp. grounded, preferred, stable, and ideal) assumption labellings of the ABA framework. Moreover, we show how our results on the translation from ABA to LP can be reapplied for a reverse translation from LP to ABA, and observe that some of the existing results in the literature are in fact special cases of our work. Overall, we show that a frequently used fragment of ABA (flat ABA frameworks with unique non-assumption contraries under complete, grounded, preferred, stable, or ideal semantics) can be seen as a form of logic programming.

Keywords: Assumption-Based Argumentation, Logic Programming

1 Introduction

Assumption-Based Argumentation (ABA) [2, 10, 27] has become one of the leading approaches for formal argumentation. On the one hand, it is an instance of Abstract Argumentation under many well-studied semantics [11, 27], on the other it generalizes logic programming, default logic and other non-monotonic reasoning systems [2, 24] and has as such proven useful for explaining [23] as well as visualizing [21] logic programs under certain semantics. In addition to the "normal" notion of acceptability used in many argumentation formalism, ABA is equipped with a dialectical notion of acceptability [26] which has for instance been applied in agent dialogues [13]. ABA has also proven useful in various application domains ranging from medicine [8] over decision making [15] and negotiation [18] to legal reasoning [12].

Although ABA was originally specified in a very general way [2], some of the more recent work (like [10, 27, 24]) has focused on flat ABA frameworks (meaning that no assumption can occur in the head of a rule) with a unique contrary for every assumption. Here, we will study a fragment of this type of ABA framework, to be referred to as *normal ABA framework*, where in addition the contraries of assumptions are non-assumptions. We will focus on some of the most commonly studied ABA semantics (complete, grounded, preferred, stable, and ideal) which we will refer to as *common ABA semantics*.

One particular question that has been studied in the literature is how ABA relates to logic programming (LP). Usually, this is done in the form of a translation from LP to ABA [2, 24, 23] and showing that ABA is powerful enough to capture LP. In the current paper, we go the other way around. That is, we provide a translation from ABA to LP and show that LP is powerful enough to capture a frequently studied fragment of ABA, i.e. normal ABA frameworks under common ABA semantics.

2 Formal Preliminaries

In the current section, we provide a number of key definitions on Assumption-Based Argumentation (ABA) and Logic Programming (LP).

Definition 1. *An* Assumption-Based Argumentation (ABA) Framework *is a tuple* $\langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ *where* $\mathcal{L}$ *is a language,* $\mathcal{R}$ *is a set of inference rules based on this language,* $\mathcal{A} \subseteq \mathcal{L}$ *is a set of assumptions, and* $^{-} : \mathcal{A} \rightarrow \mathcal{L}$ *is a function that maps each assumption* $\chi \in \mathcal{A}$ *to what is called its* contrary $\overline{\chi}$.

We say that an ABA framework is *flat* [2] iff assumptions only occur in the body of the inference rules, and not in the head. Furthermore, we notice that each assumption has a unique contrary. Although this deviates from some generalized work on ABA, where an assumption has a set of possible contraries [16, 17, 13, 14], or where a set of sentences (containing at least one assumption) is associated with a set of sentences which together form the contrary [25], in a lot of work on ABA [11, 10, 27, 22] it is common for the authors to restrict themselves to assumptions with unique contraries as originally defined [2]. In addition, we will here often restrict ourselves to a fragment of ABA where the contrary of an assumption cannot be an assumption, but only a non-assumption, i.e. where $^{-} : \mathcal{A} \rightarrow \mathcal{L} \backslash \mathcal{A}$. In Appendix A it is shown that this does not affect the expressiveness of ABA. In the current paper, we will use the term *normal ABA frameworks* for flat ABA frameworks where assumptions have unique contraries which are non-assumptions.

Definition 2. *Let* $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ *be an ABA framework. An* ABA argument $Asms \vdash x$ *for conclusion* $x \in \mathcal{L}$ *supported by assumptions* $Asms \subseteq \mathcal{A}$ *is a finite tree with nodes labelled with sentences in* $\mathcal{L}$ *or with the special symbol* `TRUE`[1]*, such that:*

- *the root is labelled with* x
- *for every node N:*
  - *if N is a leaf node, then N is labelled with an assumption or with* `TRUE`
  - *if N is not a leaf node and* $z \in \mathcal{L}$ *is the label of N, then there exists a rule in* $\mathcal{R}$ *of the form* $z \leftarrow y_1, \ldots, y_n$ *and either* $n = 0$ *and N has just a single child that is labelled with* `TRUE`*, or* $n > 0$ *and N has* n *children, labelled with* $y_1, \ldots, y_n$ *respectively*

[1] We assume that the special symbol `TRUE`, just like the special symbols `FALSE` and `UNDEFINED` do not occur in any ABA framework. So `TRUE`, `FALSE`, `UNDEFINED` $\notin \mathcal{L}$.

– *Asms is the set of all assumptions labelling leaf nodes*

We say that an ABA argument $Asms \vdash \chi$ is *trivial* iff it consists of a single node, which implies that χ is an assumption and $Asms = \{\chi\}$.

Based on the definition of an ABA argument, we proceed to introduce ABA semantics. For this, we apply the notion of assumption labellings [22].

Definition 3. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an ABA framework. An* assumption labelling *of $\mathcal{F}$ is a total function $\mathcal{L}ab : \mathcal{A} \to \{\text{IN}, \text{OUT}, \text{UNDEC}\}$. We denote by $\text{IN}(\mathcal{L}ab)$ the set of all assumptions labelled* IN *by $\mathcal{L}ab$, and similarly by $\text{OUT}(\mathcal{L}ab)$ and $\text{UNDEC}(\mathcal{L}ab)$ the sets of assumptions labelled* OUT *and* UNDEC*, respectively. An assumption labelling $\mathcal{L}ab$ is called a* complete assumption labelling *of $\mathcal{F}$ iff for each $\chi \in \mathcal{A}$ it holds that:*

1. *if $\mathcal{L}ab(\chi) = \text{IN}$ then for each argument $Asms \vdash \overline{\chi}$ it holds that $Asms \cap \text{OUT}(\mathcal{L}ab) \neq \emptyset$*
2. *if $\mathcal{L}ab(\chi) = \text{OUT}$ then there exists an argument $Asms \vdash \overline{\chi}$ such that $Asms \subseteq \text{IN}(\mathcal{L}ab)$*
3. *if $\mathcal{L}ab(\chi) = \text{UNDEC}$ then there exists an argument $Asms \vdash \overline{\chi}$ such that $Asms \cap \text{OUT}(\mathcal{L}ab) = \emptyset$, and for each argument $Asms \vdash \overline{\chi}$ it holds that $Asms \not\subseteq \text{IN}(\mathcal{L}ab)$*

The following proposition states that for complete assumption labellings, we are free to change the direction of the three if-statements. This will be a useful property for some of the proofs.

Proposition 1. *[22] Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an ABA framework, and let $\mathcal{L}ab$ be an assumption labelling of $\mathcal{F}$. $\mathcal{L}ab$ is a complete assumption labelling of T iff it holds that:*

1. *if for each argument $Asms \vdash \overline{\chi}$ it holds that $Asms \cap \text{OUT}(\mathcal{L}ab) \neq \emptyset$, then $\mathcal{L}ab(\chi) = \text{IN}$*
2. *if there exists an argument $Asms \vdash \overline{\chi}$ such that $Asms \subseteq \text{IN}(\mathcal{L}ab)$, then $\mathcal{L}ab(\chi) = \text{OUT}$*
3. *if there is an argument $Asms \vdash \overline{\chi}$ such that $Asms \cap \text{OUT}(\mathcal{L}ab) = \emptyset$, and for each argument $Asms \vdash \overline{\chi}$ it holds that $Asms \not\subseteq \text{IN}(\mathcal{L}ab)$, then $\mathcal{L}ab(\chi) = \text{UNDEC}$*

We now introduce some new results and definitions which will be needed for the comparison of ABA and LP semantics. Firstly, we observe that the set of IN assumptions of a complete assumption labelling $\mathcal{L}ab_1$ is a subset of or equal to the set of IN assumptions of another complete assumption labelling $\mathcal{L}ab_2$ if and only if the set of OUT assumptions of $\mathcal{L}ab_1$ is a subset of or equal to the set of OUT assumptions of $\mathcal{L}ab_2$.

Lemma 1. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an ABA framework, and let $\mathcal{L}ab_1$ and $\mathcal{L}ab2$ be complete assumption labellings of $\mathcal{F}$. It holds that $\text{IN}(\mathcal{L}ab_1) \subseteq \text{IN}(\mathcal{L}ab_2)$ iff $\text{OUT}(\mathcal{L}ab_1) \subseteq \text{OUT}(\mathcal{L}ab_2)$.*

Proof. "$\Rightarrow$": Assume that $\text{IN}(\mathcal{L}ab_1) \subseteq \text{IN}(\mathcal{L}ab_2)$. Let $\chi \in \text{OUT}(\mathcal{L}ab_1)$. Then, by the definition of a complete assumption labelling (Definition 3) there exists an ABA argument $Asms \vdash \overline{\chi}$ with $Asms \subseteq \text{IN}(\mathcal{L}ab_1)$. Since $\text{IN}(\mathcal{L}ab_1) \subseteq \text{IN}(\mathcal{L}ab_2)$ it follows that $Asms \subseteq \text{IN}(\mathcal{L}ab_2)$. So by Lemma 1 in [22], $\chi \in \text{OUT}(\mathcal{L}ab_2)$.
"$\Leftarrow$": Assume that $\text{OUT}(\mathcal{L}ab_1) \subseteq \text{OUT}(\mathcal{L}ab_2)$. Let $\chi \in \text{IN}(\mathcal{L}ab_1)$. Then, by the definition of a complete assumption labelling (Definition 3) it holds that each ABA argument $Asms \vdash \overline{\chi}$ has $Asms \cap \text{OUT}(\mathcal{L}ab_1) \neq \emptyset$. Since $\text{OUT}(\mathcal{L}ab_1) \subseteq \text{OUT}(\mathcal{L}ab_2)$ it follows that $Asms \cap \text{OUT}(\mathcal{L}ab_2) \neq \emptyset$. So by Lemma 1 in [22], $\chi \in \text{IN}(\mathcal{L}ab_2)$. $\square$

We now extend the notion of assumption labellings from the complete semantics as introduced in [22] to other well-known ABA semantics, which were previously defined in terms of extensions rather than labellings [2, 27]. Note that there exists a one-to-one correspondence between the assumption labellings and the assumption extensions of an ABA framework. In essence, the set of IN-labelled assumptions of a complete (resp. grounded, preferred, stable, or ideal) assumption labelling constitutes a complete (resp. grounded, preferred, stable, or (maximal) ideal) assumption extension.

Definition 4. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an ABA framework. A complete assumption labelling $\mathcal{L}ab$ of $\mathcal{F}$ is called:*

1. *a* grounded assumption labelling *iff* $\text{IN}(\mathcal{L}ab)$ *is minimal (w.r.t. $\subseteq$) among all complete assumption labellings of $\mathcal{F}$*
2. *a* preferred assumption labelling *iff* $\text{IN}(\mathcal{L}ab)$ *is maximal (w.r.t. $\subseteq$) among all complete assumption labellings of $\mathcal{F}$*
3. *a* stable assumption labelling *iff* $\text{UNDEC}(\mathcal{L}ab) = \emptyset$
4. *an* ideal assumption labelling *iff* $\text{IN}(\mathcal{L}ab)$ *is maximal (w.r.t. $\subseteq$) among all complete assumption labellings of $\mathcal{F}$ with* $\text{IN}(\mathcal{L}ab) \subseteq \bigcap\{\text{IN}(\mathcal{L}ab_{pref}) \mid \mathcal{L}ab_{pref}$ *is a preferred assumption labelling of* $\mathcal{F}\}$

As complete, grounded, preferred, stable, and ideal semantics are well-studied within ABA, we refer to these as the *common ABA semantics*.

Now that we have introduced the preliminaries of Assumption-Based Argumentation, we shift our attention to logic programming. We start with formally introducing the notion of a logic program.

Definition 5. *A* logic programming rule *is an expression* $x \leftarrow y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m$ *($n \geq 0$, $m \geq 0$) where x, each y_i ($1 \leq i \leq n$) and each z_j ($1 \leq j \leq m$) is an atom, and* `not` *represents negation as failure. We say that x is the* head *of the rule, and* $y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m$ *the body of the rule. Moreover, we say that $y_1, \ldots, y_n$ is the strong part of the body, and* $\texttt{not}\ z_1, \ldots, \texttt{not}\ z_m$ *is the weak part of the body. We assume the presence of three special atoms* `TRUE`*,* `FALSE` *and* `UNDEFINED` *that can only occur in the strong part of the body. A* NAF-literal *is an expression* $\texttt{not}\ w$*, where w is an atom. We say a rule is NAF-free iff it does not contain any NAF-literal (that is, iff $m = 0$). A* logic program *P is a finite set of logic programming rules. A logic program is NAF-free iff each of its rules is NAF-free. A logic program is called* normal[2] *iff none of its rules contains the special atoms* `TRUE`*,* `FALSE` *or* `UNDEFINED`*. The Herbrand Base of a logic program P (written as HB_P) is the set of all atoms in P. We denote by* $HB_P^{\texttt{not}} = \{\texttt{not}\ w \mid w \in HB_P\}$ *the set of all NAF literals of atoms in the Herbrand Base.*

In the following, we recall the definitions of LP semantics.

Definition 6. *A* 3-valued interpretation *of a logic program P is a pair $\langle T, F \rangle$ where $T, F \subseteq HB_P$ and $T \cap F = \emptyset$.*

[2]In general the term "normal" is used for logic programs without strong negation, which is the case for the logic programs considered here. The notion of "normal logic program" used here is thus a special case of its general usage.

When P is a NAF-free logic program (possibly containing `TRUE`, `FALSE` or `UNDEFINED`), we write $\Phi(P)$ for its unique minimal 3-valued model $\langle T_\Phi, F_\Phi\rangle$ in the sense of [19] (with minimal T_Φ and maximal F_Φ). We proceed to define the well-known Gelfond-Lifschitz reduct in the context of a 3-valued interpretation as done in [19].

Definition 7. *The reduct of a logic program P w.r.t. a 3-valued interpretation $\mathcal{M}od = \langle T, F\rangle$, written as $P^{\mathcal{M}od}$ is obtained by replacing each NAF literal* $\texttt{not}\ x$ *by* `TRUE` *if $x \in F$, by* `FALSE` *if $x \in T$, and by* `UNDEFINED` *otherwise.*

Since $P^{\mathcal{M}od}$ is a NAF-free program, it has a unique minimal 3-valued model, written as $\Phi(P^{\mathcal{M}od})$. We now recall different logic programming semantics which are based on 3-valued models [19]. Notice that although our definition of well-founded and regular models is slightly different from what is in the literature, equivalence is shown in [7]. We also define a new semantics based on 3-valued models, namely *ideal models*, inspired by the idea of ideal scenarios for logic programs [1]. In fact our ideal models coincide with ideal scenarios, as shown in Appendix B.

Definition 8. *Let P be a logic program and $\mathcal{M}od = \langle T, F\rangle$ a 3-valued interpretation of P. We say that $\mathcal{M}od$ is:*

- *a* 3-valued stable model *iff $\Phi(P^{\mathcal{M}od}) = \mathcal{M}od$*
- *a* well-founded model *iff $\mathcal{M}od$ is a 3-valued stable model where T is minimal (w.r.t. $\subseteq$) among all 3-valued stable models of P*
- *a* regular model *[29] iff $\mathcal{M}od$ is a 3-valued stable model where T is maximal (w.r.t. $\subseteq$) among all 3-valued stable models of P*
- *a* (2-valued) stable model *iff $\mathcal{M}od$ is a 3-valued stable model where $T \cup F = HB_P$*
- *an* ideal model *iff $\mathcal{M}od$ is a 3-valued stable model where T is maximal (w.r.t. $\subseteq$) among all 3-valued stable models of P with $T \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg}\rangle$ is a regular model of $P\}$*

We sometimes refer to 3-valued stable, well-founded, regular, (2-valued) stable and ideal semantics as the *common LP semantics.*

Just like was done for ABA, we can also define arguments in the context of logic programming.

Definition 9. *Let P be a logic program. An* argument *for $x \in HB_P$ (the conclusion or claim) is a tree with nodes labelled with formulas in HB_P, $HB_P^{\texttt{not}}$, or with the special symbols* `TRUE`, `FALSE` *or* `UNDEFINED` *such that:*

- *the root is labelled with x*
- *for every node N:*
  - *if N is a leaf node, then N is labelled with a NAF literal or with one of the special symbols* `TRUE`, `FALSE` *or* `UNDEFINED`
  - *if N is not a leaf node and z is the label of N, then there exists a rule in P of the form $z \leftarrow y_1, \ldots, y_n, \texttt{not}\ w_1, \ldots, \texttt{not}\ w_m$ and either $m + n = 0$ and N has just a single child, that is labelled with* `TRUE`, *or $n + m > 0$ and N has $n + m$ children, labelled with $y_1, \ldots, y_n, \texttt{not}\ w_1, \ldots, \texttt{not}\ w_m$ respectively*

Proposition 2. *Let P be a NAF-free logic program, let $\langle T, F\rangle$ be $\Phi(P)$, and let $x \in HB_P$. It holds that:*

1. *$x \in T$ iff there exists an argument for x where every leaf node is labelled with* `TRUE`
2. *$x \in F$ iff each argument for x has at least one leaf node that is labelled with* `FALSE`

3 Translating ABA Theories to Logic Programs

In order to compare ABA to logic programming, we first introduce a translation from a normal ABA framework to a logic program. The idea is to take the rules of the ABA framework and substitute each assumption by the NAF literal of its contrary. This means that different assumptions might be substituted by the same NAF literal if they have the same contrary.

Definition 10. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-}\rangle$ be a normal ABA framework. We define the associated logic program $P_{\mathcal{F}}$ as $\{x \leftarrow y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m \mid x \leftarrow y_1, \ldots, y_n, \zeta_1, \ldots, \zeta_m \in \mathcal{R}$ and $\forall i \in \{1 \ldots m\} : \overline{\zeta_i} = z_i\}$, where $HB_{P_{\mathcal{F}}} = \{w \mid w$ or $\texttt{not}\ w$ occurs in a rule of $P_{\mathcal{F}}\}$.*

As we assume that no ABA framework contains the special symbols `TRUE`, `FALSE` or `UNDEFINED`, the associated logic program will not contain any of these symbols, so $P_{\mathcal{F}}$ is a normal logic program. Note also that since $\mathcal{F}$ is a normal ABA framework, i.e. the contrary of assumptions are non-assumptions, $HB_{P_{\mathcal{F}}}$ contains no atoms which are assumptions in $\mathcal{F}$.

Example 1. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-}\rangle$ be an ABA framework with $\mathcal{A} = \{\alpha, \beta, \gamma\}$, $\mathcal{L} = \mathcal{A} \cup \{a, b, c, d, e\}$, $\overline{\alpha} = a$, $\overline{\beta} = b$, $\overline{\gamma} = e$ and $\mathcal{R} = \{a \leftarrow \beta;\ b \leftarrow \alpha;\ c \leftarrow;\ d \leftarrow b, c\}$.*
The associated logic program $P_{\mathcal{F}}$ is: $\{a \leftarrow \texttt{not}\ b;\ b \leftarrow \texttt{not}\ a;\ c \leftarrow;\ d \leftarrow b, c\}$ with $HB_{P_{\mathcal{F}}} = \{a, b, c, d\}$.

It can be observed that the translation from a normal ABA framework to a logic program can also be applied to translate ABA arguments (Definition 2) to LP arguments (Definition 9). For instance, in the above example, the ABA argument for conclusion d, constructed by applying the rules "$b \leftarrow \alpha$", "$c \leftarrow$" and "$d \leftarrow b, c$" has an associated LP argument for conclusion d, constructed by applying the rules "$b \leftarrow \texttt{not}\ a$", "$c \leftarrow$" and "$d \leftarrow b, c$". In general, we observe that each non-trivial ABA argument under $\mathcal{F}$ has an associated LP argument under $P_{\mathcal{F}}$, and vice versa.

One of the main aims of the current paper is to examine how ABA semantics are related to logic programming semantics. For this, we introduce the functions `Lab2Mod` and `Mod2Lab` to convert between ABA assumption labellings and logic programming models.

To convert an assumption labelling to a 3-valued interpretation, we start by "inverting" the labelling. That is, we construct an interpretation $\langle T', F'\rangle$ where T' contains the contraries of the assumptions that are OUT, whereas F' contains the contraries of the assumptions that are IN. However, since we started with assumptions, this will only yield the status of atoms which are contraries of assumptions. In order to obtain the status of *all* atoms in the logic program (including those that are not the contrary of any assumption in the ABA framework) we perform a simple trick: apply the Gelfond-Lifschitz reduct.

To convert a 3-valued interpretation to an assumption labelling, the idea is again to "invert" the interpretation. The assumptions whose contrary is in F will be labelled IN. The assumptions whose contrary is in T will be labelled OUT. The assumptions whose contrary is in the Herbrand Base, but not in T or F will be labelled UNDEC. The only remaining case is what to do with the assumptions whose contrary is not even in the Herbrand Base. This case occurs if there exists an assumption in the ABA framework which is not part of any inference rule itself and nor is its contrary. As these assumptions cannot have any attackers, they will simply be labelled IN.

Definition 11. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be a normal ABA framework and let $P_{\mathcal{F}}$ be the associated logic program. We define a function* `Lab2Mod` *that, given an assumption labelling $\mathcal{L}ab$ of $\mathcal{F}$, yields the 3-valued interpretation $\Phi(P_{\mathcal{F}}^{\langle T', F' \rangle})$ where $T' = \{\overline{\chi} \mid \chi \in \text{OUT}(\mathcal{L}ab)\} \cap HB_{P_{\mathcal{F}}}$ and $F' = \{\overline{\chi} \mid \chi \in \text{IN}(\mathcal{L}ab)\} \cap HB_{P_{\mathcal{F}}}$. We also define a function* `Mod2Lab` *that, given a 3-valued interpretation $\langle T, F \rangle$ of $P_{\mathcal{F}}$, yields an assumption labelling $\mathcal{L}ab$ of $\mathcal{F}$ with $\text{IN}(\mathcal{L}ab) = \{\chi \in \mathcal{A} \mid \overline{\chi} \in F\} \cup \{\chi \in \mathcal{A} \mid \overline{\chi} \notin HB_{P_{\mathcal{F}}}\}$, $\text{OUT}(\mathcal{L}ab) = \{\chi \in \mathcal{A} \mid \overline{\chi} \in T\}$ and $\text{UNDEC}(\mathcal{L}ab) = \{\chi \in \mathcal{A} \mid \overline{\chi} \in HB_{P_{\mathcal{F}}} \setminus (T \cup F)\}$*

We observe that the functions `Lab2Mod` and `Mod2Lab` provide a one-to-one mapping between the complete assumption labellings of $\mathcal{F}$ and the 3-valued stable models of $P_{\mathcal{F}}$.

Theorem 1. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be a normal ABA framework and let $P_{\mathcal{F}}$ be the associated logic program. It holds that*

1. *if $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$ then* `Lab2Mod`*($\mathcal{L}ab$) is a 3-valued stable model of $P_{\mathcal{F}}$*
2. *if $\mathcal{M}od$ is a 3-valued stable model of $P_{\mathcal{F}}$ then* `Mod2Lab`*($\mathcal{M}od$) is a complete assumption labelling of $\mathcal{F}$*
3. *when restricted to complete assumption labellings and 3-valued stable models,* `Lab2Mod` *and* `Mod2Lab` *become bijections which are each other's inverses*

Proof. 1. Let $\mathcal{L}ab$ be a complete assumption labelling of $\mathcal{F}$ and let $\langle T, F \rangle$ be `Lab2Mod`($\mathcal{L}ab$). That is, $\langle T, F \rangle = \Phi(P_{\mathcal{F}}^{\langle T', F' \rangle})$ where $T' = \{\overline{\chi} \mid \chi \in \text{OUT}(\mathcal{L}ab)\} \cap HB_{P_{\mathcal{F}}}$ and $F' = \{\overline{\chi} \mid \chi \in \text{IN}(\mathcal{L}ab)\} \cap HB_{P_{\mathcal{F}}}$. We first observe that $\langle T', F' \rangle$ is a well-defined 3-valued interpretation of $P_{\mathcal{F}}$. This is because $T', F' \subseteq HB_{P_{\mathcal{F}}}$ and $T' \cap F' = \emptyset$, the latter following from the facts that $\text{IN}(\mathcal{L}ab) \cap \text{OUT}(\mathcal{L}ab) = \emptyset$ and that two assumptions that have the same contrary also have the same label in $\mathcal{L}ab$. From the fact that $\langle T', F' \rangle$ is a well-defined 3-valued interpretation of $P_{\mathcal{F}}$, it follows that also $\langle T, F \rangle = \Phi(P_{\mathcal{F}}^{\langle T', F' \rangle})$ is a well-defined 3-valued interpretation of $P_{\mathcal{F}}$.
We proceed to show that applying the Gelfond-Lifschitz reduct (with $\langle T', F' \rangle$) does not change the status of any of the NAF-literals in $P_{\mathcal{F}}$. Let "`not` x" be a NAF-literal in some rule of $P_{\mathcal{F}}$. We distinguish three cases:

   (a) $x \in T'$. Then, by definition of `Lab2Mod`, there exists an assumption χ with $\overline{\chi} = x$ and $\chi \in \text{OUT}(\mathcal{L}ab)$. From the fact that $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$, it follows that there exists an ABA argument $Asms \vdash x$ (under $\mathcal{F}$) with $Asms \subseteq \text{IN}(\mathcal{L}ab)$. Moreover, as x is not an assumption, this argument is non-trivial. From the definition of $P_{\mathcal{F}}$ it then follows that there also exists an LP argument (under $P_{\mathcal{F}}$) for x and with each of its leaf nodes labelled either with `TRUE` or with "`not` z", where $z \in F'$. From the definition of $P_{\mathcal{F}}^{\langle T', F' \rangle}$ it then follows that there exists an argument (under $P_{\mathcal{F}}^{\langle T', F' \rangle}$) for x where every leaf node is labelled with `TRUE`. From Proposition 2 it then follows that $x \in T$, with $\langle T, F \rangle$ being the unique minimal model of $P_{\mathcal{F}}^{\langle T', F' \rangle}$.

(b) $x \in F'$. Then, by definition of `Lab2Mod`, there exists an assumption χ with $\overline{\chi} = x$ and $\chi \in \text{IN}(\mathcal{L}ab)$. From the fact that $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$, it follows that for each ABA argument $Asms \vdash x$ (under $\mathcal{F}$) (which has to be non-trivial, since x is not an assumption) it holds that $Asms \cap \text{OUT}(\mathcal{L}ab) \neq \emptyset$. So for each ABA argument $Asms \vdash x$ (under $\mathcal{F}$) there is a $\zeta \in Asms$ with $\zeta \in \text{OUT}(\mathcal{L}ab)$. From the definition of $P_{\mathcal{F}}$ it then follows that each LP argument (under $P_{\mathcal{F}}$) for x has a leaf node "`not` z" with $z \in T'$. From the definition of $P_{\mathcal{F}}^{\langle T',F' \rangle}$ it then follows that each LP argument (under $P_{\mathcal{F}}^{\langle T',F' \rangle}$) has a leaf node labelled with `FALSE`. From Proposition 2 it then follows that $x \in F$ with $\langle T, F \rangle$ being the unique minimal model of $P_{\mathcal{F}}^{\langle T',F' \rangle}$.

(c) $x \in HB_{P_{\mathcal{F}}} \setminus (T' \cup F')$. We first observe that $x \notin T'$. Hence, by definition of `Lab2Mod` there is no assumption χ with $\overline{\chi} = x$ and $\chi \in \text{OUT}(\mathcal{L}ab)$. That is, for each assumption χ with $\overline{\chi} = x$ it holds that $\chi \notin \text{OUT}(\mathcal{L}ab)$. From the fact that $\mathcal{L}ab$ is a complete assumption labelling it then follows (point 2 of Proposition 1) that there is no ABA argument $Asms \vdash x$ with $Asms \subseteq \text{IN}(\mathcal{L}ab)$. Hence, for each ABA argument $Asms \vdash x$ (which has to be non-trivial, since x is not an assumption) there is a $\zeta \in Asms$ with $\zeta \notin \text{IN}(\mathcal{L}ab)$. From the definition of $P_{\mathcal{F}}$ it then follows that each LP argument for x under $P_{\mathcal{F}}$ has a leaf node labelled with "`not` z" with $z \notin F'$. From the definition of $P_{\mathcal{F}}^{\langle T',F' \rangle}$, it follows that each LP argument for x under $P_{\mathcal{F}}^{\langle T',F' \rangle}$ has a leaf node labelled that is not labelled with `TRUE`. From Proposition 2 it then follows that $x \notin T$, with $\langle T, F \rangle$ being the unique minimal model of $P_{\mathcal{F}}^{\langle T',F' \rangle}$.
Next, we observe that $x \notin F'$. Hence, by definition of `Lab2Mod` there is no assumption χ with $\overline{\chi} = x$ and $\chi \in \text{IN}(\mathcal{L}ab)$. That is, for each assumption χ with $\overline{\chi} = x$ it holds that $\chi \notin \text{IN}(\mathcal{L}ab)$. From the fact that $\mathcal{L}ab$ is a complete assumption labelling it then follows (point 1 of Proposition 1) that there exists an ABA argument $Asms \vdash x$ (which has to be non-trivial, since x is not an assumption) with $Asms \cap \text{OUT}(\mathcal{L}ab) = \emptyset$. That is, there exists an ABA argument (under $\mathcal{F}$) for x without any leaf node that is labelled with an assumption $\zeta \in \text{OUT}(\mathcal{L}ab)$. From the definition of $P_{\mathcal{F}}$ it then follows that there exists an LP argument (under $P_{\mathcal{F}}$) for x without any leaf node that is labelled with "`not` z" where $z \in T'$. From the definition of $P_{\mathcal{F}}^{\langle T',F' \rangle}$ it then follows that there exists an argument (under $P_{\mathcal{F}}^{\langle T',F' \rangle}$) for x without any leaf node that is labelled `FALSE`. From Proposition 2 it then follows that $x \notin F$, with $\langle T, F \rangle$ being the unique minimal model of $P_{\mathcal{F}}^{\langle T',F' \rangle}$.
So, overall we obtain that $x \in HB_{P_{\mathcal{F}}}$, $x \notin T$ and $x \notin F$, so $x \in HB_{P_{\mathcal{F}}} \setminus T \cup F$.

So, overall we obtained that:
if $x \in T'$ then $x \in T$,
if $x \in F'$ then $x \in F$ and
if $x \in HB_{P_{\mathcal{F}}} \setminus (T' \cup F')$ then $x \in HB_{P_{\mathcal{F}}} \setminus (T \cup F)$.
So $\langle T', F' \rangle$ and $\langle T, F \rangle$ agree on the NAF-literals of $P_{\mathcal{F}}$. It should be noted that whenever two 3-valued interpretations $\mathcal{M}od_1$ and $\mathcal{M}od_2$ of some logic program P agree on the NAF-literals of P, the respective reducts $P^{\mathcal{M}od_1}$ and $P^{\mathcal{M}od_2}$ are equal (after all, for determining the Gelfond-Lifschitz reduct, only the NAF literals are relevant). Hence, in our particular case, we have that $P_{\mathcal{F}}^{\langle T',F' \rangle} = P_{\mathcal{F}}^{\langle T,F \rangle}$, so also $\Phi(P_{\mathcal{F}}^{\langle T',F' \rangle}) = \Phi(P_{\mathcal{F}}^{\langle T,F \rangle})$.

From $\Phi(P_{\mathcal{F}}^{\langle T',F'\rangle}) = \langle T, F\rangle$ it then directly follows that $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T,F\rangle})$, so $\langle T, F\rangle$ is a 3-valued stable model of $P_{\mathcal{F}}$. As by definition $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T',F'\rangle})$ it directly follows that $\Phi(P_{\mathcal{F}}^{\langle T',F'\rangle})$ is a 3-valued stable model of $P_{\mathcal{F}}$.

2. Let $\mathcal{M}od$ be a 3-valued stable model of $P_{\mathcal{F}}$ and let $\mathcal{L}ab = \mathtt{Mod2Lab}(\mathcal{M}od)$. Let $\chi \in \mathcal{A}$. We distinguish three cases:

   (a) $\chi \in \textsc{in}(\mathcal{L}ab)$. Then $\chi \in \{\chi \in \mathcal{A} \mid \overline{\chi} \in F\} \cup \{\chi \in \mathcal{A} \mid \overline{\chi} \notin HB_{P_{\mathcal{F}}}\}$. We distinguish two cases:

      i. $\overline{\chi} \in F$. From the fact that $\langle T, F\rangle$ is a 3-valued stable model of $P_{\mathcal{F}}$ it follows that $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T,F\rangle})$. Hence, the fact that $\overline{\chi} \in F$ means (Proposition 2) that each LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}^{\langle T,F\rangle}$ has a leaf node labelled with `FALSE`. From the definition of $P_{\mathcal{F}}^{\langle T,F\rangle}$ this then implies that each LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}$ has a leaf node labelled with some "`not` z" where $z \in T$. From the definition of $P_{\mathcal{F}}$ it then follows that each ABA argument for $\overline{\chi}$ under $\mathcal{F}$ (which has to be non-trivial, since $\overline{\chi}$ is not an assumption) has an assumption ζ that is labelled $\textsc{out}$ by $\mathcal{L}ab$. That is, for each ABA argument $Asms \vdash \overline{\chi}$ under $\mathcal{F}$ it holds that $Asms \cap \textsc{out}(\mathcal{L}ab) \neq \emptyset$.

      ii. $\overline{\chi} \notin HB_{P_{\mathcal{F}}}$. That is, $\overline{\chi}$ does not occur in any rule of $P_{\mathcal{F}}$, so also not in any rule of $\mathcal{F}$. Therefore, there is no ABA argument (under $\mathcal{F}$) for $\overline{\chi}$ (also not a trivial one, since $\overline{\chi}$ is not an assumption). Hence, trivially, for each ABA argument $Asms \vdash \overline{\chi}$ it holds that $Asms \cap \textsc{out}(\mathcal{L}ab) \neq \emptyset$.

   (b) $\chi \in \textsc{out}(\mathcal{L}ab)$. Then $\overline{\chi} \in T$. From the fact that $\langle T, F\rangle$ is a 3-valued stable model of $P_{\mathcal{F}}$ it follows that $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T,F\rangle})$. Hence, the fact that $\overline{\chi} \in T$ means (Proposition 2) that there exists an LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}^{\langle T,F\rangle}$ of which each leaf node is labelled with `TRUE`. From the definition of $P_{\mathcal{F}}^{\langle T,F\rangle}$ it then follows that there exists an LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}$ of which each leaf node is labelled either with `TRUE` or with some "`not` z" with $z \in F$. From the definition of $P_{\mathcal{F}}$ it then follows that there exists an ABA argument for $\overline{\chi}$ under $\mathcal{F}$ of which each leaf node is labelled with either `TRUE` or with some assumption ζ with $\zeta \in \textsc{in}(\mathcal{L}ab)$. That is, there exists an ABA argument $Asms \vdash \overline{\chi}$ with $Asms \subseteq \textsc{in}(\mathcal{L}ab)$.

   (c) $\chi \in \textsc{undec}(\mathcal{L}ab)$. Then $\overline{\chi} \in HB_{P_{\mathcal{F}}} \setminus (T \cup F)$. From the fact that $\langle T, F\rangle$ is a 3-valued stable model of $P_{\mathcal{F}}$ it follows that $\Phi(P_{\mathcal{F}}^{\langle T,F\rangle}) = \langle T, F\rangle$.
   The fact that $\overline{\chi} \notin T$ then implies that there is no LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}^{\langle T,F\rangle}$ where each leaf node is labelled with `TRUE`. From the definition of $P_{\mathcal{F}}^{\langle T,F\rangle}$ it then follows that there is no LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}$ (also no trivial one, since $\overline{\chi}$ is not an assumption) where each leaf node is labelled with `TRUE` or with some "`not` z" where $z \in F$. From the definition of $P_{\mathcal{F}}$ it then follows that there is no ABA argument for $\overline{\chi}$ under $\mathcal{F}$ where each leaf node is labelled with `TRUE` or with some assumption ζ with $\zeta \in \textsc{in}(\mathcal{L}ab)$. That is, there is no ABA argument $Asms \vdash \overline{\chi}$ with $Asms \subseteq \textsc{in}(\mathcal{L}ab)$.
   Similarly, the fact that $\overline{\chi} \notin F$ implies that there exists an LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}^{\langle T,F\rangle}$ that does not have any leaf that is labelled with `FALSE`. From the definition of $P_{\mathcal{F}}^{\langle T,F\rangle}$ it follows that there exists an LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}$ that does not

have any leaf that is labelled with "`not` z" with $z \in T$. From the definition of $P_{\mathcal{F}}$ it then follows that there exists an ABA argument for $\overline{\chi}$ under $\mathcal{F}$ that does not have any leaf that is labelled with an assumption ζ with $\zeta \in \text{OUT}(\mathcal{L}ab)$. That is, there exists an ABA argument $Asms \vdash \overline{\chi}$ with $Asms \cap \text{OUT}(\mathcal{L}ab) = \emptyset$.
So overall, we obtain that there is no ABA argument $Asms \vdash \overline{\chi}$ with $Asms \subseteq \text{IN}(\mathcal{L}ab)$, and there is an ABA argument $Asms \vdash \overline{\chi}$ with $Asms \cap \text{OUT}(\mathcal{L}ab) = \emptyset$.

To sum up, we have observed that:
if $\chi \in \text{IN}(Lab)$ then for each ABA argument $Asms \vdash \overline{\chi}$ it holds that $Asms \cap \text{OUT}(\mathcal{L}ab)$,
if $\chi \in \text{OUT}(\mathcal{L}ab)$ then there exists an ABA argument $Asms \vdash \overline{\chi}$ with $Asms \subseteq \text{IN}(\mathcal{L}ab)$ and
if $\chi \in \text{UNDEC}(\mathcal{L}ab)$ then there is no ABA argument $Asms \vdash \overline{\chi}$ with $Asms \subseteq \text{IN}(\mathcal{L}ab)$ and there is an ABA argument $Asms \vdash \overline{\chi}$ with $Asms \cap \text{OUT}(\mathcal{L}ab) = \emptyset$.
This means the conditions of Definition 3 are satisfied, so $\mathcal{L}ab$ is a complete assumption labelling.

3. It suffices to prove that:
if $\mathcal{L}ab$ is a complete assumption labelling, then $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab)) = \mathcal{L}ab$, and
if $\mathcal{M}od$ is is a 3-valued stable model, then $\texttt{Lab2Mod}(\texttt{Mod2Lab}(\mathcal{M}od)) = \mathcal{M}od$.
As for the first equivalence that has to be proved, let $\mathcal{L}ab$ be a complete assumption labelling of $\mathcal{F}$, and let $\chi \in \mathcal{A}$. We distinguish four cases.

   (a) $\chi \in \text{IN}(\mathcal{L}ab)$ and $\overline{\chi} \in HB_{P_{\mathcal{F}}}$. Then, by definition of F' (w.r.t. $\texttt{Lab2Mod}(\mathcal{L}ab)$) it follows that $\overline{\chi} \in F'$. As we have observed earlier (in point 1 of the current theorem) it holds that if the contrary of a particular assumption is in F', then it is also in F. Hence, $\overline{\chi} \in F$, with $\langle T, F\rangle = \texttt{Lab2Mod}(\mathcal{L}ab)$. From the definition of `Mod2Lab` it then follows that χ is labelled IN by $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$.

   (b) $\chi \in \text{IN}(\mathcal{L}ab)$ and $\overline{\chi} \notin HB_{P_{\mathcal{F}}}$. We first observe that, also in this case, $\texttt{Lab2Mod}(\mathcal{L}ab)$ is well-defined. From the definition of `Mod2Lab` it then follows that χ is labelled IN by $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$.

   (c) $\chi \in \text{OUT}(\mathcal{L}ab)$. From point 2 or Definition 3 it follows that there exists an ABA argument for $\overline{\chi}$ under $\mathcal{F}$ (which has to be non-trivial, since $\overline{\chi}$ is not an assumption). This means there is also an LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}$, so $\overline{\chi} \in HB_{P_{\mathcal{F}}}$. It then follows that $\overline{\chi} \in T'$. As we have observed earlier (in point 1 of the current theorem) it holds that if the contrary of a particular assumption is in T' then it is also in T. Hence, $\overline{\chi} \in T$, with $\langle T, F\rangle = \texttt{Lab2Mod}(\mathcal{L}ab)$. From the definition of `Mod2Lab` it then follows that χ is labelled OUT by $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$.

   (d) $\chi \in \text{UNDEC}(\mathcal{L}ab)$. From point 3 of Definition 3 it follows that there exists an ABA argument for $\overline{\chi}$ under $\mathcal{F}$ (which is non-trivial, since $\overline{\chi}$ is not an assumption). This means there is also an LP argument for $\overline{\chi}$ under $P_{\mathcal{F}}$, so $\overline{\chi} \in HB_{P_{\mathcal{F}}}$. Furthermore, from the definition of T' and F' it follows that $\overline{\chi} \notin T'$ and $\overline{\chi} \notin F'$. Hence, $\overline{\chi} \in HB_{P_{\mathcal{F}}} \setminus (T' \cup F')$. As we have observed earlier (in point 1 of the current theorem) it holds that if the contrary of a particular assumption is in $HB_{P_{\mathcal{F}}} \setminus (T' \cup F')$ then it is also in $HB_{P_{\mathcal{F}}} \setminus (T \cup F)$. Hence, $\overline{\chi} \in HB_{P_{\mathcal{F}}} \setminus (T \cup F)$, with $\langle T, F\rangle = \texttt{Lab2Mod}(\mathcal{L}ab)$. From the definition of `Mod2Lab` it then follows that χ is labelled UNDEC by $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$.

So overall, we observe that if χ is labelled IN (respectively OUT or UNDEC) by $\mathcal{L}ab$ then χ is labelled IN (respectively OUT or UNDEC) by $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$, so $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$
$\supseteq \mathcal{L}ab$. Furthermore, $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$ does not assign any additional labels other than the ones assigned by $\mathcal{L}ab$: It can easily be verified that $\mathcal{L}ab$ and $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab))$ label the same set of assumptions $\mathcal{A}$. Then, since $\text{IN}(\mathcal{L}ab) \cup \text{OUT}(\mathcal{L}ab) \cup \text{UNDEC}(\mathcal{L}ab) = \mathcal{A}$, it follows that $\texttt{Mod2Lab}(\texttt{Lab2Mod}(\mathcal{L}ab)) = \mathcal{L}ab$.
We now proceed to the second equivalence that has to be proved. Let $\mathcal{M}od = \langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle$ be a 3-valued stable model of $P_{\mathcal{F}}$. Let $x \in HB_{P_{\mathcal{F}}}$. We distinguish three cases.

(a) $x \in T_{\mathcal{M}od}$. Then from $\mathcal{M}od$ being a 3-valued stable model of $P_{\mathcal{F}}$, it follows that $\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle = \Phi(P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle})$. From Proposition 2 it then follows that there is an LP argument for x under $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$ of which each leaf node is labelled with $\texttt{TRUE}$. By definition of $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$, this implies that there exists an LP argument for x of which each leaf node is labelled either with $\texttt{TRUE}$ or with "$\texttt{not}\ z$" where $z \in F_{\mathcal{M}od}$. By definition of $P_{\mathcal{F}}$ and $\texttt{Mod2Lab}$, it then follows that there exists a (non-trivial) ABA argument for x under $\mathcal{F}$ of which each leaf node is labelled either with $\texttt{TRUE}$ or with some assumption ζ that is labelled IN by $\texttt{Mod2Lab}(\mathcal{M}od)$. From the definition of $\texttt{Lab2Mod}$, it then follows that for each such ζ it holds that $\overline{\zeta} \in F'$, with respect to $\texttt{Lab2Mod}(\texttt{Mod2Lab}(\mathcal{M}od))$. It then follows that there is an LP argument for x under $P_{\mathcal{F}}^{\langle T', F'\rangle}$ of which each leaf node is labelled with $\texttt{TRUE}$. From Proposition 2 it then follows that $x \in T$ with $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T', F'\rangle})$.

(b) $x \in F_{\mathcal{M}od}$. Then from $\mathcal{M}od$ being a 3-valued stable model of $P_{\mathcal{F}}$, it follows that $\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle = \Phi(P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle})$. From Proposition 2 it then follows that each LP argument for x under $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$ has a leaf node labelled with $\texttt{FALSE}$. By definition of $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$, this implies that each LP argument for x under $P_{\mathcal{F}}$ has a leaf node labelled with some "$\texttt{not}\ z$" where $z \in T_{\mathcal{M}od}$. By definition of $P_{\mathcal{F}}$ and $\texttt{Mod2Lab}$, it then follows that each ABA argument for x under $\mathcal{F}$ (which has to be non-trivial, since x is not an assumption) has a leaf node labelled with some assumption ζ that is labelled OUT by $\texttt{Mod2Lab}(\mathcal{M}od)$. From the definition of $\texttt{Lab2Mod}$ it then follows that for each such ζ it holds that $\overline{\zeta} \in T'$, with respect to $\texttt{Lab2Mod}(\texttt{Mod2Lab}(\mathcal{L}ab))$. It then follows that each argument for x under $P_{\mathcal{F}}^{\langle T', F'\rangle}$ has at least one leaf node labelled with $\texttt{FALSE}$. This, by Proposition 2, implies that $x \in F$, with $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T', F'\rangle})$.

(c) $x \in HB_{P_{\mathcal{F}}} \setminus (T_{\mathcal{M}od} \cup F_{\mathcal{M}od})$. Then from $\mathcal{M}od$ being a 3-valued stable model of $P_{\mathcal{F}}$, it follows that $\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle = \Phi(P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle})$. From Proposition 2 (points 1 and 2) it then follows that (1) there is no LP argument for x under $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$ where each leaf node is labelled with $\texttt{TRUE}$, and (2) there is an LP argument for x under $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$ where no leaf node is labelled with $\texttt{FALSE}$.
From (1) it directly follows that each LP argument for x under $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$ has a leaf node that is not labelled with $\texttt{TRUE}$. From the definition of $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od}, F_{\mathcal{M}od}\rangle}$ it then follows that each LP argument for x under $P_{\mathcal{F}}$ has a leaf node that is labelled with "$\texttt{not}\ z$" where $z \notin F_{\mathcal{M}od}$. From the definition of $P_{\mathcal{F}}$ and $\texttt{Mod2Lab}$ it then follows that each ABA argument for x under $\mathcal{F}$ (which has to be non-trivial, since

x is not an assumption) has a leaf node that is labelled with an assumption ζ that is not labelled IN by $\texttt{Mod2Lab}(\mathcal{M}od)$. From the definition of $\texttt{Lab2Mod}$ it then follows that $\overline{\zeta}$ is not in F', with respect to $\texttt{Lab2Mod}(\texttt{Mod2Lab}(\mathcal{M}od))$. It then follows that each LP argument for x under $P_{\mathcal{F}}^{\langle T',F'\rangle}$ has a leaf node that is not labelled with $\texttt{TRUE}$. This, by Proposition 2, implies that $x \notin T$, with $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T',F'\rangle})$.
From (2) and the definition of $P_{\mathcal{F}}^{\langle T_{\mathcal{M}od},F_{\mathcal{M}od}\rangle}$ it follows that there is an LP argument for x under $P_{\mathcal{F}}$ without any leaf being labelled with some "$\texttt{not}\ z$" with $z \in T_{\mathcal{M}od}$. From the definition of $P_{\mathcal{F}}$ and $\texttt{Mod2Lab}$ it then follows that there exists a (non-trivial) ABA argument for x without any leaf node being labelled with some assumption that is labelled OUT by $\texttt{Mod2Lab}(\mathcal{M}od)$. That is, there exists a (non-trivial) ABA argument for x where each assumption ζ is labelled IN or UNDEC by $\texttt{Mod2Lab}(\mathcal{M}od)$. From the definition of $\texttt{Lab2Mod}$ it then follows that each such ζ, $\overline{\zeta} \notin T'$, with respect to $\texttt{Lab2Mod}(\texttt{Mod2Lab}(\mathcal{M}od))$. Hence, there exists an LP argument for x under $P_{\mathcal{F}}^{\langle T',F'\rangle}$ without any leaf node labelled with $\texttt{FALSE}$. This, by Proposition 2, implies that $x \notin F$, with $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T',F'\rangle})$.
So, overall we obtain that $x \in HB_{P_{\mathcal{F}}}$, $x \notin T$ and $x \notin F$, so $x \in HB_{P_{\mathcal{F}}} \setminus (T \cup F)$.

□

Example 2. *Consider again the ABA framework $\mathcal{F}$ and its associated logic program $P_{\mathcal{F}}$ from Example 1. $\mathcal{F}$ has two complete assumption labellings: $\mathcal{L}ab_1 = \{(\alpha, \text{IN}), (\beta, \text{OUT}), (\gamma, \text{IN})\}$ and $\mathcal{L}ab_2 = \{(\alpha, \text{OUT}), (\beta, \text{IN}), (\gamma, \text{OUT})\}$. $P_{\mathcal{F}}$ has two 3-valued stable models: $\mathcal{M}od_1 = \langle\{c, b, d\}, \{a\}\rangle$ and $\mathcal{M}od_2 = \langle\{c, a\}, \{b, d\}$. It is easy to verify that $\mathcal{M}od_1 = \texttt{Lab2Mod}(\mathcal{L}ab_1)$ and $\mathcal{L}ab_1 = \texttt{Mod2Lab}(\mathcal{M}od_1)$, and $\mathcal{M}od_2 = \texttt{Lab2Mod}(\mathcal{L}ab_2)$ and $\mathcal{L}ab_2 = \texttt{Mod2Lab}(\mathcal{M}od_2)$.*

Theorem 1 is important, since in ABA complete semantics is the basis of various other semantics (like grounded, preferred, ideal, and stable), just like in logic programming 3-valued stable models are the basis of various other semantics (like well-founded, regular, ideal, and (2-valued) stable). For instance, where preferred semantics takes the complete assumption labellings and selects those with maximal IN, regular semantics takes the 3-valued stable models and selects those with maximal T. Hence, to prove equivalence between preferred semantics in ABA and regular semantics in logic programming, we need to show that there's an equivalence (through the functions $\texttt{Lab2Mod}$ and $\texttt{Mod2Lab}$) between the complete assumption labellings with maximal IN and the 3-valued stable models with maximal T. For this purpose, we first introduce the following lemma on the correspondence of the set of IN assumptions of a complete assumption labelling and T of a 3-valued stable model.

Lemma 2. *Let $\mathcal{F} = \langle\mathcal{L}, \mathcal{R}, \mathcal{A}, ^-\rangle$ be a normal ABA framework and let $P_{\mathcal{F}}$ be the associated logic program. Let $\mathcal{L}ab_1$ and $\mathcal{L}ab_2$ be complete assumption labellings of $\mathcal{F}$, and let $\mathcal{M}od_1 = \langle T_1, F_1\rangle = \texttt{Lab2Mod}(\mathcal{L}ab_1)$ and $\mathcal{M}od_2 = \langle T_2, F_2\rangle = \texttt{Lab2Mod}(\mathcal{L}ab_2)$. It holds that $\text{IN}(\mathcal{L}ab_1) \subseteq \text{IN}(\mathcal{L}ab_2)$ iff $T_1 \subseteq T_2$.*

Proof. "$\Rightarrow$": Assume that $\text{IN}(\mathcal{L}ab_1) \subseteq \text{IN}(\mathcal{L}ab_2)$. From $\mathcal{M}od_1 = \texttt{Lab2Mod}(\mathcal{L}ab_1)$ it follows that $\langle T_1, F_1\rangle = \Phi(P_{\mathcal{F}}^{\langle T_1',F_1'\rangle})$ with $T_1' = \{\overline{\chi} \mid \chi \in \text{OUT}(\mathcal{L}ab_1)\} \cap HB_{P_{\mathcal{F}}}$ and $F_1' = \{\overline{\chi} \mid \chi \in \text{IN}(\mathcal{L}ab_1)\} \cap HB_{P_{\mathcal{F}}}$. From $\mathcal{M}od_2 = \texttt{Lab2Mod}(\mathcal{L}ab_2)$ it follows that $\langle T_2, F_2\rangle = \Phi(P_{\mathcal{F}}^{\langle T_2',F_2'\rangle})$ with $T_2' = \{\overline{\chi} \mid \chi \in \text{OUT}(\mathcal{L}ab_2)\} \cap HB_{P_{\mathcal{F}}}$ and $F_2' = \{\overline{\chi} \mid \chi \in \text{IN}(\mathcal{L}ab_2)\} \cap HB_{P_{\mathcal{F}}}$. From the fact that $\text{IN}(\mathcal{L}ab_1) \subseteq \text{IN}(\mathcal{L}ab_2)$ it follows (Lemma 1) that $\text{OUT}(\mathcal{L}ab_1) \subseteq \text{OUT}(\mathcal{L}ab_2)$, so we obtain that $T_1' \subseteq T_2'$. Now, suppose that $x \in T_1$. Then (Proposition 2) there exists an LP argument for x

under $P_{\mathcal{F}}^{\langle T_1', F_1' \rangle}$ of which each leaf node is labelled with `TRUE`. From the fact that $T_1' \subseteq T_2'$ it then follows that the same argument also exists under $P_{\mathcal{F}}^{\langle T_2', F_2' \rangle}$. So $x \in T_2$.
"$\Leftarrow$": Since `Lab2Mod` and `Mod2Lab` are each other's inverses (point 3 of Theorem 1) it follows that $\mathcal{L}ab_1 = \texttt{Mod2Lab}(\mathcal{M}od_1)$ and $\mathcal{L}ab_2 = \texttt{Mod2Lab}(\mathcal{M}od_2)$. From the definition of `Mod2Lab` it then follows that $\text{OUT}(\mathcal{L}ab_1) = \{\chi \in \mathcal{A} \mid \overline{\chi} \in T_1\}$ and $\text{OUT}(\mathcal{L}ab_2) = \{\chi \in \mathcal{A} \mid \overline{\chi} \in T_2\}$. From $T_1 \subseteq T_2$ it then follows that $\text{OUT}(\mathcal{L}ab_1) \subseteq \text{OUT}(\mathcal{L}ab_2)$. From Lemma 1 it then follows that $\text{IN}(\mathcal{L}ab_1) \subseteq \text{IN}(\mathcal{L}ab_2)$. □

From the fact that for complete assumption labellings and 3-valued stable models `Lab2Mod` and `Mod2Lab` are each other's inverses, it follows that Lemma 2 can also be applied for two 3-valued stable models $\mathcal{M}od_1$ and $\mathcal{M}od_2$ of $P_{\mathcal{F}}$ and the associated assumption labellings $\mathcal{L}ab_1 = \texttt{Mod2Lab}(\mathcal{M}od_1)$ and $\mathcal{L}ab_2 = \texttt{Mod2Lab}(\mathcal{M}od_2)$ of $\mathcal{F}$. We will sometimes do so in the proof of the following theorem.

Theorem 2. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be a normal ABA framework and let $P_{\mathcal{F}}$ be the associated logic program. It holds that:*

1. *if $\mathcal{L}ab$ is a grounded assumption labelling of $\mathcal{F}$ then* `Lab2Mod`$(\mathcal{L}ab)$ *is a well-founded model of $P_{\mathcal{F}}$*
2. *if $\mathcal{M}od$ is a well-founded model of $P_{\mathcal{F}}$ then* `Mod2Lab`$(\mathcal{M}od)$ *is a grounded assumption labelling of $\mathcal{F}$*
3. *if $\mathcal{L}ab$ is a preferred assumption labelling of $\mathcal{F}$ then* `Lab2Mod`$(\mathcal{L}ab)$ *is a regular model of $P_{\mathcal{F}}$*
4. *if $\mathcal{M}od$ is a regular model of $P_{\mathcal{F}}$ then* `Mod2Lab`$(\mathcal{M}od)$ *is a preferred assumption labelling of $\mathcal{F}$*
5. *if $\mathcal{L}ab$ is a stable assumption labelling of $\mathcal{F}$ then* `Lab2Mod`$(\mathcal{L}ab)$ *is a (2-valued) stable model of $P_{\mathcal{F}}$*
6. *if $\mathcal{M}od$ is a (2-valued) stable model of $P_{\mathcal{F}}$ then* `Mod2Lab`$(\mathcal{M}od)$ *is a stable assumption labelling of $\mathcal{F}$*
7. *if $\mathcal{L}ab$ is an ideal assumption labelling of $\mathcal{F}$ then* `Lab2Mod`$(\mathcal{L}ab)$ *is an ideal model of $P_{\mathcal{F}}$*
8. *if $\mathcal{M}od$ is an ideal model of $P_{\mathcal{F}}$ then* `Mod2Lab`$(\mathcal{M}od)$ *is an ideal assumption labelling of $\mathcal{F}$*

Proof. 1. Let $\mathcal{L}ab$ be a grounded assumption labelling of $\mathcal{F}$, and let $\mathcal{M}od = \texttt{Lab2Mod}(\mathcal{L}ab)$ with $\mathcal{M}od = \langle T, F \rangle$. Then, from Theorem 1 it follows that $\mathcal{M}od$ is a 3-valued stable model of $P_{\mathcal{F}}$. In order to show that $\mathcal{M}od$ is also a well-founded model of $P_{\mathcal{F}}$ we have to additionally prove that T is *minimal* among all 3-valued stable models of $P_{\mathcal{F}}$. Let $\mathcal{M}od^* = \langle T^*, F^* \rangle$ be an arbitrary 3-valued stable model of $P_{\mathcal{F}}$. We have to prove that if $T^* \subseteq T$ then $T^* = T$. Suppose $T^* \subseteq T$. Then, according to Lemma 2, $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$, with $\mathcal{L}ab^* = \texttt{Mod2Lab}(\mathcal{M}od^*)$. From $\mathcal{L}ab$ being a grounded assumption labelling of $\mathcal{F}$, it follows that $\text{IN}(\mathcal{L}ab)$ is minimal among all complete assumption labellings of $\mathcal{F}$. Hence, from $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$ it follows that $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$, so $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$. From Lemma 2 it then follows that $T^* \supseteq T$, which together with $T^* \subseteq T$ implies that $T^* = T$. Hence, $\mathcal{M}od$ is a well-founded model of $P_{\mathcal{F}}$.

2. Let $\mathcal{M}od = \langle T, F\rangle$ be a well-founded model of $P_{\mathcal{F}}$, and let $\mathcal{L}ab = \texttt{Mod2Lab}(\mathcal{M}od)$. From Theorem 1 it then follows that $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$. In order to show that $\mathcal{L}ab$ is also a grounded assumption labelling of $\mathcal{F}$ we have to additionally prove that $\text{IN}(\mathcal{L}ab)$ is *minimal* among all complete assumption labellings of $\mathcal{F}$. Let $\mathcal{L}ab^*$ be an arbitrary complete assumption labelling of $\mathcal{F}$. We have to prove that if $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$ then $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$. Suppose $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$. Then, according to Lemma 2, $T^* \subseteq T$, with $\langle T^*, F^*\rangle = \mathcal{M}od^* = \texttt{Lab2Mod}(\mathcal{L}ab^*)$. From $\mathcal{M}od$ being a well-founded model of $P_{\mathcal{F}}$, it follows that T is minimal among all 3-valued stable models of $P_{\mathcal{F}}$. Hence, from $T^* \subseteq T$ it follows that $T^* = T$, so $T^* \supseteq T$. From Lemma 2 it then follows that $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$, which together with $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$ implies that $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$. Hence, $\mathcal{L}ab$ is a grounded assumption labelling of $\mathcal{F}$.

3. Let $\mathcal{L}ab$ be a preferred assumption labelling of $\mathcal{F}$, and let $\mathcal{M}od = \texttt{Lab2Mod}(\mathcal{L}ab)$ with $\mathcal{M}od = \langle T, F\rangle$. Then, from Theorem 1 it follows that $\mathcal{M}od$ is a 3-valued stable model of $P_{\mathcal{F}}$. In order to show that $\mathcal{M}od$ is also a regular model of $P_{\mathcal{F}}$ we have to additionally prove that T is *maximal* among all 3-valued stable models of $P_{\mathcal{F}}$. Let $\mathcal{M}od^* = \langle T^*, F^*\rangle$ be an arbitrary 3-valued stable model of $P_{\mathcal{F}}$. We have to prove that if $T^* \supseteq T$ then $T^* = T$. Suppose $T^* \supseteq T$. Then, according to Lemma 2, $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$, with $\mathcal{L}ab^* = \texttt{Mod2Lab}(\mathcal{M}od^*)$. From $\mathcal{L}ab$ being a preferred assumption labelling of $\mathcal{F}$, it follows that $\text{IN}(\mathcal{L}ab)$ is maximal among all complete assumption labellings of $\mathcal{F}$. Hence, from $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$ it follows that $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$, so $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$. From Lemma 2 it then follows that $T^* \subseteq T$, which together with $T^* \supseteq T$ implies that $T^* = T$. Hence, $\mathcal{M}od$ is a regular model of $P_{\mathcal{F}}$.

4. Let $\mathcal{M}od = \langle T, F\rangle$ be a regular model of $P_{\mathcal{F}}$, and let $\mathcal{L}ab = \texttt{Mod2Lab}(\mathcal{M}od)$. From Theorem 1 it then follows that $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$. In order to show that $\mathcal{L}ab$ is also a preferred assumption labelling of $\mathcal{F}$ we have to additionally prove that $\text{IN}(\mathcal{L}ab)$ is *maximal* among all complete assumption labellings of $\mathcal{F}$. Let $\mathcal{L}ab^*$ be an arbitrary complete assumption labelling of $\mathcal{F}$. We have to prove that if $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$ then $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$. Suppose $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$. Then, according to Lemma 2, $T^* \supseteq T$, with $\langle T^*, F^*\rangle = \mathcal{M}od^* = \texttt{Lab2Mod}(\mathcal{L}ab^*)$. From $\mathcal{M}od$ being a regular model of $P_{\mathcal{F}}$, it follows that T is maximal among all 3-valued stable models of $P_{\mathcal{F}}$. Hence, from $T^* \supseteq T$ it follows that $T^* = T$, so $T^* \subseteq T$. From Lemma 2 it then follows that $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$, which together with $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$ implies that $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$. Hence, $\mathcal{L}ab$ is a preferred assumption labelling of $\mathcal{F}$.

5. Proof by contraposition. Suppose that $\mathcal{M}od = \langle T, F\rangle = \texttt{Lab2Mod}(\mathcal{L}ab)$ is *not* a (2-valued) stable model of $P_{\mathcal{F}}$. In case $\mathcal{M}od$ is not even a 3-valued stable model of $P_{\mathcal{F}}$, Theorem 1 implies that $\mathcal{L}ab$ is not a complete assumption labelling of $\mathcal{F}$, so $\mathcal{L}ab$ is also not a stable assumption labelling of $\mathcal{F}$. In the remainder of this proof, we will therefore treat the case that $\mathcal{M}od$ is a 3-valued stable model of $P_{\mathcal{F}}$. From the fact that $\mathcal{M}od$ is not a (2-valued) stable model of $P_{\mathcal{F}}$ it then follows that $T \cup F \neq HB_{P_{\mathcal{F}}}$, so there exists a $x \in HB_{P_{\mathcal{F}}} \setminus (T \cup F)$. As $\mathcal{M}od = \texttt{Lab2Mod}(\mathcal{L}ab) = \Phi(P_{\mathcal{F}}^{\langle T', F'\rangle})$ with $T' = \{\overline{\chi} \mid \chi \in \text{OUT}(\mathcal{L}ab)\} \cap HB_{P_{\mathcal{F}}}$ and $F' = \{\overline{\chi} \mid \chi \in \text{IN}(\mathcal{L}ab)\} \cap HB_{P_{\mathcal{F}}}$, it follows that (Proposition 2) that there is no LP argument for x under $P_{\mathcal{F}}^{\langle T', F'\rangle}$ with all its leaf nodes labelled with `TRUE` and there exists an LP argument for x under $P_{\mathcal{F}}^{\langle T', F'\rangle}$ that does not have a leaf node labelled with `FALSE`. From this, it follows that there exists an

LP argument for x under $P_{\mathcal{F}}^{\langle T',F'\rangle}$ that has a leaf node labelled with `UNDEFINED`. This means there is an LP argument for x under $P_{\mathcal{F}}$ that has a leaf node labelled "`not` z" where $z \notin T'$ and $z \notin F'$. Let $\zeta \in \mathcal{A}$ be such that $\overline{\zeta} = z$. It then follows from the definition of `Lab2Mod` that $\zeta \notin \textsc{out}(\mathcal{L}ab)$ and $\zeta \notin \textsc{in}(\mathcal{L}ab)$, so $\zeta \in \textsc{undec}(\mathcal{L}ab)$. This then implies that $\textsc{undec}(\mathcal{L}ab) \neq \emptyset$, so $\mathcal{L}ab$ is not a stable assumption labelling of $\mathcal{F}$.

6. Proof by contraposition. Suppose that $\mathcal{L}ab = \texttt{Mod2Lab}(\mathcal{M}od)$ is *not* a stable assumption labelling of $\mathcal{F}$. In case $\mathcal{L}ab$ is not even a complete assumption labelling of $\mathcal{F}$, Theorem 1 implies that $\mathcal{M}od$ is not a 3-valued stable model of $P_{\mathcal{F}}$, so $\mathcal{M}od$ is also not a (2-valued) stable model of $P_{\mathcal{F}}$. In the remainder of this proof, we will therefore treat the case that $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$. From the fact that $\mathcal{L}ab$ is not a stable assumption labelling of $\mathcal{F}$ it then follows that $\textsc{undec}(\mathcal{L}ab) \neq \emptyset$, so there exists an $\chi \in \textsc{undec}(\mathcal{L}ab)$. That is, $\chi \in \textsc{undec}(\texttt{Mod2Lab}(\mathcal{M}od))$. From the definition of `Mod2Lab` it then follows that $\overline{\chi} \in HB_{P_{\mathcal{F}}} \setminus (T \cup F)$, so $T \cup F \neq HB_{P_{\mathcal{F}}}$, so $\mathcal{M}od$ is not a (2-valued) stable model of $P_{\mathcal{F}}$.

7. Let $\mathcal{L}ab$ be an ideal assumption labelling of $\mathcal{F}$, i.e. $\textsc{in}(\mathcal{L}ab)$ is maximal among all complete assumption labellings of $\mathcal{F}$ with $\textsc{in}(\mathcal{L}ab) \subseteq \bigcap\{\textsc{in}(\mathcal{L}ab_{pref}) \mid \mathcal{L}ab_{pref}$ is a preferred assumption labelling of $\mathcal{F}\}$. By Theorem 1 $\mathcal{M}od = \langle T, F\rangle = \texttt{Lab2Mod}(\mathcal{L}ab)$ is a 3-valued stable model of $P_{\mathcal{F}}$. Since for all preferred assumption labellings $\mathcal{L}ab_{pref}$ of $\mathcal{F}$ it holds that $\textsc{in}(\mathcal{L}ab) \subseteq \textsc{in}(\mathcal{L}ab_{pref})$, it follows by Lemma 2 that $T \subseteq T_{reg}$ for all $\mathcal{M}od_{reg} = \texttt{Lab2Mod}(\mathcal{L}ab_{pref}) = \langle T_{reg}, F_{reg}\rangle$. Furthermore, by Theorem 2 (point 3) all $\mathcal{M}od_{reg}$ are regular models of $P_{\mathcal{F}}$. Thus, $\mathcal{M}od$ is a 3-valued stable model of $P_{\mathcal{F}}$ with $T \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg}\rangle$ is a regular model of $P_{\mathcal{F}}\}$. To show that in addition T is *maximal* among all 3-valued stable models of $P_{\mathcal{F}}$ with $T \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg}\rangle$ is a regular model of $P_{\mathcal{F}}\}$, let $\mathcal{M}od^* = \langle T^*, F^*\rangle$ be a 3-valued stable model of $P_{\mathcal{F}}$ with $T^* \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg}\rangle$ is a regular model of $P_{\mathcal{F}}\}$. We have to prove that if $T^* \supseteq T$ then $T^* = T$. Suppose $T^* \supseteq T$. Since for every regular model $\mathcal{M}od_{reg} = \langle T_{reg}, F_{reg}\rangle$ of $P_{\mathcal{F}}$ it holds that $T^* \subseteq T_{reg}$, it follows from Lemma 2 that $\textsc{in}(\mathcal{L}ab^*) \subseteq \textsc{in}(\mathcal{L}ab_{pref})$ where $\mathcal{L}ab^* = \texttt{Mod2Lab}(\mathcal{M}od^*)$ and $\mathcal{L}ab_{pref} = \texttt{Mod2Lab}(\mathcal{M}od_{reg})$. Furthermore by Theorem 2 (point 4), all $\mathcal{L}ab_{pref}$ are preferred assumption labellings of $\mathcal{F}$. Thus, $\textsc{in}(\mathcal{L}ab^*) \subseteq \bigcap\{\textsc{in}(\mathcal{L}ab_{pref}) \mid \mathcal{L}ab_{pref}$ is a preferred assumption labelling of $\mathcal{F}\}$. But by Lemma 2 also $\textsc{in}(\mathcal{L}ab^*) \supseteq \textsc{in}(\mathcal{L}ab)$. However, since $\textsc{in}(\mathcal{L}ab)$ is also *maximal* among all complete assumption labellings of $\mathcal{F}$ with $\textsc{in}(\mathcal{L}ab) \subseteq \{\textsc{in}(\mathcal{L}ab_{pref}$ is a preferred assumption labelling of $\mathcal{F}\}$, it follows that $\textsc{in}(\mathcal{L}ab^*) = \textsc{in}(\mathcal{L}ab)$, so trivially $\textsc{in}(\mathcal{L}ab^*) \subseteq \textsc{in}(\mathcal{L}ab)$. From Lemma 3.5 it then follows that $T^* \subseteq T$, which together with $T* \supseteq T$ implies that $T^* = T$ Hence, T is maximal among all 3-valued stable labellings of $P_{\mathcal{F}}$ with $T \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg}\rangle$ is a regular model of $P_{\mathcal{F}}\}$, and thus $\mathcal{M}od$ is an ideal model of $P_{\mathcal{F}}$.

8. Let $\mathcal{M}od = \langle T, F\rangle$ be an ideal model of $P_{\mathcal{F}}$, i.e. T is maximal among all 3-valued stable models of $P_{\mathcal{F}}$ with $T \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg}\rangle$ is a regular model of $P_{\mathcal{F}}\}$. By Theorem 1 $\mathcal{L}ab = \texttt{Mod2Lab}(\mathcal{M}od)$ is a complete assumption labelling of $\mathcal{F}$. Since for all regular models $\mathcal{M}od_{reg} = \langle T_{reg}, F_{reg}\rangle$ of $P_{\mathcal{F}}$ it holds that $T \subseteq T_{reg}$, it follows by Lemma 2 that $\textsc{in}(\mathcal{L}ab) \subseteq \textsc{in}(\mathcal{L}ab_{pref})$ with $\mathcal{L}ab_{pref} = \texttt{Mod2Lab}(\mathcal{M}od_{reg})$. Furthermore, by Theorem 2 (point 4) all $\mathcal{L}ab_{pref}$ are preferred assumption labellings of $\mathcal{F}$. Thus, $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$ with $\textsc{in}(\mathcal{L}ab) \subseteq \bigcap\{\textsc{in}(\mathcal{L}ab_{pref}) \mid \mathcal{L}ab_{pref}$ is a preferred assumption labelling of $\mathcal{F}\}$. To show that in addition $\textsc{in}(\mathcal{L}ab)$ is *maximal*

among all complete assumption labellings of $\mathcal{F}$ with $\text{IN}(\mathcal{L}ab) \subseteq \bigcap\{\text{IN}(\mathcal{L}ab_{pref}) \mid \mathcal{L}ab_{pref}$ is a preferred assumption labelling of $\mathcal{F}\}$, let $\mathcal{L}ab^*$ be a complete assumption labelling of $\mathcal{F}$ with $\text{IN}(\mathcal{L}ab^*) \subseteq \bigcap\{\text{IN}(\mathcal{L}ab_{pref}) \mid \mathcal{L}ab_{pref}$ is a preferred assumption labelling of $\mathcal{F}\}$. We have to prove that if $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$ then $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$. Suppose $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$. Since for every preferred assumption labelling $\mathcal{L}ab_{pref}$ of $\mathcal{F}$ it holds that $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab_{pref})$, it follows from Lemma 2 that $T^* \subseteq T_{reg}$ where $\mathcal{M}od^* = \langle T^*, F^* \rangle = \texttt{Lab2Mod}(\mathcal{L}ab^*)$ and $\mathcal{M}od_{reg} = \langle T_{reg}, F_{reg} \rangle = \texttt{Lab2Mod}(\mathcal{L}ab_{pref})$. Furthermore by Theorem 2 (point 3), all $\mathcal{M}od_{reg}$ are regular models of $P_{\mathcal{F}}$. Thus, $T^* \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg} \rangle$ is a regular model of $P_{\mathcal{F}}\}$. But by Lemma 2 also $T^* \supseteq T$. However, since T is also *maximal* among all 3-valued stable models of $P_{\mathcal{F}}$ with $T \subseteq \bigcap\{T_{reg} \mid \langle T_{reg}, F_{reg} \rangle$ is a regular model of $P_{\mathcal{F}}\}$, it follows that $T^* = T$, so trivially $T^* \subseteq T$. From Lemma 3.5 it then follows that $\text{IN}(\mathcal{L}ab^*) \subseteq \text{IN}(\mathcal{L}ab)$, which together with $\text{IN}(\mathcal{L}ab^*) \supseteq \text{IN}(\mathcal{L}ab)$ implies that $\text{IN}(\mathcal{L}ab^*) = \text{IN}(\mathcal{L}ab)$. Hence, $\text{IN}(\mathcal{L}ab)$ is maximal among all complete assumption labellings of $\mathcal{F}$ with $\text{IN}(\mathcal{L}ab) \subseteq \bigcap\{\text{IN}(\mathcal{L}ab_{pref}) \mid \mathcal{L}ab_{pref}$ is a preferred assumption labelling of $\mathcal{F}\}$, and thus $\mathcal{L}ab$ is an ideal assumption labelling of $\mathcal{F}$.

□

4 Translating Logic Programs to ABA Theories

In the previous section, we have studied a translation from normal ABA to LP, and have observed that the various types of labellings of an ABA framework coincide with the various types of models of the associated logic program. In this section, we go the other way around. That is, we examine a translation from LP to ABA.

Definition 12. *Let P be a normal logic program. We define the* associated ABA framework *$\mathcal{F}_P = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ with $\mathcal{A} = \{\texttt{not\_w} \mid w \in HB_P\}$, $\mathcal{L} = HB_P \cup \mathcal{A}$, $\mathcal{R} = \{x \leftarrow y_1, \ldots, y_n, \texttt{not\_z}_1, \ldots,$
$\texttt{not\_z}_m \mid x \leftarrow y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m \in P\}$ and $\overline{\texttt{not\_w}} = w$ for every $\texttt{not\_w} \in \mathcal{A}$.*

We define `LP2ABA` to be the function that, given a logic program P, yields the associated ABA framework $\mathcal{F}_P$ (Definition 12). Similarly, we define `ABA2LP` to be the function that, given a normal ABA framework $\mathcal{F}$, yields the associated logic program $P_{\mathcal{F}}$ (Definition 10).

Theorem 3. *Let P be a normal logic program. It holds that $\texttt{ABA2LP}(\texttt{LP2ABA}(P)) = P$.*

Proof. Let $\mathcal{F}_P = \langle \mathcal{L}_P, \mathcal{R}_P, \mathcal{A}_P, ^- \rangle$ be $\texttt{LP2ABA}(P)$.
"$\subseteq$": Let $x \leftarrow y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m$ be a logic programming rule in $\texttt{ABA2LP}(\texttt{LP2ABA}(P))$. Then from the definition of `ABA2LP` it follows that there exists an ABA rule $x \leftarrow y_1, \ldots, y_n, \zeta_1, \ldots, \zeta_m$ in $\mathcal{R}_P$ with $\zeta_i \in \mathcal{A}_P$ and $\overline{\zeta_i} = z_i$ $(1 \leq i \leq m)$. From the definition of `LP2ABA` it then follows that $\zeta_i = \texttt{not\_z}_i$ (because $\texttt{not\_z}_i$ is the only assumption in $\mathcal{A}_P$ that has z_i as its contrary $(1 \leq i \leq m)$) and that there exists a rule $x \leftarrow y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m$ in P.
"$\supseteq$": Let $x \leftarrow y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m$ be a logic programming rule in P. Then from the definition of `LP2ABA` it follows that $\mathcal{R}_P$ contains a rule $x \leftarrow y_1, \ldots, y_n, \texttt{not\_z}_1, \ldots, \texttt{not\_z}_m$ with $\overline{\texttt{not\_z}_i} = z_i$ $(1 \leq i \leq m)$. From the definition of `ABA2LP` this then implies that $\texttt{ABA2LP}(\texttt{LP2ABA}(P))$ contains a rule $x \leftarrow y_1, \ldots, y_n, \texttt{not}\ z_1, \ldots, \texttt{not}\ z_m$. □

Theorem 4. *Let P be a logic program and let $\mathcal{F}_P = \texttt{LP2ABA}(P)$ be its associated ABA framework. It holds that:*

1. *if $\mathcal{M}od$ is a 3-valued stable model of P, then* Mod2Lab($\mathcal{M}od$) *is a complete assumption labelling of $\mathcal{F}_P$*

2. *if $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}_P$, then* Lab2Mod($\mathcal{L}ab$) *is a 3-valued stable model of P*

Proof. [3] Let $P_{\mathcal{F}_P}$ be the associated logic program of $\mathcal{F}_P$, i.e. $P_{\mathcal{F}_P} = \texttt{ABA2LP}(\mathcal{F}_P)$. From $\mathcal{F}_P = \texttt{LP2ABA}(P)$ it then follows that $P_{\mathcal{F}_P} = \texttt{ABA2LP}(\texttt{LP2ABA}(P))$. It then follows from Theorem 3 that $P_{\mathcal{F}_P} = P$.

1. Let $\mathcal{M}od$ be a 3-valued stable model of P. As $P = P_{\mathcal{F}_P}$, it directly follows that $\mathcal{M}od$ is a 3-valued stable model of $P_{\mathcal{F}_P}$. From Theorem 1 (point 2) it then follows that Mod2Lab($\mathcal{M}od$) is a complete assumption labelling of $\mathcal{F}_P$.

2. Let $\mathcal{L}ab$ be a complete assumption labelling of $\mathcal{F}_P$. From Theorem 1 (point 1) it then follows that Lab2Mod($\mathcal{L}ab$) is a 3-valued stable model of $P_{\mathcal{F}_P}$. From the fact that $P_{\mathcal{F}_P} = P$, it then directly follows that Lab2Mod($\mathcal{L}ab$) is also a 3-valued stable model of P.

□

We now extend the correspondence results from Theorem 4 to common ABA and LP semantics.

Theorem 5. *Let P be a logic program and let $\mathcal{F}_P$ = LP2ABA(P) be its associated ABA framework. It holds that:*

1. *if $\mathcal{M}od$ is a well-founded model of P, then* Mod2Lab($\mathcal{M}od$) *is a grounded assumption labelling of $\mathcal{F}_P$*

2. *if $\mathcal{L}ab$ is a grounded assumption labelling of $\mathcal{F}_P$, then* Lab2Mod($\mathcal{L}ab$) *is a well-founded model of P*

3. *if $\mathcal{M}od$ is a regular model of P, then* Mod2Lab($\mathcal{M}od$) *is a preferred assumption labelling of $\mathcal{F}_P$*

4. *if $\mathcal{L}ab$ is a preferred assumption labelling of $\mathcal{F}_P$, then* Lab2Mod($\mathcal{L}ab$) *is a regular model of P*

5. *if $\mathcal{M}od$ is a (2-valued) stable model of P, then* Mod2Lab($\mathcal{M}od$) *is a stable assumption labelling of $\mathcal{F}_P$*

6. *if $\mathcal{L}ab$ is a stable assumption labelling of $\mathcal{F}_P$, then* Lab2Mod($\mathcal{L}ab$) *is a (2-valued) stable model of P*

7. *if $\mathcal{M}od$ is an ideal model of P, then* Mod2Lab($\mathcal{M}od$) *is an ideal assumption labelling of $\mathcal{F}_P$*

8. *if $\mathcal{L}ab$ is an ideal assumption labelling of $\mathcal{F}_P$, then* Lab2Mod($\mathcal{L}ab$) *is an ideal model of P*

[3] This was proven in [24]. Here, we use a different proof which will serve as an illustration for the proofs of Theorem 5.

Proof. Similar to the proof of Theorem 4, but applying Theorem 2 instead of Theorem 1. □

Theorem 4 and Theorem 5 point out that our results regarding the translation from ABA to LP, as stated in the previous section, can be reused for the translation from LP to ABA. Hence, our work generalizes on the work of [24, 2], where only the LP to ABA direction is considered.

If it is possible to reuse the results from the ABA to LP translation for the LP to ABA translation, then is the reverse also possible? That is, we ask ourselves whether it is possible to reuse some of the existing work on the LP to ABA translation (like for instance stated in [24, 2]) to obtain similar results for the ABA to LP translation (like for instance stated in the previous section). The short answer is no, at least not in any obvious way. Our ability to reuse the results from the ABA to LP translation for the LP to ABA translation (and therefore to write the sort of proofs of Theorem 4 and Theorem 5) critically depends on the fact that $\texttt{ABA2LP}(\texttt{LP2ABA}(P)) = P$ (Theorem 3). To be able to reuse the results of the LP to ABA direction for the ABA to LP direction in a similar way as that is done in the current section would require the property that for any ABA framework $\mathcal{F}$, $\texttt{LP2ABA}(\texttt{ABA2LP}(\mathcal{F})) = \mathcal{F}$. However, this property does *not* hold, for the simple reason that when translating from ABA to LP some information gets lost (like the precise set of assumptions, some of which may not occur in any rule) as in essence only its set of rules $\mathcal{R}$ gets translated.

5 Going Beyond Normal ABA Frameworks and Common Semantics

It has to be mentioned that the fragment of ABA that we have studied so far consists of normal ABA frameworks (that is, ABA frameworks that are flat and where every assumption has a unique non-assumption contrary) under common ABA semantics (that is, under complete, grounded, preferred, stable, or ideal semantics). In this section, we briefly examine what happens when we try to go beyond this, that is, if we start to examine either ABA frameworks that are not normal or ABA semantics that are not common.

One of the ways to go beyond normal ABA frameworks is to allow for assumptions that have more than one contrary. This type of ABA frameworks could still be translated into logic programs, basically by replacing each assumption by the NAF literals for all of its contraries. For instance, if the ABA framework contains a rule $c \leftarrow a, \beta, \delta$, where β is an assumption with contraries b_1 and b_2, and δ is an assumption with contraries d_1, d_2 and d_3, then the associated logic programming rule would be $c \leftarrow a, \texttt{not}\ b_1, \texttt{not}\ b_2, \texttt{not}\ d_1, \texttt{not}\ d_2, \texttt{not}\ d_3$. Also, the functions $\texttt{Lab2Mod}$ and $\texttt{Mod2Lab}$ would have to be modified accordingly, with $\texttt{Lab2Mod}$ yielding $\langle T, F\rangle = \Phi(P_{\mathcal{F}}^{\langle T', F'\rangle})$ where $T' = \{x \mid x \in \overline{\chi} \in \mathcal{A}$ and there exists an ABA argument $Asms \vdash x$ with $Asms \subseteq \text{IN}(\mathcal{L}ab)\} \cap HB_{P_{\mathcal{F}}}$ and $F' = \{x \mid x \in \overline{\chi} \in \mathcal{A}$ and for each ABA argument $Asms \vdash x$ it holds that $Asms \cap \text{OUT}(\mathcal{L}ab) \neq \emptyset\} \cap HB_{P_{\mathcal{F}}}$, and $\texttt{Mod2Lab}$ yielding a labelling $\mathcal{L}ab$ with $\text{IN}(\mathcal{L}ab) = \{\chi \in \mathcal{A} \mid \overline{\chi} \cap HB_{P_{\mathcal{F}}} \subseteq F\}$, $\text{OUT}(\mathcal{L}ab) = \{\chi \in \mathcal{A} \mid \overline{\chi} \cap T \neq \emptyset\}$ and $\text{UNDEC}(\mathcal{L}ab) = \{\chi \in \mathcal{A} \mid \overline{\chi} \cap HB_{P_{\mathcal{F}}} \not\subseteq F$ and $\overline{\chi} \cap T = \emptyset\}$. We conjecture that under these translations, the main results of Section 3 (Theorem 1, Lemma 2 and Theorem 2) still hold.

Another way to go beyond normal ABA frameworks is to study non-flat ABA frameworks, that is, ABA frameworks where an assumption can occur in the head of a rule. Translating these into logic programs would yield logic programming rules where the head can be a NAF literal, which would go beyond the syntax and semantics of normal logic programs. Hence, as it currently stands, our theory is not able to capture the case of non-flat ABA frameworks.

Apart from going beyond normal ABA frameworks, one could also examine what happens if one were to go beyond common ABA semantics (complete, grounded, preferred, stable and ideal). Take for instance the concept of semi-stable semantics [28, 4], which has recently been formulated in the context of ABA [5, 6, 24].

Definition 13 ([24])**.** *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be an ABA framework. We say that an assumption labelling $\mathcal{L}ab$ is a* semi-stable assumption labelling *of $\mathcal{F}$ iff $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$ where* UNDEC$(\mathcal{L}ab)$ *is minimal among all complete assumption labellings of $\mathcal{F}$.*

It has been shown that there exists a one-to-one relationship between semi-stable assumption labellings (Definition 13) and semi-stable assumption extensions in the sense of [5, 6], with the set of IN labelled assumptions in a semi-stable assumption labelling constituting a semi-stable assumption extension. It has also been observed that semi-stable semantics for ABA behaves in a way that is very similar to semi-stable semantics for abstract argumentation [6]. For instance, each stable assumption labelling (extension) is also a semi-stable assumption labelling (extension), and each semi-stable assumption labelling (extension) is also a preferred assumption labelling (extension). Moreover, for ABA frameworks that have at least one stable assumption labelling (extension), it holds that each semi-stable assumption labelling (extension) is also a stable assumption labelling (extension).

In the context of logic programming, a concept somewhat similar to semi-stable semantics exists under the name of L-stable semantics.

Definition 14. *Let P be a logic program. We say that a 3-valued interpretation $\langle T, F \rangle$ of P is an* L-stable model *of P iff it is a 3-valued stable model of P where $T \cup F$ is maximal among all 3-valued stable models of P.*

So where a semi-stable assumption labelling tries to minimize the set of assumptions that are not labelled IN or OUT, an L-stable model tries to minimize the LP-atoms that are not in T or F. Although one might perhaps expect that the semi-stable assumption labellings of an ABA framework $\mathcal{F}$ coincide with the L-stable models of the associated logic program $P_{\mathcal{F}}$ (through the functions `Lab2Mod` and `Mod2Lab` as defined in Section 3) this turns out not to be the case, as is illustrated in the following example.

Example 3. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be an ABA framework with $\mathcal{A} = \{\alpha, \beta, \gamma\}$, $\mathcal{L} = \mathcal{A} \cup \{a, b, c, d\}$, $\overline{\alpha} = a$, $\overline{\beta} = b$ and $\overline{\gamma} = c$. and $\mathcal{R} =$*
$a \leftarrow \beta$
$b \leftarrow \alpha$
$c \leftarrow \gamma$
$d \leftarrow b, c$

$\mathcal{F}$ has three complete assumption labellings:
$(\emptyset, \emptyset, \{\alpha, \beta, \gamma\})$
(α, β, γ)
(β, α, γ)
The last two of these are semi-stable assumption labellings.
Translating $\mathcal{F}$ yields the following logic program $P_{\mathcal{F}}$:
$a \leftarrow \texttt{not}\ b$

$b \leftarrow \texttt{not}\ a$
$c \leftarrow \texttt{not}\ c$
$d \leftarrow b, c$
$P_{\mathcal{F}}$ has three 3-valued stable models:
$\langle \emptyset, \emptyset, \{a, b, c, d\} \rangle$
$\langle \{b\}, \{a\} \rangle$
$\langle \{a\}, \{b, d\} \rangle$
Only the last one is L-stable.
Hence, we have that the semi-stable assumption labellings of $\mathcal{F}$ do not coincide with the L-stable models of $P_{\mathcal{F}}$.

Although Example 3 illustrates that in general, ABA under semi-stable semantics does not coincide with logic programming under L-stable semantics, equivalence is restored when the ABA framework is *assumption-spanning.*

Definition 15. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an ABA framework. We say that $\mathcal{F}$ is* assumption-spanning *iff for each $x \in \mathcal{L} \setminus \mathcal{A}$ there exists a $\chi \in \mathcal{A}$ such that $\overline{\chi} = x$.*

We first observe that the ABA framework of Example 3 is not assumption-spanning, because there is no assumption that has d as its contrary. In the following example, we examine what happens if we add such an assumption (δ) and thus make the ABA framework assumption-spanning.

Example 4. *Let $\mathcal{F}' = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^- \rangle$ be an assumption-spanning ABA framework with $\mathcal{A} = \{\alpha, \beta, \gamma, \delta\}$, $\mathcal{L} = \mathcal{A} \cup \{a, b, c, d\}$, $\overline{\alpha} = a$, $\overline{\beta} = b$, $\overline{\gamma} = c$, $\overline{\delta} = d$ and $\mathcal{R} =$*
$a \leftarrow \beta$
$b \leftarrow \alpha$
$c \leftarrow \gamma$
$d \leftarrow b, c$

$\mathcal{F}'$ has three complete assumption labellings:
$(\emptyset, \emptyset, \{\alpha, \beta, \gamma\})$
$(\{\alpha\}, \{\beta\}, \{\gamma, \delta\})$
$(\{\beta, \delta\}, \{\alpha\}, \{\gamma\})$
Only the last one is semi-stable.
Translating $\mathcal{F}'$ yields the following logic program $P_{\mathcal{F}'}$:
$a \leftarrow \texttt{not}\ b$
$b \leftarrow \texttt{not}\ a$
$c \leftarrow \texttt{not}\ c$
$d \leftarrow b, c$
$P_{\mathcal{F}}$ has three 3-valued stable models:
$\langle \emptyset, \emptyset, \{a, b, c, d\} \rangle$
$\langle \{b\}, \{a\} \rangle$
$\langle \{a\}, \{b, d\} \rangle$
Again, only the last one is L-stable.
Hence, we observe that when we make the ABA framework $\mathcal{F}$ (from Example 3) assumption-spanning (by adding an assumption δ with $\overline{\delta} = d$) the resulting ABA framework $\mathcal{F}'$ has its semi-stable labellings coinciding with the L-stable models of the associated logic program $P_{\mathcal{F}'}$.

The fact that for an assumption-spanning ABA framework $\mathcal{F}$, the semi-stable assumption labellings of $\mathcal{F}$ coincide with the L-stable models of $P_{\mathcal{F}}$ is not restricted to the particular ABA framework of Example 4. In fact, it is a general property. To prove so, we need the following lemma.

Lemma 3. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be an assumption-spanning ABA framework, and let $P_{\mathcal{F}}$ be its associated logic program. Let $\mathcal{L}ab_1$ and $\mathcal{L}ab_2$ be complete assumption labellings of $\mathcal{F}$, let $\mathcal{M}od_1 = \langle T_1, F_1 \rangle$ be* $\texttt{Lab2Mod}(\mathcal{L}ab_1)$ *and let $\mathcal{M}od_2 = \langle T_2, F_2 \rangle$ be* $\texttt{Lab2Mod}(\mathcal{L}ab_2)$*. It holds that* $\textsc{undec}(\mathcal{L}ab_1) \subseteq \textsc{undec}(\mathcal{L}ab_2)$ *iff* $HB_{P_{\mathcal{F}}} \setminus (T_1 \cup F_1) \subseteq HB_{P_{\mathcal{F}}} \setminus (T_2 \cup F_2)$.

Proof. "$\Rightarrow$": Suppose $\textsc{undec}(\mathcal{L}ab_1) \subseteq \textsc{undec}(\mathcal{L}ab_2)$. Let $x \in HB_{P_{\mathcal{F}}} \setminus (T_1 \cup F_1)$. From the fact that $\mathcal{F}$ is assumption-spanning, it follows that there exists an assumption χ with $\overline{\chi} = x$. As $\mathcal{L}ab_1 = \texttt{Mod2Lab}(\mathcal{M}od_1)$ (Theorem 1, point 3) it follows from the definition of $\texttt{Mod2Lab}$ that $\chi \in \textsc{undec}(\mathcal{L}ab_1)$. From the fact that $\textsc{undec}(\mathcal{L}ab_1) \subseteq \textsc{undec}(\mathcal{L}ab_2)$ it then follows that $\chi \in \textsc{undec}(\mathcal{L}ab_2)$. As $\mathcal{L}ab_2 = \texttt{Mod2Lab}(\mathcal{M}od_2)$ (Theorem 1, point 3) it follows that $\textsc{undec}(\mathcal{L}ab_2) = \{\zeta \in \mathcal{A} \mid \overline{\zeta} \in HB_{P_{\mathcal{F}}} \setminus (T_2 \cup F_2)\}$, so from $\chi \in \textsc{undec}(\mathcal{L}ab_2)$ it follows that $x \in HB_{P_{\mathcal{F}}} \setminus (T_2 \cup F_2)$.
"$\Leftarrow$": Suppose that $HB_{P_{\mathcal{F}}} \setminus (T_1 \cup F_1) \subseteq HB_{P_{\mathcal{F}}} \setminus (T_2 \cup F_2)$. Let $\chi \in \textsc{undec}(\mathcal{L}ab_1)$. As $\mathcal{L}ab_1 = \texttt{Mod2Lab}(\mathcal{M}od_1)$ (Theorem 1, point 3) it follows that $\textsc{undec}(\mathcal{L}ab_1) = \{\zeta \in \mathcal{A} \mid \overline{\zeta} \in HB_{P_{\mathcal{F}}} \setminus (T_1 \cup F_1)\}$ so from $\chi \in \textsc{undec}(\mathcal{L}ab_1)$ it follows that $\overline{\chi} \in HB_{P_{\mathcal{F}}} \setminus (T_1 \cup T_2)$. From $HB_{P_{\mathcal{F}}} \setminus (T_1 \cup F_1) \subseteq HB_{P_{\mathcal{F}}} \setminus (T_2 \cup F_2)$ it then follows that $\overline{\chi} \in HB_{P_{\mathcal{F}}} \setminus (T_2 \cup F_2)$. As $\mathcal{L}ab_2 = \texttt{Mod2Lab}(\mathcal{M}od_2)$ (Theorem 1, point 3) it follows that $\chi \in \textsc{undec}(\mathcal{L}ab_2)$. □

Theorem 6. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be an assumption-spanning ABA framework, and let $P_{\mathcal{F}}$ be its associated logic program. It holds that:*

1. *if $\mathcal{L}ab$ is a semi-stable assumption labelling of $\mathcal{F}$ then* $\texttt{Lab2Mod}(\mathcal{L}ab)$ *is an L-stable model of $P_{\mathcal{F}}$*

2. *if $\mathcal{M}od$ is an L-stable model of $P_{\mathcal{F}}$ then* $\texttt{Mod2Lab}(\mathcal{M}od)$ *is a semi-stable assumption labelling of $\mathcal{F}$*

Proof.

1. Let $\mathcal{L}ab$ be a semi-stable assumption labelling of $\mathcal{F}$, and let $\mathcal{M}od = \texttt{Lab2Mod}(\mathcal{L}ab)$ with $\mathcal{M}od = \langle T, F \rangle$. Then, from Theorem 1 it follows that $\mathcal{M}od$ is a 3-valued stable labelling of $P_{\mathcal{F}}$. In order to show that $\mathcal{M}od$ is also an L-stable labelling of $P_{\mathcal{F}}$ we have to additionally prove that $HB_{P_{\mathcal{F}}} \setminus (T \cup F)$ is *minimal* among all 3-valued stable models of $P_{\mathcal{F}}$. Let $\mathcal{M}od^* = \langle T^*, F^* \rangle$ be an arbitrary 3-valued stable model of $P_{\mathcal{F}}$. We have to prove that if $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) \subseteq HB_{P_{\mathcal{F}}} \setminus (T \cup F)$ then $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) = HB_{P_{\mathcal{F}}} \setminus (T \cup F)$. Suppose $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) \subseteq HB_{P_{\mathcal{F}}} \setminus (T \cup F)$. Then, according to Lemma 3, $\textsc{undec}(\mathcal{L}ab^*) \subseteq \textsc{undec}(\mathcal{L}ab)$, with $\mathcal{L}ab^* = \texttt{Mod2Lab}(\mathcal{M}od^*)$. From $\mathcal{L}ab$ being a semi-stable assumption labelling of $\mathcal{F}$, it follows that $\textsc{undec}(\mathcal{L}ab)$ is minimal among all complete assumption labellings of $\mathcal{F}$. Hence, from $\textsc{undec}(\mathcal{L}ab^*) \subseteq \textsc{undec}(\mathcal{L}ab)$ it follows that $\textsc{undec}(\mathcal{L}ab^*) = \textsc{undec}(\mathcal{L}ab)$, so $\textsc{undec}(\mathcal{L}ab^*) \supseteq \textsc{undec}(\mathcal{L}ab)$. From Lemma 3 it then follows that $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) \supseteq HB_{P_{\mathcal{F}}} \setminus (T \cup F)$, which together with $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) \subseteq HB_{P_{\mathcal{F}}} \setminus (T \cup F)$ implies that $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) = HB_{P_{\mathcal{F}}} \setminus (T \cup F)$. Hence, $\mathcal{M}od$ is an L-stable model of $P_{\mathcal{F}}$.

2. Let $\mathcal{M}od = \langle T, F \rangle$ be an L-stable model of $P_{\mathcal{F}}$, and let $\mathcal{L}ab = \texttt{Mod2Lab}(\mathcal{M}od)$. From Theorem 1 it then follows that $\mathcal{L}ab$ is a complete assumption labelling of $\mathcal{F}$. In order

to show that $\mathcal{M}od$ is also a semi-stable assumption labelling of $\mathcal{F}$ we have to additionally prove that $\textsc{undec}(\mathcal{L}ab)$ is *minimal* among all complete assumption labellings of $\mathcal{F}$. Let $\mathcal{L}ab^*$ be an arbitrary complete assumption labelling of $\mathcal{F}$. We have to prove that if $\textsc{undec}(\mathcal{L}ab^*) \subseteq \textsc{undec}(\mathcal{L}ab)$ then $\textsc{undec}(\mathcal{L}ab^*) = \textsc{undec}(\mathcal{L}ab)$. Suppose $\textsc{undec}(\mathcal{L}ab^*) \subseteq \textsc{undec}(\mathcal{L}ab)$. Then, according to Lemma 3, $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) \subseteq HB_{P_{\mathcal{F}}} \setminus (T \cup F)$, with $\langle T^*, F^* \rangle = \mathcal{M}od^* = \mathtt{Lab2Mod}(\mathcal{L}ab^*)$. From $\mathcal{M}od$ being an L-stable model of $P_{\mathcal{F}}$, it follows that $HB_{P_{\mathcal{F}}} \setminus (T \cup F)$ is minimal among all 3-valued stable models of $P_{\mathcal{F}}$. Hence, from $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) \subseteq HB_{P_{\mathcal{F}}} \setminus (T \cup F)$ it follows that $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) = HB_{P_{\mathcal{F}}} \setminus (T \cup F)$, so $HB_{P_{\mathcal{F}}} \setminus (T^* \cup F^*) \supseteq HB_{P_{\mathcal{F}}} \setminus (T \cup F)$. From Lemma 3 it then follows that $\textsc{undec}(\mathcal{L}ab^*) \supseteq \textsc{undec}(\mathcal{L}ab)$, which together with $\textsc{undec}(\mathcal{L}ab^*) \subseteq \textsc{undec}(\mathcal{L}ab)$ implies that $\textsc{undec}(\mathcal{L}ab^*) = \textsc{undec}(\mathcal{L}ab)$. Hence, $\mathcal{L}ab$ is a semi-stable assumption labelling of $\mathcal{F}$.

□

From Theorem 6, it follows that when translating a logic program to an ABA framework (`LP2ABA`, Section 4) the L-stable models of the original logic program P coincide with the semi-stable models of the associated ABA framework $\mathcal{F}_P$. This is because the translation process `LP2ABA` makes sure that $\mathcal{F}_P$ is assumption-spanning.[4] Hence, the results of [24, Theorems 4 and 5] are subsumed by our current theory.

6 Discussion

In the current paper we examined the relation between ABA and LP, and found that a frequently studied fragment of ABA is subsumed by normal logic programming, and vice versa. That is, the kind of outcome that is yielded by a normal ABA framework under common ABA semantics is essentially the same as the outcome yielded by its associated normal logic program under common LP semantics. The only real difference is that whereas in ABA the outcome is defined in terms of assumptions (which correspond to the NAF-literals in the associated logic program) in logic programming the outcome is defined in terms of *all* the literals in the logic program (NAF as well as non-NAF). However, since the status of the non-NAF literals is determined solely by the status of the NAF-literals (basically, by applying the Gelfond-Lifschitz reduct, as is done by `Lab2Mod`) both approaches are equivalent.

The results of our paper enable researchers to switch freely between ABA syntax and semantics, and LP syntax and semantics. For instance, when applying normal ABA for reasoning about a particular domain, one could equally apply normal LP for the same reasoning. The advantage of doing so is that, as more people are familiar with LP than with ABA, the results can be disseminated to a wider audience.

Similarly, the equivalence between ABA and LP allows some of the techniques developed in the context of ABA to be carried over to the context of LP. For instance, the argument-based proof procedures of [11, 26], when used in a normal ABA framework, can be carried over to logic programming in a straightforward way, using the notion of LP arguments, as introduced in Definition 9.

It should be emphasized that our results regarding the equivalence between ABA and LP are restricted to normal ABA and normal LP. For instance, logic programming would struggle

[4]The precise proof uses a similar structure as the proof of Theorem 4, but applying Theorem 6 instead of Theorem 1.

to model non-flat ABA frameworks, as this would require NAF-literals to occur in the head of LP rules. Similarly, ABA would struggle to model disjunctive logic programming (where the head of a rule can be a disjunction) as it is not clear how arguments can be constructed in this context. Another issue is how the ABA-LP equivalence is affected when applying non-common ABA and LP semantics. As was observed in Section 5, ABA semi-stable in general does not coincide with LP L-stable,[5] although equivalence does hold for assumption-spanning ABA theories. Overall, we hold the issue of how the results in the current paper can be generalized beyond normal ABA frameworks, normal logic programs and common semantics to be an interesting topic for further research.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC, UK), grant ref. EP/J012084/1 (SAsSy project).

References

[1] Alferes, J. J., Dung, P. M., Pereira, L. M.: Scenario Semantics of Extended Logic Programs, *Proceedings of the 2nd International Workshop on Logic Programming and Non-monotonic Reasoning (LPNMR)* (A. Nerode, L. Pereira, Eds.), MIT Press, 1993.

[2] Bondarenko, A., Dung, P. M., Kowalski, R. A., Toni, F.: An Abstract, Argumentation-Theoretic Approach to Default Reasoning, *Artificial Intelligence*, **93**, 1997, 63–101.

[3] Brogi, A., Lamma, E., Mancarella, P., Mello, P.: Normal Logic Programs as Open Positive Programs, *Proceedings of the Joint International Conference and Symposium on Logic Programming (JICSLP) 1992* (K. R. Apt, Ed.), MIT Press, 1992.

[4] Caminada, M. W. A.: Semi-Stable Semantics, *Proceedings of Computational Models of Argument (COMMA) 2006* (P. E. Dunne, T. M. Bench-Capon, Eds.), IOS Press, 2006.

[5] Caminada, M. W. A., Sá, S., Alcântara, J., Dvořák, W.: On the Difference between Assumption-Based Argumentation and Abstract Argumentation, *Proceedings of the 25th Benelux Conference on Artificial Intelligence* (K. Hindriks, M. de Weerdt, B. van Riemsdijk, M. Warnier, Eds.), 2013.

[6] Caminada, M. W. A., Sá, S., Alcântara, J., Dvořák, W.: On the Difference between Assumption-Based Argumentation and Abstract Argumentation, *IFCoLog Journal of Logics and their Applications*, 2015, In print.

[7] Caminada, M. W. A., Sá, S., Alcântara, J., Dvořák, W.: On the Equivalence between Logic Programming Semantics and Argumentation Semantics, *International Journal of Approximate Reasoning*, **58**, 2015, 87–111.

[8] Craven, R., Toni, F., Cadar, C., Hadad, A., Williams, M.: Efficient Argumentation for Medical Decision-Making, *Proceedings of the 13th International Conference on Principles*

[5] Just like ABA semi-stable does not coincide with AA semi-stable [6] and AA semi-stable does not coincide with LP L-stable [7].

of Knowledge Representation and Reasoning (KR) (G. Brewka, T. Eiter, S. A. McIlraith, Eds.), AAAI Press, 2012.

[9] Dung, P. M.: Negations as Hypotheses: An Abductive Foundation for Logic Programming, *Proceedings of the 8th International Conference on Logic Programming (ICLP)* (K. Furukawa, Ed.), MIT Press, 1991.

[10] Dung, P. M., Kowalski, R. A., Toni, F.: Assumption-Based Argumentation, in: *Argumentation in Artificial Intelligence* (G. Simari, I. Rahwan, Eds.), Springer US, 2009, 199–218.

[11] Dung, P. M., Mancarella, P., Toni, F.: Computing Ideal Sceptical Argumentation, *Artificial Intelligence*, **171**(10-15), 2007, 642–674.

[12] Dung, P. M., Thang, P. M.: Modular Argumentation for Modelling Legal Doctrines in Common Law of Contract, *Artificial Intelligence and Law*, **17**(3), 2009, 167–182.

[13] Fan, X., Toni, F.: A General Framework for Sound Assumption-Based Argumentation Dialogues, *Artificial Intelligence*, **216**, 2014, 20–54.

[14] Fan, X., Toni, F.: On Computing Explanations in Argumentation, *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (B. Bonet, S. Koenig, Eds.), AAAI Press, 2015.

[15] Fan, X., Toni, F., Mocanu, A., Williams, M.: Dialogical Two-Agent Decision Making with Assumption-Based Argumentation, *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) 2014* (A. L. C. Bazzan, M. N. Huhns, A. Lomuscio, P. Scerri, Eds.), IFAAMAS/ACM, 2014.

[16] Gaertner, D., Toni, F.: Computing Arguments and Attacks in Assumption-Based Argumentation, *IEEE Intelligent Systems*, **22**(6), 2007, 24–33.

[17] Gaertner, D., Toni, F.: Hybrid Argumentation and Its Properties, *Proceedings of Computational Models of Argument (COMMA) 2008* (P. Besnard, S. Doutre, A. Hunter, Eds.), IOS Press, 2008.

[18] Morge, M., Mancarella, P.: Assumption-Based Argumentation for the Minimal Concession Strategy, in: *Argumentation in Multi-Agent Systems* (P. McBurney, I. Rahwan, S. Parsons, N. Maudet, Eds.), Springer Berlin Heidelberg, 2010, 114–133.

[19] Przymusinski, T. C.: The Well-Founded Semantics Coincides with the Three-Valued Stable Semantics, *Fundamenta Informaticae*, **13**(4), 1990, 445–463.

[20] Przymusinski, T. C.: Semantics of Disjunctive Logic Programs and Deductive Databases, *Proceedings of the 2nd International Conference on Deductive and Object-Oriented Databases (DOOD)* (C. Delobel, M. Kifer, Y. Masunaga, Eds.), Springer Berlin Heidelberg, 1991.

[21] Schulz, C.: Graphical Representation of Assumption-Based Argumentation, *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (B. Bonet, S. Koenig, Eds.), AAAI Press, 2015.

[22] Schulz, C., Toni, F.: Complete Assumption Labellings, *Proceedings of Computational Models of Argument (COMMA) 2014* (S. Parsons, N. Oren, C. Reed, F. Cerutti, Eds.), IOS Press, 2014.

[23] Schulz, C., Toni, F.: Justifying Answer Sets using Argumentation, *Theory and Practice of Logic Programming*, **FirstView**, 2015, 1–52.

[24] Schulz, C., Toni, F.: Logic Programming in Assumption-Based Argumentation Revisited - Semantics and Graphical Representation, *Proceedings of the 29th AAAI Conference on Artificial Intelligence* (B. Bonet, S. Koenig, Eds.), AAAI Press, 2015.

[25] Toni, F.: Assumption-Based Argumentation for Closed and Consistent Defeasible Reasoning, *New Frontiers in Artificial Intelligence, JSAI 2007 Conference and Workshops, Revised Selected Papers* (K. Satoh, A. Inokuchi, K. Nagao, T. Kawamura, Eds.), Springer Berlin Heidelberg, 2007.

[26] Toni, F.: A Generalised Framework for Dispute Derivations in Assumption-Based Argumentation, *Artificial Intelligence*, **195**, 2013, 1–43.

[27] Toni, F.: A Tutorial on Assumption-Based Argumentation, *Argument & Computation*, **5**, 2014, 89–117.

[28] Verheij, B.: Two Approaches to Dialectical Argumentation: Admissible Sets and Argumentation Stages, *Proceedings of the 8th Dutch Conference on Artificial Intelligence (NAIC)* (J.-J. C. Meyer, L. C. van der Gaag, Eds.), 1996.

[29] You, J. H., Yuan, L. Y.: Three-Valued Formalization of Logic Programming: Is It Needed?, *Proceedings of the 9th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, ACM, 1990.

[30] You, J. H., Yuan, L. Y.: On the Equivalence of Semantics for Normal Logic Programs, *The Journal of Logic Programming*, **22**(3), 1995, 211–222.

A Contraries and Expressiveness

In Definition 1 the contrary function is restricted so that each assumption has a non-assumption as its contrary. This is a small deviation from the way ABA is usually defined in then literature, where the contrary of an assumption can be an assumption or a non-assumption. In the current appendix, we show that our restriction does not affect expressiveness. That is, for each ABA framework in the "traditional" unrestricted form, there exists an equivalent ABA framework in our restricted form. As an example, consider the following ABA framework in its unrestricted form: $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, \overline{} \rangle$ with $\mathcal{A} = \{\alpha, \beta\}$, $\mathcal{L} = \mathcal{A} \cup \{c, d\}$, $\overline{\beta} = \alpha$, $\overline{\alpha} = d$ and $\mathcal{R} = \{c \leftarrow \beta\}$. This ABA framework can be translated into restricted form (satisfying Definition 1) yielding $\mathcal{F}' = \langle \mathcal{L}', \mathcal{R}', \mathcal{A}, \tilde{} \rangle$. with $\mathcal{A} = \{\alpha, \beta\}$, $\mathcal{L} = \mathcal{A} \cup \{c, d, \alpha^*\}$, $\tilde{\beta} = \alpha^*$, $\tilde{\alpha} = d$ and $\mathcal{R} = \{c \leftarrow \beta;\ \ \alpha^* \leftarrow \alpha\}$.

Hence, the idea is that every assumption that has an assumption as contrary will instead get a (brand new) non-assumption as contrary. A rule will be added from the old contrary to the new contrary.

Theorem 7. *Let $\mathcal{F} = \langle \mathcal{L}, \mathcal{R}, \mathcal{A}, ^{-} \rangle$ be an ABA framework where contraries are allowed to be assumptions or non-assumptions (that is: $^{-} : \mathcal{A} \rightarrow \mathcal{L}$). Let $\mathcal{F}' = \langle \mathcal{L}, \mathcal{R}', \mathcal{A}, \tilde{}, \rangle$ with $\mathcal{L}' = \mathcal{L} \cup \{\overline{\chi}^* \mid \chi \in \mathcal{A}$ and $\overline{\chi} \in \mathcal{A}\}$, $\mathcal{R}' = \mathcal{R} \cup \{\overline{\chi}^* \leftarrow \overline{\chi} \mid \chi \in \mathcal{A}$ and $\overline{\chi} \in \mathcal{A}\}$ and $\tilde{\chi}$ being $\overline{\chi}$ if $\overline{\chi} \in \mathcal{L} \setminus \mathcal{A}$ or $\overline{\chi}^*$ if $\overline{\chi} \in \mathcal{A}$.*
It holds that:

1. *no assumption $\chi \in \mathcal{A}$ has $\tilde{\chi} \in \mathcal{A}$*
2. *let $\chi \in \mathcal{A}$ and Asms $\subseteq \mathcal{A}$. Asms attacks χ under $\mathcal{F}$ (that is: Asms $\vdash_{\mathcal{F}} \overline{\chi}$) iff Asms attacks χ under $\mathcal{F}'$ (that is: Asms $\vdash_{\mathcal{F}'} \tilde{\chi}$)*

Proof. 1. Let $\chi \in \mathcal{A}$. We distinguish two cases:

   (a) $\overline{\chi} \in \mathcal{L} \setminus \mathcal{A}$. Then $\tilde{\chi} = \overline{\chi} \in \mathcal{L} \setminus \mathcal{A} \subseteq \mathcal{L}' \setminus \mathcal{A}$.

   (b) $\overline{\chi} \in \mathcal{A}$. Then $\tilde{\chi} = \overline{\chi}^* \in \mathcal{L}' \setminus \mathcal{A}$.

2. "$\Rightarrow$": Suppose *Asms* $\vdash_{\mathcal{F}} \overline{\chi}$. We distinguish two cases:

   (a) No rules in $\mathcal{R}$ have been used in the argument. This implies that *Asms* $= \{\overline{\chi}\}$ and that $\overline{\chi}$ is an assumption. It then follows that $\tilde{\chi} = \overline{\chi}^*$ and that there exists a rule $\overline{\chi}^* \leftarrow \overline{\chi}$ in $\mathcal{R}'$. This means there is an argument under $\mathcal{F}'$ that starts with assumption $\overline{\chi}$ and applies the rule $\overline{\chi}^* \leftarrow \overline{chi}$ to obtain conclusion $\overline{\chi}^*$. So $\{\overline{\chi}\} \vdash_{\mathcal{F}'} \overline{\chi}^*$. Since $\overline{\chi}^* = \tilde{\chi}$ it follows that this argument attacks χ under $\mathcal{F}'$. So $\{\overline{\chi}\}$ attacks χ under $\mathcal{F}'$.

   (b) At least one rule in $\mathcal{R}$ has been used to construct the argument *Asms* $\vdash_{\mathcal{F}} \overline{\chi}$. So $\overline{\chi}$ is the consequent of a rule in $\mathcal{R}$. As $\mathcal{F}$ is a flat, it follows that $\overline{\chi}$ is not an assumption, so $\tilde{\chi} = \overline{\chi}$. Since $\mathcal{R} \subseteq \mathcal{R}'$ it follows that we can construct the same argument under $\mathcal{F}'$. So *Asms* $\vdash_{\mathcal{F}'} \overline{\chi}$. As $\overline{\chi} = \tilde{\chi}$ it then follows that *Asms* attacks χ under $\mathcal{F}'$.

   "$\Leftarrow$": Suppose *Asms* $\vdash_{\mathcal{F}'} \tilde{\chi}$. We distinguish two cases:

   (a) $\tilde{\chi} = \overline{\chi}$. This means $\overline{\chi} \in \mathcal{L} \setminus \mathcal{A}$. So each rule used in the argument must be in $\mathcal{R}$ Therefore, the argument can also be constructed under $\mathcal{F}$. So *Asms* $\vdash_{\mathcal{F}} \tilde{\chi}$, which given that $\tilde{\chi} = \overline{\chi}$ implies that *Asms* $\vdash_{\mathcal{F}} \overline{\chi}$, so *Asms* attacks χ under $\mathcal{F}$.

   (b) $\tilde{\chi} = \overline{\chi}^*$. This means $\overline{\chi} \in \mathcal{A}$. As $\overline{\chi}^*$ is not an assumption in $\mathcal{F}'$, it follows that the argument has a top-rule $\overline{\chi}^* \leftarrow \overline{\chi}$. As $\overline{\chi}$ is an assumption and $\mathcal{F}'$ is a flat ABA framework, the argument cannot have any other rules. Hence, *Asms* $= \{\overline{\chi}\}$. This means that under $\mathcal{F}$ there exists an argument $\{\overline{\chi}\} \vdash_{\mathcal{F}} \overline{\chi}$. So *Asms* $= \{\overline{\chi}\}$ attacks χ under $\mathcal{F}$.

□

B Ideal Semantics

The ideal semantics for logic programs was first introduced in [1] in terms of scenarios, i.e. a logic program plus a set of NAF literals, rather than in terms of models.

Definition 16. *Let $H \subseteq HB^{\mathtt{not}}$ be a set of NAF literals and P a logic program. $\vdash$ can be considered as the standard T_P operator for logic programs treating NAF literals as atoms. Then*[6]

- *$P \cup H$ is a* scenario *of P.*
- *$P \cup H$ is* consistent *iff $\forall \mathtt{not}\ x \in H$ it holds that $P \cup H \nvdash x$.*
- *A NAF literal $\mathtt{not}\ x$ is* acceptable *w.r.t. $P \cup H$ iff $\forall K \subseteq HB^{\mathtt{not}}$ if $P \cup K \vdash x$ then $\exists \mathtt{not}\ z \in K$ such that $P \cup H \vdash z$. The set of all NAF literals which are acceptable w.r.t. $P \cup H$ is denoted $Acc(H)$.*
- *$P \cup H$ is an* admissible scenario *iff it is consistent and $H \subseteq Acc(H)$.*
- *$P \cup H$ is the* ideal scenario *if H is the maximal (w.r.t. $\subseteq$) set satisfying: For all admissible scenaria $P \cup K$, $P \cup K \cup H$ is again an admissible scenario.*
- *$P \cup H$ is a* complete scenario *iff it is consistent and $H = Acc(H)$.*
- *$P \cup H$ is a* preferred extension *iff it is a maximal (w.r.t $\subseteq$) complete scenario.*

For a set of atoms $S \subseteq HB$, $S_{HB^{\mathtt{not}}} = \{\mathtt{not}\ x \mid x \in S\}$ denotes the set of all NAF literals corresponding to atoms in S. Conversely, for a set of NAF literals $H \subseteq HB^{\mathtt{not}}$, $H_{HB} = \{x \mid \mathtt{not}\ x \in H\}$ denotes the set of all atoms corresponding to the NAF literals in H. Given a scenario $P \cup H$, $der(H) = \{x \in HB \mid P \cup H \vdash x\}$ denotes the set of atoms derivable from $P \cup H$. Furthermore, given another scenario $P \cup H_1$ ($H_1 \neq H$), $\overline{der}(P \cup H_1, H) = der(P \cup (H_1 \cup H)) \setminus der(P \cup H)$ is the set of atoms derivable from $P \cup (H_1 \cup H)$ in addition to the atoms derivable from $P \cup H$.

We prove that the ideal semantics for logic programs as originally defined (see Definition 16) is equivalent to the one given in Definition 8.

Lemma 4. *Let P be a logic program and $P \cup H_1$ a scenario. $P \cup H_1$ is an admissible scenario such that H_1 is maximal among all admissible scenaria $P \cup H$ with $H \subseteq \bigcap\{H' \mid H'$ is a preferred extension$\}$ iff $P \cup H_1$ is a complete scenario such that H_1 is maximal among all complete scenaria $P \cup C$ with $C \subseteq \bigcap\{H' \mid H'$ is a preferred extension$\}$.*

Proof. Let $H = \bigcap\{H' \mid H'$ is a preferred extension$\}$.

- From left to right: Let $P \cup H_1$ be an admissible scenario such that H_1 is maximal among all admissible scenaria $P \cup H$ with $H \subseteq \bigcap\{H' \mid H'$ is a preferred extension$\}$. Since every complete scenario is an admissible scenario by definition, there clearly cannot be a complete scenario $P \cup C$ such that $C \supseteq H_1$ and C is maximal Assume that $P \cup H_1$ is not a complete scenario, so $\exists \mathtt{not}\ x \in Acc(H_1)$ such that $\mathtt{not}\ x \notin H_1$. Then $\mathtt{not}\ x \in Acc(H)$ and thus $\mathtt{not}\ x \in Acc(H')$ for all preferred extensions $P \cup H'$ since $H = \bigcap\{H' \mid H'$ is a preferred extension$\}$. This implies that $\mathtt{not}\ x \in H'$ and thus $\mathtt{not}\ x \in H$. Since $\mathtt{not}\ x \in Acc(H_1)$, $P \cup (H_1 \cup \{\mathtt{not}\ x\})$ is an admissible scenario and $H_1 \cup \{\mathtt{not}\ x\} \subseteq H$. Thus H_1 is not maximal. Contradiction. Consequently, $P \cup H_1$ is a complete scenario and clearly there cannot be a larger complete scenario $P \cup C$ such that $C \subseteq H$ since

[6] Note that these definitions are adapted from [1]. Originally $Mand(H) = \{\mathtt{not}\ x \mid P \cup H \cup \{\neg x' \to \mathtt{not}\ x'\} \vdash \mathtt{not}\ x\}$ is also taken into consideration for all semantics, but since we are dealing with normal logic programs here, $\forall H \subseteq HB^{\mathtt{not}} : Mand(H) = H$.

every complete scenario is admissible, so $P \cup C$ would also be a larger admissible scenario (contradiction). Thus $P \cup H_1$ is a complete scenario such that H_1 is maximal among all complete scenaria $P \cup C$ with $C \subseteq \bigcap\{H' \mid H' \text{ is a preferred extension}\}$.

- From right to left: Let $P \cup H_1$ be a complete scenario such that H_1 is maximal among all complete scenaria $P \cup C$ with $C \subseteq \bigcap\{C' \mid C' \text{ is a preferred extension}\}$. Assume that there exists an admissible (but not complete) scenario $P \cup H_2$ such that $H_2 \supset H_1$ and H_2 is maximal among all admissible scenaria with $H_2 \subseteq \bigcap\{H' \mid H' \text{ is a preferred extension}\}$. Then $H_2 \subset Acc(H_2)$. Analogous to the first part of this proof, this implies that $\exists \texttt{not}\ x \in Acc(H_2)$ and that $\texttt{not}\ x \in H$, so $H_2 \cup \{\texttt{not}\ x\}$ is an admissible scenario. Contradiction since this implies that H_2 is not the maximal admissible scenario. Since by definition every complete scenario is an admissible scenario, $P \cup H_1$ is an admissible scenario such that H_1 is maximal among all admissible scenaria $P \cup H$ with $H \subseteq \bigcap\{H' \mid H' \text{ is a preferred extension}\}$.

□

Lemma 5. *Let P be a logic program. $H \subseteq HB^{\texttt{not}}$ is an ideal scenario according to Definition 16 iff $P \cup H$ is an admissible scenario where $P \cup H$ is maximal among all admissible scenaria with $H \subseteq \bigcap\{H' \mid H'$ is a preferred extension$\}$.*

Proof. From left to right: Let H be the maximal (w.r.t. $\subseteq$) set satisfying: For all admissible scenaria $P \cup K$, $P \cup K \cup H$ is again an admissible scenario. Consider first the smallest admissible scenario $P \cup K_{empty}$, where $K_{empty} = \emptyset$. Then for $P \cup K_{empty} \cup H = P \cup \emptyset \cup H$ to be admissible, $P \cup H$ has to be admissible. Now consider the largest admissible scenarios, i.e. the preferred n extensions $P \cup K_1, \ldots P \cup K_n$. Then for $P \cup K_i \cup H$ $(1 \leq i \leq n)$ to be admissible, it has to hold that $H \subseteq K_i$. Since this has to hold for all K_i, it follows that $H \subseteq \bigcap\{H' \mid H' \text{ is a preferred extension}\}$.
From right to left: Let $P \cup H$ be maximal among all admissible scenarios with $H \subseteq \bigcap\{H' \mid H'$ is a preferred extension$\}$. Then by the fundamental lemma [9] for any admissible scenario $P \cup K$, $P \cup K \cup H$ is an admissible scenario. □

Lemma 6. *Let $P \cup C_1$ and $P \cup C_2$ be two complete scenaria of P.*

1. *$C_1 \subset C_2$ iff $der(P \cup C_1) \subset (P \cup C_2)$.*
2. *$C_1 = C_2$ iff $der(P \cup C_1) = (P \cup C_2)$.*

Proof.
1. From left to right: If $C_1 \subset C_2$ then $\forall \texttt{not}\ x_1 \in C_1 : \texttt{not}\ x_1 \in C_2$ and $\exists \texttt{not}\ x_2 \in C_2$ s.t. $\texttt{not}\ x_2 \notin C_1$. Clearly $\forall x \in der(P \cup C_1) : x \in der(P \cup C_2)$, so $der(P \cup C_1) \subseteq der(P \cup C_2)$. Since $P \cup C_1$ is a complete scenario, $\texttt{not}\ x_2 \notin Acc(C_1)$. Thus, $\exists E_2 \vdash x_2$ and $\nexists \texttt{not}\ z \in E_2$ s.t. $P \cup C_1 \vdash z$, but $\exists \texttt{not}\ z' \in E_2$ s.t. $P \cup C_2 \vdash z'$. Consequently, $der(P \cup C_1) \subset der(P \cup C_2)$.
   From right to left: If $der(P \cup C_1) \subset der(P \cup C_2)$ then $Acc(C_1) \subseteq Acc(C_2)$. Since $Acc(C_1) = C_1$ and $Acc(C_2) = C_2$ it follows that $C_1 \subseteq C_2$. If $C_1 = C_2$ then clearly $der(P \cup C_1) = der(P \cup C_2)$ (contradiction). Thus, $C_1 \subset C_2$.
2. From left to right: Trivial.
   From right to left: If $der(P \cup C_1) = (P \cup C_2)$ then $Acc(C_1) = Acc(C_2)$. Since $Acc(C_1) = C_1$ and $Acc(C_2) = C_2$ it follows that $C_1 = C_2$.

□

Lemma 7. *Let $P \cup H_1, P \cup H_2, \ldots, P \cup H_n$ $(n \geq 1)$ be all n preferred extensions of P $(\forall 1 \leq i, j \leq n : i \neq j \rightarrow H_i \neq H_j)$.*

1. *There exists exactly one complete scenario $P \cup C_1$ such that C_1 is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $C \subseteq \bigcap_{1 \leq i \leq n} H_i$.*

2. *There exists exactly one complete scenario $P \cup C_1$ such that $der(P \cup C_1)$ is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $der(P \cup C) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$.*

Proof. 1. Assume there exists another complete scenario $P \cup C_2$ $(C_2 \neq C_1)$ such that C_2 is also maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $C \subseteq \bigcap_{1 \leq i \leq n} H_i$. Consequently, $C_1 \not\subset C_2$ and $C_2 \not\subset C_1$. Let $H = \bigcap_{1 \leq i \leq n} H_i$. Then clearly $C_1, C_2 \neq H$ and thus $C_1 \subset H$ and $C_2 \subset H$. It follows that $der(P \cup C_1) \subseteq der(P \cup H)$ and $der(P \cup C_2) \subseteq der(P \cup H)$. Clearly $P \cup H$ is consistent as H is a subset of every preferred extension. Since $C_1, C_2 \subset H$ it follows that $C_1 \cup C_2 \subseteq H$, and thus $P \cup (C_1 \cup C_2)$ is consistent too, and that $der(P \cup (C_1 \cup C_2)) \subseteq der(P \cup H)$. Furthermore, since $C_1 = Acc(C_1)$ and $C_2 = Acc(C_2)$ it follows that $C_1 \cup C_2 \subseteq Acc(C_1 \cup C_2)$. Note that clearly $Acc(C_1 \cup C_2) \subseteq Acc(H)$ since $der(P \cup (C_1 \cup C_2)) \subseteq der(P \cup H)$. If $C_1 \cup C_2 = Acc(C_1 \cup C_2)$ we derive a contradiction since this means that $P \cup (C_1 \cup C_2)$ is a complete scenario and $(C_1 \cup C_2) \subseteq H$, implying that C_1 is not maximal with respect to H. Thus, $C_1 \cup C_2 \subset Acc(C_1 \cup C_2)$, so $\exists \texttt{not } x \in Acc(C_1 \cup C_2)$ such that $\texttt{not } x \notin C_1 \cup C_2$. Then $\texttt{not } x \in Acc(H)$, so $\texttt{not } x \in Acc(H_i)$ for all preferred extensions H_i and consequently $\texttt{not } x \in H_i$. But then $\texttt{not } x \in H$. This means that $P \cup (C_1 \cup C_2 \cup \{\texttt{not } x\})$ is consistent (as $C_1 \cup C_2 \cup \{\texttt{not } x\}$ is a subset of / equal to H and thus a subset of every preferred extension) and $C_1 \cup C_2 \cup \{\texttt{not } x\} \subseteq Acc(C_1 \cup C_2 \cup \{\texttt{not } x\})$ (since $\texttt{not } x \in Acc(C_1 \cup C_2)$ and $C_1 \cup C_2 \subset Acc(C_1 \cup C_2)$). So either $C_1 \cup C_2 \cup \{\texttt{not } x\}$ is a complete scenario, which is a contradiction, or $C_1 \cup C_2 \cup \{\texttt{not } x\} \subset Acc(C_1 \cup C_2 \cup \{\texttt{not } x\})$. Then $\exists \texttt{not } z \in Acc(C_1 \cup C_2 \cup \{\texttt{not } x\})$ such that $\texttt{not } z \notin (C_1 \cup C_2 \cup \{\texttt{not } x\})$. Again in turns out that $\texttt{not } z \in H$, $P \cup (C_1 \cup C_2 \cup \{\texttt{not } x, \texttt{not } z\})$ is consistent, and $C_1 \cup C_2 \cup \{\texttt{not } x, \texttt{not } z\} \subseteq Acc(C_1 \cup C_2 \cup \{\texttt{not } x, \texttt{not } z\})$. We note that there are only finitely many $\texttt{not } x \in H$ with $\texttt{not } x \notin (C_1 \cup C_2)$. Let us denote the set of all such $\texttt{not } x$ by H''. If we keep adding elements from H'' to $C_1 \cup C_2$ as we did for $\texttt{not } x$ and $\texttt{not } z$, we will see that $C_1 \cup C_2 \cup H'' \subseteq Acc(C_1 \cup C_2 \cup H'')$ has to hold. If $C_1 \cup C_2 \cup H'' \subset Acc(C_1 \cup C_2 \cup H'')$ then $\exists \texttt{not } d \in Acc(C_1 \cup C_2 \cup H'')$ such that $\texttt{not } d \notin C_1 \cup C_2 \cup H''$. But then $\texttt{not } d \in H$ which is a contradiction since $H = C_1 \cup C_2 \cup H''$. Thus $C_1 \cup C_2 \cup H'' = Acc(C_1 \cup C_2 \cup H'')$ which is a contradiction since we assumed that $P \cup H$ is not a complete scenario. Therefore, there cannot exist another complete scenario $P \cup C_2$ such that C_2 is also maximal with respect to H.

2. Assume there exists another complete scenario $P \cup C_2$ $(C_2 \neq C_1)$ such that $der(P \cup C_2)$ is also maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $der(P \cup C) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$. Then $der(P \cup C_2) = der(P \cup C_1)$ or $der(P \cup C_2) \not\subset der(P \cup C_1)$ and $der(P \cup C_1) \not\subset der(P \cup C_2)$. In the first case $C_2 = C_1$ by Lemma 6(2) (contradiction), in the second case $C_2 \not\subset C_1$ and $C_1 \not\subset C_2$ by Lemma 6(1). Then $\forall \texttt{not } x \in C_2$ with $\texttt{not } x \notin C_1$ it holds that $\texttt{not } x \in Acc(C_2)$. Since $der(P \cup C_2) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$, meaning that all atoms derivable from $P \cup C_2$ are also derivable from every preferred extension, it holds that $\texttt{not } x \in Acc(H_i)$ for all preferred extensions $P \cup H_i$. Then $\texttt{not } x \in H_i$ and thus $\texttt{not } x \in H$. Consequently, $C_2 \subseteq H$. The same holds for all

$\mathtt{not}\ y \in C_1$ with $\mathtt{not}\ y \notin C_2$, so $C_1 \subseteq H$. In fact, $C_1, C_2 \subset H$ because if either of them is equal to H, e.g. $C_2 = H$, then the other one is a subset of the first, so $C_1 \subset C_2$ which by Lemma 6 implies that $der(C_1) \subset der(C_2)$ (contradiction). The proof continues analogous to the first case. Since $P \cup H$ is consistent and $C_1, C_2 \subset H$ it follows that $C_1 \cup C_2 \subseteq H$, and thus $P \cup (C_1 \cup C_2)$ is consistent too, and that $der(P \cup (C_1 \cup C_2)) \subseteq der(P \cup H)$. Furthermore, since $C_1 = Acc(C_1)$ and $C_2 = Acc(C_2)$ it follows that $C_1 \cup C_2 \subseteq Acc(C_1 \cup C_2)$. Note that clearly $Acc(C_1 \cup C_2) \subseteq Acc(H)$ since $der(P \cup (C_1 \cup C_2)) \subseteq der(P \cup H)$. If $C_1 \cup C_2 = Acc(C_1 \cup C_2)$ we derive a contradiction since this means that $P \cup (C_1 \cup C_2)$ is a complete scenario and thus by Lemma 6 $der(P \cup C_1) \subset der(P \cup (C_1 \cup C_2))$, implying that $der(P \cup C_1)$ is not maximal with respect to $\bigcap_{1 \leq i \leq n} der(P \cup H_i)$. Thus, $C_1 \cup C_2 \subset Acc(C_1 \cup C_2)$, so $\exists \mathtt{not}\ z \in Acc(C_1 \cup C_2)$ such that $\mathtt{not}\ z \notin C_1 \cup C_2$. Then $\mathtt{not}\ z \in Acc(H)$, so $\mathtt{not}\ z \in Acc(H_i)$ for all preferred extensions H_i and consequently $\mathtt{not}\ z \in H_i$. But then $\mathtt{not}\ z \in H$. This means that $P \cup (C_1 \cup C_2 \cup \{\mathtt{not}\ z\})$ is consistent (as $C_1 \cup C_2 \cup \{\mathtt{not}\ z\}$ is a subset of / equal to H and thus a subset of every preferred extension) and $C_1 \cup C_2 \cup \{\mathtt{not}\ z\} \subseteq Acc(C_1 \cup C_2 \cup \{\mathtt{not}\ z\})$ (since $\mathtt{not}\ z \in Acc(C_1 \cup C_2)$ and $C_1 \cup C_2 \subset Acc(C_1 \cup C_2)$). So either $P \cup (C_1 \cup C_2 \cup \{\mathtt{not}\ z\})$ is a complete scenario, which is again a contradiction due to the maximality of the derivations (by Lemma 6), or $C_1 \cup C_2 \cup \{\mathtt{not}\ z\} \subset Acc(C_1 \cup C_2 \cup \{\mathtt{not}\ z\})$. Then $\exists \mathtt{not}\ w \in Acc(C_1 \cup C_2 \cup \{\mathtt{not}\ z\})$ such that $\mathtt{not}\ w \notin (C_1 \cup C_2 \cup \{\mathtt{not}\ z\})$. Again in turns out that $\mathtt{not}\ w \in H$, $P \cup (C_1 \cup C_2 \cup \{\mathtt{not}\ z, \mathtt{not}\ w\})$ is consistent, and $C_1 \cup C_2 \cup \{\mathtt{not}\ z, \mathtt{not}\ w\} \subseteq Acc(C_1 \cup C_2 \cup \{\mathtt{not}\ z, \mathtt{not}\ w\})$. We note that there are only finitely many $\mathtt{not}\ v \in H$ with $\mathtt{not}\ v \notin (C_1 \cup C_2)$. Let us denote the set of all such $\mathtt{not}\ v$ by H''. If we keep adding elements from H'' to $C_1 \cup C_2$ as we did for $\mathtt{not}\ z$ and $\mathtt{not}\ w$, we will see that $C_1 \cup C_2 \cup H'' \subseteq Acc(C_1 \cup C_2 \cup H'')$ has to hold. If $C_1 \cup C_2 \cup H'' \subset Acc(C_1 \cup C_2 \cup H'')$ then $\exists \mathtt{not}\ u \in Acc(C_1 \cup C_2 \cup H'')$ such that $\mathtt{not}\ u \notin C_1 \cup C_2 \cup H''$. But then $\mathtt{not}\ u \in H$ which is a contradiction since $H = C_1 \cup C_2 \cup H''$. Thus $C_1 \cup C_2 \cup H'' = Acc(C_1 \cup C_2 \cup H'')$ which is a contradiction since then $P \cup H$ is a complete scenario and $der(P \cup H)$ is maximal with respect to $\bigcap_{1 \leq i \leq n} der(P \cup H_i)$ (by Lemma 6). Therefore, there cannot exist another complete scenario $P \cup C_2$ such that $der(P \cup C_2)$ is also maximal with respect to $\bigcap_{1 \leq i \leq n} der(P \cup H_i)$. □

Lemma 8. *Let $P \cup H_1, P \cup H_2, \ldots, P \cup H_n$ ($n \geq 1$) be all n preferred extensions of P ($\forall 1 \leq i, j \leq n : i \neq j \rightarrow H_i \neq H_j$) and let $H = \bigcap_{1 \leq i \leq n} H_i$. Then $H_i = H \cup K_i$ and $\bigcap_{1 \leq i \leq n} der(P \cup H_i) = der(P \cup H) \cup X$ with $X = \bigcap_{1 \leq i \leq n} \overline{der}(P \cup K_i, H)$.*

Proof. Since $H = \bigcap_{1 \leq i \leq n} H_i$ trivially $H \subset H_i$, so $H_i = H \cup K_i$ with $K_i \neq \emptyset$ Recall that $\overline{der}(P \cup K_i, H)$ is the set of atoms additionally derivable from $P \cup (K_i \cup H)$, so from $P \cup H_i$, as compared to $P \cup H$. Thus, $der(P \cup H_i) = der(P \cup H) \cup \overline{der}(P \cup K_i, H)$. Then $X = \bigcap_{1 \leq i \leq n} \overline{der}(P \cup K_i, H)$ denotes the set of atoms which are derivable from all $P \cup H_i$ but not from $P \cup H$. X could of course be empty, but might be non-empty in the case that an atom is derivable in different ways (i.e. using different NAF literals). Consequently, $\bigcap_{1 \leq i \leq n} der(P \cup H_i) = der(P \cup H) \cup X$. □

Lemma 9. *Let $P \cup H_1, P \cup H_2, \ldots, P \cup H_n$ ($n \geq 1$) be all n preferred extensions of P ($\forall 1 \leq i, j \leq n : i \neq j \rightarrow H_i \neq H_j$).*

- *If $P \cup C_1$ is a complete scenario such that C_1 is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $C \subseteq \bigcap_{1 \leq i \leq n} H_i$ then $der(C_1)$ is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $\overline{der}(P \cup C) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$.*

- *If $P \cup C_1$ is a complete scenario such that $der(C_1)$ is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $der(P \cup C) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$ then C_1 is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $C \subseteq \bigcap_{1 \leq i \leq n} H_i$.*

Proof. Let $H = \bigcap_{1 \leq i \leq n} H_i$.

- Let $P \cup C_1$ be a complete scenario such that C_1 is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $C \subseteq \bigcap_{1 \leq i \leq n} H_i$. Assume that there exists another complete scenario $P \cup C_2$ ($C_2 \neq C_1$) such that $\overline{der}(P \cup C_2)$ is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $der(P \cup C) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$. Then $der(P \cup C_2) \subseteq der(P \cup H) \cup X$ by Lemma 8, and due to the maximality $der(P \cup C_2) \not\subset der(P \cup C_1)$. Assume $C_2 \subset C_1$, then by Lemma 6 $der(P \cup C_2) \subset der(P \cup C_1)$ (contradiction). So either $C_1 \subset C_2$ or $C_2 \nsubseteq C_1$ and $C_1 \nsubseteq C_2$. In both cases $C_2 \nsubseteq H$ since $P \cup C_1$ is the unique (by Lemma 7) maximal complete scenario with $C_1 \subseteq H$. We split the proof in two cases depending on the relation of C_1 and H:

  1. Case 1: $C_1 = H$. Then $der(P \cup C_1) = der(P \cup H)$. Since $der(P \cup C_2)$ is maximal among all complete scenaria $der(P \cup C)$ with $der(P \cup C) \subseteq der(P \cup H) \cup X$ and since by Lemma 7 there exists only one such maximal complete scenario, it follows that $der(P \cup H) \subset der(P \cup C_2)$. Thus $der(P \cup C_2) = der(P \cup H) \cup X'$ where $X' \subseteq X$ and $X' \neq \emptyset$. By Lemma 6 $H \subset C_2$, so $C_2 = H \cup K$ ($K \subseteq HB^{\texttt{not}}, K \neq \emptyset, K \cap H = \emptyset$) and thus $der(P \cup C_2) = der(P \cup H) \cup \overline{der}(P \cup K, H)$. Since complete scenaria form a complete semilattice, there exists a preferred scenario $P \cup H_j$ such that $C_2 \subseteq H_j$. By Lemma 8, $H_j = H \cup K_j$. Thus, $K \subseteq K_j$. Clearly $K \nsubseteq Acc(H)$ since $P \cup H$ is a complete scenario, so $K \subseteq Acc(C_2)$ due to the atoms in $\overline{der}(P \cup K, H) \subseteq \overline{der}(P \cup K_j, H)$ rather than due to the atoms in $der(P \cup H)$. Since $der(P \cup C_2) = der(P \cup H) \cup X'$ it follows that $\overline{der}(P \cup K, H) = X'$. Since the atoms in $\overline{der}(P \cup K, H) = X'$ are responsible for $K \subseteq Acc(C_2)$, it follows that for any set of NAF literals which derives the atoms in X', K is acceptable with respect to this set. Since $X' \subseteq X$ this means that K is acceptable with respect to any set of NAF literals deriving X and thus acceptable with respect to all preferred extension. Then, K is a subset of all H_i of preferred extensions $P \cup H_i$ and therefore $K \subset H$. Contradiction.
  2. Case 2: $C_1 \subset H$. Then $H = C_1 \cup H'$ ($H' \subseteq HB^{\texttt{not}}, H' \neq \emptyset, C_1 \cap H' = \emptyset$). Thus $der(P \cup H) = der(P \cup C_1) \cup \overline{der}(P \cup H', C_1)$. Since $der(P \cup C_1) \subseteq der(P \cup H)$ and since $der(P \cup C_2) \subseteq der(P \cup H) \cup X$ is the unique maximal (by Lemma 7) complete scenario with respect to $der(P \cup H) \cup X$ it follows that $der(P \cup C_1) \subset der(P \cup C_2)$. Then by Lemma 6, $C_1 \subset C_2$, so $C_2 = C_1 \cup K$ ($K \subseteq HB^{\texttt{not}}, K \neq \emptyset, K \cap C_1 = \emptyset$). Then $der(P \cup C_2) = der(P \cup C_1) \cup \overline{der}(P \cup K, C_1)$. Since $der(P \cup C_2) \subseteq der(P \cup H) \cup X$ we get that $der(P \cup C_2) \subseteq der(P \cup C_1) \cup \overline{der}(P \cup H', C_1) \cup X$. Using the equivalence of $der(P \cup C_2)$ it follows that $der(P \cup C_1) \cup \overline{der}(P \cup K, C_1) \subseteq der(P \cup C_1) \cup \overline{der}(P \cup H', C_1) \cup X$. Thus, $\overline{der}(P \cup K, C_1) \subseteq \overline{der}(P \cup H', C_1) \cup X$. Since C_1 is a complete scenario, it holds that $\forall \texttt{not}\ a \in K : \texttt{not}\ a \notin Acc(C_1)$. Thus, $K \subset Acc(C_2)$ due to the atoms in $\overline{der}(P \cup K, C_1)$ rather than due to the atoms

in $der(P \cup C_1)$. Since both $\overline{der}(P \cup H', C_1)$ and X are derivable in all preferred extensions $P \cup H_i$, this means that $K \subseteq Acc(H_i)$ and thus $K \subseteq H_i$, so $K \subseteq H$. Thus $C_2 \subseteq C_1 \cup H$, but since $C_1 \subset H$ this implies that $C_2 \subseteq H$. Contradiction since $C_1 \subset C_2$ but C_1 is the unique (by Lemma 7) maximal complete scenario which is a subset of or equal to H.

- Let $P \cup C_1$ be a complete scenario such that $der(P \cup C_1)$ is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $der(P \cup C) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$. Assume that there exists another complete scenario $P \cup C'$ ($C' \neq C$) such that C' is maximal (w.r.t $\subseteq$) among all complete scenaria $P \cup C$ with $C \subseteq \bigcap_{1 \leq i \leq n} H_i$. Then by the first part, $P \cup C'$ is a complete scenario such that $der(C')$ is maximal among all complete scenaria $P \cup C$ with $der(P \cup C) \subseteq \bigcap_{1 \leq i \leq n} der(P \cup H_i)$. Contradiction since by Lemma 7 there exists a unique complete scenario which is maximal with respect to $\bigcap_{1 \leq i \leq n} der(P \cup H_i)$.

□

Theorem 8. *Let P be a logic program.*

- *If $\langle T, F\rangle$ is an ideal model of P then $P \cup F_{HB^{\mathtt{not}}}$ is an ideal scenario of P where $T = der(P \cup F_{HB^{\mathtt{not}}})$.*
- *If $P \cup H$ is an ideal scenario of P then $\langle der(P \cup H), H_{HB}\rangle$ is an ideal model of P.*

Proof.

- Let $\langle T, F\rangle$ be an ideal model. By Definition 8, $\langle T, F\rangle$ is a 3-valued stable model where T is maximal (w.r.t. $\subseteq$) among all 3-valued stable models of P with $T \subseteq \bigcap\{T' \mid \langle T', F'\rangle$ is a regular model$\}$. By Theorem 2.1 in [30], $\langle T', F'\rangle$ is a regular model iff $P \cup F'_{HB^{\mathtt{not}}}$ is a preferred extension where $T' = der(P \cup F'_{HB^{\mathtt{not}}})$. Thus $\bigcap\{T' \mid \langle T', F'\rangle$ is a regular model$\} = \bigcap\{der(P \cup H) \mid P \cup H$ is a preferred extension$\}$. Furthermore, by Theorem 3.1 in [20] and Corollary 4.16 in [3], if $\langle T, F\rangle$ is a 3-valued stable model then $P \cup F_{HB^{\mathtt{not}}}$ is a complete scenario where $T = der(P \cup F_{HB^{\mathtt{not}}})$. Thus, $P \cup F_{HB^{\mathtt{not}}}$ is a complete scenario where $der(P \cup F_{HB^{\mathtt{not}}})$ is maximal among all complete scenaria with $der(P \cup F_{HB^{\mathtt{not}}}) \subseteq \bigcap\{der(P \cup H) \mid P \cup H$ is a preferred extension$\}$. By Lemma 9, $P \cup F_{HB^{\mathtt{not}}}$ is a complete scenario where $F_{HB^{\mathtt{not}}}$ is maximal among all complete scenaria with $F_{HB^{\mathtt{not}}} \subseteq \bigcap\{H \mid P \cup H$ is a preferred extension$\}$. By Lemma 4, $P \cup F_{HB^{\mathtt{not}}}$ is an admissible scenario where $F_{HB^{\mathtt{not}}}$ is maximal among all admissible scenaria with $F_{HB^{\mathtt{not}}} \subseteq \bigcap\{H \mid P \cup H$ is a preferred extension$\}$. Then by Lemma 5, $P \cup F_{HB^{\mathtt{not}}}$ is an ideal scenario.
- Let $P \cup H$ be an ideal scenario. By Lemma 5, $P \cup H$ is an admissible scenario such that H is maximal among all admissible scenaria with $H \subseteq \bigcap\{H' \mid H'$ is a preferred extension$\}$. Then by Lemma 4, $P \cup H$ is a complete scenario such that H is maximal among all complete scenaria with $H \subseteq \bigcap\{H' \mid P \cup H'$ is a preferred extension$\}$. By Lemma 9, $P \cup H$ is a complete scenario such that $der(P \cup H)$ is maximal among all complete scenaria with $der(P \cup H) \subseteq \bigcap\{der(P \cup H') \mid P \cup H'$ is a preferred extension$\}$. By Theorem 3.1 in [20] and Corollary 4.16 in [3], if $P \cup H$ is a complete scenario then $\langle der(P \cup H), H_{HB}\rangle$ is a 3-valued stable model. Furthermore, by Theorem 2.1 in [30] $P \cup H'$ is a preferred extension iff $\langle T, F\rangle$ is a regular model with $T = der(P \cup H')$ and $F = H'_{HB}$. Consequently, $\bigcap\{der(P \cup H') \mid P \cup H'$ is a preferred extension$\} = \bigcap\{T \mid \langle T, F\rangle$ is a regular model$\}$. Thus, $\langle der(P \cup H), H_{HB}\rangle$ is maximal among all

3-valued stable models $\langle T, F\rangle$ with $T \subseteq \bigcap\{T' \mid \langle T', F'\rangle$ is a regular model$\}$. Then $\langle der(P \cup H), H_{HB}\rangle$ is an ideal model.

$\square$