Lösungshinweise zu Kapitel 10

zu Selbsttestaufgabe 10.8 (anfechtbare Ableitung) In $\mathcal{P}_{tina}$ sind

$$\begin{aligned} &\mathit{chicken}(\mathit{tina}). \\ &\mathit{bird}(\mathit{tina}) \leftarrow \mathit{chicken}(\mathit{tina}). \end{aligned}$$

enthalten, woraus man $\mathit{bird}(\mathit{tina})$ ableiten kann. Daraus lässt sich mit der in $\mathcal{P}_{tina}$ enthaltenen Regel

$$\mathit{flies}(\mathit{tina}) \prec \mathit{bird}(\mathit{tina}).$$

nun (unsicher) $\mathit{flies}(\mathit{tina})$ ableiten. Andererseits kann mit

$$\begin{aligned} &\mathit{chicken}(\mathit{tina}). \\ &\neg\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}). \end{aligned}$$

(ebenfalls unsicher) auch $\neg\mathit{flies}(\mathit{tina})$ abgeleitet werden.

zu Selbsttestaufgabe 10.10 (Monotonie von DeLP Ableitungen) Sowohl sichere als auch unsichere Ableitungen sind gemäß ihrer Definition nur von den Anteilen des Programms abhängig, die sie explizit nutzen. Eine Ableitung bzgl. eines Programms $\mathcal{P}$ ist daher auch eine Ableitung bzgl. eines beliebigen Programms $\mathcal{P}'$ mit $\mathcal{P} \subseteq \mathcal{P}'$.

zu Selbsttestaufgabe 10.13 (DeLP-Argument bei sicheren Ableitungen) Ist h sicher aus $\mathcal{P} = (\Pi, \Delta)$ ableitbar, so ist $\Pi \cup \emptyset$ nicht widersprüchlich, und es gilt $\Pi \cup \emptyset \mathrel{|\!\sim} h$. Da die leere Menge keine echte Teilmenge enthält und bereits minimal ist, ist $\langle \emptyset, h \rangle$ ein Argument und auch das einzige Argument für h.

zu Selbsttestaufgabe 10.14 (sichere DeLP Ableitungen nicht angreifbar) Sei $\langle \emptyset, h \rangle$ ein Argument; insbesondere gilt dann $\Pi \mathrel{|\!\sim} h$. Dann gibt es auch für jede Menge $\mathcal{A}$ von unsicheren Regeln eine Ableitung von h aus $\Pi \cup \mathcal{A}$ (vgl. Selbsttestaufgabe 10.10).

Gäbe es nun ein Argument $\langle \mathcal{A}, \neg h \rangle$, so würde $\Pi \cup \mathcal{A} \mathrel{|\!\sim} \neg h$ gelten. Da aber auch $\Pi \cup \mathcal{A} \mathrel{|\!\sim} h$ gelten muss, wäre damit $\Pi \cup \mathcal{A}$ widersprüchlich; daher kann $\langle \mathcal{A}, \neg h \rangle$ kein Argument sein.

zu Selbsttestaufgabe 10.27 (Spezifität) Jede Aktivierungsmenge H von $\langle \mathcal{A}_3, h_3 \rangle$ enthält die Literale $\mathit{chicken}(\mathit{tina})$ und $\mathit{scared}(\mathit{tina})$; jedes solche H aktiviert daher auch $\langle \mathcal{A}_1, h_1 \rangle$. Andererseits aktiviert $H' = \{\mathit{chicken}(\mathit{tina})\}$ zwar $\langle \mathcal{A}_1, h_1 \rangle$, aber H' aktiviert nicht $\langle \mathcal{A}_3, h_3 \rangle$. Daher gilt $\langle \mathcal{A}_3, h_3 \rangle \succ_{spec} \langle \mathcal{A}_1, h_1 \rangle$.

zu Selbsttestaufgabe 10.29 (Spezifität) Das Argument

$$\langle \mathcal{A}_1,\, h_1\rangle = \langle \{d \prec a.\},\, d\rangle$$

wird von H genau dann aktiviert, wenn $a \in H$ gilt. Das Argument

$$\langle \mathcal{A}_2,\, h_2\rangle = \langle \{\neg d \prec c.,\ c \prec a.\},\, \neg d\rangle$$

wird von H genau dann aktiviert, wenn $c \in H$ oder $a \in H$ gilt. Also aktiviert jedes H, das $\langle \mathcal{A}_1,\, h_1\rangle$ aktiviert, auch $\langle \mathcal{A}_2,\, h_2\rangle$, aber nicht umgekehrt. Daher gilt

$$\langle \mathcal{A}_1,\, h_1\rangle = \langle \{d \prec a.\},\, d\rangle \ \succ_{spec}\ \langle \{\neg d \prec c.,\ c \prec a.\},\, \neg d\rangle = \langle \mathcal{A}_2,\, h_2\rangle.$$

zu Selbsttestaufgabe 10.33 (schlagendes Gegenargument) Das Argument $\langle \mathcal{A}_3,\, h_3\rangle$ ist ein schlagendes Gegenargument zu dem Argument $\langle \mathcal{A}_1,\, h_1\rangle$. (Beachten Sie, dass wir in Selbsttestaufgabe 10.27 bereits gezeigt hatten, dass $\langle \mathcal{A}_3,\, h_3\rangle \succ_{spec} \langle \mathcal{A}_1,\, h_1\rangle$ gilt.)

zu Selbsttestaufgabe 10.45 (akzeptable Argumentationsfolge) Für die Argumentationsfolge

$$\begin{aligned}\Lambda^{tf} = [&\langle \{\mathit{flies}(\mathit{tina}) \prec \mathit{bird}(\mathit{tina}).\},\, \mathit{flies}(\mathit{tina})\rangle,\\ &\langle \{\neg\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}).\},\, \neg\mathit{flies}(\mathit{tina})\rangle,\\ &\langle \{\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}), \mathit{scared}(\mathit{tina}).\},\, \mathit{flies}(\mathit{tina})\rangle]\end{aligned}$$

untersuchen wir die vier Bedingungen aus Definition 10.43:

1. Λ^{tf} ist endlich.
2. Die Menge

   $$\begin{aligned}\Lambda^{tf}_S = \{&\langle \{\mathit{flies}(\mathit{tina}) \prec \mathit{bird}(\mathit{tina}).\},\, \mathit{flies}(\mathit{tina})\rangle,\\ &\langle \{\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}), \mathit{scared}(\mathit{tina}).\},\, \mathit{flies}(\mathit{tina})\rangle\}\end{aligned}$$

   der unterstützenden Argumente ist konkordant, da die Menge

   $$\mathcal{P}_{tina} \cup \{\mathit{flies}(\mathit{tina}) \prec \mathit{bird}(\mathit{tina}).,\ \mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}), \mathit{scared}(\mathit{tina}).\}$$

   nicht widersprüchlich ist. Weiterhin ist die Menge

   $$\Lambda^{tf}_I = \{\langle \{\neg\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}).\},\, \neg\mathit{flies}(\mathit{tina})\rangle\}$$

   der interferierenden Argumente ebenfalls konkordant, da die Menge

   $$\mathcal{P}_{tina} \cup \{\neg\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}).\}$$

   nicht widersprüchlich ist.
3. Keines der Argumente in Λ^{tf} ist ein Subargument eines vorhergehenden Arguments in Λ^{tf}.

4. Die Folge Λ^{tf} enthält keine zwei direkt aufeinanderfolgenden, blockierenden Angriffe.

Daher ist Λ^{tf} eine akzeptable Argumentationsfolge.

zu Selbsttestaufgabe 10.46 (DeLP)

1. Argument

   Ein Argument ist eine minimale, nicht-widersprüchliche Menge von unsicheren Regeln für eine Ableitung.

   Ein Beispiel für ein Argument ist $\langle\{(p -\!\prec a, b), (a -\!\prec b, c)\}, p\rangle$.

2. Subargument

   Ein Argument ist Subargument eines anderen Arguments, wenn es nur unsichere Regeln verwendet, die in dem anderen Argument auch vorkommen. Das Superargument $\langle\mathcal{A}, h\rangle$ nutzt die Schlussfolgerung des Subarguments $\langle\mathcal{A}', h'\rangle$, und aufgrund der Minimalität des Superarguments ist dieses zwangsläufig auf die Schlussfolgerung des Subarguments angewiesen.

   Das Argument $\langle\{a -\!\prec b, c\}, a\rangle$ ist ein Subargument von $\langle\{(p -\!\prec a, b), (a -\!\prec b, c)\}, p\rangle$.

3. Gegenargument

   Ein Argument ist ein Gegenargument eines anderen Arguments, wenn in der Argumentation des letzteren ein Literal geschlussfolgert wird, das zusammen mit den sicheren Regeln im Widerspruch zum geschlussfolgerten Literal des Gegenarguments steht; d.h. $\langle\mathcal{A}, h_1\rangle$ ist ein Gegenargument zu $\langle\mathcal{B}, h_2\rangle$, wenn es ein Subargument $\langle\mathcal{B}', h_2'\rangle$ von $\langle\mathcal{B}, h_2\rangle$ gibt, so dass h_1, h_2' und die sicheren Regeln zusammen widersprüchlich sind.

   Das Argument $\langle\{a -\!\prec b, c\}, a\rangle$ ist ein Gegenargument zu $\langle\{(\neg p -\!\prec \neg a, b, d), (\neg a -\!\prec c)\}, \neg p\rangle$, da $\{a, \neg a\} \cup \Pi$ widersprüchlich ist.

4. Verallgemeinerte Spezifität

   Ein Argument ist spezifischer als ein anderes Argument, wenn alle Literale, die es aktivieren, auch das andere Argument aktivieren, dieses umgekehrt aber nicht gilt.

   Das Argument $\langle\{a -\!\prec b, c\}, a\rangle$ ist spezifischer als $\langle\{\neg a -\!\prec c\}, \neg a\rangle$, da ersteres die Aktivierungsmenge $\{b, c\}$ hat und letzteres die Aktivierungsmenge $\{c\}$. Die Menge $\{b, c\}$ aktiviert auch $\langle\{\neg a -\!\prec c\}, \neg a\rangle$, die Menge $\{c\}$ jedoch nicht $\langle\{a -\!\prec b, c\}, a\rangle$.

5. Erfolgreiches Gegenargument

   Ein erfolgreiches Gegenargument ist ein schlagendes oder blockierendes Gegenargument; d.h. es ist ein Gegenargument für ein Subargument des anderen Arguments, sodass das Subargument bezüglich des Gegenarguments entweder weniger spezifisch oder unvergleichbar ist.

Das Argument $\langle\{a \prec b, c\}, a\rangle$ ist ein erfolgreiches Gegenargument zu $\langle\{(\neg p \prec \neg a, b, d), (\neg a \prec c)\}, \neg p\rangle$, da es eine widersprüchliche Schlussfolgerung zu Subargument $(\neg a \prec c)$ hat (Gegenargument) und zusätzlich spezifischer ist (erfolgreich); siehe Lösung zu Aufgabenteil (4.).

6. Konkordanz

   Eine Menge von Argumenten heißt konkordant, wenn die verwendeten unsicheren Regeln zusammen mit den Fakten und sicheren Regeln nicht widersprüchlich sind.

   Die Menge $\{\langle\{(q \prec p, b), (p \prec a, b), (a \prec b, c)\}, q\rangle, \langle\{a \prec b, c\}, a\rangle\}$ ist konkordant, da zusammen mit den Fakten (sichere Regeln gibt es hier keine) die Schlussfolgerungsmenge $\{b, c, d, q, p, a\}$ ergibt. Diese Menge ist nicht widersprüchlich.

   Ein Gegenbeispiel wäre die Menge $\{\langle\{(a \prec b, c)\}, a\rangle, \langle\{\neg a \prec c\}, \neg a\rangle\}$. Zusammen mit der Menge der Fakten würde sich hier die Schlussfolgerungsmenge $\{b, c, d, a, \neg a\}$ ergeben, welche widersprüchlich ist.

7. Akzeptable Argumentationsfolge

   Eine akzeptable Argumentationsfolge ist eine endliche Folge von Argumenten, in der

   - jedes Argument sein Vorgängerargument erfolgreich angreift,
   - kein Argument Subargument eines vorangegangen Arguments ist,
   - es keine zwei direkt aufeinanderfolgenden blockierenden Angriffe gibt,
   - sowohl die Pro- als auch die Contra-Argumente jeweils zusammengenommen konkordant sind.

   Hierbei greift ein Argument ein vorangegangenes Argument erfolgreich an, wenn das angreifende Argument nicht unspezifischer als das Subargument ist, in dem es das vorangegangene Argument angreift. Sind dabei das angreifende Argument und das Subargument, in dem das vorangegangene Argument angegriffen wird, bezüglich der Spezifität unvergleichbar, so handelt es sich um einen blockierenden Angriff.

   Die Folge

   $$\Lambda = \big\{\langle\{(\neg q \prec b)\}, \neg q\rangle, \langle\{(q \prec p, b), (p \prec a, b), (a \prec b, c)\}, q\rangle, \\ \langle\{(\neg p \prec \neg a, b, d), (\neg a \prec c)\}, \neg p\rangle\big\}$$

   ist eine akzeptable Argumentationsfolge:

   - Jedes Argument greift sein Vorgängerargument erfolgreich an:

  - Die minimale Aktivierungsmenge des ersten Arguments ist $\{b\}$ und die des zweiten Arguments sind $\{b, c\}$, $\{a, b\}$ und $\{p, b\}$. Da $\{b\} \subset \{b, c\}$, $\{b\} \subset \{a, b\}$ und $\{b\} \subset \{p, b\}$ ist, gilt

    $$\langle\{(\neg q -\!\prec b)\}, \neg q\rangle \prec_{spec} \langle\{(q -\!\prec p, b), (p -\!\prec a, b), (a -\!\prec b, c)\}, q\rangle \ .$$

    Somit schlägt das zweite Argument das erste Argument.
  - Die minimalen Aktivierungsmengen des dritten Arguments sind $\{b, c, d\}$ und $\{\neg a, b, d\}$. Ferner greift das dritte Argument das zweite in $\langle\{(p -\!\prec a, b), (a -\!\prec b, c)\}, p\rangle$ an, was die minimale Aktivierungsmengen $\{b, c\}$ und $\{a, b\}$ besitzt. Daher gilt

    $$\langle\{(p -\!\prec a, b), (a -\!\prec b, c)\}, p\rangle \not\prec_{spec} \langle\{(\neg p -\!\prec \neg a, b, d), (\neg a -\!\prec c)\}, \neg p\rangle$$

    und

    $$\langle\{(p -\!\prec a, b), (a -\!\prec b, c)\}, p\rangle \not\succ_{spec} \langle\{(\neg p -\!\prec \neg a, b, d), (\neg a -\!\prec c)\}, \neg p\rangle \ .$$

    Somit blockiert das dritte Argument das zweite Argument.

- Kein Argument ist Subargument eines vorangegangen Arguments:

  Gemäß der Erläuterung des Begriffs des *Subarguments* aus Aufgabenteil (1.) lässt sich feststellen, dass keine der in den Argumenten enthaltenen Regelmengen in der Regelmenge eines Vorgängerargumentes enthalten ist.

- Es gibt keine zwei direkt aufeinanderfolgenden blockierenden Angriffe:

  Nur das dritte Argument blockiert das zweite Argument (gemäß dem ersten Spiegelstrich). Daher ist auch diese Bedingung erfüllt.

- Sowohl die Pro- als auch die Contra-Argumente sind jeweils zusammengenommen konkordant:

  Das erste und das dritte Argument sind zusammengenommen konkordant, da zusammen mit dem Faktenwissen die Schlussfolgerungsmenge $\{b, c, d, \neg q, \neg p, \neg a\}$ entsteht, welche nicht widersprüchlich ist.

  Aus dem zweiten Argument ergibt sich zusammen mit dem Faktenwissen die Schlussfolgerungsmenge $\{b, c, d, a, p, q\}$, welche ebenfalls nicht widersprüchlich ist.

zu Selbsttestaufgabe 10.53 (DeLP-Garanten für Tina) Für die Anfrage $?\mathit{flies}(\mathit{tina})$ können wir mit den Regeln aus Δ das Argument $\langle\{\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}), \mathit{scared}(\mathit{tina}).\}, \mathit{flies}(\mathit{tina})\rangle$ bilden. Da es dazu kein erfolgreiches Gegenargument gibt, besteht der markierte dialektische Baum

$$\langle\{\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}), \mathit{scared}(\mathit{tina}).\}, \mathit{flies}(\mathit{tina})\rangle^U$$

nur aus einem mit U markiertem Wurzelknoten. Daher ist das Argument $\langle\{\mathit{flies}(\mathit{tina}) \prec \mathit{chicken}(\mathit{tina}), \mathit{scared}(\mathit{tina}).\}, \mathit{flies}(\mathit{tina})\rangle$ ebenfalls ein Garant für $\mathit{flies}(\mathit{tina})$.

zu Selbsttestaufgabe 10.54 (DeLP-Antworten für Nixon-Raute) Da weder $\mathit{pazifist}(\mathit{nixon})$ noch $\neg\mathit{pazifist}(\mathit{nixon})$ garantiert ist, werden die Anfragen $?\mathit{pazifist}(\mathit{nixon})$ und $?\neg\mathit{pazifist}(\mathit{nixon})$ beide mit *undecided* beantwortet.

zu Selbsttestaufgabe 10.55 (Aktienmarkt) Die Menge der Literale, welche aus $\mathcal{P}_{stock}$ geschlussfolgert werden können, lautet:

$$\mathcal{F} = \{\mathit{good_price}(\mathit{acme}), \mathit{in_fusion}(\mathit{acme}, \mathit{steel}), \mathit{strong}(\mathit{steel}), \mathit{buy_stock}(\mathit{acme}), \neg\mathit{buy_stock}(\mathit{acme}), \mathit{risky_company}(\mathit{acme}), \neg\mathit{risky_company}(\mathit{acme})\}$$

Mit den folgenden Teilmengen von $\mathcal{P}_{stock}$

$$\mathcal{A}_1 = \{\mathit{buy_stock}(\mathit{acme}) \prec \mathit{good_price}(\mathit{acme}).\}$$

$$\mathcal{A}_2 = \{\neg\mathit{buy_stock}(\mathit{acme}) \prec \mathit{good_price}(\mathit{acme}), \mathit{risky_company}(\mathit{acme}).,\ \underbrace{\mathit{risky_company}(\mathit{acme}) \prec \mathit{in_fusion}(\mathit{acme}, \mathit{steel}).}_{\mathcal{A}_2'}\}$$

$$\mathcal{A}_3 = \{\neg\mathit{risky_company}(\mathit{acme}) \prec \mathit{in_fusion}(\mathit{acme}, \mathit{steel}), \mathit{strong}(\mathit{steel}).\}$$

lassen sich die folgenden Argumente bilden:

$$\langle\mathcal{A}_1,\ \mathit{buy_stock}(\mathit{acme})\rangle$$
$$\langle\mathcal{A}_2,\ \neg\mathit{buy_stock}(\mathit{acme})\rangle$$
$$\langle\mathcal{A}_2',\ \mathit{risky_company}(\mathit{acme})\rangle$$
$$\langle\mathcal{A}_3,\ \neg\mathit{risky_company}(\mathit{acme})\rangle$$

Für diese Argumente gelten die Beziehungen:

$$\langle\mathcal{A}_2,\ \neg\mathit{buy_stock}(\mathit{acme})\rangle \succ_{spec} \langle\mathcal{A}_1,\ \mathit{buy_stock}(\mathit{acme})\rangle$$
$$\langle\mathcal{A}_3,\ \neg\mathit{risky_company}(\mathit{acme})\rangle \succ_{spec} \langle\mathcal{A}_2',\ \mathit{risky_company}(\mathit{acme})\rangle$$

Damit erhalten wir für die Anfrage $?\mathit{buy_stock}(\mathit{acme})$ den folgenden markierten dialektischen Baum:

$$\langle \mathcal{A}_1,\ buy_stock(acme)\rangle^U$$
$$\Big|$$
$$\langle \mathcal{A}_2,\ \neg buy_stock(acme)\rangle^D$$
$$\Big|$$
$$\langle \mathcal{A}_3,\ \neg risky_company(acme)\rangle^U$$

Die Antwort auf die Anfrage $?buy_stock(acme)$ lautet folglich *yes*, da mit $\langle \mathcal{A}_1,\ buy_stock(acme)\rangle$ ein Garant für $buy_stock(acme)$ gefunden wurde.

zu Selbsttestaufgabe 10.56 (Ameisenigel)

1. $\mathcal{P} = (\Pi, \Delta)$ mit:

$$\Pi = \left\{ \begin{array}{c} s(x) \leftarrow a(x). \\ \neg k(x) \leftarrow a(x). \\ \neg e(x) \leftarrow a(x) \wedge m(x). \\ m(ben). \\ a(ben). \\ a(anna). \\ s(emma). \\ k(nemo). \end{array} \right\}, \qquad \Delta = \left\{ \begin{array}{c} \neg e(x) \prec s(x). \\ e(x) \prec a(x). \\ e(x) \prec k(x). \end{array} \right\}$$

2. Wir konstruieren zunächst Regelketten für die folgenden Behauptungen:
   (a) Anna legt keine Eier.
   $\mathcal{R}_1 = \{\neg e(anna) \prec s(anna); s(anna) \leftarrow a(anna); a(anna)\}$
   (b) Ben legt Eier.
   $\mathcal{R}_2 = \{e(ben) \prec a(ben); a(ben)\}$
   (c) Anna legt Eier.
   $\mathcal{R}_3 = \{e(anna) \prec a(anna); a(anna)\}$
   (d) Ben legt keine Eier.
   $\mathcal{R}_4 = \{\neg e(ben) \leftarrow a(ben) \wedge m(ben); m(ben); a(ben)\}$

3. Wir konstruieren aus den Regelketten des Aufgabenteils (2.) Argumente, sofern dies möglich ist:
   (a) $\mathcal{A}_1 = \{\neg e(anna) \prec s(anna)\}$; $\langle \mathcal{A}_1, \neg e(anna)\rangle$
   (b) Hier lässt sich kein Argument bilden, da die einzige unsichere Regel, die hier zu wählen ist, einen Widerspruch zu Π ergeben würde:
   Die einzige Regel aus Δ, die hier wählbar ist, ist $e(ben) \prec a(ben)$, dann ist aber aus $\Pi \cup \{e(ben) \prec a(ben)\}$ sowohl abzuleiten, dass Ben Eier legt, als auch, dass er keine Eier legt.

(c) $\mathcal{A}_3 = \{e(anna) \multimap\!\!\prec a(anna)\}$; $\langle \mathcal{A}_3, e(anna)\rangle$

(d) $\mathcal{A}_4 = \{\}$; $\langle \emptyset, \neg e(ben)\rangle$

4. • Für $c = anna$:

$$\langle \underbrace{\{e(anna) \multimap\!\!\prec a(anna)\}}_{A_1},\ e(anna)\rangle \qquad \text{und}$$
$$\langle \underbrace{\{\neg e(anna) \multimap\!\!\prec s(anna)\}}_{A_2},\ \neg e(anna)\rangle.$$

Die einzige minimale Aktivierungsmenge von $\langle A_1,\ e(anna)\rangle$ ist $\{a(anna)\}$. Die minimalen Aktivierungsmengen von $\langle A_2, \neg e(anna)\rangle$ sind $\{a(anna)\}$ und $\{s(anna)\}$. Somit ist $\langle A_1,\ e(anna)\rangle \succ_{spec} \langle A_2, \neg e(anna)\rangle$. Daher ist $\langle A_1,\ e(anna)\rangle$ ein schlagendes Gegenargument zu $\langle A_2, \neg e(anna)\rangle$.
Es gilt:

$$\langle \mathrm{A}_2, \neg e(anna)\rangle^{\mathrm{D}} \qquad \text{und} \qquad \langle \mathrm{A}_1, e(anna)\rangle^{\mathrm{U}}$$
$$\Big|$$
$$\langle \mathrm{A}_1, e(anna)\rangle^{\mathrm{U}}$$

Damit gelten für die Antworten auf die Anfragen:

- $e(anna)$? YES
- $\neg e(anna)$? NO.

• Für $c = ben$:

$$\langle \underbrace{\emptyset}_{A_1};\ \neg e(ben)\rangle$$

Dies ein sicheres Argument, das kein Gegenargument hat und das auch kein Gegenargument zu einem anderen Argument ist, und es gilt:

$$\langle \mathrm{A}_1, \neg e(ben)\rangle^{\mathrm{U}}$$

Damit gelten für die Antworten auf die Anfragen:

- $e(ben)$? NO
- $\neg e(ben)$? YES.

- Für $c = nemo$:

  $$\langle \underbrace{\{e(nemo) \prec\!\!\!- k(nemo)\}}_{A_1},\ e(nemo)\rangle$$

  Die einzige minimale Aktivierungsmenge von $\langle A_1,\ e(nemo)\rangle$ ist $\{k(nemo)\}$.
  Es gilt:

  $$\langle \mathrm{A}_1, e(nemo)\rangle^{\mathrm{U}}$$

  Damit gelten für die Antworten auf die Anfragen:

  – $e(nemo)$? YES
  – $\neg e(nemo)$? NO.

zu Selbsttestaufgabe 10.63 (Konfliktfreiheit und angegriffene Mengen) Wir geben einige konfliktfreie Argumentmengen mit vier bzw. fünf Elementen an und die davon jeweils angegriffenen Mengen:

konfliktfreie Argumentmenge $\mathcal{S}$	angegriffene Menge $\mathcal{S}^+$
$\{A, C, D, F\}$	$\{B, E\}$
$\{A, C, D, G\}$	$\{B, E, H\}$
$\{A, C, D, H\}$	$\{B, E, G\}$
$\{A, C, E, H\}$	$\{B, F, G\}$
$\{A, C, D, F, G\}$	$\{B, E, H\}$
$\{A, C, D, F, H\}$	$\{B, E, G\}$

zu Selbsttestaufgabe 10.66 (charakteristische Funktion) Für die insgesamt sechs Argumentmengen $\mathcal{S}$ ergibt sich $\mathbf{F}(\mathcal{S})$ wie folgt:

Argumentmenge $\mathcal{S}$	$\mathbf{F}(\mathcal{S})$
$\emptyset$	$\{D\}$
$\{A\}$	$\{D\}$
$\{A, C\}$	$\{A, C, D\}$
$\{A, C, D\}$	$\{A, C, D, F\}$
$\{A, C, D, F\}$	$\{A, C, D, F\}$
$\{A, C, D, F, G\}$	$\{A, C, D, F, G\}$

zu Selbsttestaufgabe 10.76 (Extensionen) Wir bestimmen nacheinander alle zulässigen, bevorzugten, vollständigen, grundierten und stabilen Extensionen.

1. zulässige Extensionen: Eine zulässige Extension $\mathcal{S} \subseteq \mathcal{A}$ muss jedes Element in $\mathcal{S}$ verteidigen. Da die Argumente in $\mathcal{S}_{na} \subseteq \mathcal{A}$ mit

$$\mathcal{S}_{na} = \{A_1, \ldots, A_6, A_9, A_{11}\}$$

gar nicht angegriffen werden, ist jede Teilmenge von $\mathcal{S}_{na}$ eine zulässige Extension. Für jede dieser Teilmengen, die A_9 enthält, ergibt sich durch Hinzunahme von

- A_7 oder
- A_{10} oder
- A_7 und A_{10}

wieder eine zulässige Extension. Beachten Sie, dass A_8 in keiner zulässigen Extension sein kann, da A_8 von keiner Teilmenge von $\mathcal{A}$ verteidigt wird.

2. bevorzugte Extensionen: Da eine bevorzugte Extension $\mathcal{S}_b$ eine zulässige Extension ist, die bzgl. Mengeninklusion maximal ist, gilt

$$\mathcal{S}_{na} = \{A_1, \ldots, A_6, A_9, A_{11}\} \subseteq \mathcal{S}_b$$

da die Elemente in $\mathcal{S}_{na}$ gar nicht angegriffen werden. Wegen der Konfliktfreiheit von $\mathcal{S}_b$ muss

$$A_8 \notin \mathcal{S}_b$$

gelten, da A_9 das Argument A_8 angreift. Da A_7 und A_{10} von A_9 verteidigt werden, muss weiterhin

$$\{A_7, A_{10}\} \subseteq \mathcal{S}_b$$

gelten. Damit ergibt sich insgesamt

$$\mathcal{S}_b = \{A_1, \ldots, A_6, A_7, A_9, A_{10}, A_{11}\}$$

als einzige bevorzugte Extension.

3. grundierte Extensionen: Der kleinste Fixpunkt von $\mathbf{F}$ ergibt sich wie folgt:

$$\begin{aligned}
\mathbf{F}(\emptyset) &= \{A_1, \ldots, A_6, A_9, A_{11}\} \\
\mathbf{F}(\mathbf{F}(\emptyset)) &= \{A_1, \ldots, A_6, A_7, A_9, A_{10}, A_{11}\} \\
\mathbf{F}(\mathbf{F}(\mathbf{F}(\emptyset))) &= \mathbf{F}(\mathbf{F}(\emptyset)) = \mathcal{S}_b
\end{aligned}$$

Daher ist $\mathcal{S}_b$ auch die grundierte Extension.

4. vollständige Extensionen: Da eine vollständige Extension jedes Element, das von ihr verteidigt wird, enthält, müssen alle Elemente aus $\mathcal{S}_b$ auch in einer vollständigen Extension liegen. Da außerdem

$$\mathbf{F}(\mathcal{S}_b) = \mathcal{S}_b$$

gilt, ist $\mathcal{S}_b$ auch eine vollständige Extension. Weil $\mathbf{F}(\mathbf{F}(\emptyset)) = \mathcal{S}_b$ die grundierte Extension ist, ist $\mathcal{S}_b$ minimale vollständige Extension. Da $\mathcal{S}_b$ weiterhin nicht erweitert werden kann, ohne die Konfliktfreiheit zu verlieren, ist $\mathcal{S}_b$ auch eine maximale vollständige Extension. Damit ist $\mathcal{S}_b$ die einzige vollständige Extension.

5. stabile Extensionen: Da eine stabile Extension eine konfliktfreie Menge ist, die jedes Argument, das nicht in ihr enthalten ist, angreift, müssen alle Elemente aus $\mathcal{S}_b$ auch in einer stabilen Extension liegen. Da

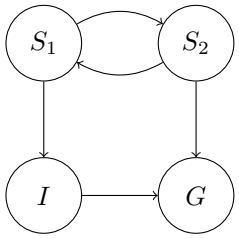
$$\mathcal{S}_b{}^+ = \{A_8\} = \mathcal{A} \setminus \mathcal{S}_b$$

gilt, ist $\mathcal{S}_b$ auch eine stabile Extension.

zu Selbsttestaufgabe 10.78 (grundierte Extension) Die grundierte Extension $\mathbf{F}^n(\emptyset)$ mit $\mathbf{F}^n(\emptyset) = \mathbf{F}(\mathbf{F}^n(\emptyset))$ ist vollständig (nach Definition) und minimal, da wegen der Monotonie von $\mathbf{F}$ für jede vollständige Extension $\mathcal{S}$ gilt: $\mathcal{S} = \mathbf{F}(\mathcal{S}) \supseteq \mathbf{F}(\emptyset)$.

zu Selbsttestaufgabe 10.79 (maximale vollständige Extensionen) Sei $\mathcal{E}$ eine maximale vollständige Extension. Annahme: $\mathcal{E}$ ist nicht bevorzugt. Dann gibt es eine zulässige Extension $\mathcal{E}'$ mit $\mathcal{E} \subsetneq \mathcal{E}'$, und dazu gibt es eine bevorzugte Extension $\mathcal{E}''$ mit $\mathcal{E}' \subseteq \mathcal{E}''$, also insgesamt auch $\mathcal{E} \subsetneq \mathcal{E}''$. Nach Theorem 10.77(2) ist $\mathcal{E}''$ auch vollständig. Dies ist ein Widerspruch zu der Voraussetzung, dass $\mathcal{E}$ maximal unter den vollständigen Extensionen ist.

zu Selbsttestaufgabe 10.82 (Abstraktes Argumentationssystem)

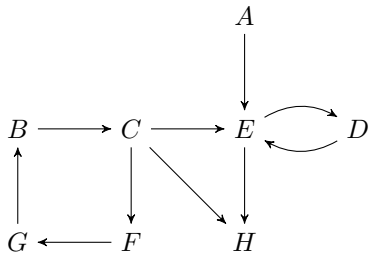


Zu beachten ist, dass es keinen Angriff von G zu I gibt, da eine Person als unschuldig angenommen wird, und I explizit besiegt werden muss, um die Schuld zu beweisen.

zu Selbsttestaufgabe 10.83 (Anwendung abstrakter Argumentationstheorie) Seien $\mathcal{A} = \{A, B, C, D, E, F, G, H\}$ eine Menge von Argumenten und $\hookrightarrow$ eine Angriffsrelation auf $\mathcal{A} \times \mathcal{A}$, die definiert ist über die Tupel

$$\{(A \hookrightarrow E), (E \hookrightarrow D), (D \hookrightarrow E), (B \hookrightarrow C), (C \hookrightarrow E),\\ (C \hookrightarrow H), (C \hookrightarrow F), (E \hookrightarrow H), (F \hookrightarrow G), (G \hookrightarrow B)\}.$$

1. Graph zu dem Argumentationssystem $\langle \mathcal{A}, \hookrightarrow \rangle$:



2. Grundierte Extension von $\langle \mathcal{A}, \hookrightarrow \rangle$:

   Die Berechnung erfolgt über die Iteration der charakteristischen Funktion $\boldsymbol{F}$ mit der Suche nach einem Fixpunkt

$$\begin{aligned} \boldsymbol{F}(\emptyset) &= \{A\} \\ \boldsymbol{F}^2(\emptyset) = \boldsymbol{F}(\{A\}) &= \{A, D\} \\ \boldsymbol{F}^3(\emptyset) = \boldsymbol{F}(\{A, D\}) &= \{A, D\} \end{aligned}$$

   Also ist $\{A, D\}$ die grundierte Extension von $(\mathcal{A}, \hookrightarrow)$.

3. Die in Aufgabenteil (2.) bestimmte Extension ist

   - **konfliktfrei**, da es kein Element $x \in \{A, D\}$ gibt, für das gilt, dass es ein Element $y \in \{A, D\}$ gibt mit $x \hookrightarrow y$;
   - **zulässig**, da $\{A, D\} \subseteq \boldsymbol{F}(\{A, D\})$;
   - **vollständig**, da $\{A, D\} = \boldsymbol{F}(\{A, D\})$;
   - nicht **bevorzugt**, da sie nicht bzgl. Mengeninklusion maximal ist; beispielsweise könnten C und G noch hinzugenommen werden und die Extension wäre immer noch zulässig;
   - nicht **stabil**, da nicht für alle Elemente $y \in \mathcal{A} \setminus \{A, D\}$ gilt, dass es ein $x \in \{A, D\}$ gibt mit $x \hookrightarrow y$. Beispielsweise wird B nicht von $\{A, D\}$ angegriffen.

4. ***Kommentar:*** Grundsätzlich ist es sinnvoll, mit den vollständigen Extensionen zu beginnen und daraus die bevorzugten und stabilen Extensionen zu bestimmen, da sowohl die bevorzugten als auch die stabilen Extensionen beide auch vollständige Extensionen sind.

   (a) vollständige Extensionen:

   Eine vollständige Extension muss (nach Definition) zulässig sein und alle Elemente, die von der Extension verteidigt werden, müssen auch enthalten sein. Daher muss

   - wenn A enthalten ist, auch D enthalten sein,

- wenn B enthalten ist, auch F enthalten sein,
- wenn C enthalten ist, auch G enthalten sein,
- wenn F enthalten ist, auch B enthalten sein,
- wenn G enthalten ist, auch C enthalten sein,
- wenn B, A oder B, D enthalten sind, auch H enthalten sein.

(1) A ist immer enthalten (da A nicht angegriffen wird), somit muss auch D immer enthalten sein. Es entsteht die Extension $\mathcal{S}_1^v$.
Es verbleiben die Möglichkeiten C, F, G, B oder H hinzuzunehmen:
(2) Nimmt man C hinzu, so muss auch G enthalten sein; umgekehrt gilt dies auch. Es entsteht die Extension $\mathcal{S}_2^v$. Weitere Elemente können nicht hinzugenommen werden, da die Extension sonst nicht mehr konfliktfrei wäre.
(3) Nimmt man F hinzu, so muss auch B enthalten sein; umgekehrt gilt dies auch. Dann muss aber auch H enthalten sein, da sowohl B, A als auch B, D enthalten sind. H kann somit nur zusammen mit F und B enthalten sein. Es entsteht die Extension $\mathcal{S}_3^v$. Weitere Elemente können nicht hinzugenommen werden, da die Extension sonst nicht mehr konfliktfrei wäre.

$\mathcal{S}_1^v = \{A, D\}$
Begründung: Da grundiert und damit die kleinste vollständige Extension. (A verteidigt D und D verteidigt sich selbst. Da beide in $\mathcal{S}_1^v$ enthalten sind, ist $\mathcal{S}_1^v$ zulässig. Da D auch das einzige Element ist, welches von A verteidigt wird, und A das einzige, welches von D verteidigt wird, ist $\mathcal{S}_1^v$ vollständig.)

$\mathcal{S}_2^v = \{G, C, D, A\}$
Begründung: G verteidigt C und umgekehrt; beide sind auch in $\mathcal{S}_1^b$ enthalten. D verteidigt sich selbst und wird von A verteidigt (beide in $\mathcal{S}_2^v$ enthalten), und A wird nicht angegriffen und somit von der leeren Menge verteidigt. Damit ist $\mathcal{S}_2^v$ zulässig.

$\mathcal{S}_3^v = \{H, B, F, D, A\}$
Begründung: H wird von D, A und B verteidigt (die alle in $\mathcal{S}_3^v$ enthalten sind), B verteidigt F und umgekehrt, und beide sind auch in $\mathcal{S}_3^v$ enthalten. D verteidigt sich selbst und wird von A verteidigt (beide in $\mathcal{S}_2^v$ enthalten), und A wird nicht angegriffen und somit von der leeren Menge verteidigt. Damit ist $\mathcal{S}_3^v$ zulässig.

(b) bevorzugte Extensionen:

Bevorzugte Extensionen sind auch vollständige Extensionen, daher sind die drei vollständigen Extensionen $\mathcal{S}_1^v$, $\mathcal{S}_2^v$ und $\mathcal{S}_3^v$ die einzigen möglichen Kandidaten und müssen untersucht werden.

$\mathcal{S}_1^b = \{G, C, D, A\}$
Begründung: Die Hinzunahme eines weiteres Knoten ist nicht möglich (z.B. E), da die Menge dann nicht mehr zulässig wäre. Damit ist $\mathcal{S}_1^b$ bevorzugt, weil maximal zulässig bezüglich Mengeninklusion.

$\mathcal{S}_2^b = \{H, B, F, D, A\}$
Begründung: Die Hinzunahme eines weiteres Knoten ist nicht möglich (z.B. G), da die Menge dann nicht mehr zulässig wäre. Damit ist $\mathcal{S}_2^b$ bevorzugt, weil maximal zulässig bezüglich Mengeninklusion.

$\mathcal{S}_1^v = \{A, D\}$ ist keine bevorzugte Extension, da $\mathcal{S}_1^v \subset \mathcal{S}_1^b$ und sie somit nicht maximal zulässig bezüglich Mengeninklusion ist (siehe Aufgabenteil (3.)).

(c) stabile Extensionen:

Da jede stabile Extension auch vollständig ist (aber nicht umgekehrt), müssen diese Extensionen unter den vollständigen Extensionen zu finden sein, es gibt also nur 3 Kandidaten.

$\mathcal{S}_1^{\text{stab}} = \{G, C, D, A\}$
Begründung: Als Begründung lässt sich anführen, dass $\{G, C, D, A\}$ konfliktfrei ist (die enthaltenen Elemente greifen sich nicht untereinander an) und alle weiteren Elemente im Graphen von $\{G, C, D, A\}$ angegriffen werden: G greift B an, C greift F, E und H an; damit sind alle Elemente im Graphen abgedeckt und somit ist $\mathcal{S}^{\text{stab}}$ stabil.

$\mathcal{S}_2^{\text{stab}} = \{H, B, F, A, D\}$
Begründung: Durch alle in der jeweiligen Menge enthaltenen Knoten werden alle nicht enthaltenen Knoten im Graphen angegriffen.

$\mathcal{S}_1^v = \{A, D\}$ ist keine stabile Extension, da nicht alle restlichen Knoten des Graphens angegriffen werden (siehe Aufgabenteil (3.)).

zu Selbsttestaufgabe 10.96 (vollständige Labelingfunktion) In allen drei Fällen sind die "⇒"-Beziehungen identisch mit den "Dann"-Bedingungen in der Definition einer vollständigen Labelingfunktion. Die "⇐"-Beziehungen ergeben sich aus der Tatsache, dass für jedes Argument genau ein Label aus $\{in, out, undec\}$ vergeben werden muss und die aufgelisteten Bedingungen exklusiv und erschöpfend sind.

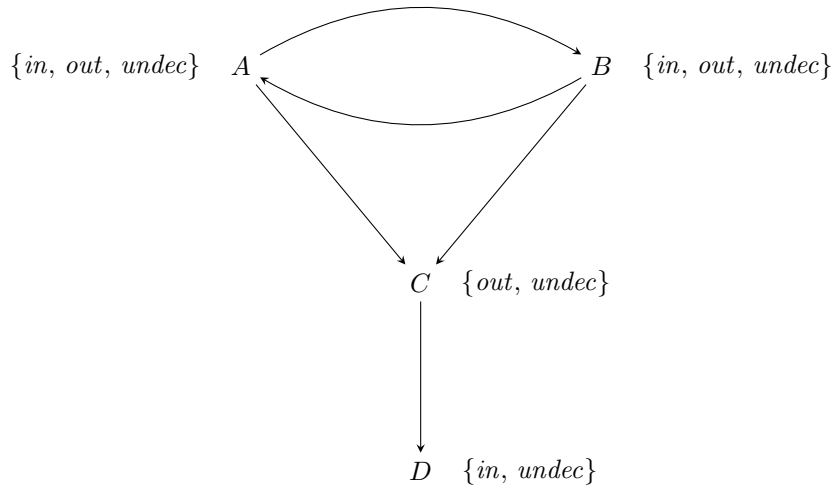
zu Selbsttestaufgabe 10.107 (Status unter Vollständigkeit) Es ist $Stat^{compl}(A) = \{in\}$ genau dann, wenn A in allen vollständigen Extensionen liegt und damit auch in deren Schnitt, d.h., in der grundierten Extension.

Sei nun $Stat^{compl}(A) = \{out\}$, d.h., $\ell(A) = out$ für alle vollständigen Labelingfunktionen ℓ, insbesondere also auch für die Labelingfunktion ℓ_g der grundierten Extension $\mathcal{E}_g$. Dann gibt es ein $B \in \mathcal{A}$ mit $\ell_g(B) = in$ und $B \hookrightarrow A$, also $A \in {\mathcal{E}_g}^+$. Ist umgekehrt $A \in {\mathcal{E}_g}^+$, so gibt es ein $B \in \mathcal{E}_g$ mit $B \hookrightarrow A$, also auch $\ell_g(B) = in$, und nach Definition 10.100 ist dann auch $\ell(B) = in$ für alle vollständigen Labelingfunktionen ℓ. Damit ist $\ell(A) = out$ für alle vollständigen Labelingfunktionen ℓ und daher $Stat^{compl}(A) = \{out\}$.

zu Selbsttestaufgabe 10.108 (Status unter Vollständigkeit) Die vollständigen Labelingfunktionen für den Graphen aus Abbildung 10.9 sind wie folgt:

A	B	C	D
out	*in*	*out*	*in*
in	*out*	*out*	*in*
undec	*undec*	*undec*	*undec*

Der Status aller Argumente unter Vollständigkeit für den Graphen aus Abbildung 10.9 ist damit wie in der folgenden Abbildung angegeben:



zu Selbsttestaufgabe 10.109 (Labeling) Es ist sinnvoll, zunächst alle zulässigen Labelingfunktionen aufzuführen und daraus die anderen zu bestimmen.

Wir starten mit der Beobachtung, dass A nicht *out* sein kann, da A keinen Angreifer hat. Also müssen wir nur die Fälle $l(A) = in$ und $l(A) = undec$ untersuchen:

$\mathbf{l(A) = in}$:

- $l(E) = in$: nicht möglich!
- $l(E) = undec$: $\Rightarrow l(D) = undec$
  - $l(H) = in$: nicht möglich
  - $l(H) = out$: $\Rightarrow l(C) = in \Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
  - $l(H) = undec$:
    - $l(C) = in$: $\Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
    - $l(C) = out$: $\Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$
    - $l(C) = undec$: $\Rightarrow l(B) = undec \Rightarrow l(G) = undec \Rightarrow l(F) = undec$
- $l(E) = out$:
  - $l(D) = in$:
    - $l(H) = in$: $\Rightarrow l(C) = out \Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$
    - $l(H) = out$: $\Rightarrow l(C) = in \Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
    - $l(H) = undec$:
      - $l(C) = in$: $\Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
      - $l(C) = out$: $\Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$
      - $l(C) = undec$: $\Rightarrow l(B) = undec \Rightarrow l(G) = undec \Rightarrow l(F) = undec$
  - $l(D) = out$: nicht möglich!
  - $l(D) = undec$:
    - $l(H) = in$: $\Rightarrow l(C) = out \Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$
    - $l(H) = out$: $\Rightarrow l(C) = in \Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
    - $l(H) = undec$:
      - $l(C) = in$: $\Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
      - $l(C) = out$: $\Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$

· $l(C) = undec$: $\Rightarrow l(B) = undec \Rightarrow l(G) = undec \Rightarrow l(F) = undec$

l(A) = undec:

- $l(E) = in$: nicht möglich!
- $l(E) = out$:
  - $l(H) = in$: $\Rightarrow l(D) = in \Rightarrow l(C) = out \Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$
  - $l(H) = out$: $\Rightarrow l(C) = in \Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
    - $l(D) = undec$
    - $l(D) = in$
  - $l(H) = undec$:
    - $l(D) = in$:
      - $l(C) = in$: $\Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
      - $l(C) = out$: $\Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$
      - $l(C) = undec$: $\Rightarrow l(B) = undec \Rightarrow l(G) = undec \Rightarrow l(F) = undec$
    - $l(D) = out$: nicht möglich!
    - $l(D) = undec$: $\Rightarrow l(C) = in \Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
- $l(E) = undec$: $\Rightarrow l(D) = undec$
  - $l(H) = in$: nicht möglich!
  - $l(H) = out$: $\Rightarrow l(C) = in \Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
  - $l(H) = undec$:
    - $l(C) = in$: $\Rightarrow l(B) = out \Rightarrow l(G) = in \Rightarrow l(F) = out$
    - $l(C) = out$: $\Rightarrow l(B) = in \Rightarrow l(G) = out \Rightarrow l(F) = in$
    - $l(C) = undec$: $\Rightarrow l(B) = undec \Rightarrow l(G) = undec \Rightarrow l(F) = undec$

Damit ergeben sich folgende Labelingfunktionen:

1. Zulässig:

	A	B	C	D	E	F	G	H	in(l)
l_1	in	out	in	undec	undec	out	in	out	$\{A, C, G\}$
l_2	in	out	in	undec	undec	out	in	undec	$\{A, C, G\}$
l_3	in	in	out	undec	undec	in	out	undec	$\{A, B, F\}$
l_4	in	undec	undec	undec	undec	undec	undec	undec	$\{A\}$
l_5	in	in	out	in	out	in	out	in	$\{A, B, D, F, H\}$
l_6	in	out	in	in	out	out	in	out	$\{A, C, D, G\}$
l_7	in	out	in	in	out	out	in	undec	$\{A, C, D, G\}$
l_8	in	in	out	in	out	in	out	undec	$\{A, B, D, F\}$
l_9	in	undec	undec	in	out	undec	undec	undec	$\{A, D\}$
l_{10}	in	in	out	undec	out	in	out	in	$\{A, B, F, H\}$
l_{11}	in	out	in	undec	out	out	in	out	$\{A, C, G\}$
l_{12}	in	out	in	undec	out	out	in	undec	$\{A, C, G\}$
l_{13}	in	in	out	undec	out	in	out	undec	$\{A, B, F\}$
l_{14}	in	undec	undec	undec	out	undec	undec	undec	$\{A\}$
l_{15}	undec	in	out	in	out	in	out	in	$\{B, D, F, H\}$
l_{16}	undec	out	in	undec	out	out	in	out	$\{C, G\}$
l_{17}	undec	out	in	in	out	out	in	out	$\{C, D, G\}$
l_{18}	undec	out	in	in	out	out	in	undec	$\{C, D, G\}$
l_{19}	undec	in	out	in	out	in	out	undec	$\{B, D, F\}$
l_{20}	undec	undec	undec	in	out	undec	undec	undec	$\{D\}$
l_{21}	undec	out	in	undec	out	out	in	undec	$\{C, G\}$
l_{22}	undec	out	in	undec	undec	out	in	out	$\{C, G\}$
l_{23}	undec	out	in	undec	undec	out	in	undec	$\{C, G\}$
l_{24}	undec	in	out	undec	undec	in	out	undec	$\{B, F\}$
l_{25}	undec	undec	undec	undec	undec	undec	undec	undec	$\{\}$

2. Vollständig:

   $l(X) = undec$ nur, wenn es keine *in*-Angreifer und mindestens einen Angreifer gibt, der nicht *out* ist.
   Daraus folgt, dass Knoten ohne Angreifer in vollständigen Labelingfunktionen nicht *undec* sein können:

	A	B	C	D	E	F	G	H	in(l)
l_5	in	in	out	in	out	in	out	in	$\{A, B, D, F, H\}$
l_6	in	out	in	in	out	out	in	out	$\{A, C, D, G\}$
l_9	in	undec	undec	in	out	undec	undec	undec	$\{A, D\}$

3. Grundiert (vollständig und $in(l)$ minimal):

	A	B	C	D	E	F	G	H	in(l)
l_9	in	undec	undec	in	out	undec	undec	undec	$\{A, D\}$

4. Bevorzugt (vollständig und $in(l)$ maximal):

	A	B	C	D	E	F	G	H	in(l)
l_5	in	in	out	in	out	in	out	in	$\{A, B, D, F, H\}$
l_6	in	out	in	in	out	out	in	out	$\{A, C, D, G\}$

5. Stabil (vollständig und $undec(l) = \emptyset$):

	A	B	C	D	E	F	G	H	in(l)
l_5	in	in	out	in	out	in	out	in	$\{A, B, D, F, H\}$
l_6	in	out	in	in	out	out	in	out	$\{A, C, D, G\}$

zu Selbsttestaufgabe 10.112 (Codierung für vollständige Extension) Nach Proposition 10.111 ist nur noch der Teil $\mathcal{P}_{\mathsf{AF}}$ des erweiterten logischen Programms $\mathcal{P}_{\mathsf{AF}} \cup \mathcal{P}_{comp}$ zu bestimmen, dieser lautet für das Argumentationssystems AF aus Beispiel 10.58 wie folgt:

$$\begin{aligned}\mathcal{P}_{\mathsf{AF}} = \{&arg(A)., arg(B)., arg(C)., arg(D)., arg(E)., arg(F)., arg(G)., arg(H).,\\ &attack(B, A)., attack(B, C)., attack(C, B)., attack(B, F)., attack(D, E).,\\ &attack(E, F)., attack(E, G)., attack(G, H)., attack(H, G).\}.\end{aligned}$$

Die zugehörige Literalmenge ist damit

$$\begin{aligned}\mathcal{F}_{\mathsf{AF}} = \{&arg(A), arg(B), arg(C), arg(D), arg(E), arg(F), arg(G), arg(H),\\ &attack(B, A), attack(B, C), attack(C, B), attack(B, F), attack(D, E),\\ &attack(E, F), attack(E, G), attack(G, H), attack(H, G)\}.\end{aligned}$$

Weiterhin ergibt sich die Literalmenge $IO(\{A, C, D, F\})$ zu

$$IO(\{A, C, D, F\} = \{in(A), out(B), in(C), in(D), out(E), in(F), out(G), out(H)\}.$$

Aus Beispiel 10.68 wissen wir, dass $\mathcal{E} = \{A, C, D, F\}$ eine vollständige Extension von AF ist mit $\mathcal{E}^+ = \{B, E\}$ und $(\mathcal{A} \backslash \mathcal{E}^+)^+ = (\mathcal{A} \backslash \{B, E\})^+ = \{A, C, D, F, G, H\}^+ = \{B, E, G, H\}$. Die zu $\{A, C, D, F\}$ gehörige Antwortmenge ist also

$$\begin{aligned}S = \mathcal{F}_{\mathsf{AF}} &\cup IO(\{A, C, D, F\})\\ &\cup \{defeated(B), defeated(E)\}\\ &\cup \{not_defended(B), not_defended(E), not_defended(G), not_defended(H)\}.\end{aligned}$$