Vorlesung “Formale Argumentation”

6. Komplexität von typischen Entscheidungsproblemen

Ringo Baumann
Professur für Formale Argumentation
und Logisches Schließen

16. Mai 2024
Leipzig

Entscheidungsprobleme

1. Credulous Acceptance (Cred_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Argument $a \in A$.
   Frage: Gilt $a \in \bigcup \sigma(F)$?

Entscheidungsprobleme

1. Credulous Acceptance (Cred_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Argument $a \in A$.
   Frage: Gilt $a \in \bigcup \sigma(F)$?

2. Skeptical Acceptance (Scept_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Argument $a \in A$.
   Frage: Gilt $a \in \bigcap \sigma(F)$?

Entscheidungsprobleme

1. Credulous Acceptance (Cred_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Argument $a \in A$.
   Frage: Gilt $a \in \bigcup \sigma(F)$?

2. Skeptical Acceptance (Scept_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Argument $a \in A$.
   Frage: Gilt $a \in \bigcap \sigma(F)$?

3. Verification (Ver_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Menge $E \subseteq A$.
   Frage: Gilt $E \in \sigma(F)$?

Entscheidungsprobleme

1. Credulous Acceptance (Cred_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Argument $a \in A$.
   Frage: Gilt $a \in \bigcup \sigma(F)$?

2. Skeptical Acceptance (Scept_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Argument $a \in A$.
   Frage: Gilt $a \in \bigcap \sigma(F)$?

3. Verification (Ver_σ)
   Gegeben: Ein AF $F = (A, R)$ und ein Menge $E \subseteq A$.
   Frage: Gilt $E \in \sigma(F)$?

4. Existence (Exists_σ)
   Gegeben: Ein AF $F = (A, R)$.
   Frage: Gilt $\sigma(F) \neq \emptyset$?

Komplexitätstheorie

... eine kurze Wiederholung.

Komplexitätstheorie beschäftigt sich mit

Komplexitätstheorie

…eine kurze Wiederholung.

Komplexitätstheorie beschäftigt sich mit

1. den Kosten, um ein Problem zu lösen.
2. dem Maß an benötigter Zeit (Anzahl Schritte) und Platzverbrauch.

Komplexitätstheorie

…eine kurze Wiederholung.

Komplexitätstheorie beschäftigt sich mit

1. den Kosten, um ein Problem zu lösen.
2. dem Maß an benötigter Zeit (Anzahl Schritte) und Platzverbrauch.
3. dem worst case Verhalten (schwierigste Instanzen).
4. dem asymptotischen Verhalten (große Instanzen).

Komplexitätstheorie

. . . eine kurze Wiederholung.

Komplexitätstheorie beschäftigt sich mit

1. den Kosten, um ein Problem zu lösen.
2. dem Maß an benötigter Zeit (Anzahl Schritte) und Platzverbrauch.
3. dem worst case Verhalten (schwierigste Instanzen).
4. dem asymptotischen Verhalten (große Instanzen).

Probleme können unterschiedlich schwer sein. Wie kann man das herausfinden?

Reduktionen

Wir wollen zeigen: Problem A ist höchstens so schwer wie Problem B. In Zeichen: $A \leq B$. Zeige dafür:

Reduktionen

Wir wollen zeigen: Problem A ist höchstens so schwer wie Problem B. In Zeichen: $A \leq B$. Zeige dafür:

Es gibt eine Reduktion R von A auf B m.d.E.

1. I ist Ja-Instanz von A gdw. $R(I)$ ist Ja-Instanz von B

Reduktionen

Wir wollen zeigen: Problem A ist höchstens so schwer wie Problem B. In Zeichen: $A \leq B$. Zeige dafür:

Es gibt eine Reduktion R von A auf B m.d.E.

1. I ist Ja-Instanz von A gdw. $R(I)$ ist Ja-Instanz von B
2. R ist effizient berechenbar (hier: in Polynomialzeit)

Intuitiv: Lösungsalgo für B ergibt Lösungsalgo für A

Reduktionen

Wir wollen zeigen: Problem A ist höchstens so schwer wie Problem B. In Zeichen: $A \leq B$. Zeige dafür:

Es gibt eine Reduktion R von A auf B m.d.E.

1. I ist Ja-Instanz von A gdw. $R(I)$ ist Ja-Instanz von B
2. R ist effizient berechenbar (hier: in Polynomialzeit)

Intuitiv: Lösungsalgo für B ergibt Lösungsalgo für A

Beispiel:

- A: Folgerungsproblem für endliches T, d.h.: Gilt $T \models \phi$?

Reduktionen

Wir wollen zeigen: Problem A ist höchstens so schwer wie Problem B. In Zeichen: $A \leq B$. Zeige dafür:

Es gibt eine Reduktion R von A auf B m.d.E.

1. I ist Ja-Instanz von A gdw. $R(I)$ ist Ja-Instanz von B
2. R ist effizient berechenbar (hier: in Polynomialzeit)

Intuitiv: Lösungsalgo für B ergibt Lösungsalgo für A

Beispiel:

- A: Folgerungsproblem für endliches T, d.h.: Gilt $T \models \phi$?
- B: Unerfüllbarkeitsproblem (UNSAT), d.h.: Gilt $Mod(\psi) = \varnothing$?

Reduktionen

Wir wollen zeigen: Problem A ist höchstens so schwer wie Problem B. In Zeichen: $A \leq B$. Zeige dafür:

Es gibt eine Reduktion R von A auf B m.d.E.

1. I ist Ja-Instanz von A gdw. $R(I)$ ist Ja-Instanz von B
2. R ist effizient berechenbar (hier: in Polynomialzeit)

Intuitiv: Lösungsalgo für B ergibt Lösungsalgo für A

Beispiel:

- A: Folgerungsproblem für endliches T, d.h.: Gilt $T \models \phi$?
- B: Unerfüllbarkeitsproblem (UNSAT), d.h.: Gilt $Mod(\psi) = \varnothing$?
- $R : (2^{\mathcal{F}})_{\text{fin}} \times \mathcal{F} \to \mathcal{F}$ mit $(T, \phi) \mapsto R(T, \phi) = \bigwedge T \land \neg\phi$

Heisenberg-Programm DFG ScaDS.AI DRESDEN LEIPZIG UNIVERSITÄT LEIPZIG

Reduktionen

Wir wollen zeigen: Problem A ist höchstens so schwer wie Problem B. In Zeichen: $A \leq B$. Zeige dafür:

Es gibt eine Reduktion R von A auf B m.d.E.

1. I ist Ja-Instanz von A gdw. $R(I)$ ist Ja-Instanz von B
2. R ist effizient berechenbar (hier: in Polynomialzeit)

Intuitiv: Lösungsalgo für B ergibt Lösungsalgo für A

Beispiel:

- A: Folgerungsproblem für endliches T, d.h.: Gilt $T \vDash \phi$?
- B: Unerfüllbarkeitsproblem (UNSAT), d.h.: Gilt $Mod(\psi) = \varnothing$?
- $R : (2^{\mathcal{F}})_{\text{fin}} \times \mathcal{F} \to \mathcal{F}$ mit $(T, \phi) \mapsto R(T, \phi) = \bigwedge T \wedge \neg\phi$
  1. Warum?
  2. Klar.

Komplexitätsklassen (allgemein)

- Komplexitätsklasse $\mathcal{C}$ ist eine Menge von Problemen, welche mit vorgegebenen Resourcen lösbar sind

Komplexitätsklassen (allgemein)

- Komplexitätsklasse $\mathcal{C}$ ist eine Menge von Problemen, welche mit vorgegebenen Resourcen lösbar sind
- Problem B ist $\mathcal{C}$-schwer, sofern für jedes Problem $A \in \mathcal{C}$ gilt: $A \leq B$ (I.A. hängt die Art der Reduktion von $\mathcal{C}$ ab.)

Komplexitätsklassen (allgemein)

- Komplexitätsklasse $\mathcal{C}$ ist eine Menge von Problemen, welche mit vorgegebenen Resourcen lösbar sind
- Problem B ist $\mathcal{C}$-schwer, sofern für jedes Problem $A \in \mathcal{C}$ gilt: $A \leq B$ (I.A. hängt die Art der Reduktion von $\mathcal{C}$ ab.)
- Problem B ist $\mathcal{C}$-vollständig, sofern B ist $\mathcal{C}$-schwer und $B \in \mathcal{C}$

Komplexitätsklassen (allgemein)

- Komplexitätsklasse $\mathcal{C}$ ist eine Menge von Problemen, welche mit vorgegebenen Resourcen lösbar sind
- Problem B ist $\mathcal{C}$-schwer, sofern für jedes Problem $A \in \mathcal{C}$ gilt: $A \leq B$ (I.A. hängt die Art der Reduktion von $\mathcal{C}$ ab.)
- Problem B ist $\mathcal{C}$-vollständig, sofern B ist $\mathcal{C}$-schwer und $B \in \mathcal{C}$

$\Rightarrow$ Falls A und B $\mathcal{C}$-vollständig, dann $A \leq B$ und $B \leq A$
(schwerste Probleme in $\mathcal{C}$)

Komplexitätsklassen (allgemein)

- Komplexitätsklasse $\mathcal{C}$ ist eine Menge von Problemen, welche mit vorgegebenen Resourcen lösbar sind
- Problem B ist $\mathcal{C}$-schwer, sofern für jedes Problem $A \in \mathcal{C}$ gilt: $A \leq B$ (I.A. hängt die Art der Reduktion von $\mathcal{C}$ ab.)
- Problem B ist $\mathcal{C}$-vollständig, sofern B ist $\mathcal{C}$-schwer und $B \in \mathcal{C}$

$\Rightarrow$ Falls A und B $\mathcal{C}$-vollständig, dann $A \leq B$ und $B \leq A$ (schwerste Probleme in $\mathcal{C}$)

$\Rightarrow$ Problem B ist $\mathcal{C}$-schwer, sofern für ein $\mathcal{C}$-vollständiges Problem A gilt: $A \leq B$ (nur eine Reduktion)

Komplexitätsklassen (speziell)

- P (Polynomial Time): Probleme, die auf deterministischer TM in Polynomialzeit lösbar sind
  P-vollständiges Problem:

Komplexitätsklassen (speziell)

- P (Polynomial Time): Probleme, die auf deterministischer TM in Polynomialzeit lösbar sind
  P-vollständiges Problem: HORNSAT, d.h. Erfüllbarkeit von aussagenlogischen Horn-Formeln

Komplexitätsklassen (speziell)

- P (Polynomial Time): Probleme, die auf deterministischer TM in Polynomialzeit lösbar sind
  P-vollständiges Problem: HORNSAT, d.h. Erfüllbarkeit von aussagenlogischen Horn-Formeln
- NP (Non-Deterministic Polynomial Time): Probleme, die auf nicht-deterministischer TM in Polynomialzeit lösbar sind (Ja-Instanz, falls es einen akzeptierenden Lauf gibt)
  NP-vollständiges Problem:

Komplexitätsklassen (speziell)

- P (Polynomial Time): Probleme, die auf deterministischer TM in Polynomialzeit lösbar sind
  P-vollständiges Problem: HORNSAT, d.h. Erfüllbarkeit von aussagenlogischen Horn-Formeln
- NP (Non-Deterministic Polynomial Time): Probleme, die auf nicht-deterministischer TM in Polynomialzeit lösbar sind (Ja-Instanz, falls es einen akzeptierenden Lauf gibt)
  NP-vollständiges Problem: SAT, d.h. Erfüllbarkeit von aussagenlogischen Formeln

Komplexitätsklassen (speziell)

- P (Polynomial Time): Probleme, die auf deterministischer TM in Polynomialzeit lösbar sind
  P-vollständiges Problem: HORNSAT, d.h. Erfüllbarkeit von aussagenlogischen Horn-Formeln
- NP (Non-Deterministic Polynomial Time): Probleme, die auf nicht-deterministischer TM in Polynomialzeit lösbar sind
  (Ja-Instanz, falls es einen akzeptierenden Lauf gibt)
  NP-vollständiges Problem: SAT, d.h. Erfüllbarkeit von aussagenlogischen Formeln
- coNP: Probleme, deren Komplemente in NP sind
  (Ja-Instanz, falls alle Läufe akzeptierend sind)
  coNP-vollständiges Problem:

Komplexitätsklassen (speziell)

- P (Polynomial Time): Probleme, die auf deterministischer TM in Polynomialzeit lösbar sind
  P-vollständiges Problem: HORNSAT, d.h. Erfüllbarkeit von aussagenlogischen Horn-Formeln
- NP (Non-Deterministic Polynomial Time): Probleme, die auf nicht-deterministischer TM in Polynomialzeit lösbar sind (Ja-Instanz, falls es einen akzeptierenden Lauf gibt)
  NP-vollständiges Problem: SAT, d.h. Erfüllbarkeit von aussagenlogischen Formeln
- coNP: Probleme, deren Komplemente in NP sind (Ja-Instanz, falls alle Läufe akzeptierend sind)
  coNP-vollständiges Problem: UNSAT, d.h. Unerfüllbarkeit von aussagenlogischen Formeln

Es gilt: $\mathrm{P} \subseteq \mathrm{NP}$ und $\mathrm{P} \subseteq \mathrm{coNP}$

Komplexitätsklassen (speziell)

- $\sum_2^p$: Probleme, die in NP sind, sofern man ein NP-Orakel verwenden darf
  $\sum_2^p$-vollständiges Problem:

Komplexitätsklassen (speziell)

- $\sum_2^p$: Probleme, die in NP sind, sofern man ein NP-Orakel verwenden darf
  $\sum_2^p$-vollständiges Problem: Gültigkeit einer QBF der Form $\exists Y \forall Z \phi(Y, Z)$

Komplexitätsklassen (speziell)

- $\sum_2^p$: Probleme, die in NP sind, sofern man ein NP-Orakel verwenden darf
  $\sum_2^p$-vollständiges Problem: Gültigkeit einer QBF der Form $\exists Y \forall Z \phi(Y, Z)$
- $\prod_2^p$: Probleme, die in coNP sind, sofern man ein NP-Orakel verwenden darf bzw. Probleme, deren Komplemente in $\sum_2^p$ sind
  $\prod_2^p$-vollständiges Problem:

Komplexitätsklassen (speziell)

- $\sum_2^p$: Probleme, die in NP sind, sofern man ein NP-Orakel verwenden darf
  $\sum_2^p$-vollständiges Problem: Gültigkeit einer QBF der Form $\exists Y \forall Z \phi(Y, Z)$
- $\prod_2^p$: Probleme, die in coNP sind, sofern man ein NP-Orakel verwenden darf bzw. Probleme, deren Komplemente in $\sum_2^p$ sind
  $\prod_2^p$-vollständiges Problem: Gültigkeit einer QBF der Form $\forall Y \exists Z \phi(Y, Z)$

Komplexitätsklassen (speziell)

- $\sum_2^p$: Probleme, die in NP sind, sofern man ein NP-Orakel verwenden darf
  $\sum_2^p$-vollständiges Problem: Gültigkeit einer QBF der Form $\exists Y \forall Z \phi(Y, Z)$
- $\prod_2^p$: Probleme, die in coNP sind, sofern man ein NP-Orakel verwenden darf bzw. Probleme, deren Komplemente in $\sum_2^p$ sind
  $\prod_2^p$-vollständiges Problem: Gültigkeit einer QBF der Form $\forall Y \exists Z \phi(Y, Z)$

Es gilt: $\text{NP}, \text{coNP} \subseteq \sum_2^p$ und $\text{NP}, \text{coNP} \subseteq \prod_2^p$

Entscheidungsprobleme für Conflict-free Sets

1. Cred_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup cf(F)$?

Entscheidungsprobleme für Conflict-free Sets

1. Cred_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup cf(F)$?
   $\Rightarrow$ Teste ob $(a, a) \in R$. Also, $\mathrm{Cred}_{cf} \in \mathrm{P}$.

Entscheidungsprobleme für Conflict-free Sets

1. Cred_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup cf(F)$?
   $\Rightarrow$ Teste ob $(a, a) \in R$. Also, $\text{Cred}_{cf} \in \text{P}$.
2. Scept_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap cf(F)$?

Entscheidungsprobleme für Conflict-free Sets

1. Cred_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup cf(F)$?
   $\Rightarrow$ Teste ob $(a, a) \in R$. Also, $\mathrm{Cred}_{cf} \in \mathrm{P}$.

2. Scept_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap cf(F)$?
   $\Rightarrow$ Trivial, da $\varnothing \in cf(F)$ und somit $\bigcap cf(F) = \varnothing$. Antwort ist also immer "Nein", egal welcher Input.

Entscheidungsprobleme für Conflict-free Sets

1. Cred_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup cf(F)$?
   $\Rightarrow$ Teste ob $(a, a) \in R$. Also, $\text{Cred}_{cf} \in \text{P}$.
2. Scept_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap cf(F)$?
   $\Rightarrow$ Trivial, da $\varnothing \in cf(F)$ und somit $\bigcap cf(F) = \varnothing$. Antwort ist also immer "Nein", egal welcher Input.
3. Ver_{cf}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in cf(F)$?

Entscheidungsprobleme für Conflict-free Sets

1. Cred_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup cf(F)$?
   $\Rightarrow$ Teste ob $(a, a) \in R$. Also, $\text{Cred}_{cf} \in \text{P}$.
2. Scept_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap cf(F)$?
   $\Rightarrow$ Trivial, da $\emptyset \in cf(F)$ und somit $\bigcap cf(F) = \emptyset$. Antwort ist also immer "Nein", egal welcher Input.
3. Ver_{cf}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in cf(F)$?
   $\Rightarrow$ Teste ob $(a, b) \in R$ mit $a, b \in E$. Also, $\text{Ver}_{cf} \in \text{P}$.
4. Exists_{cf}: Gegeben $F = (A, R)$. Gilt $cf(F) \neq \emptyset$?

Entscheidungsprobleme für Conflict-free Sets

1. Cred_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup cf(F)$?
   $\Rightarrow$ Teste ob $(a, a) \in R$. Also, $\text{Cred}_{cf} \in \mathrm{P}$.
2. Scept_{cf}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap cf(F)$?
   $\Rightarrow$ Trivial, da $\varnothing \in cf(F)$ und somit $\bigcap cf(F) = \varnothing$. Antwort ist also immer "Nein", egal welcher Input.
3. Ver_{cf}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in cf(F)$?
   $\Rightarrow$ Teste ob $(a, b) \in R$ mit $a, b \in E$. Also, $\text{Ver}_{cf} \in \mathrm{P}$.
4. Exists_{cf}: Gegeben $F = (A, R)$. Gilt $cf(F) \neq \varnothing$?
   $\Rightarrow$ Trivial, da $\varnothing \in cf(F)$ und somit $cf(F) \neq \varnothing$. Antwort ist also immer "Ja", egal welcher Input.

Entscheidungsprobleme für Admissible Sets

1. Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
3. Ver_{ad}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in ad(F)$?
4. Exists_{ad}: Gegeben $F = (A, R)$. Gilt $ad(F) \neq \varnothing$?

Entscheidungsprobleme für Admissible Sets

1. Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
3. Ver_{ad}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in ad(F)$?
4. Exists_{ad}: Gegeben $F = (A, R)$. Gilt $ad(F) \neq \emptyset$?

Welche Probleme sind trivial bzw. auf jeden Fall in P?

Entscheidungsprobleme für Admissible Sets

1. Cred$_{ad}$: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept$_{ad}$: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
   $\Rightarrow$ Trivial, da $\emptyset \in ad(F)$ und somit $\bigcap ad(F) = \emptyset$. Antwort ist also immer "Nein", egal welcher Input.

Entscheidungsprobleme für Admissible Sets

1. Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
   $\Rightarrow$ Trivial, da $\varnothing \in ad(F)$ und somit $\bigcap ad(F) = \varnothing$. Antwort ist also immer “Nein”, egal welcher Input.
3. Ver_{ad}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in ad(F)$?

Entscheidungsprobleme für Admissible Sets

1. Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
   $\Rightarrow$ Trivial, da $\varnothing \in ad(F)$ und somit $\bigcap ad(F) = \varnothing$. Antwort ist also immer "Nein", egal welcher Input.
3. Ver_{ad}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in ad(F)$?
   $\Rightarrow$ 1. Konfliktfreiheit: Teste ob $E \in cf(F)$.

Entscheidungsprobleme für Admissible Sets

1. Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
   $\Rightarrow$ Trivial, da $\varnothing \in ad(F)$ und somit $\bigcap ad(F) = \varnothing$. Antwort ist also immer "Nein", egal welcher Input.
3. Ver_{ad}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in ad(F)$?
   $\Rightarrow$ 1. Konfliktfreiheit: Teste ob $E \in cf(F)$. 2. Verteidigung: Überprüfe für jedes $a \in E$: Wenn $(b, a) \in R$, dann existiert ein $(c, b) \in R$ mit $c \in E$. Also, $\text{Ver}_{ad} \in \text{P}$.

Entscheidungsprobleme für Admissible Sets

1. Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
   $\Rightarrow$ Trivial, da $\varnothing \in ad(F)$ und somit $\bigcap ad(F) = \varnothing$. Antwort ist also immer "Nein", egal welcher Input.
3. Ver_{ad}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in ad(F)$?
   $\Rightarrow$ 1. Konfliktfreiheit: Teste ob $E \in cf(F)$. 2. Verteidigung: Überprüfe für jedes $a \in E$: Wenn $(b, a) \in R$, dann existiert ein $(c, b) \in R$ mit $c \in E$. Also, $\text{Ver}_{ad} \in \text{P}$.
4. Exists_{ad}: Gegeben $F = (A, R)$. Gilt $ad(F) \neq \varnothing$?

Entscheidungsprobleme für Admissible Sets

1. Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?
2. Scept_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap ad(F)$?
   $\Rightarrow$ Trivial, da $\emptyset \in ad(F)$ und somit $\bigcap ad(F) = \emptyset$. Antwort ist also immer "Nein", egal welcher Input.
3. Ver_{ad}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in ad(F)$?
   $\Rightarrow$ 1. Konfliktfreiheit: Teste ob $E \in cf(F)$. 2. Verteidigung: Überprüfe für jedes $a \in E$: Wenn $(b, a) \in R$, dann existiert ein $(c, b) \in R$ mit $c \in E$. Also, $\text{Ver}_{ad} \in \text{P}$.
4. Exists_{ad}: Gegeben $F = (A, R)$. Gilt $ad(F) \neq \emptyset$?
   $\Rightarrow$ Trivial, da $\emptyset \in ad(F)$ und somit $ad(F) \neq \emptyset$. Antwort ist also immer "Ja", egal welcher Input.

Was ist mit Cred_{ad}?

Komplexität von Cred_{ad}

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

1. $\text{Cred}_{ad} \in \text{NP}$.
   Benutze dafür folgenden guess-and-check Algorithmus:

Komplexität von Cred_{ad}

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

1. $\mathrm{Cred}_{ad} \in \mathrm{NP}$.
   Benutze dafür folgenden guess-and-check Algorithmus:
   - Rate $E \subseteq A \setminus \{a\}$
   - Überprüfe ob $E \cup \{a\} \in ad(F)$ $\qquad$ ($\mathrm{Ver}_{ad} \in \mathrm{P}$)

Komplexität von Cred_{ad}

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

1. $\text{Cred}_{ad} \in \text{NP}$.
   Benutze dafür folgenden guess-and-check Algorithmus:
   - Rate $E \subseteq A \setminus \{a\}$
   - Überprüfe ob $E \cup \{a\} \in ad(F)$ $\qquad$ ($\text{Ver}_{ad} \in \text{P}$)
2. Cred_{ad} ist NP-schwer. Wir zeigen $3\text{-SAT} \leq \text{Cred}_{ad}$.

Komplexität von Cred_{ad}

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

1. $\text{Cred}_{ad} \in \text{NP}$.
   Benutze dafür folgenden guess-and-check Algorithmus:
   - Rate $E \subseteq A \setminus \{a\}$
   - Überprüfe ob $E \cup \{a\} \in ad(F)$ $\qquad$ ($\text{Ver}_{ad} \in \text{P}$)
2. Cred_{ad} ist NP-schwer. Wir zeigen 3-SAT $\leq \text{Cred}_{ad}$.
   - 3-SAT ist Erfüllbarkeit von 3-KNF
   - 3-SAT ist auch NP-vollständig

Komplexität von Cred_{ad}

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

1. $\text{Cred}_{ad} \in \text{NP}$.
   Benutze dafür folgenden guess-and-check Algorithmus:
   - Rate $E \subseteq A \setminus \{a\}$
   - Überprüfe ob $E \cup \{a\} \in ad(F)$ ($\text{Ver}_{ad} \in \text{P}$)
2. Cred_{ad} ist NP-schwer. Wir zeigen 3-SAT $\leq \text{Cred}_{ad}$.
   - 3-SAT ist Erfüllbarkeit von 3-KNF
   - 3-SAT ist auch NP-vollständig

   Wir geben eine Reduktion R an, die eine 3-KNF ϕ auf ein AF $F_\phi = (A_\phi, R_\phi)$ abbildet, so daß:
   - F_ϕ ist effizient berechenbar

Komplexität von Cred_{ad}

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

1. $\text{Cred}_{ad} \in \text{NP}$.
   Benutze dafür folgenden guess-and-check Algorithmus:
   - Rate $E \subseteq A \setminus \{a\}$
   - Überprüfe ob $E \cup \{a\} \in ad(F)$ $\quad$ ($\text{Ver}_{ad} \in \text{P}$)
2. Cred_{ad} ist NP-schwer. Wir zeigen 3-SAT $\leq \text{Cred}_{ad}$.
   - 3-SAT ist Erfüllbarkeit von 3-KNF
   - 3-SAT ist auch NP-vollständig

   Wir geben eine Reduktion R an, die eine 3-KNF ϕ auf ein AF $F_\phi = (A_\phi, R_\phi)$ abbildet, so daß:

   - F_ϕ ist effizient berechenbar
   - Für ein spezielles Argument $a_\phi \in A_\phi$ gilt:

$$\phi \text{ ist erfüllbar} \quad \text{gdw.} \quad a_\phi \in \bigcup ad(F_\phi)$$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Beispiel: $\phi = (a_1 \vee a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3 \vee \neg a_4) \wedge (\neg a_1 \vee a_2 \vee a_4)$

- $\sigma(\phi) = \{a_1, a_2, a_3, a_4\}$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Beispiel: $\phi = (a_1 \vee a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3 \vee \neg a_4) \wedge (\neg a_1 \vee a_2 \vee a_4)$

- $\sigma(\phi) = \{a_1, a_2, a_3, a_4\}$
- $\phi = C_1 \wedge C_2 \wedge C_3$

Standardkonstruktion

Cred$_{ad}$: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \land C_2 \land \ldots \land C_m$ (m Klauseln)
- $C_i = l_{i1} \lor l_{i2} \lor l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Beispiel: $\phi = (a_1 \lor a_2 \lor a_3) \land (\neg a_2 \lor \neg a_3 \lor \neg a_4) \land (\neg a_1 \lor a_2 \lor a_4)$

- $\sigma(\phi) = \{a_1, a_2, a_3, a_4\}$
- $\phi = C_1 \land C_2 \land C_3$
- $l_{23} = \neg a_4$ (neg. Literal), $l_{32} = a_2$ (pos. Literal), …

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup \{a_\phi\}$

Beispiel: $\phi = (a_1 \vee a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3 \vee \neg a_4) \wedge (\neg a_1 \vee a_2 \vee a_4)$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup \{a_\phi\}$

Beispiel: $\phi = (a_1 \vee a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3 \vee \neg a_4) \wedge (\neg a_1 \vee a_2 \vee a_4)$

- $A_\phi = \{a_1, a_2, a_3, a_4\} \cup \{\overline{a_1}, \overline{a_2}, \overline{a_3}, \overline{a_4}\} \cup \{c_1, c_2, c_3\} \cup \{a_\phi\}$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup \{a_\phi\}$
- $R_\phi = \{(a, \overline{a}), (\overline{a}, a) \mid a \in \sigma(\phi)\} \cup$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup \{a_\phi\}$
- $R_\phi = \{(a, \overline{a}), (\overline{a}, a) \mid a \in \sigma(\phi)\} \cup \{(c_i, a_\phi) \mid 1 \leq i \leq m\}$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup \{a_\phi\}$
- $R_\phi = \{(a, \overline{a}), (\overline{a}, a) \mid a \in \sigma(\phi)\} \cup \{(c_i, a_\phi) \mid 1 \leq i \leq m\}$ $\cup\{(l, c_i) \mid l \in \{l_{i1}, l_{i2}, l_{i3}\}$ pos. Literal, $i \in \{1, \ldots, m\}\}$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup \{a_\phi\}$
- $R_\phi = \{(a, \overline{a}), (\overline{a}, a) \mid a \in \sigma(\phi)\} \cup \{(c_i, a_\phi) \mid 1 \leq i \leq m\}$
  $\cup\{(l, c_i) \mid l \in \{l_{i1}, l_{i2}, l_{i3}\}$ pos. Literal, $i \in \{1, \ldots, m\}\}$
  $\cup\{(\overline{l}, c_i) \mid l \in \{l_{i1}, l_{i2}, l_{i3}\}$ neg. Literal, $i \in \{1, \ldots, m\}\}$

Standardkonstruktion

Cred_{ad}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup ad(F)$?

Definition (Standardreduktion)

Gegeben eine 3-KNF ϕ mit Signatur $\sigma(\phi) = \{a_1, a_2, \ldots, a_n\}$ und

- $\phi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ (m Klauseln)
- $C_i = l_{i1} \vee l_{i2} \vee l_{i3}$ für jedes $1 \leq i \leq m$ (3-Klausel)

Definiere ein AF $F_\phi = (A_\phi, R_\phi)$ mit

- $A_\phi = \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\} \cup \{c_1, \ldots, c_m\} \cup \{a_\phi\}$
- $R_\phi = \{(a, \overline{a}), (\overline{a}, a) \mid a \in \sigma(\phi)\} \cup \{(c_i, a_\phi) \mid 1 \leq i \leq m\}$
  $\cup \{(l, c_i) \mid l \in \{l_{i1}, l_{i2}, l_{i3}\} \text{ pos. Literal}, i \in \{1, \ldots, m\}\}$
  $\cup \{(\overline{l}, c_i) \mid l \in \{l_{i1}, l_{i2}, l_{i3}\} \text{ neg. Literal}, i \in \{1, \ldots, m\}\}$

Beispiel: $\phi = (a_1 \vee a_2 \vee a_3) \wedge (\neg a_2 \vee \neg a_3 \vee \neg a_4) \wedge (\neg a_1 \vee a_2 \vee a_4)$

- $A_\phi = \{a_1, a_2, a_3, a_4\} \cup \{\overline{a_1}, \overline{a_2}, \overline{a_3}, \overline{a_4}\} \cup \{c_1, c_2, c_3\} \cup \{a_\phi\}$
- $R_\phi \quad \Rightarrow$ siehe Tafel

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Rightarrow$) Sei ϕ erfüllbar. Somit ex. Belegung $B \subseteq \sigma(\phi)$ mit $B(\phi) = 1$.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Rightarrow$) Sei ϕ erfüllbar. Somit ex. Belegung $B \subseteq \sigma(\phi)$ mit $B(\phi) = 1$. Da Konjunktion folgt $B(C_i) = 1$ für $1 \leq i \leq m$. Da C_i Disjunktion ex. ein $j_i \in \{1, 2, 3\}$ mit $B(l_{ij_i}) = 1$.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \textit{ erfüllbar} \quad \textit{gdw.} \quad \textit{es ex. ein } E \in ad(F_\phi) \textit{ mit } a_\phi \in E$$

Proof.

($\Rightarrow$) Sei ϕ erfüllbar. Somit ex. Belegung $B \subseteq \sigma(\phi)$ mit $B(\phi) = 1$. Da Konjunktion folgt $B(C_i) = 1$ für $1 \leq i \leq m$. Da C_i Disjunktion ex. ein $j_i \in \{1, 2, 3\}$ mit $B(l_{ij_i}) = 1$. Sei $L = \{l_{1j_1}, \ldots, l_{mj_m}\}$ und $E_L = \{l \mid l \in L \text{ pos. Literal}\} \cup \{\bar{l} \mid l \in L \text{ neg. Literal}\}$.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \textit{ erfüllbar} \quad \textit{gdw.} \quad \textit{es ex. ein } E \in ad(F_\phi) \textit{ mit } a_\phi \in E$$

Proof.

($\Rightarrow$) Sei ϕ erfüllbar. Somit ex. Belegung $B \subseteq \sigma(\phi)$ mit $B(\phi) = 1$. Da Konjunktion folgt $B(C_i) = 1$ für $1 \leq i \leq m$. Da C_i Disjunktion ex. ein $j_i \in \{1, 2, 3\}$ mit $B(l_{ij_i}) = 1$. Sei $L = \{l_{1j_1}, \ldots, l_{mj_m}\}$ und $E_L = \{l \mid l \in L \text{ pos. Literal}\} \cup \{\bar{l} \mid l \in L \text{ neg. Literal}\}$. Per Konstruktion ist $E_L \in cf(F_\phi)$ (Komplementäre Literale nicht gleichzeitig wahr)

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Rightarrow$) Sei ϕ erfüllbar. Somit ex. Belegung $B \subseteq \sigma(\phi)$ mit $B(\phi) = 1$. Da Konjunktion folgt $B(C_i) = 1$ für $1 \leq i \leq m$. Da C_i Disjunktion ex. ein $j_i \in \{1, 2, 3\}$ mit $B(l_{ij_i}) = 1$. Sei $L = \{l_{1j_1}, \ldots, l_{mj_m}\}$ und $E_L = \{l \mid l \in L \text{ pos. Literal}\} \cup \{\bar{l} \mid l \in L \text{ neg. Literal}\}$. Per Konstruktion ist $E_L \in cf(F_\phi)$ (Komplementäre Literale nicht gleichzeitig wahr) und auch $E_L \in ad(F_\phi)$ (c_i's und a_ϕ attackieren nicht).

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Rightarrow$) Sei ϕ erfüllbar. Somit ex. Belegung $B \subseteq \sigma(\phi)$ mit $B(\phi) = 1$. Da Konjunktion folgt $B(C_i) = 1$ für $1 \leq i \leq m$. Da C_i Disjunktion ex. ein $j_i \in \{1, 2, 3\}$ mit $B(l_{ij_i}) = 1$. Sei $L = \{l_{1j_1}, \ldots, l_{mj_m}\}$ und $E_L = \{l \mid l \in L \text{ pos. Literal}\} \cup \{\bar{l} \mid l \in L \text{ neg. Literal}\}$. Per Konstruktion ist $E_L \in cf(F_\phi)$ (Komplementäre Literale nicht gleichzeitig wahr) und auch $E_L \in ad(F_\phi)$ ($c_i's$ und a_ϕ attackieren nicht). Da Literal l_{ij_i} Disjunkt in Klausel C_i sind alle Argumente c_i von einem Element aus E_L attackiert, d.h.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Rightarrow$) Sei ϕ erfüllbar. Somit ex. Belegung $B \subseteq \sigma(\phi)$ mit $B(\phi) = 1$. Da Konjunktion folgt $B(C_i) = 1$ für $1 \leq i \leq m$. Da C_i Disjunktion ex. ein $j_i \in \{1, 2, 3\}$ mit $B(l_{ij_i}) = 1$. Sei $L = \{l_{1j_1}, \ldots, l_{mj_m}\}$ und $E_L = \{l \mid l \in L \text{ pos. Literal}\} \cup \{\bar{l} \mid l \in L \text{ neg. Literal}\}$. Per Konstruktion ist $E_L \in cf(F_\phi)$ (Komplementäre Literale nicht gleichzeitig wahr) und auch $E_L \in ad(F_\phi)$ (c_i's und a_ϕ attackieren nicht). Da Literal l_{ij_i} Disjunkt in Klausel C_i sind alle Argumente c_i von einem Element aus E_L attackiert, d.h. E_L verteidigt a_ϕ in F_ϕ. Nach Fundamentallemma $E = E_L \cup \{a_\phi\}$ admissible in F_ϕ.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Leftarrow$) Sei $E \in ad(F_\phi)$ mit $a_\phi \in E$. Da a_ϕ von allen c_i attackiert muß $E \setminus \{a_\phi\}$ jedes Argument c_i gegenattackieren.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Leftarrow$) Sei $E \in ad(F_\phi)$ mit $a_\phi \in E$. Da a_ϕ von allen c_i attackiert muß $E \setminus \{a_\phi\}$ jedes Argument c_i gegenattackieren. Genauer, für jedes c_i ex. ein Argument $l_i \in E \setminus \{a_\phi\}$ mit $(l_i, c_i) \in R_\phi$.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \textit{ erfüllbar} \quad \textit{gdw.} \quad \textit{es ex. ein } E \in ad(F_\phi) \textit{ mit } a_\phi \in E$$

Proof.

($\Leftarrow$) Sei $E \in ad(F_\phi)$ mit $a_\phi \in E$. Da a_ϕ von allen c_i attackiert muß $E \setminus \{a_\phi\}$ jedes Argument c_i gegenattackieren. Genauer, für jedes c_i ex. ein Argument $l_i \in E \setminus \{a_\phi\}$ mit $(l_i, c_i) \in R_\phi$. Per Konstruktion gilt $l_i \in \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\}$.

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Leftarrow$) Sei $E \in ad(F_\phi)$ mit $a_\phi \in E$. Da a_ϕ von allen c_i attackiert muß $E \setminus \{a_\phi\}$ jedes Argument c_i gegenattackieren. Genauer, für jedes c_i ex. ein Argument $l_i \in E \setminus \{a_\phi\}$ mit $(l_i, c_i) \in R_\phi$. Per Konstruktion gilt $l_i \in \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\}$. Betrachte $B = E \cap \sigma(\phi)$. Offensichtlich gilt $B(C_i) = 1$

Standardkonstruktion

Theorem

Gegeben eine 3-KNF ϕ. Es gilt:

$$\phi \text{ erfüllbar} \quad \text{gdw.} \quad \text{es ex. ein } E \in ad(F_\phi) \text{ mit } a_\phi \in E$$

Proof.

($\Leftarrow$) Sei $E \in ad(F_\phi)$ mit $a_\phi \in E$. Da a_ϕ von allen c_i attackiert muß $E \setminus \{a_\phi\}$ jedes Argument c_i gegenattackieren. Genauer, für jedes c_i ex. ein Argument $l_i \in E \setminus \{a_\phi\}$ mit $(l_i, c_i) \in R_\phi$. Per Konstruktion gilt $l_i \in \sigma(\phi) \cup \{\overline{a} \mid a \in \sigma(\phi)\}$. Betrachte $B = E \cap \sigma(\phi)$. Offensichtlich gilt $B(C_i) = 1$ da mindestens 1 Disjunkt auf wahr gesetzt wird. Somit $B(\phi) = 1$ was die Erfüllbarkeit von ϕ bezeugt.

$\square$

Entscheidungsprobleme für Preferred Semantics

1. Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?
2. Scept_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap pr(F)$?
3. Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?
4. Exists_{pr}: Gegeben $F = (A, R)$. Gilt $pr(F) \neq \emptyset$?

Welche Probleme sind trivial bzw. auf jeden Fall in P?

Entscheidungsprobleme für Preferred Semantics

1. Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?
2. Scept_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap pr(F)$?
3. Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?
4. Exists_{pr}: Gegeben $F = (A, R)$. Gilt $pr(F) \neq \emptyset$?

Welche Probleme sind trivial bzw. auf jeden Fall in P?

$\Rightarrow$ Exists_{pr} ist trivial, da $pr(F) \neq \emptyset$ für jedes F (VL3)

Komplexität von Ver_{pr}

Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\mathrm{Ver}_{pr}}$, d.h.

$$\overline{\mathrm{Ver}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } E \subseteq A. \text{ Gilt } E \notin pr(F)?$$

Komplexität von Ver_{pr}

Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\mathrm{Ver}_{pr}}$, d.h.

$$\overline{\mathrm{Ver}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } E \subseteq A. \text{ Gilt } E \notin pr(F)?$$

- Benutze dafür folgenden guess-and-check Algorithmus:

Komplexität von Ver_{pr}

Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\text{Ver}_{pr}}$, d.h.

  $$\overline{\text{Ver}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } E \subseteq A. \text{ Gilt } E \notin pr(F)?$$

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Überprüfe ob $E \in ad(F)$ $\qquad$ ($\text{Ver}_{ad} \in \text{P}$)

Komplexität von Ver_{pr}

Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\text{Ver}_{pr}}$, d.h.

$$\overline{\text{Ver}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } E \subseteq A. \text{ Gilt } E \notin pr(F)?$$

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Überprüfe ob $E \in ad(F)$ — ($\text{Ver}_{ad} \in \text{P}$)
  - Rate nun E' mit $A \supseteq E' \supset E$ — (Echte Obermenge)
  - Überprüfe ob $E' \in ad(F)$ — ($\text{Ver}_{ad} \in \text{P}$)

Komplexität von Ver_{pr}

Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\text{Ver}_{pr}}$, d.h.

$$\overline{\text{Ver}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } E \subseteq A. \text{ Gilt } E \notin pr(F)?$$

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Überprüfe ob $E \in ad(F)$ ($\text{Ver}_{ad} \in \text{P}$)
  - Rate nun E' mit $A \supseteq E' \supset E$ (Echte Obermenge)
  - Überprüfe ob $E' \in ad(F)$ ($\text{Ver}_{ad} \in \text{P}$)
- Somit $\overline{\text{Ver}_{pr}} \in \text{NP}$ und damit $\text{Ver}_{pr} \in \text{coNP}$

Komplexität von Ver_{pr}

Ver_{pr}: Gegeben $F = (A, R)$ und $E \subseteq A$. Gilt $E \in pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\text{Ver}_{pr}}$, d.h.

$$\overline{\text{Ver}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } E \subseteq A. \text{ Gilt } E \notin pr(F)?$$

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Überprüfe ob $E \in ad(F)$ ($\text{Ver}_{ad} \in \text{P}$)
  - Rate nun E' mit $A \supseteq E' \supset E$ (Echte Obermenge)
  - Überprüfe ob $E' \in ad(F)$ ($\text{Ver}_{ad} \in \text{P}$)
- Somit $\overline{\text{Ver}_{pr}} \in \text{NP}$ und damit $\text{Ver}_{pr} \in \text{coNP}$
- Es gilt sogar Ver_{pr} ist coNP-schwer und somit coNP-vollständig

Komplexität von Scept_{pr} und Cred_{pr}

Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:

Komplexität von Scept_{pr} und Cred_{pr}

Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \cup \{a\} \in pr(F)$ ($\text{Ver}_{pr} \in \text{coNP}$)

Komplexität von Scept_{pr} und Cred_{pr}

Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \cup \{a\} \in pr(F)$ $\qquad$ ($\mathrm{Ver}_{pr} \in \mathrm{coNP}$)
- Somit $\mathrm{Cred}_{pr} \in \mathrm{NP}^{\mathrm{coNP}} = \mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$

Komplexität von Scept_{pr} und Cred_{pr}

Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \cup \{a\} \in pr(F)$ ($\text{Ver}_{pr} \in \text{coNP}$)
- Somit $\text{Cred}_{pr} \in \text{NP}^{\text{coNP}} = \text{NP}^{\text{NP}} = \Sigma_2^p$

Geht es auch besser? (siehe Übung 3)

Komplexität von Scept_{pr} und Cred_{pr}

Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \cup \{a\} \in pr(F)$ ($\mathrm{Ver}_{pr} \in \mathrm{coNP}$)
- Somit $\mathrm{Cred}_{pr} \in \mathrm{NP}^{\mathrm{coNP}} = \mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$

Geht es auch besser? (siehe Übung 3)

Scept_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\mathrm{Scept}_{pr}}$, d.h.

$$\overline{\mathrm{Scept}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } a \in A. \text{ Gilt } a \notin \bigcap pr(F)?$$

Komplexität von Scept_{pr} und Cred_{pr}

Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \cup \{a\} \in pr(F)$ ($\mathrm{Ver}_{pr} \in$ coNP)
- Somit $\mathrm{Cred}_{pr} \in \mathrm{NP}^{\mathrm{coNP}} = \mathrm{NP}^{\mathrm{NP}} = \Sigma_2^p$

Geht es auch besser? (siehe Übung 3)

Scept_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\mathrm{Scept}_{pr}}$, d.h.

$\overline{\mathrm{Scept}_{pr}}$: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \notin \bigcap pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \in pr(F)$ ($\mathrm{Ver}_{pr} \in$ coNP)

Komplexität von Scept_{pr} und Cred_{pr}

Cred_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcup pr(F)$?

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \cup \{a\} \in pr(F)$ ($\text{Ver}_{pr} \in \text{coNP}$)
- Somit $\text{Cred}_{pr} \in \text{NP}^{\text{coNP}} = \text{NP}^{\text{NP}} = \Sigma_2^p$

Geht es auch besser? (siehe Übung 3)

Scept_{pr}: Gegeben $F = (A, R)$ und $a \in A$. Gilt $a \in \bigcap pr(F)$?

- Wir betrachten das komplementäre Problem $\overline{\text{Scept}_{pr}}$, d.h.

$$\overline{\text{Scept}_{pr}}: \text{ Gegeben } F = (A, R) \text{ und } a \in A. \text{ Gilt } a \notin \bigcap pr(F)?$$

- Benutze dafür folgenden guess-and-check Algorithmus:
  - Rate $E \subseteq A \setminus \{a\}$
  - Überprüfe ob $E \in pr(F)$ ($\text{Ver}_{pr} \in \text{coNP}$)
- Somit $\overline{\text{Scept}_{pr}} \in \text{NP}^{\text{coNP}} = \text{NP}^{\text{NP}} = \Sigma_2^p$. Also $\text{Scept}_{pr} \in \Pi_2^p$.

Vorlesung “Formale Argumentation”

6. Komplexität von typischen Entscheidungsproblemen

Ringo Baumann
Professur für Formale Argumentation
und Logisches Schließen

16. Mai 2024
Leipzig