

Prinzipien der Wissensrepräsentation

Prof. Dr. B. Nebel, Dr. S. Wölfl, Dr. M. Ragni
R. Mattmüller
Sommersemester 2010

Universität Freiburg
Institut für Informatik

Projekt P2

Abgabe: Montag, 5. Juli 2010

Abgabe per E-Mail an `mattmuel@informatik.uni-freiburg.de`

Dieses Projekt befasst sich mit ε -Konsequenzen einer Menge von plausiblen Konsequenzen.

Hinweis: Bei den Projekten geht es um die Implementierung, nicht so sehr um theoretische Aspekte. Wenn Sie an bestimmten Stellen nicht genau wissen, wie die Algorithmen und Definitionen aus der Vorlesung in die Praxis umgesetzt werden, können Sie daher gerne Fragen stellen – entweder per E-Mail oder in der Übungsgruppe.

Projekte können in C, C++, Java oder Python bearbeitet werden. Andere Programmiersprachen sind nach Absprache möglich; in diesem Fall bitte vor Bearbeitung des Projekts bei uns melden.

Die eingereichten Programme müssen die geforderten Ein- und Ausgabeformate verwenden, einige Tests bestehen und **ausreichend kommentiert** sein. Programme, die diesen Anforderungen nicht genügen, werden nicht akzeptiert, aber es besteht die Möglichkeit, innerhalb der Abgabefrist nachzubessern. Daher bitten wir darum, frühzeitig abzugeben, um ausreichend Zeit für Nachbesserungen zu haben.

Zur Erinnerung:

Sei K eine Menge von plausiblen Konsequenzen über einer Menge von Aussagevariablen V . Für $\delta > 0$ bezeichne $\mathcal{P}_\delta^K$ die Menge aller Wahrscheinlichkeitsverteilungen über den Variablenbelegungen von V , unter denen $P(\beta'|\alpha') \geq 1 - \delta$ für alle plausiblen Konsequenzen $\alpha' \sim \beta' \in K$ gilt. (Dabei sei $P(\beta'|\alpha') := 1$, falls $P(\alpha') = 0$ gilt.)

Eine plausible Konsequenz $\alpha \sim \beta$ ist eine ε -Konsequenz von K (in Symbolen: $K \models_\varepsilon \alpha \sim \beta$) wenn gilt:

Für alle $\varepsilon > 0$ existiert ein $\delta > 0$, so dass für alle Wahrscheinlichkeitsverteilungen $P \in \mathcal{P}_\delta^K$ gilt: $P(\beta|\alpha) \geq 1 - \varepsilon$.

In diesem Projekt implementieren wir ein Programm, das für die Berechnung von ε -Konsequenzen verwendet werden kann. Anstelle einer vollständigen Implementation der Definition begnügen wir uns aber damit, für *feste* Werte ε und δ zu überprüfen, ob $P(\beta|\alpha) \geq 1 - \varepsilon$ für alle Wahrscheinlichkeitsverteilungen $P \in \mathcal{P}_\delta^K$ gilt. Wir sagen in diesem Fall, dass $\alpha \sim \beta$ für die gegebenen Werte ε und δ eine ε - δ -Konsequenz von K ist.

Die Grundidee dabei ist folgende: Wir versuchen, einen Beweis durch Widerspruch zu führen, indem wir eine Wahrscheinlichkeitsverteilung $P \in \mathcal{P}_\delta^K$ mit der Eigenschaft $P(\beta|\alpha) < 1 - \varepsilon$ suchen. Dazu formalisieren wir die Eigenschaften, die eine solche Verteilung P haben muss, als lineares Programm (d. h. als

Menge von linearen Ungleichungen) und verwenden ein Lösungsverfahren für lineare Programme, um eine geeignete Verteilung P zu finden oder zu beweisen, dass keine existiert.

Sei $\mathcal{I}(V)$ die Menge aller Variablenbelegungen $\mathcal{I} : V \rightarrow \{\mathbf{T}, \mathbf{F}\}$. Wir repräsentieren eine Wahrscheinlichkeitsverteilung auf $\mathcal{I}(V)$ als Vektor von $2^{|V|}$ reellen Zahlen, wobei für alle $\mathcal{I} \in \mathcal{I}(V)$ die Zahl $p_{\mathcal{I}} \in [0, 1]$ die Wahrscheinlichkeit für die Belegung $\mathcal{I}$ bezeichnet. Damit wir tatsächlich eine Wahrscheinlichkeitsverteilung erhalten, muss gelten:

$$p_{\mathcal{I}} \geq 0 \quad \text{für alle } \mathcal{I} \in \mathcal{I}(V) \quad (1)$$

$$\sum_{\mathcal{I} \in \mathcal{I}(V)} p_{\mathcal{I}} = 1 \quad (2)$$

Als Lösungen kommen aber nicht beliebige Wahrscheinlichkeitsverteilungen in Frage, sondern nur solche, die zu P_{δ}^K gehören, d. h. bei denen für alle Aussagen $\alpha' \sim \beta' \in K$ gilt: $P(\beta'|\alpha') \geq 1 - \delta$, oder äquivalent: $P(\alpha' \wedge \beta') \geq (1 - \delta)P(\alpha')$. Als lineare Ungleichung formuliert erhalten wir also die Nebenbedingungen:

$$\sum_{\mathcal{I} \in \mathcal{I}(V), \mathcal{I} \models \alpha' \wedge \beta'} p_{\mathcal{I}} \geq \sum_{\mathcal{I} \in \mathcal{I}(V), \mathcal{I} \models \alpha'} (1 - \delta)p_{\mathcal{I}} \quad \text{für alle } \alpha' \sim \beta' \in K. \quad (3)$$

Entsprechend erhalten wir für die Bedingung $P(\beta|\alpha) < 1 - \varepsilon$ die Nebenbedingung:

$$\sum_{\mathcal{I} \in \mathcal{I}(V), \mathcal{I} \models \alpha \wedge \beta} p_{\mathcal{I}} < \sum_{\mathcal{I} \in \mathcal{I}(V), \mathcal{I} \models \alpha} (1 - \varepsilon)p_{\mathcal{I}}. \quad (4)$$

Es gilt also, dass das Ungleichungssystem (1)–(4) genau dann eine Lösung hat, wenn $\alpha \sim \beta$ keine ε - δ -Konsequenz von K ist.

Eine zu maximierende oder zu minimierende Zielfunktion gibt es hier nicht, da wir keine Präferenzen über die Gegenbeispiele haben. Weil ein lineares Programm immer eine Zielfunktion besitzen muss, können wir als triviale Zielfunktion etwa

$$\text{maximiere } 0. \quad (5)$$

verwenden.

Aufgabe P2.1 (ε - δ -Konsequenzen, 3 Punkte)

Implementieren Sie ein Programm zum Bestimmen von ε - δ -Konsequenzen. Verwenden Sie dabei das Programm `lp_solve`¹, um das zugrunde liegende lineare Programm zu lösen.

Eingabe

Ihr Programm soll wie folgt ausgeführt werden:²

```
# consequence <eps> <delta> <k> ... <k> <c>
```

Dabei sind `eps` und `delta` Fließkommazahlen in üblicher Computernotation (z. B. 0.01). Die restlichen Argumente sind plausible Konsequenzen der Form

¹Download unter <http://lpsolve.sourceforge.net> oder als Ubuntu-Paket `lp-solve`.

²Passen Sie die Kommandozeile entsprechend an, wenn Sie eine interpretierte Programmiersprache wie Java verwenden.

$\alpha \vdash \beta$, wobei die Argumente k die Menge K definieren und das Argument c die zu testende Konsequenz bezeichnet.

Plausible Konsequenzen werden in der Form $\alpha \Rightarrow \beta$ eingegeben, wobei α und β aussagenlogische Formeln sind, für die die Notation aus Projekt P1 verwendet wird.

Ausgabe

Die erste Zeile sollte je nach Ergebnis entweder „ ε - δ -consequence.“ oder „No ε - δ -consequence.“ lauten, wobei für ε und δ die entsprechenden Eingabewerte eingesetzt werden. Falls die Antwort nein lautet, sollten die nachfolgenden Zeilen eine Wahrscheinlichkeitsverteilung angeben, die ein Gegenbeispiel darstellt, wobei jeweils eine Belegung pro Zeile ausgegeben wird.

Beispiel

Da im System **P** die Kontrapositionsregel nicht gilt, kann man einfach Gegenbeispiele zur Behauptung $\{a \vdash b\} \models_{\varepsilon} \neg b \vdash \neg a$ finden. Für die Eingabe

```
# consequence 0.5 0.1 "a=>b" "~b=>~a"
```

könnte die Ausgabe beispielsweise lauten:

```
No 0.5-0.1-consequence.
p(a=F, b=F) = 0.0000
p(a=F, b=T) = 0.0000
p(a=T, b=F) = 0.0100
p(a=T, b=T) = 0.9900
```

Für die Eingabe

```
# consequence 0.5 0.1 "a=>b" "a=>b"
```

sollte die Ausgabe hingegen lauten:

```
0.5-0.1-consequence.
```

Achten Sie darauf, dass die echte Ungleichung (4) von Ihrem Programm bzw. `lp_solve` richtig, d. h. als echte Ungleichung $<$ und nicht als $\leq$, behandelt wird.