Prof. Dr. M. Grohe
E. Fluck, L. Härtel, T. Novotny

Übungsblatt 2 mit Lösungen

Abgabetermin: Montag, der 06. Mai 2024 um 14:30

Hausaufgabe 3 (Streichungsalgorithmus) **2+2=4 Punkte**

Wenden Sie den Streichungsalgorithmus aus der Vorlesung auf die folgende Formeln an:

a) $(\neg Q \vee \neg P \vee \neg T \vee R) \wedge (P) \wedge (S) \wedge (\neg R \vee \neg Q \vee \neg P) \wedge (T) \wedge (Q \vee \neg P \vee \neg T)$

b) $(R \vee \neg P \vee \neg S) \wedge (R) \wedge (S \vee \neg R \vee \neg Q) \wedge (P) \wedge (\neg R \vee P \vee \neg T) \wedge (\neg R \vee S \vee \neg P)$

Lösung:

Disclaimer: Die Lösung bezieht sich auf die aktualisierte Version von dem Algorithmus, die leider nicht rechtzeitig in Moodle verfügbar war. Die alte Version des Algorithmuses initialisiert die (partielle) Interpretation leer. Für Aufgabe (a) macht es keinen Unterschied in der Ausgabe des Algorithmuses, bei Aufgabe (b) geben wir die Unterschiede an. Wir berücksichtigen dies bei der Korrektur.

a) (1) Initialisierung auf

X	P	Q	R	S	T
$\mathfrak{A}(X)$	0	0	0	0	0

(2) Wähle (auf Zeile 8) die Klausel (P). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	1	0	0	0	0

und

$$\varphi = (\neg Q \vee \neg T \vee R) \wedge (S) \wedge (\neg R \vee \neg Q) \wedge (T) \wedge (Q \vee \neg T).$$

(3) Wähle (auf Zeile 8) die Klausel (S). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	1	0	0	1	0

und

$$\varphi = (\neg Q \vee \neg T \vee R) \wedge (\neg R \vee \neg Q) \wedge (T) \wedge (Q \vee \neg T).$$

(4) Wähle (auf Zeile 8) die Klausel (T). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	1	0	0	1	1

und

$$\varphi = (\neg Q \vee R) \wedge (\neg R \vee \neg Q) \wedge (Q).$$

(5) Wähle (auf Zeile 8) die Klausel (Q). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	1	1	0	1	1

und

$$\varphi = (R) \wedge (\neg R).$$

(6) Wähle (auf Zeile 8) die Klausel (R). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	1	1	1	1	1

und

$$\varphi = (\bot).$$

(7) Beim nächsten Durchlauf der Schleife gibt der Algorithmus (in Zeile 4) "unerfüllbar" zurück.

b) (1) Initialisierung auf

X	P	Q	R	S	T
$\mathfrak{A}(X)$	0	0	0	0	0

(2) Wähle (auf Zeile 8) die Klausel (R). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	0	0	1	0	0

und

$$\varphi = (S \vee \neg Q) \wedge (P) \wedge (P \vee \neg T) \wedge (S \vee \neg P).$$

(3) Wähle (auf Zeile 8) die Klausel (P). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	1	0	1	0	0

und

$$\varphi = (S \vee \neg Q) \wedge (S).$$

(4) Wähle (auf Zeile 8) die Klausel (S). Nach Ausführung der Zeilen 9-11 ergibt sich

X	P	Q	R	S	T
$\mathfrak{A}(X)$	1	0	1	1	0

und

$$\varphi = \top.$$

(5) Beim nächsten Durchlauf enthält jede Klausel (weil es keine Klauseln gibt) ein negatives Literal, also die Interpretation $\mathfrak{A}$ wird ausgegeben.

Da Q und T aus der Formel komplett gelöscht wurden, gibt die alte Version des Algorithmuses hier keine Belegung dieser Symbolen an. Jede Belegung der Symbole gibt eine erfüllende Interpretation.

Hausaufgabe 4 (Hornformel Äquivalenz) **2+4=6 Punkte**

(a) Sei σ eine Symbolmenge und seien $\mathfrak{A}, \mathfrak{B} : \sigma \to \{0,1\}$ zwei Interpretationen von Symbolen aus σ. Wir definieren die Interpretation $(\mathfrak{A} \cdot \mathfrak{B}) : \sigma \to \{0,1\}$ wie folgt:

$$(\mathfrak{A} \cdot \mathfrak{B})(P) = \begin{cases} 1 & \text{wenn } \mathfrak{A}(P) = \mathfrak{B}(P) = 1, \\ 0 & \text{sonst.} \end{cases}$$

Sei $\varphi \in \mathrm{AL}(\sigma)$ eine Hornformel und seien $\mathfrak{A}, \mathfrak{B}$ Modelle von φ. Beweisen Sie, dass $\mathfrak{A} \cdot \mathfrak{B}$ ein Modell von φ ist.

(b) Beweisen oder widerlegen Sie jeweils, dass die folgenden Formeln äquivalent zu einer Hornformel sind:

(i) $Q \to \Big((P \wedge R) \vee \big((\neg P \wedge Q) \to \neg R\big)\Big)$,

(ii) $Q \to \Big((P \wedge R) \vee \big((\neg P \wedge Q) \to R\big)\Big)$.

Lösung:

(a) Sei $\varphi = \bigwedge_{i=1}^{n} C_i$ eine Hornformel, das heißt, C_i enthält jeweils höchstens ein positives Literal. Sei $i \in [n]$ beliebig und seien $\mathfrak{A}, \mathfrak{B} : \sigma \to \{0,1\}$ zwei Modelle von φ, insbesondere auch Modelle von C_i. Falls C_i ein negatives Literal $\neg X$ enthält, sodass mindestens eine der beiden Interpretationen X auf 0 abbildet, dann gilt nach Definition $(\mathfrak{A} \cdot \mathfrak{B})(X) = 0$ und also $(\mathfrak{A} \cdot \mathfrak{B}) \models C_i$. Sonst gibt es in C_i positive Literale Y_1, Y_2 mit $\mathfrak{A}(Y_1) = \mathfrak{B}(Y_2) = 1$. Da in C_i höchstens ein positives Literal vorkommt, gilt $Y_1 = Y_2$ und aus der Definition folgt dass $(\mathfrak{A} \cdot \mathfrak{B})$ erfüllt C_i. Da i beliebig gewählt war, folgt $(\mathfrak{A} \cdot \mathfrak{B}) \models \varphi$.

(b) (i)

$$\begin{aligned} Q \to \Big((P \wedge R) \vee \big((\neg P \wedge Q) \to \neg R\big)\Big) \\ &\equiv \neg Q \vee (P \wedge R) \vee \big((\neg P \wedge Q) \to \neg R\big) \\ &\equiv \neg Q \vee (P \wedge R) \vee \big((P \vee \neg Q) \vee \neg R\big) \\ &\equiv (P \wedge R) \vee P \vee \neg Q \vee \neg R \\ &\equiv (P \vee P \vee \neg Q \vee \neg R) \wedge (R \vee P \vee \neg Q \vee \neg R) \\ &\equiv (P \vee \neg Q \vee \neg R) \wedge \top \\ &\equiv (P \vee \neg Q \vee \neg R). \end{aligned}$$

Dabei haben wir benutzt:

- Definition von $\to$
- Definition von $\to$ und De Morgansche Regeln

- Assoziativität und Idempotenz von $\vee$
- Distributive Regel
- $R \vee \neg R \equiv \top$ und Idempotenz von $\vee$
- $\varphi \wedge \top \equiv \varphi$

Die Formel ist also zu einer Hornformel äquivalent.

(ii) Seien $\mathfrak{A}, \mathfrak{B}$ zwei Interpretation definiert durch $\mathfrak{A}(P) = \mathfrak{A}(Q) = \mathfrak{B}(Q) = \mathfrak{B}(R) = 1$ und $\mathfrak{A}(R) = \mathfrak{B}(P) = 0$. Dies sind Modelle von der angegebenen Formel. Angenommen, es wäre zu einer Hornformel äquivalent, müsste auch die Interpretation $\mathfrak{A} \cdot \mathfrak{B}$ ein Modell sein. Allerdings kann man leicht prüfen, dass es nicht so ist.

Hausaufgabe 5 (3-Colouring) **5 Punkte**

Sei Graph $G = (V, E)$ mit Knoten in der Menge V, die potenziell unendlich ist. Wir sagen, dass der Graph G *3-färbbar* ist, wenn es eine Funktion $c\colon V \to [3]$ gibt, sodass für alle $uv \in E$ gilt dass $c(u) \neq c(v)$.

Beweisen Sie für jeden Graph G, dass G 3-färbbar ist genau dann wenn jeder endliche Teilgraph G_0 von G 3-färbbar ist.

Hinweis: Erweitern Sie die Formel vom letzten Blatt auch für unendliche Graphen, indem Sie die Formel in eine Formelmenge umschreiben.

Lösung:

Sei $G = (V, E)$ ein Graph. Wir übernehmen die Notation von Aufgabe 3 auf Blatt 1. Das heißt,

- Symbol $P_{v,c}$ steht für die Aussage: "Der Knoten v erhält Farbe c."
- Formel $\varphi_{v,\min}$ ist genau dann erfüllt, wenn der Knoten mindestens eine Farbe erhält.
- Formel $\varphi_{v,\max}$ ist genau dann erfüllt, wenn der Knoten maximal eine Farbe erhält.
- Formel φ_{uv} ist genau dann erfüllt, wenn die Kante $uv \in E$ unterschiedlich gefärbte Endpunkte hat.

Wir definieren die Formelmenge

$$\Phi_G := \{\varphi_{v,\min} \land \varphi_{v,\max} \mid v \in V\} \cup \{\varphi_{uv} \mid uv \in E\}\,.$$

Es gilt also für jeden Graphen G, dass G 3-färbbar ist genau dann wenn Φ_G erfüllbar ist.

Claim. Wenn alle endliche Teilgraphen G_0 von G 3-färbbar sind, ist auch G 3-färbbar.

Sei G ein Graph und Φ_G die oben erstellte Formelmenge. Sei $\Phi_0 \subseteq \Phi$ eine endliche Teilmenge von Φ. Sei

$$F = \{v \in V \mid P_{v,c} \in \mathrm{symb}(\Phi_0) \text{ für eine } c \in \{1, 2, 3\}\}$$

die Menge aller Knoten, deren zugehörigen Symbole in der Teilmenge Φ_0 vorkommen. Insbesondere ist F endlich. Sei $G[F]$ der von F induzierte Teilgraph von G. Wir wissen, dass $G[F]$ 3-färbbar ist, also ist die Menge $\Phi_{G[F]}$ erfüllbar und damit auch die Menge $\Phi_0 \subseteq \Phi_{G[F]}$. Nach dem Endlichkeitssatz ist auch Φ erfüllbar (da jede endliche Teilmenge von Φ erfüllbar ist). Es folgt, dass G 3-färbbar ist.

Claim. Wenn G 3-färbbar ist, dann sind alle endliche Teilgraphen G_0 von G 3-färbbar.

Sei G ein 3-färbbarer Graph und Φ_G die oben erstellte Formelmenge. Dann ist Φ_G erfüllbar. Sei G_0 ein endlicher Teilgraph von G und sei Φ_{G_0} die oben erstellte Formelmenge für G_0. Dann ist $\Phi_{G_0} \subseteq \Phi_G$ und nach dem Endlichkeitssatz ist Φ_{G_0}. Also ist G_0 3-färbbar. Da G_0 beliebig gewählt wurde, folgt das Claim.

Programmieraufgabe 6 (Normalformen) 5 Punkte

- Die Abgabe der Programmieraufgabe erfolgt über **Speichern** oder **Abgabe** in VPL. Bis zur Abgabefrist könnt ihr so oft abgeben, wie ihr wollt. Wir bewerten nur die aktuellste Abgabe.
- Ihr könnt in **assignment.py** euren eigenen Code schreiben und dabei die von uns zur Verfügung gestellten Bibliotheken benutzen. Achtet allerdings darauf, keine Dateien zu löschen und die Header der Funktionen unverändert zu lassen.
- Nicht alle Importe sind möglich, manche Bibliotheken werden also einen Fehler wie z.B. `Module assignment tries to import numpy, which does not exist` liefern, wenn ihr versucht diese zu verwenden.
- Wir empfehlen, den Code mindestens einmal zu testen, mit **Ausführen** oder Strg+F11. Dies kann einige Sekunden dauern.
- Punkte und Code sind automatisch mit eurer Abgabegruppe synchronisiert.

Diese Woche erweitern wir die Klasse `Formula` mit `BigAnd` und `BigOr`, um Formeln besser in disjunktiver und konjunktiver Normalform darstellen zu können.

Implementieren Sie zunächst das Paar von Funktionen `nnf_positive(formula: Formula) -> Formula` und `nnf_negative(formula: Formula) -> Formula`, welche als Eingabe ein Objekt der Klasse `Formula` nehmen, sich gegenseitig aufrufen dürfen, und entsprechend ein Objekt der Klasse `Formula` ausgeben, welches der Negationsnormalform der (negierten) Formel entspricht.

Schreiben Sie anschließend die Funktion `convert_to_dnf(formula: Formula) -> BigOr`, welche als Eingabe ein Objekt der Klasse `Formula` nimmt, und ein Objekt der Klasse `BigOr` ausgibt, welches einer zur Eingabe äquivalenten Formel in disjunktiver Normalform entspricht.

Schreiben Sie nun die Funktion `convert_to_cnf(formula: Formula) -> BigAnd`, welche als Eingabe ein Objekt der Klasse `Formula` nimmt, und ein Objekt der Klasse `BigAnd` ausgibt, welches einer zur Eingabe äquivalenten Formel in konjunktiver Normalform entspricht.

Berücksichtigen Sie, dass die **Negation** einer Formel in disjunktiver Normalform sich einfach als eine äquivalente Formel in konjunktiver Normalform darstellen lässt. So gilt zum Beispiel, dass

$$\neg((P \wedge Q) \vee (R \wedge S)) \equiv (\neg P \vee \neg Q) \wedge (\neg R \vee \neg S).$$

Lösung:

Eine rekursive Implementierung bietet sich für den Algorithmus der Negationsnormalform an. Dieser lässt sich direkt aus dem induktiven Beweis auf Seite 1.48a ableiten, wenn wir die Formeln φ' und φ'' als die zu implementierenden Funktionen auffassen. In der Vorlesung werden zwei Wege vorgestellt, eine Formel in DNF bzw. KNF zu überführen. Seite 1.55-a erklärt den Weg über Umformungen, Seite 1.59-b mittels Wahrheitstafeln.