

Übungen zur Vorlesung “SAT-Solving und Anwendungen” (Abgabe 2)

Aufgabe 2.1

- a) Jede einzelne Klausel (Minterm) in die kanonischen DNF von $exactlyOne(X)$ drückt aus, dass alle Variablen auf 0 gesetzt sind bis auf eine, die auf 1 gesetzt sein soll, hat also die Form

$$x_j \wedge \bigwedge_{i=1, i \neq j}^n \neg x_i$$

wobei x_j die Variable ist die auf 1 gesetzt werden soll. Eine Disjunktion dieser Klauseln, für jede Variable eine, drückt also aus, dass genau eine Variable 1 sein soll. Dies entspricht dann auch der kanonischen DNF für $exactlyOne(X)$:

$$\bigvee_{j=1}^n \left(x_j \wedge \bigwedge_{i=1, i \neq j}^n \neg x_i \right)$$

Die Anzahl der Klauseln (Minterme) in der kanonischen DNF ist also bei n Variablen n . Die Anzahl der mindestens nötigen Minterme ist ebenfalls n :

Beweis. Sei ϕ eine DNF für $exactlyOne(X)$. O.b.d.A. enthalte ϕ keine Klauseln die unerfüllbar sind, wenn doch entferne sie. Jede Klausel ϕ muss alle Variablen aus X negiert oder unnegiert enthalten. Angenommen dem wäre nicht so. Dann könnte die entsprechende Klausel, da sie ja erfüllbar ist per Voraussetzung, mit einer Belegung erfüllt werden, wobei diese Belegung alle nicht vorkommenden Variablen mit 1 belegt. Dadurch wäre ϕ schon erfüllt durch eine Belegung, die mehr als eine Variable auf 1 setzt und ϕ wäre keine DNF für $exactlyOne(X)$, Widerspruch. Also kommen in allen Klauseln von ϕ alle Variablen aus X vor und das pro Klausel entweder negiert oder nicht negiert, da die Klausel sonst unerfüllbar wäre und solche Klauseln o.b.d.A. nicht in ϕ vorkommen. Wären alle Variablen in einer Klausel negiert könnte diese durch die Belegung erfüllt werden, die alle Variablen auf 0 setzt, Widerspruch zur Annahme, ϕ wäre eine DNF für $exactlyOne(X)$. Wäre mehr als eine Variable unnegiert in einer Klausel könnte diese, und damit ϕ , von einer Belegung erfüllt werden, die mehr als eine Variable aus X auf 1 setzt, Widerspruch. Also haben alle Klauseln von ϕ die oben angegebene Form wie in der kanonischen DNF. Angenommen ϕ hätte weniger als n Klauseln. Dann würde für ein $1 \leq j \leq n$ die Klausel

$$x_j \wedge \bigwedge_{i=1, i \neq j}^n \neg x_i$$

fehlen, und alle anderen Klauseln erlauben nur exakt eine Variable auf 1 zu setzen. Damit würde die Belegung, die nur x_j auf 1 setzt ϕ nicht erfüllen, Widerspruch. Also hat ϕ doch mindestens n Klauseln. \square

Der Beweis zeigt, dass n schon die kleinste mögliche Zahl an Klauseln für eine solche Formel ist, mehr sogar: der Beweis zeigt, dass die kanonische DNF die einzige DNF ist die keine unnötigen Klauseln enthält.

- b) Anzugeben ist eine boolesche Funktion, welche 14 Variablen beinhaltet und mit nicht weniger als 120 Minterme in der DNF-Darstellung zurechtkommt. Zur Konstruktion einer solchen Funktion nutzen wir das KV-Diagramm, bekannt aus Einführung in die technische Informatik.

Konstruktion: Wir werden 14 Variablen verwenden, um 4 KV-Diagramme zu erstellen, welche nicht minimierbar sind. Des weiteren werden wir ein weiteres KV-Diagramm erstellen, welches über die Teilformeln, resultierend aus den ersten vier KV-Diagrammen ist, enthält. Wir weisen folgende Variablen Formeln zu, welche dann über ein KV-Diagramm repräsentiert werden.

$$A_{\text{var}} = \{x_1, x_2, x_3, x_4\}$$

$$B_{\text{var}} = \{x_5, x_6, x_7, x_8\}$$

$$C_{\text{var}} = \{x_9, x_{10}, x_{11}, x_{12}\}$$

$$D_{\text{var}} = \{x_{13}, x_{14}\}$$

Wir wissen - beinhaltet eine KV-Diagramm alternierende Belegung (Schachbrett), lässt sie sich nicht weiter in DNF minimieren. Wähle also folgendes Diagrammschema:

$A :=$		x_1	x_1	$\neg x_1$	$\neg x_1$	
	x_2	0	0	0	0	x_3
	x_2	0	1	0	0	$\neg x_3$
	$\neg x_2$	1	0	1	0	$\neg x_3$
	$\neg x_2$	0	0	0	0	x_3
		x_4	$\neg x_4$	$\neg x_4$	x_4	
$B :=$		x_5	x_5	$\neg x_5$	$\neg x_5$	
	x_6	0	0	1	0	x_7
	x_6	0	1	0	0	$\neg x_7$
	$\neg x_6$	1	0	1	0	$\neg x_7$
	$\neg x_6$	0	0	0	0	x_7
		x_8	$\neg x_8$	$\neg x_8$	x_8	
$C :=$		x_9	x_9	$\neg x_9$	$\neg x_9$	
	x_{10}	1	0	1	0	x_{11}
	x_{10}	0	1	0	0	$\neg x_{11}$
	$\neg x_{10}$	1	0	1	0	$\neg x_{11}$
	$\neg x_{10}$	0	0	0	0	x_{11}
		x_{12}	$\neg x_{12}$	$\neg x_{12}$	x_{12}	
$D :=$		x_{13}	$\neg x_{13}$			
	x_{14}	1	0			
	$\neg x_{14}$	0	1			

Die so entstehenden Teilformeln sind also nicht reduzierbar es gilt für D , dass sie 2 Minterme hat, A , dass die 3 hat, B 4 und C 5. Betrachte nun f :

$f(x_1, \dots, x_{14}) := A \wedge B \wedge C \wedge D$ Betrachte die Anzahl der Minterme von f : A besitzt 3, B 4 Minterme. Das Distributivgesetz sorgt dafür, dass die Anzahl der Minterme bei einer Konjunktion miteinander multipliziert werden, ersichtlich durch folgendes Beispiel: Hier werden die Anzahl der Minterme illustriert durch die Betragstriche.

$$\begin{aligned} |A \wedge B| &= |(A_1 \vee A_2 \vee A_3) \wedge B| \\ &= |A_1 B \vee A_2 B \vee A_3 B| \\ &= |A_1(B_1 \vee B_2 \vee B_3 \vee B_4) \vee A_2(B_1 \vee B_2 \vee B_3 \vee B_4) \vee A_3(B_1 \vee B_2 \vee B_3 \vee B_4)| \\ &= |A| \cdot |B| = 3 \cdot 4 = 12 \end{aligned}$$

So gilt, dass f genau $2 \cdot 3 \cdot 4 \cdot 5 = 120$ Minterme besitzt.

Aufgabe 2.2

Ganz offensichtlich kann dieses Verfahren nicht korrekt sein. Das prüfen, ob ein Minterm erfüllbar ist ist in linearer Zeit möglich und eine Formel enthält nur linear viele Minterme. Außerdem besitzt das Verfahren nach Tseitin polynomielle Laufzeit. Damit wäre, wenn das Verfahren hier korrekt wäre $P = NP$ und wir könnten uns diese Vorlesung sparen. Da das Verfahren für alle ϕ gelten müsste um korrekt zu sein wie es hier formuliert ist, genügt ein Gegenbeispiel. Betrachte

$$\phi = \neg((x \vee \neg x) \vee (x \wedge \neg x))$$

Dann ist

$$\neg\phi \equiv (x \vee \neg x) \vee (x \wedge \neg x)$$

Damit ist

$$\omega = Tseitin(\neg\phi) = ((x \vee \neg x) \vee t) \wedge (\neg t \vee x) \wedge (\neg t \vee \neg x) \wedge (t \vee \neg x \vee x)$$

mit der neuen Variable t . Jetzt ist nach dem Verfahren aus der Aufgabe:

$$\neg\omega = \neg((x \vee \neg x) \vee t) \vee \neg(\neg t \vee x) \vee \neg(\neg t \vee \neg x) \vee \neg(t \vee \neg x \vee x)$$

Die negierte zweite Klausel ist erfüllbar mit t auf 1 gesetzt und x auf 0, damit ist $\neg\omega$ erfüllbar, aber ϕ war offenbar eine Kontradiktion. Da es eine Formel gibt, für die das Verfahren fehlschlägt, ist es im allgemeinen nicht richtig.

Aufgabe 2.3

```
File: goldb-heqc-k2mul.cnf
Problem line: #vars = 11680, #clauses = 74581
Variable count: 11680
Clause count: 74581
Literal count: 23360
Maximal occurrence of a variable: 3438
Variables with maximum number of occurrences: [37]
Positive pure literals: []
Negative pure literals: []
Unit clauses: [1242, 1243, 38360, 38361]
```

File: aim-100-1_6-no-1.cnf
Problem line: #vars = 100, #clauses = 160
Variable count: 100
Clause count: 160
Literal count: 200
Maximal occurrence of a variable: 8
Variables with maximum number of occurrences: [81]
Positive pure literals: []
Negative pure literals: []
Unit clauses: []

File: aim-200-2_0-yes1-2.cnf
Problem line: #vars = 200, #clauses = 400
Variable count: 200
Clause count: 400
Literal count: 400
Maximal occurrence of a variable: 6
Variables with maximum number of occurrences: [1..200]
Positive pure literals: []
Negative pure literals: []
Unit clauses: []

File: barrel5-no.cnf
Problem line: #vars = 1407, #clauses = 5383
Variable count: 1407
Clause count: 5383
Literal count: 2814
Maximal occurrence of a variable: 127
Variables with maximum number of occurrences: [1277]
Positive pure literals: [1025, 514, 771, 1284, 262, 522, 779, 1036, 1294, 2
Negative pure literals: [-1029, -518, -775, -10, -266, -526, -783, -1040, -
Unit clauses: []

File: hanoi4-yes.cnf
Problem line: #vars = 718, #clauses = 4934
Variable count: 718
Clause count: 4934
Literal count: 1436
Maximal occurrence of a variable: 40
Variables with maximum number of occurrences: [640, 1, 259, 517, 134, 648,
Positive pure literals: [389, 390, 391, 711, 712, 713, 205, 206, 207, 527,
Negative pure literals: [-54, -71, -61, -47]
Unit clauses: [4266, 4267, 4268, 4269, 4270, 4271, 4308, 4309, 4310, 4311,

File: longmult6-no.cnf
Problem line: #vars = 2848, #clauses = 8853
Variable count: 2848
Clause count: 8853
Literal count: 5694
Maximal occurrence of a variable: 234

Variables with maximum number of occurrences: [354, 1299, 984, 1929, 669, 1
Positive pure literals: [2816, 2817, 2818, 2819, 2820, 2821, 2822, 2823, 28
Negative pure literals: [-2817, -2818, -2819, -2820, -2821, -2822, -2823, -
Unit clauses: [7558, 7559, 7560, 7561, 7562, 7563, 7564, 7565, 7566, 7567,

File: miza-sr06-md5-47-03.cnf
Problem line: #vars = 65604, #clauses = 273522
Variable count: 65604
Clause count: 273522
Literal count: 131208
Maximal occurrence of a variable: 40733
Variables with maximum number of occurrences: [1]
Positive pure literals: [1026, 1027, 1028, 1032, 1033, 1034, 1036, 1683, 16
Negative pure literals: []
Unit clauses: [0, 30818, 30819, 30820, 30821, 30822, 30823, 30824, 30825, 3

File: ssa7552-160-yes.cnf
Problem line: #vars = 1391, #clauses = 3126
Variable count: 1391
Clause count: 3126
Literal count: 2782
Maximal occurrence of a variable: 272
Variables with maximum number of occurrences: [84]
Positive pure literals: [1026, 1031, 1033, 1034, 1037, 1041, 1042, 1043, 10
Negative pure literals: [-2, -1026, -4, -5, -6, -7, -1031, -8, -9, -1033, -
Unit clauses: [0, 1, 2, 3]

hole8-no.cnf schmeißt eine Exception.