Minesweeper ist NP-vollständig

Damian Schmidt

28.06.2007

1 Einleitung

Das bekannte Spiel Minesweeper („Minenräumer“) wird mit Microsoft Windows mitgeliefert. In einem rechteckigen Feld aus quadratischen Kästchen sind hinter einigen Kästchen Minen verborgen. Ziel des Spiels ist es, möglichst schnell alle Kästchen ohne Minen aufzudecken, ohne jedoch durch Aufdecken eines Minenfelds eine Mine zur Explosion zu bringen. Deckt man ein Minenfeld auf, so ist das Spiel verloren. Alle aufgedeckten Kästchen ohne Mine sind leer oder zeigen eine Zahl von 1 bis 8 an, die der Anzahl der benachbarten Minenfelder entspricht. Noch nicht aufgedeckte Kästchen lassen sich mit einer Fahne oder einem Fragezeichen markieren. Aus den angezeigten Ziffern lässt sich meistens auf die exakten Positionen der Minen schließen. In manchen Fällen sind solche Schlüsse nur durch Einbeziehung sehr vieler Minenkästchen oder des ganzen Spielfelds möglich.

2 Definitionen

2.1 Konfiguration

Das Spielfeld von Minesweeper besteht aus $m \times n$ Kästchen, die mit den in Minesweeper üblichen Symbolen gefüllt sind. Erlaubt sind unaufgedeckte Kästchen (?), Minen (*) und leere Kästchen, die eine Zahl beinhalten, welche den benachbarten Minen entspricht. Wir definieren eine **Konfiguration** als $m \times n$ Matrix A mit den Einträgen

$$a_{i,j} \in S$$

für $1 \leq i \leq m$ und $1 \leq j \leq n$ und $S = \{?, *, 0, 1, 2, 3, 4, 5, 6, 7, 8\}$.

2.2 Konsistenz

Eine Konfiguration nennen wir **konsistent**, wenn alle unaufgedeckten Kästchen so mit Minen belegt werden können, dass es keinen Widerspruch gibt. Für die Konsistenz einer Konfiguration A gilt:

$$A \; ist \; konsistent \iff \exists \, A^* \; mit$$

$$\forall \, a_{i,j} \neq ? \, : \, a^*_{i,j} = a_{i,j} \;\; und \;\; \forall \, a_{i,j} = ? \, : \, a^*_{i,j} \in \{* \, , \, z\} \, ,$$

$$z = | \, \{ \, (x, \, y) \mid a^*_{x,y} = * \, , \, x \in \{i-1 \, , \, i \, , \, i+1\} \, , \, y \in \{j-1 \, , \, j \, , \, j+1\}\} \, |$$

wobei z die Anzahl der jeweils zum Kästchen benachbarten Minen ist.

2.3 MINESWEEPER-Problem

Wir definieren das MINESWEEPER-Problem als Frage, ob eine Konfiguration konsistent ist oder nicht und zeigen, dass es NP-vollständig ist.

Die Frage der Konsistenz ist, abgesehen von Zufall, Wahrscheinlichkeiten und damit verbundenen Strategien, zentraler Punkt des Spiels. Macht man für ein unaufgedecktes Kästchen eine Annahme (Mine vorhanden oder keine Mine vorhanden) und erhält als Folge der Annahme eine inkonsistente Konfiguration, so ist die Annahme falsch und ihr Gegenteil wahr. Oft lässt sich diese Frage nicht sofort beantworten, sondern es müssen weitere Annahmen getroffen und die entsprechenden Konfigurationen auf Konsistenz geprüft werden.

Beim Beispiel in Abbildung 1 lassen sich die Positionen der Minen eindeutig ermitteln.

3 Beweis

Durch Reduktion von SAT auf MINESWEEPER zeigen wir zunächst, dass MINESWEEPER ebenso ist schwer wie SAT und alle anderen Probleme, die in NP liegen (Satz von Cook). Im zweiten Teil zeigen wir, dass MINESWEEPER auch selbst in NP liegt. Daraus folgt, dass MINESWEEPER NP-vollständig ist.

	2	2	2	2	
	2	0	0	2	
	2	0	0	2	
	2	2	2	2	

Abbildung 1: Wo liegen die Minen?

3.1 Reduktion von SAT auf MINESWEEPER

3.1.1 Grundidee der Reduktion

SAT beschreibt das Problem der Erfüllbarkeit einer aussagenlogischen Formel, die in konjunktiver Normalform gegeben ist. Zu einer solchen Formel lässt sich in polynomieller Zeit ein elektrischer Schaltplan konstruieren, der als Eingabesignale die Variablen und als Ausgabesignal den Ergebniswert der Formel hat (wahr oder falsch).

Dieser Schaltplan wird (in polynomieller Zeit) als Minesweeper Konfiguration dargestellt, so dass das boolesche Ausgabesignal und die booleschen Eingabesignale als Kästchen mit Mine (=wahr) oder ohne Mine (=falsch) dargestellt werden.

Setzt man nun ins Kästchen des Ausgabesignals eine Mine (=wahr), und lässt die Kästchen der Eingabesignale unaufgedeckt, dann ist die Frage, ob die aussagenlogische Formel erfüllbar ist (SAT) äquivalent zu der Frage, ob die konstruierte Konfiguration konsistent ist (MINESWEEPER) und die Reduktion gelungen.

3.1.2 Konkretes Beispiel

Die Frage, ob die Formel $\neg(a \vee \neg b)$ erfüllbar ist, ist äquivalent zur Frage, ob die Minesweeper Konfiguration aus Abbildung 2 konsistent ist für den Fall, dass R wahr ist, d.h. in R eine Mine liegt.

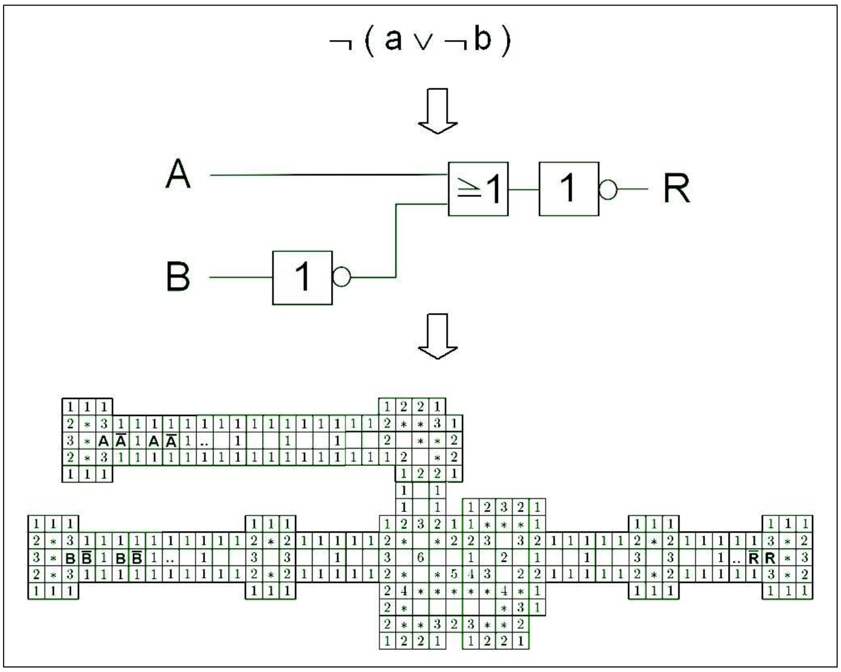

Abbildung 2: Reduktion von SAT auf MINESWEEPER

3.1.3 Schaltelemente

Damit eine aussagenlogische Formel als Minesweeper Konfiguration konstruiert werden kann, müssen alle benötigten Schaltelemente entworfen werden. Außer einer vollständigen Basis {NOT, OR} müssen Leitungen (mit Knick und Endstück), Verzweigungen, Überkreuzungen und Phasenänderungen definiert werden.

$\mathbf{X} \longrightarrow$

$\cdots$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\cdots$
$\cdots$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\cdots$
$\cdots$	x	1	x'	x	1	x'	x	1	x'	x	1	x'	x	1	x'	x	$\cdots$
$\cdots$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	$\cdots$
$\cdots$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\cdots$

Abbildung 3: Leitung

$\mathbf{X} \longrightarrow$				1	2	2	1		
$\cdots$	1	1	1	2	$*$	$*$	3	1	
$\cdots$	1	x'	x	2	x'	$*$	$*$	2	
$\cdots$	1	1	1	1	2	x	$*$	2	
					1	2	2	1	
					1	x'	1		
					1	x	1	$\mathbf{X}$	
					1	1	1	$\downarrow$	
					$\vdots$	$\vdots$	$\vdots$		

1	1	1		$\mathbf{X} \longrightarrow$				
2	$*$	3	1	1	1	1	1	$\cdots$
3	$*$	x'	x	1	x'	x	1	$\cdots$
2	$*$	3	1	1	1	1	1	$\cdots$
1	1	1						

Abbildung 4: geknickte Leitung und Endstück

				$\vdots$	$\vdots$	$\vdots$	$\uparrow$				
				1	x	1	$\mathbf{X}$				
				1	x'	1					
	$\mathbf{X} \longrightarrow$			1	1	1	$\mathbf{X}' \longrightarrow$				
$\cdots$	1	1	1	1	x	1	1	1	1	$\cdots$	
$\cdots$	x'	x	1	x'	2	x'	1	x	x'	$\cdots$	
$\cdots$	1	1	1	1	x	1	1	1	1	$\cdots$	
				1	1	1					
				1	x'	1	$\mathbf{X}$				
				1	x	1	$\downarrow$				
				$\vdots$	$\vdots$	$\vdots$					

Abbildung 5: Verzweigung

	$\mathbf{X} \longrightarrow$					1	1	2	1	1			$\mathbf{X} \longrightarrow$			
$\cdots$	1	1	1	1	1	2	$*$	3	$*$	2	1	1	1	1	1	$\cdots$
$\cdots$	x'	x	1	x'	x	3	x'	5	x	3	x'	x	1	x'	x	$\cdots$
$\cdots$	1	1	1	1	1	2	$*$	3	$*$	2	1	1	1	1	1	$\cdots$
						1	1	2	1	1						

Abbildung 6: Phasenänderung

		$\mathbf{X} \longrightarrow$				1	1	1		$\mathbf{X}' \longrightarrow$				
$\cdots$	1	1	1	1	1	2	$*$	2	1	1	1	1	1	$\cdots$
$\cdots$	x'	x	1	x'	x	3	x'	3	x	x'	1	x	x'	$\cdots$
$\cdots$	1	1	1	1	1	2	$*$	2	1	1	1	1	1	$\cdots$
						1	1	1						

Abbildung 7: NOT

				1	1	1									
			$\mathbf{U}$	1	u'	1									
			$\downarrow$	1	u	1		1	2	3	2	1			
$\mathbf{V} \longrightarrow$			1	2	3	2	1	1	$*$	$*$	$*$	1	$\mathbf{R} \longrightarrow$		
1	1	1	2	$*$	u'	$*$	2	2	3	r'	3	2	1	1	1
1	v'	v	3	v'	6	r	r'	1	r	2	r	1	r'	r	1
1	1	1	2	$*$	s	$*$	5	4	3	r'	2	2	1	1	1
			2	4	$*$	$*$	$*$	$*$	$*$	4	$*$	1			
			2	$*$	s'	a_1	a_2	a_3	r	$*$	3	1			
			2	$*$	$*$	3	2	3	$*$	$*$	2				
			1	2	2	1		1	2	2	1				

Abbildung 8: OR

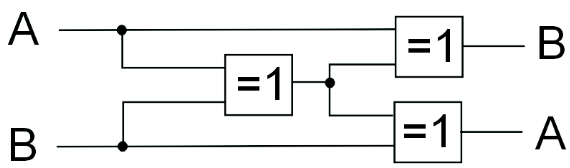

Abbildung 9: Überkreuzung von Leitungen, realisiert durch drei XOR - Gatter

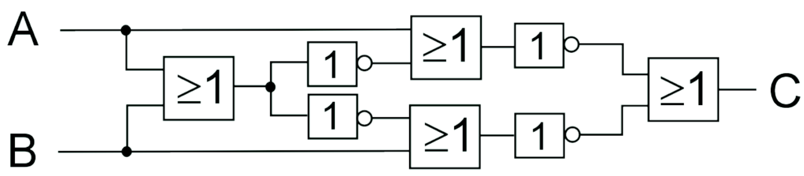

Abbildung 10: XOR - Gatter

3.1.4 Polynomielle Laufzeit der Reduktion

Für eine SAT-Eingabe der Länge n lässt sich auf einem Minesweeper - Spielfeld der Größe $c * (n * n)$ eine entsprechende Schaltung konstruieren (c ist Konstante), weil pro Literal bzw. Operator der Eingabe nur konstant viele Schaltelemente nötig sind. Anzahl der Bauelemente und Laufzeit der Konstruktion liegen folglich in $O(n^2)$.

3.2 Prüfung einer richtig geratenen Belegung in polynomieller Zeit

Gegeben sei eine richtige Eingabe für das MINESWEEPER-Problem, also eine konsistente Konfiguration, in der die zuvor unaufgedeckten Kästchen mit Minen oder Zahlen (= keine Mine) belegt sind. Weil die Spielfeldgröße $c * (n * n)$ beträgt und sich ein Kästchen in $O(1)$ auf Widersprüche prüfen lässt, geschieht die Prüfung der Gesamtkonfiguration dann in $O(n^2)$.

4 Ausblick und Anmerkungen

4.1 Beweis von P = NP anhand von Minesweeper?

Falls ein Algorithmus mit polynomieller Laufzeit zur Lösung von Minesweeper gefunden wird oder es einen Spieler gibt, der Minesweeper - Konfigurationen in polynomieller Zeit auf Konsistenz prüfen kann, so würden alle Probleme aus NP in P liegen und $P = NP$ wäre bewiesen. Auch wenn dies von Kaye ein interessanter Ansatz genannt wurde, bleibt es eher unwahrscheinlich, durch Minesweeper entsprechende Fortschritte zu erzielen. Vor allem, weil die Reduktion von SAT auf MINESWEEPER eher zu einer Verkomplizierung als zu einer Vereinfachung führt.

4.2 Weitere Schaltelemente

Zum Beweis wurde die minimale Menge der einfachsten Schaltelemente benutzt, die nötig ist, um jede aussagenlogische Formel konstruieren zu können. In den folgenden Abbildungen sind weitere Schaltelemente dargestellt.

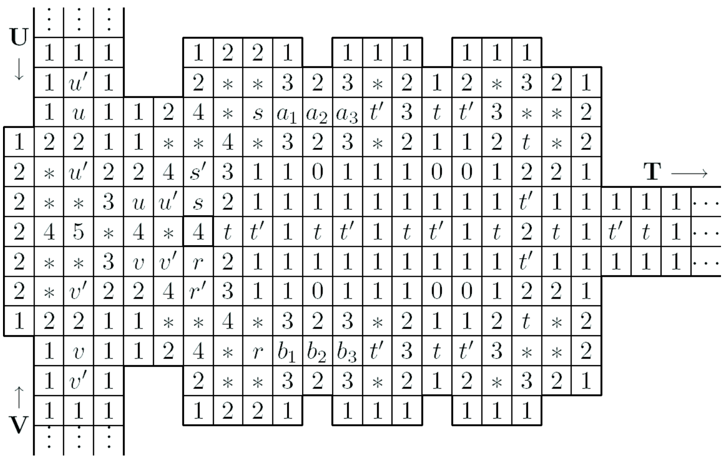

Abbildung 11: AND

Literatur

[1] R.W. Kaye, *Minesweeper is NP-complete*, Mathematical Intelligencer, vol 22, number 2, pp9-15, 2000

[2] R.W. Kaye, *Some Minesweeper Configurations*. Published in Portugese in Boletim Sociedade Portuguesea de Mathemática, Janeiro 2007 (Número especial), Lisbon. ISSN 0872-3672. pp 181-189.

[3] Richard Kaye's Minesweeper Pages,
`http://web.mat.bham.ac.uk/R.W.Kaye/minesw/`

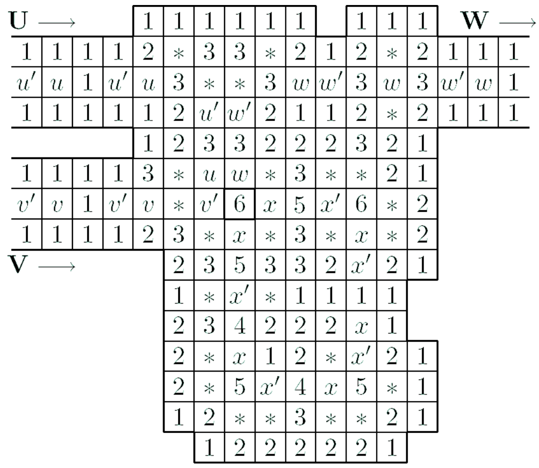

Abbildung 12: XOR

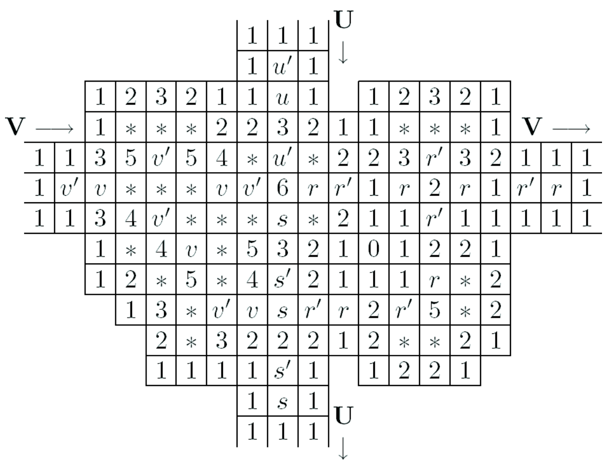

Abbildung 13: Direkte Überkreuzung