5 Algorithmus von Paturi, Pudlák und Zane

In diesem Kapitel betrachten wir einen probabilistischen Algorithmus, der der Vorgehensweise bei DLL ähnlicher ist also die im vorherigen Kapitel.

Wie bei einem DLL-Verfahren werden die Variablen der Reihe nach betrachtet und ihnen Werte zugewiesen. Dabei werden Variablen in Einerklauseln natürlich so bewertet, dass diese erfüllt sind, die übrigen Variablen werden zufällig bewertet. Die Reihenfolge, in der die Variablen betrachtet werden, ist durch eine Permutation $\pi \in S_n$ der Indizes $1, \ldots, n$ vorgegeben, die ebenfalls zufällig gewählt wird.

Ist am Ende die Formel erfüllt, wird die so gefundene Bewertung audsgegeben, andernfalls wird sie verworfen und mit einer neuen Permutation von vorn begonnen.

Die Prozedur suche(π) betrachtet die Variablen in der durch π vorgegebenen Reihenfolge und weist ihnen einen zufälligen Wert zu, außer wenn ein Wert bereits durch Einerklauseln erzwungen ist.

```
suche(π)
  for i := 1 to n do
      if Einerklausel x^ε_π(i) in F
          then ε_i := ε
          else wähle ε_i zufällig aus {0, 1}
      F := F[x_π(i) ← ε_i]
  if α ⊨ F
      then return α
      else return NIL
```

Diese Prozedur wird in einer Schleife eine bestimmte Anzahl w mal wiederholt.

```
  repeat w times
      wähle π ∈ S_n zufällig
      β := suche(π)
      if β ≠ NIL
          then return β
  return NIL
```

Wie bei den probablilistischen Algorithmen im vorhergehenden Kapitel berechnen wir die Wahrscheinlichkeit $p(n)$ dafür, dass ein Schleifendurchlauf erfolgreich ist, also eine erfüllende Bewertung findet, im Fall dass F erfüllbar ist. Daraus ergibt sich, dass $w = O(1/p(n))$ Wiederholungen genügen, um die Fehlerwahrscheinlichkeit beliebig klein zu halten, und damit eine Laufzeit von $O(1/p(n))$.

Definition. *Für eine Menge* A *von Bewertungen und* $\alpha \in A$ *ist*

$$N_A(\alpha) := |H(\alpha, 1) \cap A| - 1$$

die Anzahl der Nachbarn von α *in* A.

Lemma 19. *Ist* $A \neq \emptyset$, *so gilt*

$$\sum_{\alpha \in A} 2^{-N_A(\alpha)} \geq 1 .$$

Beweis. Die Aussage wird durch Induktion nach n bewiesen. Der Induktionsanfang für $n = 0$ ist trivial.

Sei also $n > 0$, und gelte die Aussage für Mengen von Bewertungen von $n-1$ Variablen. Für eine Bewertung α der Variablen $x_1, \ldots, x_{n-1}$ und $i = 0, 1$ sei $\alpha i := \alpha \cup [x_n \leftarrow i]$.

Sei $A \neq \emptyset$ eine Menge von Bewertungen aller n Variablen. Für $i = 0, 1$ sei $A_i := \{\, \alpha \,;\, \alpha i \in A \,\}$. Für mindestens ein i muss $A_i \neq \emptyset$ sein.

Ist $A_0 = \emptyset$, so muss $A_1 \neq \emptyset$ sein. Also ist

$$\sum_{\alpha \in A} 2^{-N_A(\alpha)} = \sum_{\alpha \in A_1} 2^{-N_A(\alpha 1)} = \sum_{\alpha \in A_1} 2^{-N_{A_1}(\alpha)} \geq 1 ,$$

wobei die letzte Ungleichung nach der Induktionshypothese gilt. Die letzte Gleichheit gilt, da für $\alpha \in A_1$ der einzige Nachbar von $\alpha 1$, der nicht von der Form β_1 ist, gerade $\alpha 0$ ist. $A_0 = \emptyset$ ist aber $\alpha 0 \notin A$.

Der Fall $A_1 = \emptyset$ ist symmetrisch. Seien nun A_0 und A_1 beide nichtleer. Dann ist

$$\begin{aligned}
\sum_{\alpha \in A} 2^{-N_A(\alpha)} &= \sum_{\alpha \in A_0} 2^{-N_A(\alpha 0)} + \sum_{\alpha \in A_1} 2^{-N_A(\alpha 1)} \\
&\geq \sum_{\alpha \in A_0} 2^{-N_{A_0}(\alpha)-1} + \sum_{\alpha \in A_1} 2^{-N_{A_1}(\alpha)-1} \\
&\geq \frac{1}{2} + \frac{1}{2} = 1 .
\end{aligned}$$

Dabei gilt die vorletzte Ungleichung wiederum, da es für $i = 0, 1$ höchstens einen Nachbarn von αi gibt, der nicht von der Form β_i ist, nämlich $\alpha 0$ $\square$

Zur Berechnung der Erfolgswahrscheinlichkeit $p(n)$ sei $A := \{\, \alpha \,;\, \alpha \models F \,\}$ die Menge der F erfüllenden Bewertungen, und sei $\alpha \in A$. Definiere

$$j(\alpha) := n - N_A(\alpha)$$

und für $i = 1, \ldots, n$ sei $\alpha_i := \alpha$ with $[x_i \leftarrow 1 - \alpha(x)]$. Das heißt, es gibt $j(\alpha)$ Variablen x_i mit $\alpha_i \not\models F$, diese heißen die *kritischen* Variablen.

Für jede kritische Variable x_i gibt es eine *kritische* Klausel C_i in F, derart dass das einzige Literal a in C_i mit $\alpha(a) = 1$ entweder x_i oder $\neg x_i$ ist.

Für eine Permutation $\pi \in S_n$ sei

$$R(\pi, \alpha) = \Big\{ x_i \text{ kritisch} \,;\, \pi^{-1}(i) \geq \pi^{-1}(j) \text{ für alle } x_j \in V(C_i) \Big\}$$

die Menge derjenigen kritischen Variablen x_i, die unter den Variablen in ihrer kritischen Klausel C_i in der Permutation π die letzte sind, und $r(\pi, \alpha) = |R(\pi, \alpha)|$.

Da die Permutation π gleichverteilt gewählt wird, ist jede der k Variablen in C_i mit gleicher Wahrscheinlichkeit $1/k$ die letzte in π. Jede der $j(\alpha)$ kritischen Variablen ist also mit Wahrscheinlichkeit $1/k$ in $R(\pi, \alpha)$, daher ist der erwartete Wert $E[r(\pi, \alpha)] = j(\alpha)/k$.

Mit jeder Variablen $x_i \in R(\pi, \alpha)$ erhöht sich die Wahrscheinlichkeit dafür, dass α bei suche(π) gefunden wird: wenn die anderen Variablen y in C_i zufällig den richtigen Wert $\alpha(y)$ erhalten, dann erhält auch x_i den Wert $\alpha(x_i)$, da zu dem Zeitpunkt, wo der Algorithmus x_i betrachtet, die Klausel C_i Einerklausel ist.

Daher ist die Wahrscheinlichkeit $p_{\alpha,\pi}$ dafür dass α gefunden wird, falls die Permutation π gewählt wurde, mindestens $p_{\alpha,\pi} \geq 2^{-n+r(\pi,\alpha)}$.

Also ist die Wahrscheinlichkeit p_α dafür, dass α gefunden wird, insgesamt

$$\begin{aligned} p_\alpha &= \sum_{\pi \in S_n} \frac{1}{n!} p_{\alpha,\pi} = \sum_{\pi \in S_n} \frac{1}{n!} 2^{-n+r(\pi,\alpha)} \\ &= 2^{-n} \sum_{\pi \in S_n} \frac{1}{n!} 2^{r(\pi,\alpha)} = 2^{-n} \cdot E[2^{r(\pi,\alpha)}] \,. \end{aligned}$$

Der Erwartungswert $E[2^{r(\pi,\alpha)}]$ ist aber nach der Jensen'schen Ungleichung mindesten $2^{E[r(\pi,\alpha)]}$, also ist

$$p_\alpha \geq 2^{-n+E[r(\pi,\alpha)]} \geq 2^{-n+j(\alpha)/k} \,.$$

Die Jensen'sche Ungleichung besagt, dass für eine Zufallsvariable X, die jeden der Werte $a_1, \ldots, a_n$ mit Wahrscheinlichkeit $1/n$ annimmt, gilt $E[2^X] \cdot 2^{E[x]}$. Dies rechnet man nach, indem man den Logarithmus zur Basis 2 betrachtet, also zeigt $\log E[2^X] \geq E[X]$. Es ist nämlich

$$\begin{aligned} \log E[2^X] &= \log\Big(\frac{1}{n} \sum_{i=1}^{n} 2^{a_i}\Big) \geq \log\Big(\Big(\prod_{i=1}^{n} 2^{a_i}\Big)^{1/n}\Big) \\ &= \frac{1}{n} \log \prod_{i=1}^{n} 2^{a_i} = \frac{1}{n} \sum_{i=1}^{n} \log 2_i^a = \frac{1}{n} \sum_{i=1}^{n} a_i = E[X] \end{aligned}$$

wobei die erste Ungleichung nach dem Satz vom arithmetischen und geometrischen Mittel gilt.

Die gesamte Erfolgswahrscheinlichkeit lässt sich nun beschränken durch

$$\begin{aligned} p(n) &= \sum_{\alpha \in A} p_\alpha \geq \sum_{\alpha \in A} 2^{-n+j(\alpha)/k} \\ &= \sum_{\alpha \in A} 2^{-n+n/k+(j(\alpha)-n)/k} = 2^{-n+n/k} \sum_{\alpha \in A} 2^{-N(\alpha)/k} \\ &\geq 2^{-n+n/k} \sum_{\alpha \in A} 2^{-N(\alpha)} \geq 2^{-n+n/k}. \end{aligned}$$

wobei die letzte Ungleichung nach Lemma 19 gilt.

Also ist $p(n) \geq 2^{-n+n/k}$, und somit erhalten wir als Schranke an die Laufzeit $O\big((2^{1-1/k})^n\big)$. Die Basis errechnet sich für kleine Werte von k wie folgt:

k	3	4	5	6	7	8
$2^{1-1/k}$	1.58741	1.68180	1.74111	1.78180	1.81145	1.83401

Dieser Algorithmus war zum Zeitpunkt seiner Veröffentlichung 1997 der erste, der eine bessere Komplexität für k-SAT mit $k \geq 4$ erreicht als der von Monien und Speckenmeyer. Ausserdem war dies der erste theoretisch analysierte probabilistische Algorithmus für k-SAT.

Eine verbesserte Form dieses Algorithmus wurde von Paturi, Pudlák, Saks und Zane etwas später angegeben. Der Algorithmus arbeitet genau wie der hier vorgestellte, aber es wird ein Präprozessor vorgeschaltet, der die Formel um alle Klauseln ergänzt, die aus den Eingabeklauseln durch Resolution erzeugt werden können, und deren Größe durch einen (von k und n abhängigen) Parameter s beschränkt ist. Die Analyse ist zu komplex, um hier präsentiert zu werden.

Dieser verbesserte Algorithmus ist derzeit der beste probabilistische Algorithmus für k-SAT mit $k \geq 4$, er erreicht die Komplexität $O(b_k^n)$ für die folgenden Werte der Basis b_k:

k	3	4	5	6	7	8
b_k	1.36406	1.49579	1.56943	1.63788	1.68753	1.72521