Nisan Wigderson Generatoren

- Konstruktion und Ananlyse -

vorgetragen von Thomas Fuchs

Inhalt

Einleitung

Konventionen
Zwei Begriffe wollen wir hier durch deutlich schärfere ersetzen:

bisher:		**hier:**
Algorithmus	<->	Schaltkreis
Laufzeit	<->	Anzahl der Gatter

Ein Schaltkreis von **s** Gattern kann mit $O(s \cdot \log s)$ Bits beschrieben werden, es gibt $2^{O(s \cdot \log s)}$ Schaltkreise mit bis zu **s** Gattern.

Jede boolsche Funktion mit **n** Eingängen kann von einem Schaltkreis der Größe $O(2^n)$ berechnet werden.

Eigenschaften von NW Generatoren
Bei Eingabe der Länge **O(l)** wird Ausgabe der Länge $2^{\square(l)}$ erzeugt.

Ausgabe ist in $2^{O(l)}$ berechenbar und von Gegnern der Größe der Größe $2^{\square(l)}$ nicht von der Gleichverteilung zu unterscheiden.

Konstruktion

Nisan-Wigderson Generatoren benutzen kombinatorische Designs, um mit einer boolschen Funktion f aus einem „seed", also einem Zufallsstring, eine wesentlich längere Ausgabe zu erzeugen.

Kombinatorisches Design
Sei $(S_1,...,S_m)$ eine Familie von Teilmengen eines Universums U mit $|S_i| = l, 1 \leq i \leq m$ und $|S_i \cap S_j| \leq \alpha, i \neq j, 1 \leq i, j \leq m$. Dann ist $(S_1,...,S_m)$ ein (l,α)-Design.

Notation
Seien $z \in \{0,1\}^t$ ein String und $S \subseteq [t]$ eine Teilmenge des Universums $[t]$. Dann bezeichnet $z_{|S}$ den String der Länge $|S|$, der aus **z** durch Auswahl der durch **S** indizierten Bits entsteht.

Beispiel: mit $z = (0,0,1,0,1,0)$ und $S = \{1,2,3,5\}$ ensteht $z_{|S} = (0,0,1,1)$

Der Nisan-Wigderson Generator
Für eine boolsche Funktion $f: \{0,1\}^l \to \{0,1\}$ und ein Design $S = (S_1,...,S_m)$ über $[t]$ mit $|S_i| = l, 1 \leq i \leq m$, ist der Nisan-Wigderson Generator definiert als Funktion $NW_{f,S}: \{0,1\}^t \to \{0,1\}^m$ durch:

$$NW_{f,S}(z) = f(z_{|S_1}) \cdot f(z_{|S_2}) \cdot ... \cdot f(z_{|S_m})$$

Wir beobachten: Die Länge der Ausgabe eines NW Generators hängt vom Design ab. Je mehr Elemente dieses hat, desto Länger die Ausgabe. Jedoch kann ein Design nur endlich viele voneinander verschiedene Elemente enthalten.
Weiterhin kann man feststellen, dass die Qualität der Ausgabe ebenso stark von dem Design abhängig ist. Es ist zu befürchten, dass ein schlecht gewähltes Design es auch schwächeren Gegnern erlaubt, die Ausgabe des NW Generators von der Gleichverteilung zu unterscheiden.

Das folgende Theorem versichert nun, dass ein Design konstruierbar ist, welches die maximale Länge der Ausgabe eines NW Generators zulässt.

Theorem
Für jede Zahl l(Länge des seeds, Stelligkeit von f) und jeden Bruch $0 < \alpha \leq 1$(skaliert das Design) existiert ein (l, logm)-Design $(S_1,...,S_m)$ über dem Universum $[t]$, mit $t = O(l/\alpha)$ und $m = 2^{\alpha}$. Solch ein Design kann in $O(2^t t m^2)$ konstruiert werden.

Analyse

Wir werden hier folgende Aussage untersuchen:

Falls es einen Schaltkreis D gibt, der NW von der Gleichverteilung unterscheidet,
dann gibt es einen Schaltkreis C, der f(x) mit entsprechender Wahrscheinlichkeit voraussagen kann.

Dies ist genau die Aussage des folgenden Hilfssatzes:

Lemma
Seien $f:\{0,1\}^l \to \{0,1\}$ eine boolsche Funktion und $S=(S_1,...,S_m)$ ein $(l,\log m)$-Design über $[t]$.
Angenommen es gibt $D:\{0,1\}^m \to \{0,1\}$, so dass
$\left|\Pr_r[D(r)=1]-\Pr_z[D(NW_{f,S}(z))=1]\right|>\delta$.
Dann existiert ein Schaltkreis **C** der Größe $O(m^2)$, so dass
$\left|\Pr_x[D(C(x))=f(x)]-\frac{1}{2}\right|\geq\frac{\delta}{m}$.

Um das Lemma zu beweisen verfolgt man folgenden Ansatz:
Wenn dieser Schaltkreis D in der Lage ist, die Ausgabe eines NW Generators von der Gleichverteiltung zu unterscheiden, dann gibt es ein Bit in der Ausgabe, an welchem der Unterschied wahrgenommen wird. Dieses Bit, also f(x), wird von D von einem Zufallsbit unterschieden.
Damit kann D in einen „Vorhersager" von f(x) umgewandelt werden.
Um dieses Bit zu finden, bedienen wir uns folgender Definition:

Hybrid Argument
Wir definieren $m+1$ Verteilungen $H_0,...,H_m$ wie folgt:
Erstellen $v=NW_{f,S}(z)$, **z** zufällig gewählt und $r\in\{0,1\}^m$, Bits in **r** gleichverteilt.
Die H_i sind definiert als Konkatenation der ersten **i** Bits von **v** und der letzten **m-i** Bits von **r**.

Also ist H_m wie $NW_{f,S}(z)$ verteilt, wohingegen H_0 der Gleichverteilung über $\{0,1\}^m$ entspricht.

Beweis
Sei $b_0\in\{0,1\}$ das Bit, an dem $NW_{f,S}(z)$ durch **D** von der Gleichverteilung über $\{0,1\}^m$ unterschieden wird.
Mit $D'(x)=b_0\oplus D(x)$ gilt:

$$\varepsilon \le \Pr_z[D(NW_{f,S}(z)) = 1] - \Pr_r[D(r) = 1]$$
$$= \Pr[D(H_m) = 1] - \Pr[D(H_0) = 1]$$
$$= \sum_{i=1}^{m} (\Pr[D(H_i) = 1] - \Pr[D(H_{i-1}) = 1])$$

Es gibt also einen Index **i**, so dass:

$$\frac{\varepsilon}{m} \le \Pr[D(H_i) = 1] - \Pr[D(H_{i-1}) = 1]$$

mit

$$H_i = f(z_{|S_1})...f(z_{|S_{i-1}})f(z_{|S_i})r_{i+1}...r_m$$
$$H_{i-1} = f(z_{|S_1})...f(z_{|S_{i-1}})r_i r_{i+1}...r_m$$

Wir haben also zunächst die Stelle i als das Bit in der Ausgabe des NW Generators identifiziert, an welcher D den Unterschied zur Gleichverteilung bemerkt.
Nun werden wir zur besseren Veranschaulichung die Notation von f etwas modiifizieren, bevor wir den „Vorhersager" konstruieren:

o.B.d.A nehmen wir an, dass $S_i = \{1,...,l\}$ ist(durch umbenennen).

Dann ist $z \in \{0,1\}^t$ repräsentiert durch das Paar (x,y) mit $x = z_{|S_i} \in \{0,1\}^l$ und $y = z_{|[t]\setminus S_i} \in \{0,1\}^{t-l}$.

Für alle $j < i$ und $z = (x,y)$ definieren wir
$f_j(x,y) = f(z_{|S_j})$.

Also hängt $f_j(x,y)$ ab von
$|S_i \setminus S_j| \le \log m$ Bits von **x** und
$l - |S_i \setminus S_j| \ge l - \log m$ Bits von **y**.

Damit ergibt sich:

$$\frac{\varepsilon}{m} \le \Pr_{x,y,r_{i+1},...,r_m}[D(f_1(x,y),...,f_{i-1}(x,y),f(x),r_{i+1},...,r_m) = 1]$$
$$- \Pr_{x,y,r_i,...,r_m}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,r_{i+1},...,r_m) = 1]$$

Das bedeuted es ist wahrscheinlicher, dass **D´** die Eingabe $f_1(x,y),...,f_{i-1}(x,y),?,r_{i+1},...,r_m$ akzeptiert, wenn bei ? $f(x)$ steht, anstelle eines Zufallsbit.
Schluß: Wenn D´ akzeptiert, dann steht f(x) an Stelle i!

Damit gelingt es uns, einen Algorithmus zu formulieren, der mit Hilfe von D´ in der Lage ist, f(x) vorherzusagen:

Algorithmus A

Eingabe: $x \in \{0,1\}^l$.
Wähle zufallige $r_i,...r_m \in \{0,1\}$.
Wähle zufälliges $y \in \{0,1\}^{t-l}$.
Berechne $f_1(x,y),...,f_{i-1}(x,y)$.
Wenn $D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=1$ Ausgabe r_i,
sonst Ausgabe $1-r_i$.

Ist **A** erfolgreich?

Sei $A(x,y,r_1,...,r_m)$ die Ausgabe von **A** bei Eingabe von **x** und zufällig gewählten $y,r_1,...,r_m$.

Dann gilt für die Wahrscheinlichkeit, dass **A** den richtigen Wert vorhersagt:

$$\begin{aligned}
&\Pr_{x,y,r}[A(x,y,r)=f(x)] \\
&=\Pr_{x,y,r}[A(x,y,r)=f(x)\mid r_i=f(x)]\cdot\Pr_{x,r_i}[r_i=f(x)] \\
&+\Pr_{x,y,r}[A(x,y,r)=f(x)\mid r_i\neq f(x)]\cdot\Pr_{x,r_i}[r_i\neq f(x)] \\
&=\frac{1}{2}\Pr_{x,y,r}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=1\mid r_i=f(x)] \\
&+\frac{1}{2}\Pr_{x,y,r}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=0\mid r_i\neq f(x)] \\
&=\frac{1}{2}+\frac{1}{2}\left[\Pr_{x,y,r}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=1\mid f(x)=b]\right. \\
&\left.-\Pr_{x,y,r}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=1\mid f(x)\neq b]\right] \\
&=\frac{1}{2}+\Pr_{x,y,r}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=1\mid f(x)=b] \\
&-\frac{1}{2}\left[\Pr_{x,y,r}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=1\mid f(x)=b]\right. \\
&\left.+\Pr_{x,y,r}[D(f_1(x,y),...,f_{i-1}(x,y),r_i,...,r_m)=1\mid f(x)\neq b]\right] \\
&=\frac{1}{2}+\Pr[D(H_i)=1]-\Pr[D(H_{i-1})=1] \\
&\geq\frac{1}{2}+\frac{\varepsilon}{m}
\end{aligned}$$

Also:

$$\Pr_{x,y,r_1,...,r_m}[A(x,y,r)=f(x)]\geq\frac{1}{2}+\frac{\varepsilon}{m}$$

Der Algorithmus A ist also genauso erfolgreich, wie im Lemma behauptet.

Für das Lemma bleibt zu zeigen:

A ist effizient und entspricht $D(C)$.

Mit konstanten Werten $c_i,...,c_m$ anstelle $r_i,...,r_m$ und // anstelle von **y** gilt:

$$\Pr_x[A(x,\square, c_i,...,c_m) = f(x)] \geq \frac{1}{2} + \frac{\square}{m}$$

Es bleiben für **A** nur noch die $f_j(x,\square), j < i$ zu berechnen, welche je von $\square \log m$ Bits von **x** abhängen und also von Schaltkreisen der Größe $O(m)$ berechenbar sind.

Davon sind $i \square 1 < m$ Schaltkreise nötig um **A** zu berechnen. Somit hat **A** die Größe $O(m^2)$.

Funktionsweise:
Ausgabe von $c_i \oplus \overline{D(\square C(x))}$. Da c_i eine Konstante ist, gilt immer entweder $A(x,\square,c) = D(C(x))$ oder $A(x,\square,c) \neq D(C(x))$. Damit ist das Lemma bewiesen.