## 7.1 Schaltnetze

Eine formale Beschreibung von „Algorithmen“ kann man wie gesehen mit Hilfe von TURING-Maschinen geben. Ein einfacher zu definierendes, verstehendes und handzuhabendes Konzept ist das BOOLEsche **Schaltnetz**, das die in einem Algorithmus auszuführenden Bitoperationen in Form eines Ablaufdiagramms darstellt. Es wird auf zwei Weisen verallgemeinert:

**probabilistisches Schaltnetz** – damit werden probabilistische Algorithmen formalisiert,

**polynomiale Schaltnetzfamilie** – mit deren Hilfe kann die Komplexität eines Algorithmus bei wachsender Größe der Eingabedaten ausgedrückt werden.

Man kommt so zum Begriff der Schaltnetzkomplexität, der 1949 von SHANNON eingeführt wurde und für die Anwendung in der Kryptologie besonders gut geeignet ist.

Ein **Schaltnetz** ist ein azyklischer gerichteter markierter Graph, dessen Knoten sämtlich den Innengrad 0 oder 2 haben.

> Das heisst, Knoten sind mit gerichteten Kanten (Pfeilen) verbunden, wobei kein geschlossener Weg entsteht, alle Knoten sind mit Markierungen (Attributen) versehen, in jedem Knoten enden 0 oder 2 Pfeile.

Ein **Eingang** ist ein Knoten mit Innengrad 0. Ein **Ausgang** ist ein Knoten mit Außengrad 0; von ihm gehen also keine weiteren Pfeile aus. Alle Knoten sind mit einem der Symbole $\oplus$ oder $\otimes$ markiert – das entspricht der Addition und Multiplikation zweier Bits im Körper $\mathbb{F}_2$. Einige Eingänge sind als konstant gekennzeichnet; an ihnen liegt stets ein festes Bit 0 oder 1 an. (Da die Außengrade unbeschränkt sind, würden im Prinzip sogar zwei konstante Eingänge reichen, aber dann braucht man unelegant viele zusätzliche Pfeile.) Eine **Eingabe** ist eine Belegung der $r$ nichtkonstanten Eingänge mit einer Bitfolge $x = (x_1, \ldots, x_r) \in \mathbb{F}_2^r$. An den inneren Knoten („Gattern“) werden die anliegenden Bits dann jeweils $\bmod 2$ addiert oder multipliziert, je nach Markierung des Knotens. Am Ausgang ensteht nach Durchlaufen des gesamten Schaltnetzes die **Ausgabe** $y \in \mathbb{F}_2^s$. Das Schaltnetz definiert also eine Funktion

$$C\colon \mathbb{F}_2^r \longrightarrow \mathbb{F}_2^s$$

und gibt deren algorithmische Zerlegung in Bitoperationen wieder. Durch ein Schaltnetz ausdrücken lässt sich jede solche Funktion, die nur durch Addition und Multiplikation gebildet werden kann, also jedes Polynom, also nach II (Einschub über BOOLEsche Abbildungen) jede Abbildung $\mathbb{F}_2^r \longrightarrow \mathbb{F}_2^s$. Im Fall $r = 2$, $s = 1$ gibt es

$$(\#\mathbb{F}_2)^{\#(\mathbb{F}_2 \times \mathbb{F}_2)} = 2^4 = 16$$

derartige Funktionen. Sie sind alle in Tabelle 1 aufgezählt, aus der man auch leicht die entsprechenden Schaltnetze entnehmen kann; sie werden als Polynome $a+bX+cY+dXY \in \mathbb{F}_2[X,Y]$ beschrieben (algebraische Normalform). Allgemein werden Schaltnetz und zugehörige Funktion, da Verwirrung nicht zu befürchten ist, mit dem gleichen Buchstaben bezeichnet.

| $a$ | $b$ | $c$ | $d$ | Polynom (ANF) | logische Operation | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | FALSE | Konstante |
| 1 | 0 | 0 | 0 | 1 | TRUE | Konstante |
| 0 | 1 | 0 | 0 | $X$ | $X$ | Projektion |
| 1 | 1 | 0 | 0 | $1+X$ | $\neg X$ | negierte Projektion |
| 0 | 0 | 1 | 0 | $Y$ | $Y$ | Projektion |
| 1 | 0 | 1 | 0 | $1+Y$ | $\neg Y$ | negierte Projektion |
| 0 | 1 | 1 | 0 | $X+Y$ | $X \oplus Y$ | XOR |
| 1 | 1 | 1 | 0 | $1+X+Y$ | $X \Longleftrightarrow Y$ | Äquivalenz |
| 0 | 0 | 0 | 1 | $XY$ | $X \wedge Y$ | AND |
| 1 | 0 | 0 | 1 | $1+XY$ | $\neg(X \wedge Y)$ | NAND |
| 0 | 1 | 0 | 1 | $X+XY$ | $X \wedge (\neg Y)$ | |
| 1 | 1 | 0 | 1 | $1+X+XY$ | $X \Longrightarrow Y$ | Implikation |
| 0 | 0 | 1 | 1 | $Y+XY$ | $(\neg X) \wedge Y$ | |
| 1 | 0 | 1 | 1 | $1+Y+XY$ | $X \Longleftarrow Y$ | Implikation |
| 0 | 1 | 1 | 1 | $X+Y+XY$ | $X \vee Y$ | OR |
| 1 | 1 | 1 | 1 | $1+X+Y+XY$ | $\neg(X \vee Y)$ | NOR |

Tabelle 1: Die 16 zweistelligen Bitoperationen