Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Dozent: Prof. Berghammer
Sommersemester 2012

BDD - Binary Decision Diagrams

Mitschrift von Sandra Dylus

Letzte Aktualisierung: 6. Juli 2012

Kontaktadresse: sad@informatik.uni-kiel.de
Hinweis: Keine Garantie auf Richtigkeit, bei Fehlerfindung bitte kontaktieren.

Inhaltsverzeichnis

1 Grundlagen

1.1 Aussagenlogik

Was ist eine Aussage? - Ein sprachliches Gebilde, von dem es Sinn macht, zu sagen, es sei wahr oder falsch.

Beispiel
Kiel liegt an der Ostsee. Kiel liegt an der Nordsee.

Beispiel
$x \in A$, $X = Y$, $\exists x : A(x)$, $\forall x : B(x)$

Kurze Wiederholung von Junktoren:

1. Negation: $\neg A$
2. Konjunktion: $A \wedge B$
3. Disjunktion: $A \vee B$
4. Implikation: $A \rightarrow B$
5. Äquivalenz: $A \leftrightarrow B$

1.1.1 Definition

Sei X eine Menge von Aussagenvariablen . Dann ist die Menge $\mathcal{A}(X)$ der <u>aussagenlogischen Formeln</u> über X wie folgt definiert:

1. Für alle $a \in X$ gilt $a \in \mathcal{A}(X)$
2. Für alle $\phi \in \mathcal{A}(X)$ gilt $(\neg\phi) \in \mathcal{A}(X)$
3. Für alle $\phi, \psi \in \mathcal{A}(X)$ gelten
   - $(\phi \wedge \psi) \in \mathcal{A}(X)$
   - $(\phi \vee \psi) \in \mathcal{A}(X)$
   - $(\phi \rightarrow \psi) \in \mathcal{A}(X)$
   - $(\phi \leftrightarrow \psi) \in \mathcal{A}(X)$
4. Es gibt keine Elemente in $\mathcal{A}(X)$, außer denen, die (1.) bis (3.) zulassen.

$\square$

Die Negation ($\neg$) bindet am stärksten, danach die Konjunktion ($\wedge$) und Disjunktion ($\vee$) und zuletzt die Implikation ($\rightarrow$) sowie die Äquivalenz ($\leftrightarrow$) (Vorrangsregeln).

Beispiel
$X = \{a, b, c, d, e\}$
$\underbrace{(\underbrace{((\underbrace{\neg a}) \vee (\underbrace{\neg b}))} \vee ((\underbrace{c \wedge d}) \wedge e)}$)
$\rightsquigarrow (\neg a \vee \neg b) \vee ((c \wedge d) \wedge e)$
$\rightsquigarrow \neg a \vee \neg b \vee (c \wedge d \wedge e)$

1.1.2 Definition

Es ist $\mathbb{B} := \{\mathbf{0}, \mathbf{1}\}$ die Menge der Wahrheitswerte. $\square$

1.1.3 Definition
Eine <u>Belegung</u> ist eine Funktion $v : X \to \mathbb{B}$. Zu $a \in X$ heißt $v(a)$ die <u>Belegung</u> von a. □

1.1.4 Definition
Die Funktionen $^-: \mathbb{B} \to \mathbb{B}, +, \cdot : \mathbb{B} \to \mathbb{B}$ sind durch die folgende Tafeln festgelegt:

a	0	1
$\overline{a}$	1	0

$+$				
a	0	0	1	1
b	0	1	0	1
$a+b$	0	1	1	1

$\cdot$				
a	0	0	1	1
b	0	1	0	1
$a \cdot b$	0	0	0	1

1.1.5 Definition
Zu einer Belegung $v : X \to \mathbb{B}$ ist der Wert $\underline{val}_v(\phi)$ für $\phi \in \mathcal{A}(X)$ induktiv wie folgt festgelegt:

1. $\underline{val}_v(a) = v(a)$ für $a \in X$
2. $\underline{val}_v(\neg\phi) = \overline{\underline{val}_v(\phi)}$ für $\phi \in \mathcal{A}(X)$
3. $\underline{val}_v(\phi \vee \psi) = \underline{val}_v(\phi) + \underline{val}_v(\psi)$ für alle $\phi, \psi \in \mathcal{A}(X)$
4. $\underline{val}_v(\phi \wedge \psi) = \underline{val}_v(\phi) \cdot \underline{val}_v(\psi)$ für alle $\phi, \psi \in \mathcal{A}(X)$
5. $\underline{val}_v(\phi \to \psi) = \overline{\underline{val}_v(\phi)} + \underline{val}_v(\psi)$ für alle $\phi, \psi \in \mathcal{A}(X)$
6. $\underline{val}_v(\phi \leftrightarrow \psi) = \underline{val}_v(\phi \to \psi) \cdot \underline{val}_v(\psi \to \phi)$ für alle $\phi, \psi \in \mathcal{A}(X)$

□

1.1.6 Definition
Die Relation $\Leftrightarrow \subseteq \mathcal{A}(X) \times \mathcal{A}(X)$ ist für alle $\phi, \psi \in \mathcal{A}(X)$ definiert durch:

$$\phi \Leftrightarrow \psi :\Longleftrightarrow \text{ für alle } v \in \mathbb{B}^X \text{ gilt } \underline{val}_v(\phi) = \underline{val}_v(\psi)$$

ϕ und ψ sind <u>logisch äquivalent</u>, falls $\phi \Leftrightarrow \psi$ gilt. □

1.1.7 Definition
Die Relation $\Rightarrow \subseteq \mathcal{A}(X) \times \mathcal{A}(X)$ ist für alle $\phi, \psi \in \mathcal{A}(X)$ definiert durch:

$$\phi \Rightarrow \psi :\Longleftrightarrow \text{ für alle } v \in \mathbb{B}^X \text{ gilt, wenn } \underline{val}_v\phi = 1, \text{ dann } \underline{val}_v\psi = 1$$

ϕ <u>impliziert logisch</u> ψ, falls $\phi \Rightarrow \psi$ gilt. □

1.1.8 Satz
Für alle $\phi, \psi \in \mathcal{A}(X)$ gelten:

1. $\phi \Leftrightarrow \psi$ ist äquivalent zu $\underline{val}_v(\phi \leftrightarrow \psi) = 1$ für alle $v \in \mathbb{B}^X$
2. $\phi \Rightarrow \psi$ ist äquivalent zu $\underline{val}_v(\phi \to \psi) = 1$ für alle $v \in \mathbb{B}^X$

□

1.1.9 Satz

1. $\Leftrightarrow$ ist Äquivalenzrelation auf $\mathcal{A}(X)$, d.h. reflexiv, symmetrisch und transitiv
2. $\Rightarrow$ ist Quasiordnung auf $\mathcal{A}(X)$, d.h. reflexiv und transitiv
3. für alle $\phi, \psi, \rho \in \mathcal{A}(X)$ gilt: $\phi \Rightarrow \psi$ und $\psi \Rightarrow \rho$ impliziert $\phi \Rightarrow \rho$
4. für alle $\phi, \psi \in \mathcal{A}(X)$ gilt $\phi \Leftrightarrow \psi$ genau dann, wenn $\phi \Rightarrow \psi$ und $\psi \Rightarrow \phi$ gelten

□

1.1.10 Beispiel
Seien $a, b, c \in X$. Dann gilt:

$$\begin{aligned} a \to (b \to c) &\overset{Def.}{\Leftrightarrow} \neg a \vee (b \to c) \\ &\overset{Def.}{\Leftrightarrow} \neg a \vee (\neg b \vee c) \\ &\overset{Ass.}{\Leftrightarrow} (\neg a \vee \neg b) \vee c \\ &\overset{DeM.}{\Leftrightarrow} \neg(a \wedge b) \vee c \\ &\overset{Def.}{\Leftrightarrow} (a \wedge b) \to c \end{aligned}$$

□

1.2 Boolesche Algebra

1.2.1 Definition
Ein Verband ist eine algebraische Struktrur $(V, \sqcap, \sqcup)$ mit $V \neq \emptyset$ und $\sqcap, \sqcup : V \times V \to V$, so dass für alle $x, y \in V$ gilt:

1. $x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z$
2. $x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$
3. $x \sqcup y = y \sqcup x$
4. $x \sqcap y = y \sqcap x$
5. $x \sqcap (y \sqcup x) = x$
6. $x \sqcup (y \sqcap x) = x$

□

1.2.2 Beispiel
1. $(\mathbb{B}, \cdot, +)$ ist Verband
2. $(2^M, \cap, \cup)$ ist Verband
3. $(\mathbb{N}, \underline{ggT}, \underline{kgV})$ ist Verband
4. $(\mathbb{N}, \underline{min}, \underline{max})$ ist Verband

□

1.2.3 Satz
Es sei $(V, \sqcap, \sqcup)$ ein Verband. Dann gelten für alle $x, y \in V$ die folgenden Eigenschaften:

1. $x \sqcup x = x$ und $x \sqcap x = x$
2. $x \sqcap y = x \Leftrightarrow x \sqcup y = y$

Beweis

(1)

$$\begin{aligned} x \sqcup x &= x \sqcup (x \sqcap (x \sqcup x)) \\ &= x \sqcup ((x \sqcup x) \sqcap x) \\ &= x \end{aligned}$$

$$\begin{aligned} x \sqcap x &= x \sqcap (x \sqcup (x \sqcap x)) \\ &= x \sqcap ((x \sqcap x) \sqcup x) \\ &= x \end{aligned}$$

(2) $\Rightarrow$ Sei $x \sqcap y = x$, dann

$$\begin{aligned} x \sqcup y &= (x \sqcap y) \sqcup y \\ &= y \sqcup (x \sqcap y) \\ &= y \end{aligned}$$

$\Leftarrow$ Sei $x \sqcup y = y$, dann

$$\begin{aligned} x \sqcap y &= x \sqcap (x \sqcup y) \\ &= x \sqcap (y \sqcup x) \\ &= x \end{aligned}$$

$\square$

1.2.4 Satz
Es sei $(V, \sqcap, \sqcup)$ ein Verband. Definiert man auf V eine Relation $\sqsubseteq \subseteq V \times V$ für alle $x, y \in V$ durch $x \sqsubseteq y :\Longleftrightarrow x \sqcap y = x (\Longleftrightarrow x \sqcup y = y$, so ist $(V, \sqsubseteq)$ eine geordnete Menge.

Beweis
Reflexivität: Sei $x \in V$

$$x \sqsubseteq x \overset{Def.}{\Leftrightarrow} x \sqcap x = x \overset{Satz 1.2.3(1)}{\Leftrightarrow} \underline{wahr}$$

Antisymmetrie: Seien $x, y \in V$

$$x \sqsubseteq y \wedge y \sqsubseteq x \overset{Def.}{\Leftrightarrow} x \sqcap y = x \wedge y \sqcap x = y \overset{Komm.}{\Leftrightarrow} x = y$$

Transitivität: Seien $x, y, z \in V$

$$\begin{aligned} x \sqsubseteq y \wedge y \sqsubseteq z &\overset{Def.}{\Leftrightarrow} & x \sqcap y = x \wedge y \sqcap z = y \\ &\overset{Vor.+Ass.}{\Rightarrow} & x \sqcap z = (x \sqcap y) \sqcap z = x \sqcap (y \sqcap z) = x \sqcap y = x \\ &\overset{Def.}{\Leftrightarrow} & x \sqsubseteq z \end{aligned}$$

$\square$

Beispiel
1. $(V, \sqcap, \sqcup, \sqsubseteq)$
2. $(\mathbb{B}, \cdot, +, \Rightarrow)$
3. $(2^M, \cap, \cup, \subseteq)$
4. $(\mathbb{N}, \underline{ggT}, \underline{kgV}, \backslash)$
5. $(\mathbb{N}, \underline{min}, \underline{max}, \leq)$

1.2.5 Definition
Ein Verband $(V, \sqcap, \sqcup)$ heißt distributiv, falls für alle $x, y, z \in V$ gelten:

1. $x \sqcup (y \sqcap z) = (x \sqcup y) \sqcap (x \sqcup z)$
2. $x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$

$\square$

1.2.6 Definition

Eine Boolesche Algebra ist eine algebraische Struktur $(V, \sqcap, \sqcup, ^{-}, \mathbf{0}, \mathbf{L})$ mit $\sqcap, \sqcup : V \times V \to V, ^{-} : V \to V$, $\mathbf{0}, \mathbf{L} \in V$, so dass folgende Eigenschaften gelten:

1. $(V, \sqcap, \sqcup)$ ist ein Verband
2. für alle $x \in V$ gilt $x \sqcap \neg x = \mathbf{0}$ und $x \sqcup \neg x = \mathbf{L}$

□

Daraus kann man Folgendes observieren:

$$\begin{aligned} x \sqsubseteq \mathbf{L} \quad &\Leftrightarrow \quad x \sqcup \mathbf{L} = \mathbf{L} \\ &\Leftrightarrow \quad x \sqcup (x \sqcup \neg x) = x \sqcup \neg x \\ &\Leftrightarrow \quad (x \sqcup x \sqcup) \neg x = x \sqcup \neg x \\ &\Leftrightarrow \quad x \sqcup \neg x = x \sqcup \neg x \end{aligned}$$

$$\begin{aligned} x \sqsubseteq \mathbf{0} \quad &\Leftrightarrow \quad \mathbf{0} \sqcap x = \mathbf{0} \\ &\Leftrightarrow \quad (x \sqcap \neg x) \sqcap x = x \sqcap \neg x \\ &\Leftrightarrow \quad x \sqcap \neg x = x \sqcap \neg x \end{aligned}$$

Also ist $\mathbf{L}$ größtes Element und $\mathbf{0}$ kleinstes Element von V in $(V, \sqsubseteq)$.

	Boolesche Algebra	Aussagenlogik
Trägermenge	V	$\mathcal{A}(X)$
Operationen	$\sqcap$	$\wedge$
	$\sqcup$	$\vee$
	$^{-}$	$\neg$
Elemente	$\mathbf{O}$	falsch
	$\mathbf{L}$	wahr
Ordnung	$\sqsubseteq$	$\rightarrow$
Gleichheit	$=$	$\leftrightarrow$

Tabelle 1: Vergleich der Booleschen Algebra und Aussagenlogik

1.2.7 Satz

Sei $(V, \sqcap, \sqcup, ^{-}, \mathbf{O}, \mathbf{L})$ eine Boolesche Algebra. Dann gelten für alle $x, y \in V$ folgende Eigenschaften:

1. $\overline{\overline{x}} = x$
2. $\overline{x \sqcap y} = \overline{x} \sqcup \overline{y}$ und $\overline{x \sqcup y} = \overline{x} \sqcap \overline{y}$
3. $x \sqsubseteq y \Leftrightarrow \neg x \sqcap y = \mathbf{L} \Leftrightarrow x \sqcap \neg y = \mathbf{0}$
4. $\overline{\mathbf{0}} = \mathbf{L}$ und $\overline{\mathbf{L}} = \mathbf{0}$

□

1.2.8 Satz

Gilt eine Gleichung $t_1 = t_2$ in allen Booleschen Verbänden, so gilt auch $t_1^d = t_2^d$, wobei t_i^d aus t_i dadurch entsteht, dass man $\sqcap$ und $\sqcup$ sowie $\mathbf{0}$ und $\mathbf{L}$ vertauscht. □

1.2.9 Definition

Es sei $(V, \sqcap, \sqcup, ^{-}, \mathbf{0}, \mathbf{L})$ eine Boolesche Algebra, dann heißt $a \in V$ Atom, falls $a \neq \mathbf{0}$ und für alle $x \in V$ gilt $x \sqsubseteq a \Rightarrow x = \mathbf{0} \vee x = a$. $\mathcal{A}t(V)$ sei die Menge der Atome von V.

□

Daraus folgern wir: falls $|V| < \infty$ und $|V| \geq 2$, dann gilt $\mathcal{A}t(V) \neq \mathbf{0}$ und für alle $x \in V$ gibt es $a \in \mathcal{A}t(V)$ mit $a \sqsubseteq x$.

1.2.10 Lemma
In einer Booleschen Algebra $(V, \sqcap, \sqcup, ^{-}, \mathbf{0}, \mathbf{L})$ gelten die folgenden Eigenschaften:

1. für alle $a, b \in \mathcal{A}t(V)$ mit $a \neq b$ gilt $a \sqcap b = \mathbf{0}$
2. für alle $a \in \mathcal{A}t(V)$ mit $x \in V$ gilt $a \not\sqsubseteq x \Rightarrow a \sqsubseteq \overline{x}$
3. für alle $a \in \mathcal{A}t(V)$ und $x_1, \ldots, x_n \in V, n \geq 1$ gilt $a \sqsubseteq \bigsqcup_{i=1}^{n} x_i \Leftrightarrow \exists j \in \{1, \ldots, n\} : a \sqsubseteq x_j$

Beweis

(1) Angenommen $a \sqcap b \neq \mathbf{0}$. Dann gilt $a \sqcap b = a$, da $a \sqcap b \sqsubseteq a$, und $a \sqcap b = b$, da $a \sqcap b = b$. Also gilt $a = b$ und das widerspricht $a \neq b$.
(2) $a \not\sqsubseteq x \Leftrightarrow a \sqcap \overline{x} \neq a$.
Da $a \sqcap \overline{x} \sqsubseteq a$ folgt $a \sqcap \overline{x} = \mathbf{0}$. Also $a \sqsubset \overline{x}$ nach Satz 1.2.7 (3).
(3)

$$
\begin{aligned}
\nexists j \in \{1, \ldots, n\} : a \sqsubseteq x_j \quad &\Leftrightarrow \quad \forall j \in \{1, \ldots, n\} : a \not\sqsubseteq x_j \\
&\Rightarrow \quad \forall j \in \{1, \ldots, n\} : a \sqsubseteq \overline{x_j} \\
&\Rightarrow \quad a \sqcap \bigsqcup_{i=1}^{n} x_i = \bigsqcup_{i=1}^{n} (a \sqcap x_i) = \bigsqcup_{i=1}^{n} \mathbf{0} \\
&\Rightarrow \quad a \not\sqsubseteq \bigsqcup_{i=1}^{n} x_i \quad \text{da } a \in \mathcal{A}t(V)
\end{aligned}
$$

1.2.11 Satz
Ist $(V, \sqcap, \sqcup, ^{-}, \mathbf{0}, \mathbf{L})$ eine endliche Boolesche Algebra, so ist die Funktion

$$f : V \to 2^{\mathcal{A}t(V)}, \qquad f(x) = \{a \in \mathcal{A}t(V) | a \sqsubseteq x\}$$

eine bijektive Funktion. Also gilt $|V| = 2^{|\mathcal{A}t(V)|}$.

Beweis

Injektivität: Seien $x, y \in V$ mit $x \neq y$, z.z. ist $f(x) \neq f(y)$. Aus $x \neq y$ folgt ($x \not\sqsubseteq y$ oder $y \not\sqsubseteq x$). Es gelte o.B.d.A. $x \not\sqsubseteq y$. Dann gilt $x \sqcap \overline{y} \neq \mathbf{0}$. Also gibt es, da $|V| < \infty$, ein Atom $a \in \mathcal{A}t(V)$ mit $a \sqsubseteq x \sqcap \overline{y}$.

(1) $a \sqsubseteq x$ (2) $a \sqsubseteq \overline{y}$

Aus (1) und $a \in \mathcal{A}t(V)$ folgt $a \in f(x)$. Aus (2) folgt $a \not\sqsubseteq y$. Wäre $a \subseteq y$, dann golte $a \sqsubseteq \overline{y}$ und $a \subseteq y$, also $a \sqsubseteq y \sqcap \overline{y} = \mathbf{0}$, daraus folgt wiederum $a = \mathbf{0}$, dies steht aber im Widerspruch zu $a \in \mathcal{A}t(V)$. Also gilt $a \notin f(y)$.
Da $a \in f(x)$ und $a \notin f(y)$ gilt $f(x) \neq f(y)$.
Surjektivität: Sei $A \in 2^{\mathcal{A}t(V)}$. Da $|V| < \infty$ gibt es $a_1, \cdots a_n \in \mathcal{A}t(V)$ mit $A = \{a_1, \cdots, a_n$.
Beh.: $f(\bigsqcup_{i=1}^{n} a_i) = \{a_1, \cdots, a_n\}$ für $n \geq 1$ und $f(\mathbf{0}) = \emptyset$ für $n = 0$
Bew.: $\subseteq$
Sei $b \in f(\bigsqcup_{i=1}^{n} a_i)$, d.h. $b \in \mathcal{A}t(V)$ unf $b \sqsubseteq \bigsqcup_{i=1}^{n} a_i$. Lemma 1.2.10 (3) zeigt, dass es $j \in \{1, \cdots, n\}$ mit $b \sqsubseteq a_j$ gibt. Wegen $b \in \mathcal{A}t(V)$ gilt $b = a_j$, also $b \in A$.
$\supseteq$
Sei $b \in A$, d.h. $b = a_j$ für ein $j \in \{1, \ldots, n\}$. Dann gilt $b \in \mathcal{A}t(V)$ und $b \sqsubseteq \bigsqcup_{i=1}^{n} a_i$ (da $b \sqcup \bigsqcup_{i=1}^{n} a_i = a_j \sqcup \bigsqcup_{i=1}^{n} a_i = \bigsqcup_{i=1}^{n} a_i$). Die Definition von f bringt letztendlich $b \in f(\bigsqcup_{i=1}^{n} a_i)$.

$\square$

Weiterhin gelten für f noch folgene Eigenschaften:

1. $f(x \sqcup y) = f(x) \cup f(y)$
2. $f(x \sqcap y) = f(x) \cap (fy)$
3. $f(\overline{x}) = \overline{f(x)} = \mathcal{A}t(V) \setminus f(x)$
4. $f(\mathbf{0}) = \emptyset$
5. $f(\mathbf{L}) = \mathcal{A}t(V)$

1.3 Boolesche Funktionen

1.3.1 Definition
Ist $(V, \sqcap, \sqcup, ^{-}, \mathbf{0}, \mathbf{L})$ eine Boolesche Algebra und $n \in V$ mit $n \geq 1$, dann ist V_n die Menge der n-stelligen Funktionen $f : V^n \to V$. □

1.3.2 Definition
Zu $(\mathbb{B}, \cdot, +, ^{-}, \mathbf{0}, \mathbf{L})$ und $n \geq 1$ heißt $f \in \mathbb{B}_n$ <u>Boolesche Funktion</u> (Schaltfunktion). □

1.3.3 Satz
Ist $(V, \sqcap, \sqcup, ^{-}, \mathbf{0}, \mathbf{L})$ eine Boolesche Algebra, so wird auch $(V_n, \widetilde{\sqcap}, \widetilde{\sqcup}, \widetilde{^{-}}, \widetilde{\mathbf{0}}, \widetilde{\mathbf{L}})$ zu einer Booleschen Algebra, indem man definiert:
$\widetilde{\mathbf{0}} : V^n \to V \qquad \widetilde{\mathbf{0}}(x_1, \ldots, x_n) = \mathbf{0}$
$\widetilde{\mathbf{L}} : V^n \to V \qquad \widetilde{\mathbf{L}}(x_1, \ldots, x_n) = \mathbf{L}$
und für alle $f : V^n \to V$ Funktionen $f \widetilde{\sqcap} g : V^n \to V$, $f \widetilde{\sqcup} g : V^n \to V$, $\widetilde{\overline{f}} : V^n \to V$ definiert durch:
$(f \widetilde{\sqcap} g)(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \sqcap g(x_1, \ldots, x_n)$
$(f \widetilde{\sqcup} g)(x_1, \ldots, x_n) = f(x_1, \ldots, x_n) \sqcup g(x_1, \ldots, x_n)$
$\widetilde{\overline{f}}(x_1, \ldots, x_n) = \overline{f(x_1, \ldots, x_n)}$

Beweis
Seien $f, g : V^n \to V$, dann:

$$\begin{aligned}
& f \sqcap g = g \sqcap f \\
\Leftrightarrow\ & \forall x_1, \ldots, x_n \in V : (f \sqcap g)(x_1, \ldots, x_n) = (g \sqcap f)(x_1, \ldots, x_n) \\
\Leftrightarrow\ & \forall x_1, \ldots, x_n) \in V : f(x_1, \ldots, x_n) \sqcap g(x_1, \ldots, x_n) = g(x_1, \ldots, x_n) \sqcap f(x_1, \ldots, x_n) \\
\Leftrightarrow\ & \forall x_1, \ldots, x_n \in V : \underline{\text{wahr}}
\end{aligned}$$

Rest analog. □

Die Mächtigkeit von V_n sei kurz festzuhalten: $|V_n| = |V|^{(|V|^n)} = (2^a)^b = 2^{a \cdot 2^{a \cdot n}}$

Des Weiteren schauen wir uns kurz Boolesche Funktionen auf $\mathbb{B}$ an. Dabei ist $\mathbf{0} : \mathbb{B}^n \to \mathbb{B}$ eine Kontraktion und $\mathbf{L} : \mathbb{B}^n \to \mathbb{B}$ eine Tautologie. Weiter gilt $+ : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ mit $+ \in \mathbb{B}_2$, $\cdot : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ mit $\cdot \in \mathbb{B}_2$ und $^{-} : \mathbb{B} \to \mathbb{B}$ mit $^{-} \in \mathbb{B}$. Allgemein gilt $f : \mathbb{B}^n \to \mathbb{B}, f(x_1, \ldots, x_n) = t$, wobei t mittels $x_1, \ldots, x_n, +, \cdot$ und $^{-}$ aufgebaut ist.

1.3.4 Definition
Seien $n \geq 1$ und $x := (x_1, \ldots, x_n), a := (a_1, \ldots, a_n)$ Vektoren aus $\mathbb{B}^n$. Dann definiert man:

1. $x_i^{a_i} = x_i$, falls $a_i = 1$ (positives Literal)
2. $x_i^{a_i} = \overline{x_i}$, falls $a_i = 0$ (negatives Literal)
3. $m_a(x) = \prod_{i=1}^{n} x_i^{a_i}$ ist Minterm x bzgl. a

□

1.3.5 Satz
Für alle $f \in \mathbb{B}_n$ mit $n \geq 1$ und $f \neq \mathbf{0}$ gilt:

$$f(x_1, \ldots, x_n) = \sum_{a \in f^{-1}(1)} m_a(x) \qquad \text{DNF}$$

Falls $f(x_1, \ldots, x_n) = \sum_{a \in f^{-1}(1)} m_a(x) = \mathbf{0}$ definiert, dann ist $f \neq 0$ nicht wahr. □

1.3.6 Beispiel
$f : \mathbb{B}^3 \to \mathbb{B}, f(x, y, z) \mathrel{\widehat{=}}$ "$x + y + z$"gerade

x	0	0	0	0	1	1	1	1
y	0	0	1	1	0	0	1	1
z	0	1	0	1	0	1	0	1
$f(x, y, z)$	1	0	0	1	0	1	1	0

$f^{-1}(1) = \{(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0)\}$
$f(x, y, z) = \bar{x} \cdot \bar{y} \cdot \bar{z} + \bar{x} \cdot y \cdot z + x \cdot \bar{y} \cdot z + x \cdot y \cdot \bar{z}$

Im Folgenden werden wir das Nicod-Nor und Sheffer-Nand verwenden.

1. $\nabla \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ mit $x \nabla y = \overline{x + y}$
2. $\triangle \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ mit $x \triangle y = \overline{x \cdot y}$

1.3.7 Satz 1. Jede Boolesche Funktion $f : \mathbb{B}^n \to \mathbb{B}$ ist durch ∇ (bzw. durch $\triangle$) darstellbar.

2. ∇ und $\triangle$ sind die einzigen Funktionen aus $\mathbb{B}_2$, mit denen man alle $f : \mathbb{B}^n \to \mathbb{B}$ darstellen kann.

Beweis

(1) Es genügt $+, \cdot, ^-$ darzustellen, dann folgt die Behauptung aus Satz 1.3.5 (DNF).

$$x + y = \overline{\overline{x + y}} = \overline{\bar{x} \cdot \bar{y}} = \bar{x} \triangle \bar{y} = (x \triangle x) \triangle (y \triangle y) \tag{1}$$
$$x \cdot y = \overline{\overline{x \cdot y}} = \overline{\bar{x} + \bar{y}} = \overline{x \triangle y} = (x \triangle y) \triangle (x \triangle y) \tag{2}$$
$$\bar{x} = \overline{x \cdot x} = x \triangle x \tag{3}$$

Analog für ∇.
(2) Sei $\otimes : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$, so dass jede Funktion $f : \mathbb{B}^n \to \mathbb{B}$ nur durch $\otimes$ und Variablen darstellbar ist.
$0 \otimes 0 = 0$ erlaubt nicht, $^-$ darzustellen, da $\bar{0} = 1$.
$1 \otimes 1 = 1$ erlaubt nicht, $^-$ darzustellen, da $\bar{1} = 0$.
Also muss $0 \otimes 0 = 1$ sowie $1 \otimes 1 = 0$ gelten.

Die restlichen Auswertungen erfordern eine Fallunterscheidung:
Fall 1: $0 \otimes 1 = 1$ und $1 \otimes 0 = 0$, dann wäre $x \otimes y = \bar{x}$
Fall 2: $0 \otimes 1 = 0$ und $1 \otimes 0 = 1$, dann wäre $x \otimes y = \bar{y}$
Somit können Fall 1 und Fall 2 nicht auftreten.

Fall 3: $0 \otimes 1 = 0$ und $1 \otimes 0 = 0$, dann wäre $\otimes = \nabla$
Fall 4: $0 \otimes 1 = 1$ und $1 \otimes 0 = 1$, dann wäre $\otimes = \triangle$

Insgesamt sind also folgende Wertetafeln möglich:

x	0	0	1	1
y	0	1	0	1
$x \otimes y$	1	0	0	0

x	0	0	1	1
y	0	1	0	1
$x \otimes y$	1	1	1	0

□

Sei $X = \{x_1, \dots, x_n\}$ Menge von Booleschen Variablen.

a) Zu jedem $\phi \in \mathcal{A}(X)$ gibt es eine Boolesche Funktion $f_\phi : \mathbb{B}^n \to \mathbb{B}$, so dass für alle Belegungen $V : X \to \mathbb{B}$ gilt:
$$f_\phi(v(x_1), \dots, v(x_n)) = \underline{val}_v(\phi)$$

b) Zu jeder Booleschen Funktion $f : \mathbb{B}^n \to \mathbb{B}$ gibt es $\phi_f \in \mathcal{A}(X)$, so dass für alle Belegungen $V : X \to \mathbb{B}$ gilt:
$$f(v(x_1), \dots, v(x_n)) = \underline{val}_v(\phi_f)$$

1.3.8 Beispiel
Sei $X = \{x, y, z\}$.
$\phi \in \mathcal{A}(X) : \neg x \vee \neg(\neg y \vee z) \vee \neg x$

Durch Aufstellen der Wertetafel ergibt sie wie folgt:

a	0	0	0	0	1	1	1	1
b	0	0	1	1	0	0	1	1
c	0	1	0	1	0	1	0	1
$f_\phi(a,b,c)$	1	1	1	1	0	0	1	0

Und daraus ergibt sich dann die Funtkion $f_\phi(a,b,c) = \bar{a} + (\overline{\bar{b} + c}) + \bar{a} = \bar{a} + b \cdot \bar{c}$.

Sei $V : X \to \mathbb{B}$ Belegung. Dann

$$\begin{aligned}
\underline{val}_v(\phi) &= \underline{val}_v(\neg x \vee \neg(\neg y \vee z) \vee \neg x) \\
&= \underline{val}_v(\neg x) + \underline{val}_v(\neg(\neg y \vee z)) + \underline{val}_v(\neg x) \\
&= \overline{v(x)} + \underline{val}_v(\overline{\neg y \vee z}) + \overline{v(z)} \\
&= \overline{v(x)} + \overline{\overline{v(y)} + v(z)} + \overline{v(x)} \\
&= f_\phi(v(x), v(y), v(z))
\end{aligned}$$

<u>Grundprobleme</u>

1. Erfüllbarkeit: Gegeben sei $\phi \in \mathcal{A}(X)$, gibt es $V : X \to \mathbb{B}$ mit $\underline{val}_v(\phi) = 1$
Äquivalent dazu: Gilt $f_\phi \neq \mathbf{0}$

2. Berechnung: Gegeben sei $\phi \in \mathcal{A}(X)$ und $V : X \to \mathbb{B}$. Bestimme $\underline{val}_v(\phi)$.
Äquivalent dazu: $f_\phi(v(x_1), \dots, v(x_n))$

3. Äquivalenztest: Gegeben sei $\phi, \psi \in \mathcal{A}(X)$, gilt $\phi \Leftrightarrow \psi$
Äquivalent dazu: $f_\phi = f_\psi$

4. Erfüllende Belegung: Gegeben sei $\phi \in \mathcal{A}(X)$, bestimme $\{v \in \mathbb{B}^X | \underline{val}_v \phi = 1\}$
Äquivalent dazu: $f_\phi^{-1}(1)$

5. Anzahl der erfüllenden Belegungen: Gegeben sei $\phi \in \mathcal{A}(X)$, bestimme $|\{v \in \mathbb{B}^X | \underline{val}_v(\phi) = 1\}|$
Äquivalent dazu: $|f_\phi^{-1}(1)|$

1.4 Einige spezielle Konstruktionen

Definiere wie folgt: $\mathbb{B}_n$ Menge, $f : \mathbb{B}^n \to \mathbb{B}$
$\mathbf{0} : \mathbb{B}^n \to \mathbb{B}$, $\mathbf{L} : \mathbb{B}^n \to \mathbb{B}$
$\sqcap : \mathbb{B}_n \times \mathbb{B}_n \to \mathbb{B}_n$, $\sqcup : \mathbb{B}_n \times \mathbb{B}_n \to \mathbb{B}_n$, $^- : \mathbb{B}_n \to \mathbb{B}_n$.
Dann ist $(\mathbb{B}_n, \sqcap, \sqcup, ^-, \mathbf{0}, \mathbf{L})$ eine Boolesche Algebra.

1.4.1 Definition
Es sei $f : \mathbb{B}^n \to \mathbb{B}$ Boolesche Funktion. Eine Varibale x_i heißt <u>wesentlich</u>[1], falls es $a_1, \dots, a_{n-1} \in \mathbb{B}$ gibt mit $f(a_1, \dots, a_{i-1}, 1, a_i, \dots, a_{n-1}) \neq f(a_1, \dots, a_{i-1}, 0, a_i, \dots, a_{n-1})$. □

[1] auch: Stelle i in f ist wesentlich.

Man kann mit Hilfe von zwei Schritten, feststellen, welche Variablen wesentlich sind:

1. Konstruktion einer Formel $\phi_f \in \mathcal{A}(X)$ zu f
2. Äquivalenzumformungen, um Variablen zu entfernen

Wesentliche Variablen werden dabei nicht entfernt, bleiben also stehen.

1.4.2 Beispiel
Sei $X = \{x, y, z\}$ und $f : \mathbb{B}^3 \to \mathbb{B}, f(x, y, z) = x + y \cdot x + z \cdot x$
Formel $\phi_f : x \vee (y \wedge x) \vee (z \wedge x)$

$$\begin{array}{lll} & x \vee (y \wedge x) \vee (z \wedge x) & \\ \text{logisch äquivalent zu:} & x \vee (z \wedge x) & \text{Absorption} \\ \text{logisch äquivalent zu:} & x & \text{Absorption} \end{array}$$

Also $f(a, b, c) = a$ für alle $a, b, c \in \mathbb{B}$. Das heißt x ist wesentlich, y und z hingegen sind unwesentlich.

1.4.3 Definition
Es sei $f : \mathbb{B}^n \to \mathbb{B}$ eine Boolesche Funktion. Zu $i, 1 \leq i \leq n$ ist der positive Co-Faktor definiert als

$$f_{x_i} : \mathbb{B}^n \to \mathbb{B}^n, f_{x_i}(y_1, \ldots, y_n) = f(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n)$$

und der negative Co-Faktor definiert als

$$f_{\overline{x_i}} : \mathbb{B}^n \to \mathbb{B}^n, f_{\overline{x_i}}(y_1, \ldots, y_n) = f(y_1, y_{i-1}, 0, y_{i+1}, \ldots, y_n).$$

□

Im Folgenden wird die Shannon-Zerlegung verwendet:

$$f = x_i \cdot f_{x_i} + \overline{x_i} \cdot f_{\overline{x_i}}$$

1.4.4 Definition
Die Projektionsfunktion $x_i : \mathbb{B}^n \to \mathbb{B}$ ist für $i, 1 \leq i \leq n$ definiert durch $x_i(y_1, \ldots, y_n) = y_i$.

□

Für die Operatoren der Booleschen Algebra bedeutet das wie folgt:

1. $x_i \cdot f_{x_i} \mathrel{\widehat{=}} x_i \sqcap f_{x_i}$ mit $\sqcap : \mathbb{B}_n \times \mathbb{B}_n \to \mathbb{B}_n$
2. $g + h \mathrel{\widehat{=}} g \sqcup h$ mit $\sqcup : \mathbb{B}_n \times \mathbb{B}_n \to \mathbb{B}_n$
3. $\bar{\ } \mathrel{\widehat{=}} \bar{\ } : \mathbb{B}_n \to \mathbb{B}_n$

1.4.5 Satz
Für alle $f \in \mathbb{B}_n$ und alle $i, 1 \leq i \leq n$ gilt

$$f = x_i \cdot f_{x_i} + \overline{x_i} \cdot f_{\overline{x_i}}.$$

Beweis
Sei $(y_1, \ldots, y_n) \in \mathbb{B}^n$ beliebig.
Zu zeigen: $f(y_1, \ldots, y_n) = (x_i \cdot f_{x_i} + \overline{x_i} \cdot f_{\overline{x_i}})(y_1, \ldots, y_n)$.

$$\begin{array}{rl} & (x_i \cdot f_{x_i} + \overline{x_i} \cdot f_{\overline{x_i}})(y_1, \ldots, y_n) \\ = & (x_i \cdot f_{x_i})(y_1, \ldots, y_n) + (\overline{x_i} \cdot f_{\overline{x_i}})(y_1, \ldots, y_n) \\ = & x_i(y_1, \ldots, y_n) \cdot f_{x_i}(y_1, \ldots, y_n) + \overline{x_i}(y_1, \ldots, y_n) \cdot f_{\overline{x_i}}(y_1, \ldots, y_n) \\ = & y_i \cdot f(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n) + \overline{x_i}(y_1, \ldots, y_n) \cdot f_{\overline{x_i}}(y_1, \ldots, y_n) \\ = & y_i \cdot f(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n) + \overline{y_i} \cdot f(y_1, y_{i-1}, 0, y_{i+1}, \ldots, y_n) \end{array}$$

Fall 1: $y_i = 1$

$$(x_i \cdot f_{x_i} + \overline{x_i} \cdot f_{\overline{x_i}})(y_1, \ldots, y_n) = f(y_1, \ldots, y_i, \ldots, y_n) + 0 = f(y_1, \ldots, y_n)$$

Fall 2: $y_i = 0$

$$(x_i \cdot f_{x_i} + \overline{x_i} \cdot f_{\overline{x_i}})(y_1, \ldots, y_n) = 0 + f(y_1, \ldots, y_i, \ldots, y_n) = f(y_1, \ldots, y_n)$$

□

1.4.6 Definition
Es sei $f : \mathbb{B}^n \to \mathbb{B}$, zu i, $1 \leq i \leq n$ heißt die Funktion

$$\exists x_i f : \mathbb{B}^n \to \mathbb{B}, \quad \exists x_i f = f_{x_i} + f_{\overline{x_i}}$$

die Existenzialquantifizierung und

$$\forall x_i f : \mathbb{B}^n \to \mathbb{B}, \quad \forall x_i f = f_{x_i} \cdot f_{\overline{x_i}}$$

die Allquantifizierung von f nach x_i. □

1.4.7 Satz
Es seien $f : \mathbb{B}^n \to \mathbb{B}^n$ und i, $1 \leq i \leq n$. Dann gilt für alle $(y_1, \ldots, y_n) \in \mathbb{B}$

$$(\exists x_i f)(y_1, \ldots, y_n) = 1 \Leftrightarrow \exists a \in \mathbb{B} : f(y_1, \ldots, y_{i-1}, a, y_{i+1}, \ldots, y_n) = 1$$

$$(\forall x_i f)(y_1, \ldots, y_n) = 1 \Leftrightarrow \forall a \in \mathbb{B} : f(y_1, \ldots, y_{i-1}, a, y_{i+1}, \ldots, y_n) = 1$$

Beweis
Für $\exists x_i f$:

$$\begin{array}{rcl}
(\exists x_i f)(y_1, \ldots, y_n) & = & (f_{x_i} + f_{\overline{x_i}})(y_1, \ldots, y_n) \\
& = & f_{x_i}(y_1, \ldots, y_n) + f_{\overline{x_i}}(y_1, \ldots, y_n) \\
& = & y_i \cdot f_{x_i}(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n) + \overline{y_i} \cdot f_{\overline{x_i}}(y_1, \ldots, y_{i-1}, 0, y_{i+1}, \ldots, y_n) \\
(\exists x_i f)(y_1, \ldots, y_n) = 1 & \Leftrightarrow & y_i \cdot f(y_1, \ldots, 1, \ldots, y_n) + \overline{y_i} \cdot f(y_1, \ldots, 0, \ldots, y_n) = 1 \\
& & y_i = 1 \text{ oder } y_i = 0 \\
& \Leftrightarrow & f(y_1, \ldots, 1, \ldots, y_n) = 1 \text{ oder } f(y_1, \ldots, 0, \ldots, y_n) = 1
\end{array}$$

□

2 Binäre Entscheidungsdiagramme

Folgende Themen sollen behandelt werden:

- Grundlagen Graphentheorie
- Grundlagen BDDs
- BDDs und Boolesche Funktionen
- BDDs für spezielle Funktionen

2.1 Graphentheoretische Grundlagen

Es soll die übliche Konvention gelten: $g = (V, P), V \neq \emptyset, |V| < \infty$ Knotenmenge , $P \subseteq V \times V$ Pfeilmenge , $(x, y) \in P, x, y \in V$.

2.1.1 Definition
Ein gerichteter (Multi-) Graph ist ein 4-Tupel $g = (V, P, \alpha, \omega)$ mit

(1) V endlich und $V \neq \emptyset, x \in V$ heißt Knoten

(2) P endliche Menge, $p \in P$ heißt Pfeil

(3) $\alpha, \omega : P \rightarrow V$ sind Funktionen, dabei heißen $\alpha(p)$ und $\omega(p)$ Anfangs- bzw. Endknoten von $p \in P$.

Gilt für $x, y \in V$ und $p \in P$, dass $\alpha(p) = x$ und $\omega(p) = y$, so heißt x Vorgänger von y und y Nachfolger von x.

(4) $d^+g(x) = |\{p \in P \mid \alpha(p) = x\}|$ heißt Außengrad von $x \in V$

(5) $d^-g(x) = |\{p \in P \mid \omega(p) = x\}|$ heißt Innengrad von $x \in V$

□

2.1.2 Definition
Sei $g = (V, P, \alpha, \omega)$ ein Graph.

(1) Eine Folge $(p_1, \ldots, p_n) \in P^+$[2] heißt Weg von $\alpha(p_1)$ nach $\omega(p_n)$, falls für alle $i, 1 \leq i \leq n$ gilt: $\omega(p_i) = \alpha(p_{i+1})$. Sind alle paarweise verschieden, so heißt der Weg einfach.

(2) Ein Weg $(p_1, \ldots, p_n)$ heißt Kreis, falls $\omega(p_n) = \alpha(p_1)$. Ein Kreis heißt einfach, falls alle Pfeile paarweise verschieden sind.

(3) Knotenlisten von Wegen nach Pfeilen[3] $(p_1, \ldots, p_n)$ lassen sich wie folgt darstellen: $(\alpha(p_1), \ldots, \alpha(p_n), \omega(p_n))$.

(4) Seien $x, y \in V$. Dann heißt y "von x aus erreichbar", falls $x = y$ oder ein Weg $(p_1, \ldots, p_n) \in P^+$ mit $\alpha(p_1) = x$ und $\omega(p_n) = y$ existiert.

(5) g heißt kreisfrei, falls es keinen Kreis gibt.

□

2.1.3 Beispiel
BILD
Die Weglänge entspricht der Anzahl der Pfeile.

[2] nichtleere Liste bzw $n \geq 1$
[3] Pfeile sind eindeutig

2.1.4 Definition
Seien $g = (V, P, \alpha, \omega)$ Graph und $x \in V$. Dann heißt x

(1) Quelle, falls $d^{-}g(x) = 0$ ($\omega(p) \neq x$ für alle $p \in P$)

(2) Senke, falls $d^{+}g(x) = 0$ ($\alpha(p) \neq x$ für alle $p \in P$)

(3) Wurzel, falls jeder Knoten $y \in V$ von x aus erreichbar ist.

□

2.1.5 Definition
Sei $g = (V, P, \alpha, \omega)$ Graph. Dann heißt g

(1) Knotenmarkiert, falls es eine Funktion $m : V \to M$ gibt ($m(x)$ heißt Marke von $x \in V$).

(2) Pfeilmarkiert, falls es eine Funktion $m : P \to M$ gibt ($m(x)$ heißt Marke von $p \in P$).

□

2.2 Grundlagen BDDs

2.2.1 Beispiel
Hier könnte Ihre Werbung stehen!

2.2.2 Definition
Es sei $X = \{x_1, \ldots . x_n\}$ eine Menge von Variablen. Ein BDD zu X ist ein gerichteter Graph $g = (V, P, \alpha, \omega)$ mit den folgenden Eigenschaften

(1) Der Graph g ist kreistfrei, hat genau eine Wurzel und genau zwei Senken, genannt $\mathbb{I}$ und $\mathbb{O}$.

(2) Jeder Knoten ungleich $\mathbb{I}$ und $\mathbb{O}$ heißt innerer Knoten und trägt eine Variable aus X als Marke. $\mathbb{I}$ heißt 1-Senke und trägt die 1 als Marke und $\mathbb{O}$ heißt 0-Senke und trägt die 0 als Marke. Falls $V' = V \cup \{\mathbb{I}, \mathbb{O}\}$, dann V innere Marken.

(3) Jeder Knoten ungleich $\mathbb{I}$ und $\mathbb{O}$ hat genau zwei ausgehende Pfeile. Ein Pfeil ist mit 1 markiert und heißt 1-Pfeil, der andere ist mit 0 markiert und heißt 0-Pfeil. 1- und 0-Nachfolger von $x \in V$ sind entsprechend definiert.

(4) Für alle Wege von der Wurzel zu einem Knoten sind die Knotenmarkierungen der Knoten der Knotenlisten paarweise verschieden.

□

2.2.3 Bezeichnungen (1) $\mathbb{I}$ 1-Senke, $\mathbb{O}$ 0-Senke.

(2) Für $x \in V$ ist $\underline{var}(x) \in X$ Markierung ($\underline{var}(\mathbb{I}) = 1$ und $\underline{var}(\mathbb{O}) = 0$).

(3) $\underline{size}(g) = |V|$ ist die Anzahl der inneren Knoten, $\underline{height}(g)$ bezeichnet die Länge eines längsten Weges von der Wurzel zu einer Senke[4], $\underline{width}(g) = \max_{i \in \{1,\ldots,n\}} |V_i|$ mit $V_i = \{a \in V | \underline{var}(a) = x_i\}$.

(4) Zu $a \in V$ heißt ist 1-Nachfolger, $\underline{then}(a)$ und 0-Nachfolger $\underline{else}(a)$.

□

[4] Höhe von g

2.2.4 Definition
Sei $(X, >)$ Variablenordnung, wobei $>$ lineare Striktordnung auf X. Ein BDD g heißt geordnet oder OBDD, falls für alle Wege $(p_1, \ldots, p_n)$ von der Wurzel bis zur einer Senke gilt:

$$\underline{var}(p_i) < \underline{var}(p_{i+1}) \text{ für } i, 1 \leq i \leq n-1$$

□

Beispiel
Regelbsp. Entfernen eines redundanten Knotens

2.2.5 Regel
Es sei $g = (V, P, \alpha, \omega)$ ein OBDD. Ein innerer Knoten $a \in V$ heißt redundant, falls $\underline{then}(a) = \underline{else}(a)$. Das Entfernen eines redundanten Knotens a funktioniert wie folgt:

(1) a wird aus V entfernt.

(2) Jeder Pfeil $p \in P$ mit $\alpha(p) = a$ oder $\omega(p) = a$ wird aus P entfernt.

(3) Für jeden Vorgängerknoten von a, füge einen
  (i) 1-Pfeil nach $\underline{then}(a)$, falls es ein 1-Pfeil war.
  (ii) 0-Pfeil nach $\underline{else}(a)$, sonst.

□

2.2.6 Definition
Sei $g = (V, P, \alpha, \omega)$ ein OBDD und $a \in V$ innerer Knoten. Das durch a induzierte Unter-OBDD $g' = (V', P', \alpha\,\omega')$ ist wie folgt definiert:

(1) $V' = \{b \in V \mid b \text{ von } a \text{ aus erreichbar}\}$

(2) $P' = \{p \in P \mid \alpha(p) \in V' \wedge \omega(p) \in V'\}$

(3) $\alpha' : P' \to V'$ ist definiert durch $\alpha'(p) = \alpha(p)$ für alle $p \in P$.
$\omega' : P' \to V'$ ist definiert durch $\omega'(p) = \omega(p)$ für alle $p \in P$.

(4) Jedes $a \in V'$ hat die gleiche Marke wie in g.

(5) Jeder $p \in P$ hat den selben Typ (0- oder 1-Pfeil) wie in g.

□

2.2.7 Definition
Seien $g = (V, P, \alpha, \omega)$ und $g' = (V', P', \alpha', \omega')$ OBDDs. Dann heißen g ung g' <u>strukturgleich</u>[5], falls es bijektive Funktionen $\Phi : V \to V'$ und $\Psi : P \to P'$ gibt, so dass für alle $p \in P$ gilt

(1) $\alpha'(\Psi(p)) = \Phi(\alpha(p))$

(2) $\omega'(\Psi(p)) = \Phi(\omega(p))$

(3) p ist 1-Pfeil $\to \Psi(p)$ ist 1-Pfeil

(4) p ist 0-Pfeil $\to \Psi(p)$ ist 0-Pfeil

und für alle $a \in V$ gilt

(5) $\underline{var}(\Phi(a)) = \underline{var}(a)$

[5] auch: isomorph

□

2.2.8 Beispiel

Hier sollte ein Bild sein.

2.2.9 Regel

Sei $g = (V, P, \alpha, \omega)$ OBDD. Innere Knoten $a, b \in V$ heißen <u>äquivalent</u>, falls gilt:

$$\begin{aligned} \underline{then}(a) &= \underline{then}(b) \\ \underline{else}(a) &= \underline{else}(b) \\ \underline{var}(a) &= \underline{var}(b) \end{aligned}$$

Das Verschmelzen von a und b[6] funktioniert wie folgt:

(1) b wird aus V entfernt.

(2) Jeder Pfeil $p \in P$ mit $\alpha(p) = b$ bzw. $\omega(p) = b$ wird aus P entfernt.

(3) Für alle Vorgänger c von b füge
   - (i) 1-Pfeil von c nach a ein gdw. 1-Pfeil von c nach b existierte,
   - (ii) 0-Pfeil von c nach a ein, sonst.

□

2.2.10 Definition

Sei g ein OBDD über Variablenmenge X. Dann heißt g

(1) <u>vollständig</u>, falls für jeden Weg $(p_1, \ldots, p_n)$ von der Wurzel zu einer Senke ($\mathbb{I}$ oder $\mathbb{O}$) gilt: $n = |X|$.

(2) <u>quasi-reduziert</u> (kurz: QOBDD), falls es vollständig ist und es keine äquivalenten Knoten gibt.

(3) <u>reduziert</u> (kurz: ROBDD), falls es keine redundanten und äquivalenten Knoten besitzt.

Ist $\pi = (x_1, \ldots, x_n)$ die Variablenordnung, so nennt man $V_i = \{a \in V \mid \underline{var}(a) = x_i\}$ die i-te <u>Schicht</u>, wobei $1 \leq i \leq n$ und $\{\mathbb{I}, \mathbb{O}\}$ heißt die $n+1$-te Schicht.

□

2.2.11 Beispiel

Ein weiteres Beispiel

2.3 Darstellung von Booleschen Funktionen durch BDDs

2.3.1 Definition

Sei $g = (V, P, \alpha, \omega)$ OBDD.

$$f_{\mathbb{I}} : \mathbb{B}^n \to \mathbb{B}, \qquad f_{\mathbb{I}}(x_1, \ldots, x_n) = 1$$

$$f_{\mathbb{O}} : \mathbb{B}^n \to \mathbb{B}, \qquad f_{\mathbb{O}}(x_1, \ldots, x_n) = 0$$

und für jeden inneren Knoten $a \in V$ ist

$$f_a : \mathbb{B}^n \to \mathbb{B}, \qquad f_a(x_1, \ldots, x_n) = x \cdot f_{then(a)}(x_1, \ldots, x_n) + \overline{x} \cdot f_{else(a)}(x_1, \ldots, x_n)$$

mit $x := \underline{var}(a)$ definiert. Ist w Wurzel von g, so ist f_w die von g dargestellte Funktion. □

[6] durch Entfernen von b

2.3.2 Beispiel
Sei $X = \{x, y, z, u\}$ und $\pi = (x, y, z, u)$.
Hier fehlt ein Bild.

$$\begin{aligned}
f_a(x,y,z,u) &= x \cdot f_c(x,y,z,u) + \overline{x} \cdot f_b(x,y,z,u) \\
&= x \cdot (z \cdot f_d(x,y,z,u) + \overline{z} \cdot f_{\mathbb{O}}(x,y,z,u)) + \overline{x} \cdot (y \cdot f_c(x,y,z,u) + \overline{y} \cdot f_{\mathbb{O}}(x,y,z,u)) \\
&= x \cdot (z \cdot (u \cdot f_{\mathbb{I}}(x,y,z,u) + \overline{u} \cdot f_{\mathbb{O}}(x,y,z,u)) + \overline{z} \cdot 0) + \\
&\quad\; \overline{x} \cdot (y \cdot (z \cdot (u \cdot f_{\mathbb{I}}(x,y,z,u) + \overline{u} \cdot f_{\mathbb{O}}(x,y,z,u)) + \overline{z} \cdot 0)) \\
&= x \cdot (z \cdot (u \cdot 1 + \overline{u} \cdot 0)) + \overline{x} \cdot (y \cdot (z \cdot (u \cdot 1 + \overline{u} \cdot 0))) \\
&= x \cdot z \cdot u + \overline{x} \cdot y \cdot z \cdot u
\end{aligned}$$

2.3.3 Satz
Sei g ein OBDD. Zu einem Weg $(p_1, \ldots, p_n)$ von der Wurzel w bis zur $\mathbb{I}$ (1-Senke) sei definiert:

$$m(p_1, \ldots, p_k) = \prod_{i=1}^{k} \underline{var}(\alpha(p_i))^{a_i}$$

wobei $x_i^{a_i}$ wie in Def. 1.3.4 und $a_i = 1$ gdw. p_i 1-Pfeil bzw. $a_i = 0$ gdw. p_i 0-Pfeil ist, dann gilt

$$f_w(x_1, \ldots, x_n) = \sum_{w \in \mathcal{W}} m(w)$$

wobei $\mathcal{W}$ die Menge der Wege von w nach $\mathbb{I}$ ist.

Beweis
Induktion über die Höhe des OBDDs
<u>IA:</u> Höhe sei 1.
Hier fehlen wieder Bilder.
1. Fall:

$$f_w(x_1, \ldots, x_n) = x \cdot f_{\mathbb{I}}(x_1, \ldots, x_n) + \overline{x} \cdot f_{\mathbb{O}}(x_1, \ldots, x_n) = x$$

Dann gilt $\mathcal{W} = \{(p)\}$, $m((p)) = x$ und schließlich $\sum_{w \in \mathcal{W}} m(w) = m((p)) = x = f_w(x_1, \ldots, x_n)$.
Analog für Fall 2 bis 4. <u>IS:</u> Sei Höhe > 1 und g habe folgende Form (Bild).

$$\begin{aligned}
f_a(x_1, \ldots, x_n) &= x \cdot f_b(x_1, \ldots, x_n) + \overline{x} \cdot f_c(x_1, \ldots, x_n) \\
&= x \cdot \sum_{w \in \mathcal{W}_1} m(w) + \overline{x} \cdot \sum_{v \in \mathcal{W}_2} m(v) \\
&= \sum_{w \in \mathcal{W}_1} x \cdot m(w) + \sum_{v \in \mathcal{W}_2} \overline{x} \cdot m(v) \\
&= \sum_{w \in \mathcal{W}, w_1 = p} m(w) + \sum_{v \in \mathcal{W}, v_1 = q} m(v) \\
&= \sum_{w \in \mathcal{W}} m(w)
\end{aligned}$$

□

2.3.4 Konvention
Bei ROBDDs werden auch die 1- und 0-Senke als OBDDs aufgefasst, mit $\mathbb{I}$ bzw. $\mathbb{O}$ bezeichnet und es wird $f_{\mathbb{I}}$ und $f_{\mathbb{O}}$ wie in Def. 2.3.1 definiert. □

2.3.5 Satz
Zu jeder Booleschen Funktion existiert ein OBDD, welcher die Funktion darstellt. Dies gilt auch, wenn man QOBDDs bzw. ROBDDs betrachtet. □

2.3.6 Satz
Es sei f eine Boolesche Funktion. Wird f durch zwei ROBDDs (oder QOBDDs) g_1 und g_2 dargestellt, so sind g_1 und g_2 strukturgleich. □

Es ist festzuhalten, dass die Variablenordnung Einfluss auf die Größe hat.

2.3.7 Beispiel
BILD

2.4 OBDDs für wichtige Funktionen

Gaaaanz viele Bilder.

3 Algorithmen

- Minimierung
- Konstruktion (Zusammenbau, Synthese)
- Operation für Basisfragen von logischen Formeln

3.1 Minimierung

Gegeben: OBDD
Aufgabe: Konstruiere einen ROBDD durch Anwenden der Regeln „Entfernen redundanter Knoten" und „Verschmelzen äquivalenter Knoten" soweit wie möglich.

3.1.1 OBDDs als Zeigergeflecht

Sei g ein OBDD. Jeder Knoten a wird als Record mit 4 Komponenten dargestellt.

(1) a.id - Knotenbezeichnung, natürliche Zahl (alle verschieden)

(2) a.var - Beschriftung, d.h. Variable[7]

(3) a.then - Verweis auf $then(a)$, falls a innerer Knoten, sonst nil

(4) a.else - Verweis auf $else(a)$, falls a innerer Knoten, sonst nil

Zusätzlich: Mengen $S_1, \dots, S_n$ mit $S_i = \{a | \underline{var}(a) = x_i\}$. Im konkreten S_i gibt es eine Liste von Verweise auf Records mit $a.var = x_i$ (bzw. $a.var = i$).

3.1.2 Beispiel

Bildchen

3.1.3 Algorithmus Reduce

Eingabe: OBDD g zu einer Variablenordnung (in Zeigergeflechtdarstellung)
Ausgabe: ROBDD - Geflechtdarstellung zu g

`Reduce`(g):

(1) Durchlaufe g in DFS-Ordnung und setze dabei
0−Senke.id := 0
1−Senke.id := 1
z [0] = 0−Senke
z [1] = 1−Senke

(2) k := 2
for i=n−1 until 1 do

  (1) Suche in S_i alle redundanten Knoten und markiere sie als entfernbar.

  (2) Sortiere die Mengen S_i nach der folgenden Ordnung:
  $a \sqsubseteq b \Leftrightarrow$ (a.then.id, a.else.id) $\leq_{lex.}$ (b.then.id, b.else.id)
  Damit stehen äquivalente Knoten in der Liste immer hintereinander.

  (3) Suche die verschmelzbaren Knoten und markiere alle bis auf einen als entfernbar. Für jede Teilliste $a_1, \dots, a_n$ von äquivalenten Knoten wird a_1 Vertreterknoten und $a_2, \dots, a_n$ werden als entfernbar markiert.
  k := k+1
  a1.id = k
  z[k] = a_1

[7] Index der Variable bei Implementierung

(4) Alle in (1) und (3) als entferbar markierten Knoten werden entfernt und auf sie verweisende Zeiger mithilfe von z entsprechend umgelenkt.

3.2 Auswertung und Erfüllbarkeit

Auswertung:
Gegeben sei ein OBDD zur Funktion $f : \mathbb{B}^n \to \mathbb{B}$ und $b_1, \dots b_n \in \mathbb{B}^n$. Bestimme $f(b_1, \dots, b_n)$.

Erfüllbarkeit:
Gegeben sei ein OBDD zur Funktion $f : \mathbb{B}^n \to \mathbb{B}$. Gibt es $(b_1, \dots, b_n) \in \mathbb{B}^n$ mit $f(b_1, \dots, b_n) = 1$?

3.2.1 Auswertungsalgorithmus Value

Eingabe: OBDD g zu f und $(b_1, \dots, b_n) \in \mathbb{B}^n$.

`Value(g,b)`

$a := g$
while a keine Senke do
    if $b[a.var] = 0$ then $a := a.else$
    else $a := a.then$
return $a.var$

3.2.2 Beispiel

$X = \{x_1, x_2, x_3, x_4\}, \;\; \pi = (x_1, x_2, x_3, x_4), \;\; b = (0, 1, 0, 1)$
$f(x) = x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_1 \cdot x_2 \cdot \overline{x_3} \cdot x_4 + x_1 \cdot \overline{x_2} \cdot x_3 \cdot x_4 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 + \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4 + \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4$

3.2.3 Erfüllbarkeitsalgorithmus Satisfy

Eingabe: OBDD g und globales Feld B
Ausgabe: 1 und Belegung von B, die den Wert 1 liefert, sonst 0

`Satisfy(g,B)`[8]

$a := g$
if a Senke then return $a.var$
else $B[a.var] = 1$
if $Satisfy(a.then, B)$ then return 1
$B[a.var] := 0$
if $Satisfy(a.else, B)$ then return 1

3.2.4 Beispiel

Sei $X = \{x_1, x_2, x_3\}$ und $\pi = (x_1, x_2, x_3)$, sowie $f(x) = x_1 \cdot \overline{x_2} \cdot x_3 = \overline{x_1} \cdot \overline{x_2} \cdot x_3$

3.2.5 Satz

Sei $f : \mathbb{B}^n \to \mathbb{B}$ Boolesche Funktion, die durch ein QOBDD dargestellt wird. Dann ist die Anzahl der erfüllenden Belegungen gegeben durch $\underline{anz}(a)$, wobei anz rekursiv wie folgt definiert ist[9]:

$$\begin{aligned}
\underline{anz}(\mathbb{O}) &= 0\\
\underline{anz}(\mathbb{I}) &= 1\\
\underline{anz}(a) &= \underline{anz}(a.then) + \underline{anz}(a.else), \;\; \text{wobei } a \text{ keine Senke}
\end{aligned}$$

$\square$

3.2.6 Beispiel

QOBDD zur Funktion von Beispiel 2.2.1
$X = \{x_1, x_2, x_3\}, \;\; \pi = (x_1, x_2, x_3)$

[8] Komplexität: $O(\underline{size}(g))$
[9] Für ROBDDs gilt dies jedoch nicht.

Sei g ein ROBDD über $X = \{x_1, \dots, x_n\}$. Für Knoten a von g wird $\underline{level}(a)$ definiert als

- Index von $\underline{var}(a)$, falls a innerer Knoten
- $n + 1$, falls a Senke

3.2.7 Satz

Sei g ein ROBDD, welches Funktion $f : \mathbb{B}^n \to \mathbb{B}$ darstellt. Dann ist die Anzhal der erfüllenden Belegungen von f gegeben durch $\underline{anz}(g)$[10], wobei die Funktion anz auf ROBDD Knoten definiert ist durch:

$$\begin{aligned}
\underline{anz}(\mathbb{I}) &= 1 \\
\underline{anz}(\mathbb{O}) &= 0 \\
\underline{anz}(a) &= \frac{1}{2} \cdot \left(\underline{anz}(\underline{then}(a)) \cdot 2^{\,\underline{level}(\underline{then}(a)) - \underline{level}(a)} + \underline{anz}(\underline{else}(a)) \cdot 2^{\,\underline{level}(\underline{else}(a)) - \underline{level}(a)} \right)
\end{aligned}$$

□

3.2.8 Beispiel

Boolesche Addition.

ROBDD zur Funktion von Beispiel 2.2.1.

3.3 Äquivalenztest

Problem: $f_1, f_2 : \mathbb{B}^n \to \mathbb{B}$. Gilt $f_1 = f_2$? (D.h. $f_1(a_1, \dots, a_n) = f_2(a_1, \dots, a_n)$ f.a. $(a_1, \dots, a_n) \in \mathbb{B}^n$)
Lösung: Formuliere f_1 und f_2 über gleichen Variablen $X = \{x_1, \dots, x_n\}$. Lege (möglichst günstige) Variablenordnung fest und stelle f_1 und f_2 durch BDDs[11] g_1, g_2 dar.
Dann gilt: $f_1 = f_2 \Leftrightarrow g_1$ und g_2 strukturgleiche BDDs.

3.3.1 Algorithmus Equiv

`Equiv`(a,b)	if a 1-Senke then return „b 1-Senke“ else if a 0-Senke then return „b 0-Senke“ else if b Senke then return false else if $var(a) = var(b)$ then return $\texttt{Equiv}(\underline{then}(a), \underline{then}(b)) \wedge \texttt{Equiv}(\underline{else}(a), \underline{else}(b))$ else return false

Notwendig zur Effizienzsteigerung: Computed Table
Computed Table CT mit Einträgen $((a, b,), r)$ als Paare, wobei a, b BDD-Knoten sind und r ein Wahrheitswert ist. Dabei gilt:

$$((a, b), r) \text{ in CT } \Rightarrow \texttt{Equiv}(a, b) = r$$

Die Tabellenoperationen umfassen dabei folgende Funktionalitäten:

- Einfügen eines Paares
- Testen, ob zu (a, b) ein Eintrag existiert
- Auslesen des Wahrheitwertes r zum Eintrag $((a, b), r)$

Die Implementierung von CTs erfolgt zumeist durch AVL-Bäume, wodurch logarithmische Zugriffszeiten möglich sind.

[10] wenn g Wurzel ist
[11] am besten sogar als ROBDDs

3.4 Negation

Gegeben sei eine Funktion $f : \mathbb{B}^n \to \mathbb{B}$ und die Negation $\bar{f} : \mathbb{B}^n \to \mathbb{B}, \bar{f}(x_1, \dots, x_n) = \overline{f(x_1, \dots, x_n)}$.
Problem: Ist g OBDD zur Darstellung von f, wie bekommt man OBDD zur Darstellung von $\bar{f}$?
Bei ROBDDS und QOBDDs wird mit $\bar{g}$ das BDD bezeichnet, welches $\bar{f}$ darstellt. Man erhält also die ROBDD- bzw. QOBDD-Negation durch die Berechnung von $\bar{g}$ aus g.

3.4.1 Satz
Sei g das ROBDD (QOBDD), welches $f : \mathbb{B}^n \to \mathbb{B}$ darstellt. Ist $\bar{g}$ das BDD, welches aus g dadurch entsteht, dass man $\mathbb{O}$ und $\mathbb{I}$ vertauscht, so stellt $\bar{g}$ die Funktion $\bar{f}$ dar. □

3.4.2 Definition
Es seien g_1, g_2 OBDDs mit fester Variablenordnung und $x \in X$, so dass $\underline{var}(a) < x$ für alle Knoten a von g_1 und g_2 gilt. Dann ist OBDD $\underline{cons}(x, g_1, g_2)$ festgelegt durch:

$$\begin{aligned} \underline{var}(\underline{cons}(x, g_1, g_2)) &= x \\ \underline{then}(\underline{cons}(x, g_1, g_2)) &= g1 \\ \underline{else}(\underline{cons}(x, g_1, g_2)) &= g2 \end{aligned}$$

□

Implementierung:

`Cons`(x, g_1, g_2)

$new(a)$
$a.id := \underline{max}(g_1.id, g_2.id) + 1$
$a.var := x$
$a.then := g1$
$a.else := g2$
$return$ `Reduce`(a)

3.4.3 Beispiel
$X = \{x, y, z\}, \pi = (x, y, z)$ $f_1(x, y, z) = y + z, f_2(x, y, z) = y \cdot z)$

(1) Konstruiere $\underline{cons}(x, g_1, g_2)$.[12]

(2) Konstruiere $\underline{cons}(x, g_1, g_1) = g_1$.

3.4.4 Satz
Seien $f_1, f_2 : \mathbb{B}^n \to \mathbb{B}$ dargestellt durch OBDDs g_1 bzw. g_2. Dann wird durch $\underline{cons}(x, g_1, g_2)$ die Funktion $f : \mathbb{B}^n \to \mathbb{B}$ dargestellt, wobei gilt:

$$f = x \cdot f_1 + \bar{x} \cdot f_2$$

□

3.4.5 Algorithmus Negation
Voraussetzung: Computed Table CT mit Einträgen (a, r), wobei a und r OBDD-Knoten sind.

[12] $f(x, y, z) = x \cdot y + x \cdot \bar{y} \cdot z = \bar{x} \cdot y \cdot z = x \cdot f_1 + \bar{x} \cdot f_2$

$\texttt{Neg}(a)$

if a = 1-Senke then return $\mathbb{O}$
else if a = O-Senke then return $\mathbb{I}$
else if $\exists r : (a, r)$ in CT then return r mit (a, r) in CT
else $r_1 := \texttt{Neg}(\underline{then}(a))$
$\quad r_2 := \texttt{Neg}(\underline{else}(a))$
$\quad r \ := \underline{cons}(\underline{var}(a), r_1, r_2)$
$\quad \underline{add}(a, r)$ to CT
$\quad$ return r

3.4.6 Beispiel
Sei $X = \{y, z\}$ und $f = y + z$. Konstruiere $\underline{cons}(y, \mathbb{I}, \underline{cons}(z, \mathbb{I}, \mathbb{O}))$.
Wie sieht die Negation aus?

3.5 Binäre Operationen

Es gibt 16 Funktionen von $\mathbb{B} \times \mathbb{B}$ nach $\mathbb{B}$, da $|\mathbb{B}^{\mathbb{B}\times\mathbb{B}}| = 2^4 = 16$. So sind bspw. $+, \cdot, \rightarrow, \leftrightarrow$ solche Funktionen.

Wenn $f : \mathbb{B}^n \rightarrow \mathbb{B}$, $g : \mathbb{B}^n \rightarrow \mathbb{B}, \otimes : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$, dann ist $f \otimes g : \mathbb{B}^n \rightarrow \mathbb{B}$, wobei $f \otimes g$ definiert ist durch:

$$(f \otimes g)(x_1, \ldots, x_n) = f(x_1, \otimes, x_n) \otimes g(x_1, \ldots, x_n)$$

Also $\otimes : \mathbb{B}_n \times \mathbb{B}_n \rightarrow \mathbb{B}_n$.

Beispiele:

$$+ : f + g : \mathbb{B}^n \rightarrow \mathbb{B} \quad (f \sqcup g)$$
$$\cdot : f \cdot g : \mathbb{B}^n \rightarrow \mathbb{B} \quad (f \sqcap g)$$

Problem:
Gegeben seien Funktionen $f_1, f_2 : \mathbb{B}^n \rightarrow \mathbb{B}, \otimes : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B}$ und OBDDs g_1 für f_1 und g_2 für f_2.
Konstruiere g_1 und g_2 OBDD zur Darstellung von $f_1 \otimes f_2$ (Binäre Synthese).

Im Folgenden sollen zwei Ansätze zur Binären Synthese betrachtet werden.

(1) ITH-Funktion (if-then-else)

(2) Apply-Funktion

Zu (1): ITH-Funktion
$ITH : \mathbb{B}_n \times \mathbb{B}_n \times \mathbb{B}_n \rightarrow \mathbb{B}_n,\ ITH(f, g, h) = f \cdot g + \bar{f}h$

Nun soll gezeigt werden, dass die Funktionen $f_1 + f_2, f_1 \cdot f_2, \bar{f}$ mit der ITH-Funktion darstellbar sind.

$$ITH(f, \mathbf{0}, \mathbf{L}) = f \cdot \mathbf{0} + \bar{f} \cdot \mathbf{L} = \bar{f}$$
$$ITH(f_1, f_2, \mathbf{0}) = f_1 \cdot f_2 + \overline{f_1} \cdot \mathbf{0} = f_1 \cdot f_2$$
$$ITH(f_1, \mathbf{L}, f_2) = f_1 \cdot \mathbf{L} + \overline{f_1} \cdot f_2 = f_1 + \overline{f_1} \cdot f_2 = (f_1 + \overline{f_1}) \cdot (f_1 + f_2) = f_1 + f_2$$

Damit ist jede Funktion $f : \mathbb{B}^n \rightarrow \mathbb{B}$ durch einen Term in ITH und $\mathbf{0}, \mathbf{L}$ ausdrückbar.

Zu (2): Apply-Funktion
Apply : $\mathbb{B}_n \times \mathbb{B}_n \times \mathbb{B}_2 \rightarrow \mathbb{B}_n$, wobei die ersten zwei Argumente, sowie das Resultat OBDDs sind und das dritte Argument ein Operation ($\otimes$) ist.
Die Apply-Funktion soll nun im weiteren Verlauf verwendet werden.

3.5.1 Satz
Es sei $f, g : \mathbb{B}^n \to \mathbb{B}$ und $\otimes : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$. Dann gilt für jede Variable x_i (aufgefasst als Projektion) $x_i : \mathbb{B}_n \to \mathbb{B}$:

$$f \otimes g = x_i \cdot (f_{x_i} \otimes g_{x_i}) + \overline{x_i} \cdot (f_{\overline{x_i}} \otimes g_{\overline{x_i}})$$

Beweis
Seien $(y_1, \ldots, y_n) \in \mathbb{B}^n$. Dann gilt $f \otimes g(y_1, \ldots, y_n) = f(y_1, \ldots, y_n) \otimes g(y_1, \ldots, y_n)$.

Rechte Seite:

$(x_i \cdot (f_{x_i} \otimes g_{x_i}) + \overline{x_i} \cdot (f_{\overline{x_i}} \otimes g_{\overline{x_i}}))(y_1, \ldots, y_n)$
$= x_i \cdot (f_{x_i} \otimes g_{x_i})(y_1, \ldots, y_n) + \overline{x_i} \cdot (f_{\overline{x_i}} \otimes g_{\overline{x_i}})(y_1, \ldots, y_n)$
$= x_i(y_1, \ldots, y_n) \cdot (f_{x_i}(y_1, \ldots, y_n) \otimes g_{x_i}(y_1, \ldots, y_n)) + \overline{x_i}(y_1, \ldots, y_n) \cdot (f_{\overline{x_i}}(y_1, \ldots, y_n) \otimes g_{\overline{x_i}}(y_1, \ldots, y_n))$
$= y_i \cdot f(y_1, \ldots, y_{i-1}, 1, y_{i+1} \ldots, y_n) \otimes g(y_1, \ldots, y_{i-1}, 1, y_{i+1} \ldots, y_n)$
$\quad + \overline{y_i} \cdot f(y_1, \ldots, y_{i-1}, 0, y_{i+1}, \ldots, y_n) \otimes g(y_1, \ldots, y_{i-1}, 0, y_{i+1}, \ldots, y_n)$
Fall 1: $y_i = 1$
$= f(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n) \otimes g(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n) + 0$
$= f(y_1, \ldots, y_n) \otimes g(y_1, \ldots, y_n)$
Fall 2: $y_1 = 0$
$= 0 + f(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n) \otimes g(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n)$
$= f(y_1, \ldots, y_n) \otimes g(y_1, \ldots, y_n)$

$\square$

3.5.2 Binäre Synthese bei QOBDDs
Voraussetzung: $f_1, f_2 : \mathbb{B}^n \to \mathbb{B}$ werden durch die QOBDDs g_1 bzw. g_2 (mit gleicher Variablenordnung) dargestellt und $\otimes : \mathbb{B} \to \mathbb{B}$ sei die gegebene binäre Operation.
Dann berechnet $\texttt{Apply}(g_1, g_2, \otimes)$ das QOBDD für $g_1 \otimes g_2$ (mit CT).

$\texttt{Apply}(a, b, \otimes)$	if $a \wedge b$ Senken then return $a \otimes b$ als QOBDD elseif $\exists r : ((a, b), r)$ in CT then return r mit $((a, b), r)$ in CT else $r_1 := \texttt{Apply}(\underline{then}(a), \underline{then}(b), \otimes)$ $r_2 := \texttt{Apply}(\underline{else}(a), \underline{else}(b), \otimes)$ $r := \texttt{Cons}(\underline{var}(a), r_1, r_2)$ $\underline{add}((a, b), r)$ to CT return r

3.5.3 Beispiel
(1) Seien x und $f_1, f_2 : \mathbb{B}^n \to \mathbb{B}, \otimes : \mathbb{B} \to \mathbb{B}$ gegeben mit

$$f_1(x) = x, \;\; f_2(x) = \bar{x}, \;\; \otimes = \cdot$$
$$(f_1 \cdot f_2)(x) = x \cdot \bar{x} = 0$$

($\texttt{Apply}(g_1, g_2, \otimes)$ durchführen, wobei g_1 und g_2 QOBDDs zu f_1 bzw. f_2.)

(2) $X = x, y,\ \pi = (x, y),\ f_1 : \mathbb{B}^2 \to \mathbb{B},\ f_2 : \mathbb{B}^2 \to \mathbb{B},\ \otimes : \mathbb{B} \to \mathbb{B}.$

$$\begin{aligned}
&f_1(x, y) = x \\
&f_2(x, y) = x + y \\
&\otimes = \cdot \\
&(f_1 \cdot f_2)(x, y) = f_1(x, y) \cdot f_2(x, y) = x \cdot (x + y) = x
\end{aligned}$$

3.5.4 Algorithmus Apply bei ROBDDs

Im Gegensatz zu den QOBBDs müssen hier 4 Fälle unterschieden werden.

(1) a und b sind Senken:
$\texttt{Apply}(a, b, \otimes) = a \otimes b$ (als ROBDD)

(2) a und b sind keine Senken und $\underline{var}(a) = \underline{var}(b) = x$:
$\texttt{Apply}(a, b, \otimes) = \texttt{Cons}(x, \texttt{Apply}(\underline{then}(a), \underline{then}(b), \otimes), \texttt{Apply}(\underline{else}(a), \underline{else}(b), \otimes))$

(3) a ist keine Senke und b beliebig, aber $\underline{var}(b) < \underline{var}(a) = x$, falls b keine Senke ist:
$\texttt{Apply}(a, b, \otimes) = \texttt{Cons}(x, \texttt{Apply}(\underline{then}(a), b, \otimes), \texttt{Apply}(\underline{else}(a), b, \otimes))$

(4) b keine Senke und a beliebig, aber $\underline{var}(a) < \underline{var}(b) = x$, falls a keine Senke ist:
$\texttt{Apply}(a, b, \otimes) = \texttt{Cons}(x, \texttt{Apply}(a, \underline{then}(b), \otimes), \texttt{Apply}(a, \underline{else}(b), \otimes))$

3.5.5 Beispiel

Seien $X = \{x\},\ f_1, f_2 : \mathbb{B}^2 \to \mathbb{B}, \otimes : \mathbb{B} \to \mathbb{B}.$

$$\begin{aligned}
&f_1(x, y) = x \\
&f_2(x, y) = x + y \\
&\otimes = \cdot \\
&(f_1 \otimes f_2)(x, y) = x \cdot (x + y) = x
\end{aligned}$$

3.6 BDD-Pakete

Anwendung:

(1) Schaltungsentfwurf, VLSI-Design

(2) Symbolisches Model Checking

(3) Algorithmik, große Graphen

(4) Spieltheorie (Stefan Bolus)

(5) Relationentheorie

Pakete:

(1) CUDD

(2) CMU-BDD-Paket

(3) BuDDy

(4) CrocoPat

(5) RelView und Kure

4 Anwendung der Spieltheorie

In diesem Abschnitt werden folgende Themen bearbeitet:

- Einfache Spiele und QOBDDs
- Berechnung von Schlüsselspielern
- Bestimmung der Wünschenswert-Relation

Kurz zur Spieltheorie:
Man unterscheidet zwischen konkurrierenden und kooperierenden Spielen. Zweiteres wird noch in einfache Spiele[13] und nicht-einfach Spiele unterteilt.
Mit BDDs kann man einfache Spiele besonders gut berechnen. Einfache Spiele entsprechen monotonen Funktionen.

4.1 Einfache Spiele und QOBDDs

4.1.1 Definition
Ein einfaches Spiel ist ein Paar $(X, \mathcal{W})$ mit:

(1) X endliche, nicht-leere Menge von Spielern

(2) $\mathcal{W} \subseteq 2^X$ mit der Eigenschaft, dass für alle $Y, Z \in 2^X$:

$$Y \in \mathcal{W} \wedge Y \subseteq Z \Rightarrow Z \in W$$

$Y \in 2^X$ heißt Koalition, $Y \in \mathcal{W}$ heißt gewinnende Koalition und $Y \in 2^X \backslash \mathcal{W}$ verlierende Koalition.

$\square$

4.1.2 Definition
Sei $(X, \mathcal{W})$ ein einfacher Spiel. Ein Paar (Q, w) mit $Q \in \mathbb{N}_{>0}$ und $w : X \to \mathbb{N}$ heißt gewichtete Dartellung, falls für alle $Y \in 2^X$ gilt:

$$Y \in \mathcal{W} \Leftrightarrow \sum_{x \in Y} w(x) \geq Q$$

Besitzt ein Spiel eine gewichtete Darstellung, so heißt es gewichtetes Mehrheitsspiel (GMS). $\square$

Normalfall der Schreibweise ist dabei wie folgt:
$X = \{x_1, \ldots, x_n\}, [Q; w_1, \ldots, w_n]$, wobei $w_i = w(x_i)$ und Q Quote.

4.1.3 Beispiel
(1) Sei $(X, \mathcal{W})$ mit $X = \{x_1, x_2, x_3\},\ \mathcal{W} = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_1, x_3\}.\{x_1, x_2, x_3\}\}$.
$(X, \mathcal{W})$ ist GMS mit $[6; 5, 5, 1]$.
Warum sind die Mengen in $\mathcal{W}$ gewinnende Koalitionen?

(2) Sei $(X, \mathcal{W})$ einfaches Spiel, gegeben durch $X = \{x_1, x_2, x_2, x_4, x_5\}$ und $[312; 239, 146, 93, 76, 68]$[14].
Hierbei stehen die Elemente aus X für die folgenden Partein:
$x_1 \mathrel{\widehat{=}}$ CDU/CSU, $x_2 \mathrel{\widehat{=}}$ SPD, $x_3 \mathrel{\widehat{=}} =$ FDP, $x_4 \mathrel{\widehat{=}}$ Die Linke, $x_5 \mathrel{\widehat{=}}$ Die Grünen und die Gewichte entsprechen der Sitzverteilung nach der Bundestagswahl 2009.
Wie sieht nun $\mathcal{W}$ aus?

[13] Nutzen ist hier 0 oder 1
[14] oder $[3; 2, 1, 1, 1, 0]$ in der minimalen Variante

4.1.4 Definition
Sei $(X, \mathcal{W})$ einfaches Spiel. Dann heißt dies Vektorgerichtetes Mehrheitsspiel (VGMS), falls es $k \in \mathbb{N}_{>0}$ und $Q^{(1)}, \ldots, Q^{(k)} \in \mathbb{N}_{>0}$ und $w^{(j)} : X \to \mathbb{N}$ für alle $j \in \{1, \ldots, k\}$ gibt mit der Eigenschaft, dass für alle $Y \in 2^X$ gilt:

$$Y \in \mathcal{W} \Leftrightarrow \forall j \in \{1, \ldots, k\} : \sum_{x \in Y} w^{(j)}(x) \geq Q^{(j)}$$

□

$(Q^{(j)}, w^{(j)})$ heißt j-tes GMS von VGMS. Weiter gilt:

$$Y \text{ gewinnend in VGMS } \Leftrightarrow Y \text{ gewinnend in jedem der einzelnen GMS}$$

Normfall der Schreibweise ist dabei wie folgt:
$[Q^{(1)}, w_1^{(1)}, \ldots, w_n^{(1)}] \wedge \cdots \wedge [Q^{(k)}, w_1^{(k)}, \ldots, w_n^{(k)}]$ mit $w_i^{(j)} = w^{(j)}(x_i)$ und $x_1, \ldots, x_n$ als Reihenfolge.

4.1.5 Beispiel
Vertrag von Nizza
$X = \{x_1, \ldots, x_{27}\}$
$[265; 29, 29, 29, 29, 27, 27, 14, 13, 12, 12, 12, 12, 12, 10, 10, 10, 7, 7, 7, 7, 7, 4, 4, 4, 4, 4, 3] \wedge [14; 1, \ldots, 1] \wedge$
$[620; 170, 123, 122, 120, 82, 89, 47, 33, 22, 21, 21, 21, 21, 18, 17, 17, 11, 11, 11, 8, 8, 4, 4, 3, 2, 1, 1]$
VGMS kann dabei nicht auf GMS minimiert bzw. zurürckgeführt werden.

4.1.6 Konvention
Bei einem einfachen Spiel $(X, \mathcal{W})$ ist im Folgenden $X = \{x_1, \ldots, x_n\}$ und X ist auch die Menge der Variablen zur Definition von $f : \mathbb{B}^n \to \mathbb{B}$ und der darstellenden QOBDDs Variablenordnung sei $\pi = (x_1, \ldots, x_n)$. □

4.1.7 Definition
Sei $(X, \mathcal{W})$ einfaches Spiel. Zu $Y \in 2^X$ ist der charakteristische Verktor $\chi(Y) \in \mathbb{B}^n$ definiert durch:

$$\chi(Y)_i = 1 \Leftrightarrow x_i \in Y \text{ für alle } i \in \{1, \ldots, n\}$$

□

4.1.8 Definition
Sei $(X, \mathcal{W})$ einfaches Spiel.

(1) Eine Funktion $f : \mathbb{B}^n \to \mathbb{B}$ stellt $(X, \mathcal{W})$ dar, falls für alle $Y \in 2^X$ gilt:

$$Y \in W \Leftrightarrow f(\chi(Y)) = 1$$

(2) Ein QOBDD g über X stellt $(X, \mathcal{W})$ dar, falls es die Funktion f darstellt, die auch $(X, \mathcal{W})$ darstellt.

□

4.1.9 Beispiel
(1) Spiel von Bsp. 4.1.3 (1)
(2) Bundestag Sept. 2009

4.1.10 Definition
Sei g ein QOBDD und $w = (p_1, \ldots, p_k)$ ein Weg von einem inneren Knoten in g zu $\mathbb{I}$. Dann definiert man

$$\underline{eins}(w) = \{\underline{var}(\alpha(p_i)) \mid 1 \leq i \leq k \wedge p_i \text{ 1-Pfeil}\}$$

□

4.1.11 Satz
Es sei $(X, \mathcal{W})$ ein einfaches Spiel und r Wurzel des QOBDDs, das $(X, \mathcal{W})$ darstellt. Dann gilt für alle $Y \in 2^X$:

$$Y \in W \Leftrightarrow \text{ Es gibt Weg } w \text{ von } r \text{ nach } \mathbb{I} \text{ mit } Y = \underline{eins}(w)$$

□

4.1.12 Definition
Es sei ein QOBDD g mit Knotenmengen V gegeben. Dann ist die Funktion $\underline{set} : V \to 2^X$ definiert durch:

$$\underline{set}(a) = \begin{cases} \{\underline{eins}(w) \mid w \text{ ist Weg von } a \text{ nach } \mathbb{I}\} & , a \text{ ist innerer Knoten} \\ \{\emptyset\} & , a = \mathbb{I} \\ \emptyset & , a = \mathbb{O} \end{cases}$$

□

4.1.13 Satz
Es seien g QOBDD und a innerer Knoten. Dann gilt:

$$\underline{set}(a) = \{Y + \underline{var}(a) \mid Y \in \underline{set}(\underline{then}(a))\} \cup \underline{set}(\underline{else}(a))$$

Beweis

$$\begin{aligned} \underline{set}(a) &= \{eins(w) \mid w \text{ Weg von } a \text{ nach } \mathbb{I}\} \\ &= \{eins(w) \mid w \text{ Weg von } a \text{ nach } \mathbb{I} \ \wedge \ w_1 \text{ 1-Pfeil}\} \cup \{eins(w) \mid w \text{ Weg von } a \text{ nach } \mathbb{I} \ \wedge \ w_1 \text{ 0-Pfeil}\} \\ &= \{eins(w) + \underline{var}(a) \mid w \text{ Weg von } \underline{then}(a) \text{ nach } \mathbb{I}\} \cup \{eins(w) \mid w \text{ Weg von } \underline{else}(a) \text{ nach } \mathbb{I}\} \\ &= \{Y + \underline{var}(a) \mid \exists \text{ Weg } w \text{ von } \underline{then}(a) \text{ nach } \mathbb{I} \text{ mit } Y = eins(w)\} \cup \underline{set}(\underline{else}(a)) \\ &= \{Y + \underline{var}(a) \mid Y \in \underline{set}(\underline{then}(a)\} \cup \underline{set}(\underline{else}(a)) \end{aligned}$$

□

Wenn $(X, \mathcal{W})$ ein einfaches Spiel ist und r die Wurzel des dargestellten QOBDDs, dann gilt $\underline{set}(r) = \mathcal{W}$.

4.1.14 Satz
Es sei $(X, \mathcal{W})$ ein GMS mit gewichteter Darstellung $[Q; w_1, \ldots w_n]$. Definiere für alle $i \in \{1, \ldots, n+1\}$ und $q \in \mathbb{R}$ (die Funktion $f : \{1, \ldots, n+1\} \times \mathbb{R} \to QOBDD$):

$$f(i, q) = \begin{cases} \mathbb{I} & , i = n+1 \wedge q \leq 0 \\ \mathbb{O} & , i = n+1 \wedge q > 0 \\ \underline{cons}(x_i, f(i+1, q - w_i), f(i+1, q)) & , \text{sonst} \end{cases}$$

Dann gilt:

$$\underline{set}(f(i, q)) = \left\{ Y \in 2^{\{x_i, \ldots, x_n\}} \middle| \sum_{x \in Y} w(x) \geq q \right\}$$

Insbesondere gilt:

$$\underline{set}(f(1, Q)) = \left\{ Y \in 2^X \middle| \sum_{x \in Y} w(x) \geq Q \right\} = \mathcal{W}$$

Beweis
Induktion nach Differenz von $n + 1 - i$

IA: $n+1-i=0$, d.h. $i=n+1$
Fall 1. $q \leq 0$. Dann gilt

$$f(i,q) = \mathbb{I}$$
$$\left\{ Y \in 2^{\emptyset} \,\middle|\, \sum_{x \in Y} w(x) \geq q \right\} = \{\emptyset\} = \underline{set}(\mathbb{I}) = \underline{set}(f(i,q))$$

Fall 2. $q > 0$. Dann gilt

$$f(i,q) = \mathbb{O}$$
$$\left\{ Y \in 2^{\emptyset} \,\middle|\, \sum_{x \in Y} w(x) \geq q \right\} = \emptyset = \underline{set}(\mathbb{O}) = \underline{set}(f(i,q))$$

IS: Sei $n+1-i > 0$, d.h. $i < n+1$. Dann gilt für alle $Y \in 2^X$

$$\begin{aligned}
Y \in \underline{set}(f(i,q)) &\Leftrightarrow Y \in \underline{set}(\underline{cons}(x_i, f(i+1, q-w_i), f(i+1,q))) \\
&\Leftrightarrow (Y \subseteq \{x_i, \ldots, x_n\} \wedge x_i \in Y \wedge \sum_{x \in Y - x_i} w(x) \geq q - w_i) \\
&\quad \vee (Y \subseteq \{x_i, \ldots, x_n\} \wedge x_i \in Y \wedge \sum_{x \in Y - x_i} w(x) \geq q) \\
&\Leftrightarrow (Y \subseteq \{x_i, \ldots, x_n\} \wedge x_i \in Y \wedge \sum_{x \in Y} w(x) \geq q) \\
&\quad \vee (Y \subseteq \{x_i, \ldots, x_n\} \wedge x_i \in Y \wedge \sum_{x \in Y} w(x) \geq q) \\
&\quad \Leftrightarrow Y \subseteq \{x_1, \ldots, x_n\} \wedge x_i \in Y \wedge \sum_{x \in Y} \leq q
\end{aligned}$$

□

4.1.15 Beispiel
Sei $X = \{x_1, x_2, x_3\}$ gewichtete Darstellung $[6; 5, 5, 1]$. Bestimme $f(1,6)$.

4.2 Bestimmung von Schlüsselspielern

4.2.1 Definition
Sei $(X, \mathcal{W})$ einfaches Spiel.

(1) Eine gewinnende Koalition $Y \in \mathcal{W}$ heißt minimal gewinnend, falls $Z \notin \mathcal{W}$ für alle $Z \subseteq Y$.

(2) $\mathcal{W}_{min} = \{Y \in W \mid Y \text{ minimal gewinnend}\}$

□

4.2.2 Definition
Sei $(X, \mathcal{W})$ einfaches Spiel. Ein Spieler $x \in X$ heißt

(1) Diktator, falls $\mathcal{W}_{min} = \{\{x\}\}$,

(2) Vetospieler, falls für alle $Y \in \mathcal{W}$ gilt: $x \in Y$,

(3) belanglos, falls für alle $Y \in \mathcal{W}_{min}$ gilt: $x \notin Y$.

□

Der Diktator ist am mächtigsten; Vetospieler können nichts erzwingen, aber alles verhindern; belanglose Spieler haben keinerlei Macht.

4.2.3 Satz
In einem einfachen Spiel gibt es höchstens einen Diktator. □

4.2.4 Satz
Ist $x \in X$ ein Diktator im einfachen Spiel $(X, \mathcal{W})$, so gilt für alle $y \in X \setminus \{x\} : y$ ist belanglos.

Beweis
Da x Diktator ist, gilt $\mathcal{W}_{min} = \{\{x\}\}$. Ist $Y \in \mathcal{W}_{min}$, so gilt $Y = \{\{x\}\}$, also gilt $y \notin Y$, da $y \neq x$. Damit ist y belanglos.

□

4.2.5 Beispiel
(1) Im Spiel, dass den Bundestag von 1957 modelliert ist x_1 Diktator. Dabei sei $X = \{x_1, x_2, x_3, x_4\}$ mit gewichteter Darstellung $[260; 277, 181, 44, 17]$.

(2) Sei $X = \{x_1, \ldots, x_15\}$ mit gewichteter Darstellung $[39; 7, 7, 7, 7, 7, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$. Dieses Spiel modelliert den UN Sicherheitsrat. Die Darstellung als VGMS mit 2 GMS sieht dabei wie folgt aus: $[5; 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] \ \wedge \ [9; 1, \ldots, 1]$

(3) Im Spiel, das den 1. EU Vetrag von 1958 modelliert, ist x_6 (Luxemburg) ein belangloser Spieler. Dabei sei $X = \{x_1, \ldots, x_6\}$ GMS mit gewichteter Darstellung $[12; 4, 4, 4, 2, 2, 1]$.

4.2.6 Lemma
Es sei $(X, \mathcal{W})$ einfach Spiel. Dann gilt für alle $x \in X$:

$$x \text{ belanglos } \Leftrightarrow \forall\, Y \in \mathcal{W} : Y - x \in \mathcal{W}$$

Beweis
„$\Rightarrow$“: Sei $Y \in \mathcal{W}$ beliebig. Der Fall, dass $x \notin Y$ ist klar, daher sei $x \in Y$. Definiere:

$$\mathcal{K} := \{Z \in \mathcal{W} \mid x \in Z \wedge Z \subseteq Y\}$$

$\mathcal{K} \neq \emptyset$, da $Y \in \mathcal{K}$. Wegen $|X| < \infty$ gilt auch $\mathcal{K} < \infty$ und damit existiert in $\mathcal{K}$ mindestens ein Element Z_0.
Angenommen $Y - x \notin \mathcal{W}$. Dann gilt $Z_0 - x \notin \mathcal{W}$, denn $Z_0 - x \in \mathcal{W}$ würde $Y - x \in \mathcal{W}$ implizieren (Monotonie). Also gilt Z_0 minimal gewinnend mit $x \in Z_0$. Das ist ein Widerspruch sazu, dass x belanglos ist.
„$\Leftarrow$“: Sei x nicht belanglos. Dann gibt es $Z \in \mathcal{W}_{min}$ mit $x \in Z$. Da $Z \in \mathcal{W}_{min}$ folgt $Z \in \mathcal{W}$. Also gilt nach Annahme $Z - x \in \mathcal{W}$ und damit wiederum $Z \notin \mathcal{W}_{min}$. Widerspruch! □

Folgende Aussagen sind ebenfall äquivalent:

$$x \text{ belanglos } \Leftrightarrow \forall\, Y \in 2^X : Y \in \mathcal{W} \leftrightarrow Y - x \in \mathcal{W}$$

Sei nun f eine Funktion, die QOBDD darstellt und $x = x_i$. Dann gilt:

$$x_i \text{ belanglos } \Leftrightarrow \forall\, y_1, \ldots, y_n : \ f(y_1, \ldots, y_{i-1}, 1, y_{i+1}, \ldots, y_n) = f(y_1, \ldots, y_{i-1}, 0, y_{i+1}, \ldots, y_n)$$

4.2.7 Satz
Sei $(X, \mathcal{W})$ durch QOBDD g dargestellt. Dann gilt für alle $x \in X$:

$$x_i \text{ belanglos } \Leftrightarrow \text{ für alle } a \in V \text{ mit } \underline{var}(a) = x_i \text{ gilt } \underline{then}(a) = \underline{else}(a)$$

□

4.2.8 Satz
Seien $(X, \mathcal{W})$ einfach Spiel, g darstellendes QOBDD und $x \in X$. Dann gilt:

$$x \text{ Vetospieler } \Leftrightarrow \text{ Für alle Knoten } a \text{ mit } \underline{var}(a) = x \text{ gilt } \underline{set}(\underline{else}(a)) = \emptyset$$

Beweis
Es soll bewiesen werden:

$$x \text{ kein Vetospieler } \Leftrightarrow \exists \text{ Knoten } a \text{ und } \underline{var}(a) = x \text{ und } \underline{set}(\underline{else}(a)) \neq \emptyset$$

Zum Beweis sei o.B.d.A. $x = x_i$ angenommen.

„$\Rightarrow$“ Sei x_i kein Vetospieler. Dann gibt es $Y \in W$ mit $x \notin Y$. Also gibt es einen Weg von der Wurzel r zu $\mathbb{I}$ mit $Y = \underline{eins}(w)$. Sei $w = (p_1, \ldots, p_n)$. Dann ist $\alpha(p_i) \notin Y$ und somit p_i 0-Pfeil.
Es gilt mit $a := \alpha(p_i)$, dass $\underline{var}(a) = x_i$. Definiere nun $w' := (p_{i+1}, \ldots, p_n)$. Dann ist w' Weg von $\underline{else}(a)$ nach $\mathbb{I}$. (Damit gilt $\underline{eins}(w) \neq \emptyset$.) Weil w' ein Weg von $\underline{else}(a)$ nach $\mathbb{I}$ ist, gilt $\underline{set}(\underline{else}(a)) \neq \emptyset$.

„$\Leftarrow$“ Es sei a ein Knoten mit $\underline{var}(a) = x_i$ und $\underline{set}(\underline{else}(a)) \neq \emptyset$. Damit gibt es einen Weg w' von $\underline{else}(a)$ nach $\mathbb{I}$. Ergänze w' zu einem Weg von r nach $\mathbb{I}$ namens w. Sei nun $Y := \underline{set}(w)$. Dann gilt $Y \in \mathcal{W}$. Nach Konstruktion von w gilt, dass w_i ein 0-Pfeil ist. Damit $x_i \notin \underline{set}(w)$, also $x_i \notin Y$. Also ist x_i kein Vetospieler. □

4.2.9 Satz
Es seien $(X, \mathcal{W})$ einfach Spiel, g darstellendes QOBDD und x_i Spieler. Dann gilt x_i Vetospieler genau dann, wenn für alle Knoten der Schicht i, $\mathbb{I}$ über alle Wege über den $\underline{else}$-Nachfolger nicht erreichbar ist. □

4.2.10 Lemma
Es sei $(X, \mathcal{W})$ ein einfaches Spiel. Dann gilt für alle $x \in X$:

$$x \text{ Diktator } \Leftrightarrow \mathcal{W} = \{Y \in 2^X \mid x \in Y\}$$

Beweis
„$\Longrightarrow$“ zu zeigen: $Y \in \mathcal{W} \Leftrightarrow x \in Y$ für alle $Y \in 2^X$
„$\Rightarrow$“ Wenn $Y \in \mathcal{W}$, dann gibt es $Z \in \mathcal{W}_{min}$ mit $Z \subseteq Y$ (da $|X| < \infty$). Da x Diktator ist, gilt $\mathcal{W}_{min} = \{\{x\}\}$, also $Z = \{x\}$, also $x \in Y$.
„$\Leftarrow$“ Sei $x \in Y$. Dann $\{x\} \subseteq Y$. Da x ein Diktator ist, gilt $\{x\} \in \mathcal{W}_{min} \subseteq W$. Monotonie und $\{x\} \in Y$ bringen $Y \in \mathcal{W}$.

„$\Longleftarrow$“ Es gilt $\{x\} \in \mathcal{W}$, da $\{x\} \in 2^X$ und $x \in \{x\}$ und vorausgesetzer Gleichheit. Also gilt $\{x\} \in \mathcal{W}_{min}$. Sei $Y \in \mathcal{W}_{min}$. Dann gelten $Y \in \mathcal{W}$ und $x \in Y$ wegen gefordeter Gleichheit. Also gilt $Y = \{x\}$ und letztendlich ist damit $\{x\}$ ein Diktator.

□

Im Folgenden soll ein spezieller QOBDD namens $\underline{ith}$ betrachtet werden. Dafür ist mit a als Wurzel des QOBDDs definiert:

$$\underline{set}(\underline{ith}(i)) = \{eins(w) \mid w \text{ Weg von } a \text{ nach } \mathbb{I}\} = \{Y \in 2^X \mid x_i \in Y\}$$

4.2.11 Satz
Es sei $(X, \mathcal{W})$ einfaches Spiel und g das darstellende QOBDD. Es ist $x_i \in X$ genau dann Diktator, wenn g gleich $\underline{ith}(i)$ ist.

□

BILD von $\underline{ith}$.

4.3 Die Wünschenswert-Relation

4.3.1 Definition
Sei $(X, \mathcal{W})$ einfaches Spiel. Die Relation $\preccurlyeq_I$ auf X ist für alle $x, y \in X$ definiert durch:

$$\begin{aligned} x \preccurlyeq_I y &\Leftrightarrow \forall\, Y \in 2^X : x \notin Y \wedge y \notin Y \wedge Y + x \in \mathcal{W} \Rightarrow Y + y \in \mathcal{W} \\ &\Leftrightarrow \forall Y \in 2^{X-x-y} : Y + x \in \mathcal{W} \Rightarrow Y + y \in \mathcal{W} \end{aligned}$$

$\preccurlyeq_I$ heißt Wünschenswert-Relation (auf Spielern) und $x \preccurlyeq_I y$ bedeutet „y ist wünschenswerter als x“.

$\square$

4.3.2 Satz
Für alle einfachen Spiele $(X, \mathcal{W})$ ist $\preccurlyeq_I$ eine Quasi-Ordnung[15].

Beweis
Reflexivität:
trivial

Transitivität:
Seien $x, y, z \in X$ mit $x \preccurlyeq_I y$ und $y \preccurlyeq_I z$. zum Beweis von $x \preccurlyeq_I z$ sei $Y \in 2^X$ mit $x \notin Y$ und $z \notin Y$ beliebig vorgegeben. Weiterhin sei $Y + x \in \mathcal{W}$.
Fall 1: $y \notin Y$ Wegen $x \notin Y$, $y \notin Y$ und $x \preccurlyeq_I y$ sowie $Y + x \in \mathcal{W}$ folgt $Y + x \in \mathcal{W}$. Analog ist $Y + x \in W$.
Fall 2: $y \in Y$.
Unterfall 1. $x = z$ trivial
Unterfall 2. $x \neq z$ Definiere $Y' := Y - y$. Dann $x, y, z \notin Y'$.

$$\begin{aligned}
Y + x \in \mathcal{W} &\Leftrightarrow Y' + y + x \in \mathcal{W} \\
(x \neq y) &\Leftrightarrow Y' + x + y \in \mathcal{W} \\
(y \notin Y' + x, z \notin Y' + x, y \preccurlyeq_I z) &\Rightarrow Y' + x + z \in \mathcal{W} \\
x \neq z &\Leftrightarrow Y' + z + x \in \mathcal{W} \\
x \notin Y' + z, y \notin Y' + z, x \preccurlyeq_I y &\Leftrightarrow Y' + z + y \in \mathcal{W} \\
y \neq z &\Leftrightarrow Y' + y + z \in \mathcal{W} \\
&\Leftrightarrow Y + z \in \mathcal{W}
\end{aligned}$$

$\square$

4.3.3 Satz
Ist $(X, \mathcal{W})$ GMS mit gewichteter Darstellung (Q, w), so gilt für alle $x, y \in X$:

$$w(x) \leq w(y) \Rightarrow x \preccurlyeq_I y$$

Beweis
Sei $Y \in 2^X$ mit $x \notin Y$ und $y \notin Y$. Dann gilt

$$\begin{aligned}
Y + x \in \mathcal{W} &\Leftrightarrow w(x) + \sum_{z \in Y} w(z) \geq Q \\
&\Rightarrow w(y) + \sum_{z \in Y} w(z) \geq Q \\
&\Leftrightarrow Y + y \in \mathcal{W}
\end{aligned}$$

$\square$

4.3.4 Satz
Die Relation $\preccurlyeq_I$ und $\approx_I$ auf X sind für ein einfach Spiel $(X, \mathcal{W})$ und alle Spieler $x, y \in X$ definiert durch:

$$x \prec_I y \Leftrightarrow (x \preccurlyeq_I y) \ \wedge \ \neg(y \preccurlyeq_I x)$$

„y echt wünschenswerter als x“

[15] also reflexiv und transitiv

$$x \approx_I y \Leftrightarrow (x \preccurlyeq_I y) \ \wedge \ (y \preccurlyeq_I x)$$
„x,y gleich wünschenswert“

Dabei gilt, dass $\prec_I$ eine strikte Quasi-Ordnung[16] und $\approx_I$ eine Äquivalenzrelation[17] ist. □

4.3.5 Lemma
Es sei $(X, \mathcal{W})$ ein einfaches Spiel, dann sind für alle $x, y \in X$ mit $x \neq y$ die folgenden zwei Aussagen äquivalent:

(1) $x \preccurlyeq_I y$
(2) $\{Y - x \mid x \in Y \wedge Y \in \mathcal{W} \wedge y \notin Y\} \subseteq \{Y - y \mid y \in Y \wedge Y \in \mathcal{W}\}$

Beweis
„(1) $\Rightarrow$ (2)“ Es gelte $x \preccurlyeq_I y$. Sei $Z \in \{Y - x \mid x \in Y \wedge Y \in \mathcal{W} \wedge y \notin Y\}$. Also gibt es Y mit $Y \in \mathcal{W}$, $x \in Y$, $y \notin Y$ und $Z = Y - x$.

$$\begin{aligned} Y \in W &\Leftrightarrow Y - x + x \in W && \text{wegen } x \in Y \text{ gilt } Y = Y - x + x \\ &\Leftrightarrow Z + x \in \mathcal{W} \\ &\Rightarrow Z + y \in \mathcal{W} && \text{wegen } x \preccurlyeq_I y, x \notin Z \text{ und } y \notin Z \end{aligned}$$

Definiere nun $Y' := Z + y$. Dann gilt $Y' \in \mathcal{W}$. Weiter gilt $y \in Y'$. Also ist $Y' - y \in \{Y - y \mid y \in Y \wedge Y \in \mathcal{W}\}$. Weiter ist $y \notin Y$, also $Y' - y = Z + y - y = Z$ und letztendlich $Z = Y' - y$. Damit gilt $Z \in \{Y - y \mid y \in Y \wedge Y \in \mathcal{W}\}$.

„(2) $\Rightarrow$ (1)“ Es gelte die Mengeninklusion. Zum Beweis von $x \preccurlyeq_I y$ sei $Y \in 2^X$ mit $x \notin Y$, $y \notin Y$ und $Y + x \in \mathcal{W}$ gegeben.
Definiere nun $Z = Y + x$. Dann ist $Z \in \mathcal{W}$, $x \in Z$ und $y \notin Z$ (wegen $x \neq y$). Damit gilt $Z - x \in \{Y - x \mid x \in Y \wedge Y \in \mathcal{W} \wedge y \notin Y\}$. Folglich gilt auch $Z \in \{Y - y \mid y \in Y \wedge Y \in \mathcal{W}\}$. Wegen $Z - x = Y$ ist $Y \in \{Y - x \mid x \in Y \wedge Y \in \mathcal{W} \wedge y \notin Y\}$. Dann gibt es $Z' \in W$ mit $y \in Z'$ und $Y = Z' - y$. Dann ist $Y + y = Z' - y + y = Z' \in \mathcal{W}$.

□

Die Voraussetzung für solch ein Vorgehen ist eine Operation `Remove` auf QOBDDs, so dass für alle Knoten a und alle $i \in \{1, \ldots, n\}$ gilt:

$$\underline{set}(\texttt{Remove}(a, i))) = \{Y - x_i \mid Y \in \underline{set}(a) \wedge x_i \in Y\}$$

4.3.6 Lemma (1) Für alle QOBDDs $\underline{ith}(i)$ gilt:

$$\underline{set}(\texttt{Neg}(\underline{ith}(i)) = 2^X \setminus \underline{set}(\underline{ith}(i)) = 2^{X - x_i}$$

(2) Für alle QOBDDs g_1 und g_2 gilt mit $g_1 \sqcap g_2 = \texttt{Apply}(g_1, g_2, \cdot)$:

$$\underline{set}(g_1 \sqcap g_2) = \underline{set}(g_1) \cap \underline{set}(g_2)$$

□

4.3.7 Satz
Es sei $(X, \mathcal{W})$ einfaches Spiel. Dann sind für alle $x_i, x_j \in X$ mit $i \neq j$ die folgenden Aussagen äquivalent:

(1) $x_i \preccurlyeq_I x_j$
(2) Ist g das QOBDD zur Darstellung von $(X, \mathcal{W})$ mit Wurzel r so ist
$\texttt{Remove}(r \sqcap \texttt{Neg}(\underline{ith}(j)), i) \sqcap \texttt{Neg}(\texttt{Remove}(r, j))$ gleich dem QOBDD zur Darstellung von $\mathbf{0} : \mathbb{B}^n \to \mathbb{B}$

[16] asymmetrisch und transitiv bzw. irreflexiv und transitiv
[17] symmetrisch, transitiv und reflexiv

Beweis
Sei r Wurzel von g. Dann gilt $\underline{set}(r) = \mathcal{W}$.

$$\begin{aligned} x_i \preccurlyeq_I x_j &\Leftrightarrow \{Y - x_i \mid x_i \in Y \wedge Y \in \mathcal{W} \wedge x_j \notin Y\} \subseteq \{Y - x_j \mid x_j \in Y \wedge Y \in \mathcal{W}\} \\ &\Leftrightarrow \{Y - x_i \mid x_i \in Y \wedge Y \in \underline{set}(r) \wedge x_j \notin Y\} \subseteq \{Y - x_j \mid x_j \in Y \wedge Y \in \underline{set}(r)\} \end{aligned}$$

$$\begin{aligned} \{Y - x_i \mid x_i \in Y \wedge Y \in \underline{set}(r) \wedge x_j \notin Y\} &= \{Y - x_i \mid Y \in 2^{X - x_j} \wedge Y \in \underline{set}(r) \wedge x_i \in Y\} \\ &= \{Y - x_i \mid Y \in \underline{set}(\texttt{Neg}(\underline{ith}(j))) \wedge Y \in \underline{set}(r) \wedge x_i \in Y\} \\ &= \{Y - x_i \mid Y \in \underline{set}(r \sqcap \texttt{Neg}(\underline{ith}(j))) \wedge x_i \in Y\} \\ &= \underline{set}(\texttt{Remove}(r \sqcap \texttt{Neg}(\underline{ith}(j)), i)) \end{aligned}$$

$$\{Y - x_j \mid x_j \in Y \wedge Y \in \underline{set}(r)\} = \underline{set}(\texttt{Remove}(r, j))$$

Also gilt nun insgesamt:

$$\begin{aligned} x_i \preccurlyeq_I x_j &\Leftrightarrow \underline{set}(\underbrace{\texttt{Remove}(r \sqcap \texttt{Neg}(\underline{ith}(j)), i)}_{f_1 : \mathbb{B}^n \to \mathbb{B}}) \subseteq \underline{set}(\underbrace{\texttt{Remove}(r, j)}_{f_2 : \mathbb{B}^n \to \mathbb{B}}) \\ &\Leftrightarrow f_1 \sqsubseteq f_2 \\ &\Leftrightarrow f_1 \sqcap \overline{f_2} = \mathbf{0} \\ &\Leftrightarrow \texttt{Remove}(r \sqcap \texttt{Neg}(\underline{ith}(j)), i) \sqcap \texttt{Neg}(\texttt{Remove}(r, j)) \\ &\quad \text{ist QOBDD zur Darstellung der Funktion } \mathbf{0} : \mathbb{B}^n \to \mathbb{B} \end{aligned}$$

□

Beispiel
$\underline{set}(r) = \{\{x_1, x_2, x_3\}, \{x_1, x_3\}, \{x_2, x_3\}\}$
$\underline{set}(\texttt{Remove}(r, 1)) = \{\{x_2, x_3\}, \{x_3\}\}$
Bildchen...

4.3.8 Satz

Die Funktion

$$\texttt{Remove}(a, i) = \begin{cases} \underline{cons}(x_i, \mathbb{O}_{i+1}, \underline{then}(a)) & , \text{ falls } \underline{var}(a) = x_i \\ \underline{cons}(\underline{var}(a), \texttt{Remove}(\underline{then}(a), i), \texttt{Remove}(\underline{else}(a), i)) & , \text{ falls } \underline{var}(a) > x_i \\ a & , \text{ sonst} \end{cases}$$

erfüllt die Gleichung

$$\underline{set}(\texttt{Remove}(a, i)) = \{Y - x_i \mid Y \in \underline{set}(a) \wedge x_i \in Y\}$$

4.3.9 Algorithmus Remove

$\texttt{Remove}(a, i)$	if $\underline{var}(a) = x_i$ then return $(\underline{cons}(x_i, \mathbb{O}_{i+1}, \underline{then}(a))$ elseif $\exists r$ in CT with (a, i, r) then return r with (a, i, r) in CT else $r := \underline{cons}(\underline{var}(a), \texttt{Remove}(\underline{then}(a), i), \texttt{Remove}(\underline{else}(a), i))$ add (a, i, r) to CT return r

4.3.10 Beispiel

Sei $X = \{x_1, x_2, x_3, x_4\}, \pi = (x_1, x_2, x_3)$
$\underline{set}(a) = \{\{x_1, x_2, x_3, x_4\}, \{x_1, x_2, x_4\}, \{x_1, x_3, x_4\}, \{x_1, x_4\}, \{x_2, x_3, x_4\}, \{x_2, x_4\}, \{x_3, x_4\}, \{x_4\}\}$
$\underline{set}(\texttt{Remove}(a, 3)) = \{\{x_1, x_2, x_4\}, \{x_1, x_4\}, \{x_2 . x_4\}, \{x_4\}\}$

4.3.11 Beispiel

Siehe Tabelle 4.3.11.

$\preccurlyeq_I$	CDU/CSU	SPD	FDP	Die Linke	Die Grünen
CDU/CSU	X	O	O	O	O
SPD	X	X	Y	Y	O
FDP	X	X	X	Y	O
Die Linke	X	X	X	X	O
Die Grünen	X	X	X	X	X

$\approx_I$	CDU/CSU	SPD	FDP	Die Linke	Die Grünen
CDU/CSU	X	O	O	O	O
SPD	O	X	X	X	O
FDP	O	X	X	X	O
Die Linke	O	X	X	X	O
Die Grünen	O	O	O	O	X

$\prec_I$	CDU/CSU	SPD	FDP	Die Linke	Die Grünen
CDU/CSU	O	O	O	O	O
SPD	X	O	O	O	O
FDP	X	O	O	O	O
Die Linke	X	O	O	O	O
Die Grünen	X	X	X	X	O

Tabelle 2: Alle Einträge bis auf die Ys ergeben sich aus der Quote ($w(x) < w(y) \Rightarrow x \preccurlyeq_I y$)

4.4 Eigenschaften von einfachen Spielen

Bisher wurde nur zwischen GMS und VGMS unterschieden.

4.4.1 Definition
Ein einfaches Spiel $(X, \mathcal{W})$ heißt

(1) echt, falls für alle $Y \in \mathcal{W}$ gilt: $X \setminus Y \notin \mathcal{W}$

(2) fest, falls für alle $Y \in 2^X \setminus W$ gilt: $X \setminus Y \in W$

(3) entscheidend, falls es echt und fest ist

(4) dual-gleichwertig, falls es echt oder fest ist

□

4.4.2 Satz
Für alle einfachen Spiele $(X, \mathcal{W})$ gilt:

$$(X, \mathcal{W}) \text{ entscheidend } \Rightarrow |\mathcal{W}| = 2^{|X|-1}$$

Beweis
Definiere $\Phi : 2^X \to 2^X,\ \Phi(Y) = X \setminus Y$. Dann gilt $\Phi^2 = id$, also Φ bijektiv.

$$\begin{aligned}
(X, \mathcal{W}) \text{ echt } &\Leftrightarrow \Phi(\mathcal{W}) \subseteq 2^X \setminus \mathcal{W} \\
&\Rightarrow |\mathcal{W}| = |\Phi(\mathcal{W})| \leq |2^X \setminus \mathcal{W}| \\
(X, \mathcal{W}) \text{ fest } &\Leftrightarrow \Phi(2^X \setminus \mathcal{W}) \subseteq \mathcal{W} \\
&\Rightarrow |2^X \setminus \mathcal{W}| = |\Phi(2^X \setminus \mathcal{W})| \leq |W|
\end{aligned}$$

□

4.4.3 Satz
Ist $(X, \mathcal{W})$ ein GMS, so gilt:

(1) $(X, \mathcal{W})$ ist echt oder fest

(2) $|\mathcal{W}| = 2^{|X|-1} \Rightarrow (X, \mathcal{W})$ entscheidend

Beweis
(1) Es sei (Q, w) gewichtete Darstellung und $(X, \mathcal{W})$ weder echt noch fest.

$$(X, \mathcal{W}) \text{ nicht echt } \Rightarrow \exists Y \in 2^X : Y \in \mathcal{W} \ \wedge \ X \setminus Y \in \mathcal{W}$$

Dann gilt für so ein Y:

$$\sum_{x \in Y} w(x) \geq Q \text{ und } \sum_{x \in X \setminus Y} w(x) \geq X \text{ also auch } \sum_{x \in X} w(x) = \sum_{x \in Y} w(x) + \sum_{x \in X \setminus Y} \geq 2 \cdot Q$$

$$(X, \mathcal{W}) \text{ nicht fest } \Rightarrow \exists Z \in 2^X : Z \in 2^X \setminus \mathcal{W} \ \wedge \ X \setminus Z \in 2^X \setminus \mathcal{W}$$

Dann gilt für so ein Z:

$$\sum_{x \in Z} w(x) < Q \text{ und } \sum_{x \in X \setminus Z} w(x) < X \text{ also insgesamt } \sum_{x \in X} w(x) = \sum_{x \in Z} w(x) + \sum_{x \in X \setminus Z} < 2 \cdot Q$$

Widerspruch!

(2) Sei $|\mathcal{W}| = 2^{|X|-1}$, aber $(X, \mathcal{W})$ nicht entscheidend.
Fall 1: Es sei $(X, \mathcal{W})$ echt, aber nicht fest.

$$\begin{aligned} (X, W) \text{ echt } &\Leftrightarrow \Phi(\mathcal{W}) \subseteq 2^X \setminus \mathcal{W} \\ &\Rightarrow |\mathcal{W}| \leq |2^x \setminus \mathcal{W}| \end{aligned}$$

Da $|\mathcal{W}| = 2^{|X|-1}$ gilt und $(\mathcal{W}, 2^X \setminus \mathcal{W})$ eine Partition bilden, gilt:

$$|W| = |2^X \setminus \mathcal{W}| = |\Phi(\mathcal{W})|$$

Also gilt $\Phi(\mathcal{W}) = 2^X \setminus \mathcal{W}$. Und damit auch $\mathcal{W} = \Phi(\Phi(\mathcal{W})) = \Phi(2^X \setminus \mathcal{W})$. Also insgesamt $\Phi(2^X \setminus \mathcal{W}) \subseteq \mathcal{W}$, d.h. $(X, \mathcal{W})$ ist fest.
Widerspruch!

Fall 2: Analog.

□

4.4.4 Algorithmus Decisive

Der folgende Algorithmus testet, ob ein GMS $(X, \mathcal{W})$ entscheidend ist.

`Decisive`(Q,w)

(1) Berechne das QOBDD zur Darstellung des Spiels mit Hilfe von Satz 4.1.14
(2) Zähle in diesem QOBDD die Anzahl m der erfüllenden Belegungen nach Satz 3.2.5
(3) return $m = 2^{|X|-1}$

□

4.4.5 Definition

Sei $(X, \mathcal{W})$ einfaches Spiel.

(1) Eine Koalition $Y \in 2^X$ heißt blockierend, falls $X \setminus Y \notin \mathcal{W}$.

(2) Es heißt $\mathcal{W}^d = \{Y \in 2^X \mid Y \text{ blockierend}\}$ Dual von $\mathcal{W}$.

(3) $(X, \mathcal{W}^d)$ ist das zu $(X, \mathcal{W})$ duale einfache Spiel.

□

Beachte dabei, dass $Y \in \mathcal{W}^d$ und $Y \subseteq Z \Rightarrow Z \in \mathcal{W}^d$ gilt.

4.4.6 Beispiel
(1) (fest und echt)
$X = \{x_1, x_2, x_3\},\ \mathcal{W} = \{\{x_1, x_2, x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}\}$[18]

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$\mathcal{W}$	0	0	0	1	0	1	1	1
$\mathcal{W}^d$	0	0	0	1	0	1	1	1

(2) (nicht fest, aber echt)
$X = \{x_1, x_2, x_3\},\ \mathcal{W} = \{\{x_1, x_2, x_3\}, \{x_2, x_3\}\}$

x_1	0	0	0	0	1	1	1	1
x_2	0	0	1	1	0	0	1	1
x_3	0	1	0	1	0	1	0	1
$\mathcal{W}$	0	0	0	1	0	0	0	1
$\mathcal{W}^d$	0	1	1	1	0	1	1	1

4.4.7 Satz
Es sei $(X, \mathcal{W})$ einfaches Spiel. Dann gelten die Äquivalenzen:

(1) $(X, \mathcal{W})$ echt $\Leftrightarrow \mathcal{W} \subseteq \mathcal{W}^d$

(2) $(X, \mathcal{W})$ fest $\Leftrightarrow \mathcal{W}^d \subseteq \mathcal{W}$

Beweis
(1) „$\Rightarrow$“ Sei $Y \in 2^X$, dann gilt:

$$\begin{aligned} Y \in \mathcal{W} &\Rightarrow X \setminus Y \in 2^X \setminus \mathcal{W} && \text{wegen } (X, \mathcal{W}) \text{ echt} \\ &\Leftrightarrow Y \in \mathcal{W}^d && \text{Def. von } \mathcal{W}^d \end{aligned}$$

Also $\mathcal{W} \subseteq W^d$.
„$\Leftarrow$“ Seien $\mathcal{W} \subseteq W^d$ und $Y \in 2^X$.

$$\begin{aligned} Y \in W &\Rightarrow Y \in W^d && \text{wegen } \mathcal{W} \subseteq \mathcal{W}^d \\ &\Leftrightarrow X \setminus Y \in 2^x \setminus \mathcal{W} && \text{Def. von echt} \end{aligned}$$

Also ist $(X, \mathcal{W})$ echt.

(2) Analog.

□

Für a als QOBDD-Knoten mit $\underline{var}(a) = x_i$ soll im Folgenden gelten:

$$\underline{set}(\texttt{Compl}(a)) = \{Y \in 2^{\{x_i, \ldots, x_n\}} \mid \{x_i, \ldots, x_n\} \setminus Y \in \underline{set}(a)\}$$

[18] Bei $W = W^d$ spricht man von einem Selbstdual.

4.4.8 Satz
Es sei $(X, \mathcal{W})$ einfaches Spiel, dargestellt durch QOBDD mit Wurzel r. Dann gilt

$$(X, \mathcal{W}) \text{ echt } \Leftrightarrow \text{ Das QOBDD } \texttt{Compl}(r) \sqcap r \text{ ist gleich dem QOBDD zur Darstellung von } \mathbf{0}$$

Beweis

$$\begin{aligned}
(X, \mathcal{W}) \text{ echt } &\Leftrightarrow \mathcal{W} \subseteq \mathcal{W}^d && \text{Def. von echt}\\
&\Leftrightarrow W \subseteq \{Y \in 2^X \mid X \setminus Y \notin \mathcal{W}\} && \text{Def. von } \mathcal{W}^d\\
&\Leftrightarrow \underline{set}(r) \subseteq \{Y \in 2^X \mid X \setminus Y \notin \underline{set}(r)\} && \mathcal{W} = \underline{set}(r)\\
&\Leftrightarrow \underline{set}(r) \subseteq \{Y \in 2^x \mid X \setminus Y \in \underline{set}(\texttt{Neg}(r))\} && \text{Eig. von } \texttt{Neg}\\
&\Leftrightarrow \underline{set}\underbrace{(r)}_{f_1} \subseteq \underline{set}\underbrace{(\texttt{Compl}(\texttt{Neg}(r)))}_{f_2} && (*)\\
&\Leftrightarrow f_1 \sqsubseteq f_2\\
&\Leftrightarrow \overline{f_2} \sqcap f_1 = \mathbf{0}\\
&\Leftrightarrow \texttt{Neg}(\texttt{Compl}(\texttt{Neg}(r))) \sqcap r \text{ ist QOBDD für } \mathbf{0}\\
&\Leftrightarrow \texttt{Compl}(r) \sqcap r \text{ ist QOBDD für } \mathbf{0}
\end{aligned}$$

□

4.4.9 Satz
Es sei $(X, \mathcal{W})$ einfaches Spiel, dargestellt durch QOBDD mit Wurzel r. Dann gilt:

$$(X, \mathcal{W}) \text{ fest } \Leftrightarrow \text{Das QOBDD } \texttt{Neg}(\texttt{Compl}(r) \sqcap \texttt{Neg}(r)) \text{ ist gleich dem QOBDD zur Darstellung von } \mathbf{0}$$

Beweis

$$\begin{aligned}
(X, \mathcal{W}) \text{ fest } &\Leftrightarrow \mathcal{W}^d \subseteq \mathcal{W}\\
&\Leftrightarrow \{Y \in 2^X \mid X \setminus Y \notin \mathcal{W}\} \subseteq \mathcal{W}\\
\& \{Y \in 2^X \mid X \setminus Y \notin \underline{set}(r)\} \subseteq \underline{set}(r)\\
&\Leftrightarrow \{Y \in 2^X \mid X \setminus Y \in \underline{set}(\texttt{Neg}(r)\} \subseteq \underline{set}(r)\\
&\Leftrightarrow \underline{set}\underbrace{(\texttt{Compl}(\texttt{Neg}(r))}_{f_1} \subseteq \underline{set}\underbrace{(r)}_{f_2}\\
&\Leftrightarrow f_1 \sqsubseteq f_2\\
&\Leftrightarrow \overline{f_2} \sqcap f_1 = \mathbf{0}\\
&\Leftrightarrow \texttt{Neg}(r) \sqcap \texttt{Compl}(\texttt{Neg}(r)) \text{ ist das QOBDD für } \mathbf{0}
\end{aligned}$$

□

Sei a wieder QOBDD-Knoten mit $\underline{var}(a) = x_i$. Dann wissen wir schon, dass gilt:

$$\underline{set}(a) = \{eins(w) \mid w \text{Weg von } a \text{ nach } \mathbb{I}\}$$

Daraus können wir nun observieren, dass folgende Gleichheit für das Komplement gelten muss:

$$\begin{aligned}
&\underline{set}(\texttt{Compl}(a)) = \{null(w) \mid w \text{ Weg von } a \text{ nach } \mathbb{I}\}\\
&\text{mit } null(p_1, \dots, p_m) = \{\underline{var}(\alpha(p_i)) \mid p_i \text{ ist 0-Pfeil } \wedge 1 \leq i \leq m\}
\end{aligned}$$

4.4.10 Satz
Sei a QOBDD-Knoten. Die Funktion `Compl` mit

$$\texttt{Compl}(a) = \begin{cases} \mathbb{I} & \text{, falls } a = \mathbb{I} \\ \mathbb{O} & \text{, falls } a = \mathbb{O} \\ \underline{cons}(\underline{var}(a), \texttt{Compl}(\underline{else}(a)), \texttt{Compl}(\underline{then}(a))) & \text{, sonst} \end{cases}$$

erfüllt die Eigenschaft $\underline{set}(\texttt{Compl}(a)) = \{Y \in 2^{\{x_i,\ldots,x_n\}} \mid \{x_i, \ldots, x_n\} \setminus Y \in \underline{set}(a)\}$. □

4.4.11 Beispiel
Sei einfaches Spiel $(X, \mathcal{W})$ gegeben mit $X = \{x_1, x_2, x_3, x_4\}$ und $\mathcal{W}$ wie folgt:

x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
x_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\mathcal{W}$	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	1
$\mathcal{W}^d$	1	1	1	1	1	0	1	0	1	1	0	0	0	0	0	0

Dieses Spiel ist nicht fest und nicht echt.
Sei $Z := \{x_2, x_4\}$, dann gilt $Z \notin \mathcal{W}$ und $X \setminus Z \notin \mathcal{W}$, also ist $(X, \mathcal{W})$ nicht fest.
Sei dann $Y := \{x_2, x_3\}$, dann gilt $Y \in \mathcal{W}$ und $X \setminus Y \in \mathcal{W}$, also $(X, \mathcal{W})$ auch nicht echt.