Präsentation
BDDs
Binary Decision Diagrams

Günther Eder — Andreas Weichhart

28.01.2011

Überblick 1/2

1. Entstehung BDDs
2. Boolean Expressions
3. Nachteil KNF/DNF
4. Shannon Expansion
5. INF
6. Definition BDDs

Überblick 2/2

7. OBDDs & ROBDDs
8. Konstruktion
9. Vorteile BDDs
10. Komplexität
11. Implementierung
12. Anwendung
13. Zusammenfassung

Entstehung

Was sind BDDs?

- Datenstruktur zur Repräsentation von Boolschen Funktionen
- Shannon Expansion

Entstehung

Was sind BDDs?

- Datenstruktur zur Repräsentation von Boolschen Funktionen
- Shannon Expansion

Wann entstanden sie?

- C.Y. Lee (1959)
- Shaldon B. Akers (1978)
- Randal Bryant (1986)

Boolean Expressions

Kurze Wiederholung

Variable	$1, 0$
Operatoren	$\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$
Boolean Expression	$t := x \| 0 \| 1 \| \neg t \| t \wedge t \| t \vee t \| t \Rightarrow t \| t \Leftrightarrow t$

Boolean Expressions

Kurze Wiederholung

Variable	$1, 0$
Operatoren	$\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$
Boolean Expression	$t := x \vert 0 \vert 1 \vert \neg t \vert t \wedge t \vert t \vee t \vert t \Rightarrow t \vert t \Leftrightarrow t$

	$\neg$
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\Rightarrow$	0	1
0	1	1
1	0	1

$\Leftrightarrow$	0	1
0	1	0
1	0	1

Boolean Expressions

Kurze Wiederholung

Variable	$1, 0$
Operatoren	$\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$
Boolean Expression	$t := x \mid 0 \mid 1 \mid \neg t \mid t \wedge t \mid t \vee t \mid t \Rightarrow t \mid t \Leftrightarrow t$

	$\neg$
0	1
1	0

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\Rightarrow$	0	1
0	1	1
1	0	1

$\Leftrightarrow$	0	1
0	1	0
1	0	1

Beispiel

$$\neg x_1 \wedge x_2 \vee x_3 \Rightarrow x_4 \quad = \quad (((\neg x_1) \wedge x_2) \vee x_3) \Rightarrow x_4$$

Nachteil KNF / DNF

Nachteil KNF / DNF

KNF $(x_0^1 \vee x_1^1) \wedge (x_0^2 \vee x_1^2) \wedge ... \wedge (x_0^n \vee x_1^n)$

Nachteil KNF / DNF

KNF $(x_0^1 \vee x_1^1) \wedge (x_0^2 \vee x_1^2) \wedge \ldots \wedge (x_0^n \vee x_1^n)$

Umformung in DNF

$$
\begin{gathered}
(x_0^1 \wedge x_0^2 \wedge \ldots \wedge x_0^{n-1} \wedge x_0^n) \vee \\
(x_0^1 \wedge x_0^2 \wedge \ldots \wedge x_0^{n-1} \wedge x_1^n) \vee \\
\vdots \\
(x_1^1 \wedge x_1^2 \wedge \ldots \wedge x_1^{n-1} \wedge x_0^n) \vee \\
(x_1^1 \wedge x_1^2 \wedge \ldots \wedge x_1^{n-1} \wedge x_1^n) \vee
\end{gathered}
$$

KNF proportional zu $n \Rightarrow$ DNF jedoch $n2^n$

Shannon Expansion

Idee

Zerlegung Boolscher Funktionen in Summe zweier Teilfunktionen.

$$F = x \cdot F_x + x' \cdot F_{x'}$$

Shannon Expansion

Idee

Zerlegung Boolscher Funktionen in Summe zweier Teilfunktionen.

$F = x \cdot F_x + x' \cdot F_{x'}$

$$\begin{aligned} \text{Beispiel:} \quad & f = xyz + xy'z + x'y'z + x'yz + x'y'z' \\ \text{umschreiben:} \quad & f = x' \cdot g_x' + x \cdot g_x \\ \text{expandieren:} \quad & f = x'(y'z + yz + y'z') + x(yz + y'z) \end{aligned}$$

INF - If then else normal form

Definition

$$x \rightarrow y_0, y_1 \quad = \quad (x \wedge y_0) \vee (\neg x \wedge y_1)$$

INF - If then else normal form

Definition

$$x \rightarrow y_0, y_1 \quad = \quad (x \wedge y_0) \vee (\neg x \wedge y_1)$$

Beispiel

$$\begin{aligned} \neg x &= (x \rightarrow 0, 1) \\ x \Leftrightarrow y &= x \rightarrow (y \rightarrow 1, 0), (y \rightarrow 0, 1) \end{aligned}$$

INF - If then else normal form

Definition

$$x \rightarrow y_0, y_1 \quad = \quad (x \wedge y_0) \vee (\neg x \wedge y_1)$$

Beispiel

$$\begin{aligned} \neg x \quad &= \quad (x \rightarrow 0, 1) \\ x \Leftrightarrow y \quad &= \quad x \rightarrow (y \rightarrow 1, 0), (y \rightarrow 0, 1) \end{aligned}$$

Jede Boolean Expression hat äquivalente INF oder ITE

If Then Else: $ITE(F, G, H) = F \cdot G + \overline{F} \cdot H$

Definition BDDs

BDD ist Graph mit den Eigenschaften

- Wurzelgraph
- gerichtet
- azyklisch

Definition BDDs

BDD ist Graph mit den Eigenschaften

- Wurzelgraph
- gerichtet
- azyklisch

Definition BDD

- Ein oder zwei Terminal Nodes mit Ausgangsgrad 0.
- Eine Menge von Nodes mit Ausgangsgrad 2.
  Die Ausgangskanten sind gegeben durch zwei Funktionen $low(u)$ und $high(u)$.
- Eine Variable $var(u)$ verknüpft mit jedem Knoten.

OBDDs & ROBDDs

OBDD - Ordered BDD

- alle Variablen respektieren Ordnungsrelation.
  $x_1 < x_2 < ... < x_n$

OBDDs & ROBDDs

OBDD - Ordered BDD

- alle Variablen respektieren Ordnungsrelation.
  $x_1 < x_2 < \ldots < x_n$

ROBDD - Reduced Ordered BDD

- Eindeutigkeit
  $var(u) = var(v), low(u) = low(v), high(u) = high(v)$
  $\Rightarrow u = v$
- ohne Redundanz
  $low(u) \neq high(u)$

Konstruktion v. Wahrheitstabelle

1. Boolsche Funktion umwandeln

Boolsche Funktion	$\rightarrow$	Wahrheitstabelle				
$a \vee (\overline{a} \wedge \overline{b})$	$\rightarrow$	a	$\vee$	$(\overline{a}$	$\wedge$	$\overline{b})$
		0		1		1
		0		1		0
		1		0		1
		1		0		0

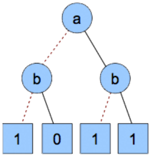
Konstruktion v. Wahrheitstabelle

1. *Boolsche Funktion umwandeln*

Boolsche Funktion	$\rightarrow$	Wahrheitstabelle				
$a \vee (\overline{a} \wedge \overline{b})$	$\rightarrow$	a	$\vee$	$(\overline{a}$	$\wedge$	$\overline{b})$
		0		1		1
		0		1		0
		1		0		1
		1		0		0

2. *Isomorphe Teilbäume umformen u. redundante Knoten entfernen*

decision tree — isomorph — BDD



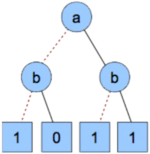
Konstruktion v. Wahrheitstabelle

1. *Boolsche Funktion umwandeln*

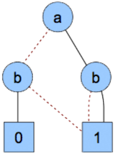
Boolsche Funktion	$\rightarrow$	Wahrheitstabelle				
$a \lor (\overline{a} \land \overline{b})$	$\rightarrow$	a	$\lor$	$(\overline{a}$	$\land$	$\overline{b})$
		0		1		1
		0		1		0
		1		0		1
		1		0		0

2. *Isomorphe Teilbäume umformen u. redundante Knoten entfernen*

decision tree isomorph BDD



$\rightarrow$



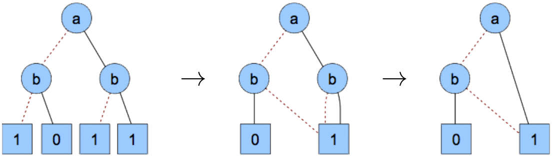
Konstruktion v. Wahrheitstabelle

1. *Boolsche Funktion umwandeln*

Boolsche Funktion	$\rightarrow$	Wahrheitstabelle				
$a \vee (\overline{a} \wedge \overline{b})$	$\rightarrow$	a	$\vee$	$(\overline{a}$	$\wedge$	$\overline{b})$
		0		1		1
		0		1		0
		1		0		1
		1		0		0

2. *Isomorphe Teilbäume umformen u. redundante Knoten entfernen*

decision tree → isomorph → BDD



Vorteile von BDDs

- einfache Evaluierung
- lexikographisch kleinsten/größten Wert der die Gleichung erfüllt
- Lösungen zählen
- Liste aller Lösungen in $O(nN)$
- Wahrscheinlichkeit für Lösung

Komplexität

Implementierungen	*Zeitkomplexität*	
Reduce	$O(\|G\| \cdot log\|G\|)$	G, G_1, G_2...Graphen
Apply	$O(\|G_1\| \cdot \|G_2\|)$	$\|G\|$...Anzahl d. Knoten
Restrict	$O(\|G\| \cdot log\|G\|)$	$\|S_f\|$...Anzahl der
Compose	$O(\|G_1\|^2 \cdot \|G_2\|)$	Erfüllbaren Fkt.
Satisfy-one	$O(n)$	
Satisfy-all	$O(n \cdot \|S_f\|)$	
Satisfy-count	$O(\|G\|)$	

Komplexität

Implementierungen	*Zeitkomplexität*	
Reduce	$O(\lvert G\rvert \cdot log\lvert G\rvert)$	G, G_1, G_2...Graphen
Apply	$O(\lvert G_1\rvert \cdot \lvert G_2\rvert)$	$\lvert G\rvert$...Anzahl d. Knoten
Restrict	$O(\lvert G\rvert \cdot log\lvert G\rvert)$	$\lvert S_f\rvert$...Anzahl der
Compose	$O(\lvert G_1\rvert^2 \cdot \lvert G_2\rvert)$	Erfüllbaren Fkt.
Satisfy-one	$O(n)$	
Satisfy-all	$O(n \cdot \lvert S_f\rvert)$	
Satisfy-count	$O(\lvert G\rvert)$	

Worst Case

- Maximal Komplexität (Wahrheitstabelle) $O(2^{2^n})$
  z.B. integer Multiplizierer

Im Schnitt

- lineare Komplexität

Implementierung

Grundlegendes Design

- Node Pointers
- Node Indices

Implementierung

Grundlegendes Design

- Node Pointers
- Node Indices

BF vs. DF

Implementierung

Grundlegendes Design

- Node Pointers
- Node Indices

BF vs. DF

unique Tables

- Abbildung von Node auf Tripel $F = (v, G, H)$
- *hash* Tabelle
- garantiert Eindeutigkeit
- exakte Performance Benchmarks

Anwendung

Anwendungsgebiete

- Formelüberprüfung auf Äquivalenz
- Modellüberprüfung
- pipelined microprocessors werden geprüft
- automatische Testfallgeneration
- routen/optimieren von FPGAs

konkrete Anwendung

- MS SLAM Project. entwickelt ca. 2000

Zusammenfassung

- DNF, KNF und INF/ITE
  if-then-else Normal Form
- BDDs
  effiziente Datenstruktur für Boolsche Funktionen
- Implementierung
  z.B. unique Table
- Anwendung
  z.B. Formelüberprüfung auf Äquivalenz

Quellenangaben

- Rudolf Mühlbauer
  Binary Decision Diagrams Implementation Details
  Institute for Formal Models and Verification, Linz, Mai 2009.
- Sheldon B. Akers
  Binary Decision Diagrams
  IEEE Transactions on Computers, 27(6):509-516, 1978.
- Randal E. Bryant
  Graph-Based Algorithms for Boolean Function Manipulation
  IEEE Transactions on Computers, 35:677-691, 1986.
- Henrik Reif Andersen
  An Introduction to Binary Decision Diagrams
  Course Notes on the WWW, 1997.

Diese Präsentation ist auf http://www.gu3.at/uni zu finden.

Fragen?

Fragen zum Thema
können sie jetzt stellen!

Ende

Vielen Dank für Ihre Aufmerksamkeit!