Kapitel 7

Formale Spezifikation von Hardware:

1. **Boolesche Ausdrücke**
2. Binäre Entscheidungsdiagramme (BDDs)
3. Anwendung: Formale Verifikation

binary decision diagram

Albert-Ludwigs-Universität Freiburg

Dr. Tobias Schubert, Dr. Ralf Wimmer

Professur für Rechnerarchitektur

WS 2016/17

UNI FREIBURG

Motivation

- Der Entwurf von ReTI hat (hoffentlich) gezeigt, dass Hardware-Synthese komplex und fehleranfällig ist.
- Es gibt automatische Methoden, um Fehler zu finden oder ihre Abwesenheit nachweisen zu können.
- Für ihre Anwendbarkeit muss ein Schaltkreis formal vollständig spezifiziert werden.
- Wir schauen uns daher boolesche Funktionen nochmals (und genauer) an und lernen effiziente Algorithmen und Datenstrukturen zu ihrer Handhabung.

2

Motivation

- Der Entwurf von ReTI hat (hoffentlich) gezeigt, dass Hardware-Synthese komplex und fehleranfällig ist.
- Es gibt automatische Methoden, um Fehler zu finden oder ihre Abwesenheit nachweisen zu können.
- Für ihre Anwendbarkeit muss ein Schaltkreis formal vollständig spezifiziert werden.
- Wir schauen uns daher boolesche Funktionen nochmals (und genauer) an und lernen effiziente Algorithmen und Datenstrukturen zu ihrer Handhabung.

Boolesche Algebren - allgemein

Operation $\wedge$ $\vee$

- Es sei M eine Menge auf der zwei binäre Operationen $\cdot$ und $+$ und eine unäre ~~Option~~ $\sim$ definiert sind.

- Das Tupel $(M, \cdot, +, \sim)$ heißt boolesche Algebra, falls M eine nichtleere Menge ist und für alle $x, y, z \in M$ die folgenden Axiome gelten:

Kommutativität	$x + y = y + x$	$x \cdot y = y \cdot x$
Assoziativität	$x + (y + z) = (x + y) + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Absorption	$x + (x \cdot y) = x$	$x \cdot (x + y) = x$
Distributivität	$x + (y \cdot z) = (x + y) \cdot (x + z)$	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
Komplement	$x + (y \cdot \sim y) = x$	$x \cdot (y + \sim y) = x$

Beispiele boolescher Algebren

$\{0,1\}$

- $(\mathbb{B}, \wedge, \vee, \neg)$

Potenzmenge

$S = \{a, b, c\}$

$Pot(S) = \{\emptyset,$ $\{a\},$ $\{b\},$ $\dots,$ $\{a,b,c\}\}$

- Boolesche Algebra der Teilmengen einer Menge S: $(Pot(S), \cap, \cup, ^C)$
- Boolesche Algebra der booleschen Funktionen in n Variablen: $(\mathbb{B}_n, \cdot, +, \sim)$

⇒ **Allgemein**: Lässt sich eine Aussage direkt aus den Axiomen herleiten, dann gilt sie in allen booleschen Algebren!

- Man darf beim Beweis der Aussage aber auch wirklich nur die Axiome verwenden und keine Eigenschaften der konkreten booleschen Algebra.

5

Boolesche Algebra der Teilmengen von S $(Pot(S), \cap, \cup, {}^C)$

Komplement/ Inverse/ Negation

- Menge: Potenzmenge von S
- $\cdot$: $Pot(S) \times Pot(S) \to Pot(S)$; $(M_1, M_2) \mapsto M_1 \cap M_2$
- $+$: $Pot(S) \times Pot(S) \to Pot(S)$; $(M_1, M_2) \mapsto M_1 \cup M_2$
- ^{C}: $Pot(S) \to Pot(S)$; $M \mapsto M^C := S \setminus M$

Satz

$(Pot(S), \cap, \cup, {}^C)$ ist eine boolesche Algebra.

- **Beweis**: Nachrechnen, dass alle Axiome gelten.

  Beispiel: Absorption
  - Seien $M_1, M_2 \in Pot(S)$.
  - Dann ist $(M_1 + (M_1 \cdot M_2)) = (M_1 \cup \overbrace{(M_1 \cap M_2)}^{M' \subseteq M_1}) = M_1$
    und $(M_1 \cdot (M_1 + M_2)) = (M_1 \cap (M_1 \cup M_2)) = M_1$.

Boolesche Algebra der Funktionen in n Variablen $(\mathbb{B}_n, \cdot, +, \sim)$

- Menge: $\mathbb{B}_n$ (Menge der booleschen Funktionen in n Variablen)
- $\cdot\colon \mathbb{B}_n \times \mathbb{B}_n \to \mathbb{B}_n$; $(f \cdot g)(\alpha) = f(\alpha) \cdot g(\alpha)$ für alle $\alpha \in \mathbb{B}^n$
- $+\colon \mathbb{B}_n \times \mathbb{B}_n \to \mathbb{B}_n$; $(f + g)(\alpha) = f(\alpha) + g(\alpha)$ für alle $\alpha \in \mathbb{B}^n$
- $\sim\colon \mathbb{B}_n \to \mathbb{B}_n$; $(\sim f)(\alpha) = 1 \Leftrightarrow f(\alpha) = 0$ für alle $\alpha \in \mathbb{B}^n$

Satz

$(\mathbb{B}_n, \cdot, +, \sim)$ ist eine boolesche Algebra.

- **Beweis**: Nachrechnen, dass alle Axiome gelten.

  Beispiel: Kommutativität
  - Seien $f, g \in \mathbb{B}_n$.
  - Für alle $\alpha \in \mathbb{B}^n$ gilt: $(f+g)(\alpha) = \underbrace{f(\alpha)+g(\alpha)}_{+:\ \mathbb{B}\times\mathbb{B}\to\mathbb{B}} = g(\alpha)+f(\alpha) = \underbrace{(g+f)(\alpha)}_{+:\ \mathbb{B}_n\times\mathbb{B}_n\to\mathbb{B}_n}$.
  - Also $f+g = g+f$.

Weitere, aus den Axiomen ableitbare Regeln:

- Existenz neutraler Elemente:
  $\exists \mathbf{0} : x + \mathbf{0} = x,\ x \cdot \mathbf{0} = \mathbf{0} \quad \exists \mathbf{1} : x \cdot \mathbf{1} = x,\ x + \mathbf{1} = \mathbf{1}$
- Doppeltes Komplement:
  $(\sim(\sim x)) = x$
- Eindeutigkeit des Komplements:
  $(x \cdot y = \mathbf{0}$ und $x + y = \mathbf{1}) \Rightarrow y = (\sim x)$
- Idempotenz:
  $x + x = x \quad x \cdot x = x$
- de Morgan-Regel:
  $\sim(x + y) = (\sim x) \cdot (\sim y) \quad \sim(x \cdot y) = (\sim x) + (\sim y)$
- Consensus-Regel: / Resolution
  $(x \cdot y) + ((\sim x) \cdot z) = (x \cdot y) + ((\sim x) \cdot z) + (y \cdot z)$
  $(x + y) \cdot ((\sim x) + z) = (x + y) \cdot ((\sim x) + z) \cdot (y + z)$
- Diese Regeln gelten in allen booleschen Algebren!

Dualitätsprinzip bei booleschen Algebren

Prinzip der Dualität

Gilt eine aus den Gesetzen der booleschen Algebra abgeleitete Gleichung p, so gilt auch die zu p duale Gleichung, die aus p hervorgeht durch gleichzeitiges Vertauschen von $+$ und $\cdot$, sowie **0** und **1**.

Consensusregel

- **Beispiel**:
  - $(x \cdot y) + ((\sim x) \cdot z) + (y \cdot z) = (x \cdot y) + ((\sim x) \cdot z)$
  - $(x + y) \cdot ((\sim x) + z) \cdot (y + z) = (x + y) \cdot ((\sim x) + z)$

Boolesche Ausdrücke - allgemein

- Formal vollständige Definition boolescher Ausdrücke
  - Syntax (korrekte Schreibweise) $\rightarrow$ Def. boolescher Ausdrücke $BE(X_n)$
  - Semantik (Bedeutung) $\rightarrow$ Interpretationsfunktion Ψ von $BE(X_n)$

$\rightarrow$ Zweck: Einem Rechner unzweifelhaft „beibringen”, was und was nicht ein boolescher Ausdruck ist und was seine Funktion bezüglich einer booleschen Algebra ist.

$\rightarrow$ Zum Beispiel: Unterschied zwischen dem Ausdruck „$\underbrace{(x_1 \cdot (\sim x_2))}_{BE}$” und der Funktion $\underbrace{f = x_1 \wedge \neg x_2}_{\psi}$.

Syntax boolescher Ausdrücke

- Sei $X_n = \{x_1, \ldots, x_n\}$ eine endliche Menge von Symbolen/Variablen.
- Sei $A = \underline{X_n \cup \{0, 1, +, \cdot, \sim, (,)\}}$ ein Alphabet.

Definition

Die Menge $\underline{BE(X_n)}$ der vollständig geklammerten booleschen Ausdrücke über X_n ist eine Teilmenge von $\underline{A^*}$, die folgendermaßen induktiv definiert ist:

- $\underline{0, 1}$ und $\underline{x_i \in X_n\ i = 1, \ldots, n}$ sind boolesche Ausdrücke
- Sind g und h boolesche Ausdrücke, so auch
  - die Disjunktion $(\underline{g + h})$,
  - die Konjunktion $\underline{(g \cdot h)}$,
  - die Negation $\underline{(\sim g)}$.

11

UNI FREIBURG

Schreibweise von $BE(X_n)$

- **Konvention**: Negation $\sim$ bindet stärker als Konjunktion $\cdot$, Konjunktion $\cdot$ bindet stärker als Disjunktion $+$.
  - → Klammern können weggelassen werden, ohne dass Mehrdeutigkeiten entstehen.
- Je nach Kontext (betrachtete boolesche Algebra) schreibt man auch
  - statt $0, 1$: Die entsprechenden neutralen Elemente,
  - statt $\cdot$: $\wedge, \cap$,
  - statt $+$: $\vee, \cup$,
  - statt $\sim x$: $\neg x, x^C, x', \overline{x}$.

→ So „vereinfachte" Ausdrücke entsprechen zwar nicht genau der obigen Definition, es gibt aber für jeden solchen Ausdruck einen äquivalenten vollständig geklammerten Ausdruck im Sinne der Definition.

- **Beispiel**: Der äquivalente vollständige geklammerte Ausdruck für „$x_1 \wedge \neg x_2$" wäre „$(x_1 \cdot (\sim x_2))$".

12

Semantik boolescher Ausdrücke

typischerweise $\mathbb{B}_n$

- Sei $\widetilde{M} = (M, \cdot, +, \sim)$ eine beliebige boolesche Algebra.
- Seien $0, 1 \in M$ die neutralen Elemente von B.

Definition

Jedem booleschen Ausdruck $BE(X_n)$ kann durch eine Interpretationsfunktion $\Psi : BE(X_n) \to M_n$ eine boolesche Funktion $M_n : M^n \to M$ zugeordnet werden.

$\mathbb{B}^n \to \mathbb{B}$

Ψ wird folgendermaßen induktiv definiert:

- $\Psi(0) = \mathbf{0}$; $\Psi(1) = \mathbf{1}$;
  $\Psi(x_i)(\alpha_1, \ldots, \alpha_n) = \alpha_i$ für alle $\alpha \in M^n$ (Projektion)
- $\Psi((g + h)) = \Psi(g) + \Psi(h)$ (Disjunktion)
- $\Psi((g \cdot h)) = \Psi(g) \cdot \Psi(h)$ (Konjunktion)
- $\Psi((\sim g)) = \sim (\Psi(g))$ (Negation)

13

Interpretation boolescher Ausdrücke

- Sei e ein boolescher Ausdruck.
  - $\Psi(e)(\alpha)$ für ein $\alpha \in M^n$ ergibt sich durch Ersetzen von x_i durch α_i in e, für alle i und Rechnen in der booleschen Algebra $\widetilde{M}$.

    n-elem. Tupel aus M

  - Gilt $\Psi(e) = f$ für eine boolesche Funktion $f \in M_n$, so sagen wir, dass e ein boolescher Ausdruck für f ist, bzw. dass e die boolesche Funktion f beschreibt.
  - Zwei boolesche Ausdrücke e_1 und e_2 heißen äquivalent ($e_1 \equiv e_2$) genau dann, wenn $\Psi(e_1) = \Psi(e_2)$.
    Sie sind gleich, wenn $e_1 = e_2$.
- Wir betrachten folgend nur noch die Interpretation in $B = (\mathbb{B}, \wedge, \vee, \neg)$.

14

Boolesche Ausdrücke ↔ boolesche Funktionen

Lemma 1

Zu jedem booleschen Ausdruck $e \in BE(X_n)$ existiert eine boolesche Funktion f, die durch e beschrieben wird.

- **Beweis**: $f := \Psi(e)$

Lemma 2

Zu jeder booleschen Funktion f existiert ein boolescher Ausdruck, der f beschreibt.

Minterme

BE

- **Beweis**: Es gilt: $f = \Psi(\overbrace{\sum_{\alpha \in ON(f)} m(\alpha)}^{\text{Minterme}})$.

  m.a.W. Die DNF ist ein Boolescher Ausdruck.

  mit anderen Worten

Zusammenhang mit Schaltkreisen

Lemma 3

Zu jedem booleschen Ausdruck $e \in BE(X_n)$ gibt es einen kombinatorischen Schaltkreis, der e implementiert.

$e \in BE(x_n) \rightsquigarrow f = \psi(e)$

- **Beweis**: Übung

SK_f

- Zu jeder booleschen Funktion gibt es einen kombinatorischen Schaltkreis, der sie implementiert (zum Beispiel zweistufige Umsetzung der DNF/KDNF).
- Zu jedem kombinatorischen Schaltkreis gibt es sowohl eine boolesche Funktion, als auch einen boolescher Ausdruck.