Boolean Logic

Specifications section, **Logic** topic, **Lecture 2**

Pavel Panchekha

CS 6110, U of Utah

9 January 2020

Software Verification

No Bugs

Bugs are bad

Challenges

Writing a specification for `quicksort`

Reasoning about predicates like sorted

Combining facts about lists and predicates

Propagating facts through the program

Expanding the specification to make it provable

Verification is hard!

Recent Successes

Quark

COMPCERT

Memory Model

IronClad

ROSETTE

Isabelle

SLAM

Z3

TLA+

KODKOD

Dafny

seL4

frenetic

alloy

Class Progress

Logical reasoning | Program logics | Static analysis

First-order Logic | Decision Procedures

Boolean logic | Syntax | Proof | Algorithm

Boolean logic

Language for stating **facts about booleans**

And/or/not, conjunctive form, universality

Proofs of boolean logic facts **by refutation**

Compact evidence of truth/falsity

Algorithm to **automatically find proofs**

Efficient in simple cases, core of modern solvers

Boolean Specifications

Syntax, Semantics, and Transformations

Specifications

Generally describe **facts** about the program ???

Which may or may not, but **should**, be true

Phrased in terms of **values** that flow through program

Arguments, returns, variables, loop indices, system calls, …

Pre condition | Post condition | Types | Invariants | Effects | …

Today, simplest kind of value: booleans

Facts about Booleans

True	False	Both	Neither	One+	One
x	$\neg x$	$x \land y$	$\neg(x \lor y)$	$x \lor y$	$x \veebar y$

If	Iff	Two of 3	Always	Never	…
$x \Rightarrow y$	$x \Leftrightarrow y$	${}_2\{x, y, z\}_2$	$\top$	$\bot$	

Facts about Booleans

$$x, y := v \mid \neg x \mid x \wedge y \mid x \vee y$$

Variable

Syntax

That's what you can **say**; what does it **mean?**

Depends on the **values of the variables**

$[\![v]\!] =$ "v **is True**"

$[\![x \wedge y]\!] =$ **both** $[\![x]\!]$ **and** $[\![y]\!]$

$[\![\neg x]\!] =$ **not** $[\![x]\!]$

$[\![x \vee y]\!] =$ **either** $[\![x]\!]$ **or** $[\![y]\!]$ **or both**

Semantics

Conjunctive form

$$x \wedge \neg x = \bot \qquad \neg\neg x = x \qquad x \vee y = \neg(\neg x \wedge \neg y)$$

Conjunctive form: and, then or, then not, then variables

$$\begin{aligned}(x \wedge y) \vee (a \wedge b) &= (x \vee (a \wedge b)) \wedge (y \vee (a \wedge b)) \\ &= (x \vee a) \wedge (x \vee b) \wedge (y \vee a) \wedge (y \vee b)\end{aligned}$$

Any boolean logic fact can be **put into conjunctive form**

But can it be **done quickly**?

Conjunctive form

$$(x \land y) \lor (a \land b)$$

Conjunctive form

$$(\quad A \quad) \vee (\quad B \quad)$$

$$A = x \wedge y$$

$$B = a \wedge b$$

Conjunctive form

$$(\quad A \quad) \vee (\quad B \quad)$$

One per term

$$(A \wedge x \wedge y) \vee (\neg A \wedge \neg(x \wedge y))$$

$$(B \wedge a \wedge b) \vee (\neg B \wedge \neg(a \wedge b))$$

Each is small

"Tseytin transformation" to conjunctive form

Constant factor increase in expression size

What use is a Spec?

That's what you can **say**; what does it **mean?**

Depends on the **values of the variables**

We want to use the spec **before** running a program

Variables **don't have values** yet!

"**Could it** be true?" "**Must it** be true?" "**If** this, **then** that?"

Satisfiability

Validity

Implication

Deduction and Proof

Satisfiability, Validity, and Resolution

What use is a Spec?

"**Could it** be true?" "**Must it** be true?" "**If** this, **then** that?"

Satisfiability | **Validity** | **Implication**

What evidence could one provide?

"**Could** $(x \wedge y) \vee (a \wedge b)$ be true?" **Yes.**

"Try True for x and y, and False for a and b."

"**Must** $(x \wedge y) \vee (a \wedge b)$ be true?" **No.**

"Try False for x and y, and False for a and b."

What use is a Spec?

"**Could it** be true?" "**Must it** be true?" "**If** this, **then** that?"

Satisfiability | Validity | Implication

What evidence could one provide?

"**If** $(x \wedge y) \vee (a \wedge b)$ is true, **then must** $x \vee \neg a$ be true?"

"**If** $(x \wedge y) \vee (a \wedge b)$ is true, **then could** $x \vee \neg a$ be false?"

"**Could** $\big((x \wedge y) \vee (a \wedge b)\big) \wedge \neg(x \vee \neg a)$ be true?" **Yes.**

"Try True for x and y, and False for a and b."

Evidence

"**Must** $\underbrace{(x \vee y)}_{\textbf{A}} \wedge \underbrace{(\neg x \vee z)}_{\textbf{B}} \wedge \underbrace{(\neg y \vee z)}_{\textbf{C}} \wedge \underbrace{\neg z}_{\textbf{D}}$ be false?" **Yes.**

x	y	z	A	B	C	D
T	T	T	T	T	T	
T	T		T			T
T		T	T	T	T	
T			T		T	T
	T	T	T	T	T	
	T		T	T		T
		T		T	T	
				T	T	T

Evidence

"**Must** $(x \lor y) \land (\neg x \lor z) \land (\neg y \lor z) \land \neg z$ be false?" **Yes.**

$$\begin{tikzcd}[column sep=small]
(x \lor y) \arrow[dr] & & (\neg x \lor z) \arrow[dl] & (\neg y \lor z) \arrow[d] & \neg z \arrow[dd] \\
& y \lor z \arrow[rr] & & z \arrow[dr] & \\
& & & & \bot
\end{tikzcd}$$

If $(x \lor y)$ **and** $(\neg x \lor z)$ **then** $(y \lor z)$

$$\frac{x \lor y \qquad \neg x \lor z}{y \lor z}$$

Logical Resolution

Evidence

If $(x \vee y)$ **and** $(\neg x \vee z)$ **then** $(y \vee z)$

$$\frac{x \vee y \qquad\qquad \neg x \vee z}{y \vee z}$$

Logical Resolution

Logical resolution can **prove anything must be false**.

Or **disprove anything could be true**, naturally…

Compact: resolution proof hard to find, **easy to check**

Proof by Resolution

Logical resolution can **prove falsehood**.

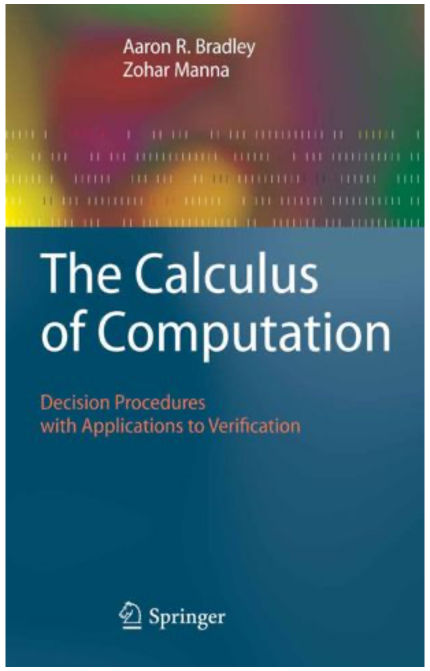
1. Put into conjunctive form
2. Group “or” terms: with x and with $\neg x$
3. Resolve every pair of terms across groups
4. New, equivalent formula without x

We’ll see “variable elimination” again in this class

Course Updates

Recitation, Textbook, Assignment

Textbook



The Calculus of Computation

Aaron R. Bradley
Zohar Manna

Springer, 2007

Topic 1 readings from this book (today: Chapter 1)

Available electronically through SpringerLink

I prefer to **read day after lecture**, before assignment

That way **you can skim** if you already understand it

Assignment 1

Solving n-Queens using the **miniSat** solver

- Given a board size, find **all valid n-Queens solutions**
- **Encode the problem** to boolean logic
- **Transform the encoding** into conjunctive form
- **Invoke the solver** and parse the output
- **Repeat multiple times** to find all solutions

Due 16 January, in a week (start now!)

- Demo, install help **tomorrow** at **13:00** in **MEB 3485**

Proof Search

Davis-Putnam-Logemann-Loveland

Proof by Resolution

Logical resolution can **prove anything must be false**.

1. Put into conjunctive form
2. Group "or" terms: with x and with $\neg x$
3. Resolve every pair of terms across groups
4. New, equivalent formula without x

Rewrites one fact into another form

Proof by Resolution

Algorithm with **one variable**: fact being rewritten

- Like working in **assembly language**…

Let's **re-imagine this algorithm**

- Result is known as **DPLL** "Davis-Putnam-Logemann-Loveland"

Let's **re-imagine this algorithm**

Special Cases

Logical resolution can **prove anything must be false**.

What if one group is empty?

1. Put into conjunctive form

2. Group "or" terms: with x and with $\neg x$

Then there's only x and no $\neg x$

3. Resolve every pair of terms across groups

So x should be True

4. New, equivalent formula without x

So terms with x don't matter

Special Cases

Logical resolution can **prove anything must be false**.

1. Put into conjunctive form
2. Group "or" terms: with x and without x
3. Resolve every pair of terms across groups
4. New equivalent formula without x

What if one term is a singleton?

Then x must be True

Special Cases

Logical resolution can **prove anything must be false**.

1. Put into conjunctive form

2. **Otherwise, both terms are "or"s**

3. Resolve every pair of terms across groups

4. **Then result is also an "or"**

DPLL algorithm

Proof by resolution, **presented as a program**

1. Put into conjunctive form
2. Check for **variables all alone**
3. Check for **variables with one polarity**
4. Pick a variable and **try setting** to True
5. If it doesn’t work, it **must be** False

Proof by Resolution

DPLL algorithm is still **core of modern SAT solvers**

Two **additional improvements** possible

- Non-chronological backtracking
- Conflict-driven clausal learning

One **knob to tune**: which variable to pick

- Most common variable? Least common? Least controversial?

Next class:

First-order Logic

To do:

- ☐ Course feedback
- ☐ Reading in textbook
- ☐ Assignment 1

Boolean logic

Language for stating **facts about booleans**

- And/or/not, conjunctive form, universality

Proofs of boolean logic facts **by refutation**

- Compact evidence of truth/falsity

Algorithm to **automatically find proofs**

- Efficient in simple cases, core of modern solvers

TO INFINITY...

2 BOOLEANS

∞ INTEGERS

NEW SYNTAX

FOR ALL

THERE EXISTS

THEORIES

ONLY THE LIMITS
OF YOUR MIND

MAKE IT
IMPOSSIBLE

Next class:

First-order Logic

To do:

- ☐ Course feedback
- ☐ Reading in textbook
- ☐ Assignment 1