Binary Decision Diagrams

EECS 4315

www.cse.yorku.ca/course/4315/

Model checking

- *Explicit*: states and transitions are represented explicitly.

  Drawback: the state space of interesting systems is usually too large to represent explicitly.

Model checking

- *Explicit*: states and transitions are represented explicitly.

  Drawback: the state space of interesting systems is usually too large to represent explicitly.

- *Symbolic*: (sets of) states and (sets of) transitions are represented symbolically.

  Key idea: exploit the fact that the state space of most systems is not random.

Model checking

- *Explicit*: states and transitions are represented explicitly.

  Drawback: the state space of interesting systems is usually too large to represent explicitly.

- *Symbolic*: (sets of) states and (sets of) transitions are represented symbolically.

  Key idea: exploit the fact that the state space of most systems is not random.

  We focus on one symbolic approach:

  - BDD based

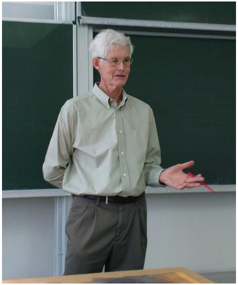
Satisfiability

Cook's theorem

Satisfiability checking of Boolean expressions is NP-complete.

Stephen Cook

- recipient of the ACM Turing award (1982)
- fellow of the Royal Society of London (1998)
- fellow of the Royal Society of Canada (1984)
- member of the National Academy of Sciences (1985)
- member of the American Academy of Arts and Sciences (1986)



Source: Jiri Janicek

Tautology

Theorem

Tautology checking of Boolean expressions is co-NP-complete.

Disjunctive normal form

Definition

A *literal* is a variable or its negation.

Disjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *disjunctive normal form (DNF)* if it is a disjunction of conjunctions of literals.

Disjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *disjunctive normal form (DNF)* if it is a disjunction of conjunctions of literals.

Proposition

Any Boolean expression is equivalent to one in DNF.

Disjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *disjunctive normal form (DNF)* if it is a disjunction of conjunctions of literals.

Proposition

Any Boolean expression is equivalent to one in DNF.

Proposition

Satisfiability checking of Boolean expressions in DNF is in P.

Disjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *disjunctive normal form (DNF)* if it is a disjunction of conjunctions of literals.

Proposition

Any Boolean expression is equivalent to one in DNF.

Proposition

Satisfiability checking of Boolean expressions in DNF is in P.

Proposition

Tautology checking of Boolean expressions in DNF is co-NP-complete.

Conjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *conjunctive normal form (CNF)* if it is a conjunction of disjunctions of literals.

Conjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *conjunctive normal form (CNF)* if it is a conjunction of disjunctions of literals.

Proposition

Any Boolean expression is equivalent to one in CNF.

Conjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *conjunctive normal form (CNF)* if it is a conjunction of disjunctions of literals.

Proposition

Any Boolean expression is equivalent to one in CNF.

Proposition

Satisfiability checking of Boolean expressions in CNF is NP-complete.

Conjunctive normal form

Definition

A *literal* is a variable or its negation.

Definition

A Boolean expression is in *conjunctive normal form (CNF)* if it is a conjunction of disjunctions of literals.

Proposition

Any Boolean expression is equivalent to one in CNF.

Proposition

Satisfiability checking of Boolean expressions in CNF is NP-complete.

Proposition

Tautology checking of Boolean expressions in CNF is in P.

If-then-else normal form

Notation

$$
\begin{array}{rcl}
0 & : & \text{false} \\
1 & : & \text{true} \\
x \to t_1, t_0 & : & (x \wedge t_1) \vee (\neg x \wedge t_0)
\end{array}
$$

Definition

The set of Boolean expressions in *if-then-else normal form (INF)* is defined by

$$t ::= 0 \mid 1 \mid x \to t, t$$

If-then-else normal form

Question

Give a Boolean expression in INF equivalent to $x_1 \wedge (\neg x_2 \vee x_3)$.

If-then-else normal form

Question

Give a Boolean expression in INF equivalent to $x_1 \wedge (\neg x_2 \vee x_3)$.

Answer

$$
\begin{aligned}
t &= x_1 \to t_1, t_0 \\
t_0 &= x_2 \to t_{01}, t_{00} \\
t_1 &= x_2 \to t_{11}, t_{10} \\
t_{00} &= x_3 \to 0, 0 \\
t_{01} &= x_3 \to 0, 0 \\
t_{10} &= x_3 \to 1, 1 \\
t_{11} &= x_3 \to 1, 0
\end{aligned}
$$

If-then-else normal form

Shannon's expansion theorem

For every Boolean expression t and variable x,

$$t = x \to t[1/x], t[0/x].$$

Proposition

Any Boolean expression is equivalent to one in INF.

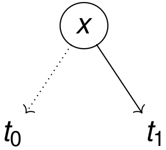
Decision trees

Boolean expressions in INF can be viewed as binary trees known as *decision trees*.

Two types of leaves: 0 and 1

$\boxed{0}$ $\boxed{1}$

One type of internal nodes: $x \rightarrow t_1, t_0$



Decision trees

Question

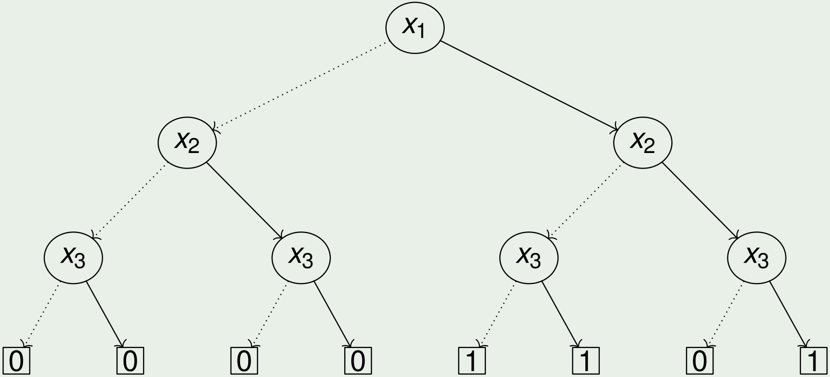
Draw the decision tree for the Boolean expression in INF equivalent to $x_1 \wedge (\neg x_2 \vee x_3)$.

Decision trees

Question

Draw the decision tree for the Boolean expression in INF equivalent to $x_1 \wedge (\neg x_2 \vee x_3)$.

Answer



If-then-else normal form

$$
\begin{aligned}
t &= x_1 \to t_1, t_0 \\
t_0 &= x_2 \to t_{01}, t_{00} \\
t_1 &= x_2 \to t_{11}, t_{10} \\
t_{00} &= x_3 \to 0, 0 \\
t_{01} &= x_3 \to 0, 0 \\
t_{10} &= x_3 \to 1, 1 \\
t_{11} &= x_3 \to 1, 0
\end{aligned}
$$

Question

Identify all equal subexpressions.

If-then-else normal form

$$
\begin{aligned}
t &= x_1 \to t_1, t_0 \\
t_0 &= x_2 \to t_{01}, t_{00} \\
t_1 &= x_2 \to t_{11}, t_{10} \\
t_{00} &= x_3 \to 0, 0 \\
t_{01} &= x_3 \to 0, 0 \\
t_{10} &= x_3 \to 1, 1 \\
t_{11} &= x_3 \to 1, 0
\end{aligned}
$$

Question

Identify all equal subexpressions.

Answer

There are multiple occurrences of 0 and 1. Furthermore, t_{00} and t_{01} are equal.

Binary decision diagram

Question

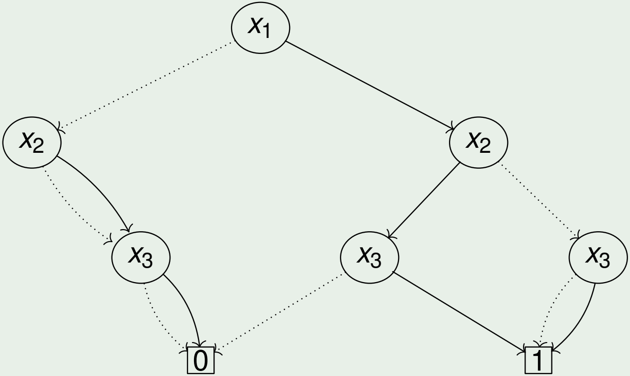
Identify the equal subtrees in the decision tree for the Boolean expression in INF equivalent to $x_1 \wedge (\neg x_2 \vee x_3)$.

Binary decision diagram

Question

Identify the equal subtrees in the decision tree for the Boolean expression in INF equivalent to $x_1 \wedge (\neg x_2 \vee x_3)$.

Answer



Binary decision diagram

Definition

A *binary decision diagram (BDD)* is a rooted directed acyclic graph where

- two (external) nodes where have out-degree zero and are labelled 0 and 1,
- and all other (internal) nodes have out-degree two, with one outgoing edge called the low edge and the other called the high edge, and are labelled with a variable.

Notation

Let u be an internal node.
$var(u)$ denotes the variable with which node u is labelled.
$low(u)$ denotes the successor of node u along its low edge (corresponding to the case that value of $var(u)$ is low, that is, 0).
$high(u)$ denotes the successor of node u along its high edge (corresponding to the case that value of $var(u)$ is high, that is, 1).

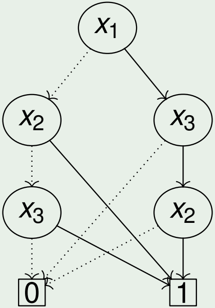
Ordered binary decision diagrams

Definition

A BDD is *ordered* if on all paths through the graph the variables respect a given linear order $x_1 < x_2 < \cdots < x_n$.

Question

Is the BDD



ordered?

Reduced ordered binary decision diagrams

Definition

An ordered BDD is *reduced* if

- *unique*: no two distinct internal nodes u and v have the same variable, low- and high-successor, that is,

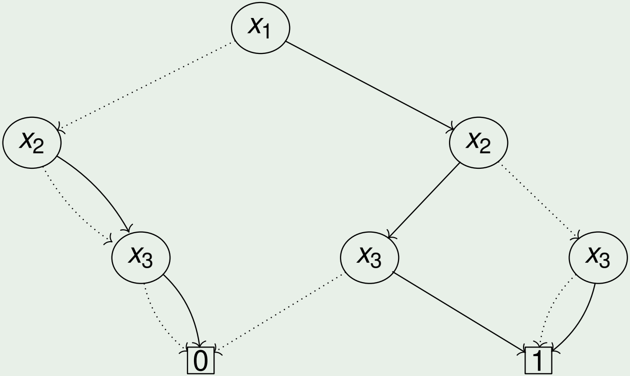
  if $\mathit{var}(v) = \mathit{var}(u), \mathit{low}(v) = \mathit{low}(u),$ and $\mathit{high}(v) = \mathit{high}(u)$ then $u = v$.

- *non-redundant*: no internal node u has identical low- and high-successor, that is,

$$\mathit{low}(u) \neq \mathit{high}(u).$$

Question

Is the ordered BDD



reduced?

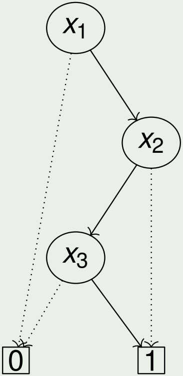
Question

What is the corresponding reduced ordered BDD?

Question

What is the corresponding reduced ordered BDD?

Answer



Canonicity lemma

Lemma

For a Boolean expression t with variables $x_1, x_2, \ldots, x_n$ and a linear order $x_1 < x_2 < \cdots < x_n$, there exists a unique reduced ordered BDD which is equivalent to t.

For the remainder, we restrict our attention to reduced ordered BDDs and simply call them BDDs.

- member of the National Academy of Engineering (2003),
- recipient of the Paris Kanellakis Theory and Practice Award (1997)
- recipient of the IEEE Emanuel R. Piore Award (2007)
- his paper on BDDs is one of the most cited computer science papers (more than 9660 citations)

Source: Randal Bryant

BDDs

Proposition

Satisfiability checking of BDDs is constant time.

Proposition

Tautology checking of BDDs is constant time.

The variable order matters

Question

Draw the BDD corresponding to

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6)$$

for the variable ordering

$$x_1 < x_2 < x_3 < x_4 < x_5 < x_6$$

The variable order matters

Question

Draw the BDD corresponding to

$$(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6)$$

for the variable ordering

$$x_1 < x_4 < x_5 < x_2 < x_3 < x_6$$

The variable order matters

Theorem

Deciding whether a given variable order is optimal is NP-hard.

Heuristics are used to find good variable orderings. For more details, see, for example,
I. Wegener. *Branching Programs and Binary Decision Diagrams: Theory and Applications.* 2000.