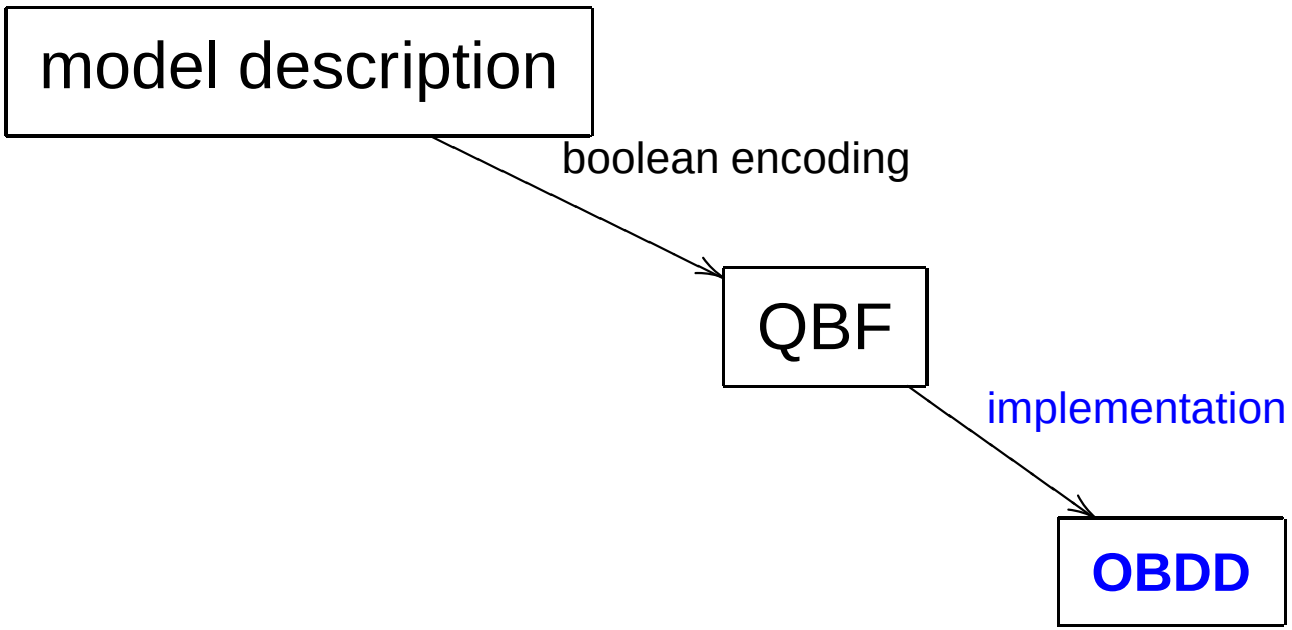
Computer aided verification

Lecture 6:
Symbolic verification I: OBDD

Symbolic model checking

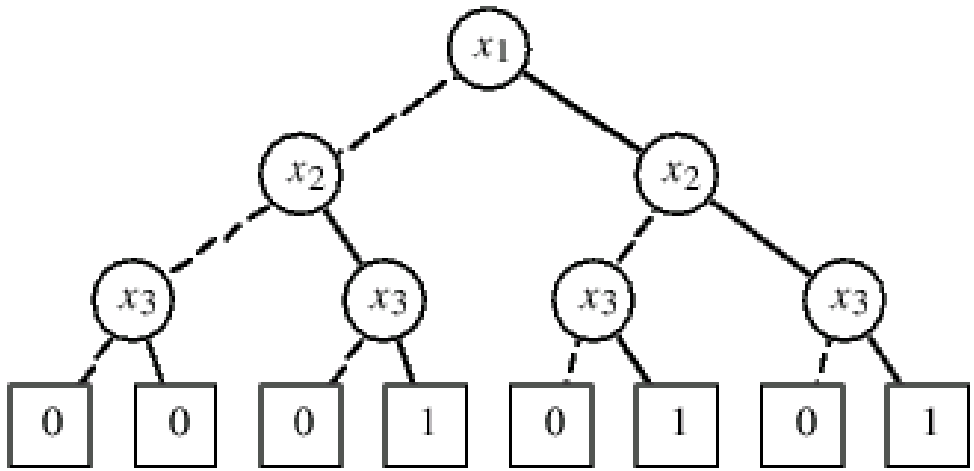


model checking = operations on OBDDs

I. OBDD

Ordered Binary Decision Diagrams

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



[Bryant 1992]

– $f : \{0,1\}^3 \rightarrow \{0,1\}$

– fixed order on variables: $x_1 < x_2 < x_3$

OBDD = rooted acyclic graph

Attributes of a node v:

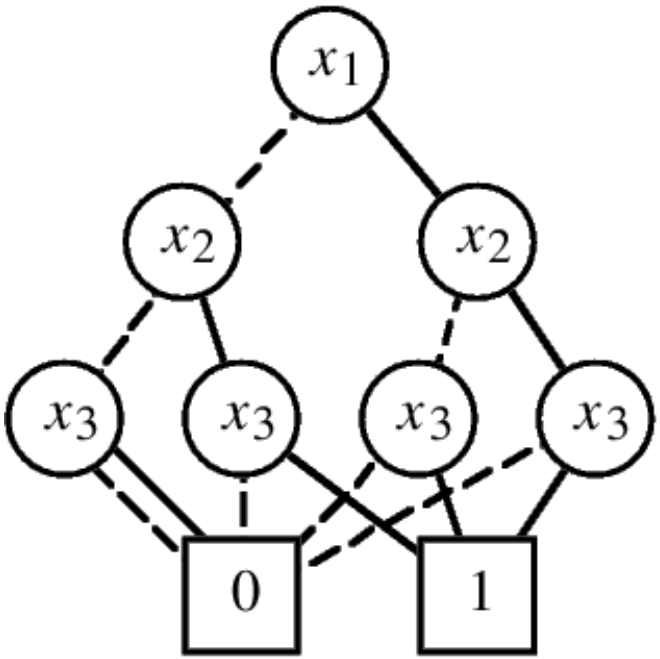
- when v is a leaf
  - $\mathrm{val}(v) \in \{0, 1\}$
- when v is not a leaf
  - $\mathrm{var}(v) \in \{x_1, x_2, \ldots\}$
  - $\mathrm{lo}(v), \mathrm{hi}(v)$ – 2 nodes

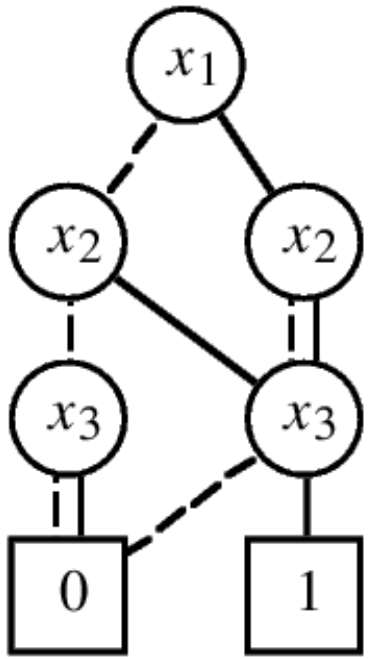
The order of variables must be obeyed on every path.

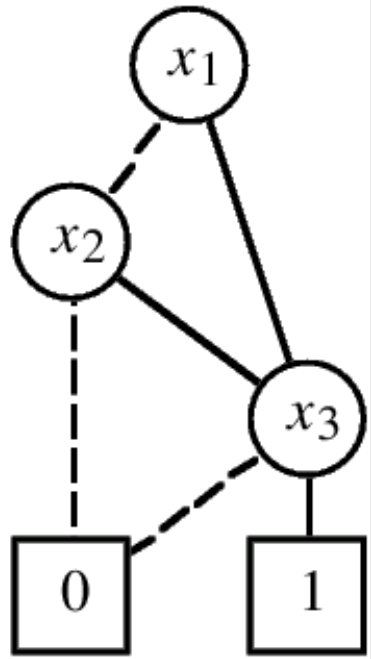
Simplifying OBDD

- remove redundant leaves
- remove redundant non-leaves
- remove redundant tests

OBDD $\longmapsto$ ROBDD







[Bryant 1992]

Canonical form: reduced OBDD

Canonical form for a boolean function:

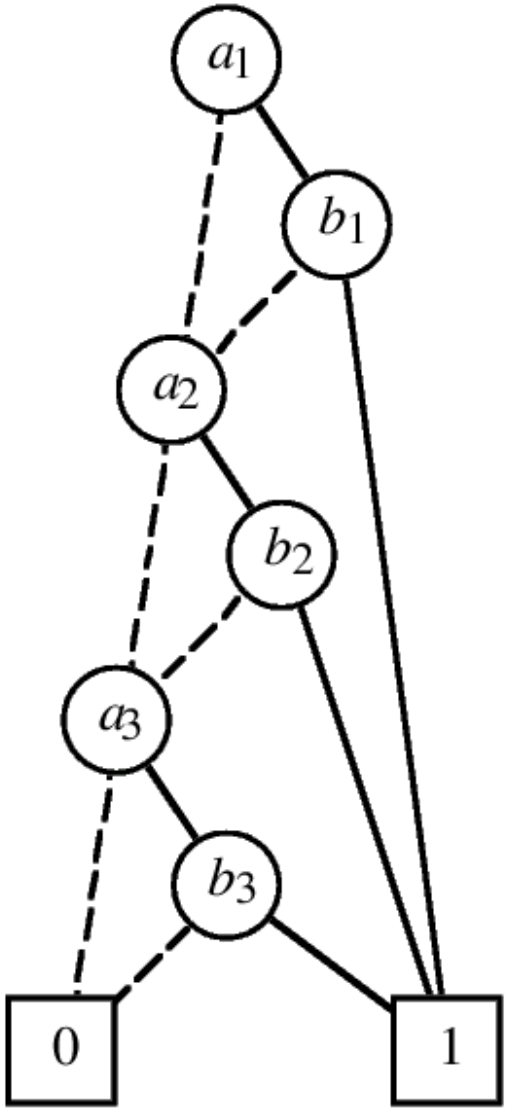
- independent from the initial OBDD
- (strongly) dependent on the order of variables

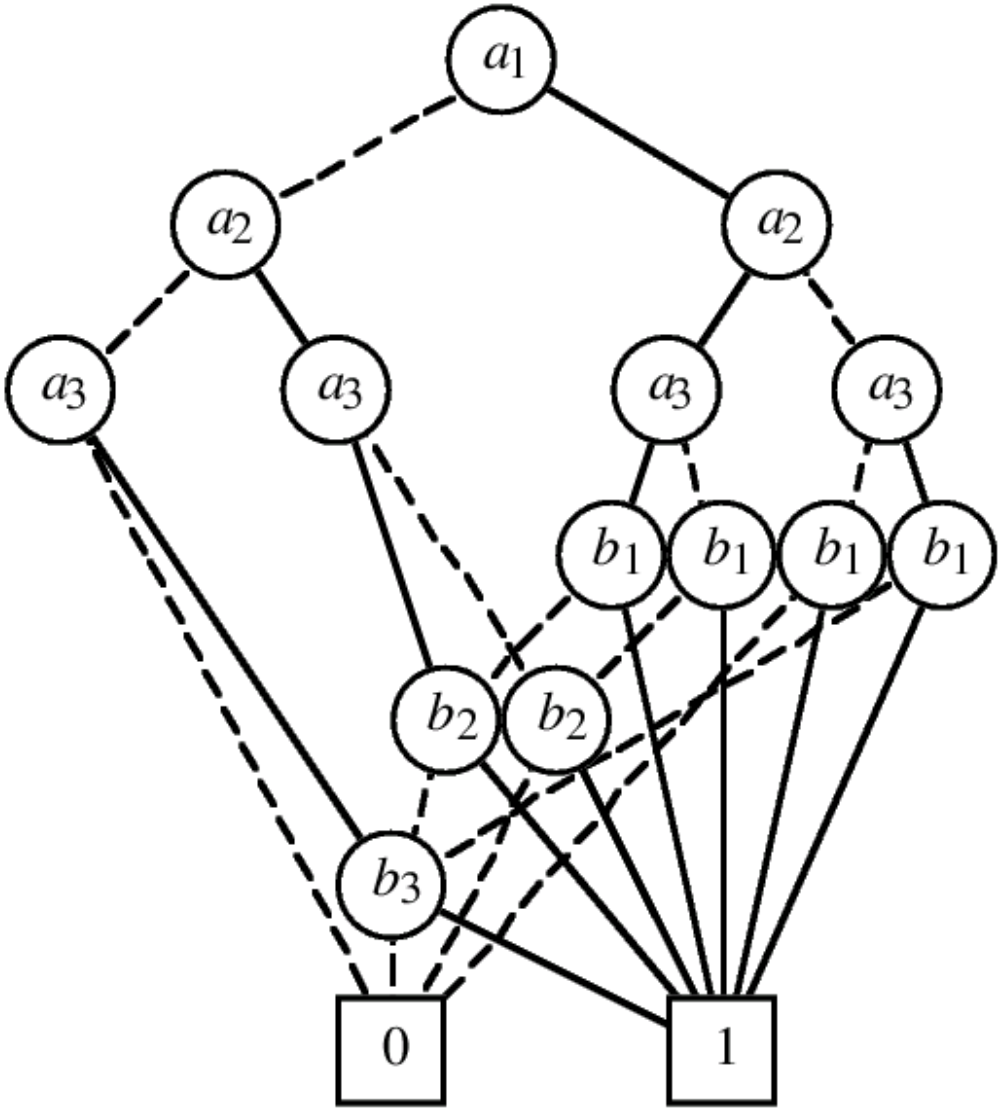
Naive construction of an OBDD for a boolean formula ϕ:

$$\phi \longmapsto \text{decision tree} \longmapsto \text{canonical OBDD}$$

Finding an appropriate order of variables is crucial!

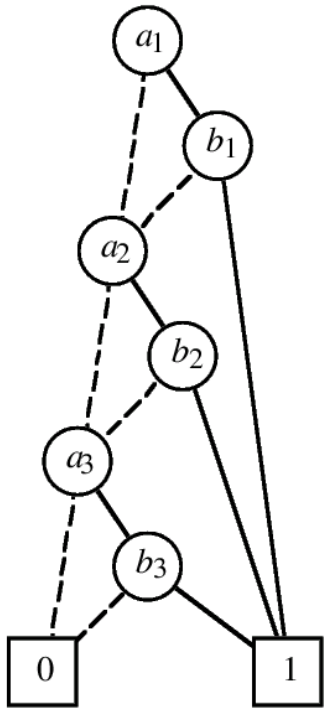
$$(a_1 \wedge b_1) \vee (a_2 \wedge b_2) \vee (a_3 \wedge b_3)$$

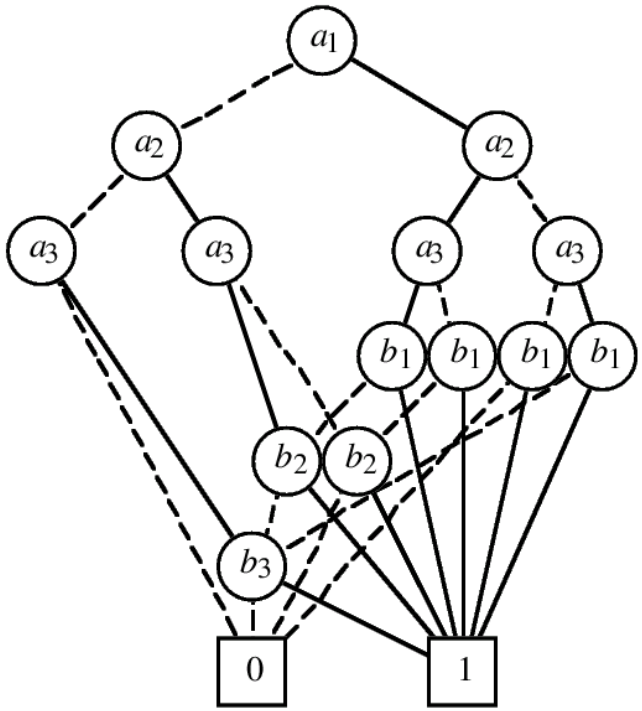




[Bryant 1992]

$$f(a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n) = (a_1 \wedge b_1) \vee (a_2 \wedge b_2) \vee \ldots \vee (a_n \wedge b_n)$$





$2 \cdot n$ $\qquad$ $2 \cdot (2^n - 1)$

OBDD size

Heuristic: closely related variables should be close in the order

example of a boolean function	lower bound	upper bound
symmetric functions	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$
addition (any bit)	$\mathcal{O}(n)$	$\mathcal{O}(2^n)$
multiplication (middle bit)	$\mathcal{O}(2^n)$	$\mathcal{O}(2^n)$

Shannon's expansion

$$f \quad = \quad \neg x \wedge f\big|_{x \leftarrow 0} \quad \vee \quad x \wedge f\big|_{x \leftarrow 1}$$

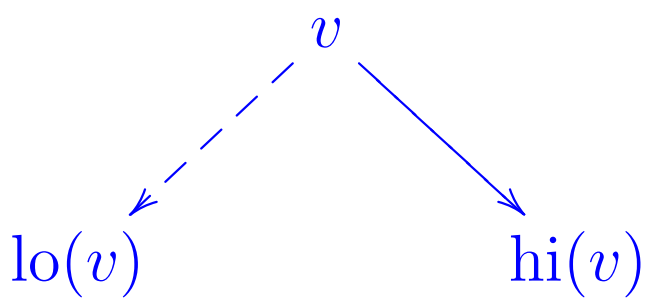
$$f\big|_{x_i \leftarrow b}(x_1, \ldots, x_n) \quad = \quad f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n)$$

Shannon's expansion

$$f \;=\; \neg x \wedge f\big|_{x \leftarrow 0} \;\vee\; x \wedge f\big|_{x \leftarrow 1}$$

$$f\big|_{x_i \leftarrow b}(x_1, \ldots, x_n) \;=\; f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_n)$$

$$v \;=\; \neg x \wedge \mathrm{lo}(v) \;\vee\; x \wedge \mathrm{hi}(v)$$



$x = \mathrm{var}(v)$

OBDDs as an abstract data type

the order of variables the same in all OBDDs

Operations:

$f \vee g, \quad f \wedge g, \quad \neg f, \quad \texttt{false}, \quad \texttt{true} \qquad BF \mapsto OBDD$

$f|_{x \leftarrow 0}, \quad f|_{x \leftarrow 1}$

$\exists x.\ f, \quad \forall x.\ f \qquad QBF \mapsto OBDD$

$f = g$

Note: operations on **functions**, not on values $\{0, 1\}$.

Implementation of unary operations

– $f|_{x \leftarrow b}$

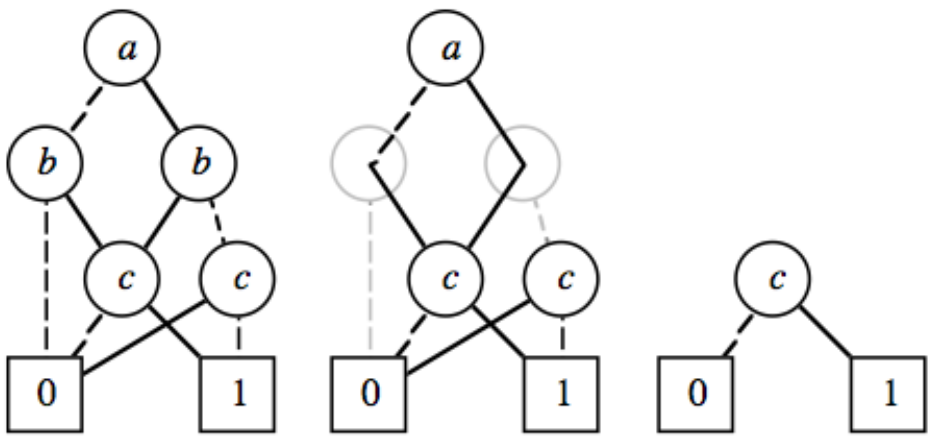
traverse nodes n of OBDD representing f:

– if $x > \mathrm{var}(n)$ then recursively traverse $\mathrm{lo}(n)$ and $\mathrm{hi}(n)$;

– if $x < \mathrm{var}(n)$ then stop;

– otherwise replace n by: $\begin{cases} \mathrm{lo}(n) & \text{if } b = 0 \\ \mathrm{hi}(n) & \text{if } b = 1 \end{cases}$

Example: OBDD for $f|_{b \leftarrow 1}$



[Bryant 1992]

Implementation of unary operations (cont.)

- $\exists x.\ f \quad = \quad f\big|_{x \leftarrow 0} \vee f\big|_{x \leftarrow 1}$

  the order of variables

  remains the same!

- $\neg f$ **?**

Implementation of binary operations

the order of variables the same in all OBDDs

- $\bullet : \{0,1\}^2 \to \{0,1\}$

$$f \bullet g = \neg x \wedge (f \bullet g)|_{x \leftarrow 0} \vee x \wedge (f \bullet g)|_{x \leftarrow 1}$$
$$f \bullet g = \neg x \wedge (f|_{x \leftarrow 0} \bullet g|_{x \leftarrow 0}) \vee x \wedge (f|_{x \leftarrow 1} \bullet g|_{x \leftarrow 1})$$

Apply(f, g, $\bullet$) (think of f, g as roots of OBDDs)

- f, g leaves: $\mathrm{val}(f \bullet g) = \mathrm{val}(f) \bullet \mathrm{val}(g)$
- f leaf, g not: $f \bullet g = op(g)$
- $\mathrm{var}(f) = \mathrm{var}(g) = x$:

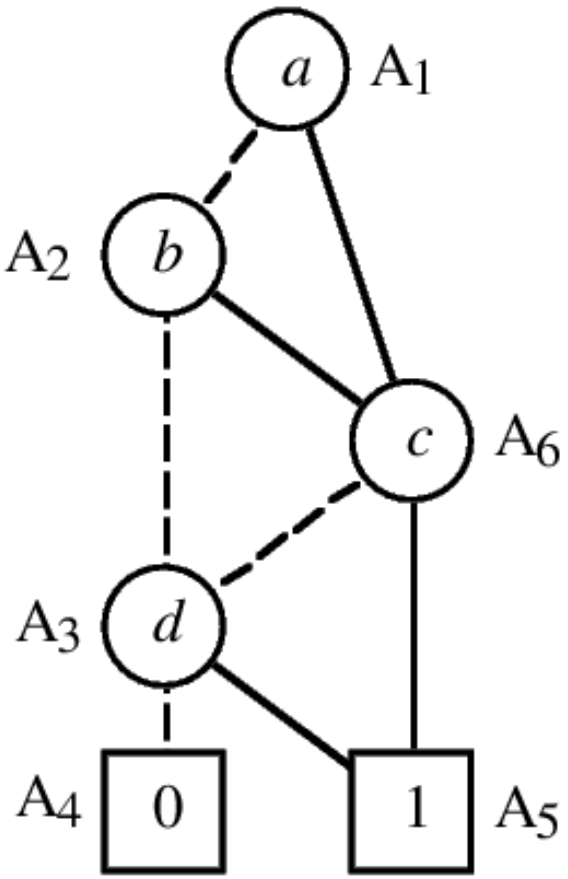
$$\mathrm{lo}(f \bullet g) = \mathrm{lo}(f) \bullet \mathrm{lo}(g) \qquad \mathrm{hi}(f \bullet g) = \mathrm{hi}(f) \bullet \mathrm{hi}(g)$$

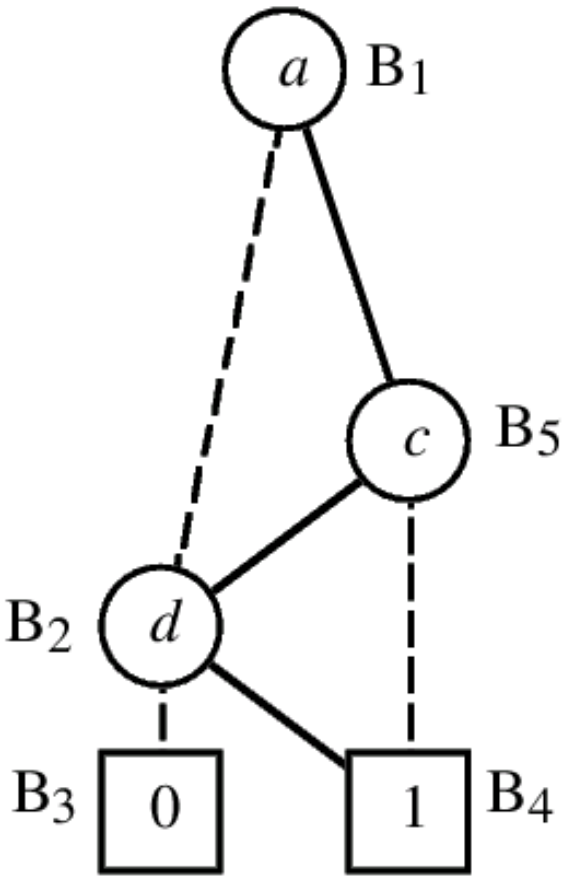
- $\mathrm{var}(f) = x < y = \mathrm{var}(g)$:

$$\mathrm{lo}(f \bullet g) = \mathrm{lo}(f) \bullet g \qquad \mathrm{hi}(f \bullet g) = \mathrm{hi}(f) \bullet g$$

$$f \bullet g = \neg x \wedge (f \bullet g)|_{x \leftarrow 0} \vee x \wedge (f \bullet g)|_{x \leftarrow 1}$$
$$f \bullet g = \neg x \wedge (f|_{x \leftarrow 0} \bullet g|_{x \leftarrow 0}) \vee x \wedge (f|_{x \leftarrow 1} \bullet g|_{x \leftarrow 1})$$

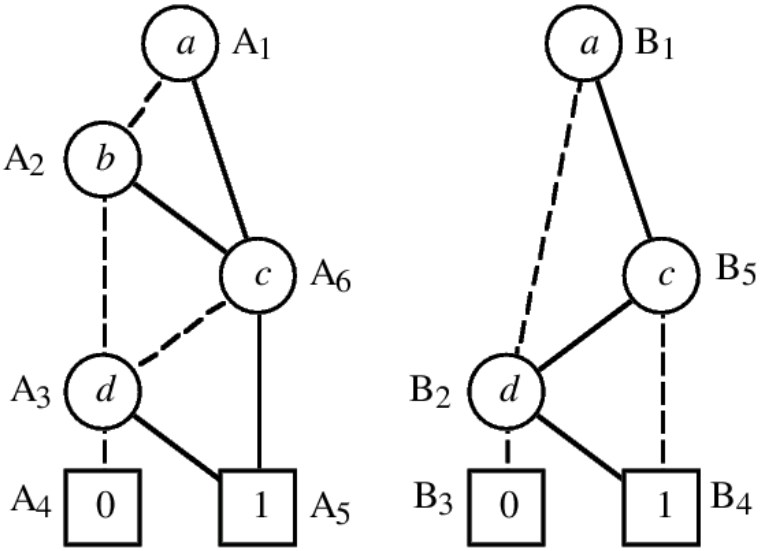
Example: input OBDDs

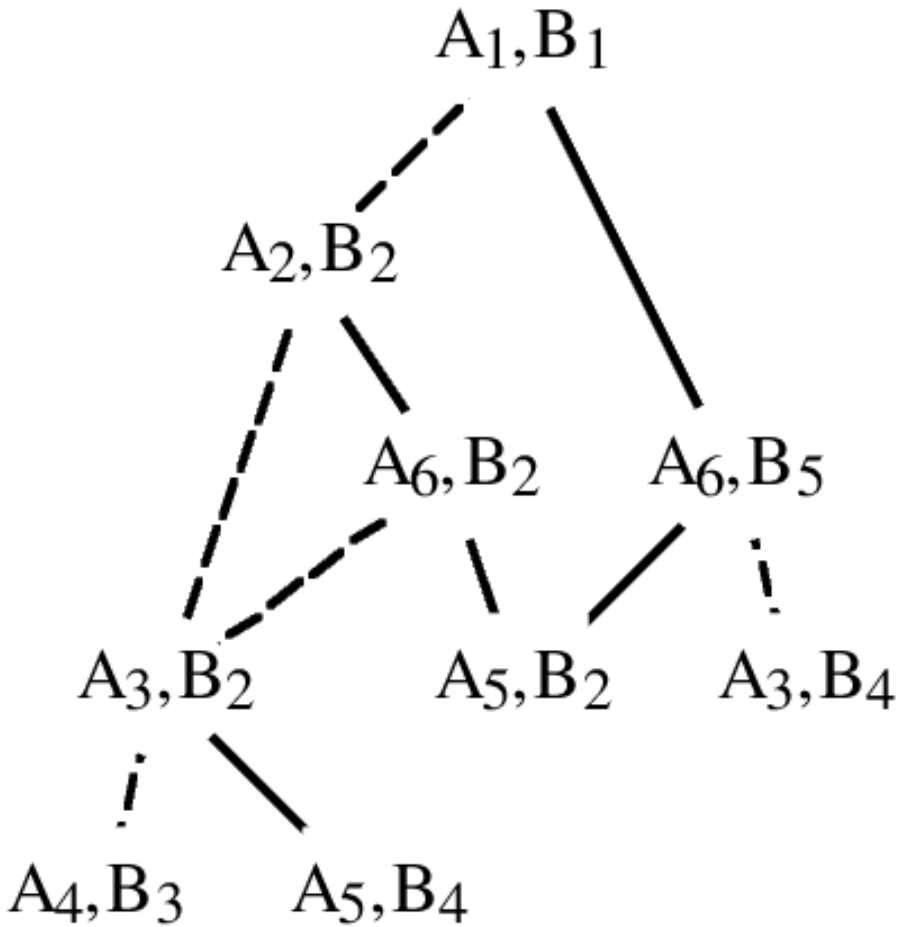




[Bryant 1992]

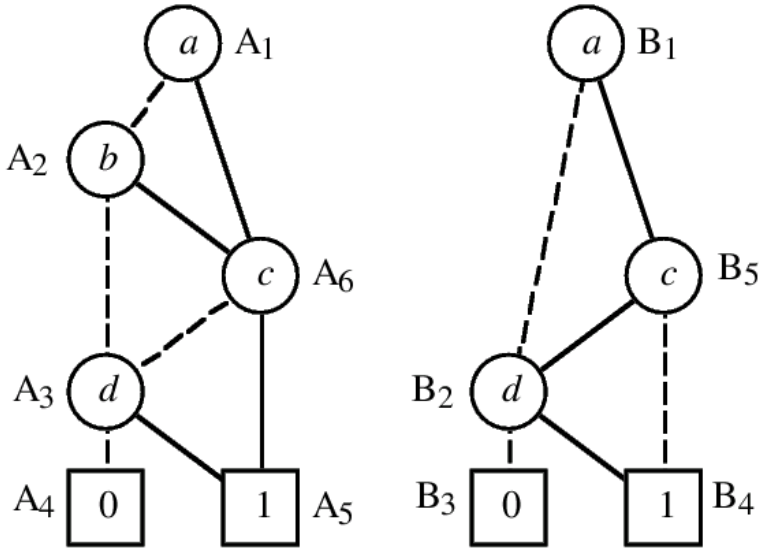
Example: recursive calls





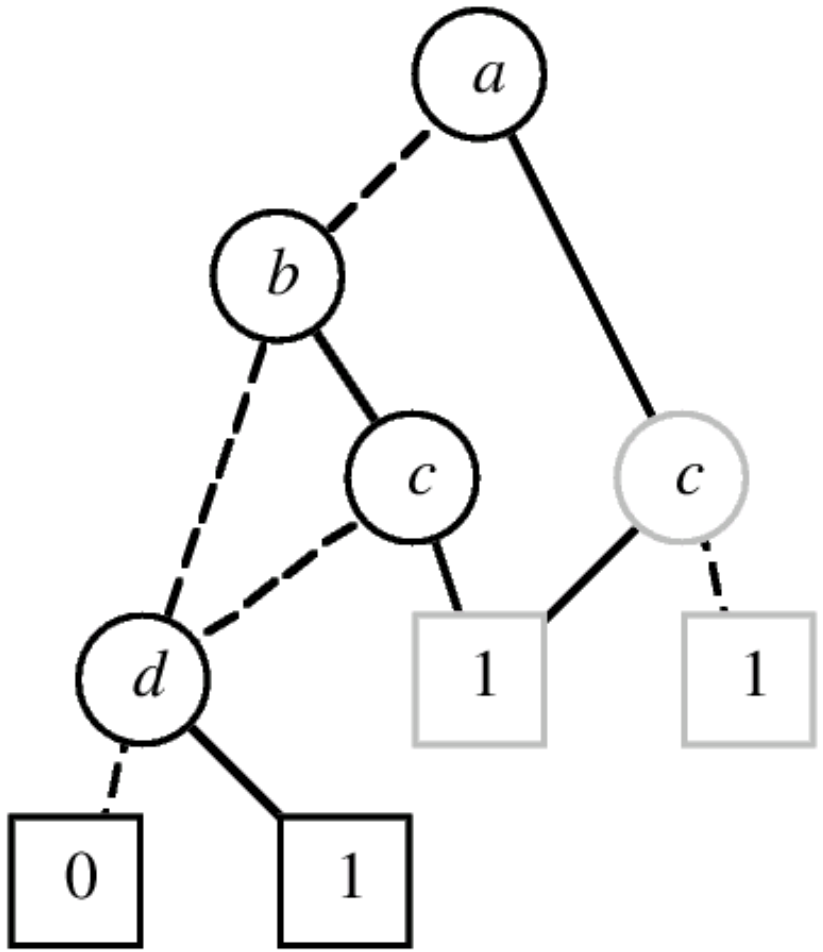
[Bryant 1992]

Example: $f \vee g = a \vee (b \wedge c) \vee d$



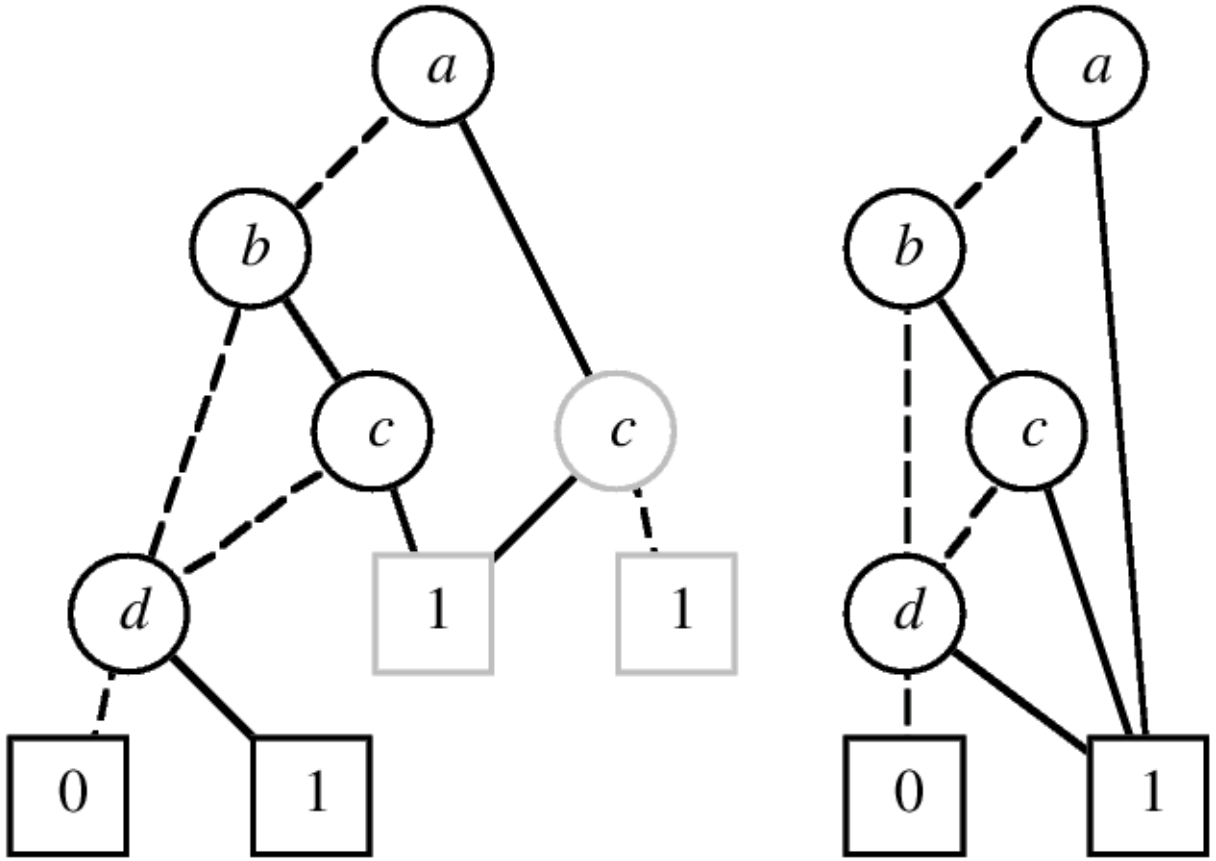
$f = ((a \vee b) \wedge c) \vee d$

$g = (a \wedge \neg c) \vee d$



[Bryant 1992]

Example: reduced $f \vee g$



[Bryant 1992]

Implementation of binary operations

$\mathrm{Apply}(\ f,\ g,\ \bullet\ )$

- running time: $\mathcal{O}(|f| \cdot |g|)$
- result in the canonical form

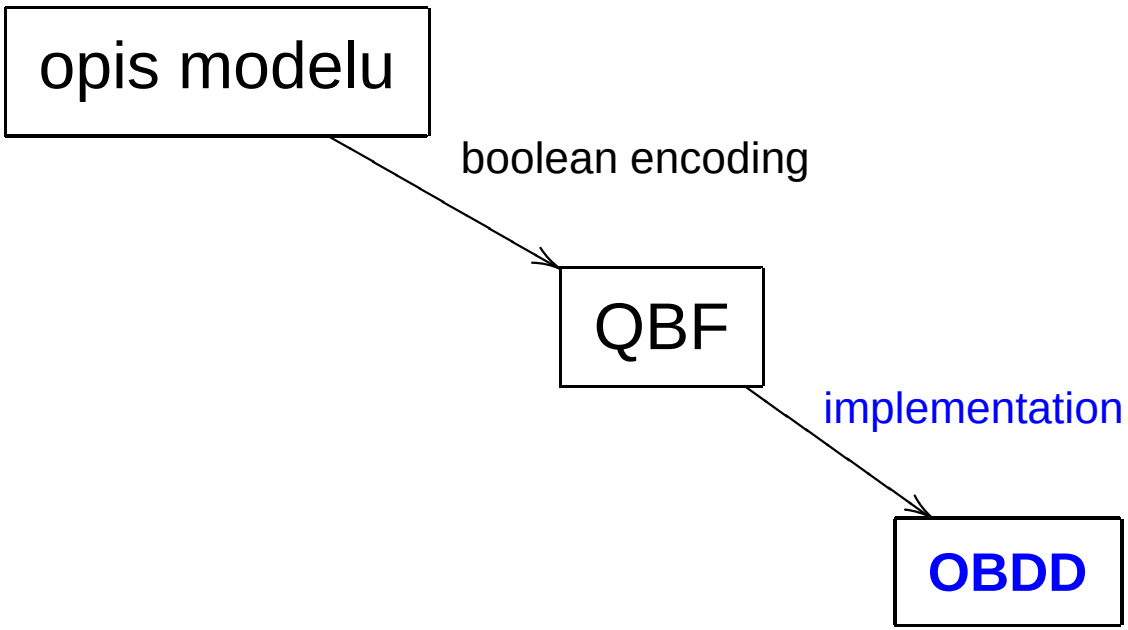
Question: $f \iff g$ **?** $\qquad\qquad$ $f = g$ **?**

Variations

- one OBDD shared by all functions
  - $=$ in constant time
- edges representing $\neg$
- …

II. Boolean encoding

Boolean encoding



model checking = operations on OBDDs

Model description

– S described by m boolean variables: $S \equiv \{0,1\}^m$

– transition relation $R : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$

$$R(x_1, \ldots, x_m, x'_1, \ldots, x'_m) \in \{0,1\}$$

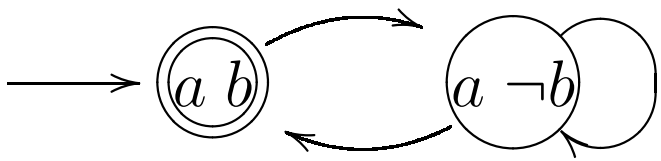
– initial states $S_0 : \{0,1\}^m \to \{0,1\}$

$$S_0(x_1, \ldots, x_m) \in \{0,1\}$$

– atomic properties $L_p = \{s \mid p \in L(s)\} : \{0,1\}^m \to \{0,1\}$

$$L_p(x_1, \ldots, x_m) \in \{0,1\}$$

Example



(reachable states)

$$
\begin{aligned}
R &= (x \wedge \neg x') \vee (\neg x \wedge \neg x') \vee (\neg x \wedge x') \\
S_0 &= x \\
L_b &= x \\
L_a &= 1
\end{aligned}
$$

From a model to OBDD

description of a Kripke structure

Kripke structure ⟼ OBDD

NO!

From a model to OBDD

description of a Kripke structure

NO!

Kripke structure ⟼ OBDD

description of a Kripke structure ⟼ OBDD

YES!

Compositional model desciption

Synchronous processes:

$$R \;=\; R_1 \;\wedge\; R_2 \wedge \;\ldots\; \wedge \;R_n$$

Asynchronous processes (interleaving):

$$R \;=\; R'_1 \;\vee\; R'_2 \vee \;\ldots\; \vee \;R'_n$$

$$R'_i \;=\; R_i \wedge (\textstyle\bigwedge_{j \neq i} \mathrm{Id}_j)$$

Asynchronous processes (simultaneous execution):

$$R \;=\; R'_1 \;\wedge\; R'_2 \wedge \;\ldots\; \wedge \;R'_n$$

$$R'_i \;=\; R_i \vee \mathrm{Id}_i$$

Restriction to reachable states

$$\widehat{R}(x_1, \ldots, x_m, x'_1, \ldots, x'_m) = 1$$

$$\Updownarrow$$

$$R(x_1, \ldots, x_m, x'_1, \ldots, x'_m) \ \wedge$$

$(x_1, \ldots, x_m), (x'_1, \ldots, x'_m)$ reachable

Restriction to reachable states

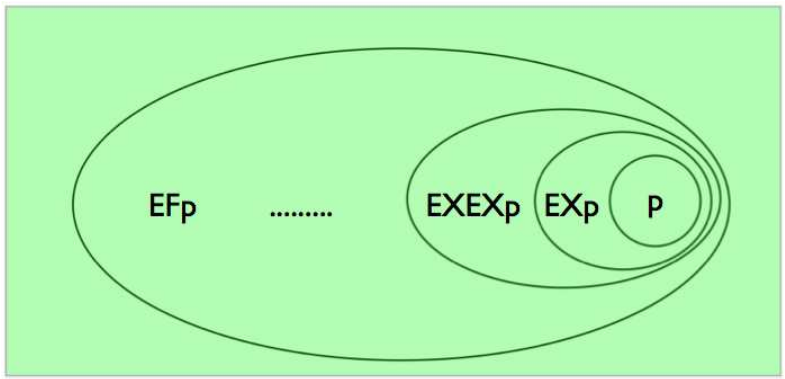
$$\widehat{R}(x_1, \ldots, x_m, x'_1, \ldots, x'_m) = 1$$

$$\Updownarrow$$

$$R(x_1, \ldots, x_m, x'_1, \ldots, x'_m) \ \wedge$$

$(x_1, \ldots, x_m)$ reachable

III. Symbolic verification



Symbolic model checking

CTL ($\neg$, $\wedge$, **EX**, **E_U_**, **EG**) (these connectives are sufficient)

$$\mathrm{Check} : \text{CTL} \mapsto \text{OBDD}$$

$\mathrm{Check}(\phi)$ represents $\{s \mid s \vDash \phi\}$

Example: $\mathrm{Check}(p)$ represents L_p