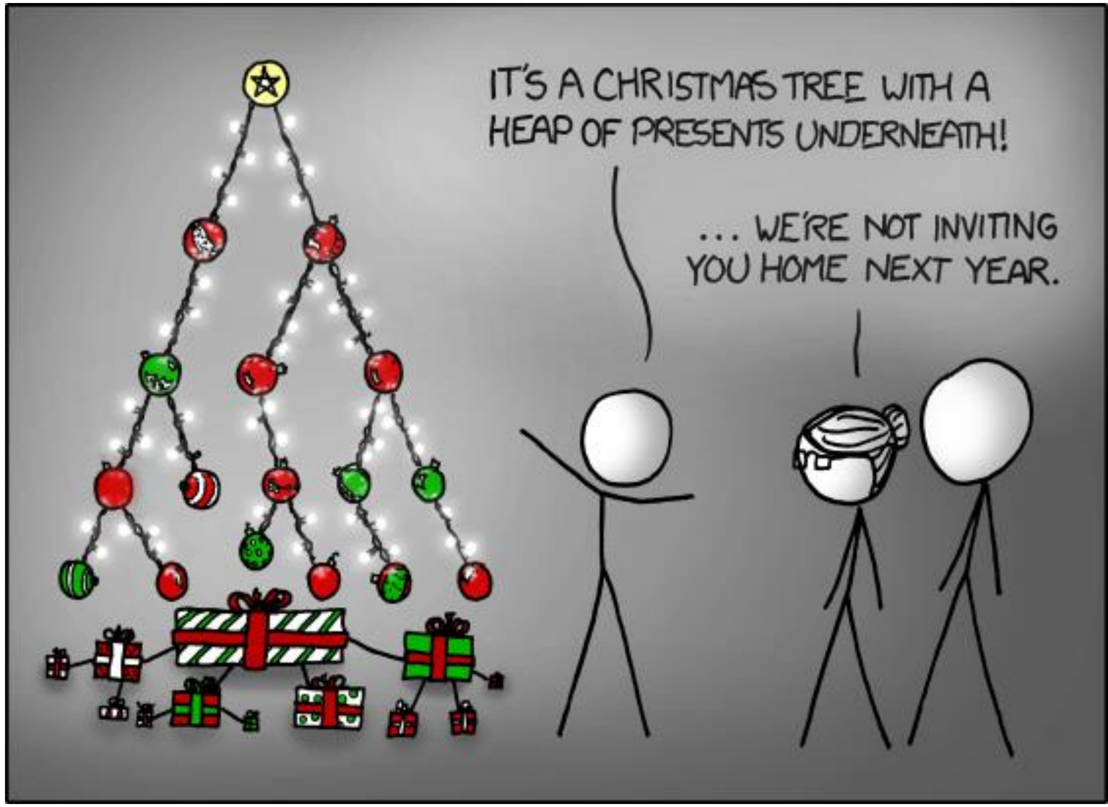
Logic and Computability

Binary Decision Diagrams (BDDs)



Bettina Könighofer
bettina.koenighofer@iaik.tugraz.at

Stefan Pranger
stefan.pranger@iaik.tugraz.at

https://xkcd.com/835/

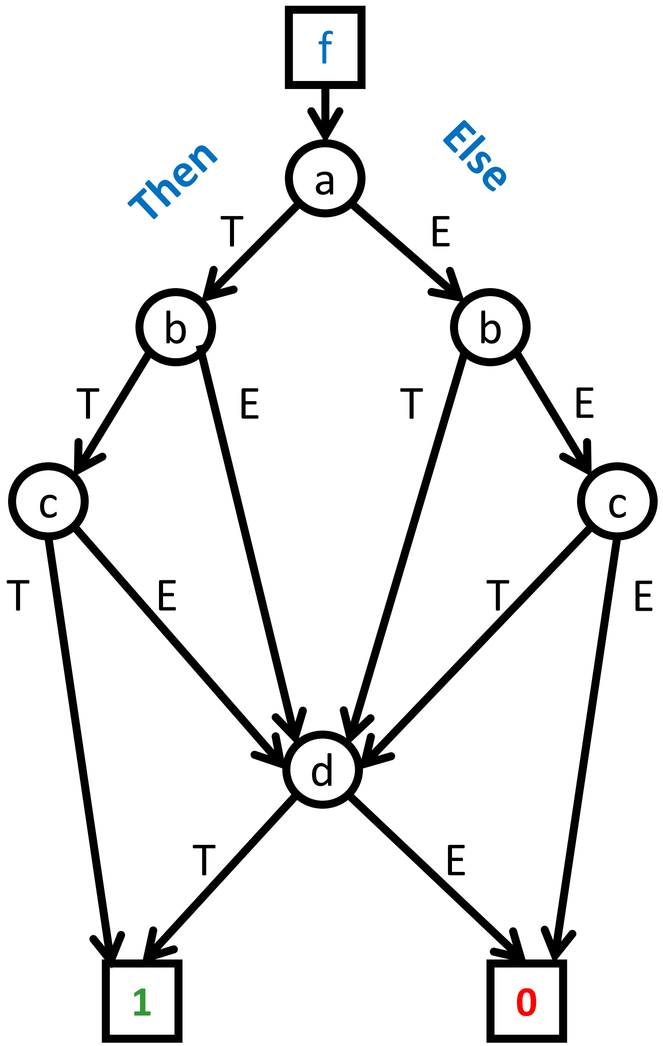
Motivation – BDDs

- Efficient Representation of Boolean Formulas
  - Small for many practical cases
  - Efficient Manipulation
    - Boolean Operations

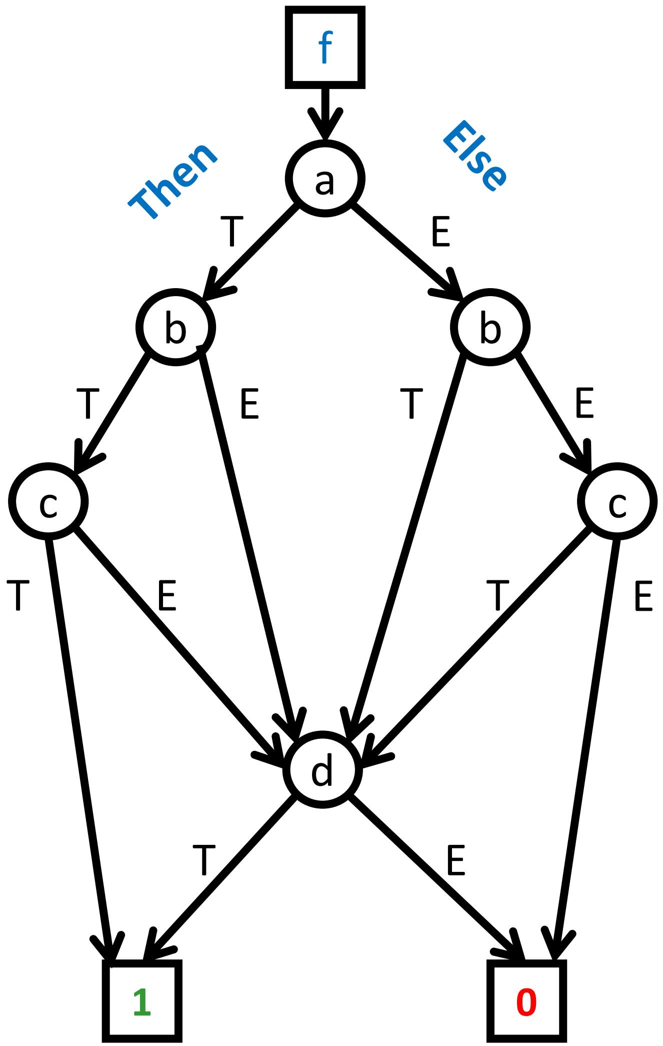
Outline

- What are Binary Decision Diagrams (BDDs)?
  - Intuitive Explanation
  - Formal Definition
- From BDDs to Reduced Ordered BDDs (ROBDDs)
- **Construct Formula from ROBDD**
- **Construct ROBDD from Formula**
- Pros and Cons of BDDs

Binary Decision Diagram (BDD)



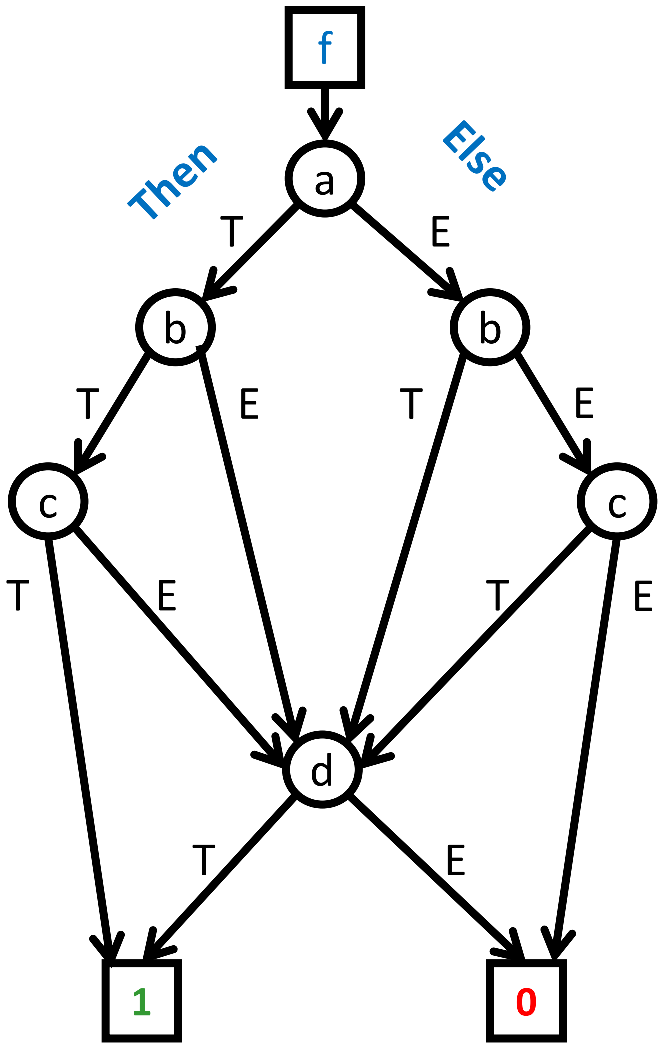
Binary Decision Diagram (BDD)



$M := \{a = T, b = T, c = T, d = T\}$

M is a satisfying assignment

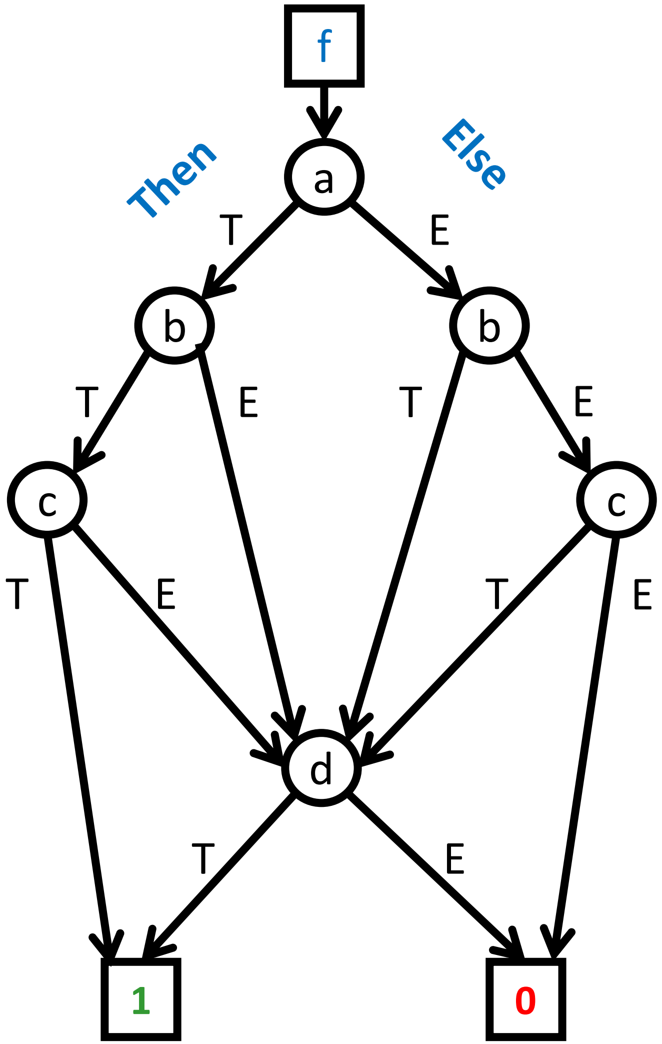
Binary Decision Diagram (BDD)

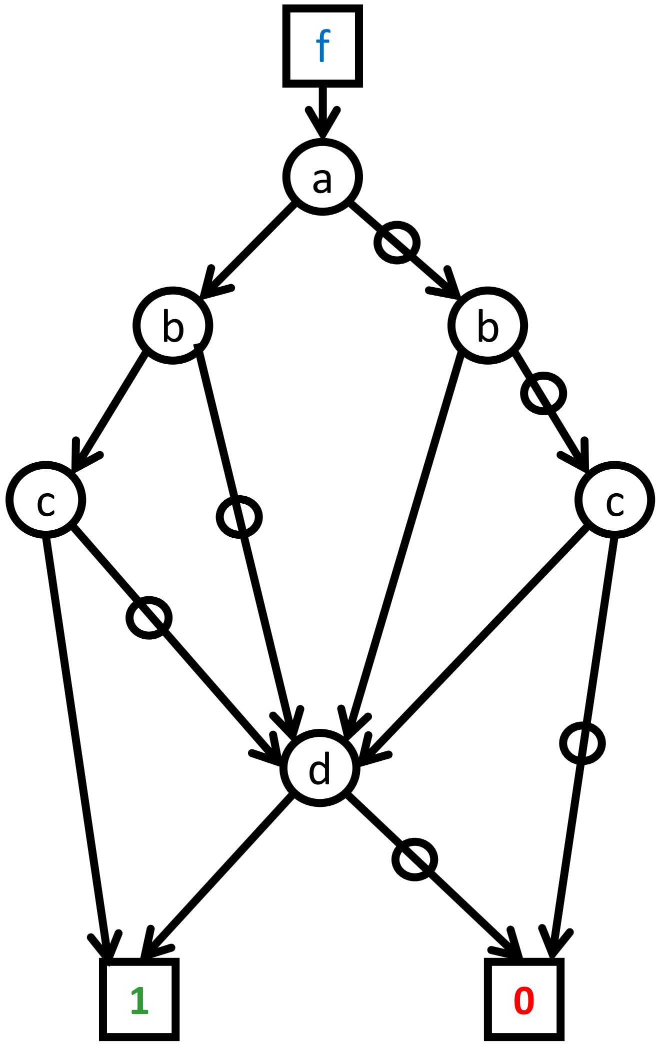


Find the formula f that is represented by this BDD:

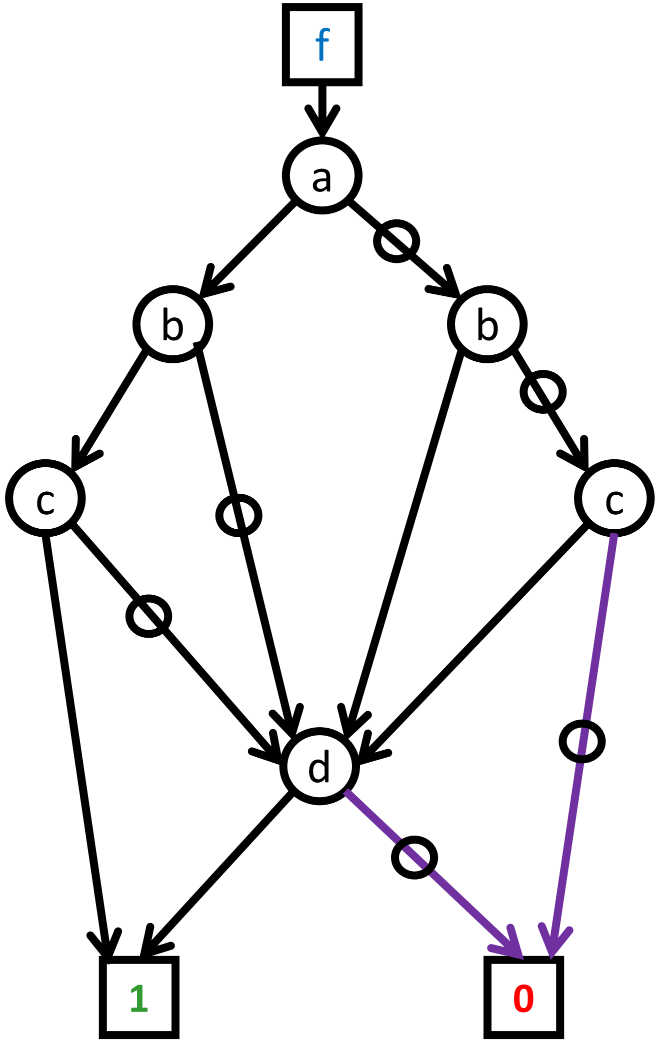
$$f := (a \wedge b \wedge c) \vee (a \wedge b \wedge \neg c \wedge d) \vee (a \wedge \neg b \wedge d) \vee (\neg a \wedge b \wedge d) \vee (\neg a \wedge \neg b \wedge \mathrm{c} \wedge d)$$

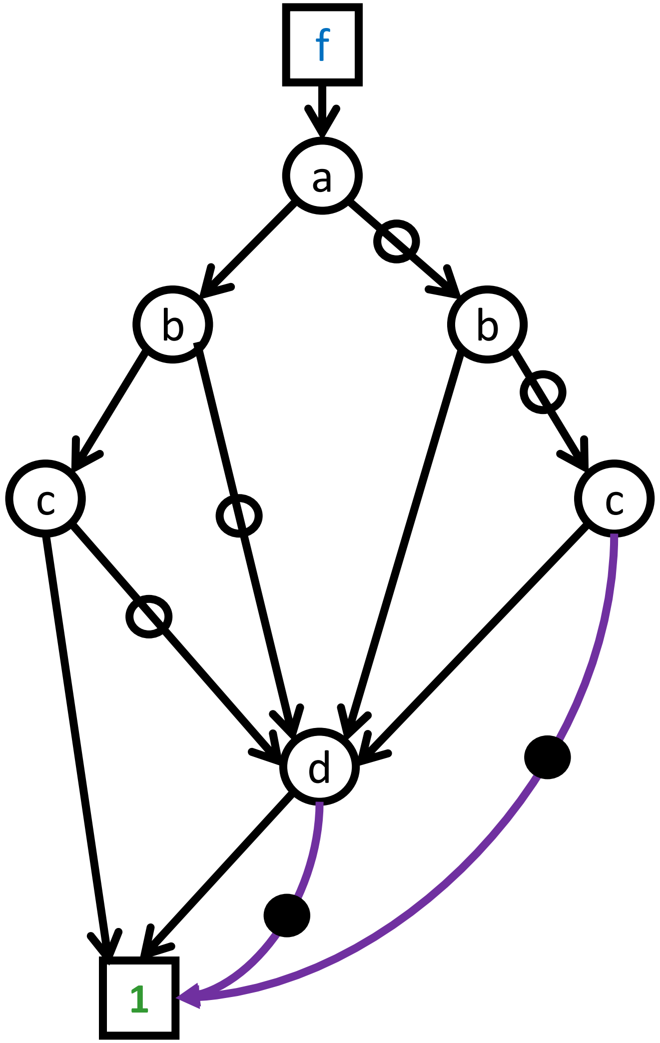
Binary Decision Diagram (BDD)



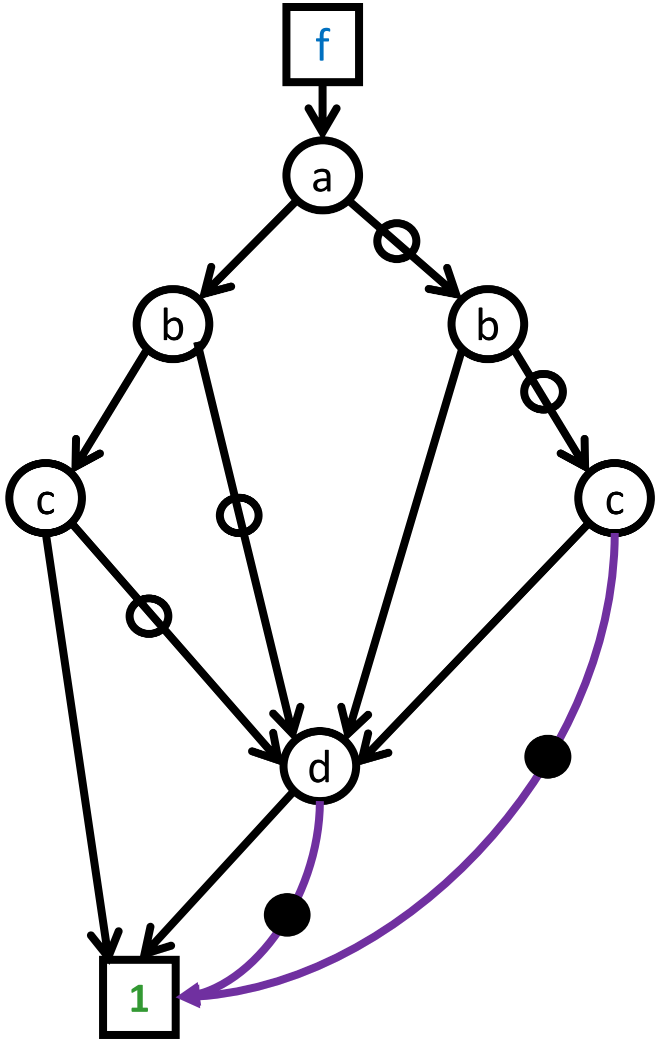


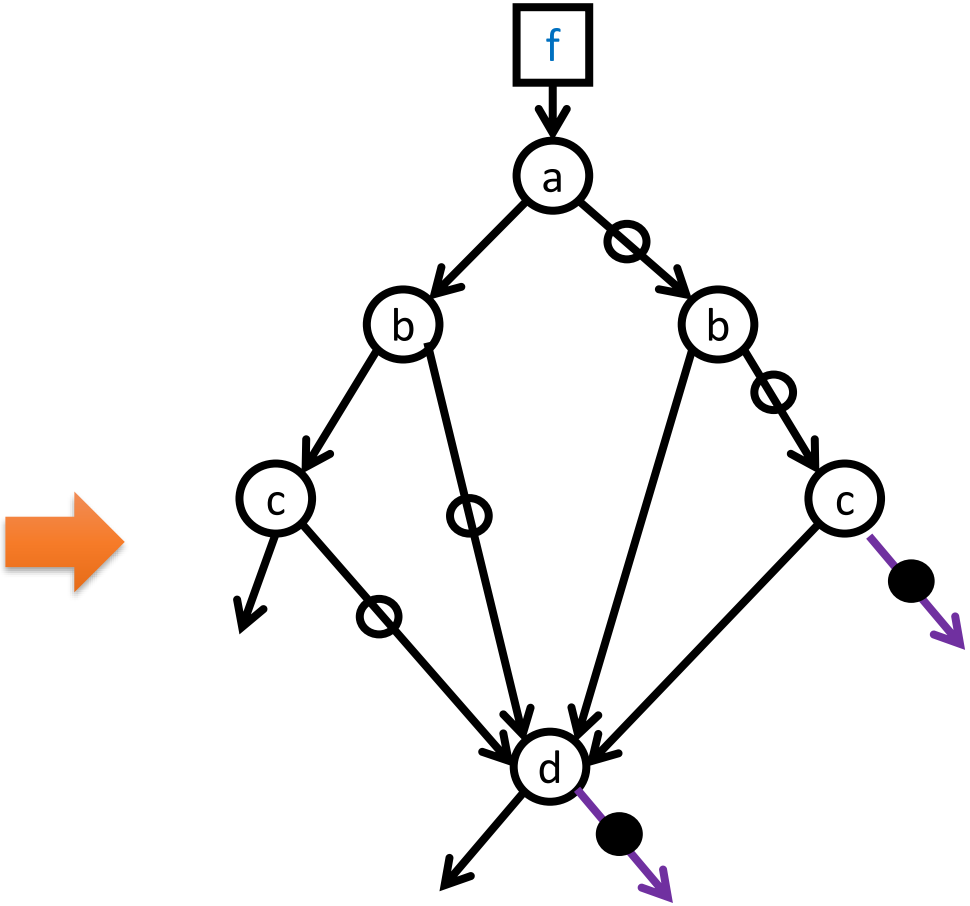
BDD with Complimented Edges



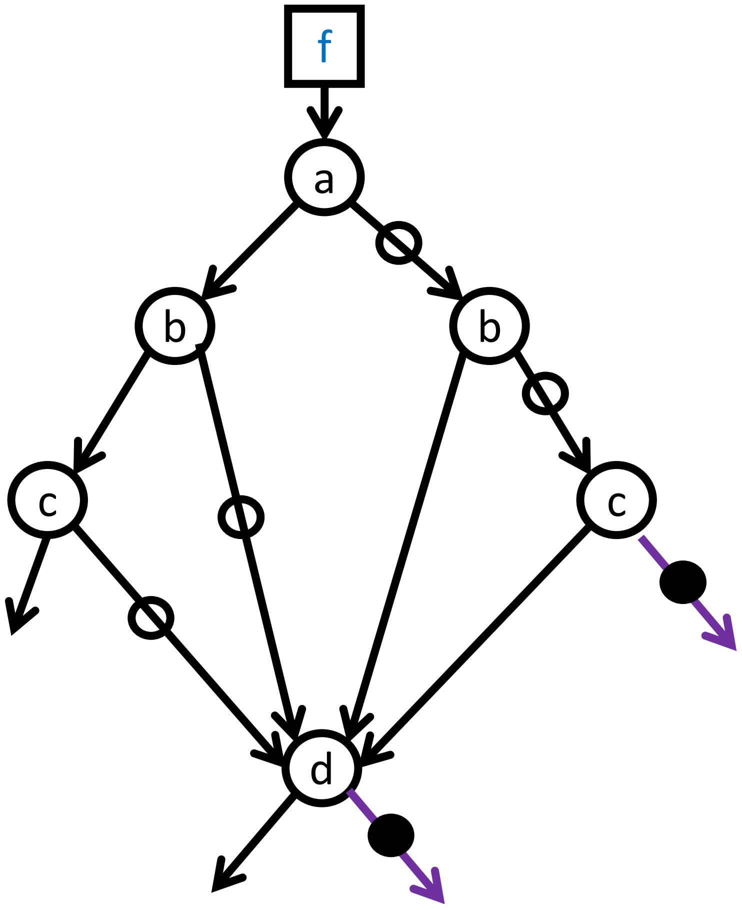


BDD with Complimented and dangling Edges



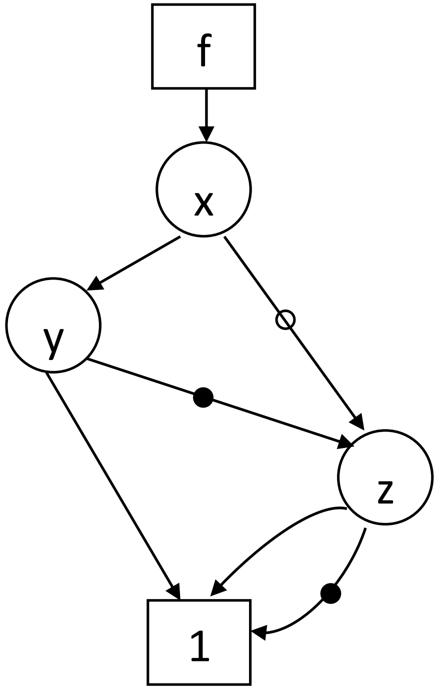


From now on....



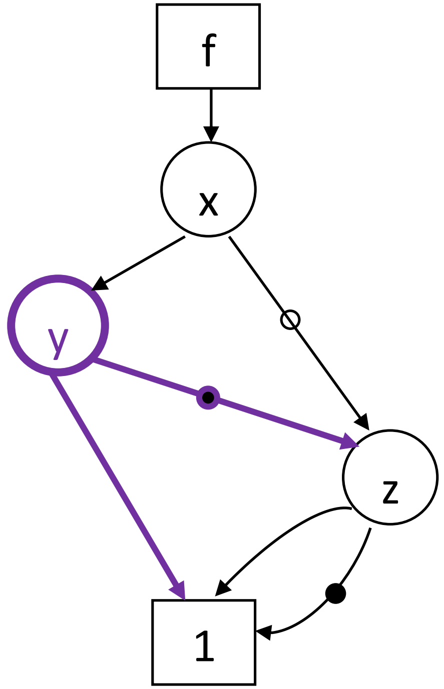
Definition of BDDs

- Directed Acyclic Graph
- $(V \cup \Phi \cup \{\mathbf{1}\},\ E)$
  - Internal Nodes $v \in V$
  - Function Nodes $f_i \in \Phi$
  - Terminal Node $\mathbf{1}$
  - Edges E
    - "Complement" attribute



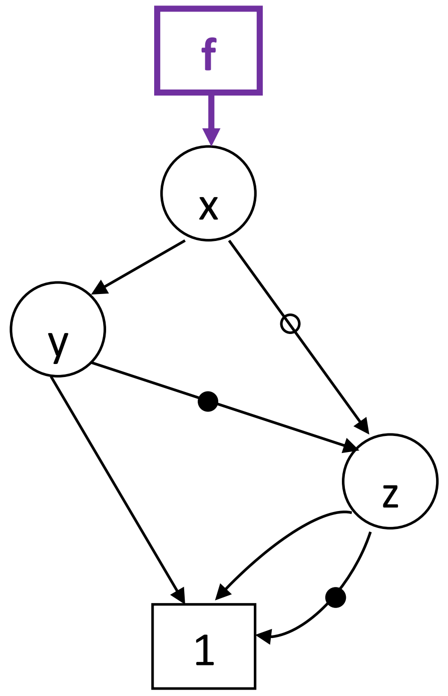
Definition of BDDs: Internal Node

- Label $l(v) \in \{x_1, \ldots, x_n\}$
  - Variables of f
- Out–degree: 2
  - Then-Edge T
  - Else-Edge E
    - Marked with (empty) circle
    - Can have complement attribute (full cycle)



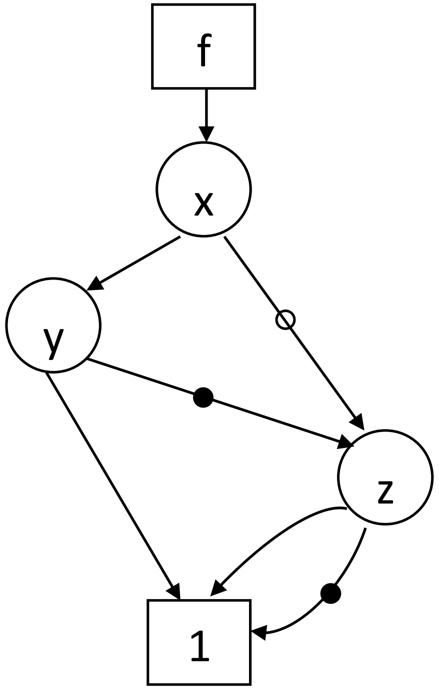
Definition of BDDs: Function Node

- Represents Boolean Formula f_i
- In-degree: 0
- Out-degree: 1
  - Edge can have complement attribute



Definition of BDDs: Terminal Node

- Constant Function **True**
- Out-degree: 0

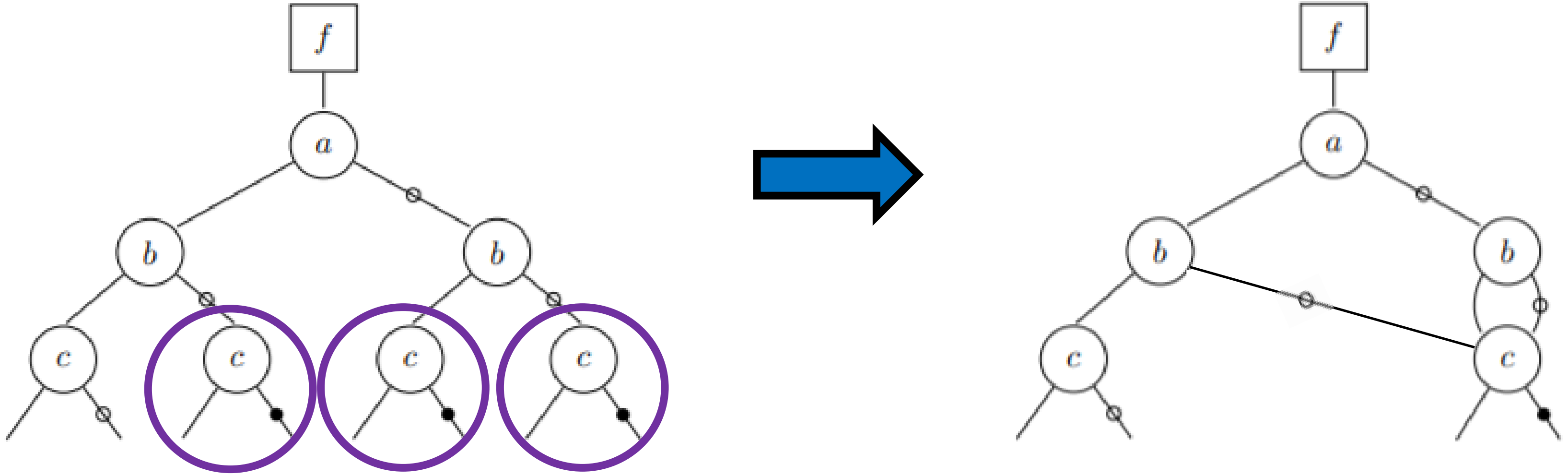


Outline

- What are Binary Decision Diagrams (BDDs)?
  - Intuitive Explanation
  - Formal Definition
- **From BDDs to Reduced Ordered BDDs (ROBDDs)**
- Construct Formula from ROBDD
- Construct ROBDD from Formula
- Pros and Cons of BDDs

From BDD to Reduced BDD

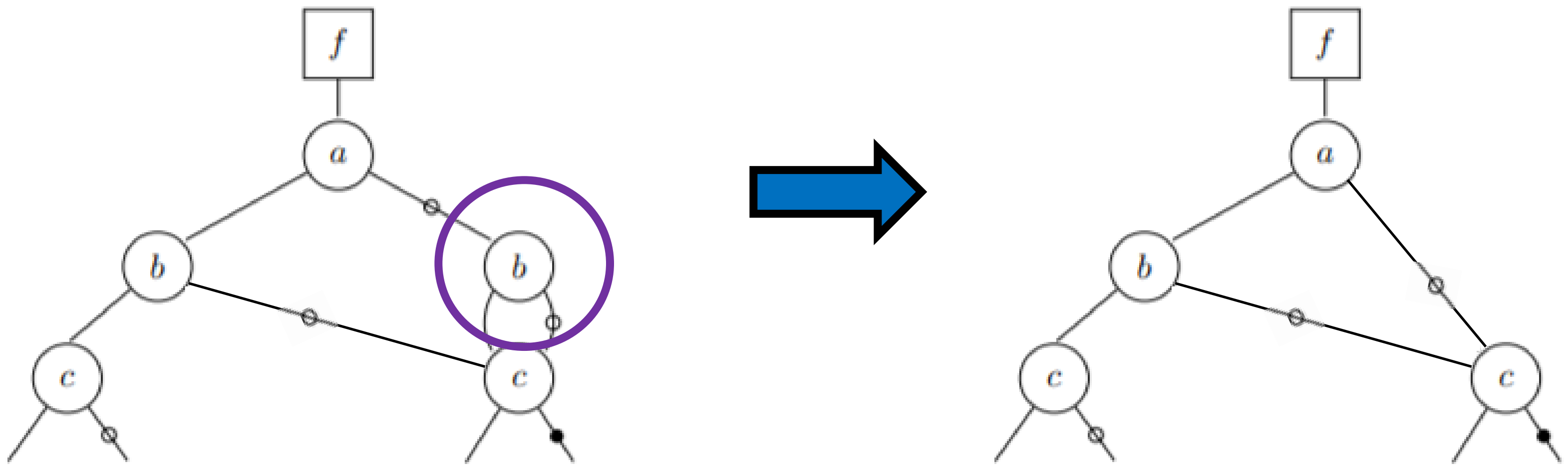
1. No duplicate sub-BDDs



From BDD to Reduced BDD

1. No duplicate sub-BDDs
2. No redundant nodes

} Reduced BDD

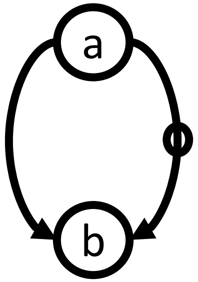


From BDD to Reduced BDD

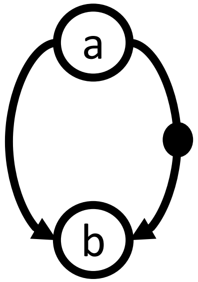
1. No duplicate sub-BDDs
2. No redundant nodes

} Reduced BDD

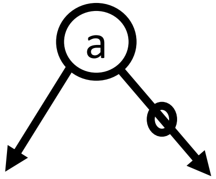
Redundant



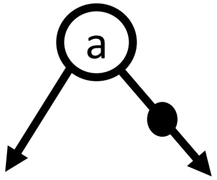
NOT Redundant



Redundant (special case)

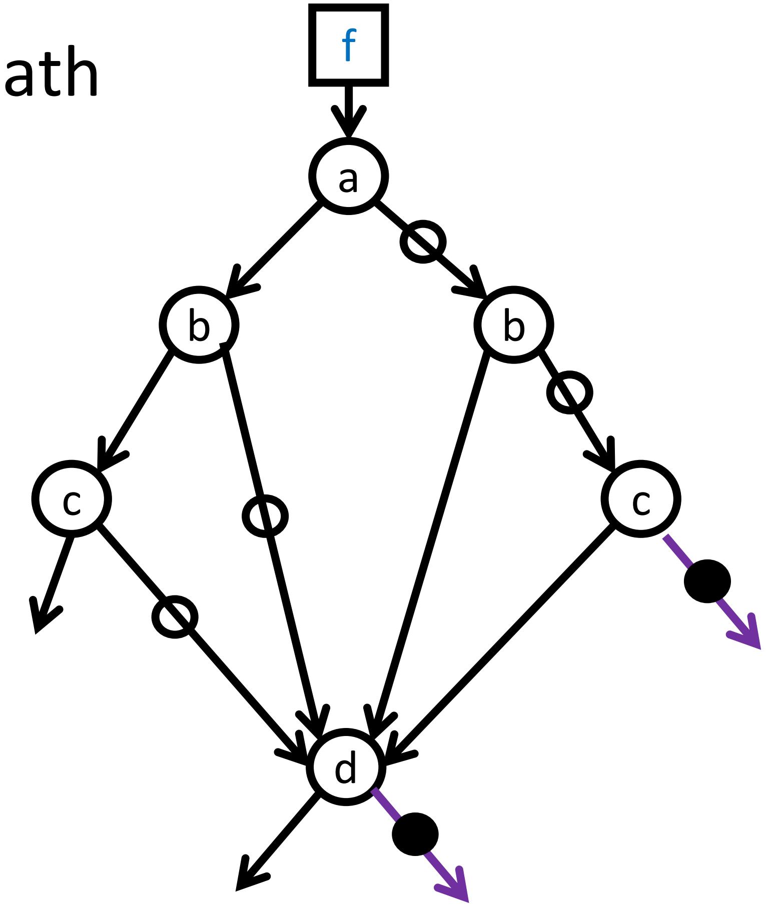


NOT Redundant (special case)



From RBDD to Reduced Ordered BDD (ROBDD)

- Ordering on the variables along any path
  - E.g., $a < b < c < \mathrm{d}$
- A ROBDD gives a **canonical** representation of a formula
  - For given variable ordering
  - Meaning:
    - If two formulas are semantically equivalent, they will be represented by the exact same ROBDD
  - Allows Equivalence Checking in constant time



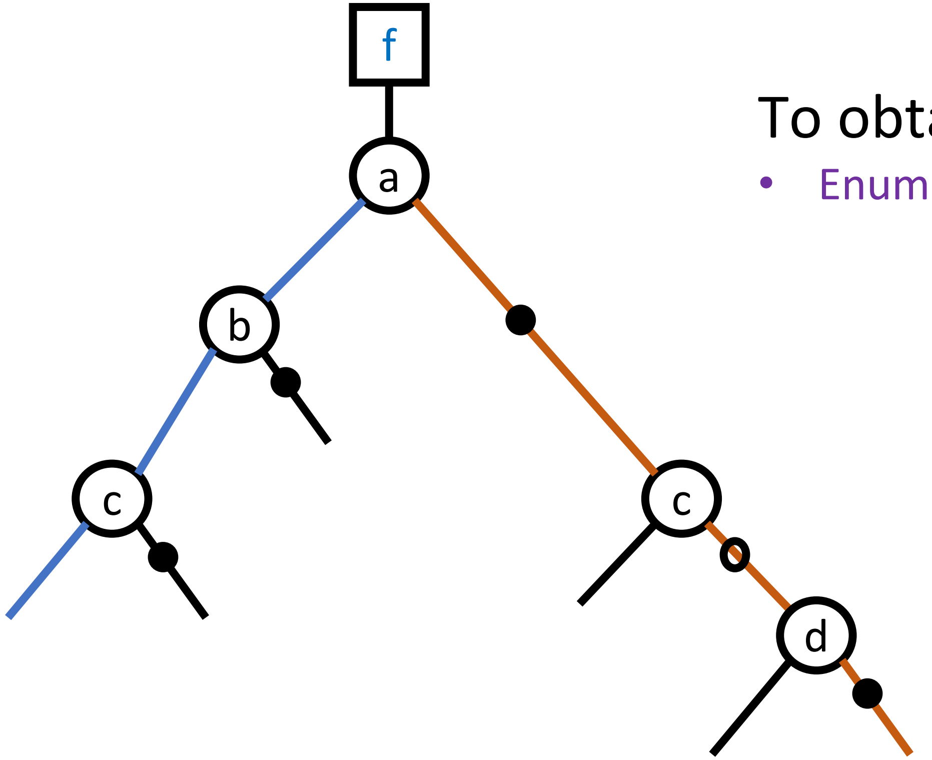
Outline

- What are Binary Decision Diagrams (BDDs)?
  - Intuitive Explanation
  - Formal Definition
- From BDDs to Reduced Ordered BDDs (ROBDDs)

- Construct Formula from ROBDD
- Construct ROBDD from Formula

- Pros and Cons of BDDs

From ROBDD to Formula, Example 1

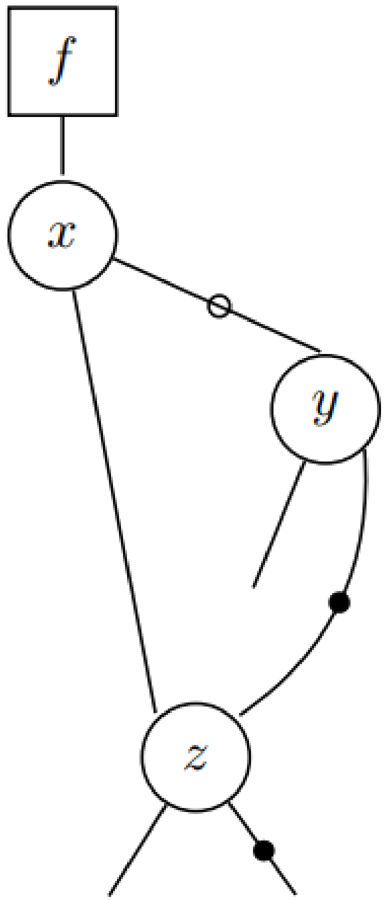


To obtain a formula in DNF:

- Enumerate all paths with an even number of negations (full cycles)

$$f = (a \land b \land c) \lor (\neg a \land \neg c \land \neg d)$$

From ROBDD to Formula, Example 2



To obtain a formula in DNF:

- Enumerate all paths with an even number of negations (full cycles)

$$f = (x \land z) \lor (\neg x \land y) \lor (\neg x \land \neg y \land \neg z)$$

Outline

- What are Binary Decision Diagrams (BDDs)?
  - Intuitive Explanation
  - Formal Definition
- From BDDs to Reduced Ordered BDDs (ROBDDs)
- Construct Formula from ROBDD
- Construct ROBDD from Formula
- Pros and Cons of BDDs

From Formula to BDD

1. Compute all Cofactors
2. Draw ROBDD from Cofactors
3. Shift Negations Upwards

From Formula to BDD – Step 1: Cofactors

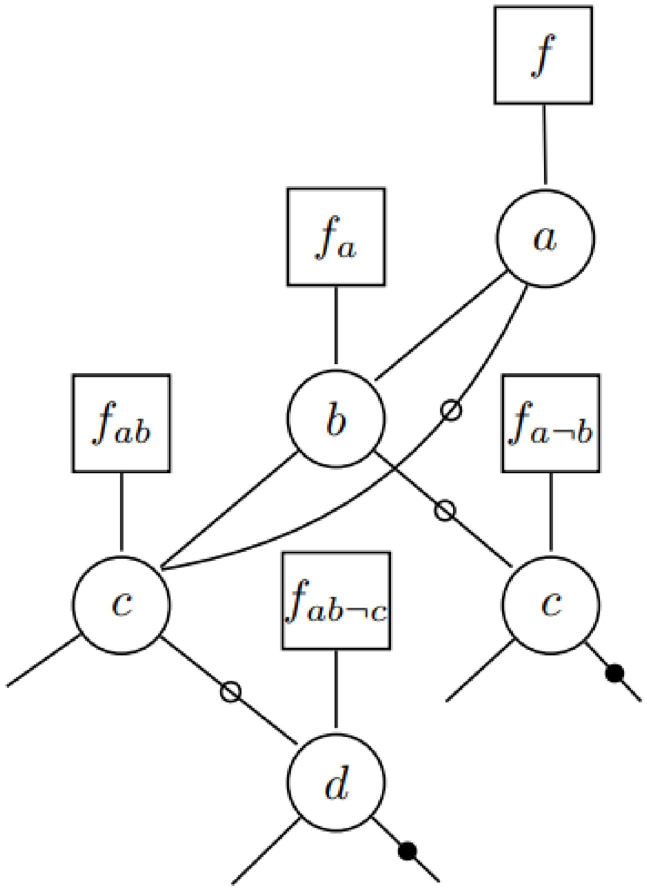
- Boolean formula f w.r.t. a variable x
  - Positive Cofactor f_x: f with x set to ⊤
  - Negative Cofactor $f_{\neg x}$: f with x set to ⊥

- Example:
  - $f = (x \wedge y) \vee (\neg x \wedge z)$
    - $f_x = y$
    - $f_{\neg x} = z$

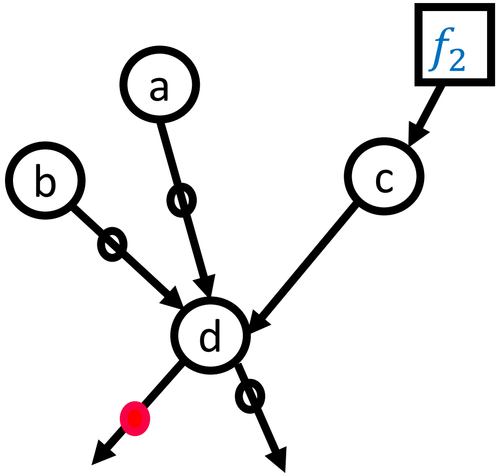
Example: From Formula to BDD

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

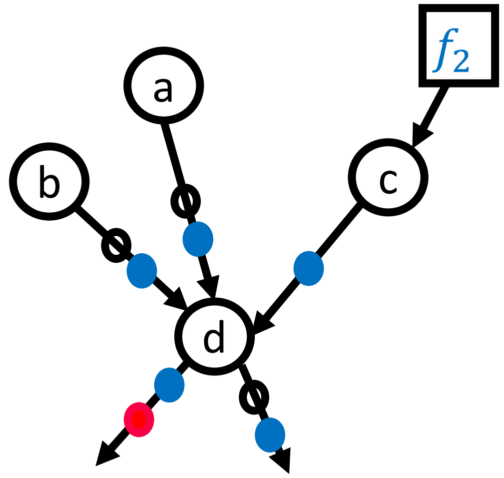
$$f_a = b \wedge \neg c \wedge d \vee c$$
$$f_{ab} = \neg c \wedge d \vee c$$
$$f_{abc} = \top$$
$$f_{ab\neg c} = d$$
$$f_{ab\neg cd} = \top$$
$$f_{ab\neg c\neg d} = \bot$$
$$f_{a\neg b} = c$$
$$f_{a\neg bc} = \top$$
$$f_{a\neg b\neg c} = \bot$$
$$f_{\neg a} = \neg c \wedge d \vee c = f_{ab}$$



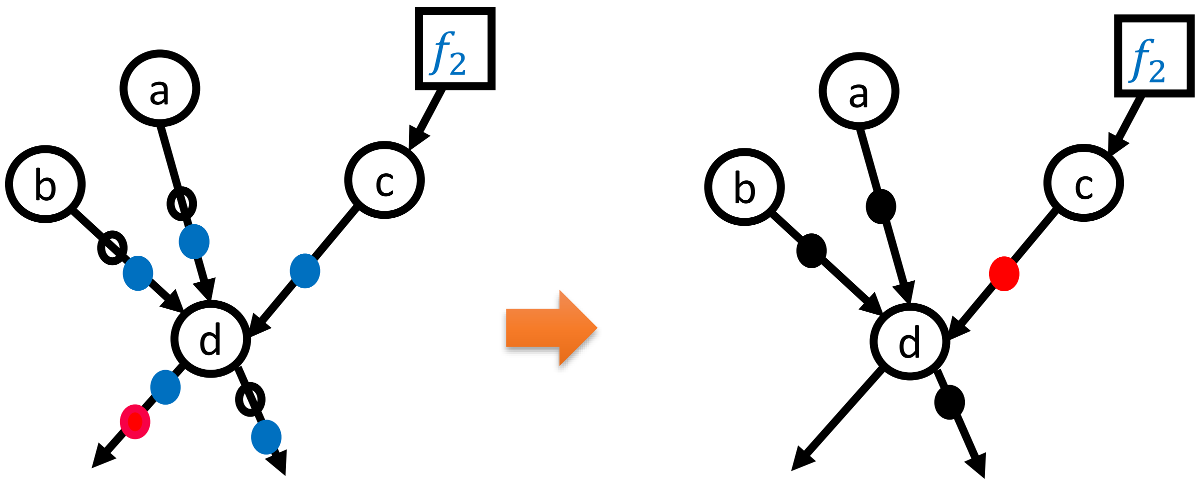
From Formula to BDD – Step 3: Shift Negations Upwards



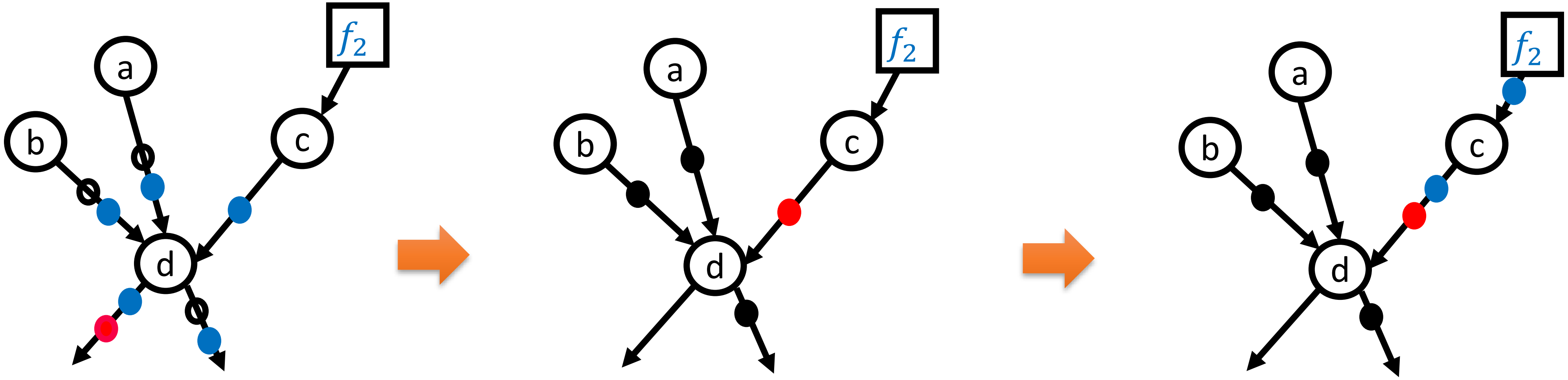
From Formula to BDD – Step 3: Shift Negations Upwards



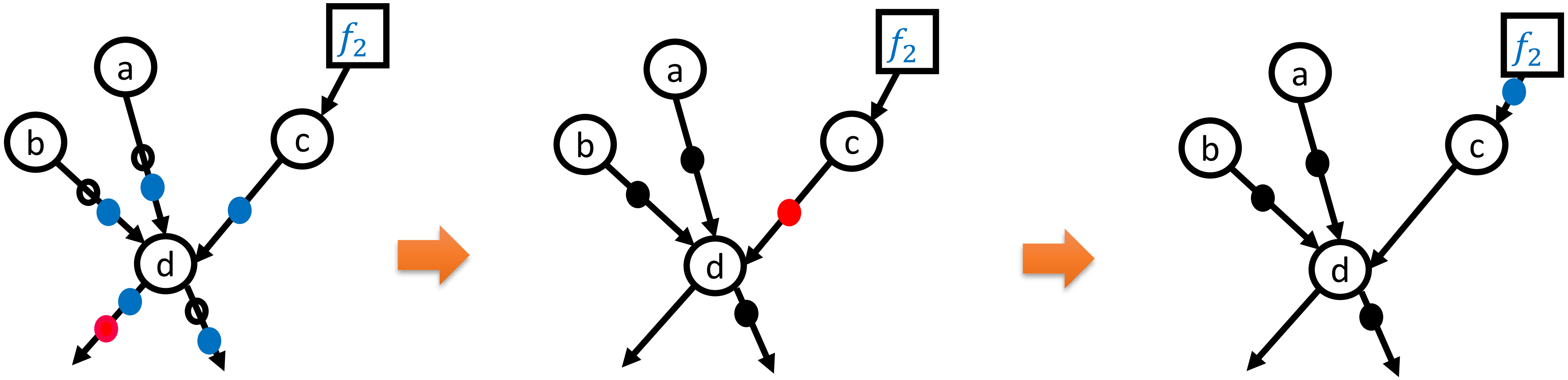
From Formula to BDD – Step 3: Shift Negations Upwards



From Formula to BDD – Step 3: Shift Negations Upwards



From Formula to BDD – Step 3: Shift Negations Upwards



Example: From Formula to BDD

$$f = (a \land \neg c) \lor (\neg a \land (b \lor (\neg b \land c)))$$

Next: Shift negations upwards

$f_a = \neg c$

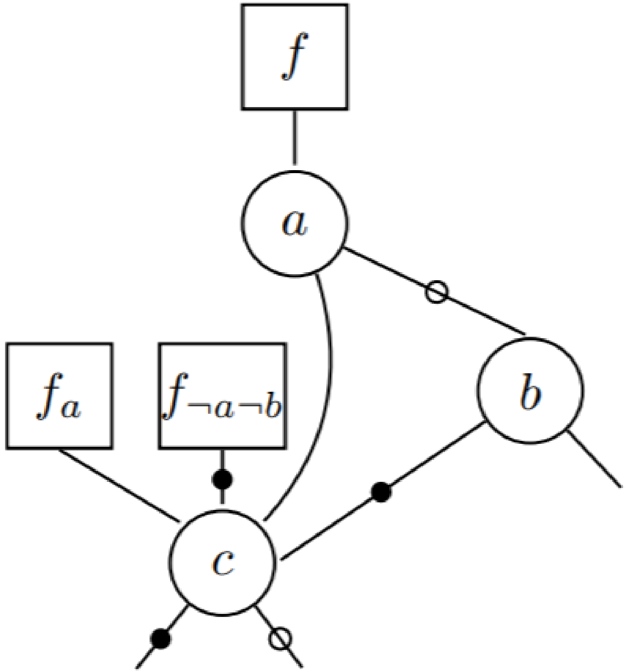
$f_{ac} = \bot$

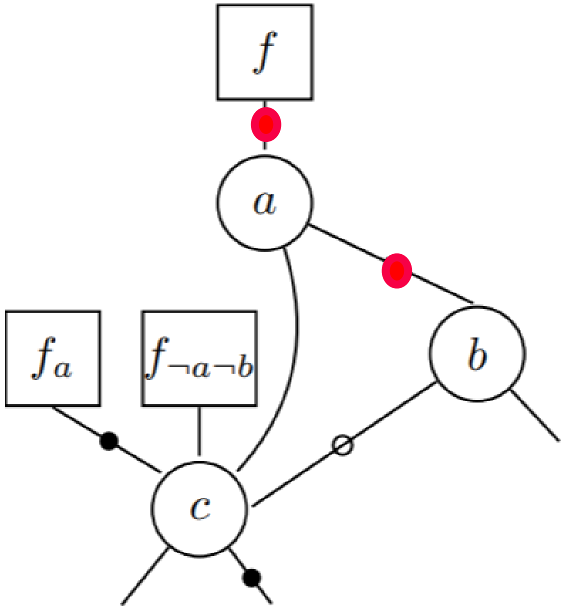
$f_{a\neg c} = \top$

$f_{\neg a} = b \lor (\neg b \land c)$

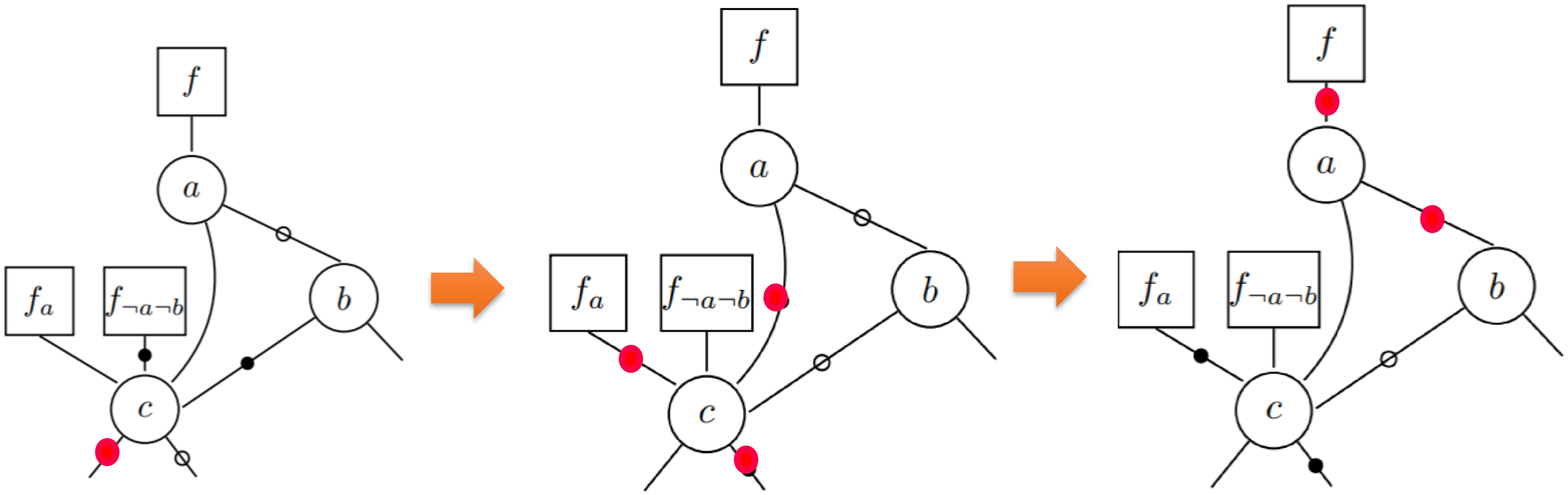
$f_{\neg ab} = \top$

$f_{\neg a \neg b} = c = \neg f_a$





Example: From Formula to BDD

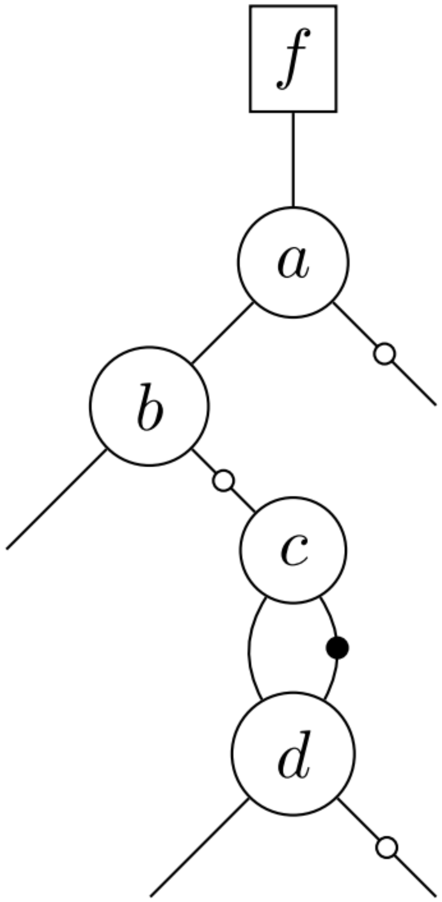


Example: From Formula to BDD

[Lecture] Construct a ROBDD for the following formula using alphabethic ordering:

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

$$f_a = b \vee (c \leftrightarrow d)$$
$$f_{ab} = \top$$
$$f_{a\neg b} = c \leftrightarrow d$$
$$f_{a\neg bc} = d$$
$$f_{a\neg bcd} = \top$$
$$f_{a\neg bc\neg d} = \bot$$
$$f_{a\neg b\neg c} = \neg d = f_{abc}$$
$$f_{\neg a} = \top$$



Example: From Formula to BDD

[Lecture] Construct a ROBDD for the following formula using alphabethic ordering:

$$f = (r \wedge p) \vee (\neg r \wedge \neg p) \vee (s \wedge \neg r) \vee (\neg s \wedge r) \vee (\neg r \wedge q)$$

$$f_p = r \vee (s \wedge \neg r) \vee (\neg s \wedge r) \vee (\neg r \wedge q)$$

$$f_{pq} = r \vee (s \wedge \neg r) \vee (\neg s \wedge r) \vee \neg r = \top$$

$$f_{p\neg q} = r \vee (s \wedge \neg r) \vee (\neg s \wedge r)$$

$$f_{p\neg qr} = \top$$

$$f_{p\neg q\neg r} = s$$

$$f_{p\neg q\neg rs} = \top$$

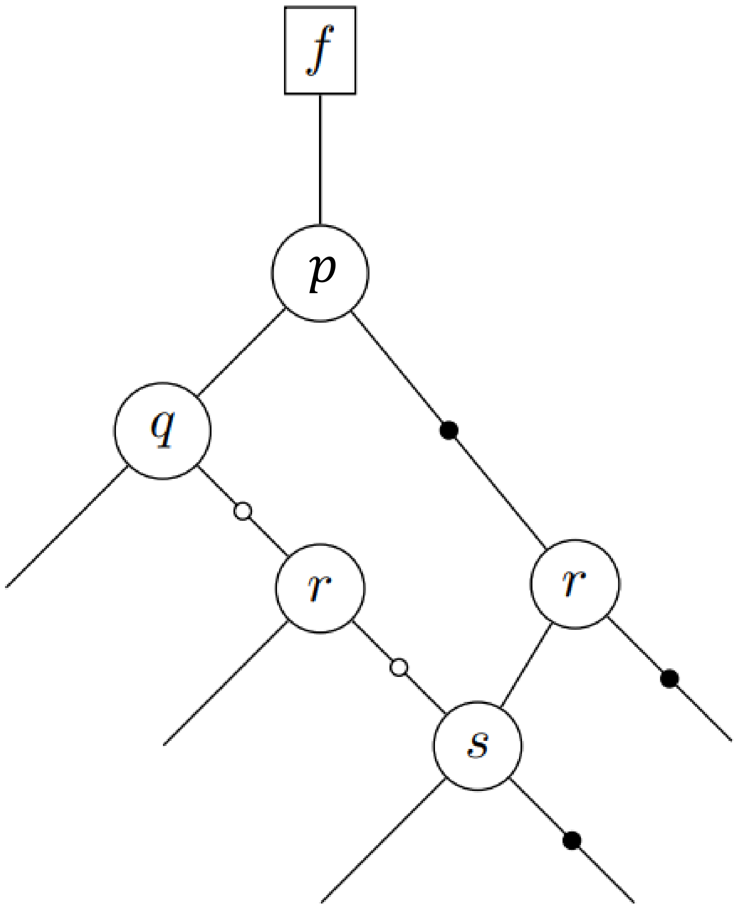
$$f_{p\neg q\neg r\neg s} = \bot$$

$$f_{\neg p} = \neg r \vee (s \wedge \neg r) \vee (\neg s \wedge r) \vee (\neg r \wedge q)$$

$$= \neg r \vee (s \wedge \neg r) \vee (\neg s \wedge r)$$

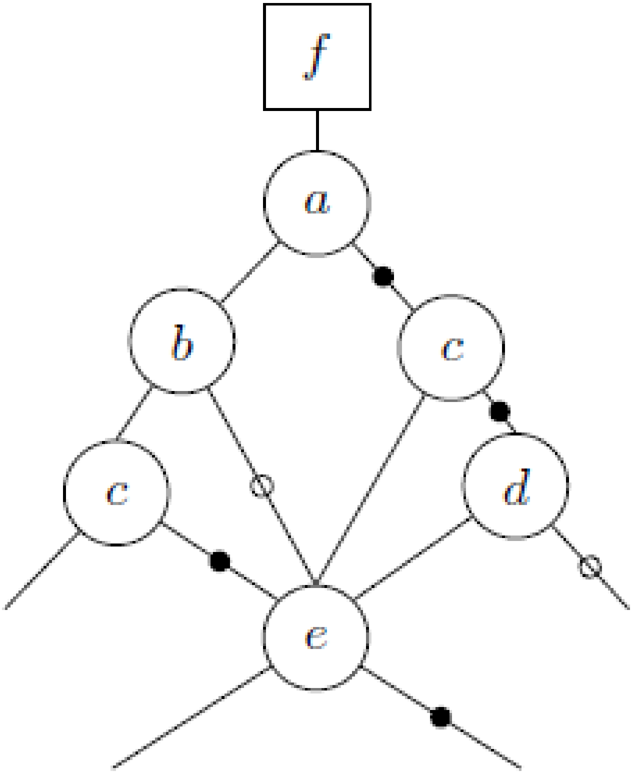
$$f_{\neg pr} = \neg s = f_{p\neg q\neg r}$$

$$f_{\neg p\neg r} = \top$$



Example: From BDD to Formula

[Lecture] Given the *Binary Decision Diagram (BDD)* below. Construct the formula f in disjunctive normal form (DNF) that is represented by the BDD.



Solution:

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

Outline

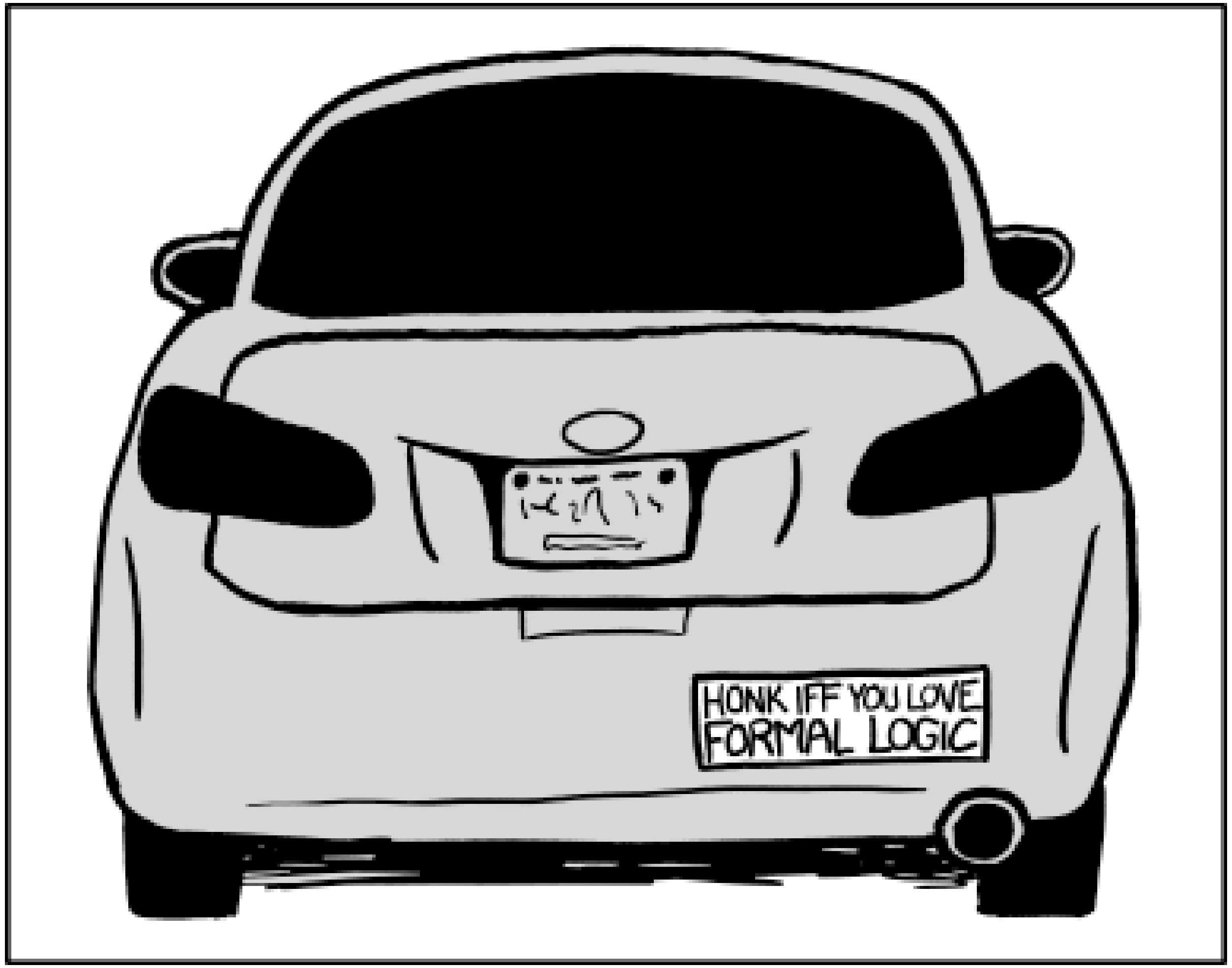
- What are Binary Decision Diagrams (BDDs)?
  - Intuitive Explanation
  - Formal Definition
- From BDDs to Reduced Ordered BDDs (ROBDDs)
- Construct Formula from ROBDD
- Construct ROBDD from Formula
- Pros and Cons of BDDs

Advantages / Disadvantages of BDDs

+ Size-Efficiency
  - Worst case: exponential
  - Often: BDDs contain a lot of redundancy. Eliminating redundancy results in small BDD
+ Efficient Operations
  - e.g. AND, OR: Polynomial time
  - Equivalence Check: Constant time
  - Satisfiability and Validity Check: Constant Time

– Variable order
  - Big impact
  - Hard to optimize

Thank You



https://xkcd.com/1033/