Binary Decision Diagrams
An Introduction and Some Applications

Manas Thakur
PACE Lab, IIT Madras
## Motivating Example

**Binary decision tree for a truth table**

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Motivating Example

Binary decision tree for a truth table

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Collapse redundant nodes
Motivating Example

Collapse redundant nodes

Diagram showing a network of nodes labeled x1, x2, x3, x4 with connections illustrating redundant nodes and their collapse.
Collapse redundant nodes
Collapse redundant nodes
Eliminate unnecessary nodes
We got an ROBDD!!
Overview

1. Motivating Example
2. Introduction
3. Constructing ROBDDs
4. Applications
5. Conclusion
Binary Decision Diagrams

Definition

A Binary Decision Diagram is a rooted DAG with

- One or two terminal nodes of out-degree zero labeled 0 or 1
- A set of variable nodes of out-degree two
A BDD is ordered if on all paths through the graph, the variables respect a given linear order.

\[ b_1 < b_2 < \ldots < b_n \]
Ordered Binary Decision Diagrams (OBDDs)

- A BDD is ordered if on all paths through the graph, the variables respect a given linear order.
  \[ b_1 < b_2 < ... < b_n \]

- An unordered BDD
Ordered Binary Decision Diagrams (OBDDs)

- The size of a BDD depends on the variable ordering

<table>
<thead>
<tr>
<th>( b_1 )</th>
<th>( b_2 )</th>
<th>( b_3 )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- The problem of finding the best variable ordering in OBDDs is NP-Complete
Reduced Ordered Binary Decision Diagrams (ROBDDs)

Definition
An OBDD is reduced if it satisfies the following properties:

- **Uniqueness**
  \[ \text{low}(u) = \text{low}(v) \text{ and } \text{high}(u) = \text{high}(v) \text{ implies } u = v \]

- **Non-redundant tests**
  \[ \text{low}(u) \neq \text{high}(u) \]

We already saw an example of ROBDDs!!
Properties of ROBDDs

- Size is correlated to amount of redundancy, NOT size of relation
  - Insight: As the relation gets larger, the number of don’t-care bits increases, leading to fewer necessary nodes (usually)

- **Canonicity**: For every Boolean function, there is exactly one ROBDD representing it
  - Hence, satisfiability, tautology-check, and equivalence can be tested in deterministic time
  - For Boolean expressions, this problem is NP-Complete
Normal forms for Boolean expressions

- Disjunctive Normal Form (DNF)
  - \((a_1 \land a_2 \land \ldots \land a_n) \lor \ldots \lor (a_1 \land a_2 \land \ldots \land a_n)\)
  - Satisfiability: easy; Tautology check: hard
Normal forms for Boolean expressions

- **Disjunctive Normal Form (DNF)**
  - \((a_1 \land a_2 \land ... \land a_n) \lor ... \lor (a_1 \land a_2 \land ... \land a_n)\)
  - Satisfiability: easy; Tautology check: hard

- **Conjunctive Normal Form (CNF)**
  - \((a_1 \lor a_2 \lor ... \lor a_n) \land ... \land (a_1 \lor a_2 \lor ... \lor a_n)\)
  - Tautology check: easy; Satisfiability: hard
Normal forms for Boolean expressions

- **Disjunctive Normal Form (DNF)**
  - \((a_1 \land a_2 \land ... \land a_n) \lor ... \lor (a_1 \land a_2 \land ... \land a_n)\)
  - Satisfiability: easy; Tautology check: hard

- **Conjunctive Normal Form (CNF)**
  - \((a_1 \lor a_2 \lor ... \lor a_n) \land ... \land (a_1 \lor a_2 \lor ... \lor a_n)\)
  - Tautology check: easy; Satisfiability: hard

- Reduction?
Normal forms for Boolean expressions

- **Disjunctive Normal Form (DNF)**
  - \((a_1 \land a_2 \land ... \land a_n) \lor ... \lor (a_1 \land a_2 \land ... \land a_n)\)
  - Satisfiability: easy; Tautology check: hard

- **Conjunctive Normal Form (CNF)**
  - \((a_1 \lor a_2 \lor ... \lor a_n) \land ... \land (a_1 \lor a_2 \lor ... \lor a_n)\)
  - Tautology check: easy; Satisfiability: hard

- **Reduction?**
  - No hopes since conversion between CNF and DNF is exponential
An If-then-else Normal Form (INF) is a Boolean expression built from the if-then-else operator and the constants 0 and 1, such that all tests are performed only on variables.

\[ x \rightarrow y_0, y_1 = (x \land y_0) \lor (\neg x \land y_1) \]
If-then-else Normal Form (INF)

- An If-then-else Normal Form (INF) is a Boolean expression built from the if-then-else operator and the constants 0 and 1, such that all tests are performed only on variables.
  \[ x \rightarrow y_0, y_1 = (x \land y_0) \lor (\neg x \land y_1) \]

- More examples
  \[ x = x \rightarrow (1, 0) \]
If-then-else Normal Form (INF)

- An If-then-else Normal Form (INF) is a Boolean expression built from the if-then-else operator and the constants 0 and 1, such that all tests are performed only on variables.
  \[ x \rightarrow y_0, y_1 = (x \land y_0) \lor (\neg x \land y_1) \]

- More examples
  \[ x = x \rightarrow (1, 0) \]
  \[ \neg x = (x \rightarrow 0, 1) \]
If-then-else Normal Form (INF)

- An If-then-else Normal Form (INF) is a Boolean expression built from the if-then-else operator and the constants 0 and 1, such that all tests are performed only on variables.

\[ x \rightarrow y_0, y_1 = (x \land y_0) \lor (\neg x \land y_1) \]

- More examples

\[
\begin{align*}
  x &= x \rightarrow (1, 0) \\
  \neg x &= (x \rightarrow 0, 1) \\
  x \lor y &= (x \rightarrow 1, y)
\end{align*}
\]
An If-then-else Normal Form (INF) is a Boolean expression built from the if-then-else operator and the constants 0 and 1, such that all tests are performed only on variables.

\[ x \rightarrow y_0, y_1 = (x \land y_0) \lor (\neg x \land y_1) \]

More examples

\[ x = x \rightarrow (1, 0) \]
\[ \neg x = (x \rightarrow 0, 1) \]
\[ x \lor y = (x \rightarrow 1, y) \]
\[ x \land y = (x \rightarrow y, 1) \]
If-then-else Normal Form (INF)

- An If-then-else Normal Form (INF) is a Boolean expression built from the if-then-else operator and the constants 0 and 1, such that all tests are performed only on variables.

\[ x \rightarrow y_0, y_1 = (x \land y_0) \lor (\neg x \land y_1) \]

- More examples

\[
\begin{align*}
x &= x \rightarrow (1, 0) \\
\neg x &= (x \rightarrow 0, 1) \\
x \lor y &= (x \rightarrow 1, y) \\
x \land y &= (x \rightarrow y, 1) \\
x \iff y &= x \rightarrow (y \rightarrow 1, 0), (y \rightarrow 0, 1)
\end{align*}
\]
Example: \[ t = (x_1 \leftrightarrow y_1) \land (x_2 \leftrightarrow y_2) \]

\[ t = x_1 \rightarrow t_1, t_0 \]
Example: \[ t = (x_1 \Leftrightarrow y_1) \land (x_2 \Leftrightarrow y_2) \]

\[ t = x_1 \rightarrow t_1, t_0 \]
\[ t_0 = y_1 \rightarrow 0, t_{00} \]
Example: \( t = (x_1 \Leftrightarrow y_1) \land (x_2 \Leftrightarrow y_2) \)

\[
\begin{align*}
  t &= x_1 \rightarrow t_1, t_0 \\
  t_0 &= y_1 \rightarrow 0, t_{00} \\
  t_1 &= y_1 \rightarrow t_{11}, 0
\end{align*}
\]
Example: \( t = (x_1 \iff y_1) \land (x_2 \iff y_2) \)

\[
\begin{align*}
t &= x_1 \rightarrow t_1, t_0 \\
t_0 &= y_1 \rightarrow 0, t_{00} \\
t_1 &= y_1 \rightarrow t_{11}, 0 \\
t_{00} &= x_2 \rightarrow t_{001}, t_{000} 
\end{align*}
\]
Example: \( t = (x_1 \leftrightarrow y_1) \land (x_2 \leftrightarrow y_2) \)

\[
\begin{align*}
t &= x_1 \rightarrow t_1, t_0 \\
t_0 &= y_1 \rightarrow 0, t_{00} \\
t_1 &= y_1 \rightarrow t_{11}, 0 \\
t_{00} &= x_2 \rightarrow t_{001}, t_{000} \\
t_{11} &= x_2 \rightarrow t_{111}, t_{110}
\end{align*}
\]
Example: $t = (x_1 \Leftrightarrow y_1) \land (x_2 \Leftrightarrow y_2)$

\[
\begin{align*}
t &= x_1 \rightarrow t_1, t_0 \\
t_0 &= y_1 \rightarrow 0, t_{00} \\
t_1 &= y_1 \rightarrow t_{11}, 0 \\
t_{00} &= x_2 \rightarrow t_{001}, t_{000} \\
t_{11} &= x_2 \rightarrow t_{111}, t_{110} \\
t_{000} &= y_2 \rightarrow 0, 1
\end{align*}
\]
Example: \( t = (x_1 \Leftrightarrow y_1) \land (x_2 \Leftrightarrow y_2) \)

\[
\begin{align*}
    t &= x_1 \rightarrow t_1, t_0 \\
    t_0 &= y_1 \rightarrow 0, t_{00} \\
    t_1 &= y_1 \rightarrow t_{11}, 0 \\
    t_{00} &= x_2 \rightarrow t_{001}, t_{000} \\
    t_{11} &= x_2 \rightarrow t_{111}, t_{110} \\
    t_{000} &= y_2 \rightarrow 0, 1 \\
    t_{001} &= y_2 \rightarrow 1, 0
\end{align*}
\]
Example: $t = (x_1 \Leftrightarrow y_1) \land (x_2 \Leftrightarrow y_2)$

\[
\begin{align*}
t &= x_1 \rightarrow t_1, t_0 \\
t_0 &= y_1 \rightarrow 0, t_{00} \\
t_1 &= y_1 \rightarrow t_{11}, 0 \\
t_{00} &= x_2 \rightarrow t_{001}, t_{000} \\
t_{11} &= x_2 \rightarrow t_{111}, t_{110} \\
t_{000} &= y_2 \rightarrow 0, 1 \\
t_{001} &= y_2 \rightarrow 1, 0 \\
t_{110} &= y_2 \rightarrow 0, 1
\end{align*}
\]
Example: \( t = (x_1 \iff y_1) \land (x_2 \iff y_2) \)

\[
\begin{align*}
t &= x_1 \rightarrow t_1, t_0 \\
t_0 &= y_1 \rightarrow 0, t_{00} \\
t_1 &= y_1 \rightarrow t_{11}, 0 \\
t_{00} &= x_2 \rightarrow t_{001}, t_{000} \\
t_{11} &= x_2 \rightarrow t_{111}, t_{110} \\
t_{000} &= y_2 \rightarrow 0, 1 \\
t_{001} &= y_2 \rightarrow 1, 0 \\
t_{110} &= y_2 \rightarrow 0, 1 \\
t_{111} &= y_2 \rightarrow 1, 0
\end{align*}
\]
Constructing ROBDDs

Example: 
\[ t = (x_1 \Leftrightarrow y_1) \land (x_2 \Leftrightarrow y_2) \]

\[ t = x_1 \rightarrow t_1, t_0 \]
\[ t_0 = y_1 \rightarrow 0, t_{00} \]
\[ t_1 = y_1 \rightarrow t_{11}, 0 \]
\[ t_{00} = x_2 \rightarrow t_{001}, t_{000} \]
\[ t_{11} = x_2 \rightarrow t_{111}, t_{110} \]
\[ t_{000} = y_2 \rightarrow 0, 1 \]
\[ t_{001} = y_2 \rightarrow 1, 0 \]
\[ t_{110} = y_2 \rightarrow 0, 1 \]
\[ t_{111} = y_2 \rightarrow 1, 0 \]
Example: $t = (x_1 \Leftrightarrow y_1) \land (x_2 \Leftrightarrow y_2)$

$t = x_1 \rightarrow t_1, t_0$
$t_0 = y_1 \rightarrow 0, t_{00}$
$t_1 = y_1 \rightarrow t_{11}, 0$
$t_{00} = x_2 \rightarrow t_{001}, t_{000}$
$t_{11} = x_2 \rightarrow t_{111}, t_{110}$
$t_{000} = y_2 \rightarrow 0, 1$
$t_{001} = y_2 \rightarrow 1, 0$
$t_{110} = y_2 \rightarrow 0, 1$
$t_{111} = y_2 \rightarrow 1, 0$
Example: $t = (x_1 \leftrightarrow y_1) \land (x_2 \leftrightarrow y_2)$

$t = x_1 \rightarrow t_1, t_0$
$t_0 = y_1 \rightarrow 0, t_{00}$
$t_1 = y_1 \rightarrow t_{11}, 0$
$t_{00} = x_2 \rightarrow t_{001}, t_{000}$
$t_{11} = x_2 \rightarrow t_{111}, t_{110}$
$t_{000} = y_2 \rightarrow 0, 1$
$t_{001} = y_2 \rightarrow 1, 0$
$t_{110} = y_2 \rightarrow 0, 1$
$t_{111} = y_2 \rightarrow 1, 0$
Let’s move ahead

1. Motivating Example
2. Introduction
3. Constructing ROBDDs
4. Applications
5. Conclusion
Applications of BDDs

- Correctness of Combinational Circuits
- Equivalence of Combinational Circuits
- Model Checking

And yes, Program Analysis!
BDDs for representing Points-to relation

- Points-to analysis using BDDs. Berndl et al. PLDI’03.
- Let $a, b, c$ be reference variables and $A, B, C$ be object references.
- The points-to relation $(a, A), (a, B), (b, A), (b, B), (c, A), (c, B), (c, C)$ is represented as:
BDDs for representing Points-to relation

- Points-to analysis using BDDs. Berndl et al. PLDI’03.
- Let a, b, c be reference variables and A, B, C be object references.
- The points-to relation \((a, A), (a, B), (b, A), (b, B), (c, A), (c, B), (c, C)\) is represented as:

  On the board.

On the Board

1 Represent program information using relations

Express relations as BDDs

On the Board

1. Represent program information using relations
   *Express relations as BDDs*

2. Write program analyses as Datalog queries
   *Express queries as BDD operations*

On the Board

1. Represent program information using relations
   *Express relations as BDDs*

2. Write program analyses as Datalog queries
   *Express queries as BDD operations*

3. Get solutions!
   *Perform operations on BDDs*
Pointers for the enthusiast

- An Introduction to Binary Decision Diagrams. Tutorial by Henrik Reif Andersen.
- Fun with Binary Decision Diagrams. Video lecture by Donald Knuth.
Conclusion

BDDs are very interesting and useful!
Conclusion

BDDs are very interesting and useful!