Model Checking for Propositions

- Problem: Would like an efficient way to answer for a given proposition \( P \):
  - Is \( P \) a tautology?
  - Is \( P \) satisfiable?
    - Note: And general algorithm to answer the first can be used to answer the second and vice versa.
  - Does a given valuation satisfy \( P \)?
- Difficulty: Answering if \( P \) is satisfiable is NP-complete
- Algorithms exist with good performance in general practice
- BDDs are one such

Binary Decision Trees

- Binary decision tree is a (rooted, directed) edge and vertex labeled tree with two types of vertices — internal nodes, and leaves — such that:
  - Internal nodes have exactly two out edges
  - Leaves have no out edges
  - Internal nodes are labeled by atomic propositions (variables)
  - Leaves are labeled by \( \text{true} \) or \( \text{false} \).
  - Left edges labeled \( \text{false} \) and right edges labeled \( \text{true} \).
  - For each path (branch) in the tree, each atomic proposition may label at most one vertex of that path.

Example:

\[(A \land B) \lor (\neg C)\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example: Different Variable Ordering - Different Tree

\[(A \land B) \lor (\neg C)\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example:

\[(A \land B) \lor (\neg C)\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Elsa L Gunter
2112 SC, UIUC
eugunter@illinois.edu
http://courses.engr.illinois.edu/cs477

Slides based in part on previous lectures by Mahesh Vishwanathan, and by Gul Agha
January 31, 2014
Alternate Syntax for Propositional Logic

- Still have constants \( \{T, F\} \)
- Still have countable set \( AP \) of propositional variables a.k.a. atomic propositions
- Only one ternary connective: the conditional if_then_else
  - First argument only a variable
  - Second and third arguments propositions

Example:

\[ P = (A \land B) \lor \neg C \], variables \{A, B, C\} \\
\[ P_0 = (A \land B) \lor \neg C \] \\
\[ P_1 = \text{if } A \text{ then } (T \land B) \lor \neg C \text{ else } (F \land B) \lor \neg C \]
Example:

\[ P = (A \land B) \lor (\neg C) \], variables \{A, B, C\}

\[
P_0 = (A \land B) \lor (\neg C)
\]

\[
P_1 = \text{if } A \text{ then } (T \land B) \lor (\neg C) \text{ else } (F \land B) \lor (\neg C)
\]

\[
P_2 = \text{if } B \text{ then } (\text{if } A \text{ then } (T \land T) \lor (\neg C) \text{ else } (F \land T) \lor (\neg C))
\]

\[ \text{else if } A \text{ then } (T \land F) \lor (\neg C) \text{ else } (F \land F) \lor (\neg C) \]

\[
P_3 = \text{if } B \text{ then } (\text{if } A \text{ then } T \lor (\neg C) \text{ else } F \lor (\neg C))
\]

\[ \text{else if } A \text{ then } F \lor (\neg C) \text{ else } F \lor (\neg C) \]

Example:

\[ P = (A \land B) \lor (\neg C) \], variables \{A, B, C\}

\[
P_0 = (A \land B) \lor (\neg C)
\]

\[
P_1 = \text{if } A \text{ then } (T \land B) \lor (\neg C) \text{ else } (F \land B) \lor (\neg C)
\]

\[
P_2 = \text{if } B \text{ then } (\text{if } A \text{ then } (T \land T) \lor (\neg C) \text{ else } (F \land T) \lor (\neg C))
\]

\[ \text{else if } A \text{ then } (T \land F) \lor (\neg C) \text{ else } (F \land F) \lor (\neg C) \]

\[
P_3 = \text{if } C \text{ then } (\text{if } B \text{ then } (\text{if } A \text{ then } T \text{ else } F) \text{ else } (F \text{ and } F)) \text{ else } (A \text{ and } F \text{ else } F)
\]

\[ \text{else if } B \text{ then } (\text{if } A \text{ then } T \text{ else } T) \text{ else } (A \text{ and } T \text{ else } T) \]

\[ P_3 \text{ corresponds to second binary decision tree given earlier} \]

- Any proposition is strict if \textit{then}_ \textit{else}_ \textit{form} corresponds directly to a binary decision tree that accepts exactly the valuations that satisfy (model) the proposition.

Example:

\[ P = (A \land B) \lor (\neg C) \], variables \{A, B, C\}

\[
P_0 = (A \land B) \lor (\neg C)
\]

\[
P_1 = \text{if } A \text{ then } (T \land B) \lor (\neg C) \text{ else } (F \land B) \lor (\neg C)
\]

\[
P_2 = \text{if } B \text{ then } (\text{if } A \text{ then } (T \land T) \lor (\neg C) \text{ else } (F \land T) \lor (\neg C))
\]

\[ \text{else if } A \text{ then } (T \land F) \lor (\neg C) \text{ else } (F \land F) \lor (\neg C) \]

\[
P_3 = \text{if } C \text{ then } (\text{if } B \text{ then } (\text{if } A \text{ then } T \text{ else } F) \text{ else } (F \text{ and } F)) \text{ else } (A \text{ and } F \text{ else } F)
\]

\[ \text{else if } B \text{ then } (\text{if } A \text{ then } T \text{ else } T) \text{ else } (A \text{ and } T \text{ else } T) \]

Example:

- Binary decision trees may contain (much) redundancy
- Binary Decision Diagram (BDD): Replace trees by (rooted) directed acyclic graphs
- Require all other conditions still hold
- Generalization of binary decision trees
- Allows for sharing of common subtrees.
- Accepts / rejects valuations as with binary decision trees.
Reduced Ordered Binary Decision Diagrams

- Problem: given proposition may correspond to many different BDDs
- How to create a (compact) canonical BDD for a proposition such that two different propositions are logically equivalent if and only if they have the same (isomorphic) canonical BDD
- Start: order propositional variables \( v_i < v_j \).
- Bryant showed you can obtain such a canonical BDD by requiring
  - Variables should appear in order on each path for root to leaf
  - No distinct duplicate (isomorphic) subtrees (including leaves)

Achieving Canonical Form

- Start with an Ordered BDD (all edges in correct order)
- Repeat following until none apply
  - Remove duplicate leaves: Eliminate all but one leaf with a given label and redirect all edges to the eliminated leaves to the remaining one
  - Remove duplicate nonterminals: If node \( n \) and \( m \) have the same variable label, their left edges point to the same node and their right edges point to the same node, remove one and redirect edges that pointed to it to the other
  - Remove redundant tests: If both out edges node \( n \) point to node \( m \), eliminate \( n \) and redirect all edges coming in to \( n \) to \( m \)
- Bryant gave procedure to do the above that terminates in linear time

Example