CS-E4800 AI Exercises 4 Solutions: Logic and Constraint Satisfaction January 31, 2017

Demonstration: Binary Decision Diagrams

Shannon expansion of a formula ϕ with respect to variable x is

$$(x \wedge \phi[\top/x]) \vee (\neg x \wedge \phi[\bot/x])$$

where $\phi[\psi/x]$ denotes substituting all occurrences of x in ϕ by ψ.

Binary Decision Diagrams *(BDD) are a representation of propositional formulas with many useful properties, including constant-time test for logical equivalence.*

We will illustrate properties of BDDs, including their construction by repeated applications of Shannon expansion.

BDDs are essentially a *normal form* of propositional formulas. They are widely used in AI and e.g. in computer-aided verification, for handling both Boolean circuits and state-space search (reachability). The important properties of BDDS include

- constant-time *logical equivalence* testing
- poly-time *model-counting* (how many satisfying assignments there are?)
- poly-time *logical consequence* (set-inclusion)
- poly-time *conjunction*, *disjunction*, *negation*, and relational operations

These good properties are bought at the cost of *potentially exponential size and construction time.* However, if relevant BDDs have been constructed (without too much time and memory consumed), the efficient operations that become available justify the effort in many applications. BDDs, along with explicit state-space search algorithms (breadth-first, A$*$, greedy-best-first, etc.) and satisfiability-based state-space search (lecture 5), is one of the three main approaches to state-space search.

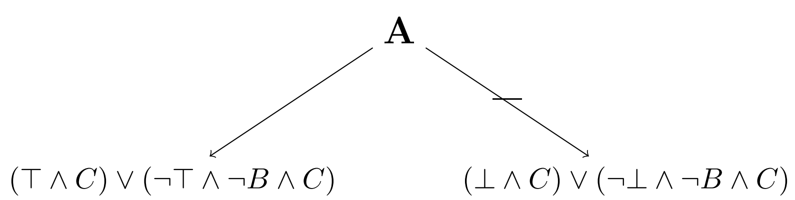
The demonstration: Construct a BDD for $(A \wedge C) \vee (\neg A \wedge \neg B \wedge C)$.

The ordering in which the variables are eliminated is A, B, C, D. Different (but logically equivalent) BDDs are obtained by using other orderings.

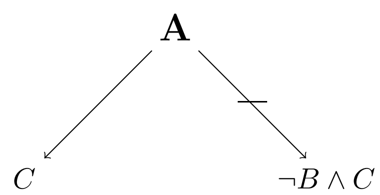
First apply the Shannon expansion with A. This leads to the following formula.

$$((\top \wedge C) \vee (\neg\top \wedge \neg B \wedge C)) \vee (\bot \wedge C) \vee (\neg\bot \wedge \neg B \wedge C)$$

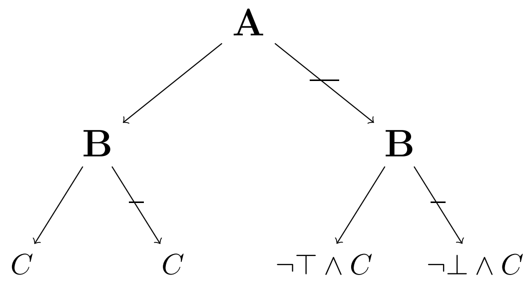
We start to construct a corresponding BDD. The root of the BDD is A, and the two children nodes correspond to the two disjuncts of the above formula, one with $A = \top$ and the other with $A = \bot$.



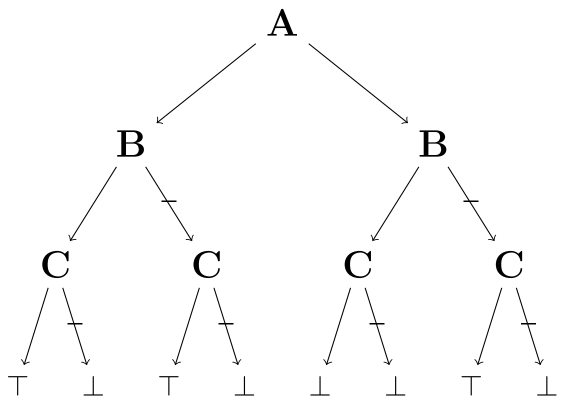
We do logical simplifications:



Next we apply Shannon expansion with B to the formulas in the leaf nodes, and split each disjunct, to form new leaf nodes.



The final Shannon expansions are with C, in all four lead nodes.



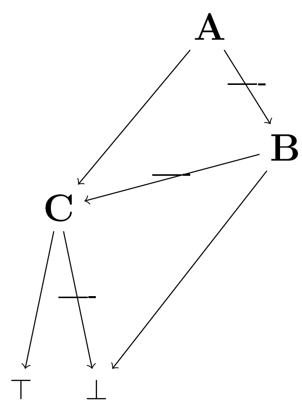
Finally, there are two rules to make sure that the BDD is *reduced*. This means that no isomorphic subtrees may occur.

1. If two nodes have the same label and arcs going to the same (isomorphic) subtrees, merge the nodes: eliminate one of the nodes and re-direct all its incoming arcs to the other node.
2. If the two arcs from a node go to the same child node, delete the node, and re-direct the incoming arc directly tot eh child node.

For the above tree, this means the following:

- The 1st, 2nd and 3rd subtree with label C are merged
- The leftmost subtree with B has two arcs going to the same child node: erase B, and re-direct the incoming node from A to the child node.

The resulting graph is a Binary Decision Diagram.



1 Exercise

2 Exercise

3 Exercise

4 Exercise

Let $X = \{C, D, E, F\}$. Give formulas that represent the following sets. Give as small (simple, intuitive) a formula as possible.

1. $\{0000, 0001\}$
2. $\{0011, 1100\}$
3. $\{0000, 0010, 0110, 1110\}$

Solution

1. $\neg C \wedge \neg D \wedge \neg E$
2. $(C \leftrightarrow D) \wedge (E \leftrightarrow F) \wedge (C \leftrightarrow \neg E)$
3. $\neg F \wedge (C \rightarrow D) \wedge (D \rightarrow E)$

5 Exercise

Let $X = \{C, D, E, F\}$. What sets do the following formulas correspond to?

1. $(C \rightarrow D) \wedge (D \rightarrow E) \wedge (E \rightarrow F)$
2. $E \wedge (C \vee D)$
3. $(C \leftrightarrow D) \wedge (D \leftrightarrow E) \wedge (E \leftrightarrow F)$
4. $(C \leftrightarrow D) \wedge (D \leftrightarrow E)$

Solution

1. $\{0000, 0001, 0011, 0111, 1111\}$
2. $\{0110, 0111, 1010, 1011, 1110, 1111\}$
3. $\{0000, 1111\}$
4. $\{0000, 0001, 1110, 1111\}$