

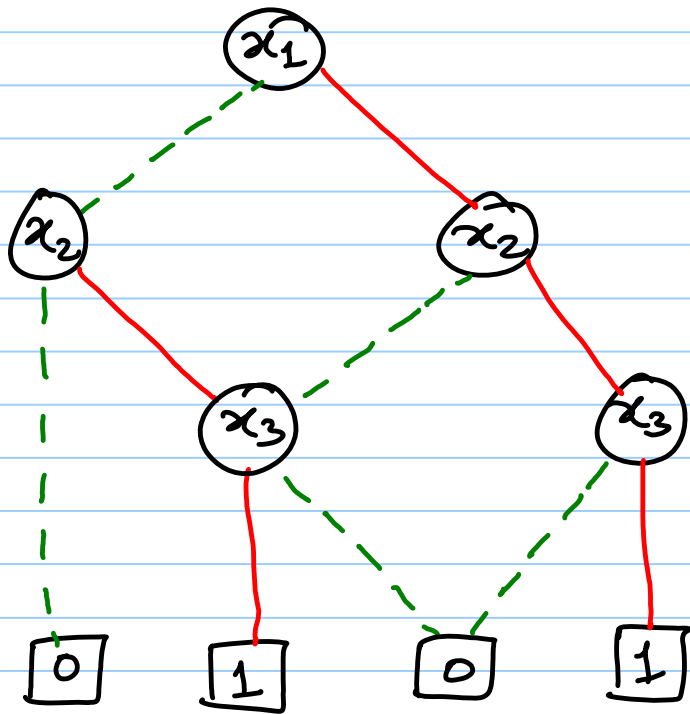
Data structure of BDD nodes

- Let n be a BDD node.
- If n is a leaf node then
 - $val(n)$ denotes its value
- If n is a non-leaf node then
 - $var(n)$ denotes its variable
 - $low(n)$ denotes the node pointed to by green (0) edge
 - $high(n)$ denotes the node pointed to by red (1) edge
- Let $label(n)$ be a function from nodes to integers.

Algorithm REDUCE

1. Let $\text{label}(n) \leftarrow 0$ if $\text{val}(n) = 0$ and n is a leaf (leaf)
 $\text{label}(n) \leftarrow 1$ if $\text{val}(n) = 1$ and n is a leaf
2. If $\text{label}(\text{low}(n)) = \text{label}(\text{high}(n))$ then (redundant)
 $\text{label}(n) \leftarrow \text{label}(\text{low}(n))$
3. If there exists a node m such that $\text{var}(n) = \text{var}(m)$
and $\text{label}(\text{low}(n)) = \text{label}(\text{low}(m))$ and (isomorphic)
 $\text{label}(\text{high}(n)) = \text{label}(\text{high}(m))$ then
 $\text{label}(n) \leftarrow \text{label}(m)$
4. Otherwise, $\text{label}(n) \leftarrow \text{next-label}$
5. Redirect the edges bottom-up according to reduction rules.

Example of application of REDUCE



Example of application of REDUCE

