p84 13
MJCG

1999

Specification & Verification 2: Solution Notes to Question 2

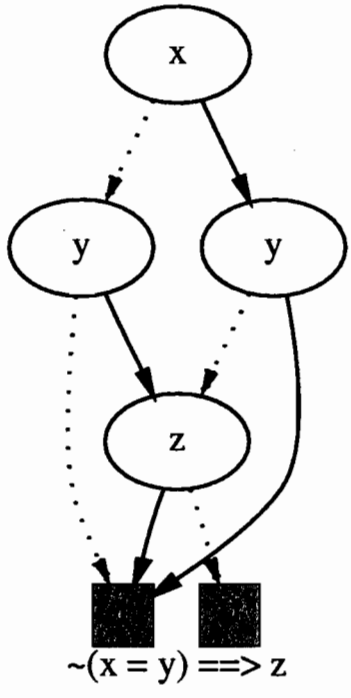
BDDS are an efficient representation of boolean formulae based on encoding as conditionals:

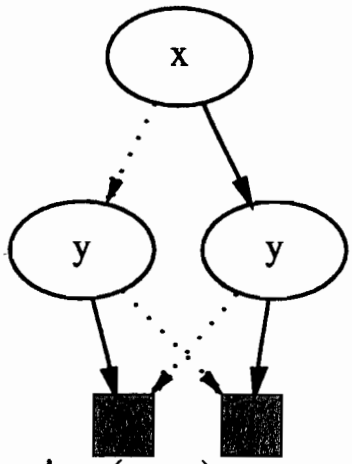
$\neg x \quad = if\ x\ then\ false\ else\ true$
$x \wedge y \quad = if\ x\ then\ y\ else\ false$
$x \vee y \quad = if\ x\ then\ true\ else\ y$

There are two atomic nodes: 1 (true) and 0 (false) and then a compound node, representing a conditional, has a variable label and two pointers to the BDDs representing the true and false cases. Efficient algorithms use "hash-cons" and "memoisation" to construct BDDs.

For a given variable ordering, a BDD is a canonical representation. Thus testing logical equality reduces to pointer equality.

$\forall x.\ P(x)$ is reduced to $P(true) \wedge P(false)$ and $\exists x.\ P(x)$ is reduced to $P(true) \vee P(false)$. The BDDs of these are then computed.





The final part of the question is mostly bookwork. In the lectures a more complex case (three variables not two) is presented, so pure regurgitation won't work (also other minor details have changed).

$$
\begin{aligned}
&\exists \overline{x}\overline{y}.P(\overline{x},\overline{y}) \wedge \mathcal{R}((\overline{x},\overline{y}),(x,y)) \\
&= \exists \overline{x}\overline{y}.P(\overline{x},\overline{y}) \wedge ((x = E_1(\overline{x},\overline{y}) \wedge y = \overline{y}) \vee \\
&\qquad\qquad (x = \overline{x} \wedge y = E_2(\overline{x},\overline{y}))) \\
&= (\exists \overline{x}\overline{y}.P(\overline{x},\overline{y}) \wedge x = E_1(\overline{x},\overline{y}) \wedge y = \overline{y}) \vee \\
&\quad (\exists \overline{x}\overline{y}.P(\overline{x},\overline{y}) \wedge x = \overline{x} \wedge y = E_2(\overline{x},\overline{y}))) \\
&= ((\exists \overline{x}.P(\overline{x},y) \wedge x{=}E_1(\overline{x},y)) \wedge (\exists \overline{y}.y{=}\overline{y})) \vee \\
&\quad ((\exists \overline{x}.x{=}\overline{x}) \wedge (\exists \overline{y}.P(x,\overline{y}) \wedge y{=}E_2(x,\overline{y}))) \\
&= (\exists \overline{x}.P(\overline{x},y) \wedge x = E_1(\overline{x},y)) \vee \\
&\quad (\exists \overline{y}.P(x,\overline{y}) \wedge y = E_2(x,\overline{y}))
\end{aligned}
$$

Computing the BDD of the last formula doesn't involve computing the BDD of the transition relation.

The significance is that the kind of simplification described here enables systems to be analysed without computing the BDD of their transition relations, which might be too big to work with.