# Model Checking WS 2011: Assignment 10

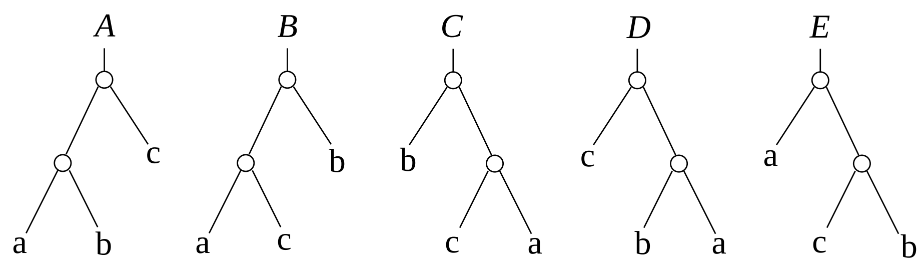
Institute for Formal Models and Verification, JKU Linz

Due 19.01.2012

*For exercises 37 and 38, assume that AIG construction with syntactic sharing, detection of commutativity and unique-tables is implemented as demonstrated in the lecture.*

**Exercise 37**

a) Which of the following logically equivalent AIGs $A$, $B$, $C$, $D$ and $E$, where $a$, $b$, and $c$ are distinct AIG variable nodes, can be recognized as equivalent by syntactic sharing and detection of commutativity? Check *all* possible pairs and justify your answers.




b) AIG t5 is constructed bottom-up in the following four incremental steps. Draw the resulting AIG after step 4 including the effects of all previous steps:

(a) $\texttt{t0} = \texttt{and\_aig}(\texttt{var\_aig}(0), \texttt{var\_aig}(1))$, $\texttt{t1} = \texttt{and\_aig}(\texttt{var\_aig}(0), \texttt{var\_aig}(2))$

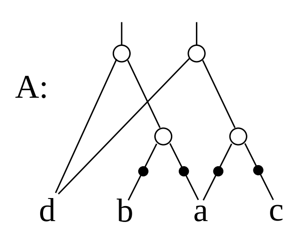
(b) $\texttt{t2} = \texttt{or\_aig}(\texttt{t0}, \texttt{t1})$

(c) $\texttt{t3} = \texttt{and\_aig}(\texttt{var\_aig}(1), \texttt{var\_aig}(0))$, $\texttt{t4} = \texttt{and\_aig}(\texttt{var\_aig}(2), \texttt{var\_aig}(1))$

(d) $\texttt{t5} = \texttt{xor\_aig}(\texttt{t3}, \texttt{t4})$

**Exercise 38**

Given the AIG $A$ shown on the right where $a$, $b$, $c$ and $d$ are distinct AIG variable nodes with variable IDs 0,1,2 and 3, respectively. That is, AIG $A$ has been constructed and all of its nodes have been inserted into the unique-table.



a) After calling

$$\texttt{and\_aig(or\_aig(var\_aig(1), var\_aig(0)), or\_aig(var\_aig(0), var\_aig(2)))}$$

for incrementally constructing an AIG for $(b \vee a) \wedge (a \vee c)$ on the given AIG $A$, draw the resulting AIG and determine all *reference counts*[1] and the total number of nodes.

b) The equivalence $(b \vee a) \wedge (a \vee c) \equiv a \vee (b \wedge c)$, when read from left to right, can be used as an AIG simplification rule: instead of constructing an AIG for $(b \vee a) \wedge (a \vee c)$, an AIG for $a \vee (b \wedge c)$ is constructed. Do you consider this rule as reasonable? Justify your answer.

c) Apply the simplification rule from b): after calling

$$\texttt{or\_aig(var\_aig(0), and\_aig(var\_aig(1), var\_aig(2)))}$$

for incrementally constructing an AIG for $a \vee (b \wedge c)$ *on the original AIG $A$ shown above*, draw the resulting AIG and determine all reference counts and the total number of nodes.

d) Compare and interpret the results from a) and c). Then answer b) again.

**Exercise 39**

Let $f(a,b,c) := (a \wedge \neg b) \vee (\neg a \wedge c)$. Draw a binary decision diagram[2] representing $f$ (try to find a *minimal* one) where

a) $a$ is the root node.

b) $c$ is the root node.

*Please consistently use the graphical notation introduced in the lecture (slide 150).*

**Exercise 40**

a) Draw a binary decision tree with variable order[3] $a > b > c$ for the boolean function $f(a,b,c)$ which returns *true* if, and only if, exactly one argument is *true*. Then draw the ROBDD $F$ for $f$ resulting from the decision tree by applying the reduction rules.

b) Eliminate two nodes in the ROBDD $F$ from the first part by inserting negated edges without changing the function denoted by $F$.

---

[1] The reference count of a node is how often that node is pointed to. For example, in the AIG $A$ shown above, the two AND-nodes on top are roots of the AIG and have a reference count of one.

[2] See slide 150 for an example.

[3] Note that $a$ is "above" $b$ if $a > b$.