SOLUTION NOTES

Logic and Proof 2001 Paper 6 Question 11 (LCP)

The first part is largely bookwork with the obvious substitutions. Ordered **binary** decision diagrams involve variables that are examined in a predetermined order. We convert a binary decision tree to an optimized graph by removing redundant tests (those in which the decision does not affect the outcome) and by identifying equivalent subgraphs. Clearly all of these features are independent of whether the decisions are binary or ternary.

The algorithm for converting an expression to an OTDD is again bookwork. Two OTDDs that both test the variable P are combined into a single one that tests P. Each of the three decisions leads to an OTDD generated by a recursive call. If the variables differ, we make them identical by inserting a redundant test, as described in the notes, and work as in the previous case.

For the $\otimes$ connective, there are obvious optimizations based on $0 \otimes i = 0$ and $1 \otimes i = i$. The case $2 \otimes i$ can simply be evaluated if i is a number and otherwise are handled by the general algorithm. An important efficiency refinement is to keep a hash table of inputs and the resulting outputs, since it is not unusual for the same expression to be converted repeated in the recursive calls.

The attached diagram outlines the construction of the OTDD.

A pure literal is one whose complement doesn't appear anywhere in the clauses. (Bookwork again.) Clause methods are concerned with satisfiability. If the clauses can be satisfied then they can be satisfied while making all pure literals true, since that can only help matters. So we may as well assume that the pure literals are true; therefore the clauses containing them are also true and can be discarded. Discarding clauses can result in further pure literals, and so forth. This step is explicitly built into the Davis-Putnam procedure, but it is also applicable to resolution.

The clauses are satisfiable. There are no pure literals or unit clauses, so we try a case split on P. If $P =$ **true** then the clauses become

$$\{\neg R\} \qquad \{\neg Q, R\} \qquad \{Q, R\}$$

In this case we have a unit clause; we put $R =$ **false** and get the contradictory clauses $\{\neg Q\}$ and $\{Q\}$. There is no model with $P =$ **true**.

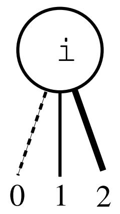
Turning to $P =$ **false**, we are left with the clauses

$$\{R\} \qquad \{\neg Q\} \qquad \{\neg Q, R\}$$

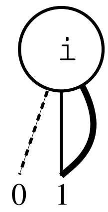
We again have a unit clause, giving $R =$ **true** and leaving the clause $\{\neg Q\}$, giving $Q =$ **false**. We have arrived at the interpretation $P = Q =$ **false** and $R =$ **true**, which we can check by plugging it into the original clauses.

dashed line: variable is 0
thin line: variable is 1
thick line: variable is 2

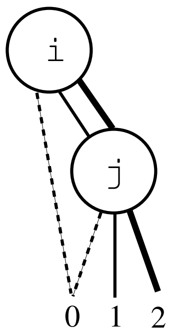
OTDD for the variable *i*



OTDD for the expression $i \otimes i$



OTDD for the expression $(i \otimes i) \otimes j$



OTDD for the expression $((i \otimes i) \otimes j) \otimes 2$