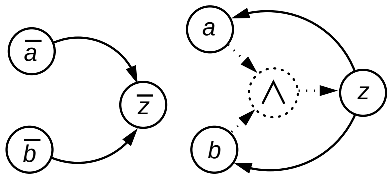

Figure 7.7: Implication graph (solid) and 3-SAT term (dotted) for 2-input AND.

a *transitive closure* [122] graph so that when a node is set to true, all reachable nodes are also set to true. This allows very efficient analysis of signal implications, because the transitive closure can determine more global signal relationships in the graph than other branch-and-bound search methods. Figure 7.7 (solid lines) shows the implication graph for the "if ... then" clauses of Equation 7.6. Note that only binary implications (those involving two literals) can be represented by an edge. The node with $\wedge$ (dotted lines) denotes an ANDing operator, and represents the 3-SAT term of Equation 7.8 (last clause of Equation 7.6) [291]. We cover these algorithms briefly in advanced topics (see Section 7.5.4.)

Computational Complexity. Ibarra and Sahni [317] analyzed the computational complexity of ATPG. They found that it is an *NP-Complete* problem, which means that no polynomial expression for the compute time function was found, and the problem is presumed to have exponential complexity. We will informally discuss how this arises. In the worst case, with no_pi inputs, there are 2^{no_pi} different input combinations to try in depth-first fashion in the binary decision tree. When no_ff flip-flops are present in the circuit, there are potentially 4^{no_ff} different initial flip-flop states for ATPG to consider. This is because a flip-flop can be in either 0 or 1 state in the fault-free circuit and also in 0 or 1 state in the faulty circuit. Thus, the state-space of a flip-flop contains four elements (also see Section 8.2.) Finally, the work to forward simulate or reverse simulate all logic gates, as appropriate, rises proportionately to n, the number of logic gates. In the worst case, this work has to be done for all potential combinations of PIs and initial flip-flop states. The complete expression for the worst-case ATPG computational complexity is:

$$O(n \times 2^{no\_pi} \times 4^{no\_ff})$$

The above proof considers ATPG to be mathematically equivalent to the problem of *Boolean satisfiability.*

The entire history of ATPG algorithms has been a process of improving heuristic algorithms and procedures to (1) find all necessary signal assignments for a test as early as possible, and (2) search as little of the above decision space as possible. The worst-case decision space is $2^{no\_pi} \times 4^{no\_ff}$. For logic simulation, the computational complexity is $O(n)$. For combinational fault simulation, the complexity is $O(n^2)$[277], and for sequential fault simulation, the complexity is estimated to be between $O(n^2)$ and $O(n^3)$, based on empirical measurements. This means that, whenever possible, we will use fault simulation to avoid ATPG computations. For