A Circuit for Coloring

15-251: Great Theoretical Ideas in Computer Science

November 25, 2008

Suppose that we have a graph \( G = (V, E) \) on \( n \) nodes. We’d like to test colorability by devising a circuit. We’ll have a Boolean variable indicating the presence or absence of each possible edge, and three Boolean variables that encode which color is assigned to each node. Specifically,

\[
X_{ij} = \begin{cases} 
1 & \text{if } (i, j) \text{ is an edge} \\
0 & \text{otherwise};
\end{cases}
\]

\[
Y_{ik} = \begin{cases} 
1 & \text{if node } i \text{ is colored } k \in \{\text{red, blue, green}\} \\
0 & \text{otherwise}.
\end{cases}
\]

We ensure that an assignment of Boolean variables \( Y_{ik} \), for \( i = 1, 2, \ldots, n \) and \( k = 1, 2, 3 \) is a valid assignment of colors with following circuit \( \mathcal{E}(Y) \):

\[
\mathcal{E}(Y) = \bigwedge_{i=1}^{n} \{ (Y_{i1} \land \neg Y_{i2} \land \neg Y_{i3}) \lor (Y_{i2} \land \neg Y_{i1} \land \neg Y_{i3}) \lor (Y_{i3} \land \neg Y_{i1} \land \neg Y_{i2}) \}.
\]

In words, if a Boolean variable is true for a color, then the Boolean variables for the other two colors have to be false.

We then test whether an assignment of colors is a valid coloring with the following circuit:

\[
\mathcal{C}(X, Y) = \bigwedge_{i,j=1}^{n} \{ \neg X_{ij} \lor \mathcal{K}(Y_i, Y_j) \}.
\]

In this expression, if \((i, j)\) is not an edge, so that \(\neg X_{ij}\) is true, then we don’t care about the coloring of the endpoints. However, if it is an edge, so that \(\neg X_{ij}\) is false, then we demand that \(\mathcal{K}(Y_i, Y_j)\) is true, where the circuit \(\mathcal{K}(Y_i, Y_j)\) checks that the colors on the two neighboring nodes are not the same:

\[
\mathcal{K}(Y_i, Y_j) = \bigwedge_{k=1}^{3} \neg (Y_{ik} \land Y_{jk})
\]

Putting the pieces together, our overall circuit to test colorability of the graph is then

\[
V(X, Y) = \mathcal{E}(Y) \land \mathcal{C}(X, Y).
\]
This can be efficiently constructed from the graph, in $O(n^2)$ operations. Note that for a given graph, the Boolean variables $X$ are fixed, and we feed in Boolean variables $Y$ to test whether or not it encodes a valid coloring of the graph.

Now suppose that we want to test whether a given graph $G = (V, E)$ has an independent set of size $k$. In a similar fashion as above, we introduce Boolean variables $Z_i$, for $i = 1, 2, \ldots, n$, where now

$$Z_i = \begin{cases} 1 & \text{if node } i \text{ is an element of the independent set} \\ 0 & \text{otherwise.} \end{cases}$$

Then a circuit $I(X, Z)$ that is true if and only if $Z$ encodes an independent set for the graph encoded by $X$ is

$$I(X, Z) = \bigwedge_{i,j=1}^n \{\neg X_{ij} \lor \neg (Z_i \land Z_j)\}.$$ 

The idea here is that if both $i$ and $j$ are in the putative independent set, so that $Z_i$ and $Z_j$ are both true, then we require that $(i, j)$ is not an edge, so that $\neg X_{ij}$ is true.

But note that we have ignored the size $k$ of the independent set, which is an input to the problem. Can you figure out how to modify the construction to account for $k$?