Week 12 Continuous Assessment

1803086

May 13, 2021

1. The number of subsets with $n-2$ elements of C is given by the binomial cooefficient. $\binom{n}{2}$, which equates to $\frac{1}{2}(n-1)n$. We can therefore, run the existing $2^n \cdot n^2$ TSP algorithm as such:

$$TSP(C\ c_i), i = 1, \ldots, \binom{n}{2}$$

Where each c_i is a 2 element subset of C.

We then take the minimum of these $\frac{1}{2}(n-1)n$ values, in $\frac{1}{2}(n-1)n$ time. We therefore run our $2^n \cdot n^2$ algorithm $\frac{1}{2}(n-1)n$ times, giving an runtime complexity of $2^n \cdot \frac{1}{2}(n-1)n^3$ before taking the minimum, in $\frac{1}{2}(n-1)$ time, giving an overall runtime of $2^n \cdot \frac{n^4}{4}(n-1)^2$ which is in $2^n \cdot n^{O(1)}$.

As for this algorithm's correctness, we can assume the result of each call to the predefined TSP function is correct. Each one of these results will be the shortest route which visits **all** the nodes in the (n-2) subset provided to it. Therefore after running all $\binom{n}{2}$ instances, we are left with an array of the shortest distances visiting $n-2$ nodes, therefore the smallest among them is the shortest overall path through the graph which visits **exactly** $n-2$ nodes.

2.
$$d(x, y, z) = 1 \iff x = y = 1 \text{ or } x = z = 0$$

(a)

$$\begin{aligned}
d(x, y, z) =& (x \wedge y) \vee (\neg x \wedge \neg z) \\
&\equiv \neg(\neg(x \wedge y) \wedge \neg(\neg x \wedge \neg z)) \\
&\equiv \neg((\neg x \vee \neg y) \wedge (x \vee z)) \\
&\equiv \neg((\neg x \wedge x) \vee (\neg x \wedge z) \vee (\neg y \wedge x) \vee (\neg y \wedge z)) \\
&\equiv \neg(\neg x \wedge z) \wedge \neg(\neg y \wedge x) \wedge \neg(\neg y \wedge z) \\
&\equiv (x \vee z) \wedge (y \vee x) \wedge (y \vee \neg z)
\end{aligned}$$

(b) i.

$$
\begin{aligned}
x \vee y \iff & \\
& d(0, \quad d(\\
& \qquad d(\neg x, \neg y, d(x,x,x)), \\
& \qquad d(\neg x, \neg y, d(x,x,x)), \\
& \qquad d(\neg x, \neg y, d(x,x,x)) \\
& \qquad) \\
& , d(\neg x, \neg y, d(x,x,x)))
\end{aligned}
$$

Where:

- $\neg x = d(0, d(x,x,x), x)$
- $\neg y = d(0, d(y,y,y), y)$

ii. $x \wedge y \iff d(x, y, d(x,x,x))$

(c) By the change-of-basis theorem

Let the circuits Ω_1 defined using the operators $\{\neg, \wedge, \vee\}$ and let the circuits defined over $\{d, 0\}$ be the set Ω_2

for each circuit $c_i \in \Omega_1$ let there be an equivalent circuit $c_i' \in \Omega_2$ of size s_i and depth l_i which computes c_i. Also let $s = \max_i s_i$ and $l = \max_i l_i$.

Given a circuit $C \in \Omega_1$, we can construct a logically equivalent circuit $C' \in \Omega_2$ by replacing all non-source nodes labelled with c_i by the circuit c_i'.

With $\Omega_1 = \{\neg, \vee, \wedge\}$ and $\Omega_2 = \{0, d\}$ we can show that every Ω_1 circuit C can be polynomially transformed to a Ω_2 circuit C'.

for the operator $\neg$, we have seen in (b) that this can be transformed into $d(0, d(x,x,x), x)$. This node can therefore be replaced with a circuit of depth 6.

The operator $\vee$ has been shown to be equivalent to the Ω_2 circuit of depth 9.

The $\wedge$ operator has been shown to be equivalent to a Ω_2 circuit of depth 6.

Therefore we can see that in this case $l = 9$ and so the depth of C' is at most $9 \cdot \texttt{depth}(C)$ which is in $O(\texttt{depth}(C))$

3. (a) Show that if $\bigvee L_j = 0$ for some $j = 1, \ldots, m$ then $f(\vec{x}, \vec{y}, \vec{z}) = 0$

For any set of literals L_j, for $\bigvee L_j$ to equal 0, all literals in L_j must also equal 0.

If $\bigvee L_j = 0$ then no conjunction of the remaining $L_i, i = 1, \ldots, m, i \neq j$, can now satisfy $\bigwedge_{j=1}^{m} \bigvee L_j$ as this would take the form:

$$
\left(\bigwedge_{j=1, j\neq i}^{m} \bigvee L_j\right) \wedge 0
$$

Which cannot be satisfied.

The fact that this is said to be a **minimal** CNF, implies that there are no dual literals in any set of literals L_j, therefore, we do not have any set of literals L_j constructed purely from these dual literals. This allows us to assume that if

(b) In order for $f(\vec{x}, \vec{y}, \vec{z})$ to be false, one or more of (x_i, y_i, z_i) must be false for each $i = 1, \ldots, n$.

This is the case as for the disjunction over $i = 1, \ldots, n$ of $(x_i \wedge y_i \wedge z_i)$ to be false all $(x_i \wedge y_i \wedge z_i)$ must be false. For this to be the case in each set of literals, grouped by i, one or more must be false in order to falsify the conjunction.

4. In complexity theory, one strives to find the most optimal algorithms to solve a problem.

   Whilst we have not yet the means to necessarily find these, we can posit as to whether algorithms within various complexity classes exist for such problems.

   Current theories regarding complexity classes and how they might relate to one another, (for example NP-Hard, NP and P) give us broad upper and or lower bounds for complexity but for some problems these bounds are still too generous. The paper gives as an example the potential for a $n^{O(\log n)}$ time algorithm for 3SAT, which under current theories regarding NP-Hardness, is possible, though thought to be extremely unlikely.

   The Exponential Time Hypothesis (ETH) is proposed as a means by which these classes can be *tightened* around certain problems. Allowing us to rule out the possibility or show the potential for algorithms solving hard problems in a certain complexity.

   The core of the ETH states that "There is a positive real s s.t. 3-CNF-Sat with parameter n cannot be solved in time $2^{sn}(n+m)^{O(1)}$" and as such if we can reduce a problem to 3-CNF-Sat in such a way that takes subexponential time, we can also say that there is no subexponential time algorithm to solve it.

   The paper uses two methods of reduction which satisfy this requirement: many-to-one, Fixed Paramter Tractable and Turing reduction.