CS60-454/554
Design and Analysis of Algorithms
Winter 2017

Assignment 4

Due Date: April 4, 2017, before lecture

The following rules apply to all assignments handed out in this course.

- For every algorithm you present, you *must* include:
  1. a description of the idea underlying the algorithm, possibly with examples to show how it works,
  2. the algorithm in a pseudo-code,
  3. a correctness proof of the algorithm,
  4. an analysis of the time complexity of the algorithm.
- Type your solutions if your hand-writing is *not* legible. The marks you will receive depends not only on the correctness or efficiency of the algorithm, but also the presentation.

Decision Problem Π:

Given a CNF logical expression, C, in which every clause contains exactly three literals. Is that a way of assigning truth values to the variables in C so that C is evaluated to the `true` value with every clause in it having *exactly one literal* being assigned the `true` value?

Prove that Π is `NP-complete` by proving $\Pi \in$ `NP` and `3SAT` $\propto \Pi$.

[**Hint:** Let $(x \vee y \vee z)$ be a clause in a problem instance of `3SAT`. Construct a CNF logical expression of Π which consists of the three clauses $(x \vee a \vee b), (y \vee c \vee d)$ and $(z \vee e \vee f)$ and six other clauses each of which contains exactly two of a, b, c, d, e and f.]

Solution:

(i) $\Pi \in$ `NP`:

nondeterministic Algorithm `3SAT-1in3`;

Input: A CNF logical expression C of `3SAT` with variable set U;

Output: $\begin{cases} Yes, & \text{if } C \text{ is satisfiable with every clause containing exactly one } \textit{true} \text{ literal;} \\ No, & \text{otherwise.} \end{cases}$

begin

1. Write down a guess which is a truth assignment $t : U \to \{true, false\}$;
2. for each literal $u_{j_i}, 1 \leq j \leq m, 1 \leq i \leq 3$, in C, scan the guess string t to look for $t(u_{j_i})$ and replace u_{j_i} with $t(u_{j_i})$.

3. **for** each c_j in C **do**

    if (c_j does not contain exactly one *true*) **then return** "*No*"; **stop**;

4. Evaluate C;

   if ($C \equiv true$) **then** report ("*Yes*")

   else report ("*No*");

end.

Since the length of the guess t is proportional to $|U|$, Step 1 takes $O(n)$ time as $|U| = n$. Since there are $3m$ literals and finding the value of each literal takes $O(n)$ time, Step 2 takes $O(mn)$ time. Since there are m clauses to check and each clause contains 3 literals, Step 3 takes $O(m)$ time. Evaluating C involves $2m$ '$\vee$' operators and $m-1$ '$\wedge$' operators and as each of these logical operations takes $O(1)$ time to evaluate, The evaluation of C thus takes $O(m)$ time. Reporting the result takes $O(1)$.

Hence, the non-deterministic algorithm takes $O(n)+O(mn)+O(m)+O(m)+O(1) = O(mn)$ time which is polynomial.

Thus, $\Pi \in NP$. $\square$

(ii) `3SAT` $\propto \Pi$:

Let $C = \bigwedge_{j=1}^{m} c_j$ be a problem instance of `3SAT`.

We are to create a polynomial transformation that transforms C into a problem instance $\tilde{C} = \bigwedge_{j=1}^{m} \tilde{c}_j$ of Π.

Since $C = \bigwedge_{j=1}^{m} c_j$ is satisfiable if and only if $c_j, 1 \le j \le m$, is satisfiable, it suffices to consider transforming each $c_j, 1 \le j \le m$, to $\tilde{c}_j$.

Let $c_j = (x \vee y \vee z)$.

Since c_j is *true* implies that there can be *more than one* of x, y and z having the *true* value, we must put the three literals in different clauses in $\tilde{c}_j$ as we can have exactly one literal having the *true* value in every clause in $\tilde{c}_j$. This means that in $\tilde{c}_j$, we must have:

$$(x \vee a \vee b) \wedge (y \vee c \vee d) \wedge (z \vee e \vee f).$$

Naturally, we have to relate these three clauses. So, we introduce the clauses:

$$(a \vee c \vee e) \wedge (b \vee d \vee f),$$

resulting in $\tilde{c}_j \equiv (x \vee a \vee b) \wedge (y \vee c \vee d) \wedge (z \vee e \vee f) \wedge (a \vee c \vee e) \wedge (b \vee d \vee f)$.

Unfortunately, this transformation is only one-sided! Specifically, we have $\tilde{C}$ is satisfiable implies C is satisfiable but C is satisfiable does not imply $\tilde{C}$ is satisfiable (verify this yourself).

To make the transformation two-sided, we noticed that we have to break $(a \vee c \vee e)$ into three pairs $(a \vee c)$ and $(c \vee e)$ and $(e \vee a)$ and add three new literals g, h, i to turn each pair into a clause having exactly there literals. Specifically, we replace $(a \vee c \vee e)$ with:

$$(a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i)$$

Likewise, we replace $(b \vee d \vee f)$ with:

$$(b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell)$$

We thus have $\tilde{c}_j = (x \vee a \vee b) \wedge (y \vee c \vee d) \wedge (z \vee e \vee f) \wedge (a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i)$
$\wedge\ (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell)$.

To create $\tilde{c}_j$ from c_j, we just have to generate 9 clause each with 3 literals. This can be done in $O(1)$ time. Hence, generating $\tilde{C}$ from C takes $O(m)$ time.

It remains to prove that C is satisfiable $\Leftrightarrow$ $\tilde{C}$ is satisfiable.

$\Rightarrow$) Suppose C is satisfiable.

Then there exists a truth assignment t for C such that $\bigwedge_{j=1}^{m} c_j \equiv true$.

Let $c_j \equiv (x \vee y \vee z)$.

Let $\tilde{t}(x) = t(x), \tilde{t}(y) = t(y), \tilde{t}(z) = t(z)$.

(a) $t(x) = t(y) = t(z) = true$:

Then $\tilde{t}(x) = \tilde{t}(y) = \tilde{t}(z) = true$.

$$
\begin{aligned}
\tilde{c}_j \equiv\ & (x \vee a \vee b) \wedge (y \vee c \vee d) \wedge (z \vee e \vee f) \wedge (a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
& \wedge (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell). \\
\equiv\ & (true \vee a \vee b) \wedge (true \vee c \vee d) \wedge (true \vee e \vee f) \\
& \wedge (a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
& \wedge (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell). \quad (\because \tilde{t}(x) = \tilde{t}(y) = \tilde{t}(z) = true) \\
\equiv\ & (true \vee false \vee false) \wedge (true \vee false \vee false) \wedge (true \vee false \vee false) \\
& \wedge (false \vee false \vee g) \wedge (false \vee false \vee h) \wedge (false \vee false \vee i) \\
& \wedge (false \vee false \vee j) \wedge (false \vee false \vee k) \wedge (false \vee false \vee \ell). \\
& (\because (true \vee a \vee b) \wedge (true \vee c \vee d) \wedge (true \vee e \vee f) \Rightarrow a \equiv b \equiv c \equiv d \equiv e \equiv f \equiv false) \\
\equiv\ & (true \vee false \vee false) \wedge (true \vee false \vee false) \wedge (true \vee false \vee false) \\
& \wedge (false \vee false \vee true) \wedge (false \vee false \vee true) \wedge (false \vee false \vee true) \\
& \wedge (false \vee false \vee true) \wedge (false \vee false \vee true) \wedge (false \vee false \vee true) \\
& \qquad (\text{let } g \equiv h \equiv i \equiv j \equiv k \equiv \ell \equiv true) \\
\equiv\ & true.
\end{aligned}
$$

Hence, $\tilde{t}(x) = \tilde{t}(y) = \tilde{t}(z) = \tilde{t}(g) = \tilde{t}(h) = \tilde{t}(i) = \tilde{t}(j) = \tilde{t}(k) = \tilde{t}(\ell) = true$, and
$\tilde{t}(a) = \tilde{t}(b) = \tilde{t}(c) = \tilde{t}(d) = \tilde{t}(e) = \tilde{t}(f) = false$,

is a truth assignment such that c_j is evaluated to the `true` value with every clause in it having *exactly one literal* being assigned the `true` value.

Thus, $\tilde{C}$ is satisfiable.

(b) Exactly two of x, y, z is *true*:

Owing to symmetricity, we consider the case $t(x) = t(y) = true, t(z) = false$ only (the other cases are similar).

Then $\tilde{t}(x) = \tilde{t}(y) = true, \tilde{t}(z) = false$.

$$
\begin{aligned}
\tilde{c}_j \equiv\ & (x \vee a \vee b) \wedge (y \vee c \vee d) \wedge (z \vee e \vee f) \wedge (a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
\equiv\ & (true \vee a \vee b) \wedge (true \vee c \vee d) \wedge (false \vee e \vee f)
\end{aligned}
$$

$$
\begin{aligned}
&\wedge(a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
&\wedge\ (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell). \qquad (\because \tilde{t}(x) = \tilde{t}(y) = true; \tilde{t}(z) = false) \\
\equiv\ &(true \vee false \vee false) \wedge (true \vee false \vee false) \wedge (false \vee e \vee f) \\
&\wedge(false \vee false \vee g) \wedge (false \vee e \vee h) \wedge (false \vee e \vee i) \\
&\wedge\ (false \vee false \vee j) \wedge (false \vee f \vee k) \wedge (false \vee f \vee \ell). \\
&\qquad\qquad (\because (true \vee a \vee b) \wedge (true \vee c \vee d) \Rightarrow a \equiv b \equiv c \equiv d \equiv false) \\
\equiv\ &(true \vee false \vee false) \wedge (true \vee false \vee false) \wedge (false \vee false \vee true) \\
&\wedge(false \vee false \vee g) \wedge (false \vee false \vee h) \wedge (false \vee false \vee i) \\
&\wedge\ (false \vee false \vee j) \wedge (false \vee true \vee k) \wedge (false \vee true \vee \ell). \\
&\qquad\qquad\qquad\qquad (\text{let } f \equiv true, e \equiv false) \\
\equiv\ &(true \vee false \vee false) \wedge (true \vee false \vee false) \wedge (false \vee false \vee true) \\
&\wedge(false \vee false \vee true) \wedge (false \vee false \vee true) \wedge (false \vee false \vee true) \\
&\wedge\ (false \vee false \vee true) \wedge (false \vee true \vee false) \wedge (false \vee true \vee false). \\
&\qquad\qquad\qquad (\text{let } g \equiv h \equiv i \equiv j \equiv true, k \equiv \ell \equiv false) \\
\equiv\ &true.
\end{aligned}
$$

Hence, $\tilde{t}(x) \equiv \tilde{t}(y) \equiv \tilde{t}(f) \equiv \tilde{t}(g) \equiv \tilde{t}(h) \equiv \tilde{t}(i) \equiv \tilde{t}(j) \equiv true$, and

$$\tilde{t}(z) \equiv \tilde{t}(a) \equiv \tilde{t}(b) \equiv \tilde{t}(c) \equiv \tilde{t}(d) \equiv \tilde{t}(e) \equiv \tilde{t}(k) \equiv \tilde{t}(\ell) \equiv false,$$

is a truth assignment such that c_j is evaluated to the `true` value with every clause in it having *exactly one literal* being assigned the `true` value.

Thus, $\tilde{C}$ is satisfiable.

(c) Exactly one of x, y, z is $true$:

Owing to symmetricity, we consider the case $t(x) = true, t(y) = t(z) = false$ only.

Then $\tilde{t}(x) = true, \tilde{t}(y) = \tilde{t}(z) = false$

$$
\begin{aligned}
\tilde{c}_j \equiv\ &(x \vee a \vee b) \wedge (y \vee c \vee d) \wedge (z \vee e \vee f) \wedge (a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
&\wedge\ (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell) \\
\equiv\ &(true \vee a \vee b) \wedge (false \vee c \vee d) \wedge (false \vee e \vee f) \\
&\wedge(a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
&\wedge\ (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell) \qquad (\because \tilde{t}(x) = true; \tilde{t}(y) = \tilde{t}(z) = false) \\
\equiv\ &(true \vee false \vee false) \wedge (false \vee c \vee d) \wedge (false \vee e \vee f) \\
&\wedge(false \vee c \vee g) \wedge (false \vee e \vee h) \wedge (c \vee e \vee i) \\
&\wedge\ (false \vee d \vee j) \wedge (false \vee f \vee k) \wedge (d \vee f \vee \ell)\ (\because (true \vee a \vee b) \Rightarrow a \equiv b \equiv false) \\
\equiv\ &(true \vee false \vee false) \wedge (false \vee c \vee d) \wedge (false \vee false \vee true) \\
&\wedge(false \vee c \vee g) \wedge (false \vee false \vee h) \wedge (c \vee false \vee i) \\
&\wedge\ (false \vee d \vee j) \wedge (false \vee true \vee k) \wedge (d \vee true \vee \ell) \qquad (\text{let } f \equiv true; e \equiv false) \\
\equiv\ &(true \vee false \vee false) \wedge (false \vee true \vee false) \wedge (false \vee false \vee true) \\
&\wedge(false \vee true \vee g) \wedge (false \vee false \vee h) \wedge (true \vee false \vee i)
\end{aligned}
$$

$$
\begin{aligned}
&\wedge (false \vee false \vee j) \wedge (false \vee true \vee k) \wedge (false \vee true \vee \ell) \qquad (\text{let } c \equiv true; d \equiv false) \\
&\equiv (true \vee false \vee false) \wedge (false \vee true \vee false) \wedge (false \vee false \vee true) \\
&\quad \wedge (false \vee true \vee false) \wedge (false \vee false \vee h) \wedge (true \vee false \vee false) \\
&\quad \wedge (false \vee false \vee j) \wedge (false \vee true \vee false) \wedge (false \vee true \vee false) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{let } g \equiv i \equiv k \equiv \ell \equiv false) \\
&\equiv (true \vee false \vee false) \wedge (false \vee true \vee false) \wedge (false \vee false \vee true) \\
&\quad \wedge (false \vee true \vee false) \wedge (false \vee false \vee true) \wedge (true \vee false \vee false) \\
&\quad \wedge (false \vee false \vee true) \wedge (false \vee true \vee false) \wedge (false \vee true \vee false) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (\text{let } h \equiv j \equiv true) \\
&\equiv true.
\end{aligned}
$$

Hence, $\tilde{t}(x) \equiv \tilde{t}(c) \equiv \tilde{t}(f) \equiv \tilde{t}(h) \equiv \tilde{t}(j) \equiv true$, and

$$\tilde{t}(y) \equiv \tilde{t}(z) \equiv \tilde{t}(a) \equiv \tilde{t}(b) \equiv \tilde{t}(e) \equiv \tilde{t}(e) \equiv \tilde{t}(d) \equiv \tilde{t}(g) \equiv \tilde{t}(i) \equiv \tilde{t}(k) \equiv \tilde{t}(\ell) \equiv false,$$

is a truth assignment such that c_j is evaluated to the `true` value with every clause in it having *exactly one literal* being assigned the `true` value.

Thus, $\tilde{C}$ is satisfiable.

$\Leftarrow$) Suppose $\tilde{C}$ is satisfiable.

Then there exists a truth assignment $\tilde{t}$ for $\tilde{C}$ such that $\bigwedge_{j=1}^{m} \tilde{c}_j \equiv true$.

For each $\tilde{c}_j, 1 \leq j \leq m$, since $c_j = (x, y, z)$, let $t(x) = \tilde{t}(x), t(y) = \tilde{t}(y), t(z) = \tilde{t}(z)$.

(a) $\tilde{t}(x) \equiv true$ or $\tilde{t}(y) \equiv true$:

Then $t(x) \equiv true$ or $t(y) \equiv true$

$$
\begin{aligned}
&\Rightarrow (x \vee y \vee z) \equiv (true \vee y \vee z) \equiv true, \text{ or} \\
&\quad\ (x \vee y \vee z) \equiv (x \vee true \vee z) \equiv true.
\end{aligned}
$$

Hence, any truth assignment t with $t(x) = true$ or $t(y) = true$ is a truth assignment such that c_j is evaluated to the `true` value with every clause in it having *exactly one literal* being assigned the `true` value.

(b) $\tilde{t}(x) = false$ and $\tilde{t}(y) = false$:

Then,

$$
\begin{aligned}
\tilde{c}_j &\equiv (x \vee a \vee b) \wedge (y \vee c \vee d) \wedge (z \vee e \vee f) \wedge (a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
&\quad \wedge (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell) \\
&\equiv (false \vee a \vee b) \wedge (false \vee c \vee d) \wedge (z \vee e \vee f) \\
&\quad \wedge (a \vee c \vee g) \wedge (a \vee e \vee h) \wedge (c \vee e \vee i) \\
&\quad \wedge (b \vee d \vee j) \wedge (b \vee f \vee k) \wedge (d \vee f \vee \ell) \qquad (\because \tilde{t}(x) = \tilde{t}(y) = false)
\end{aligned}
$$

Since every clause can have exactly one literal having the *true* value,

$(false \vee a \vee b) \wedge (false \vee c \vee d)$ implies that exactly one of a and b is true and exactly one of b and d is *true*.

Owing to symmetricity, we consider the case $\tilde{t}(a) = \tilde{t}(d) = true, \tilde{t}(b) = \tilde{t}(c) = false$. We then have:

$$\tilde{c}_j \equiv (false \vee true \vee false) \wedge (false \vee false \vee true) \wedge (z \vee e \vee f)$$
$$\wedge (true \vee false \vee g) \wedge (true \vee e \vee h) \wedge (false \vee e \vee i)$$
$$\wedge\ (false \vee true \vee j) \wedge (false \vee f \vee k) \wedge (true \vee f \vee \ell).$$

Since $(true \vee e \vee h) \Rightarrow e \equiv h \equiv false$, and

$(true \vee f \vee \ell) \Rightarrow f \equiv \ell \equiv false,$

we thus have $(z \vee e \vee f) \equiv (z \vee false \vee false) \Rightarrow \tilde{t}(z) = true.$

Hence, $t(z) = true \qquad (\because t(z) = \tilde{t}(z))$

$\Rightarrow (x \vee y \vee z) \equiv (false \vee false \vee true) \equiv true.$

$\Rightarrow c_j \equiv true$

Hence, $t(x) = t(y) = false$, and $t(z) = true$ is a truth assignment such that c_j is evaluated to the `true` value.

Thus, C is satisfiable. $\square$

Since $\Pi \in$ `NP` and `3SAT` $\propto \Pi$, Π is `NP-complete`. $\blacksquare$