CS 341: Algorithms
Module 8: Intractability and Undecidability

Armin Jamshidpey, Eugene Zima

Based on lecture notes by many previous CS 341 instructors

David R. Cheriton School of Computer Science, University of Waterloo

Spring 2019

Decision Problems

Decision Problem: Given a problem instance I, answer a certain question "yes" or "no".

Problem Instance: Input for the specified problem.

Problem Solution: Correct answer ("yes" or "no") for the specified problem instance. I is a yes-instance if the correct answer for the instance I is "yes". I is a no-instance if the correct answer for the instance I is "no".

Size of a problem instance: Size(I) is the number of bits required to specify (or encode) the instance I.

The Complexity Class P

Algorithm Solving a Decision Problem: An algorithm A is said to solve a decision problem Π provided that A finds the correct answer ("yes" or "no") for every instance I of Π in finite time.
Polynomial-time Algorithm: An algorithm A for a decision problem Π is said to be a polynomial-time algorithm provided that the complexity of A is $O(n^k)$, where k is a positive integer and $n = \mathit{Size}(\mathrm{I})$.
The Complexity Class P denotes the set of all decision problems that have polynomial-time algorithms solving them. We write $\Pi \in P$ if the decision problem Π is in the complexity class P.

Cycles in Graphs

Cycle Problem

Instance: An undirected graph $G = (V, E)$.
Question: Does G contain a cycle?

Hamiltonian Cycle Problem

Instance: An undirected graph $G = (V, E)$.
Question: Does G contain a hamiltonian cycle?

A hamiltonian cycle is a cycle that passes through every vertex in V exactly once.

Knapsack Problems

0-1 Knapsack-Dec Problem

Instance: a list of profits, $P = [p_1, \cdots, p_n]$; a list of weights, $W = [w_1, \cdots, w_n]$; a capacity, M; and a target profit, T.
Question: Is there an n-tuple $[x_1, x_2, \cdots, x_n] \in \{0, 1\}^n$ such that $\sum w_i x_i \leq M$ and $\sum p_i x_i \geq T$?

Rational Knapsack-Dec Problem

Instance: a list of profits, $P = [p_1, \cdots, p_n]$; a list of weights, $W = [w_1, \cdots, w_n]$; a capacity, M; and a target profit, T.
Question: Is there an n-tuple $[x_1, x_2, \cdots, x_n] \in [0, 1]^n$ such that $\sum w_i x_i \leq M$ and $\sum p_i x_i \geq T$?

Polynomial-time Turing Reductions

Suppose Π_1 and Π_2 are problems (not necessarily decision problems). A (hypothetical) algorithm B to solve Π_2 is called an oracle for Π_2.

Suppose that A is an algorithm that solves Π_1, assuming the existence of an oracle B for Π_2. (B is used as a subroutine within the algorithm A.)

Then we say that A is a Turing reduction from Π_1 to Π_2, denoted $\Pi_1 \leq^T \Pi_2$.

A Turing reduction A is a polynomial-time Turing reduction if the running time of A is polynomial, under the assumption that the oracle B has unit cost running time.

If there is a polynomial-time Turing reduction from Π_1 to Π_2, we write $\Pi_1 \leq_P^T \Pi_2$

Informally: Existence of a polynomial-time Turing reduction means that if we can solve Π_2 in polynomial time, then we can solve Π_1 in polynomial time.

The two subset sum problems **SubS-D**, and **SubS-F** both have as input an array of integers $a_1, a_2, \ldots, a_n$ and an integer S (target value).

SubS-D is a decision problem: the question is whether or not there is subset of integers $a_1, a_2, \ldots, a_n$ that sums to S.

SubS-F does not have yes/no answers. The answer to **SubS-F** is the set of integers $i_1, i_2, \ldots, i_k$, $1 \le i_1 < i_2 < \cdots < i_k \le n$, such that $a[i_1] + a[i_2] + \cdots + a[i_k] = S$.

These problems are polynomial-time Turing equivalent.

SubS-D $\leq_P^T$ **SubS-F** is trivial...
SubS-F $\leq_P^T$ **SubS-D**:

```
SubS-F(a,S) {
  if SubS-D(a,S) {
    for i = 1 to n do
      if SubS-D(a\a[i],S)
         a = a \ a[i]
    od
  return a
  }
  else FAIL
}
```

The number of steps in this algorithm is equal to the length of the given array a. Each step requires 1 call to **SubS-D**.

Consider the following variations of the CLIQUE problem (clique in a graph is a set of vertices such that each pair of vertices in this set is connected by an edge):

- **CLIQUE-D**
  Input: a graph $G = (V, E)$, and an integer k.
  Output: TRUE, if graph G has a cliques of size $\geq k$, and FALSE otherwise.
- **CLIQUE-O**
  Input: a graph $G = (V, E)$. Output: integer n – size of the largest clique in graph G.
- **CLIQUE-F**
  Input: a graph $G = (V, E)$. Output: subset V' – vertices that form largest clique in graph G.

They are all polynomial time Turing reducible to each-other.

CLIQUE-D $\leq_P^T$ **CLIQUE-O** is trivial:

```
CLIQUE-D(V,E,k) {
  n = CLIQUE-O(V,E);
  return (k <= n);
}
```

CLIQUE-O $\leq_P^T$ **CLIQUE-D**:

```
CLIQUE-O(V,E) {
  n = |V|;
  while not CLIQUE-D(V,E,n) do
    n--;
  od
  return n;
}
```

CLIQUE-F $\leq_P^T$ CLIQUE-D:

```
CLIQUE-F(V,E) {
  n = |V|;
  while not CLIQUE-D(V,E,n) do
    n--;
  od
  for i = 1 to |V| do
    if CLIQUE-D(V\{V[i]},E',n)
\\ E' is the set of edges after removing V[i]
      then V=V\{V[i]}; E = E';
    fi
  od
  return V;
}
```

Travelling Salesperson Problems

Problem: TSP-Optimization

Instance: A graph G and edge weights $w : E \to \mathbb{Z}^+$
Find: A hamiltonian cycle H in G such that $w(H) = \sum_{e \in H} w(e)$ is minimized.

Problem: TSP-Optimal Value

Instance: A graph G and edge weights $w : E \to \mathbb{Z}^+$
Find: The minimum T such that there exists a hamiltonian cycle H in G with $w(H) = T$.

Problem: TSP-Decision

Instance: A graph G and edge weights $w : E \to \mathbb{Z}^+$ and a target T.
Question: Does there exist a hamiltonian cycle H in G with $w(H) \leq T$?

TSP-Optimal Value $\leq_P^T$ TSP-Dec

Algorithm 1: TSP-OptimalValue-Solver (G, w)

external TSP-Dec-Solver
$hi \leftarrow \sum_{e \in E} w(e)$
$lo \leftarrow 0$
if not TSP-Dec-Solver(G, w, hi) **then return** (∞)
while $hi > lo$ **do**
    $mid \leftarrow \lfloor \frac{hi+lo}{2} \rfloor$
    if TSP-Dec-Solver(G, w, mid) **then**
        $hi \leftarrow mid$
    else $lo \leftarrow mid + 1$
return(hi)

This is a standard binary search technique.

TSP-Optimization $\leq_P^T$ TSP-Dec

Algorithm 2: TSP-Optimization-Solver($G = (V, E), w$)

external TSP-OptimalValue-Solver, TSP-Dec-Solver
$T^* \leftarrow$ TSP-OptimalValue-Solver(G, w)
if $T^* = \infty$ **then return** ("no hamiltonian cycle exists")
$w_0 \leftarrow w$
$H \leftarrow \emptyset$
for all $e \in E$ **do**
  $w_0[e] \leftarrow \infty$
  if not TSP-Dec-Solver(G, w_0, T^*) **then**
    $w_0[e] \leftarrow w[e]$
    $H \leftarrow H \cup \{e\}$
return(H)

Proof of Correctness

Clearly H contains a hamiltonian cycle of minimum weight T^* at the end of the algorithm (note that H just consists of the edges that are not deleted from G). We claim that H is precisely a hamiltonian cycle.

Suppose not; then $C \cup \{e\} \subseteq H$, where C is a hamiltonian cycle of weight T^* and $e \in G \setminus C$. Consider the iteration when e was added to H. Let G' denote the graph G at this point in time. G' contains a hamiltonian cycle of weight T^* but $G' \setminus \{e\}$ does not, so e is included in H. We are assuming that $C \cup \{e\} \subseteq H$, which implies

$$C \subseteq H \setminus \{e\}\,.$$

Since $H \subseteq G'$, we have

$$C \subseteq H \setminus \{e\} \subseteq G' \setminus \{e\}\,.$$

Therefore e would not have been added to H, which is a contradiction.