Carlos Mc Gregor c.mcgregormuro@mail.utoronto.ca

CSC373S: Algorithm Design, Analysis & Complexity

Lecture 31

Friday March 24, 2017
based on notes by Denis Pankratov

Many several classes of *complexity problems*! Check *Complexity Zoo* at `https://complexityzoo.uwaterloo.ca/Complexity_Zoo`

co-NP Problems

Definition: $L \subseteq \{0,1\}^*$, the complement of L is $\hat{L} = \{0,1\}^* \setminus L$

Definition: co-$NP = \{L | \hat{L} \in NP\}$
co-NP = class of decision problems such that NO-instances can be verified efficiently

$L_{3-SAT} \in NP$
Certificate assignment to variables of formula.
Verifier: evaluates formula on the assignment

Certificate for NO-instances, unsatisfied formulas.
All possible assignments $\rightarrow$ exponential length $2^n, n =$ number of variables.

No known short & easily verifiable certificate for NO-instances of L_{3-SAT}
L_{3-SAT} is not known to be in co-NP

Exercise: $P \subseteq NP \cap$ co-NP
Figure 1: Photo

Definition: $L \in$ co-NPC (co-NP complete) if:

1. $L \in$ co-NP
2. $\forall \tilde{L} \in$ coNP, $\tilde{L} \leq_p L$

Definition: Let $L \in NP$ the corresponding search problem is to find the certificate that makes the verifier output 1.

Example: K-clique search verifier
Input: $G = (V, E) -$undirected graph $k \in \mathbb{Z}_{>0}$
Output: $S \subseteq V : S - k$-clique if it exists
NIL otherwise

Definition: $L \subseteq \{0,1\}^*$
Let A be an oracle (black-box) for L: $(\forall x) x \in L \Leftrightarrow A(x) = 1$

L is self-reducible if the corresponding search problem can be solved in polytime using oracle A (access to the oracle costs 1 unit of time)

Example: 3-SAT search verifier
Input: φ-3-CNF w/ n variables, m clauses
Output: $x \in \{T, F\}^n$ so that $\varphi(x) = T$ if there is such x_i
NIL otherwise

Claim: 3-SAT is self reducible
Proof: Let A be an oracle for L_{3-SAT}
Restrict (φ, i, v)

```
/* plugs x_i = v into φ, simplifies it, returns a smaller formula */

φ' ← ∅
for each clause C ∈ φ:
   if C does not contain x_i or ¬x_i:
      φ' ← φ' ∪ {C}
   if C is not satisfied by x_i = v, let l be the literal com to x_i
      φ' ← φ' ∪ {C \ l}
return φ'
```

3-SAT**-**$SEARCH(\varphi)$**:**

```
if A(φ) = 0:
   return NIL
for i = 0 to n
   φ_T ← Restrict (φ, i, T)
   φ_F ← Restrict (φ, i, F)
if A(φ_T) = 1:        set x_i = T
   φ ← φ_T
else:
   set x_i = F
   φ ← φ_F
return x
```

LI: in iteration i the newly obtained φ is botained from the original φ by setting values of $x_1, .., x_i$ as in the previous steps & newly derived formula is satisfiable.
Proof: exercise Termination: all variables $x_1, .., x_n$ are assigned $\Rightarrow \varphi = \varnothing \Leftarrow$ the assignment satisfies the original φ
Runtime: Restrict runs in $\mathcal{O}(m), m$ being the number of clauses

Overall $\mathcal{O}(nm)$
If our impl. of oracle A runs in $T(n, m)$, then the runtime is $\mathcal{O}(n(T(n, m) + m))$

Claim: k-clique is self reducible

Proof: Let A be an oracle for dec. ver.

```
def K-CLIQUE-SEARCH(G, k):
    if A(G,k) = 0:
        return None
```