ECE 595, Section 10
Numerical Simulations
Lecture 4: **NP**-hardness

Prof. Peter Bermel

January 14, 2013

Outline

- Recap from Friday
- Class NP
- Non-deterministic Turing machines
- Reducibility
- Cook-Levin theorem
- Coping with NP Hardness

Recap from Friday

- Church-Turing thesis used to demonstrate demonstrate several models of computation:
  - Turing machines
  - Register machines
  - Lambda calculus
- Big-oh notation: DTIME(n2)
- Polynomial complexity: $\mathrm{P} = \cup_{c \geq 1} \mathbf{DTIME}(n^c)$,

Examples in **P**

- CONNECTED – the set of all connected graphs
- TRIANGLEFREE – the set of all graphs without a triangle
- BIPARTITE – the set of all bipartite (distinct) graphs
- TREE – the set of all trees

Class **NP** Definition

- Decision problems that can be efficiently *verified*
- The language L subset {0,1}* is in **NP** if there exists a polynomial p: N → N and a polynomial time TM M s.t. for every x:
  - $x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)}$
- If $x \in L$ and $u \in \{0,1\}^{p(|x|)}$ satisfies *M*(*x*,*u*)=1, then u is a **certificate** for x

Examples of **NP**

- Independent set – find a *k*-size subset of a given graph *G*'s vertices
- Traveling salesman – given a set of *n* nodes and pairwise distances, find a closed circuit that visits every node once of length no more than *k*
- Linear/integer programming – given a set of m linear inequalities for n variables, find a set of rational/integer solutions
- Graph isomorphism – given two nxn adjacency matrices M_1 and M_2, decide if they define the same graph
- Composite numbers – decide if number *N* is prime
- Factoring – for a number *N*, find a factor *M* in the interval [*L*,*U*]

Relationship of **P** & **NP**

- **P** is a subset of **NP**
- Formally, $\mathrm{P} \subseteq \mathrm{NP} \subseteq \cup_{c \geq 1} \mathbf{DTIME}\left(2^{n^c}\right)$
- Recall **DTIME**(T) is computable in cT(n) time

Relationship of **P** & **NP**

- Key open question: does **P**=**NP**?
- Some **NP** problems are in **P**:
  - Connectivity – breadth-first search
  - Composite numbers – Agrawal, Kayal, and Saxena solved this a few years ago
  - Linear programming – ellipsoidal algorithm of Khachiyan

Relationship of **P** & **NP**

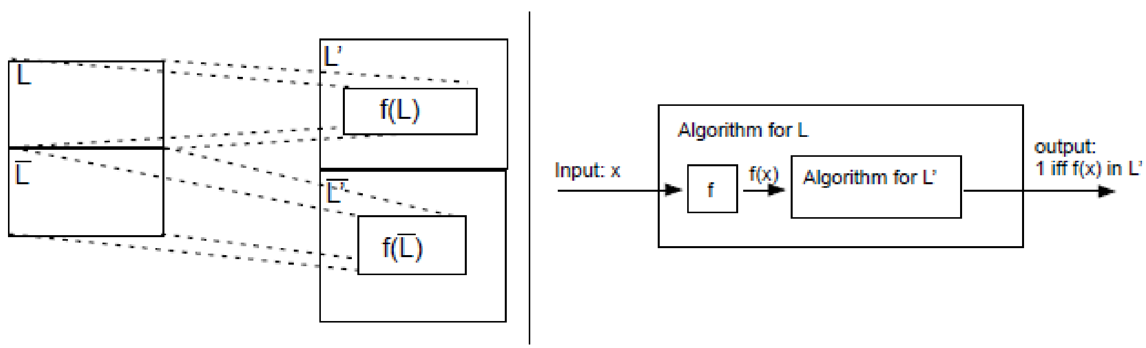
- Problems *not* shown to be in **P**:
  - Independent set
  - Traveling salesman
  - Subset sum
  - Integer programming
- This general group is known as **NP** complete: i.e., not in **P** unless **P**=**NP**

Non-deterministic Turing machines

- Has two transition functions δ_0 and δ_1
- The NDTM has another internal state, q_{accept}
- M outputs 1 if some (non-deterministic) sequence of choices → q_{accept}
- Can also define **NTIME** s.t. NDTM takes cT(n)-time $\forall x \in L \Leftrightarrow M(x) = 1$
- Alternative definition: $\mathrm{NP} = \bigcup_{c\in\mathbb{N}} \mathbf{NTIME}(n^c)$

Reducibility

- Karp reductions relate the various example problems to one another
- Language *A* reduces to *B* if there is a polynomial-time computable function *f,* s.t. $\forall x \in \{0,1\}^*$, $x \in A$ iff $f(x) \in B$
- B is **NP**-hard if $A \leq_p B,\ \forall A \in \mathrm{NP}$



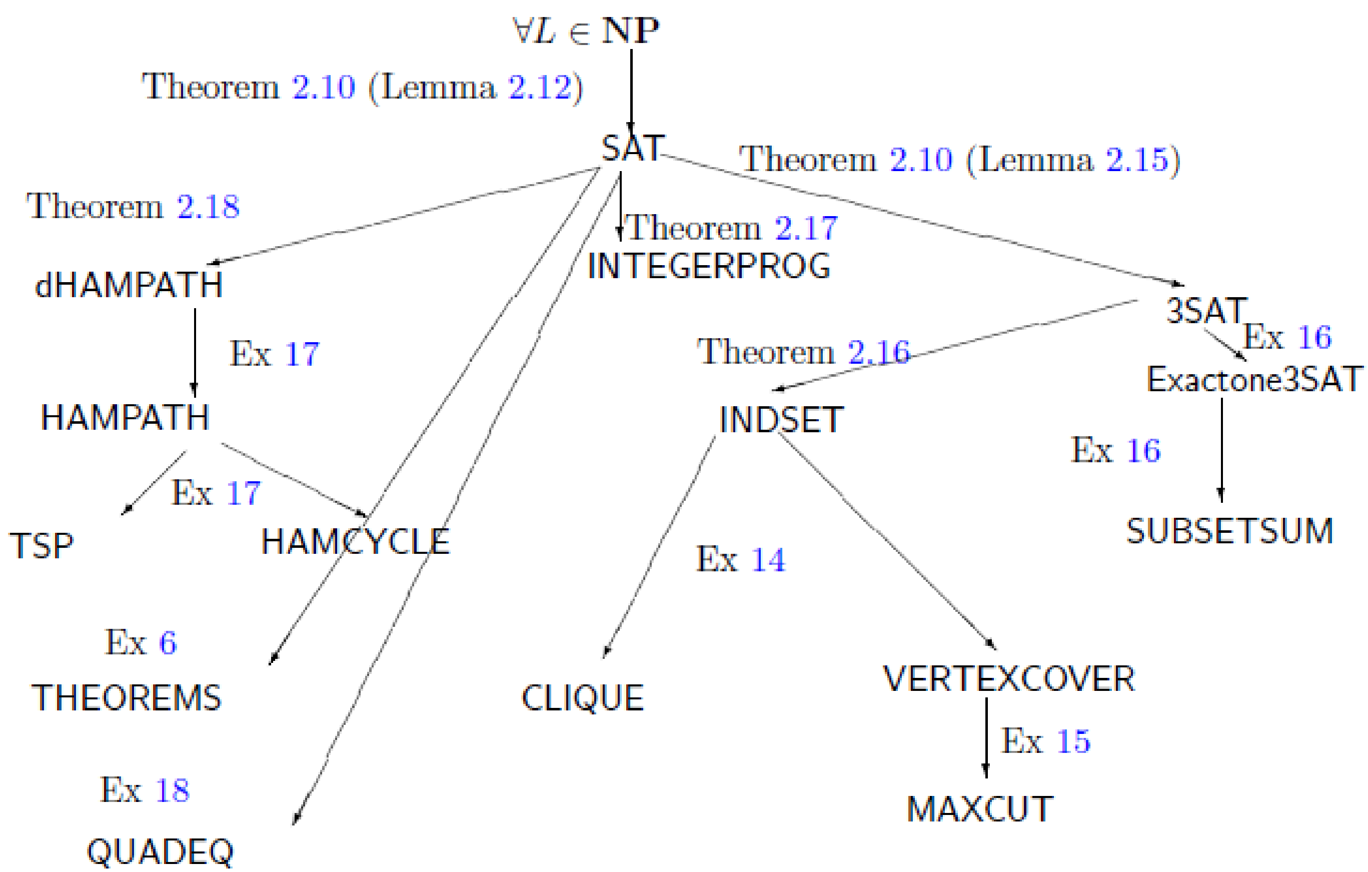
NP completeness

- By transitivity of Karp reduction, there should be a language that's "hardest," known as **NP**-complete
- The TMSAT language is **NP**-complete
  - $TMSAT = \{\langle \alpha, x, 1^n, 1^t \rangle : \exists u \in \{0,1\}^n\}$
  - M_α denotes TM represented by α
  - M_α outputs 1 on $\langle x, u \rangle$

Cook-Levin theorem

- CNF: conjunctive normal form used for a Boolean formula consisting of AND's and OR's
- kCNF: CNF consisting of k AND's: $\bigwedge_{i=1}^{k} \left(\bigvee_j v_{ij} \right)$
- Cook-Levin Theorem:
  - Let SAT be the language of all satisfiable CNF formulae: SAT is NP-complete
  - Let 3SAT be the language of all satisfiable 3CNF formulae: 3SAT is NP-complete

Web of reductions



Coping with **NP**-hardness

- Approximate/heuristic solutions
- Example: traveling salesman
- Key considerations:
  - Maximum error
  - Average-case complexity

Related Complexity Classes

- coNP – $x \in L \Leftrightarrow \forall u \in \{0,1\}^{p(|x|)}$ s.t. M(x,u)=1
- EXP = $\cup_{c \geq 0}$ $\mathbf{DTIME}\left(2^{n^c}\right)$
- NEXP= $\cup_{c \geq 0}$ $\mathbf{NTIME}\left(2^{n^c}\right)$
- Must have $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{EXP} \subseteq \mathrm{NEXP}$

What if **P=NP**?

- Any decision (or equivalently, search) problem could be solved in a reasonable amount of time
- Perfectly optimal designs for all areas of engineering could be chosen
- Automated theory generation for experimental results
- No privacy in the digital domain

Next Class

- Discussion of solving linear algebraic equations
- Please read Chapter 2 of “Numerical Recipes” by W.H. Press *et al*.