CS21
Decidability and Tractability

Lecture 23

March 3, 2014

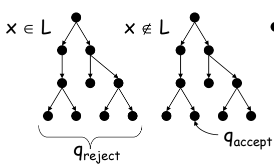
Outline

- the class co-NP
- the class NP ∩ coNP
- the class PSPACE
  - a PSPACE-complete problem
  - PSPACE and 2-player games

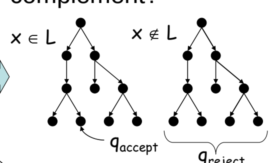
coNP

- Is NP closed under complement?

Can we transform this machine:



into this machine?



coNP

- language L is in coNP iff its complement (co-L) is in NP
- it is believed that NP ≠ coNP
- note: P = NP implies NP = coNP
  - proving NP ≠ coNP would prove P ≠ NP
  - another major open problem…

coNP

- canonical coNP-complete language:

  UNSAT = {φ : φ is an unsatisfiable 3-CNF formula}
  - proof?

coNP

Disjunctive Normal Form = OR of ANDs

- another example

  3-DNF-TAUTOLOGY = {φ : φ is a 3-DNF formula and for all x, φ(x) =1}
  - proof?
- another example:

  EQUIV-CIRCUIT = {(C_1, C_2) : C_1 and C_2 are Boolean circuits and for all x, C_1(x) = C_2(x)}
  - proof?

Quantifier characterization of coNP

- recall that a language L is in NP if and only if it is expressible as:

  $L = \{x \mid \exists y, |y| \leq |x|^k, (x, y) \in R\}$

  where R is a language in P.

Theorem: language L is in coNP if and only if it is expressible as:

  $L = \{x \mid \forall y, |y| \leq |x|^k, (x, y) \in R\}$

  where R is a language in P.

Proof interpretation of coNP

- What is a proof?
- Good formalization comes from NP:

  $L = \{x \mid \exists y, |y| \leq |x|^k, (x, y) \in R\}$, and $R \in P$

  "proof" "short" proof "proof verifier"

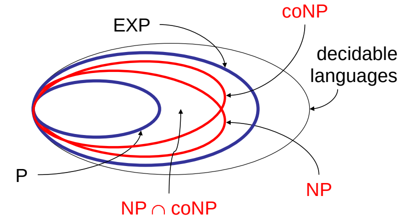
- NP languages have short proofs of membership
- co-NP languages have short proofs of non-membership
- coNP-complete languages are least likely to have short proofs of membership

coNP

- what complexity class do the following languages belong in?
  - COMPOSITES = {x : integer x is a composite}
  - PRIMES = {x : integer x is a prime number}
  - GRAPH-NONISOMORPHISM = {(G, H) : G and H are graphs that are not isomorphic}
  - EXPANSION = {(G = (V,E), $\alpha > 0$): every subset $S \subset V$ of size at most |V|/2 has at least $\alpha|S|$ neighbors}

coNP

- Picture of the way we believe things are:



NP ∩ coNP

- Might guess NP ∩ coNP = P by analogy with RE (since RE ∩ coRE = DECIDABLE)
- Not believed to be true.
- A problem in NP ∩ coNP not believed to be in P:

  L = {(x, k): integer x has a prime factor p < k}

  (decision version of factoring)

NP ∩ coNP

- **Theorem**: This language is in NP ∩ coNP:

  L = {(x, k): integer x has a prime factor p < k}

Proof:
- In NP (why?)
- In coNP (what certificate demonstrates that x has *no* small prime factor?)
- Use this claim: PRIMES is in NP:

  PRIMES = $\{x : \forall\, 1 < y < x, y \text{ does not divide } x\}$

PRIMES in NP

Theorem: (Pratt 1975) PRIMES is in NP.

PRIMES = {x : ∀ 1 < y < x, y does not divide x}

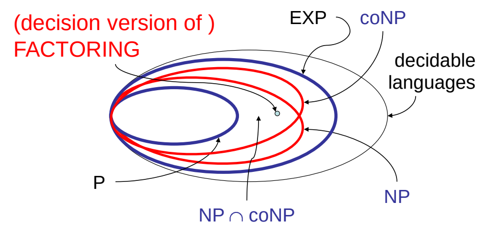
- Proof outline:
  - Step 1: give "∃" characterization of PRIMES
  - Step 2: this ⇒ short certificate of primality
  - Step 3: certificate checkable in poly time

(we will skip, because…)

Theorem: (M. Agrawal, N. Kayal, N. Saxena 2002)
PRIMES is in P.

Summary

- Picture of the way we believe things are:



Space complexity

Definition: the space complexity of a TM M is a function

$$f: \mathbf{N} \rightarrow \mathbf{N}$$

where f(n) is the maximum number of tape cells M scans on any input of length n.

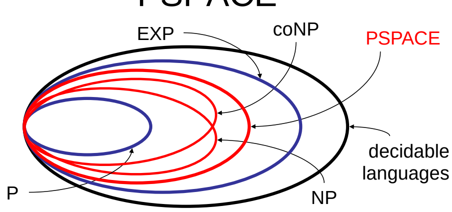
- "M uses space f(n)," "M is a f(n) space TM"

Space complexity

Definition: SPACE(t(n)) = {L : there exists a TM M that decides L in space O(t(n))}

$$\text{PSPACE} = \cup_{k \geq 1} \text{SPACE}(n^k)$$

PSPACE



- NP ⊂ PSPACE, coNP ⊂ PSPACE (proof?)
- PSPACE ⊂ EXP (proof?)
- containments believed to be proper