coNP problems

- **NP** is the class of problems with succinct certificates.
- **coNP** is the class of problems with succinct disqualifications.
- VALIDITY:
  INSTANCE: a Boolean expression ϕ
  QUESTION: is ϕ valid?
- VALIDITY is in **coNP**: for a expression ϕ which is not valid, a falsifying truth assignment is a succinct disqualification.
- HAMILTON PATH COMPLEMENT and SAT COMPLEMENT are also in **coNP**.
- $\mathbf{P} \subseteq \mathbf{coNP}$

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

coNP-completeness

- HAMILTON PATH COMPLEMENT and VALIDITY are **coNP**-complete:

  Every language $L \in \mathbf{coNP}$ is reducible to VALIDITY, because $\overline{L} \in \mathbf{NP}$ and, hence, there is a reduction R from $\overline{L}$ to SAT such that for every string x, $x \in \overline{L}$ iff $R(x) \in$ SAT but then $x \in L$ iff $R(x) \notin$ SAT iff $\neg R(x) \in$ VALIDITY.

  Similarly for HAMILTON PATH COMPLEMENT.

Proposition. If L is **NP**-complete, then its complement $\overline{L} = \Sigma^* - L$ is **coNP**-complete.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

coNP and NP

- It is open whether $\mathbf{NP} = \mathbf{coNP}$.
- If $\mathbf{P} = \mathbf{NP}$, then $\mathbf{NP} = \mathbf{coNP}$.
- It is possible that $\mathbf{P} \neq \mathbf{NP}$ but $\mathbf{NP} = \mathbf{coNP}$ (however, it is strongly believed that $\mathbf{NP} \neq \mathbf{coNP}$).
- The problems in **coNP** that are **coNP**-complete are the least likely to be in **P** and also in **NP** (see below).

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

coNP and NP—cont'd

Proposition. If a **coNP**-complete problem is in **NP**, $\mathbf{NP} = \mathbf{coNP}$.

Proof. Let L be a **coNP**-complete problem that is in **NP**. Consider $L' \in \mathbf{coNP}$. Then there is a reduction R from L' to L. Then $L' \in \mathbf{NP}$, because L' can be decided by a polynomial time NTM which on input x computes first $R(x)$ and then starts the NTM for L. Thus $\mathbf{coNP} \subseteq \mathbf{NP}$.

Consider $L' \in \mathbf{NP}$. Then $\overline{L'} \in \mathbf{coNP}$ and there is a reduction R from $\overline{L'}$ to L. Then $\overline{L'} \in \mathbf{NP}$ and hence $L' \in \mathbf{coNP}$. Thus $\mathbf{NP} \subseteq \mathbf{coNP}$.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

PRIMES

- PRIMES:
  INSTANCE: an integer N in binary
  QUESTION: Is N a prime number?
- PRIMES $\in$ **coNP** (disqualification: divisor).
- Note that a $O(\sqrt{N})$ algorithm for PRIMES testing all relevant divisor candidates is only pseudopolynomial.
- PRIMES $\in$ **NP** (to be shown)
- August 2002:
  M. Agrawal, N. Kayal, N. Saxena: *PRIMES is in* **P** !!

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

The class coNP ∩ NP

- Problems in **coNP** ∩ **NP** have succinct certificates and disqualifications.
- **P** ⊆ **coNP** ∩ **NP**
- Certain kind of optimization problems are in **coNP** ∩ **NP**:
  if two problems in **NP** are *dual*: each is *reducible to the complement* of the other, then both are in **coNP** ∩ **NP**.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

The class coNP ∩ NP—cont'd

Example.
MAX FLOW(D): does a given network N have a flow from s to t of at least K.
MIN CUT(D): Given a network and a budget B, is there are set of edges of capacity of at most B such that deleting these disconnects s from t.

Now by max flow–min cut theorem, N *has* a flow of value at least K iff it *does not have* a cut of capacity $K-1$ or less.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

PRIMES—cont'd

A succinct certificate for primality can be obtained using the following theorem.

Theorem. A number $p > 1$ is prime iff there is a number $1 < r < p$ such that $r^{p-1} = 1 \bmod p$ and, furthermore, $r^{\frac{p-1}{q}} \neq 1 \bmod p$ for all prime divisors q of $p-1$.

Corollary. PRIMES is in **NP** ∩ **coNP**.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

PRIMES—cont'd

Proof.

- The succinct certificate $C(p)$ for primality of p is $C(p) = (r; q_1, C(q_1), \ldots, q_k, C(q_k))$ where $C(q_i)$ is a primality certificate for q_i.
- The certificate is polynomial in the length of p (in $\log p$).
- The certificate can be checked by division and exponentiation.
- Ordinary multiplication and division doable in polynomial time in the length of the input (in binary).
- Exponentiation r^{p-1} can be done in polynomial time by starting with repeated squaring $r^2, r^4, \ldots r^{2^l}$.
- Hence, the certificate can be checked in polynomial time.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Function problems

- We have studied decision problems but many problems in practice require a more complicated answer than "yes" / "no".

  Example. Find a satisfying truth assignment for a formula. Compute an optimal tour for TSP.
- Such problems are called *function problems*.
- Decision problems are useful surrogates of function problems only in the context of *negative complexity results*.

  Example. SAT and TSP(D) are **NP**-complete. Then unless $\mathbf{P} = \mathbf{NP}$, there is no polynomial time algorithm for finding a satisfying truth assignment or an optimal tour.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Function problems—cont'd

- FSAT: given a Boolean expression ϕ, if ϕ is satisfiable then return a satisfying truth assignment of ϕ otherwise return "no".
- If FSAT can solved in polynomial time, then so can SAT.
- If SAT can solved in polynomial time, then so can FSAT using the following algorithm given ϕ with variables $x_1, \ldots, x_n$ ($\phi[x := \mathbf{true}]$ denotes ϕ where variable x is replaced by **true**):

  if $\phi \notin$ SAT then return "no";
  for all $x \in \{x_1, \ldots, x_n\}$ do
  if $\phi[x := \mathbf{true}] \in$ SAT then $T(x) := \mathbf{true}$; $\phi := \phi[x := \mathbf{true}]$
  else $T(x) := \mathbf{false}$; $\phi := \phi[x := \mathbf{false}]$;
  return T;

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Function problems—cont'd

- If TSP can solved in polynomial time, then so can TSP(D).
- If TSP(D) can solved in polynomial time, then so can TSP.
- An optimal tour can be found using an algorithm which
  1. finds the cost C of an optimal tour by binary search (with a polynomial number of calls to the polynomial time TSP(D));
  2. finds an optimal tour using C (with a polynomial number of calls to the polynomial time TSP(D)).

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

A TSP algorithm using a TSP(D) algorithm as a subroutine:

/*Find the cost C of an optimal tour by binary search*/
$C := 0$; $C_u := 2^n$;
while $(C_u > C)$ do
  if there is a tour of cost $\lfloor (C_u + C)/2 \rfloor$ or less then
    $C_u := \lfloor (C_u + C)/2 \rfloor$
  else $C := \lfloor (C_u + C)/2 \rfloor + 1$;
/* Find an optimal tour */
For all intercity distances do
  set the distance to $C + 1$;
  if there is a tour of cost C or less, freeze the distance to $C + 1$
  else restore the original distance and add it to the tour;
endfor

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Relationship between decision and function problems

- Let L be a language in **NP** having a polynomial time decidable and polynomially balanced relation R_L such that for all strings x: there is a string y with $R_L(x,y)$ iff $x \in L$.
- The function problem associated with L, denoted by FL: Given x, find a string y such that $R_L(x,y)$ if such a string y exists; if no such string exists, return "no".
- The class of all function problems associated as above with languages in **NP** is called **FNP**.
- **FP** is the subclass of **FNP** solvable in polynomial time.
- FSAT is in **FNP** and FHORNSAT is in **FP** (but it is open whether TSP is in **FNP**).

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Reductions of function problems

- A function problem A *reduces* to a function problem B if there are string functions R, S computable in logarithmic space such that for all strings x, z: if x is an instance of A, then $R(x)$ is an instance of B and if z is a correct output of $R(x)$, then $S(z)$ is a correct output of x.
- A problem A is *complete* for a class F$\mathcal{C}$ of function problems if it is in F$\mathcal{C}$ and every problem in F$\mathcal{C}$ reduces to A.
- **FP** and **FNP** are closed under reductions and reductions among function problems compose.
- FSAT is **FNP**-complete.
- **FP** $=$ **FNP** iff **P** $=$ **NP**.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Total functions

- There are certain important problems in **FNP** that are guaranteed to never return "no".
- FACTORING: Given an integer N, find its prime decomposition $N = p_1^{k_1} \cdots p_m^{k_m}$ (No known polynomial time algorithm).
- FACTORING seems to be different from the other hard problems in **FNP**: it is a total function.
- A problem L in **FNP** is called *total* if for every string x there is at least one string y such that $R_L(x,y)$.
- The subclass of **FNP** containing all total function problems is denoted by **TFNP**.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Total functions—cont'd

There are also other problems in **TFNP** with no known polynomial time algorithm.

Example. HAPPYNET:
INSTANCE: an undirected graph (V,E) with integer weights w on edges.
QUESTION: find a state of the graph where all nodes are happy.

- A state is a mapping $S : V \mapsto \{-1,+1\}$
- A node i is happy in S if $S(i)\sum_{[i,j]\in E} S(j)w[i,j] \geq 0$
- Every graph has a happy state:
  A happy state can be found using the algorithm:
  Start with any S and while there is an unhappy node, flip it.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

HAPPYNET—cont'd

- This algorithm in not polynomial but pseudopolynomial $\mathrm{O}(W)$ where W is the sum of all weights.
- No polynomial algorithm known.
- HAPPYNET equivalent with finding stable states in neural networks in the Hopfield model.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio

Total functions—cont'd

- ANOTHER HAMILTON CYCLE is **FNP**-complete.
- ANOTHER HAMILTON CYCLE for cubic graphs is in **TFNP**.
- EQUAL SUMS:
  Given n positive integers $a_1,\ldots,a_n$ such that $\sum_{i=1}^{n} a_i < 2^n - 1$, find two different subsets that have the same sum.
- EQUAL SUMS in **TFNP**.

© 2004 Teknillinen korkeakoulu, Tietojenkäsittelyteorian laboratorio