CS38
Introduction to Algorithms

Lecture 16

May 22, 2014

Outline

- Linear programming
  - LP duality
  - ellipsoid algorithm

  * slides from Kevin Wayne

- coping with intractibility
  - NP-completeness

LP Duality

Primal problem.

$$\begin{array}{lllllll} (P) & \max & 13A & + & 23B & & \\ & \text{s.t.} & 5A & + & 15B & \le & 480 \\ & & 4A & + & 4B & \le & 160 \\ & & 35A & + & 20B & \le & 1190 \\ & & A & , & B & \ge & 0 \end{array}$$

Idea. Add nonnegative combination (C, H, M) of the constraints s.t.

$$\begin{array}{lll} 13A + 23B & \le & (5C + 4H + 35M)\ A \ +\ (15C + 4H + 20M)\ B \\ & \le & 480C + 160H + 1190M \end{array}$$

Dual problem. Find best such upper bound.

$$\begin{array}{lllllllll} (D) & \min & 480C & + & 160H & + & 1190M & & \\ & \text{s.t.} & 5C & + & 4H & + & 35M & \ge & 13 \\ & & 15C & + & 4H & + & 20M & \ge & 23 \\ & & C & , & H & , & M & \ge & 0 \end{array}$$

LP Duals

Canonical form.

$$\begin{array}{llll} (P) & \max\ c^T x & & \\ & \text{s.t.}\ \ Ax & \le & b \\ & x & \ge & 0 \end{array} \qquad \begin{array}{llll} (D) & \min\ y^T b & & \\ & \text{s.t.}\ \ A^T y & \ge & c \\ & y & \ge & 0 \end{array}$$

Double Dual

Canonical form.

$$\begin{array}{llll} (P) & \max\ c^T x & & \\ & \text{s.t.}\ \ Ax & \le & b \\ & x & \ge & 0 \end{array} \qquad \begin{array}{llll} (D) & \min\ y^T b & & \\ & \text{s.t.}\ \ A^T y & \ge & c \\ & y & \ge & 0 \end{array}$$

Property. The dual of the dual is the primal.

Pf. Rewrite (D) as a maximization problem in canonical form; take dual.

$$\begin{array}{llll} (D') & \max\ -y^T b & & \\ & \text{s.t.}\ \ -A^T y & \le & -c \\ & y & \ge & 0 \end{array} \qquad \begin{array}{llll} (DD) & \min\ -c^T z & & \\ & \text{s.t.}\ \ -(A^T)^T z & \ge & -b \\ & z & \ge & 0 \end{array}$$

Taking Duals

LP dual recipe.

Primal (P)	maximize	minimize	Dual (D)
constraints	$a x = b_i$ $a x \le b$ $a x \ge b_i$	y_i unrestricted $y_i \ge 0$ $y_i \le 0$	variables
variables	$x_j \le 0$ $x_j \ge 0$ unrestricted	$a^T y \ge c_j$ $a^T y \le c_j$ $a^T y = c_j$	constraints

Pf. Rewrite LP in standard form and take dual.

Strong duality

LP Strong Duality

Theorem. [Gale-Kuhn-Tucker 1951, Dantzig-von Neumann 1947]
For $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, if (P) and (D) are nonempty, then max = min.

$$(P)\quad \max c^T x \quad \text{s. t.} \quad Ax \le b,\ x \ge 0$$

$$(D)\quad \min y^T b \quad \text{s. t.} \quad A^T y \ge c,\ y \ge 0$$

Generalizes:
- Dilworth's theorem.
- König-Egervary theorem.
- Max-flow min-cut theorem.
- von Neumann's minimax theorem.
- …

Pf. [ahead]

LP Weak Duality

Theorem. For $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, if (P) and (D) are nonempty, then max $\le$ min.

$$(P)\quad \max c^T x \quad \text{s. t.} \quad Ax \le b,\ x \ge 0$$

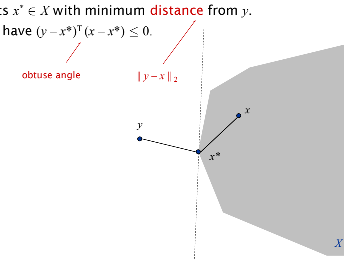
$$(D)\quad \min y^T b \quad \text{s. t.} \quad A^T y \ge c,\ y \ge 0$$

Pf. Suppose $x \in \mathbb{R}^m$ is feasible for (P) and $y \in \mathbb{R}^n$ is feasible for (D).
- $y \ge 0,\ A x \le b \quad \Rightarrow \quad y^T A x \le y^T b$
- $x \ge 0,\ A^T y \ge c \quad \Rightarrow \quad y^T A x \ge c^T x$
- Combine: $c^T x \le y^T A x \le y^T b$.

Projection Lemma

Weierstrass' theorem. Let X be a compact set, and let $f(x)$ be a continuous function on X. Then min $\{ f(x) : x \in X \}$ exists.

Projection lemma. Let $X \subset \mathbb{R}^m$ be a nonempty closed convex set, and take y not in X. Then there exists $x^* \in X$ with minimum distance from y. Moreover, for all $x \in X$ we have $(y - x^*)^T (x - x^*) \le 0$.

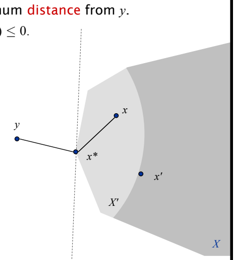


Projection Lemma

Weierstrass' theorem. Let X be a compact set, and let $f(x)$ be a continuous function on X. Then min $\{ f(x) : x \in X \}$ exists.

Projection lemma. Let $X \subset \mathbb{R}^m$ be a nonempty closed convex set, and take y not in X. Then there exists $x^* \in X$ with minimum distance from y. Moreover, for all $x \in X$ we have $(y - x^*)^T (x - x^*) \le 0$.

Pf.
- Define $f(x) = \| y - x \|$.
- Want to apply Weierstrass, but X not necessarily bounded.
- X not empty $\Rightarrow$ there exists $x' \in X$.
- Define $X' = \{ x \in X : \| y - x \| \le \| y - x' \| \}$ so that X' is closed, bounded, and min $\{ f(x) : x \in X \}$ = min $\{ f(x) : x \in X' \}$.
- By Weierstrass, min exists.



Projection Lemma

Weierstrass' theorem. Let X be a compact set, and let $f(x)$ be a continuous function on X. Then min $\{ f(x) : x \in X \}$ exists.

Projection lemma. Let $X \subset \mathbb{R}^m$ be a nonempty closed convex set, and take y not in X. Then there exists $x^* \in X$ with minimum distance from y. Moreover, for all $x \in X$ we have $(y - x^*)^T (x - x^*) \le 0$.

Pf.
- x^* min distance $\Rightarrow$ $\| y - x^* \|^2 \le \| y - x \|^2$ for all $x \in X$.
- By convexity: if $x \in X$, then $x^* + \epsilon (x - x^*) \in X$ for all $0 < \epsilon < 1$.
- $\| y - x^* \|^2 \le \| y - x^* - \epsilon (x - x^*) \|^2$
  $= \| y - x^* \|^2 + \epsilon^2 \| (x - x^*) \|^2 - 2\epsilon (y - x^*)^T (x - x^*)$
- Thus, $(y - x^*)^T (x - x^*) \le \frac{1}{2} \epsilon \| (x - x^*) \|^2$.
- Letting $\epsilon \to 0^+$, we obtain the desired result.

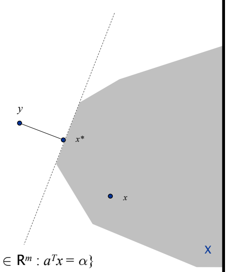
Separating Hyperplane Theorem

Theorem. Let $X \subset \mathrm{R}^m$ be a nonempty closed convex set, and take y not in X. Then there exists a hyperplane $H = \{ x \in \mathrm{R}^m : a^T x = \alpha \}$ where $a \in \mathrm{R}^m$, $\alpha \in \mathrm{R}$ that separates y from X.

$a^T x \geq \alpha$ for all $x \in X$
$a^T y < \alpha$

Pf.

- Let x^* be closest point in X to y.
- By projection lemma, $(y - x^*)^T (x - x^*) \leq 0$ for all $x \in X$
- Choose $a = x^* - y$ not equal 0 and $\alpha = a^T x^*$.
- If $x \in X$, then $a^T(x - x^*) \geq 0$; thus $\Rightarrow a^T x \geq a^T x^* = \alpha$.
- Also, $a^T y = a^T (x^* - a) = \alpha - \|a\|^2 < \alpha$ ▪



13

Farkas' Lemma

Theorem. For $A \in \mathrm{R}^{m \times n}$, $b \in \mathrm{R}^m$ exactly one of the following two systems holds:

$$\text{(I)} \quad \exists x \in \Re^n \quad \text{s.t.} \quad \begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned}$$

$$\text{(II)} \quad \exists y \in \Re^m \quad \text{s.t.} \quad \begin{aligned} A^T y &\geq 0 \\ y^T b &< 0 \end{aligned}$$

Pf. [not both] Suppose x satisfies (I) and y satisfies (II). Then $0 > y^T b = y^T A x \geq 0$, a contradiction.

Pf. [at least one] Suppose (I) infeasible. We will show (II) feasible.

- Consider $S = \{ Ax : x \geq 0 \}$ and note that b not in S.
- Let $y \in \mathrm{R}^m$, $\alpha \in \mathrm{R}$ be a hyperplane that separates b from S: $y^T b < \alpha$, $y^T s \geq \alpha$ for all $s \in S$.
- $0 \in S \Rightarrow \alpha \leq 0 \Rightarrow y^T b < 0$
- $y^T A x \geq \alpha$ for all $x \geq 0 \Rightarrow y^T A \geq 0$ since x can be arbitrarily large.

14

Another Theorem of the Alternative

Corollary. For $A \in \mathrm{R}^{m \times n}$, $b \in \mathrm{R}^m$ exactly one of the following two systems holds:

$$\text{(I)} \quad \exists x \in \Re^n \quad \text{s.t.} \quad \begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

$$\text{(II)} \quad \exists y \in \Re^m \quad \text{s.t.} \quad \begin{aligned} A^T y &\geq 0 \\ y^T b &< 0 \\ y &\geq 0 \end{aligned}$$

Pf. Apply Farkas' lemma to:

$$\text{(I')} \quad \exists x \in \Re^n, s \in \Re^m \quad \text{s.t.} \quad \begin{aligned} Ax + Is &= b \\ x, s &\geq 0 \end{aligned}$$

$$\text{(II')} \quad \exists y \in \Re^m \quad \text{s.t.} \quad \begin{aligned} A^T y &\geq 0 \\ I y &\geq 0 \\ y^T b &< 0 \end{aligned}$$

15

LP Strong Duality

Theorem. [strong duality] For $A \in \mathrm{R}^{m \times n}$, $b \in \mathrm{R}^m$, $c \in \mathrm{R}^n$, if (P) and (D) are nonempty then max = min.

$$\text{(P)} \quad \max c^T x \quad \text{s.t.} \quad \begin{aligned} Ax &\leq b \\ x &\geq 0 \end{aligned}$$

$$\text{(D)} \quad \min y^T b \quad \text{s.t.} \quad \begin{aligned} A^T y &\geq c \\ y &\geq 0 \end{aligned}$$

Pf. [max $\leq$ min] Weak LP duality.
Pf. [min $\leq$ max] Suppose max < α. We show min < α.

$$\text{(I)} \quad \exists x \in \Re^n \quad \text{s.t.} \quad \begin{aligned} Ax &\leq b \\ -c^T x &\leq -\alpha \\ x &\geq 0 \end{aligned}$$

$$\text{(II)} \quad \exists y \in \Re^m, z \in \Re \quad \text{s.t.} \quad \begin{aligned} A^T y - cz &\geq 0 \\ y^T b - \alpha z &< 0 \\ y, z &\geq 0 \end{aligned}$$

- By definition of α, (I) infeasible $\Rightarrow$ (II) feasible by Farkas' Corollary.

16

LP Strong Duality

$$\text{(II)} \quad \exists y \in \Re^m, z \in \Re \quad \text{s.t.} \quad \begin{aligned} A^T y - cz &\geq 0 \\ y^T b - \alpha z &< 0 \\ y, z &\geq 0 \end{aligned}$$

Let y, z be a solution to (II).

Case 1. $[z = 0]$

- Then, $\{ y \in \mathrm{R}^m : A^T y \geq 0, y^T b < 0, y \geq 0 \}$ is feasible.
- Farkas Corollary $\Rightarrow$ $\{ x \in \mathrm{R}^n : Ax \leq b, x \geq 0 \}$ is infeasible.
- Contradiction since by assumption (P) is nonempty.

Case 2. $[z > 0]$

- Scale y, z so that y satisfies (II) and $z = 1$.
- Resulting y feasible to (D) and $y^T b < \alpha$.

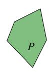
17

Ellipsoid algorithm

May 22, 2014 CS38 Lecture 16 18

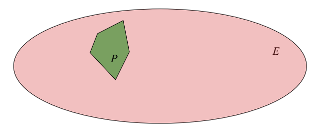
Geometric Divide-and-Conquer

To find a point in P:



Geometric Divide-and-Conquer

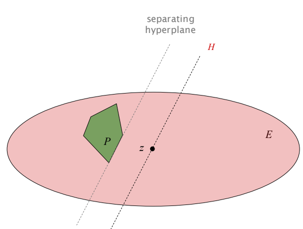
To find a point in P:
- Maintain ellipsoid E containing P.



Geometric Divide-and-Conquer

To find a point in P:
- Maintain ellipsoid E containing P.
- If center of ellipsoid z is in P stop;
  otherwise find hyperplane separating z from P.

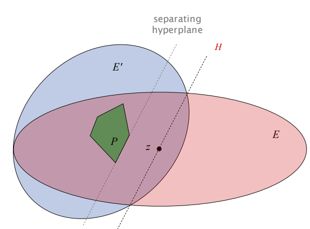
and consider corresponding half-ellipsoid $\frac{1}{2} E = E \cap H$



Geometric Divide-and-Conquer

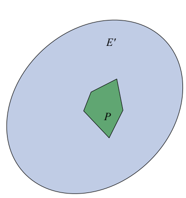
To find a point in P:
- Maintain ellipsoid E containing P.
- If center of ellipsoid z is in P stop;
  otherwise find hyperplane separating z from P.
- Find smallest ellipsoid E' containing half-ellipsoid.

L-J ellipsoid



Geometric Divide-and-Conquer

To find a point in P:
- Maintain ellipsoid E containing P.
- If center of ellipsoid z is in P stop;
  otherwise find hyperplane separating z from P.
- Find smallest ellipsoid E' containing half-ellipsoid.
- Repeat.



Optimization to Feasibility

Standard form.

$$\begin{array}{lrcl} \max c^T x & & & \\ \text{s. t.} & Ax & = & b \\ & x & \geq & 0 \end{array}$$

$Ax \leq b$ form.

$$\begin{array}{llrcl} \exists\ x, y & & & & \\ & \text{s. t.} & Ax & \leq & b \\ & & -Ax & \leq & -b \\ & & -x & \leq & 0 \\ & & A^T y & \leq & c \\ & & c^T x - b^T y & \leq & 0 \end{array}$$

$Ax \leq b$
$x \geq 0$
dual feasible
optimal

Ellipsoid Algorithm

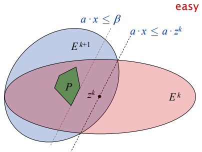
Goal. Given $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, find $x \in \mathbb{R}^n$ such that $\underbrace{Ax \le b}_{P}$.

Ellipsoid algorithm.

- Let E_0 be an ellipsoid containing P.
- $k = 0$.
- While center z^k of ellipsoid E^k is not in P:
  - find a constraint, say $a \cdot x \le \beta$, that is violated by z^k
  - let E^{k+1} be min volume ellipsoid containing $\underbrace{E^k \cap \{ x : a \cdot x \le a \cdot z^k \}}_{\text{half-ellipsoid } \frac{1}{2} E}$
  - $k = k + 1$

enumerate constraints

easy to compute



Shrinking Lemma

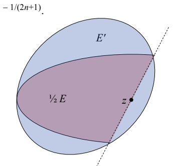
Ellipsoid. Given $D \in \mathbb{R}^{n \times n}$ positive definite and $z \in \mathbb{R}^n$, then

$$E = \{ x \in \Re^n : (x - z)^T D^{-1} (x - z) \le 1 \}$$

is an ellipsoid centered on z with $\text{vol}(E) = \sqrt{\det(D)} \times vol(B(0, 1))$

unit sphere

Key lemma. Every half-ellipsoid $\frac{1}{2} E$ is contained in an ellipsoid E' with $\text{vol}(E') / \text{vol}(E) \le e^{-1/(2n+1)}$.



Shrinking Lemma: Unit Sphere

Special case. E = unit sphere, $H = \{ x : x_1 \ge 0 \}$.

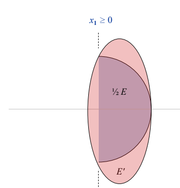
$$E = \{ x : \sum_{i=1}^{n} (x_i)^2 \le 1 \} \qquad E' = \{ x : \left(\tfrac{n+1}{n}\right)^2 (x_1 - \tfrac{1}{n+1})^2 + \tfrac{n^2-1}{n^2} \sum_{i=2}^{n} (x_i)^2 \le 1 \}$$

Claim. E' is an ellipsoid containing $\frac{1}{2} E = E \cap H$.

Pf. If $x \in \frac{1}{2} E$:

$$\begin{aligned}
&\left(\frac{n+1}{n}\right)^2 \left(x_1 - \frac{1}{n+1}\right)^2 + \frac{n^2-1}{n^2} \sum_{i=2}^{n} x_i^2 \\
&= \frac{n^2+2n+1}{n^2} x_1^2 - \left(\frac{n+1}{n}\right)^2 \frac{2x_1}{n+1} + \frac{1}{n^2} + \frac{n^2-1}{n^2} \sum_{i=2}^{n} x_i^2 \\
&= \frac{2n+2}{n^2} x_1^2 - \frac{2n+2}{n^2} x_1 + \frac{1}{n^2} + \frac{n^2-1}{n^2} \sum_{i=1}^{n} x_i^2 \\
&= \frac{2n+2}{n^2} x_1 (x_1 - 1) + \frac{1}{n^2} + \frac{n^2-1}{n^2} \sum_{i=1}^{n} x_i^2 \\
&\le 0 + \frac{1}{n^2} + \frac{n^2-1}{n^2} \\
&= 1
\end{aligned}$$

$0 \le x_1 \le 1$ $\quad$ $\sum x_i^2 \le 1$



Shrinking Lemma: Unit Sphere

Special case. E = unit sphere, $H = \{ x : x_1 \ge 0 \}$.

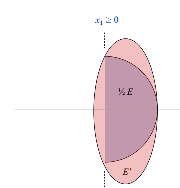
$$E = \{ x : \sum_{i=1}^{n} (x_i)^2 \le 1 \} \qquad E' = \{ x : \left(\tfrac{n+1}{n}\right)^2 (x_1 - \tfrac{1}{n+1})^2 + \tfrac{n^2-1}{n^2} \sum_{i=2}^{n} (x_i)^2 \le 1 \}$$

Claim. E' is an ellipsoid containing $\frac{1}{2} E = E \cap H$.

Pf. Volume of ellipsoid is proportional to side lengths:

$$\begin{aligned}
\frac{vol(E')}{vol(E)} &= \left(\frac{n^2}{n^2-1}\right)^{\frac{n-1}{2}} \left(\frac{n}{n+1}\right) \\
&= \left(1 + \frac{1}{n^2-1}\right)^{\frac{n-1}{2}} \left(1 - \frac{1}{n+1}\right) \\
&\le e^{\frac{1}{n^2-1} \frac{n-1}{2}} \; e^{\frac{-1}{n+1}} \\
&= e^{\frac{-1}{2(n+1)}}
\end{aligned}$$

$1 + x \le e^x$



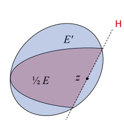
Shrinking Lemma

Shrinking lemma. The min volume ellipsoid containing the half-ellipsoid $\frac{1}{2} E = E \cap \{ x : a \cdot x \le a \cdot z \}$ is defined by:

$$z' = z - \frac{1}{n+1} \frac{Da}{\sqrt{a^T D a}}, \quad D' = \frac{n^2}{n^2-1} \left(D - \frac{2}{n+1} \frac{Daa^T D}{a^T D a} \right)$$

$$E' = \{ x \in \Re^n : (x - z')^T (D')^{-1} (x - z') \le 1 \}$$

Moreover, $\text{vol}(E') / \text{vol}(E) < e^{-1/(2n+1)}$.



Pf sketch.

- We proved E = unit sphere, $H = \{ x : x_1 \ge 0 \}$
- Ellipsoids are affine transformations of unit spheres.
- Volume ratios are preserved under affine transformations.

Shrinking Lemma

Shrinking lemma. The min volume ellipsoid containing the half-ellipsoid $\frac{1}{2} E = E \cap \{ x : a \cdot x \le a \cdot z \}$ is defined by:

$$z' = z - \frac{1}{n+1} \frac{Da}{\sqrt{a^T D a}}, \quad D' = \frac{n^2}{n^2-1} \left(D - \frac{2}{n+1} \frac{Daa^T D}{a^T D a} \right)$$

$$E' = \{ x \in \Re^n : (x - z')^T (D')^{-1} (x - z') \le 1 \}$$

Moreover, $\text{vol}(E') / \text{vol}(E) < e^{-1/(2n+1)}$.

Corollary. Ellipsoid algorithm terminates after at most $2(n+1) \ln(\text{vol}(E_0) / \text{vol}(P))$ steps.

Ellipsoid Algorithm

Theorem. Linear Programming problems can be solved in polynomial time.

Pf sketch.
- Shrinking lemma.
- Set initial ellipsoid E_0 so that $\text{vol}(E_0) \le 2^{cnL}$.
- Perturb $Ax \le b$ to $Ax \le b + \varepsilon \Rightarrow$ either P is empty or $\text{vol}(P) \ge 2^{-cnL}$.
- Bit complexity (to deal with square roots).
- Purify to vertex solution.

Caveat. This is a theoretical result. Do not implement.

$O(mn^3 L)$ arithmetic ops on numbers of size $O(L)$, where L = number of bits to encode input

Coping with intractability

Decision problems + languages

- A problem is a function:

  $f:\Sigma^* \to \Sigma^*$
- Simple. Can we make it simpler?
- Yes. Decision problems:

  $f:\Sigma^* \to \{\text{accept, reject}\}$
- Does this still capture our notion of problem, or is it too restrictive?

Decision problems + languages

- Example: factoring:
  - given an integer m, find its prime factors

    $f_{factor}: \{0,1\}^* \to \{0,1\}^*$
- Decision version:
  - given 2 integers m,k, accept iff m has a prime factor p < k
- Can use one to solve the other and vice versa. True in general.

Decision problems + languages

- For most complexity settings a problem is a decision problem:

  $f:\Sigma^* \to \{\text{accept, reject}\}$
- Equivalent notion: language

  $L \subseteq \Sigma^*$

  the set of strings that map to "accept"
- Example: L = set of pairs (m,k) for which m has a prime factor p < k

Search vs. Decision

- Definition: given a graph G = (V, E), an independent set in G is a subset V' ⊆ V such that for all u,w ∈ V' (u,w) ∉ E
- A problem:

  given G, find the largest independent set
- This is called a search problem
  - searching for *optimal* object of some type
  - comes up frequently

Search vs. Decision

- We want to talk about languages (or decision problems)
- Most search problems have a natural, related decision problem by adding a bound "k"; for example:
  - search problem: given G, find the largest independent set
  - decision problem: given (G, k), is there an independent set of size *at least* k

The class NP

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

$$P = \cup_{k \geq 1} \text{TIME}(n^k)$$

Definition: NTIME(t(n)) = {L : there exists a NTM M that decides L in time O(t(n))}

$$NP = \cup_{k \geq 1} \text{NTIME}(n^k)$$

Poly-time verifiers

- NP = {L : L decided by poly-time NTM}
- Very useful alternate definition of NP:

Theorem: 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.

- poly-time TM M_R deciding R is a "verifier"

Poly-time verifiers

- NP = {L : L decide "witness" or "certificate" e NTM}
- Very useful alternate definition efficiently verifiable

Theorem: language L is in NP i f it is expressible as:

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

where R is a language in P.

- poly-time TM M_R deciding R is a "verifier"

Poly-time verifiers

- Example: 3SAT expressible as

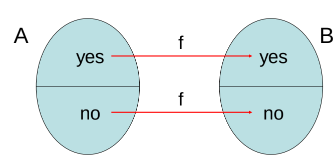
3SAT = {φ : φ is a 3-CNF formula for which ∃ assignment A for which (φ, A) ∈ R}

R = {(φ, A) : A is a sat. assign. for φ}

- satisfying assignment A is a "witness" of the satisfiability of φ (it "certifies" satisfiability of φ)
- R is decidable in poly-time

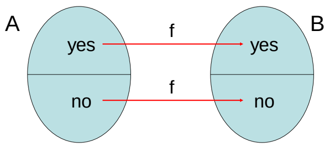
Poly-time reductions

- Type of reduction we will use:
  - "many-one" poly-time reduction



reduction from language A to language B

Poly-time reductions



- function f should be poly-time computable

Definition: $f : \Sigma^* \rightarrow \Sigma^*$ is poly-time computable if for some $g(n) = n^{O(1)}$ there exists a g(n)-time TM M_f such that on every $w \in \Sigma^*$, M_f halts with f(w) on its tape.

Poly-time reductions

Definition: $A \leq_P B$ ("A reduces to B") if there is a poly-time computable function f such that for all w

$$w \in A \Leftrightarrow f(w) \in B$$

- condition equivalent to:
  - YES maps to YES *and* NO maps to NO
- meaning is:
  - B is at least as "hard" (or expressive) as A

Poly-time reductions

Theorem: if $A \leq_P B$ and $B \in P$ then $A \in P$.

Proof:
- a poly-time algorithm for deciding A:
- on input w, compute f(w) in poly-time.
- run poly-time algorithm to decide if $f(w) \in B$
- if it says "yes", output "yes"
- if it says "no", output "no"

Hardness and completeness

- Reasonable that can efficiently transform one problem into another.
- Surprising:
  - can often find a special language L so that **every** language in a given complexity class reduces to L!
  - powerful tool

Hardness and completeness

- Recall:
  - a language L is a set of strings
  - a complexity class C is a set of languages

Definition: a language L is C-hard if for every language $A \in C$, A poly-time reduces to L; i.e., $A \leq_P L$.

meaning: L is at least as "hard" as anything in C

Hardness and completeness

- Recall:
  - a language L is a set of strings
  - a complexity class C is a set of languages

Definition: a language L is C-complete if L is C-hard and $L \in C$

meaning: L is a "hardest" problem in C

Lots of NP-complete problems

- logic problems
  - 3-SAT = {φ : φ is a satisfiable 3-CNF formula}
  - NAE3SAT, (3,3)-SAT
  - Max-2-SAT
- finding objects in graphs
  - independent set
  - vertex cover
  - clique
- sequencing
  - Hamilton Path
  - Hamilton Cycle and TSP
- problems on numbers
  - subset sum
  - knapsack
  - partition
- splitting things up
  - max cut
  - min/max bisection