College of Computer & Information Science — Fall 2007
Northeastern University — Handout 1
CS G399: Algorithmic Power Tools I — 4 December 2007

Lecture Outline:

- Recursion Theory
- Randomness
- Volume Estimation

In this lecture, we have an overview of the Kleene hierarchy, followed by a discussion of the power of randomized algorithms. We look at the connections between the Kleene hierarchy and randomized classes such as BPP. We then introduce volume estimation.

1 Recursion Theory

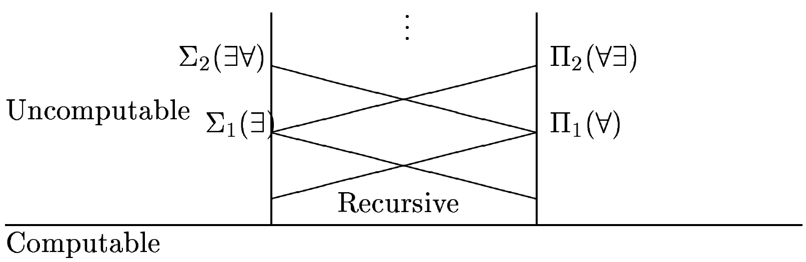

Figure 1: Kleene Hierarchy

Recursion theory is an important branch of Computer Science, although it is not very active currently. It categorizes problems by computability; see the Kleene hierarchy. Problems in this hierarchy can be characterized by logical formulae, with a move up and left in the diagram corresponding to adding an existential quantifier, and moving up and right corresponding to adding a universal quantifier.

Problems in Σ_1 can be accepted by a Turing machine, but not rejected; similarly, problems in Π_1 can be rejected by a Turing machine, but not accepted. Given an oracle to decide some level in the hierarchy, some Turing machine can accept or reject the next level up, depending on which quantifier is added.

The recursive problems are also divided into a hierarchy (fig. 2). This hierarchy divides the computable problems by the amount of resources that a machine requires to decide them. Here, Σ_1^P is the class NP, and Π_1^P is the class co-NP. There are polynomially many levels to this hierarchy. Since everything in the hierarchy can be written as a quantified boolean formula, it is reducible to QBF; thus, it is closed by the class PSPACE

The next time complexity class is EXPTIME. We know that P $\subset$ EXPTIME, and P $\subseteq$ PSPACE $\subseteq$ EXPTIME, but it is not known whether any two of these classes are equal.

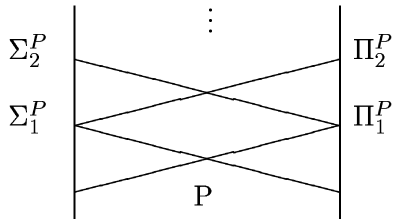

Figure 2: Time/space hierarchy

2 Randomness

Randomness is a resource that allows us to do interesting things. For instance, consider the graph nonisomorphism problem. This is in co-NP, so suppose we have access to a god. Even if the oracle is untrustworthy (may give the wrong answer), we can get a reliable answer using randomness. If we have graphs G_0 and G_1, and the god says that they are nonisomorphic, then we randomly select one, permute it, and ask the god to identify it. If the graphs are nonisomorphic, the god can tell which it is by checking for isomorphism with G_0 and G_1. If the god lied, then there is no way to tell, and the response has only a 50% chance of being correct. We can then repeat this experiment as many times as necessary, until a desired probability of being correct is reached.

2.1 BPP

BPP is the class of problems solvable given polynomial time and random coins. More precisely, for a language L in BPP, there is some randomized machine that decides L in polynomial time, with at most a $1/3$ chance of deciding wrong. Note that any probability less than .5 of deciding wrong allows an arbitrarily small error probability to be reached by iterating, and using whichever answer was given the most times.

Related to BPP is the class RP, where machines always answer correctly when the correct answer is “yes”, but have some probability of answering wrong when the correct answer is “no”. co-RP is the opposite of this situation; the answer is always right in the “no” case, and may be wrong for the “yes” case.

ZPP is the class of problems that a machine can decide correctly in expected polynomial time. $\mathrm{ZPP} = \mathrm{RP} \cap \mathrm{co} - \mathrm{RP}$, by bisimulation. We also know that $P \subseteq \mathrm{ZPP} \subseteq \mathrm{BPP}$, but nobody knows any more than that. It does seem, though, that there is something weird going on in these classes.

For instance, primeness testing used to be the canonical problem known to be in BPP but not known to be in P, until recently a polynomial time algorithm for it was found. A problem in BPP that is still not known to be in P is determining whether the determinant of a matrix with both constant and variable entries is 0. Since in this case the determinant can have exponentially many terms, it is not nown to be in P. However, we know the maximum degree of the determinant polynomial by the size of the matrix, so we can randomly try to guess the roots.

E is the class of problems that are accepted in $2^{O(n)}$ time, and SUBEXP is the class accepted in

$2^{O(n^\epsilon)}$. Surprisingly, if E is different from SUBEXP, then P = BPP. If P $\neq$ BPP, then the entire polynomial hierarchy collapses.

A toy example of speeding up computation with randomness is matrix product verification. Suppose we have matrices A and B, and we want to verify $AB = C$. Performing the multiplication is roughly $O(n^3)$. However, we can choose a random vector r, and check whether $(AB - C)r = 0$. By algebra, this is $A(Br) = Cr$. Since the matrix-vector multiplications can be done in quadratic time, the verification becomes $O(n^2)$.

3 Volume Estimation

Toda's theorem says that $PH \subseteq P^{\sharp P}$. $\sharp P$ is equivalent to counting, which is equivalent to volume estimation (i.e., volume estimation is very hard).

The idea is to generate a polytope whose volume is equal to the number of things we are counting. This is difficult to do deterministically, but easy with randomness (This is a big deal).

Our model is n-dimensional space, and the goal is to estimate the volume of a convex object. We requires as input the dimension n and $r1 \leq r2$ so that the object contains $B(r_1)$ and is contained by $B(r_2)$. $(B_{r_1} \subseteq k \subseteq B_{r_2})$. Alternatively, we can require that the object contains the origin, and the unit ball contains the object. $(0^n \in k \subseteq B(1))$

To estimate the volume, we select m points at random from $B(r_2)$. For each point $p_1 \ldots p_m$, we check whether it is inside the object. We claim, then, that $\frac{\mathtt{Vol}(\mathtt{Conv}(\langle p_i \rangle))}{\mathtt{Vol}(B(1))} \leq m/(2-\epsilon)^n$.

We can create an approximation of the polytope itself as follows. For each point p_i, let B_i be the ball with the line segment from the origin to p_i as a diameter.(see figure 3 Then we use $\bigcup_i B_i$ as an approximation; we claim that $\mathtt{conv}(p_i) \subseteq \bigcup_i B_i$.

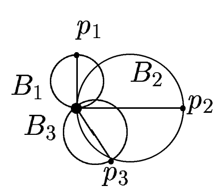

Figure 3: Ball construction

We demonstrate this claim as follows. Consider a point in the convex hull. If B_i contains it, then the angle from pole to pole is ≥ 90 deg. For points outside a ball, this angle is acute. We need to show that $\forall x \notin \bigcup B_i$ and $\forall v \in \mathtt{conv}(p_i)$, the line $\overline{0v}$ subtends < 90 deg at x. This is true because: $\forall i, \overline{0p_i}$ subtends < 90 deg at x. Therefore $\forall i, (p_i - x)\dot{(}0 - x) > 0$, so $\langle v - x, 0 - x\rangle > 0$.