Randomized Methods in Computation*
Lecture 6: Uniform Generation

Notes taken by Eden Chlamtac and Shimon Kogan

Spring 2001

Summary: This lecture continues the discussion of approximate counting and uniform generation using an NP-oracle begun in the previous lecture, where we saw an approximate counting algorithm. Here we will see the equivalence (up to Turing reductions) of approximate counting to uniform generation of NP-witnesses. Together with the approximate counting algorithm from last lecture, this yields a uniform generation algorithm using an NP-oracle. We conclude with an alternative (direct) uniform generation algorithm, which uses n-wise independent hash functions (as opposed to the pairwise independent hash functions used for approximate counting).

1 Introduction

Recall from the previous lecture that we were concerned with answering quantitative questions about sets for which we have an oracle. Specifically, given some NP-relation $R \subseteq \{0,1\}^* \times \{0,1\}^*$, and some input string $x \in \{0,1\}^n$, we would like to answer quantitative questions regarding the set

$$S = R(x) \stackrel{\text{def}}{=} \{y : (x,y) \in R\} \subseteq \{0,1\}^n$$

(Note that we assume if $y \in R(x)$ then $|y| = n = |x|$. This is without loss of generality— it can be achieved with suitable padding.)

The previous lecture was concerned with *approximate counting*—approximating the size of S. In this lecture we are concerned with the closely related problem of *uniform generation*, or sampling with uniform distribution an element in S. In fact, we will show that the two computational problems are equivalent. That is, we show that *Approximate Counting and Uniform Generation are computationally equivalent.*

2 Approximate Counting Reduces to Uniform Generation

At our current state of knowledge (i.e., having seen n approximate counting algorithm in the previous lecture but not a uniform generation one), this is the less interesting direction of the equivalence. Still, it is simpler to explain and is needed to establish the computational equivalence of the two tasks.

* Lecture Notes for a course given by Oded Goldreich at the Weizmann Institute of Science, Israel.

2.1 Intuition

Assume we have a generic uniform generation algorithm UniGen, which for any NP-relation R and input x outputs $\mathrm{UniGen}_R(x)$—a string chosen with uniform distribution from the set $R(x)$.

Using UniGen we will generate a sufficiently large sample from $|S|$ to estimate what fraction of the strings in $|S|$ start with a particular bit σ. Denoting this subset of S by S_σ, we estimate

$$p_\sigma \stackrel{\text{def}}{=} \frac{|S_\sigma|}{|S|}$$

We then recursively approximate the quantity $|S_\sigma|$, and then produce an approximation for $|S|$ using the formula

$$|S| = \frac{1}{p_\sigma} \cdot |S_\sigma|$$

The recursion will proceed to S_σ such that $p_\sigma > \frac{1}{2}$ (the larger of the two values p_σ and $p_{\bar{\sigma}}$), since we need a lower bound on p_σ in order to get a good bound on an estimate of the form $\frac{1}{p_\sigma \pm \epsilon}$.

Our goal is to define an algorithm Approx, which, given some arbitrarily small parameters $\epsilon > 0$, $\delta > 0$, satisfies:

$$\Pr[\mathrm{Approx}_R(x) \notin (1 \pm \epsilon) \cdot |R(x)|] < \delta$$

2.2 Formal Definition

Recall that our uniform generation algorithm relates to the NP-relation R. In order to apply it to subsets of S we need to define new NP-relations. Formally, we define

$$R' \stackrel{\text{def}}{=} \{((x, \alpha), \beta) : \alpha, \beta \in \{0,1\}^*, (x, \alpha \circ \beta) \in R\}$$

(Here, $\alpha \circ \beta$ denotes the concatenation of the two strings α and β.) Clearly, if R is an NP-relation, then R' is also an NP-relation. Moreover, for any $\alpha_0 \in \{0,1\}^*$, R' satisfies

$$R'(x, \alpha_0) = \{\beta : (x, \alpha_0 \circ \beta) \in R\} \cong \{\alpha \circ \beta \in R(x) : \alpha = \alpha_0\}$$

Note that the set on the right is of the sort we want to quantify in the recursion.

Note that since the error and deviation bounds will increase with successive approximations at various levels of recursion, we will have to start out with stricter bounds. Specifically, we take

$$\tilde{\epsilon} = \tfrac{\epsilon}{3n} \text{ and } \tilde{\delta} = \tfrac{\delta}{n}$$

(Here ϵ and δ are the deviation and error parameters in the performance specification noted above).

Since $|R(x)| = |R'(x, \lambda)|$ (where λ is the empty string), for the sake of uniformity of notation, we present an algorithm for approximating $|R'(x, \alpha)|$ where the string $\alpha \in \{0,1\}^*$ is part of the input (along with $x \in \{0,1\}^n$ of course), and initially $\alpha = \lambda$.

2.3 Algorithm $\mathrm{Approx}_{R'}(x, \alpha)$

1. We let $n \stackrel{\text{def}}{=} |x|$ as before. If $|\alpha| = n$ return $\begin{cases} 1 & \text{if } \alpha \in R(x) \\ 0 & \text{otherwise} \end{cases}$

2. Otherwise, for $i = 1, \ldots, m$ uniformly and independently generate sample points in $R'(x, \alpha)$:

$$s_i \leftarrow \mathrm{UniGen}_{R'}(x, \alpha)$$

where

$$m = O\left(\frac{1}{\tilde{\epsilon}^2} \log \frac{1}{\tilde{\delta}}\right)$$

3. For $\sigma = 0, 1$ set

$$\tilde{p}_\sigma \stackrel{\mathrm{def}}{=} \frac{|\{i : s_i \in R'(x, \alpha \circ \sigma)\}|}{m}$$

(an approximation for $p_\sigma \stackrel{\mathrm{def}}{=} \frac{|R'(x,\alpha\circ\sigma)|}{|R'(x,\alpha)|}$)

4. For σ s.t. $\tilde{p}_\sigma \geq \tilde{p}_{\bar{\sigma}}$ return

$$\frac{1}{\tilde{p}_\sigma} \cdot \mathrm{Approx}_{R'}(x, \alpha \circ \sigma)$$

2.4 Analysis

We would like to see how the error and deviation bounds propagate through the recursion. First note that in step (2), at every level of the recursion, by choice of m and applying Chernoff bound, we get:

$$\Pr[|p_\sigma - \tilde{p}_\sigma| > \tilde{\epsilon}] < \tilde{\delta}$$

At the ith level of recursion (where $|\alpha| = i$) we will denote the error and deviation bounds by δ_i and ϵ_i, so that:

$$\Pr[\mathrm{Approx}_{R'}(x, \alpha) \notin (1 \pm \epsilon_i) \cdot |R'(x, \alpha)|] < \delta_i$$

Given these two probability bounds, let us show how the error and deviation propagate from level $i + 1$ of the recursion up to level i. Recall that:

$$\mathrm{Approx}_{R'}(x, \alpha) \stackrel{\mathrm{def}}{=} \frac{\mathrm{Approx}_{R'}(x, \alpha \circ \sigma)}{\tilde{p}_\sigma}$$

We know that with probability $> 1 - (\tilde{\delta} + \delta_{i+1})$ we have both $|p_\sigma - \tilde{p}_\sigma| \leq \tilde{\epsilon}$ and

$$\mathrm{Approx}_{R'}(x, \alpha \circ \sigma) \in (1 \pm \epsilon_{i+1}) \cdot |R'(x, \alpha)|$$

In this case it holds that:

$$\begin{aligned}
\frac{\mathrm{Approx}_{R'}(x, \alpha \circ \sigma)}{\tilde{p}_\sigma} &= \frac{1 \pm \epsilon_{i+1}}{\tilde{p}_\sigma} \cdot |R'(x, \alpha \circ \sigma)| \\
&= \frac{1 \pm \epsilon_{i+1}}{\tilde{p}_\sigma} \cdot p_\sigma \cdot |R'(x, \alpha)| \\
&= \frac{(\tilde{p}_\sigma \pm \tilde{\epsilon}) \cdot (1 \pm \epsilon_{i+1})}{\tilde{p}_\sigma} \cdot |R'(x, \alpha)| \\
&= \left((1 \pm \epsilon_{i+1}) \pm \frac{\tilde{\epsilon}}{\tilde{p}_\sigma}(1 \pm \epsilon_{i+1})\right) \cdot |R'(x, \alpha)| \\
&= \left(1 \pm \left(\epsilon_{i+1} + \frac{\tilde{\epsilon}}{\tilde{p}_\sigma}(1 + \epsilon_{i+1})\right)\right) \cdot |R'(x, \alpha)|
\end{aligned}$$

where the second equality follows from from the equality $|R'(x, \alpha \circ \sigma)| = p_\sigma |R'(x, \alpha)|$, which follows immediately from the definition of p_σ. Now it remains to bound

$$\epsilon_i \stackrel{\text{def}}{=} \epsilon_{i+1} + \frac{\tilde{\epsilon}}{\tilde{p}_\sigma}(1 + \epsilon_{i+1})$$

Recall that by our choice of $\tilde{p}_\sigma$, we know

$$\tilde{p}_\sigma \geq 1/2$$

Also, as will be apparent from the following analysis, if we assume w.l.o.g. that $\epsilon < 1/2$ then it is easy to show that

$$\epsilon_i < 1/2$$

for all i.
Keeping these two inequalities in mind, we have

$$\begin{aligned}\epsilon_i &= \epsilon_{i+1} + \frac{\tilde{\epsilon}}{\tilde{p}_\sigma}(1 + \epsilon_{i+1}) \\ &< \epsilon_{i+1} + \frac{\tilde{\epsilon}}{1/2}(1 + (1/2)) \\ &= \epsilon_{i+1} + 3\tilde{\epsilon}\end{aligned}$$

So clearly the bounds we chose—

$$\tilde{\epsilon} = \tfrac{\epsilon}{3n} \text{ and } \tilde{\delta} = \tfrac{\delta}{n}$$

are sufficient.

2.5 An alternative reduction of Approximate Counting to Uniform Generation

Here is another way of reducing approximate counting to uniform generation, suggested by Udi Wieder. Rather than recursively taking smaller and smaller subsets of S, we may take some predefined set K (with known cardinality and easily decidable) of comparable size to S, such that $S \bigcap K = \emptyset$, and then get a good approximation of S by using repeatedly generating random samples in $S \cup K$, to approximate

$$\beta \stackrel{\text{def}}{=} \frac{|S|}{|S \bigcup K|}$$

Since $|K|$ is known, this will yield an approximation of $|S|$ derived from the following equation:

$$|S| = |K| \cdot \frac{\beta}{1 - \beta}$$

Here we will want to be able to bound β both in terms of upper- and lower-bounds. The situation is similar to that which we had with p_σ in the last algorithm. If we had $\beta \approx 0$ or $\beta \approx 1$ then knowing we had a good approximation of the form $\beta \pm \epsilon$ would not help, since even a small deviation will no longer yield a good relative approximation for β or $\frac{1}{1-\beta}$. Hence, it is essential that we find a set K of comparable size to S (where $S \subseteq \{0,1\}^n$).

Specifically, for all $i = 1, \ldots, n$ we define

$$K_i \stackrel{\text{def}}{=} \{1^{n+1} \circ y : y \in \{0,1\}^i\} \subseteq \{0,1\}^{n+i+1}$$

$$S_i \stackrel{\text{def}}{=} \{0^{i+1} \circ s : s \in S\} \subseteq \{0,1\}^{n+i+1}$$

Note that these sets satisfy the following properties for all i:

- $|K_i| = 2^i$
- $|S_i| = |S|$
- $K_i \cap S_i = \emptyset$ (since $z \in K_i$ start with 1, and $z \in S_i$ start with 0)

Intuitively, $S_i \cup K_i$ is a witness set for the input $x \circ 01^i$. This motivates the definition of the following NP-relation:

$$R'' \stackrel{\text{def}}{=} \left\{ (x \circ 01^i, z) : \begin{array}{l} 1 \leq i \leq |x| \wedge x \in \{0,1\}^* \\ \wedge \left[(z = 0^{i+1} \circ y \wedge (x,y) \in R) \vee ((z = 1^{n+1} \circ y) \wedge y \in \{0,1\}^i) \right] \end{array} \right\}$$

Clearly, for some $1 \leq i \leq n$ we have $\frac{1}{2}|K_i| \leq |S| \leq |K_i|$ (where $S = R(x)$ and $x \in \{0,1\}^n$ as above). If β is defined as above for $K = K_i$ (i.e. $\beta = \frac{|S|}{|S|+|K_i|}$), then we get $\frac{1}{3} \leq \beta \leq \frac{1}{2}$. Hence if we can find such an i, knowing β is bounded as above, we can guarantee a good approximation of $|S|$. Here, since the deviation does not need to propagate through any recursion (there is no recursion), we can take $\tilde{\epsilon} \stackrel{\text{def}}{=} \frac{\epsilon}{9}$. The error remains as in the last algorithm (we still apply union bound on the probabilities), i.e. $\tilde{\delta} \stackrel{\text{def}}{=} \frac{\delta}{n}$. Here's the formal algorithm:

1. For $i = 1$ to n do:
   - For $j = 1, \ldots, m$ uniformly and independently generate sample points in $S \bigcup K_i$:
   $$s_j \leftarrow \text{UniGen}_{R''}(x \circ 01^i)$$
   where again
   $$m = O\left(\frac{1}{\tilde{\epsilon}^2} \log \frac{1}{\tilde{\delta}}\right)$$
   - Set $\beta_i \stackrel{\text{def}}{=} \frac{|\{j : s_j \in S_i\}|}{m}$
2. For $i_0 \stackrel{\text{def}}{=} \min\left\{i : \beta_i \geq \frac{1}{4}\right\}$ return
$$2^{i_0} \cdot \frac{\beta_{i_0}}{1 - \beta_{i_0}}$$

3 Uniform Generation Reduces to Approximate Counting

Given our current state of knowledge, this is indeed the more interesting reduction.

3.1 Intuition

Suppose we had perfect counting. In particular, suppose we could calculate the exact quantity $|S|$, and for $\sigma = 0, 1$ we could calculate the size of the set of all elements in S which start with σ (denoted S_σ), as in the previous section. In particular, we could compute the exact values

$$p_\sigma \stackrel{\text{def}}{=} \frac{|S_\sigma|}{|S|}$$

We want to output $y \in S$ with uniform distribution. So

- with probability p_0 output (recursively) $y \in S_0$ (with uniform distribution)
- with probability p_1 output (recursively) $y \in S_1$ (with uniform distribution)

3.2 Formal Definitions

Now assume we have an approximate counting algorithm Approx which for any NP-relation R and inputs x, ϵ, δ returns $\text{Approx}_R(x, \epsilon, \delta)$ where

$$\Pr[\text{Approx}_R\,(x, \delta, \epsilon) \notin (1 \pm \epsilon)|R(x)|] < \delta$$

and

$$\text{Approx}_R(x, \delta, \epsilon) = |R(x)| \text{ if } |R(x)| = 0$$

Now using Approx we'll generate a random element of $R(x)$ bit by bit in a recursive manner. This we do once more using the modified form of R which we defined in the previous section:

$$R' \stackrel{\text{def}}{=} \{((x, \alpha), \beta) : \alpha, \beta \in \{0,1\}^*, (x, \alpha \circ \beta) \in R\}$$

where we assume that $\Pr[\text{Approx}_{R'}\,(x, \alpha) \notin (1 \pm \epsilon)|R'(x, \alpha)|] < \delta$ and $\text{Approx}_{R'}\,(x, \alpha) = |R'(x, \alpha)|$ if $|R'(x, \alpha)| = 0$. First we'll define an auxiliary algorithm ShApprox which returns slightly shifted approximation values. Then using this algorithm we'll define the uniform generation algorithm UniGen. As always we assume that in R the length of all witnesses is equal to the length of the input x.

3.3 Algorithm $\text{ShApprox}_{R'}\,(x, \alpha)$

1. Let $n = |x|, i = |\alpha|$
2. Return $\begin{cases} \frac{4n^2+4n-4i}{4n^2+1} \cdot \text{Approx}_{R'}\,(x, \alpha) & if \quad i < n \\ \text{Round}(\text{Approx}_{R'}\,(x, \alpha)) & if \quad i = n \end{cases}$

In all the calls made to Approx by this algorithm we fix ϵ and δ to be $\epsilon = \frac{1}{4n^2}$ and $\delta = \frac{1}{3n^2}$. Notice that by the above definition we have that for $i = n$ ShApprox returns 1 if $\alpha \in R(x)$ and 0 otherwise. The purpose of this shifted approximation (as we'll show later in the analysis) is to ensure that with high probability we have:

$$\text{ShApprox}_{R'}\,(x, \alpha \circ 0) + \text{ShApprox}_{R'}\,(x, \alpha \circ 1) \leq \text{ShApprox}_{R'}\,(x, \alpha)$$

3.4 Algorithm $\text{UniGen}_{R'}(x, \alpha)$

1. Let $n = |x|, i = |\alpha|$
2. If $i = n$ return λ (where λ is the empty string)
3. For $\sigma = 0, 1$ set

   $$\tilde{p}_\sigma \stackrel{\text{def}}{=} \frac{\text{ShApprox}_{R'}\,(x, \alpha \circ \sigma)}{\text{ShApprox}_{R'}\,(x, \alpha)}$$

   In this calculation, reuse value returned by calculation of $\text{ShApprox}_{R'}\,(x, \alpha)$ in the previous level of recursion (unless this is the first level of recursion, in which case we apply $\text{ShApprox}_{R'}\,(x, \lambda)$ once and use the same value in calculating p_0 and p_1).
4. If $\tilde{p}_0 + \tilde{p}_1 > 1$ return($\perp$)
5. With probability $\tilde{p}_0$ return $(0 \circ \text{UniGen}_{R'}(x, \alpha \circ 0))$
   With probability $\tilde{p}_1$ return $(1 \circ \text{UniGen}_{R'}(x, \alpha \circ 1))$
   Otherwise return($\perp$)

We invoke the algorithm by $\text{UniGen}_{R'}(x, \lambda)$, i.e. $\text{UniGen}_R(x) = \text{UniGen}_{R'}(x, \lambda)$.

3.5 Analysis

Let's begin with a definition

Definition 1 *We say that a call to* $\mathrm{Approx}_{R'}(x,\alpha)$ *is* successful *if*

$$\mathrm{Approx}_{R'}(x,\alpha) \in (1 \pm \epsilon)\,|R'(x,\alpha)|$$

First of all notice that

$$\Pr\left[\text{all calls to Approx are successful during the running of UniGen}\right] \geq 1 - \frac{1}{n}$$

since $\Pr\left[\text{Approx unsuccessful}\right] < \delta = \frac{1}{3n^2}$ and at each stage of the recursion we invoke Approx (via ShApprox) at most 3 times.
Now we'll begin establishing the main results.

Claim 3.1 *For all* $0 \leq i < n$ *if the call to* $\mathrm{Approx}_{R'}(x,\alpha)$ *in* $\mathrm{ShApprox}_{R'}(x,\alpha)$ *is successful then*

$$\left(1 + \frac{n-i-1}{n^2}\right) \cdot |R'(x,\alpha)| \leq \mathrm{ShApprox}_{R'}(x,\alpha) \leq \left(1 + \frac{n-i}{n^2}\right) \cdot |R'(x,\alpha)|$$

Proof: For $0 \leq i < n$ we have

$$\mathrm{ShApprox}_{R'}(x,\alpha) = \frac{4n^2+4n-4i}{4n^2+1} \cdot \mathrm{Approx}_{R'}(x,\alpha)$$

so if $\mathrm{Approx}_{R'}(x,\alpha) \in (1 \pm \epsilon)\,|R'(x,\alpha)|$ (for $\epsilon = \frac{1}{4n^2}$) then

$$\frac{4n^2+4n-4i}{4n^2+1} \cdot \left(1 - \frac{1}{4n^2}\right) \cdot |R'(x,\alpha)| \leq \mathrm{ShApprox}_{R'}(x,\alpha) \leq \frac{4n^2+4n-4i}{4n^2+1} \cdot \left(1 + \frac{1}{4n^2}\right) \cdot |R'(x,\alpha)|$$

Now

$$\frac{4n^2+4n-4i}{4n^2+1} \cdot \left(1 + \frac{1}{4n^2}\right) = \frac{(4n^2+4n-4i)\cdot(4n^2+1)}{(4n^2+1)\cdot 4n^2} = 1 + \frac{n-i}{n^2}$$

and thus the upper bound follows.
For the lower bound we begin by noticing that

$$\frac{4n^2+4n-4i}{4n^2+1} \cdot \left(1 - \frac{1}{4n^2}\right) = \frac{(n^2+n-i)\cdot(4n^2-1)}{(4n^2+1)\cdot n^2}$$

Now since $(n^2+n-i)(4n^2-1) > (n^2+n-i-1)(4n^2+1)$, we get that

$$\frac{(n^2+n-i)(4n^2-1)}{(4n^2+1)n^2} > \frac{n^2+n-i-1}{n^2} = 1 + \frac{n-i-1}{n^2}$$

and the lower bound follows. ■

Corollary 3.2 *If all calls to* Approx *during the running of* UniGen *are successful then in any stage of the recursion* $\tilde{p}_0 + \tilde{p}_1 \leq 1$

Proof: Assume all calls to Approx are successful. Let $i \stackrel{\text{def}}{=} |\alpha|$. Now if $0 \le i \le n-2$ then by the lemma above

$$\begin{aligned}
\text{ShApprox}_{R'}(x, \alpha \circ 0) &+ \text{ShApprox}_{R'}(x, \alpha \circ 1) \\
&\le \left(1 + \frac{n-i-1}{n^2}\right) \cdot |R'(x, \alpha \circ 0)| + \left(1 + \frac{n-i-1}{n^2}\right) \cdot |R'(x, \alpha \circ 1)| \\
&= \left(1 + \frac{n-i-1}{n^2}\right) \cdot |R'(x, \alpha)| \\
&\le \text{ShApprox}_{R'}(x, \alpha)
\end{aligned}$$

On the other hand, for $i = n-1$ we have for every $\sigma \in \{0, 1\}$

$$|R'(x, \alpha \circ \sigma)| = \text{Round}(\text{Approx}_{R'}(x, \alpha \circ \sigma)) = \text{ShApprox}_{R'}(x, \alpha \circ \sigma)$$

where the first equality is due to the fact that $|R'(x, \alpha \circ \sigma)| \le 1$ and so rounding the value of the good approximation must yield exactly the correct value, which is preserved by ShApprox. Again, by Claim 3.1: $|R'(x, \alpha)| \le \text{ShApprox}_{R'}(x, \alpha)$, and so it follows that

$$\text{ShApprox}_{R'}(x, \alpha \circ 0) + \text{ShApprox}_{R'}(x, \alpha \circ 1) = |R'(x, \alpha)| \le \text{ShApprox}_{R'}(x, \alpha)$$

We conclude that in both cases

$$\tilde{p}_0 + \tilde{p}_1 = \frac{\text{ShApprox}_{R'}(x, \alpha \circ 0)}{\text{ShApprox}_{R'}(x, \alpha)} + \frac{\text{ShApprox}_{R'}(x, \alpha \circ 1)}{\text{ShApprox}_{R'}(x, \alpha)} \le 1$$

■

Lemma 3.3 *Assume that all calls to* Approx *are successful then for all* x, α, β

$$\Pr[\text{UniGen}_{R'}(x, \alpha) = \beta] = \begin{cases} \frac{1}{\text{ShApprox}_{R'}(x,\alpha)} & \textit{if } \alpha \circ \beta \in R(x) \\ 0 & \textit{otherwise} \end{cases}$$

furthermore if $\alpha \circ \beta \notin R(x)$ *then* $\Pr[\text{UniGen}_{R'}(x, \alpha) = \beta] = 0$ *regardless of the success of the* Approx *calls.*

Proof: First of all, we show that if $\alpha \circ \beta \notin R(x)$ then

$$\Pr[\text{UniGen}_{R'}(x, \alpha) = \beta] = 0$$

Observe that if $\alpha \circ \beta \notin R(x)$ then $R'(x, \alpha \circ \beta) = 0$. Now let $\beta = \gamma \circ \sigma$, where $|\gamma| = |\beta| - 1$ and look at the last recursive step of $\text{UniGen}_{R'}(x, \alpha)$ (i.e. the call to $\text{UniGen}_{R'}(x, \alpha \circ \gamma)$). If we have not arrived to this stage then the output was $\perp$, which is certainly not β. In case we did arrive at this step, the probability to output σ and thus finish outputting β as is

$$\tilde{p}_\sigma = \frac{\text{ShApprox}_{R'}(x, \alpha \circ \gamma \circ \sigma)}{\text{ShApprox}_{R'}(x, \alpha \circ \gamma)}$$

Now

$$\text{ShApprox}_{R'}(x, \alpha \circ \gamma \circ \sigma) = \text{Approx}_{R'}(x, \alpha \circ \gamma \circ \sigma) = 0$$

because by our definition of Approx we have that $\text{Approx}_{R'}(x, \alpha) = 0$ if $R'(x, \alpha) = 0$. Thus we get $\tilde{p}_\sigma = 0$ and so σ will not be output and the claim follows. Also notice that we have not used in any part of the above proof the assumption that all calls to Approx were successful.

We now turn to the more interesting case; that is, when $\alpha \circ \beta \in R(x)$. We will be proven the claim for this case by downward induction on $|\alpha|$. The claim is trivially true for $|\alpha| = n$ as in that case $\beta = \lambda$ and

$$\Pr\left[\mathrm{UniGen}_{R'}(x,\alpha) = \lambda\right] = 1 = \frac{1}{\mathrm{ShApprox}_{R'}(x,\alpha)}$$

Now we'll assume that the claim holds for $|\alpha| > k$ and prove it for $|\alpha| = k$. Let $\beta = \sigma \circ \beta'$ then

$$\Pr\left[\mathrm{UniGen}_{R'}(x,\alpha) = \beta\right] = \tilde{p}_\sigma \cdot \Pr\left[\mathrm{UniGen}_{R'}(x,\alpha\circ\sigma) = \beta'\right]$$

Now by the induction hypothesis

$$\Pr\left[\mathrm{UniGen}_{R'}(x,\alpha\circ\sigma) = \beta'\right] = \frac{1}{\mathrm{ShApprox}_{R'}(x,\alpha\circ\sigma)}$$

So we get that

$$\begin{aligned}
\Pr\left[\mathrm{UniGen}_{R'}(x,\alpha) = \beta\right] &= \tilde{p}_\sigma \cdot \frac{1}{\mathrm{ShApprox}_{R'}(x,\alpha\circ\sigma)} \\
&= \frac{\mathrm{ShApprox}_{R'}(x,\alpha\circ\sigma)}{\mathrm{ShApprox}_{R'_\alpha}(x)} \cdot \frac{1}{\mathrm{ShApprox}_{R'}(x,\alpha\circ\sigma)} \\
&= \frac{1}{\mathrm{ShApprox}_{R'}(x,\alpha)}
\end{aligned}$$

■

Corollary 3.4 *Assuming all calls to* Approx *are successful we have for all* $\beta \in R(x)$

$$\Pr\left[\mathrm{UniGen}_{R}(x) = \beta\right] = \frac{1}{\mathrm{ShApprox}_{R'}(x,\lambda)}$$

Proof: Take $\alpha = \lambda$ in Lemma 3.3. ■

Corollary 3.5 *If all calls to* Approx *are successful then for any* $\beta \in R(x)$ *we have*

$$\Pr\left[\mathrm{UniGen}_R(x) = \beta \mid \mathrm{UniGen}_R(x) \neq \perp\right] = \frac{1}{|R(x)|}$$

Proof: By Corollary 3.4 we have

$$\Pr\left[\mathrm{UniGen}_{R}(x) = \beta\right] = \frac{1}{\mathrm{ShApprox}_{R'}(x,\lambda)}$$

and this quantity is independent of β so all valid witnesses will be generated with the same probability. ■

Now we'll try to bound the probability that $\mathrm{UniGen}_{R'}(x,\alpha) = \perp$.

Lemma 3.6 *Under the assumption that all calls to* Approx *are successful we have that*

$$\Pr\left[\mathrm{UniGen}_{R}(x) = \perp\right] \leq \frac{2}{n}$$

Proof: By Claim 3.1 we know that

$$\left(1+\frac{n-i-1}{n^2}\right)\cdot|R'(x,\alpha)| \leq \mathrm{ShApprox}_{R'}(x,\alpha) \leq \left(1+\frac{n-i}{n^2}\right)\cdot|R'(x,\alpha)|$$

Thus we have that in each recursive step of the algorithm the following inequality holds:

$$\begin{aligned}
\tilde{p}_0+\tilde{p}_1 &= \frac{\mathrm{ShApprox}_{R'}(x,\alpha\circ 0)}{\mathrm{ShApprox}_{R'}(x,\alpha)}+\frac{\mathrm{ShApprox}_{R'}(x,\alpha\circ 1)}{\mathrm{ShApprox}_{R'}(x,\alpha)} \\
&\geq \frac{\left(1+\frac{n-i-2}{n^2}\right)\cdot(|R'(x,\alpha\circ 0)|+|R'(x,\alpha\circ 1)|)}{\left(1+\frac{n-i}{n^2}\right)\cdot|R'(x,\alpha)|} \\
&= \frac{n^2+n-i-2}{n^2+n-i} \geq 1-\frac{2}{n^2}
\end{aligned}$$

Since there are n recursion levels and by the above the probability of outputting $\perp$ at Stage 5 of the UniGen algorithm is smaller then $\frac{2}{n^2}$ at each recursion level, we conclude that

$$\Pr[\mathrm{UniGen}_R(x)=\perp] \leq n\cdot\frac{2}{n^2}=\frac{2}{n}$$

■

Corollary 3.7

$$\Pr[\mathrm{UniGen}_R(x)=\perp] \leq \frac{3}{n}$$

Proof: Let AS denote the event *All calls to* Approx *are Successful.*
Using Lemma 3.6 and the fact that $\Pr[\mathrm{AS}] \geq 1-\frac{1}{n}$ we have that

$$\begin{aligned}
\Pr[\mathrm{UniGen}_R(x)=\perp] &= \Pr[(\mathrm{UniGen}_R(x)=\perp)\wedge \mathrm{AS}]+\Pr\left[(\mathrm{UniGen}_R(x)=\perp)\wedge\overline{\mathrm{AS}}\right] \\
&\leq \Pr[\mathrm{UniGen}_R(x)=\perp \mid \mathrm{AS}]+\Pr\left[\overline{\mathrm{AS}}\right] \\
&\leq \frac{2}{n}+\frac{1}{n}
\end{aligned}$$

■

Combining Corollaries 3.5 and 3.7, we conclude that the procedure UniGen_R behaves as desired.

4 An alternate algorithm for uniform generation

4.1 Intuition

Now we shall sketch an alternative algorithm for uniform generation. In addition to using approximate counting this algorithm will use hash functions in a direct way. The hash functions will be of high independence (and thus have stronger "random properties") than pairwise independence used before.

4.2 Definitions

Let $H(n,m,t)$ denote a collection of t-wise independent hash functions of n bits to m bits. This means that for any $y_1,\ldots,y_t \in \{0,1\}^m$ and any distinct $x_1,\ldots,x_t \in \{0,1\}^n$ we have for a hash function h uniformly chosen from $H(n,m,t)$ that

$$\Pr\left[h(x_1) = y_1 \wedge \ldots \wedge h(x_t) = y_t\right] = 2^{-mt}$$

Let $h \in H(n,m,t)$ then for $\alpha \in \{0,1\}^m$ we let $h^{-1}(\alpha) = \{y \in \{0,1\}^n \mid h(y) = \alpha\}$.
Recall that $S = R(x)$ and $n = |x|$.

4.3 Algorithm $\mathrm{UniGen}_R(x)$

1. If $|S| \leq 2n^5$ compute a listing $y_1,\ldots,y_s$ of the members of the set $|S|$, select j at random from $1,\ldots,s$, output y_j and halt.
2. Otherwise, let $N \approx |S|$ (we obtain a good approximation of $|S|$ using approximate counting), and let $m = \log_2 N - 5\log n$
3. Select h uniformly in $H(n,m,n)$
4. If $\exists \alpha \in \{0,1\}^m$ s.t. $|S \cap h^{-1}(\alpha)| \geq 2n^5$ then output $\perp$
5. Select α uniformly in $\{0,1\}^m$
6. Compute $S \cap h^{-1}(\alpha)$ and for every $y \in S \cap h^{-1}(\alpha)$ output it with probability $\frac{1}{2n^5}$, otherwise output $\perp$

The implementation of Steps $1,4,6$ involves the use of an NP-oracle. In each case we build an appropriate NP-relations and use the oracle to check membership in those relations.
For example in Step 4 we would use the following NP-relation and corresponding NP-set:

$$S'' = \{(x,h) : \exists \alpha, y_1,\ldots,y_k \text{ such that } ((x,h),(\alpha,y_1,\ldots,y_k)) \in R''\}$$

$$R'' = \left\{ ((x,h),(\alpha,y_1,\ldots,y_k)) : \begin{array}{l} k = 2|x|^5 \text{ and } y_1,\ldots,y_k \text{ are distinct} \\ \text{and } h \in H(|x|,m,|x|) \\ \text{and } R(x,y_i) = 1 \text{ (for every } i \in [k]) \\ \text{and } h(y_i) = \alpha \text{ (for every } i \in [k]) \end{array} \right\}$$

4.4 Analysis

To show that the algorithm indeed is a uniform generator we note that if $|S| \leq 2n^5$ then the algorithm halts in Step 1 outputting a uniformly chosen element of S. If $|S| > 2n^5$ then considering any possible (fixed) choice of h in Step 3 and assuming a non-$\perp$ result in Step 4 we have that the randomization is only over the choice of α in step 5 and the choices in step 6. And so we have that the probability that any fixed $y \in S$ will be output is:

$$\begin{aligned} \Pr[\text{y is output}] &= \Pr\left[h(y) = \alpha : \alpha \xleftarrow{R} \{0,1\}^m\right] \cdot \frac{1}{2n^2} \\ &= 2^{-m} \cdot \frac{1}{2n^2} \end{aligned}$$

and this quantity is independent of y.

Another important claim is that the probability of a non-$\perp$ output is a positive constant. First notice that if h is drawn at random from $H(n, m, n)$ then the expected size of $|S \cap h^{-1}(\alpha)|$ is $\frac{|S|}{2^m} = n^5$, and from this fact it's possible to show via standard t-wise independence techniques that the probability that $|S \cap h^{-1}(\alpha)| < 2n^5$ for all α is a positive constant. We conclude that the probability of not halting in Step 4 of the algorithm is a positive constant.

Conditioned on not halting in Step 4, the probability that any fixed $y \in S$ is output in Step 6 is $2^{-m} \cdot \frac{1}{2n^5}$, so it follows that the probability of outputting $\perp$ in Step 6 equals

$$1 - |S| \cdot \frac{2^{-m}}{2n^5} = 1 - |S| \cdot \frac{2^{5\log_2 n - log_2 N}}{2n^5} = 1 - \frac{|S|}{2N}$$

Now since N is a good approximation of $|S|$, we have $\frac{|S|}{2N} > \frac{1}{3}$. Thus the probability of outputting an element of S in Step 6 is at least 1/3. Finally, as we have shown that the probability of halting in Step 4 is a positive constant, and the probability of a non-$\perp$ result in Step 6 is a positive constant, we conclude that the probability of the algorithm to output an element of S is a positive constant.