here’s the story i believe:
lets say you have a counting program \( \text{COUNT}(f) \), where \( f \) is a formula.

how do i generate a random assignment of a formula \( f \)?

do as follows:

for \( i = 1, n \)
\[
c_1 = \text{COUNT}(f_{xi=False}) \\
c_2 = \text{COUNT}(f_{xi=True})
\]
set \( xi \) to True with probability \( c_2/(c_1 + c_2) \)
else set \( xi \) to False
end for

so, this gets a random assignment selected uniformly at random by
using an exact \( \text{COUNT} \) program.

Question: How can we use sampling to get (approx) counting?
(look up story. i believe one of the early papers by
valiant covers this.)

Now I know how to do counting using a uniform sampling algorithm. I’ll write
it up with more details in latex later tonight. But here’s the idea.

Giving a formula \( F \) on variables \( x_1, x_2, \ldots, x_n \). Let \( S(F) \) be the number of
solutions of \( F \).

\[
S(F) = (S(F)/S(F ^ (x_1=A_1))) \cdot S(F ^ (x_1=A_1)),
\]
\( A_1 \) is either True or False.

\( (S(F)/S(F ^ (x_1=A_1))) \) can be estimated by sampling the solutions of \( F \),
and see the ratio that satisfies \( x_1 = A_1 \).

To guarantee stability, always pick \( A_1 \) so that at least half of the
solutions of \( F \) satisfy \( x_1 = A_1 \). i.e., keeping \( (S(F)/S(F ^ (x_1=A_1))) \leq 2 \).

Continue doing this to calculate \( S(F ^ (x_1=A)) \), until all variables are
fixed. Then we know the last factor is always 1 as long as a solution
exists.

So \( S(F) \) is the product of these ratios \( S(f) = (S(F)/S(F ^ x_1=A)) \cdot (S(F ^
 x_1 = A_1)/S(F ^ x_1=A_1 ^ x_2=A_2)) \cdot \ldots \cdot 1. \)

Generated by unregistered txt2pdf v.7.1 © SANFACE Software 2004
Available at http://www.sanface.com-txt2pdf.html