Absynth*
Bounded Expectations: Resource Analysis for Probabilistic Programs

Van Chan Ngo | Carnegie Mellon University

*the name originally comes from Tom Reps. Absynth is built based on Patis
Overview

- Static analysis for deriving bounds on the expected resource usage of probabilistic programs
- Fully automated using off-the-shelf LP solving
- Multivariate polynomial bounds on the inputs
- Prototype implementation and benchmark design*

*available upon request at channgo@cmu.edu
Simulates a random walk that ends when the walker passes the boundary.

Each time unit:

- Goes forward 1 step with $p = 3/4$
- Goes backward 1 step with $p = 1/4$

What is the expected value of elapsed time?

```c
int x, n;
x = 0;
while (x < n) {
    x = x + 1
    [3/4]
    x = x - 1;
tick(1);
}
```
Simple Programs

Simulates a random walk that ends when the walker passes the boundary

Each time unit:

- Goes forward 1 step with $p = \frac{3}{4}$
- Goes backward 1 step with $p = \frac{1}{4}$

What is the expected value of elapsed time?

```c
int x, n;
x = 0;
while (x < n) {
    x = x + 1 \left\lfloor \frac{3}{4} \right\rfloor;
    x = x - 1;
    tick(1);
}
```
Trapped Miner

A miner is sent to a mine \( n \) time independently.

With \( p = \frac{1}{2} \), the miner is trapped.

When being trapped, 3 doors to open:

- Door 1: takes 3 hours to safety
- Door 2: takes 5 hours to the mine
- Door 3: takes 7 hours to the mine

What is the expected time to reach safety?
Trapped Miner

A miner is sent to a mine n time independently

With $p = 1/2$, the miner is trapped

When being trapped, 3 doors to open

- Door 1: takes 3 hours to safety
- Door 2: takes 5 hours to the mine
- Door 3: takes 7 hours to the mine

What is the expected time to reach safety?

$$7.5 \max(0, n)$$
Architecture

Base Functions

\[ M := 1 | x | M_1 \cdot M_2 | \max(0, P) \]

Potential & Rewrite Functions

\[ P := k \cdot M | P_1 + P_2 \]
while (x < n) {
    x = x + 1
    [3/4]
    x = x - 1;
    tick(1);
} //termination point

c1 [p] c2 : probabilistic branching

id = e bop R : sampling assignment
R is probability distribution

if * c1 else c2 : non-deterministic choice