Automated Termination Analysis of Polynomial Probabilistic Programs

Marcel Moosbrugger*, Ezio Bartocci, Joost-Pieter Katoen, Laura Kovács

Probabilistic Termination

> Does a probabilistic program with loops stop?

Important for safety analysis of stochastic systems

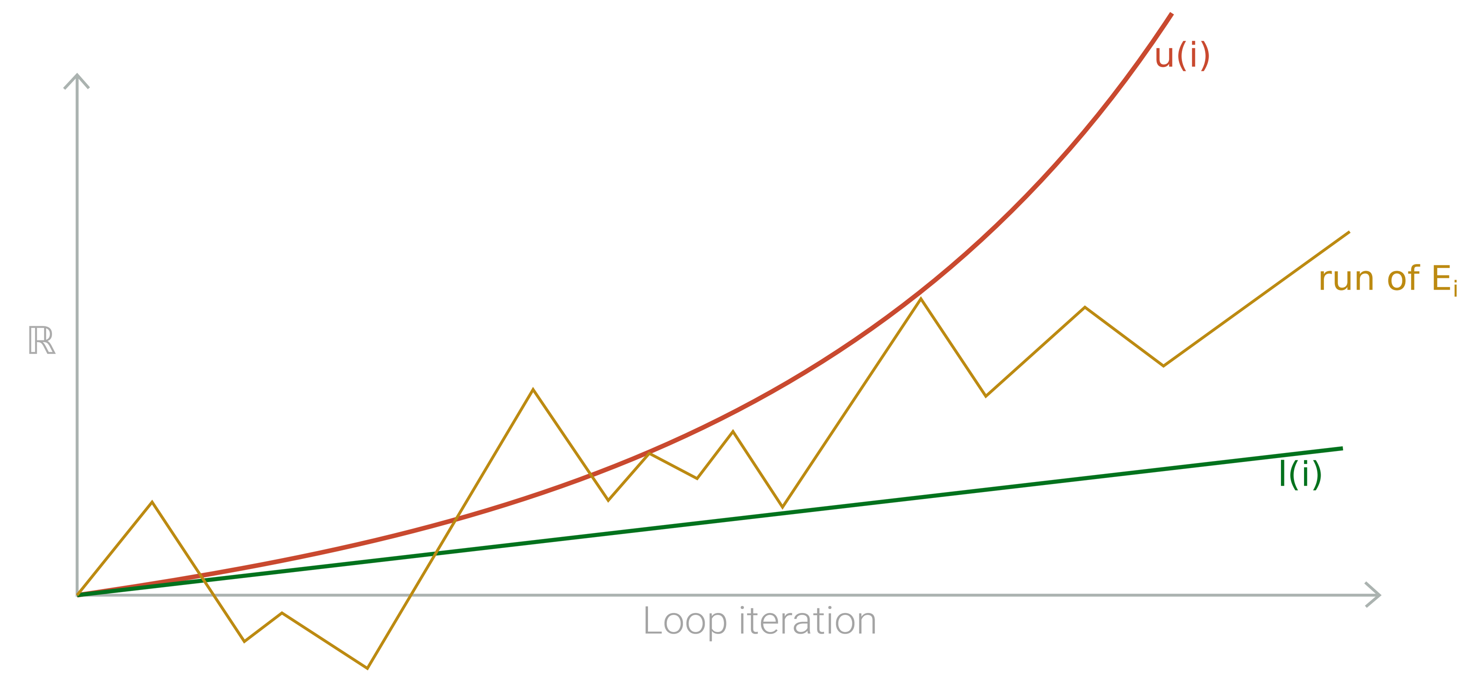
Hard (in general undecidable) problem

Finite Expected Termination **PAST**	$\neq$	Termination with probability 1 **AST**

Our tool Amber **automatically checks PAST, AST and their negations** for specific probabilistic loops with loop guard **G,** supporting **polynomial arithmetic**. The techniques are based on **algebraic recurrences**.

Asymptotic Bounding Functions (ABF)

Used to automate proof rules. Let E be a polynomial over program variables.



Lower bounding function l(i) for E: Eventually and almost-surely E is greater-or-equal than l(i)

Upper bounding function u(i) for E: Eventually and almost-surely E is smaller-or-equal than u(i)

Automated Proof Rules

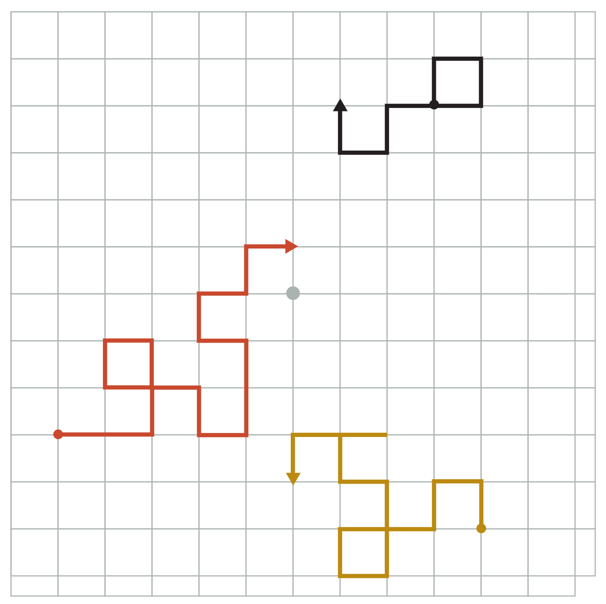


Figure: The symmetric 2D random walk is a stochastic process which is AST but not PAST.

Ranking-Supermartingale-Rule:
If $E(G_{i+1} - G_i \mid F_i) \leq -\epsilon$ then the program is PAST

Supermartingale-Rule:
1. $E(G_{i+1} - G_i \mid F_i) \leq 0$
2. $P(G_{i+1} \leq G_i - d(G_i) \mid F_i) \geq p(G_i)$ for antitone functions p,d

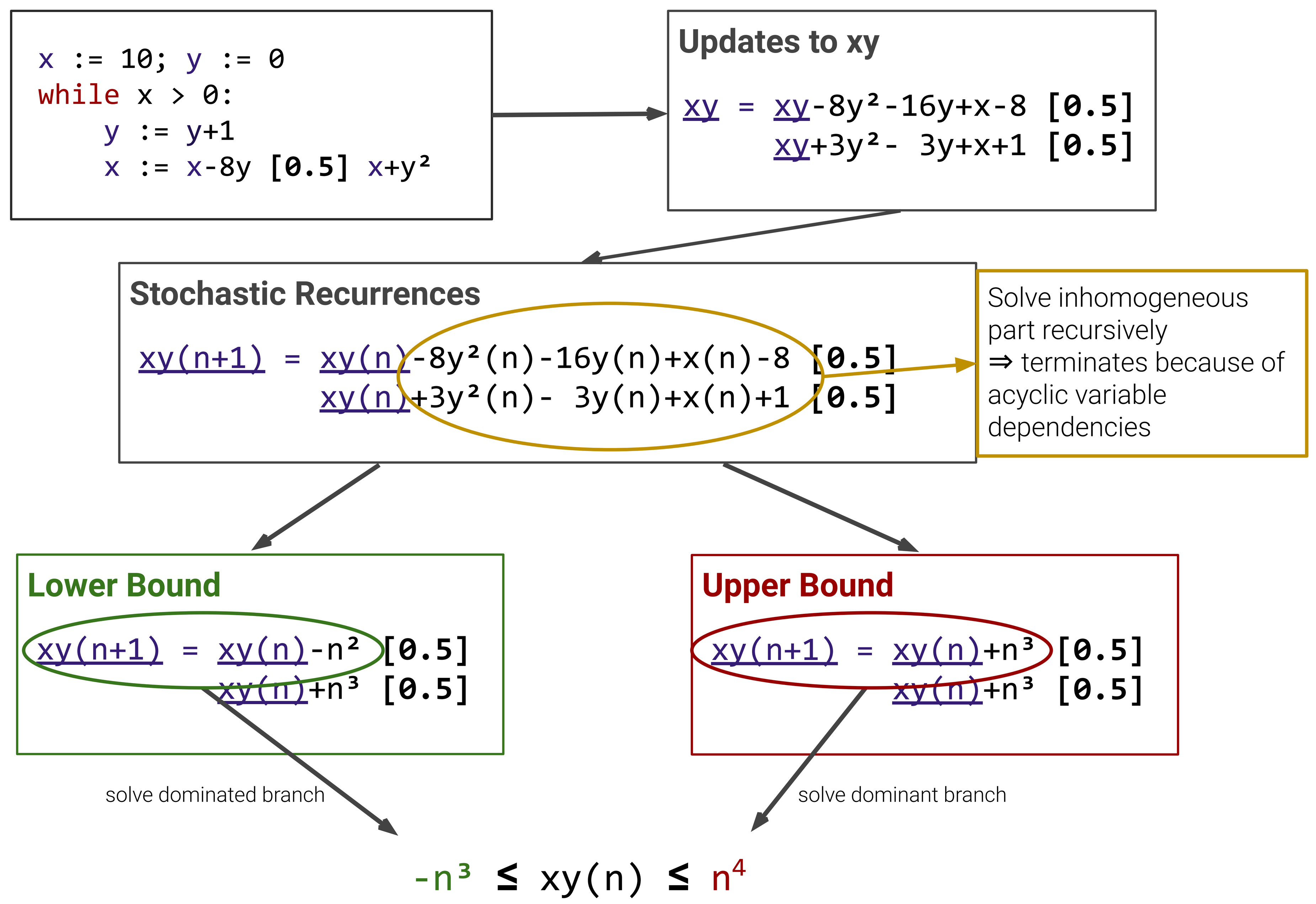
If (1) and (2) hold, then the program is AST

Repulsing-AST-Rule:
1. $E(G_i - G_{i+1} \mid F_i) \leq -\epsilon$
2. Differences of G bounded by c

If (1) and (2) hold, the program is not AST.

Conditions turn out to be **polynomial inequalities over program variables.**
We use **Asymptotic Bounding Functions** to check all of them.

Computing ABFs



Visit **probing-lab.github.io** for more