Probabilistic Higher-Order Recursion Schemes and Termination Probabilities

Naoki Kobayashi
University of Tokyo

joint work with

Ugo Dal Lago (University of Bologna)
Charles Grellois (Aix-Marseille University)

Our Interest

◆ **Model Checking of Probabilistic and Higher-Order Systems**
(with applications to verification of probabilistic functional programs)

cf.

- **Model checking of probabilistic procedural programs (probabilistic pushdown [Esparza+ 04], recursive Markov chains [Etessami&Yannakakis 04])**
- **Model checking of higher-order programs [Knapik+02, Ong06, K09,...]**

This Talk

- pHORS: probabilistic extension of higher-order recursion schemes
- Termination problems for pHORS
  - Undecidability of AST of order-2 pHORS
  - Fixpoint characterization of termination probabilities
  - Approximate computation of termination probabilities

This Talk

- pHORS: probabilistic extension of higher-order recursion schemes
- Termination problems for pHORS
  - Undecidability of AST of order-2 pHORS
  - Fixpoint characterization of termination probabilities
  - Approximate computation of termination probabilities

pHORS

- **A set of (simply-typed) rules of the form**

$$F\,x_1 \ldots x_n = t_L \oplus_p \; t_R$$

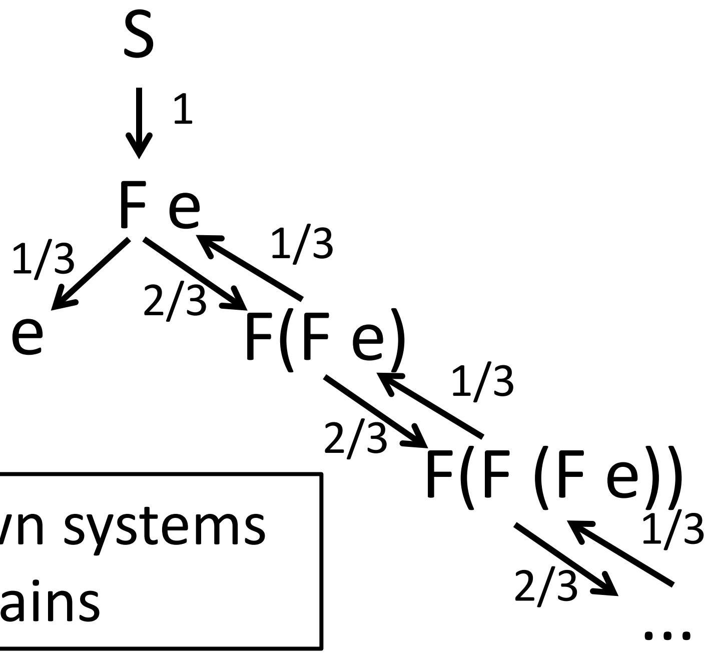
where:

$$t ::= e \text{ (termination)} \mid \Omega \text{ (divergence)} \mid x \mid F \mid t_1 t_2$$

Order-1 pHORS (Random Walk):

$S = F\,e \oplus_1 \Omega$

$F\,x = x \oplus_{1/3} F\,(F\,x)$



Order-1 pHORS ≈ Probabilistic pushdown systems
≈ recursive Markov chains

Termination Probabilities and Verification Problems

- **Termination probability of pHORS G**

TP(G): the probability that $S_G \rightarrow^* e$

$$
\begin{aligned}
G_1: \quad & S = F\ e \\
& F\ x = x \oplus_p F\ (F\ x)
\end{aligned}
$$

$TP(G_1)$ = the least solution of $z=p+(1-p)\,z^2$

$$
= \begin{cases} 1 & \text{if } p \geq 0.5 \\ p/(1-p) & \text{if } p<0.5 \end{cases}
$$

Thus, $TP(G_1)=1$ iff $p \geq 0.5$

Termination Probabilities and Verification Problems

- **Termination probability of pHORS G**

  TP(G): the probability that $S_G \rightarrow^* e$

- **Problems of interest**
  - **Decision problems:**

    Input: G, a rational number $r \in [0,1]$

    Output: whether $TP(G) \sim r$ (where $\sim \in \{=, >, <\}$)

    (Special case: almost sure termination $TP(G)=1$)

Known to be decidable for probabilistic pushdown (or recursive Markov chains) [Esparza+ 04][Etessami&Yannakakis 04]), hence also for order-1 pHORS

Termination Probabilities and Verification Problems

- **Termination probability of pHORS G**

  TP(G): the probability that $S_G \to^* e$

- **Problems of interest**
  - **Decision problems:**

    Input: G, a rational number $r \in [0,1]$

    Output: whether $TP(G) \sim r$ (where $\sim \in \{=, >, <\}$)

    (Special case: almost sure termination TP(G)=1)
  - **Approximation:**

    Input: G, a rational number $\varepsilon>0$

    Output: r such that $|TP(G)-r| < \varepsilon$

Termination Probabilities and Verification Problems

- ♦ Problems of interest
  - Decision problems:

    Input: G, a rational number r ∈ [0,1]

    Output: whether TP(G) ∼ r (where ∼ ∈ {=, >, <})
  - Approximation:

    Input: G, a rational number ε>0

    Output: r such that |TP(G)-r| < ε
- ♦ Why termination?
  - A fundamental property of programs
  - Used as a basis of other model checking procedures for probabilistic pushdown [Etessami+][Esparza+]

Outline

- pHORS: probabilistic extension of higher-order recursion schemes
- Termination Problems
- Undecidability of AST of order-2 pHORS
  - summary of results
  - proof ideas
- Fixpoint characterization of termination probabilities
- Approximate computation of termination probabilities
- Conclusion

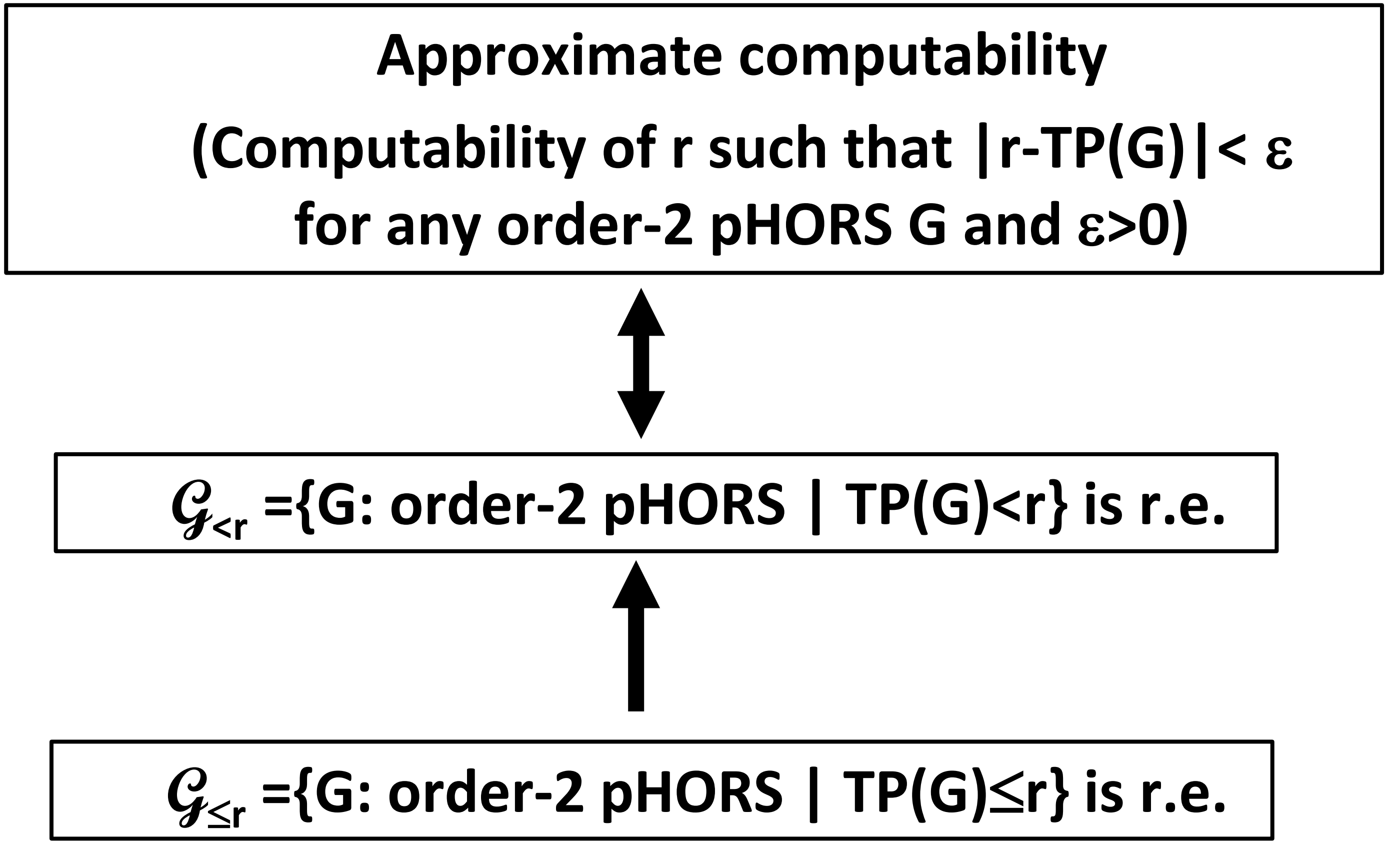
Undecidability of AST (Almost Sure Termination)

- **The following decision problem is undecidable**
  - **Input: order-2 pHORS G**
  - **Output: whether TP(G)=1.**
- **More precisely, the following sets are not recursively enumerable** (for $r \in (0,1]$)
  - $\mathcal{G}_{=r}$ **= {G: order-2 pHORS | TP(G)=r}**
  - $\mathcal{G}_{\geq r}$ **={G: order-2 pHORS | TP(G)≥r}**

  cf. $\mathcal{G}_{>r}$ **={G: order-2 pHORS | TP(G)>r} is r.e.**

  open: whether $\mathcal{G}_{<r}$ **and** $\mathcal{G}_{\leq r}$ **are r.e.**

Relationship between open problems



Approximate computability

(Computability of r such that $|r-TP(G)|< \varepsilon$ for any order-2 pHORS G and $\varepsilon>0$)

$\mathcal{G}_{<r}$ ={G: order-2 pHORS | TP(G)<r} is r.e.

$\mathcal{G}_{\leq r}$ ={G: order-2 pHORS | TP(G)≤r} is r.e.

Outline

- pHORS: probabilistic extension of higher-order recursion schemes
- Termination Problems
- Undecidability of AST of order-2 pHORS
  - summary of results
  - proof ideas
- Fixpoint characterization of termination probabilities
- Approximate computation of termination probabilities
- Conclusion

Proof Idea

- Reduction from Hilbert's 10th Problem (unsolvability of Diophantine equations)
  - Given polynomials $P(x_1,\ldots,x_n)$ and $Q(x_1,\ldots,x_n)$ (with non-negative coefficients), $\exists x_1,\ldots,x_n . P(x_1,\ldots,x_n) < Q(x_1,\ldots,x_n)$ is undecidable (corollary of unsolvability of Diophantine)

    Note: $D(x_1,\ldots,x_n)=0$ iff $D(x_1,\ldots,x_n)^2 < 1$

Proof Idea

- Reduction from Hilbert's 10th Problem (unsolvability of Diophantine equations)
  - Given polynomials $P(x_1,\ldots,x_n)$ and $Q(x_1,\ldots,x_n)$ (with non-negative coefficients), $\exists x_1,\ldots,x_n.P(x_1,\ldots,x_n) < Q(x_1,\ldots,x_n)$ is undecidable (corollary of unsolvability of Diophantine)
  - Given $P(x_1,\ldots,x_n)$ and $Q(x_1,\ldots,x_n)$, one can effectively construct an order-2 pHORS $G_{P,Q}$ s.t.

    $TP(G_{P,Q})<1$ iff $\exists x_1,\ldots,x_n.P(x_1,\ldots,x_n) < Q(x_1,\ldots,x_n)$

Construction of $G_{P,Q}$ (order-3 case)

- **Church-encode natural numbers**

  $[n]:nat = \lambda s.\lambda z.s^n z$

  (where $nat=(o\to o)\to o\to o$)

- **Construct $Test_<$: $nat\to nat\to o$ such that:**

  $m<n$ iff $Test_<$ m n is not AST

- **Let $G_{P,Q}$ run $Test_<$ $(P(x_1,\dots,x_n))$ $(Q(x_1,\dots,x_n))$ for all $x_1,\dots,x_n$:**

  $S = TestAll\ 0 \dots 0$

  $$TestAll\ x_1 \dots x_n = Test_<\ (P(x_1,\dots,x_n))\ (Q(x_1,\dots,x_n)) \oplus TestAll\ (x_1+1)\ x_2 \dots x_n \oplus \dots \oplus TestAll\ x_1 \dots x_{n-1}\ (x_n+1)$$

Construction of $G_{P,Q}$ (order-3 case)

- Church-encode natural numbers
  - $[n] = \lambda s.\lambda z.s^n\, z : (o \rightarrow o) \rightarrow o \rightarrow o$
- Construct $Test_< : nat \rightarrow nat \rightarrow o$ s.t.

  m<n iff Church m n is not AST
- Let $G_{P,Q}$ run $Test_< (P(x_1,\ldots,x_n))\ (Q(x_1,\ldots,x_n))$ for all $x_1,\ldots,x_n$:

  $S = TestAll\ 0\ \ldots.\ 0$

  $TestAll\ x_1\ \ldots\ x_n = Test_< (P(x_1,\ldots,x_n))\ (Q(x_1,\ldots,x_n))$
  $\oplus\ TestAll\ (x_1+1)\ x_2\ \ldots\ x_n \oplus \ldots$
  $\oplus\ TestAll\ x_1\ \ldots\ x_{n-1}\ (x_n+1)$

Construction of Test$_<$ m n

- Recall:
  - F e where F x = x $\oplus_p$ F(F x) is non-AST iff $p<0.5$
- Parametrize F by $\oplus_p$:
  - F’ g e where F’ g x = g x (F’ g (F’ g x)) is non-AST if g: o→o→o chooses the first branch with prob. <0.5
- Define Test$_<$ by:
  - Test$_<$ m n = F’ $\boxed{\text{(LT m n)}}$ e
  - LT m n x y = $((H\ x)^m\ y) \oplus_{0.5} ((H\ y)^n\ x)$
  - H x y = x $\oplus_{0.5}$ y

Chooses the first branch with prob. < 0.5 iff m<n

Chooses x with prob. $1 – 1/2^m$

Chooses x with prob. $1/2^n$

$$\tfrac{1}{2}\,(1-1/2^m) + \tfrac{1}{2}(1/2^n) = \tfrac{1}{2} + \tfrac{1}{2}\,(1/2^n - 1/2^m) < \tfrac{1}{2} \text{ iff } m<n$$

$G_{P,Q}$ for order-3 case

S = TestAll Zero .... Zero

TestAll x_1 ... x_n = Test$_<$ (P x_1 ... x_n) (Q x_1 ... x_n)

$\oplus$ TestAll (x_1+1) x_2 ... x_n $\oplus$ ...

$\oplus$ TestAll x_1 ... x_{n-1} (x_n+1)

Run **Test$_<$ (P x_1 ... x_n) (Q x_1 ... x_n)** for all **x_1,...,x_n**

Test$_<$ m n = F' (LT m n) e

LT m n x y = ((H x)^m y) $\oplus_{0.5}$ ((H y)^n x)

H x y = x $\oplus_{0.5}$ y

non-AST iff m<n

Zero s z = z

P x_1 ... x_n = ...

Q x_1 ... x_n = ...

Define natural numbers and polynomials using Church encoding

$G_{P,Q}$ is non-AST iff P(x_1,...,x_n)<Q(x_1,...,x_n) is satisfiable

$G_{P,Q}$ for order-3 case

S = TestAll Zero .... Zero

TestAll x_1 … x_n = Test$_<$ (P x_1 … x_n) (Q x_1 … x_n)
⊕ TestAll (x_1+1) x_2 … x_n ⊕ …
⊕ TestAll x_1 … x_{n-1} (x_n+1)

Run **Test$_<$ (P x_1 … x_n) (Q x_1 … x_n)** for all **x_1,…,x_n**

Test$_<$ m n = F' (LT m n) e

Does not work for order-2 case,
because Church numerals are order-2 functions

Zero s z = z

P x_1 … x_n = …

Q x_1 … x_n = …

Define natural numbers and polynomials using Church encoding

$G_{P,Q}$ is non-AST iff P(x_1,…,x_n)<Q(x_1,…,x_n) is satisfiable

Ideas for Order-2 Case

- Represent natural numbers as order-1 probabilistic functions

  $[n] = \lambda x.\lambda y.\ x \oplus_{p(n)} y$ where $p(n)=1 - 1/2^n$

  $\text{Zero}\ x\ y = y$ $\quad$ $\text{Succ}\ n\ x\ y = x \oplus_{1/2} (n\ x\ y)$

  Prob("Succ n x y chooses y")

  $= 1/2 \cdot 1/2^n = 1/2^{n+1}$

Ideas for Order-2 Case

- Represent natural numbers as order-1 probabilistic functions

  $[n] = \lambda x.\lambda y.\ x \oplus_{p(n)} y$ where $p(n)=1 - 1/2^n$

  $\text{Zero } x\ y = y$ $\quad$ $\text{Succ } n\ x\ y = x \oplus_{1/2} (n\ x\ y)$

  $\text{Add } m\ n\ x\ y = m\ x\ (n\ x\ y)$

  Prob("Add m n x y chooses y")

  $= 1/2^m \cdot 1/2^n = 1/2^{m+n}$

Ideas for Order-2 Case

- Represent natural numbers as order-1 probabilistic functions

$[n] = \lambda x.\lambda y.\ x \oplus_{p(n)} y$ where $p(n)=1 - 1/2^n$

Zero x y = y Succ n x y = $x \oplus_{1/2}$ (n x y)

Add m n x y = m x (n x y)

S = TestAll Zero .... Zero

TestAll $x_1 \ldots x_n$ = $\text{Test}_<$ (P $x_1 \ldots x_n$) (Q $x_1 \ldots x_n$)

$\oplus$ TestAll (x_1+1) $x_2 \ldots x_n \oplus \ldots \oplus$ TestAll $x_1 \ldots x_{n-1}$ (x_n+1)

$\text{Test}_<$ m n = F' (LT m n) e

LT m n x y = (m x y) $\oplus_{0.5}$ (n y x)

Ideas for Order-2 Case

- Represent natural numbers as order-1 probabilistic functions

$[n] = \lambda x.\lambda y.\ x \oplus_{p(n)} y$ where $p(n)=1 - 1/2^n$

Zero $x\ y = y$ Succ $n\ x\ y = x \oplus_{1/2} (n\ x\ y)$

Add $m\ n\ x\ y = m\ x\ (n\ x\ y)$

S = TestAll Zero .... Zero

TestAll $x_1 \dots x_n$ = $\text{Test}_<$ (P $x_1 \dots x_n$) (Q $x_1 \dots x_n$) $\oplus$ TestAll $(x_1+1)\ x_2 \dots x_n \oplus \dots \oplus$ TestAll $x_1 \dots x_{n-1}\ (x_n+1)$

$\text{Test}_<$ m n = F' (LT m n) e

LT m n x y = (m x y) $\oplus_{0.5}$ (n y x)

This cannot be defined!

Ideas for Order-2 Case

- Represent natural numbers as order-1 probabilistic functions

  $[n] = \lambda x.\lambda y.\ x \oplus_{p(n)} y$ where $p(n)=1 - 1/2^n$

  Zero x y = y   Succ n x y = x $\oplus_{1/2}$ (n x y)

  Add m n x y = m x (n x y)

- Pass around the values of $x_1^{k1} \ldots x_n^{kn}$ (for $k_1 \leq d_1, \ldots, k_n \leq d_n$, where d_i is the degree of P+Q in x_i)

$v_{k1,\ldots,kn}$ holds the value of $x_1^{k1} \ldots x_n^{kn}$

S = TestAll One Zero ..... Zero

TestAll $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$=

$\mathrm{Test}_<$ (P $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$) (Q $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$)

$\oplus$ TestAll ($\mathrm{Inc}_1\ \vec{v}$) $\oplus \ldots \oplus$ TestAll ($\mathrm{Inc}_n\ \vec{v}$)

Inc_i updates $v_{k1,\ldots,kn}$ to the value of $x_1^{k1} \ldots (x_i+1)^{ki}\ x_n^{kn}$

Ideas for Order-2 Case

- Pass around the values of $x_1^{k1} \ldots x_n^{kn}$ (for $k_1 \leq d_1, \ldots, k_n \leq d_n$, where d_i is the degree of P+Q in x_i)

Example: P $=x^2y$, Q $= x^2+y$

TestAll $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$ =

v_{jk} holds the value of $x^j y^k$

$\quad$ Test$_<$ (P $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$) (Q $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$)

$\quad \oplus$ TestAll (Inc$_{x,00}$ $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$) ...
(Inc$_{x,21}$ $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$)

$\quad \oplus$ TestAll (Inc$_{y,00}$ $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$) ...
(Inc$_{y,21}$ $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$)

P $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21} = v_{21}$ $\quad$ Q $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$ = Add $v_{20}\ v_{01}$

Inc$_{x,00}$ $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$ = One

...

Inc$_{x,21}$ $v_{00}\ v_{01}\ v_{10}\ v_{11}\ v_{20}\ v_{21}$ = Add v_{21} (Add v_{11} (Add $v_{11}\ v_{01}$))

(since $(x+1)^2y = x^2y+2xy+y$)

Construction of $G_{P,Q}$

S = TestAll One Zero .... Zero

TestAll $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$ =

$\mathrm{Test}_{<}$ **(P** $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$**) (Q** $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$**)**

$\oplus$ **TestAll** $(\mathrm{Inc}_1\ \vec{v}) \oplus \ldots \oplus$ **TestAll** $(\mathrm{Inc}_n\ \vec{v})$

Run $\mathrm{Test}_{<}$ $(P(x_1,\ldots,x_n))$ $(Q(x_1,\ldots,x_n))$ for all $x_1,\ldots,x_n$

$\mathrm{Test}_{<}$ **m n = F' (LT m n) e**

LT m n x y = (m x y) $\oplus_{0.5}$ **(n y x)**

F' g x = g x (F' g (F' g x))

non-AST iff m<n

Zero x y = y

Succ n x y = x $\oplus_{1/2}$ **(n x y)**

Add m n x y = m x (n x y)

Encode natural numbers as order-1 probabilistic functions

P $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$ **= ...**

$\mathrm{Inc}_{i,\ k1,\ldots,kn}$ $v_{0,\ldots,0} \ldots v_{d1,\ldots,dn}$ **= ...**

Can be expressed as linear combinations of v

Summary of Undecidability Results

- **The following decision problem is undecidable**
  - **Input: order-2 pHORS G**
  - **Output: whether TP(G)=1.**
- **More precisely, the following sets are not recursively enumerable (for r$\in$(0,1])**
  - **$\mathcal{G}_{=r}$ = {G: order-2 pHORS | TP(G)=r}**
  - **$\mathcal{G}_{\geq r}$ ={G: order-2 pHORS | TP(G)$\geq$r}**

  cf. $\mathcal{G}_{>r}$ ={G: order-2 pHORS | TP(G)>r} is r.e.

  open: whether $\mathcal{G}_{<r}$ and $\mathcal{G}_{\leq r}$ are r.e.

Note: A hope remains on approximate computation:

Input: G, a rational number ε>0

Output: r such that |TP(G)-r| < ε

Outline

- pHORS: probabilistic extension of higher-order recursion schemes
- Termination Problems
- Undecidability of AST of order-2 pHORS
- Fixpoint characterization of termination probabilities
  - Order-n characterization
  - Order-(n-1) characterization
- Approximate computation of termination probabilities
- Conclusion

Order-n (Least) Fixpoint Characterization of Termination Prob. for Order-n pHORS

◆ **Just replace**

e (termination) with 1

Ω (divergence) with 0

$t_L \oplus_p t_R$ with $p[t_L] + (1\text{-}p)[t_R]$

Order-1 pHORS:
$S = F\ e$
$F\ x = x \oplus_{1/3} F\ (F\ x)$

Order-1 fixpoint equations
$S = F\ 1$
$F\ x = 1/3 \cdot x + 2/3 \cdot F\ (F\ x)$

The least solution: $S = 0.5$, $F(x) = 0.5\ x$

Order-(n-1) Fixpoint Characterization?

(cf. Order-0 equations for termination probabilities of probabilistic PDS [Etessami+; Esparza+])

- **Easy in order-1 case (cf. probabilistic pushdown)**

Order-1 pHORS:

$S = F\ e$

$F\ x = x \oplus_{1/3} F\ (F\ x)$

Order-0 fixpoint equations

$S = F_0 + F_1 \cdot 1$

$F_1 = 1/3 \cdot 1 + 2/3 \cdot F_1 \cdot F_1$

$F_0 = 1/3 \cdot 0 + 2/3 \cdot (F_0 + F_1 \cdot F_0)$

F_0: prob. of terminating without using x

F_1: prob. of using x

Order-1 function of arity k can be expressed as

$(F_0, F_1, \ldots, F_k) \in \mathrm{Real}^{k+1}$

where F_0 : prob. of terminating without using any arguments

F_i: prob. of using the i-th argument

Order-(n-1) Fixpoint Characterization: General Case

- How to translate an order-2 function F of type $(o \to o) \to o$?
  - Naive solution: as a function from
    $(g_0, g_1) \in \mathrm{Real}^2$ (g_0: prob. that the argument g terminates,
    g_1: prob. that g uses its argument)
    to the termination probability of F(g).

  => Does not work when g contains a variable:

  e.g. J x = F (H x)

  There is no way to calculate the prob. that J uses x from the translation of F.

Order-(n-1) Fixpoint Characterization: General Case

- **In a context where order-0 variables $x_1,\ldots,x_k$ are visible, a function $\lambda y_1\ldots y_\ell.\ \lambda z_1\ldots z_m.\ t$ is translated to:**

order($\mathbf{y}_\ell$)> 0

order-0

$$(\mathbf{t_0}, \underbrace{\mathbf{t_1}, \ldots, \mathbf{t_m}}, \underbrace{\mathbf{t_{m+1}}, \ldots, \mathbf{t_{m+k}}}, \mathbf{t_{m+k+1}})$$

Returns the prob. of reaching the current reachability target (given the translation of y's)

Returns prob. of reaching z_i

Returns prob. of reaching x_j

Returns prob. of reaching a "fresh" variable t does not know

Order-(n-1) Fixpoint Characterization: General Case

- **In a context where order-0 variables $x_1,\ldots,x_k$ are visible, a function $\lambda y_1 \ldots y_\ell .\ \lambda z_1 \ldots z_m .\ t$ is translated to:**

$$(t_0, \underbrace{t_1, \ldots, t_m}, \underbrace{t_{m+1}, \ldots, t_{m+k}}, t_{m+k+1})$$

Returns the prob. of reaching the current reachability target

Returns prob. of reaching z_i

Returns prob. of reaching x_j

Returns prob. of reaching a "fresh" variable t does not know

e.g. $\lambda y.\lambda z.\ y\ (x \oplus_{1/3} z)$ is translated to:

$$(\lambda(y_0,y_1,y_2).y_0,\ \lambda(y_0,y_1,y_2).2/3\cdot y_1,\ \lambda(y_0,y_1,y_2).1/3\cdot y_1,\ \lambda(y_0,y_1,y_2).y_2\ )$$

Translation Relation for Order-(n-1) Fixpoint Characterization

$$\Gamma; \overbrace{x_1,\ldots,x_k}^{\text{order-0 variables}} \vdash t: \kappa_1 \to \ldots \to \kappa_\ell \to o^m \to o$$

$$\hookrightarrow (\underset{\text{prob. of reaching the current reachability target}}{t_0}, \underbrace{t_1, \ldots, t_m}_{\text{prob. of reaching each order-0 argument}}, \underbrace{t_{m+1}, \ldots, t_{m+k}}_{\text{prob. of reaching } x_j}, \underset{\text{prob. of reaching a “fresh” variable t does not know}}{t_{m+k+1}})$$

Translation Rule for Ω (divergence)

$$\Gamma;\ \overbrace{x_1,\ldots,x_k}^{\text{order-0 variables}} \vdash \Omega : \kappa_1 \to \ldots \to \kappa_\ell \to o^m \to o$$

$$\hookrightarrow (0,\ \underbrace{t_1, \ldots, t_m,}_{}\ \ \underbrace{0, \ldots, 0\,,}_{}\ \ 0\,)$$

prob. of reaching the current reachability target

prob. of reaching each order-0 argument

prob. of reaching x_j

prob. of reaching a “fresh” variable t does not know

Translation Rule for Order-0 Variables

$$\Gamma; \overbrace{x_1,\ldots,x_k}^{\text{order-0 variables}} \vdash x_i : \kappa_1 \to \ldots \to \kappa_\ell \to o^m \to o$$

$$\hookrightarrow (0, \underbrace{t_1, \ldots, t_m}, \underbrace{0^{i-1}, 1, 0^{k-i}}, 0)$$

prob. of reaching the current reachability target

prob. of reaching each order-0 argument

prob. of reaching x_j

prob. of reaching a “fresh” variable t does not know

Translation Rule for Variables

$$\Gamma; \overbrace{x_1,\ldots,x_k}^{\text{order-0 variables}} \vdash y: \kappa_1 \to \ldots \to \kappa_\ell \to o^m \to o$$

$$\hookrightarrow (y_0, \underbrace{y_1, \ldots, y_m}, \underbrace{y_{m+1}, \ldots, y_{m+1}}, y_{m+1})$$

prob. of reaching the current reachability target

prob. of reaching each order-0 argument

prob. of reaching x_j

prob. of reaching a “fresh” variable t does not know

Translation Rule for Applications (order-1 case)

$$\dfrac{\begin{array}{l}\Gamma;\, \overbrace{x_1,\ldots,x_k}^{\text{order-0 variables}} \vdash\ s: o^{m+1}\rightarrow o \\ \quad \hookrightarrow (s_0,\, s_1,\, \ldots,\, s_{m+1}, s_{m+2}\ \ldots,\ s_{m+k+1}\,,\, s_{m+k+2}\,) \\ \Gamma;\, x_1,\ldots,x_k \vdash\ t: o\ \hookrightarrow (t_0,\, t_1\ \ldots,\ t_k\,,\, t_{k+1}\,)\end{array}}{\begin{array}{l}\Gamma;\, x_1,\ldots,x_k \vdash\ s\ t: o^{m}\rightarrow o \\ \quad \hookrightarrow (s_0 + s_1\cdot t_0\,,\, s_2,\, \ldots,\, s_{m+1}, \\ \qquad\qquad s_{m+2} + s_1\cdot t_1, \ldots,\, s_{m+k+2} + s_1\cdot t_{k+1})\end{array}}$$

Translation Rule for Applications (higher-order case)

$$\begin{array}{l}
\Gamma; \overbrace{x_1,\ldots,x_k}^{\text{order-0 variables}} \vdash\; s:\; \kappa_1 \to \ldots \to \kappa_\ell \to o^m \to o \\
\quad \hookrightarrow (s_0, s_1, \ldots, s_m, s_{m+1}, \ldots, s_{m+k}, s_{m+k+1}) \\
\Gamma; x_1,\ldots,x_k \vdash\; t: \kappa_1 \\
\quad \hookrightarrow (t_0, t_1 \ldots, t_n, t_{n+1}, \ldots, t_{n+k}, t_{n+k+1}) \\
\hline
\Gamma; x_1,\ldots,x_k \vdash\; s\,t: \kappa_2 \to \ldots \to \kappa_\ell \to o^m \to o \\
\quad \hookrightarrow (s_0(t_0, t_1 \ldots, t_n, t_{n+k+1}), \ldots, s_m(t_0, t_1 \ldots, t_n, t_{n+k+1}), \\
\qquad\quad s_{m+1}(t_{n+1}, t_1 \ldots, t_n, t_{n+k+1}), \ldots, \\
\qquad\quad s_{m+k+1}(t_{n+k+1}, t_1 \ldots, t_n, t_{n+k+1}))
\end{array}$$

Translation of Rewriting rules

$$\begin{array}{l}
y_1,\ldots,y_\ell;\ \overbrace{x_1,\ldots,x_k}^{\text{order-0 variables}} \ |\!-\ s : o \ \hookrightarrow (s_0, s_1, \ldots,\ s_k\ , s_{k+1}\ ) \\
y_1,\ldots,y_\ell;\ x_1,\ldots,x_k \ |\!-\ t : o \ \hookrightarrow (t_0, t_1, \ldots,\ t_k\ , t_{k+1}\ ) \\
\hline
\\
\quad F\ y_1,\ldots,y_\ell\ x_1,\ldots,x_k = s \oplus_p t \\
\quad \hookrightarrow \{\ F_0\ (y_{1,0},\ldots)\ \ldots\ (y_{\ell,0},\ldots) = ps_0 + (1\text{-}p)t_0\ , \\
\qquad \ldots, \\
\qquad F_k\ \ (y_{1,0},\ldots)\ \ldots\ (y_{\ell,0},\ldots) = ps_k + (1\text{-}p)t_k\ \}
\end{array}$$

Example

$S' = S\ e\ \Omega \quad S\ x\ y = F\ (C\ x\ y) \quad F\ g = g\ H$

$H\ x = x \oplus_{0.5} \Omega \quad C\ x\ y\ f = (f\ x) \oplus_{0.3} (f\ y)$

(S': o, S: o->o->o, H: o->o, F: ((o->o)->o)->o,
C: o->o->(o->o)->o

$S_0 = F_0(C_0\ 0\ 0,\ C_0\ 0\ 0) \quad S_1=F_0(C_0\ 1\ 0,\ C_0\ 0\ 0)$

$S_2= F_0(C_0\ 0\ 1,\ C_0\ 0\ 0)$

$F_0(g_0, g_1) = g_0(H_0, H_1, H_0)$

$C_0\ \ x_0\ \ y_0\ (f_0, f_1, f_2)\ = 0.3(f_0+f_1 \cdot x_0\ )+0.7(f_0+f_1 \cdot y_0)$

$H_0=0 \quad H_1 = 0.5$

$$S_1=F_0(C_0\ 1\ 0,\ C_0\ 0\ 0)\ = C_0\ 1\ 0\ (H_0, H_1, H_0)$$
$$= 0.3(H_0 + H_1 \cdot 1\ ) + 0.7(H_0 + H_1 \cdot 0\ ) = 0.15$$

Correctness of Fixpoint Characterization

If G $\hookrightarrow \mathcal{E}_G$, then:

TP(G) is the least solution of $\mathcal{E}_G$

(more precisely, TP(G, S e) = lfp($\mathcal{E}_G$)(S_1)).

For any order-n pHORS G,

$\mathcal{E}_G$ (such that G $\hookrightarrow \mathcal{E}_G$) is

- a system of order-(n-1) fixpoint equations; and
- constructible in polynomial time

Outline

- pHORS: probabilistic extension of higher-order recursion schemes
- Termination Problems
- Undecidability of AST of order-2 pHORS
- Fixpoint characterization of termination probabilities
- Approximate computation of termination probabilities (for order-2 pHORS)
- Related work and conclusion

Summary of the Talk So far

- TP(G)$\sim$r is undecidable for $\sim \in \{=, \geq\}$
  - A hope remains on approximate computability:

    Input: G, a rational number $\varepsilon > 0$

    Output: r such that $|TP(G)-r| < \varepsilon$
- TP(G) can be characterized as order-(n-1) fixpoint equations

  (order-1 equations for order-2 pHORS)
  - immediately yields a method for computing a lower bound for TP(G)
    - Given f=F(f), the least solution can be lower-approximated by $F^k(\bot)$
  - how about upper-approximation?

Non-Solution 1:

Upper-approximation of greatest fixpoint

- For f=F(f),

  $\mathrm{lfp}(F) \leq \mathrm{gfp}(F) \leq F^n(\lambda x.1)$,

  but $F^n(\lambda x.1)$ may be too imprecise as an upper bound for lfp(F).

  e.g. For $F = \lambda f.\lambda x.f(x)$,

  $\mathrm{lfp}(F)=\lambda x.0$, but $\mathrm{gfp}(F) = F^n(\lambda x.1)= \lambda x.1$

Non-Solution 2:

Upper-approximation by Polynomials

- Example: $f(x)= 1/3 \cdot x + 2/3 \cdot f(f(x))$
  Template: $f(x) = c_0 + c_1 x$

Sufficient condition for upper-approximation:

$$f(x) \geq 1/3 \cdot x + 2/3 \cdot f(f(x))$$

$$= 1/3 \cdot x + 2/3(c_0 + c_1 (c_0 + c_1 x)),$$

i.e. $c_0 \geq 2/3(c_0 + c_1 c_0)$

$c_1 \geq 1/3 + 2/3 c_1^2$,

yielding $c_1= 1/2$, $c_0=0$, i.e. $f(x)=1/2 x$

Non-Solution 2:
Upper-approximation by Polynomials

- **Imprecise for: $f(x_1, x_2)= x_1 + x_2 \cdot f(x_1, x_2)$ (where x_1, x_2 are constrained by $0\leq x_1+x_2 \leq 1$)**
  - **least solution:**

$$f(x_1, x_2)= \begin{cases} 0 & \text{if } x_1 =0 \\ x_1/(1-x_2) & \text{if } x_1 >0 \end{cases}$$

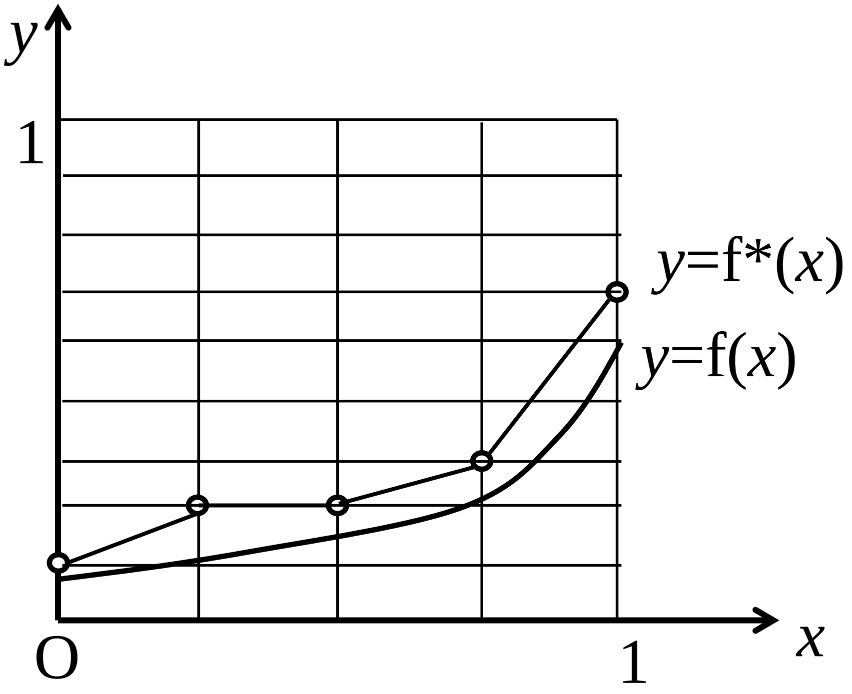
Since $f(x_1, x_2)=1$ for $x_1 = \varepsilon>0$ and $x_2 = 1-\varepsilon$,

$f'(0,1)=1 > f(0,1)$

for any sound polynomial upper-approximation f' of f (due to the continuity of f')

Our Approach: Discretization (à la Finite Element Method)

- **Decompose [0,1] into a finite number of intervals, and use a step-wise linear function f* as an upper bound**
- **f* is determined by a finite number of points (x_0, y_0), ... (x_n, y_n)**
- **Sufficient condition for sound approximation:**
  $y_i \geq f^*(x_i)$ for i=0,...,n
- **y_i can be computed by:**
  - **using decidability of theories of real arithmetics; or**
  - **discretization of codomain**



Example

$f(x) = 0.25x + 0.75f(f(x))$

- discrete points:
  $(0, y_0)$, $(0.5, y_1)$, $(1,y_2)$
- constraints:

  $y_0 \geq 0.25\cdot 0 + 0.75f^*(f^*(0))$, $y_1 \geq 0.25\cdot 0.5 + 0.75\, f^*(f^*(0.5))$,
  $y_2 \geq 0.25\cdot 1 + 0.75f^*(f^*(1))$

  $$\text{where } f^*(x) = \begin{cases} (1-2x)y_0 + 2xy_1 & \text{if } x \in [0, 0.5] \\ (2-2x)y_1 + (2x-1)y_2 & \text{if } x \in (0.5, 1] \end{cases}$$

- with discretization of y_i to $\{0, 0.25, 0.5, 0.75, 1\}$:

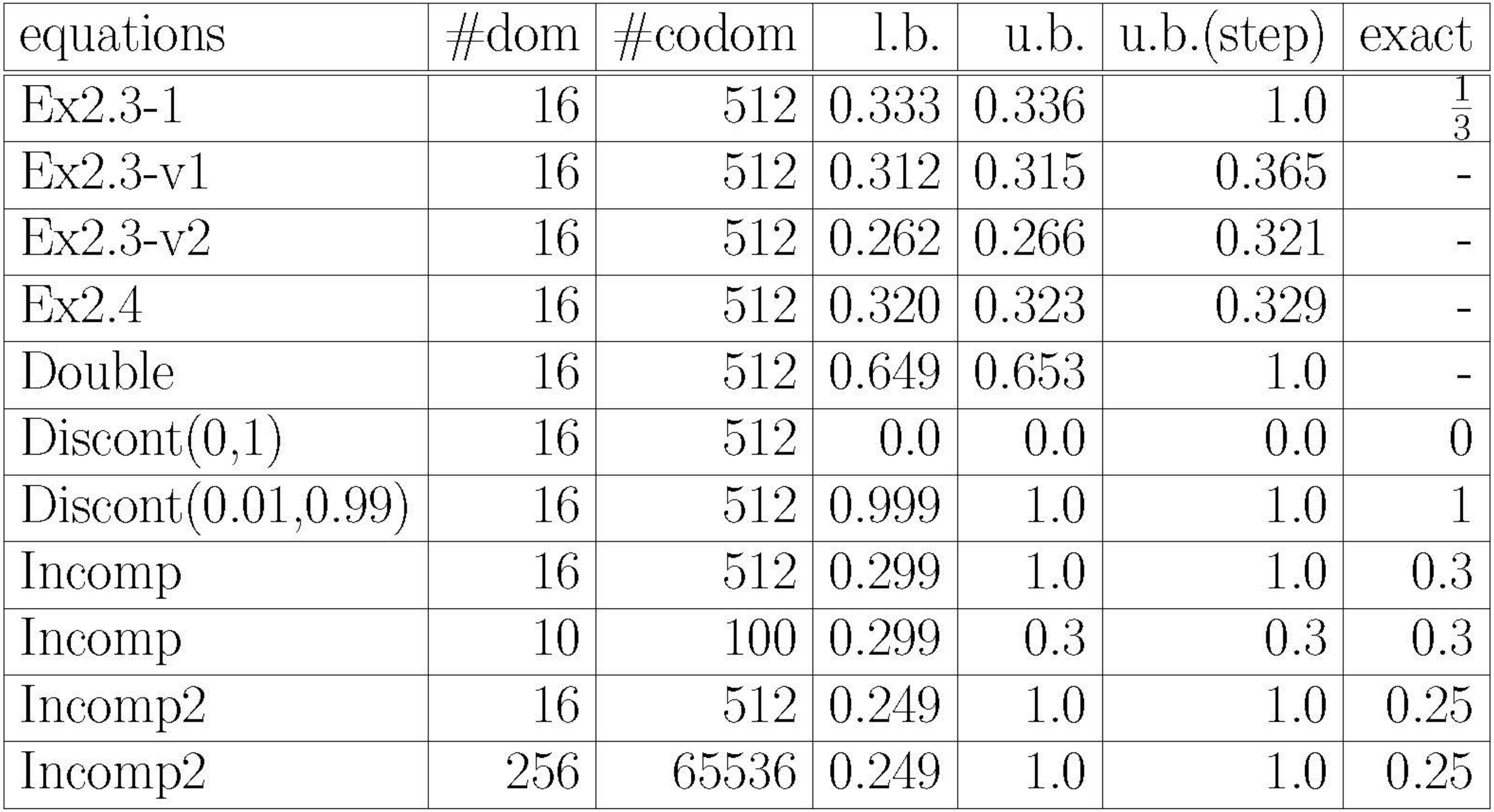
  $(y_{0,}\ y_{1,}\ y_2)^{(0)} = (0, 0, 0)$

  $(y_{0,}\ y_{1,}\ y_2)^{(1)} = (0, 0.25, 0.25)$

  $(y_{0,}\ y_{1,}\ y_2)^{(2)} = (y_{0,}\ y_{1,}\ y_2)^{(3)} = (0, 0.25, 0.5)$

  yielding $f^*(x)=0.5x$ (exact solution: $f(x)=1/3\cdot x$)

Experimental Results



equations	#dom	#codom	l.b.	u.b.	u.b.(step)	exact
Ex2.3-1	16	512	0.333	0.336	1.0	$\frac{1}{3}$
Ex2.3-v1	16	512	0.312	0.315	0.365	-
Ex2.3-v2	16	512	0.262	0.266	0.321	-
Ex2.4	16	512	0.320	0.323	0.329	-
Double	16	512	0.649	0.653	1.0	-
Discont(0,1)	16	512	0.0	0.0	0.0	0
Discont(0.01,0.99)	16	512	0.999	1.0	1.0	1
Incomp	16	512	0.299	1.0	1.0	0.3
Incomp	10	100	0.299	0.3	0.3	0.3
Incomp2	16	512	0.249	1.0	1.0	0.25
Incomp2	256	65536	0.249	1.0	1.0	0.25

Experimental Results

$f(x_1, x_2) = x_1 + x_2 \cdot f(x_1, x_2)$

The exact solution is:

$$f(x_1, x_2) = \begin{cases} 0 & \text{if } x_1 = 0 \\ x_1/(1-x_2) & \text{if } x_1 > 0,\ x_2 < 1 \end{cases}$$

equati					p)	exact
Ex2.3-					.0	$\frac{1}{3}$
Ex2.3-					65	-
Ex2.3-					21	-
Ex2.4					29	-
Double	16	512	0.649	0.653	1.0	-
Discont(0,1)	16	512	0.0	0.0	0.0	0
Discont(0.01,0.99)	16	512	0.999	1.0	1.0	1
Incomp	16	512	0.299	1.0	1.0	0.3
Incomp	10	100	0.299	0.3	0.3	0.3
Incomp2	16	512	0.249	1.0	1.0	0.25
Incomp2	256	65536	0.249	1.0	1.0	0.25

Experimental Results

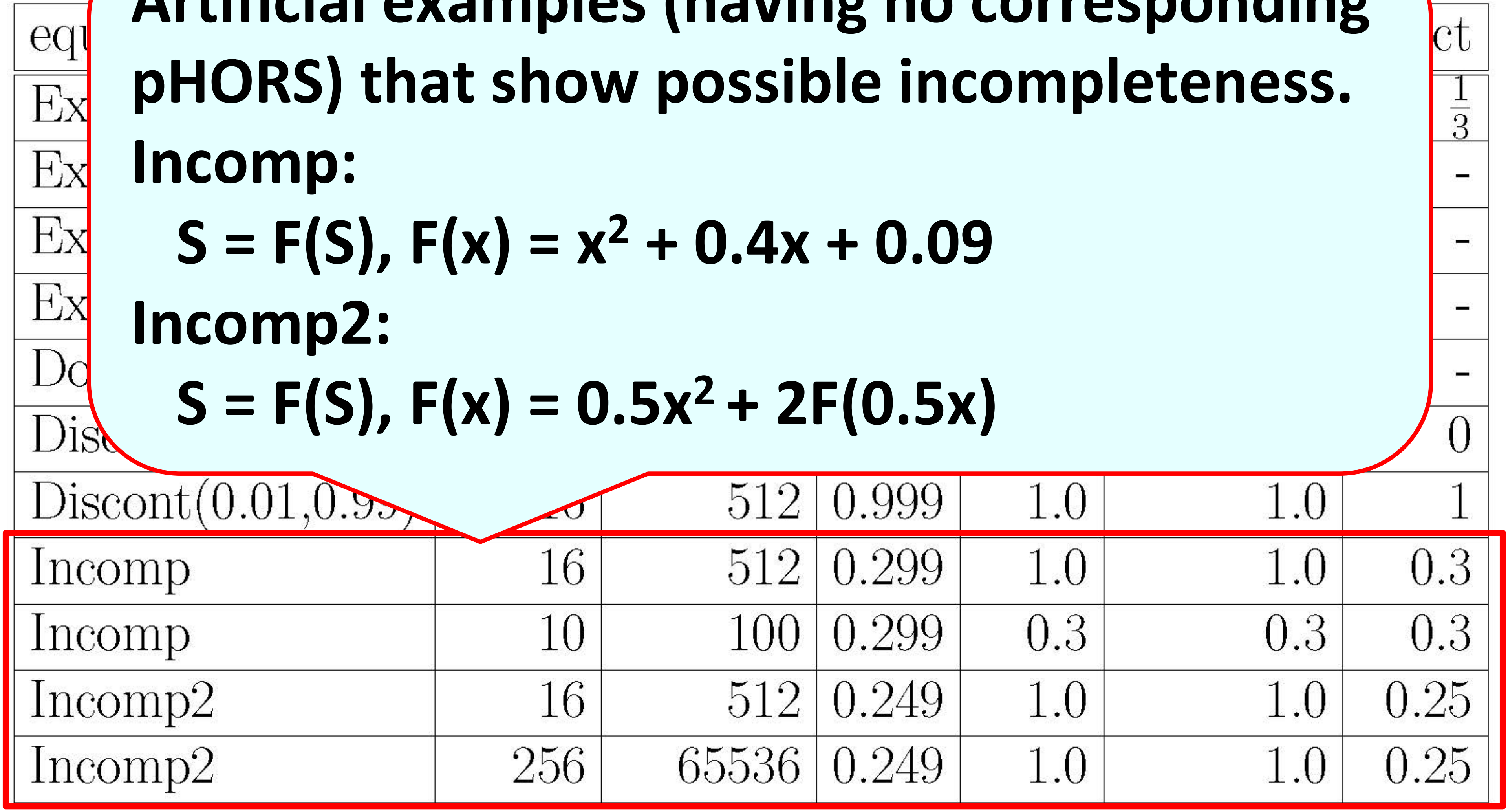
Artificial examples (having no corresponding pHORS) that show possible incompleteness.

Incomp:

$S = F(S),\ F(x) = x^2 + 0.4x + 0.09$

Incomp2:

$S = F(S),\ F(x) = 0.5x^2 + 2F(0.5x)$



equ						ct
Ex						$\frac{1}{3}$
Ex						-
Ex						-
Ex						-
Do						-
Dis						0
Discont(0.01,0.9		512	0.999	1.0	1.0	1
Incomp	16	512	0.299	1.0	1.0	0.3
Incomp	10	100	0.299	0.3	0.3	0.3
Incomp2	16	512	0.249	1.0	1.0	0.25
Incomp2	256	65536	0.249	1.0	1.0	0.25

Experimental Results

Artificial examples (having no corresponding pHORS) that show possible incompleteness.

Incomp:

$S = F(S),\ F(x) = x^2 + 0.4x + 0.09$

$S \geq F(S)$ iff $(S-0.3)^2 \leq 0$ iff $S=0.3$

Incomp2:

$S = F(S),\ F(x) = 0.5x^2 + 2F(0.5x)$

equ						ct
Ex						$\frac{1}{3}$
Ex						-
Ex						-
Ex						-
Do						-
Dis						0
Discont(0.01,0.9		512	0.999	1.0	1.0	1
Incomp	16	512	0.299	1.0	1.0	0.3
Incomp	10	100	0.299	0.3	0.3	0.3
Incomp2	16	512	0.249	1.0	1.0	0.25
Incomp2	256	65536	0.249	1.0	1.0	0.25

Outline

- pHORS: probabilistic extension of higher-order recursion schemes
- Termination Problems
- Undecidability of AST of order-2 pHORS
- Fixpoint characterization of termination
- Approximate computation of termination probabilities
- Related Work and Conclusion

Related Work

- Model checking of probabilistic pushdownsystems/recursive Markov chains [Esparza+ 04, Etessami&Yannakakis 04,...]
  - termination probabilities as polynomial equations (special case of our order-(n-1) fixpoint characterization)
  - studies of linear-time/branching-time model checking problems
- Model checking of (non-probabilistic) HORS [Knapik+02, Ong06, Kobayashi09, ...]
- Type-based characterization of termination probabilities of probabilistic functional programs [Dal Lago&Grellois, Breuvart&Dal Lago]
  - do not provide a method for precise approximation

Conclusion

- pHORS as a model of probabilistic functional programs
- Undecidability of AST of order-2 pHORS
- Order-(n-1) Fixpoint Characterization of Termination Probability of order-n pHORS
- Sound (but possibly incomplete) method for approximate computation of TP(G) for order-2 pHORS

Future Work

- **Settling the question of approximate computability of TP(G) with arbitrary precision**

  Input: G, a rational number $\varepsilon>0$

  Output: r such that $|TP(G)-r| < \varepsilon$

  (equivalent to the question of whether $\mathcal{G}_{<r}$ ={G: order-2 pHORS | TP(G)<r} is r.e.)

- **Practical method for approximate computation of TP(G) for pHORS of arbitrary order**
- **Model checking of pHORS**