Probabilistic Program Analysis

Probablistic Abstract Interpretation

Alessandra Di Pierro
University of Verona, Italy
alessandra.dipierro@univr.it

Herbert Wiklicky
Imperial College London, UK
herbert@doc.ic.ac.uk

Approximation and Correctness

Data-flow analyses can be re-formulated in a different scenario where correctness is guaranteed by construction.

Classically, the theory of Abstract Interpretation allows us to

- construct simplified (computable) abstract semantics
- construct approximate solutions
- obtain the correctness of the approximate solution by construction.

Notions of Approximation

In order theoretic structures we are looking for
Safe Approximations

$$s^* \sqsubseteq s \quad \text{or} \quad s \sqsubseteq s^*$$

In quantitative, vector space structures we want
Close Approximations

$$\|s - s^*\| = \min_x \|s - x\|$$

Abstract Interpretation

Some problems may be have too costly solutions or be uncomputable on a concrete space (complete lattice).

- Solution: find abstract descriptions on which computations are easier, then relate the concrete and abstract solutions.
- Basic idea: analyse the program using an *abstract semantics* which only registers those aspects of the program that are relevant for the specific analysis.
- Example: for the parity analysis of the factorial program (see previous lecture), we used as an abstract domain the lattice

$$\bot \leq \textbf{even}, \textbf{odd} \leq \top$$

which captures the abstract property we were interested in.

Abstract Interpretation

The standard theory of *Abstract Interpretation* was introduced by Cousot& Cousot in 1977.
It states that the correctness of an abstract semantics is guaranteed by establishing a *categorical adjunction* between the concrete and abstract properties (lattices).

Definition

Let $\mathcal{C} = (\mathcal{C}, \leq)$ and $\mathcal{D} = (\mathcal{D}, \sqsubseteq)$ be two partially ordered set. If there are two functions $\alpha : \mathcal{C} \rightarrow \mathcal{D}$ and $\gamma : \mathcal{D} \rightarrow \mathcal{C}$ such that for all $c \in \mathcal{C}$ and all $d \in \mathcal{D}$:

$$c \leq_{\mathcal{C}} \gamma(d) \text{ iff } \alpha(c) \sqsubseteq d,$$

then $(\mathcal{C}, \alpha, \gamma, \mathcal{D})$ form a Galois connection.

Galois Connections

Definition

Let $\mathcal{C} = (\mathcal{C}, \leq_{\mathcal{C}})$ and $\mathcal{D} = (\mathcal{D}, \leq_{\mathcal{D}})$ be two partially ordered sets with two order-preserving functions $\alpha : \mathcal{C} \mapsto \mathcal{D}$ and $\gamma : \mathcal{D} \mapsto \mathcal{C}$. Then $(\mathcal{C}, \alpha, \gamma, \mathcal{D})$ form a Galois connection iff

(i) $\alpha \circ \gamma$ is reductive i.e. $\forall d \in \mathcal{D}, \alpha \circ \gamma(d) \leq_{\mathcal{D}} d$,

(ii) $\gamma \circ \alpha$ is extensive i.e. $\forall c \in \mathcal{C}, c \leq_{\mathcal{C}} \gamma \circ \alpha(c)$.

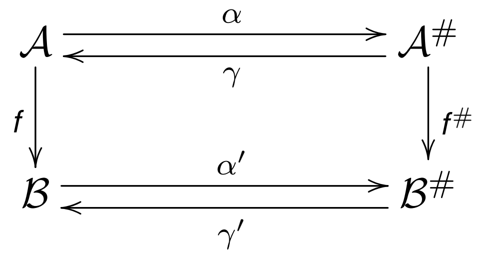
Proposition

Let $(\mathcal{C}, \alpha, \gamma, \mathcal{D})$ be a Galois connection. Then α and γ are quasi-inverse, i.e.

(i) $\alpha \circ \gamma \circ \alpha = \alpha$

(ii) $\gamma \circ \alpha \circ \gamma = \gamma$

General Construction



Correct approximation:

$$\alpha' \circ f \leq_\# f^\# \circ \alpha.$$

Induced semantics:

$$f^\# = \alpha \circ f \circ \gamma.$$

Probabilistic Abstraction Domains

A probabilistic domain is essentially a vector space which represents the distributions *Dist*(S) on the state space S of a probabilistic transition system, i.e. for finite state spaces

$$\mathcal{V}(S) = \{ (v_s)_{s \in S} \mid v_s \in \mathbb{R}\}.$$

The notion of *norm* is essential for our treatment; we will consider normed vector spaces.
In the finite setting we can identify $\mathcal{V}(S)$ with the Hilbert space $\ell^2(S)$.

Norm and Operator Norm

A norm on a vector space $\mathcal{V}$ is a map $\|.\| : \mathcal{V} \mapsto \mathbb{R}$ such that for all $v, w \in \mathcal{V}$ and $c \in \mathbb{C}$:

- $\|v\| \geq 0$,
- $\|v\| = 0 \Leftrightarrow v = o$,
- $\|cv\| = |c|\|v\|$,
- $\|v + w\| \leq \|v\| + \|w\|$,

with $o \in \mathcal{V}$ the zero vector.

We can always use a norm to define a topology on a vector space via the distance function $d(v, w) = \|v - w\|$.

$$\|\mathbf{M}\| = \sup_{v \in \mathcal{V}} \frac{\|\mathbf{M}(v)\|}{\|v\|} = \sup_{\|v\|=1} \|\mathbf{M}(v)\|.$$

Generalised Inverse

Definition

Let $\mathcal{C}$ and $\mathcal{D}$ be two finite-dimensional vector spaces and $\mathbf{A} : \mathcal{C} \to \mathcal{D}$ a linear map. Then the linear map $\mathbf{A}^\dagger = \mathbf{G} : \mathcal{D} \to \mathcal{C}$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$ iff

(i) $\mathbf{A} \circ \mathbf{G} = \mathbf{P}_A$,

(ii) $\mathbf{G} \circ \mathbf{A} = \mathbf{P}_G$,

where $\mathbf{P}_A$ and $\mathbf{P}_G$ denote orthogonal projections onto the ranges of $\mathbf{A}$ and $\mathbf{G}$.

Least Squares Solutions

Definition

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Then $\mathbf{u} \in \mathbb{R}^n$ is called a least squares solution to $\mathbf{Ax} = \mathbf{b}$ if

$$\|\mathbf{Au} - \mathbf{b}\| \leq \|\mathbf{Av} - \mathbf{b}\|, \text{ for all } \mathbf{v} \in \mathbb{R}^n.$$

Theorem

Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{b} \in \mathbb{R}^m$. Then $\mathbf{A}^\dagger\mathbf{b}$ is the minimal least squares solution to $\mathbf{Ax} = \mathbf{b}$.

Extraction Functions

An extraction function $\eta : C \mapsto D$ is a mapping from a set of values to their descriptions in D.

Proposition

Given an extraction function $\eta : \mathcal{C} \mapsto \mathcal{D}$, the quadruple $(\mathcal{P}(\mathcal{C}), \alpha_\eta, \gamma_\eta, \mathcal{P}(\mathcal{D}))$ is a Galois connection with α_η and γ_η defined by:

$$\alpha_\eta(C') = \{\eta(c) \mid c \in C'\} \text{ and } \gamma_\eta(D') = \{v \mid \eta(v) \in D'\}$$

Vector Space Lifting

Free vector space construction on a set S:

$$\mathcal{V}(S) = \{\sum x_s s \mid x_s \in \mathbb{R}, s \in S\}$$

An obvious way to lift an extraction function to a linear map between vector spaces is to construct the free vector spaces on $\mathcal{C}$ and $\mathcal{D}$ and define:

Vector Space lifting: $\vec{\alpha} : \mathcal{V}(\mathcal{C}) \rightarrow \mathcal{V}(\mathcal{D})$

$$\vec{\alpha}(p_1 \cdot \vec{c}_1 + p_2 \cdot \vec{c}_2 + \ldots) = p_i \cdot \eta(c_1) + p_2 \cdot \eta(c_2) \ldots$$

Support Set: **supp** : $\mathcal{V}(\mathcal{C}) \rightarrow \mathcal{P}(\mathcal{C})$

$$\mathbf{supp}(\vec{x}) = \{c_i \mid \langle c_i, p_i \rangle \in \vec{x} \text{ and } p_i \neq 0\}$$

Relation with Classical Abstractions

Lemma

Let $\vec{\alpha}$ be a probabilistic abstraction function and let $\vec{\gamma}$ be its Moore-Penrose pseudo-inverse.

Then $\vec{\gamma} \circ \vec{\alpha}$ is extensive with respect to the inclusion on the support sets of vectors in $\mathcal{V}(\mathcal{C})$, i.e. $\forall \vec{x} \in \mathcal{V}(\mathcal{C})$,

$$\mathbf{supp}(\vec{x}) \subseteq \mathbf{supp}(\vec{\gamma} \circ \vec{\alpha}(\vec{x})).$$

Analogously we can show that $\vec{\alpha} \circ \vec{\gamma}$ is reductive. Therefore,

Proposition

$(\vec{\alpha}, \vec{\gamma})$ form a Galois connection wrt the support sets of $\mathcal{V}(\mathcal{C})$ and $\mathcal{V}(\mathcal{D})$, ordered by inclusion.

Examples of Lifted Abstractions

Parity Abstraction operator on $\mathcal{V}(\{1, \ldots, n\})$ (with n even):

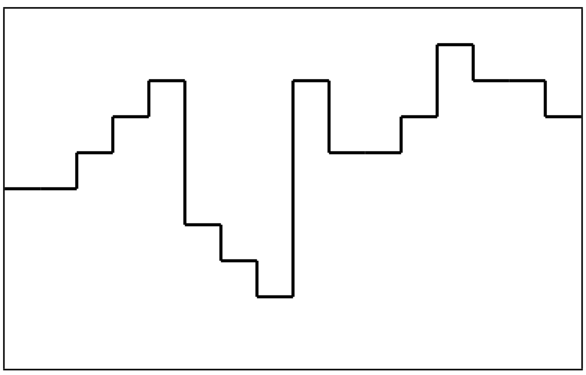
$$\mathbf{A}_p = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix} \qquad \mathbf{A}_p^\dagger = \begin{pmatrix} \frac{2}{n} & 0 & \frac{2}{n} & 0 & \ldots & 0 \\ 0 & \frac{2}{n} & 0 & \frac{2}{n} & \ldots & \frac{2}{n} \end{pmatrix}$$

Sign Abstraction operator on $\mathcal{V}(\{-n, \ldots, 0, \ldots, n\})$:

$$\mathbf{A}_s = \begin{pmatrix} 1 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{pmatrix} \qquad \mathbf{A}_s^\dagger = \begin{pmatrix} \frac{1}{n} & \ldots & \frac{1}{n} & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & \frac{1}{n} & \ldots & \frac{1}{n} \end{pmatrix}$$

Example: Function Approximation (ctd.)

Concrete and abstract domain are step-functions on $[a, b]$.
The set of (real-valued) step-function $\mathcal{T}_n$ is based on the sub-division of the interval into n sub-intervals.



Each step function in $\mathcal{T}_n$ corresponds to a vector in $\mathbb{R}^n$, e.g.

$$\begin{pmatrix} 5 & 5 & 6 & 7 & 8 & 4 & 3 & 2 & 8 & 6 & 6 & 7 & 9 & 8 & 8 & 7 \end{pmatrix}$$

Example: Abstraction Matrices

$$\mathbf{A}_8 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

Example: Abstraction Matrices

$$\mathbf{G}_8 = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Compute the abstractions of f as $f\mathbf{A}_j$.

In a similar way we can also compute the over- and under-approximation of f in $\mathcal{T}_i$ based on the pointwise ordering and its reverse.

Approximation Estimates

Compute the *least square error* as

$$\|f - f\mathbf{AG}\|.$$

$$\begin{aligned}\|f - f\mathbf{A}_8\mathbf{G}_8\| &= 3.5355 \\ \|f - f\mathbf{A}_4\mathbf{G}_4\| &= 5.3151 \\ \|f - f\mathbf{A}_2\mathbf{G}_2\| &= 5.9896 \\ \|f - f\mathbf{A}_1\mathbf{G}_1\| &= 7.6444\end{aligned}$$

Concrete Semantics (LOS)

$$\mathbf{T}(P) = \sum_{\langle i,p_{ij},j\rangle \in flow(P)} p_{ij} \cdot \mathbf{T}(\ell_i, \ell_j),$$

where

$$\mathbf{T}(\ell_i, \ell_j) = \mathbf{N} \otimes \mathbf{E}(\ell_i, \ell_j),$$

with **N** an operator representing a state update while the second factor realises the transfer of control from label ℓ_i to label ℓ_j.

Abstract Semantics

Moore-Penrose Pseudo-Inverse of a Tensor Product is:

$$(\mathbf{A}_1 \otimes \mathbf{A}_2 \otimes \ldots \otimes \mathbf{A}_n)^\dagger = \mathbf{A}_1^\dagger \otimes \mathbf{A}_2^\dagger \otimes \ldots \otimes \mathbf{A}_n^\dagger$$

Via linearity we can construct $\mathbf{T}^\#$ in the same way as $\mathbf{T}$, i.e

$$\mathbf{T}^\#(P) = \sum_{\langle i, p_{ij}, j\rangle \in \mathcal{F}(P)} p_{ij} \cdot \mathbf{T}^\#(\ell_i, \ell_j)$$

with local abstraction of individual variables:

$$\mathbf{T}^\#(\ell_i, \ell_j) = (\mathbf{A}_1^\dagger \mathbf{N}_{i1} \mathbf{A}_1) \otimes (\mathbf{A}_2^\dagger \mathbf{N}_{i2} \mathbf{A}_2) \otimes \ldots \otimes (\mathbf{A}_v^\dagger \mathbf{N}_{iv} \mathbf{A}_v) \otimes \mathbf{M}_{ij}$$

Argument

$$\begin{aligned}
\mathbf{T}^\# &= \mathbf{A}^\dagger \mathbf{T} \mathbf{A} \\
&= \mathbf{A}^\dagger (\sum_{i,j} p_{ij} \mathbf{T}(i,j)) \mathbf{A} \\
&= \sum_{i,j} \mathbf{A}^\dagger p_{ij} \mathbf{T}(i,j) \mathbf{A} \\
&= \sum_{i,j} p_{ij} (\bigotimes_k \mathbf{A}_k)^\dagger \mathbf{T}(i,j) (\bigotimes_k \mathbf{A}_k) \\
&= \sum_{i,j} p_{ij} (\bigotimes_k \mathbf{A}_k^\dagger)(\bigotimes_k \mathbf{N}_{ik})(\bigotimes_k \mathbf{A}_k) \\
&= \sum_{i,j} p_{ij} \bigotimes_k (\mathbf{A}_k^\dagger \mathbf{N}_{ik} \mathbf{A}_k)
\end{aligned}$$

Parity Analysis

Determine at each program point whether a variable is *even* or *odd*.

Parity Abstraction operator on $\mathcal{V}(\{0, \ldots, n\})$ (with n even):

$$\mathbf{A}_p = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ \vdots & \vdots \\ 0 & 1 \end{pmatrix} \qquad \mathbf{A}^\dagger = \begin{pmatrix} \frac{2}{n} & 0 & \frac{2}{n} & 0 & \ldots & 0 \\ 0 & \frac{2}{n} & 0 & \frac{2}{n} & \ldots & \frac{2}{n} \end{pmatrix}$$

Example

$[m \leftarrow 1]^1$;
while $[n > 1]^2$ **do**
    $[m \leftarrow m \times n]^3$;
    $[n \leftarrow n - 1]^4$
end while
$[\textbf{stop}]^5$

$$\begin{aligned} \mathbf{T} &= \mathbf{U}(\mathrm{m} \leftarrow 1) \otimes \mathbf{E}(1,2) \\ &+ \mathbf{P}(n > 1) \otimes \mathbf{E}(2,3) \\ &+ \mathbf{P}(n \leq 1) \otimes \mathbf{E}(2,5) \\ &+ \mathbf{U}(\mathrm{m} \leftarrow m \times n) \otimes \mathbf{E}(3,4) \\ &+ \mathbf{U}(\mathrm{n} \leftarrow n - 1) \otimes \mathbf{E}(4,2) \\ &+ \mathbf{I} \otimes \mathbf{E}(5,5) \end{aligned} \qquad \begin{aligned} \mathbf{T}^\# &= \mathbf{U}^\#(\mathrm{m} \leftarrow 1) \otimes \mathbf{E}(1,2) \\ &+ \mathbf{P}^\#(n > 1) \otimes \mathbf{E}(2,3) \\ &+ \mathbf{P}^\#(n \leq 1) \otimes \mathbf{E}(2,5) \\ &+ \mathbf{U}^\#(\mathrm{m} \leftarrow m \times n) \otimes \mathbf{E}(3,4) \\ &+ \mathbf{U}^\#(\mathrm{n} \leftarrow n - 1) \otimes \mathbf{E}(4,2) \\ &+ \mathbf{I}^\# \otimes \mathbf{E}(5,5) \end{aligned}$$

Abstract Semantics

Abstraction: $\mathbf{A} = \mathbf{A}_{p_m} \otimes \mathbf{I}_n \otimes \mathbf{I}_l$, i.e. m abstract (parity) but n and the labels are not abstracted.

$$
\begin{aligned}
\mathbf{T}^{\#} &= \mathbf{U}^{\#}(m \leftarrow 1) \otimes \mathbf{E}(1,2) \\
&+ \mathbf{P}^{\#}(n > 1) \otimes \mathbf{E}(2,3) \\
&+ \mathbf{P}^{\#}(n \leq 1) \otimes \mathbf{E}(2,5) \\
&+ \mathbf{U}^{\#}(m \leftarrow m \times n) \otimes \mathbf{E}(3,4) \\
&+ \mathbf{U}^{\#}(n \leftarrow n - 1) \otimes \mathbf{E}(4,2) \\
&+ \mathbf{I}^{\#} \otimes \mathbf{E}(5,5)
\end{aligned}
$$

Abstract Semantics

$$
\mathbf{U}^{\#}(m \leftarrow 1) = \\
= \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & & \dots & 1 \end{pmatrix}
$$

Abstract Semantics

$$\mathbf{U}^{\#}(n \leftarrow n-1) =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Abstract Semantics

$$\mathbf{P}^{\#}(n > 1) =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

Abstract Semantics

$$\mathbf{P}^{\#}(n \leq 1) =$$

$$= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Abstract Semantics

$$\mathbf{U}^{\#}(m \leftarrow m \times n) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix} +$$

$$+ \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \ddots \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \ddots \end{pmatrix}$$

Implementation

Implementation of concrete and abstract semantics of Factorial using **octave**. Ranges: $n \in \{1, \ldots, d\}$ and $m \in \{1, \ldots, d!\}$.

d	dim($\mathbf{T}(F)$)	dim($\mathbf{T}^{\#}(F)$)
2	45	30
3	140	40
4	625	50
5	3630	60
6	25235	70
7	201640	80
8	1814445	90
9	18144050	100

Using uniform initial distributions $\mathbf{d_0}$ for n and m.

Scalablity

The abstract probabilities for m being **even** or **odd** when we execute the abstract program for various d values are:

d	**even**	**odd**
10	0.81818	0.18182
100	0.98019	0.019802
1000	0.99800	0.0019980
10000	0.99980	0.00019998

References

- Nielson-Nielson-Hankin: *Principles of Program Analysis*, Springer Verlag, 1999.
- Di Pierro-Wiklicky: *Probabilistic semantics and analysis*, Vol. 6155 of LNCS, Springer Verlag, 2010.
- Cousot-Cousot: *Abstract Interpretation: A Unified Lattice Model for Static Analysis of Programs by Construction or Approximation of Fxpoints*, Proceedings POPL'77, ACM Press, 1977.