Interval Analysis of Unreliable Programs

A thesis submitted in partial fulfillment of
The requirements for the degree of Master of Technology
in
Computer Science and Engineering

By
Dibyendu Das
Roll No. 14CS60R34
Under the supervision of
Prof Soumyajit Dey
Dept. of CSE, IIT Kharagpur

Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Contents

Certification

This is to certify that the project entitled "*Interval Analysis of Unreliable Programs*" being submitted by Dibyendu Das is a bonafide work done by him in the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur under my supervision and guidance for M.Tech Project.

Prof. Soumyajit Dey
Department of Computer Science and Engineering
IIT Kharagpur

Declaration

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature and acknowledgement of collaborative researchand discussions. The work was done under the guidance of Prof. Soumyajit Dey, at Indian institute of Technology, Kharagpur.

Dibyendu Das
Department of Computer Science and Engineering
IIT Kharagpur

Acknowledgements

This thesis is the result of the project work performed under the guidance of Prof. Soumyajit Dey at the Department of Computer Science and Engineering of the Indian Institute of Technology, Kharagpur. I am deeply grateful to my supervisor for having given me the opportunity of working under him and guiding me seamlessly for the whole period. He gave me exposure to all the related research going on in this field, which helped me enormously. Without the help, encouragement and patient support I received from my guide, this report would never have materialized.

Dibyendu Das
Department of Computer Science and Engineering
IIT Kharagpur

Abstract

Advancement of chip technology will make futute computer chips more and more fast. But this speed gain doesn't come free of cost. There is going to be a trade-off between speed and efficiency, i.e accuracy of the computation. In order to achieve this ultra speed we'll simply have to let our computers make more mistakes in computations.

Our objective in this project is to statically analyse codes written for these kind of architecture. We have used a C-type language to model the programs written for this architecture. One example of such unreliable programs is RELY[3]. Our static analysis primarily focuses on *Interval Analysis* of this sort of programs. There are 2 types of failures of the hardware components namely: *permanent failure model* (where the hardware stops on failure) and *transient failure model* (where, on failure, the hardware continues with the next operations). We've only taken transient failure model into consideration.

The goal of this analysis is to be able to predict, **for each program variable at each program point, the interval/range within which it belongs with a particular certainty/confidence**. say for example, the program has n variables namely $x_1, x_2 \cdots x_n$. Our analysis will produce output something like $x_i = \langle [a, b], p_{ab} \rangle$ for each i, at every program point. This means that variable x_i, at that program point lines in the interval $[a, b]$ with probability p_{ab}.

Abstract Interpretation is what we've based our technique on. For that we've come up with two new complete lattices (one concrete and one abstract) for our probabilistic domains and established Galois Connection between them. Right now we're only concerned for program variables storing integer values.

The major applications or this work can be *calculating the success probability of the overall program*, *static branch prediction* for such languages. Other application could be *detection of most reliable path in the program*.

1. Introduction

As transistors get smaller, they also become less reliable. So far, computer-chip designers have been able to work around that problem, but in the future, it could mean that computers stop improving at the rate weve come to expect.

A third possibility, which some researchers have begun to float, is that we could simply let our computers make more mistakes. This reliability won't be a major issue in some cases. If, for instance, a few pixels in each frame of a high-definition video are improperly decoded, viewers probably wont notice — but relaxing the requirement of perfect decoding could yield gains in speed or energy efficiency.

Emerging high-performance architectures are anticipated to contain unreliable components that may exhibit soft errors, which silently corrupt the results of computations. Full detection and masking of soft errors is challenging, expensive, and, for some applications, unnecessary. For example, approximate computing applications such as *i*) Multimedia processing, *ii*) High definition video decoding, *iii*) Image manipulation, *iv*) Machine learning, *v*) Big data analytics can often naturally tolerate soft errors.

For handling the future unreliable chips, a research group at MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) has developed a new programming framework called **RELY**[3] that enables software developers to specify when errors may be tolerable. The system then calculates the probability that the software will perform as it's intended.

Here, we present an amended form of the popular static analysis tool called **Range Analysis** or **Interval Analysis** for unreliable programs as such. We call it **Probabilistic Interval Analysis**. Our model is so modified as to incorporate the unreliabilities occur because of unreliable ALU and memory operations.

2. Objective

We propose a general abstract interpretation based method for the static analysis of programs that run on unreliable hardwares. Our method does Interval analysis for programs like these. The objectibe is to calculate the success probability of the overall program and other similar tasks like detecting the most reliable path in the program.

More formally, for a program with n integer variables, we are interested in answering questions like, with what probability does each program variable lie in a range of values at each program point. In other words, goal is to find out the probability that $x_i \in [a, b]$ at program point p_i.

3. Related Work

There have already been some work on static analysis of programs with *imprecise probabilistic inputs* [2], which heavily relies on Dempster-Shafer structures (DSI) or P-boxes. This analyzer assumes each input value to be lying within a *confidence box* $[\underline{P}, \overline{P}]$. Based on this assumption it works on to perform the interval analysis of each variable at each program point.

Few more works are there that analyze programs with non-deterministic and probabilistic behavior [4][6]. Those use general abstract interpretation based method for the static analysis of programs using random generators or random inputs and also allow non-deterministic inputs, not necessarily following a random distribution.

Our work is orthogonal to all these as we're concerned with the imprecision seeded into the hardware rather than in input. So, in my case inputs come from reliable sources with fully reliable values.

4. Unreliable Programs

Following is an example program snippet and its corresponding **C**ontrol **F**low **G**raph for RELY like programming languages.

```
//v0
x =• 0
//v1
while (x <• 10) // v2
      x =• x +• 3
//v3
```

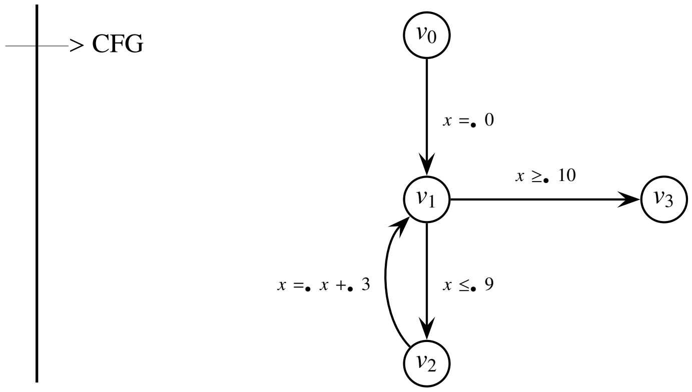


Figure 1: Example program.

Here each operation is probabilistic i.e each operation produces correct values with some probability. Unreliable operators are denoted by the corresponding operator followed by a *subscript bullet* (•). Each operator has a success probability associated with it i.e the operator succeeds and produces correct result with a predefined probability. These probability values are set by the hardware manufacturer of the chip. Upon failure the probabilistic operators take random values from their corresponding domains. For arithmatic operators this domain is $\mathbb{Z} \cap$ [MININT, MAXINT], for boolean operators it is $\{true, false\}$ etc. We define the success probabilities of some of the operators as follows :

Operator	Probability of successful execution (set by hardware manufacturer)	Failure probability	Domain of operation
$+_\bullet$	$Pr(+_\bullet)$	$1 - Pr(+_\bullet)$	$\{a \in \mathbb{Z} \quad \mid \quad \texttt{MININT} \leq a \leq \texttt{MAXINT}\}$
$-_\bullet$	$Pr(-_\bullet)$	$1 - Pr(-_\bullet)$	
$\times_\bullet$	$Pr(\times_\bullet)$	$1 - Pr(\times_\bullet)$	
$\div_\bullet$	$Pr(\div_\bullet)$	$1 - Pr(\div_\bullet)$	
$=_\bullet$	$Pr(Wr)$	$1 - Pr(Wr)$	
$>_\bullet$	$Pr(>_\bullet)$	$1 - Pr(>_\bullet)$	$\{true,\ false\}$
$\geq_\bullet$	$Pr(\geq_\bullet)$	$1 - Pr(\geq_\bullet)$	

⋮

E.g: This program statement

$$\sigma : x \; =_{\bullet} \; x \; +_{\bullet} \; 3$$

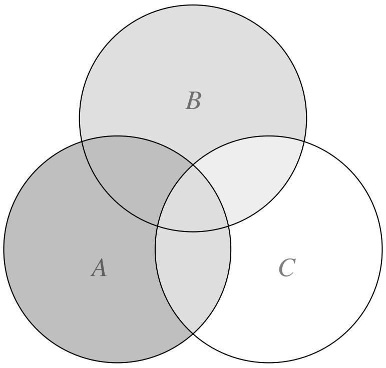
involves 3 probabilistic operations namely Read (**Rd**) of the variable x, Add (**+**) & Write (**Wr**) to the variable x. So, the probability that σ executes successfully is $Pr(Rd) \cdot Pr(+_{\bullet}) \cdot Pr(Wr)$. Now we compute the probability with which program variable x holds correct value after the execution of σ.

These three operations are independent of each other. Corresponding to three operations we have three events namely:

- A: **Read** executes successfully.
- B: **Add** ($+_{\bullet}$) executes successfully.
- C: **Write** executes successfully.

$\therefore \quad Pr(A) = Pr(Rd), \quad Pr(B) = Pr(+_{\bullet}) \quad \text{and} \quad Pr(C) = Pr(Wr).$

All three events are pairwise independent but none of them are mutually exclusive.



After σ is executed the variable x will hold the desired value with probability

$$Rel_{cur}(x) = Rel_{prev}(x) \cdot \left(Pr(Rd) + \frac{1 - Pr(Rd)}{\texttt{MAXINT} - \texttt{MININT} + 1} \right) \cdot \left(Pr(+_{\bullet}) + \frac{1 - Pr(+_{\bullet})}{\texttt{MAXINT} - \texttt{MININT} + 1} \right) \cdot \left(Pr(Wr) + \frac{1 - Pr(Wr)}{\texttt{MAXINT} - \texttt{MININT} + 1} \right)$$

Where, $\underline{Rel_{prev}(x)}$ is the probability with which variable x holds the correct value just before this statement is executed.

We are interested in static analysis of these type of probabilistic programs by employing the theory of abstraction using Galois Connection (GC).

5. The Probabilistic Concrete Domain, *L*

From the above arguments we can infer that in the concrete domain each possible program state is associated with some probability. We represent program state as a tuple of set of values and its associated probability. Set of all possible program states is thus represented by

$$\mathbb{Z}_p = \Big\{\langle a, p_a\rangle \quad \Big| \quad a \in \mathbb{Z}, p_a \in \mathbb{R}, \texttt{MININT} \leq a \leq \texttt{MAXINT}, 0 \leq p_a \leq 1\Big\}$$

Throughout our discussion we assume that our program variables take values between `MININT` and `MAXINT` (including both). $\langle a_i, p_{a_i}\rangle$ denotes that at a particular program point v_i, variable x_i takes the value a_i with probability p_{a_i}. For $1 \leq i \leq n$, We define two utility functions as follows :

$$\mathbf{val}\Big(\big(\langle a_1, p_{a_1}\rangle, \langle a_2, p_{a_2}\rangle \cdots \langle a_n, p_{a_n}\rangle\big), i\Big) = a_i \tag{1}$$

$$\mathbf{prob}\Big(\big(\langle a_1, p_{a_1}\rangle, \langle a_2, p_{a_2}\rangle \cdots \langle a_n, p_{a_n}\rangle\big), i\Big) = p_{a_i} \tag{2}$$

$$\tag{3}$$

In this case, the concrete domain L is a subset of the power set of $\mathbb{Z}_p$. For program with a single integer variable, the concrete domain is $L = \Big\{S \in 2^{\mathbb{Z}_p} \ : \ S \text{ is finite} \bigwedge \big|\{\mathbf{val}(t) \mid \forall t \in S\}\big| = \big|S\big|\Big\}$[1], set of all feasible sets of program states. In general, for program with n integer variables, this concrete domain will be : $\bigwedge_i \forall S_i \in L, \quad \Big\{S_1 \times S_2 \times \cdots \times S_n\Big\} = L_n$.

We define a partial order $\sqsubseteq_L$ on L as follows :

$$\forall A, B \in L \qquad A \sqsubseteq_L B$$

iff

i) $\Big\{\mathbf{val}(t) \,\Big|\, \forall t \in A\Big\} \subseteq \Big\{\mathbf{val}(t) \,\Big|\, \forall t \in B\Big\}$

ii) $\forall t_a \in A, \ \forall t_b \in B \quad \Big(\big(\mathbf{val}(t_a) = \mathbf{val}(t_b)\big) \Rightarrow \big(\mathbf{prob}(t_a) \geq \mathbf{prob}(t_b)\big)\Big)$

all the above conditions hold simultaneously. As an example,

$$\Big\{\langle a_1, p_{a_1}\rangle, \ \langle a_2, p_{a_2}\rangle \ \cdots \ \langle a_n, p_{a_n}\rangle\Big\} \sqsubseteq_L \Big\{\langle a_1, p'_{a_1}\rangle, \ \langle a_2, p'_{a_2}\rangle \ \cdots \ \langle a_m, p'_{a_m}\rangle\Big\} \qquad \text{iff}$$
$$n \leq m, \text{so that } \{a_1, a_2 \ \cdots \ a_n\} \subseteq \{a_1, a_2 \ \cdots \ a_m\} \qquad \text{and} \qquad \bigwedge_n (p_{a_i} \geq p'_{a_i})$$

$(L, \ \sqsubseteq_L)$ is a **poset**, proof of which is given in Appendix A. To show this poset to be a **lattice**, we need to show that for any two elements in L there exist a unique infimum and a unique supremum. More formally

$$\forall x, y \in L \ \Bigg(\exists L_{lub} \in \Big\{u \in L \ \Big| \ (x \sqsubseteq_L u) \wedge (y \sqsubseteq_L u)\Big\} \ s.t \Big(\forall z \ \ \Big(z \in \Big\{u \in L \ \Big| \ (x \sqsubseteq_L u) \wedge (y \sqsubseteq_L u)\Big\} \Rightarrow L_{lub} \sqsubseteq_L z\Big)\Big)$$
$$\bigwedge$$
$$\exists L_{glb} \in \Big\{l \in L \ \Big| \ (l \sqsubseteq_L x) \wedge (l \sqsubseteq_L y)\Big\} \ s.t \Big(\forall z \ \ \Big(z \in \Big\{l \in L \ \Big| \ (l \sqsubseteq_L x) \wedge (l \sqsubseteq_L y)\Big\} \Rightarrow z \sqsubseteq_L L_{glb}\Big)\Big)\Bigg)$$

where L_{lub} is the least upper bound and L_{glb} is the greatest lower bound of x and y in L. We now define the least upper bound or $l.u.b$, $\bigsqcup_L$ and the greatest lower bound or $g.l.b$, $\bigsqcap^L$ for any two elements A and B of L as follows.

[1] This way we ensure that values in each element of L are unique. For example, $\Big\{\langle a, p_a\rangle, \ \langle b, p_b\rangle\Big\} \in L$ but $\Big\{\langle a, p_a\rangle, \ \langle b, p_{b_1}\rangle, \ \langle b, p_{b_2}\rangle\Big\} \notin L$

$$A \bigsqcup_{L} B \quad = \quad \left\{t_a \,\middle|\, t_a \in A \bigwedge \mathbf{val}(t_a) \notin \{\mathbf{val}(t_b) \mid t_b \in B\}\right\} \bigcup \left\{t_b \,\middle|\, t_b \in B \bigwedge \mathbf{val}(t_b) \notin \{\mathbf{val}(t_a) \mid t_a \in A\}\right\} \bigcup$$
$$\left\{\langle \mathbf{val}(t_a), \mathbf{min}(\mathbf{prob}(t_a), \mathbf{prob}(t_b))\rangle \,\middle|\, t_a \in A, t_b \in B \bigwedge \mathbf{val}(t_a) = \mathbf{val}(t_b)\right\} \tag{4}$$

$$A \bigsqcap^{L} B \quad = \quad \left\{\langle \mathbf{val}(t_a), \mathbf{max}(\mathbf{prob}(t_a), \mathbf{prob}(t_b))\rangle \,\middle|\, t_a \in A, t_b \in B \bigwedge \mathbf{val}(t_a) = \mathbf{val}(t_b)\right\} \tag{5}$$

Soundness of these definitions is proved in Appendix B. Greatest and least element of this lattice $(L, \sqsubseteq_L)$ are $\top_L$ and $\bot_L$ respectively, where

$\top_L = \{\langle a, 0\rangle \,|\, a \in [\texttt{MININT}, \texttt{MAXINT}]\} = \{\langle \texttt{MININT}, 0\rangle \cdots \langle -1, 0\rangle, \langle 0, 0\rangle, \langle 1, 0\rangle \cdots \langle \texttt{MAXINT}, 0\rangle\}$

$\bot_L = \{\,\}$

The **Hasse Diagram** of lattice $(L, \sqsubseteq_L)$ can be found in Appendix C. $\forall S \subseteq L, \quad S$ has a unique *l.u.b* and *g.l.b* in L. Hence L is a complete lattice.

Next, we shall define a function on L_n called the *strongest post-condition function*[1].

Consider P to be a predicate defined over program variables. Alternatively, we can also think of P as a set of possible program states which satisfy the relation P. For example, $p = (\langle 5, p_1\rangle) \equiv [x = 5]$ is a program state (p_1 is a place-holder here) and $P : [x > 10]$ is a predicate such that $p \notin P$. Similarly, consider a program with three integer variables x, y and z such that $p = (\langle 5, p_1\rangle, \langle 10, p_2\rangle, \langle 10, p_3\rangle) \equiv [x = 5,\ y = 10,\ z = 10]$ and $p' = (\langle 15, p'_1\rangle, \langle 0, p'_2\rangle, \langle 1, p'_3\rangle) \equiv [x = 15,\ y = 0,\ z = 1]$ are two program states. Given a predicate $P : [x + y \geq z]$, we may observe that both $p, p' \in P$. Let P denote a (pre-condition) predicate which is true before the execution of some program statement S. Strongest post condition (sp) is a function which takes as argument a program statement S (or a sequence of statements) and a predicate P and returns the strongest predicate that holds after executing S from any program state which satisfies P. It is strongest in the sense that for any other such predicate Q which holds for any state resulting from execution of S given P, we shall always have $sp(P, S) \Rightarrow Q$ (implies Q) or in a set theoretic notation $sp(P, S) \subseteq Q$. As example, we consider next arithmetic expressions (with assignment) and logical expressions as examples of S.

We assume that there are n variables in the program, namely, $x_1, x_2 \cdots x_n$.

$$
\begin{aligned}
& sp\Big(P,\ x_{i_f} =_{\bullet} x_{i_1} \text{🅐}_{2}\, x_{i_2} \text{🅐}_{3} \cdots \text{🅐}_{k}\, x_{i_k}\Big) \\
& = \Bigg\{ \Big(\langle a_1, p_1 \rangle, \langle a_2, p_2 \rangle \cdots \langle a_n, p_n \rangle \Big) \Bigg|\ \exists t \in P' \Bigg(a_{i_f} = \mathbf{val}(t, i_f) \bigwedge p_{i_f} = \Bigg(\sum_{\forall t' \in P',\ \mathbf{val}(t', i_f) = \mathbf{val}(t, i_f)} \mathbf{prob}(t', i_f) \Bigg) \cdot \\
& \quad \left(Pr(Wr) + \frac{1 - Pr(Wr)}{\mathtt{MAXINT} - \mathtt{MININT} + 1} \right) \cdot \left(Pr(Rd) + \frac{1 - Pr(Rd)}{\mathtt{MAXINT} - \mathtt{MININT} + 1} \right)^{l} \cdot \prod_{j=2}^{k} \left(Pr\Big(\text{🅐}_{j}\Big) + \frac{1 - Pr\Big(\text{🅐}_{j}\Big)}{\mathtt{MAXINT} - \mathtt{MININT} + 1} \right) \bigwedge \\
& \qquad \forall j \in \mathbb{N} \setminus \{i_f\} \Big(j \leq n \ \Rightarrow a_j = \mathbf{val}(t, j) \bigwedge p_j = \mathbf{prob}(t, j) \Big) \Bigg) \Bigg\}
\end{aligned} \tag{6}
$$

where, $l = \Big| \{ i_k \mid x_{i_k} \text{ is a variable} \} \Big|$ and

$$
\begin{aligned}
P' = & \Bigg\{ \Big(\langle a_1, p_1 \rangle, \langle a_2, p_2 \rangle \cdots \langle a_n, p_n \rangle \Big) \Bigg|\ \exists t \in P \Bigg(a_{i_f} = \Big(v(t, x_{i_1})\ \text{Ⓐ}_{2}\ v(t, x_{i_2})\ \text{Ⓐ}_{3} \cdots \text{Ⓐ}_{k}\ v(t, x_{i_k}) \Big) \bigwedge p_{i_f} = \prod_{j=1}^{k} p(t, x_{i_j}) \bigwedge \\
& \qquad \forall j \in \mathbb{N} \setminus \{i_f\} \Big(j \leq n \ \Rightarrow a_j = \mathbf{val}(t, j) \bigwedge p_j = \mathbf{prob}(t, j) \Big) \Bigg) \Bigg\}
\end{aligned}
$$

$$
\begin{aligned}
& sp\Bigg(P,\ \Big(x_{i_1} \text{🅐}_{i_2}\, x_{i_2} \text{🅐}_{i_3} \cdots \text{🅐}_{i_k}\, x_{i_k} \Big)\ \text{🅒}\ \Big(x_{j_1} \text{🅐}_{j_2}\, x_{j_2} \text{🅐}_{j_3} \cdots \text{🅐}_{j_m}\, x_{j_m} \Big) \Bigg) \\
& = \Bigg\{ \Big(\langle a_1, p_1 \rangle, \langle a_2, p_2 \rangle \cdots \langle a_n, p_n \rangle \Big) \Bigg|\ \exists t \in P \Bigg(\forall j \in \mathbb{N} \Bigg(j \leq n \ \Rightarrow a_j = \mathbf{val}(t, j) \bigwedge p_j = \left(Pr(Rd) + \frac{1 - Pr(Rd)}{\mathtt{MAXINT} - \mathtt{MININT} + 1} \right)^{l} \cdot \\
& \quad \left(Pr(\text{🅒}) + \frac{1 - Pr(\text{🅒})}{2} \right) \cdot \prod_{j=2}^{k} \left(Pr\Big(\text{🅐}_{i_j}\Big) + \frac{1 - Pr\Big(\text{🅐}_{i_j}\Big)}{\mathtt{MAXINT} - \mathtt{MININT} + 1} \right) \cdot \prod_{i=2}^{m} \left(Pr\Big(\text{🅐}_{j_i}\Big) + \frac{1 - Pr\Big(\text{🅐}_{j_i}\Big)}{\mathtt{MAXINT} - \mathtt{MININT} + 1} \right) \cdot \\
& \qquad \mathbf{prob}(t, j) \Bigg) \bigwedge \Big(v(t, x_{i_1})\ \text{Ⓐ}_{i_2}\ v(t, x_{i_2})\ \text{Ⓐ}_{i_3} \cdots \text{Ⓐ}_{i_k}\ v(t, x_{i_k}) \Big)\ \text{Ⓒ}\ \Big(v(t, x_{j_1})\ \text{Ⓐ}_{j_2}\ v(t, x_{j_2})\ \text{Ⓐ}_{j_3} \cdots \text{Ⓐ}_{j_m}\ v(t, x_{j_m}) \Big) \Bigg) \Bigg\}
\end{aligned} \tag{7}
$$

where, $l = \left| \left\{ x \in \left\{ x_{i_1}, x_{i_2} \cdots x_{i_k} \right\} \bigcup \left\{ x_{j_1}, x_{j_2} \cdots x_{j_m} \right\} \middle| x \text{ is a variable} \right\} \right|$

In the above equations,

$$
v(t, x_i) = \begin{cases} \mathbf{val}(t, i) & \text{if } x_i \text{ is a variable} \\ x_i = C & \text{if } x_i \text{ is a constant, say } C \end{cases} \qquad \text{and} \qquad p(t, x_i) = \begin{cases} \mathbf{prob}(t, i) & \text{if } x_i \text{ is a variable} \\ 1 & \text{if } x_i \text{ is a constant} \end{cases}
$$

$\forall i$, 🅐_{i} is a probabilistic arithmatic operation and Ⓐ_{i} is a deterministic arithmatic operation.

i.e $\text{🅐}_{i} \in \{+_{\bullet}, -_{\bullet}, \times_{\bullet}, \div_{\bullet}, \%_{\bullet}, \cdots\}$ $\text{Ⓐ}_{i} \in \{+, -, \times, \div, \%, \cdots\}$

🅒 is a probabilistic comparison operation and Ⓒ is a deterministic comparison operation.

i.e $\text{🅒} \in \{<_{\bullet}, >_{\bullet}, \geq_{\bullet}, \leq_{\bullet}, ==_{\bullet}, \neq_{\bullet}\}$ $\text{Ⓒ} \in \{>, <, \geq, \leq, ==, \neq\}$

$$sp\Big(P,\ x =_\bullet 0\Big) = \Bigg\{\Bigg(\Bigg\langle 0,\ Pr(Wr) + \frac{1 - Pr(Wr)}{\texttt{MAXINT} - \texttt{MININT} + 1}\Bigg\rangle\Bigg)\Bigg\} \tag{8}$$

$$\begin{aligned} sp\Big(P,\ x =_\bullet x +_\bullet 3\Big) &= \Bigg\{\Bigg(\Bigg\langle \mathbf{val}(t,1) + 3,\ \mathbf{prob}(t,1) \cdot \Bigg(Pr(Rd) + \frac{1 - Pr(Rd)}{\texttt{MAXINT} - \texttt{MININT} + 1}\Bigg) \cdot \\ &\Bigg(Pr(+_\bullet) + \frac{1 - Pr(+_\bullet)}{\texttt{MAXINT} - \texttt{MININT} + 1}\Bigg) \cdot \Bigg(Pr(Wr) + \frac{1 - Pr(Wr)}{\texttt{MAXINT} - \texttt{MININT} + 1}\Bigg)\Bigg\rangle\Bigg)\ \Bigg|\ t \in P\Bigg\} \end{aligned} \tag{9}$$

$$sp\Big(P,\ x \leq_\bullet 9\Big) = \Bigg\{\Bigg(\Bigg\langle \mathbf{val}(t,1),\ \mathbf{prob}(t,1) \cdot \Bigg(Pr(\leq_\bullet) + \frac{1 - Pr(\leq_\bullet)}{2}\Bigg)\Bigg\rangle\Bigg)\ \Bigg|\ t \in P \bigwedge \mathbf{val}(t,1) \leq 9\Bigg\} \tag{10}$$

$$sp\Big(P,\ x \geq_\bullet 10\Big) = \Bigg\{\Bigg(\Bigg\langle \mathbf{val}(t,1),\ \mathbf{prob}(t,1) \cdot \Bigg(Pr(\geq_\bullet) + \frac{1 - Pr(\geq_\bullet)}{2}\Bigg)\Bigg\rangle\Bigg)\ \Bigg|\ t \in P \bigwedge \mathbf{val}(t,1) \geq 10\Bigg\} \tag{11}$$

and so on . . .

Here, P is a set of program states of the form $\{\langle a,\ p_a\rangle\}$, with unique a's, i.e $P \in L$.

Further, for any program σ, $sp(P = \perp_L = \{\ \},\ \sigma) = \perp_L = \{\ \}$. If the predicate P defines a relation which is not satisfied by any possible program state, or speaking set theoretically if P is an empty set, then it is not possible for σ to execute at all since there is no state to start from. Thus the set of states reachable is also empty. Hence, the strongest post-condition is the empty set.

Using the notion of sp, we are interested in computing the set of reachable program states for every program point which is known as ***collecting semantics***. For each program point v_i, let g_i denotes the set of reachable program states. We can consider these collection of program states, (g_i) as **pre-condition** predicates for the succeeding program statement. In that case, they are linked by sp to give the following system of equation.

$$\begin{aligned} g_0 &= \Big\{\langle a,\ 1\rangle\ \Big|\ \forall a \in \mathbb{Z} \bigwedge \texttt{MININT} \leq a \leq \texttt{MAXINT}\Big\} \\ g_1 &= sp(g_0,\ x =_\bullet 0) \bigsqcup_L sp(g_2, x =_\bullet x +_\bullet 3) \\ g_2 &= sp(g_1,\ x \leq_\bullet 9) \\ g_3 &= sp(g_1,\ x \geq_\bullet 10) \end{aligned}$$

We can derive an overall function $\mathcal{F}$,

$$\mathcal{F}(g_0,\ g_1,\ g_2,\ g_3) = \Big(g_0,\ sp(g_0,\ x =_\bullet 0) \bigsqcup_L sp(g_2,\ x =_\bullet x +_\bullet 3),\ sp(g_1,\ x \leq_\bullet 9),\ sp(g_1,\ x \geq_\bullet 10)\Big)$$

that maps each g_i to new values of g_i iteratively. The least fixed point of $\mathcal{F}$ is the final solution. We can arrive at the solution by computing the sequence, $\mathcal{F}^n(\perp_L,\ \perp_L,\ \perp_L,\ \perp_L)$ starting with the least elements, i.e. the empty sets.

For the sake of the example we assume all the success probabilities to be $1 - 10^{-2}$, i.e $Pr(+_\bullet) = Pr(-_\bullet) = Pr(Rd) = Pr(Wr) = Pr(\leq_\bullet) = \cdots = 1 - 10^{-2}$ and value of MININT and MAXINT to be -32768 and 32767 respectively. Therefore, $\Bigg(Pr(+_\bullet) + \dfrac{1 - Pr(+_\bullet)}{\texttt{MAXINT} - \texttt{MININT} + 1}\Bigg) = 0.9900001526 = x$ and $\Bigg(Pr(\leq_\bullet) + \dfrac{1 - Pr(\leq_\bullet)}{2}\Bigg) = 0.995 = y$ (say)

Solving iteratively would give us a solution as follows

$\Big(\perp_L,\ \perp_L,\ \perp_L,\ \perp_L\Big) \rightarrow \Big(\Big\{\langle a,\ 1\rangle\ \Big|\ \forall a \in \mathbb{Z} \bigwedge \texttt{MININT} \leq a \leq \texttt{MAXINT}\Big\},\ \bullet,\ \bullet,\ \bullet\Big) \rightarrow \Big(\bullet,\ \Big\{\langle 0,\ x\rangle\Big\},\ \bullet,\ \bullet\Big) \rightarrow$
$\Big(\bullet,\ \bullet,\ \Big\{\langle 0,\ xy\rangle\Big\},\ \bullet\Big) \rightarrow \Big(\bullet,\ \Big\{\langle 0,\ x\rangle, \langle 3,\ x^4y\rangle\Big\},\ \bullet,\ \bullet\Big) \rightarrow \Big(\bullet,\ \bullet,\ \Big\{\langle 0,\ xy\rangle, \langle 3,\ x^4y^2\rangle\Big\},\ \bullet\Big) \rightarrow$

$$\Big(\bullet,\ \big\{\langle 0,\ x\rangle, \langle 3,\ x^4y\rangle, \langle 6,\ x^7y^2\rangle\big\},\ \bullet,\ \bullet\Big) \rightarrow \Big(\bullet,\ \bullet,\ \big\{\langle 0,\ xy\rangle, \langle 3,\ x^4y^2\rangle, \langle 6,\ x^7y^3\rangle\big\},\ \bullet\Big) \rightarrow$$
$$\Big(\bullet,\ \big\{\langle 0,\ x\rangle, \langle 3,\ x^4y\rangle, \langle 6,\ x^7y^2\rangle, \langle 9,\ x^{10}y^3\rangle\big\},\ \bullet,\ \bullet\Big) \rightarrow \Big(\bullet,\ \bullet,\ \big\{\langle 0,\ xy\rangle, \langle 3,\ x^4y^2\rangle, \langle 6,\ x^7y^3\rangle, \langle 9,\ x^{10}y^4\rangle\big\},\ \bullet\Big) \rightarrow$$
$$\Big(\bullet,\ \big\{\langle 0,\ x\rangle, \langle 3,\ x^4y\rangle, \langle 6,\ x^7y^2\rangle, \langle 9,\ x^{10}y^3\rangle, \langle 12,\ x^{13}y^4\rangle\big\},\ \bullet,\ \bullet\Big) \rightarrow \Big(\bullet,\ \bullet,\ \bullet,\ \big\{\langle 12,\ x^{13}y^5\rangle\big\}\Big)$$

The '$\bullet$' indicates, repetition of value from previous iteration. The overall solution is therefore

$$\Big($$
$$\big\{\langle a,\ 1\rangle \mid \forall a \in \mathbb{Z} \bigwedge -32768 \le a \le 32767\big\},$$
$$\big\{\langle 0,\ 0.9900001526\rangle, \langle 3,\ 0.955793619\rangle, \langle 6,\ 0.922768992\rangle, \langle 9,\ 0.890885433\rangle, \langle 12,\ 0.860103516\rangle\big\},$$
$$\big\{\langle 0,\ 0.985050152\rangle, \langle 3,\ 0.951014651\rangle, \langle 6,\ 0.918155147\rangle, \langle 9,\ 0.886431006\rangle\big\},$$
$$\big\{\langle 12,\ 0.855802998\rangle\big\}$$
$$\Big)$$

In general, we may need infinite number of iterations to converge. This is because the height of the lattice L is ∞.

6. The Probabilistic Interval Abstract Domain, *M*

In this domain, M, the elements are of the form of tuples, $\langle [a, b], p_{ab}\rangle$, i.e with each interval there is a corresponding probability. Program variables are assigned these tuples at different program points instead of set of concrete values. E.g: program variable x_i takes value $\langle [a, b], p_{ab}\rangle$ at a particular program point v_i, signifies that value of x_i lies in the interval $[a, b]$ at that program point with probability p_{ab}.

We define a utility function $\underline{p.m.f}$ (probability mass of an interval) :

$$p.m.f\Big(\langle [a, b], p_{ab}\rangle\Big) = \frac{p_{ab}}{b - a + 1} \tag{12}$$

We now define a binary relation $\sqsubseteq_M$ on this set of abstract values, M. Before doing that we recall the case as it was with trivial Interval Abstract Domain[5]. Two intervals are related to each other by a binary relation $\sqsubseteq_{int}$ iff one interval is contained in the other.

$$[a,\ b] \sqsubseteq_{int} [c,\ d] \iff (c \le a) \wedge (b \le d) \tag{13}$$

Intuitively we lose precision in the interval that comes higher in the order.

Similar intuition is applicable in defining $\sqsubseteq_M$ on **Probabilistic Interval Abstract Domain** also. Two elements of M are related by $\sqsubseteq_M$ if we lose precision in the element that comes higher in the order. Here, by precision we mean the *accuracy (probability mass)* with which we can determine whether a program variable takes its values from the corresponding interval. For example, if one element $\langle [a, b], p_{ab}\rangle$ is more precise than another $\langle [a', b'], p'_{ab}\rangle$, this means that, with *more accuracy (higher probability mass)* we can determine that program varuiable x_i (say) takes its values from interval $[a, b]$ than it takes values from $[a', b']$.

We define,

$$\langle [a,b], p_{ab} \rangle \sqsubseteq_M \langle [c,d], p_{cd} \rangle$$

iff

i) $[a,b] \sqsubseteq_{int} [c,d]$

ii) $p.m.f\Big(\langle [a,b], p_{ab} \rangle\Big) \geq \; p.m.f\Big(\langle [c,d], p_{cd} \rangle\Big)$

all the above conditions hold simultaneously. So, clearly tuples whose intervals aren't related by $\sqsubseteq_{int}$ will not partake in $\sqsubseteq_M$.

$\sqsubseteq_M$ is a **partial order relation**. For its proof please refer to Appendix D. Formal definition of the **Probabilistic Interval Abstract Domain**, M, is :

$M = \{ \langle [a,b], p_{ab} \rangle \mid a, b \in \mathbb{Z}, p_{ab} \in \mathbb{R}, \texttt{MININT} \leq a \leq b \leq \texttt{MAXINT}, 0 \leq p_{ab} \leq 1 \} \bigcup \{ \perp_M \}$
where $\perp_M$ is the least element of M. $\perp_M = \langle [\;], 1 \rangle$ i.e the tuple of empty interval and probability 1. For n number of program variables this domain will be M^n.

The poset $(M, \sqsubseteq_M)$ is a **lattice**. For that we show that for any two elements in M there exist a unique infimum and a unique supremum. We now define the least upper bound or $l.u.b$, $\bigsqcup\limits_{M}$ and the greatest lower bound or $g.l.b$, $\overset{M}{\bigsqcap}$ for any two elements of M as follows.

$$\left.\begin{aligned} i \bigsqcup_{M} \perp_M \quad &= \quad i & \forall i \in M \\ \langle [a,b], p_{ab} \rangle \bigsqcup_{M} \langle [c,d], p_{cd} \rangle \quad &= \quad \langle [x,y], p_{xy} \rangle & \end{aligned}\right\} \tag{14}$$

where, $\quad x = \mathbf{min}(a,c) \quad y = \mathbf{max}(b,d) \quad p_{xy} = (y{-}x{+}1) \cdot \mathbf{min}\left(p.m.f\Big(\langle [a,\ b],\ p_{ab} \rangle\Big),\ p.m.f\Big(\langle [c,\ d],\ p_{cd} \rangle\Big),\ \dfrac{1}{y-x+1} \right)$

$$\left.\begin{aligned} i \overset{M}{\bigsqcap} \perp_M \quad &= \quad \perp_M & \forall i \in M \\ \langle [a,b], p_{ab} \rangle \overset{M}{\bigsqcap} \langle [c,d], p_{cd} \rangle \quad &= \begin{cases} \langle [x,y], p_{xy} \rangle & \text{if } x \leq y \\ \perp_M & \text{otherwise} \end{cases} \end{aligned}\right\} \tag{15}$$

where, $\quad x = \mathbf{max}(a,c) \quad y = \mathbf{min}(b,d) \quad p_{xy} = (y-x+1) \cdot \mathbf{max}\left(p.m.f\Big(\langle [a,\ b],\ p_{ab} \rangle\Big),\ p.m.f\Big(\langle [c,\ d],\ p_{cd} \rangle\Big)\right)$

Soundness of these definitions is proved in Appendix E. For the **Hasse Diagram** of this lattice $(M, \sqsubseteq_M)$ please refer to Appendix F.

The greatest and least element of $(M, \sqsubseteq_M)$ are $\top_M$ and $\perp_M$ respectively, where

$\top_M = \langle [\texttt{MININT}, \texttt{MAXINT}], 0 \rangle$

$\perp_M = \langle [\;], 1 \rangle$

7. Galois connection between $(L, \sqsubseteq_L)$ and $(M, \sqsubseteq_M)$

For two complete lattices $(L, \sqsubseteq_L)$ and $(M, \sqsubseteq_M)$ a pair (α, γ) of monotonic functions

$$\alpha : L \rightarrow M \qquad \text{and} \qquad \gamma : M \rightarrow L$$

is called a **Galois connection** if

$$\forall l \in L : l \sqsubseteq_L \gamma\Big(\alpha(l)\Big) \qquad \text{and} \qquad \forall m \in M : \alpha\Big(\gamma(m)\Big) \sqsubseteq_M m$$

Before going on with the definition of (α, γ) we define few utility functions over L, the concrete domain. $\forall S \in L$:

$$\mathbf{MIN_v}\Big(S\Big) = v_m \qquad \text{where,} \quad v_m \in \{\mathbf{val}(t) \mid \forall t \in S\} \bigwedge \forall v \Big((v \in \{\mathbf{val}(t) \mid \forall t \in S\}) \Rightarrow v_m \leq v\Big) \tag{16a}$$

$$\mathbf{MAX_v}\Big(S\Big) = v_M \qquad \text{where,} \quad v_M \in \{\mathbf{val}(t) \mid \forall t \in S\} \bigwedge \forall v \Big((v \in \{\mathbf{val}(t) \mid \forall t \in S\}) \Rightarrow v \leq v_M\Big) \tag{16b}$$

$$\mathbf{MIN_p}\Big(S\Big) = p_m \qquad \text{where,} \quad p_m \in \{\mathbf{prob}(t) \mid \forall t \in S\} \bigwedge \forall p \Big((p \in \{\mathbf{prob}(t) \mid \forall t \in S\}) \Rightarrow p_m \leq p\Big) \tag{16c}$$

So, $\mathbf{MIN_v}$ and $\mathbf{MAX_v}$ returns respectively the **minimum** and **maximum** of all **values** among all the tuples of a member set of L. Where as $\mathbf{MIN_p}$ returns the **minimum** of all **probabilities** among all the tuples in a set of L. For example, if $S = \{\langle a_1, p_{a_1}\rangle, \langle a_2, p_{a_2}\rangle, \langle a_3, p_{a_3}\rangle\}$ then $\mathbf{MIN_v}\Big(S\Big) = \mathbf{min}(a_1, a_2, a_3)$, $\mathbf{MAX_v}\Big(S\Big) = \mathbf{max}(a_1, a_2, a_3)$ and $\mathbf{MIN_p}\Big(S\Big) = \mathbf{min}(p_{a_1}, p_{a_2}, p_{a_3})$

We now define (α, γ) as follows :

$$\alpha(S) = \begin{cases} \langle [\,],\ 1\rangle = \bot_M & S = \bot_L = \{\,\} \\ \Big\langle \Big[\mathbf{MIN_v}(S),\ \mathbf{MAX_v}(S)\Big],\ \mathbf{min}\Big(1,\ \mathbf{MIN_p}(S) * \Big(\mathbf{MAX_v}(S) - \mathbf{MIN_v}(S) + 1\Big)\Big)\Big\rangle & \text{otherwise} \end{cases} \tag{17}$$

$$\left.\begin{aligned} \gamma\Big(\langle [\,],\ 1\rangle\Big) \quad &= \quad \{\,\} = \bot_L \\ \gamma\Big(\langle [a,\ b],\ p_{ab}\rangle\Big) \quad &= \quad \Big\{\Big\langle i,\ p.m.f(\langle [a, b], p_{ab}\rangle)\Big\rangle \ \Big|\ i \in [a, b]\Big\} \end{aligned}\right\} \tag{18}$$

For the proof of monotonicity of α and γ please refer to Appendix G.

$\forall A \in L \quad A \sqsubseteq_L \gamma\Big(\alpha(\bot_L)\Big)$. And this can be proved easily by expanding the definitions of α and γ.

If $\quad A = \bot_L, \quad$ we have $\quad \bot_L \sqsubseteq_L \gamma\Big(\alpha(\bot_L)\Big) \quad [\ \because\ \alpha(\bot_L) = \bot_M$ and $\gamma(\bot_M) = \bot_L]$

Otherwise, $\quad \gamma\Big(\alpha(A)\Big) = \left\{\left\langle i,\ \mathbf{min}\left(\dfrac{1}{\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1},\ \mathbf{MIN_p}\,(A)\right)\right\rangle\ \middle|\ i \in [\mathbf{MIN_v}(A), \mathbf{MAX_v}(A)]\right\}$

And $\quad A \sqsubseteq_L \left\{\left\langle i,\ \mathbf{min}\left(\dfrac{1}{\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1},\ \mathbf{MIN_p}\,(A)\right)\right\rangle\ \middle|\ i \in [\mathbf{MIN_v}(A), \mathbf{MAX_v}(A)]\right\}$[2]

[2] Trivially follows from the definition of $\sqsubseteq_L$

Similarly, we can prove that $\forall m \in M \quad \boldsymbol{\alpha}\big(\boldsymbol{\gamma}(m)\big) \sqsubseteq_M m$

If $m = \bot_M$, we have $\boldsymbol{\alpha}\big(\boldsymbol{\gamma}(\bot_M)\big) \sqsubseteq_M \bot_M \quad [\because \ \boldsymbol{\gamma}(\bot_M) = \bot_L$ and $\boldsymbol{\alpha}(\bot_L) = \bot_M]$

Otherwise, $\boldsymbol{\alpha}\Big(\boldsymbol{\gamma}\big(\langle [a,b], p_{ab}\rangle\big)\Big) = \boldsymbol{\alpha}\Big(\Big\{\Big\langle i,\ p.m.f(\langle [a,b], p_{ab}\rangle)\Big\rangle \ \Big|\ i \in [a,b]\Big\}\Big) = \langle [a,b], p_{ab}\rangle$

$\therefore \quad \boldsymbol{\alpha}\big(\boldsymbol{\gamma}(m)\big) = m \sqsubseteq_M m \qquad \forall m \in M \bigwedge m \neq \bot_M$

8. Safe approxmation of strongest post-condition function sp in the abstract domain M

For a program statement σ, we can think of the strongest post-condition function sp as $sp(\sigma) : L \to L$. This is a transformation from a function with two arguments i.e. $sp(P, \sigma)$ to a function $sp(\sigma)$ with a single argument P which is an element of L (a collection of program states). Given a σ, we can construct the function $sp(\sigma)$ which maps a collection of concrete program states to another collection of concrete program states. We are interested in ***safe approximation*** of such functions in our abstract domain M. Let $sp^{\#}(\sigma) : M \to M$ be our choice of safe approximation in the abstract domain M. For safe approximation of sp in L by $sp^{\#}$ in M we require the following to hold.

$$\forall m \in M, \quad \boldsymbol{\alpha}\big(sp(\boldsymbol{\gamma}(m),\ \sigma)\big) \sqsubseteq_M sp^{\#}(m,\ \sigma) \quad \text{or equivalently} \quad sp\big(\boldsymbol{\gamma}(m),\ \sigma\big) \sqsubseteq_L \boldsymbol{\gamma}\big(sp^{\#}(m,\ \sigma)\big)$$

The functions $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ are defined earlier for the Probalilistic Interval domain abstraction. In case $sp^{\#}$ is the ***most precise safe approximation***, we can write

$$\boldsymbol{\alpha}\big(sp(\boldsymbol{\gamma}(m),\ \sigma)\big) = sp^{\#}(m,\ \sigma)$$

Therefore, using $\boldsymbol{\alpha}$ and $\boldsymbol{\gamma}$ from equations 17 and 18 respectively, we can use the above relation to compute $sp^{\#}$.

A few examples of $sp^{\#}(m,\sigma)$ for different kinds of program statements are as follows.

$$\begin{aligned}
sp^{\#}\Big(\langle[a,b],p_{ab}\rangle,\ x =_\bullet 0\Big) &= \boldsymbol{\alpha}\Big(sp(\boldsymbol{\gamma}(\langle[a,b],p_{ab}\rangle)),\ x =_\bullet 0)\Big)\\
&= \boldsymbol{\alpha}\Big(sp\Big(\Big\{\Big\langle i,\ p.m.f\big(\langle[a,b],p_{ab}\rangle\big)\Big\rangle \ \Big|\ i \in [a,b]\Big\},\ x =_\bullet 0\Big)\Big) \qquad \text{[from 18]}\\
&= \boldsymbol{\alpha}\left(\left\{\left\langle 0,\ Pr(Wr) + \frac{1-Pr(Wr)}{\texttt{MAXINT}-\texttt{MININT}+1}\right\rangle\right\}\right) \qquad \text{[from 8]}\\
&= \left\langle\big[0,0\big],\ Pr(Wr) + \frac{1-Pr(Wr)}{\texttt{MAXINT}-\texttt{MININT}+1}\right\rangle \qquad \text{[from 17]}
\end{aligned} \tag{19}$$

$$\begin{aligned}
sp^{\#}\Big(\langle[a,b],p_{ab}\rangle,\ x =_\bullet x +_\bullet 3\Big) &= \left\langle\big[a+3,b+3\big],\ p_{ab}\cdot\left(Pr(Rd) + \frac{1-Pr(Rd)}{\texttt{MAXINT}-\texttt{MININT}+1}\right)\cdot\right.\\
&\left.\left(Pr(+_\bullet) + \frac{1-Pr(+_\bullet)}{\texttt{MAXINT}-\texttt{MININT}+1}\right)\cdot\left(Pr(Wr) + \frac{1-Pr(Wr)}{\texttt{MAXINT}-\texttt{MININT}+1}\right)\right\rangle
\end{aligned} \tag{20}$$

$$sp^{\#}\Big(\langle[a,b],p_{ab}\rangle,\ x \leq_\bullet 9\Big) = \begin{cases}
\left\langle\big[a,b\big],\ p_{ab}\cdot\left(Pr(\leq_\bullet) + \frac{1-Pr(\leq_\bullet)}{2}\right)\right\rangle & \text{if } b < 9\\
\left\langle\big[a,9\big],\ p_{ab}\cdot\left(Pr(\leq_\bullet) + \frac{1-Pr(\leq_\bullet)}{2}\right)\cdot\frac{9-a+1}{b-a+1}\right\rangle & \text{if } a \leq 9 \leq b\\
\bot_M & \text{if } 9 < a
\end{cases} \tag{21}$$

$$sp^{\#}\Big(\langle[a,b],p_{ab}\rangle,\ x \geq_\bullet 10\Big) = \begin{cases}
\left\langle\big[a,b\big],\ p_{ab}\cdot\left(Pr(\geq_\bullet) + \frac{1-Pr(\geq_\bullet)}{2}\right)\right\rangle & \text{if } 10 < a\\
\left\langle\big[10,b\big],\ p_{ab}\cdot\left(Pr(\geq_\bullet) + \frac{1-Pr(\geq_\bullet)}{2}\right)\cdot\frac{b-10+1}{b-a+1}\right\rangle & \text{if } a \leq 10 \leq b\\
\bot_M & \text{if } b < 10
\end{cases} \tag{22}$$

$$\begin{aligned}
sp^{\#}\Big(\bot_M = \langle[\ ],1\rangle,\ \sigma\Big) &= \boldsymbol{\alpha}\Big(sp(\boldsymbol{\gamma}(\bot_M),\ \sigma)\Big)\\
&= \boldsymbol{\alpha}\Big(sp\Big(\bot_L = \{\ \},\ \sigma\Big)\Big)\\
&= \boldsymbol{\alpha}\,(\bot_L) \qquad [\because\ sp(\bot_L,\ \sigma) = \bot_L = \{\ \},\quad \text{for any program statement } \sigma]\\
&= \bot_M
\end{aligned} \tag{23}$$

and so on . . .

Let us now try the reachability computation in the abstract domain. Let the set of abstract program states at the different program points be defined as $g_0^{\#},\ g_1^{\#},\ g_2^{\#},\ g_3^{\#}$ and the abstract version of the overall function $\mathcal{F}$ be defined as $\mathcal{F}^{\#}$. Then we shall have,

$$\begin{aligned}
g_0^{\#} &= \langle[\texttt{MININT},\ \texttt{MAXINT}],\ 1\rangle\\
g_1^{\#} &= sp^{\#}(g_0^{\#},\ x =_\bullet 0)\bigsqcup_M sp^{\#}(g_2^{\#},\ x =_\bullet x +_\bullet 3)\\
g_2^{\#} &= sp^{\#}(g_1^{\#},\ x \leq_\bullet 9)\\
g_3^{\#} &= sp^{\#}(g_1^{\#},\ x \geq_\bullet 10)
\end{aligned}$$

$$\mathcal{F}^{\#}\Big(g_0^{\#},\ g_1^{\#},\ g_2^{\#},\ g_3^{\#}\Big) = \Big(\langle[\texttt{MININT},\ \texttt{MAXINT}],\ 1\rangle,\ sp^{\#}(g_0^{\#},\ x =_\bullet 0)\bigsqcup_M sp^{\#}(g_2^{\#},\ x =_\bullet x{+_\bullet}3),\ sp^{\#}(g_1^{\#},\ x \leq_\bullet 9),\ sp^{\#}(g_1^{\#},\ x \geq_\bullet 10)\Big)$$

For demonstration purpose we again assume all the success probabilities to be $1 - 10^{-2}$, i.e $Pr(+_\bullet) = Pr(-_\bullet) = Pr(Rd) = Pr(Wr) = Pr(\leq_\bullet) = \quad \cdots \quad = 1 - 10^{-2}$ and value of MININT and MAXINT to be -32768 and 32767 respectively. Therefore, $\left(Pr(+_\bullet) + \dfrac{1 - Pr(+_\bullet)}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) = 0.9900001526 = x$ and $\left(Pr(\leq_\bullet) + \dfrac{1 - Pr(\leq_\bullet)}{2}\right) = 0.995 = y$ (say)

The least fixed point of $\mathcal{F}^\#$, i.e $(\mathcal{F}^\#)^n(\bot_M, \bot_M, \bot_M, \bot_M)$ will be our final solution. The computation of $(\mathcal{F}^\#)^n(\bot_M, \bot_M, \bot_M, \bot_M)$ shall be as follows

$\left(\bot_M,\ \bot_M,\ \bot_M,\ \bot_M\right) \rightarrow \left(\langle[\texttt{MININT}, \texttt{MAXINT}],\ 1\rangle,\ \bullet,\ \bullet,\ \bullet\right) \rightarrow \left(\bullet,\ \langle[0,0],\ x\rangle,\ \bullet,\ \bullet\right) \rightarrow \left(\bullet,\ \bullet,\ \langle[0,0],\ xy\rangle,\ \bullet\right) \rightarrow$
$\left(\bullet,\ \langle[0,3],\ 1\rangle,\ \bullet,\ \bullet\right) \rightarrow \left(\bullet,\ \bullet,\ \langle[0,3],\ y\rangle,\ \bullet\right) \rightarrow \left(\bullet,\ \langle[0,6],\ 1\rangle,\ \bullet,\ \bullet\right) \rightarrow \left(\bullet,\ \bullet,\ \langle[0,6],\ y\rangle,\ \bullet\right) \rightarrow$
$\left(\bullet,\ \langle[0,9],\ 1\rangle,\ \bullet,\ \bullet\right) \rightarrow$
$\left(\bullet,\ \bullet,\ \langle[0,9],\ y\rangle,\ \bullet\right) \rightarrow \left(\bullet,\ \langle[0,12],\ 1\rangle,\ \bullet,\ \bullet\right) \rightarrow$
$\left(\bullet,\ \bullet,\ \langle[0,9],\ y \cdot \left(\frac{10}{13}\right)\rangle,\ \bullet\right) \rightarrow \left(\bullet,\ \langle[0,12],\ x^3y\rangle,\ \bullet,\ \bullet\right) \rightarrow$
$\left(\bullet,\ \bullet,\ \langle[0,9],\ x^3y \cdot y \cdot \left(\frac{10}{13}\right)\rangle,\ \bullet\right) \rightarrow \left(\bullet,\ \langle[0,12],\ x^6y^2\rangle,\ \bullet,\ \bullet\right) \rightarrow$
$\left(\bullet,\ \bullet,\ \langle[0,9],\ x^6y^2 \cdot y \cdot \left(\frac{10}{13}\right)\rangle,\ \bullet\right) \rightarrow \left(\bullet,\ \langle[0,12],\ x^9y^3\rangle,\ \bullet,\ \bullet\right) \rightarrow$

The '•' indicates, repetition of value from previous iteration.

Appendix A.

Proof of $\sqsubseteq_L$ being a partial order

We prove that $\sqsubseteq_L$ is a partial order relation as follows.

Reflexivity :

Reflexivity follows trivially from the definition of $\sqsubseteq_L \qquad \therefore \forall A \in L \quad A \sqsubseteq_L A$.

Transitivity :

$$A \sqsubseteq_L B \qquad \bigwedge \qquad B \sqsubseteq_L C \qquad \text{(Given)}$$

$$\left\{\mathbf{val}(t) \,\middle|\, \forall t \in A\right\} \subseteq \left\{\mathbf{val}(t) \,\middle|\, \forall t \in B\right\} \qquad \bigwedge \qquad \left\{\mathbf{val}(t) \,\middle|\, \forall t \in B\right\} \subseteq \left\{\mathbf{val}(t) \,\middle|\, \forall t \in C\right\} \tag{A.1}$$

$$\forall t_a \in A, \forall t_b \in B \quad \left(\left(\mathbf{val}(t_a) = \mathbf{val}(t_b)\right) \Rightarrow \left(\mathbf{prob}(t_a) \geq \mathbf{prob}(t_b)\right)\right) \quad \bigwedge$$
$$\forall t_b \in B, \forall t_c \in C \quad \left(\left(\mathbf{val}(t_b) = \mathbf{val}(t_c)\right) \Rightarrow \left(\mathbf{prob}(t_b) \geq \mathbf{prob}(t_c)\right)\right) \tag{A.2}$$

From A.1 we get $\left\{\mathbf{val}(t) \,\middle|\, \forall t \in A\right\} \subseteq \left\{\mathbf{val}(t) \,\middle|\, \forall t \in C\right\}$ and from A.2 we get $\forall t_a \in A, \forall t_c \in C \quad \left(\left(\mathbf{val}(t_a) = \mathbf{val}(t_c)\right) \Rightarrow \left(\mathbf{prob}(t_a) \geq \mathbf{prob}(t_c)\right)\right)$.

$\therefore A \sqsubseteq_L C$

Anti-Symmetricity :

$$A \sqsubseteq_L B \quad \bigwedge \quad B \sqsubseteq_L A \tag{Given}$$

$$\Big\{\mathbf{val}(t) \,\Big|\, \forall t \in A\Big\} \subseteq \Big\{\mathbf{val}(t) \,\Big|\, \forall t \in B\Big\} \quad \bigwedge \quad \Big\{\mathbf{val}(t) \,\Big|\, \forall t \in B\Big\} \subseteq \Big\{\mathbf{val}(t) \,\Big|\, \forall t \in A\Big\} \tag{A.3}$$

$$\begin{aligned} &\forall t_a \in A, \forall t_b \in B \quad \Big(\Big(\mathbf{val}(t_a) = \mathbf{val}(t_b)\Big) \Rightarrow \Big(\mathbf{prob}(t_a) \geq \mathbf{prob}(t_b)\Big)\Big) \quad \bigwedge \\ &\forall t_b \in B, \forall t_a \in A \quad \Big(\Big(\mathbf{val}(t_b) = \mathbf{val}(t_a)\Big) \Rightarrow \Big(\mathbf{prob}(t_b) \geq \mathbf{prob}(t_a)\Big)\Big) \end{aligned} \tag{A.4}$$

From A.3 we get $\Big\{\mathbf{val}(t) \,\Big|\, \forall t \in A\Big\} = \Big\{\mathbf{val}(t) \,\Big|\, \forall t \in B\Big\}$ and from A.4 we get $\forall t_a \in A, \forall t_b \in B \quad \Big(\Big(\mathbf{val}(t_a) = \mathbf{val}(t_b)\Big) \Rightarrow \Big(\mathbf{prob}(t_a) = \mathbf{prob}(t_b)\Big)\Big)$.

$\therefore A = B$

Appendix B.

Soundness of the $l.u.b \bigsqcup_L$ and $g.l.b \bigsqcap^L$ operators in L

Let, A and B are two elements in L. From definition 4 we get the least upper bound U of A and B as

$$\begin{aligned} U = A \bigsqcup_L B = &\Big\{t_a \,\Big|\, t_a \in A \bigwedge \mathbf{val}(t_a) \notin \{\mathbf{val}(t_b) \mid t_b \in B\}\Big\} \bigcup \Big\{t_b \,\Big|\, t_b \in B \bigwedge \mathbf{val}(t_b) \notin \{\mathbf{val}(t_a) \mid t_a \in A\}\Big\} \bigcup \\ &\Big\{\langle \mathbf{val}(t_a), \mathbf{min}(\mathbf{prob}(t_a), \mathbf{prob}(t_b))\rangle \,\Big|\, t_a \in A, t_b \in B \bigwedge \mathbf{val}(t_a) = \mathbf{val}(t_b)\Big\} \end{aligned} \tag{B.1}$$

Now, C be an upper bound of both A and B.

$\therefore A \sqsubseteq_L C \qquad and \qquad B \sqsubseteq_L C$

Using our definition of $\sqsubseteq_L$ we get :

$$\Big\{\mathbf{val}(t) \,\Big|\, \forall t \in A\Big\} \subseteq \Big\{\mathbf{val}(t) \,\Big|\, \forall t \in C\Big\} \tag{B.2}$$

$$\Big\{\mathbf{val}(t) \,\Big|\, \forall t \in B\Big\} \subseteq \Big\{\mathbf{val}(t) \,\Big|\, \forall t \in C\Big\} \tag{B.3}$$

$$\forall t_a \in A,\ \forall t_c \in C \quad \Big(\Big(\mathbf{val}(t_a) = \mathbf{val}(t_c)\Big) \Rightarrow \Big(\mathbf{prob}(t_a) \geq \mathbf{prob}(t_c)\Big)\Big) \tag{B.4}$$

$$\forall t_b \in B,\ \forall t_c \in C \quad \Big(\Big(\mathbf{val}(t_b) = \mathbf{val}(t_c)\Big) \Rightarrow \Big(\mathbf{prob}(t_b) \geq \mathbf{prob}(t_c)\Big)\Big) \tag{B.5}$$

We have to show that $U \sqsubseteq_L C$. From B.1 it's obvious that

$$\begin{aligned} &\forall t \quad \Big(t \in U \Rightarrow t \in A \vee t \in B\Big) \\ \Rightarrow \quad &\forall t \quad \Big(t \in U \Rightarrow t \in A \cup B\Big) \\ \Rightarrow \quad &U \subseteq A \cup B \end{aligned} \tag{B.6}$$

Combining B.2 and B.3 we get

$$\begin{aligned}
&\left\{\mathbf{val}(t) \,\middle|\, \forall t\ (t \in A \vee t \in B)\right\} \subseteq \left\{\mathbf{val}(t) \,\middle|\, \forall t \in C\right\} \\
&\Rightarrow \left\{\mathbf{val}(t) \,\middle|\, \forall t \in A \cup B\right\} \subseteq \left\{\mathbf{val}(t) \,\middle|\, \forall t \in C\right\} \\
&\Rightarrow \left\{\mathbf{val}(t) \,\middle|\, \forall t \in U\right\} \subseteq \left\{\mathbf{val}(t) \,\middle|\, \forall t \in C\right\} \qquad \text{[from } \mathit{B}.6\text{]} \qquad \text{(B.7)}
\end{aligned}$$

Combining B.4 and B.5 we get

$$\begin{aligned}
&\forall t \in A \cup B,\ \forall t_c \in C \quad \Big(\big(\mathbf{val}(t) = \mathbf{val}(t_c)\big) \Rightarrow \big(\mathbf{prob}(t) \geq \mathbf{prob}(t_c)\big)\Big) \\
&\forall t_u \in U,\ \forall t_c \in C \quad \Big(\big(\mathbf{val}(t_u) = \mathbf{val}(t_c)\big) \Rightarrow \big(\mathbf{prob}(t_u) \geq \mathbf{prob}(t_c)\big)\Big) \qquad \text{[from } \mathit{B}.6\text{]} \qquad \text{(B.8)}
\end{aligned}$$

B.7 and B.8 together proves that $U \sqsubseteq_L C$

We omit the similar proof for greatest lower bound for the sake of brevity.

Appendix C.

Hasse Diagram of the Probabilistic Concrete Lattice $(L, \sqsubseteq_L)$

The lattice $(L, \sqsubseteq_L)$ is best viewed in 3 dimensions. In the following few pages we've given several different (cross sectional) views of the same lattice.

In all the following diagrams, **directed solid lines** represent $g.l.b \rightarrow l.u.b$ relationship i.e the element at the arrow head is the least upper bound of the element at the arrow base and vice versa. There doesn't exist any other element between the connecting elements. Whereas **directed dashed lines** represent the existence of infinite number of elements. Probability of exactly one tuple decreases monotonically along any dashed edge.

In all the following diagrams :

- $m =$ `MININT`
- $M =$ `MAXINT`
- $R = M - m$
- a_1, a_2, a_3 are placeholders.

$$\begin{tikzcd}
\{\langle a_1, 1\rangle\} \arrow[d] \\
\{\langle a_1, 0\rangle\}
\end{tikzcd}$$

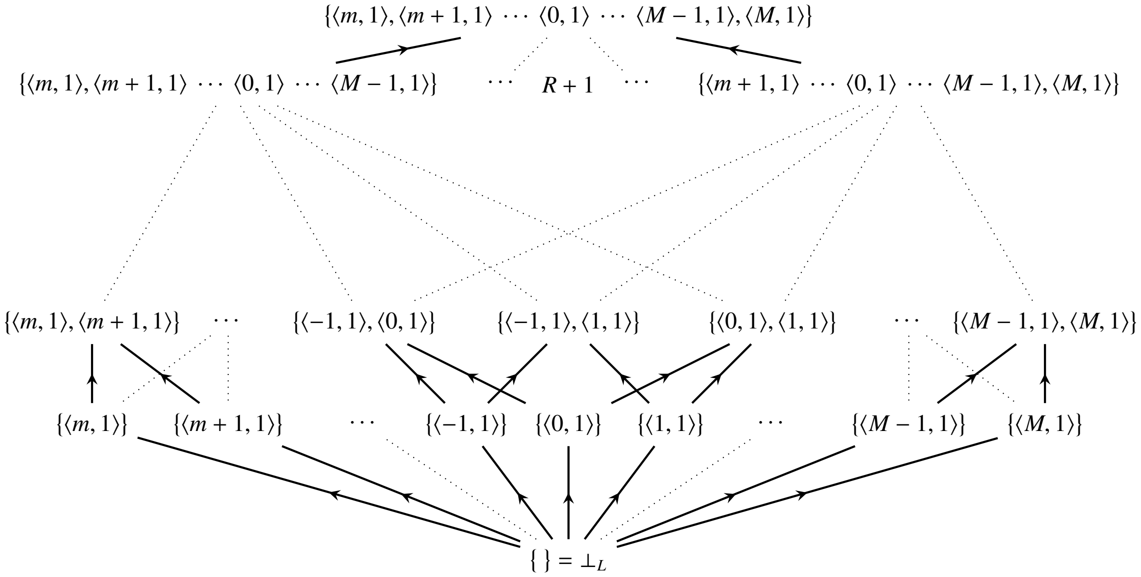
$$\begin{tikzcd}
& \{\langle a_1, 1\rangle, \langle a_2, 1\rangle\} \arrow[dl, dashed] \arrow[dr, dashed] & \\
\{\langle a_1, 1\rangle, \langle a_2, 0\rangle\} \arrow[dr, dashed] & & \{\langle a_1, 0\rangle, \langle a_2, 1\rangle\} \arrow[dl, dashed] \\
& \{\langle a_1, 0\rangle, \langle a_2, 0\rangle\} &
\end{tikzcd}$$

$$\begin{tikzcd}
& \{\langle a_1, 1\rangle, \langle a_2, 1\rangle, \langle a_3, 1\rangle\} \arrow[dl, dashed] \arrow[d] \arrow[dr, dashed] & \\
\{\langle a_1, 1\rangle, \langle a_2, 1\rangle, \langle a_3, 0\rangle\} \arrow[d] \arrow[dr, dashed] & \{\langle a_1, 1\rangle, \langle a_2, 0\rangle, \langle a_3, 1\rangle\} \arrow[dl, dashed] \arrow[dr, dashed] & \{\langle a_1, 0\rangle, \langle a_2, 1\rangle, \langle a_3, 1\rangle\} \arrow[dl, dashed] \arrow[d] \\
\{\langle a_1, 1\rangle, \langle a_2, 0\rangle, \langle a_3, 0\rangle\} \arrow[dr, dashed] & \{\langle a_1, 0\rangle, \langle a_2, 1\rangle, \langle a_3, 0\rangle\} \arrow[d] & \{\langle a_1, 0\rangle, \langle a_2, 0\rangle, \langle a_3, 1\rangle\} \arrow[dl, dashed] \\
& \{\langle a_1, 0\rangle, \langle a_2, 0\rangle, \langle a_3, 0\rangle\} &
\end{tikzcd}$$

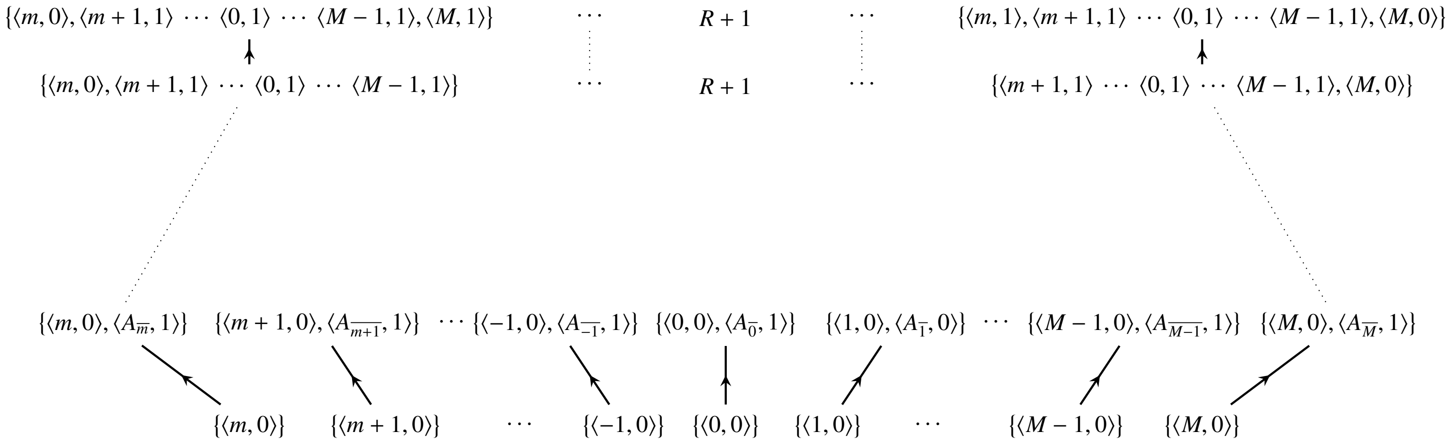
$$\vdots$$

Top View (Cross Sectional)[3]

[3] For brevity we've shown only for elements with $1, 2$ & 3 tuples.



Frontal View


{⟨m, 0⟩, ⟨m + 1, 1⟩ ··· ⟨0, 1⟩ ··· ⟨M − 1, 1⟩, ⟨M, 1⟩}
··· R + 1 ···
{⟨m, 1⟩, ⟨m + 1, 1⟩ ··· ⟨0, 1⟩ ··· ⟨M − 1, 1⟩, ⟨M, 0⟩}
{⟨m, 0⟩, ⟨m + 1, 1⟩ ··· ⟨0, 1⟩ ··· ⟨M − 1, 1⟩}
··· R + 1 ···
{⟨m + 1, 1⟩ ··· ⟨0, 1⟩ ··· ⟨M − 1, 1⟩, ⟨M, 0⟩}
{⟨m, 0⟩, ⟨A_m̄, 1⟩}
{⟨m + 1, 0⟩, ⟨A_(m+1)‾, 1⟩}
··· {⟨−1, 0⟩, ⟨A_(−1)‾, 1⟩}
{⟨0, 0⟩, ⟨A_0̄, 1⟩}
{⟨1, 0⟩, ⟨A_1̄, 0⟩}
··· {⟨M − 1, 0⟩, ⟨A_(M−1)‾, 1⟩}
{⟨M, 0⟩, ⟨A_M̄, 1⟩}
{⟨m, 0⟩}
{⟨m + 1, 0⟩}
···
{⟨−1, 0⟩}
{⟨0, 0⟩}
{⟨1, 0⟩}
···
{⟨M − 1, 0⟩}
{⟨M, 0⟩}

Frontal View (Cross Sectional)

Here, Each element has exactly one tuple with 0 probability and $A_{\overline{p}} \in \{m, m+1 \ \cdots \ M-1, M\} \setminus \{p\} \qquad \forall p \in [m, M]$

$$\begin{tikzcd}
\{\langle m,1\rangle,\langle m+1,1\rangle\ \cdots\ \langle 0,1\rangle\ \cdots\ \langle M-1,1\rangle,\langle M,1\rangle\} \arrow[rrr, dotted] & & & \{\langle m,0\rangle,\langle m+1,0\rangle\ \cdots\ \langle 0,0\rangle\ \cdots\ \langle M-1,0\rangle,\langle M,0\rangle\} = \top_L \\
\{\langle a_1,1\rangle,\langle a_2,1\rangle,\langle a_3,1\rangle\} \arrow[u, dotted] \arrow[r, dashed] & \{\langle a_1,0\rangle,\langle a_2,1\rangle,\langle a_3,1\rangle\} \arrow[r, dashed] & \{\langle a_1,0\rangle,\langle a_2,0\rangle,\langle a_3,1\rangle\} \arrow[r, dashed] & \{\langle a_1,0\rangle,\langle a_2,0\rangle,\langle a_3,0\rangle\} \arrow[u, dotted] \\
\{\langle a_1,1\rangle,\langle a_2,1\rangle\} \arrow[u] \arrow[r, dashed] & \{\langle a_1,0\rangle,\langle a_2,1\rangle\} \arrow[u] \arrow[r, dashed] & \{\langle a_1,0\rangle,\langle a_2,0\rangle\} \arrow[u] & \\
\{\langle a_1,1\rangle\} \arrow[u] \arrow[r, dashed] & \{\langle a_1,0\rangle\} \arrow[u] & & \\
\{\ \} = \bot_L \arrow[u] & & &
\end{tikzcd}$$

Lateral View (Cross Sectional)

Here,

- $\bot_L = \{\ \}$ $\qquad \top_L = \{\langle m,0\rangle,\langle m+1,0\rangle\ \cdots\ \langle 0,0\rangle\ \cdots\ \langle M-1,0\rangle,\langle M,0\rangle\}$
- a_1, a_2 and a_3 are placeholders with $a_1, a_2, a_3 \in [m, M]$

Appendix D.

Proof of $\sqsubseteq_M$ being a partial order

We prove that $\sqsubseteq_M$ is a partial order relation as follows.

Reflexivity :

It's trivial to show that $\langle [a,b], p_{ab}\rangle \sqsubseteq_M \langle [a,b], p_{ab}\rangle$. It follows directly from our definition of $\sqsubseteq_M$.

Transitivity :

$\langle [a,b], p_{ab}\rangle \sqsubseteq_M \langle [c,d], p_{cd}\rangle$ and $\langle [c,d], p_{cd}\rangle \sqsubseteq_M \langle [e,f], p_{ef}\rangle$ (Given)

$$[a,b] \sqsubseteq_{int} [c,d] \quad \wedge \quad [c,d] \sqsubseteq_{int} [e,f] \tag{D.1}$$

$$p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \geq\ p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big) \quad \wedge \quad p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big) \geq\ p.m.f\Big(\langle [e,f], p_{ef}\rangle\Big) \tag{D.2}$$

From D.1 we get $[a,b] \sqsubseteq_{int} [e,f]$ and from D.2 we get $p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \geq\ p.m.f\Big(\langle [e,f], p_{ef}\rangle\Big)$.

$\therefore \langle [a,b], p_{ab}\rangle \sqsubseteq_M \langle [e,f], p_{ef}\rangle$

Anti-Symmetricity :

$$\langle [a,b], p_{ab}\rangle \sqsubseteq_M \langle [c,d], p_{cd}\rangle \qquad \text{and} \qquad \langle [c,d], p_{cd}\rangle \sqsubseteq_M \langle [a,b], p_{ab}\rangle \qquad \text{(Given)}$$

$$[a,b] \sqsubseteq_{int} [c,d] \implies (c \le a) \wedge (b \le d) \qquad \Big[\text{using definition 13}\Big] \tag{D.3}$$

$$[c,d] \sqsubseteq_{int} [a,b] \implies (a \le c) \wedge (d \le b) \qquad \Big[\text{using definition 13}\Big] \tag{D.4}$$

$$p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \ge p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big) \tag{D.5}$$

$$p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big) \ge p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \tag{D.6}$$

Combining D.3 and D.4 we get,

$(c \le a) \wedge (a \le c) \implies a = c$ and

$(b \le d) \wedge (d \le b) \implies b = d$

$\therefore \underline{a = c,\ b = d}$

From D.5 and D.6 we get,

$$p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) = p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big)$$

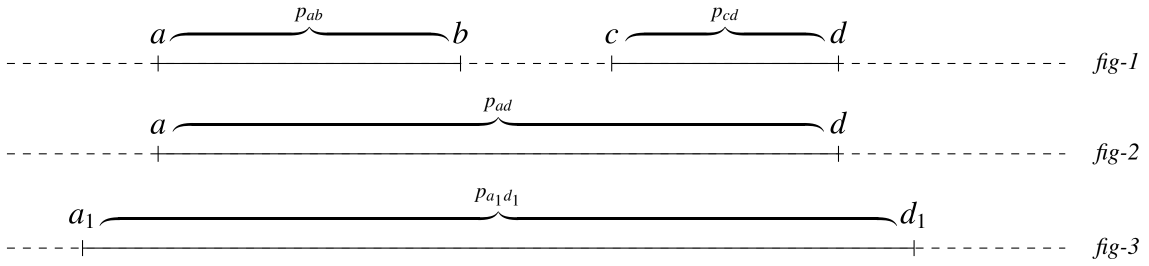
$$\Rightarrow \frac{p_{ab}}{b-a+1} = \frac{p_{cd}}{d-c+1} \qquad \Big[\text{using definition 12}\Big]$$

$$\Rightarrow \underline{p_{ab} = p_{cd}} \qquad \Big[\because a = c,\ b = d \quad \text{and} \quad b-a+1 \ne 0\Big]$$

$\therefore \langle [a,b], p_{ab}\rangle = \langle [c,d], p_{cd}\rangle$

Appendix E.

Proof of correctness of the $l.u.b \bigsqcup_M$ and $g.l.b \bigsqcap^M$ operators in M



For the two elements $\langle [a,b], p_{ab}\rangle$ and $\langle [c,d], p_{cd}\rangle$ (in *fig-B1*) suppose the $l.u.b$ is $\langle [x,y], p_{xy}\rangle$, i.e

$$\langle [a,b], p_{ab}\rangle \bigsqcup_M \langle [c,d], p_{cd}\rangle = \langle [x,y], p_{xy}\rangle$$

From definition 14 we get,

$$x = \mathbf{min}(a,c) = a \qquad y = \mathbf{max}(b,d) = d \qquad p_{xy} = (d-a+1)\cdot \mathbf{min}\left(p.m.f\Big(\langle [a,\ b],\ p_{ab}\rangle\Big),\ p.m.f\Big(\langle [c,\ d],\ p_{cd}\rangle\Big),\ \frac{1}{d-a+1}\right)$$

$\therefore$ $\langle [a,d], p_{ad}\rangle$ (in *fig-2*) is the $l.u.b$ of the elements in *fig-1* with $p_{ad} = p_{xy}$. We show all upper bounds of $\langle [a,b], p_{ab}\rangle$ and $\langle [c,d], p_{cd}\rangle$ come higher in the order than $\langle [a,d], p_{ad}\rangle$.

It's trivial to show that for any tuple of the form $\langle [a,d], p'_{ad}\rangle$, where $p'_{ad} \le p_{ad}$[4], $\langle [a,d], p_{ad}\rangle \sqsubseteq_M \langle [a,d], p'_{ad}\rangle$ (by the definition of $\sqsubseteq_M$). So, $\langle [a,d], p_{ad}\rangle$ remains the $l.u.b$.

[4] If $p'_{ad} > p_{ad}$, then $\langle [a,d], p'_{ad}\rangle$ is **NOT** an upper bound of both $\langle [a,b], p_{ab}\rangle$ and $\langle [c,d], p_{cd}\rangle$

Now, the other form of tuple i.e $\langle [a_1, d_1], p_{a_1d_1}\rangle$ (in *fig-3*), where $[a,d] \sqsubseteq_{int} [a_1,d_1]$, will be an upper bound of $\langle [a,b], p_{ab}\rangle$ and $\langle [c,d], p_{cd}\rangle$ if

$$\begin{aligned}
&\langle [a,b], p_{ab}\rangle \sqsubseteq_M \langle [a_1,d_1], p_{a_1d_1}\rangle \qquad \bigwedge \qquad \langle [c,d], p_{cd}\rangle \sqsubseteq_M \langle [a_1,d_1], p_{a_1d_1}\rangle\\
&\Rightarrow p.m.f\Big(\langle [a_1,d_1], p_{a_1d_1}\rangle\Big) \le p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \bigwedge p.m.f\Big(\langle [a_1,d_1], p_{a_1d_1}\rangle\Big) \le p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big) \Big[\because \text{we know } [a,d] \sqsubseteq_{int} [a_1,d_1]\Big]\\
&\Rightarrow p_{a_1d_1} \le (d_1 - a_1 + 1)\cdot p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \bigwedge p_{a_1d_1} \le (d_1 - a_1 + 1)\cdot p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big) \qquad \Big[\text{using definition 12}\Big]\\
&\Rightarrow p_{a_1d_1} \le \mathbf{min}\Big((d_1 - a_1 + 1)\cdot p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big),\ (d_1 - a_1 + 1)\cdot p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big),\ 1\Big) \qquad \Big[\because p_{a_1d_1} \le 1\Big]\\
&\Rightarrow p_{a_1d_1} \le (d_1 - a_1 + 1)\ \cdot\ \mathbf{min}\left(p.m.f\Big(\langle [a,\ b],\ p_{ab}\rangle\Big),\ p.m.f\Big(\langle [c,\ d],\ p_{cd}\rangle\Big),\ \frac{1}{d_1 - a_1 + 1}\right)
\end{aligned}$$

We need to prove that, $p.m.f\Big(\langle [a,\ d],\ p_{ad}\rangle\Big) \ge p.m.f\Big(\langle [a_1,\ d_1],\ p_{a_1d_1}\rangle\Big)$, in order to prove that $\langle [a,\ d],\ p_{ad}\rangle$ still remains the *l.u.b*.

Without any loss of generality we assume that

$$p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \le p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big) \tag{E.1}$$

$$\frac{1}{d_1 - a_1 + 1} < \frac{1}{d - a + 1} \tag{E.2}$$

$$\left.\begin{aligned}
p_{ad} &= (d - a + 1)\ \cdot\ \mathbf{min}\left(p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big),\ p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big),\ \frac{1}{d - a + 1}\right)\\
&= (d - a + 1)\ \cdot\ \mathbf{min}\left(p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big),\ \frac{1}{d - a + 1}\right) \qquad [\text{from } E.1]\\
p_{a_1d_1} &= (d_1 - a_1 + 1)\ \cdot\ \mathbf{min}\left(p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big),\ p.m.f\Big(\langle [c,d], p_{cd}\rangle\Big),\ \frac{1}{d_1 - a_1 + 1}\right)\\
&= (d_1 - a_1 + 1)\ \cdot\ \mathbf{min}\left(p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big),\ \frac{1}{d_1 - a_1 + 1}\right) \qquad [\text{from } E.1]
\end{aligned}\right\} \text{(Given)}$$

In view of E.2 there are three cases to consider now.

Case 1 : $\quad p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \le \dfrac{1}{d_1 - a_1 + 1}$

$$\therefore p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) < \frac{1}{d - a + 1} \qquad [\text{from } E.2]$$

$$\therefore p_{ad} = (d - a + 1)\ \cdot\ p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \qquad p_{a_1d_1} = (d_1 - a_1 + 1)\ \cdot\ p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big)$$

$$\therefore p.m.f\Big(\langle [a,d], p_{ad}\rangle\Big) = p.m.f\Big(\langle [a_1,d_1], p_{a_1d_1}\rangle\Big)$$

Case 2 : $\quad \dfrac{1}{d_1 - a_1 + 1} < p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) < \dfrac{1}{d - a + 1}$

$$\therefore p_{ad} = (d - a + 1)\ \cdot\ p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \qquad p_{a_1d_1} = 1$$

$$\therefore p.m.f\Big(\langle [a,d], p_{ad}\rangle\Big) = p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) \qquad p.m.f\Big(\langle [a_1,d_1], p_{a_1d_1}\rangle\Big) = \frac{1}{d_1 - a_1 + 1}$$

$$\therefore p.m.f\Big(\langle [a,d], p_{ad}\rangle\Big) > p.m.f\Big(\langle [a_1,d_1], p_{a_1d_1}\rangle\Big)$$

Case 3 : $\quad \dfrac{1}{d - a + 1} \le p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big)$

$$\therefore p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big) > \frac{1}{d_1 - a_1 + 1} \qquad [\text{from } E.2]$$

$$\therefore p_{ad} = 1 \qquad p_{a_1d_1} = 1$$

$$\therefore p.m.f\Big(\langle [a,d], p_{ad}\rangle\Big) = \ p.m.f\Big(\langle [a_1,d_1], p_{a_1d_1}\rangle\Big)$$

In all the above three cases $p.m.f\Big(\langle [a,d], p_{ad}\rangle\Big) \ge p.m.f\Big(\langle [a_1,d_1], p_{a_1d_1}\rangle\Big)$. Hence $\langle [a,d], p_{ad}\rangle$ is the *l.u.b*

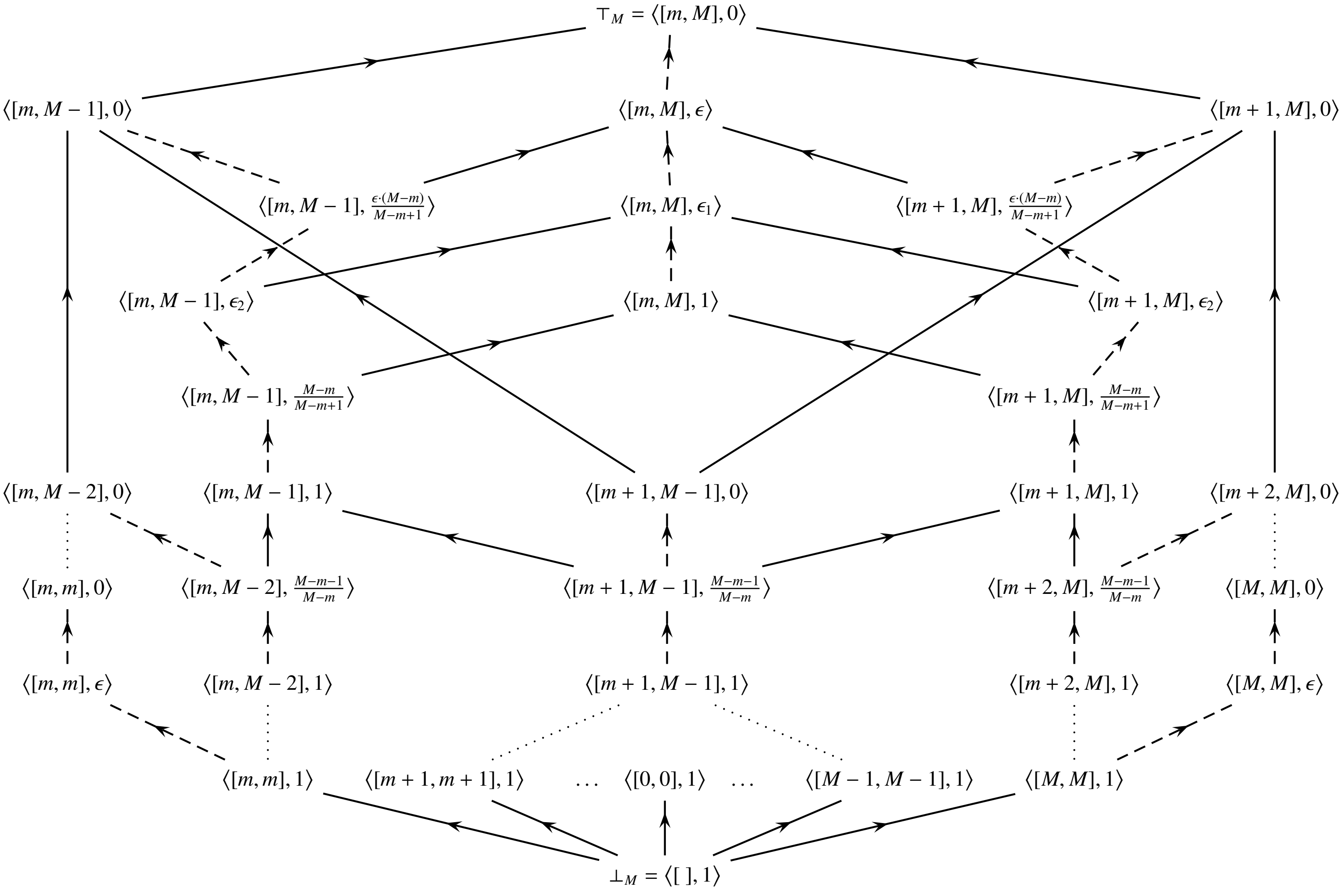
Analogous arguments can be applied to show the soundness of the greatest lower bound, $\overset{M}{\sqcap}$ definition. For brevity we're omitting the proof for *g.l.b* here.

Appendix F.

Hasse Diagram of the Probabilistic Interval Lattice $(M, \sqsubseteq_M)$

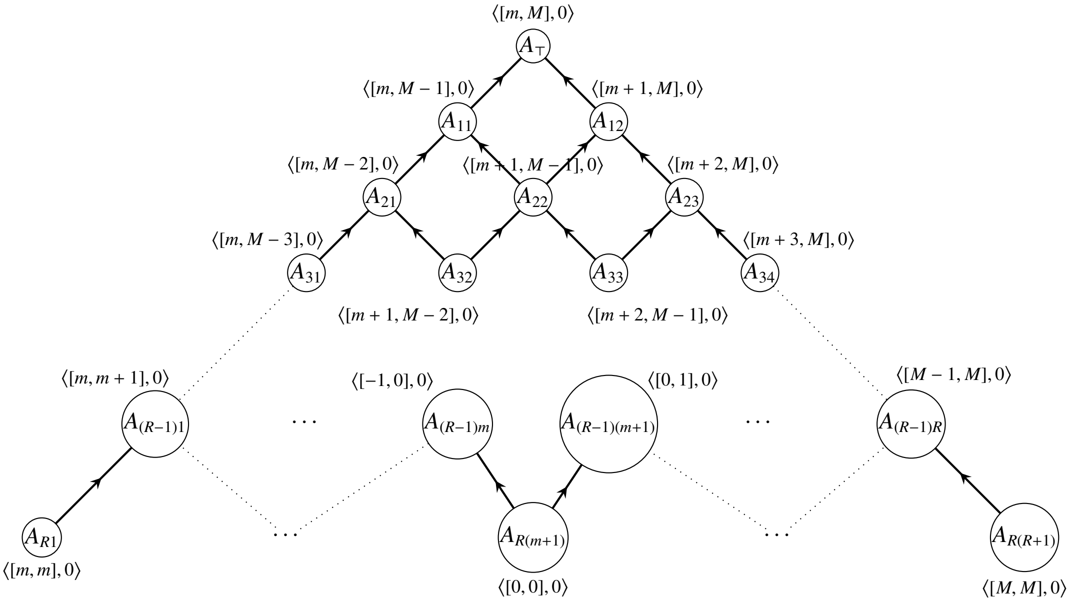
The lattice $(M, \sqsubseteq_M)$ is best viewed in 3 dimensions. For that reason we've given 4 different views of the same lattice.

In all the following diagrams, **solid lines** represent direct parent-child relationship i.e there doesn't exist any other element between the connecting nodes and **dashed lines** represent the existence of infinite number of elements having same intervals as that of the connecting nodes but probability that varies monotonically between the connecting nodes.


$\top_M = \langle [m, M], 0\rangle$
$\langle [m, M-1], 0\rangle$
$\langle [m, M], \epsilon\rangle$
$\langle [m+1, M], 0\rangle$
$\langle [m, M-1], \frac{\epsilon\cdot(M-m)}{M-m+1}\rangle$
$\langle [m, M], \epsilon_1\rangle$
$\langle [m+1, M], \frac{\epsilon\cdot(M-m)}{M-m+1}\rangle$
$\langle [m, M-1], \epsilon_2\rangle$
$\langle [m, M], 1\rangle$
$\langle [m+1, M], \epsilon_2\rangle$
$\langle [m, M-1], \frac{M-m}{M-m+1}\rangle$
$\langle [m+1, M], \frac{M-m}{M-m+1}\rangle$
$\langle [m, M-2], 0\rangle$
$\langle [m, M-1], 1\rangle$
$\langle [m+1, M-1], 0\rangle$
$\langle [m+1, M], 1\rangle$
$\langle [m+2, M], 0\rangle$
$\langle [m, m], 0\rangle$
$\langle [m, M-2], \frac{M-m-1}{M-m}\rangle$
$\langle [m+1, M-1], \frac{M-m-1}{M-m}\rangle$
$\langle [m+2, M], \frac{M-m-1}{M-m}\rangle$
$\langle [M, M], 0\rangle$
$\langle [m, m], \epsilon\rangle$
$\langle [m, M-2], 1\rangle$
$\langle [m+1, M-1], 1\rangle$
$\langle [m+2, M], 1\rangle$
$\langle [M, M], \epsilon\rangle$
$\langle [m, m], 1\rangle$
$\langle [m+1, m+1], 1\rangle$
$\dots$
$\langle [0, 0], 1\rangle$
$\dots$
$\langle [M-1, M-1], 1\rangle$
$\langle [M, M], 1\rangle$
$\bot_M = \langle [\,], 1\rangle$

2D Projectional View

*legends are defined in the next page



$\langle [m, M], 0\rangle$
$A_\top$
$\langle [m, M-1], 0\rangle$
A_{11}
$\langle [m+1, M], 0\rangle$
A_{12}
$\langle [m, M-2], 0\rangle$
A_{21}
$\langle [m+1, M-1], 0\rangle$
A_{22}
$\langle [m+2, M], 0\rangle$
A_{23}
$\langle [m, M-3], 0\rangle$
A_{31}
A_{32}
A_{33}
$\langle [m+3, M], 0\rangle$
A_{34}
$\langle [m+1, M-2], 0\rangle$
$\langle [m+2, M-1], 0\rangle$
$\langle [m, m+1], 0\rangle$
$A_{(R-1)1}$
$\langle [-1, 0], 0\rangle$
$A_{(R-1)m}$
$\langle [0, 1], 0\rangle$
$A_{(R-1)(m+1)}$
$\langle [M-1, M], 0\rangle$
$A_{(R-1)R}$
A_{R1}
$\langle [m, m], 0\rangle$
$A_{R(m+1)}$
$\langle [0, 0], 0\rangle$
$A_{R(R+1)}$
$\langle [M, M], 0\rangle$

<u>Frontal View</u>

In the 2D Projectional View :

- m = `MININT`
- M = `MAXINT`
- $0 < \epsilon, \epsilon_1, \epsilon_2 < 1$
- $\epsilon_2 = \dfrac{\epsilon_1 \cdot (M-m)}{M-m+1}$ [5]

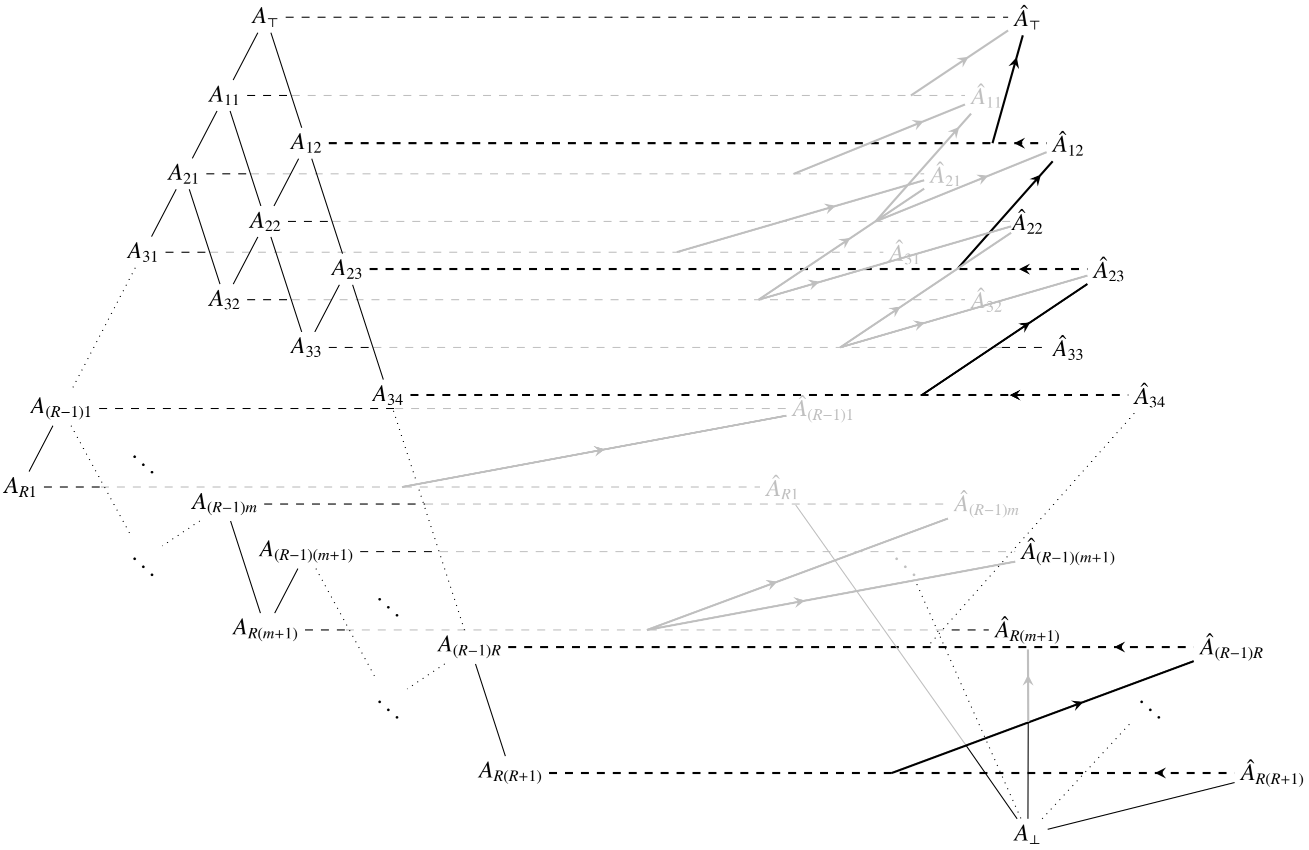
In the Frontal & Lateral View :

$A_{Ri} \equiv A_{(M-m)i}$ and $\hat{A}_{Ri} \equiv \hat{A}_{(M-m)i}$ $\forall i$

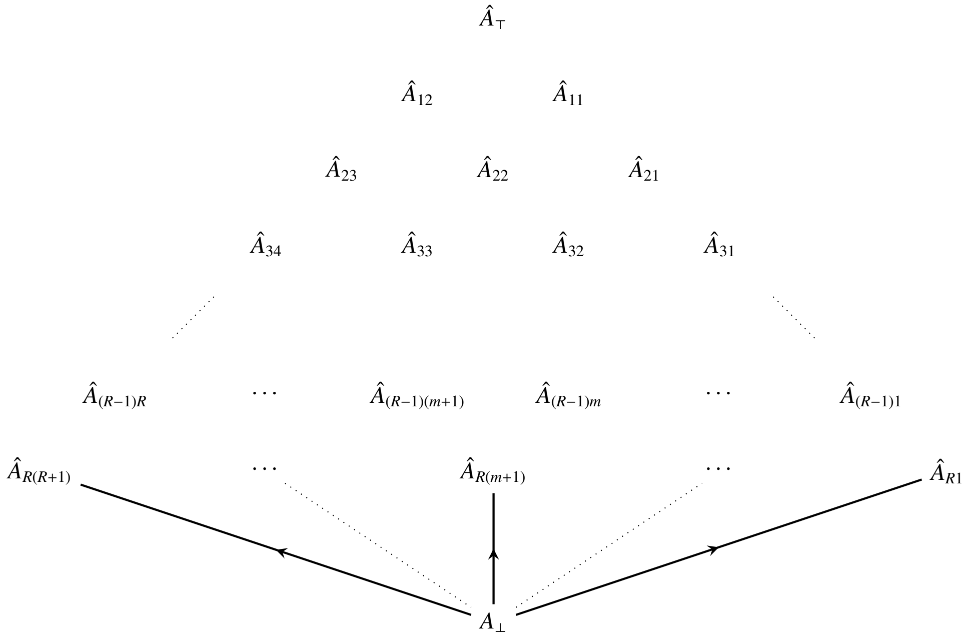
For every pair of indices i, j (and $\top$) the elements denoted by A_{ij} (and $A_\top$) and $\hat{A}_{ij}$ (and $\hat{A}_\top$) have the same interval, but probability in element A_{ij} (and $A_\top$) is 0 , wherese it is 1 for $\hat{A}_{ij}$ (and $\hat{A}_\top$).

E.g: $A_{22} \equiv \langle [m+1,\ M-1],\ 0\rangle$ but $\hat{A}_{22} \equiv \langle [m+1,\ M-1],\ 1\rangle$

[5] In general probability of an element = $\dfrac{\text{(Probability of its direct parent)} \times \text{(No. of elements in its own interval)}}{\text{(No. of elements in its direct parent's interval)}}$



Lateral View (Skewed)



Rear View

Appendix G.

Proof of monotonicity of Galois connection (α, γ)

Let, $\langle [a,b], p_{ab} \rangle$ and $\langle [c,d], p_{cd} \rangle$ be two elements in M, with $\langle [a,b], p_{ab} \rangle \sqsubseteq_M \langle [c,d], p_{cd} \rangle$. We show that

$$\boldsymbol{\gamma}\Big(\langle [a,b], p_{ab} \rangle\Big) \sqsubseteq_L \boldsymbol{\gamma}\Big(\langle [c,d], p_{cd} \rangle\Big)$$ [6]

$\because \langle [a,b], p_{ab} \rangle \sqsubseteq_M \langle [c,d], p_{cd} \rangle$, from our definition of $\sqsubseteq_M$ we have

$$\begin{aligned} &[a,b] \sqsubseteq_{int} [c,d] \qquad \text{and} \\ &p.m.f(\langle [a,b], p_{ab} \rangle) \geq p.m.f(\langle [c,d], p_{cd} \rangle) \end{aligned} \tag{G.1}$$

$$\begin{aligned} & [a,b] \sqsubseteq_{int} [c,d] \\ \Rightarrow\ & c \leq a \bigwedge b \leq d && \text{[using definition 13]} \\ \Rightarrow\ & \{i \mid i \in [a,b]\} \subseteq \{i \mid i \in [c,d]\} \end{aligned} \tag{G.2}$$

$$\boldsymbol{\gamma}\Big(\langle [a,b], p_{ab} \rangle\Big) = \Big\{\langle i, p.m.f(\langle [a,b], p_{ab} \rangle) \rangle \ \Big| \ i \in [a,b]\Big\} = X \text{ (say)} \qquad \text{and}$$

$$\boldsymbol{\gamma}\Big(\langle [c,d], p_{cd} \rangle\Big) = \Big\{\langle i, p.m.f(\langle [c,d], p_{cd} \rangle) \rangle \ \Big| \ i \in [c,d]\Big\} = Y \text{ (say)}$$

From G.2 we get $\{\mathbf{val}(t_x) \mid t_x \in X\} \subseteq \{\mathbf{val}(t_y) \mid t_y \in Y\}$ and

from G.1 we get $\forall t_x \in X, \forall t_y \in Y \left(\Big(\mathbf{val}(t_x) = \mathbf{val}(t_y)\Big) \Rightarrow \Big(\mathbf{prob}(t_x) \geq \mathbf{prob}(t_y)\Big)\right)$

$\therefore X \sqsubseteq_L Y$ [from our definition of $\sqsubseteq_L$].

$\therefore \boldsymbol{\gamma}\Big(\langle [a,b], p_{ab} \rangle\Big) \sqsubseteq_L \boldsymbol{\gamma}\Big(\langle [c,d], p_{cd} \rangle\Big)$

This completes the proof that $\boldsymbol{\gamma}$ is monotonic.

Now, suppose A and B are two elements in L, with $A \sqsubseteq_L B$ and $A, B \neq \{\ \}$. We show that

$$\boldsymbol{\alpha}\Big(A\Big) \sqsubseteq_M \boldsymbol{\alpha}\Big(B\Big)$$ [7]

$\because A \sqsubseteq_L B$, from the definition of $\sqsubseteq_L$ we have

$$\Big\{\mathbf{val}(t) \ \Big| \ \forall t \in A\Big\} \subseteq \Big\{\mathbf{val}(t) \ \Big| \ \forall t \in B\Big\} \tag{G.3}$$

$$\forall t_a \in A,\ \forall t_b \in B \quad \left(\Big(\mathbf{val}(t_a) = \mathbf{val}(t_b)\Big) \Rightarrow \Big(\mathbf{prob}(t_a) \geq \mathbf{prob}(t_b)\Big)\right) \tag{G.4}$$

$$\boldsymbol{\alpha}(A) = \left\langle \Big[\mathbf{MIN_v}(A),\ \mathbf{MAX_v}(A)\Big],\ \mathbf{min}\Big(1,\ \mathbf{MIN_p}(A) * \Big(\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1\Big)\Big)\right\rangle \qquad \text{and}$$

$$\boldsymbol{\alpha}(B) = \left\langle \Big[\mathbf{MIN_v}(B),\ \mathbf{MAX_v}(B)\Big],\ \mathbf{min}\Big(1,\ \mathbf{MIN_p}(B) * \Big(\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1\Big)\Big)\right\rangle$$

From G.3 we get

$$\begin{aligned} & \mathbf{MIN_v}(B) \leq \mathbf{MIN_v}(A) \bigwedge \mathbf{MAX_v}(A) \leq \mathbf{MAX_v}(B) \\ \Rightarrow\ & [\mathbf{MIN_v}(A), \mathbf{MAX_v}(A)] \sqsubseteq_{int} [\mathbf{MIN_v}(B), \mathbf{MAX_v}(B)] && \text{[using definition 13]} \end{aligned} \tag{G.5}$$

Let $t_a \in A$ and $t_b \in B$ such that $\mathbf{prob}(t_a) = \mathbf{MIN_p}(A) \wedge \mathbf{val}(t_a) = \mathbf{val}(t_b)$. Two such tuples always exist because $A, B \neq \{\ \}$ and G.3 holds. So, From G.4 we get

$$\begin{aligned} & \mathbf{prob}(t_a) \geq \mathbf{prob}(t_b) \\ \Rightarrow\ & \mathbf{MIN_p}(A) \geq \mathbf{prob}(t_b) && [\because \mathbf{prob}(t_a) = \mathbf{MIN_p}(A)] \\ \Rightarrow\ & \mathbf{MIN_p}(A) \geq \mathbf{prob}(t_b) \geq \mathbf{MIN_p}(B) && [\because \mathbf{MIN_p}(B) \text{ is the minimum probability among all tuples in } B] \\ \Rightarrow\ & \mathbf{MIN_p}(A) \geq \mathbf{MIN_p}(B) \end{aligned} \tag{G.6}$$

[6] This is trivial to show that, if $\bot_M \sqsubseteq_M \langle [a,b], p_{ab} \rangle$ then $\boldsymbol{\gamma}\Big(\bot_M\Big) = \bot_L \sqsubseteq_L \boldsymbol{\gamma}\Big(\langle [a,b], p_{ab} \rangle\Big)$ using definition 18

[7] It's a trivial case, if $\bot_L \sqsubseteq_L S, \forall S \in L$ then $\boldsymbol{\alpha}\Big(\bot_L\Big) = \bot_M \sqsubseteq_M \boldsymbol{\alpha}\Big(S\Big)$ using definition 17

From G.5 we have

$$\begin{aligned}
&\big(\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1\big) \leq \big(\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1\big) \\
\Rightarrow\ &\frac{1}{\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1} \geq \frac{1}{\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1}
\end{aligned} \tag{G.7}$$

Now, depending upon the value of $p.m.f\big(\alpha(A)\big)$ and $p.m.f\big(\alpha(B)\big)$ we have 4 different cases to consider.

Case 1 : $p.m.f\big(\alpha(A)\big) = \mathbf{MIN_p}(A)$ and $p.m.f\big(\alpha(B)\big) = \mathbf{MIN_p}(B)$

From G.6 it's obvious that $p.m.f\big(\alpha(A)\big) \geq p.m.f\big(\alpha(B)\big)$

Case 2 : $p.m.f\big(\alpha(A)\big) = \mathbf{MIN_p}(A)$ and $p.m.f\big(\alpha(B)\big) = \dfrac{1}{\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1}$

$$\because\ p.m.f\big(\alpha(B)\big) = \frac{1}{\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1}$$

$$\Rightarrow\ 1 \leq \mathbf{MIN_p}(B) * \big(\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1\big)$$

$$\Rightarrow\ \mathbf{MIN_p}(B) \geq \frac{1}{\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1}$$

$$\Rightarrow\ \mathbf{MIN_p}(A) \geq \frac{1}{\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1} \qquad \text{[From } G.6\text{]}$$

$$\therefore\ p.m.f\big(\alpha(A)\big) \geq p.m.f\big(\alpha(B)\big)$$

Case 3 : $p.m.f\big(\alpha(A)\big) = \dfrac{1}{\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1}$ and $p.m.f\big(\alpha(B)\big) = \mathbf{MIN_p}(B)$

$$\because\ p.m.f\big(\alpha(B)\big) = \mathbf{MIN_p}(B)$$

$$\Rightarrow\ 1 \geq \mathbf{MIN_p}(B) * \big(\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1\big)$$

$$\Rightarrow\ \mathbf{MIN_p}(B) \leq \frac{1}{\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1}$$

$$\Rightarrow\ \mathbf{MIN_p}(B) \leq \frac{1}{\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1} \qquad \text{[From } G.7\text{]}$$

$$\therefore\ p.m.f\big(\alpha(A)\big) \geq p.m.f\big(\alpha(B)\big)$$

Case 4 : $p.m.f\big(\alpha(A)\big) = \dfrac{1}{\mathbf{MAX_v}(A) - \mathbf{MIN_v}(A) + 1}$ and $p.m.f\big(\alpha(B)\big) = \dfrac{1}{\mathbf{MAX_v}(B) - \mathbf{MIN_v}(B) + 1}$

From G.7 it's obvious that $p.m.f\big(\alpha(A)\big) \geq p.m.f\big(\alpha(B)\big)$

So, in all the 4 cases $p.m.f\big(\alpha(A)\big) \geq p.m.f\big(\alpha(B)\big)$. This along with G.5 proves that $\alpha\big(A\big) \sqsubseteq_M \alpha\big(B\big)$. This completes the proof that α is monotonic.

References

[1] Lecturecise 14: From lattices to abstract interpretation, 2013. http://lara.epfl.ch/w/_media/sav13:lecturecise14.pdf.

[2] Assale Adje, Olivier Bouissou, Jean Goubault-Larrecq, Eric Goubault, and Sylvie Putot. Static analysis of programs with imprecise probabilistic inputs. 5th *International Conference VSTTE*, pages 22–47, 2013. http://www.lix.polytechnique.fr/Labo/Sylvie.Putot/Publications/vstte13.pdf.

[3] Michael Carbin, Sasa Misailovic, and Martin C. Rinard. Verifying quantitative reliability for programs that execute on unreliable hardware. *SIGPLAN Not.*, 48(10), 2013. http://doi.acm.org/10.1145/2544173.2509546.

[4] Monniaux David. Backwards abstract interpretation of probabilistic programs. 10th *European Symposium on Programming*, pages 367–382, 2001. http://www-verimag.imag.fr/~monniaux/biblio/Monniaux_ESOP01.pdf.

[5] F. Nielson, H.R. Nielson, and C. Hankin. *Principles of Program Analysis*. Springer, 1999.

[6] Sriram Sankaranarayanan, Aleksandar Chakarov, and Sumit Gulwani. Static analysis for probabilistic programs: inferring whole program properties from finitely many paths. *SIGPLAN Not.*, pages 447–458, 2013. http://doi.acm.org/10.1145/2499370.2462179.