Interval Analysis of Unreliable Programs[☆]

Dibyendu Das[a,*]

[a]*Department of Computer Science & Engineering, IIT Kharagpur, West Bengal 721302, India*

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 extra speed we'll simply have to let our computers make more mistakes in computations.

Our objective here 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*.

Keywords: Interval Analysis, Abstract Interpretation, Static Analysis, Probabilistic Programming Language

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 popular static analysis tool for probabilistic programs
[*]Corresponding author
Email address: dibyendu.das.in@gmail.com (Dibyendu Das)

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 a static analysis tool called **Range Analysis** or **Interval Analysis** for unreliable programs written for these type of architecture. 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. 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 [5][7]. 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.

3. Unreliable Programs

Following is an example program snippet and its corresponding **C**ontrol **F**low **G**raph of a program written for an unreliable system in which every operation (ALU or memory) is unrel.

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

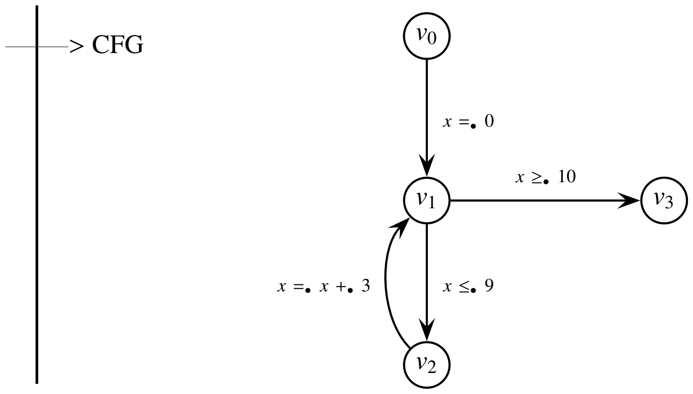


Figure 1: Example program.

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* ($\bullet$). 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 $\mathbb{Z} \cap [\texttt{MININT}, \texttt{MAXINT}]$, 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
$+_\bullet$	$Pr(+_\bullet)$	$1 - Pr(+_\bullet)$	
$-_\bullet$	$Pr(-_\bullet)$	$1 - Pr(-_\bullet)$	
$\times_\bullet$	$Pr(\times_\bullet)$	$1 - Pr(\times_\bullet)$	$\{a \in \mathbb{Z} \mid \texttt{MININT} \leq a \leq \texttt{MAXINT}\}$
$\div_\bullet$	$Pr(\div_\bullet)$	$1 - Pr(\div_\bullet)$	
$=_\bullet$	$Pr(Wr)$	$1 - Pr(Wr)$	
$>_\bullet$	$Pr(>_\bullet)$	$1 - Pr(>_\bullet)$	$\{true,\ false\}$
$\geq_\bullet$	$Pr(\geq_\bullet)$	$1 - Pr(\geq_\bullet)$	

$\vdots$

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.

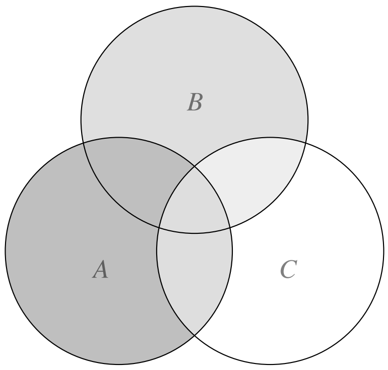
These three operations are independent of each other. Corresponding to three operations we have three events namely:

- A: **Read** executes successfully.

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

$\therefore \quad Pr(A) = Pr(Rd), \quad Pr(B) = Pr(+_\bullet) \quad$ 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).

4. The Probabilistic Concrete Domain, $\mathcal{L} : (L, \sqsubseteq_L)$

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 sets values and its associated probability. For program with a single integer variable, the concrete domain is $\quad L \subseteq 2^{\mathbb{Z}} \times \mathbb{R} \quad$ and

$$L = \left\{ (S, p) \,\middle|\, S \in 2^{\mathbb{Z}} \setminus \phi \bigwedge p \in \mathbb{R} \bigwedge \forall s \in S, \texttt{MININT} \leq s \leq \texttt{MAXINT} \bigwedge 0 \leq p \leq 1 \right\} \bigcup \{(\phi, 1)\}$$

Throughout our discussion we assume that our program variables take values between MININT and MAXINT (including both). $\langle S_i, p_i \rangle$ denotes that at a particular program point, say v_i, variable x_i (say) takes its values from set S_i with probability p_i. In general, for program with n integer variables, this concrete domain will be L^n

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

$$\forall \langle S_1, p_1 \rangle, \langle S_2, p_2 \rangle \in L \qquad \langle S_1, p_1 \rangle \sqsubseteq_L \langle S_2, p_2 \rangle \qquad \Longleftrightarrow \qquad S_1 \subseteq S_2 \quad \bigwedge \quad p_1 \geq p_2$$

As an example,

$$\langle \{a_1, a_2, \cdots a_n\}, p_1 \rangle \sqsubseteq_L \langle \{a_1, a_2, \cdots a_m\}, p_2 \rangle \qquad \text{iff} \qquad n \leq m, \text{so that } \{a_1, a_2 \cdots a_n\} \subseteq \{a_1, a_2 \cdots a_m\} \qquad \text{and} \qquad p_1 \geq p_2$$

$(L, \sqsubseteq_L)$ 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

$$\forall x, y \in L \left(\exists L_{lub} \in \left\{ u \in L \;\middle|\; (x \sqsubseteq_L u) \wedge (y \sqsubseteq_L u) \right\} s.t \left(\forall z \;\left(z \in \left\{ u \in L \;\middle|\; (x \sqsubseteq_L u) \wedge (y \sqsubseteq_L u) \right\} \Rightarrow L_{lub} \sqsubseteq_L z \right)\right) \right.$$

$$\bigwedge$$

$$\left. \exists L_{glb} \in \left\{ l \in L \;\middle|\; (l \sqsubseteq_L x) \wedge (l \sqsubseteq_L y) \right\} s.t \left(\forall z \;\left(z \in \left\{ l \in L \;\middle|\; (l \sqsubseteq_L x) \wedge (l \sqsubseteq_L y) \right\} \Rightarrow z \sqsubseteq_L L_{glb} \right)\right) \right)$$

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.

$$\langle S_1, p_1 \rangle \bigsqcup_L \langle S_2, p_2 \rangle = \left\langle S_1 \bigcup S_2, \mathbf{min}(p_1, p_2) \right\rangle \quad (1)$$

$$\langle S_1, p_1 \rangle \bigsqcap^L \langle S_2, p_2 \rangle = \left\langle S_1 \bigcap S_2, \mathbf{max}(p_1, p_2) \right\rangle \quad (2)$$

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

$\top_L = \langle \{a \in \mathbb{Z} \;:\; \text{MININT} \le a \le \text{MAXINT}\}, 0 \rangle$

$\bot_L = \langle \{\;\}, 1 \rangle$

The **Hasse Diagram** of lattice $(L, \sqsubseteq_L)$ can be found in Appendix C. $\forall S \subseteq L$, 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 $p = \langle (\{5\}, p_1), (\{10\}, p_2), (\{10\}, p_3) \rangle \equiv [x = 5,\ y = 10,\ z = 10]$ and $p' = \langle (\{15\}, p_1), (\{0\}, p_2), (\{1\}, p_3) \rangle \equiv [x = 15,\ y = 0,\ z = 1]$ are two program states (p_1, p_2, p_3 are place-holders here). Given a predicate $P : [x + y \ge z]$, 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, $x_1, x_2 \cdots x_n$. At any program point, a predicate with n variables will be of the form $P : \langle (S_1, p_1), (S_2, p_2), \cdots, (S_n, p_n) \rangle$, s.t at that program point each x_i will have values from the set S_i with probability p_i. In the following equations,

$$\mathcal{V}\Big((v_1, v_2, \cdots, v_n), x_i\Big) = \begin{cases} v_i & \text{if } x_i \text{ is a variable} \\ x_i = C & \text{if } x_i \text{ is a constant, say } C \end{cases} \qquad \text{and} \qquad \mathcal{P}(P, x_i) = \begin{cases} p_i & \text{if } x_i \text{ is a variable} \\ 1 & \text{if } x_i \text{ is a constant} \end{cases}$$

$\forall i$, $\colorbox{black}{$\color{white}\mathtt{A_i}$}$ is a probabilistic arithmatic operation and $\textcircled{$\mathtt{A_i}$}$ is a deterministic arithmatic operation.

i.e $\colorbox{black}{$\color{white}\mathtt{A_i}$} \in \{+_\bullet, -_\bullet, \times_\bullet, \div_\bullet, \%_\bullet, \cdots\}$ $\quad \textcircled{$\mathtt{A_i}$} \in \{+, -, \times, \div, \%, \cdots\}$

$\colorbox{black}{$\color{white}\mathtt{C}$}$ is a probabilistic comparison operation and $\textcircled{C}$ is a deterministic comparison operation.

i.e $\colorbox{black}{$\color{white}\mathtt{C}$} \in \{<_\bullet, >_\bullet, \geq_\bullet, \leq_\bullet, ==_\bullet, \neq_\bullet\}$ $\quad \textcircled{C} \in \{>, <, \geq, \leq, ==, \neq\}$

$$\begin{aligned}
& sp\Big(P,\ x_{i_f} =_\bullet x_{i_1} \colorbox{black}{$\color{white}\mathtt{A_2}$}\, x_{i_2} \colorbox{black}{$\color{white}\mathtt{A_3}$} \cdots \colorbox{black}{$\color{white}\mathtt{A_k}$}\, x_{i_k}\Big) \\
= \ & sp\Big(\langle (S_1, p_1), (S_2, p_2), \cdots, (S_{i_f}, p_{i_f}), \cdots, (S_n, p_n)\rangle,\ x_{i_f} =_\bullet x_{i_1} \colorbox{black}{$\color{white}\mathtt{A_2}$}\, x_{i_2} \colorbox{black}{$\color{white}\mathtt{A_3}$} \cdots \colorbox{black}{$\color{white}\mathtt{A_k}$}\, x_{i_k}\Big) \\
= \ & \langle (S_1, p_1), (S_2, p_2), \cdots, (S^*_{i_f}, p^*_{i_f}), \cdots, (S_n, p_n)\rangle = P^* \qquad (3) \\
& \text{where,} \quad S^*_{i_f} = \Big\{\mathcal{V}(t, x_{i_1}) \textcircled{$\mathtt{A_2}$} \mathcal{V}(t, x_{i_2}) \textcircled{$\mathtt{A_3}$} \cdots \textcircled{$\mathtt{A_k}$} \mathcal{V}(t, x_{i_k}) \ \Big| \ \forall t \in S_1 \times S_2 \times \cdots \times S_n\Big\} \quad \text{and} \\
& \qquad p^*_{i_f} = \left(Pr(Wr) + \frac{1 - Pr(Wr)}{\mathtt{MAXINT} - \mathtt{MININT} + 1}\right) \cdot \left(Pr(Rd) + \frac{1 - Pr(Rd)}{\mathtt{MAXINT} - \mathtt{MININT} + 1}\right)^{|vars|} \cdot \prod_{j=1}^{k} \mathcal{P}(P, x_{i_j}) \cdot \\
& \qquad\qquad \prod_{j=2}^{k} \left(Pr\left(\colorbox{black}{$\color{white}\mathtt{A_j}$}\right) + \frac{1 - Pr\left(\colorbox{black}{$\color{white}\mathtt{A_j}$}\right)}{\mathtt{MAXINT} - \mathtt{MININT} + 1}\right) \\
& \text{where,} \quad vars = \Big\{x_{i_j} \ : \ \forall j \in \{1, 2, \cdots, k\} \text{ and } x_{i_j} \text{ is a variable}\Big\}
\end{aligned}$$

$$\begin{aligned}
& sp\left(P,\ \left(x_{i_1} \colorbox{black}{$\color{white}\mathtt{A_{i_2}}$}\, x_{i_2} \colorbox{black}{$\color{white}\mathtt{A_{i_3}}$} \cdots \colorbox{black}{$\color{white}\mathtt{A_{i_k}}$}\, x_{i_k}\right) \colorbox{black}{$\color{white}\mathtt{C}$} \left(x_{j_1} \colorbox{black}{$\color{white}\mathtt{A_{j_2}}$}\, x_{j_2} \colorbox{black}{$\color{white}\mathtt{A_{j_3}}$} \cdots \colorbox{black}{$\color{white}\mathtt{A_{j_m}}$}\, x_{j_m}\right)\right) \\
= \ & sp\left(\langle (S_1, p_1), (S_2, p_2), \cdots, (S_n, p_n)\rangle,\ \left(x_{i_1} \colorbox{black}{$\color{white}\mathtt{A_{i_2}}$}\, x_{i_2} \colorbox{black}{$\color{white}\mathtt{A_{i_3}}$} \cdots \colorbox{black}{$\color{white}\mathtt{A_{i_k}}$}\, x_{i_k}\right) \colorbox{black}{$\color{white}\mathtt{C}$} \left(x_{j_1} \colorbox{black}{$\color{white}\mathtt{A_{j_2}}$}\, x_{j_2} \colorbox{black}{$\color{white}\mathtt{A_{j_3}}$} \cdots \colorbox{black}{$\color{white}\mathtt{A_{j_m}}$}\, x_{j_m}\right)\right) \\
= \ & \langle (S^*_1, p^*_1), (S^*_2, p^*_2), \cdots, (S^*_n, p^*_n)\rangle = P^* \qquad (4) \\
& \text{where,} \quad \forall i \in \{1, 2, \cdots, n\} \\
& \qquad S^*_i = \{t_i \ : \ \forall (t_1, t_2, \cdots, t_n) \in Q\} \quad \text{and} \\
& \qquad p^*_i = p_i \cdot \left(Pr(Rd) + \frac{1 - Pr(Rd)}{\mathtt{MAXINT} - \mathtt{MININT} + 1}\right)^{|vars|} \cdot \left(Pr\left(\colorbox{black}{$\color{white}\mathtt{C}$}\right) + \frac{1 - Pr\left(\colorbox{black}{$\color{white}\mathtt{C}$}\right)}{2}\right) \cdot \\
& \qquad\qquad \prod_{j=2}^{k} \left(Pr\left(\colorbox{black}{$\color{white}\mathtt{A_{i_j}}$}\right) + \frac{1 - Pr\left(\colorbox{black}{$\color{white}\mathtt{A_{i_j}}$}\right)}{\mathtt{MAXINT} - \mathtt{MININT} + 1}\right) \cdot \prod_{i=2}^{m} \left(Pr\left(\colorbox{black}{$\color{white}\mathtt{A_{j_i}}$}\right) + \frac{1 - Pr\left(\colorbox{black}{$\color{white}\mathtt{A_{j_i}}$}\right)}{\mathtt{MAXINT} - \mathtt{MININT} + 1}\right) \\
& \text{where,} \quad vars = \Big\{x \in \{x_{i_1}, x_{i_2} \cdots x_{i_k}\} \cup \{x_{j_1}, x_{j_2} \cdots x_{j_m}\} \ : \ x \text{ is a variable}\Big\} \quad \text{and} \\
& Q = \left\{t \in S_1 \times S_2 \times \cdots \times S_n \ \middle| \ \left(\mathcal{V}(t, x_{i_1}) \textcircled{$\mathtt{A_{i_2}}$} \mathcal{V}(t, x_{i_2}) \textcircled{$\mathtt{A_{i_3}}$} \cdots \textcircled{$\mathtt{A_{i_k}}$} \mathcal{V}(t, x_{i_k})\right) \textcircled{C} \left(\mathcal{V}(t, x_{j_1}) \textcircled{$\mathtt{A_{j_2}}$} \mathcal{V}(t, x_{j_2}) \textcircled{$\mathtt{A_{j_3}}$} \cdots \textcircled{$\mathtt{A_{j_m}}$} \mathcal{V}(t, x_{j_m})\right)\right\}
\end{aligned}$$

For our example program the definition of sp will be as follows,

$$sp\Big(\langle S,p\rangle,\ x =_\bullet 0\Big) = \left\langle \{0\},\ Pr(Wr) + \frac{1 - Pr(Wr)}{\texttt{MAXINT} - \texttt{MININT} + 1} \right\rangle \tag{5}$$

$$\begin{aligned} sp\Big(\langle S,p\rangle,\ x =_\bullet x +_\bullet 3\Big) = \Bigg\langle & \{v+3\ :\ \forall v \in S\},\ p \cdot \left(Pr(Rd) + \frac{1 - Pr(Rd)}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) \cdot \\ & \left(Pr(+_\bullet) + \frac{1 - Pr(+_\bullet)}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) \cdot \left(Pr(Wr) + \frac{1 - Pr(Wr)}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) \Bigg\rangle \end{aligned} \tag{6}$$

$$sp\Big(\langle S,p\rangle,\ x \leq_\bullet 9\Big) = \left\langle \{v \in S\ :\ v \leq 9\},\ p \cdot \left(Pr(\leq_\bullet) + \frac{1 - Pr(\leq_\bullet)}{2}\right)\left(Pr(Rd) + \frac{1 - Pr(Rd)}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) \right\rangle \tag{7}$$

$$sp\Big(\langle S,p\rangle,\ x \geq_\bullet 10\Big) = \left\langle \{v \in S\ :\ v \geq 10\},\ p \cdot \left(Pr(\geq_\bullet) + \frac{1 - Pr(\geq_\bullet)}{2}\right)\left(Pr(Rd) + \frac{1 - Pr(Rd)}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) \right\rangle \tag{8}$$

Further, for any program σ, $sp(P = \bot_L = \{\ \},\ \sigma) = \bot_L = \{\ \}$. 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.

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

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, $\mathcal{F}^n(\bot_L,\ \bot_L,\ \bot_L,\ \bot_L)$ starting with the least elements, i.e. the empty sets.

For the sake of the example we assume all the success probabilities to be $1 - 10^{-4}$, i.e $Pr(+_\bullet) = Pr(-_\bullet) = Pr(Rd) = Pr(Wr) = Pr(\leq_\bullet) = \quad \cdots \quad = 1 - 10^{-4}$ and values of MININT and MAXINT to be -32768 and 32767 respectively.

$$\therefore \quad \left(Pr(+_\bullet) + \frac{1 - Pr(+_\bullet)}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) = 0.99990000152 = x \quad \text{and} \quad \left(Pr(\leq_\bullet) + \frac{1 - Pr(\leq_\bullet)}{2}\right) = 0.99995 = y \text{ (say)}$$

Solving iteratively would give us a solution as follows

$$\Big(\bot_L,\ \bot_L,\ \bot_L,\ \bot_L\Big) \to \Big(\Big\langle\Big\{a \in \mathbb{Z}\Big|\texttt{MININT} \leq a \leq \texttt{MAXINT}\Big\},\ 1\Big\rangle,\ \bullet,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \Big\langle\Big\{0\Big\},\ x\Big\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \Big\langle\Big\{0\Big\},\ x^2y\Big\rangle,\ \bullet\Big) \to$$
$$\Big(\bullet,\ \Big\langle\Big\{0,3\Big\},\ x^5y\Big\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \Big\langle\Big\{0,3\Big\},\ x^6y^2\Big\rangle,\ \bullet\Big) \to \Big(\bullet,\ \Big\langle\Big\{0,3,6\Big\},\ x^9y^2\Big\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \Big\langle\Big\{0,3,6\Big\},\ x^{10}y^3\Big\rangle,\ \bullet\Big) \to$$

$\Big(\bullet,\ \big\langle\{0,3,6,9\},\ x^{13}y^{3}\big\rangle,\ \bullet,\ \bullet\Big)\ \rightarrow\ \Big(\bullet,\ \bullet,\ \big\langle\{0,3,6,9\},\ x^{14}y^{4}\big\rangle,\ \bullet\Big)\ \rightarrow\ \Big(\bullet,\ \big\langle\{0,3,6,9,12\},\ x^{17}y^{4}\big\rangle,\ \bullet,\ \bullet\Big)\ \rightarrow$
$\Big(\bullet,\ \bullet,\ \bullet,\ \big\langle\{12\},\ x^{18}y^{5}\big\rangle\Big)$

The '•' indicates, repetition of value from previous iteration. The overall solution is therefore
$\Big(\big\langle\{a\in\mathbb{Z}\,|\,\texttt{MININT}\le a\le\texttt{MAXINT}\},\ 1\big\rangle,\ \big\langle\{0,3,6,9,12\},\ x^{17}y^{4}\big\rangle,\ \big\langle\{0,3,6,9\},\ x^{14}y^{4}\big\rangle,\ \big\langle\{12\},\ x^{18}y^{5}\big\rangle\Big) =$
$\Big(\big\langle\{a\in\mathbb{Z}\,|\,-32768\le a\le 32767\},\ 1\big\rangle,\ \big\langle\{0,3,6,9,12\},\ 0.99810173981\big\rangle,\ \big\langle\{0,3,6,9\},\ 0.99840122568\big\rangle,\ \big\langle\{12\},\ 0.99795203106\big\rangle\Big)$

In general, we may need infinite number of iterations to converge. This is because the height of the lattice L is ∞. But for practical purpose we stop iterating as soon as the values converge.

5. The Probabilistic Interval Abstract Domain, $\mathcal{M}$: $(M, \sqsubseteq_M)$

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) :

$$p.m.f\big(\langle[a,b], p_{ab}\rangle\big) = \frac{p_{ab}}{b-a+1} \tag{9}$$

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[6]. Two intervals are related to each other by a binary relation $\sqsubseteq_{int}$ iff one interval is contained in the other.

$$[a,\ b] \sqsubseteq_{int} [c,\ d] \iff (c \le a) \wedge (b \le d) \tag{10}$$

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 a partial order $\sqsubseteq_M$ on M as follows :

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

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 $M \subseteq \mathbb{Z}^2 \times \mathbb{R}$, such that,

$M = \{\langle[a,b],p_{ab}\rangle \mid a,b \in \mathbb{Z}, p_{ab} \in \mathbb{R}, \texttt{MININT} \leq a \leq b \leq \texttt{MAXINT}, 0 \leq p_{ab} \leq 1\} \bigcup \{\ \perp_M\ \}$

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 $l.u.b$, $\underset{M}{\bigsqcup}$ and the greatest lower bound or $g.l.b$, $\overset{M}{\bigsqcap}$ for any two elements of M as follows.

$$\left.\begin{array}{rcll} i \underset{M}{\bigsqcup} \perp_M & = & i & \forall i \in M \\ \langle[a,b],p_{ab}\rangle \underset{M}{\bigsqcup} \langle[c,d],p_{cd}\rangle & = & \langle[x,y],p_{xy}\rangle & \end{array}\right\} \tag{11}$$

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

$$\left.\begin{array}{rcll} i \overset{M}{\bigsqcap} \perp_M & = & \perp_M & \forall i \in M \\ \langle[a,b],p_{ab}\rangle \overset{M}{\bigsqcap} \langle[c,d],p_{cd}\rangle & = & \begin{cases} \langle[x,y],p_{xy}\rangle & \text{if } x \leq y \\ \perp_M & \text{otherwise} \end{cases} & \end{array}\right\} \tag{12}$$

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

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

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

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

6. 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. $\forall \langle S, p\rangle \in L$, such that $S \neq \phi$

$$\mathcal{V}_m\big(\langle S, p\rangle\big) = v_m \qquad \text{where,} \quad v_m \in S \bigwedge \forall v \in S \;\Rightarrow\; v_m \leq v \tag{13a}$$

$$\mathcal{V}_M\big(\langle S, p\rangle\big) = v_M \qquad \text{where,} \quad v_M \in S \bigwedge \forall v \in S \;\Rightarrow\; v_M \geq v \tag{13b}$$

$\therefore$ $\mathcal{V}_m$ and $\mathcal{V}_M$ returns respectively the **minimum** and **maximum** of all **values** of a member of L whose value set is not empty. For example, if $l = \Big\langle\{a_1, a_2, a_3\}, p\Big\rangle \in L$ then $\mathcal{V}_m\big(l\big) = \mathbf{min}(a_1, a_2, a_3)$ and $\mathcal{V}_M\big(l\big) = \mathbf{max}(a_1, a_2, a_3)$

We now define (α, γ) as follows :

$$\alpha(\langle S, p\rangle) = \begin{cases} \langle [\ ],\ 1\rangle = \perp_M & \langle S, p\rangle = \langle\{\ \},\ 1\rangle = \perp_L \\ \Big\langle\Big[\mathcal{V}_m\big(\langle S, p\rangle\big),\ \mathcal{V}_M\big(\langle S, p\rangle\big)\Big],\ \mathbf{min}\Big(1,\ p\cdot\Big(\mathcal{V}_M\big(\langle S, p\rangle\big) - \mathcal{V}_m\big(\langle S, p\rangle\big) + 1\Big)\Big)\Big\rangle & \text{otherwise} \end{cases} \tag{14}$$

$$\left.\begin{aligned} \gamma\Big(\langle[\ ],\ 1\rangle\Big) &= \Big\langle\{\ \},\ 1\Big\rangle = \perp_L \\ \gamma\Big(\langle[a,\ b],\ p_{ab}\rangle\Big) &= \Big\langle\Big\{i \in \mathbb{Z}\ :\ a \leq i \leq b\Big\},\ p.m.f\Big(\langle[a, b], p_{ab}\rangle\Big)\Big\rangle \end{aligned}\right\} \tag{15}$$

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

Now we show that α and γ follow the conditions for Galois Connection.

$\forall\langle S, p\rangle \in L \quad \langle S, p\rangle \sqsubseteq_L \gamma\big(\alpha(\langle S, p\rangle)\big)$. And this can be proved easily by expanding the definitions of α.

If $\langle S, p\rangle = \langle\phi, 1\rangle = \perp_L$, we trivially have $\langle\{\}, 1\rangle = \gamma\big(\alpha(\langle\{\}, 1\rangle)\big)$ i.e $\langle\{\}, 1\rangle \sqsubseteq_L \gamma\big(\alpha(\langle\{\}, 1\rangle)\big)$

Otherwise, $\gamma\big(\alpha(\langle S, p\rangle)\big) = \left\langle\left\{i \in \mathbb{Z}\ \middle|\ \mathcal{V}_m\big(\langle S, p\rangle\big) \leq i \leq \mathcal{V}_M\big(\langle S, p\rangle\big)\right\},\ \mathbf{min}\left(p,\ \dfrac{1}{\mathcal{V}_M\big(\langle S, p\rangle\big) - \mathcal{V}_m\big(\langle S, p\rangle\big) + 1}\right)\right\rangle$

and $\langle S, p\rangle \sqsubseteq_L \left\langle\left\{i \in \mathbb{Z}\ \middle|\ \mathcal{V}_m\big(\langle S, p\rangle\big) \leq i \leq \mathcal{V}_M\big(\langle S, p\rangle\big)\right\},\ \mathbf{min}\left(p,\ \dfrac{1}{\mathcal{V}_M\big(\langle S, p\rangle\big) - \mathcal{V}_m\big(\langle S, p\rangle\big) + 1}\right)\right\rangle$[1]

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

If $m = \langle[], 1\rangle = \perp_M$, we have $\alpha\big(\gamma(\langle[], 1\rangle)\big) = \langle[], 1\rangle$ i.e $\alpha\big(\gamma(\langle[], 1\rangle)\big) \sqsubseteq_M \langle[], 1\rangle$

Otherwise, $\alpha\Big(\gamma\big(\langle[a, b], p_{ab}\rangle\big)\Big) = \alpha\left(\left\langle\{i \in \mathbb{Z}\ :\ a \leq i \leq b,\},\ \dfrac{p_{ab}}{b - a + 1}\right\rangle\right) = \langle[a, b], p_{ab}\rangle$[2]

$\therefore\quad \alpha\big(\gamma(m)\big) = m \sqsubseteq_M m \qquad \forall m \in M$

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

[2]$\because p_{ab} \leq 1$

7. 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 $sp(\sigma) : L \rightarrow L$. 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 $sp^{\#}(\sigma) : M \rightarrow M$ 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 14 and 15 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. For notational convenience we write

$$Rel(Rd) = Pr(Rd) + \frac{1 - Pr(Rd)}{\texttt{MAXINT} - \texttt{MININT} + 1} \qquad Rel(Wr) = Pr(Wr) + \frac{1 - Pr(Wr)}{\texttt{MAXINT} - \texttt{MININT} + 1}$$

$$Rel(+_\bullet) = Pr(+_\bullet) + \frac{1 - Pr(+_\bullet)}{\texttt{MAXINT} - \texttt{MININT} + 1} \qquad Rel(\leq_\bullet) = Pr(\leq_\bullet) + \frac{1 - Pr(\leq_\bullet)}{2}$$

$$Rel(\geq_\bullet) = Pr(\geq_\bullet) + \frac{1 - Pr(\geq_\bullet)}{2}$$

$$sp^{\#}\Big(\langle[a,b],p\rangle,\ x =_{\bullet} 0\Big) = \boldsymbol{\alpha}\Big(sp(\boldsymbol{\gamma}(\langle[a,b],p\rangle),\ x =_{\bullet} 0)\Big)$$

$$= \boldsymbol{\alpha}\Big(sp\Big(\Big\langle\{i \in \mathbb{Z}\ :\ a \leq i \leq b\},\ p.m.f\big(\langle[a,b],p\rangle\big)\Big\rangle,\ x =_{\bullet} 0\Big)\Big) \qquad \text{[from 15]}$$

$$= \boldsymbol{\alpha}\left(\left\langle\left\{0\right\},\ Rel(Wr)\right\rangle\right) \qquad \text{[from 5]}$$

$$= \left\langle\left[0,0\right],\ Rel(Wr)\right\rangle \qquad \text{[from 14]} \tag{16}$$

$$sp^{\#}\Big(\langle[a,b],p\rangle,\ x =_{\bullet} x +_{\bullet} 3\Big) = \left\langle\left[a+3, b+3\right],\ p \cdot Rel(Rd) \cdot Rel(+_{\bullet}) \cdot Rel(Wr)\right\rangle \tag{17}$$

$$sp^{\#}\Big(\langle[a,b],p\rangle,\ x \leq_{\bullet} 9\Big) = \begin{cases} \left\langle\left[a,b\right],\ p \cdot Rel(Rd) \cdot Rel(\leq_{\bullet})\right\rangle & \text{if } b < 9 \\ \left\langle\left[a,9\right],\ \dfrac{p \cdot (9-a+1)}{b-a+1} \cdot Rel(Rd) \cdot Rel(\leq_{\bullet})\right\rangle & \text{if } a \leq 9 \leq b \\ \bot_M & \text{if } 9 < a \end{cases} \tag{18}$$

$$sp^{\#}\Big(\langle[a,b],p\rangle,\ x \geq_{\bullet} 10\Big) = \begin{cases} \left\langle\left[a,b\right],\ p \cdot Rel(Rd) \cdot Rel(\geq_{\bullet})\right\rangle & \text{if } 10 < a \\ \left\langle\left[10,b\right],\ \dfrac{p \cdot (b-10+1)}{b-a+1} \cdot Rel(Rd) \cdot Rel(\geq_{\bullet})\right\rangle & \text{if } a \leq 10 \leq b \\ \bot_M & \text{if } b < 10 \end{cases} \tag{19}$$

$$sp^{\#}\Big(\bot_M = \langle[\ ],1\rangle,\ \sigma\Big) = \boldsymbol{\alpha}\Big(sp(\boldsymbol{\gamma}(\bot_M),\ \sigma)\Big)$$

$$= \boldsymbol{\alpha}\Big(sp\Big(\bot_L,\ \sigma\Big)\Big)$$

$$= \boldsymbol{\alpha}\left(\bot_L\right) \qquad [\because\ sp(\bot_L,\ \sigma) = \bot_L, \quad \text{for any program statement } \sigma]$$

$$= \bot_M \tag{20}$$

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 $1 - 10^{-4}$, i.e $Pr(+_{\bullet}) = Pr(-_{\bullet}) = Pr(Rd) = Pr(Wr) = Pr(\leq_{\bullet}) = \cdots = 1 - 10^{-4}$ and value of MININT and MAXINT to be -32768 and 32767 respectively. Therefore, $\left(Pr(+_{\bullet}) + \dfrac{1 - Pr(+_{\bullet})}{\texttt{MAXINT} - \texttt{MININT} + 1}\right) = 0.9999000015 = x$ and $\left(Pr(\leq_{\bullet}) + \dfrac{1 - Pr(\leq_{\bullet})}{2}\right) = 0.99995 = y$ (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

$$\Big(\bot_M,\ \bot_M,\ \bot_M,\ \bot_M\Big) \to \Big(\langle\texttt{[MININT,MAXINT]},\ 1\rangle,\ \bullet,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \langle[0,0],\ x\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \langle[0,0],\ x^2y\rangle,\ \bullet\Big) \to \Big(\bullet,\ \langle[0,3],\ 1\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \langle[0,3],\ xy\rangle,\ \bullet\Big) \to \Big(\bullet,\ \langle[0,6],\ 1\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \langle[0,6],\ xy\rangle,\ \bullet\Big) \to \Big(\bullet,\ \langle[0,9],\ 1\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \langle[0,9],\ xy\rangle,\ \bullet\Big) \to \Big(\bullet,\ \langle[0,12],\ 1\rangle,\ \bullet,\ \bullet\Big) \to \Big(\bullet,\ \bullet,\ \bullet,\ \langle[10,12],\ \tfrac{3}{13}xy\rangle\Big)$$

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

$$\Big(\Big\langle\texttt{[MININT,}\leq\texttt{MAXINT]},\ 1\Big\rangle,\ \Big\langle[0,12],\ 1\Big\rangle,\ \Big\langle[0,9],\ xy\Big\rangle,\ \Big\langle[10,12],\ \tfrac{3}{13}xy\Big\rangle\Big) =$$
$$\Big(\Big\langle\texttt{[MININT,}\leq\texttt{MAXINT]},\ 1\Big\rangle,\ \Big\langle[0,12],\ 1\Big\rangle,\ \Big\langle[0,9],\ 0.9998500065\Big\rangle,\ \Big\langle[10,12],\ 0.23073461688\Big\rangle\Big)$$

In general, we may need infinite number of iterations to converge. This is because the height of the lattice M is ∞. But for practical purpose we stop iterating as soon as the intervals are fixed.

8. Fixpoint Approximation with Convergence Acceleration by Widening

Let's consider the following code segment.

```
x =• 0
while (x <• 1000)
    x =• x +• 1
```

Probabilistic interval analysis will need around 1000 iterations to converge to interval $[1000, 1000]$ for x at the end of the program. This seems too slow. To reduce the number of iterations to reach a fixed point, we rely on a technique called *widening*. The technique of widening basically refers to methods for accelerating the convergence of fixed point iterations.

Let's first look at the formal definition of the widening operator.
A widening operator $\nabla : M \times M \mapsto M$ is such that,

Correctness :

- $\forall x, y \in M,\quad \boldsymbol{\gamma}(x)\ \sqsubseteq_M \boldsymbol{\gamma}(x \nabla y)$
- $\forall x, y \in M,\quad \boldsymbol{\gamma}(y)\ \sqsubseteq_M \boldsymbol{\gamma}(x \nabla y)$

Convergence :

- For all increasing chains $x^0 \sqsubseteq_M x^1 \sqsubseteq_M \dots$, the increasing chain defined by $y^0 = x^0,\ \dots, y^{i+1} = y^i \nabla x^{i+1},\ \dots$ is not strictly increasing.

8.1. Fixpoint approximation with widening

For every *collecting semantics* in abstract domain, $g_j^\#$, we write $(g_j^\#)^i$ to represent its value after i^{th} iteration.

Following this notation, the upward iteration sequence with widening[4]:

$(g_j^\#)^0 = \bot_M$

$$(g_j^\#)^{i+1} = \begin{cases} (g_j^\#)^i & \text{if } sp^\#\big((g_j^\#)^i\big) \sqsubseteq_M (g_j^\#)^i \\ (g_j^\#)^i \nabla sp^\#\big((g_j^\#)^i\big) & \text{otherwise} \end{cases}$$

is ultimately stationary and its limit $\hat{A}$ is a sound upper approximation of $\text{lfp}^{\bot_M} sp^\#$ i.e $\text{lfp}^{\bot_M} sp^\# \sqsubseteq_M \hat{A}$.

8.2. Interval Widening with threshold set

There are several possible schemes for implementing widening operation. We shall outline one such method as follows.

A **threshold set** T is a finite set of numbers (including MININT and MAXINT),

For a given program, we identify the set of constants used. Let this be the set C. We are restricting ourselves to programs with integer variables only. Now our threshold set $T = C \bigcup \{\text{MININT, MAXINT}\}$.

Given two elements $\langle[a,b],p\rangle$ and $\langle[a',b'],p'\rangle$ in M, $\forall X \in M$, we define the widening with threshold operator ∇_T as follows:

$$X \nabla_T \bot_M = X \tag{21a}$$

$$\bot_M \nabla_T X = X \tag{21b}$$

$$\langle[a,b],p\rangle \nabla_T \langle[a',b'],p'\rangle = \begin{cases} \langle[a,b],p\rangle & \text{if } \langle[a',b'],p'\rangle \sqsubseteq_M \langle[a,b],p\rangle \\ \langle[x,y],p_{xy}\rangle & \text{otherwise} \end{cases} \tag{21c}$$

where,

$x = \mathbf{max}\{l \in T \mid l \leq a'\}$,

$y = \mathbf{min}\{h \in T \mid h \geq b'\}$ and

$p_{xy} = (y - x + 1) \cdot \mathbf{min}\left(\mathbf{max}\left(p.m.f\big(\langle[a,b],p\rangle\big),\ p.m.f\big(\langle[a',b'],p'\rangle\big)\right),\ \dfrac{1}{y-x+1}\right)$

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 $\sqsubseteq_L$ $\therefore \forall\langle S,p\rangle \in L$ $\langle S,p\rangle \sqsubseteq_L \langle S,p\rangle$.

Transitivity :

$$\langle S_1, p_1\rangle \sqsubseteq_L \langle S_2, p_2\rangle \quad \bigwedge \quad \langle S_2, p_2\rangle \sqsubseteq_L \langle S_3, p_3\rangle \tag{Given}$$

$$S_1 \subseteq S_2 \quad \bigwedge \quad S_2 \subseteq S_3 \tag{A.1}$$

$$p_1 \geq p_2 \quad \bigwedge \quad p_2 \geq p_3 \tag{A.2}$$

From A.1 we get $S_1 \subseteq S_3$ and from A.2 we get $p_1 \geq p_3$.

$\therefore \langle S_1, p_1\rangle \sqsubseteq_L \langle S_3, p_3\rangle$

Anti-Symmetricity :

$$\langle S_1, p_1\rangle \sqsubseteq_L \langle S_2, p_2\rangle \quad \bigwedge \quad \langle S_2, p_2\rangle \sqsubseteq_L \langle S_1, p_1\rangle \tag{Given}$$

$$S_1 \subseteq S_2 \quad \bigwedge \quad S_2 \subseteq S_1 \tag{A.3}$$

$$p_1 \geq p_2 \quad \bigwedge \quad p_2 \geq p_1 \tag{A.4}$$

From A.3 we get $S_1 = S_2$ and from A.4 we get $p_1 = p_2$.

$\therefore \langle S_1, p_1\rangle = \langle S_2, p_2\rangle$

Appendix B.

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

Let, $\langle S_1, p_1\rangle$ and $\langle S_2, p_2\rangle$ are two elements in L. From definition 1 we get the least upper bound $\langle S_u, p_u\rangle$ of $\langle S_1, p_1\rangle$ and $\langle S_2, p_2\rangle$ as

$$\langle S_u, p_u\rangle = \langle S_1, p_1\rangle \bigsqcup_L \langle S_2, p_2\rangle = \left\langle S_1 \bigcup S_2, \mathbf{min}(p_1, p_2)\right\rangle \tag{B.1}$$

Now, $\langle S_c, p_c\rangle$ be an upper bound of both $\langle S_1, p_1\rangle$ and $\langle S_2, p_2\rangle$.

$\therefore \langle S_1, p_1\rangle \sqsubseteq_L \langle S_c, p_c\rangle \quad$ *and* $\quad \langle S_2, p_2\rangle \sqsubseteq_L \langle S_c, p_c\rangle$

Using our definition of $\sqsubseteq_L$ we get :

$$S_1 \subseteq S_c \tag{B.2}$$

$$S_2 \subseteq S_c \tag{B.3}$$

$$p_1 \geq p_c \tag{B.4}$$

$$p_2 \geq p_c \tag{B.5}$$

We have to show that $\langle S_u, p_u \rangle \sqsubseteq_L \langle S_c, p_c \rangle$. From B.2 and B.3 it's obvious that

$$S_1 \cup S_2 \subseteq S_c \tag{B.6}$$

Combining B.4 and B.5 we get

$$\mathbf{min}(p_1, p_2) \geq p_c \tag{B.7}$$

From B.6 and B.7 we get

$$\begin{aligned} & \left\langle S_1 \bigcup S_2, \mathbf{min}(p_1, p_2) \right\rangle \sqsubseteq_L \langle S_c, p_c \rangle \\ \Rightarrow \quad & \langle S_u, p_u \rangle \sqsubseteq_L \langle S_c, p_c \rangle \end{aligned}$$

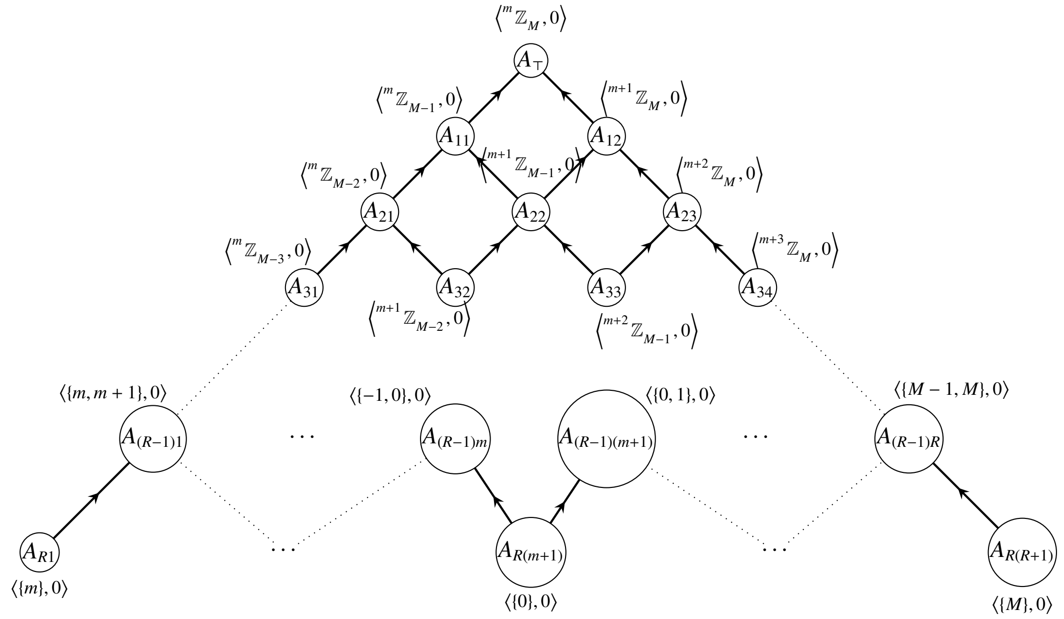
We omit the similar proof for greatest lower bound for the sake of brevity.

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 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. Only probability decreases monotonically (keeping the set of values same) along any dashed edge.



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

Frontal View

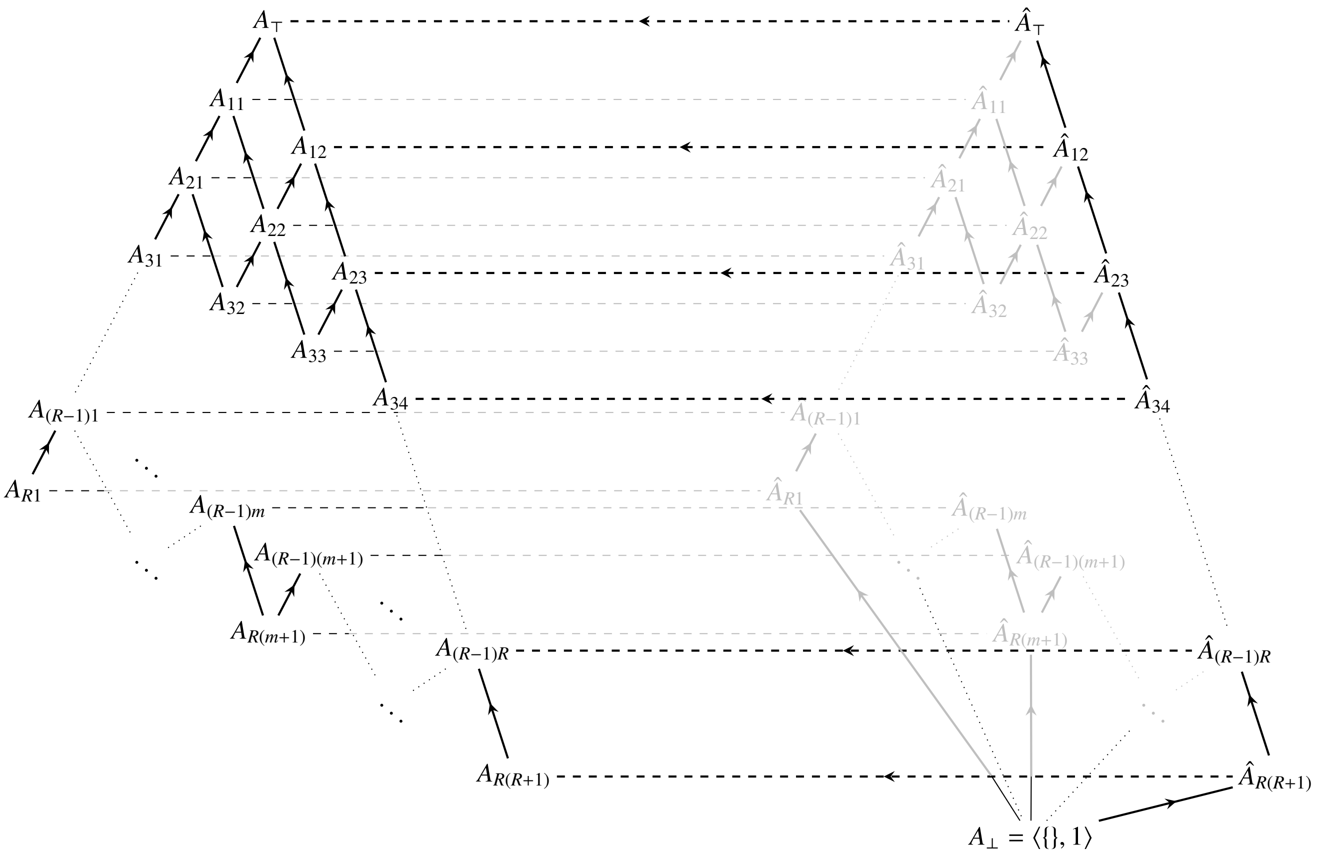
- ${}^{A}\mathbb{Z}_{B} = \{i \in \mathbb{Z} \mid A \leq i \leq B\}$
- $m = \texttt{MININT}$
- $M = \texttt{MAXINT}$

In the Frontal & Lateral View :

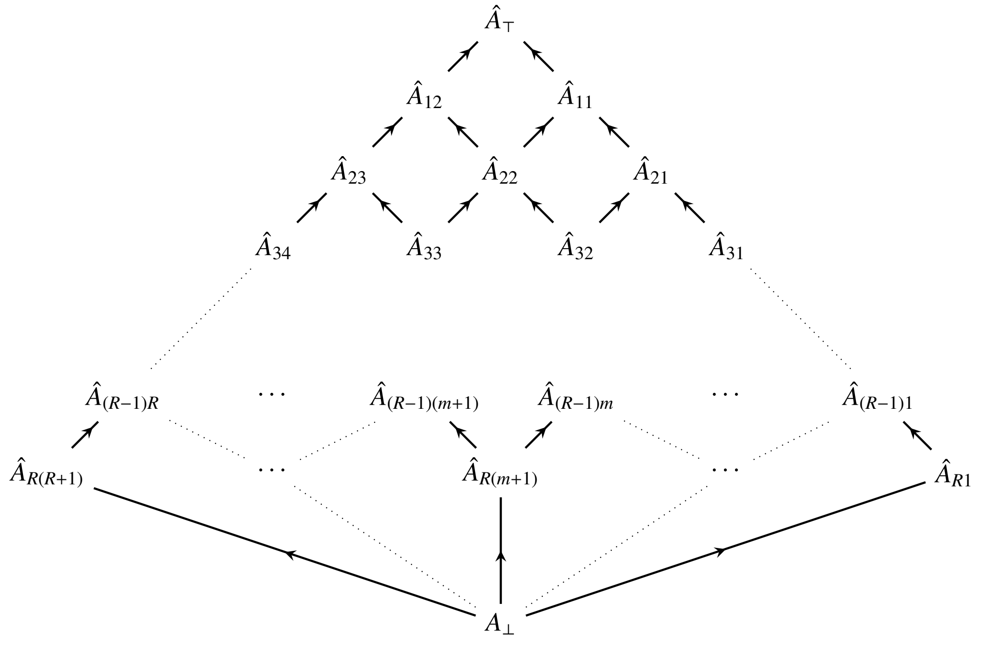
$A_{Ri} \equiv A_{(M-m)i}$ and $\hat{A}_{Ri} \equiv \hat{A}_{(M-m)i}$ $\forall i$

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: $A_{22} \equiv \left\langle {}^{m+1}\mathbb{Z}_{M-1},\ 0\right\rangle$ but $\hat{A}_{22} \equiv \left\langle {}^{m+1}\mathbb{Z}_{M-1},\ 1\right\rangle$



Lateral View (Skewed)



Rear View

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 :

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

$$[a,b] \sqsubseteq_{int} [c,d] \implies (c \leq a) \wedge (b \leq d) \qquad \Big[\text{using definition 10}\Big] \tag{D.3}$$

$$[c,d] \sqsubseteq_{int} [a,b] \implies (a \leq c) \wedge (d \leq b) \qquad \Big[\text{using definition 10}\Big] \tag{D.4}$$

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

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

Combining D.3 and D.4 we get,

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

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

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

From D.5 and D.6 we get,

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

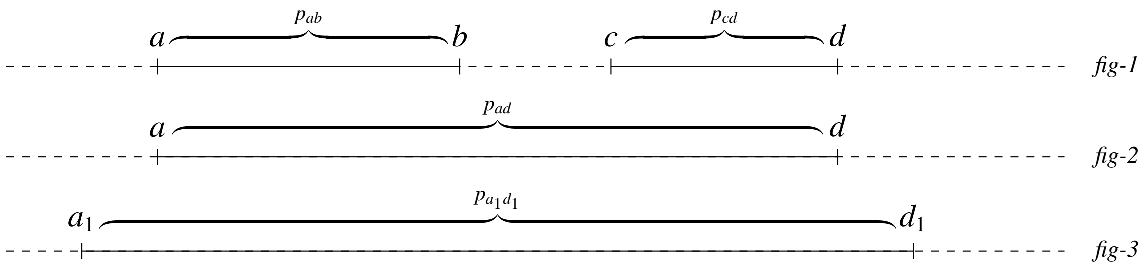
$\Rightarrow \dfrac{p_{ab}}{b-a+1} = \dfrac{p_{cd}}{d-c+1}$ $\qquad \Big[\text{using definition 9}\Big]$

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

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

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 11 we get,

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

$\therefore$ $\langle[a,d],p_{ad}\rangle$ (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 $p'_{ad} \le p_{ad}$[3], $\langle[a,d],p_{ad}\rangle \sqsubseteq_M \langle[a,d],p'_{ad}\rangle$ (by the definition of $\sqsubseteq_M$). So, $\langle[a,d],p_{ad}\rangle$ remains the *l.u.b*.

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

$$\langle[a,b],p_{ab}\rangle \sqsubseteq_M \langle[a_1,d_1],p_{a_1d_1}\rangle \qquad \wedge \qquad \langle[c,d],p_{cd}\rangle \sqsubseteq_M \langle[a_1,d_1],p_{a_1d_1}\rangle$$

$$\Rightarrow p.m.f\big(\langle[a_1,d_1],p_{a_1d_1}\rangle\big) \le p.m.f\big(\langle[a,b],p_{ab}\rangle\big) \ \wedge\ p.m.f\big(\langle[a_1,d_1],p_{a_1d_1}\rangle\big) \le p.m.f\big(\langle[c,d],p_{cd}\rangle\big) \ \big[\because \text{we know } [a,d] \sqsubseteq_{int} [a_1,d_1]\big]$$

$$\Rightarrow p_{a_1d_1} \le (d_1-a_1+1)\cdot p.m.f\big(\langle[a,b],p_{ab}\rangle\big) \ \wedge\ p_{a_1d_1} \le (d_1-a_1+1)\cdot p.m.f\big(\langle[c,d],p_{cd}\rangle\big) \qquad \big[\text{using definition } 9\big]$$

$$\Rightarrow p_{a_1d_1} \le \mathbf{min}\Big((d_1-a_1+1)\cdot p.m.f\big(\langle[a,b],p_{ab}\rangle\big),\ (d_1-a_1+1)\cdot p.m.f\big(\langle[c,d],p_{cd}\rangle\big),\ 1\Big) \qquad \big[\because p_{a_1d_1} \le 1\big]$$

$$\Rightarrow p_{a_1d_1} \le (d_1-a_1+1)\cdot\mathbf{min}\Big(p.m.f\big(\langle[a,\ b],\ p_{ab}\rangle\big),\ p.m.f\big(\langle[c,\ d],\ p_{cd}\rangle\big),\ \frac{1}{d_1-a_1+1}\Big)$$

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

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

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

[3]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 $\langle[c,d],p_{cd}\rangle$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 $(M, \sqsubseteq_M)$

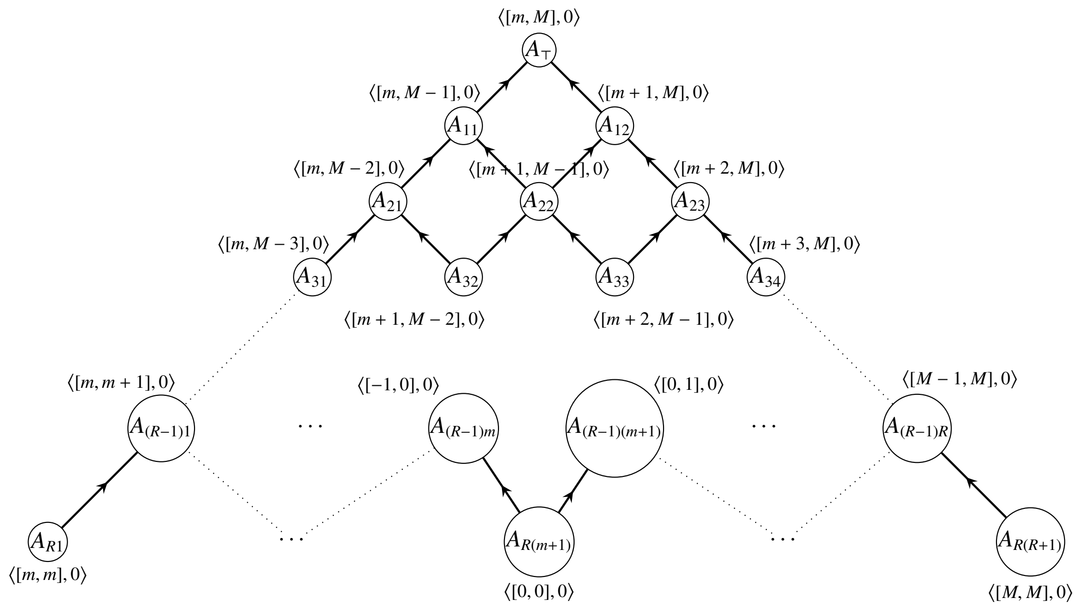
The lattice $(M, \sqsubseteq_M)$ is best viewed in 3 dimensions. For that reason we've given 4 different views of the same lattice.

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.

$$\begin{tikzcd}[column sep=small]
 & & & & \top_M = \langle [m,M], 0\rangle & & & & \\
\langle [m,M-1],0\rangle \arrow[urrrr] & & & & \langle [m,M],\epsilon\rangle \arrow[u, dashed] & & & & \langle [m+1,M],0\rangle \arrow[ullll] \\
 & & \langle [m,M-1], \frac{\epsilon\cdot(M-m)}{M-m+1}\rangle \arrow[ull, dashed] \arrow[urr] & & \langle [m,M],\epsilon_1\rangle \arrow[u, dashed] & & \langle [m+1,M], \frac{\epsilon\cdot(M-m)}{M-m+1}\rangle \arrow[ull] \arrow[urr, dashed] & & \\
 & \langle [m,M-1],\epsilon_2\rangle \arrow[ur, dashed] \arrow[urrr] & & & \langle [m,M],1\rangle \arrow[u, dashed] & & & \langle [m+1,M],\epsilon_2\rangle \arrow[ulll] \arrow[ul, dashed] & \\
 & \langle [m,M-1], \frac{M-m}{M-m+1}\rangle \arrow[u, dashed] \arrow[urrr] & & & & & & \langle [m+1,M], \frac{M-m}{M-m+1}\rangle \arrow[ulll] \arrow[u, dashed] & \\
\langle [m,M-2],0\rangle \arrow[uuuu] & \langle [m,M-1],1\rangle \arrow[u, dashed] & & & \langle [m+1,M-1],0\rangle \arrow[uuuullll] \arrow[uuuurrrr] & & & \langle [m+1,M],1\rangle \arrow[u, dashed] & \langle [m+2,M],0\rangle \arrow[uuuu] \\
\langle [m,m],0\rangle \arrow[u, dotted, no head] & \langle [m,M-2], \frac{M-m-1}{M-m}\rangle \arrow[ul, dashed] \arrow[u] & & & \langle [m+1,M-1], \frac{M-m-1}{M-m}\rangle \arrow[ulll] \arrow[u, dashed] \arrow[urrr] & & & \langle [m+2,M], \frac{M-m-1}{M-m}\rangle \arrow[u] \arrow[ur, dashed] & \langle [M,M],0\rangle \arrow[u, dotted, no head] \\
\langle [m,m],\epsilon\rangle \arrow[u, dashed] & \langle [m,M-2],1\rangle \arrow[u, dashed] & & & \langle [m+1,M-1],1\rangle \arrow[u, dashed] & & & \langle [m+2,M],1\rangle \arrow[u, dashed] & \langle [M,M],\epsilon\rangle \arrow[u, dashed] \\
 & \langle [m,m],1\rangle \arrow[ul, dashed] \arrow[u, dotted, no head] & \langle [m+1,m+1],1\rangle \arrow[urr, dotted, no head] & \dots & \langle [0,0],1\rangle & \dots & \langle [M-1,M-1],1\rangle \arrow[ull, dotted, no head] & \langle [M,M],1\rangle \arrow[u, dotted, no head] \arrow[ur, dashed] & \\
 & & & & \bot_M = \langle [\ ],1\rangle \arrow[ulll] \arrow[ull] \arrow[u] \arrow[urr] \arrow[urrr] & & & &
\end{tikzcd}$$

2D Projectional View

*legends are defined in the next page



Frontal View

In the 2D Projectional View :

- m = `MININT`
- M = `MAXINT`
- $0 < \epsilon, \epsilon_1, \epsilon_2 < 1$
- $\epsilon_2 = \dfrac{\epsilon_1 \cdot (M - m)}{M - m + 1}$ [4]

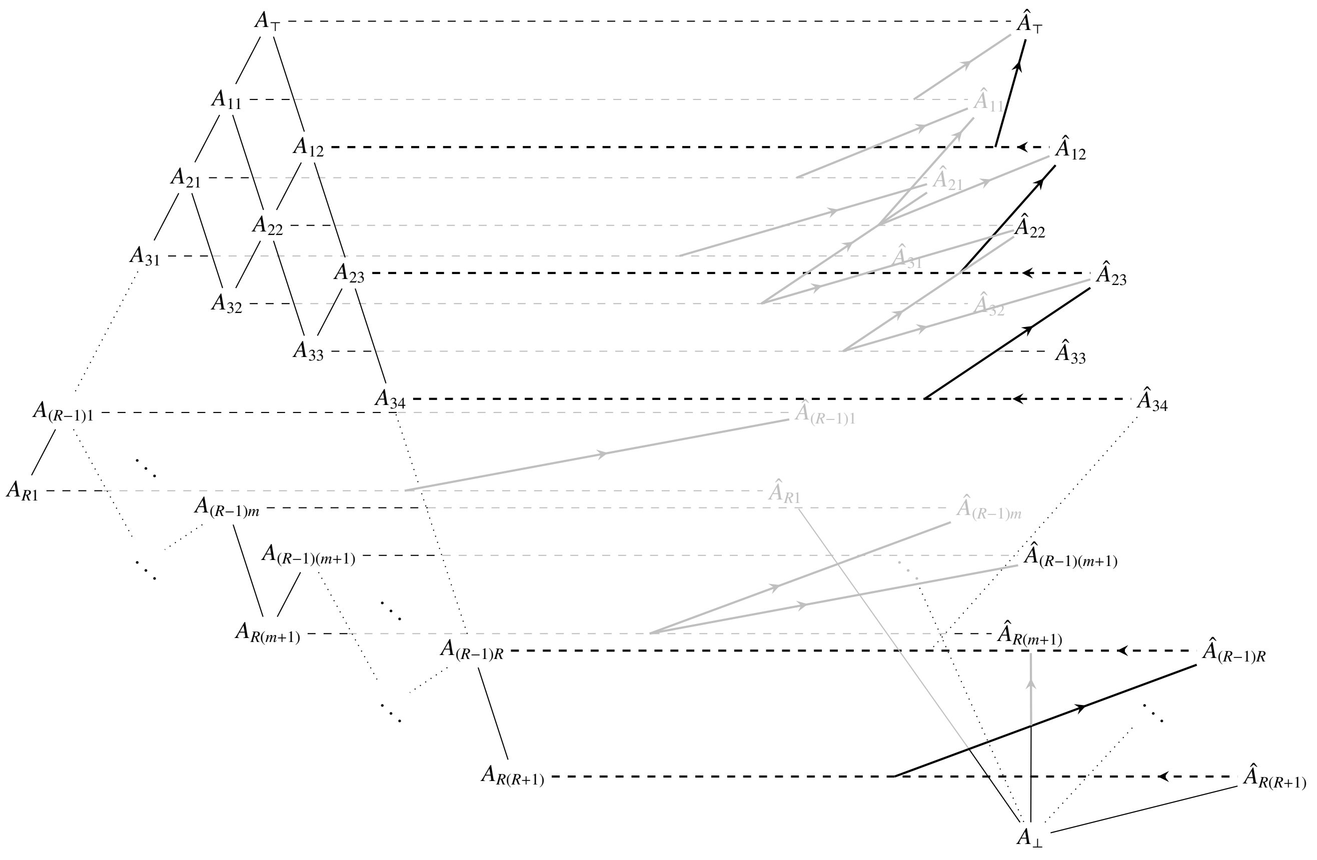
In the Frontal & Lateral View :

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

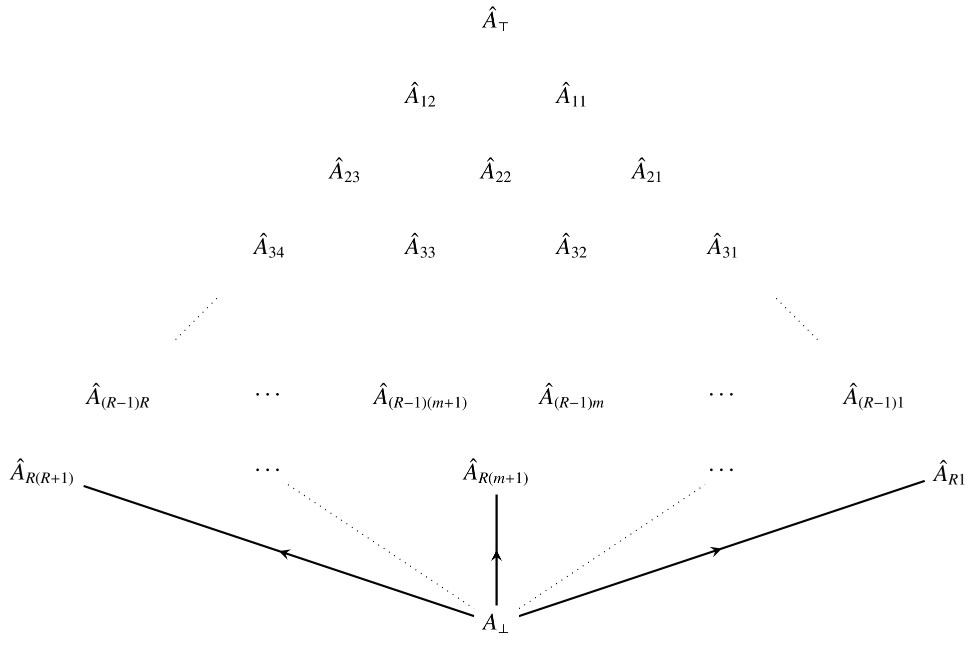
For every pair of indices i, j (and $\top$) the elements denoted by A_{ij} (and $A_\top$) and $\hat{A}_{ij}$ (and $\hat{A}_\top$) have the same interval, but probability in element A_{ij} (and $A_\top$) is 0 , wherese it is 1 for $\hat{A}_{ij}$ (and $\hat{A}_\top$).

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

[4]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)



$\hat{A}_\top$
$\hat{A}_{12}$ $\hat{A}_{11}$
$\hat{A}_{23}$ $\hat{A}_{22}$ $\hat{A}_{21}$
$\hat{A}_{34}$ $\hat{A}_{33}$ $\hat{A}_{32}$ $\hat{A}_{31}$
$\hat{A}_{(R-1)R}$ $\cdots$ $\hat{A}_{(R-1)(m+1)}$ $\hat{A}_{(R-1)m}$ $\cdots$ $\hat{A}_{(R-1)1}$
$\hat{A}_{R(R+1)}$ $\cdots$ $\hat{A}_{R(m+1)}$ $\cdots$ $\hat{A}_{R1}$
$A_\perp$

Rear View

Appendix G.

Proof of monotonicity of Galois connection (α, γ)

We know,

$$\begin{aligned}\boldsymbol{\gamma}\Big(\langle [\ ],\ 1\rangle\Big) &= \Big\langle \{\ \},\ 1\Big\rangle = \perp_L \\ \boldsymbol{\gamma}\Big(\langle [a,\ b],\ p_{ab}\rangle\Big) &= \Big\langle\Big\{i \in \mathbb{Z}\ :\ a \le i \le b\Big\},\ p.m.f\Big(\langle [a,b], p_{ab}\rangle\Big)\Big\rangle\end{aligned}$$

Let, $\langle [a,b], p_{ab}\rangle$ and $\langle [c,d], p_{cd}\rangle$ be two elements in M, with $\langle [a,b], p_{ab}\rangle \sqsubseteq_M \langle [c,d], p_{cd}\rangle$[5]. We show that

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

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

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

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

$$\begin{aligned} & [a,b] \sqsubseteq_{int} [c,d] \\ \Rightarrow\ & c \le a \bigwedge b \le d && \text{[using definition 10]} \\ \Rightarrow\ & \{i \in \mathbb{Z} \mid a \le i \le b\} \subseteq \{i \in \mathbb{Z} \mid c \le i \le d\} \end{aligned} \tag{G.2}$$

From G.1 and G.2 we get, $\boldsymbol{\gamma}\Big(\langle [a,b], p_{ab}\rangle\Big) \sqsubseteq_L \boldsymbol{\gamma}\Big(\langle [c,d], p_{cd}\rangle\Big)$ [from our definition of $\sqsubseteq_L$]

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

We know,

$$\boldsymbol{\alpha}\Big(\langle S, p\rangle\Big) = \begin{cases} \langle [\ ],\ 1\rangle = \perp_M & \langle S, p\rangle = \langle \phi, 1\rangle = \perp_L \\ \Big\langle\Big[\mathcal{V}_m\Big(\langle S,p\rangle\Big),\ \mathcal{V}_M\Big(\langle S,p\rangle\Big)\Big],\ \mathbf{min}\Big(1,\ p\cdot\Big(\mathcal{V}_M\Big(\langle S,p\rangle\Big) - \mathcal{V}_m\Big(\langle S,p\rangle\Big) + 1\Big)\Big)\Big\rangle & \text{otherwise}\end{cases}$$

Now, suppose $\langle S_1, p_1\rangle$ and $\langle S_2, p_2\rangle$ are two elements in L, with $\langle S_1, p_1\rangle \sqsubseteq_L \langle S_2, p_2\rangle$ and $S_1 \ne \phi$[6]. We show that

$$\boldsymbol{\alpha}\Big(\langle S_1, p_1\rangle\Big) \sqsubseteq_M \boldsymbol{\alpha}\Big(\langle S_2, p_2\rangle\Big)$$

$\because \langle S_1, p_1\rangle \sqsubseteq_L \langle S_2, p_2\rangle$, from the definition of $\sqsubseteq_L$ we have

$$S_1 \subseteq S_2 \tag{G.3}$$

$$p_1 \ge p_2 \tag{G.4}$$

$$\boldsymbol{\alpha}(\langle S_1, p_1\rangle) = \Big\langle\Big[\mathcal{V}_m\Big(\langle S_1,p_1\rangle\Big),\ \mathcal{V}_M\Big(\langle S_1,p_1\rangle\Big)\Big],\ \mathbf{min}\Big(1,\ p_1\cdot\Big(\mathcal{V}_M\Big(\langle S_1,p_1\rangle\Big) - \mathcal{V}_m\Big(\langle S_1,p_1\rangle\Big) + 1\Big)\Big)\Big\rangle \quad \text{and}$$

$$\boldsymbol{\alpha}(\langle S_2, p_2\rangle) = \Big\langle\Big[\mathcal{V}_m\Big(\langle S_2,p_2\rangle\Big),\ \mathcal{V}_M\Big(\langle S_2,p_2\rangle\Big)\Big],\ \mathbf{min}\Big(1,\ p_2\cdot\Big(\mathcal{V}_M\Big(\langle S_2,p_2\rangle\Big) - \mathcal{V}_m\Big(\langle S_2,p_2\rangle\Big) + 1\Big)\Big)\Big\rangle$$

[5] If $\perp_M \sqsubseteq_M \langle [a,b], p_{ab}\rangle$, we trivially have $\boldsymbol{\gamma}\Big(\perp_M\Big) \sqsubseteq_L \boldsymbol{\gamma}\Big(\langle [a,b], p_{ab}\rangle\Big)$ $[\because \boldsymbol{\gamma}\Big(\perp_M\Big) = \perp_L$ is the least element in $\mathcal{L}]$

[6] If $S_1 = \phi$, we have $\langle S_1, p_1\rangle = \perp_L \sqsubseteq_L \langle S_2, p_2\rangle \Rightarrow \boldsymbol{\alpha}\Big(\perp_L\Big) \sqsubseteq_M \boldsymbol{\alpha}\Big(\langle S_2, p_2\rangle\Big)$, $[\because \boldsymbol{\alpha}\Big(\perp_L\Big) = \langle [], 1\rangle = \perp_M$, the least element of $\mathcal{M}]$.

From G.3 we get,

$$\mathcal{V}_m\big(\langle S_2, p_2\rangle\big) \le \mathcal{V}_m\big(\langle S_1, p_1\rangle\big) \bigwedge \mathcal{V}_M\big(\langle S_1, p_1\rangle\big) \le \mathcal{V}_M\big(\langle S_2, p_2\rangle\big)$$
$$\Rightarrow \Big[\mathcal{V}_m\big(\langle S_1, p_1\rangle\big), \mathcal{V}_M\big(\langle S_1, p_1\rangle\big)\Big] \sqsubseteq_{int} \Big[\mathcal{V}_m\big(\langle S_2, p_2\rangle\big), \mathcal{V}_M\big(\langle S_2, p_2\rangle\big)\Big] \qquad \text{[using definition 10]} \tag{G.5}$$

From G.5 we get,

$$\mathcal{V}_M\big(\langle S_1, p_1\rangle\big) - \mathcal{V}_m\big(\langle S_1, p_1\rangle\big) + 1 \le \mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1$$
$$\Rightarrow \frac{1}{\mathcal{V}_M\big(\langle S_1, p_1\rangle\big) - \mathcal{V}_m\big(\langle S_1, p_1\rangle\big) + 1} \ge \frac{1}{\mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1} \tag{G.6}$$

$$p.m.f\big(\alpha(\langle S_i, p_i\rangle)\big) = \mathbf{min}\left(p_i, \frac{1}{\mathcal{V}_M\big(\langle S_i, p_i\rangle\big) - \mathcal{V}_m\big(\langle S_i, p_i\rangle\big) + 1}\right) \qquad \forall i \in \{1, 2\}$$

Now, depending upon the value of $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big)$ and $p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big)$ we have 4 different cases to consider.

Case 1 : $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) = p_1$ and $p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big) = p_2$

From G.4 it's obvious that $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) \ge p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big)$

Case 2 : $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) = p_1$ and $p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big) = \frac{1}{\mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1}$

$$\because\ p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big) = \frac{1}{\mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1}$$
$$\Rightarrow\ p_2 \ge \frac{1}{\mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1}$$
$$\Rightarrow\ p_1 \ge \frac{1}{\mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1} \qquad \text{[From } G.4\text{]}$$
$$\therefore\ p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) \ge p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big)$$

Case 3 : $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) = \frac{1}{\mathcal{V}_M\big(\langle S_1, p_1\rangle\big) - \mathcal{V}_m\big(\langle S_1, p_1\rangle\big) + 1}$ and $p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big) = p_2$

$$\because\ p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big) = p_2$$
$$\Rightarrow\ \frac{1}{\mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1} \ge p_2$$
$$\Rightarrow\ \frac{1}{\mathcal{V}_M\big(\langle S_1, p_1\rangle\big) - \mathcal{V}_m\big(\langle S_1, p_1\rangle\big) + 1} \ge p_2 \qquad \text{[From } G.6\text{]}$$
$$\therefore\ p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) \ge p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big)$$

Case 4 : $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) = \frac{1}{\mathcal{V}_M\big(\langle S_1, p_1\rangle\big) - \mathcal{V}_m\big(\langle S_1, p_1\rangle\big) + 1}$ and $p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big) = \frac{1}{\mathcal{V}_M\big(\langle S_2, p_2\rangle\big) - \mathcal{V}_m\big(\langle S_2, p_2\rangle\big) + 1}$

From G.6 it's obvious that $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) \ge p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big)$

So, in all the 4 cases $p.m.f\big(\alpha(\langle S_1, p_1\rangle)\big) \ge p.m.f\big(\alpha(\langle S_2, p_2\rangle)\big)$. This along with G.5 proves that $\alpha\big(\langle S_1, p_1\rangle\big) \sqsubseteq_M \alpha\big(\langle S_2, p_2\rangle\big)$. 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. 5^{th} *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] Patrick Cousot. A tutorial on abstract interpretation. In *VMCAI'05 Industrial day on Automatic Tools for Program Verification*, pages 104–106, 2005. http://cs.nyu.edu/~pcousot/COUSOTtalks/VMCAI05_TOOLS.shtml.

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

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

[7] 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.