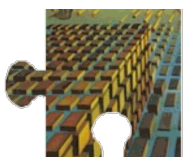
Algorithmes et méthodes pour la fiabilité numérique :

Partie 2 : une approche sémantique

Matthieu Martel

Laboratoire ELIAUS-DALI
Université de Perpignan Via Domitia

matthieu.martel@univ-perp.fr



DALI
Digits, Architectures et Logiciels Informatiques

ELIAUS

UPVD
Université de Perpignan Via Domitia

Introduction

Computer arithmetics :

not intuitive (usual algebraic laws do not hold)

difficult to predict the precision of a computation

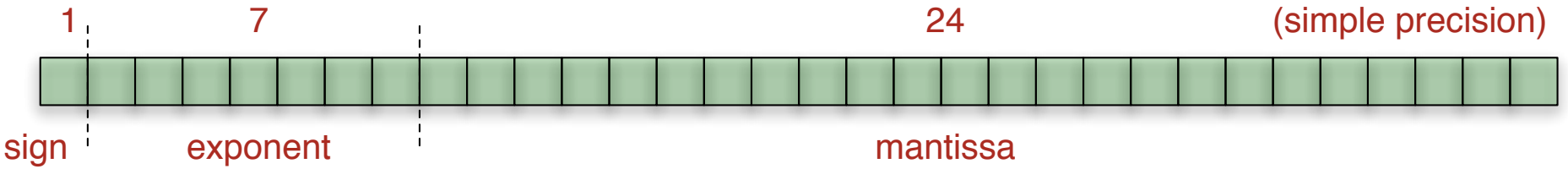
no method to improve the numerical quality of an implementation

a few empirical rules (Horner scheme, sort, etc.)

[David Goldberg : *What Every Computer Scientist Should Know About Floating Point Arithmetic*, 1991, ACM Computing Surveys]

[David Monniaux : *The Pitfalls of Verifying Floating-Point Computations*, 2007, ACM TOPLAS]

Floating-Point Arithmetic: the IEEE754 Standard

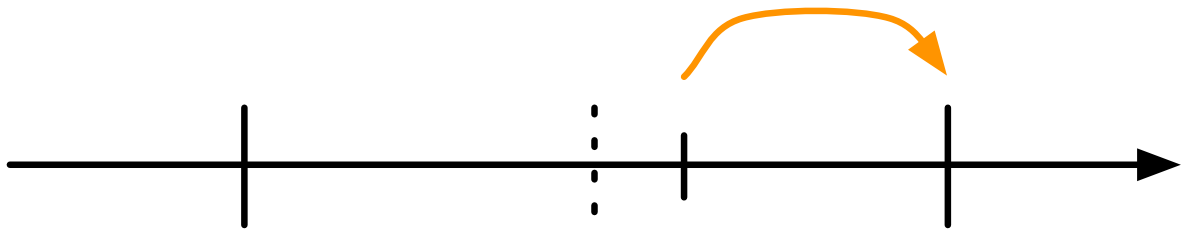


Floating-point numbers: $\pm d_0.d_1 d_2 \ldots d_{p-1} \beta^e$

Specifies basis ($\beta = 2$), mantissa size p and range for the exponent e

$$\beta = 2,\ p = 3,\ -1 \leq e \leq 1$$

IEEE754 Standard: roundoffs



4 rounding modes: towards $-\infty$, towards $+\infty$, to the nearest, towards 0

$\uparrow_\circ : \mathbb{R} \to \mathbb{F}$ computes the roundoff of a real number

for elementary operations $\Diamond \in \{+, -, \times, \div, \sqrt{}\}$:

$$x_1 \Diamond_{\mathbb{F},\circ} x_2 = \uparrow_\circ (x_1 \Diamond_{\mathbb{R}} x_2)$$

$\downarrow_\circ : \mathbb{R} \to \mathbb{R}$ computes the roundoff error: $\downarrow_\circ (r) = r - \uparrow_\circ (r)$

Fixed-Point Arithmetic

Hardware:

General processors with no floating-point unit

Reconfigurable circuits, FPGAs

Application domains:

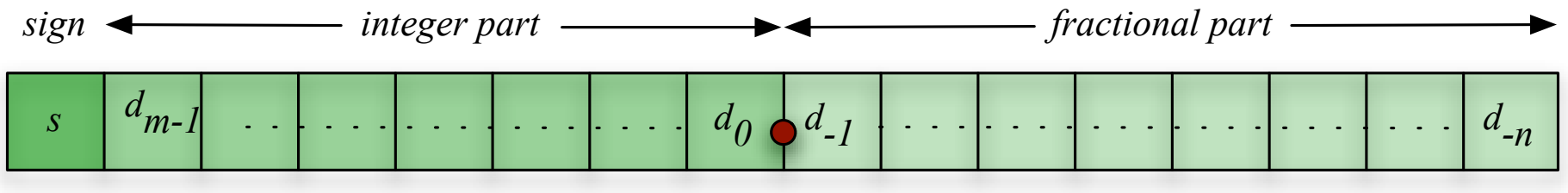
Cost-critical embedded systems: cellular phones, automotive industry, etc.

Domain-specific circuits, small series, prototypes

No standard comparable to IEEE754

[R. Rocher, D. Menard, N. Herve and O. Sentieys: *Fixed-Point Configurable Hardware Components*, Journal on Embedded Systems, 2006]

Fixed-Point Arithmetic (2)

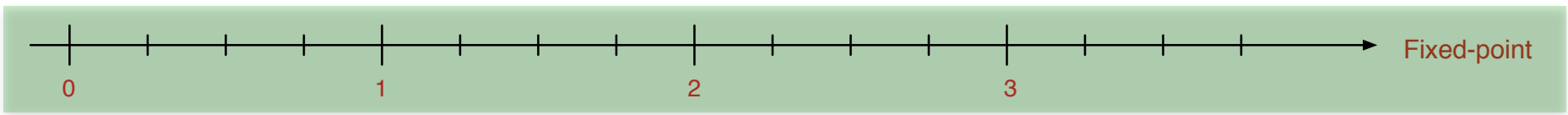


$$x = s \cdot (d_{m-1} \dots d_0 . d_{-1} \dots d_{-n}) = s \cdot \sum_{i=-n}^{m-1} d_i \beta^i$$

Basis $\beta = 2$, numbers made of $m + n + 1$ digits

Minimal number m of digits needed to encode the integer part:

$$\leftarrow_\circ (x) = \min \{ m \in \mathbb{N} \ : \ \lfloor x \rfloor \leq \beta^m \}$$



$$\beta = 2, \ m = 2, \ n = 2$$

Quality Indicators

In floating-point arithmetic:

Enhancing the quality of the implementation of a formula $f(\mathbf{x})$ consists of minimizing the roundoff error on the result

Goal: minimize $\downarrow_\circ (\mathtt{f}(\mathbf{x}))$, for all the possible vectors of inputs $\mathbf{x}$

In fixed-point arithmetic:

Enhancing the quality of the implementation of a formula $f(\mathbf{x})$ consists of finding the minimal size for the integer part of numbers such that no overflow arises during the computation

Goal: minimize $\leftarrow_\circ (y)$, for all intermediary result y arising during the computation of $\mathtt{f}(\mathbf{x})$ for all the possible vectors of inputs $\mathbf{x}$

Example: are 16 bits enough for the integer part to perform the computation without overflow?

Example

Mathematical formula:

$$E = (a + (b + (c + d))) \times e$$

$\texttt{a} \in [-14, -13]$ $\quad$ $\texttt{b} \in [-3, -2]$
$\texttt{c} \in [3, 3.5]$ $\quad$ $\texttt{d} \in [12.5, 13.5]$ $\quad$ $\texttt{e} = 2$

Implementations:

```
(a+(b+(c+d)))* e // straightforward implementation
(((a+b)+c)+d) * e // by associativity
(a*e+(b*e+(c*e+d*e))) // by distributivity
(a+b)*e+(c+d)*e // mix
...
```

In floating-point arithmetic:

What is the precision in the worst case?
Which operation of an implementation makes mostly the error grow?
What is the best implementation?

Example

Mathematical formula:

$$E = (a + (b + (c + d))) \times e$$

$\mathtt{a} \in [-14, -13]$ $\quad$ $\mathtt{b} \in [-3, -2]$
$\mathtt{c} \in [3, 3.5]$ $\quad$ $\mathtt{d} \in [12.5, 13.5]$ $\quad$ $\mathtt{e} = 2$

Implementations:

```
(a+(b+(c+d)))* e // straightforward implementation
(((a+b)+c)+d) * e // by associativity
(a*e+(b*e+(c*e+d*e))) // by distributivity
(a+b)*e+(c+d)*e // mix
...
```

In fixed-point arithmetic:

How many digits for the integer part, for a given implementation?
Which implementation requires the minimal number of digits for the integer part?

Measuring the Quality of an Implementation

Left-to-right evaluation of expressions:

$$\frac{v = v_1 + v_2}{v_1 + v_2 \;\rightarrow\; v} \qquad \frac{e_1 \;\rightarrow\; e_1'}{e_1 + e_2 \;\rightarrow\; e_1' + e_2} \qquad \frac{e_2 \;\rightarrow\; e_2'}{v_1 + e_2 \;\rightarrow\; v_1 + e_2'}$$

Semantics with quality indicators:

A value is a pair (x, μ) where:

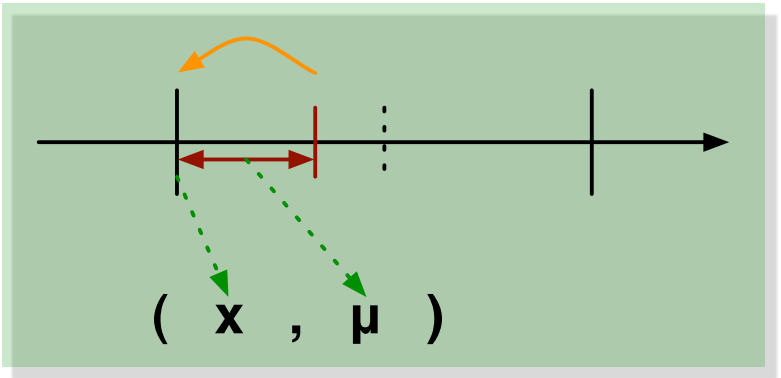
- x denotes the computer number, i.e. a fixed-point or a floating-point number
- μ measures the quality of x, (number of digits of the integer part or the roundoff error)

Abstraction by intervals:

$\{(x_i, \mu_i) \;:\; i \in I\}$ abstracted by $(x^\sharp, \mu^\sharp)$ where $x^\sharp$ and $\mu^\sharp$ are intervals

Measuring in the Floating-Point Format

Non standard value for $x_{\mathbb{R}}$:



Elementary operations:

$$\begin{array}{ll} & (x_1 \;,\; \mu_1) \\ + & (x_2 \;,\; \mu_2) \\ \hline = & ((x_1+x_2)\;,\;(\mu_1+\mu_2)) \\ \hline = & (\uparrow(x_1+x_2),\; (\mu_1+\mu_2 + \downarrow(x_1+x_2))) \end{array}$$

$$\begin{array}{ll} & (x_1 \;,\; \mu_1) \\ \times & (x_2 \;,\; \mu_2) \\ \hline = & ((x_1x_2)\;,\;(x_2\mu_1+x_1\mu_2+\mu_1\mu_2)) \\ \hline = & (\uparrow(x_1x_2)\;,\;(x_2\mu_1+x_1\mu_2+\mu_1\mu_2 + \downarrow(x_1x_2))) \end{array}$$

Example: For $E =$ `(a+(b+(c+d)))*e` we obtain

$$E_{\text{float}} = \left([-3,4],[-2.861022949\cdot 10^{-6},0]\right)$$

Semantics: values

- Values:

$$v = f\vec{\varepsilon}_f + e\vec{\varepsilon}_e, \quad f \in \mathbb{F},\ e \in \mathbb{R}$$

- Example : $\frac{1}{3}$ represented by:

$$\begin{array}{rcl} v & = & \uparrow_\circ \left(\frac{1}{3}\right) \vec{\varepsilon}_f + \left(\frac{1}{3} - \downarrow_\circ \left(\frac{1}{3}\right) \vec{\varepsilon}_e \\ & = & 0.333333\vec{\varepsilon}_f + \left(\frac{1}{3} - 0.333333\right) \vec{\varepsilon}_e \end{array}$$

- Interpretation of a constant d :

$$[\![d]\!]_{\mathbb{E}} = \uparrow_\circ (d)\vec{\varepsilon}_f + \downarrow_\circ (d)\vec{\varepsilon}_e$$

Elementary operations

$$x_1 = f_1\vec{\varepsilon}_f + e_1\vec{\varepsilon}_e \quad \text{et} \quad x_2 = f_2\vec{\varepsilon}_f + e_2\vec{\varepsilon}_e$$

$$[\![x_1 + x_2]\!]_{\mathbb{E}} = \uparrow_\circ (f_1 + f_2)\vec{\varepsilon}_f + [e_1 + e_2 + \downarrow_\circ (f_1 + f_2)]\,\vec{\varepsilon}_e$$

$$[\![x_1 - x_2]\!]_{\mathbb{E}} = \uparrow_\circ (f_1 - f_2)\vec{\varepsilon}_f + [e_1 - e_2 + \downarrow_\circ (f_1 - f_2)]\,\vec{\varepsilon}_e$$

$$[\![x_1 \times x_2]\!]_{\mathbb{E}} = \uparrow_\circ (f_1 \times f_2)\vec{\varepsilon}_f + [e_1 f_2 + e_2 f_1 + e_1 e_2 + \downarrow_\circ (f_1 \times f_2)]\,\vec{\varepsilon}_e$$

$$[\![\frac{1}{x_1}]\!]_{\mathbb{E}} = \uparrow_\circ \left(\frac{1}{f_1}\right)\vec{\varepsilon}_f + \left[\downarrow_\circ \left(\frac{1}{f_1}\right) + \sum_{n\geq 1}(-1)^n \frac{e_1^n}{f_1^{n+1}}\right]\vec{\varepsilon}_e$$

Division

$$\frac{1}{1+x} = \sum_{n\geq 0}(-1)^n x^n \text{ pour tout } x \text{ t.q. } -1 < x < 1$$

We have :

$$\frac{1}{f+e} = \frac{1}{f} \times \frac{1}{1+\frac{e}{f}} = \frac{1}{f} \times \sum_{n\geq 0}(-1)^n \frac{e^n}{f^n}$$

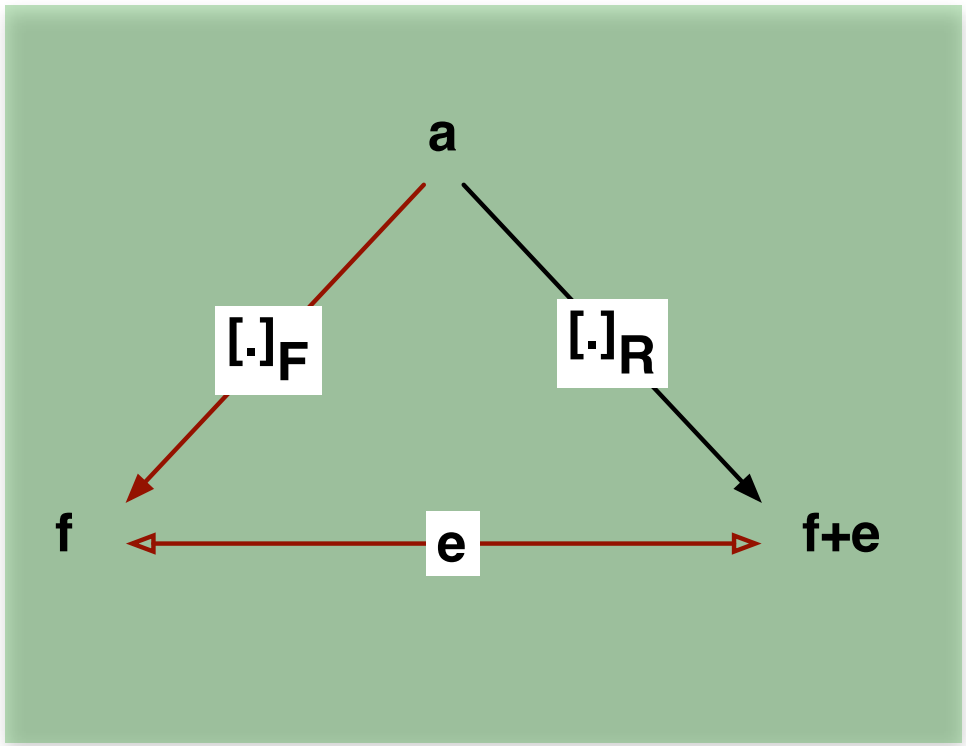
and :

$$[\![\frac{1}{x}]\!]_{\mathbb{E}} = \uparrow_{\circ}\left(\frac{1}{f}\right)\vec{\varepsilon}_f + \left[\downarrow_{\circ}\left(\frac{1}{f}\right) + \sum_{n\geq 1}(-1)^n \frac{e^n}{f^{n+1}}\right]\vec{\varepsilon}_e$$

Holds for $-1 < \frac{e}{f} < 1$ or equivalently for $|e| \leq |f|$, i.e. as long as the error is less than the floating-point number

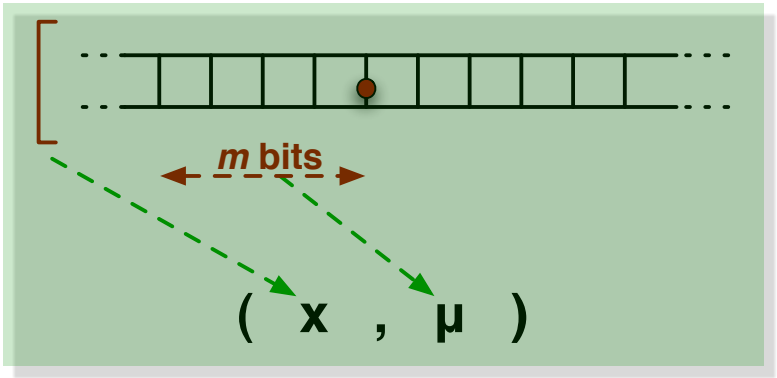
Property

Let a be an arithmetic expression.
If $[\![a]\!]_{\mathbb{E}} = f\vec{\varepsilon}_f + e\vec{\varepsilon}_e$ then $[\![a]\!]_{\mathbb{F}} = f$ and $[\![a]\!]_{\mathbb{R}} = f + e$.



Measuring in the Fixed-Point Format

Non standard value for $x_{\mathbb{R}}$:



Elementary operations:

$$\begin{array}{rl} & (x_1 \quad , \quad \mu_1) \\ + & (x_2 \quad , \quad \mu_2) \\ \hline = & ((x_1+x_2) \ , \quad \max \left| \begin{array}{l} \mu_1, \\ \mu_2, \\ \leftarrow(x_1+x_2)) \end{array} \right. \end{array}$$

$$\begin{array}{rl} & (x_1 \quad , \quad \mu_1) \\ \times & (x_2 \quad , \quad \mu_2) \\ \hline = & ((x_1 x_2) \ , \quad \max \left| \begin{array}{l} \mu_1, \\ \mu_2, \\ \leftarrow(x_1 x_2)) \end{array} \right. \end{array}$$

Example: For $E =$ `(a+(b+(c+d)))*e` we obtain $E_{\text{fixed}} = \big([-3,4],5\big)$

3 bits are enough for the integer part of the final result
But if `c`$=3.5$ and `d`$=13.5$ then `c+d`$=17$ and $\leftarrow_\circ(17) = 5$

Operations

$$(x_1^\sharp, \mu_1^\sharp) + (x_2^\sharp, \mu_2^\sharp) = \left(x_1^\sharp + x_2^\sharp, \max(\mu_1^\sharp, \mu_2^\sharp, \leftarrow_\circ^\sharp (x_1^\sharp + x_2^\sharp))\right)$$

$$(x_1^\sharp, \mu_1^\sharp) - (x_2^\sharp, \mu_2^\sharp) = \left(x_1^\sharp - x_2^\sharp, \max(\mu_1^\sharp, \mu_2^\sharp, \leftarrow_\circ^\sharp (x_1^\sharp - x_2^\sharp))\right)$$

$$(x_1^\sharp, \mu_1^\sharp) \times (x_2^\sharp, \mu_2^\sharp) = \left(x_1^\sharp \times x_2^\sharp, \max(\mu_1^\sharp, \mu_2^\sharp, \leftarrow_\circ^\sharp (x_1^\sharp \times x_2^\sharp))\right)$$

Semi-Automatic Improvement of the Quality (2)

Global indicators:

- Do not indicate why a quality indicator is bad
- No hint on how to improve the quality of a computation

Definition of a new semantics:

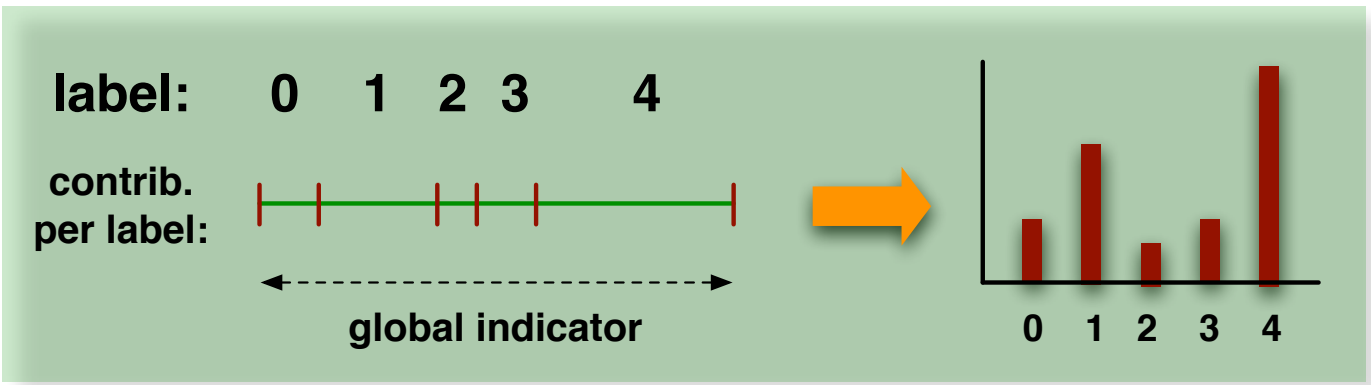
- Value = interval for the computer representable number + sequence of indicators
- Computes a safe approximation of the result
- Shows the main sources of quality losts by relating the indicators to operations

Semi-Automatic Improvement of the Quality

Labeled expressions: $e^{\ell} ::= v^{\ell} \mid e_1^{\ell_1} +^{\ell} e_2^{\ell_2} \mid e_1^{\ell_1} -^{\ell} e_2^{\ell_2} \mid e_1^{\ell_1} \times^{\ell} e_2^{\ell_2}$

Indicator tuples: $x_{\mathbb{S}} = (x, \langle \mu_{\ell_1}, \ldots, \mu_{\ell_n}, \mu_{\hbar} \rangle)$

x: computer representation of the number $x_{\mathbb{S}}$
The tuple $\langle \mu_{\ell_1}, \ldots, \mu_{\ell_n}, \mu_{\hbar} \rangle$ has one component per label $\ell_1, \ldots, \ell_n$
$\mu_{\hbar}$ special label used for higher order error terms



Semi-Automatic Improvement of the Quality (3)

Intuitively, in floating-point arithmetic:

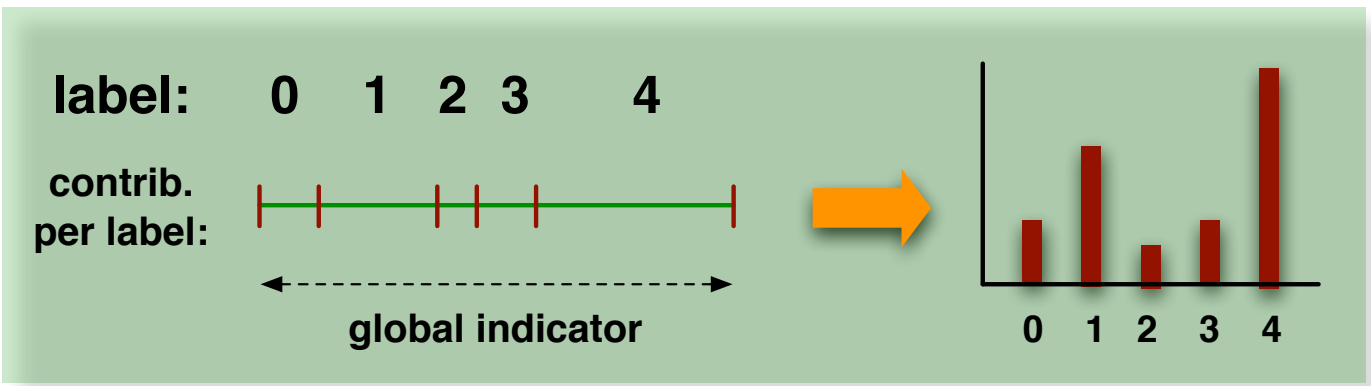
μ_{ℓ_k} is the contribution to the global error of the error introduced at ℓ_k

$$\mu = \mu_{\hbar} + \sum_{i=1}^{n} \mu_{\ell_i}$$

Intuitively, in fixed-point arithmetic:

μ_{ℓ_k} is the size of the integer part of the intermediary result at ℓ_k

$$\mu = \max \{\ell_k,\ \ell_k \in \mathrm{Lab}(e^{\ell})\}$$

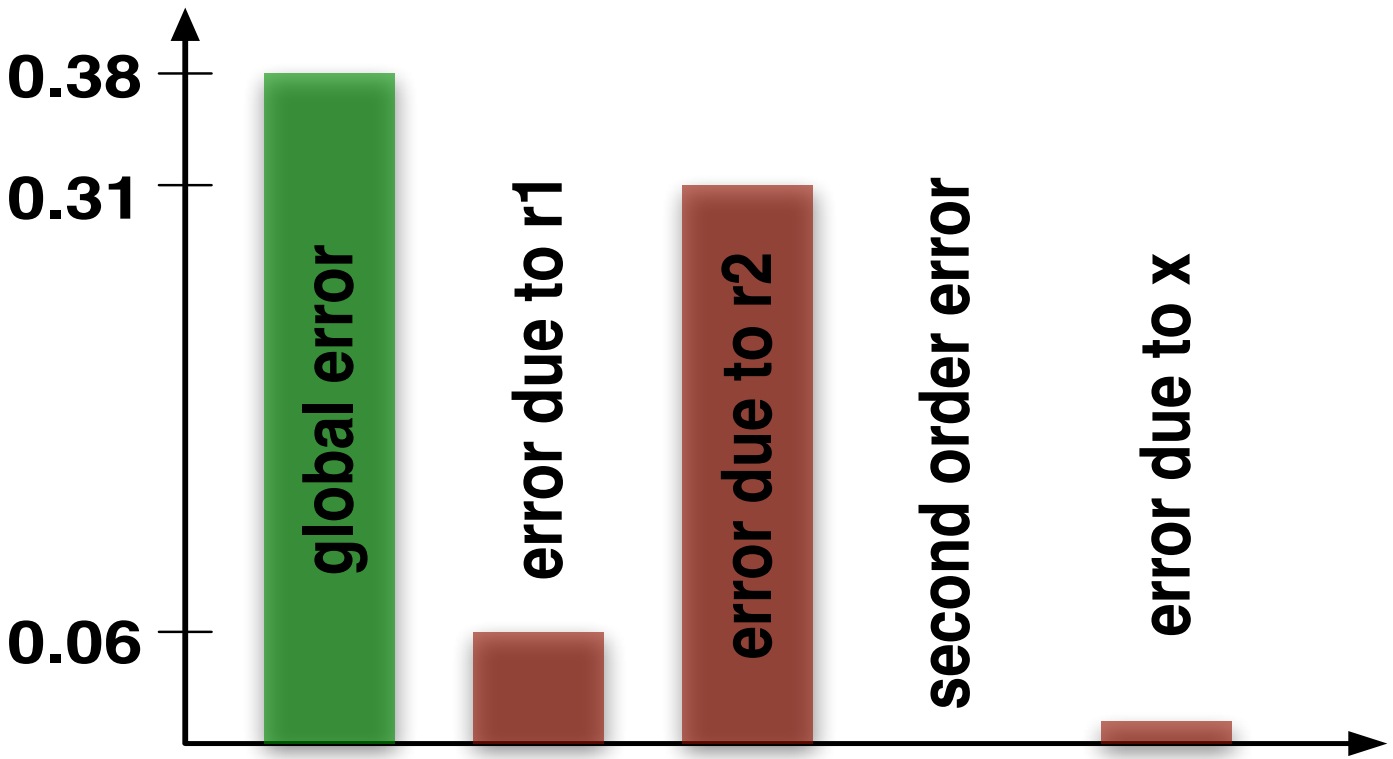


Introductory Example

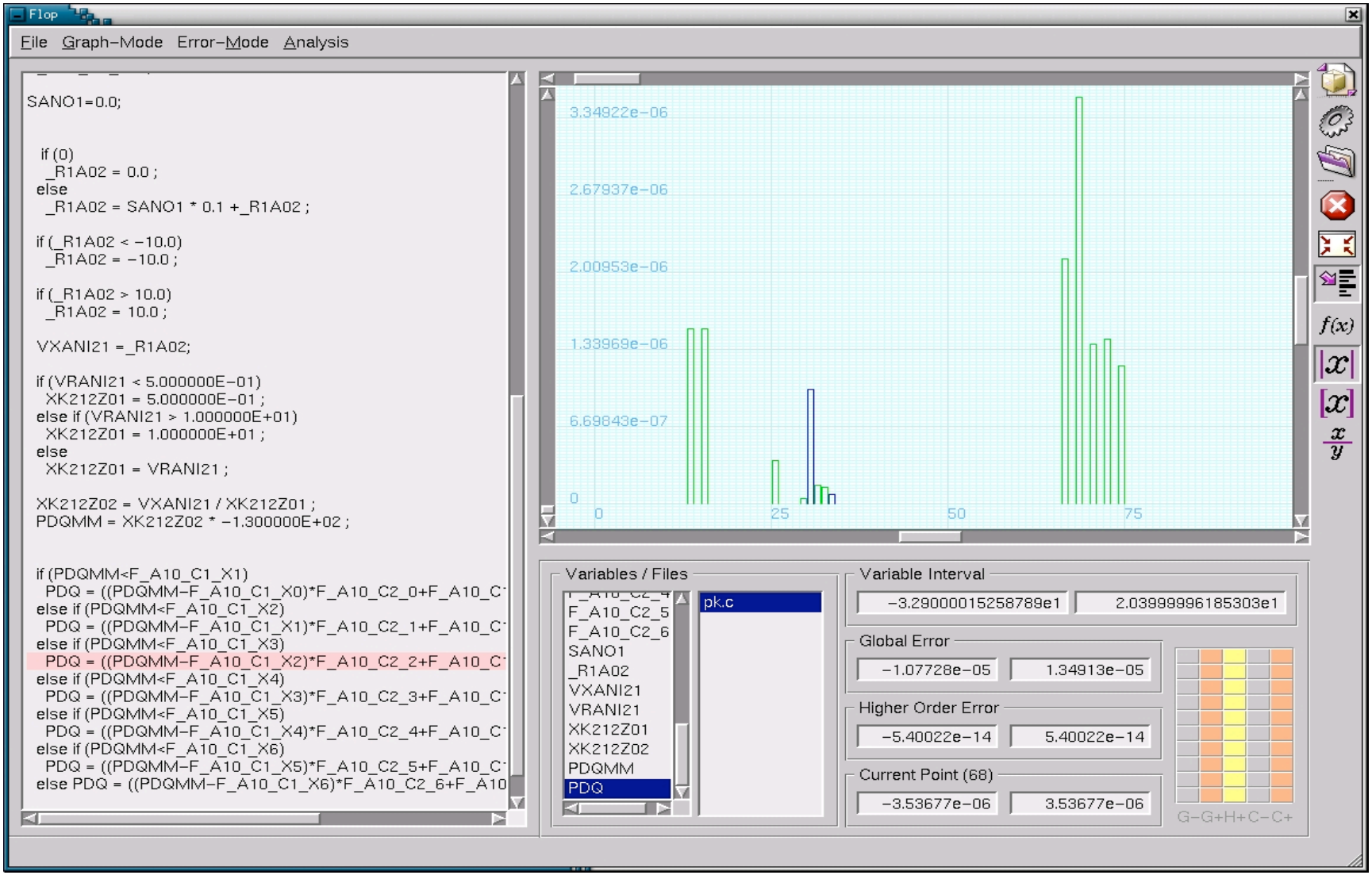
$$
\begin{array}{rrcrl}
 & 621.3\vec{\varepsilon} & + & 0.05\vec{\varepsilon}_{\ell_1} & r_1^{\ell_1} \\
\times^{\ell_3} & 1.287\vec{\varepsilon} & + & 0.0005\vec{\varepsilon}_{\ell_2} & r_2^{\ell_2} \\
\hline
= & 799.6131\vec{\varepsilon}\cdot\vec{\varepsilon} & & & \text{Result} \\
 & & + & 0.06435\vec{\varepsilon}\cdot\vec{\varepsilon}_{\ell_1} & \text{Error due to } r_1^{\ell_1} \\
 & & + & 0.31065\vec{\varepsilon}\cdot\vec{\varepsilon}_{\ell_2} & \text{Error due to } r_2^{\ell_2} \\
 & & + & 0.000025\vec{\varepsilon}_{\ell_1}\cdot\vec{\varepsilon}_{\ell_2} & \text{Second order term} \\
\hline
= & 799.6\vec{\varepsilon} & & & \text{Machine result} =\uparrow_\circ (r_1 \times r_2) \\
 & & + & 0.06435\vec{\varepsilon}_{\ell_1} & \text{Error due to } r_1^{\ell_1} \\
 & & + & 0.31065\vec{\varepsilon}_{\ell_2} & \text{Error due to } r_2^{\ell_2} \\
 & & + & 0.000025\vec{\varepsilon}_{\ell_1\ell_2} & \text{Second order term} \\
 & & + & 0.0131\vec{\varepsilon}_{\ell_3} & \text{Error due to } \times^{\ell_3} =\downarrow_\circ (r_1 \times r_2)
\end{array}
$$

Introductory example (2)

$$799.988125 = 799.6\vec{\varepsilon} + 0.06435\vec{\varepsilon}_{\ell_1} + 0.31065\vec{\varepsilon}_{\ell_2} + 0.000025\vec{\varepsilon}_{\ell_1\ell_2} + 0.0131\vec{\varepsilon}_{\ell_3}$$



Fluctuat


Flop
File Graph-Mode Error-Mode Analysis
SANO1=0.0;
if (0)
_R1A02 = 0.0 ;
else
_R1A02 = SANO1 * 0.1 +_R1A02 ;
if (_R1A02 < -10.0)
_R1A02 = -10.0 ;
if (_R1A02 > 10.0)
_R1A02 = 10.0 ;
VXANI21 =_R1A02;
if (VRANI21 < 5.000000E-01)
XK212Z01 = 5.000000E-01 ;
else if (VRANI21 > 1.000000E+01)
XK212Z01 = 1.000000E+01 ;
else
XK212Z01 = VRANI21 ;
XK212Z02 = VXANI21 / XK212Z01 ;
PDQMM = XK212Z02 * -1.300000E+02 ;
if (PDQMM<F_A10_C1_X1)
PDQ = ((PDQMM-F_A10_C1_X0)*F_A10_C2_0+F_A10_C
else if (PDQMM<F_A10_C1_X2)
PDQ = ((PDQMM-F_A10_C1_X1)*F_A10_C2_1+F_A10_C
else if (PDQMM<F_A10_C1_X3)
PDQ = ((PDQMM-F_A10_C1_X2)*F_A10_C2_2+F_A10_C
else if (PDQMM<F_A10_C1_X4)
PDQ = ((PDQMM-F_A10_C1_X3)*F_A10_C2_3+F_A10_C
else if (PDQMM<F_A10_C1_X5)
PDQ = ((PDQMM-F_A10_C1_X4)*F_A10_C2_4+F_A10_C
else if (PDQMM<F_A10_C1_X6)
PDQ = ((PDQMM-F_A10_C1_X5)*F_A10_C2_5+F_A10_C
else PDQ = ((PDQMM-F_A10_C1_X6)*F_A10_C2_6+F_A10
3.34922e-06
2.67937e-06
2.00953e-06
1.33969e-06
6.69843e-07
0
0
25
50
75
f(x)
Variables / Files
F_A10_C2_5
F_A10_C2_6
SANO1
_R1A02
VXANI21
VRANI21
XK212Z01
XK212Z02
PDQMM
PDQ
pk.c
Variable Interval
-3.290000152587891e1
2.039999961853031e1
Global Error
-1.07728e-05
1.34913e-05
Higher Order Error
-5.40022e-14
5.40022e-14
Current Point (68)
-3.53677e-06
3.53677e-06
G-G+H+C-C+

Abstract Semantics of Indicator Tuples (Floating-Point)

$$x^\sharp_{\mathbb{S}} = \left(x^\sharp, \langle \mu^\sharp_{\ell_1}, \ldots, \mu^\sharp_{\ell_n}, \mu^\sharp_{\hbar} \rangle\right) \text{ and } y^\sharp_{\mathbb{S}} = \left(y^\sharp, \langle \nu^\sharp_{\ell_1}, \ldots, \nu^\sharp_{\ell_n}, \nu^\sharp_{\hbar} \rangle\right)$$

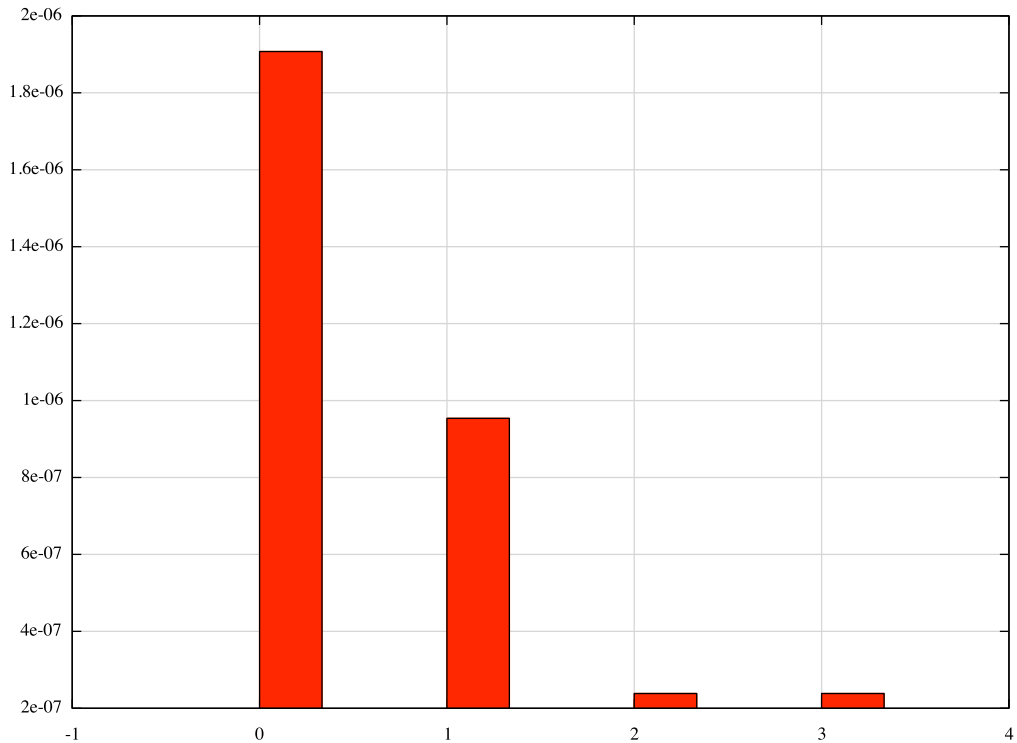
Elementary operations:

$$\begin{array}{lll} & (x & , \langle \mu_1, \mu_2, \ldots, \mu_k, \mu_h \rangle) \\ + & (y & , \langle \nu_1, \nu_2, \ldots, \nu_k, \nu_h \rangle) \\ \hline = & ((x+y) & , \langle \mu_1+\nu_1, \\ & & \quad \ldots \\ & & \quad \mu_h+\nu_h \rangle) \\ \hline = & (\uparrow(x_1+x_2) & , \langle \mu_1+\nu_1, \\ & & \quad \ldots \\ & & \quad \mu_h+\nu_h \rangle) \end{array} \longleftarrow +\downarrow(x_1+x_2)))$$

Example (floating-point)

$$E^{\ell} = (\mathtt{a} +^{\ell_2} (\mathtt{b} +^{\ell_1} (\mathtt{c} +^{\ell_0} \mathtt{d}))) \times^{\ell_3} \mathtt{e}$$

$$E^{\ell}_{\text{float}} = ([-3, 4], \langle \quad [-1.90734863281250 \cdot 10^{6}, 1.90734863281250 \cdot 10^{6}],$$
$$[-9.53674316406250 \cdot 10^{7}, 9.53674316406250 \cdot 10^{7}],$$
$$[-2.38418579101562 \cdot 10^{7}, 2.38418579101563 \cdot 10^{7}],$$
$$[-2.38418579101562 \cdot 10^{7}, 2.38418579101563 \cdot 10^{7}] \rangle)$$

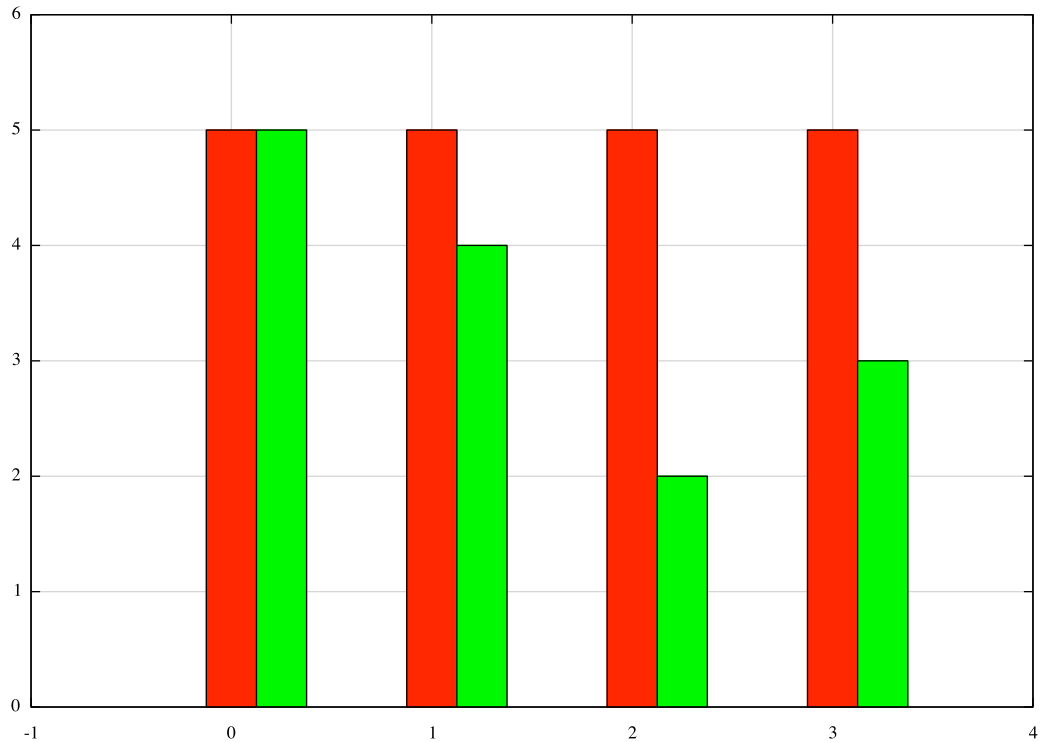


Example (fixed-point)

$$E^{\ell} = (\mathtt{a} +^{\ell_2} (\mathtt{b} +^{\ell_1} (\mathtt{c} +^{\ell_0} \mathtt{d}))) \times^{\ell_3} \mathtt{e}$$

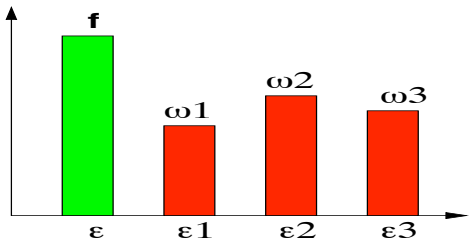
Local max: $E^{\ell}_{\text{fixed}} = \left([-3, 4], \langle 5, 4, 2, 3 \rangle\right)$

Global max: $E^{\ell}_{\text{fixed}} = \left([-3, 4], \langle 5, 5, 5, 5 \rangle\right)$



Sémantique générale $[\![.]\!]_{\mathbb{E}}^{\mathcal{L}^*}$

1^er ordre : $r^{\mathcal{L}^*} = f\vec{\varepsilon} + \sum_{\ell \in \mathcal{L}} \omega^\ell \vec{\varepsilon}_\ell$



- $f \in \mathbb{F}$ est le flottant utilisé par la machine au lieu de la valeur exacte r
- $\vec{\varepsilon}$ est une variable formelle toujours attachée au flottant f
- $\mathcal{L}$ ensemble des labels du programme
- $\vec{\varepsilon}_\ell$ variable formelle correspondant à l'erreur due au point $\ell \in \mathcal{L}$
- $\omega^\ell \in \mathbb{R}$ poids de l'erreur introduite au point $\ell \in \mathcal{L}$
- La valeur exacte dans $\mathbb{R}$ est : $r = f + \sum_{\ell \in \mathcal{L}} \omega^\ell$

Représentation des Erreurs d'Ordre Supérieur

- Erreur du 1^{er} ordre due au point ℓ attachée à la variable $\vec{\varepsilon}_\ell$
- Erreur du second ordre = produit de deux termes d'erreur du 1^{er} ordre, attaché à la variable formelle $\vec{\varepsilon}_{\ell_1 \ell_2}$

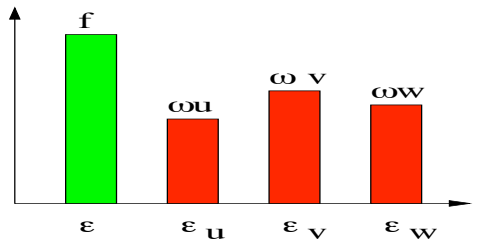
Exemple :

$$(f_1 \vec{\varepsilon} + \omega^{\ell_1} \vec{\varepsilon}_{\ell_1}) \times (f_2 \vec{\varepsilon} + \omega^{\ell_2} \vec{\varepsilon}_{\ell_2}) = f_1 f_2 \vec{\varepsilon} + f_2 \omega^{\ell_1} \vec{\varepsilon}_{\ell_1} + f_1 \omega^{\ell_2} \vec{\varepsilon}_{\ell_2} + \omega^{\ell_1} \omega^{\ell_2} \vec{\varepsilon}_{\ell_1 \ell_2}$$

- Plus généralement $\vec{\varepsilon}_{\ell_1 \ldots \ell_n}$ est la variable correspondant à l'erreur d'ordre n due aux points $\ell_1, \ldots, \ell_n$

Représentation des Erreurs d'Ordre Supérieur (2)

$$r^{\mathcal{L}^*} = f\vec{\varepsilon} + \sum_{u \in \overline{\mathcal{L}^+}} \omega^u \vec{\varepsilon}_u$$



- $f \in \mathbb{F}$ est le nombre flottant utilisé par la machine au lieu de la valeur exacte $r \in \mathbb{R}$
- $\overline{\mathcal{L}^+} \subseteq \mathcal{L}^*$ est un ensemble de mots sur l'alphabet $\mathcal{L}$ =Lab(prg)
- Pour tout mot $u = \ell_1 \dots \ell_n \in \overline{\mathcal{L}^+}$, $\vec{\varepsilon}_u$ est une variable formelle correspondant à l'erreur d'ordre n due aux points $\ell_1, \dots, \ell_n$
- $\omega^u \in \mathbb{R}$ est le coefficient de $\vec{\varepsilon}_u$
- La valeur exacte dans $\mathbb{R}$ est : $r = f + \sum_{u \in \overline{\mathcal{L}^+}} \omega^\ell$

Représentation des Erreurs d'Ordre Supérieur (3)

$$r^{\mathcal{L}^*} = f\vec{\varepsilon} + \sum_{u \in \overline{\mathcal{L}^+}} \omega^u \vec{\varepsilon}_u$$

- $\mathcal{L}$ ensemble des étiquettes, utilisé comme alphabet, $\mathcal{L}^*$ mots de $\mathcal{L}$
- ϵ mot vide, $\vec{\varepsilon} = \vec{\varepsilon}_\epsilon =$ variable attachée au flottant $f = \omega^\epsilon$
- $\mathcal{L}^+ = \mathcal{L} \setminus \{\epsilon\}$
- $\overline{\mathcal{L}^+}$ mots de $\mathcal{L}^+$ composés des mêmes lettres ($\vec{\varepsilon}_{\ell_1\ell_2} = \vec{\varepsilon}_{\ell_2\ell_1}$)

Opérations Elémentaires (Addition)

$$r_1 = f_1\vec{\varepsilon} + \sum_{u\in\overline{\mathcal{L}^+}} \omega_1^u \vec{\varepsilon}_u \qquad r_2 = f_2\vec{\varepsilon} + \sum_{u\in\overline{\mathcal{L}^+}} \omega_2^u \vec{\varepsilon}_u$$

$$r_1 +^{\ell_i} r_2 \overset{\text{def}}{=} \uparrow_\circ (f_1+f_2)\vec{\varepsilon} + \sum_{u\in\overline{\mathcal{L}^+}} (\omega_1^u+\omega_2^u)\vec{\varepsilon}_u + \downarrow_\circ (f_1+f_2)\vec{\varepsilon}_{\ell_i}$$

$$\begin{array}{rrcr} & 621.3\vec{\varepsilon} & + & 0.05\vec{\varepsilon}_{\ell_1} \\ +^{\ell_3} & 1.287\vec{\varepsilon} & + & 0.0005\vec{\varepsilon}_{\ell_2} \\ \hline = & 622.5\vec{\varepsilon} & & \\ & & + & 0.05\vec{\varepsilon}_{\ell_1} \\ & & + & 0.0005\vec{\varepsilon}_{\ell_2} \\ & & + & 0.087\vec{\varepsilon}_{\ell_3} \end{array}$$

$r_1^{\ell_1}$

$r_2^{\ell_2}$

Résultat machine $=\uparrow_\circ (r_1+r_2)$

Erreur due à $r_1^{\ell_1}$

Erreur due à $r_2^{\ell_2}$

Erreur due à $+^{\ell_3} = \downarrow_\circ (r_1+r_2)$

Opérations Elémentaires (Multiplication)

$$r_1 \times^{\ell_i} r_2 \overset{\text{def}}{=} \uparrow_\circ (f_1 f_2)\vec{\varepsilon} + \sum_{\substack{u \in \overline{\mathcal{L}^*},\ v \in \overline{\mathcal{L}^*} \\ |u.v| > 0}} \omega_1^u \omega_2^v \vec{\varepsilon}_{u.v} + \downarrow_\circ (f_1 f_2)\vec{\varepsilon}_{\ell_i}$$

	$621.3\vec{\varepsilon}$	$+$	$0.05\vec{\varepsilon}_{\ell_1}$	$r_1^{\ell_1}$
$\times^{\ell_3}$	$1.287\vec{\varepsilon}$	$+$	$0.0005\vec{\varepsilon}_{\ell_2}$	$r_2^{\ell_2}$
$=$	$799.6\vec{\varepsilon}$			Résultat machine $=\uparrow_\circ (r_1 \times r_2)$
		$+$	$0.06435\vec{\varepsilon}_{\ell_1}$	Erreur due à $r_1^{\ell_1}$
		$+$	$0.31065\vec{\varepsilon}_{\ell_2}$	Erreur due à $r_2^{\ell_2}$
		$+$	$0.000025\vec{\varepsilon}_{\ell_1 \ell_2}$	Terme du second ordre
		$+$	$0.0131\vec{\varepsilon}_{\ell_3}$	Erreur due à $\times^{\ell_3} = \downarrow_\circ (r_1 \times r_2)$

Opérations Elémentaires (Division)

- Inverse : obtenu par un développement en séries (problème : convergence)

$$(r_1)^{-1^{\ell_i}} \stackrel{\text{def}}{=} \uparrow_\circ (f_1^{-1})\vec{\varepsilon} + \frac{1}{f_1}\sum_{n\geq 1}(-1)^n \left(\sum_{u\in\mathcal{W}^+} \frac{\omega_1^u}{f_1}\vec{\varepsilon}_u\right)^n + \downarrow_\circ (f_1^{-1})\vec{\varepsilon}_{\ell_i}$$

- Division : combine produit et inverse

$$r_1 \div^{\ell_i} r_2 \stackrel{\text{def}}{=} \uparrow_\circ \left(\frac{f_1}{f_2}\right)\vec{\varepsilon} + \sum_{u\in\mathcal{W}} \frac{\omega_1^u}{f_2}\sum_{n\geq 0}(-1)^n \left(\sum_{v\in\mathcal{W}^+} \frac{\omega_2^v}{f_2}\vec{\varepsilon}_v\right)^n \vec{\varepsilon}_u + \downarrow_\circ \left(\frac{f_1}{f_2}\right)\vec{\varepsilon}_{\ell_i}$$

- remarque : $r_1 \div^{\ell_i} r_2 \neq r_1 \times^{\ell_i} r_2^{-1^{\ell_i}}$

Convergence

- Rayon de convergence de la série

$$\frac{1}{1+x} = \sum_{n \geq 0} (-1)^n x^n \text{ pour tout } x \text{ t.q. } -1 < x < 1$$

- Pour $x = \sum_{u \in \mathcal{W}^+} \frac{\omega_1^u}{f_1}$ il faut

$$-1 < \sum_{u \in \mathcal{W}^+} \frac{\omega_1^u}{f_1} < 1$$

- c.à.d.

$$-f_1 < \sum_{u \in \mathcal{W}^+} \omega_1^u < f_1$$

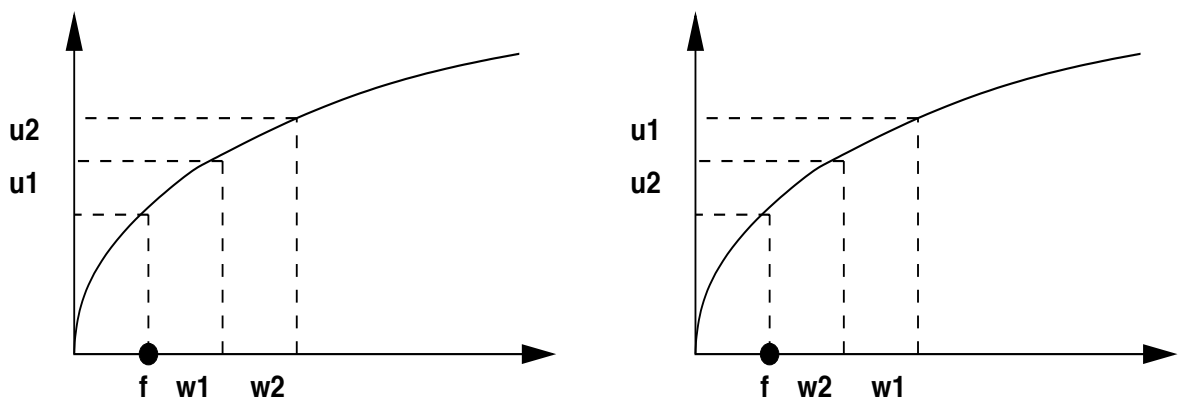
- L'erreur globale doit être inférieure au flottant

Racine carrée

- Développement en Série : si $-1 < \sum_{u\in\mathcal{W}^+} \frac{\omega^u}{f} < 1$ alors

$$\begin{aligned}\sqrt{r}^{\ell_i} &= \uparrow_\circ (\sqrt{f})\vec{\varepsilon} + \downarrow_\circ (\sqrt{f})\vec{\varepsilon}_{\ell_i} \\ &+ \sum_{n\geq 1}\left[\frac{\frac{1}{2}(-\frac{1}{2})\dots(\frac{3}{2}-n)}{n!}\sqrt{f}\left(\sum_{u\in\mathcal{W}^+}\frac{\omega^u}{f}\vec{\varepsilon}_u\right)^n\right]\end{aligned}$$

- Comment $\sqrt{\ }$ transforme $f + \omega_1\vec{\varepsilon}_1 + \omega_2\vec{\varepsilon}_2$ en $f' + \nu_1\vec{\varepsilon}_1 + \nu_2\vec{\varepsilon}_2$?



Correction (1)

$$r_1 = \sum_{u \in \mathcal{W}} \omega_1^u \vec{\varepsilon}_u \qquad r_2 = \sum_{u \in \mathcal{W}} \omega_2^u \vec{\varepsilon}_u \qquad r = r_1 \Diamond^{\ell_i} r_2 = \sum_{u \in \mathcal{W}} \omega^u \vec{\varepsilon}_u$$

- Principe : montrer que $\mathbb{R}(r_1) \Diamond \mathbb{R}(r_2) = \mathbb{R}(r)$

$$\sum_{u \in \mathcal{W}} \omega^u = \left(\sum_{u \in \mathcal{W}} \omega_1^u \right) \Diamond \left(\sum_{u \in \mathcal{W}} \omega_2^u \right)$$

- **Trop faible, par exemple:**

$$(f_1 \vec{\varepsilon} + \omega^{\ell_1} \vec{\varepsilon}_{\ell_1}) +^{\ell_{\ell_i}} (f_2 \vec{\varepsilon} + \omega^{\ell_2} \vec{\varepsilon}_{\ell_2}) \overset{\text{faux!}}{=} \uparrow_\circ (f_1 + f_2) \vec{\varepsilon} + \omega^{\ell_1} \vec{\varepsilon}_{\ell_2} + \omega^{\ell_2} \vec{\varepsilon}_{\ell_1} + \downarrow_\circ (f_1 + f_2) \vec{\varepsilon}_{\ell_i}$$

Correction (2)

$$r_1 = \sum_{u \in \mathcal{W}} \omega_1^u \vec{\varepsilon}_u \qquad r_2 = \sum_{u \in \mathcal{W}} \omega_2^u \vec{\varepsilon}_u \qquad r = r_1 \Diamond^{\ell_i} r_2 = \sum_{u \in \mathcal{W}} \omega^u \vec{\varepsilon}_u$$

- Principe : étudier les variations des coefficients ω_1^u et ω_2^u
- Les variations de $\omega_1^{u_1} \dots \omega_1^{u_k} \dots \omega_2^{u_{k+1}} \dots \omega_2^{u_n}$ sont données par $\frac{\partial^n}{\partial \omega_{k_1}^{u_1} \dots \omega_{k_n}^{u_n}}$

Propriété

$$\frac{\partial (r_1 \Diamond r_2)}{\partial \omega_1^{u_0}} = \frac{\partial r}{\partial \omega_1^{u_0}} \quad \textit{et} \quad \frac{\partial (r_1 \Diamond r_2)}{\partial \omega_2^{u_0}} = \frac{\partial r}{\partial \omega_2^{u_0}}$$

Propriété

$$\frac{\partial^n (r_1 \Diamond r_2)}{\partial \omega_{k_1}^{u_1} \dots \omega_{k_n}^{u_n}} = \frac{\partial^n r}{\partial \omega_{k_1}^{u_1} \dots \omega_{k_n}^{u_n}}$$

Preuve (multiplication)

D'un côté nous avons : $\frac{\partial (r_1 \times^{\ell i} r_2)}{\partial \omega_1^{u_0}} = \frac{\partial}{\partial \omega_1^{u_0}} \left(\sum_{u,v \in \mathcal{W}} \omega_1^u \omega_2^v \vec{\varepsilon}_{uv} \right)$ En utilisant l'égalité

$$\sum_{u,v \in \mathcal{W}} \omega_1^u \omega_2^v \vec{\varepsilon}_{uv} = \sum_{v \in \mathcal{W}} \omega_1^{u_0} \omega_2^v \vec{\varepsilon}_{u_0 v} + \sum_{u \in \mathcal{W} \setminus \{u_0\}, v \in \mathcal{W}} \omega_1^u \omega_2^v \vec{\varepsilon}_{uv}$$

on obtient : $\frac{\partial (r_1 \times^{\ell i} r_2)}{\partial \omega_1^{u_0}} = \sum_{v \in \mathcal{W}} \omega_2^v \vec{\varepsilon}_{u_0 v}$ Par ailleurs,

$$\frac{\partial}{\partial \omega_1^{u_0}} \left(\sum_{u \in \mathcal{W}} \omega_1^u \vec{\varepsilon}_u \times \sum_{v \in \mathcal{W}} \omega_2^u \vec{\varepsilon}_u \right) =$$

$$\sum_{u \in \mathcal{W}} \omega_1^u \vec{\varepsilon}_u \times \frac{\partial}{\partial \omega_1^{u_0}} \left(\sum_{v \in \mathcal{W}} \omega_2^v \vec{\varepsilon}_v \right) + \sum_{v \in \mathcal{W}} \omega_2^v \vec{\varepsilon}_v \times \frac{\partial}{\partial \omega_1^{u_0}} \left(\sum_{u \in \mathcal{W}} \omega_1^u \vec{\varepsilon}_u \right) =$$

$$0 + \left(\sum_{v \in \mathcal{W}} \omega_2^v \vec{\varepsilon}_v \right) \times \vec{\varepsilon}_{u_0} = \sum_{v \in \mathcal{W}} \omega_2^v \vec{\varepsilon}_{u_0 v}$$

Propriétés de $[\![.]\!]_{\mathbb{E}}^{\mathcal{L}^*}$

Propriété

a expression arithmétique telle que $[\![a]\!]_{\mathbb{E}} = f\vec{\varepsilon}_f + e\vec{\varepsilon}_e$ *et* $[\![a]\!]_{\mathbb{W}} = \omega\vec{\varepsilon} + \sum_{\ell\in\mathcal{L}} \omega^\ell \vec{\varepsilon}_\ell$. *Alors* $\omega = f$ *et* $\sum_{\ell\in\mathcal{L}} \omega^\ell = e$.

Remarques concernant $[\![.]\!]^{\mathcal{L}^*}_{\mathbb{E}}$

- Comme $[\![.]\!]_{\mathbb{E}}$, $[\![.]\!]_{\mathbb{W}}$ utilise des réels
- Nouvelles sémantiques : $[\![.]\!]_{\mathbb{WI}}$, $[\![.]\!]_{\mathbb{WS}}$
- Les erreurs du premier ordre de $[\![.]\!]_{\mathbb{W}}$ peuvent être estimées à partir de $[\![.]\!]_{\mathbb{D}}$. Soit d_i donnée définie au point ℓ d'un prg calculant $g(d_1, \ldots, d_n)$:

$$\downarrow_{\circ} (d_i) \times \frac{\partial g}{\partial d_i}(d_1, \ldots, d_n) \approx \omega^{\ell}$$

Restriction à l'Ordre n : $[\![.]\!]^{\mathcal{L}^n}$

- n opérations introduisent éventuellement des erreurs d'ordre n
  - Comment réduire la taille des valeurs?
- En pratique :
  - Seules les erreurs du premier ordre sont significatives
  - Très rarement, celles d'ordre 2 ne sont pas négligeables
- Principe :
  - Détailler la provenance des erreurs des n premiers ordres uniquement
  - Vérifier que les erreurs d'ordre supérieur sont globalement négligeables

Erreurs d'Ordre n : Principe

- Détaille la contribution des erreurs d'ordre $\leq n$
- Indique globalement le poids des erreurs $> n$
  - On a $\omega_{hi} =$ somme des erreurs d'ordre $> n$

Opérations Elémentaires ($\mathcal{W} = \overline{\mathcal{L}^n}$)

- $\overline{\mathcal{L}^n} = \{u \in \overline{\mathcal{L}^*} \ : \ |u| \leq n\} \cup \{hi\}$
- Concaténation : $u \cdot_n v = \begin{cases} u.v \text{ si } |u.v| \leq n \\ hi \text{ sinon} \end{cases}$

$$r_1 +^{\ell_i} r_2 \stackrel{\text{def}}{=} \uparrow_\circ (f_1 + f_2)\vec{\varepsilon} + \sum_{u \in \mathcal{W}^+} (\omega_1^u + \omega_2^u)\vec{\varepsilon}_u + \downarrow_\circ (f_1 + f_2)\vec{\varepsilon}_{\ell_i}$$

$$r_1 \times^{\ell_i} r_2 \stackrel{\text{def}}{=} \uparrow_\circ (f_1 f_2)\vec{\varepsilon} + \sum_{\substack{u \in \mathcal{W},\ v \in \mathcal{W} \\ |u.v| > 0}} \omega_1^u \omega_2^v \vec{\varepsilon}_{u.v} + \downarrow_\circ (f_1 f_2)\vec{\varepsilon}_{\ell_i}$$

Relation entre $[\![.]\!]^{\mathcal{L}^n}$ et $[\![.]\!]^{\mathcal{L}^m}$

- $[\![.]\!]^{\mathcal{L}^n}$ détaille la provenance des erreurs jusqu'à l'ordre n et calcule la somme des erreurs d'ordre $> n$
- Pour $m < n$ la sémantique à l'ordre m est une approximation de celle à l'ordre n :

$$\langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n})), \subseteq \rangle \quad \underset{\alpha^{n,m}}{\overset{\gamma^{m,n}}{\leftrightarrows}} \quad \langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^m})), \subseteq \rangle$$

- On obtient une chaine de sémantiques de plus en plus précises :

$$[\![.]\!]^{\mathcal{L}^*}(a_0^{\ell_0}) \leftrightarrows \ldots [\![.]\!]^{\mathcal{L}^n}(a_0^{\ell_0}) \leftrightarrows [\![.]\!]^{\mathcal{L}^{n-1}}(a_0^{\ell_0}) \ldots \leftrightarrows [\![.]\!]^{\mathcal{L}^0}(a_0^{\ell_0})$$

Relation entre $[\![.]\!]^{\mathcal{L}^n}$ et $[\![.]\!]^{\mathcal{L}^m}$

$$\alpha^{n,m}\left(\sum_{u\in\overline{\mathcal{L}^n}}\omega^u\vec{\varepsilon}_u\right)\stackrel{\text{def}}{=}\sum_{u\in\overline{\mathcal{L}^m}\setminus\{hi\}}\omega^u\vec{\varepsilon}_u+\left(\sum_{u\in(\overline{\mathcal{L}^n}\setminus\overline{\mathcal{L}^m})\cup\{hi\}}\omega^u\right)\vec{\varepsilon}_{hi}$$

$$\gamma^{m,n}\left(\sum_{u\in\overline{\mathcal{L}^m}}\nu^u\vec{\varepsilon}_u\right)\stackrel{\text{def}}{=}\left\{\sum_{u\in\overline{\mathcal{L}^n}}\omega^u\vec{\varepsilon}_u\ :\ \left|\begin{array}{l}\omega^u=\nu^u\text{ if }u\in\overline{\mathcal{L}^m}\setminus\{hi\}\\ \sum_{u\in(\overline{\mathcal{L}^n}\setminus\overline{\mathcal{L}^m})\cup\{hi\}}\omega^u=\nu^{hi}\end{array}\right.\right\}$$

$$\alpha^{n,m}(X)=\{\alpha^{n,m}(x)\ :\ x\in X\}\quad\gamma^{m,n}(X)=\cup_{x\in X}\gamma^{m,n}(x)$$

Correction

Propriété

Soit ℓ_i un point de contrôle, soit $r^{\mathcal{L}^n}, s^{\mathcal{L}^n} \in \mathcal{F}(\mathbb{R}, \overline{\mathcal{L}^n})$, des séries d'erreurs telles que $r^{\mathcal{L}^m} = \alpha^{n,m}(r^{\mathcal{L}^n})$, $s^{\mathcal{L}^m} = \alpha^{n,m}(s^{\mathcal{L}^n})$, $1 \leq m \leq n$. Pour toute opération $\Diamond \in \{+, -, \times, \div\}$ nous avons

$$r^{\mathcal{L}^n} \Diamond^{\ell_i} s^{\mathcal{L}^n} \in \gamma^{m,n}(r^{\mathcal{L}^m} \Diamond^{\ell_i} s^{\mathcal{L}^m})$$

Preuve (multiplication)

Soit

$$r^{\mathcal{L}^n} = \sum_{u \in \overline{\mathcal{L}^n}} \omega_r^u \vec{\varepsilon}_u$$

et

$$s^{\mathcal{L}^n} = \sum_{u \in \overline{\mathcal{L}^n}} \omega_s^u \vec{\varepsilon}_u$$

On a :

$$\begin{array}{rcl} t^{\mathcal{L}^n} & = & r^{\mathcal{L}^n} \times^{\ell_i} s^{\mathcal{L}^n} \\ & = & \uparrow_\circ (\omega_r^\epsilon \omega_s^\epsilon) \vec{\varepsilon}_\epsilon + \sum_{\substack{u,\, v \,\in\, \overline{\mathcal{L}^n} \\ |u.v| > 0}} \omega_r^u \omega_s^v \vec{\varepsilon}_{u.v} + \downarrow_\circ (\omega_r^\epsilon \omega_s^\epsilon) \vec{\varepsilon}_{\ell_i} \end{array} \tag{1}$$

De même, si $r^{\mathcal{L}^m} = \sum_{u \in \overline{\mathcal{L}^m}} \nu_r^u \vec{\varepsilon}_u$ et $s^{\mathcal{L}^m} = \sum_{u \in \overline{\mathcal{L}^m}} \nu_s^u \vec{\varepsilon}_u$ alors

$$\begin{array}{rcl} t^{\mathcal{L}^m} & = & r^{\mathcal{L}^m} \times^{\ell_i} s^{\mathcal{L}^m} \\ & = & \uparrow_\circ (\nu_r^\epsilon \nu_s^\epsilon) \vec{\varepsilon}_\epsilon + \sum_{\substack{u,\, v \,\in\, \overline{\mathcal{L}^m} \\ |u.v| > 0}} \nu_r^u \nu_s^v \vec{\varepsilon}_{u.v} + \downarrow_\circ (\nu_r^\epsilon \nu_s^\epsilon) \vec{\varepsilon}_{\ell_i} \end{array} \tag{2}$$

Par definition de $\gamma^{m,n}$,

$$\gamma^{m,n}(t^{\mathcal{L}^m}) =$$

$$\left\{ \sum_{u \in \overline{\mathcal{L}^n}} \omega^u \vec{\varepsilon}_u \ : \ \left| \begin{array}{l} \omega^u = \nu_t^u \text{ if } u \in \overline{\mathcal{L}^m} \setminus \{hi\} \\ \sum_{u \in (\overline{\mathcal{L}^n} \setminus \overline{\mathcal{L}^m}) \cup \{hi\}} \omega^u = \nu_t^{hi} \end{array} \right. \right\} \tag{3}$$

où, dans (3),

$$\nu_t^u = \sum_{u_1 u_2 = u} \nu_r^{u_1} \nu_s^{u_2}$$

Grain d'Erreur : $[\![.]\!]^{\mathcal{T}^n}$

- But : limiter le nombre de termes dans les séries d'erreurs
- Principe :
  - Ne plus considérer les erreurs introduites par des opérations élémentaires
  - Calculer les erreurs due à des morceaux de code partitionnant le programme
- Exemples :
  - Erreur due à une formule intermédiaire
  - Erreur introduite par un bloc de programme C
  - $[\![.]\!]^{\mathcal{L}^*}$, $[\![.]\!]^{\mathcal{L}^n}$ correspondent à une partition particulière, celle des singletons de Lab(prg)

Grain d'Erreur : Principe

- $\mathcal{J} = \{J_1, \ldots J_m\}$ partition de l'ensemble $\mathcal{L}$ des étiquettes
- Pour les termes du premier ordre:
- Pour les termes d'ordre supérieur : $\omega^{J_1 J_2} = \sum_{\ell_1 \in J_1, \ell_2 \in J_2} \omega^{\ell_1} \omega^{\ell_2}$
- Sémantique des opérations : $\mathcal{W} = \overline{\mathcal{J}^n}$

$$r_1 +^{\ell_i} r_2 \stackrel{\text{def}}{=} \uparrow_\circ (f_1 + f_2)\vec{\varepsilon} + \sum_{u \in \mathcal{W}^+} (\omega_1^u + \omega_2^u)\vec{\varepsilon}_u + \downarrow_\circ (f_1 + f_2)\vec{\varepsilon}_{\ell_i}$$

Correction de $[\![.]\!]^{\mathcal{J}^n}$

- Ordre partiel $\dot{\subseteq}$: $\mathcal{J}_1$ est plus précis que $\mathcal{J}_2$ si $\mathcal{J}_2$ regroupe des éléments de $\mathcal{J}_1$
- La sémantique $[\![.]\!]^{\mathcal{J}_2^n}$ utilisant la partition $\mathcal{J}_2$ t.q. $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$ est une approximation de la sémantique $[\![.]\!]^{\mathcal{J}_1^n}$

$$\langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_1^n})), \subseteq \rangle \quad \underset{\alpha^{\mathcal{J}_1^n, \mathcal{J}_2^n}}{\overset{\gamma^{\mathcal{J}_2^n, \mathcal{J}_1^n}}{\leftrightarrows}} \quad \langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_2^n})), \subseteq \rangle$$

Correction de $[\![.]\!]^{\mathcal{J}^n}$

$$\langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_1^n})), \subseteq \rangle \quad \underset{\alpha^{\mathcal{J}_1^n, \mathcal{J}_2^n}}{\overset{\gamma^{\mathcal{J}_2^n, \mathcal{J}_1^n}}{\leftrightarrows}} \quad \langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_2^n})), \subseteq \rangle$$

$$\tau_{\mathcal{J}_1^n, \mathcal{J}_2^n}(J_1.u) = J_2.\tau_{\mathcal{J}_1^n, \mathcal{J}_2^n}(u) \text{ avec } J_1 \in \mathcal{J}_1,\ J_1 \subseteq J_2,\ J_2 \in \mathcal{J}_2$$

$$\alpha^{\mathcal{J}_1^n, \mathcal{J}_2^n}\left(\sum_{u \in \overline{\mathcal{J}_1^n}} \omega_i^u \vec{\varepsilon}_u\right) \overset{\text{def}}{=} \sum_{u \in \overline{\mathcal{J}_1^n}} \omega^u \vec{\varepsilon}_{\tau_{\mathcal{J}_1^n, \mathcal{J}_2^n}(u)}$$

$$\gamma^{\mathcal{J}_2^n, \mathcal{J}_1^n}\left(\sum_{v \in \overline{\mathcal{J}_2^n}} \nu^u \vec{\varepsilon}_v\right) \overset{\text{def}}{=} \left\{ \sum_{u \in \overline{\mathcal{J}_1^n}} \omega^u \vec{\varepsilon}_u \ : \sum_{\tau_{\mathcal{J}_1^n, \mathcal{J}_2^n}(u) = v} \omega^u = \nu^v \right\}$$

Correction

Propriété

Soit ℓ_i un point de contrôle, soit $\mathcal{J}_1$ et $\mathcal{J}_2$ des partitions de $\mathcal{L}$ telles que $\mathcal{J}_1 \dot{\subseteq} \mathcal{J}_2$ et soient $r^{\mathcal{J}_1^n}, s^{\mathcal{J}_1^n} \in \mathcal{F}(\mathbb{R}, \overline{\mathcal{J}_1^n})$. Si $r^{\mathcal{J}_2^n} = \alpha^{\mathcal{J}_1^n, \mathcal{J}_2^n}(r^{\mathcal{J}_1^n})$, $s^{\mathcal{J}_2^n} = \alpha^{\mathcal{J}_1^n, \mathcal{J}_2^n}(s^{\mathcal{J}_1^n})$ alors pour tout opérateur $\Diamond \in \{+, -, \times, \div\}$ nous avons :

$$r^{\mathcal{J}_1^n} \Diamond^{\ell_i} s^{\mathcal{J}_1^n} \in \gamma^{\mathcal{J}_2^n, \mathcal{J}_1^n}(r^{\mathcal{J}_2^n} \Diamond^{\ell_i} s^{\mathcal{J}_2^n})$$

Preuve (multiplication)

On utilise les notations :

$$r^{\mathcal{J}_1^n} = \sum_{u \in \overline{\mathcal{J}_1^n}} \omega_r^u \vec{\varepsilon}_u, \qquad s^{\mathcal{J}_1^n} = \sum_{u \in \overline{\mathcal{J}_1^n}} \omega_s^u \vec{\varepsilon}_u,$$

$$r^{\mathcal{J}_2^n} = \sum_{u \in \overline{\mathcal{J}_2^n}} \nu_r^u \vec{\varepsilon}_u, \qquad s^{\mathcal{J}_2^n} = \sum_{u \in \overline{\mathcal{J}_2^n}} \nu_s^u \vec{\varepsilon}_u.$$

Soit $\tau(u) = \tau_{\mathcal{J}_1^n, \mathcal{J}_2^n}(u)$. Nous avons :

$$t^{\mathcal{J}_1^n} = r^{\mathcal{J}_1^n} \times^{\ell_i} s^{\mathcal{J}_1^n} = \uparrow_\circ (\omega_r^\epsilon \omega_s^\epsilon) \vec{\varepsilon}_\epsilon + \sum_{\substack{u, v \in \overline{\mathcal{J}_1^n} \\ |u.v| > 0}} \omega_r^u \omega_s^v \vec{\varepsilon}_{u.v} + \downarrow_\circ (\omega_r^\epsilon \omega_s^\epsilon) \vec{\varepsilon}_{\ell_i}$$

$$t^{\mathcal{J}_2^n} = r^{\mathcal{J}_2^n} \times^{\ell_i} s^{\mathcal{J}_2^n} = \uparrow_\circ (\nu_r^\epsilon \nu_s^\epsilon) \vec{\varepsilon}_\epsilon + \sum_{\substack{u, v \in \overline{\mathcal{J}_2^n} \\ |u.v| > 0}} \nu_r^u \nu_s^v \vec{\varepsilon}_{u.v} + \downarrow_\circ (\nu_r^\epsilon \nu_s^\epsilon) \vec{\varepsilon}_{\ell_i}$$

Le point principal de la preuve consiste à montrer que pour tout $u \in \overline{\mathcal{J}_2^n}$, $\sum_{\tau(v)=u} \omega_t^v = \nu_t^u$.

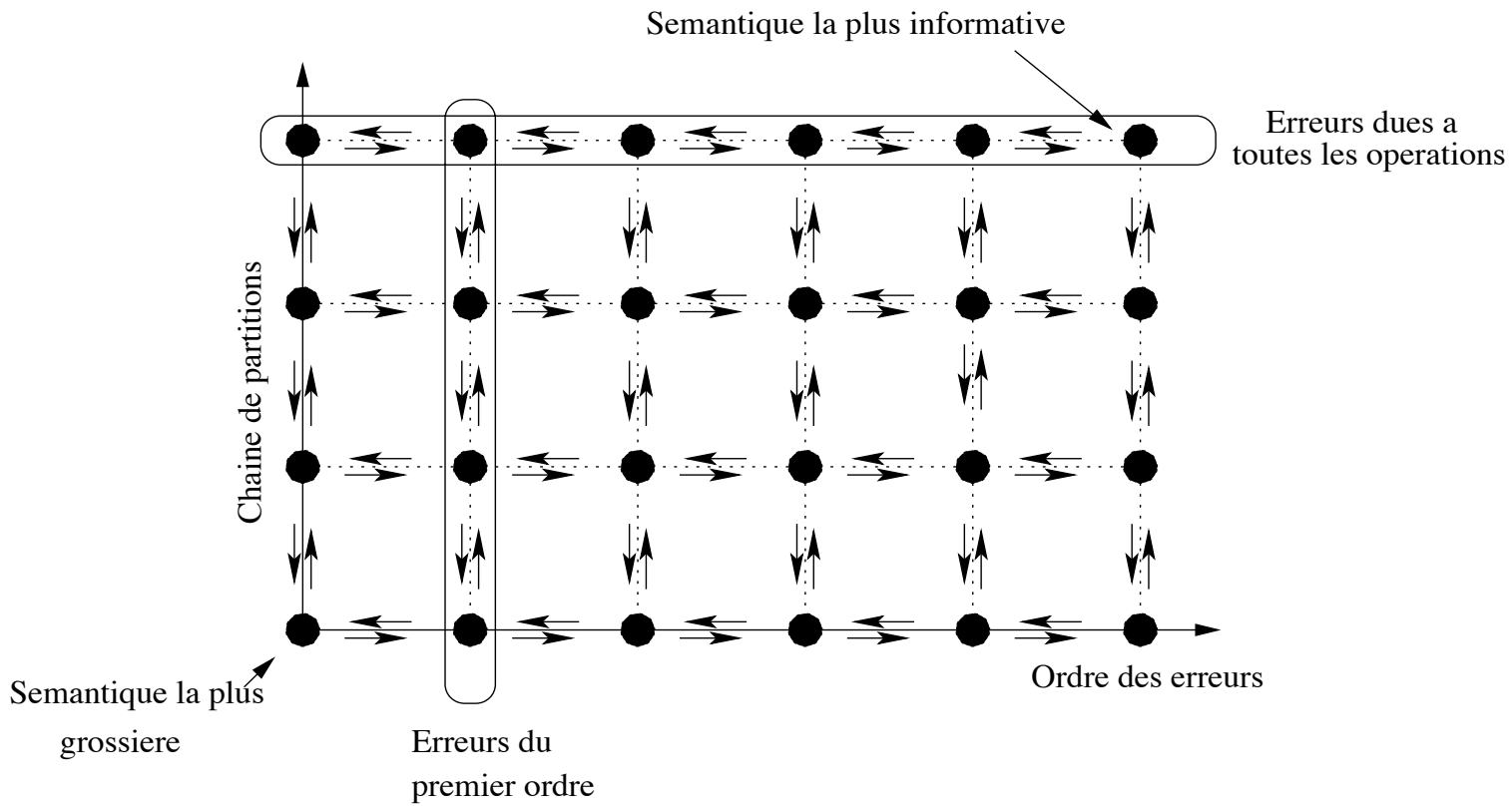
$$\sum_{\tau(v)=u} \omega_t^v = \sum_{\tau(v_1 . v_2)=u} \omega_r^{v_1} \omega_s^{v_2} = \sum_{\substack{\tau(v_1).\tau(v_2) = u_1.u_2 \\ u_1.u_2 = u}} \omega_r^{v_1} \omega_s^{v_2}$$

$$= \sum_{u_1 . u_2 = u} \left(\sum_{\substack{\tau(v_1) = u_1 \\ \tau(v_2) = u_2}} \omega_r^{v_1} \omega_s^{v_2} \right)$$

Relation Entre les Différentes Sémantiques

$$
\begin{array}{ccccccc}
[.]^{\mathcal{J}_0^*}(a_0^{\ell 0}) & \leftrightarrows & \dots & \underset{\alpha^{n+1,n}}{\overset{\gamma^{n,n+1}}{\longleftrightarrows}} & [.]^{\mathcal{J}_0^n}(a_0^{\ell 0}) & \underset{\alpha^{n,n-1}}{\overset{\gamma^{n-1,n}}{\longleftrightarrows}} & \dots & \leftrightarrows & [.]^{\mathcal{J}_0^0}(a_0^{\ell 0}) \\
\uparrow\downarrow & & \uparrow\downarrow & & {}_{\alpha}\mathcal{J}_1^n,\mathcal{J}_0^n \uparrow\downarrow {}_{\gamma}\mathcal{J}_0^n,\mathcal{J}_1^n & & \uparrow\downarrow & & \uparrow\downarrow \\
[.]^{\mathcal{J}_1^*}(a_0^{\ell 0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{J}_1^n}(a_0^{\ell 0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{J}_1^0}(a_0^{\ell 0}) \\
\uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow \\
\dots & \leftrightarrows & \dots & \leftrightarrows & \dots & \leftrightarrows & \dots & \leftrightarrows & \dots \\
\uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow \\
[.]^{\mathcal{L}^*}(a_0^{\ell 0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{L}^n}(a_0^{\ell 0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{L}^0}(a_0^{\ell 0})
\end{array}
$$

Relation Entre les Différentes Sémantiques



Partitionnement des points de contrôle

Analyse statique fondée sur $[\![.]\!]^{\mathcal{J}^n}$:

- intervalles de flottants pour le terme flottant
- intervalles multi-précision pour les termes d'erreurs
- choix d'une partition

Le choix d'une partition a une grande importance sur la précision de l'analyse

Abstraction par des intervalles

$$\langle \wp(\mathcal{F}(\mathbb{R}, \overline{\mathcal{I}^n})), \subseteq \rangle \quad \underset{\alpha^{\mathcal{I}}}{\overset{\gamma^{\mathcal{I}}}{\leftrightarrows}} \quad \langle \mathcal{F}(\mathcal{I}_{\mathbb{F}}, \overline{\mathcal{I}^n}), \sqsubseteq \rangle$$

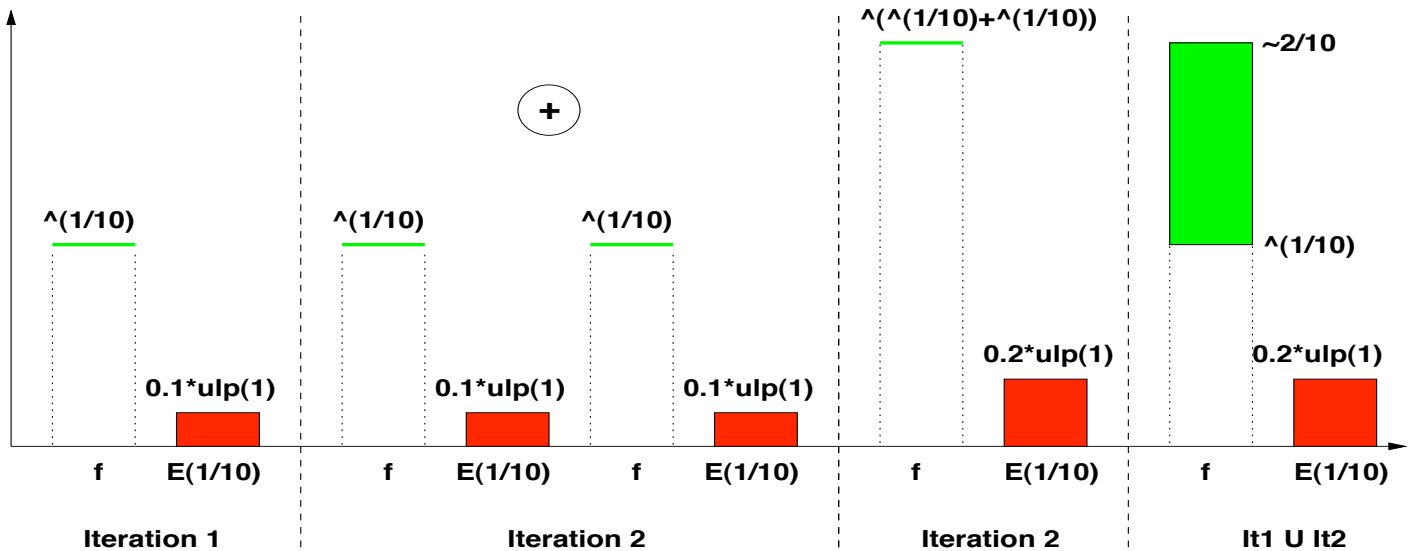
$$\sum_{u \in \overline{\mathcal{I}^n}} \omega_1^u \vec{\varepsilon}_u \sqsubseteq \sum_{u \in \overline{\mathcal{I}^n}} \omega_2^u \vec{\varepsilon}_u \; ; \iff ; \forall u \in \overline{\mathcal{I}^n}, \; \omega_1^u \subseteq \omega_2^u$$

$$\alpha^{\mathcal{I}} \left(\left\{ \sum_{u \in \overline{\mathcal{I}^n}} \omega_i^u \vec{\varepsilon}_u \; : \; i \in I \right\} \right) \overset{\text{def}}{=} \sum_{u \in \overline{\mathcal{I}^n}} \Phi \left(\{ \omega_i^u \; : \; i \in I \} \right) \vec{\varepsilon}_u$$

$$\gamma^{\mathcal{I}} \left(\sum_{u \in \overline{\mathcal{I}^n}} \nu^u \vec{\varepsilon}_u \right) \overset{\text{def}}{=} \left\{ \sum_{u \in \overline{\mathcal{I}^n}} \omega^u \vec{\varepsilon}_u \; : \; \forall u \in \overline{\mathcal{I}^n}, \; \omega^u \in \nu^u \right\}$$

Exemple

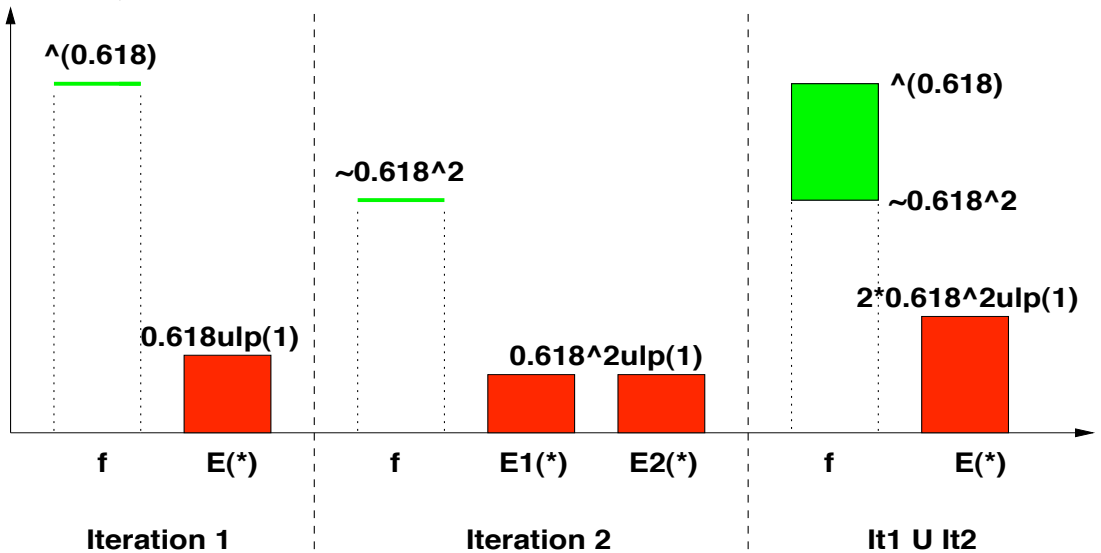
```
t = 0.0;
for (;;)
  t = t+1/10;
```



$t_1 < t_2$, par widening on obtient : $t = [\uparrow_\circ (1/10), +\infty]\vec{\varepsilon} + [-\infty, +\infty]\vec{\varepsilon}_{1/10}$

Autre Exemple

```
t = 1.0;
for (i=1; i<=20; i++)
  t = t*0.618;
```



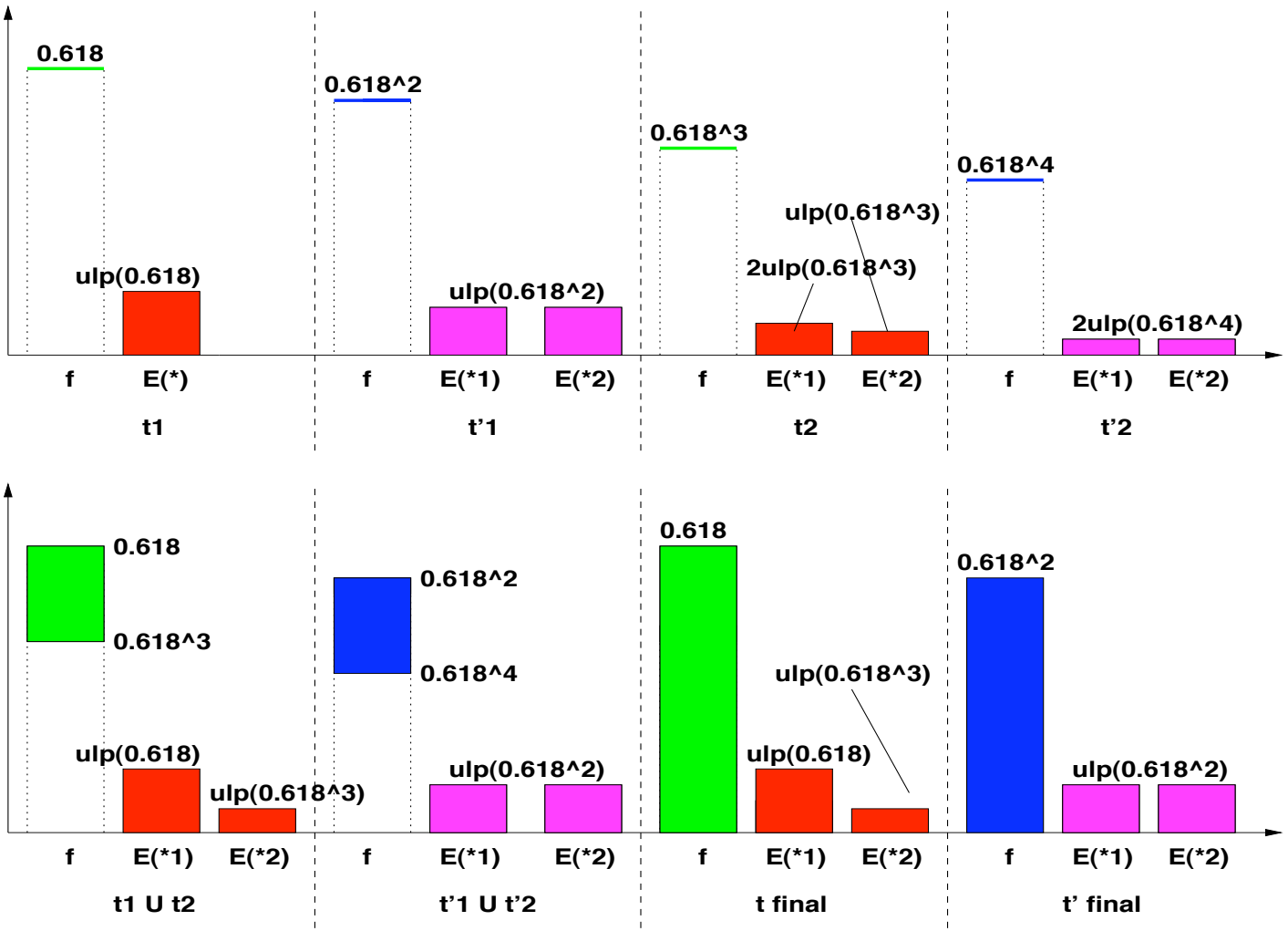
$0.618 < 2 \times 0.618^2$, l'erreur calculée augmente alors que l'erreur réelle diminue

Dépliage des boucles

- Une solution : réécrire le programme comme suit :

```
t = 1;
for (i=1 ; i<=20 ; i++)
{
  t = t*0.618; (multiplication 1)
  if (i>20) break;
  i++;
  t = t*0.618; (multiplication 2)
}
```

Dépliage des boucles

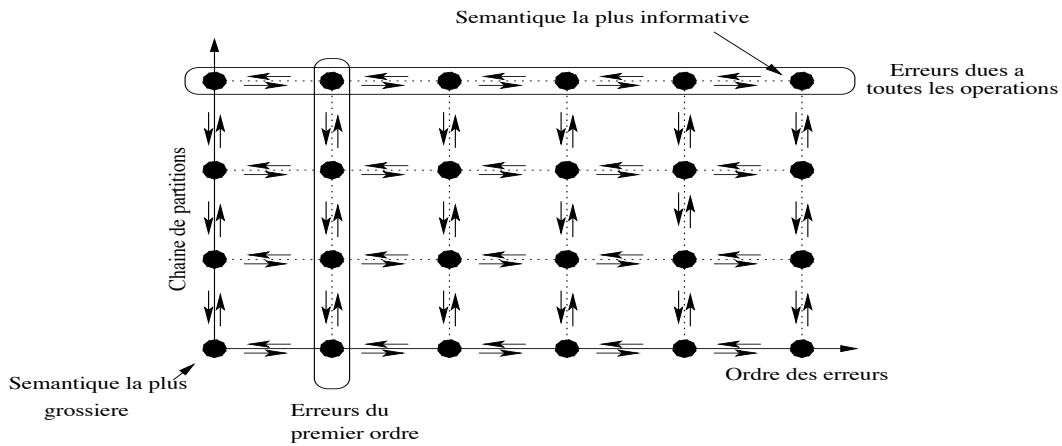


Cas des Boucles

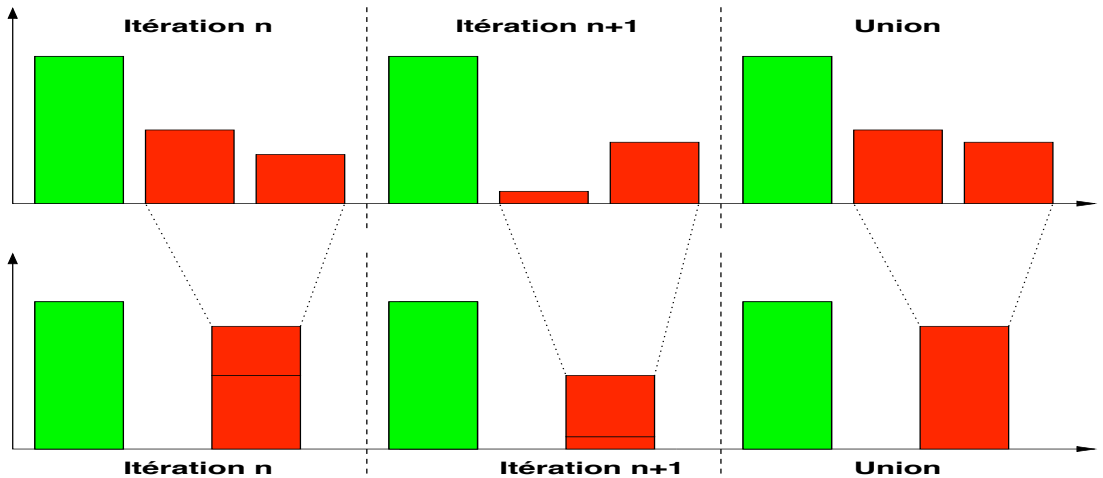
- L'analyse dépend de deux paramètres :
  - Nombre de fois où le corps d'une boucle est déplié
  - Nombre d'itérations du code déplié avant widening
- Pas de méthode connue pour estimer ces paramètres a priori
- Approches alternatives :
  - Partitionnement dynamique
  - Analyse spécifique : exposants de Lyapunov

Partitionnement Dynamique

- Principe :
  - Regrouper les erreurs dues à certains points de contrôle (pour limiter l'occupation mémoire)
  - En isoler certain autres (pour éviter les pertes de précision)
  - Faire ce choix dynamiquement, c.à.d. en cours d'analyse



Partitionnement Dynamique (2)

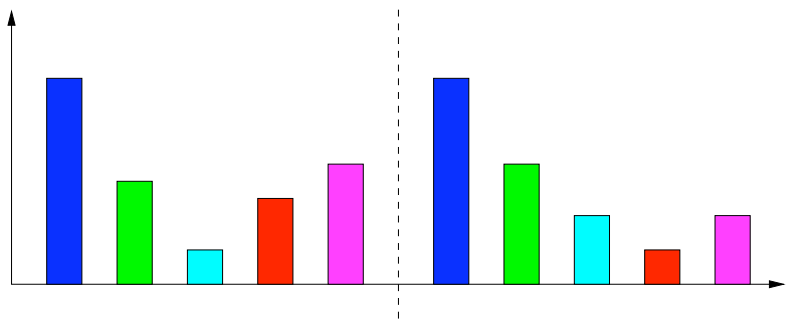


- Difficulté :
  - Des points regroupés ne peuvent ensuite être dissociés (correction de l'analyse)
  - Trouver la plus grande partition pour laquelle on peut affirmer que le calcul est stable

Partitionnement Dynamique : Complexité

Problème DP : Etant donnés deux tuples d'entiers $W = \langle \omega_1, \ldots, \omega_n \rangle$ et $W' = \langle \omega'_1, \ldots, \omega'_n \rangle$, existe-t-il une partition $\mathcal{P}$ de $\{1, \ldots, n\}$ de taille t telle que

$$\forall X \in \mathcal{P}, \sum_{i \in X} \omega_i \leq \sum_{i \in X} \omega'_i$$



Problème NP-Complet (preuve à partir de 2-Partition)

2-Partition : rappel

Etant donné un ensemble d'entiers positifs $A = \{a_1, \ldots, a_n\}$, existe-t-il un sous-ensemble I de A tel que :

$$\sum_{a_i \in I} a_i = \sum_{a_i \in A \setminus I} a_i$$

Preuve (NP-Complétude de DP)

- DP appartient à NP car vérification en temps polynomial.
- Soit $A = \{a_1, \ldots, a_n\}$ une instance I_1 de Partition. Construction à partir de I_1, de I_2 instance de DP

$$
\begin{aligned}
t &= 2 \\
\omega_i &= \left(\sum_{a_j \in A} a_j\right) + a_i,\ 1 \leq i \leq n \\
\omega'_i &= \left(\sum_{a_j \in A} a_j\right) - a_i,\ 1 \leq i \leq n \\
\omega_{n+1} = \omega_{n+2} &= 0 \\
\omega'_{n+1} = \omega'_{n+2} &= \sum_{a_j \in A} a_j
\end{aligned}
$$

Preuve (NP-Complétude de DP)

- On suppose connaître un algorithme A polynomial pour DP
- A trouve une solution à I_2 qui satisfait :

$$\forall X \in \mathcal{P}, \sum_{i \in X} \omega_i \leq \sum_{i \in X} \omega'_i \tag{4}$$

- Remarque 1 : puisque $t = 2$, $\mathcal{P} = \{X, \overline{X}\}$, où $\overline{X} = \{1, 2, \ldots, n+2\} \setminus X$.
- Remarque 2 : $n+1$ et $n+2$ ne sont pas dans la même classe car $\omega'_i < \omega_i$, $1 \leq i \leq n$. Si $n+1$ et $n+2$ sont dans X (resp. $\overline{X}$), alors (4) n'est pas respectée par $\overline{X}$ (resp. X).

Preuve (NP-Complétude de DP)

Nous avons pour X :

$$\begin{aligned}
\sum_X \omega_i &\leq \sum_X \omega'_i \\
|X| \sum_A a_i + \sum_X a_i &\leq |X| \sum_A a_i - \sum_X a_i + \omega'_{n+1} \\
\sum_X a_i &\leq \omega'_{n+1} - \sum_X a_i \\
\sum_X a_i &\leq \sum_A a_i - \sum_X a_i = \sum_{\overline{X}} a_i
\end{aligned}$$

Pour $\overline{X}$, on obtient par la même preuve :

$$\sum_{\overline{X}} a_i \leq \sum_X a_i$$

On en déduit l'égalité