Interprétation abstraite pour la précision numérique : de la détection à la correction d'erreurs

Matthieu Martel

matthieu.martel@univ-perp.fr

Introduction

Flottants utilisés à la place des réels pour les calculs sur ordinateurs

Arithmétiques très différentes :

- Opérations non associatives, non distributives, non inversibles
  - $\sqrt{2.0} = 1.414\ldots = x \quad x^2 = 1.9999\ldots$
  - dans les doubles :

$$x + 16.0 = 16.0 \text{ ssi } x \in [-8.881784197001252e^{-16}, 1.776356839400250 46e^{-15}]$$

- Nombre fini de décimales
  - Exceptions : overflow, underflow, NaN.
    `if (x<>0) y = 1/(x*x)`
  - Pertes de précision : erreurs “subjectives” ne déclenchent pas d’exception/interruption

Bugs connus

- Missile Patriot (1991) : $\frac{1}{10} = \frac{1}{2^4} + \frac{1}{2^5} + \frac{1}{2^8} + \ldots = 0.0001100\underline{1100}$. Codé sur 24 bits : erreur de $9.5 \cdot 10^{-8}$. Au bout de 100 heures, $100 \times 3600 \times 10 \times 9.5 \cdot 10^{-8} = 0.34$ secondes. Vitesse du Scud : $1676 ms^{-1}$. Erreur > 500 mètres.
- Autres exemples : bourse de Vancouver, élections allemandes, plateforme pétrolière
- Flottants de + en + utilisés dans des systèmes embarqués critiques (avions, voitures, centrales nucléaires, etc.)
- Besoin de techniques formelles de validation $\neq$ méthodes classiques des numériciens (tests/comparaison à des résultats obtenus analytiquement ou expérimentalement)

Plan

1. Généralités sur les nombres flottants
2. Détection : sémantique des séries d'erreurs
3. Correction : transformation sémantique d'expressions

Généralités sur les flottants

1. Norme IEEE 754
2. Mesure des erreurs

La norme IEEE 754

$$\begin{aligned} f &= \pm d_0.d_1 d_2 \ldots d_{p-1} \beta^e \\ &= \pm \left(d_0 + d_1 \beta^{-1} + \ldots d_{p-1} \beta^{-(p-1)}\right) \beta^e \end{aligned} \tag{1}$$

- β : base, e : exposant ($e_{min} \leq e \leq e_{max}$), p : précision, $d_0.d_1 \ldots d_{p-1}$ mantisse avec $0 \leq d_i < \beta$
- Nombres *flottants* : nombres représentables par (1)
- Nombres *normalisés* : nombres flottants t.q. $d_0 \neq 0$ $\Rightarrow$ unicité
- Nombres non-représentables :
  - Hors des bornes : $x < \beta^{e_{min}}$ ou $x > \beta^{e_{max}}$
  - Précision : π, 0.1 en base 2

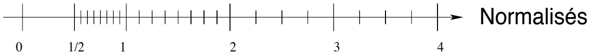
Norme IEEE 754 : formats

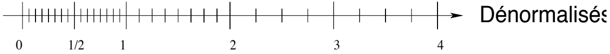
Type	Mantisse (bits)	e_{max}	e_{min}	Exposant (bits)
Single	23+1	+127	-126	8
Single Extended	≥ 32	$\geq$+1023	$\leq$-1022	$\geq$11
Double	52+1	+1023	-1022	11
Double Extended	$\geq$64	$>$16383	$\leq$-16382	$\geq$15

- Rayon d'un proton : $1.2 \times 10^{-15} m$
- Masse d'un electron : $9.1 \times 10^{-31} kg$
- Masse de la voie lactée : $2.2 \times 10^{41} kg$
- Age de l'univers : $4.17 \times 10^{17} s$

Valeurs particulières

Valeur	Signification	Exposant	Mantisse
$\pm\infty$	Dépacement	$e_{max}+1$	0
NaN	Not a Number	e_{max}	$\neq 0$
± 0	Zéro signé	$-e_{max}$	0
$0.dd\ldots$	Dénormalisés	$-e_{max}$	$\neq 0$





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

Opérations élémentaires sur les flottants

- Entre valeurs particulières :

$$-\infty \times -\infty = +\infty, \quad \pm 0 \div \pm\infty = \pm 0, \quad \pm\infty \div \pm\infty = \mathit{NaN}, \ \mathit{etc.}$$

- Pour les valeurs "classiques" → garantie de l'arrondi
- 4 modes d'arrondi : vers 0, vers $+\infty$, au plus près, vers $-\infty$

$$\uparrow_\circ : \mathbb{R} \to \mathbb{F}$$

$$\text{Pour } \Diamond \in \{+, -, \times, \div, \sqrt{\ }\}, \ x \Diamond_{\mathbb{F}} y = \uparrow_\circ (x \Diamond_{\mathbb{R}} y)$$

- Obtenu en utilisant des bits de garde (bits supplémentaires)
- Donne une sémantique précise aux opérations sur les flottants

Mesure des erreurs

- Erreurs déclenchant une exception : Overflow Underflow NaN
- Pertes de précision (ne déclenchent pas d'exception) :
  - Absorption $x + y = x$ si $x \gg y$ (cf. $x + 16 = 16$)
  - Branchement / Test instable
  - Elimination catastrophique (cancellation), $x - y$ avec $x \approx y$

$$A = \sqrt{s(s-a)(s-b)(s-c)} \quad \text{avec } s = \frac{a+b+c}{2}$$

Si le triangle est plat, $a \approx b + c$ et $s \approx a$

Si $a = 9.00$, $b = c = 4.53$:

$s_{\mathbb{R}}$	$A_{\mathbb{R}}$	$s_{\mathbb{F}}$	$A_{\mathbb{F}}$	Erreur
9.03	2.34	9.04	2.71	15%

Ulp, erreurs relatives

- *ulp* : **U**nit in the **L**ast **P**lace, ordre de grandeur du plus petit chiffre significatif d'un nombre
- <u>ex :</u> si $p = 3$, $\beta = 10$, $x = 1.23 \cdot 10^4$, $ulp(x) = 1 \cdot 10^2$
- Erreur relative : $e_r = \left| \frac{r_{\text{exact}} - r_{\text{approché}}}{r_{\text{exact}}} \right|$
- Elimination catastrophique $\Rightarrow$ erreur relative importante
- aire du triangle : dans $\mathbb{R}$, $9.03 - 9.0 = 0.03$; dans $\mathbb{F}$, $9.04 - 9.0 = 0.04$; $e_r = \left| \frac{0.03 - 0.04}{0.03} \right| = 33\%$

ϵ-machine, Erreurs d'ordre supérieur

- ϵ-machine : maximum de l'erreur relative due à un arrondi au voisinage d'un point x.

$$\epsilon = ulp(1) \qquad x = 1.23 \cdot 10^4,\ ulp(x) = 1 \cdot 10^2 = 10^4\epsilon$$

- Erreurs d'ordre supérieur :

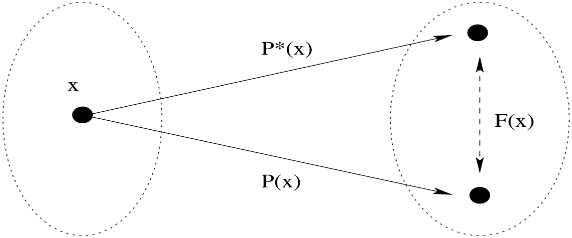
$$x_1 = f_1 + \epsilon_1,\ x_2 = f_2 + \epsilon_2, \quad x_1 \times x_2 = \underbrace{f_1 f_2}_{resultat} + \underbrace{f_1\epsilon_2 + f_2\epsilon_1}_{1er\ ordre} + \underbrace{\epsilon_1\epsilon_2}_{2d\ ordre}$$

- plus généralement, des erreurs d'ordre n peuvent apparaître au cours d'un calcul.
- les erreurs d'ordre supérieur sont généralement négligeables par rapport à celles d'ordre 1 mais $\exists$ des exceptions

Erreur en avant

$F(x) = \text{dist}(p(x), p^*(x))$

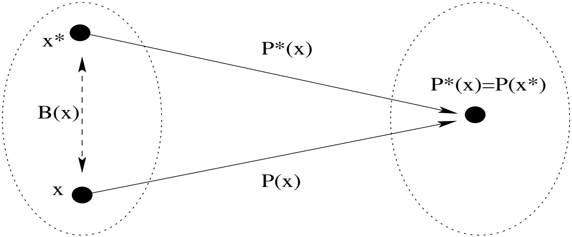
$$\begin{cases} p & : \text{ calcul exact} \\ p^* & : \text{ calcul approché} \end{cases}$$



Estime la précision de la solution en fonction de la précision d'une entrée particulière

Erreur en arrière

$$B(x) = \text{Inf } \left\{ \text{dist}(x, x^*) \ : \ x^* = p^{-1}(p^*(x)) \right\}$$



Détermine si une solution approchée est égale à la solution exacte d'un problème voisin

Introduction

(i) Définir une sémantique concrète pour les nombres flottants

Fondée sur la Norme IEEE 754

Valeur = flottant + somme d'erreurs commises durant le calcul

(ii) Définir une analyse statique (par interprétation abstraite) à partir de la sémantique précédente

Majore les erreurs pour une classe d'exécutions

Valeur abstraite = intervalle pour le flottant + somme d'intervalles d'erreurs

Donne une estimation de la précision des calculs

Permet de comprendre la provenance des erreurs

Utilisation de la Norme IEEE 754

Arrondi des nombres réels (vers 0, $-\infty$, $+\infty$ et au plus près) :

$$\uparrow_{\circ}: \mathbb{R} \rightarrow \mathbb{F}$$

Résultats des opérations : pour $\lozenge \in \{+, -, \times, \div, \sqrt{}\}$ on a :

$$x \lozenge_{\mathbb{F}} y = \uparrow_{\circ} (x \lozenge_{\mathbb{R}} y)$$

Erreur d'arrondi :

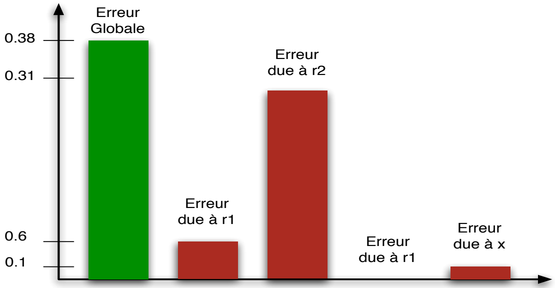
$$\downarrow_{\circ} : \ \mathbb{R} \rightarrow \mathbb{R} \qquad\qquad \downarrow_{\circ} (r) = r - \uparrow_{\circ} (r)$$

Exemple introductif

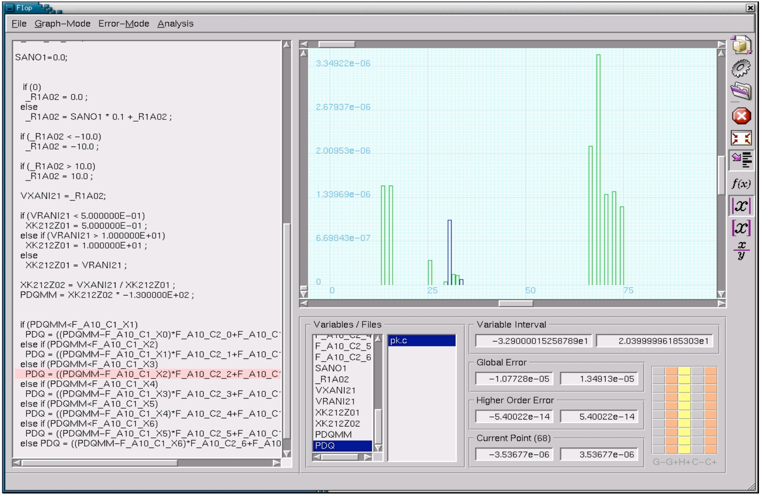
$$
\begin{array}{rrclcl}
 & 621.3\vec{\varepsilon} & + & 0.05\vec{\varepsilon}_{\ell_1} & & \color{darkred}{r_1^{\ell_1}} \\
\times^{\ell_3} & 1.287\vec{\varepsilon} & + & 0.0005\vec{\varepsilon}_{\ell_2} & & \color{darkred}{r_2^{\ell_2}} \\
\hline
= & 799.6131\vec{\varepsilon}\cdot\vec{\varepsilon} & & & & \color{darkred}{\text{Résultat}} \\
 & & + & 0.06435\vec{\varepsilon}\cdot\vec{\varepsilon}_{\ell_1} & & \color{darkred}{\text{Erreur due à } r_1^{\ell_1}} \\
 & & + & 0.31065\vec{\varepsilon}\cdot\vec{\varepsilon}_{\ell_2} & & \color{darkred}{\text{Erreur due à } r_2^{\ell_2}} \\
 & & + & 0.000025\vec{\varepsilon}_{\ell_1}\cdot\vec{\varepsilon}_{\ell_2} & & \color{darkred}{\text{Terme du second ordre}} \\
\hline
= & 799.6\vec{\varepsilon} & & & & \color{darkred}{\text{Résultat machine } =\uparrow_\circ (r_1 \times r_2)} \\
 & & + & 0.06435\vec{\varepsilon}_{\ell_1} & & \color{darkred}{\text{Erreur due à } r_1^{\ell_1}} \\
 & & + & 0.31065\vec{\varepsilon}_{\ell_2} & & \color{darkred}{\text{Erreur due à } r_2^{\ell_2}} \\
 & & + & 0.000025\vec{\varepsilon}_{\ell_1\ell_2} & & \color{darkred}{\text{Terme du second ordre}} \\
 & & + & 0.0131\vec{\varepsilon}_{\ell_3} & & \color{darkred}{\text{Erreur due à } \times^{\ell_3} =\downarrow_\circ (r_1 \times r_2)}
\end{array}
$$

Exemple Introductif (suite)

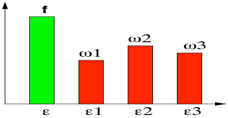
$$799.988125 = 799.6\vec{\varepsilon} + 0.06435\vec{\varepsilon}_{\ell_1} + 0.31065\vec{\varepsilon}_{\ell_2} + 0.000025\vec{\varepsilon}_{\ell_1\ell_2} + 0.0131\vec{\varepsilon}_{\ell_3}$$



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
Variables / Files
F_A10_C2_5
F_A10_C2_6
SANO1
_R1A02
VXANI21
VRANI21
XK212Z01
XK212Z02
PDQMM
PDQ
pk.c
Variable Interval
-3.29000015258789e1
2.03999996185303e1
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+

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



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

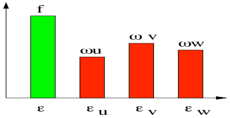
<u>Exemple :</u>

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

- 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)

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

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

	$621.3\vec{\varepsilon}$	$+$	$0.05\vec{\varepsilon}_{\ell_1}$	$r_1^{\ell_1}$
$+^{\ell_3}$	$1.287\vec{\varepsilon}$	$+$	$0.0005\vec{\varepsilon}_{\ell_2}$	$r_2^{\ell_2}$
$=$	$622.5\vec{\varepsilon}$			Résultat machine $= \uparrow_\circ (r_1 + r_2)$
		$+$	$0.05\vec{\varepsilon}_{\ell_1}$	Erreur due à $r_1^{\ell_1}$
		$+$	$0.0005\vec{\varepsilon}_{\ell_2}$	Erreur due à $r_2^{\ell_2}$
		$+$	$0.087\vec{\varepsilon}_{\ell_3}$	Erreur due à $+^{\ell_3} = \downarrow_\circ (r_1 + r_2)$

Opérations Elémentaires (Multiplication)

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

	$621.3\vec{\varepsilon}$	$+$	$0.05\vec{\varepsilon}_{\ell_1}$	$r_1^{\ell_1}$
$\times^{\ell_3}$	$1.287\vec{\varepsilon}$	$+$	$0.0005\vec{\varepsilon}_{\ell_2}$	$r_2^{\ell_2}$
$=$	$799.6\vec{\varepsilon}$			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 $\times^{\ell_3}$ $=\downarrow_\circ (r_1 \times r_2)$

Cas de la division

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

Nous avons :

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

et :

$$[\![\frac{1}{f\vec{\varepsilon}+e\vec{\varepsilon}_\ell}]\!]^{\mathcal{L}^*} = \uparrow_\circ \left(\frac{1}{f}\right)\vec{\varepsilon}_f + \left[\downarrow_\circ \left(\frac{1}{f}\right) + \sum_{n\geq 1}(-1)^n \frac{e^n}{f^{n+1}}\right]\vec{\varepsilon}_\ell$$

Valable pour $-1 < \frac{e}{f} < 1$ ou pour $|e| \leq |f|$, c.à.d. tant que l'erreur est inférieure au flottant.

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

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:**

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

Correction (2)

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

- 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

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

Propriété

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

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,

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

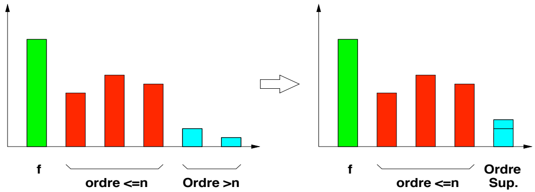
$$\sum_{u \in \mathcal{W}} \omega_1^u \vec{\varepsilon}_u \times \frac{\partial}{\partial \omega_1^{u_0}} \left(\sum_{v \in \mathcal{W}} \omega_2^v \vec{\varepsilon}_v \right) + \sum_{v \in \mathcal{W}} \omega_2^v \vec{\varepsilon}_v \times \frac{\partial}{\partial \omega_1^{u_0}} \left(\sum_{u \in \mathcal{W}}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

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

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 :

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

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

$$\begin{aligned} t^{\mathcal{L}^m} &= r^{\mathcal{L}^m} \times^{\ell_i} s^{\mathcal{L}^m} \\ &= \uparrow_\circ (\nu_r^\epsilon \nu_s^\epsilon)\vec{\varepsilon}_\epsilon + \sum_{\substack{u, v \in \overline{\mathcal{L}^m} \\ |u.v| > 0}} \nu_r^u \nu_s^v \vec{\varepsilon}_{u.v} + \downarrow_\circ (\nu_r^\epsilon \nu_s^\epsilon)\vec{\varepsilon}_{\ell_i} \end{aligned} \tag{3}$$

Preuve (multiplication)

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

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

$$\left\{ \sum_{u \in \overline{\mathcal{L}^n}} \omega^u \vec{\varepsilon}_u \ : \ \left| \ \begin{array}{l} \omega^u = \nu_t^u \text{ if } u \in \overline{\mathcal{L}^m} \setminus \{hi\} \\ \sum_{u \in (\overline{\mathcal{L}^n} \setminus \overline{\mathcal{L}^m}) \cup \{hi\}} \omega^u = \nu_t^{hi} \end{array} \right. \right\} \tag{4}$$

où, dans (4),

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

et

$$\nu_t^{hi} = \sum_{u_1 u_2 = u \in (\overline{\mathcal{L}^n} \setminus \overline{\mathcal{L}^m}) \cup \{hi\}} \nu_r^{u_1} \nu_s^{u_2}$$

Nous devons montrer que $\sum_{u \in (\overline{\mathcal{L}^n} \setminus \overline{\mathcal{L}^m}) \cup \{hi\}} \omega_t^u = \nu_t^{hi}$. Notation : $M = \overline{\mathcal{L}^m} \setminus \{hi\}$ et $N = (\overline{\mathcal{L}^n} \setminus \overline{\mathcal{L}^m}) \cup \{hi\}$. On a :

$$\begin{aligned}
\sum_{u \in N} \omega_t^u &= \sum_{\substack{u \in \overline{\mathcal{L}^n},\, v \in \overline{\mathcal{L}^n} \\ u.v \in N}} \omega_r^u \omega_s^v \\
&= \sum_{\substack{u, v \in M \\ u.v \in N}} \omega_r^u \omega_s^v + \sum_{u, v \in N} \omega_r^u \omega_s^v \\
&+ \sum_{\substack{u \in M,\, v \in N \\ u.v \in N}} \omega_r^u \omega_s^v + \sum_{\substack{u \in N,\, v \in M \\ u.v \in N}} \omega_r^u \omega_s^v
\end{aligned}$$

Preuve (multiplication)

Puisque $r^{\mathcal{L}^m} = \alpha^{n,m}(r^{\mathcal{L}^n})$ et $s^{\mathcal{L}^m} = \alpha^{n,m}(s^{\mathcal{L}^n})$, $\sum_{u \in N} \omega_r^u = \nu_r^{hi}$ et $\sum_{u \in N} \omega_s^u = \nu_s^{hi}$. Donc,

$$\sum_{u \in N} \omega_t^u = \sum_{\substack{u, v \in M \\ u.v \in N}} \omega_r^u \omega_s^v + \nu_r^{hi} \nu_s^{hi} + \sum_{u \in M} \omega_r^u \nu_s^{hi} + \sum_{v \in M} \nu_r^{hi} \omega_s^v$$

Puisque $r^{\mathcal{L}^m} = \alpha^{n,m}(r^{\mathcal{L}^n})$ et $s^{\mathcal{L}^m} = \alpha^{n,m}(s^{\mathcal{L}^n})$, pour tout mot u tel que $|u| \leq m$, nous avons $\omega_r^u = \nu_r^u$ et $\omega_s^u = \nu_s^u$, donc

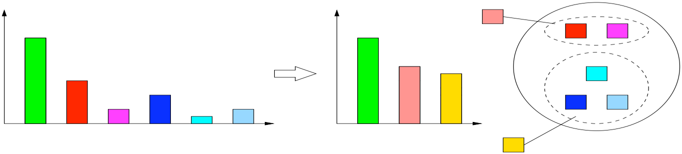
$$\begin{aligned}
\sum_{u \in N} \omega_t^u &= \sum_{\substack{u, v \in M \\ u.v \in N}} \nu_r^u \nu_s^v + \nu_r^{hi} \nu_s^{hi} + \sum_{u \in M} \nu_r^u \nu_s^{hi} + \sum_{v \in M} \nu_r^{hi} \nu_s^v \\
&= \sum_{\substack{u, v \in \overline{\mathcal{L}^m} \\ u.v \in N}} \nu_r^u \nu_s^v = \nu_t^{hi}
\end{aligned}$$

Grain d'Erreur : $[\![.]\!]^{\mathcal{J}^n}$

- 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:

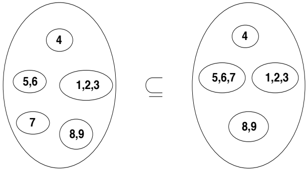


- Pour les termes d'ordre supérieur : $\omega^{J_1 J_2} = \sum_{\ell_1 \in J_1, \ell_2 \in J_2} \omega^{\ell_1} \omega^{\ell_2}$
- Sémantique des opérations : $\mathcal{W} = \overline{\mathcal{J}^n}$

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

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

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

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

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

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

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 $u \in \overline{\mathcal{J}_2^n}$, $\sum_{\tau(v)=u} \omega_t^v = \nu_t^u$.

Preuve (multiplication)

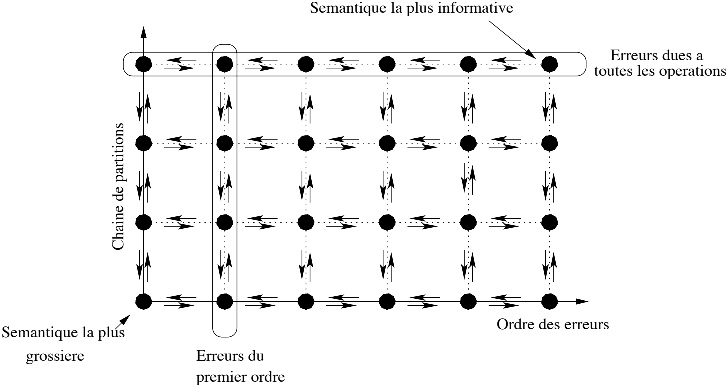
Le point principal de la preuve consiste à montrer que pour tout $u \in \overline{\mathcal{J}_2^n}$, $\sum_{\tau(v)=u} \omega_t^v = \nu_t^u$.

$$\begin{aligned}
\sum_{\tau(v)=u} \omega_t^v &= \sum_{\tau(v_1 . v_2)=u} \omega_r^{v_1} \omega_s^{v_2} = \sum_{\substack{\tau(v_1).\tau(v_2) = u_1.u_2 \\ u_1.u_2 = u}} \omega_r^{v_1} \omega_s^{v_2} \\
&= \sum_{u_1 . u_2 = u} \left(\sum_{\substack{\tau(v_1) = u_1 \\ \tau(v_2) = u_2}} \omega_r^{v_1} \omega_s^{v_2} \right) \\
&= \sum_{u_1 . u_2 = u} \left(\sum_{\tau(v_1)=u_1} \omega_r^{v_1} \times \sum_{\tau(v_2)=u_2} \omega_s^{v_2} \right) = \sum_{u_1 . u_2 = u} \nu_r^{u_1} \nu_s^{u_2} = \nu_t^u
\end{aligned}$$

Relation Entre les Différentes Sémantiques

$$\begin{array}{ccccccccc}
[.]^{\mathcal{J}_0^*}(a_0^{\ell_0}) & \leftrightarrows & \dots & \underset{\alpha^{n+1,n}}{\overset{\gamma^{n,n+1}}{\leftrightarrows}} & [.]^{\mathcal{J}_0^n}(a_0^{\ell_0}) & \underset{\alpha^{n,n-1}}{\overset{\gamma^{n-1,n}}{\leftrightarrows}} & \dots & \leftrightarrows & [.]^{\mathcal{J}_0^0}(a_0^{\ell_0}) \\
\uparrow\downarrow & & \uparrow\downarrow & & {}_{\alpha^{\mathcal{J}_1^n,\mathcal{J}_0^n}}\uparrow\downarrow_{\gamma^{\mathcal{J}_0^n,\mathcal{J}_1^n}} & & \uparrow\downarrow & & \uparrow\downarrow \\
[.]^{\mathcal{J}_1^*}(a_0^{\ell_0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{J}_1^n}(a_0^{\ell_0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{J}_1^0}(a_0^{\ell_0}) \\
\uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow \\
\dots & \leftrightarrows & \dots & \leftrightarrows & \dots & \leftrightarrows & \dots & \leftrightarrows & \dots \\
\uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow & & \uparrow\downarrow \\
[.]^{\mathcal{L}^*}(a_0^{\ell_0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{L}^n}(a_0^{\ell_0}) & \leftrightarrows & \dots & \leftrightarrows & [.]^{\mathcal{L}^0}(a_0^{\ell_0})
\end{array}$$

Relation Entre les Différentes Sémantiques



Partitionnement des points de contrôle

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

- 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

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

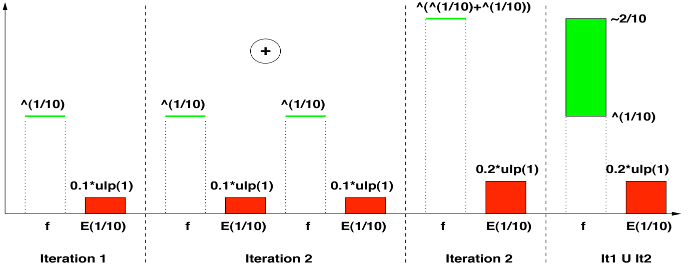
$$\sum_{u \in \overline{\mathcal{J}^n}} \omega_1^u \vec{\varepsilon}_u \sqsubseteq \sum_{u \in \overline{\mathcal{J}^n}} \omega_2^u \vec{\varepsilon}_u \; ; \iff ; \; \forall u \in \overline{\mathcal{J}^n}, \; \omega_1^u \subseteq \omega_2^u$$

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

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

Exemple

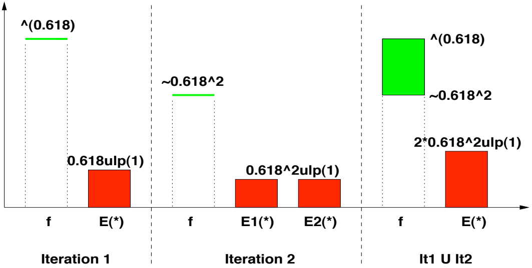
```
t = 0.0;
for (;;)
  t = t+1/10;
```



$t_1 < t_2$, 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;
```



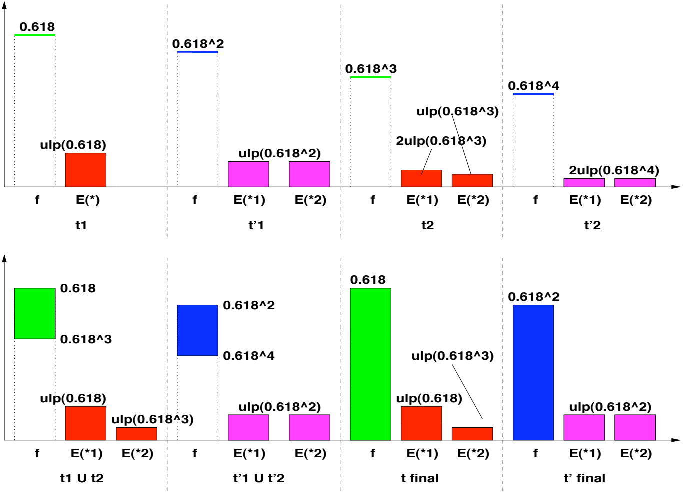
$0.618 < 2 \times 0.618^2$, l'erreur calculée augmente alors que l'erreur réelle diminue

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

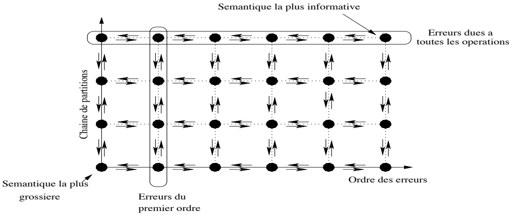


Cas des Boucles

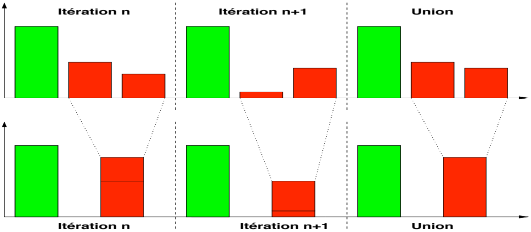
- L'analyse dépend de deux paramètres :
  - Nombre de fois où le corps d'une boucle est déplié
  - Nombre d'itérations du code déplié avant widening
- Pas de méthode connue pour estimer ces paramètres a priori
- Approches alternatives :
  - Partitionnement dynamique
  - Analyse spécifique : exposants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

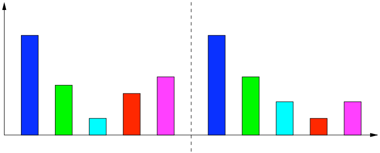


- Difficulté :
  - Des points regroupés ne peuvent ensuite être dissociés (correction de l'analyse)
  - Trouver la plus grande partition pour laquelle on peut affirmer que le calcul est stable

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 :

$$\forall X \in \mathcal{P},\ \sum_{i \in X} \omega_i \leq \sum_{i \in X} \omega'_i \tag{5}$$

- 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 (5) 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

Amélioration de la précision

Principe

Transformation sémantique pour améliorer la précision de l'évaluation des expressions arithmétiques

Modèle

$x^2+2x+1 = (x+1)^2 = \dots$

Implémentation

`((x*x)+(2*x))+1`

$\neq$ `(x+1)*(x+1)` $\neq \dots$

- **Première sémantique :** une expression décrit une formule mathématique usuelle
  - l'évaluation en précision ∞ donne résultat exact
  - lois algébriques autorisées (associativité, commutativité, etc.)
- **Seconde sémantique :**
  - les expressions suivent l'arithmétique des nombres flottants
  - erreurs d'arrondi, lois algébriques non-valides
- **But :** transformer une expression e en une expression e' t.q. :
  - e et e' sont égales dans l'arithmétique usuelle
  - e' s'évalue plus précisément que e sur ordinateur

Motivation - approche

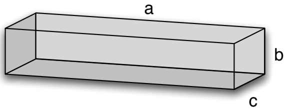
Motivation :

- erreurs dues aux flottants difficiles à comprendre et corriger
  - arithmétique non-intuitive, déboguage ardu
  - pas de méthodologie pour améliorer la précision d'un calcul
  - quelques “recettes de cuisine” (schéma de Horner, tri, etc.)
- séries d'erreurs :
  - permettent d'identifier les sources d'erreurs
  - n'offrent pas d'aide directe à la correction

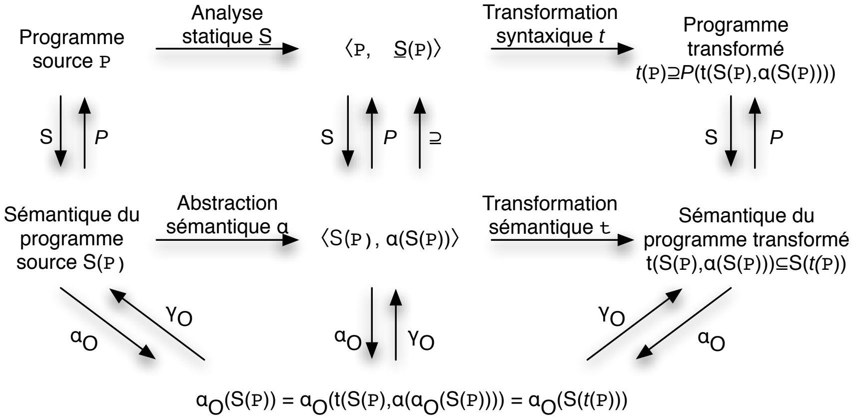
Approche :

- transformation sémantique [P. and R. Cousot, POPL'02]
- exemple: aire d'un parallélépipède rectangle allongé :

$$A = 2 \times \big((a \times b) + (b \times c) + (c \times a)\big)$$



  - implémentation directe : `2*(((a*b)+(b*c))+(c*a))`
  - transformé en : `(((c*a)*2)+(2*(b*c)))+(2*(a*b))`
  - plus petites valeurs sommées d'abord, produit distribué


Programme source P
Analyse statique S
⟨P, S(P)⟩
Transformation syntaxique t
Programme transformé t(P)⊇P(t(S(P),α(S(P))))
S
P
S
P
⊇
S
P
Sémantique du programme source S(P)
Abstraction sémantique α
⟨S(P), α(S(P))⟩
Transformation sémantique t
Sémantique du programme transformé t(S(P),α(S(P)))⊆S(t(P))
γO
αO
αO
γO
γO
αO
αO(S(P)) = αO(t(S(P),α(αO(S(P)))) = αO(S(t(P)))

Sémantique concrète idéale

Sémantique fondée sur l'arithmétique des réels : $\rightarrow_{\mathbb{R}}$

$$\frac{v = v_1 +_{\mathbb{R}} v_2}{v_1 + v_2 \rightarrow v}$$

$$\frac{v = v_1 \times_{\mathbb{R}} v_2}{v_1 \times v_2 \rightarrow v}$$

$$\frac{e_1 \rightarrow e_1'}{e_1 + e_2 \rightarrow e_1' + e_2}$$

$$\frac{e_1 \rightarrow e_1'}{e_1 \times e_2 \rightarrow e_1' \times e_2}$$

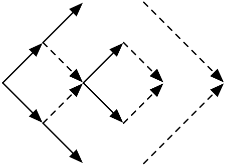
$$\frac{e \equiv e_1 \quad e_1 \rightarrow e_1' \quad e_1' \equiv e'}{e \rightarrow e'}$$

Modèle
$x^2 + 2x + 1 = (x+1)^2 = \ldots$

Implémentation
`((x*x)+(2*x))+1`
$\neq$ `(x+1)*(x+1)` $\neq \ldots$

- $(e_1 + e_2) + e_3 \equiv e_1 + (e_2 + e_3)$
- $e_1 + e_2 \equiv e_2 + e_1$
- $e \equiv e + 0$
- $(e_1 \times e_2) \times e_3 \equiv e_1 \times (e_2 \times e_3)$
- $e_1 \times e_2 \equiv e_2 \times e_1$
- $e \equiv e \times 1$
- $e_1 \times (e_2 + e_3) \equiv e_1 \times e_2 + e_1 \times e_3$

Propriété (confluence faible) Soit e une expression arithmétique. Si $e \rightarrow_{\mathbb{R}} e_1$ et $e \rightarrow_{\mathbb{R}} e_2$ alors il existe e' t.q. $e_1 \rightarrow_{\mathbb{R}}^{*} e'$ et $e_2 \rightarrow_{\mathbb{R}}^{*} e'$



Sémantique pour l'estimation des erreurs

La Norme IEEE754

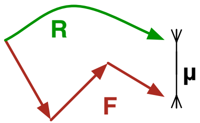
- $\uparrow_\circ : \ \mathbb{R} \to \mathbb{F}$ spécifie l'arrondi d'un nombre réel
- opérations élémentaires $\Diamond \in \{+, -, \times, \div, \sqrt{}\}$:

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

- $\downarrow_\circ : \mathbb{R} \to \mathbb{R}$ donne l'erreur d'arrondi : $\downarrow_\circ (r) = r - \uparrow_\circ (r)$

Estimation des erreurs : l'arithmétique $\mathbb{E}$

- $\mathbb{E} = \mathbb{F} \times \mathbb{R}$. Dans $(x, \mu) \in \mathbb{E}$, μ mesure la distance entre le résultat machine x et le résultat exact
- pour $v_1 = (x_1, \mu_1)$ et $v_2 = (x_2, \mu_2)$:



$$v_1 +_{\mathbb{E}} v_2 = (\uparrow_\circ (x_1 +_{\mathbb{R}} x_2), [\mu_1 + \mu_2 + \downarrow_\circ (x_1 +_{\mathbb{R}} x_2)])$$

$$v_1 \times_{\mathbb{E}} v_2 = (\uparrow_\circ (x_1 \times_{\mathbb{R}} x_2), [\mu_1 x_2 +_{\mathbb{R}} \mu_2 x_1 +_{\mathbb{R}} \mu_1 \mu_2 +_{\mathbb{R}} \downarrow_\circ (x_1 \times_{\mathbb{R}} x_2)])$$

Sémantique concrète de l'implémentation

La sémantique $\to_{\mathbb{E}}$

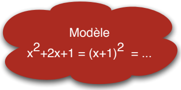
$$\frac{v = v_1 +_{\mathbb{E}} v_2}{v_1 + v_2 \;\to\; v}$$

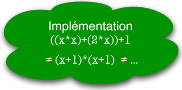
$$\frac{v = v_1 \times_{\mathbb{E}} v_2}{v_1 \times v_2 \;\to\; v}$$

$$\frac{e_1 \;\to\; e_1'}{e_1 + e_2 \;\to\; e_1' + e_2}$$

$$\frac{e_1 \;\to\; e_1'}{e_1 \times e_2 \;\to\; e_1' \times e_2}$$

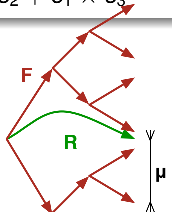
$$\frac{e \equiv e_1 \quad e_1 \;\to\; e_1' \quad e_1' \equiv e'}{e \;\to\; e'}$$





- $(e_1 + e_2) + e_3 \equiv e_1 + (e_2 + e_3)$
- $e_1 + e_2 \equiv e_2 + e_1$
- $e \equiv e + 0$
- $(e_1 \times e_2) \times e_3 \equiv e_1 \times (e_2 \times e_3)$
- $e_1 \times e_2 \equiv e_2 \times e_1$
- $e \equiv e \times 1$
- $e_1 \times (e_2 + e_3) \equiv e_1 \times e_2 + e_1 \times e_3$

Propriété (absence de confluence) En général, pour une expression arithmétiue e, il existe des pas de réduction $e \to_{\mathbb{E}} e_1$ et $e \to_{\mathbb{E}} e_2$ t.q. il n'existe pas d'expression e' t.q. $e_1 \to_{\mathbb{E}}^{*} e'$ et $e_2 \to_{\mathbb{E}}^{*} e'$

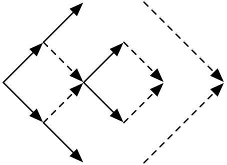


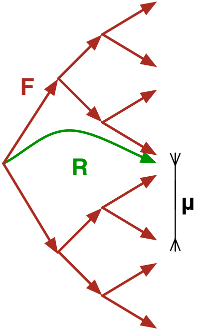
Bilan

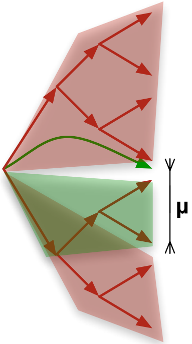
Sémantique machine ($\mathbb{E}$)

Sémantique abstraite

Sémantique idéale ($\mathbb{R}$)







Etape suivante :

- **objectif:** utiliser $\rightarrow_{\mathbb{E}}$ pour trouver le meilleur chemin
- trop de chemins!
- abstraction de $\rightarrow_{\mathbb{E}}$ fondée sur une sémantique non-standard $\longrightarrow$

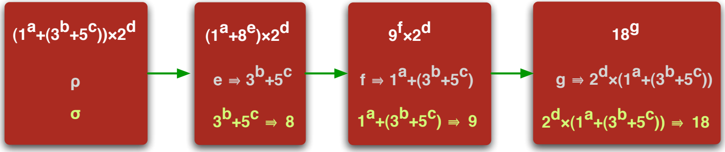
Sémantique non-standard (fondée sur $\longrightarrow_{\mathbb{E}}$)

Principe

- retenir comment chaque valeur a été obtenue
- une étiqutte $\ell \in \mathcal{L}$ est attachée à chaque valeur des expressions
- deux environnements :
  - $\rho : \mathcal{L} \rightarrow \text{Expr} \quad \rho(\ell) = e$ si $e \rightarrow^* v^\ell$
  - $\sigma : \text{Expr} \rightarrow \mathbb{E} \quad \sigma(e) = v$ si $e \rightarrow^* v$

$$\frac{v = v_1 +_{\mathbb{E}} v_2 \quad \ell \notin \text{Dom}(\rho)}{\langle \rho, \sigma, v_1^{\ell_1} + v_2^{\ell_2} \rangle \longrightarrow \langle \rho[\ell \mapsto \rho(\ell_1) + \rho(\ell_2)], \sigma[\rho(\ell_1) + \rho(\ell_2) \mapsto v], v^\ell \rangle}$$

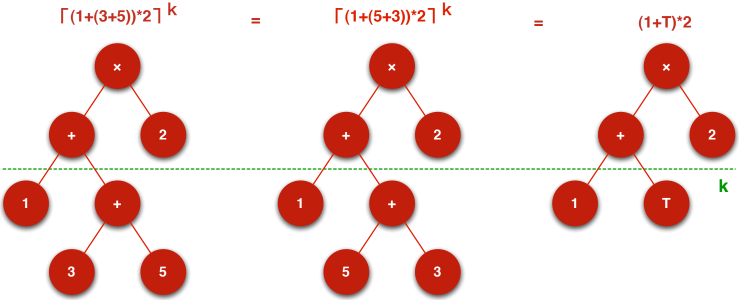
$$\frac{e \equiv e_1 \quad \langle \rho, \sigma, e_1 \rangle \longrightarrow \langle \rho', \sigma', e_1' \rangle \quad e_1' \equiv e'}{\langle \rho, \sigma, e \rangle \longrightarrow \langle \rho', \sigma', e' \rangle}$$



Sémantique abstraite

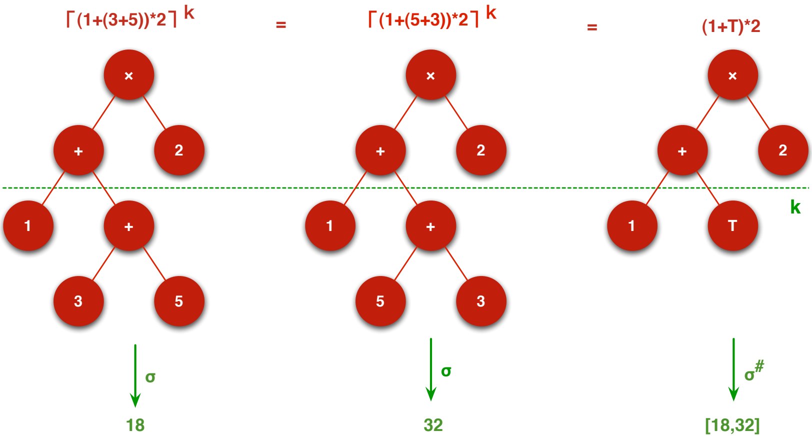
Principe

- Fusionner des traces dans lesquelles des sous-expressions ont été évaluées à peu près de la même manière
- $\mathrm{Expr}^\sharp_k$ ensemble des expressions de hauteur au plus k
- $\ulcorner e \urcorner^k$ remplace par $\top$ tous les sous-arbres de niveau k qui ne sont pas des valeurs



- les valeurs des environnements abstraits sont des intervalles

Sémantique abstraite (2)



$$\frac{v^\sharp = \bigcup_{\substack{\eta_1 \in \rho^\sharp(\ell_1)\\ \eta_2 \in \rho^\sharp(\ell_2)}} \sigma^\sharp(\eta_1) +^\sharp_{\mathbb{E}} \sigma^\sharp(\eta_2) \qquad E = \bigcup_{\substack{\eta_1 \in \rho^\sharp(\ell_1)\\ \eta_2 \in \rho^\sharp(\ell_2)}} \ulcorner \eta_1 + \eta_2 \urcorner^k \qquad \sigma^{\sharp\prime} = \sigma^\sharp \bigodot_{\substack{\eta_1 \in \rho^\sharp(\ell_1),\ \eta_2 \in \rho^\sharp(\ell_2)\\ \eta = \ulcorner \eta_1 + \eta_2 \urcorner^k\\ \nu = \sigma^\sharp(\eta_1) + \sigma^\sharp(\eta_2)}} [\eta \mapsto \sigma^\sharp(\eta) \cup}{\langle \rho^\sharp, \sigma^\sharp, v_0^{\ell_0} + v_1^{\ell_1} \rangle \xrightarrow{\ell = \ell_1 + \ell_2} {}_k\langle \rho^\sharp[\ell \mapsto \rho^\sharp(\ell) \cup E], \sigma^{\sharp\prime}, v^\ell \rangle}$$

Sémantique abstraite : exemple

Sémantique non-standard

$(1^a+(3^b+5^c))\times 2^d$		$(1^a+8^e)\times 2^d$		$9^f\times 2^d$		18^g
ρ	$\rightarrow$	$e \Rightarrow 3^b+5^c$	$\rightarrow$	$f \Rightarrow 1^a+(3^b+5^c)$	$\rightarrow$	$g \Rightarrow 2^d\times(1^a+(3^b+5^c))$
σ		$3^b+5^c \Rightarrow 8$		$1^a+(3^b+5^c) \Rightarrow 9$		$2^d\times(1^a+(3^b+5^c)) \Rightarrow 18$

Sémantique abstraite

$(1^a+(3^b+5^c))\times 2^d$		$(1^a+8^e)\times 2^d$		$9^f\times 2^d$		18^g
$\rho^\#$	$\rightarrow$	$e \Rightarrow 3^b+5^c$	$\rightarrow$	$f \Rightarrow 1^a+(3^b+5^c)$	$\rightarrow$	$g \Rightarrow 2^d\times(1^a+T)$
$\sigma^\#$		$3^b+5^c \Rightarrow [8]$		$1^a+(3^b+5^c) \Rightarrow [9]$		$2^d\times(1^a+T) \Rightarrow [18]$

Nouvelle relation d'équivalence

Dans la sémantique non-standard :

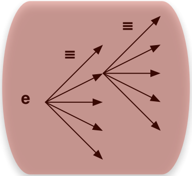
- utilisation de $\equiv$ (équivalence mathématique)
- à partir de e, une transition par expression e' t.q. $e \equiv e'$
- nombre exponentiel de transitions (pire cas: $\sum_{i=1}^{n} x_i$)

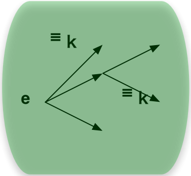
Dans la sémantique abstraite :

- nouvelle relation d'équivalence $e \sim_k e'$ ssi $\ulcorner e \urcorner^k = \ulcorner e' \urcorner^k$
- la sémantqiue abstraite utilise la relation quotient $\equiv_k$ ssi $\equiv / \sim_k$

$$\frac{e \equiv_k e_1 \quad e_1 \rightarrow e_1' \quad e_1' \equiv_k e'}{e \rightarrow e'}$$

- Au pire, la classe par $\equiv_k$ de e contient $O(n^k)$ éléments

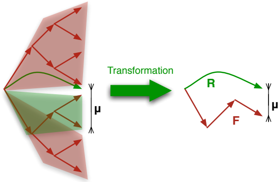




Transformation sémantique

Principe :

1. choisir la trace abstraite qui minimise l'erreur
2. construire une nouvelle trace qui n'utilise pas $\equiv$
   - des actions sont attachées aux transitions abstraites
   - les actions indiquent comment de nouvelles étiquettes ont été générées



Exemple

$$2\times(1+(3+5)) \xrightarrow{4=1+3} 2\times(4+5)) \xrightarrow{8=2\times 4} 8+2\times 5 \xrightarrow{10=2\times 5} 8+10 \xrightarrow{18=8+10} 18$$

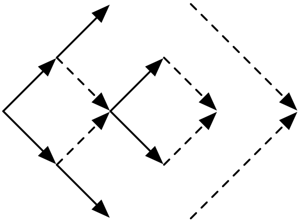
$$2\times(1+3)+2\times 5 \longrightarrow 2\times 4+2\times 5 \longrightarrow 8+2\times 5 \longrightarrow 8+10 \longrightarrow 18$$

Transformation sémantique (correction)

Correction :

- la trace d'origine et la trace transformée sont égales à observation près
- intuitivement les deux traces mènent au même résultat réel
- $\alpha_{\mathcal{O}}$ abstraction observationnelle :

$$\begin{array}{rcccc} \alpha_{\mathcal{O}} & : & \mathbb{E} & \to & \mathbb{R} \\ & & (x, \mu) & \mapsto & x + \mu \end{array}$$



Résultats expérimentaux

En simple précision, $1.0 + 5e^{-8} = 1.0$ tandis que $1.0 + (2 \times 5e^{-8}) \neq 1.0$

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

avec

$$a = [56789, 98765] \quad b = [0, 1] \quad c = [0, 5e^{-8}] \quad d = [0, 5e^{-8}]$$

`(a*((b+c)+d)) -> ((a*b)+(a*(c+d)))` k=2

borne d'erreur sur `(a*((b+c)+d))` : [-1.5679E-2,1.5680E-2]

borne d'erreur sur `((a*b)+(a*(c+d)))` : [-7.8125E-3,7.8126E-3]

Résultats expérimentaux

$s = \sum_{i=0}^{4} x_i$, avec $x_i = [2^i, 2^{i+1}]$:

`a`, `b`, `c`, `d` et `e` remplacent x_0, x_1, x_2, x_3 et x_4

k	Expression	borne d'erreur
expression source	`(((e+d)+c)+b)+a`	[-7.6293E-6,7.6294E-6]
$k = 1$	`(b+a)+(c+(e+d))`	[-5.9604E-6,5.9605E-6]
$k = 2$	`(c+(b+a))+(e+d)`	[-4.5299E-6,4.5300E-6]
$k = 3$	`(d+(c+(a+b)))+e`	[-3.5762E-6,3.5763E-6]

Polynômes, $x = ([0, 2], [0, 0.0005])$:

Cas	Expression	borne d'erreur
expression source	`x+(x*x)`	[-1.800074334E-3,1.001074437E-3]
$k = 2$	`(1.0+x)*x`	[-9.000069921E-4,1.010078437E-4]
expression source	`(x*(x*x))+(x*x)`	[-1.802887642E-3,3.191200091E-3]
$k = 3$	`(x+1.0)*(x*x)`	[-1.818142851E-4,1.390014781E-3]
$k = 4$	`((1.0+x)*x)*x`	[-9.091078216E-5,1.100112212E-3]

Travaux en cours...

- Traiter un langage complet (boucles, affectations, conditions, etc.)
  - Calculs définis sur plusieurs lignes de code
  - Dépliage de boucles, modification des affectations
- Amélioration de l'abstraction (optimisaton globale)
- Amélioration de la relation d'équivalence (plus de lois)
- Développement d'autres applications :
  - calculs en virgule fixe
  - offuscation de code sans perte de précision
- Vers un outil (langae source? destination?)

Travaux en cours

Plus généralement...

- Analyses relationnelles
- Analyses probabilistes
- Analyses de systèmes synchrones
- Itérations sur les politiques
- Systèmes hybrides discrets/continus
- Amélioration de la précision
- ...

Systèmes hybrides

Embedded programs as Hybrid systems

We are interested in the **verification and validation** of **industrial** systems. These are generally composed of two distinct parts: a discrete embedded program and the continuous physical environment surrounding it. Each subsystem has its own particularities and modelling each requires a **good expertise of the domain** (see Figure 1). Therefore, we try to analyse these systems without using models where the discrete and continuous worlds are mixed.

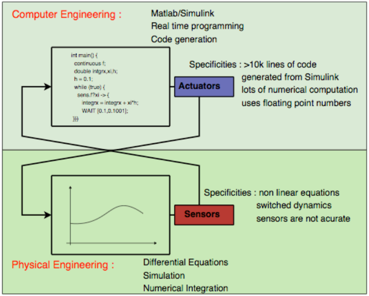


Figure 1: An hybrid system

We thus proposed [1] a model for hybrid systems in which we basically add hybrid actions (like sensors and actuators) to an imperative language. The program is then coupled to a description of the continuous physical environment. We base our analysis on this model.

Goal of the analysis

We want to prove that the **uncertainties** due to the implementation of the system does not affect its behaviour. These uncertainties are due first to the imprecision of the **numerical computations in floating point numbers**, but also to the **imprecision of the sensors**. In addition to that, the time at which the sensors and/or actuators are activated is in practice unknown, which introduces new potential errors. Our analysis aims at proving that the differences between the implementation where all these errors occur and the ideal model with perfect sensors/actuators and real numbers computations does not modify the behaviour of the system (see Figure 2).

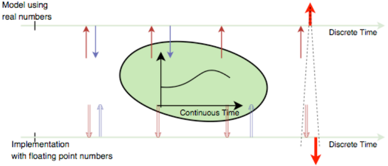
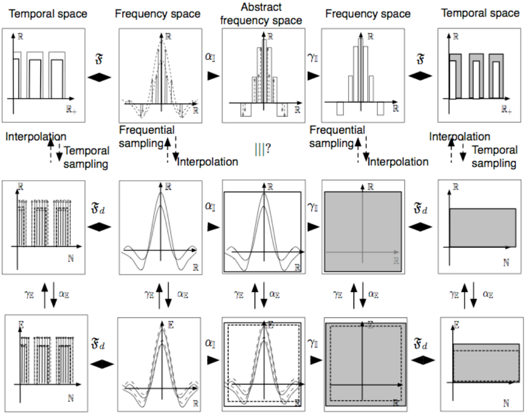


Figure 2: Actions taken by the ideal model (top) and by the implementation (bottom)

We thus only focus on some actions: sensors (dark red arrows) and actuators (blue) activations, as well as alarms (red), and we show that these actions occur almost at the same time in both cases.

Systèmes synchrones



Mathematical ideal model

- Continuous periodic time functions
- Discrete frequency functions: IDEAL SPECTRA

Sampling time operation

- Discrete time functions with real range
- Periodic frequency functions: COMPUTED SPECTRA

Quantization operation

- Introduce rounding errors in signal values
- Add noise to frequency spectra