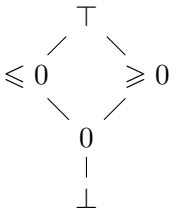
Interprétation abstraite (suite)

November 29, 2021

Rappel des épisodes précédents

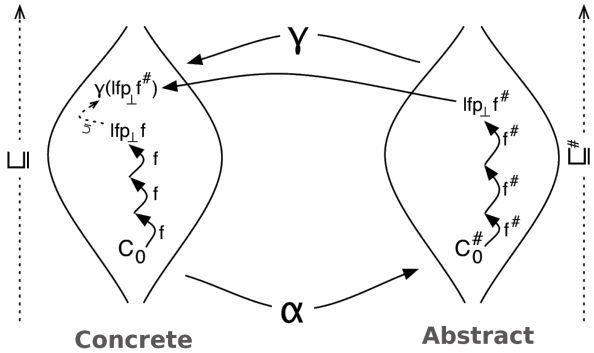


- Analyse de programmes avec approximations: **interprétation abstraite**.
- Trouver des invariants avec un algorithme de calcul de point-fixe.
- Un framework générique avec des résultats théoriques intéressants.
- Exemples de domaines non-relationels et à hauteur finie: Signes, Constantes, ...

▶ Pour l'instant, pas de domaine relationel, ni de treillis à hauteur infinie!

Credit: diagramme de P. Roux

Illustration du calcul du point-fixe (concret vs abstrait)



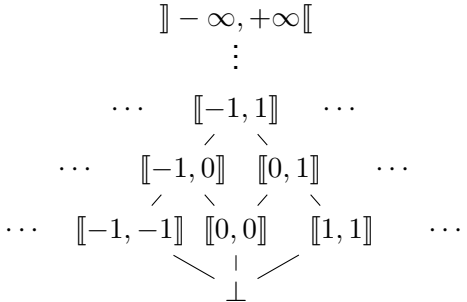
Le résultat principal: la sûreté

Sûreté de l'interprétation abstraite pour des domaines non-relationels de hauteur finie

Si le calcul se termine, alors le résultat est **sur**

$$R \subseteq \gamma\left(R^{\sharp}\right)$$

Le domaines de intervalles, basé sur le treillis $(\mathcal{D}^\sharp, \sqsubseteq^\sharp)$



$$\begin{aligned}
\gamma(]-\infty, +\infty[) &=]-\infty, +\infty[\\
\gamma(]-\infty, n]) &=]-\infty, n] \\
\gamma([n, +\infty[) &= [n, +\infty[\\
\gamma([n_1, n_2]) &= [n_1, n_2] \\
\gamma(\bot) &= \emptyset
\end{aligned}$$

$$\alpha(S) = \begin{cases} [n_1, n_2] & \text{avec } n_1 = \min S \text{ et } n_2 = \max S \\ \bot & \text{quand } S = \emptyset \end{cases}$$

Exercice: définir l'opération abstraite + et définir l'union abstraite.

Un exemple qui se termine

```
int x=0;
while (x<1000) {
  x=x+1;
}
```

Itérations $[0,0]$, $[0,1]$, $[0,2]$, $[0,3]$,...

- Intervalle strictement croissant pendant 1000 itérations, puis stabilisation: $[0,1000]$ est un **invariant**.

Sûreté et terminaison

On a toujours la sûreté, mais:

Mauvaise nouvelle!

- En général, cela ne se termine pas!
  Parce que le treillis a des chaines ascendantes infinies
  (e.g., $(\llbracket 0, n \rrbracket)_{n \in \mathbb{N}}$).
- Et même lorsque cela se termine, c'est peut-être long...

Une solution: l'extrapolation!

```
int x=0;
while (x<1000) {
  x=x+1;
}
```

$[0,0]$, $[0,1]$, $[0,2]$, $[0,3] \rightarrow [0,+\infty)$

$[0,\infty[$ est stable!

- insérer entre chaque itération un **widening** qui empeche de suivre une chaine ascendante infinie, en "sautant" plus haut.

Le widening

Definition (widening)

Le widening $\triangledown$ est une opération binaire ($\triangledown : \mathcal{D}^\sharp \times \mathcal{D}^\sharp \to \mathcal{D}^\sharp$) aux propriétés suivantes:

- $\forall x^\sharp, y^\sharp, \quad x^\sharp \sqcup^\sharp y^\sharp \ \sqsubseteq^\sharp \ x^\sharp \triangledown y^\sharp$;
- pour toute séquence $\left(x_n^\sharp\right)_{n\in\mathbb{N}}$, la séquence croissante

$$\begin{cases} y_0^\sharp &= x_0^\sharp \\ y_{i+1}^\sharp &= y_i^\sharp \triangledown x_{i+1}^\sharp \end{cases}$$

est stationnaire.

Widening sur les intervalles

Idée : quand les bornes d'un intervalle sont stables, on les garde, sinon on les remplace par ∞.

$$x^\sharp \triangledown y^\sharp = \begin{cases} [\![a, b]\!] & \text{if } x^\sharp = [\![a, b]\!], y^\sharp = [\![c, d]\!], c \geqslant a, d \leqslant b \\ [\![a, +\infty[\![& \text{if } x^\sharp = [\![a, b]\!], y^\sharp = [\![c, d]\!], c \geqslant a, d > b \\ ]\!-\infty, b]\!] & \text{if } x^\sharp = [\![a, b]\!], y^\sharp = [\![c, d]\!], c < a, d \leqslant b \\ ]\!-\infty, +\infty[\![& \text{if } x^\sharp = [\![a, b]\!], y^\sharp = [\![c, d]\!], c < a, d > b \\ y^\sharp & \text{if } x^\sharp = \bot \\ x^\sharp & \text{if } y^\sharp = \bot \end{cases}$$

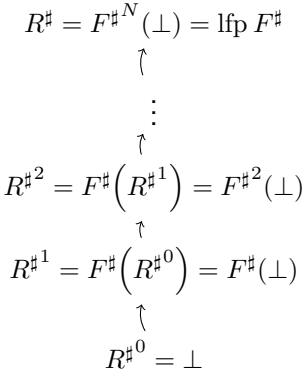
Exercice

- $[\![0, 2]\!] \triangledown [\![0, 1]\!] = ???$
- $[\![0, 1]\!] \triangledown [\![0, 2]\!] = ???$

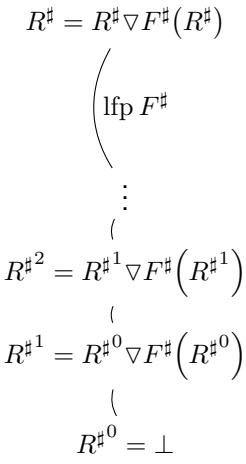
Remarks:

- $\triangledown$ n'est pas commutatif
- $x \triangledown y$ est souvent défini avec $x \sqsubseteq y$

Widening, Illustration



$F^\sharp$ stationnaire



$F^\sharp$ non stationnaire, widening

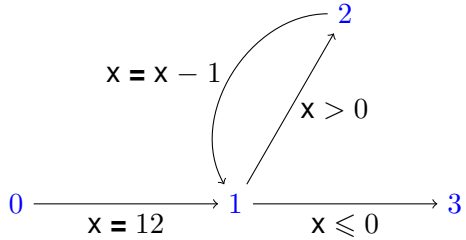
Exemple de calcul de point-fixe abstrait

$_0$x = 12;

while $_1$(x > 0) **{**

  $_2$ x = x − 1;

}$_3$



$$
\begin{aligned}
R_0^{\sharp\, i+1} &= \top \\
R_1^{\sharp\, i+1} &= R_1^{\sharp\, i} \nabla_{\mathrm{nr}} \left(R_0^{\sharp\, i+1} \left[x \mapsto [\![12, 12]\!] \right] \sqcup^\sharp_{\mathrm{nr}} R_2^{\sharp\, i} \left[x \mapsto R_2^{\sharp\, i}(x) -^\sharp [\![1, 1]\!] \right] \right) \\
R_2^{\sharp\, i+1} &= R_1^{\sharp\, i+1} \left[x \mapsto R_1^{\sharp\, i+1}(x) \sqcap^\sharp [\![1, +\infty[\![\right] \\
R_3^{\sharp\, i+1} &= R_1^{\sharp\, i+1} \left[x \mapsto R_1^{\sharp\, i+1}(x) \sqcap^\sharp]\!]-\infty, 0]\!] \right]
\end{aligned}
$$

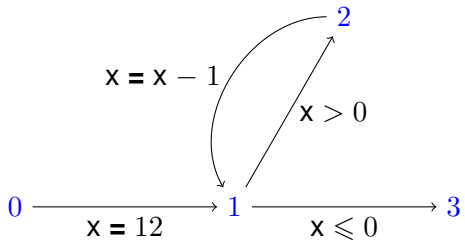
Exemple de calcul de point-fixe abstrait

$_0$x = 12;

while $_1$(x > 0) {

  $_2$ x = x $- 1$;

}$_3$



l	$R_k^{\sharp\,0}$	$R_k^{\sharp\,1}$	$R_k^{\sharp\,2}$	$R_k^{\sharp\,3}$
0	$\bot$			
1	$\bot$			
2	$\bot$			
3	$\bot$			

$$
\begin{aligned}
R_0^{\sharp\,i+1} &= \top \\
R_1^{\sharp\,i+1} &= R_1^{\sharp\,i} \nabla_{\mathrm{nr}} \left(R_0^{\sharp\,i+1} \left[x \mapsto [\![12, 12]\!] \right] \sqcup^\sharp_{\mathrm{nr}} R_2^{\sharp\,i} \left[x \mapsto R_2^{\sharp\,i}(x) -^\sharp [\![1, 1]\!] \right] \right) \\
R_2^{\sharp\,i+1} &= R_1^{\sharp\,i+1} \left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp [\![1, +\infty[\![\right] \\
R_3^{\sharp\,i+1} &= R_1^{\sharp\,i+1} \left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp]\!]-\infty, 0]\!] \right]
\end{aligned}
$$

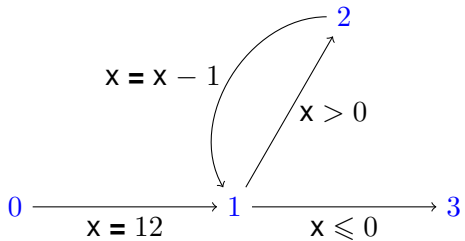
Exemple de calcul de point-fixe abstrait

$_0$x = 12;

while $_1$(x $>$ 0) **{**

  $_2$ x = x $-$ 1;

}$_3$



l	$R_k^{\sharp\,0}$	$R_k^{\sharp\,1}$	$R_k^{\sharp\,2}$	$R_k^{\sharp\,3}$
0	$\bot$	$\top$		
1	$\bot$	$[\![12,12]\!]$		
2	$\bot$	$[\![12,12]\!]$		
3	$\bot$	$\bot$		

$$
\begin{aligned}
R_0^{\sharp\,i+1} &= \top \\
R_1^{\sharp\,i+1} &= R_1^{\sharp\,i} \triangledown_{\mathrm{nr}} \left(R_0^{\sharp\,i+1} \left[x \mapsto [\![12,12]\!] \right] \sqcup^\sharp_{\mathrm{nr}} R_2^{\sharp\,i} \left[x \mapsto R_2^{\sharp\,i}(x) -^\sharp [\![1,1]\!] \right] \right) \\
R_2^{\sharp\,i+1} &= R_1^{\sharp\,i+1} \left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp [\![1,+\infty[\![\right] \\
R_3^{\sharp\,i+1} &= R_1^{\sharp\,i+1} \left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp]\!]-\infty,0]\!] \right]
\end{aligned}
$$

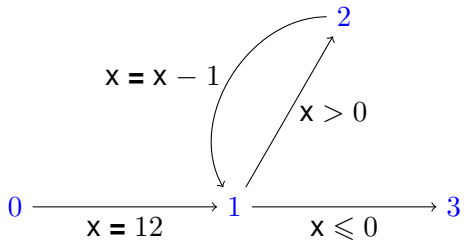
Exemple de calcul de point-fixe abstrait

$_0$x = 12;

while $_1$(x > 0) {

  $_2$ x = x $- 1$;

}$_3$



l	$R_k^{\sharp\,0}$	$R_k^{\sharp\,1}$	$R_k^{\sharp\,2}$	$R_k^{\sharp\,3}$
0	$\bot$	$\top$	$\top$	
1	$\bot$	$[\![12,12]\!]$	$]\!]-\infty,12]\!]$	
2	$\bot$	$[\![12,12]\!]$	$[\![1,12]\!]$	
3	$\bot$	$\bot$	$]\!]-\infty,0]\!]$	

$$
\begin{aligned}
R_0^{\sharp\,i+1} &= \top \\
R_1^{\sharp\,i+1} &= R_1^{\sharp\,i} \nabla_{\mathrm{nr}} \left(R_0^{\sharp\,i+1} \left[x \mapsto [\![12,12]\!] \right] \sqcup^\sharp_{\mathrm{nr}} R_2^{\sharp\,i} \left[x \mapsto R_2^{\sharp\,i}(x) -^\sharp [\![1,1]\!] \right] \right) \\
R_2^{\sharp\,i+1} &= R_1^{\sharp\,i+1} \left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp [\![1,+\infty[\![\right] \\
R_3^{\sharp\,i+1} &= R_1^{\sharp\,i+1} \left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp]\!]-\infty,0]\!] \right]
\end{aligned}
$$

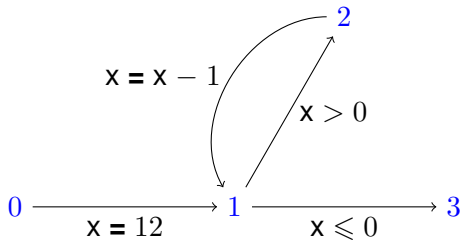
Exemple de calcul de point-fixe abstrait

$_0$x = 12;

while $_1$(x $>$ 0) {

$_2$ x = x $-$ 1;

}$_3$



l	$R_k^{\sharp\,0}$	$R_k^{\sharp\,1}$	$R_k^{\sharp\,2}$	$R_k^{\sharp\,3}$
0	$\bot$	$\top$	$\top$	$\top$
1	$\bot$	$[\![12,12]\!]$	$]\!]-\infty,12]\!]$	$]\!]-\infty,12]\!]$
2	$\bot$	$[\![12,12]\!]$	$[\![1,12]\!]$	$[\![1,12]\!]$
3	$\bot$	$\bot$	$]\!]-\infty,0]\!]$	$]\!]-\infty,0]\!]$

$$
\begin{aligned}
R_0^{\sharp\,i+1} &= \top \\
R_1^{\sharp\,i+1} &= R_1^{\sharp\,i} \triangledown_{\mathrm{nr}} \left(R_0^{\sharp\,i+1}\,[x \mapsto [\![12,12]\!]] \sqcup^\sharp_{\mathrm{nr}} R_2^{\sharp\,i}\left[x \mapsto R_2^{\sharp\,i}(x) -^\sharp [\![1,1]\!]\right]\right) \\
R_2^{\sharp\,i+1} &= R_1^{\sharp\,i+1}\left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp [\![1,+\infty[\![\right] \\
R_3^{\sharp\,i+1} &= R_1^{\sharp\,i+1}\left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp]\!]-\infty,0]\!]\right]
\end{aligned}
$$

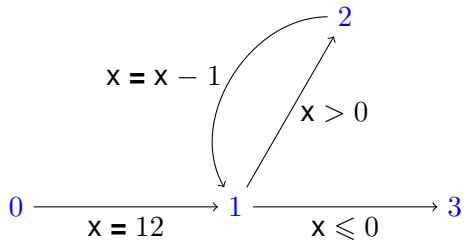
Exemple de calcul de point-fixe abstrait

$_0$x = 12;

while $_1$(x > 0) {

$_2$ x = x $- 1$;

}$_3$



l	$R_k^{\sharp\,0}$	$R_k^{\sharp\,1}$	$R_k^{\sharp\,2}$	$R_k^{\sharp\,3}$
0	$\bot$	$\top$	$\top$	$\top$
1	$\bot$	$[\![12,12]\!]$	$]\!]-\infty,12]\!]$	$]\!]-\infty,12]\!]$
2	$\bot$	$[\![12,12]\!]$	$[\![1,12]\!]$	$[\![1,12]\!]$
3	$\bot$	$\bot$	$]\!]-\infty,0]\!]$	$]\!]-\infty,0]\!]$

$$
\begin{aligned}
R_0^{\sharp\,i+1} &= \top \\
R_1^{\sharp\,i+1} &= R_1^{\sharp\,i} \triangledown_{\mathrm{nr}} \left(R_0^{\sharp\,i+1}\,[x \mapsto [\![12,12]\!]] \sqcup^\sharp_{\mathrm{nr}} R_2^{\sharp\,i}\left[x \mapsto R_2^{\sharp\,i}(x) -^\sharp [\![1,1]\!]\right]\right) \\
R_2^{\sharp\,i+1} &= R_1^{\sharp\,i+1}\left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp [\![1,+\infty[\![\right] \\
R_3^{\sharp\,i+1} &= R_1^{\sharp\,i+1}\left[x \mapsto R_1^{\sharp\,i+1}(x) \sqcap^\sharp]\!]-\infty,0]\!]\right]
\end{aligned}
$$

Pourtant $x = 0$ à la fin!(des infos sur la précision plus tard...)

Quand les intervalles ne sont pas suffisants

```
assume(x >= 0 && x <= 1);
y = x;
z = x-y;
```

- L'humain (intelligent) voit que $z = 0$ (car $y = x$).
- Le domaine des intervalles ne prends pas en compte $y = x$ et ne voit donc pas que $z = 0$.

Rappels

Pour les valeurs numériques, on peut:

- (non relationel) abstraire $\mathcal{P}(\text{Var} \to \mathbb{Z})$ en $\text{Var} \to \mathcal{P}(\mathbb{Z})$ puis $\mathcal{P}(\mathbb{Z})$ en $\mathcal{D}^\sharp$
- relationel: abstraire $\mathcal{P}(\text{Var} \to \mathbb{Z})$ directement en $\mathcal{D}^\sharp$
  - $+$ plus précis
  - $-$ plus compliqué et coûteux
  - octagones, polyèdres, ...

Comment prendre en compte les relations

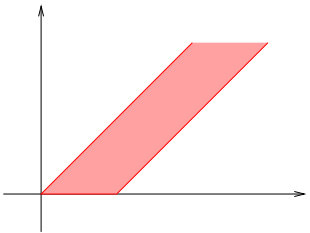
En utilisant de **domaines relationels**

Exemple: bornes sur les différences entre variables

- un intervalle pour chaque variable
- une information $x - y \leqslant C$ pour chaque paire de variables .
- (on représente $x = y$ par $x - y \leqslant 0$ et $y - x \leqslant 0$.)

Comment le **calculer?**

Bornes sur les différences: exemple



Supposons $x - y \leqslant 4$, on calcule $z = x + 3$, on en déduit $z - y \leqslant 7$. Supposons $x - z \leqslant 20$, $x - y \leqslant 4$ et $y - z \leqslant 6$, on en déduit $x - z \leqslant 10$.

- On sait comment **calculer** ces relations.

Pourquoi c'est utile?

Soit $t(0..\boldsymbol{n})$ un tableau dans le programme.
Le programme écrit dans $t(i)$.

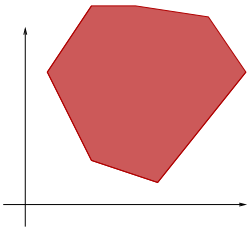
On veut savoir si $0 \leqslant i \leqslant n$, autrement dit trouver des bornes sur i et $n - i$...

Est ce qu'on peut faire mieux?

Peut-on prendre en compte des relations telles que $2x + 3y \leqslant 6$?

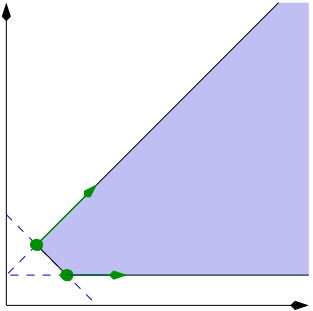
Associer un ensemble d'**inégalités linéaires** à chaque point du prgoramme.

Autrement dit, un **polyèdre convexe**.



Le domaine des polyèdres

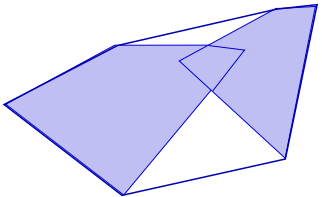
Représentation par polyèdre convexe :



- Algorithmes efficaces (test d'inclusions, transformations . . .) affine transformation . . .)

Le domaine des polyèdres

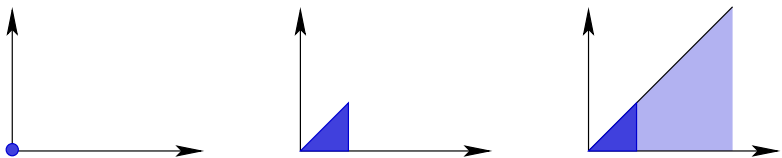
- Intersection: union des contraintes
- Transformation affine : $a(P) = \{CX + D \mid X \in P\}$.
- Join: Enveloppe convexe (perte de précision)



Le domaine des polyèdres

Widening : $P \nabla Q$ Il existe plusieurs versions du widening sur les polyèdres. Une version basique est la suivante:

Contraintes de $P \nabla Q$ = les contraintes de P, sans celles qui ne sont pas satisfaites par Q.



Astuce (!) : $\{x = y = 0\} = \{0 \leqslant y \leqslant x \leqslant 0\}$

Problemes et limitations

Sources de complexité :

- nombre de points de contrôle
- nombre de variables

Sources d'imprécision :

- Enveloppes convexes
- Widening

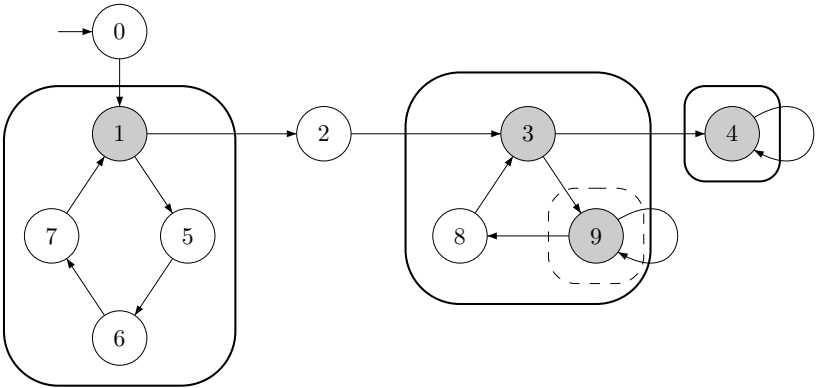
Complexity

(En general) Plus on est précis, plus c'est couteux:

- Intervalles: algorithmes en $O(n)$, n est le nombre de variables.
- Differences $x - y \leqslant C$: algorithmes en $O(n^3)$
- Octogones $\pm x \pm y \leqslant C$ (Miné) : algorithmes $O(n^3)$
- Polyèdres (Cousot / Halbwachs): algorithmes en $O(2^n)$.

Boucles imbriquées

(Bourdoncle, 1992) Utiliser les composantes fortement connexes :



Les noeuds gris sont les **noeuds de widening**

Améliorer la précision après convergence

```
int x=0; // [0, 0]
while (x<1000) {// [0, +infty)
  x=x+1;
}
```

On a $[0, +\infty)$ au lieu de $[0, 999]$. On fait une itération de plus de la boucle : $\{0\} \sqcup [1, 1000] = [0, 1000]$. On voit que $[0, 1000]$ est bien un invariant.

- Cette approche est appelée **narrowing** ou séquence descendante : on s'arrête lorsqu'on obtient un invariant, ou après k applications de la fonction de transition.

Widening retardé

Halbwachs 1993 / Goubault 2001 / Blanchet et al. 2003

Fixer k et calculer :

$$X_n = \begin{cases} \bot & \text{si } n = 0 \\ F(X_{n-1}) & \text{si } n < k \\ X_{n-1} \nabla F(X_{n-1}) & \text{sinon.} \end{cases}$$

- Similaire au déroulage de boucles. Couteux mais utile.