Validation par analyse statique

Partie : Interprétation abstraite, cours 1/3

Pierre-Loïc Garoche

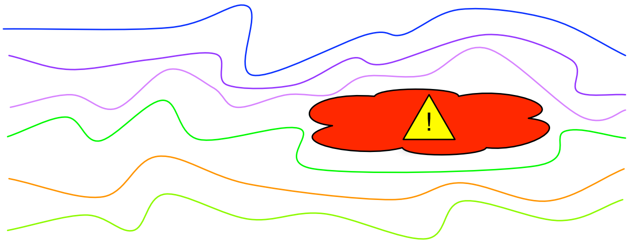
(merci à Pierre Roux pour ses contributions à ce cours)

ENAC

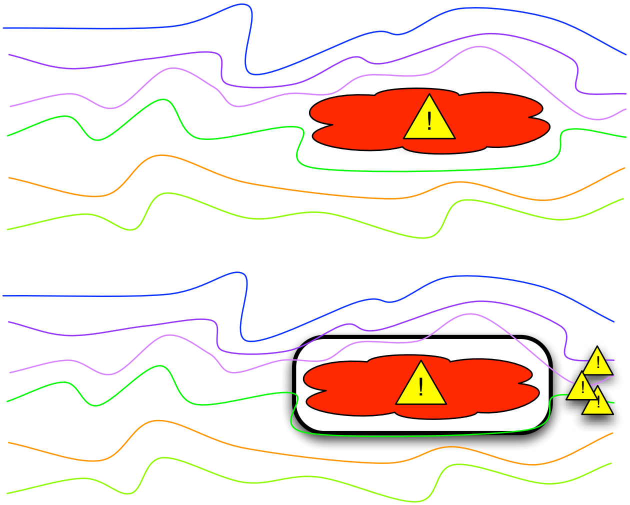
ENSEEIHT 2A

2022-2023

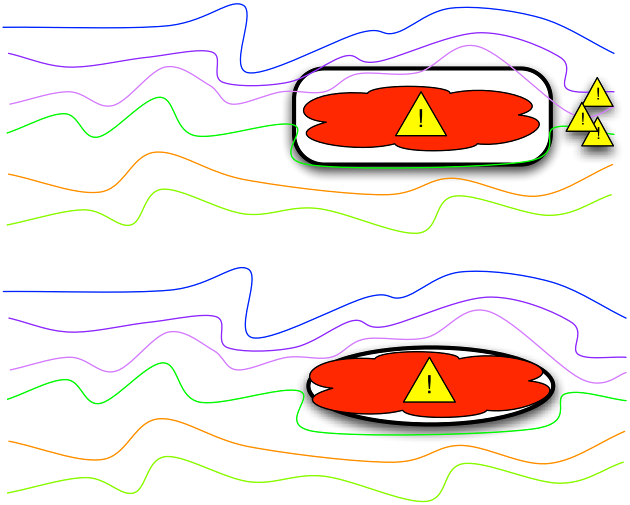
L'interprétation abstraite d'un coup d'oeil



L'interprétation abstraite d'un coup d'oeil



L'interprétation abstraite d'un coup d'oeil



Plan des 3 cours sur l'interprétation abstraite

1. Introduction à l'interprétation abstraite (aujourd'hui)
    - Exemple graphique
    - Sémantique collectrice d'un langage *C*-like
2. Abstractions numériques simples
    - domaine des signes
    - domaine des constantes
    - intervalles et accélération de convergence
3. Abstractions numériques relationnelles et bref état de l'art
    - domaine des polyèdres
    - aperçu d'autres analyses
    - quelques outils et applications

Un exemple graphique
- Un peu de dessin
- Notion de point fixe
- Notion d'abstraction
- Meilleure abstraction
- Opérations abstraites

Une approche plus... langage
- Syntaxe
- Sémantique
- Ordres partiels

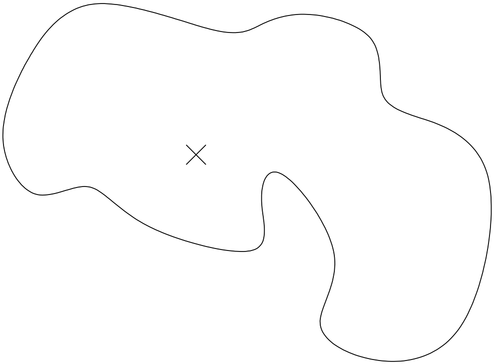
Un exemple graphique

But : donner les intuitions sur les principes généraux.

- $+$ simple et intuitif
- $-$ peu formel

. Exemple extrait du cours de L. Mauborgne à l'EJCP'06.

Objets

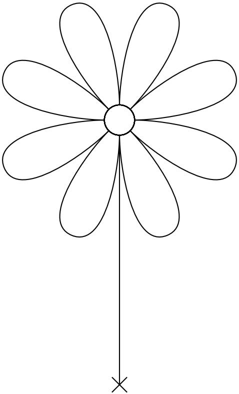


Définition

Un *objet* est défini par :

1. une origine (un point du plan)
2. un ensemble de points

Un objet : une fleur

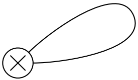


Des outils pour ce langage

Pour définir des objets, et les manipuler, nous avons besoin :

- d'objets de base (primitives, constantes)
- de fonctions pour les modifier

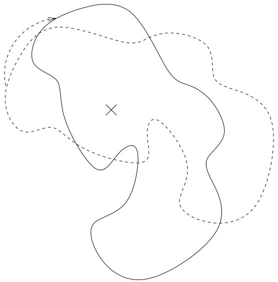
Constante : objet pétale

pétale = 

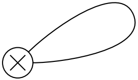
Fonction : rotation

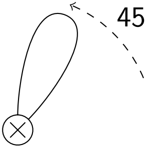
Définition

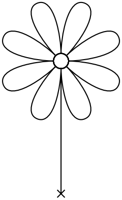
$\mathrm{r}[a](o)$ est la rotation d'angle a de l'objet o autour de son origine.

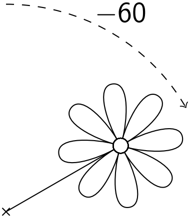


Exemples de rotations

pétale = 

r[45](pétale) = 

fleur = 

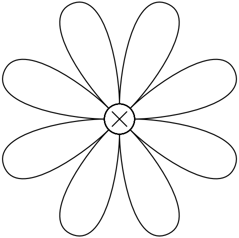
$\mathrm{r}[-60]$(fleur) = 

Fonction : union d'objets

Définition

$o_1 \sqcup o_2$ est l'union des objets o_1 et o_2 à l'origine.

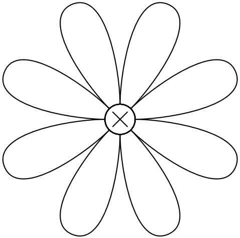
$$\begin{aligned}\text{corolle} = {} & \text{pétale} \sqcup \mathrm{r}[45](\text{pétale}) \sqcup \mathrm{r}[90](\text{pétale}) \sqcup \\ & \mathrm{r}[135](\text{pétale}) \sqcup \mathrm{r}[180](\text{pétale}) \sqcup \mathrm{r}[225](\text{pétale}) \sqcup \\ & \mathrm{r}[270](\text{pétale}) \sqcup \mathrm{r}[315](\text{pétale})\end{aligned}$$



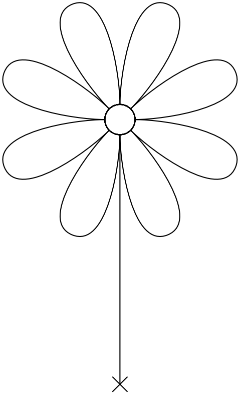
Fonction : tige

Définition

$\mathrm{tige}(o)$ ajoute une tige à o en partant de l'origine
et déplace ensuite l'origine au bas de la tige.

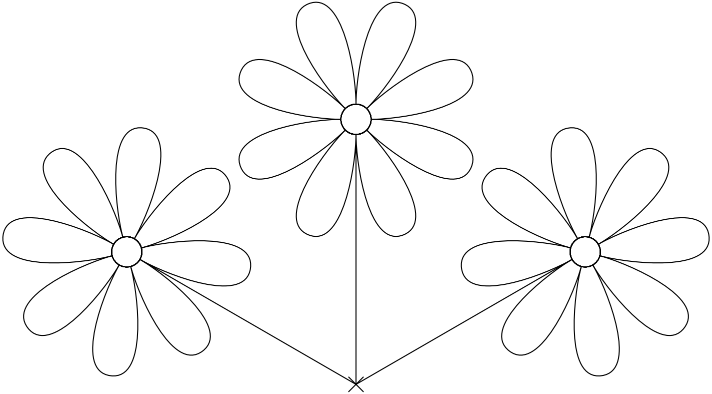


corolle



fleur = tige(corolle)

Construction de l'objet bouquet



$$\text{bouquet} = \mathrm{r}[60](\text{fleur}) \sqcup \text{fleur} \sqcup \mathrm{r}[-60](\text{fleur})$$

Un exemple graphique

- Un peu de dessin
- Notion de point fixe
- Notion d'abstraction
- Meilleure abstraction
- Opérations abstraites

Une approche plus... langage

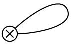
- Syntaxe
- Sémantique
- Ordres partiels

La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2

La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



1

La corolle construite de façon itérative...

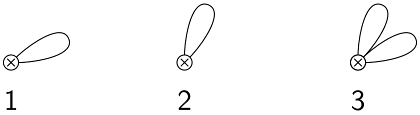
1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2

1

2

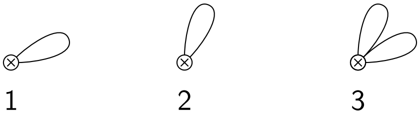
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2

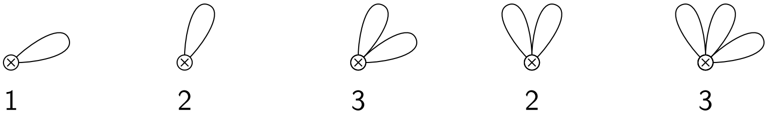


La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2

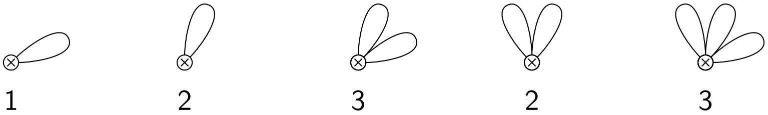
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



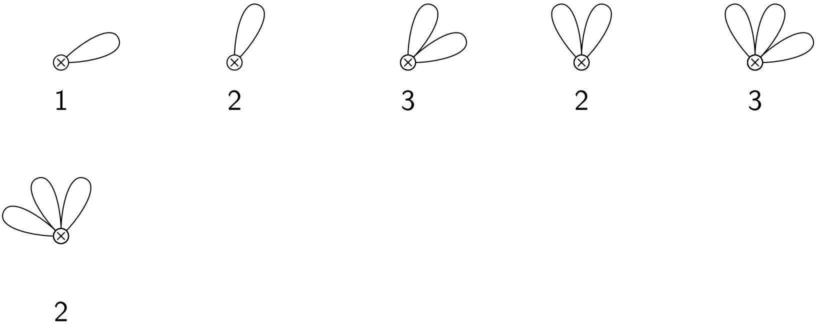
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



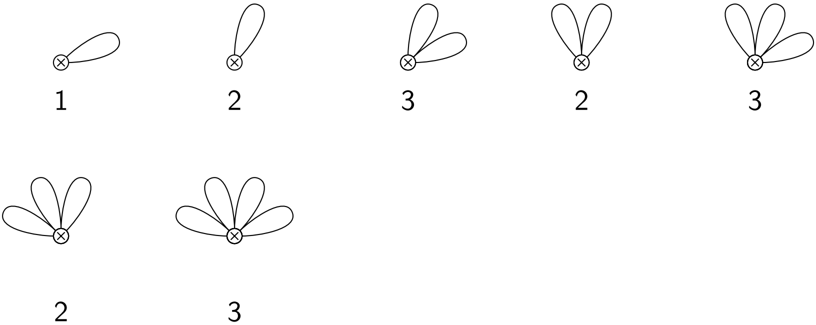
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2

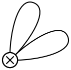


La corolle construite de façon itérative...

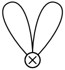
1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2

1

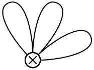
2



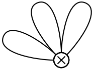
3



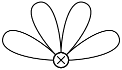
2



3



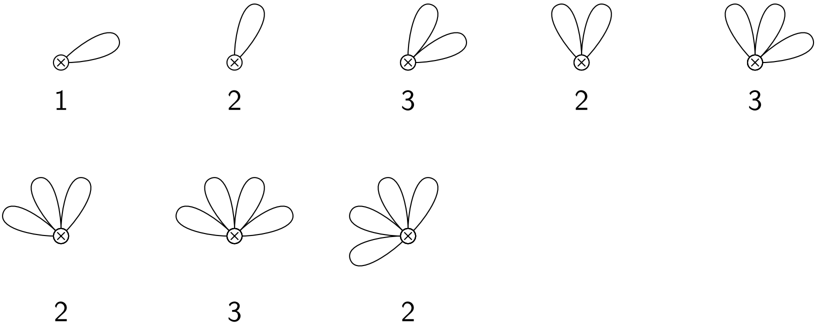
2



3

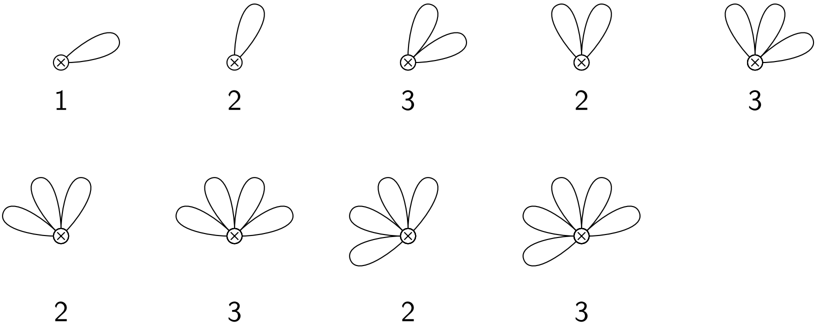
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



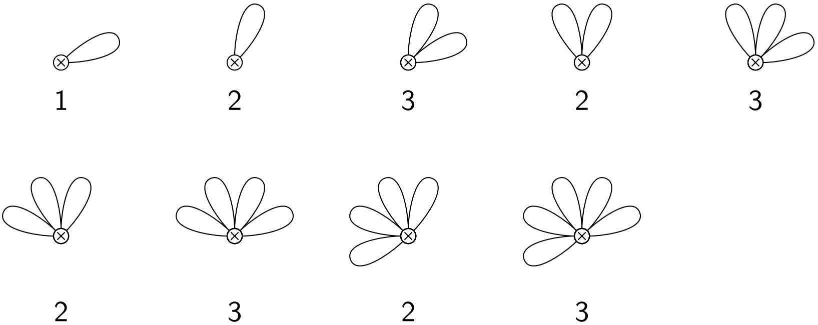
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



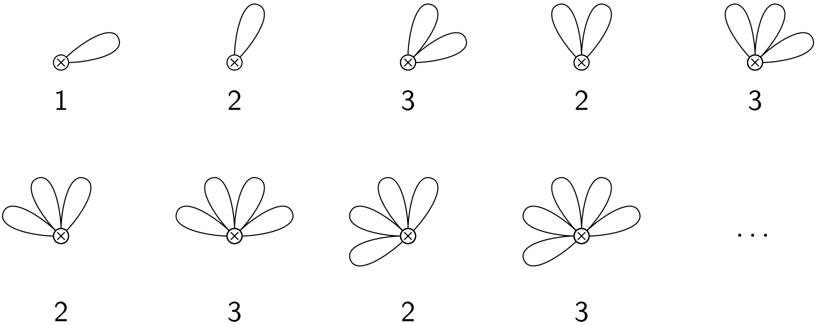
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



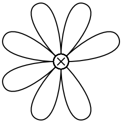
La corolle construite de façon itérative...

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \sqcup$ pétale)
4. retourner en 2



...est un point fixe

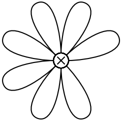
1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



3

...est un point fixe

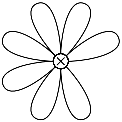
1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



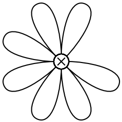
3

...est un point fixe

1. prendre un pétale (x := pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



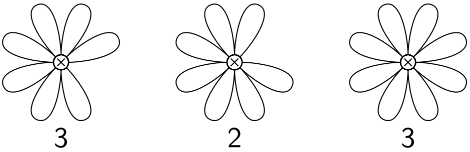
3



2

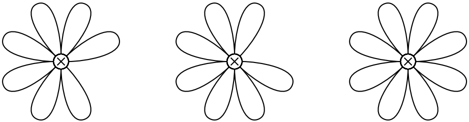
...est un point fixe

1. prendre un pétale (x := pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



...est un point fixe

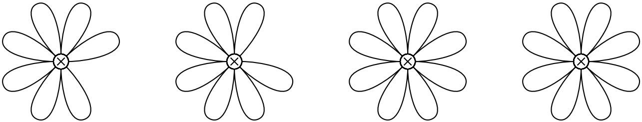
1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



3 2 3

...est un point fixe

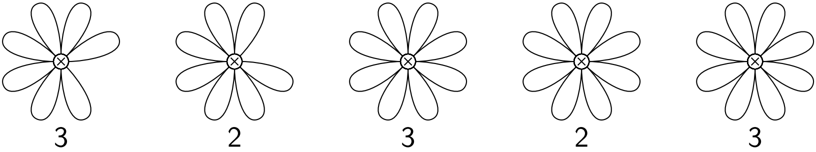
1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



3 2 3 2

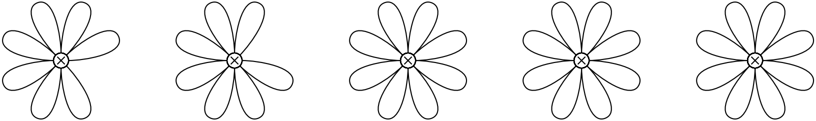
...est un point fixe

1. prendre un pétale (x := pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



...est un point fixe

1. prendre un pétale ($x :=$ pétale)
2. effectuer une rotation de 45 degré ($x := \mathrm{r}[45](x)$)
3. faire l'union avec pétale ($x := x \cup$ pétale)
4. retourner en 2



3 2 3 2 3

Définition (corolle)

C'est le plus petit objet X tel que :

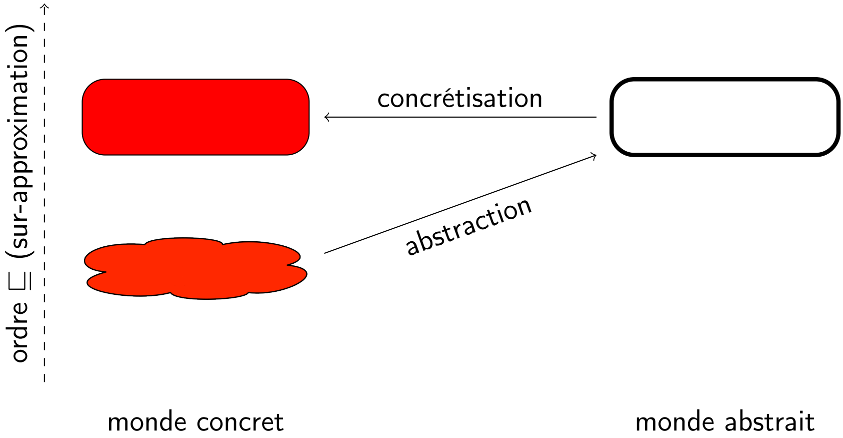
- pétale $\subseteq X$
- $\mathrm{r}[45](X) \sqcup$ pétale $\subseteq X$.

Noté : $$\text{corolle} = \mathrm{lfp}(X \mapsto \text{pétale} \sqcup \mathrm{r}[45](X))$$

Un exemple graphique
- Un peu de dessin
- Notion de point fixe
- Notion d'abstraction
- Meilleure abstraction
- Opérations abstraites

Une approche plus... langage
- Syntaxe
- Sémantique
- Ordres partiels

Concret — abstrait

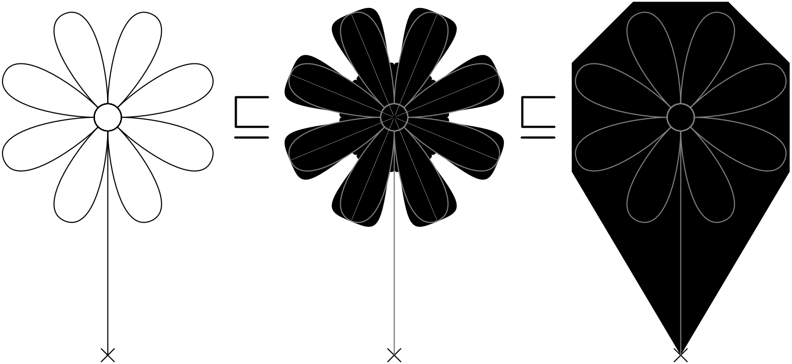


Sur-approximation

Définition (ordre $\sqsubseteq$ entre les objets concrets)

Un objet o' sur-approxime un objet o si :

- ils ont la même origine ;
- tout point de o est un point de o'.

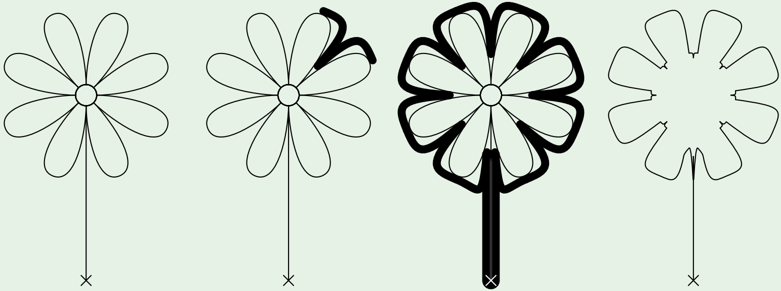


Les objets abstraits

Idée

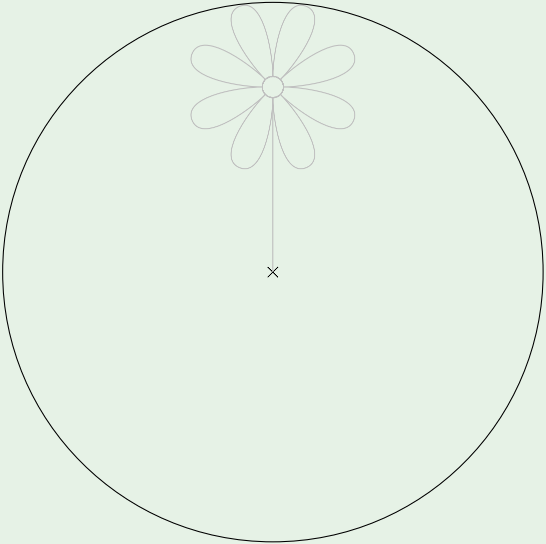
Un objet abstrait est une représentation simplifiée d'un objet.

Exemple d'abstraction : contours



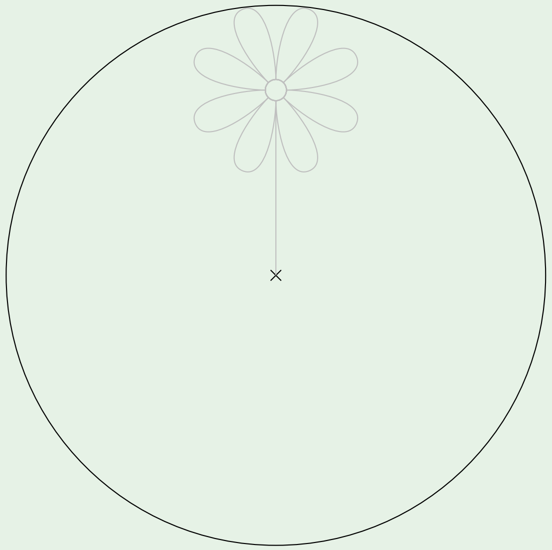
Autres exemples d'abstraction

Cercles centrés à l'origine

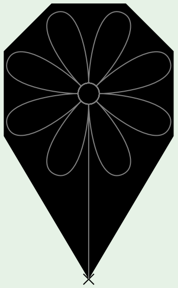


Autres exemples d'abstraction

Cercles centrés à l'origine



Polygones convexes



L'abstraction n'est pas injective

Attention

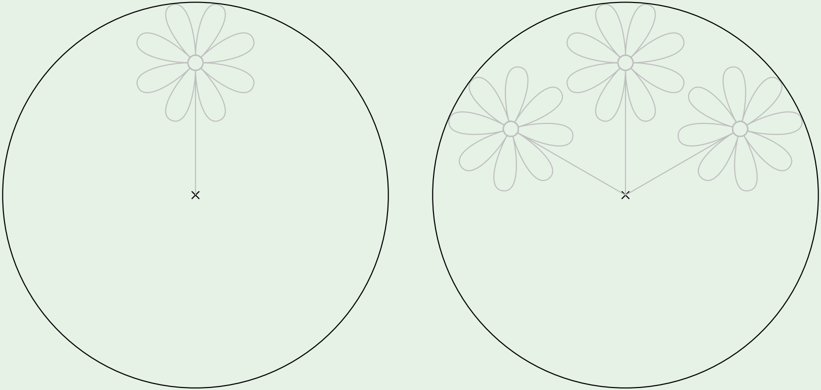
Plusieurs objets concrets peuvent être représentés par le même objet abstrait (sinon il n'y a pas vraiment d'abstraction).

L'abstraction n'est pas injective

Attention

Plusieurs objets concrets peuvent être représentés par le même objet abstrait (sinon il n'y a pas vraiment d'abstraction).

Exemple : fleur et bouquet sont dans le même cercle



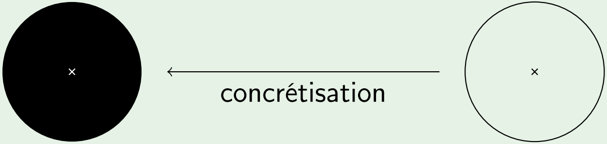
Concrétisation

Idée

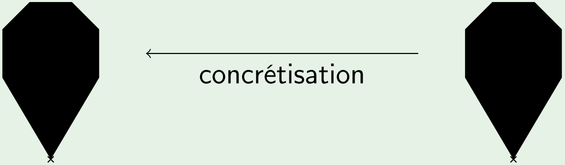
La concrétisation est la « réciproque » de l'abstraction.

Exemple

- Pour contours et cercles : remplissage de l'intérieur



- Pour polygones : identité



Sur-approximation dans l'abstrait

Définition (ordre $\sqsubseteq^\sharp$ entre les objets abstraits)

Dépend de l'abstraction choisie.

Exemples : $o \sqsubseteq^\sharp o'$ si :

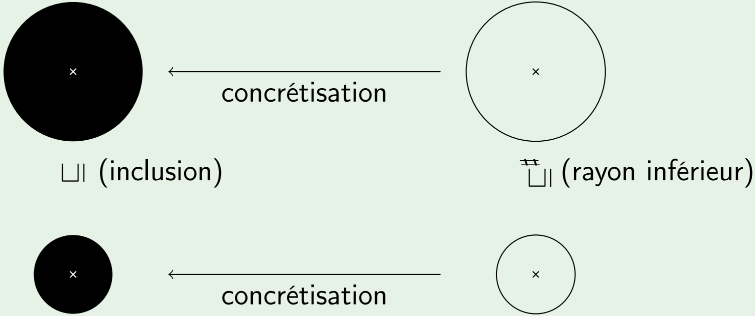
- polygones convexes :
  - o et o' ont la même origine
  - o est inclus dans o'
- contour :
  - o et o' ont la même origine
  - l'intérieur de o est inclus dans celui de o'
- cercles centrés à l'origine :
  - le rayon de o est inférieur à celui de o'

Correction de l'ordre abstrait par rapport à l'ordre concret

Définition ($\sqsubseteq^\sharp$ abstrait correctement $\sqsubseteq$)

$$\forall o, o', \quad o \sqsubseteq^\sharp o' \Rightarrow \text{concrétisation}(o) \sqsubseteq \text{concrétisation}(o')$$

Exemple (cercles centrés à l'origine)



Notations

Définition (domaine abstrait $\mathcal{D}^\sharp$)

Un domaine abstrait spécifie :

- un ensemble $\mathcal{D}^\sharp$ d'éléments abstraits ;
- des opérations abstraites représentant dans l'abstrait l'utilisation des opérations concrètes dans le concret $\mathcal{D}$.

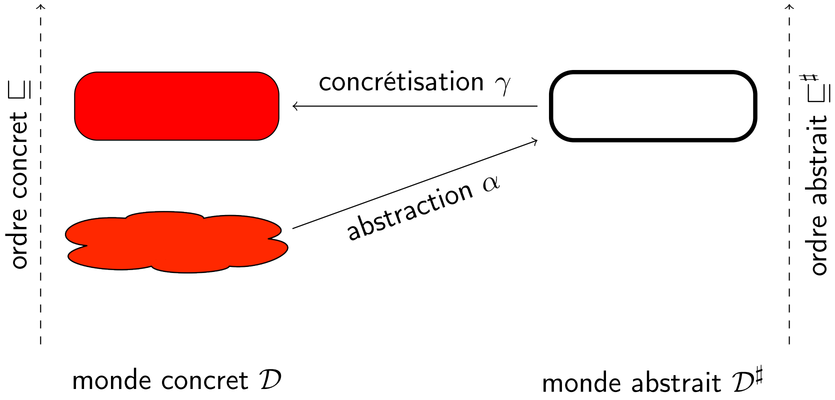
Définition (abstraction α)

Une fonction d'abstraction α associe à chaque objet concret o un objet abstrait $o^\sharp$, simplification de o.

Définition (concrétisation γ)

Une fonction de concrétisation γ associe à chaque objet abstrait $o^\sharp$ le plus grand objet concret o qu'il approxime.

Concret — Abstrait : résumé

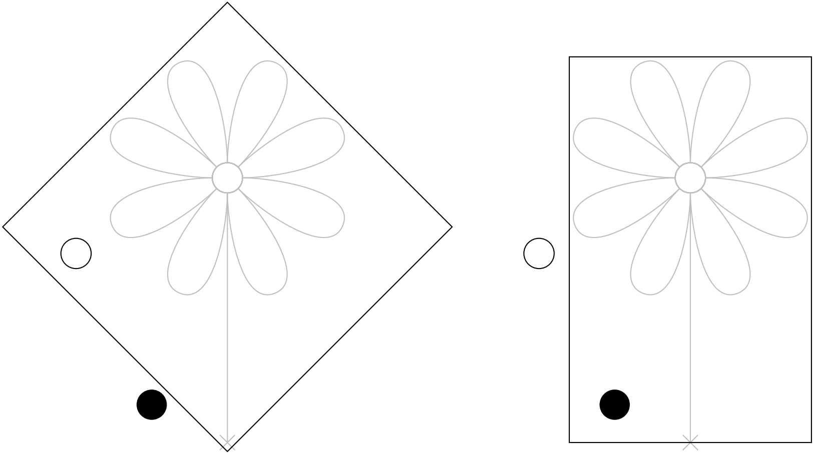


Un exemple graphique
Un peu de dessin
Notion de point fixe
Notion d'abstraction
Meilleure abstraction
Opérations abstraites

Une approche plus... langage
Syntaxe
Sémantique
Ordres partiels

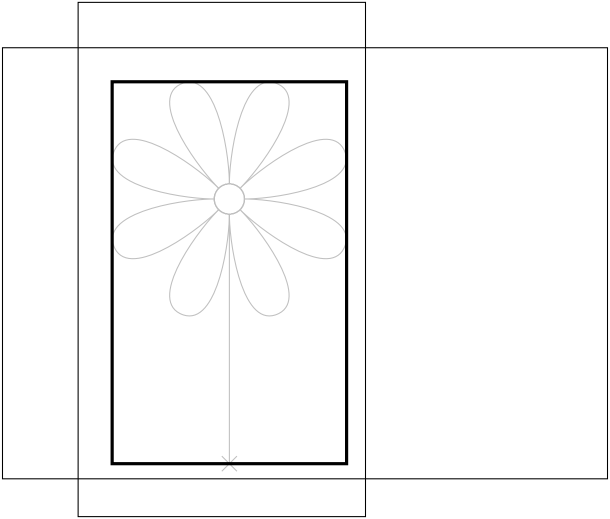
Comparaison d'abstractions

Abstractions différentes : pas toujours comparables



Meilleure abstraction : exemple

Si on n'autorise que les rectangles parallèles au bord,
on a une *meilleure* (i.e. plus petite) abstraction



Meilleure abstraction : définition

Définition

Un objet o a une *meilleure abstraction* dans $\mathcal{D}^\sharp$
si l'ensemble des objets abstraits $o^\sharp \in \mathcal{D}^\sharp$ qui l'approximent
$\left\{ o^\sharp \in \mathcal{D}^\sharp \;\middle|\; o \sqsubseteq \gamma(o^\sharp) \right\}$ a un minimum.

Meilleure abstraction : définition

Définition

Un objet o a une *meilleure abstraction* dans $\mathcal{D}^\sharp$
si l'ensemble des objets abstraits $o^\sharp \in \mathcal{D}^\sharp$ qui l'approximent
$\left\{o^\sharp \in \mathcal{D}^\sharp \,\middle|\, o \sqsubseteq \gamma(o^\sharp)\right\}$ a un minimum.

Exemple

- La fleur n'a pas de meilleure abstraction
  dans le monde des rectangles quelconques.
- La fleur a une meilleure abstraction
  dans le monde des rectangles parallèles au bord.

C.f. deux slides précédentes.

Meilleure abstraction : exemples

Exemple

- Tout objet a une meilleure abstraction
  dans le monde des contours (et c'est le contour).

Meilleure abstraction : exemples

Exemple

- Tout objet a une meilleure abstraction
  dans le monde des contours (et c'est le contour).
- Tout objet a une meilleure abstraction
  dans le monde des cercles centrés à l'origine
  (et c'est le cercle circonscrit).

Meilleure abstraction : exemples

Exemple

- Tout objet a une meilleure abstraction
  dans le monde des contours (et c'est le contour).
- Tout objet a une meilleure abstraction
  dans le monde des cercles centrés à l'origine
  (et c'est le cercle circonscrit).
- Certains objets n'ont pas de meilleure abstraction
  dans le monde des polygones convexes (ex. un cercle).

Correspondance de Galois

Définition

(α, γ) forme une *correspondance de Galois* entre $\mathcal{D}$ et $\mathcal{D}^\sharp$ si :

$$\forall x \in \mathcal{D}, \forall y \in \mathcal{D}^\sharp, \quad \alpha(x) \sqsubseteq^\sharp y \Leftrightarrow x \sqsubseteq \gamma(y).$$

Correspondance de Galois

Définition

(α, γ) forme une *correspondance de Galois* entre $\mathcal{D}$ et $\mathcal{D}^\sharp$ si :

$$\forall x \in \mathcal{D}, \forall y \in \mathcal{D}^\sharp, \quad \alpha(x) \sqsubseteq^\sharp y \Leftrightarrow x \sqsubseteq \gamma(y).$$

Exemple

- (contour, remplissage) est une correspondance de Galois entre le monde concret et le monde des contours.

Correspondance de Galois

Définition

(α, γ) forme une *correspondance de Galois* entre $\mathcal{D}$ et $\mathcal{D}^\sharp$ si :

$$\forall x \in \mathcal{D}, \forall y \in \mathcal{D}^\sharp, \quad \alpha(x) \sqsubseteq^\sharp y \Leftrightarrow x \sqsubseteq \gamma(y).$$

Exemple

- (contour, remplissage) est une correspondance de Galois entre le monde concret et le monde des contours.
- (cercle circonscrit, remplissage) est une correspondance de Galois entre le monde concret et le monde des cercles.

Correspondance de Galois

Définition

(α, γ) forme une *correspondance de Galois* entre $\mathcal{D}$ et $\mathcal{D}^\sharp$ si :

$$\forall x \in \mathcal{D}, \forall y \in \mathcal{D}^\sharp, \quad \alpha(x) \sqsubseteq^\sharp y \Leftrightarrow x \sqsubseteq \gamma(y).$$

Exemple

- (contour, remplissage) est une correspondance de Galois entre le monde concret et le monde des contours.
- (cercle circonscrit, remplissage) est une correspondance de Galois entre le monde concret et le monde des cercles.
- Il n'existe pas de correspondance de Galois entre le monde concret et le monde des polygones convexes.

Correspondance de Galois et meilleure abstraction

Théorème

Il existe une correspondance de Galois (α, γ) entre $\mathcal{D}$ et $\mathcal{D}^\sharp$
si et seulement si
tout objet de $\mathcal{D}$ a une meilleure abstraction dans $\mathcal{D}^\sharp$
(et la meilleure abstraction est alors donnée par α).

Un exemple graphique
- Un peu de dessin
- Notion de point fixe
- Notion d'abstraction
- Meilleure abstraction
- Opérations abstraites

Une approche plus... langage
- Syntaxe
- Sémantique
- Ordres partiels

Opérations abstraites et leur correction

Pour manipuler les objets abstraits, on a besoin d'opérations
const$^\sharp : \mathcal{D}^\sharp$, *unaire*$^\sharp : \mathcal{D}^\sharp \to \mathcal{D}^\sharp$ ou *binaire*$^\sharp : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$
alter ego des opérations concrètes
const (ex. pétale), *unaire* (ex. rotation) ou *binaire* (ex. union).

Opérations abstraites et leur correction

Pour manipuler les objets abstraits, on a besoin d'opérations $\mathit{const}^\sharp : \mathcal{D}^\sharp$, $\mathit{unaire}^\sharp : \mathcal{D}^\sharp \to \mathcal{D}^\sharp$ ou $\mathit{binaire}^\sharp : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$
alter ego des opérations concrètes
const (ex. pétale), *unaire* (ex. rotation) ou *binaire* (ex. union).

Définition (correction des opérations abstraites)

- $\mathit{const} \sqsubseteq \gamma(\mathit{const}^\sharp)$

Opérations abstraites et leur correction

Pour manipuler les objets abstraits, on a besoin d'opérations
$\mathit{const}^\sharp : \mathcal{D}^\sharp$, $\mathit{unaire}^\sharp : \mathcal{D}^\sharp \to \mathcal{D}^\sharp$ ou $\mathit{binaire}^\sharp : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$
alter ego des opérations concrètes
const (ex. pétale), *unaire* (ex. rotation) ou *binaire* (ex. union).

Définition (correction des opérations abstraites)

- $\mathit{const} \sqsubseteq \gamma(\mathit{const}^\sharp)$
- $\forall x \in \mathcal{D}^\sharp, \quad \mathit{unaire}(\gamma(x)) \sqsubseteq \gamma(\mathit{unaire}^\sharp(x))$

Opérations abstraites et leur correction

Pour manipuler les objets abstraits, on a besoin d'opérations
$\mathit{const}^\sharp : \mathcal{D}^\sharp$, $\mathit{unaire}^\sharp : \mathcal{D}^\sharp \to \mathcal{D}^\sharp$ ou $\mathit{binaire}^\sharp : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$
alter ego des opérations concrètes
const (ex. pétale), *unaire* (ex. rotation) ou *binaire* (ex. union).

Définition (correction des opérations abstraites)

- $\mathit{const} \sqsubseteq \gamma(\mathit{const}^\sharp)$
- $\forall x \in \mathcal{D}^\sharp, \quad \mathit{unaire}(\gamma(x)) \sqsubseteq \gamma(\mathit{unaire}^\sharp(x))$
- $\forall x, y \in \mathcal{D}^\sharp, \quad \mathit{binaire}(\gamma(x), \gamma(y)) \sqsubseteq \gamma(\mathit{binaire}^\sharp(x, y))$

Opérations abstraites et leur correction

Pour manipuler les objets abstraits, on a besoin d'opérations
$\mathit{const}^\sharp : \mathcal{D}^\sharp$, $\mathit{unaire}^\sharp : \mathcal{D}^\sharp \to \mathcal{D}^\sharp$ ou $\mathit{binaire}^\sharp : (\mathcal{D}^\sharp \times \mathcal{D}^\sharp) \to \mathcal{D}^\sharp$
alter ego des opérations concrètes
const (ex. pétale), *unaire* (ex. rotation) ou *binaire* (ex. union).

Définition (correction des opérations abstraites)

- $\mathit{const} \sqsubseteq \gamma(\mathit{const}^\sharp)$
- $\forall x \in \mathcal{D}^\sharp, \quad \mathit{unaire}(\gamma(x)) \sqsubseteq \gamma(\mathit{unaire}^\sharp(x))$
- $\forall x, y \in \mathcal{D}^\sharp, \quad \mathit{binaire}(\gamma(x), \gamma(y)) \sqsubseteq \gamma(\mathit{binaire}^\sharp(x, y))$

Très important (les opérations abstraites n'ont aucun sens sinon).

Illustration de la correction

Dans le monde des rectangles *parallèles au bord*.

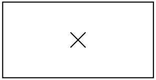


Illustration de la correction

Dans le monde des rectangles *parallèles au bord*.

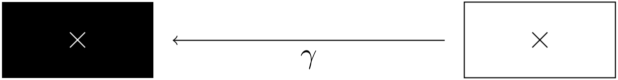


Illustration de la correction

Dans le monde des rectangles *parallèles au bord*.

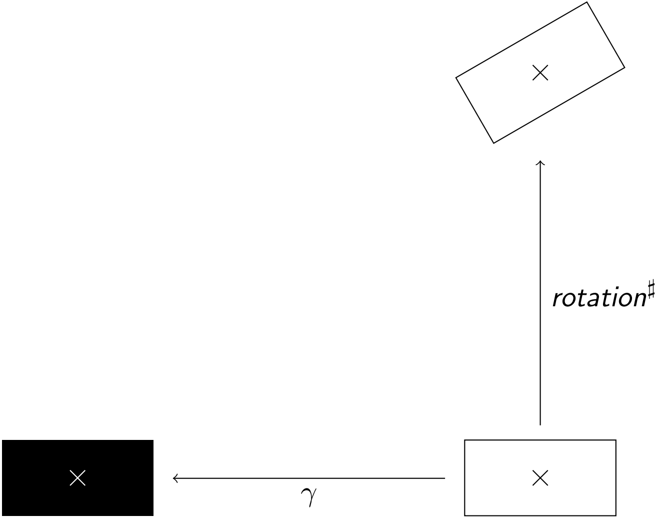


Illustration de la correction

Dans le monde des rectangles *parallèles au bord*.

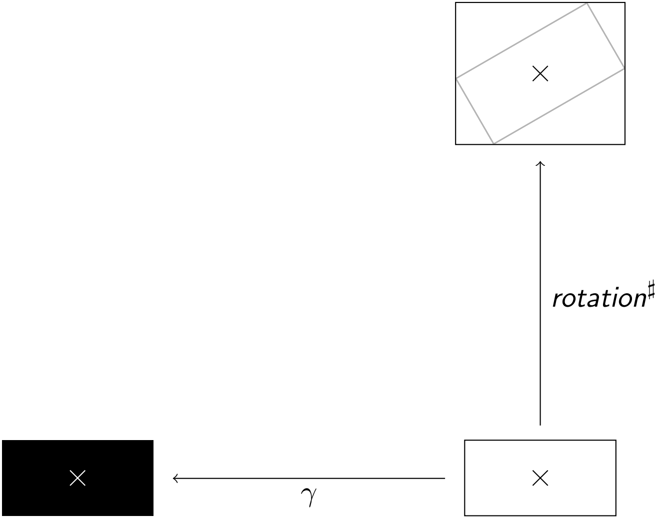


Illustration de la correction

Dans le monde des rectangles *parallèles au bord*.

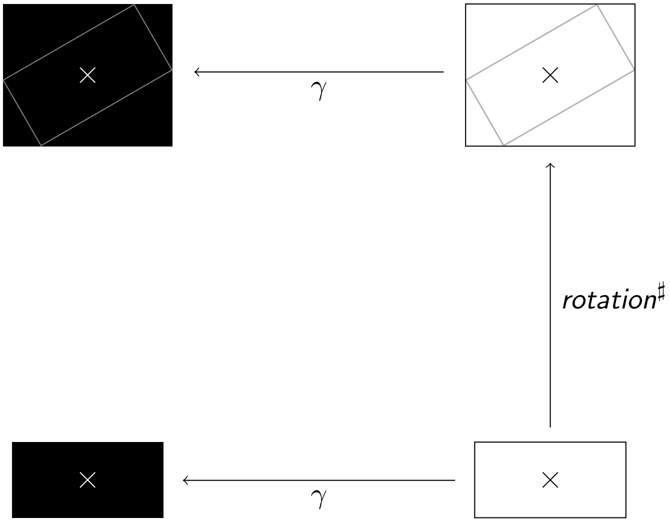
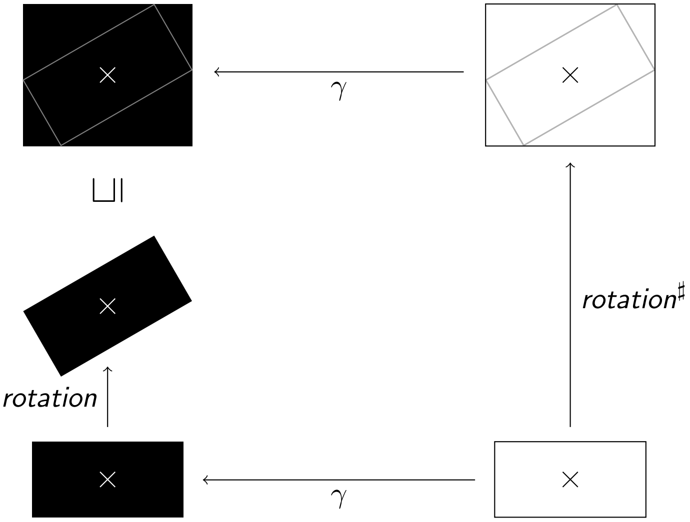


Illustration de la correction

Dans le monde des rectangles *parallèles au bord*.



On a bien $rotation(\gamma(.)) \sqsubseteq \gamma(rotation^\sharp(.))$.

En présence d'une meilleure abstraction

Les meilleures opérations abstraites sont définies par :

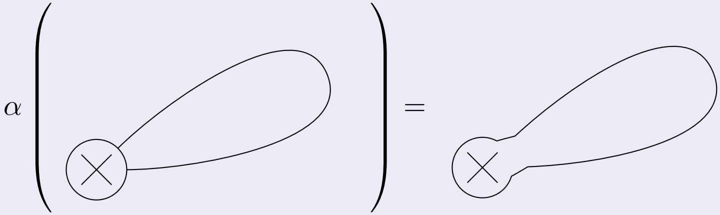
- $const^\sharp = \alpha(const)$
- $unaire^\sharp(x) = \alpha(unaire(\gamma(x)))$
- $binaire^\sharp(x, y) = \alpha(binaire(\gamma(x), \gamma(y)))$
- $\ldots$

Pétale abstrait

On reprend l'abstraction « contours ».

Définition

$const^\sharp = \alpha(const)$:



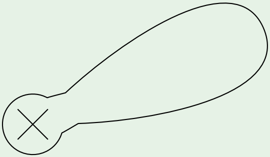
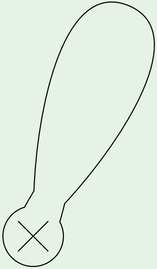
Rotation abstraite

Définition

$unaire^\sharp(.) = \alpha(unaire(\gamma(.))$:

$$\mathrm{r}^\sharp[a](x) = \alpha(\mathrm{r}[a](\gamma(x))) = \mathrm{r}[a](x)$$

Exemple

$\mathrm{r}^\sharp[45]$ () $=$ 

Remarque : l'opération est *exacte*

$$\mathrm{r}[a](\gamma(x)) = \gamma(\mathrm{r}^\sharp[a](x))$$

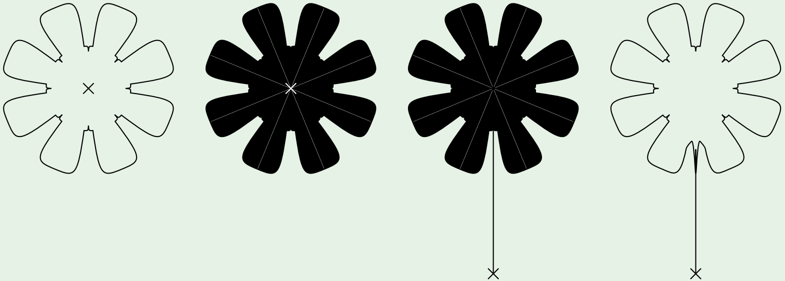
Tige abstraite

Définition

$unaire^\sharp(.) = \alpha(unaire(\gamma(.))$:

$$\mathrm{tige}^\sharp[a](x) = \alpha(\mathrm{tige}[a](\gamma(x)))$$

Exemple



corolle abstraite

γ(corolle abstraite)

tige(γ(corolle abstraite))

α(tige(γ(corolle abstraite)))

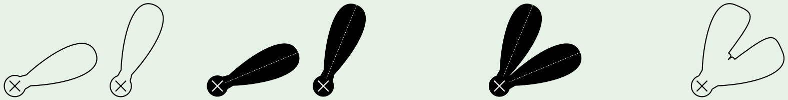
Union abstraite

Définition

$$binaire^{\sharp}(.,.) = \alpha(binaire(\gamma(.), \gamma(.)) :$$

$$x \sqcup^{\sharp} y = \alpha(\gamma(x) \cup \gamma(y))$$

Exemple

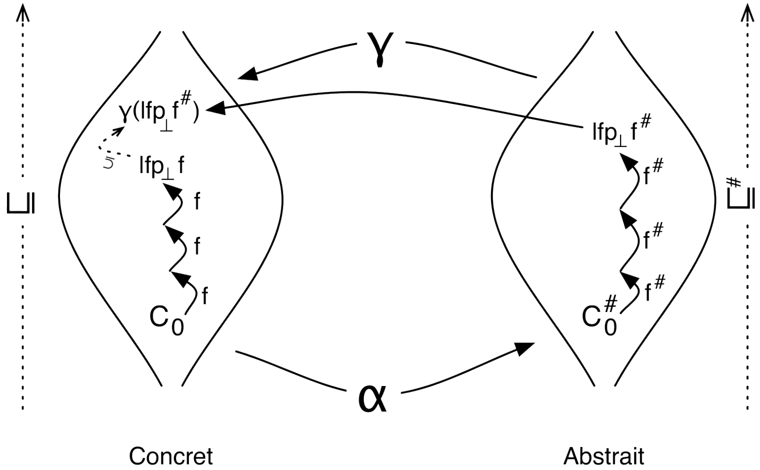


p_1, p_2 $\quad$ $\gamma(p_1), \gamma(p_2)$ $\quad$ $\gamma(p_1) \cup \gamma(p_2)$ $\quad$ $\alpha(\gamma(p_1) \cup \gamma(p_2))$

Point fixe abstrait pour la corolle

- On avait défini : corolle $= \mathrm{lfp} F$
  avec $F : X \mapsto$ pétale $\sqcup \mathrm{r}[45](X)$.
- On définit : corolle abstraite $= \mathrm{lfp} F^\sharp$
  avec $F^\sharp : X \mapsto$ pétale abstrait $\sqcup^\sharp \mathrm{r}^\sharp[45](X)$.
- Toutes les opérations élémentaires sont correctes
  donc par construction la corolle abstraite approxime bien
  la vraie corolle (corolle $\sqsubseteq \gamma$(corolle abstraite)).

Cadre général de l'interprétation abstraite



Un exemple graphique
- Un peu de dessin
- Notion de point fixe
- Notion d'abstraction
- Meilleure abstraction
- Opérations abstraites

Une approche plus... langage
- Syntaxe
- Sémantique
- Ordres partiels

Un langage jouet

Syntaxe

$$
\begin{aligned}
\mathit{stm} ::=\ & v = \mathit{expr}\ \mathbf{;}\ \mid\ \mathit{stm}\ \mathit{stm} \\
& \mid\ \mathbf{if}\ \mathbf{(}\mathit{expr} > 0\mathbf{)}\ \mathbf{\{}\ \mathit{stm}\ \mathbf{\}}\ \mathbf{else}\ \mathbf{\{}\ \mathit{stm}\ \mathbf{\}} \\
& \mid\ \mathbf{while}\ \mathbf{(}\mathit{expr} > 0\mathbf{)}\ \mathbf{\{}\ \mathit{stm}\ \mathbf{\}}
\end{aligned}
$$

$$
\begin{aligned}
\mathit{expr} ::=\ & v \mid n \mid \mathbf{rand(}n,\ n\mathbf{)} \\
& \mid \mathit{expr} + \mathit{expr} \mid \mathit{expr} - \mathit{expr} \mid \mathit{expr} \times \mathit{expr} \mid \mathit{expr} / \mathit{expr}
\end{aligned}
$$

$v \in \mathbb{V}$, un ensemble de variables

$n \in \mathbb{Z}$ (on ne manipule que des entiers)

$\mathbf{rand(}n_1,\ n_2\mathbf{)}$ représente le choix aléatoire d'un entier entre n_1 et n_2 (sert à simuler une entrée).

. Suite très inspirée du cours de A. Miné au MPRI.

Un langage jouet (suite et fin)

Exemple

$x = \textbf{rand}(0, 12)\,;\ y = 42\,;$
while $(x > 0)$ **{**
$\quad x = x - 2\,;$
$\quad y = y + 4\,;$
}

Une exécution
(valeurs à l'entrée de la boucle) :

x	7	5	3	1	-1
y	42	46	50	54	58

Remarques

- un langage très simple, sans fonctions, sans...
- mais représentatif d'un langage impératif comme C
- dont c'est d'ailleurs un sous ensemble
- et on peut tout calculer (c'est Turing-complet)

Un exemple graphique

- Un peu de dessin
- Notion de point fixe
- Notion d'abstraction
- Meilleure abstraction
- Opérations abstraites

Une approche plus... langage

- Syntaxe
- Sémantique
- Ordres partiels

Graphe de flot de contrôle

On va utiliser les graphes de flot de contrôle des programmes

Définition

Un *graphe de flot de contrôle* (L, A) est composé d'un ensemble de points de programme L, d'un point d'entrée $0 \in L$ et d'arêtes $A \subseteq L \times com \times L$ avec :
$com ::= v = expr \mid expr > 0$

Graphe de flot de contrôle

On va utiliser les graphes de flot de contrôle des programmes

Définition

Un *graphe de flot de contrôle* (L, A) est composé d'un ensemble de points de programme L, d'un point d'entrée $0 \in L$ et d'arêtes $A \subseteq L \times com \times L$ avec :

$com ::= v = expr \mid expr > 0$

Exemple

$_0\mathsf{x} = \mathbf{rand}(0,\, 12)\,;_1\mathsf{y} = 42\,;$

$\mathbf{while}\ _2(\mathsf{x} > 0)\ \mathbf{\{}$

$\quad _3\mathsf{x} = \mathsf{x} - 2\,;$

$\quad _4\mathsf{y} = \mathsf{y} + 4\,;$

$\mathbf{\}}_5$

$$\begin{tikzcd}
 & & & 4 \arrow[dr, "\mathsf{y} = \mathsf{y} + 4"'] & & 3 \arrow[ll, "\mathsf{x} = \mathsf{x} - 2"'] & \\
0 \arrow[r, "\mathsf{x} = \mathbf{rand}(0,\, 12)"'] & 1 \arrow[rrr, "\mathsf{y} = 42"'] & & & 2 \arrow[ur, "\mathsf{x} > 0"'] \arrow[rr, "\mathsf{x} \leqslant 0"'] & & 5
\end{tikzcd}$$

Sémantique concrète, expressions

Sémantique des expressions : $[\![e]\!]_{\mathrm{E}} : (\mathbb{V} \to \mathbb{Z}) \to \mathcal{P}(\mathbb{Z})$

$$
\begin{array}{ll}
[\![v]\!]_{\mathrm{E}}(\rho) & = \{\rho(v)\} \\
[\![n]\!]_{\mathrm{E}}(\rho) & = \{n\} \\
[\![\mathbf{rand}(n_1, n_2)]\!]_{\mathrm{E}}(\rho) & = \{n \in \mathbb{Z} \mid n_1 \leqslant n \leqslant n_2\} \\
[\![e_1 + e_2]\!]_{\mathrm{E}}(\rho) & = \{n_1 + n_2 \mid n_1 \in [\![e_1]\!]_{\mathrm{E}}(\rho) \wedge n_2 \in [\![e_2]\!]_{\mathrm{E}}(\rho)\} \\
\ldots &
\end{array}
$$

Sémantique concrète, expressions

Sémantique des expressions : $[\![e]\!]_{\mathrm{E}} : (\mathbb{V} \to \mathbb{Z}) \to \mathcal{P}(\mathbb{Z})$

$$
\begin{array}{ll}
[\![v]\!]_{\mathrm{E}}(\rho) & = \{\rho(v)\} \\
[\![n]\!]_{\mathrm{E}}(\rho) & = \{n\} \\
[\![\mathbf{rand}(n_1, n_2)]\!]_{\mathrm{E}}(\rho) & = \{n \in \mathbb{Z} \mid n_1 \leqslant n \leqslant n_2\} \\
[\![e_1 + e_2]\!]_{\mathrm{E}}(\rho) & = \{n_1 + n_2 \mid n_1 \in [\![e_1]\!]_{\mathrm{E}}(\rho) \wedge n_2 \in [\![e_2]\!]_{\mathrm{E}}(\rho)\} \\
\ldots &
\end{array}
$$

Remarque : environnement

On nomme généralement *environnement* les fonctions $\rho : \mathbb{V} \to \mathbb{Z}$ qui associent une valeur à chaque variable.

Sémantique concrète, expressions (suite et fin)

Remarque : cas d'erreur

On peut rencontrer deux types d'erreur à l'exécution :

- **rand**(n_1, n_2) avec $n1 > n2$:
  $[\![\textbf{rand}(n_1, n_2)]\!]_{\mathrm{E}} = \{x \in \mathbb{Z} \mid n_1 \leqslant x \leqslant n_2\} = \emptyset$;

Sémantique concrète, expressions (suite et fin)

Remarque : cas d'erreur

On peut rencontrer deux types d'erreur à l'exécution :

- **rand**(n_1, n_2) avec $n1 > n2$:
  $[\![\textbf{rand}(n_1, n_2)]\!]_{\mathrm{E}} = \{x \in \mathbb{Z} \mid n_1 \leqslant x \leqslant n_2\} = \emptyset$;
- division par zéro : $[\![e/0]\!]_{\mathrm{E}} = \emptyset$.

Sémantique concrète, expressions (suite et fin)

Remarque : cas d'erreur

On peut rencontrer deux types d'erreur à l'exécution :

- **rand**(n_1, n_2) avec $n1 > n2$:
  $[\![\textbf{rand}(n_1, n_2)]\!]_{\mathrm{E}} = \{x \in \mathbb{Z} \mid n_1 \leqslant x \leqslant n_2\} = \emptyset$;
- division par zéro : $[\![e/0]\!]_{\mathrm{E}} = \emptyset$.

On suppose donc que le programme lève une exception
et abandonne son exécution dans ces deux cas.

Sémantique concrète, commandes

Sémantique des commandes : $[\![c]\!]_{\mathrm{C}} : \mathcal{P}(\mathbb{V} \to \mathbb{Z}) \to \mathcal{P}(\mathbb{V} \to \mathbb{Z})$

$$[\![v = e]\!]_{\mathrm{C}}(R) = \{\rho[v \mapsto n] \mid \rho \in R, n \in [\![e]\!]_{\mathrm{E}}(\rho)\}$$
$$[\![e > 0]\!]_{\mathrm{C}}(R) = \{\rho \mid \rho \in R, \exists n \in [\![e]\!]_{\mathrm{E}}(\rho), n > 0\}$$

Sémantique concrète, commandes

Sémantique des commandes : $\llbracket c \rrbracket_{\mathrm{C}} : \mathcal{P}(\mathbb{V} \to \mathbb{Z}) \to \mathcal{P}(\mathbb{V} \to \mathbb{Z})$

$$\llbracket v = e \rrbracket_{\mathrm{C}}(R) = \{\rho[v \mapsto n] \mid \rho \in R, n \in \llbracket e \rrbracket_{\mathrm{E}}(\rho)\}$$
$$\llbracket e > 0 \rrbracket_{\mathrm{C}}(R) = \{\rho \mid \rho \in R, \exists n \in \llbracket e \rrbracket_{\mathrm{E}}(\rho), n > 0\}$$

Remarque : $e \leqslant 0$

$e \leqslant 0$ n'est qu'une jolie façon d'écrire $1 - e > 0$ (sucre syntaxique).

Sémantique concrète, programme

Sémantique des programmes : $[\![(L, A)]\!] : L \to \mathcal{P}(\mathbb{V} \to \mathbb{Z})$

À chaque point de programme, on associe le meilleur invariant.

Sémantique concrète, programme

Sémantique des programmes : $\llbracket (L, A) \rrbracket : L \to \mathcal{P}(\mathbb{V} \to \mathbb{Z})$

À chaque point de programme, on associe le meilleur invariant.

C'est la plus petite solution (au sens de l'inclusion $\subseteq$) du système

$$\begin{cases} R_0 = \mathbb{V} \to \mathbb{Z} & \\ R_{l'} = \displaystyle\bigcup_{(l,c,l') \in A} \llbracket c \rrbracket_{\mathrm{C}} (R_l) & l' \neq 0 \end{cases}$$

Sémantique concrète, programme

Sémantique des programmes : $[\![(L, A)]\!] : L \to \mathcal{P}(\mathbb{V} \to \mathbb{Z})$

À chaque point de programme, on associe le meilleur invariant.

C'est la plus petite solution (au sens de l'inclusion $\subseteq$) du système

$$\left\{ \begin{array}{ll} R_0 = \mathbb{V} \to \mathbb{Z} & \\ R_{l'} = \displaystyle\bigcup_{(l,c,l') \in A} [\![c]\!]_{\mathrm{C}} \left(R_l \right) & \qquad l' \neq 0 \end{array} \right.$$

Une telle solution existe toujours
d'après le théorème de Knaster-Tarski...

Exemple

$_0$x = **rand**(0, 12) ; $_1$y = 42 ;

while $_2$(x > 0) **{**

  $_3$x = x − 2 ;

  $_4$y = y + 4 ;

}$_5$

$$\begin{tikzcd}
 & & & 4 \arrow[dr, "y = y + 4"'] & & 3 \arrow[ll, "x = x - 2"'] & \\
0 \arrow[rr, "x = \mathbf{rand}(0,\ 12)"'] & & 1 \arrow[rr, "y = 42"'] & & 2 \arrow[ur, "x > 0"'] \arrow[rr, "x \leqslant 0"'] & & 5
\end{tikzcd}$$

équations

$R_0 = \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$

Exemple

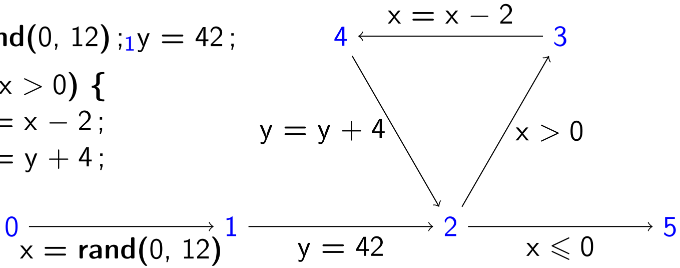
$_0$x = **rand**(0, 12) ;$_1$y = 42 ;

while $_2$(x > 0) **{**

  $_3$x = x − 2 ;

  $_4$y = y + 4 ;

}$_5$



équations

$R_0 = \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$

$R_1 = \{\, x \in [\![0, 12]\!], y \in \mathbb{Z} \,\}$

Exemple

$_{0}\mathsf{x} = \mathbf{rand}(0,\ 12)\ ;_{1}\mathsf{y} = 42\ ;$

$\mathbf{while}\ _{2}(\mathsf{x} > 0)\ \mathbf{\{}$

$\quad _{3}\mathsf{x} = \mathsf{x} - 2\ ;$

$\quad _{4}\mathsf{y} = \mathsf{y} + 4\ ;$

$\mathbf{\}}_{5}$

$$
\begin{tikzcd}
 & & 4 \arrow[dr, "\mathsf{y} = \mathsf{y} + 4"'] & & 3 \arrow[ll, "\mathsf{x} = \mathsf{x} - 2"'] & \\
0 \arrow[r, "\mathsf{x} = \mathbf{rand}(0,\ 12)"'] & 1 \arrow[rr, "\mathsf{y} = 42"'] & & 2 \arrow[ur, "\mathsf{x} > 0"'] \arrow[rr, "\mathsf{x} \leqslant 0"'] & & 5
\end{tikzcd}
$$

équations

$R_0 = \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$

$R_1 = \{\, x \in [\![0, 12]\!], y \in \mathbb{Z} \,\}$

$R_2 = R_1[y \mapsto 42] \cup R_4[y \mapsto y + 4]$

Exemple

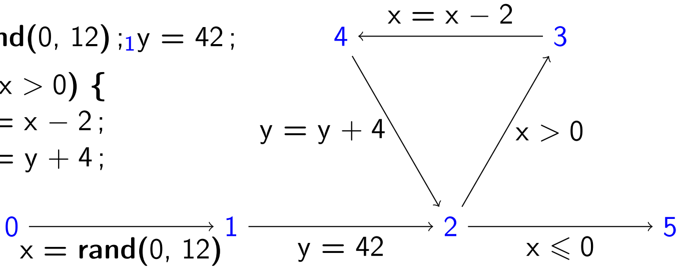
$_0$x = **rand**(0, 12) ;$_1$y = 42 ;

while $_2$(x > 0) **{**

  $_3$x = x − 2 ;

  $_4$y = y + 4 ;

}$_5$



équations

$R_0 = \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$
$R_1 = \{\, x \in [\![0, 12]\!], y \in \mathbb{Z} \,\}$
$R_2 = R_1[y \mapsto 42] \cup R_4[y \mapsto y + 4]$

$R_3 = R_2 \cap \{\, x > 0, y \in \mathbb{Z} \,\}$

Exemple

$_{0}\mathsf{x} = \mathbf{rand}(0,\ 12)\ ;_{1}\mathsf{y} = 42\ ;$

$\mathbf{while}\ _{2}(\mathsf{x} > 0)\ \{$

$\quad _{3}\mathsf{x} = \mathsf{x} - 2\ ;$

$\quad _{4}\mathsf{y} = \mathsf{y} + 4\ ;$

$\}_{5}$

$$
\begin{tikzcd}
 & & & 4 \arrow[dr, "y = y + 4"'] & & 3 \arrow[ll, "x = x - 2"'] & \\
0 \arrow[rr, "x = \mathbf{rand}(0,\ 12)"'] & & 1 \arrow[rr, "y = 42"'] & & 2 \arrow[ur, "x > 0"'] \arrow[rr, "x \leqslant 0"'] & & 5
\end{tikzcd}
$$

équations

$$R_0 = \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$$
$$R_1 = \{\, x \in [\![0, 12]\!], y \in \mathbb{Z} \,\}$$
$$R_2 = R_1[y \mapsto 42] \cup R_4[y \mapsto y + 4]$$

$$R_3 = R_2 \cap \{\, x > 0, y \in \mathbb{Z} \,\}$$

$$R_4 = R_3[x \mapsto x - 2]$$

Exemple

$_0\mathsf{x} = \mathbf{rand}(0,\ 12)\ ;_1\mathsf{y} = 42\ ;$

$\mathbf{while}\ _2(\mathsf{x} > 0)\ \mathbf{\{}$

$\quad _3\mathsf{x} = \mathsf{x} - 2\ ;$

$\quad _4\mathsf{y} = \mathsf{y} + 4\ ;$

$\mathbf{\}}_5$

$$
\begin{tikzcd}
 & & 4 \arrow[dr, "\mathsf{y} = \mathsf{y} + 4"'] & & 3 \arrow[ll, "\mathsf{x} = \mathsf{x} - 2"'] \\
0 \arrow[r, "\mathsf{x} = \mathbf{rand}(0,\ 12)"'] & 1 \arrow[rr, "\mathsf{y} = 42"'] & & 2 \arrow[ur, "\mathsf{x} > 0"'] \arrow[r, "\mathsf{x} \leqslant 0"'] & 5
\end{tikzcd}
$$

équations

$R_0 = \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$

$R_1 = \{\, x \in [\![0, 12]\!], y \in \mathbb{Z} \,\}$

$R_2 = R_1[y \mapsto 42] \cup R_4[y \mapsto y + 4]$

$R_3 = R_2 \cap \{\, x > 0, y \in \mathbb{Z} \,\}$

$R_4 = R_3[x \mapsto x - 2]$

$R_5 = R_2 \cap \{\, x \leqslant 0, y \in \mathbb{Z} \,\}$

Exemple

$_{0}\mathrm{x} = \textbf{rand}(0,\ 12)\,;_{1}\mathrm{y} = 42\,;$

$\textbf{while}\ _{2}(\mathrm{x} > 0)\ \{$

$\quad _{3}\mathrm{x} = \mathrm{x} - 2\,;$

$\quad _{4}\mathrm{y} = \mathrm{y} + 4\,;$

$\}_{5}$

$$
\begin{tikzcd}
 & & 4 \arrow[dr, "\mathrm{y} = \mathrm{y} + 4"'] & & 3 \arrow[ll, "\mathrm{x} = \mathrm{x} - 2"'] & \\
0 \arrow[r, "\mathrm{x} = \textbf{rand}(0,\ 12)"'] & 1 \arrow[rr, "\mathrm{y} = 42"'] & & 2 \arrow[ur, "\mathrm{x} > 0"'] \arrow[rr, "\mathrm{x} \leqslant 0"'] & & 5
\end{tikzcd}
$$

équations	plus petite solution
$R_0 = \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$	$= \{\, x \in \mathbb{Z}, y \in \mathbb{Z} \,\}$
$R_1 = \{\, x \in [\![0, 12]\!], y \in \mathbb{Z} \,\}$	$= \{\, x \in [\![0, 12]\!], y \in \mathbb{Z} \,\}$
$R_2 = R_1[y \mapsto 42] \cup R_4[y \mapsto y + 4]$	$= \{x \in [\![-1, 12]\!], y \in [\![42, 66]\!] \cap 4\mathbb{Z} + 2 \mid 2x + y \in [\![42, 66]\!]\}$
$R_3 = R_2 \cap \{\, x > 0, y \in \mathbb{Z} \,\}$	$= \{x \in [\![1, 12]\!], y \in [\![42, 66]\!] \cap 4\mathbb{Z} + 2 \mid 2x + y \in [\![42, 66]\!]\}$
$R_4 = R_3[x \mapsto x - 2]$	$= \{x \in [\![-1, 10]\!], y \in [\![42, 66]\!] \cap 4\mathbb{Z} + 2 \mid 2x + y \in [\![38, 62]\!]\}$
$R_5 = R_2 \cap \{\, x \leqslant 0, y \in \mathbb{Z} \,\}$	$= \{x \in [\![-1, 0]\!], y \in [\![42, 66]\!] \cap 4\mathbb{Z} + 2 \mid 2x + y \in [\![42, 66]\!]\}$

Un exemple graphique
- Un peu de dessin
- Notion de point fixe
- Notion d'abstraction
- Meilleure abstraction
- Opérations abstraites

Une approche plus... langage
- Syntaxe
- Sémantique
- Ordres partiels

Rappels

Définition (ordre)

Un *ordre* $\sqsubseteq$ est une relation binaire

- réflexive ($\forall x, x \sqsubseteq x$) ;
- transitive ($\forall x, y, z, (x \sqsubseteq y \wedge y \sqsubseteq z) \Rightarrow x \sqsubseteq z$) ;
- antisymétrique ($\forall x, y, (x \sqsubseteq y \wedge y \sqsubseteq x) \Rightarrow x = y$).

Définition (borne supérieure)

Une *borne supérieure* $\bigsqcup : \mathcal{P}(S) \to S$ associe à tout sous ensemble S' de S son plus petit majorant

- $\forall x \in S', x \sqsubseteq \bigsqcup S'$
- $\forall y \in S, (\forall x \in S', x \sqsubseteq y) \Rightarrow \bigsqcup S' \sqsubseteq y$

Treillis complet

Définition (treillis complet)

Un ensemble S muni d'un ordre $\sqsubseteq$ est un *treillis complet* s'il admet une borne supérieure $\bigsqcup S'$.

Un treillis complet est automatiquement muni

- d'une borne inférieure (plus grand minorant) : $\bigsqcap S' = \bigsqcup \{x \mid \forall y \in S', x \sqsubseteq y\}$;
- d'un plus petit élément (bottom) : $\bot = \bigsqcup \emptyset = \bigsqcap S$;
- d'un plus grand élément (top) : $\top = \bigsqcup S = \bigsqcap \emptyset$.

Treillis complet, exemples

Exemple

$\mathbb{Z}$ n'est pas un treillis complet ($\bigsqcup \mathbb{Z}$ n'existe pas).

Treillis complet, exemples

Exemple

$\mathbb{Z}$ n'est pas un treillis complet ($\bigsqcup \mathbb{Z}$ n'existe pas).

Exemple

$\bar{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$ est un treillis complet.

Treillis complet, exemples

Exemple

$\mathbb{Z}$ n'est pas un treillis complet ($\bigsqcup \mathbb{Z}$ n'existe pas).

Exemple

$\bar{\mathbb{Z}} = \mathbb{Z} \cup \{-\infty, +\infty\}$ est un treillis complet.

Exercice

- Montrer que pour tout ensemble S, l'*ensemble de ses parties* $\mathcal{P}(S)$ muni de l'ordre inclusion $\subseteq$ est un treillis complet.
- À quoi correspondent la borne supérieure $\sqcup$?
la borne inférieure $\sqcap$? $\perp$ et $\top$?

Treillis complet, autres exemples

Exercice

Soit A un ensemble quelconque et $(B, \sqsubseteq_B)$ un treillis complet, montrer que $A \to B$, les *fonctions* de A dans B forment un treillis complet muni de l'ordre usuel sur les fonctions $f \sqsubseteq_{A \to B} g$ si pour tout $x \in A$, $f(x) \sqsubseteq_B g(x)$.

Treillis complet, autres exemples

Exercice

Soit A un ensemble quelconque et $(B, \sqsubseteq_B)$ un treillis complet, montrer que $A \to B$, les *fonctions* de A dans B forment un treillis complet muni de l'ordre usuel sur les fonctions $f \sqsubseteq_{A \to B} g$ si pour tout $x \in A$, $f(x) \sqsubseteq_B g(x)$.

Exemple

$$\begin{tikzcd}[column sep=small] & \{0, 1\} \arrow[dl, dash] \arrow[dr, dash] & \\ \{0\} & & \{1\} \end{tikzcd}$$

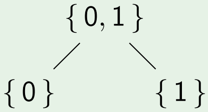
n'est pas un treillis complet ($\bigsqcup \emptyset$ n'existe pas).

Treillis complet, autres exemples

Exercice

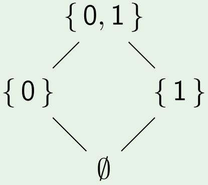
Soit A un ensemble quelconque et $(B, \sqsubseteq_B)$ un treillis complet, montrer que $A \to B$, les *fonctions* de A dans B forment un treillis complet muni de l'ordre usuel sur les fonctions $f \sqsubseteq_{A \to B} g$ si pour tout $x \in A$, $f(x) \sqsubseteq_B g(x)$.

Exemple



n'est pas un treillis complet ($\bigsqcup \emptyset$ n'existe pas).

Exemple



est un treillis complet (c.f. exercice du slide précédent).

Théorème de Knaster-Tarski

Définition

Une fonction f d'un treillis complet dans lui même est *monotone* si

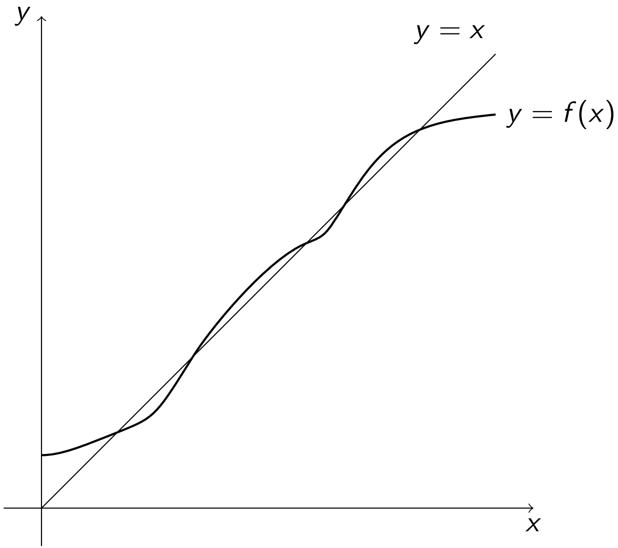
$$\forall x, y \in S, \quad x \sqsubseteq y \Rightarrow f(x) \sqsubseteq f(y)$$

Théorème

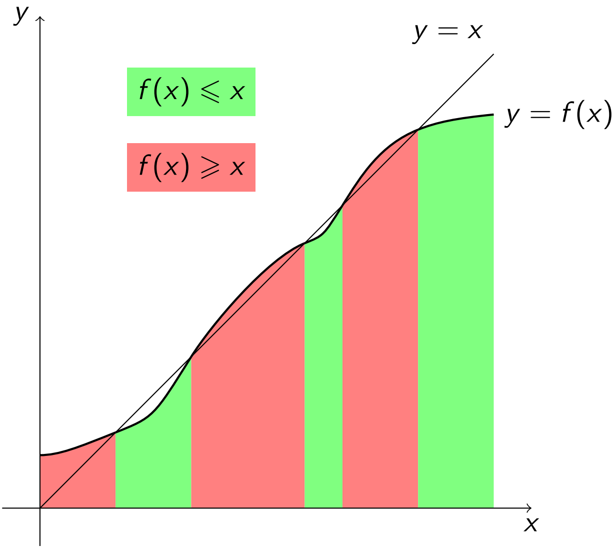
Si S est un treillis complet et f une fonction monotone sur ce treillis alors f admet un plus petit point fixe

$$\mathrm{lfp}\, f = \bigsqcap \{x \in S \mid f(x) \sqsubseteq x\}.$$

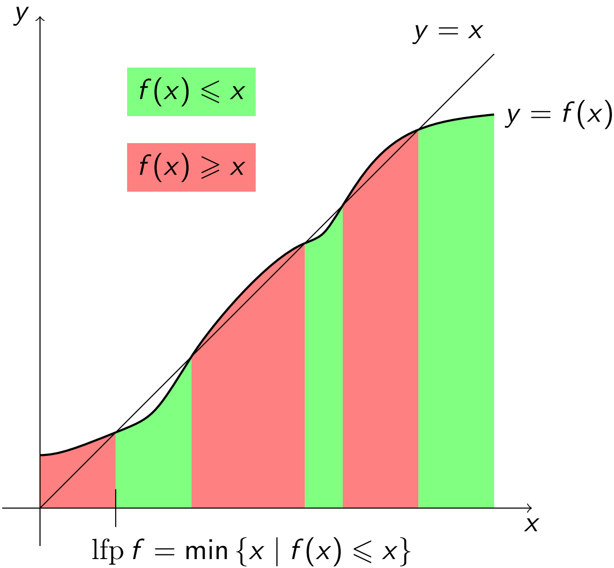
Théorème de Knaster-Tarski, illustration



Théorème de Knaster-Tarski, illustration



Théorème de Knaster-Tarski, illustration



Théorème de Knaster-Tarski, démonstration

Notons $P = \{x \in S \mid f(x) \sqsubseteq x\}$ et $p = \bigsqcap P$.

- p est un point fixe :
  - Soit $x \in P$ quelconque (P est non vide car $\top \in P$), $p \sqsubseteq x$ donc par croissance de f, $f(p) \sqsubseteq f(x)$ et $f(x) \sqsubseteq x$ car $x \in P$ donc $f(p) \sqsubseteq x$.
    Ainsi $f(p)$ est un minorant de P donc $f(p) \sqsubseteq p$ ($p = \bigsqcap P$).
  - Par croissance de f, $f(f(p)) \sqsubseteq f(p)$ donc $f(p) \in P$.
    Or $p = \bigsqcap P$ donc $p \sqsubseteq f(p)$.
  - Ainsi $p = f(p)$.
- et c'est le plus petit :
  - Tous les point fixes sont dans P (si $f(x) = x$ alors $f(x) \sqsubseteq x$).
  - p est un minorant de P.

Notre système a une solution

- $L \to \mathcal{P}(\mathbb{V} \to \mathbb{Z})$ est un treillis complet (c.f. exercices).
- La fonction $F : (L \to \mathcal{P}(\mathbb{V} \to \mathbb{Z})) \to (L \to \mathcal{P}(\mathbb{V} \to \mathbb{Z}))$

$$F(R) = \left\{ \begin{array}{lll} 0 & \mapsto & (\mathbb{V} \to \mathbb{Z}) \\ l' & \mapsto & \displaystyle\bigcup_{(l,c,l') \in A} [\![c]\!]_{\mathrm{C}} (R(l)) \end{array} \right.$$

  est monotone.
- Donc notre sémantique est bien définie.

Problème

Malheureusement, la sémantique concrète n'est pas calculable.

Problème

Malheureusement, la sémantique concrète n'est pas calculable.

On va donc en calculer une surapproximation.

Problème

Malheureusement, la sémantique concrète n'est pas calculable.

On va donc en calculer une surapproximation. la prochaine fois