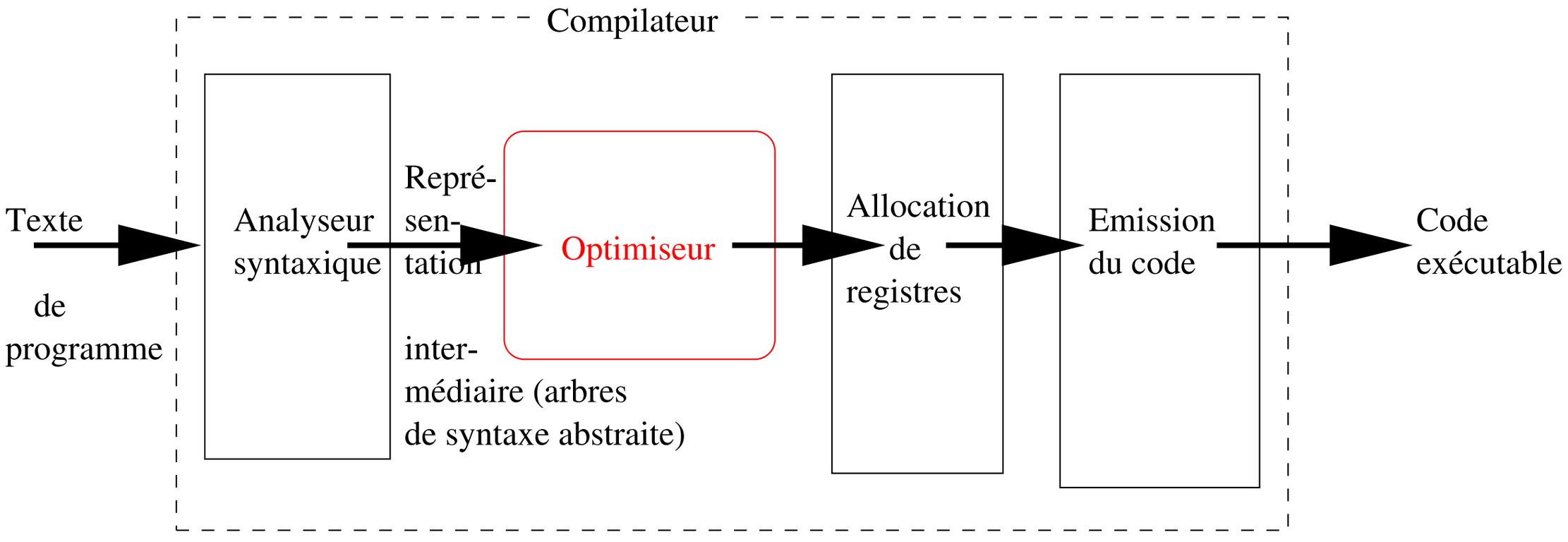
Analyse statique de programmes

Analyse statique

- But: découvrir **automatiquement** des propriétés (simples) des programmes
- Applications:
  - compilateurs optimisants (gcc, etc.)
  - preuve automatique d'absence d'erreurs à l'exécution (Polyspace, Astrée, etc.)
  - base d'autres outils (en sécurité par ex.)

Architecture d'un compilateur (pas le vôtre!)



Un exemple

```
int f (int m, int n) { int p = m+n;  return p+3; }
```

compilé par gcc 2.8.1 sans option en:

Mnémoniques assembleur	Explication
`movl 8(%ebp),%eax`	mettre `m` dans `%eax`
`movl 12(%ebp),%edx`	mettre `n` dans `%edx`
`leal (%edx,%eax),%ecx`	calculer `%ecx` := `%eax`+`%edx`
`movl %ecx,-4(%ebp)`	inutile… **car** *p* ne sera jamais plus utilisée
`movl -4(%ebp),%edx`	remplaçable par `movl %eax,%edx`
`addl $3,%edx`	ajouter 3 à `%edx`
`movl %edx,%eax`	retourner `%edx`

Propriété à démontrer, automatiquement

Quelques propriétés

- *live variables*: variables définies à la ligne *n* qui seront possiblement utilisées plus tard
- *dead code*: code qui ne sera jamais exécuté
- *array overflow*: débordement de tableau
- *arithmetic overflow*: débordement arithmétique (cf. Ariane 5)

Le théorème de Rice

- **Thm**: toute propriété extensionnelle et non-triviale de prgms est **indécidable**.
- Extensionnelle = ne dépend que de la sémantique
- Non-triviale: fausse d'au moins 1 prgm,
  et vraie d'au moins 1 prgm
- (pour un langage de prog. Turing-complet.)

Exemple

- La propriété « y a-t-il du code mort dans mon code? » est indécidable
- (Mais… ceci n'est pas extensionnel: comment utilise-t-on Rice ici?)
- Morale: « toute propriété intéressante de programmes est indécidable ».

L'interprétation abstraite

- On va construire des **approximations** calculables de propriétés *P* de prgms:
  - Si l'analyseur dit OK, *P* est vraie
  - Sinon, on ne sait pas…
  - mais l'analyseur termine

Illustration: la règle des signes

- But: détecter quelles vars *x* sont ≥ 0 et à quelles lignes d'un prgm π
- Deux signes: '+' et '–'
- Règles de calcul:
  '+' x '+' = '+' '+' x '–' = '–' etc.
- Quelles règles pour l'addition?

Top

- ‘+’ + ‘–’ = $\top$ (top, « je ne sais pas, ça peut être n’importe quoi »)
- Pas $\bot$ (bottom): pourquoi?

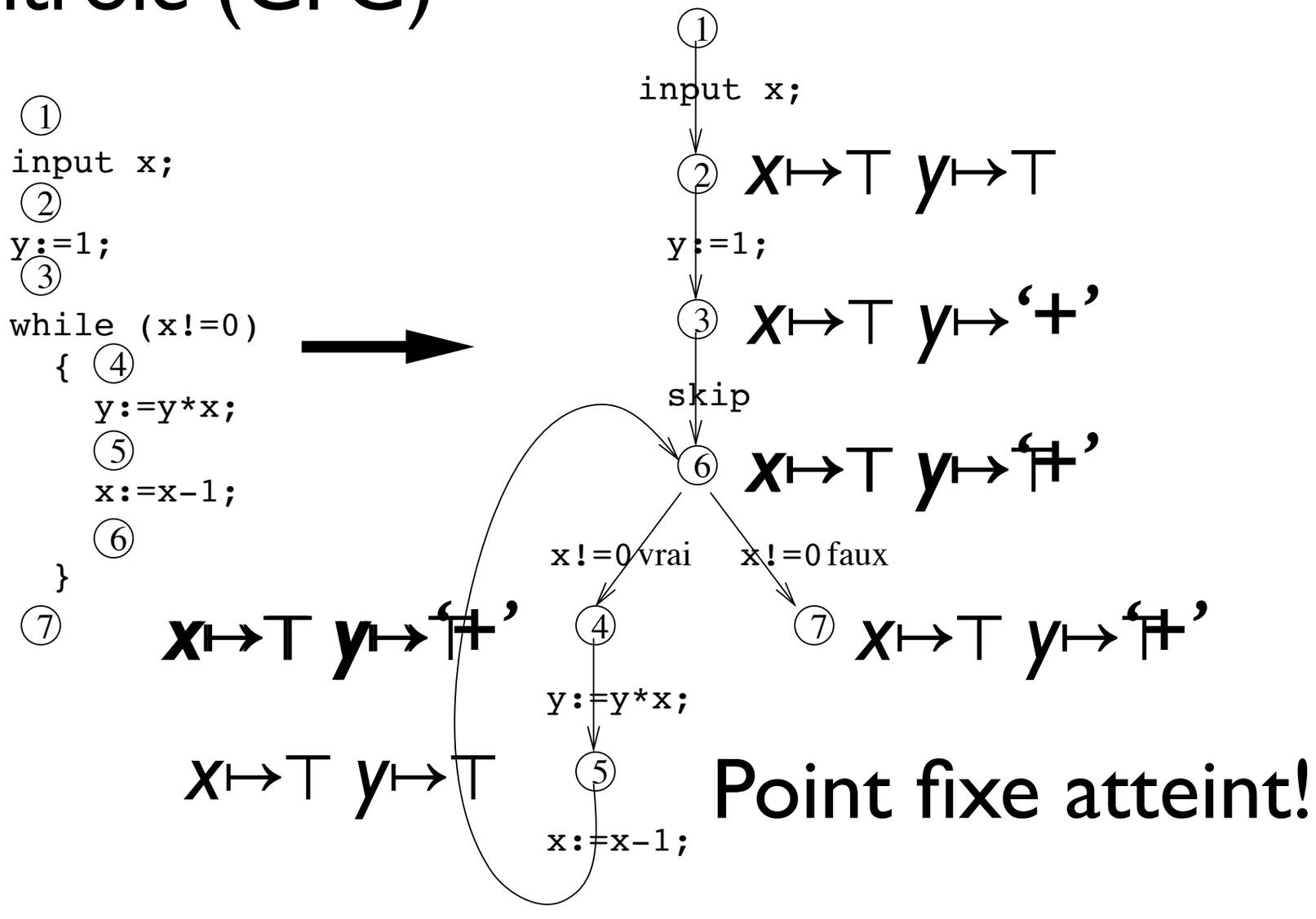
Concrétisation

- = fonction de sémantique γ des valeurs abstraites (ici, les signes) vers l'ensemble des propriétés (=sous-ensembles des valeurs possibles)

- $\gamma(\text{`+'}) = \{n \in \mathbf{Z} \mid n \geq 0\}$
  $\gamma(\text{`–'}) = \{n \in \mathbf{Z} \mid n < 0\}$
  $\gamma(\top) = \mathbf{Z}$ … monotone!
  $\gamma(\bot) = \varnothing$

Exemple

- En pratique on analyse des graphes de flot de contrôle (CFG)



Sémantique abstraite

- Soit $L^\#$ un treillis complet
  (de valeurs abstraites),
  $\text{Env}^\# = \text{Var (x Ligne)} \rightarrow L^\#$
  (treillis complet aussi)

- $[\![\text{skip}]\!]^\# \rho^\# = \rho^\#$
  $[\![x{=}e]\!]^\# \rho^\# = \rho^\#[x \mapsto [\![e]\!]^\# \rho^\# \vee \rho^\#(x)]$
  $[\![\text{if } e \text{ then } c \text{ else } c']\!]^\# \rho^\# = [\![c]\!]^\# \rho^\# \vee [\![c']\!]^\# \rho^\#$
  $[\![\text{while } e \text{ do } c]\!]^\# \rho^\# = \text{lfp}_{\rho^\#} ([\![c]\!]^\#)$

lfp_V

- $f: L^\# \to L^\#$ est **inflationnaire** ssi $f(x) \geq x$
- **Thm**: Si $f: L^\# \to L^\#$ est mon.+inflationnaire, pour tout $V \in L^\#$, f a un plus petit point fixe **au-dessus de V**. On le note $\mathrm{lfp}_V(f)$.
  (Note: monotonie pas nécessaire [Bourbaki-Witt])
- *Dém.*: Tarski sur le treillis complet $\uparrow V$.
  — Inflationnarité nécessaire: pourquoi?
  — Quels sont les sups, les infs, dans $\uparrow V$?

Autre démonstration

- **Thm**: Si $f: L^\# \to L^\#$ est mon.+inflationnaire, pour tout $V \in L^\#$, f a un plus petit point fixe **au-dessus de V**. On le note $\mathrm{lfp}_V(f)$.

- On pose $f^V(x^\#) = f(x^\#) \vee V$.
  Alors $\mathrm{lfp}_V(f) = \mathrm{lfp}(f^V)$
  (les points fixes de f^V sont exactement les points fixes de $f \geq V$.)

Correction

- (Supposons une sémantique petit pas avec des règles de la forme $(c, \rho) \rightarrow (c', \rho')$ pour simplifier)
- **Prop**: si $\rho(x) \in \gamma(\rho^{\#}(x))$ pour tout x,
  et si $(c, \rho) \rightarrow^{*} (c', \rho')$
  alors $\rho'(x) \in \gamma((\llbracket c \rrbracket^{\#}\rho^{\#})(x))$ pour tout x.
- **Dém.**: récurrence + analyse de cas

Faire mieux

- Peut-on, connaissant γ et la sémantique concrète $(c, \rho) \rightarrow (c', \rho')$, **déduire** la sémantique abstraite?
  … de sorte que cette dernière soit la plus précise possible?
- Correspondances de Galois

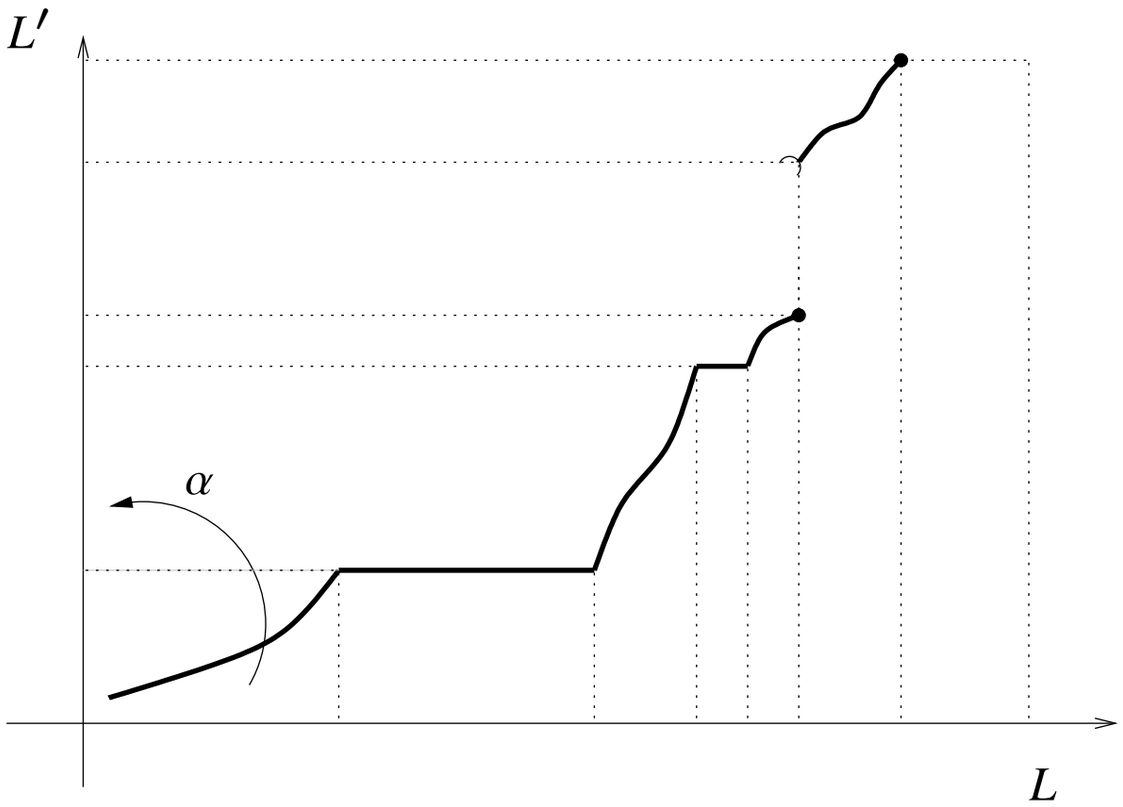
Correspondances de Galois

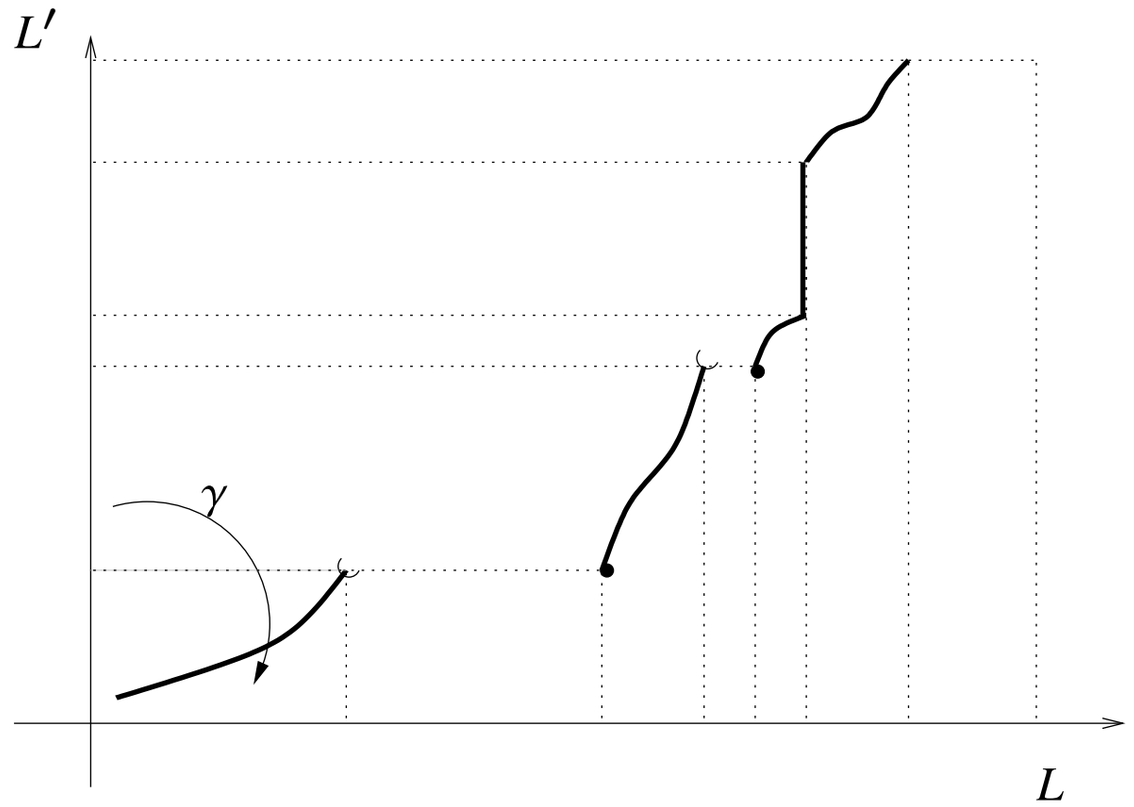
- Soit $\gamma : L^\# \to L$, $\alpha : L \to L^\#$ monotones.
  Les propriétés suivantes sont équivalentes:
  - pour tous $x, y^\#$, $\alpha(x) \leq y^\#$ ssi $x \leq \gamma(y^\#)$
  - pour tous $x, y^\#$, $y^\# \leq \alpha(\gamma(y^\#))$, $x \leq \gamma(\alpha(x))$
  - On dit alors que $\alpha \dashv \gamma$ est une **correspondance de Galois**

Correspondances de Galois

- Soit $\gamma : L^\# \to L$, $\alpha : L \to L^\#$ monotones, L et $L^\#$ treillis complets et $\alpha \dashv \gamma$ (pensez $L = \mathbb{P}(\mathrm{Val})$)
- Alors $\alpha(x)$ est la valeur abstraite $y^\#$ **la plus précise** (la plus petite) qui soit **correcte** par rapport à x ($x \leq \gamma(y^\#)$)

Correspondances de Galois





Galois et points fixes

- **Thm**: Soit L treillis complet, $f : L \to L$ monotone, et $L{:}\alpha \dashv \gamma{:} L^{\#}$, alors

$$\mathrm{lfp}(f) \leq \gamma(\mathrm{lfp}^{\#}(f^{\#})) \text{ où } f^{\#} = \alpha \circ f \circ \gamma$$

Galois et points fixes

- **Thm**: Soit L treillis complet, $f : L \to L$ monotone, et $L{:}\alpha \dashv \gamma{:} L^{\#}$, alors

  $\mathrm{lfp}(f) \leq \gamma(\mathrm{lfp}^{\#}(f^{\#}))$ où $f^{\#} = \alpha \circ f \circ \gamma$

- **Dém.**: $\mathrm{lfp}^{\#}(f^{\#}) = f^{\#}(\mathrm{lfp}^{\#}(f^{\#}))$

Galois et points fixes

- **Thm**: Soit L treillis complet, $f : L \to L$ monotone, et $L{:}\alpha \dashv \gamma{:} L^{\#}$, alors

  $$\mathrm{lfp}(f) \leq \gamma(\mathrm{lfp}^{\#}(f^{\#})) \text{ où } f^{\#} = \alpha \circ f \circ \gamma$$

- **Dém.**: $\mathrm{lfp}^{\#}(f^{\#}) = f^{\#}(\mathrm{lfp}^{\#}(f^{\#}))$
  donc $\alpha(f(\gamma(\mathrm{lfp}^{\#}(f^{\#})))) \leq \mathrm{lfp}^{\#}(f^{\#})$ (en fait =)

Galois et points fixes

- **Thm**: Soit L treillis complet, $f : L \to L$ monotone, et $L{:}\alpha \dashv \gamma{:} L^{\#}$, alors

  $$\mathrm{lfp}(f) \leq \gamma(\mathrm{lfp}^{\#}(f^{\#})) \text{ où } f^{\#} = \alpha \circ f \circ \gamma$$

- **Dém.**: $\mathrm{lfp}^{\#}(f^{\#}) = f^{\#}(\mathrm{lfp}^{\#}(f^{\#}))$
  donc $\alpha(f(\gamma(\mathrm{lfp}^{\#}(f^{\#})))) \leq \mathrm{lfp}^{\#}(f^{\#})$ (en fait =)
  donc $f(\gamma(\mathrm{lfp}^{\#}(f^{\#}))) \leq \gamma(\mathrm{lfp}^{\#}(f^{\#}))$ $(\alpha \dashv \gamma)$

Galois et points fixes

- **Thm**: Soit L treillis complet, $f : L \to L$ monotone, et $L{:}\alpha \dashv \gamma{:} L^{\#}$, alors

  $$\mathrm{lfp}(f) \leq \gamma(\mathrm{lfp}^{\#}(f^{\#})) \text{ où } f^{\#} = \alpha \circ f \circ \gamma$$

- **Dém.**: $\mathrm{lfp}^{\#}(f^{\#}) = f^{\#}(\mathrm{lfp}^{\#}(f^{\#}))$
  donc $\alpha(f(\gamma(\mathrm{lfp}^{\#}(f^{\#})))) \leq \mathrm{lfp}^{\#}(f^{\#})$ (en fait =)
  donc $f(\gamma(\mathrm{lfp}^{\#}(f^{\#}))) \leq \gamma(\mathrm{lfp}^{\#}(f^{\#}))$ $(\alpha \dashv \gamma)$
  i.e., $\gamma(\mathrm{lfp}^{\#}(f^{\#}))$ post-point fixe de f

Galois et points fixes

- **Thm**: Soit L, $L^\#$ treillis complet, $f : L \to L$ monotone, et $L:\alpha \dashv \gamma: L^\#$, alors

  $$\mathrm{lfp}(f) \leq \gamma(\mathrm{lfp}^\#(f^\#)) \text{ où } f^\# = \alpha \circ f \circ \gamma$$

- **Dém.**: $\mathrm{lfp}^\#(f^\#) = f^\#(\mathrm{lfp}^\#(f^\#))$
  donc $\alpha(f(\gamma(\mathrm{lfp}^\#(f^\#)))) \leq \mathrm{lfp}^\#(f^\#)$ (en fait =)
  donc $f(\gamma(\mathrm{lfp}^\#(f^\#))) \leq \gamma(\mathrm{lfp}^\#(f^\#))$ $(\alpha \dashv \gamma)$
  i.e., $\gamma(\mathrm{lfp}^\#(f^\#))$ post-point fixe de f
  donc $\geq$ le plus petit, qui est $\mathrm{lfp}(f)$ (Tarski).

Galois et points fixes

- **Thm**: Soit L, $L^\#$ treillis complet, $f : L \to L$ mon.+inflationnaire, et $L : \alpha \dashv \gamma : L^\#$, alors si $V \leq \gamma(V^\#)$ et $V^\# \in \mathrm{Im}\ \alpha$ on a:

  $\mathrm{lfp}_V(f) \leq \gamma(\mathrm{lfp}^\#_{V^\#}(f^\#))$ où $f^\# = \alpha \circ f \circ \gamma$

- **Dém.**: soit $V^\# = \alpha(V')$

  $\alpha \dashv \gamma$ se restreint à $\uparrow V' : \alpha \dashv \gamma : \uparrow V^\#$

  puis $\mathrm{lfp}_V(f) \leq \mathrm{lfp}_{V'}(f)$ (par définition de lfp_V.)

Galois et points fixes

- **Thm**: Soit L, $L^\#$ treillis complet, $f : L \to L$ mon.+inflationnaire, et $L{:}\alpha \dashv \gamma{:} L^\#$, alors si $V \leq \gamma(V^\#)$ et γ **injective** on a:

  $\mathrm{lfp}_V(f) \leq \gamma(\mathrm{lfp}^\#_{V^\#}(f^\#))$ où $f^\# = \alpha \circ f \circ \gamma$

- **Dém.**: il suffit de voir que α est alors surjective. En effet, on a toujours $\gamma(x^\#) = \gamma(\alpha(\gamma(x^\#)))$ (exercice).

Correction

- Pour ϱ : Var $\to \mathbb{P}$(Val)
- $[\![\text{skip}]\!]^\flat\ \varrho = \varrho$

$[\![x{=}e]\!]^\flat\ \varrho = \varrho[x \mapsto \varrho(x) \cup [\![e]\!]^\flat \varrho]$

$[\![\text{if } e \text{ then } c \text{ else } c']\!]^\flat \varrho = [\![c]\!]^\flat\ \varrho \cup [\![c']\!]^\flat \varrho$

$[\![\text{while } e \text{ do } c]\!]^\flat \varrho = \text{lfp}_\varrho\ ([\![c]\!]^\flat)$

Correction

- On démontre une fois pour toutes:
- **Prop**: si $(c, \rho) \rightarrow^* (c', \rho')$
  et pour tout x, $\rho(x) \in \varrho(x)$, alors
  pour tout x, $\rho'(x) \in (\llbracket c \rrbracket \flat \varrho)\ (x)$.

Transfert

- **Thm**: si ϱ est correct par rapport à $\rho^\#$, (i.e., pour tout x, $\varrho(x) \subseteq \gamma(\rho^\#(x))$), alors $[\![c]\!]^\flat \varrho$ est correct par rapport à $[\![c]\!]^\# \rho^\#$.

- **Dém.**: on utilise 'Galois et points fixes' pour les boucles.
- Fournit automatiquement des interprétations abstraites correctes $[\![c]\!]^\# \rho^\#$.

Calculabilité

- Si $L^\#$ est de hauteur finie (ex: pour les signes, hauteur 2), alors $\mathrm{lfp}^{\#}_{V\#}(f^\#)$ est le sup de la chaîne croissante **finie** des $f^{\#n}(V^\#)$.
- Sinon, techniques d'élargissement (Cousot), d'itérations sur les politiques (Goubault&Putot — pas moi).