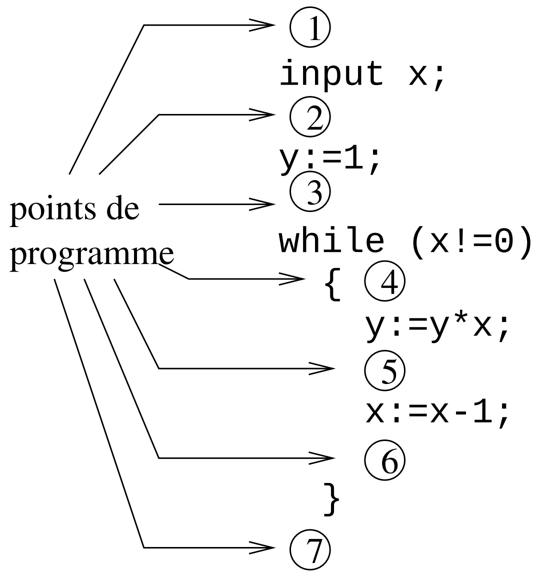
Interprétation abstraite

- Graphes de contrôle;
- Sémantique opérationnelle à petit pas;
- Équations sémantiques, points fixes;
- Résolution d'équations de points fixes;
- Élargissements.

Problème 6 : points de programmes intermédiaires

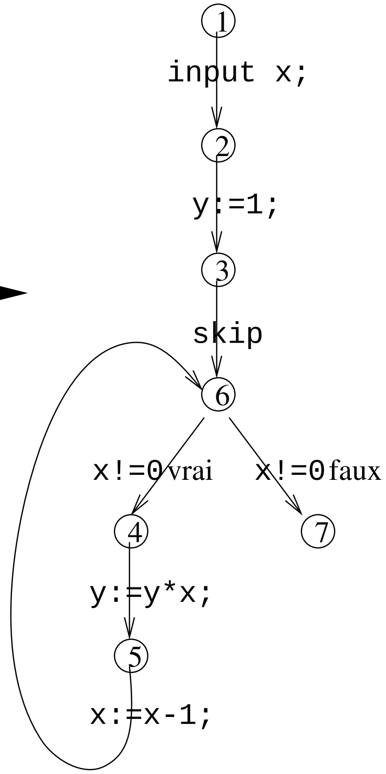
On étiquette chaque point de programme. Toute instruction est comprise entre deux points de programmes (“avant” et “après”).



Graphe de contrôle

On représente le programme par un graphe orienté, le graphe de contrôle :

```
1
input x;
2
y:=1;
3
while (x!=0)
   { 4
     y:=y*x;
     5
     x:=x-1;
     6
   }
7
```



Graphes orientés

Formellement, un graphe orienté (avec étiquettes dans $\mathcal{L}$) est un septuplet $(\mathrm{PP}, E, \partial^0, \partial^1, \delta, I, F)$, où :

- PP est un ensemble de sommets; (= points de programmes)
- E est un ensemble d'arcs; ($\sim$ lignes de programme)
- $\partial^0 : E \to \mathrm{PP}$ envoie chaque arc vers sa source;
- $\partial^1 : E \to \mathrm{PP}$ envoie chaque arc vers sa cible;
- $\delta : E \to \mathcal{L}$ envoie chaque arc vers son étiquette; ($\sim$ instruction)
- $I \in \mathrm{PP}$ est le sommet initial; (= entrée du programme)
- $F \in \mathrm{PP}$ est le sommet final (= sortie du programme).

On note $① \xrightarrow{\texttt{y:=1;}} ②$ l'arc e tel que $\partial^0 e = 1$, $\partial^1 e = 2$, $\delta(e) = \texttt{y := 1}$.

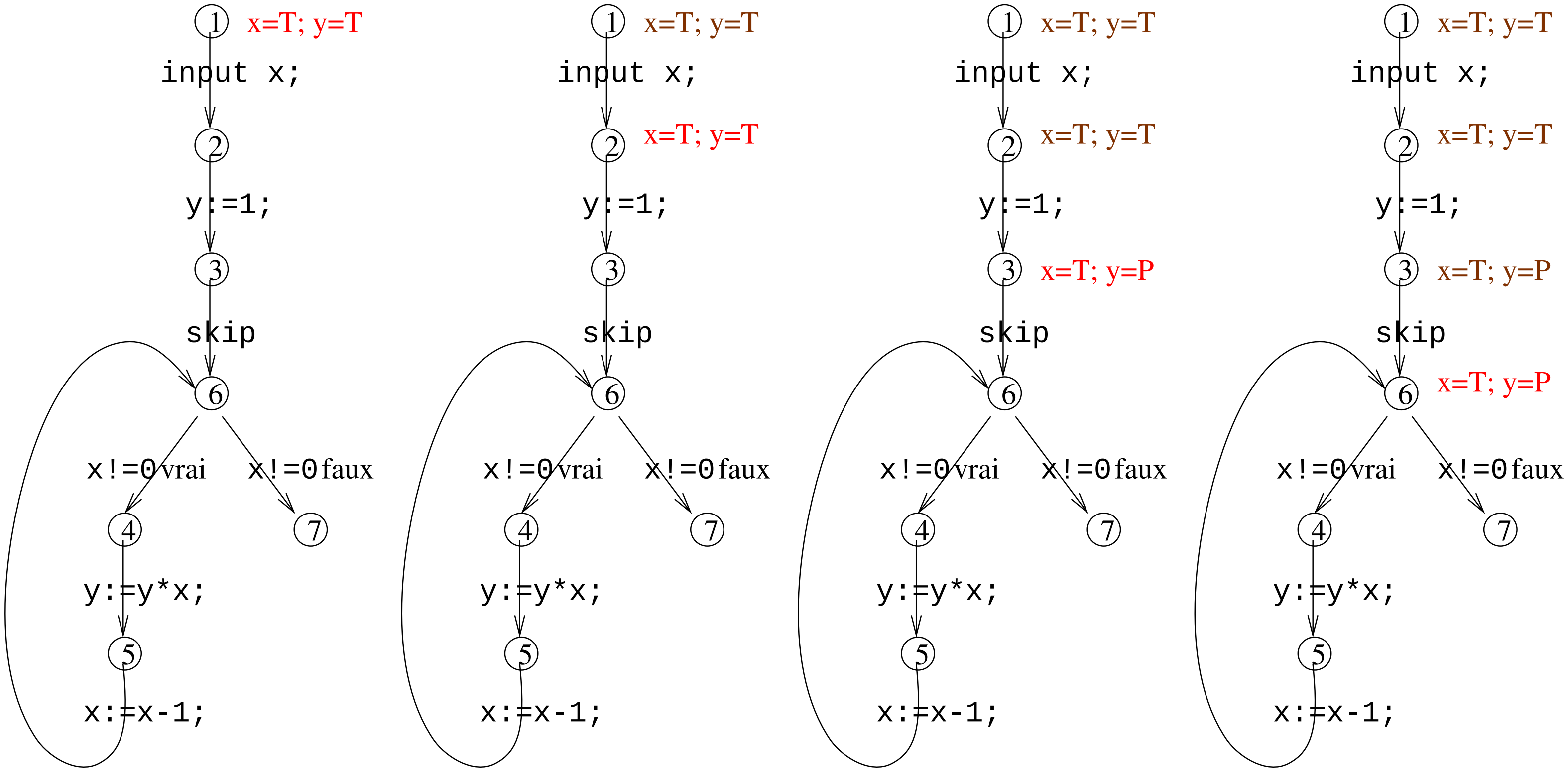
Dans les graphes de contrôles, les étiquettes sont appelées transitions :

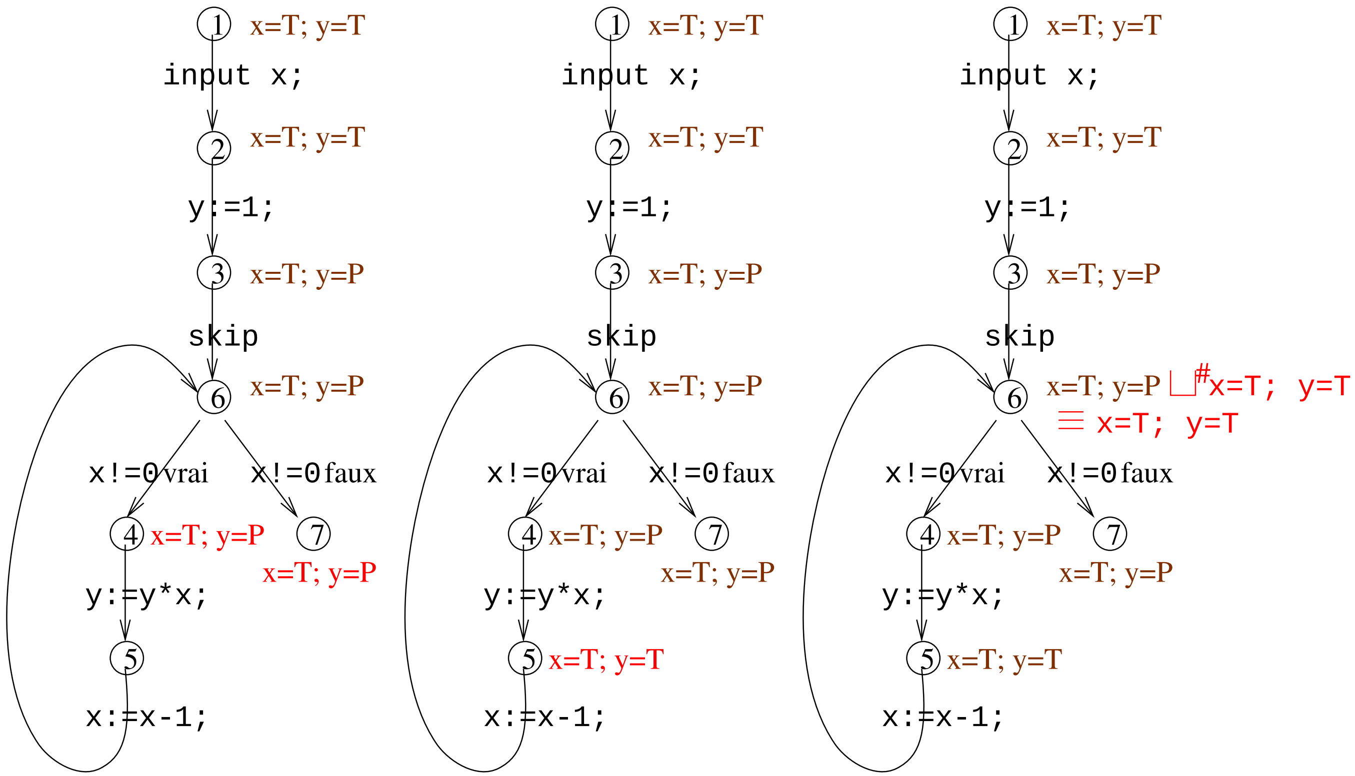
$$
\begin{array}{rcll}
t, \ldots \in \mathcal{T}rans & \hat{=} & x := e & \text{transitions} \\
 & | & \texttt{input}\; x & \\
 & | & e \text{ vrai?} & \\
 & | & e \text{ faux?} & \\
 & | & \texttt{skip} &
\end{array}
$$

En Caml :

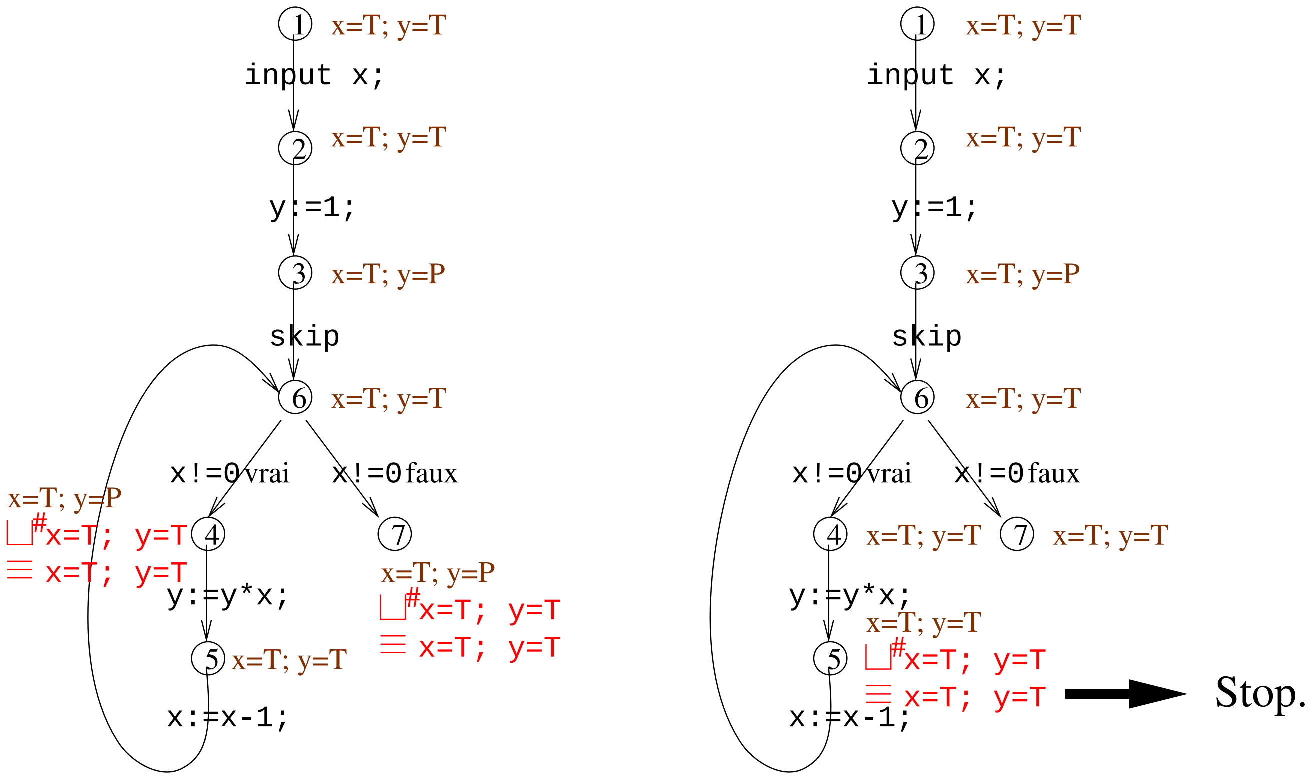
```
type etiq = SET of var * expr
   | INPUT of var
   | TEST of expr * bool (* e vrai?, e faux? *)
   | SKIP;;
```

Propagation des informations





1 x=T; y=T
input x;
2 x=T; y=T
y:=1;
3 x=T; y=P
skip
6 x=T; y=P
x!=0 vrai
x!=0 faux
4 x=T; y=P
7
x=T; y=P
y:=y*x;
5
x:=x-1;
1 x=T; y=T
input x;
2 x=T; y=T
y:=1;
3 x=T; y=P
skip
6 x=T; y=P
x!=0 vrai
x!=0 faux
4 x=T; y=P
7
x=T; y=P
y:=y*x;
5 x=T; y=T
x:=x-1;
1 x=T; y=T
input x;
2 x=T; y=T
y:=1;
3 x=T; y=P
skip
6 x=T; y=P ⊔# x=T; y=T
≡ x=T; y=T
x!=0 vrai
x!=0 faux
4 x=T; y=P
7
x=T; y=P
y:=y*x;
5 x=T; y=T
x:=x-1;



1 x=T; y=T
input x;
2 x=T; y=T
y:=1;
3 x=T; y=P
skip
6 x=T; y=T
x!=0 vrai
x!=0 faux
x=T; y=P
⊔# x=T; y=T
≡ x=T; y=T
4
y:=y*x;
5 x=T; y=T
x:=x-1;
7
x=T; y=P
⊔# x=T; y=T
≡ x=T; y=T
1 x=T; y=T
input x;
2 x=T; y=T
y:=1;
3 x=T; y=P
skip
6 x=T; y=T
x!=0 vrai
x!=0 faux
4 x=T; y=T
7 x=T; y=T
y:=y*x;
x=T; y=T
5 ⊔# x=T; y=T
≡ x=T; y=T
x:=x-1;
Stop.

En résumé :

- On propage des environnements abstraits $\rho^\sharp$ le long du graphe de contrôle …
- en prenant le sup $\sqcup^\sharp$ à chaque sommet de jonction (tout sommet ayant ≥ 2 prédécesseurs) …
- et en s'arrêtant dès que le sup ne fait plus augmenter l'information.

Ceci termine toujours … mais pourquoi ?

Ceci est correct … mais pourquoi ?

Une parenthèse sur les environnements abstraits

$$\rho^\sharp \in \mathcal{E}nv^\sharp \hat{=} \mathcal{V}ar \to L^\sharp \qquad \mathcal{E}nv^\sharp, \sqsubseteq^\sharp_{NR} \underset{\alpha_{NR}}{\overset{\gamma_{NR}}{\leftrightarrows}} \mathbb{P}(\mathcal{E}nv), \subseteq$$

(abstrait) (concret)

$\rho_1^\sharp \sqsubseteq^{\sharp *} \rho_2^\sharp \Leftrightarrow \forall x \in \mathcal{V}ar \cdot \rho_1(x) \sqsubseteq^\sharp \rho_2(x)$

$\gamma_{NR}(\rho^\sharp) \hat{=} \{\rho \mid \forall x \in \mathcal{V}ar \cdot \rho(x) \in \gamma(\rho^\sharp(x))\}$

$\alpha_{NR}(A) \hat{=} \lambda x \in \mathcal{V}ar \cdot \alpha\{\rho(x) \mid \rho \in A\}$

: interprétation non relationnelle

(perd les relations entre différentes variables)

… mais on pourrait plus généralement utiliser un treillis $\mathcal{E}nv^\sharp$ arbitraire (on verra ça plus tard).

Théorème : $(\alpha_{NR}, \gamma_{NR})$ est une correspondance de Galois.

Démonstration :

$$\begin{array}{rcll} \alpha_{NR}(A) \sqsubseteq^{\sharp *} \rho^\sharp & \Leftrightarrow & \forall x \cdot \alpha\{\rho(x) \mid \rho \in A\} \sqsubseteq^\sharp \rho^\sharp(x) & \\ & \Leftrightarrow & \forall x \cdot \{\rho(x) \mid \rho \in A\} \subseteq \gamma(\rho^\sharp(x)) & (1) \\ A \subseteq \gamma_{NR}(\rho^\sharp) & \Leftrightarrow & A \subseteq \{\rho \mid \forall x \cdot \rho(x) \in \gamma(\rho^\sharp(x))\} & (2) \end{array}$$

$(1) \Rightarrow (2)$: si $\rho \in A$, par (1) $\rho(x) \in \gamma(\rho^\sharp(x))$ pour tout x, donc $\rho \in \{\rho \mid \forall x \cdot \rho(x) \in \gamma(\rho^\sharp(x))\}$.
$(2) \Rightarrow (1)$: si $v \in \{\rho(x) \mid \rho \in A\}$, $v = \rho(x)$ pour un $\rho \in A$; par (2) $\rho(x) \in \gamma(\rho^\sharp(x))$, donc $v \in \gamma(\rho^\sharp(x))$.

Correction, sémantique opérationnelle petit pas

Correction : s'assurer que la sémantique abstraite et la sémantique concrète se correspondent
... ceci nécessite en premier la définition de la sémantique (concrète) du langage à analyser.

Elle est définie à l'aide de jugements :

$$p_1, \rho_1 \vdash instr \Rightarrow p_2, \rho_2 \quad (\text{si } cond)$$

"au point de programme p_1, dans l'environnement ρ_1, exécuter l'instruction $instr$ permet d'atteindre le point de programme p_2, et l'environnement devient ρ_2, si $cond$ est vraie."

Pour tout $p_1 \xrightarrow{instr} p_2$ dans le programme, on crée une règle $p_1, \rho_1 \vdash instr \Rrightarrow p_2, \rho_2$ (si $cond$) :

$instr$	règle	éq. sémantique $cond[\![instr]\!](\rho_1, \rho_2)$
$x{:=}e$	$p_1, \rho_1 \vdash x{:=}e \Rrightarrow p_2, \rho_2$	(si $\rho_2 = \rho_1[x \leftarrow \mathcal{E}[\![e]\!]\rho_1]$)
$\texttt{input}\, x$	$p_1, \rho_1 \vdash \texttt{input}\, x \Rrightarrow p_2, \rho_2$	(si $\forall y \neq x \cdot \rho_2(y) = \rho_1(y)$)
e vrai?	$p_1, \rho_1 \vdash e$ vrai? $\Rrightarrow p_2, \rho_2$	(si $\mathcal{E}[\![e]\!]\rho_1 = \texttt{true}$ et $\rho_2 = \rho_1$)
e faux?	$p_1, \rho_1 \vdash e$ faux? $\Rrightarrow p_2, \rho_2$	(si $\mathcal{E}[\![e]\!]\rho_1 = \texttt{false}$ et $\rho_2 = \rho_1$)
$\texttt{skip}$	$p_1, \rho_1 \vdash \texttt{skip} \Rrightarrow p_2, \rho_2$	(si $\rho_2 = \rho_1$)

Traces

Une trace (d'exécution) T est une suite $(p_1^T, \rho_1^T), (p_2^T, \rho_2^T), \ldots, (p_k^T, \rho_k^T), \ldots$ telle que :

$$p_1^T, \rho_1^T \vdash instr_1 \Rightarrow p_2^T, \rho_2^T \vdash instr_2 \Rightarrow \ldots \Rightarrow p_k^T, \rho_k^T \vdash \ldots$$

et $p_1^T = I$.

L'ensemble des environnements $Env(p)$ au point de programme p est $\{\rho | \exists T \cdot \exists k \in \mathbb{N} \cdot \rho = \rho_k^T \wedge p_k^T = p\}$.

Le but de l'interprétation abstraite sera de calculer $Env^\sharp \in \mathcal{E}nv^\sharp$ tel que pour tout point de programme p, $Env(p) \subseteq \gamma_{NR}(Env^\sharp(p))$ (correction), c'est-à-dire $\forall \rho \in Env(p) \cdot \forall x \in \mathcal{V}ar \cdot \rho(x) \in \gamma(Env^\sharp(p)(x))$.

Sémantique collectante

Supposons que l'on ait une estimation E de $Env \hat{=} \lambda p \cdot Env(p)$.

On peut calculer une meilleure estimation E' de Env en voyant ce qui se passe après une étape de calcul :

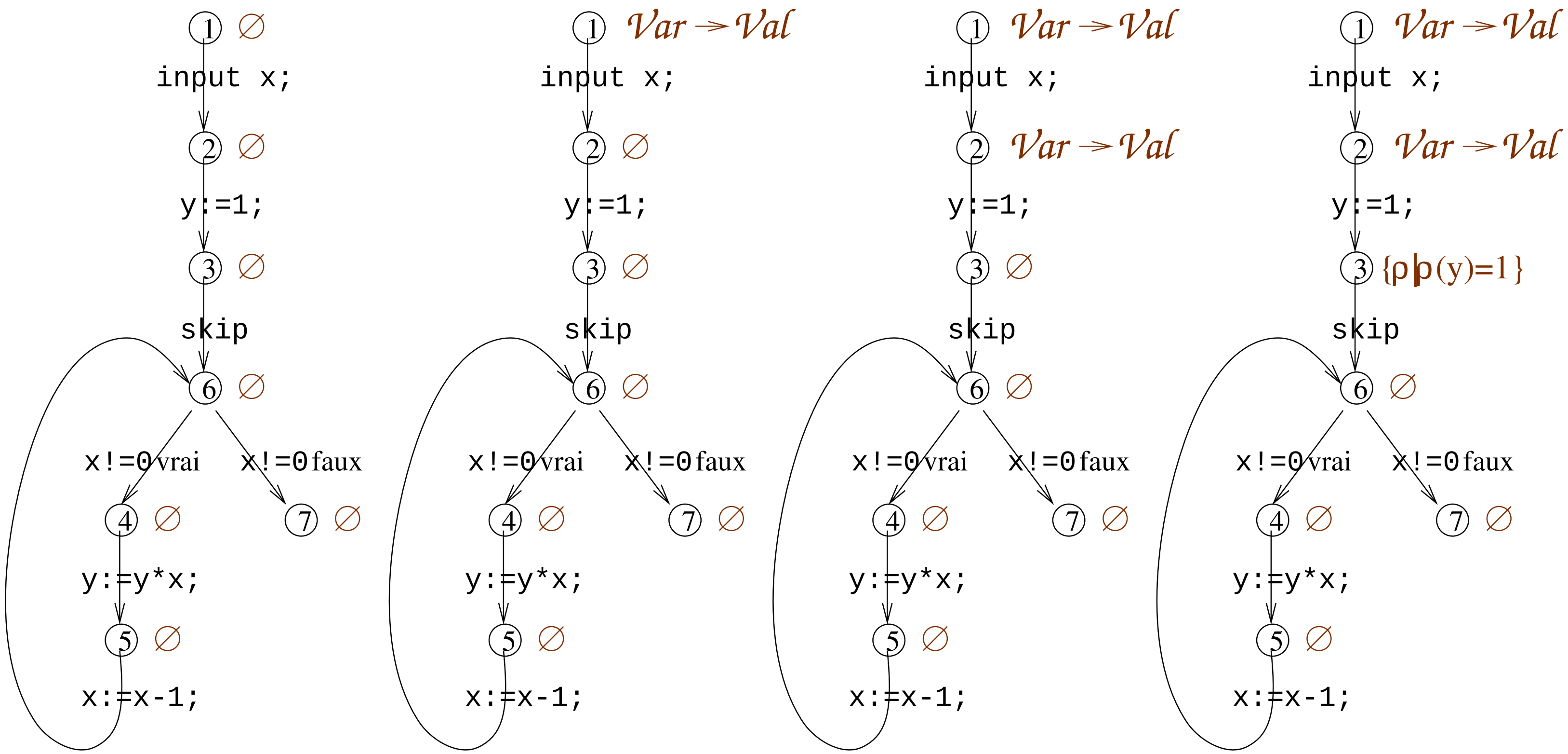
$E'(I) \hat{=} \mathcal{V}ar \to \mathcal{V}al$

(tous les environnements possibles)

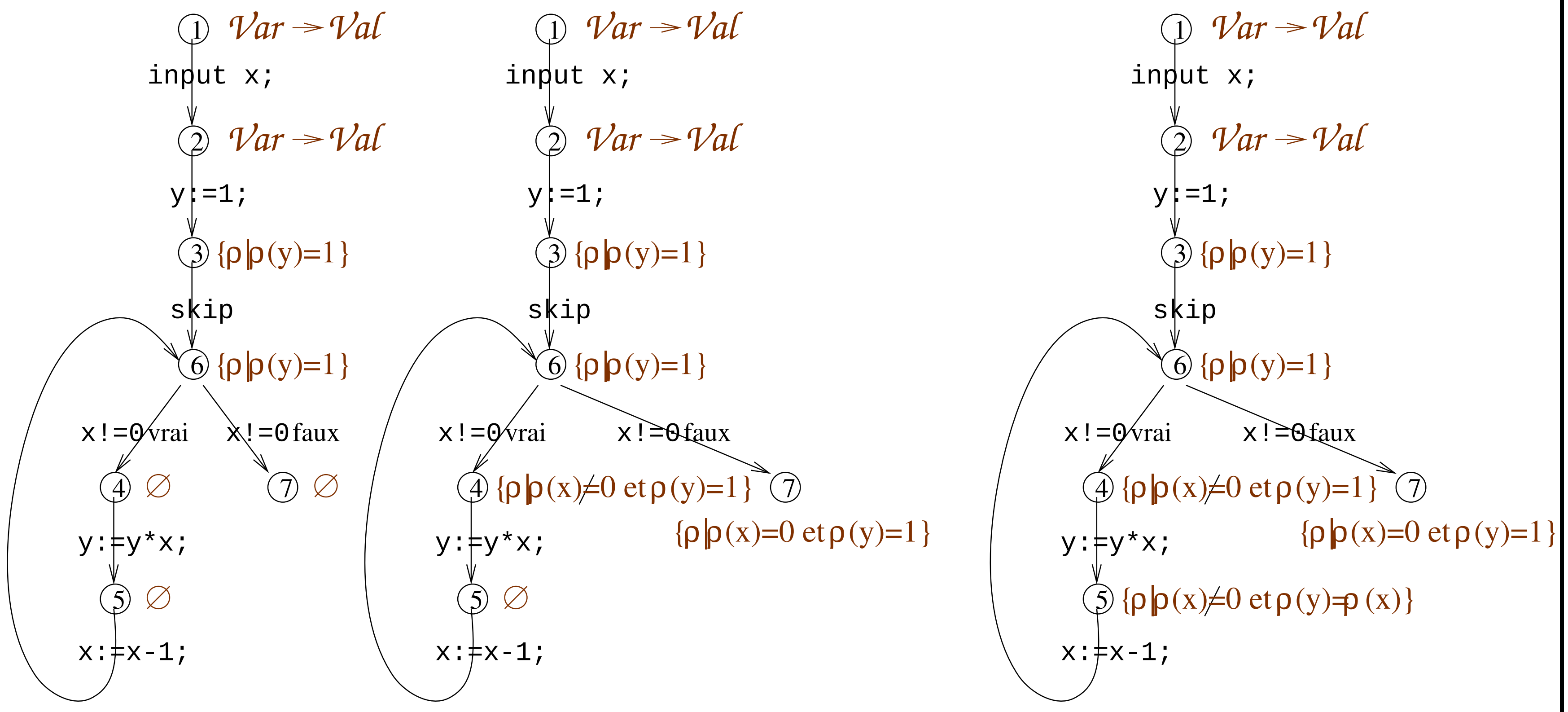
$E'(p) \hat{=} E(p) \cup \{\rho \mid \exists p_1, \rho_1 \in E(p_1), instr \cdot p_1, \rho_1 \vdash instr \Rightarrow p, \rho\} \quad (p \neq I)$

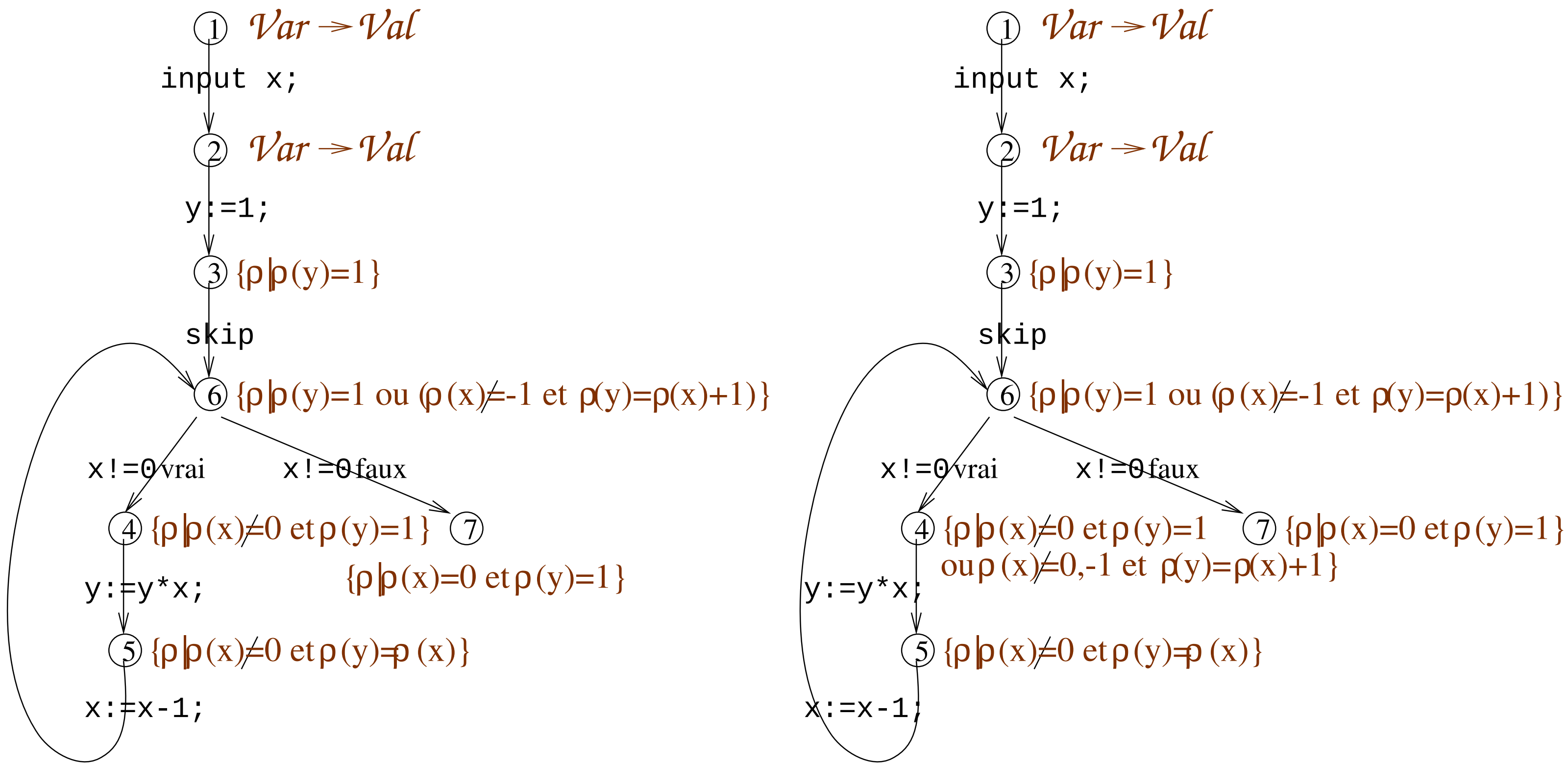
($E(p)$ plus résultats d'une étape de calcul)

E' est une fonction de E, la sémantique collectante avant $\mathcal{P}_{\to}(E)$ du programme.

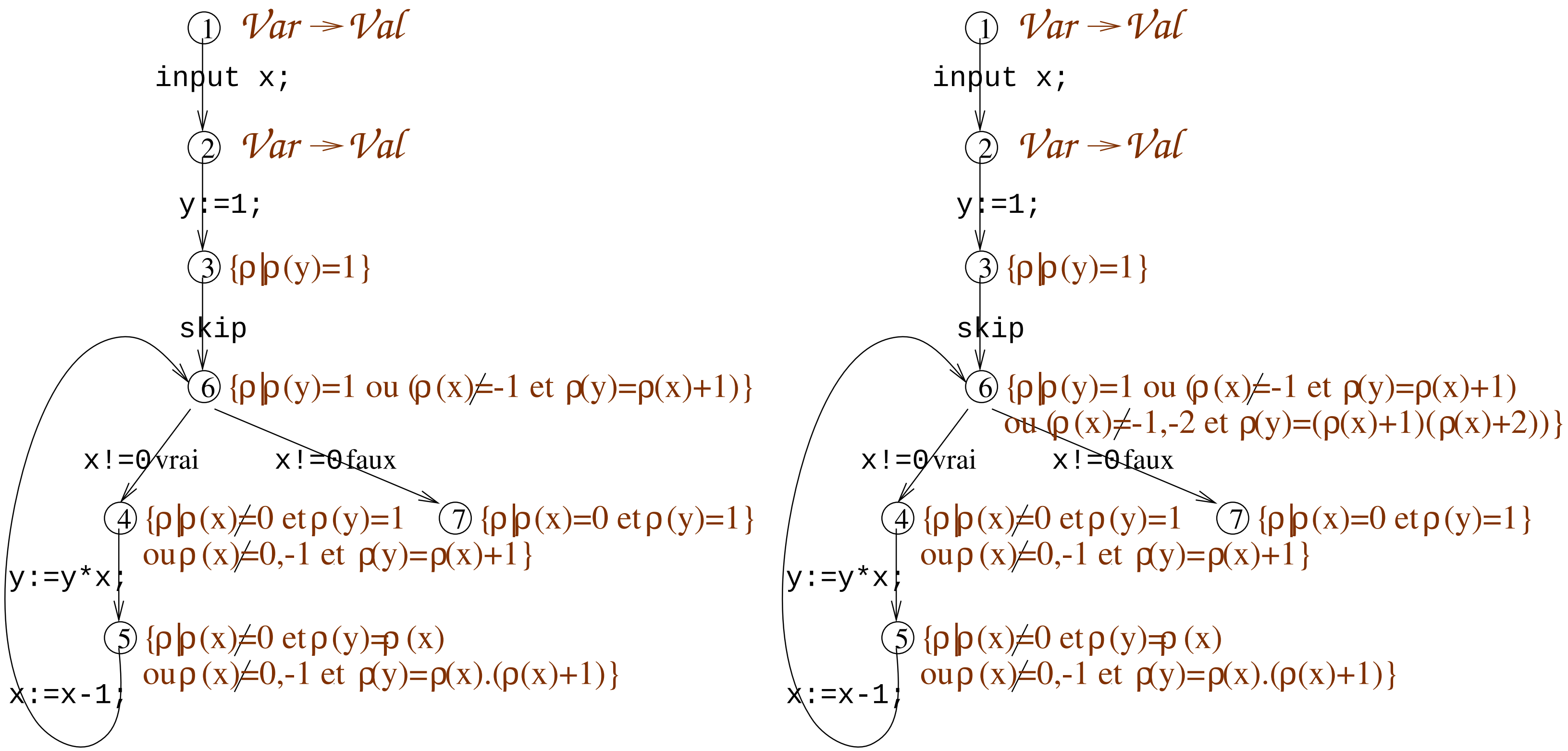


1 ∅
input x;
2 ∅
y:=1;
3 ∅
skip
6 ∅
x!=0 vrai
x!=0 faux
4 ∅
7 ∅
y:=y*x;
5 ∅
x:=x-1;
1 Var → Val
input x;
2 ∅
y:=1;
3 ∅
skip
6 ∅
x!=0 vrai
x!=0 faux
4 ∅
7 ∅
y:=y*x;
5 ∅
x:=x-1;
1 Var → Val
input x;
2 Var → Val
y:=1;
3 ∅
skip
6 ∅
x!=0 vrai
x!=0 faux
4 ∅
7 ∅
y:=y*x;
5 ∅
x:=x-1;
1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 ∅
x!=0 vrai
x!=0 faux
4 ∅
7 ∅
y:=y*x;
5 ∅
x:=x-1;


1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 {ρ|ρ(y)=1}
x!=0 vrai
x!=0 faux
4 ∅
7 ∅
y:=y*x;
5 ∅
x:=x-1;
1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 {ρ|ρ(y)=1}
x!=0 vrai
x!=0 faux
4 {ρ|ρ(x)≠0 et ρ(y)=1}
7 {ρ|ρ(x)=0 et ρ(y)=1}
y:=y*x;
5 ∅
x:=x-1;
1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 {ρ|ρ(y)=1}
x!=0 vrai
x!=0 faux
4 {ρ|ρ(x)≠0 et ρ(y)=1}
7 {ρ|ρ(x)=0 et ρ(y)=1}
y:=y*x;
5 {ρ|ρ(x)≠0 et ρ(y)=ρ(x)}
x:=x-1;



1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 {ρ|ρ(y)=1 ou (ρ(x)≠-1 et ρ(y)=ρ(x)+1)}
x!=0 vrai
x!=0 faux
4 {ρ|ρ(x)≠0 et ρ(y)=1}
7
{ρ|ρ(x)=0 et ρ(y)=1}
y:=y*x;
5 {ρ|ρ(x)≠0 et ρ(y)=ρ(x)}
x:=x-1;
1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 {ρ|ρ(y)=1 ou (ρ(x)≠-1 et ρ(y)=ρ(x)+1)}
x!=0 vrai
x!=0 faux
4 {ρ|ρ(x)≠0 et ρ(y)=1
ou ρ(x)≠0,-1 et ρ(y)=ρ(x)+1}
7 {ρ|ρ(x)=0 et ρ(y)=1}
y:=y*x;
5 {ρ|ρ(x)≠0 et ρ(y)=ρ(x)}
x:=x-1;



1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 {ρ|ρ(y)=1 ou (ρ(x)≠-1 et ρ(y)=ρ(x)+1)}
x!=0 vrai
x!=0 faux
4 {ρ|ρ(x)≠0 et ρ(y)=1
ou ρ(x)≠0,-1 et ρ(y)=ρ(x)+1}
7 {ρ|ρ(x)=0 et ρ(y)=1}
y:=y*x;
5 {ρ|ρ(x)≠0 et ρ(y)=ρ(x)
ou ρ(x)≠0,-1 et ρ(y)=ρ(x).(ρ(x)+1)}
x:=x-1;
1 Var → Val
input x;
2 Var → Val
y:=1;
3 {ρ|ρ(y)=1}
skip
6 {ρ|ρ(y)=1 ou (ρ(x)≠-1 et ρ(y)=ρ(x)+1)
ou (ρ(x)≠-1,-2 et ρ(y)=(ρ(x)+1)(ρ(x)+2))}
x!=0 vrai
x!=0 faux
4 {ρ|ρ(x)≠0 et ρ(y)=1
ou ρ(x)≠0,-1 et ρ(y)=ρ(x)+1}
7 {ρ|ρ(x)=0 et ρ(y)=1}
y:=y*x;
5 {ρ|ρ(x)≠0 et ρ(y)=ρ(x)
ou ρ(x)≠0,-1 et ρ(y)=ρ(x).(ρ(x)+1)}
x:=x-1;

Passage à la limite

On calcule $E_\infty \hat{=} \lambda p \cdot \bigcup_{k \in \mathbb{N}} E_k(p)$, avec $E_0 \hat{=} \lambda p \cdot \emptyset$ et $E_{k+1} \hat{=} \mathcal{P}_\rightarrow(E_k)$:

Point 1 : $\mathcal{V}ar \rightarrow \mathcal{V}al$

Point 2 : $\mathcal{V}ar \rightarrow \mathcal{V}al$

Point 3 : $\{\rho \mid \rho(y) = 1\}$

Point 4 : $\{\rho \mid \exists k \in \mathbb{N} \cdot \rho(x) \notin \{0, -1, \ldots, -k\} \wedge \rho(y) = \prod_{i=1}^{k}(\rho(x) + i)\}$

Point 5 : $\{\rho \mid \exists k \in \mathbb{N} \cdot \rho(x) \notin \{0, -1, \ldots, -k\} \wedge \rho(y) = \prod_{i=0}^{k}(\rho(x) + i)\}$

Point 6 : $\{\rho \mid \exists k \in \mathbb{N} \cdot \rho(x) \notin \{-1, \ldots, -k\} \wedge \rho(y) = \prod_{i=1}^{k}(\rho(x) + i)\}$

Point 7 : $\{\rho \mid \exists k \in \mathbb{N} \cdot \rho(x) = 0 \wedge \rho(y) = k!\}$

Note : $E_\infty = \mathcal{P}_\rightarrow(E_\infty)$, i.e. E_∞ est un point fixe de $\mathcal{P}_\rightarrow$.

Une caractérisation de Env

Ce qu'on a calculé par cette itération infinie, c'est Env elle-même :

Théorème : $Env \hat{=} \lambda p \cdot Env(p) = lfp_{\subseteq^*}(\mathcal{P}_{\rightarrow})$, où $lfp_{\subseteq^*}$ est l'opérateur "plus petit point fixe" et $E \subseteq^* E'$ ssi pour tout p, $E(p) \subseteq E'(p)$.

Démonstration :

• Env est un point fixe de $\mathcal{P}_{\rightarrow}$:

Pour $p = I$, $\mathcal{P}_{\rightarrow}(Env)(I) = \mathcal{V}ar \rightarrow \mathcal{V}al = Env(I)$. Pour $p \neq I$:

$$\begin{aligned}
\mathcal{P}_{\rightarrow}(Env)(p) &= Env(p) \cup \{\rho \mid \exists p_1, \rho_1 \in Env(p_1), instr \cdot p_1, \rho_1 \vdash instr \Rightarrow p, \rho\} \\
&= \{\rho_k^T \mid T, k \geq 1, p_k^T = p\} \cup \{\rho \mid \exists T, k \geq 1, instr \cdot p_k^T, \rho_k^T \vdash instr \Rightarrow p, \rho\} \\
&= \{\rho_k^T \mid T, k \geq 1, p_k^T = p\} \cup \{\rho_{k+1}^{T'} \mid T', k \geq 1, p_{k+1}^{T'} = p\} \\
&\qquad (T' \hat{=} (p_1^T, \rho_1^T), \ldots, (p_k^T, \rho_k^T), (p, \rho); \quad T \hat{=} (p_1^{T'}, \rho_1^{T'}), \ldots, (p_k^{T'}, \rho_k^{T'})) \\
&= \{\rho_k^T \mid T, k \geq 1, p_k^T = p\} = Env(p) \\
&\qquad (\text{car } \{\rho_{k+1}^{T'} \mid T', k \geq 1, p_{k+1}^{T'} = p\} \subseteq \{\rho_k^T \mid T, k \geq 1, p_k^T = p\})
\end{aligned}$$

Donc pour tout p, $\mathcal{P}_{\rightarrow}(Env)(p) = Env(p)$, donc $\mathcal{P}_{\rightarrow}(Env) = Env$.

• Env est plus petit que tout point fixe E de $\mathcal{P}_{\rightarrow}$:

Montrons d'abord que : (1) $Env_k(p) \hat{=} \{\rho \mid \exists T \cdot \rho = \rho_k^T \wedge p_k^T = p\} \subseteq E(p) \ldots$ par récurrence sur $k \geq 1$.

Pour $k = 1$, $Env_1(p) = \begin{cases} \mathcal{V}ar \rightarrow \mathcal{V}al & \text{si } p = I \\ \emptyset & \text{sinon} \end{cases} \subseteq E(p)$

car $E(I) = \mathcal{P}_{\rightarrow}(E)(I) = \mathcal{V}ar \rightarrow \mathcal{V}al$.

Sinon, $Env_{k+1}(p) = \{\rho \mid \exists T \cdot \rho = \rho_{k+1}^T \wedge p_{k+1}^T = p\}$

$= \{\rho \mid \exists T', instr \cdot p_k^{T'}, \rho_k^{T'} \vdash instr \Rrightarrow p, \rho\}$

$\subseteq \mathcal{P}_{\rightarrow}(Env_k)(p)$ (définition de $\mathcal{P}_{\rightarrow}$)

$\subseteq \mathcal{P}_{\rightarrow}(E)(p)$ (hypothèse de récurrence et monotonie de $\mathcal{P}_{\rightarrow}$)

$= E(p)$ (E point fixe de $\mathcal{P}_{\rightarrow}$)

Donc $Env(p) = \bigcup_{k \geq 1} Env_k(p) \subseteq E(p)$ par (1).