Journée en l'honneur de Nicolas Halbwachs

Lundi 4 juin 2018 — 10h20 — 11h15

VERIMAG, Université Grenoble Alpes (France)

Analyse statique de dépendance par interprétation abstraite

Patrick Cousot

New York University, Courant Institute of Mathematics, Computer Science

pcousot@cs.nyu.edu   cs.nyu.edu/~pcousot

Motivation

Dépendance

Notion qui intervient dans de nombreux raisonnements sur les programmes :

- Non-interférence (confidentialité, intégrité)
- Sécurité, vie privée
- Découpage de programmes (*slicing*)
- Dépendances temporelles en Lustre, Signal
- etc.

Les définitions existantes

- sont données à priori (par exemple Cheney, Ahmed, and Acar, 2011; D. E. Denning and P. J. Denning, 1977),
- sans justification sémantiques (à l'exception de Assaf, Naumann, Signoles, Totel, and Tronel, 2017; Urban and Müller, 2018)

Notre objectif est de montrer les principes, pas d'obtenir une analyse de dépendance puissante

Sémantique

Syntaxe des programmes

- Programmes C limités aux entiers, affectations, séquences, conditionnelles, iterations
- Les programmes sont étiquetés pour nommer des points de programme
  - $\mathsf{at}[\![\mathtt{S}]\!]$: point d'entrée de `S`
  - $\mathsf{after}[\![\mathtt{S}]\!]$: point de sortie de `S`
  - $\mathsf{in}[\![\mathtt{S}]\!]$: points accessibles pendant l'exécution de `S` (sauf $\mathsf{after}[\![\mathtt{S}]\!]$)
  - $\mathsf{break\text{-}to}[\![\mathtt{S}]\!]$: point de branchement si `S` contient un **`break ;`** (alors $\mathsf{escape}[\![\mathtt{S}]\!] = \mathsf{tt}$)

Traces d'exécution

- Programme:

$$\ell_1\ \mathtt{x = 0\ ;\ while}\ \ell_2\ (\mathtt{tt})\ \{\ \ell_3\ \mathtt{x = x+1\ ;}\ \}\ \ell_4$$

- Trace d'execution infinie: $\ell_1 \xrightarrow{\mathtt{x = 0} = 0} \ell_2 \xrightarrow{\mathtt{tt}} \ell_3 \xrightarrow{\mathtt{x = x + 1} = 1} \ell_2 \xrightarrow{\mathtt{tt}} \ell_3$ $\xrightarrow{\mathtt{x = x + 1} = 2} \ell_2 \ldots \ell_2 \xrightarrow{\mathtt{tt}} \ell_3 \xrightarrow{\mathtt{x = x + 1} = n} \ell_2 \xrightarrow{\mathtt{tt}} \ell_3 \xrightarrow{\mathtt{x = x + 1} = n+1} \ell_2 \ldots$

- Trace: suite finie ou infinie de points de programme séparés par des actions ($\mathtt{x = A} = \mathit{valeur}$, $\mathtt{B}$, $\neg\mathtt{B}$, et **break ;**)

Valeur d'une variable (et d'une expression)

- Le valeur d'une variable le long d'une trace est la dernière valeur affectée (ou initialement 0)

$$
\begin{array}{rcll}
\boldsymbol{\rho}(\pi^{\ell} \xrightarrow{\ \mathsf{x} \,=\, \mathsf{E} \,=\, \upsilon\ } \ell')\mathsf{x} & \triangleq & \upsilon & \\
\boldsymbol{\rho}(\pi^{\ell} \xrightarrow{\ \cdots\ } \ell')\mathsf{x} & \triangleq & \boldsymbol{\rho}(\pi^{\ell}) & \text{otherwise} \\
\boldsymbol{\rho}(\ell)\mathsf{x} & \triangleq & 0 &
\end{array}
$$

Sémantique $\widehat{\mathcal{S}}^*[\![\mathsf{S}]\!]$ de traces préfixes

- Étant donnée une trace d'initialisation $\pi_0\mathsf{at}[\![\mathsf{S}]\!]$ se terminant à l'entrée $\mathsf{at}[\![\mathsf{S}]\!]$ d'une commande S, la sémantique de traces préfixes $\widehat{\mathcal{S}}^*[\![\mathsf{S}]\!](\pi_0\mathsf{at}[\![\mathsf{S}]\!])$ est l'ensemble des préfixes des traces d'exécution continuant $\pi_0\mathsf{at}[\![\mathsf{S}]\!]$
- $\widehat{\mathcal{S}}^*[\![\mathsf{S}]\!]$ est définie par induction sur la syntax des commandes S et par point fixe pour l'itération.

Définition structurelle de la sémantique de traces préfixes I

- À l'entrée $\mathsf{at}[\![\mathsf{S}]\!]$ d'une commande S:
  - $$\frac{}{\mathsf{at}[\![\mathsf{S}]\!] \in \widehat{\mathcal{S}}^*[\![\mathsf{S}]\!](\pi_1\mathsf{at}[\![\mathsf{S}]\!])}$$
- Affectation $\mathsf{S} ::= {}^{\ell}\ \mathtt{x = A\ ;}$:
  - $$\frac{v = \mathscr{A}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi^{\ell})}{\ell \xrightarrow{\mathtt{x = A} = v} \mathsf{after}[\![\mathsf{S}]\!] \in \widehat{\mathcal{S}}^*[\![\mathsf{S}]\!](\pi^{\ell})}$$

Définition structurelle de la sémantique de traces préfixes II

- Conditionnelle $\texttt{S} ::= \texttt{if}\ ^{\ell}\ \texttt{(B)}\ \texttt{S}_t$:

  - $$\frac{\mathscr{B}[\![\texttt{B}]\!]\rho(\pi_1{}^{\ell}) = \mathrm{ff}}{\ell \xrightarrow{\neg(\texttt{B})} \mathrm{after}[\![\texttt{S}]\!] \in \widehat{\mathscr{S}}^*[\![\texttt{S}]\!](\pi_1{}^{\ell})}$$

  - $$\frac{\mathscr{B}[\![\texttt{B}]\!]\rho(\pi_1{}^{\ell}) = \mathrm{tt}, \quad \pi_2 \in \widehat{\mathscr{S}}^*[\![\texttt{S}_t]\!](\pi_1{}^{\ell} \xrightarrow{\texttt{B}} \mathrm{at}[\![\texttt{S}_t]\!])}{\ell \xrightarrow{\texttt{B}} \mathrm{at}[\![\texttt{S}_t]\!] \frown \pi_2 \in \widehat{\mathscr{S}}^*[\![\texttt{S}]\!](\pi_1{}^{\ell})}$$

Définition structurelle de la sémantique de traces préfixes III

- Itération `S ::= while` $^\ell$ `(B) S`$_b$ (par règles):

  - $$\frac{}{\ell \in \widehat{\mathcal{S}}^*[\![\mathtt{S}]\!](\pi_1\ell)}$$

  - $$\frac{\ell\pi_2\ell \in \widehat{\mathcal{S}}^*[\![\mathtt{S}]\!](\pi_1\ell), \quad \mathcal{B}[\![\mathtt{B}]\!]\rho(\pi_1\ell\pi_2\ell) = \mathrm{ff}}{\ell\pi_2\ell \xrightarrow{\neg(\mathtt{B})} \mathrm{after}[\![\mathtt{S}]\!] \in \widehat{\mathcal{S}}^*[\![\mathtt{S}]\!](\pi_1\ell)}$$

  - $$\frac{\begin{array}{c}\ell\pi_2\ell \in \widehat{\mathcal{S}}^*[\![\mathtt{S}]\!](\pi_1\ell), \quad \mathcal{B}[\![\mathtt{B}]\!]\rho(\pi_1\ell\pi_2\ell) = \mathrm{tt},\\ \pi_3 \in \widehat{\mathcal{S}}^*[\![\mathtt{S}_b]\!](\pi_1\ell\pi_2\ell \xrightarrow{\mathtt{B}} \mathrm{at}[\![\mathtt{S}_b]\!])\end{array}}{\ell\pi_2\ell \xrightarrow{\mathtt{B}} \mathrm{at}[\![\mathtt{S}_b]\!] \smallfrown \pi_3 \in \widehat{\mathcal{S}}^*[\![\mathtt{S}]\!](\pi_1\ell)}$$

Définition structurelle de la sémantique de traces préfixes IV

- Itération $\mathtt{S} ::= \mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b$ (par point fixe):

$$\mathtt{S} \quad ::= \quad \mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b$$

$$\widehat{\mathcal{S}}^*[\![\mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b]\!] \quad = \quad \mathsf{lfp}^{\dot{\subseteq}}\ \mathsf{F}^*[\![\mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b]\!]$$

$$\mathsf{F}^*[\![\mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b]\!](X)(\pi_1\ell') \quad \triangleq \quad \varnothing \qquad \text{when} \quad \ell' \neq \ell$$

$$\mathsf{F}^*[\![\mathtt{while}\ \ell\ \mathtt{(B)}\ \mathtt{S}_b]\!](X)(\pi_1\ell) \quad \triangleq \quad \{\ell\}$$

$$\cup \left\{\ell'\pi_2\ell' \xrightarrow{\neg(\mathsf{B})} \mathsf{after}[\![\mathtt{S}]\!] \mid \ell'\pi_2\ell' \in X(\pi_1\ell') \wedge \mathscr{B}[\![\mathsf{B}]\!]\boldsymbol{\rho}(\pi_1\ell'\pi_2\ell') = \mathsf{ff} \wedge \ell' = \ell\right\}$$

$$\cup \left\{\ell'\pi_2\ell' \xrightarrow{\mathsf{B}} \mathsf{at}[\![\mathtt{S}_b]\!] \smallfrown \pi_3 \mid \ell'\pi_2\ell' \in X(\pi_1\ell') \wedge \mathscr{B}[\![\mathsf{B}]\!]\boldsymbol{\rho}(\pi_1\ell'\pi_2\ell') = \mathsf{tt} \wedge \pi_3 \in \widehat{\mathcal{S}}^*[\![\mathtt{S}_b]\!](\pi_1\ell'\pi_2\ell' \xrightarrow{\mathsf{B}} \mathsf{at}[\![\mathtt{S}_b]\!]) \wedge \ell' = \ell\right\}$$

Sémantique $\widehat{\mathcal{S}}^{+\infty}[\![\mathsf{s}]\!]$ de traces maximales

- Traces préfixes finies maximales, et
- Traces infinies obtenues comme limites $\lim \mathscr{T}$ d'un ensemble de traces préfixes $\mathscr{T}$:

$$\lim \mathscr{T} \quad \triangleq \quad \{\pi \in \mathbb{T}^{\infty} \mid \forall n \in \mathbb{N} \,.\, \exists p \geqslant n \,.\, \pi[0..p] \in \mathscr{T}\} \,.$$

Propriétés

Propriété

- Une propriété est représentée par l'ensembles des éléments possédant cette propriété
- Entiers pairs: $2\mathbb{Z} \triangleq \{2k \mid k \in \mathbb{Z}\}$
- x possède la propriété P est donc $x \in P$
- L'implication est $P_1 \subseteq P_2$

Propriétés sémantiques

- Une sémantique de traces appartient à $\mathbb{T}^{+} \to \wp(\mathbb{T}^{+\infty})$
- Isomorphe à $\wp(\mathbb{T}^{+} \times \mathbb{T}^{+\infty})$ (représentation d'une relation par son image droite)
- Une propriété sémantique appartient à $\wp(\wp(\mathbb{T}^{+} \times \mathbb{T}^{+\infty}))$
- Une abstraction classique est d'une propriété sémantique en une propriété de traces:

$$\langle \wp(\wp(\mathbb{T}^{+} \times \mathbb{T}^{+\infty})),\ \subseteq\rangle \underset{\lambda P \cdot \cup P}{\overset{\lambda Q \cdot \wp(Q)}{\leftrightarrows}} \langle \wp(\mathbb{T}^{+} \times \mathbb{T}^{+\infty}),\ \subseteq\rangle$$

(sureté (*safety*) et vivacité (*liveness*) sont des propriétés de traces)

Dépendance, informellement

Dépendance fonctionnelle

- Une fonction $f(\ldots, x, \ldots)$ dépend de son paramètre x si et seulement si changer uniquement ce paramètre change le résultat

$$\exists x_1, x_2 \,.\, f(\ldots, x_1, \ldots) \neq f(\ldots, x_2, \ldots)$$

- Exemple: $f(x, y) = x - (y - y)$ dépend de x mais pas de y
- Définition:

$$\mathcal{Fd}^{ni} \triangleq \{ f \mid \exists x_1, \ldots, x_n, x_i' \,.\, f(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n) \neq$$

$$\mathcal{Fd} \triangleq \bigcup_{n \in \mathbb{N}_*} \bigcup_{1 \leqslant i \leqslant n} \mathcal{Fd}^{ni}$$

Non-interférence

- On se donne des variables basses L (par exemple "public" respectivement "fiable/non corrompu") des variables hautes H ("privé/confidentiel" respectivement "douteux/corrompu")
- La non-interférence Goguen and Meseguer, 1982, 1984 est définie comme "si des exécutions coomencent avec les mémes valeurs des variables basses alors, si elles terminent, les variables basses sont égales (donc changer uniquement les variables hautes initiales ne change pas les variables basses finales)
- La propriété de non-interférence est donc

$$\begin{aligned}\mathcal{N}i(L,H) \quad = \quad & \{\Pi \in \wp(\mathbb{T}^{+} \times \mathbb{T}^{\infty}) \mid \forall \langle \pi_0,\ \pi \rangle, \langle \pi'_0,\ \pi' \rangle \in \Pi \cap (\mathbb{T}^{+} \times \mathbb{T}^{+}) \ . \\ & \quad (\forall \mathsf{x} \in L \ . \ \boldsymbol{\rho}(\pi_0)\mathsf{x} = \boldsymbol{\rho}(\pi'_0)\mathsf{x}) \Rightarrow (\forall \mathsf{x} \in L \ . \ \boldsymbol{\rho}(\pi_0 \frown \pi)\mathsf{x} = \boldsymbol{\rho}(\pi'_0 \frown \pi')\mathsf{x})\}\end{aligned}$$

- L'interférence en cours de calcul et la non-terminaison ne sont pas pris en compte.

Dépendance locale

- ℓ_1 `y = 0 ;`ℓ_2 `y = x ;`ℓ_3
  - le futur de `y` en ℓ_1 est la valeur initiale y_0 de `y` en ℓ_1
    Changer la valeur initiale de `x` ne change pas le futur de `y` en ℓ_1 donc `y` ne dépend pas de la valeur initiale de `x` en ℓ_1
  - le futur de `y` en ℓ_2 est 0.
    Changer la valeur initiale de `x` ne change pas le futur de `y` en ℓ_2 donc `y` ne dépend pas de la valeur initiale de `x` en ℓ_2
  - le futur de `y` en ℓ_3 est la valeur initiale x_0 de x.
    Changer la valeur initiale de `x` change le futur de `y` en ℓ_3 donc `y` dépend de la valeur initiale de `x` en ℓ_3

$\Rightarrow$ la notion de dépendance des variables initiales est locale.

Idée générale

- $\mathtt{y}$ dépend de la valeur initiale x_0 de $\mathtt{x}$ en ℓ si et seulement si changer x_0 change les observations futures $\mathtt{y}$ en ℓ
- Plus généralement, changer une abstraction du passé en ℓ change une abstraction du futur aprés ℓ

La dépendance dépend des valeurs

```
if ℓ0 (x == 1) { ℓ1 y = 1 ;ℓ2 } else { ℓ3 y = 2 ;ℓ4 };ℓ5 y = 3 ;ℓ6.
```

- y ne dépend pas de la valeur initiale x_0 de x en ℓ_0, ℓ_1, ℓ_2, ℓ_3, ℓ_4, and ℓ_6
- y dépend de la valeur initiale de x en ℓ_5 (où le futur de y est 1 ou 2 selon x_0)

```
if ℓ0 (x == 1) { ℓ1 y = 1 ;ℓ2 } else { ℓ3 y = 1 ;ℓ4 };ℓ5 y = 3 ;ℓ6.
```

- y ne x ne dépend pas de la valeur initiale x_0 de x en ℓ_0, ℓ_1, …, ℓ_5, ℓ_6

$\Rightarrow$ la dépendance dépend des valeurs[1]

[1](Contrairement à D. E. Denning and P. J. Denning, 1977 qui affirment que “toutes les structures conditionnelles engendrent des flots implicites”, ce qui signifie que toute variable affectée dans un alternative de la conditionnelle dépend des variables apparaissant dans le test.)

Le futur doit comporter des observations multiples

ℓ_1 `y = 0 ;while` ℓ_2 `(`$\mathtt{tt}$`) {` ℓ_3 `y = y + 1 ;`ℓ_4 `y = y + x ;`ℓ_5 `}`

.

- À la première itération y est toujours 1 en ℓ_4
- Aux itérations suivantes y dépend de la valeur initiale de x

- L'observation d'une seule valeur (la première) est insuffisante
- Correct en fin d'exécution auquel cas il n'y a qu'une seule valeur possible

Dépendance dans une itération

$$\mathtt{S} \quad = \quad {}^{\ell_1}\ \mathtt{y = x\ ;while}\ {}^{\ell_0}\ \mathtt{(x < 5)\ \{}\ {}^{\ell_1}\ \mathtt{x = x + 1\ ;}{}^{\ell_2}\ \mathtt{y = y + 1\ ;\ \}}{}^{\ell_3}.$$

- si les valeurs initiales de x et y sont $x_0 = y_0 = 0$, alors l'observation future de y en ℓ_0 est $0 \cdot 1 \cdot 2 \cdot 3 \cdot 4$;
- si $x_0 = 2$ et $y_0 = 0$ elle est $2 \cdot 3 \cdot 4$;
- y dépend de la valeur initiale de x.

$\Rightarrow$ L'observation de la dernière valeur prise (en cas de terminaison) est insuffisante
$\Rightarrow$ Il peut y avoir des dépendances entre variables n'apparaissant pas dans la même affectation ou le même test (voir analyse polyèdrale)
$\Rightarrow$ Il faut observer à partir de la première valeur

Dépendance dans une itération

$\mathtt{P}_u \triangleq$ `while` ℓ `(0==0) x=x+1;`

- On peut observer 17 puis 42
- On peut observer 18 puis 43
- `x` en ℓ dépend de x_0

$\mathtt{P}_0 \triangleq$ `x=0; while` ℓ `(0==0) x=x+1;`

- On peut observer 17 puis 42
- On peut observer 18 puis 43
- `x` en ℓ ne dépend pas de x_0

- Il faut faire des observations maximales des valeurs successives prises en un point:
  - $x_0 \cdot x_0 + 1 \cdot x_0 + 2 \cdot x_0 + 17 \cdot x_0 + 18 \cdot \ldots x_0 + 42 \cdot x_0 + 43 \cdot \ldots$ pour $\mathtt{P}_u$
  - $0 \cdot 1 \cdot 2 \cdot \ldots 17 \cdot 18 \cdot \ldots \cdot 42 \cdot 43 \cdot \ldots$ pour $\mathtt{P}_0$

Observations avec ou sans répétitions

```
ℓ0 i = 0 ;
ℓ1 y = 0 ;
while ℓ2 (0 == 0) {
   if ℓ3 (x >= 0 || i % 2 == 0)
      ℓ4 y = y + 1 ;
   ℓ5 i = i + 1 ;
}
ℓ6
```

- Observation de y en ℓ_5:
  - Si $x_0 \geqslant 0$, on observe $1 \cdot 2 \cdot 3 \cdot \ldots$
  - Si $x_0 < 0$, pourrait observer $1 \cdot 2 \cdot 3 \cdot \ldots$ alors que c'est $1 \cdot 1 \cdot 2 \cdot 2 \cdot 3 \cdot 3 \cdot \ldots$

$\Rightarrow$ il faut tenir compte des répétitions dans les suites d'observations

Observations avec ou sans répétitions (suite)

```
ℓ0 i = 0 ;
ℓ1 y = 0 ;
while ℓ2 (0 == 0) {
   if ℓ3 (x >= 0 || i % 2 == 0)
      ℓ4 y = y + 1 ;
   ℓ5 i = i + 1 ;
}
ℓ6
```

- Observation maximale de y en ℓ_4: toujours $0 \cdot 1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots$
- Donc y en ℓ_4 ne dépend pas de x_0
- D. E. Denning and P. J. Denning, 1977 affirment le contraire, pourquoi?

Observations vides

```
if (x=0) { y=x; ℓ}
```

- Observation de y en ℓ:
  - si $x_0 = 0$ alors "0"
  - si $x_0 \neq 0$ alors "" (observations vide: les exécutions n'atteignant jamais ℓ)

$\Rightarrow$ Dépendance si on prend en compte les observations vides

$\Rightarrow$ Pas de dépendance si on prend pas en compte les observations vides

$\Rightarrow$ Le choix est arbitraire!

$\Rightarrow$ On choisit (arbitrairement) de ne pas faire d'observations vides

Observations temporelles

$\texttt{P}_0 \triangleq$ `x=0; while` ℓ `(0==0) x=x+1;`

$\texttt{P}'_0 \triangleq$ `y=x; x=0; while` ℓ `(0==0) { z=y; while (z>0) { z=z-1; } x=x+1; }`

- L'observation de `x` en ℓ est 0,1,2,...17...42 ... dans les deux cas
- Pour $\texttt{P}_0$, changer y_0 ne change rien
- Pour $\texttt{P}'_0$, changer y_0 change les dates des observations pas la séquence de valeurs observées

$\Rightarrow$ Pour simplifier, on choisit (arbitrairement) d'ignorer les dates des changements de valeur

$\Rightarrow$ On observe juste la séquence maximale des valeurs prises (avec répétition)

"*timing channel*" I

```
while ℓ (x > 0) x = x - 1 ;
```

- Est-ce que y ≠ x en ℓ depend de la valeur initiale x_0 de x?
  - L'observation de y en ℓ est $y_0 \cdot y_0 \cdot \ldots \cdot y_0$ répété $x_0 + 1$ fois.
  - Donc changer x_0 change l'observation de y en ℓ

⇒ C'est un canal caché ("covert/side channel") Lampson, 1973 plus précisément un canal de synchronisation ("timing channel") Russo, Hughes, Naumann, and Sabelfeld, 2006; Sabelfeld and Myers, 2003

⇒ Pour simplifier, on choisit (arbitrairement) d'ignorer les canaux de synchronisation

⇒ Les séquences d'observations doivent différer par au moins une donnée

"*timing channel*" II

ℓ_0 `i = 0 ;`
ℓ_1 `y = 0 ;`
`while` ℓ_2 `(i < 5) {`
  `if` ℓ_3 `(x >= 0 || i % 2 == 0)`
    ℓ_4 `y = y + 1 ;`
  ℓ_5 `i = i + 1 ;`
`}`
ℓ_6

Observation de y en ℓ_4:

- $x_0 \geqslant 0 \longrightarrow 0 \cdot 1 \cdot 2 \cdot 3 \cdot 4$
- $x_0 < 0 \longrightarrow 0 \cdot 1 \cdot 2$

$\Rightarrow$ canal de synchronisation ("timing channel") (les séquences d'observations ne diffèrent pas par au moins une donnée)

Dépendance, formellement

Observations futures

- trace d'initialisation $\pi_0 \in \mathbb{T}^+$
- trace (non vide) de continuation $\pi \in \mathbb{T}^{+\infty}$
- $\mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0,\ \pi)$ est la séquence de valeurs de la y observees successivement au point ℓ dans la trace π continuant π_0[2]

$$
\begin{array}{rcl}
\mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0, \ell) & \triangleq & \boldsymbol{\rho}(\pi_0)\mathsf{y} \\
\mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0, \ell') & \triangleq & \ni \\
\mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0, \ell \xrightarrow{a} \ell''\pi) & \triangleq & \boldsymbol{\rho}(\pi_0)\mathsf{y} \cdot \mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0 \frown \ell \xrightarrow{a} \ell'', \ell''\pi) \\
\mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0, \ell' \xrightarrow{a} \ell''\pi) & \triangleq & \mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0 \frown \ell' \xrightarrow{a} \ell'', \ell''\pi)
\end{array}
$$

- $\mathsf{future}[\![\mathsf{y}]\!]\ell(\pi_0, \pi)$ est la séquence vide $\ni$ si ℓ n'apparait pas dans π

[2] définition bi-inductive P. Cousot and R. Cousot, 2009

Différences entre observations futures ω et ω'

(1) Observation des canaux de synchronisation ("timing channels"):
$\mathsf{tdep}(\omega, \omega') \triangleq \omega \neq \omega'$

(2) Observation des changements de valeurs de la variable y: $\mathsf{vdep}(\omega, \omega')$ où

$$\mathsf{vdep}(\omega, \omega') \quad \triangleq \quad \exists \omega_0, \omega_1, \omega_1', \nu, \nu' \,.\, \omega = \omega_0 \cdot \nu \cdot \omega_1 \wedge \omega' = \omega_0 \cdot \nu' \cdot \omega_1' \wedge \nu \neq \nu'$$

(3) Observation des changements de valeurs et observations vides: $\mathsf{edep}(\omega, \omega')$ où

$$\mathsf{edep}(\omega, \omega') \quad \triangleq \quad \mathsf{vdep}(\omega, \omega') \vee (\omega = \ni \wedge \omega' \neq \ni) \vee (\omega \neq \ni \wedge \omega' = \ni)$$

- D. E. Denning and P. J. Denning, 1977 postulent une définition correspondant à (3)
- Par esprit de contradiction, on élabore la définition structurale de dépendance dans le cas (2)

Définition formelle de la dépendance (de données)

- Propriété de dépendance:

$$\mathcal{D}_{\mathsf{dep}}{}^{\ell}\langle \mathsf{x},\ \mathsf{y}\rangle \quad \triangleq \quad \{\Pi \in \wp(\mathbb{T}^{+} \times \mathbb{T}^{+\infty}) \mid \exists \langle \pi_0,\ \pi_1\rangle, \langle \pi'_0,\ \pi'_1\rangle \in \Pi\ .\ (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}\}\ .\ \boldsymbol{\rho}(\pi_0)\mathsf{z} = \boldsymbol{\rho}(\pi'_0)\mathsf{z}) \wedge \mathsf{dep}(\mathsf{future}[\![\mathsf{y}]\!]^{\ell}(\pi_0,\ \pi_1), \mathsf{future}[\![\mathsf{y}]\!]^{\ell}(\pi'_0,\ \pi'_1))\}$$

- choisir dep = vdep (dépendance de données), ou tdep (canal de synchronisation (“timing channel”)) ou edep (qui est vdep plus possibilité d’une absence d’observation)
- y dépend de la valeur initiale de x au point ℓ du programme P est :

$$\widehat{\mathcal{S}}^{+\infty}[\![\mathsf{P}]\!] \quad \in \quad \mathcal{D}_{\mathsf{dep}}{}^{\ell}\langle \mathsf{x},\ \mathsf{y}\rangle$$

- Pas de distinction nécessaire entre flots explicites et implicites comme dans D. E. Denning and P. J. Denning, 1977

Pourquoi des traces maximales?

- Avec des traces préfixes, si une trace est dans la sémantique, ses préfixes le sont également, ce qui introduit des canaux de synchronization (“timing channels”) artificiels

Traces préfixes pour les dépendances de valeurs

- Pour les dépendances de valeurs, la sémantique de traces maximales peut être remplacée par sémantique de traces préfixes sans problème:

  Lemma $\mathcal{S}^{+\infty}[\![\mathsf{P}]\!] \in \mathcal{D}_{\mathsf{vdep}}\,^{\ell}\langle \mathsf{x},\ \mathsf{y}\rangle \Leftrightarrow \mathcal{S}^{*}[\![\mathsf{P}]\!] \in \mathcal{D}_{\mathsf{vdep}}\,^{\ell}\langle \mathsf{x},\ \mathsf{y}\rangle$

- Idem si on inclut les observations vides (les préfixes de $\mathcal{S}^{*}[\![\mathsf{P}]\!]\pi_0$ ne sont jamais vides)

Dépendance, abstraction

Abstraction en dépendance de données

- L'abstraction d'une propriété sémantique $\mathcal{S} \in \wp(\wp(\mathbb{T}^+ \times \mathbb{T}^{+\infty}))$ en une propriété de dépendance de données $\alpha^{\mathrm{d}}(\mathcal{S}) \in \mathbb{L} \to \wp(\mathbb{V} \times \mathbb{V})$ est :

$$\boxed{\alpha^{\mathrm{d}}(\mathcal{S})\ell \quad \triangleq \quad \{\langle \mathsf{x},\ \mathsf{y}\rangle \mid \mathcal{S} \in \mathcal{D}_{\mathsf{vdep}}\,\ell\langle \mathsf{x},\ \mathsf{y}\rangle\}}$$

- C'est une correspondance de Galois :

Lemma 1 $\langle \wp(\wp(\mathbb{T}^+ \times \mathbb{T}^{+\infty})),\ \subseteq\rangle \xrightleftharpoons[\alpha^{\mathrm{d}}]{\gamma^{\mathrm{d}}} \langle \mathbb{L} \to \wp(\mathbb{V} \times \mathbb{V}),\ \supseteq^{\mathrm{d}}\rangle$ où la concrétisation d'une propriété de dépendance $\mathbf{D} \in \mathbb{L} \to \wp(\mathbb{V} \times \mathbb{V})$ est :

$$\gamma^{\mathrm{d}}(\mathbf{D}) \quad \triangleq \quad \bigcap_{\ell \in \mathbb{L}} \ \bigcap_{\langle \mathsf{x},\ \mathsf{y}\rangle \in \mathbf{D}(\ell)} \mathcal{D}_{\mathsf{vdep}}\,\ell\langle \mathsf{x},\ \mathsf{y}\rangle$$

(plus il y a de sémantiques, moins il y a de dépendances communes)

Dépendance, analyse statique

Dépendance potentielle

- $\alpha^{\mathrm{d}}(\{\mathcal{S}^{*}[\![\mathtt{s}]\!]\})$ n'est pas calculable (théorème de Rice)
- On conçoit une sur-approximation:

> *Sémantique de dépendance potentielle* $\widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}$ *:*
>
> $$\alpha^{\mathrm{d}}(\{\mathcal{S}^{+\infty}[\![\mathtt{s}]\!]\}) \quad \dot{\subseteq} \quad \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{s}]\!]$$

- L'abstraction de D. E. Denning and P. J. Denning, 1977 est purement syntaxique
- On fait un petit peu mieux en prenant en compte la sémantique de façon simple.

Méthode de conception

- Par calcul (en principe, peut être vérifié en Coq comme Jourdan, Laporte, Blazy, Leroy, and Pichardie, 2015)
- Par induction structurelle sur la syntaxe du programme
- Par approximation de point fixe pour les itérations :

  Theorem (sur-approximation de point fixe) Si $\langle \mathscr{C}, \sqsubseteq, \bot, \top, \sqcup, \sqcap \rangle$ et $\langle \mathscr{A}, \preccurlyeq, 0, 1, \curlyvee, \curlywedge \rangle$ sont des treillis complets, $\langle \mathscr{C}, \sqsubseteq \rangle \underset{\alpha}{\overset{\gamma}{\leftrightarrows}} \langle \mathscr{A}, \preccurlyeq \rangle$ est une correspondance de Galois, $f \in \mathscr{C} \xrightarrow{\nearrow} \mathscr{C}$ et $\overline{f} \in \mathscr{A} \xrightarrow{\nearrow} \mathscr{A}$ sont croissantes et $\alpha \circ f \mathrel{\dot{\preccurlyeq}} \overline{f} \circ \alpha$ (*semi-commutation*) alors $\mathsf{lfp}^{\sqsubseteq} f \sqsubseteq \gamma(\mathsf{lfp}^{\preccurlyeq} \overline{f})$.

- Domaine fini, pas besoin d'élargissement

Sémantique de dépendance potentielle de l'affectation $\texttt{S ::= x = A ;}$

$$
\begin{array}{rcl}
\widehat{\mathcal{S}}^{\,\mathrm{d}}_{\exists}[\![\mathtt{S}]\!]\,\ell & = & (\!\!|\ \ell = \mathrm{at}[\![\mathtt{S}]\!]\ ?\ \{\langle \mathtt{y},\ \mathtt{y}\rangle \mid \mathtt{y} \in \mathcal{V}\} \\
 & & [\!]\ \ell = \mathrm{after}[\![\mathtt{S}]\!]\ ?\ \{\langle \mathtt{y},\ \mathtt{x}\rangle \mid \mathtt{y} \in \widehat{\mathcal{S}}^{\,\mathrm{d}}_{\exists}[\![\mathtt{A}]\!]\} \cup \{\langle \mathtt{y},\ \mathtt{y}\rangle \mid \mathtt{y} \neq \mathtt{x}\} \\
 & & \mathbin{\raisebox{0.3ex}{:}}\ \varnothing\ |\!\!) \\
\widehat{\mathcal{S}}^{\,\mathrm{d}}_{\exists}[\![\mathtt{A}]\!] & \triangleq & \{\mathtt{y} \mid \exists \rho \in \mathbb{E}\mathrm{v}\ .\ \exists \nu \in \mathbb{V}\ .\ \mathscr{E}[\![\mathtt{A}]\!]\rho \neq \mathscr{E}[\![\mathtt{A}]\!]\rho[\mathtt{y} \leftarrow \nu]\} \quad \subseteq \quad \mathrm{vars}[\![\mathtt{A}]\!]
\end{array}
$$

Exemple:

- après $\texttt{x = y - y ;}$, $\texttt{x}$ ne dépend pas de y.
- après $\texttt{x = y ; x = y - x ;}$, $\texttt{x}$ dépend des valeurs initiales de $\texttt{x}$ et $\texttt{y}$ (pour être plus précis, il faut conserver une information sur les valeurs)

Preuve I

The case $\ell = \mathsf{at}[\![\mathsf{S}]\!]$ was handled in (43.27). Assume $\ell = \mathsf{after}[\![\mathsf{S}]\!]$.

$$\alpha^{\mathrm{d}}(\{\mathcal{S}^{+\infty}[\![\mathsf{S}]\!]\})\ \mathsf{after}[\![\mathsf{S}]\!]$$

$$= \alpha^{\mathrm{d}}(\{\mathcal{S}^{+}[\![\mathsf{S}]\!]\})\ \mathsf{after}[\![\mathsf{S}]\!] \quad \text{(def. (7.6) of } \mathcal{S}^{+\infty}[\![\mathsf{S}]\!] \text{ since the assignment } \mathsf{S} \text{ has only finite prefix traces)}$$

$$= \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \mathcal{S}^{+}[\![\mathsf{S}]\!] \in \mathcal{D}_{\mathsf{vdep}}(\mathsf{after}[\![\mathsf{S}]\!])\langle \mathsf{x}', \mathsf{y}\rangle\} \quad \text{(def. (43.20) of } \alpha^{\mathrm{d}} \text{ and def. } \subseteq\text{)}$$

$$= \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \exists \langle \pi_0, \pi_1 \mathsf{after}[\![\mathsf{S}]\!]\pi_2\rangle, \langle \pi'_0, \pi'_1 \mathsf{after}[\![\mathsf{S}]\!]\pi'_2\rangle \in \mathcal{S}^{+}[\![\mathsf{S}]\!] \mathrel{.} (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\} \mathrel{.} \boldsymbol{\rho}(\pi_0)\mathsf{z} = \boldsymbol{\rho}(\pi'_0)\mathsf{z}) \wedge \mathsf{after}[\![\mathsf{S}]\!] \notin \pi_1 \wedge \mathsf{after}[\![\mathsf{S}]\!] \notin \pi'_1 \wedge \mathsf{vdep}(\mathsf{future}[\![\mathsf{y}]\!]\mathsf{after}[\![\mathsf{S}]\!](\pi_0 \frown \pi_1\mathsf{after}[\![\mathsf{S}]\!], \mathsf{after}[\![\mathsf{S}]\!]\pi_2), \mathsf{future}[\![\mathsf{y}]\!]\mathsf{after}[\![\mathsf{S}]\!](\pi'_0 \frown \pi'_1\mathsf{after}[\![\mathsf{S}]\!], \mathsf{after}[\![\mathsf{S}]\!]\pi'_2))\}$$

(def. (43.14) of $\mathcal{D}_{\mathsf{vdep}}\,\ell\langle \mathsf{x}', \mathsf{y}\rangle$ and (43.12) of $\mathsf{future}[\![\mathsf{y}]\!]\mathsf{after}[\![\mathsf{S}]\!]$ starting at the first occurence of ℓ in $\pi 1 \mathsf{after}[\![\mathsf{S}]\!]\pi 2$ and $\pi'_1\mathsf{after}[\![\mathsf{S}]\!]\pi'_2$)

$$= \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \exists \langle \pi_0, \pi_1 \mathsf{after}[\![\mathsf{S}]\!]\pi_2\rangle, \langle \pi'_0, \pi'_1 \mathsf{after}[\![\mathsf{S}]\!]\pi'_2\rangle \in \{\langle \pi\mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathcal{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle \mid \pi\mathsf{at}[\![\mathsf{S}]\!] \in \mathbb{T}^{+}\} \mathrel{.} (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\} \mathrel{.} \boldsymbol{\rho}(\pi_0)\mathsf{z} = \boldsymbol{\rho}(\pi'_0)\mathsf{z}) \wedge \mathsf{after}[\![\mathsf{S}]\!] \notin \pi_1 \wedge \mathsf{after}[\![\mathsf{S}]\!] \notin \pi'_1 \wedge \mathsf{vdep}(\mathsf{future}[\![\mathsf{y}]\!]\mathsf{after}[\![\mathsf{S}]\!](\pi_0 \frown \pi_1\mathsf{after}[\![\mathsf{S}]\!], \mathsf{after}[\![\mathsf{S}]\!]\pi_2), \mathsf{future}[\![\mathsf{y}]\!]\mathsf{after}[\![\mathsf{S}]\!](\pi'_0 \frown \pi'_1\mathsf{after}[\![\mathsf{S}]\!], \mathsf{after}[\![\mathsf{S}]\!]\pi'_2))\}$$

(def. maximal finite trace semantics in Section **6.4** and (6.10))

$$= \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \exists \langle \pi_0\mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathcal{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle, \langle \pi'_0\mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathcal{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle \mathrel{.} (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\} \mathrel{.} \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z} = \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z}) \wedge \mathsf{vdep}(\mathsf{future}[\![\mathsf{y}]\!]\mathsf{after}[\![\mathsf{S}]\!](\pi_0\mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathcal{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!], \mathsf{after}[\![\mathsf{S}]\!]), \mathsf{future}[\![\mathsf{y}]\!]\mathsf{after}[\![\mathsf{S}]\!](\pi'_0\mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathcal{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!], \mathsf{after}[\![\mathsf{S}]\!]))\} \quad \text{(def. } \in\text{)}$$

Preuve II

$$= \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \exists \langle \pi_0 \mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle, \langle \pi'_0 \mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle \ . \\ (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\} \ . \ \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z} = \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z}) \wedge \mathsf{vdep}(\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{y} \cdot \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!])\mathsf{y}, \\ \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{y} \cdot \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!])\mathsf{y})\} \qquad \text{≀def. (43.12) of future}[\![\mathsf{y}]\!]\text{∫}$$

$$\subseteq \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \exists \langle \pi_0 \mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle, \langle \pi'_0 \mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle \ . \\ (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\} \ . \ \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z} = \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z}) \wedge ((\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{y} \neq \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{y}\cdot) \vee (\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{y} = \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{y}\cdot) \wedge \\ \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!])\mathsf{y} \neq \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!])\mathsf{y}))\}$$

$$\text{≀(43.13) so that vdep}(a \cdot b,\ c \cdot d) \text{ if and only if (1) } a \neq c \text{ or (2) } a = c \wedge b \neq d.\text{∫}$$

$$\subseteq \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \exists \langle \pi_0 \mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle, \langle \pi'_0 \mathsf{at}[\![\mathsf{S}]\!], \mathsf{at}[\![\mathsf{S}]\!] \xrightarrow{\mathsf{x} = \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])} \mathsf{after}[\![\mathsf{S}]\!]\rangle \ . \\ (\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\} \ . \ \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z} = \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z}) \wedge ((\mathsf{y} = \mathsf{x}') \vee (\mathsf{y} = \mathsf{x} \wedge \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!]) \neq \mathscr{E}[\![\mathsf{A}]\!]\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])))\} \\ \text{≀def. (6.2) of } \boldsymbol{\rho}\text{∫}$$

$$\subseteq \{\langle \mathsf{x}', \mathsf{y}\rangle \mid ((\mathsf{y} = \mathsf{x}') \vee (\mathsf{y} = \mathsf{x} \wedge \exists \rho, \nu \ . \ \mathscr{E}[\![\mathsf{A}]\!]\rho \neq \mathscr{E}[\![\mathsf{A}]\!]\rho[\mathsf{x}' \leftarrow \nu]))\}$$

≀letting $\rho = \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])$ and $\nu = \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])(\mathsf{x}')$ so that $\forall \mathsf{z} \in \mathbb{V} \setminus \{\mathsf{x}'\} \ . \ \boldsymbol{\rho}(\pi_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z} = \boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!])\mathsf{z}$ implies that $\boldsymbol{\rho}(\pi'_0\mathsf{at}[\![\mathsf{S}]\!]) = \rho[\mathsf{x}' \leftarrow \nu]$∫

$$\subseteq \{\langle \mathsf{x}', \mathsf{x}'\rangle \mid \mathsf{x}' \neq \mathsf{x}\} \cup \{\langle \mathsf{x}', \mathsf{x}\rangle \mid \exists \rho, \nu \ . \ \mathscr{E}[\![\mathsf{A}]\!]\rho \neq \mathscr{E}[\![\mathsf{A}]\!]\rho[\mathsf{x}' \leftarrow \nu]\} \qquad \text{≀case analysis∫}$$

Preuve III

$= \{\langle \mathtt{x}', \mathtt{x}' \rangle \mid \mathtt{x}' \neq \mathtt{x}\} \cup \{\langle \mathtt{x}', \mathtt{x} \rangle \mid \mathtt{x}' \in \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{A}]\!]\}$

$\wr$by defining the functional dependency of an expression A as $\widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{A}]\!] \triangleq \{\mathtt{x}' \mid \exists \rho, \nu \,.\, \mathscr{E}[\![\mathtt{A}]\!]\rho \neq \mathscr{E}[\![\mathtt{A}]\!]\rho[\mathtt{x}' \leftarrow \nu]\}$$\wr$

$\square$

Sémantique de dépendance potentielle de la conditionnelle $\mathtt{S} ::= \mathtt{if\ (B)\ S}_t$

$$
\begin{aligned}
\widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{S}]\!]\,\ell \quad = \quad & (\!\![\ \ell = \mathsf{at}[\![\mathtt{S}]\!] \mathrel{?} \{\langle \mathtt{x}',\ \mathtt{x}'\rangle \mid \mathtt{x}' \in \mathbb{V}\} \\
& \ \|\ \ell \in \mathsf{in}[\![\mathtt{S}_t]\!] \cup (\!\![\ \mathsf{escape}[\![\mathtt{S}_t]\!] \mathrel{?} \{\mathsf{break\text{-}to}[\![\mathtt{S}_t]\!]\} \mathrel{\text{\large ⁏}} \varnothing\ ]\!\!) \mathrel{?} \\
& \qquad\qquad\qquad\qquad \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{S}_t]\!]\,\ell\ \rceil\ \mathsf{inondet}(\mathtt{B}, \mathtt{B}) \\
& \ \|\ \ell = \mathsf{after}[\![\mathtt{S}]\!] \mathrel{?} (\widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{S}_t]\!]\ \mathsf{after}[\![\mathtt{S}_t]\!]\ \rceil\ \mathsf{inondet}(\mathtt{B}, \mathtt{B})) \\
& \qquad\qquad\qquad \cup \{\langle \mathtt{x}',\ \mathtt{x}'\rangle \mid \mathtt{x}' \in \mathsf{inondet}(\neg\mathtt{B}, \neg\mathtt{B})\} \\
& \qquad\qquad\qquad\quad \cup \{\langle \mathtt{x}',\ \mathtt{y}\rangle \mid \mathtt{x}' \in \mathsf{inondet}(\mathtt{B}, \neg\mathtt{B}) \wedge \mathtt{y} \in \mathsf{mod}[\![\mathtt{S}_t]\!]\} \\
& \ \mathrel{\text{\large ⁏}} \varnothing\ ]\!\!)
\end{aligned}
$$

$$
\mathsf{inondet}(\mathtt{B}_1, \mathtt{B}_2) \triangleq \{\mathtt{x}' \mid \exists \rho, \nu\ .\ \rho(\mathtt{x}') \neq \nu \wedge \mathscr{B}[\![\mathtt{B}_1]\!]\rho = \mathscr{B}[\![\mathtt{B}_2]\!]\rho[\mathtt{x}' \leftarrow \nu] = \mathrm{tt}\}
$$

$$
\begin{aligned}
\mathsf{mod}[\![\mathtt{x = E ;}]\!] &\triangleq \{\mathtt{x}\} \\
\mathsf{mod}[\![\mathtt{;}]\!] \triangleq \mathsf{mod}[\![\,\epsilon\,]\!] \triangleq \mathsf{mod}[\![\mathtt{break ;}]\!] &\triangleq \varnothing \\
\mathsf{mod}[\![\mathtt{while\ (B)\ S}]\!] = \mathsf{mod}[\![\mathtt{if\ (B)\ S}]\!] &\triangleq \mathsf{mod}[\![\mathtt{S}]\!] \\
\mathsf{mod}[\![\mathtt{if\ (B)\ S}_t\ \mathtt{else\ S}_f]\!] &\triangleq \mathsf{mod}[\![\mathtt{S}_t]\!] \cup \mathsf{mod}[\![\mathtt{S}_f]\!] \\
\mathsf{mod}[\![\mathtt{\{\ Sl\ \}}]\!] &\triangleq \mathsf{mod}[\![\mathtt{Sl}]\!] \\
\mathsf{mod}[\![\mathtt{Sl\ S}]\!] &\triangleq \mathsf{mod}[\![\mathtt{Sl}]\!] \cup \mathsf{mod}[\![\mathtt{S}]\!]
\end{aligned}
$$

Note sur la sémantique de dépendance potentielle de la conditionnelle $\mathtt{S} ::= \mathtt{if\ (B)\ S}_t$

- Les observations vides ne sont pas prises en compte
- ℓ_0 `if (x=0) { y=x;` ℓ_1 `}` ℓ_2
  - y ne dépend pas de x en ℓ_0 ni ℓ_1
  - y dépend de x en ℓ_2

Sémantique de dépendance potentielle de la composition séquentielle $\mathtt{Sl} ::= \mathtt{Sl}'\ \mathtt{S}$

$$
\begin{aligned}
\widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{Sl}]\!]\,\ell \quad \triangleq \quad & (\!|\ \ell \in \mathrm{labx}[\![\mathtt{Sl}']\!] \ ?\ \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{Sl}']\!]\,\ell \\
& \|\ \ell \in \mathrm{labx}[\![\mathtt{S}]\!] \setminus \{\mathrm{at}[\![\mathtt{S}]\!]\} \ ?\ \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{Sl}']\!]\,\mathrm{at}[\![\mathtt{S}]\!] \mathbin{\text{;}} \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{S}]\!]\,\ell \\
& \mathbin{\text{:}} \varnothing\ |\!)
\end{aligned}
$$

Sémantique de dépendance potentielle de l'itération $\texttt{S ::= while}\ \ell\ \texttt{(B)}\ \texttt{S}_b$

$$\widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\texttt{S}]\!]\,\ell' \quad = \quad (\mathsf{lfp}^{\dot{\subseteq}}\,\mathsf{F}^{\mathrm{d}}[\![\texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b]\!])\,\ell'$$

$$\begin{aligned}
&\mathsf{F}^{\mathrm{d}}[\![\texttt{while}\ \ell\ \texttt{(B)}\ \texttt{S}_b]\!]\,X\,\ell' \quad = \\
&\quad (\!|\ \ell' = \ell \ \text{?}\ \mathbb{1}_{\mathbb{V}} \cup X(\ell) \cup (X(\ell) \,\text{⨾}\, \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\texttt{S}_b]\!]\,\ell) \\
&\quad \|\ \ell' \in \mathsf{in}[\![\texttt{S}]\!] \cup (\!|\ \mathsf{escape}[\![\texttt{S}]\!]\ \text{?}\ \{\mathsf{break\text{-}to}[\![\texttt{S}]\!]\} \mathbin{:} \varnothing\ |\!)\ \text{?}\ X(\ell') \cup (X(\ell) \,\text{⨾}\, \widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\texttt{S}_b]\!]\,\ell') \\
&\quad \|\ \ell' = \mathsf{after}[\![\texttt{S}]\!]\ \text{?}\ X(\ell) \cup \{\langle \mathsf{x}', \mathsf{y}\rangle \mid \mathsf{x}' \in \mathsf{vars}[\![\texttt{B}]\!] \wedge \mathsf{y} \in \mathsf{mod}[\![\texttt{S}_b]\!]\} \\
&\quad \mathbin{:} \varnothing\ |\!)
\end{aligned}$$

- Peut être raffiné comme pour la conditionnelle

Exemple

$$\texttt{S} \quad = \quad \texttt{while}\ ^{\ell_0}\ (\mathtt{tt})\ \{\ ^{\ell_1}\ \texttt{y = z ;}^{\ell_2}\ \texttt{z = x ; }\}^{\ell_3}.$$

Le système d'équations $X = \mathsf{F}^{\mathsf{d}}[\![\mathsf{S}]\!](X)$ est

$$\begin{cases} X(\ell_0) & = & \{\langle \mathsf{v},\ \mathsf{v}\rangle \mid \mathsf{v} \in \mathbb{V}\} \cup (X(\ell_2) \fatsemi \{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{x},\ \mathsf{z}\rangle, \langle \mathsf{y},\ \mathsf{y}\rangle\}) \\ X(\ell_1) & = & X(\ell_0) \\ X(\ell_2) & = & X(\ell_2) \cup (X(\ell_1) \fatsemi \{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{z},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{z}\rangle\}) \\ X(\ell_3) & = & \varnothing \end{cases}$$

Les itérations chaotiques sont

ℓ	ℓ_0, ℓ_1	ℓ_2	ℓ_3
$X^0(\ell)$	$\varnothing$	$\varnothing$	$\varnothing$
$X^1(\ell)$	$\{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{y},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{z}\rangle\}$	$\{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{z},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{z}\rangle\}$	$\varnothing$
$X^2(\ell)$	$\{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{x},\ \mathsf{z}\rangle, \langle \mathsf{y},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{z}\rangle\}$	$\{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{x},\ \mathsf{y}\rangle, \langle \mathsf{x},\ \mathsf{z}\rangle, \langle \mathsf{z},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{z}\rangle\}$	$\varnothing$
$X^3(\ell)$	$\{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{x},\ \mathsf{y}\rangle, \langle \mathsf{x},\ \mathsf{z}\rangle, \langle \mathsf{y},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{z}\rangle\}$	$\{\langle \mathsf{x},\ \mathsf{x}\rangle, \langle \mathsf{x},\ \mathsf{y}\rangle, \langle \mathsf{x},\ \mathsf{z}\rangle, \langle \mathsf{z},\ \mathsf{y}\rangle, \langle \mathsf{z},\ \mathsf{z}\rangle\}$	$\varnothing$
$X^4(\ell)$	$X^3(\ell_0) = X^3(\ell_1)$	$X^3(\ell_2)$	$\varnothing$

- La valeur initiale x_0 de x coule dans ("*flows*) dans x en ℓ_0 à l'entrée de la boucle, dans z après la première itération et donc dans y après la deuxième itération.
- La valeur initiale y_0 de y coule seulement dans y en ℓ_0 à l'entrée de la boucle.
- La valeur initiale z_0 de z coule dans z en ℓ_0 à l'entrée de la boucle et ensuite dans y après la première itération.

La sémantique de dépendance potentielle n'est pas purement structurelle[3]

- Analyse séparée des commandes :

ℓ_0 y = x ; ℓ_1	x et y en ℓ_1 dépendent de x en ℓ_0.
ℓ_1 y = y - x ; ℓ_2	x et y en ℓ_2 dépendent de x en ℓ_1.

- Composition des analyses dans la composition séquentielle des des commandes :

ℓ_0 y = x ; ℓ_1 y = y - x ; ℓ_2	y en ℓ_2 dépend de x en ℓ_1 qui dépend de x en ℓ_0 donc, par composition, y en ℓ_2 dépend de x en ℓ_0.

- Cependant, $y = 0$ en ℓ_2 et donc y en ℓ_2 ne dépend pas de x en ℓ_0.
- Une définition syntaxique purement structurelle de la dépendance comme $\widehat{\mathcal{S}}^{\mathrm{d}}_{\exists}[\![\mathtt{S}]\!]$ est forcément imprécise (car elle ne tient pas compte des valeurs des variables)

[3] on dirait compositionelle en sémantique dénotationelle.

Amélioration de la précision

- Pour être précis il faut tenir compte des valeurs possibles des variables
- Produit réduit avec une analyse d'accessibilité (par exemple Cortesi, Ferrara, Halder, and Zanioli, 2018; Zanioli and Cortesi, 2011)

Conclusion

La dépendence de données est une interprétation abstraite

- L'analyse de dépendance est une interprétation abstraite
- Ceci englobe non-interférence, "taint" analysis, *etc*.
- L'analyse de dépendance des données pour détecter le parallélisme dans des codes séquentiels Padua and Wolfe, 1986 est également une interprétation abstraite Tzolovski, 1997, Tzolovski, 2002, Ch. 5.
- On a considéré des cas particuliers de dépendance.

Conjecture: toutes les dépendances sont des interprétations abstraites

- La sémantique est un ensemble de calculs $\langle \pi\ell, \ell\pi' \rangle$ (où $\ell \notin \pi$).
- On définit une abstraction du passé $\pi\ell$ (l'état initial dans notre cas)
- On définit une abstraction du futur (la suite des valeurs d'une variable y observées dans $\ell\pi'$ à chaque point ℓ dans $\ell\pi'$).
- On définit une différence des passés (changer uniquement la valeur d'une variable dans notre cas)
- On définit une différence des futurs (tdep, vdep ou edep dans notre cas)
- La dépendance est alors l'abstraction du futur dépend de l'abstraction du passé ssi un changement de l'abstraction du passé change l'abstraction du futur.
- En variant les abstractions et la différence on change les notions de dépendance (et on devrait pouvoir retrouver toute la littérature comme ça).
- De bons exemples sont Giacobazzi and Mastroeni, 2018 pour la non-interférence et Barthe, Grégoire, and Laporte, 2017 pour la protection contre les attaques par des canaux indirects (*side-channels*)

Bibliographie

References I

Assaf, Mounir, David A. Naumann, Julien Signoles, Eric Totel, and Frédéric Tronel (2017). "Hypercollecting semantics and its application to static analysis of information flow". In: *POPL*. ACM, pp. 874–887 (3).

Barthe, Gilles, Benjamin Grégoire, and Vincent Laporte (2017). "Provably secure compilation of side-channel countermeasures". In: *IACR Cryptology ePrint Archive* 2017, p. 1233 (56).

Cheney, James, Amal Ahmed, and Umut A. Acar (2011). "Provenance as dependency analysis". In: *Mathematical Structures in Computer Science* 21.6, pp. 1301–1337 (3).

Cortesi, Agostino, Pietro Ferrara, Raju Halder, and Matteo Zanioli (2018). "Combining Symbolic and Numerical Domains for Information Leakage Analysis". In: *Trans. Computational Science* 31, pp. 98–135 (33, 53).

Cousot, Patrick and Radhia Cousot (2009). "Bi-inductive structural semantics". In: *Inf. Comput.* 207.2, pp. 258–283 (4, 9, 7, 33).

References II

Denning, Dorothy E. and Peter J. Denning (1977). "Certification of Programs for Secure Information Flow". In: *Commun. ACM* 20.7, pp. 504–513 (1, 3–5, 8, 10, 12, 33, 22, 27, 34, 35, 41).

Giacobazzi, Roberto and Isabella Mastroeni (2018). "Abstract Non-Interference: A Unifying Framework for Weakening Information-flow". In: *ACM Trans. Priv. Secur.* 21.2, 9:1–9:31 (56).

Goguen, Joseph A. and José Meseguer (1982). "Security Policies and Security Models". In: *IEEE Symposium on Security and Privacy*. IEEE Computer Society, pp. 11–20 (2, 3, 19).

– (1984). "Unwinding and Inference Control". In: *IEEE Symposium on Security and Privacy*. IEEE Computer Society, pp. 75–87 (2, 19).

Jourdan, Jacques-Henri, Vincent Laporte, Sandrine Blazy, Xavier Leroy, and David Pichardie (2015). "A Formally-Verified C Static Analyzer". In: *POPL*. ACM, pp. 247–259 (17, 13, 12, 16, 5, 42).

References III

Lampson, Butler W. (1973). "A Note on the Confinement Problem". In: *Commun. ACM* 16.10, pp. 613–615 (5, 30).

Padua, David A. and Michael Wolfe (1986). "Advanced Compiler Optimizations for Supercomputers". In: *Commun. ACM* 29.12, pp. 1184–1201 (33, 55).

Russo, Alejandro, John Hughes, David A. Naumann, and Andrei Sabelfeld (2006). "Closing Internal Timing Channels by Transformation". In: *ASIAN*. Vol. 4435. Lecture Notes in Computer Science. Springer, pp. 120–135 (5, 30).

Sabelfeld, Andrei and Andrew C. Myers (2003). "Language-based information-flow security". In: *IEEE Journal on Selected Areas in Communications* 21.1, pp. 5–19 (5, 30).

Tzolovski, Stanislav (1997). "Data Dependence as Abstract Interpretations". In: *SAS*. Vol. 1302. Lecture Notes in Computer Science. Springer, p. 366 (33, 55).

– (15 June 2002). "Raffinement d'analyses par interprétation abstraite". Thèse de doctorat. Palaiseau, France: École polytechnique (33, 55).

References IV

Urban, Caterina and Peter Müller (2018). "An Abstract Interpretation Framework for Input Data Usage". In: *ESOP*. Vol. 10801. Lecture Notes in Computer Science. Springer, pp. 683–710 (3, 33).

Zanioli, Matteo and Agostino Cortesi (2011). "Information Leakage Analysis by Abstract Interpretation". In: *SOFSEM*. Vol. 6543. Lecture Notes in Computer Science. Springer, pp. 545–557 (33, 53).

Fin, Merci

Bon éméritat (mérité) Nicolas!