Pretty-big-step-semantics-based Certified Abstract Interpretation

Martin Bodin, Thomas Jensen, Alan Schmitt

9 Janvier 2014

JFLA 2014

Objectif

Comment passer de façon systématique d'une sémantique à une analyse de programme certifiée ?

Sémantique à grands pas

↓

Analyse de programme

Contexte

Ce que nous avons

- Une formalisation d'un gros langage (JavaScript).
- Sémantique écrite sous une forme particulière (*Pretty-Big-Step*).

Ce que nous voulons

- Définir des analyses.
- Analyses *certifiées*.

Contexte

Ce que nous avons

- Une formalisation d'un gros langage (JAVASCRIPT).
  700+ règles
- Sémantique écrite sous une forme particulière (*Pretty-Big-Step*).

Ce que nous voulons

- Définir des analyses.
- Analyses *certifiées*.

Un Langage jouet : O'WHILE

$s ::=$
- $\mid$ skip
- $\mid$ $s_1 ; s_2$
- $\mid$ $\mathrm{if}\ e\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2$
- $\mid$ $\mathrm{while}\ e\ \mathrm{do}\ s$
- $\mid$ $\mathtt{x} = e$
- $\mid$ $e_1.\mathtt{f} = e_2$
- $\mid$ $\mathrm{delete}\ e.\mathtt{f}$

$e ::=$
- $\mid$ c
- $\mid$ $\mathtt{x}$
- $\mid$ $e_1\ \mathtt{op}\ e_2$
- $\mid$ $\{\}$
- $\mid$ $e.\mathtt{f}$

Pretty-Big-Step

Un problème des sémantiques à grands pas

Duplication des règles dès l'ajout d'exceptions, de divergence, etc.

$$\frac{E, H, \underline{e} \to r \qquad E' = E[\mathrm{x} \mapsto v]}{S, \underline{\mathrm{x} = e} \to E', H}\ \textsc{Asg}$$

Le Pretty-Big-Step

- Une sémantique à grands pas.
- Des formes intermédiaires comme en petits pas.

$$\frac{E, H, \underline{e} \to r \qquad E, H, \underline{\mathrm{x} =_1 r} \to r'}{E, H, \underline{\mathrm{x} = e} \to r'}\ \textsc{Asg} \qquad \frac{E' = E[\mathrm{x} \mapsto v]}{S, \underline{\mathrm{x} =_1 (E, H, v)} \to E', H}\ \textsc{Asg1}$$

Pretty-Big-Step

Un problème des sémantiques à grands pas

Duplication des règles dès l'ajout d'exceptions, de divergence, etc.

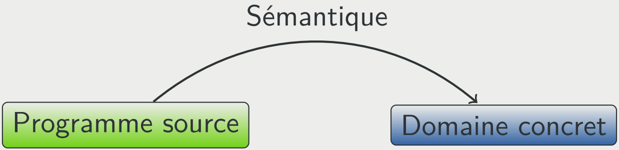
$$\frac{E, H, \underline{e} \rightarrow r \qquad E' = E[\mathtt{x} \mapsto v]}{S, \underline{\mathtt{x} = e} \rightarrow E', H}\ \textsc{Asg}$$

Le Pretty-Big-Step

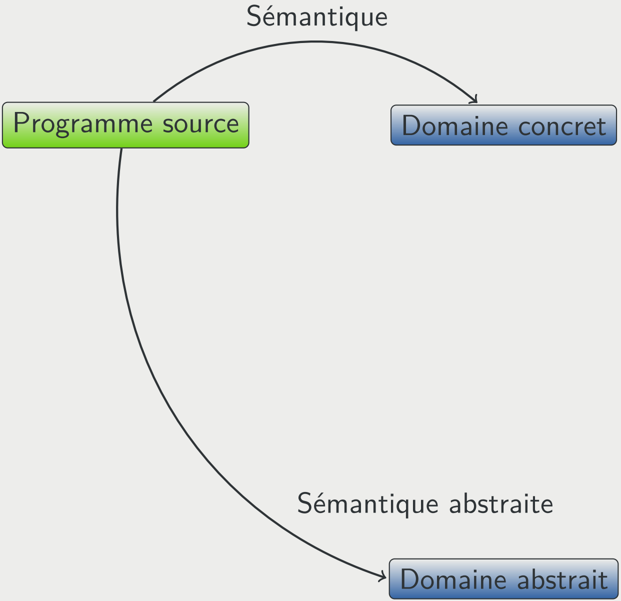
- Une sémantique à grands pas.
- Des formes intermédiaires comme en petits pas.

$$\frac{E, H, \underline{e} \rightarrow r \qquad E, H, \underline{\mathtt{x}\ =_1 r} \rightarrow r'}{E, H, \underline{\mathtt{x} = e} \rightarrow r'}\ \textsc{Asg} \qquad \frac{E' = E[\mathtt{x} \mapsto v]}{S, \underline{\mathtt{x}\ =_1 (E, H, v)} \rightarrow E', H}\ \textsc{Asg1}$$

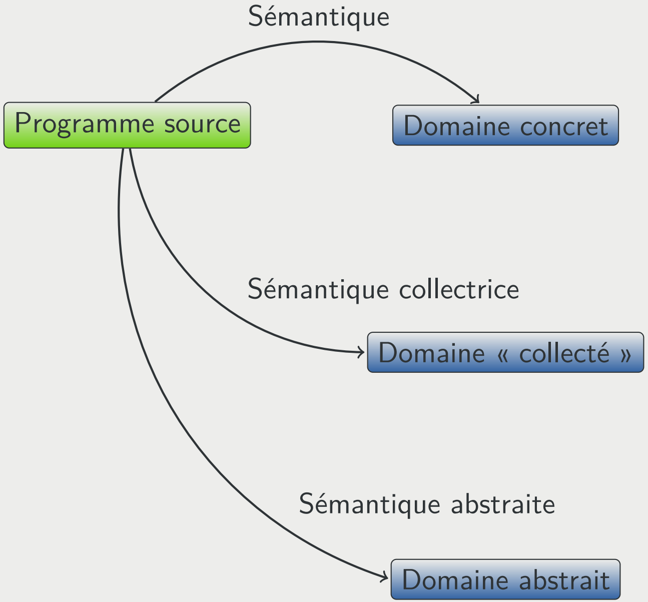
Les Analyses : le Schéma classique



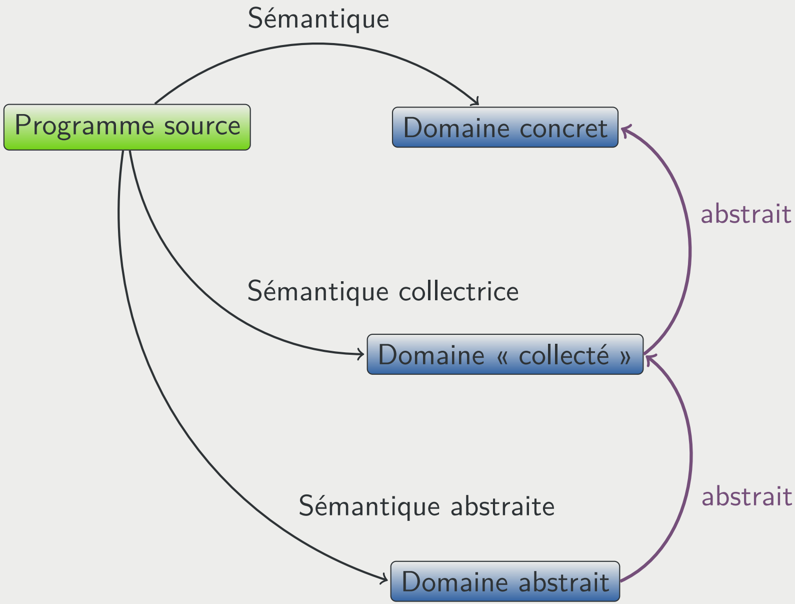
Les Analyses : le Schéma classique



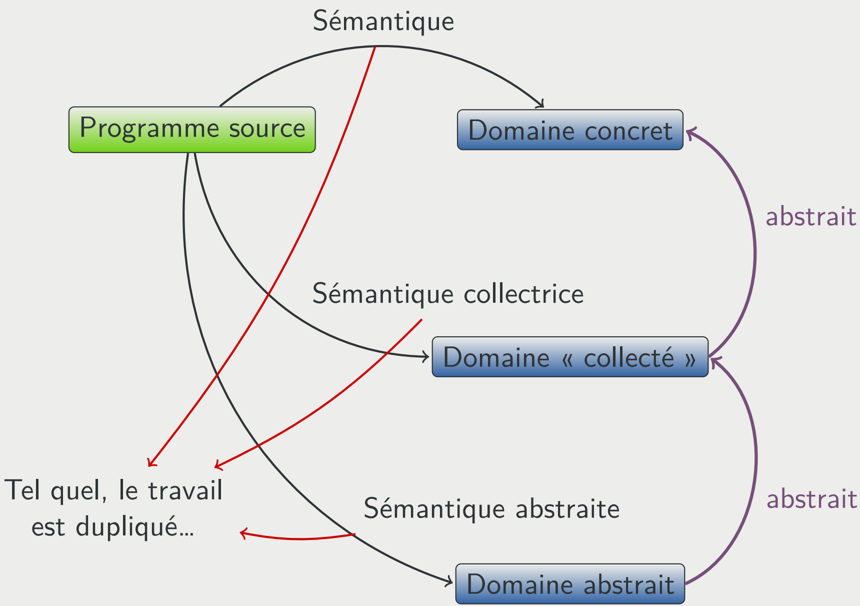
Les Analyses : le Schéma classique

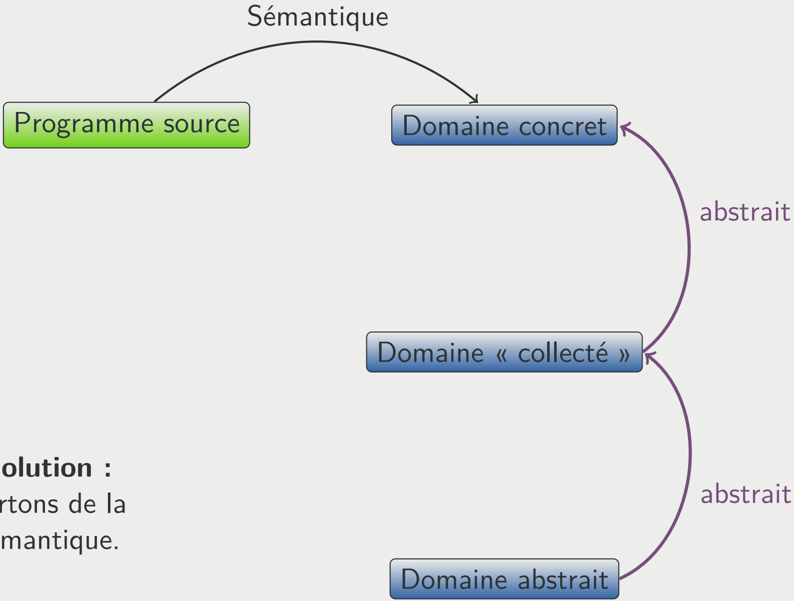


Les Analyses : le Schéma classique



Les Analyses : le Schéma classique

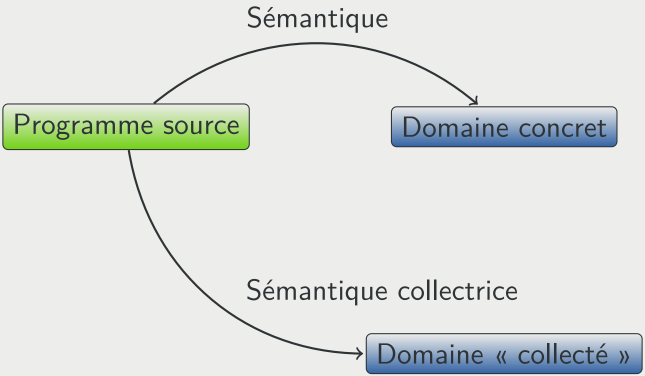




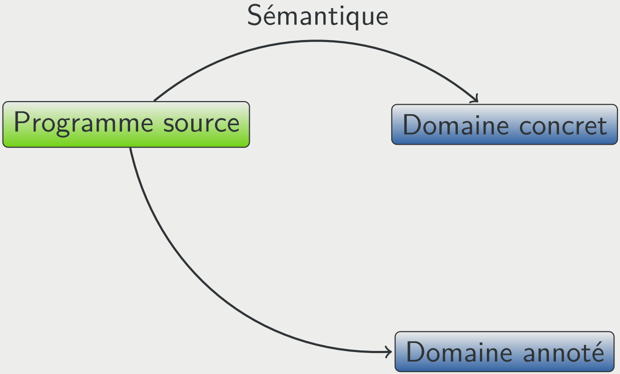
Solution :
partons de la sémantique.

1. Annotations
   - Traces
   - Flots directs
2. Sémantique abstraite

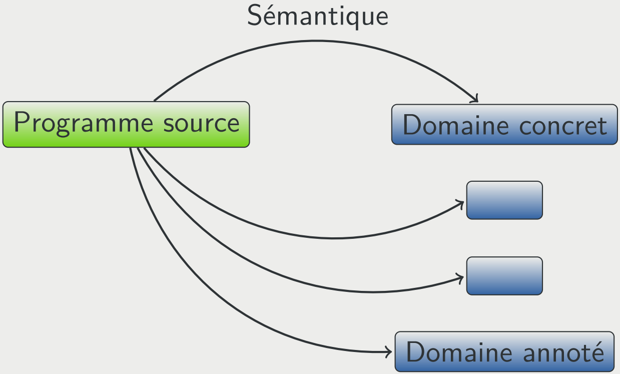
Annoter la sémantique



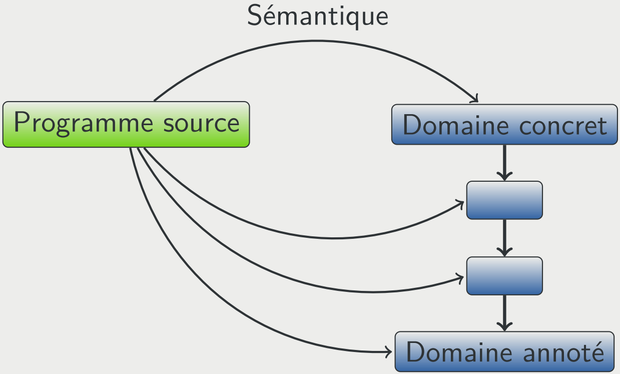
Annoter la sémantique



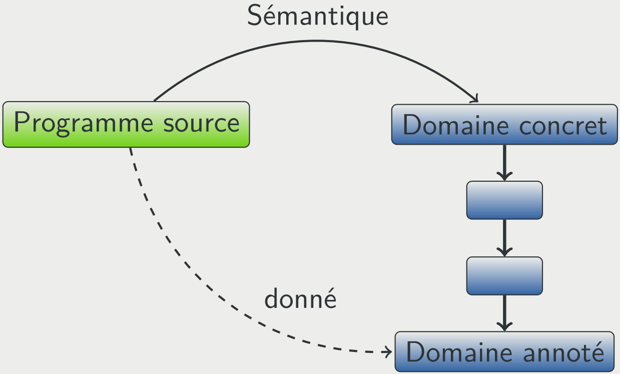
Annoter la sémantique



Annoter la sémantique



Annoter la sémantique



Un exemple de chaîne d'annotations

- Nous voulons détecter des flots du type $\mathrm{x}^{p_1} \looparrowright \mathrm{y}^{p_2}$.

```
t = x ;
/* ... */
y = t
```

- On a besoin d'avoir les traces.
- On a besoin d'avoir un environnement stockant les dates de dernières modifications.

Un exemple de chaîne d'annotations

- Nous voulons détecter des flots du type $\mathtt{x}^{\tau_1} \leadsto \mathtt{y}^{\tau_2}$.

```
t = x ;
/* ... */
y = t
```

- On a besoin d'avoir les traces.
- On a besoin d'avoir un environnement stockant les dates de dernières modifications.

Un exemple de chaîne d'annotations

- Nous voulons détecter des flots du type $\mathtt{x}^{\tau_1} \looparrowright \mathtt{y}^{\tau_2}$.

```
t = x ;
/* ... */
y = t
```

- On a besoin d'avoir les traces.
- On a besoin d'avoir un environnement stockant les dates de dernières modifications.

Un exemple de chaîne d'annotations

$S_\tau, t \to r$	Sémantique standard
$\downarrow$	
$\tau, S_\tau, t \to \tau', r$	Ajout des traces
$\downarrow$	
$\tau, M_\tau, S_\tau, t \to \tau', M_{\tau'}, r$	Dernières modifications
$\downarrow$	
$\tau, M_\tau, D, F_\tau, S_\tau, t \to \tau', M_{\tau'}, D', F_{\tau'}, r$	Flots directs

On a rendu le flot explicite dans la dérivation…
mais *sans ajouter d'information* !

Les Annotations sont locales

N'oublions pas les 700+ règles de JavaScript !

On se refuse tout copier/coller.

En Pretty-Big-step, il n'y a que trois types de règles

$$\frac{}{a \to a} \qquad \frac{a \to a}{a \to a} \qquad \frac{a \to a \qquad a \to a}{a \to a}$$

$$\frac{\dfrac{}{a_1 \to a_2} \qquad \dfrac{\dfrac{}{a_4 \to a_5}}{a_3 \to a_6}}{a_0 \to a_7}$$

Les Annotations sont locales

N'oublions pas les 700+ règles de JavaScript !

On se refuse tout copier/coller.

En Pretty-Big-step, il n'y a que trois types de règles

$$\frac{}{a \to a} \qquad \frac{a \to a}{a \to a} \qquad \frac{a \to a \qquad a \to a}{a \to a}$$

$$\dfrac{\dfrac{}{a_1 \to a_2} \qquad \dfrac{\dfrac{}{a_4 \to a_5}}{a_3 \to a_6}}{a_0 \to a_7}$$

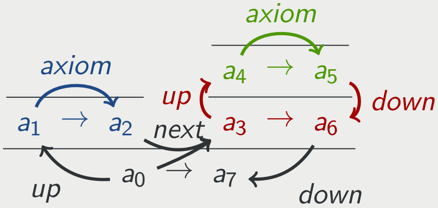
Les Annotations sont locales

N'oublions pas les 700+ règles de JavaScript !

On se refuse tout copier/coller.

En Pretty-Big-step, il n'y a que trois types de règles

$$\frac{}{a \to a} \qquad \frac{a \to a}{a \to a} \qquad \frac{a \to a \qquad a \to a}{a \to a}$$

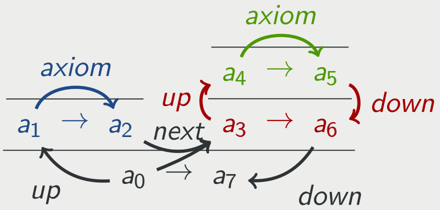


Définition des Traces

Les **traces** τ représentent un point dans l'arbre de dérivation, c'est à dire le *temps d'exécution*.

Sensible au flot

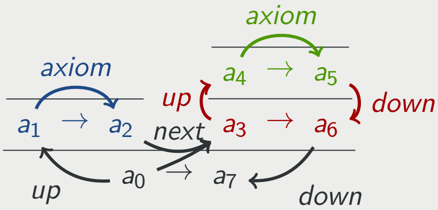
$$\textsc{Asg}\ \frac{S, e \to r_0 \qquad S, \mathtt{x} =_1 r_0 \to r}{S, \mathtt{x} = e \to r}$$



Les **traces** τ représentent un point dans l'arbre de dérivation, c'est à dire le *temps d'exécution*.

Sensible au flot

$$\textsc{Asg}\ \frac{S, e \to r_0 \qquad S, \mathtt{x} =_1 r_0 \to r}{\tau_0, S, \mathtt{x} = e \to r}$$



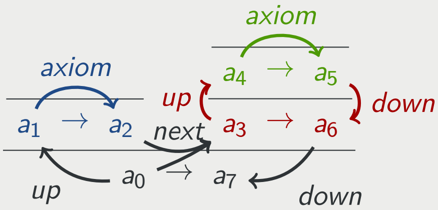
Définition des Traces

Les **traces** τ représentent un point dans l'arbre de dérivation, c'est à dire le *temps d'exécution*.

Sensible au flot

$$\tau_1 = \tau_0 \mathbin{+\!\!+} [\textsc{AsgE}]$$

$$\textsc{Asg}\ \frac{\tau_1, S, e \to r_0 \qquad S, \mathtt{x} =_1 r_0 \to r}{\tau_0, S, \mathtt{x} = e \to r}$$



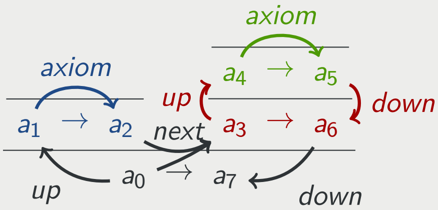
Définition des Traces

Les **traces** τ représentent un point dans l'arbre de dérivation, c'est à dire le *temps d'exécution.*

Sensible au flot

$$\tau_1 = \tau_0 \mathbin{+\!\!+} [\textsc{AsgE}]$$

$$\textsc{Asg}\ \frac{\tau_1, S, e \to \tau_2, r_0 \qquad S, \mathtt{x} =_1 r_0 \to r}{\tau_0, S, \mathtt{x} = e \to r}$$

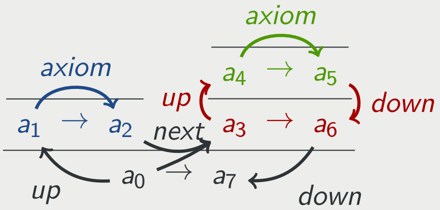


Définition des Traces

Les **traces** τ représentent un point dans l'arbre de dérivation, c'est à dire le *temps d'exécution.*

Sensible au flot

$$\textsc{Asg}\ \frac{\tau_1 = \tau_0 + [\textsc{AsgE}] \qquad \tau_3 = \tau_2 + [\textsc{Asg1}] \qquad \tau_1, S, e \rightarrow \tau_2, r_0 \qquad \tau_3, S, \mathtt{x} =_1 r_0 \rightarrow r}{\tau_0, S, \mathtt{x} = e \rightarrow r}$$



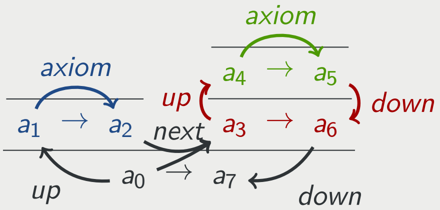
Définition des Traces

Les **traces** τ représentent un point dans l'arbre de dérivation, c'est à dire le *temps d'exécution.*

Sensible au flot

$$\tau_1 = \tau_0 +\!\!\!+ [\mathrm{ASGE}] \qquad \tau_3 = \tau_2 +\!\!\!+ [\mathrm{ASG1}]$$

$$\mathrm{ASG}\ \frac{\tau_1, S, \underline{e} \rightarrow \tau_2, r_0 \qquad \tau_3, S, \underline{\mathtt{x} =_1 r_0} \rightarrow \tau_4, r}{\tau_0, S, \underline{\mathtt{x} = e} \rightarrow \quad r}$$

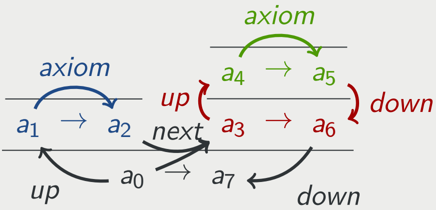


Définition des Traces

Les **traces** τ représentent un point dans l'arbre de dérivation, c'est à dire le *temps d'exécution*.

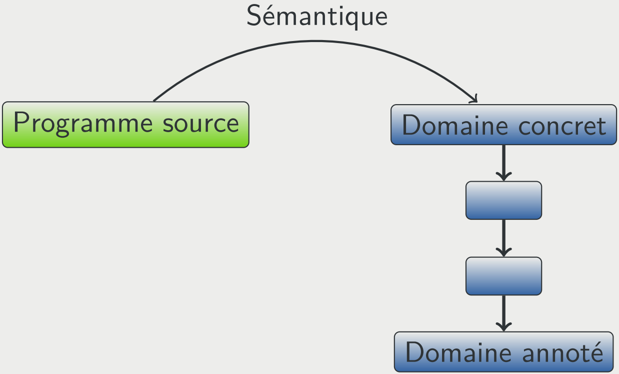
Sensible au flot

$$\textsc{Asg}\ \frac{\tau_5 = \tau_4 \mathbin{+\!\!+} [\textsc{Asg}] \qquad \begin{array}{c}\tau_1 = \tau_0 \mathbin{+\!\!+} [\textsc{AsgE}] \\ \tau_1, S, \underline{e} \to \tau_2, r_0\end{array} \qquad \begin{array}{c}\tau_3 = \tau_2 \mathbin{+\!\!+} [\textsc{Asg1}] \\ \tau_3, S, \underline{\mathtt{x} =_1 r_0} \to \tau_4, r\end{array}}{\tau_0, S, \underline{\mathtt{x} = e} \to \tau_5, r}$$



Et les 700+ règles de JavaScript ?

Si l'on doit définir plusieurs fonctions par règles, cela ne simplifie pas tant que cela la tâche...



En fait, la plupart du temps, ces fonctions ne font que propager leur annotation.

Partir de la sémantique

$$\frac{}{S, \mathrm{skip} \to S}\ \textsc{Skip} \qquad \frac{S, s_1 \to r \qquad S, r ;_1 s_2 \to r'}{S, s_1 ; s_2 \to r'}\ \textsc{Seq} \qquad \frac{S', s \to r}{S, S' ;_1 s \to r}\ \textsc{Seq1} \qquad \frac{S, e \to r \qquad S, \mathrm{if1}(r, s_1, s_2) \to r'}{S, \mathrm{if}\ e\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2 \to r'}\ \textsc{If}$$

$$\frac{S', s_1 \to r}{S, \mathrm{if1}((S', \mathrm{true}), s_1, s_2) \to r}\ \textsc{IfTrue} \qquad \frac{S', s_2 \to r}{S, \mathrm{if1}((S', \mathrm{false}), s_1, s_2) \to r}\ \textsc{IfFalse} \qquad \frac{S, e \to r \qquad S, \mathrm{while1}(r, e, s) \to r'}{S, \mathrm{while}\ e\ \mathrm{do}\ s \to r'}\ \textsc{While}$$

$$\frac{S', s \to r \qquad S', \mathrm{while2}(r, e, s) \to r'}{S, \mathrm{while1}((S', \mathrm{true}), e, s) \to r'}\ \textsc{WhileTrue1} \qquad \frac{S', \mathrm{while}\ e\ \mathrm{do}\ s \to r}{S, \mathrm{while2}(S', e, s) \to r}\ \textsc{WhileTrue2}$$

$$\frac{}{S, \mathrm{while1}((S', \mathrm{false}), e, s) \to S'}\ \textsc{WhileFalse} \qquad \frac{E, H, e \to r \qquad E, H, \mathtt{x} =_1 r \to r'}{E, H, \mathtt{x} = e \to r'}\ \textsc{Asg}$$

$$\frac{E' = E[\mathtt{x} \mapsto v]}{S, \mathtt{x} =_1 (E, H, v) \to E', H}\ \textsc{Asg1} \qquad \frac{S, e_1 \to r \qquad S, r.\mathtt{f} =_1 e_2 \to r'}{S, e_1.\mathtt{f} = e_2 \to r'}\ \textsc{FldAsg} \qquad \frac{S', e \to r \qquad S', l.\mathtt{f} =_2 r \to r'}{S, (S', l).\mathtt{f} =_1 e \to r'}\ \textsc{FldAsg1}$$

$$\frac{H[l] = o \qquad o' = o[\mathtt{f} \mapsto v] \qquad H' = H[l \mapsto o']}{S, l.\mathtt{f} =_2 (E, H, v) \to E, H'}\ \textsc{FldAsg2} \qquad \frac{S, e \to r \qquad S, \mathrm{delete1}\ r.\mathtt{f} \to r'}{S, \mathrm{delete}\ e.\mathtt{f} \to r'}\ \textsc{Del}$$

$$\frac{H[l] = o \qquad o[\mathtt{f}] \neq \bot \qquad o' = o[\mathtt{f} \mapsto \bot] \qquad H' = H[l \mapsto o']}{S, \mathrm{delete1}\ (E, H, l).\mathtt{f} \to E, H'}\ \textsc{Del1} \qquad \frac{}{S, c \to S, c}\ \textsc{Cst} \qquad \frac{E[\mathtt{x}] = v}{E, H, \mathtt{x} \to E, H, v}\ \textsc{Var}$$

$$\frac{S, e_1 \to r \qquad S, r\ \mathrm{op}_1\ e_2 \to r'}{S, e_1\ \mathrm{op}\ e_2 \to r'}\ \textsc{Bin} \qquad \frac{S', e_2 \to r \qquad S', v_1\ \mathrm{op}_2\ r \to r'}{S, (S', v_1)\ \mathrm{op}_1\ e_2 \to r'}\ \textsc{Bin1} \qquad \frac{v = v_1\ \mathrm{op}\ v_2}{S, v_1\ \mathrm{op}_2\ (S, v_2) \to S, v}\ \textsc{Bin2}$$

$$\frac{H[l] = \bot \qquad H' = H[l \mapsto \{\}]}{E, H, \{\} \to E, H', l}\ \textsc{Obj} \qquad \frac{S, e \to r \qquad S, r\ ._1\mathtt{f} \to r'}{S, e.\mathtt{f} \to r'}\ \textsc{Fld} \qquad \frac{H'[l] = o \qquad o[\mathtt{f}] = v}{E, H, (E', H', l)\ ._1\mathtt{f} \to E', H', v}\ \textsc{Fld1} \qquad \frac{\mathtt{abort}(t_e) = r}{S, t_e \to r}\ \textsc{Abort}$$

Partir de la sémantique

$$\frac{}{S, \texttt{skip} \to S}\ \textsc{Skip} \qquad \frac{S, s_1 \to r \qquad S, r ;_1 s_2 \to r'}{S, s_1 ; s_2 \to r'}\ \textsc{Seq} \qquad \frac{S', s \to r}{S, S' ;_1 s \to r}\ \textsc{Seq1} \qquad \frac{S, e \to r \qquad S, \texttt{if1}(r, s_1, s_2) \to r'}{S, \texttt{if}\ e\ \texttt{then}\ s_1\ \texttt{else}\ s_2 \to r'}\ \textsc{If}$$

$$\frac{S', s_1 \to r}{S, \texttt{if1}((S', \texttt{true}), s_1, s_2) \to r}\ \textsc{IfTrue} \qquad \frac{S', s_2 \to r}{S, \texttt{if1}((S', \texttt{false}), s_1, s_2) \to r}\ \textsc{IfFalse} \qquad \frac{S, e \to r \qquad S, \texttt{while1}(r, e, s) \to r'}{S, \texttt{while}\ e\ \texttt{do}\ s \to r'}\ \textsc{While}$$

$$\frac{S', s \to r \qquad S', \texttt{while2}(r, e, s) \to r'}{S, \texttt{while1}((S', \texttt{true}), e, s) \to r'}\ \textsc{WhileTrue1} \qquad \frac{S', \texttt{while}\ e\ \texttt{do}\ s \to r}{S, \texttt{while2}(S', e, s) \to r}\ \textsc{WhileTrue2}$$

$$\frac{}{S, \texttt{while1}((S', \texttt{false}), e, s) \to S'}\ \textsc{WhileFalse} \qquad \frac{E, H, e \to r \qquad E, H, \texttt{x} =_1 r \to r'}{E, H, \texttt{x} = e \to r'}\ \textsc{Asg}$$

$$\frac{E' = E[\texttt{x} \mapsto v]}{S, \texttt{x} =_1 (E, H, v) \to E', H}\ \textsc{Asg1} \qquad \frac{S, e_1 \to r \qquad S, r.\texttt{f} =_1 e_2 \to r'}{S, e_1.\texttt{f} = e_2 \to r'}\ \textsc{FldAsg} \qquad \frac{S', e \to r \qquad S', l.\texttt{f} =_2 r \to r'}{S, (S', l).\texttt{f} =_1 e \to r'}\ \textsc{FldAsg1}$$

$$\frac{H[l] = o \qquad o' = o[\texttt{f} \mapsto v] \qquad H' = H[l \mapsto o']}{S, l.\texttt{f} =_2 (E, H, v) \to E, H'}\ \textsc{FldAsg2} \qquad \frac{S, e \to r \qquad S, \texttt{delete1}\ r.\texttt{f} \to r'}{S, \texttt{delete}\ e.\texttt{f} \to r'}\ \textsc{Del}$$

$$\frac{H[l] = o \qquad o[\texttt{f}] \neq \bot \qquad o' = o[\texttt{f} \mapsto \bot] \qquad H' = H[l \mapsto o']}{S, \texttt{delete1}\ (E, H, l).\texttt{f} \to E, H'}\ \textsc{Del1} \qquad \frac{}{S, c \to S, c}\ \textsc{Cst} \qquad \frac{E[\texttt{x}] = v}{E, H, \texttt{x} \to E, H, v}\ \textsc{Var}$$

$$\frac{S, e_1 \to r \qquad S, r\ \texttt{op}_1\ e_2 \to r'}{S, e_1\ \texttt{op}\ e_2 \to r'}\ \textsc{Bin} \qquad \frac{S', e_2 \to r \qquad S', v_1\ \texttt{op}_2\ r \to r'}{S, (S', v_1)\ \texttt{op}_1\ e_2 \to r'}\ \textsc{Bin1} \qquad \frac{v = v_1\ \texttt{op}\ v_2}{S, v_1\ \texttt{op}_2\ (S, v_2) \to S, v}\ \textsc{Bin2}$$

$$\frac{H[l] = \bot \qquad H' = H[l \mapsto \{\}]}{E, H, \{\} \to E, H', l}\ \textsc{Obj} \qquad \frac{S, e \to r \qquad S, r\ ._1\texttt{f} \to r'}{S, e.\texttt{f} \to r'}\ \textsc{Fld} \qquad \frac{H'[l] = o \qquad o[\texttt{f}] = v}{E, H, (E', H', l)\ ._1\texttt{f} \to E', H', v}\ \textsc{Fld1} \qquad \frac{\texttt{abort}(t_e) = r}{S, t_e \to r}\ \textsc{Abort}$$

Flots directs

$$S_\tau, t \to r$$

Sémantics standard

$$\downarrow$$

$$\tau, S_\tau, t \to \tau', r$$

Ajout des traces

τ : Traces

$$\downarrow$$

$$\tau, M_\tau, S_\tau, t \to \tau', M_{\tau'}, r$$

Dernières modifications

M_τ : Var $\to$ Traces

$$\downarrow$$

$$\tau, M_\tau, D, S_\tau, t \to \tau', M_{\tau'}, D', r$$

Dépendance des expressions

$\mathtt{x}^\tau, l.\mathtt{f}^\tau, l \in D$

$$\downarrow$$

$$\tau, M_\tau, D, F_\tau, S_\tau, t \to \tau', M_{\tau'}, D', F_{\tau'}, r$$

Flots directs

$s \rightsquigarrow t \in F_\tau$

Ces flots $s \rightsquigarrow t$ représentent une dépendance entre :

- Une **source** : un objet l, une variable $\mathtt{x}^\tau$ ou un champ d'un object $l.\mathtt{f}^\tau$.
- Une **cible** : une variable $\mathtt{x}^\tau$ ou un champ d'un object $l.\mathtt{f}^\tau$.

Côté Coq

```
Section LastModified.
Variable Locations : Annotations.

Definition ModifiedAnnots := annot_s_r Locations.
Record LastModifiedHeaps : Type :=
  makeLastModifiedHeaps {
    LCEnvironment : heap var ModifiedAnnots;
    LCHeap : heap loc (heap prop_name ModifiedAnnots)
  }.

Definition LastModified :=
  ConstantAnnotations LastModifiedHeaps.
```

```
Definition LastModifiedAxiom_s (r : LastModifiedHeaps)
  E H t o (R : red_stat Locations E H t o) :=
  let LCE := LCEnvironment r in
  let LCH := LCHeap r in
  let (_, tau) := extract_annot_s R in
  match R with
  | red_stat_ext_stat_assign_1 _ _ _ _ _ _ _ x _ _ ⇒
    let LCE' := write LCE x tau
    in makeLastModifiedHeaps LCE' LCH
  | red_stat_stat_delete _ _ _ _ _ l _ f _ _ _ _ _ _ ⇒
    let aob := read LCH l
    in let LCH' := write LCH l (write aob f tau)
    in makeLastModifiedHeaps LCE LCH'
  | red_stat_ext_stat_set_2 _ _ _ _ _ _ _ _ l _ f _ _ _ _ _ _ ⇒
    let aob := read LCH l
    in let LCH' := write LCH l (write aob f tau)
    in makeLastModifiedHeaps LCE LCH'
  | _ ⇒ makeLastModifiedHeaps LCE LCH
  end.
```

Côté Coq

```
Definition annotLastModified :=
  makeIterativeAnnotations LastModified
    (init_e Transmit) (axiom_e Transmit) (up_e Transmit) (down_e
        Transmit) (next_e Transmit)
    (up_s_e Transmit) (next_e_s Transmit)
    (init_s Transmit) LastModifiedAxiom_s (up_s Transmit) (down_s
        Transmit) (next_s Transmit).

End LastModified.
```

Flots directs

$S_\tau, t \to r$	Sémantics standard
$\downarrow$	
$\tau, S_\tau, t \to \tau', r$	Ajout des traces τ : Traces
$\downarrow$	
$\tau, M_\tau, S_\tau, t \to \tau', M_{\tau'}, r$	Dernières modifications M_τ : Var $\to$ Traces
$\downarrow$	
$\tau, M_\tau, D, S_\tau, t \to \tau', M_{\tau'}, D', r$	Dépendance des expressions $\mathtt{x}^\tau, l.\mathtt{f}^\tau, l \in D$
$\downarrow$	
$\tau, M_\tau, D, F_\tau, S_\tau, t \to \tau', M_{\tau'}, D', F_{\tau'}, r$	Flots directs $s \rightsquigarrow t \in F_\tau$

Ces flots $s \rightsquigarrow t$ représentent une dépendance entre :

- Une **source** : un objet l, une variable $\mathtt{x}^\tau$ ou un champ d'un object $l.\mathtt{f}^\tau$.
- Une **cible** : une variable $\mathtt{x}^\tau$ ou un champ d'un object $l.\mathtt{f}^\tau$.

Correction des annotations

Il reste maintenant à prouver certaines propriétés de correction sur ces annotations.

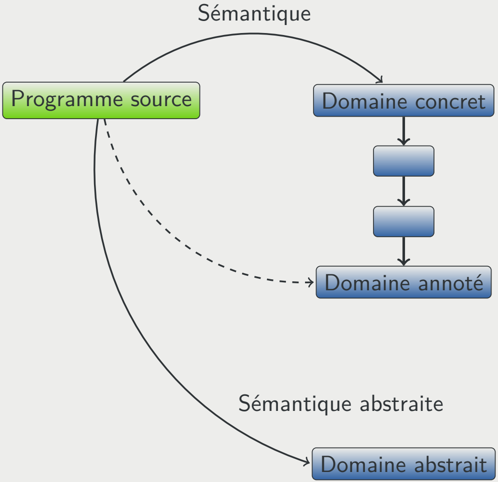
Par exemple

Si $\tau, M_\tau, D, F_\tau, S_\tau, t \rightarrow \tau', M_{\tau'}, D', F_{\tau'}, r$
et $\mathtt{x}^{\tau_1} \mathrel{\subset\!\!\!\!\rightsquigarrow} \mathtt{y}^{\tau_2} \in F_{\tau'}$,
alors $M_{\tau_2}[\mathtt{x}] = \tau_1$.

« Si F affirme que $\mathtt{x}$ au temps τ_1 flotte dans $\mathtt{y}$ au temps τ_2, alors $\mathtt{x}$ n'a pas été modifié entre-temps. »

Mais cela est facilité car la sémantique annotée et la sémantique standard représentent exactement la même dérivation.

1. Annotations
   - Traces
   - Flots directs
2. Sémantique abstraite



Une Abstraction standard: *points-to*

- Les objets sont abstraits par le point de programme où ils ont étés alloués.
$$l^\sharp \in \mathrm{Loc}^\sharp = \mathcal{P}\left(PP\right)$$
- Les valeurs abstraites sont à la fois des locations abstraites et l'ensemble des variables dont elles dépendent.
$$v^\sharp \in \mathrm{Val}^\sharp = \mathrm{Loc}^\sharp \times \mathcal{P}\left(\mathrm{Var} \times PP\right)$$

Flots abstraits

- L'environment et le tas stockent les « temps » de dernières modifications.

$$E^\sharp \in \mathrm{Env}^\sharp = \mathrm{Var} \to \left(\mathcal{P}\left(PP\right) \times \mathrm{Val}^\sharp\right)$$

$$H^\sharp \in \mathrm{Heap}^\sharp = \mathrm{Loc}^\sharp \to \mathrm{Field} \to \left(\mathcal{P}\left(PP\right) \times \mathrm{Val}^\sharp\right)$$

- Les flots abstraits sont abstraits de la même manière :

$$\mathrm{Store}^\sharp = (\mathrm{Var} \times PP) + (PP \times \mathrm{Field} \times PP) \qquad \mathrm{Source}^\sharp = PP + \mathrm{Store}^\sharp$$

$$F \in \mathrm{Dep}^\sharp = \mathcal{P}\left(\mathrm{Source}^\sharp \times \mathrm{Store}^\sharp\right)$$

$$\frac{E^\sharp, H^\sharp, \underline{e} \to^\sharp v^\sharp, d^\sharp}{E^\sharp, H^\sharp, \underline{\mathrm{x}^p \mathrel{\color{green}{=}} e} \to^\sharp E^\sharp\left[\mathrm{x} \mapsto \left(\{p\}, v^\sharp\right)\right], H^\sharp, \left(v_l^\sharp \cup d^\sharp\right) \rightsquigarrow^\sharp \mathrm{x}^p} \ \textsc{Asg}$$

$$\frac{\begin{array}{c} E^\sharp \sqsubseteq E_0^\sharp \qquad H^\sharp \sqsubseteq H_0^\sharp \\ E_0^\sharp, H_0^\sharp, \underline{s} \to^\sharp E_1^\sharp, H_1^\sharp, F^\sharp \\ E_1^\sharp \sqsubseteq E_0^\sharp \qquad H_1^\sharp \sqsubseteq H_0^\sharp \end{array}}{E^\sharp, H^\sharp, \underline{\color{green}{\mathrm{while}}\ e\ \color{green}{\mathrm{do}}\ s} \to^\sharp E_0^\sharp, H_0^\sharp, F^\sharp} \ \textsc{While}$$

Correction de l'analyse

On définit une relation d'abstraction $\prec$ entre

- Traces et PP,
- Heap et $\mathrm{Heap}^\sharp$,
- ...

Théorème (En cours)

Si

$$[], \emptyset, [], \emptyset, \emptyset, \underline{s} \to \tau, M_\tau, F_\tau, E_\tau, H_\tau$$

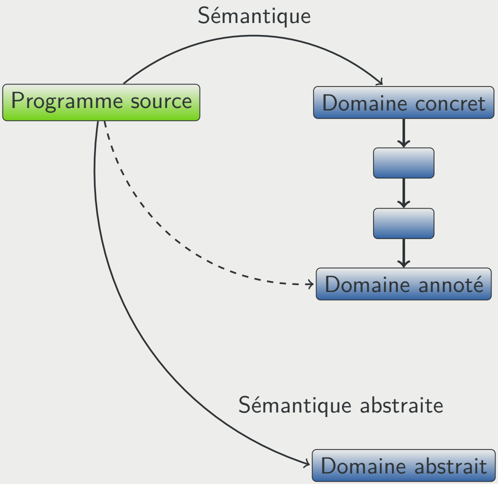
et

$$\bot, \bot, \underline{s} \to^\sharp E^\sharp, H^\sharp, F^\sharp$$

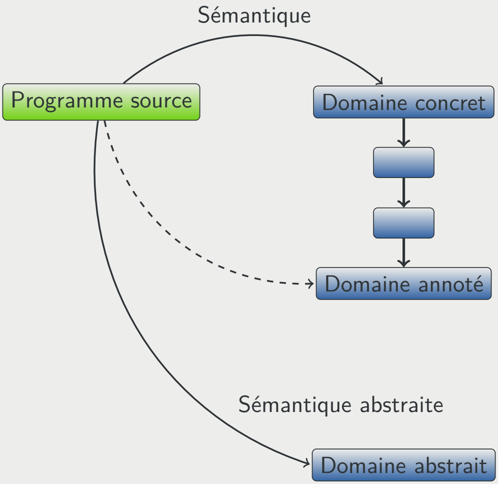
alors $E \prec E^\sharp$, $H \prec H^\sharp$ et $F_\tau \prec F^\sharp$.

Conclusion

- Une architecture générale pour instrumenter une sémantique.
- Passe à l'échelle.
- Formalisé en Coq.



Merci pour votre écoute !



$$\frac{E[\mathtt{x}] = v \qquad M[\mathtt{x}] = \tau_0}{\tau, M, \boldsymbol{d}, E, H, \underline{\mathtt{x}} \to \tau', M, \boldsymbol{d} \cup \{\mathtt{x}^{\tau_0}\}, E, H, v}\ \textsc{Var} \qquad \frac{H[l] = \bot \qquad H' = H[l \mapsto \underline{\{\}}] \qquad M' = M[l \mapsto \tau']}{\tau, M, \boldsymbol{d}, E, H, \underline{\{\}} \to \tau', M, \boldsymbol{d} \cup \left\{l^{\tau'}\right\}, E, H', l}\ \textsc{Obj}$$

$$\frac{H'[l] = o \qquad o[\mathtt{f}] = v \qquad M[(l, M[l], f)] = \tau_0}{\tau, M, \boldsymbol{d}, E, H, \underline{(E', H', l)\ ._1\mathtt{f}} \to \tau', M, \boldsymbol{d} \cup \{(l, M[l], f)^{\tau_0}\}, E', H', v}\ \textsc{Fld1}$$

$$\frac{\tau_1, M, \emptyset, S, \underline{e} \to \tau_2, M', \boldsymbol{d}, r \qquad \tau_3, M', \emptyset, S, \underline{\mathrm{if1}(r, s_1, s_2)} \to \tau_4, M'', \emptyset, r'}{\tau_0, M, \emptyset, S, \underline{\mathrm{if}\ e\ \mathrm{then}\ s_1\ \mathrm{else}\ s_2} \to \tau_5, M'', \emptyset, r'}\ \textsc{If}$$

$$\frac{\tau_1, M, \emptyset, S, e \to \tau_2, M', \boldsymbol{d}, r \qquad \tau_3, M', \emptyset, S, \underline{\mathrm{while1}(r, x, s)} \to \tau_4, M'', \emptyset, r'}{\tau_0, M, \emptyset, S, \underline{\mathrm{while}\ e\ \mathrm{do}\ s} \to \tau_5, M'', \emptyset, r'}\ \textsc{While}$$

$$\frac{\tau_1, M, \emptyset, S, e \to \tau_2, M', \boldsymbol{d}, r \qquad \tau_3, M', \boldsymbol{d}, S, \underline{\mathtt{x}\ =_1\ r} \to \tau_4, M'', \emptyset, r'}{\tau_0, M, \emptyset, S, \underline{\mathtt{x} = e} \to \tau_5, M'', \emptyset, r'}\ \textsc{Asg} \qquad \frac{E' = E[\mathtt{x} \mapsto v] \qquad M' = M[\mathtt{x} \mapsto \tau']}{\tau, M, \boldsymbol{d}, S, \underline{\mathtt{x}\ =_1\ (E, H, v)} \to \tau', M', \emptyset, E', H}\ \textsc{Asg1}$$

$$\frac{\tau_1, M, \emptyset, S, e_1 \to \tau_2, M', \boldsymbol{d}, r \qquad \tau_3, M', \emptyset, S, \underline{r.\mathtt{f}\ =_1\ e_2} \to \tau_4, M'', \emptyset, r'}{\tau_0, M, \emptyset, S, \underline{e_1.\mathtt{f} = e_2} \to \tau_5, M'', \emptyset, r'}\ \textsc{FldAsg}$$

$$\frac{\tau_1, M, \emptyset, S', e \to \tau_2, M', \boldsymbol{d}, r \qquad \tau_3, M', \boldsymbol{d}, S', \underline{l.\mathtt{f}\ =_2\ r} \to \tau_4, M'', \emptyset, r'}{\tau_0, M, \emptyset, S, \underline{(S', x).\mathtt{f}\ =_1\ e} \to \tau_5, M'', \emptyset, r'}\ \textsc{FldAsg1}$$

$$\frac{H[l] = o \qquad o' = o[\mathtt{f} \mapsto v] \qquad H' = H'\left[l \mapsto o'\right] \qquad M' = M[(l, M[l], f) \mapsto \tau']}{\tau, M, \boldsymbol{d}, S, \underline{l.\mathtt{f}\ =_2\ (E, H, v)} \to \tau', M', \emptyset, E, H'}\ \textsc{FldAsg2}$$

$$\frac{\tau_1, M, \emptyset, S, e \to \tau_2, M', \boldsymbol{d}, r \qquad \tau_3, M', \emptyset, S, \underline{\mathrm{delete1}\ r.\mathtt{f}} \to \tau_4, M'', \emptyset, r'}{\tau_0, M, \emptyset, S, \underline{\mathrm{delete}\ e.\mathtt{f}} \to \tau_5, M'', \emptyset, r'}\ \textsc{Delete}$$

1. Annotations
    - Traces
    - Flots directs

2. Sémantique abstraite