Arnaud Carayol
5/3/2015

GdT Sémantique - Vérif

"Simulation des schémas de récursion par des CPDA"

Rappels

Labeled Recursion Scheme

$N = \{A, B, C \dots\}$ non-terminaux typés

$V = \{x, y, \varphi, \psi, \dots\}$ variables typées

règles : $\underbrace{F\, x_1 \dots x_n}_{\text{type } 0} \xrightarrow{a \in \Sigma} \underbrace{t}_{\text{type } 0} \in Term^0(N \cup \{x_1, \dots, x_n\})$

Labeled ~~Recursion~~ Transition System (LTS)

$$F\, t_1 \dots t_n \xrightarrow{a \in \Sigma} t[x_1 \leftarrow t_1, \dots, x_n \leftarrow t_n]$$

dès que le schéma a une règle $F\, x_1 \dots x_n \xrightarrow{a \in \Sigma} t$

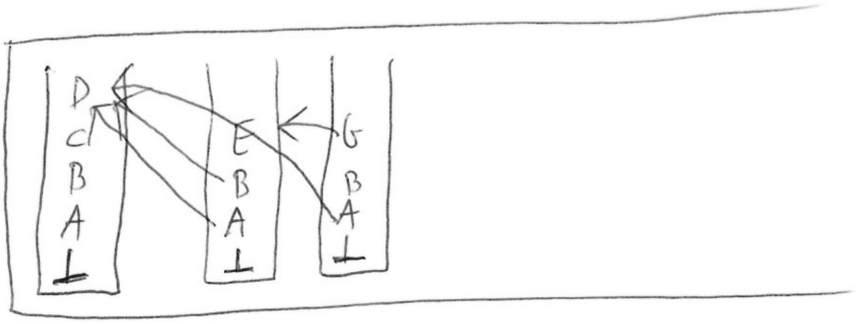
Arbre : il est obtenu par dépliement du LTS à partir de $S : 0 \in N$

Collapsible Pushdown Automata (CPDA)

Pour la simplicité de l'exposé, on fera tout à l'ordre 2 aujourd'hui. L'encodage sera similaire à tout ordre (malgré une vraie difficulté technique à partir de l'ordre 3).

<u>2-CPDA</u> : états finis + une pile dont les éléments sont des piles "usuelles", plus des liens

exemple



<u>opérations</u> :

pop_1 — ~~ajout~~ retrait de symbole au sommet de la pile qui est au sommet de la pile de piles

$push_D$ — ajout du symbole D ——— (sans lien)

$push^l_D$ ——— (avec lien)

$push_2$ — copie de la pile au sommet de la pile de piles

pop_2 — retrait de ———

collapse — effacement de tout ce qui est au-dessus de la cible du lien de l'élément le plus haut

<u>Automate</u> : Q états finis

règles $\underset{\in Q}{p}, \underset{\in \text{ alphabet de pile } \Gamma}{A} \xrightarrow{\overset{\in\Sigma}{a} \cup \{\varepsilon\}} \underset{\in Q}{q}, \underset{\text{opération}}{\ell}$

<u>LTS associé</u> : configurations $Q \times Stacks_2^{\ell}(\Gamma)$

transitions $(p, s) \longrightarrow (q, \ell(s))$

si $top_0(s) = A$

et $p, A \rightarrow q, \ell \in \Delta$

<u>NB</u> : on doit autoriser les ε-transitions.

<u>Th</u> : Les arbres définis par les HORS étiquetés d'ordre n sont isomorphes aux arbres définis par les CPDA d'ordre n.

<u>Commençons par une preuve à l'ordre 1.</u>

$$\left.\begin{array}{l} S \xrightarrow{\varepsilon} A\ V \\ A\ x \xrightarrow{a} A\ (B\ x) \\ A\ x \xrightarrow{c} x \\ B\ y \xrightarrow{b} y \end{array}\right\}\ \text{HORS}$$

LTS associé :

$$\begin{tikzcd} S \arrow[r] & A\ V \arrow[r, "a"] \arrow[d, "c"] & A\ (B\ V) \arrow[r, "a"] \arrow[d, "c"] & A(B\ B\ V \arrow[d, "c"] \\ & V & B\ V \arrow[l, "b"'] & BBV \arrow[l, "b"'] & \arrow[l] \end{tikzcd}$$

1-CPDA correspondant

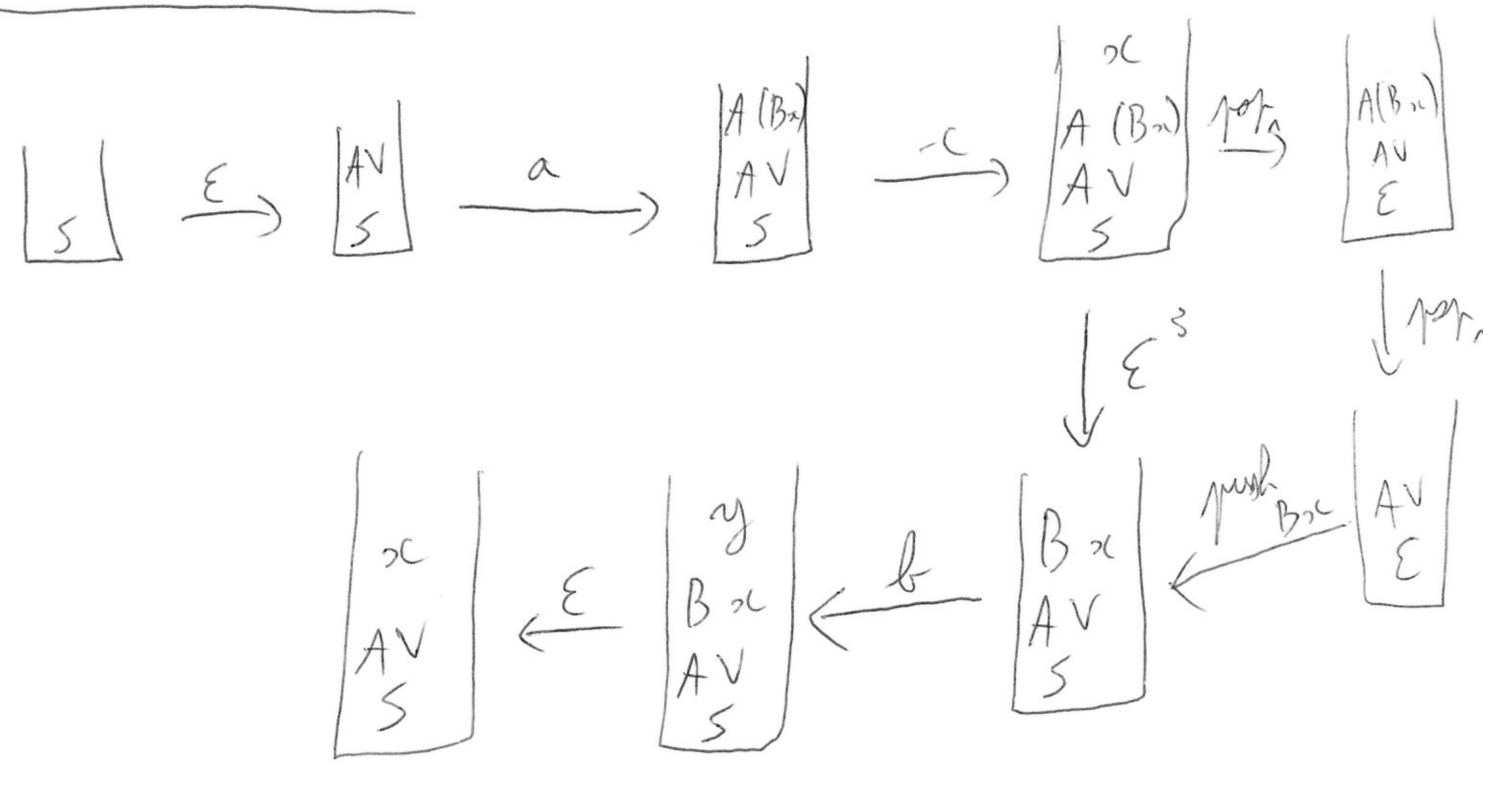


Schéma d'ordre 2

$$Z \xrightarrow{e} G\,(H\ X)$$

$$G\ z \xrightarrow{(} F\ \ G\ \ z\ \ (H\ z)$$

$$G\ z \xrightarrow{*} X$$

$$H\ u \xrightarrow{*} u$$

⚠ terme partiellement appliqué ici

$$F\ \varphi\ x\ y \xrightarrow{(} F\ (F\ \varphi\ x)\ \ y\ \ (H\ y)$$

$$F\ \varphi\ x\ y \xrightarrow{)} \varphi\ (H\ y)$$

$$F\ \varphi\ x\ y \xrightarrow{*} x$$

$$\begin{array}{|c|} \\ \\ Z \\ \hline \end{array} \xrightarrow{e} \begin{array}{|c|} \\ G(H\,X) \\ Z \\ \hline \end{array} \xrightarrow{C} \begin{array}{|c|} F\,G\,z(H_z) \\ G(H\,X) \\ Z \\ \hline \end{array} \xrightarrow{?} \begin{array}{|c|} \varphi(H_y) \\ F\,G\,z(H_z) \\ G(H\,X) \\ Z \\ \hline \end{array}$$

<u>PROBLEME</u> ! si on fait ainsi avec des piles d'ordre comment reprendre le calcul une fois φ évalué ?

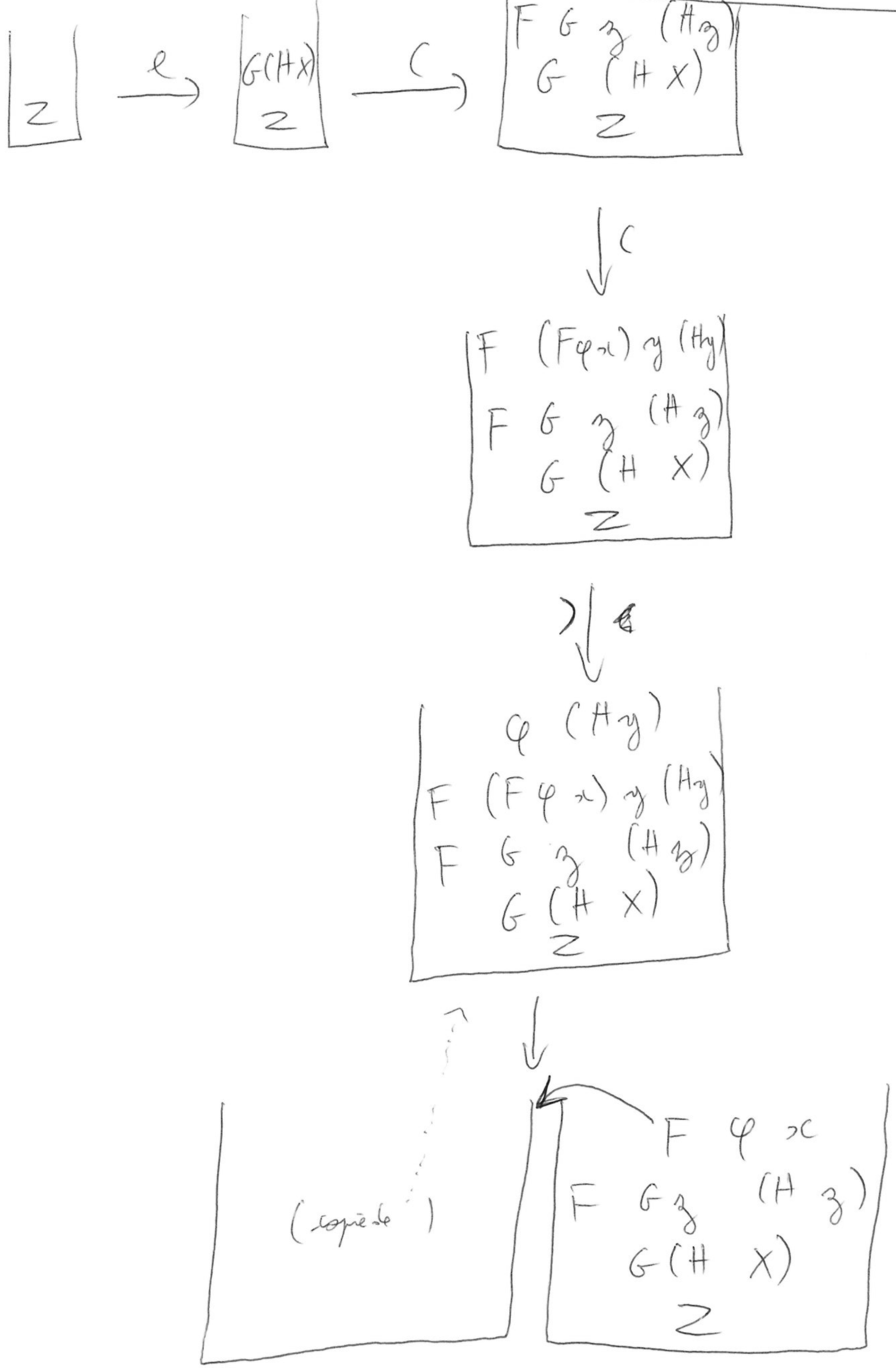
la méthode de recherche est destructrice..

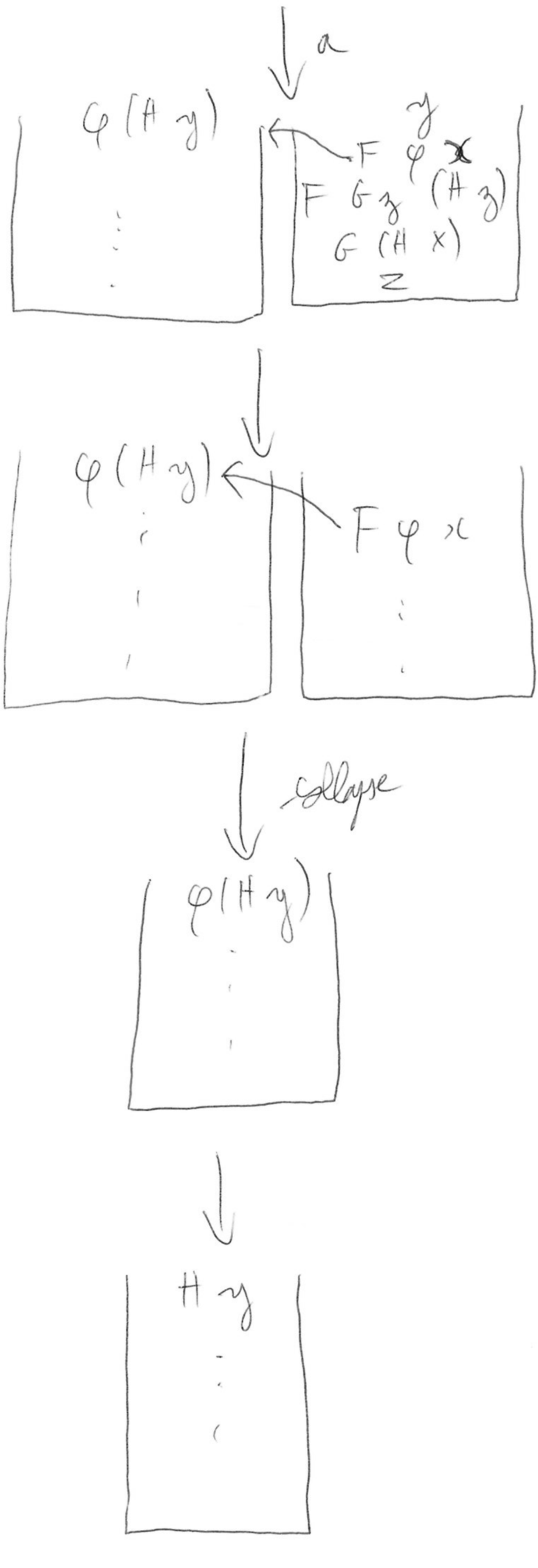
↳ on fait des copies, en travaillant dans une 2-pile

<u>étape suivante</u> : copie + recherche de la valeur de φ dans la nouvelle copie : on fait un "pop" et on le trouve

$$\xrightarrow{*} \begin{array}{|c|} \varphi(H_y) \\ F\,G\,z(H_z) \\ G(H\,X) \\ Z \\ \hline \end{array} \longleftarrow \begin{array}{|c|} \\ G \\ G(H\,X) \\ Z \\ \hline \end{array}$$

qui bloque ensuite en se réécrivant en X (symbole sans règle, qui sert à stopper le calcul).


Z
e
G (H X)
Z
C
F G z (H z)
G (H X)
Z
C
F (F φ x) y (H y)
F G z (H z)
G (H X)
Z
φ (H y)
F (F φ x) y (H y)
F G z (H z)
G (H X)
Z
(copie de)
F φ x
F G z (H z)
G (H X)
Z



a
φ (H y)
y
F φ x
F G z (H z)
G (H x)
z
φ (H y)
F φ x
collapse
φ (H y)
H y