Université Bordeaux 1
Licence, module Compilation, 2017/2018

TD5-Sémantique

Sémantique de IMP

Exercice 5.1

1-Vérification. On en déduit que $[\![c]\!]_p$ est aussi point-fixe de la fonction Φ dont $[\![c]\!]_g$ est le plus petit point fixe. Donc

$$[\![c]\!]_g \subseteq [\![c]\!]_p$$

2- Supposons que $n = m + 1$ et $(c, \rho) \to (C, \rho') \to^m (\varepsilon, \rho'')$ On distingue tous les cas de règles utilisées au premier pas, et on conclut en utilisant l'hypothèse de récurrence.
Donc

$$[\![c]\!]_p \subseteq [\![c]\!]_g.$$

3- Immédiat.

Exercice 5.2

On note c_i ($i \in [1, 10]$ ces commandes, prises dans l'ordre de lecture.
On trouve que :

$$[\![c_1]\!] : \rho \mapsto \rho[x \mapsto 1], \quad [\![c_2]\!] = \emptyset, \quad [\![c_3]\!] : \rho \mapsto \rho[x \mapsto 1], \quad [\![c_4]\!] = \emptyset,$$

$$[\![c_4]\!] = [\![c_5]\!] = [\![c_6]\!] = \emptyset, \quad [\![c_7]\!] : \rho \mapsto \rho[x \mapsto 10]$$

$$[\![c_8]\!] = \emptyset, \quad [\![c_9]\!] = \rho \mapsto \rho[x \mapsto 0], \quad [\![c_{10}]\!] = \rho \mapsto \rho[x \mapsto 1].$$

Exercice 5.3

$$c_1 \equiv c_3 \equiv c_{10}, \quad c_2 \equiv c_4 \equiv c_5 \equiv c_6 \equiv c_8$$

Sémantique de Léa

Exercice 5.4

On note P_i ($i \in [1, 3]$ ces programmes, pris dans l'ordre de lecture.

$$[\![P_1]\!] : u \mapsto 11, \quad [\![P_2]\!] = [\![P_3]\!] = \emptyset.$$

Attributs

Exercice 5.5

1- Soient $f, g \in \mathcal{F}(D, I)$. Alors $f \cap g$ est encore une fonction appartenant à $\mathcal{F}$: $\inf f, g = f \cap g$. Par contre il se peut que f, g ne possèdent pas de borne supérieure. Par exemple :

$$D = \{0, 1\} = I; \quad f = \{(0, 0)\}, \quad g = \{(0, 1)\},$$

Si $f \sqsubseteq h$ et $g \sqsubseteq h$, alors $f(0) = h(0) = g(0)$, ce qui est faux. Donc un tel majorant h n' existe pas. A fortiori, f, g n'ont pas de *plus petit majorant*. La structure $(\mathcal{F}, \sqsubseteq)$ n'est donc *pas* un treillis.
2- Oui, toute suite croissante admet une borne supérieure : soit $(f_n)_{n \in \mathbb{N}}$ une suite croissante. Alors $f := \bigcup \{f_n \mid n \geq 0\}$ est un ensemble, qui est, dans $(\mathcal{P}(D \times I), \subseteq)$, le plus petit majorant

de tous les f_n. Vérifions que c'est (le graphe d') une fonction.
Soient $(x,y),(x,z) \in f$. Alors il existe des entiers n, m tels que

$$(x,y) \in f_n, \;\; (x,z) \in f_m$$

Comme la suite est croissante on a aussi

$$(x,y) \in f_{n+m}, \;\; (x,z) \in f_{n+m}.$$

Comme f_{n+m} est une fonction, $y = z$.
3- Soit $F : \mathcal{F}(D,I) \mapsto \mathcal{F}(D,I)$ application continue. Posons

$$f := \sup(\{F^n(\emptyset) \mid n \geq 0\}).$$

Comme $\emptyset \sqsubseteq F(\emptyset)$ et F est croissante

$$\forall n \in \mathbb{N}, \;\; F^n(\emptyset) \sqsubseteq F^{n+1}(\emptyset)$$

donc, par le résultat de la question 2, $\{F^n(\emptyset) \mid n \geq 0\}$ admet bien une borne supérieure.
Comme F est continue, $F(f) = f$.

Exercice 5.6

Notons $D := \mathrm{Dom}(T)$. Pour chaque triplet p, β, i ($p \in P, \beta \in \mathcal{A}, i \in [0, n_p]$ i.e. que i est une position dans la règle p), on définit une application

$$\Phi_{p,\beta,i} : \prod_{\alpha \in \mathcal{A}} \mathcal{F}(D, V_\alpha) \to \mathcal{F}(D, V_\beta)$$

par, pour tout $\vec{\alpha} = (\hat{\alpha})_{\alpha \in \mathcal{A}}$

$$\Phi_{p,\beta,i}(\vec{\alpha}) := \hat{\beta}$$

avec

$$\mathrm{Dom}(\hat{\beta}) := \{u \odot i \mid u \odot i_1 \in \mathrm{Dom}(\hat{\alpha}_1), \ldots, u \odot i_j \in \mathrm{Dom}(\hat{\alpha}_j), \ldots, u \odot i_t \in \mathrm{Dom}(\hat{\alpha}_t)\}$$

et pour tout $u \odot i \in \mathrm{Dom}(\hat{\beta})$

$$\hat{\beta}(u \odot i) := f_{p,\beta,i}(\hat{\alpha}_1(u \odot i_1), \ldots, \hat{\alpha}_j(u \odot i_j), \ldots, \hat{\alpha}_t(u \odot i_t)). \tag{1}$$

On pose maintenant :

$$\Phi_\beta := \bigcup_{p \in P, i \in [0, n_p]} \Phi_{p,\beta,i}. \tag{2}$$

On définit alors une application

$$\Phi : \prod_{\alpha \in \mathcal{A}} \mathcal{F}(D, V_\alpha) \to \prod_{\alpha \in \mathcal{A}} \mathcal{F}(D, V_\alpha)$$

par $\forall \vec{\alpha} \in \prod_{\alpha \in \mathcal{A}} \mathcal{F}(D, V_\alpha)$,

$$\Phi(\vec{\alpha}) = (\Phi_\alpha(\vec{\alpha}))_{\alpha \in \mathcal{A}}.$$

Fait 0 :Un vecteur $\vec{\alpha}$ est une décoration de T ssi $\vec{\alpha}$ est un point fixe de Φ.
En effet, dans le cas où $\Phi(\vec{\alpha}) = \vec{\alpha}$, la définition (1) de $\Phi(\vec{\alpha})$ entrîne que les équations (6) sont vérifiées. Réciproquement, si les équations (6) sont vérifiées , alors le vecteur défini par les

relations (1) est $\vec{\alpha}$ lui-même.
Fait 1 :$\Phi_{p,\beta,i}(\vec{\alpha})$ est une fonction.
Pour chaque attribut β et chaque noeud $u \odot i$, la définition (1) n'assigne qu'une valeur (au plus) à chaque $\hat{\beta}(u \odot i)$.

Fait 2 :$\Phi_{\beta}(\vec{\alpha})$ est une fonction.
Soit v un noeud de T et soit S le non-terminal $S = T(v)$.
Soit p (resp. q) la règle de dérivation qui fait passer de v à ses fils (resp. du père de v à ses fils)
cas 0 : $\beta \in \mathcal{A}_0(S)$
Alors $\hat{\beta}(v)$ ne fait l'objet que d'une relation d'égalité (1), celle qui correspond au triplet $(p, \beta, 0)$.
cas 1 : $\beta \in \mathcal{A}_1(S)$
- si $v \neq \varepsilon$, v se décompose en $v = u \cdot i$ avec $i \geq 1$, donc $v = u \odot i$.
Alors $\hat{\beta}(v)$ ne fait l'objet que d'une relation d'égalité (1), celle qui correspond au triplet (q, β, i).
- si $v = \varepsilon$,
Alors $\hat{\beta}(v)$ ne fait l'objet d'aucune relation (1), car aucune règle n'aboutit à ce noeud v. Donc, $\varepsilon \notin \mathrm{Dom}(\hat{\beta})$.

Fait 3 :$\Phi_{p,\beta,i}$ est une application croissante.
Immédiat sur la définition.

Fait 4 :Φ est une application croissante.
Supposons $\vec{\alpha} \sqsubseteq \vec{\beta}$. Par le fait 3, pour tout triplet (p, β, i), $\Phi_{p,\beta,i}(\vec{\alpha}) \sqsubseteq \Phi_{p,\beta,i}(\vec{\beta})$. On en déduit que $\bigcup_{p \in P, i \in [0,n_p]} \Phi_{p,\beta,i}(\vec{\alpha}) \sqsubseteq \bigcup_{p \in P, i \in [0,n_p]} \Phi_{p,\beta,i}(\vec{\beta})$.

Fait 5 : Toute application croissante $\prod_{\alpha \in \mathcal{A}} \mathcal{F}(D, V_\alpha) \to \prod_{\alpha \in \mathcal{A}} \mathcal{F}(D, V_\alpha)$ préserve la borne supérieure des suites croissantes.
En effet, comme D est fini, toute suite croissante est ultimement stationnaire. Donc sa borne supérieure est son maximum. Toute application croissante préserve le maximum. L'exercice 5.5 permet de conclure que toute application croissante de $(\prod_{\alpha \in \mathcal{A}} \mathcal{F}(D, V_\alpha), \sqsubseteq)$ dans lui-même admet un plus petit point fixe.
Prouvons qu'il existe une plus petite décoration :
par le Fait 0, une décoration est un point fixe de Φ et par le Fait 4 Φ est croissante, par le Fait 5 Φ préserve la borne supérieure des suites croissantes et par le raisonnement de la question 3 de l'exercice 5, Φ admet un plus petit point fixe.
2- Soit la grammaire :

$$\sigma \to TU, \;\; U \to a, \;\; U \to b.$$

les attributs :

$$\mathcal{A} = \{\alpha\}, \;\; \mathcal{A}_0(\sigma) = \mathcal{A}_1(T) = \mathcal{A}_1(U).$$

et les règles sémantiques :

$$T.\alpha := U.\alpha, \;\; \sigma.\alpha := T.\alpha, \;\; U.\alpha := \sigma.\alpha.$$

Soit $\mathcal{T} : \{\varepsilon, 1, 11, 2, 21\} \to \{\sigma, T, U, a, b\}$ l'arbre planaire défini par

$$\varepsilon \mapsto \sigma, \;\; 1 \mapsto T, \;\; 11 \mapsto a, \;\; 2 \mapsto U, \;\; 21 \mapsto b.$$

Donnons deux solutions maximales $\hat{\alpha}_1, \hat{\alpha}_2$:

$$\forall u \in \mathrm{Dom}(\mathcal{T}), \hat{\alpha}_1(u) := 1, \;\; \forall u \in \mathrm{Dom}(\mathcal{T}), \hat{\alpha}_1(u) := 2.$$

Exercice 5.7

Soit $G = \langle X, N, P, \sigma \rangle$ une grammaire en forme normale quadratique de Chomsky, On pose $\mathcal{A} = \{m, M\}$ (m évoque minimum M évoque Maximum).
- $m \in \mathcal{A}_0(\sigma), M \in \mathcal{A}_0(\sigma)$,
- si $S \neq \sigma$ est binaire, alors $m \in \mathcal{A}_1(S), M \in \mathcal{A}_0(S)$,
- si S est unaire, alors $m \in \mathcal{A}_1(S), M \in \mathcal{A}_1(S)$. Pour toutes les règle syntaxiques $\sigma \to TU$, $S \to TU$, $S \to s$) on pose les règles sémantiques :

$$\sigma \to TU; \;\; T.m := \sigma.m + 1, \; U.m := T.M + 1, \; \sigma.M := U.M$$
$$S \to TU; \;\; T.m := S.m + 1, \; U.m := T.M + 1, \; S.M := U.M$$
$$S \to s; \;\; S.M := S.m$$

On vérifie que la valeur de l'attribut $\hat{M}(u)$ dans la pp-décoration est bien le numéro du noeud u dans l'ordre préfixe.

Exercice 5.8

1- La règle $B.true = newlabelQ$ n'est pas conforme à la définition [Knuth 68] : si newlabelQ est une fonction à 0 arguments, c' est une fonction constante, et elle ne vérifie pas l' hypothèse que " newlabelQ creates a new label each time it is called".
2- En revanche, en suivant l'idée de l' exercice précédent, on peut remplacer newlabelQ par un attribut `num`, dont la valeur dans la pp-décoration sera le numéro du noeud dans l'ordre préfixe.

Exercice 5.9

1- Cette traduction est correcte si le langage de départ est `IMP`
2- Reprenons la fonction `foo` de l'exercice plus haut et considérons l'instruction I :

```
if (false and foo(0)) then write(1) else write(2)
```

La sémantique de I est $[\![\mathrm{I}]\!] = \emptyset$
Tandis que le code à trois adresses calcule : $u \mapsto 2$ (pour toute suite $u \in (\mathbb{Z}/N\mathbb{Z})^*$).

Bisimulations

Exercice 5.10

Notons A_1, A_2, A_3 les automates pris dans l'ordre de lecture.
Disons que (A_i, q) est bisimilaire à (A_j, r) si il existe une bisimulation de A_i vers A_j qui possède (q, r). Comparons A_1 et A_2 :

$$R := \{(0,0)\} \cup \{(i,j) \mid i \in [1,3], j \in [1,3]\}$$

est une bisimulation de A_1 vers A_2.
Supposons qu'il existe une bisimulation R' de A_1 vers A_3 telle que $(1,1) \in R'$. Alors on aurait $(1,1) \in R'$ ou $(1,3) \in R'$. Or $L(A_1, 1) \neq L(A_3, 1)$ et $L(A_1, 1) \neq L(A_3, 3)$. Donc, par l'exercice

suivant, Q1, il n'est pas possible que $(A_1, 1), (A_3, 1)$ soient bisimilaires, ni que $(A_1, 1), (A_3, 3)$ soient bisimilaires.
On vérifie que, si R est une bisimulation de A vers B et R' une bisimulation de B vers C alors $R \circ R' := \{(p, r) \mid \exists q, (p, q) \in R, (q, r) \in R'\}$ est une bisimulation de A vers C.
Donc $(A_2, 0)$,$(A_3, 0)$ ne sont pas bisimilaires.

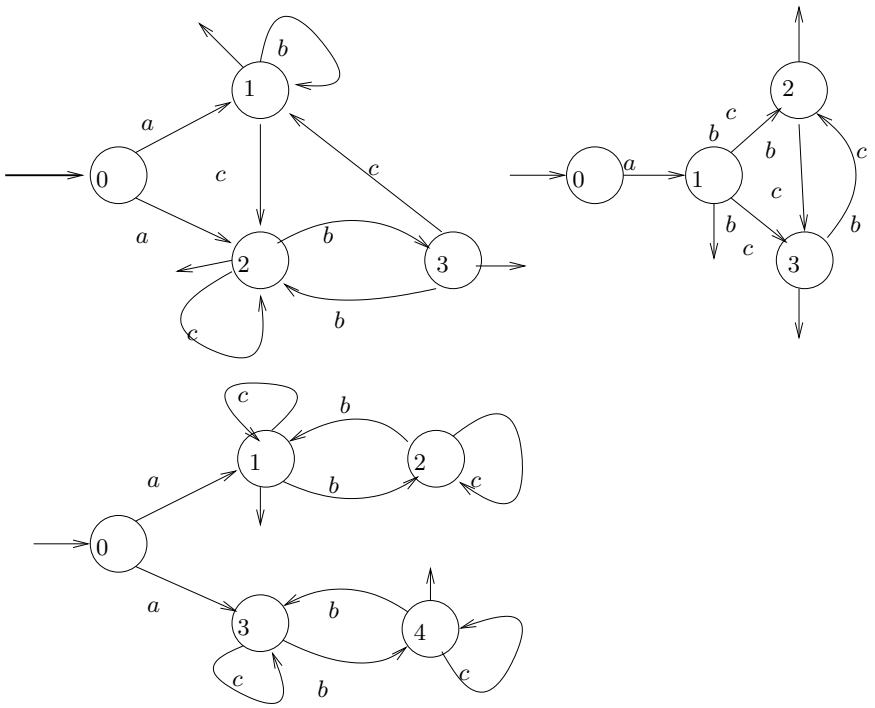

FIGURE 1 – exercice : trois automates

Exercice 5.11
1- Soient $\mathcal{A}_i = \langle A, Q_i, q_{i,-}, Q_{i,+}, \delta_i \rangle$ des automates finis (pour $i \in \{1, 2\}$). Soit R une bisimulation de A_1 vers A_2. On montre par récurrence sur $|u|$ que :

$$\forall p_1, p_1' \in Q_1, \forall p_2 \in Q_2, (p_1, p_2) \in R \text{ et } p_1 \to^u_{A_1} p_1' \Rightarrow \exists p_2' \in Q_2, (p_1', p_2') \in R \text{ et } p_2 \to^u_{A_2} p_2'$$

Par la clause (B1),

$$p_1' \in Q_{1,+}, (p_1', p_2') \in R \Rightarrow p_2' \in Q_{2,+}.$$

Donc $L(A_1) \subseteq L(A_2)$.
Par des arguments analogues (en échangeant les indices $1, 2$), on montre que $L(A_2) \subseteq L(A_1)$.
Finalement

$$L(A_1) = L(A_2).$$

2- Reprenons les 3 automates de la Figure 1. $L(A_1, 0) = L(A_3, 0)$ mais il n'existe pas de bisimulation de A_1 vers A_3 qui possède $(0, 0)$
3- Soit $\mathcal{A} = \langle A, Q, q, Q_+, \delta \rangle$ un automate *déterministe*. La relation binaire sur Q :

$$p \sim q \Leftrightarrow L(A, p) = L(A, q)$$

est une bisimulation. Donc, lorsque A est déterministe, la réciproque de l'énoncé de la question 1 est vraie.

Exercice 5.12

1- On vérifie que, si A, B sont deux automates et $\mathcal{B}$ est un ensemble de bisimulation de A vers B, alors $R := \bigcup \mathcal{B}$ est une bisimulation de A vers B. On vérifie aisément qu'elle est plus grande que tousles éléments de $\mathcal{B}$. Donc R est la *plus grande* bisimulation de A vers B.
Dans le cas où A=B, comme les bisimulaton sont closes par composition,on peut en conclure que la clôture réf ;exise et transitive d' une bisimulation de A vers A est aussi une bisimulation de A vers A. La plus grande bisimulation de A est donc une relation d' équivalence.
2- On considère l'ensemble ordonné $(E, \subseteq)$ des relations binaires qui respectent la relation d'équivalence $\sim_0$ définie par

$$p \sim_0 q \Leftrightarrow (p \in Q_+ \Leftrightarrow q \in Q_+).$$

Soit $\Phi : E \to E$ définie par :

$$\Phi(R) := \{(p,q) \mid \forall a \in A, [\forall p' \in \delta(p,a), \exists q' \in \delta(q,a), (p',q') \in R] \text{ et } [\forall q' \in \delta(q,a), \exists p' \in \delta(p,a), (p',q') \in R]\}$$

On vérifie que, pour tout $R \in E$, R est une bisimulation ssi

$$R \subseteq \Phi(R).$$

Cette application Φ est croissante. Lorsque Q est fini, E est fini, donc toute application croissante $E \to E$ est continue. Donc Φ admet un plus grand point fixe

$$\sim := \bigcap_{n \geq 0} \Phi(\sim_0)$$

qui est la plus grande bisimulation.
Algorithme : calculer $\bigcap_{n\geq 0} \Phi^n(\sim_0)$

$R :=\sim_0$
$S := \Phi(R);$
while $S \subset R$ *do*
begin $R := S;\ S := \Phi(R)$ end
return(R)

NB : l'algorithme de Moore (qui calcule l'équivalence de Nerode d'un automate fini déterministe), est l'instanciation de cet algorithme dans le cas des automates finis déterministes.
3- Idem question 1.
4- Idem question 2 pour une structure de Kripke, finie, non-déterministe.
5- Soit

$$R := \{(0,0)\} \cup \{(i,j) \mid i \in [1,3], j \in [1,3]\}$$

C'est une relation d' équivalence qui a deux classes. Notons B_1 la plus grande bisimulation de A_1. On doit avoir $R \subseteq B_1$. Donc B_1 est soit l'équivalence à une seule classe, soit R. Mais on a vu que 0 et 1 ne sont pas bisimilaires. Donc $B_1 = R$.

Exercice 5.13
1- $\mathcal{K}'$ n'est pas déterministe : voir la règle d'affectation.
2- Soit $\rho \in \mathbb{Z}^V$. Posons

$$R := \{(\rho, \rho') \in \mathbb{Z}^V \times \mathcal{E}^V \mid \forall x \in V, \rho(x) = [\![\rho'(x)]\!]$$

R est une bisimulation de $\mathcal{K}$ vers $\mathcal{K}'$ telle que $(\rho, \rho \circ \mu) \in R$.
3- Soit $\mathcal{K}''$ obtenue à partir de $\mathcal{K}'$ en faisant un choix déterministe de transition à chaque action d'affectation. Posons

$$R'' := \{(\rho, \rho'') \in \mathbb{Z}^V \times Q'' \mid \forall x \in V, \rho(x) = [\![\rho''(x)]\!]\}$$

R'' est une bisimulation de $\mathcal{K}$ vers $\mathcal{K}''$ telle que $(\rho, \rho \circ \mu) \in R$.
4- On en conclut que, pour toute commande c et tout symbole de prédicat $P \in \mathcal{P}$ (i.e. expression e),

$$(\rho, c, \tau) \in \delta_{\mathcal{K}}, (\rho \circ \mu, c, \tau'') \in \delta_{\mathcal{K}''} \Rightarrow e^{\mathcal{K}}(\tau) = e^{\mathcal{K}''}(\tau'').$$

En ce sens, les calculs de ces deux structures sont indiscernables au moyen des prédicats e. Comme ces prédicats expriment toutes les "valeurs" de toutes les variables, en ce sens les deux structures calculent la "même" chose.
5- Un interpréteur qui simule une structure $\mathcal{K}''$ présenter un intérêt, dans le cas où, à cause des branches non-parcourues des If-Then-Else, le calcul dans $\mathcal{K}''$ évite d'évaluer certaines expressions, dont la représentation dans $\mathbb{Z}$ (en base 2, par exemple) est beaucoup plus longue que l'expression elle-même.

ANNEXE

Grammaire d'attributs.
Une grammaire d'attributs [Knuth, 68] est un tuple

$$\mathcal{G} := \langle X, N, P, \mathcal{A}, (V_\alpha)_{\alpha\in\mathcal{A}}, R\rangle$$

où
- $\langle X, N, P\rangle$ est une grammaire algébrique,
- $\mathcal{A}$ est un ensemble, appelé ensemble des *attributs* ;
- pour chaque $\alpha \in \mathcal{A}$, V_α est un ensemble, appelé l'ensemble des *valeurs* de l'attribut α
- l'ensemble R est constitué des *règles sémantiques*, décrites ci-dessous.

À chaque non-terminal $S \in N$ sont associés deux ensembles disjoints d'attributs $\mathcal{A}_0(S)$, $\mathcal{A}_1(S) \subseteq \mathcal{A}$; les membres de $\mathcal{A}_0(S)$ (resp. $\mathcal{A}_1(S)$) sont appelés attributs *synthétisés* (resp. *hérités*) du non-terminal S.
À chaque règle (syntaxique) $p \in P$

$$S_{p,0} \to S_{p,1}S_{p,2}\dots S_{p,n_p} \tag{3}$$

est associé un ensemble de règles (sémantiques) $R_{p,\alpha,i}$, pour $i \in [0, n_p], \alpha \in \mathcal{A}(S_{p,i})$; chacune de ces règles est de la forme

$$(R_{p,\alpha,i}) : \quad (\alpha, i) = f_{p,\alpha,i}((\alpha_1, i_1), \dots, (\alpha_t, i_t)) \tag{4}$$

avec $t \in \mathsf{N}, i_j \in [0, n_p], \alpha_j \in \mathcal{A}(S_{p,i_j})$, et de fonctions

$$f_{p,\alpha,i} : V_{\alpha_1} \times V_{\alpha_2} \times \ldots \times V_{\alpha_t} \to V_\alpha \tag{5}$$

Les indices doivent respecter la convention que :
- si $i = 0$, alors $\alpha \in \mathcal{A}_0(S_{p,0})$ et $\forall j \in [1, t], i_j \in [1, n_p]$
- si $i \in [1, n_p]$, alors $\alpha \in \mathcal{A}_1(S_{p,i})$.

R est l'ensemble de ces règles $(R_{p,\alpha,i})$.
L'idée intuitive est que la valeur de l'attribut α de la variable $S_{p,i}$:
- sur un noeud père de la règle p (i.e. $i = 0$) et si α est synthétisé, est l'image par $f_{p,\alpha,0}$ des attributs de ses fils
- sur un noeud fils de la règle p (i.e. $i \geq 1$), et si α est hérité, est l'image par $f_{p,\alpha,i}$ des attributs de son père, de ses frères et de lui-même.

NB1 : Pour un attribut α, le fait d'être "synthétisé" ou "hérité" dépend du non-terminal $S \in V$ considéré, mais pas de la règle p où apparait S.
NB2 : Il est possible que $\mathcal{A}_0(S_1) \cap \mathcal{A}_1(S_2) \neq \emptyset$ i.e. que α soit synthétisé pour un non-terminal S_1 mais hérité pour un autre non-terminal S_2 ; on peut cependant normaliser l'ensemble des attributs de façon que tout attribut α soit :
- membre d'au moins un ensemble $\mathcal{A}(S)$
- synthétisé (pour *tous* les non-terminaux S tels que $\alpha \in \mathcal{A}(S)$)
- ou bien hérité (pour *tous* les non-terminaux S tels que $\alpha \in \mathcal{A}(S)$).

Une *décoration* d'un arbre de dérivation T par la grammaire $\mathcal{G}$ est une famille de fonctions

$$\hat{\alpha} : \mathrm{Dom}(T) \to V_\alpha$$

vérifiant, pour tout noeud $u \in \mathrm{Dom}(T)$, tel que u possède n_p fils et

$$p = (T(u), T(u1)T(u2)\dots T(un_p))$$

on a

$$\hat{\alpha}(ui) = f_{p,\alpha,i}(\hat{\alpha_1}(ui_1), \dots, \hat{\alpha_t}(ui_t)) \tag{6}$$

où $u0$ dénote le mot u et ui dénote le mot u suivi de la lettre i (si $i \neq 0$).
Structure de Kripke.
Une structure de Kripke, sur l'alphabet d'actions A et la signature propositionnelle $\mathcal{P}$ est un tuple

$$\mathcal{K} := \langle A, Q, \delta, (P^{\mathcal{K}})_{P \in \mathcal{P}} \rangle$$

tel que

- A est un ensemble, l'ensemble des actions
- Q est un ensemble, l'ensemble des états
- $\delta \subset Q \times A \times Q$ est l'ensemble des transitions
- pour tout $P \in \mathcal{P}$, $P^{\mathcal{K}} : Q \to \{0, 1\}$ est une application à valeurs booléennes.

Un automate $\mathcal{A}$ sur l'alphabet d'entrée A et l'ensemble d' états Q peut être vu comme une structure de Kripke en choisissant une structure propositionnelle à un seul prédicat F, interprété par

$$F^{\mathcal{A}}(q) := 1 \text{ ssi } q \text{ est final .}$$

Bisimulation.
Soit A un alphabet, $\mathcal{P}$ une signature propositionnelle et, pour chaque $i \in \{0, 1\}$, $\mathcal{K}_i := \langle A, Q, \delta, (P)^{\mathcal{K}_i}_{P \in \{\mathcal{P}\}} \rangle$ une structure de Kripke sur $A, \mathcal{P}$. Une relation binaire $R \subseteq Q_0 \times Q_1$ est une *bisimulation* de $\mathcal{K}_1$ vers $\mathcal{K}_2$ ssi : (B1)$\forall (p, q) \in R, \forall P \in \mathcal{P}, P^{\mathcal{K}_0}(p) = P^{\mathcal{K}_1}(q)$
(B2)$\forall (p, q) \in R, \forall (p, a, p') \in \delta_0, \exists q' \in Q_1, (q, a, q') \in \delta_1$ et $(q, q') \in R$
(B3)$\forall (p, q) \in R, \forall (q, a, q') \in \delta_1, \exists p' \in Q_0, (p, a, p') \in \delta_0$ et $(p, p') \in R$