Alphabet : Σ = ensemble de lettres ou symboles.

mot sur Σ = suite finie de lettres

Σ^* = tous les mots sur Σ

ϵ = mot vide ($\epsilon \in \Sigma^*$)

Langage = partie de Σ^* (contient ou non ϵ)

Concaténation : $(\Sigma^*, .)$ est un monoïde libre (loi interne associative avec élément neutre)

Homomorphisme : $h : \Sigma^* \to \Gamma^*$ tel que : $\begin{cases} h(\epsilon) = \epsilon \\ h(uv) = h(u)h(v) \end{cases}$

Synthaxe = écriture des mots

Sémantique = signification des mots

Arbre libre : Graphe non orienté connexe et sans circuit
Feuille = un seul voisin
Etiquette = A un ensemble et (V, Γ) un arbre libre, à chaque élément de V on associe un élément de A.

Arbre enraciné : Arbre libre tel que $\exists\, r \in V \mid \forall x \in V\ \exists\, !c$ chemin entre x et r
Arité d'un sommet = nombre d'enfants
Hauteur de l'arbre = longueur du plus grand chemin entre la racine et une feuille.

Arbre ordonné : Arbre dans lequel l'ensemble des enfants de chaque sommet est totalement ordonné.

Définition inductive : Soit E un ensemble, on définit X le plus petit sous ensemble de E tel que :

$$\begin{cases} (B) : B \subset X \\ (I) : (r_i :: E^{n_i} \to E) \end{cases}$$

B est la base et I l'ensemble des règles d'induction.

Notations (exemples)

$$\mathbb{N} = \begin{cases} (B) : 0 \in \mathbb{N} \\ (I) : \forall n \in \mathbb{N}, n+1 \in \mathbb{N} \end{cases} = \left\{ \frac{}{0} \ \frac{n}{n+1} \right\}$$

$$\Sigma^* = \begin{cases} (B) : \epsilon \in \Sigma^* \\ (I) : \forall w \in \Sigma^*, a \in \Sigma, wa \in \Sigma^* \end{cases} = \left\{ \frac{}{\epsilon} \ \frac{w}{wa} \right\}$$

$$\{a^n b c^n\} = \left\{ \frac{}{b} \ \frac{x}{axc} \right\}$$

Preuve par induction $\mathcal{P}$ est vraie sur X si et seulement si :

$$\begin{cases} \mathcal{P} \text{ vraie sur } B \\ \mathcal{P} \text{ stable par } (I) \end{cases}$$

Fonction définie inductivement :

On définit f sur B

$$\forall x = r_i(x_1, \ldots, x_{n_i}) \in X, f(x) = r_i(f(x_1), \ldots, f(x_{n_i}))$$

Arbre ordonné

$$\begin{cases} \varnothing \in V \\ \forall (x_1, \ldots, x_n) \in X \text{ liste ordonnee}, \forall v \in V, (v, x_1, \ldots, x_n) \in X \end{cases}$$

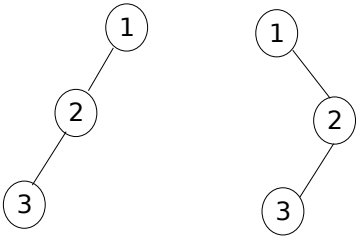
Arbre binaire

$$\begin{cases} \varnothing \in A \\ \forall A_g, A_d \in A, r \in X, (A_g, r, A_d) \in A \end{cases}$$

Arbre binaire étiqueté par un ensemble A

$$\begin{cases} \varnothing \in AB \\ \forall g, d \in AB, \forall a \in A, (g, a, d) \in AB \end{cases}$$

ATTENTION Les deux arbres ci-dessous sont les mêmes arbres ordonnés, mais pas les mêmes arbres binaires :



Arbre de gauche : $(((\varnothing, 3, \varnothing), 2, \varnothing), 1, \varnothing)$

Arbre de droite : $(\varnothing, 1, ((\varnothing, 3, \varnothing), 2, \varnothing)$

Logique N^o 5 Arbres binaires (déf explicite)

Définition par récurrence :

$$\left.\begin{array}{l} AB_0 = \{\varnothing\} \\ AB_{n+1} = AB_n \bigcup \{(g,a,d), a \in A,\ g,d \in AB_n\} \end{array}\right\} AB_{rec} = \bigcup_{\mathbb{N}} AB_n$$

Rappel :

$$\begin{cases} (B) : \varnothing \in AB \\ (I) \forall g,d \in AB, \forall a \in A, (g,a,d) \in AB \end{cases}$$

Propriété : $AB_{rec} = AB$

Preuve :

1. Montrons que $AB_{rec} \subset AB$, (montrons que $AB_n \subset AB \forall n$)
   $AB_0 = \{\varnothing\} \subset AB$
   Supposons $AB_n \subset AB$ et montrons que $AB_{n+1} \subset AB$
   Soit $x \in AB_{n+1} = AB_n \bigcup \{(g,a,d), a \in A, g,d \in AB_n\}$
   Si $x \in AB_n$ alors $x \in AB$ par hypothèse de récurrence
   Sinon $x = (g,a,d)$, $g,d \in AB_n \subset AB$ et $a \in A$ donc $x \in AB$ d'après (I)

2. Montrons que $AB \subset AB_{rec}$ (il suffit de montrer que AB_{rec} respecte (B) et (I))
   $\varnothing \in AB_0 \subset AB_{rec}$ donc AB_{rec} vérifie (B).
   Soient $g, g \in AB_{req}$, $\exists\, p \in \mathbb{N} \mid\ g,d \in AB_p$
   Donc $\forall a \in A, (g,a,d) \in AB_{p+1} \subset AB_{rec}$
   Donc AB_{rec} respecte (I)

Définition

$$\begin{cases} (B) : (\varnothing, a\varnothing) \in ABS \forall a \in A \\ (I) : \forall g, d \in ABS, \forall a \in A, (g, a, d) \in ABS \end{cases}$$

Propriété : $n = 2f - 1$ (n = nombre de sommets, f = nombre de feuilles)

Preuve ;

1. Montrons que $\mathcal{P}$ est vraie sur B
   Pour $(\varnothing, a, \varnothing)$, on a $n = 1$ et $f = 1$ donc $\mathcal{P}$ vraie ($2 \times 1 - 1 = 1$)

2. Supposons $\mathcal{P}$ vraie pour g et d dans ABS et montrons que $\mathcal{P}$ est vraie pour (g, a, d).
   $f = f_g + f_d$ et $n = n_g + n_d + 1$
   Donc $n = 2f_g - 1 + 2f_d - 1 + 1 = 2f - 1$

Soit $F = \{f_0, \ldots, f_n, \ldots\}$ un ensemble de **symboles de fonctions** (prenant un certain nombre d'**arguments**, pas le même nombre pour chaque fonction).

Soit $\varphi : F \to \mathbb{N}$ qui à une fonction associe le **nombre d'arguments** qu'elle prend (aussi appelé **arité** de la fonction)

$F_n = \{f \in F \mid \varphi(f) = n\}$ ($F_0 =$ les constantes (on peut donc les utiliser comme arguments des autres fonctions).

Termes sur F : On définit T inductivement :

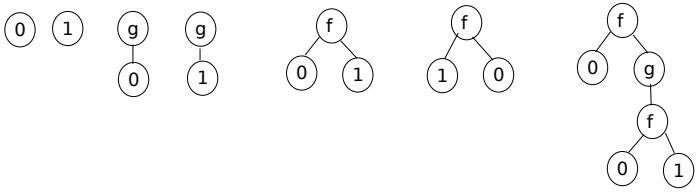
$$\begin{cases} (B) : F_0 \subset T \\ (I) : \forall t_1, \ldots, t_n \in T, f \in F_n, f(t_1, \ldots, t_n) \in T \end{cases}$$

Lien avec les arbres ordonnés étiquetés par F

$F = \{0, 1, f, g\}$ $F_0 = \{0, 1\}]$ $F_1 = \{g\}$ $F_2 = \{f\}$

Donnons une liste (non exhaustive !) d'éléments de T :

0 ; 1 ; $g(0)$; $g(1)$; $f(0,1)$; $f(1,0)$; $f(0, g(f(0,1)))\ldots$



Soit X définit **inductivement** par une base (B) et un ensemble de règles (I). On peut aussi définir X par **récurrence** :

$$\left.\begin{array}{l} X_0 = B \\ X_{n+1} = X_n \bigcup \{r_i(x_1, \ldots, x_{n_i}), x_1, \ldots, x_{n_i} \in X_n\} \end{array}\right\} X = \bigcup_{\mathbb{N}} X_n$$

(pour la preuve, voir l'exemple des arbres binaires définis par récurrence)

Lien avec les termes

$F = B \bigcup I$ $(B = F_0)$

Ensemble des dérivations de X $= D =$ termes sur F

On pose h :

$$\begin{array}{lll} D & \longrightarrow & X \\ b \in B & \longrightarrow & b \\ r_i(t_1, \ldots, t_{n_i}) & \longrightarrow & r_i(h(t_1), \ldots, h(t_{n_i})) \end{array}$$

La propriété $X = \bigcup X_n$ se réécrit alors

$$X = h(D) = \{h(d), d \in D\}$$

Si h est **injective**, on parle de définition **non ambigüe** (le chemin pour atteindre un élément est unique.