LOGIQUE

L3 Informatique

Salim Lardjane

Université Bretagne Sud

4. Les termes du premier ordre

Motivation

On a construit précédemment des *démonstrations* du fait que $\vdash_{\mathcal{S}_1} A \Rightarrow A$.

Supposons toutefois qu'on vous donne le premier pas d'une prétendue démonstration :

$$1)\ \ (A \Rightarrow ((A \Rightarrow A) \Rightarrow A)) \Rightarrow ((A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow A))$$

sans vous en donner la justification et que vous vous posez la question (légitime) :

comment puis-je être sûr(e) que cette fbf est un premier pas possible dans une démonstration ?

Motivation

Si l'on représente les formules (chaînes structurées) comme des **arbres**, répondre à la question revient à trouver des remplacements à faire dans les schémas d'axiomes de façon à obtenir la fbf recherchée (dans une démonstration c'est la seule possibilité pour le premier pas).

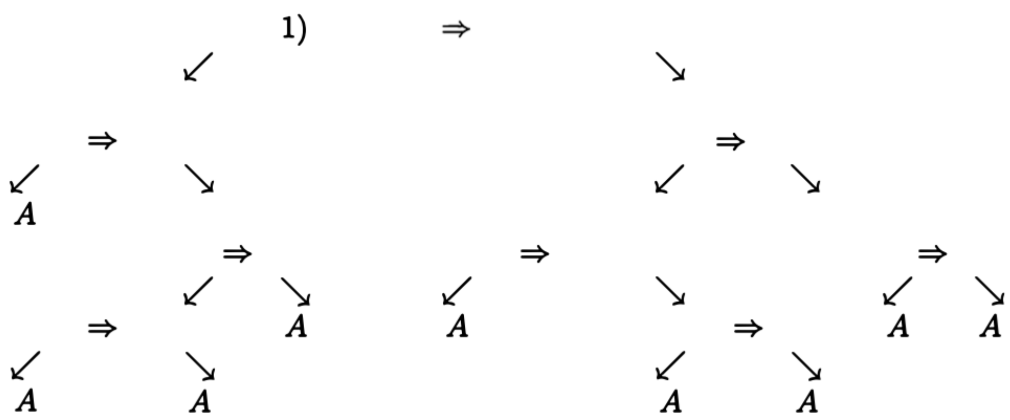
On essaie donc les schémas d'axiomes un à un.

Motivation

L'arbre correspondant à la formule :

$$1)\ \ (A \Rightarrow ((A \Rightarrow A) \Rightarrow A)) \Rightarrow ((A \Rightarrow (A \Rightarrow A)) \Rightarrow (A \Rightarrow A))$$

est le suivant :



Motivation

On utilisera X, Y, Z pour les méta-variables apparaissant dans les schémas d'axiomes pour mieux rappeler que ce sont elles que l'on doit remplacer.

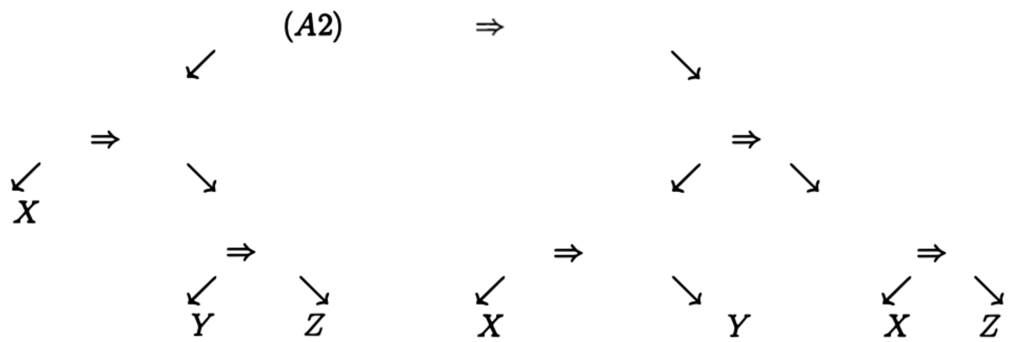
Avec cette notation, l'axiome A_1 du système formel $\mathcal{S}_1$ correspond à l'arbre suivant :

$$(A1) \quad \begin{tikzcd} & \Rightarrow \arrow[dl] \arrow[dr] & & \\ X & & \Rightarrow \arrow[dl] \arrow[dr] & \\ & Y & & X \end{tikzcd}$$

On voit que l'on ne peut pas remplacer de façon cohérente les variables dans A_1 pour pouvoir identifier A_1 avec 1) : il faudrait faire $X \leftarrow A \Rightarrow A$ (feuille la plus à droite) et $X \leftarrow A \Rightarrow ((A \Rightarrow A) \Rightarrow A)$ (feuille la plus à gauche).

Motivation

Essayons maintenant avec A_2 :



Nous découvrons que l'on peut identifier les arbres 1) et A_2 en faisant : $X \leftarrow A$; $Y \leftarrow A \Rightarrow A$; $Z \leftarrow A$.

La conclusion est donc que 1) peut être le premier pas d'une démonstration dans $\mathcal{S}_1$.

Motivation

Dans l'exemple précédent, nous avons considéré que seulement un des arbres contenait des variables.

Essayons de voir comment faire si les deux termes à rendre identiques contiennent des variables.

Dans ce qui suit $x, y, z, \ldots$ dénotent des variables (des objets qui peuvent être remplacé par d'autres objets) et $a, b, c, \ldots, f, g, \ldots$ dénotent des constantes (des objets qui ne peuvent pas être remplacés par d'autres objets).

Comme nous cherchons à concevoir un algorithme général, si l'on identifie $f, g, ...$ à des symboles de fonction, on ne leur attribue *aucune propriété particulière* (associativité, commutativité, etc.).

Motivation

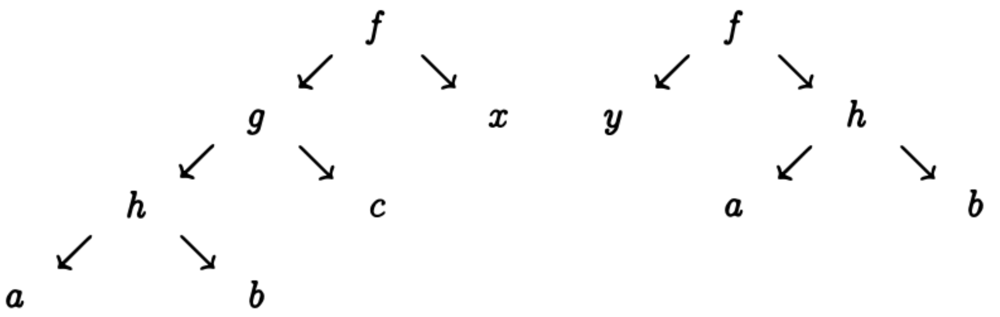
$$\begin{tikzcd} & f \arrow[dl] \arrow[dr] & \\ a & & x \end{tikzcd} \qquad \begin{tikzcd} & f \arrow[dl] \arrow[dr] & \\ y & & b \end{tikzcd}$$

On peut rendre identiques ces deux arbres en faisant : $x \leftarrow b$, $y \leftarrow a$.

$$\begin{tikzcd} & f \arrow[dl] \arrow[dr] & & \\ y & & g \arrow[dl] \arrow[dr] & \\ & a & & x \end{tikzcd} \qquad \begin{tikzcd} & f \arrow[dl] \arrow[dr] & & \\ x & & g \arrow[dl] \arrow[dr] & \\ & y & & b \end{tikzcd}$$

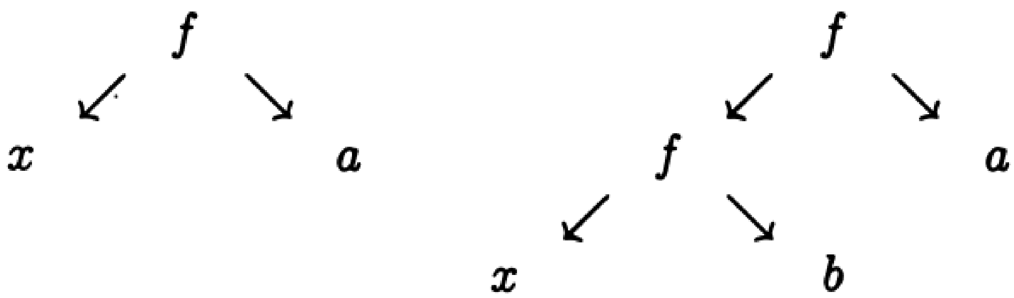
Ce deux arbres ne peuvent pas être rendus identiques : ça exigerait de faire $y \leftarrow a$ et $y \leftarrow b$.

Filtrage et unification



On peut rendre identiques ces deux arbres en faisant : $y \leftarrow g(h(a,b),c)$ et $x \leftarrow h(a,b)$.

Essayer d'unifier les deux arbres suivants :



pose un problème. Lequel ?

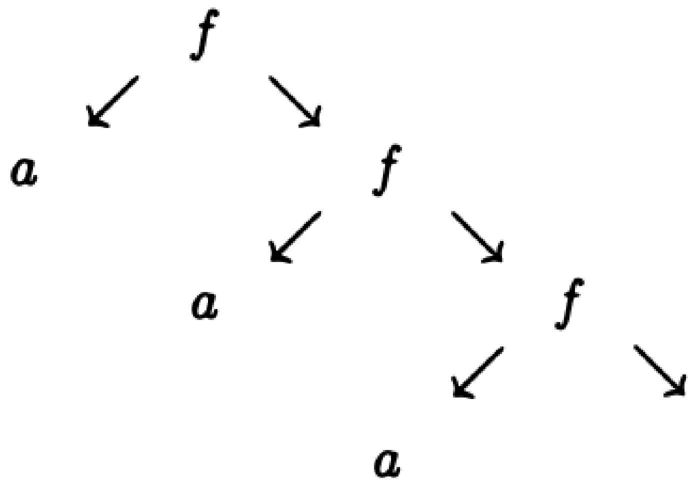
Une classification des arbres

1. Arbres finis

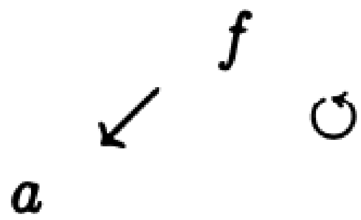
$$\begin{array}{ccccccc} & & f & & & & \\ & \swarrow & & \searrow & & & \\ a & & & & g & & \\ & & & \swarrow & \downarrow & \searrow & \\ & & b & & x & & y \end{array}$$

Une classification des arbres

2. Arbres infinis rationnels (c'est-à-dire avec nombre fini de sous-arbres)

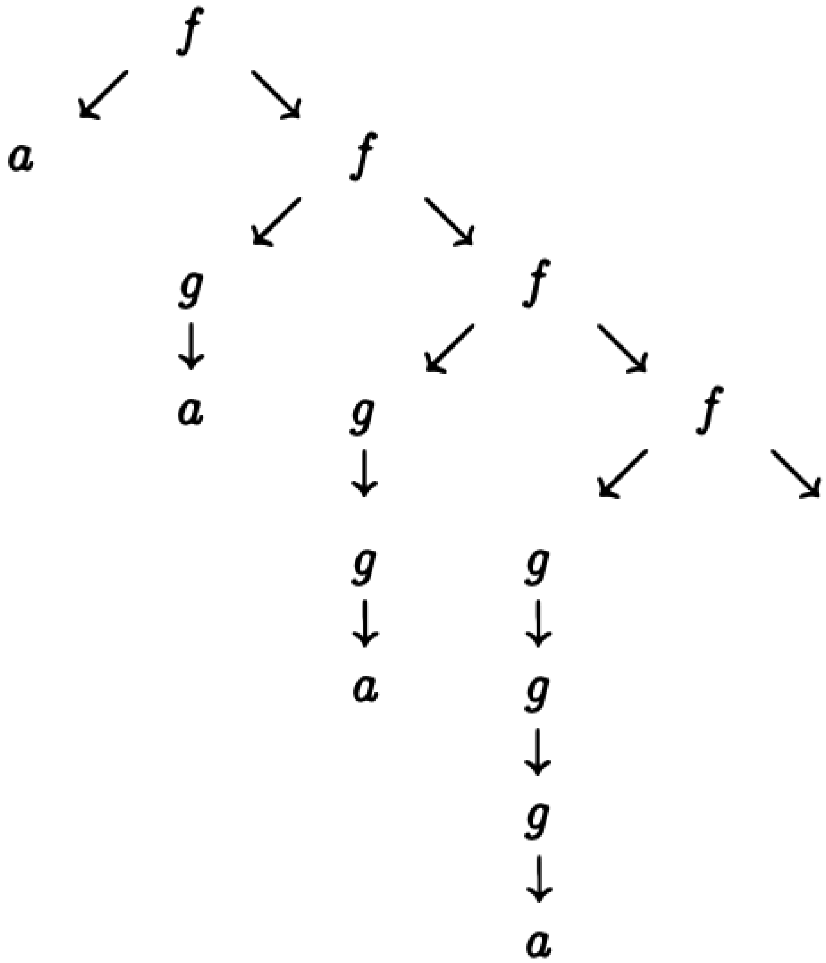


C'est-à-dire :



En notation linéaire : $f(a, f(a, f(a, \ldots.$

3. Arbres infinis non rationnels



En notation linéaire :

$$f(a, f(g(a), f(g(g(a))), f(\ldots$$

Nous formalisons dans les sections suivantes ces concepts traités jusqu'ici de façon intuitive.

Termes du premier ordre

Soient :
$\mathcal{V}$ un ensemble de symboles de variables :

$$\mathcal{V} = \{x_1, x_2, \ldots\}$$

$\mathcal{F}$ un ensemble de symboles de fonctions, contenant en particulier les constantes, dénotées $\mathcal{C}$ ($\mathcal{C} \subseteq \mathcal{F}$).

A chaque symbole de fonction est associé son *arité* ($n_i \geq 0\,;\ i \geq 1$), c'est-à-dire son nombre d'arguments.

Les constantes ont l'arité 0.

$$\mathcal{F} = \{f_1^{(n_1)}, f_2^{(n_2)}, \ldots\}$$

$$\mathcal{V} \cap \mathcal{F} = \emptyset$$

Termes du premier ordre

Définition 1 (termes). L'ensemble des *termes* construits sur $\{\mathcal{V}, \mathcal{F}\}$, noté $\Sigma(\mathcal{V}, \mathcal{F})$ est le plus petit ensemble tel que :

1. Si $x \in \mathcal{V}$, alors $x \in \Sigma(\mathcal{V}, \mathcal{F})$;

2. Si $a \in \mathcal{C}$, alors $a \in \Sigma(\mathcal{V}, \mathcal{F})$;

3. Si $f_k^{(n)} \in \mathcal{F}$ et $t_1, \ldots, t_n \in \Sigma(\mathcal{V}, \mathcal{F})$, alors $f_k^{(n)}(t_1, \ldots, t_n) \in \Sigma(\mathcal{V}, \mathcal{F})$.

Termes du premier ordre

Remarques.

a) La règle 2 est contenue dans la règle 3. Elle a été ajoutée pour une plus grande clarté.

b) Les termes sans variable (c'est-à-dire $\Sigma(\mathcal{F})$) sont appelés *termes clos* ou *termes fermés*.

c) Les arbres infinis ne sont pas des termes.

d) Sauf mention contraire nous noterons les variables $u, v, x, ...$ et les constantes $a, b, c, ...$

Termes du premier ordre

Définition 2 (variables, constantes, profondeur d'un terme).

– L'ensemble des *variables d'un terme* t, noté $Var(t)$ ou $V(t)$ est défini par :

$$V(t) = \begin{cases} \{t\} & \text{si} \quad t \in \mathcal{V} \\ \emptyset & \text{si} \quad t \in \mathcal{C} \\ \bigcup_{i=1}^{n} V(t_i) & \text{si} \quad t = f_k^{(n)}(t_1, \ldots, t_n) \end{cases}$$

– L'ensemble des *constantes d'un terme* t, noté $Const(t)$ ou $C(t)$ est défini par :

$$C(t) = \begin{cases} \emptyset & \text{si} \quad t \in \mathcal{V} \\ \{t\} & \text{si} \quad t \in \mathcal{C} \\ \bigcup_{i=1}^{n} C(t_i) & \text{si} \quad t = f_k^{(n)}(t_1, \ldots, t_n) \end{cases}$$

Termes du premier ordre

– La *profondeur d'un terme* t, notée $Prof(t)$ ou $P(t)$ est définie par :

$$P(t) = \begin{cases} 1 & \text{si} \quad t \in \mathcal{V} \\ 1 & \text{si} \quad t \in \mathcal{C} \\ 1 + \max(P(t_1), \ldots, P(t_n)) & \text{si} \quad t = f_k^{(n)}(t_1, \ldots, t_n) \end{cases}$$

On aurait pu prendre dans les deux premier cas $P(t) = 0$.

Substitutions

Définition 3 (substitution).

Une *substitution* est une application :

$$\sigma : \mathcal{V} \to \Sigma(\mathcal{V}, \mathcal{F})$$

qui est l'identité presque partout, c'est-à-dire sauf en un nombre fini de points.

On appelle *domaine* d'une substitution σ l'ensemble :

$$dom(\sigma) = \{x|\sigma(x) \neq x\}$$

On appelle *codomaine* (ang. *range*) d'une substitution σ l'ensemble :

$$codom(\sigma) = \{y|\exists x.\sigma(x) = y\}$$

Substitutions

On note une substitution en donnant ses valeurs pour les points où elle n'est pas l'identité (c'est-à-dire les valeurs sur son domaine) :

$$\{x_1 \leftarrow t_1, x_2 \leftarrow t_2, \ldots, x_n \leftarrow t_n\}$$

ou

$$\{x_1 \mapsto t_1, x_2 \mapsto t_2, \ldots, x_n \mapsto t_n\}$$

Une substitution :

$$\sigma : \mathcal{V} \to \Sigma(F)$$

est dite *close* ou *fermée*.

Substitutions

Si l'on a défini une substitution :

$$\sigma : \mathcal{V} \to \Sigma(\mathcal{V}, \mathcal{F})$$

on étend σ à $\Sigma(\mathcal{V}, \mathcal{F})$:

$$\overline{\sigma} : \Sigma(\mathcal{V}, \mathcal{F}) \to \Sigma(\mathcal{V}, \mathcal{F})$$

de la façon suivante :

$$\overline{\sigma}(t) = \begin{cases} \sigma(t) & \text{si} & t \in \mathcal{V} \\ t & \text{si} & t \in \mathcal{C} \\ f_k^{(n)}(\overline{\sigma}(t_1), \ldots, \overline{\sigma}(t_n)) & \text{si} & t = f_k^{(n)}(t_1, \ldots, t_n) \end{cases}$$

Dans la suite, on utilisera aussi σ pour l'extension.

Symbole d'égalité

L'égalité est d'habitude utilisée soit dans le sens de l'identité, comme par exemple dans :

$$(a + b)^2 = a^2 + 2ab + b^2$$

c'est-à-dire lorsque, pour *n'importe quelles* valeurs de a et b, l'identité se vérifie, soit au sens conditionnel, comme par exemple dans :

$$4 \times x = 16 \times y$$

où seulement *certaines* valeurs (éventuellement aucune) vérifient l'égalité.

Symbole d'égalité

Dans le cas de plusieurs solutions, certaines sont *plus intéressantes* que d'autres.

Par exemple, si nous nous intéressons aux solutions sur les entiers positifs de l'équation ci-dessus :

$$\{x = 4, y = 1\},\ \{x = 8, y = 2\},\ \{x = 12, y = 3\}, \ldots$$

sont des solutions.

La solution la plus générale est $\{x = 4 \times y\ ;\ y \in \mathbb{N}\}$.

Notation : pour mettre en évidence qu'il s'agit d'une égalité *conditionnelle*, on notera

$$4 \times x \doteq 16 \times y$$

et, s'il s'agit de termes, $t_1 \doteq t_2$.

Ordre des substitutions

Définition 4. On dit qu'une substitution σ est *plus générale* qu'une substitution γ si et seulement s'il existe une substitution λ telle que :

$$\gamma = \lambda \circ \sigma$$

où $\circ$ désigne la composition des fonctions.

Unification

Définition 5. Étant donnée l'équation $t_1 \doteq t_2$, avec $t_1, t_2 \in \Sigma(\mathcal{V}, \mathcal{F})$, le problème de *l'unification* est celui de trouver la substitution *la plus générale*, appelée *upg* (unificateur le plus général), telle que :

$$\sigma(t_1) = \sigma(t_2) \quad \text{(identité syntaxique)}$$

Si un seul des termes contient des variables (disons t_1), le problème qui consiste à trouver σ telle que

$$\sigma(t_1) = t_2$$

est appelé le problème du *filtrage*.

L'*algorithme d'unification*, donné ci-après, soit construit l'upg solution d'un ensemble d'équations dans les termes, soit il détecte qu'il n'existe pas de solution.

Des exemples seront vus en TD.

Unification

algorithme UNIFICATION
entrée : $\mathcal{S} = \{t_1 \doteq s_1, \ldots, t_n \doteq s_n\}$
sortie : soit une substitution σ (le **upg**) solution de $\mathcal{S}$
soit $\perp$ (pas de solution)
test d'arrêt : ***clash*** ou **cycle** ou aucune règle ne s'applique
%Γ denote un ensemble fini d'équations, au début $\Gamma = \mathcal{S}$
début
%$V(t)$ *et* $V(\Gamma)$ dénotent l'ensemble des variables dans t et dans Γ respectivement
%$\Gamma\{x \leftarrow t\}$ signifie :remplacer toutes les occurrences de la variable x dans Γ par t
%$Expr1 \longrightarrow Expr2$ signifie : remplacer $Expr1$ par $Expr2$
Pour chaque $t_i \doteq s_i$ $(1 \leq i \leq n)$ dans Γ appliquer les règles suivantes :
1 • $\{t \doteq t\} \bigcup \Gamma \longrightarrow \Gamma$
2 • si t n'est pas une variable :
$\{t \doteq x\} \bigcup \Gamma \longrightarrow \{x \doteq t\} \bigcup \Gamma$
3 • si x variable : $\{x \doteq t\} \bigcup \{x \doteq s\} \bigcup \Gamma \longrightarrow \{x \doteq t\} \bigcup \{t \doteq s\} \bigcup \Gamma$
4 • $\{f(t_1, \ldots, t_n) \doteq f(s_1, \ldots, s_n)\} \bigcup \Gamma \longrightarrow \{t_1 \doteq s_1\} \bigcup, \ldots, \bigcup \{t_n \doteq s_n\} \bigcup \Gamma$
5 • si $x \notin V(t)$ et $x \in V(\Gamma)$:
$\{x \doteq t\} \bigcup \Gamma \longrightarrow \{x \doteq t\} \bigcup \Gamma\{x \leftarrow t\}$
% par exemple si l'on a $x = h(y)$, $y = f(u)$ il **faut faire** $x = h(f(u))$, $y = f(u)$
6 • $\{f(t_1, \ldots, t_n) \doteq g(s_1, \ldots, s_m)\} \bigcup \Gamma \longrightarrow \perp$ (***clash*** **ou conflit**)
% c'est-à-dire les symboles de fonction sont constants
7 • si $x \in V(t)$ et t n'est pas une variable, c'est-à-dire si $t \neq x$:
% test d'occurrence (*occurs check*)
$\{x \doteq t\} \bigcup \Gamma \longrightarrow \perp$ **(cycle)**
% c'est-à-dire les termes infinis ne sont pas admis dans σ
fin