# Aspects Fondamentaux du Calcul : Le Problème de l'Unification

# Unification

1. but: résoudre des équations

# Unification: équations

1. Le but de l'unification est de résoudre des équations.
2. Un *problème équationnel* est un ensemble fini de paires (Attention il s'agit bien de *paires* et pas de *couples* !) de termes:

$$P = \{s_1 \stackrel{?}{=} t_1, ..., s_n \stackrel{?}{=} t_n\}$$

3. $s_i \stackrel{?}{=} t_i$ est une *équation*

# Convention

1. Si $t_i$ dans $s_i \stackrel{?}{=} t_i$ est une variable $x_i$,
   on écrira plutôt $x_i \stackrel{?}{=} s_i$ que $s_i \stackrel{?}{=} x_i$
2. La variable isolée, si elle existe est toujours à gauche.

# Solution

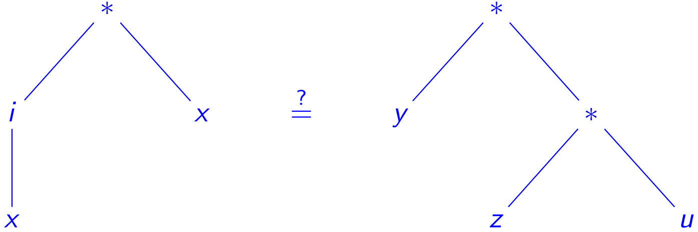
- Une *solution* de $P$ ou un *unificateur* de $P$ est une substitution $\sigma$ telle que

$$\begin{aligned}\sigma(s_1) &= \sigma(t_1)\\ &\vdots\\ \sigma(s_n) &= \sigma(t_n)\end{aligned}$$

- L'*ensemble des unificateurs* de $P$ est noté $\mathcal{U}(P)$.
- Si $\mathcal{U}(P) \neq \varnothing$ on dit que $P$ est *unifiable*.

# Exemple 1

Soit le problème :



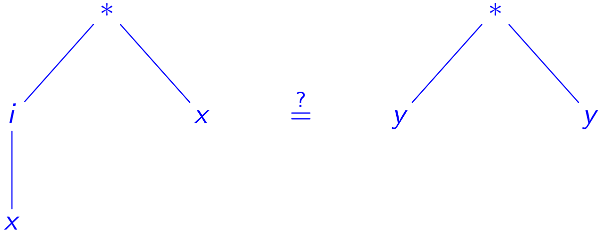

Les substitutions

$$\{x \mapsto i(u) * u, \quad y \mapsto i(i(u) * u), \quad z \mapsto i(u)\}$$
$$\{x \mapsto z * u, \quad y \mapsto i(z * u)\}$$

sont chacune des solutions.

# Exemple 2

Le problème :




n'a pas de solution, car il faudrait que $\sigma(x) = i(\sigma(x))$. Ce qui est impossible.

# Ordre sur les substitutions et MGU

1. L'ensemble $Sub(T(\Sigma, V))$ est muni d'un ordre $\lesssim$ défini par

$$\sigma \lesssim \tau \text{ si et seulement si } (\exists \rho \in Sub(T(\Sigma, V)) \quad \tau = \rho\sigma.$$

   $\sigma$ est dite *plus générale* que $\tau$.

2. L'*unificateur le plus général* de $P$ (ou *mgu*) est une substitution $\sigma$ telle que
   - $\sigma$ est un unificateur de $P$, c-à-d $\sigma \in \mathcal{U}(P)$.
   - Si $\theta \in \mathcal{U}(P)$ alors $\sigma \lesssim \theta$.

# Forme résolue

- Soit $P$ un problème contenant $x \stackrel{?}{=} t$.
  $x \stackrel{?}{=} t$ est en *forme résolue* dans $P$ si $x$ n'apparaît nulle part ailleurs dans $P$.
  En particulier,
  - $x \notin Var(t)$
  - et il n'y a pas d'autre équation de la forme $x \stackrel{?}{=} s$.
- $x$ est alors dite *résolue*.
- *Un problème est résolu* si toutes ses équations sont en forme résolue.

# Forme résolue et MGU

**Definition**

Si $P = \{x_1 \overset{?}{=} t_1, ..., x_n \overset{?}{=} t_n\}$ est résolu,
alors $\sigma = \{x_1 \mapsto t_1, ..., x_n \mapsto t_n\}$ est un mgu
de $P$.

On note $\sigma_P$ au lieu de $\{x_1 \mapsto t_1, ..., x_n \mapsto t_n\}$

# Unification: en pratique

- unification = résolution d'équations symboliques
- principe: faire "coller" 2 termes l'un en face de l'autre, en attrapant le "dur" (terme) avec le "mou" (variable)
- technique: exploration des 2 arbres en parallèle pour engendrer une substitution
  fonction partielle des variables avec les termes

# Algorithme

soit $\mathcal{P}$ un problème, et $\sigma$ une solution courante

- $\sigma = \varnothing$
- tant que $\mathcal{P}$ n'est pas vide
  - on prend une équation dans $\mathcal{P}$, plusieurs cas:
    - **décomposition**: $c(s_1, \ldots, s_n) = c(t_1, \ldots, t_n)$
      on ajoute les équations $s_i = t_i$ dans $\mathcal{P}$
    - **conflit**: $c(s_1, \ldots, s_n) = d(t_1, \ldots, t_l)$ **Echec**
    - **trivial**: $t = t$, rien à faire
    - **élimination de variable**: $x = t$ et $x \notin var(t)$
      $\sigma' = \{x \mapsto t\}$
      $\mathcal{P} = \sigma'\mathcal{P}$, $\sigma = \sigma'\sigma \cup \sigma'$
    - **cyclicité**: $x = t$ et $x \in var(t)$ et $x \neq t$ **Echec**
    - **orientation**: $x = t$ et $t \notin var(\mathcal{P})$
      $\mathcal{P} \cup \{x = t\}$
- fin tant que
- renvoyer $\sigma$ (**succès**)

# Algorithme: remarques

- quand on applique une substitution $x \mapsto t$, la variable $x$ disparaît
- algorithme non déterministe ("prendre").
  comment choisir l'équation pour être plus efficace?
- l'algorithme termine (test de cyclicité)
- élimination d'une variable: on préserve l'ensemble des solutions

# Algorithme: propriétés

- une execution qui réussit, est de la forme: $[\mathcal{P}, \varnothing] \Rightarrow \ldots \Rightarrow [\varnothing, \sigma]$ (séquence de couples [ problème, substitution])
    - **correction:** si $[\mathcal{P}, \varnothing] \Rightarrow [\varnothing, \sigma]$ alors $\sigma$ unifie les équations de $\mathcal{P}$ i.e., $\forall t = t' \in \mathcal{P}, \quad \sigma t = \sigma t'$
    - **complétude:** si $\sigma'$ unifie les équations de $\mathcal{P}$, alors tout calcul $[\mathcal{P}, \varnothing] \Rightarrow \ldots$ se termine sur un $\sigma$ qui généralise $\sigma'$ (i.e., $\exists \rho \; \sigma' = \rho\sigma$)
    - **idempotence** la substitution $\sigma$ obtenue est idempotente si $x \mapsto t \in \sigma$, alors $x$ n'a pas d'autre occurence dans $\sigma$
- donc $[\mathcal{P}, \varnothing] \Rightarrow \ldots \Rightarrow$ *echec* ssi $\mathcal{P}$ n'a pas de solution
- **conclusion:** le procédure d'unification calcule un unificateur le plus général lorsqu'il existe
  ou $\sigma$ décrit toutes les solutions au problème de départ

# Algorithme: complexité

- cet algorithme peut être exponentiel en espace et en temps
- (pour une équation (et peut être plus), i.e., cas de l'unification en Prolog) représenter les données qu'on manipule (termes, substitutions) de manière efficace permet d'obtenir des algorithmes plus efficaces (temps $\mathcal{O}(n^2)$, espace en $\mathcal{O}(n)$ en utilisant des DAG par exemple)
  il y aurait même des algo linéaires en temps (pas vérifié)
- le filtrage en Caml est de la "demi-unification"
  - filtrage: le motif "mange" la valeur (toutes les variables à gauche)
  - unification: les 2 côtés de l'égalité sont acteurs et ont le même rôle

# Des exemples

Variables: $\{x, y, z, t\}$

- $\{c(x, y) = c(f(a), g(a, b))\}$, solution $\{x \mapsto f(a), y \mapsto g(a, b)\}$
- $\{c(f(a), g(a, b)) = c(x, y)\}$, solution $\{x \mapsto f(a), y \mapsto g(a, b)\}$
- $\{f(t, c(e), d) = f(a, x, d)\}$, solution $\{t \mapsto a, x \mapsto c(e)\}$
- $\{f(a, b, x) = f(y, c, d)\}$, solution *echec*
- $\{c(x, y) = c(z, t)\}$, solution $\{z \mapsto x, t \mapsto y\}$ ou $\{y \mapsto t, x \mapsto z\}$ ou $\ldots$
- $\{c(x, a, b) = c(c, x, b)\}$, problème: terme incorrect
- $\{f(a) = f(a)\}$, solution $\{\}$
- $\{p(x, c, x) = p(a, y, a)\}$, solution $\{x \mapsto a, y \mapsto c\}$
- $\{f(x, g(x)) = f(z, z)\}$, solution *echec*