La définition $\mathtt{f = x} \mapsto \mathtt{sin\ (cos\ (sin\ x))}$ s'écrit parfois `f x = sin (cos (sin x))`. L'avantage de l'écriture $\mathtt{f = x} \mapsto \mathtt{sin\ (cos\ (sin\ x))}$ est qu'elle permet de distinguer deux opérations très différentes : la *construction* de la fonction $\mathtt{x} \mapsto \mathtt{sin\ (cos\ (sin\ x))}$ de la *définition* proprement dite qui consiste à attribuer un nom à un objet déjà construit. Il est souvent important, en informatique, d'avoir des notations qui permettent de construire des objets sans nécessairement leur attribuer un nom.

Dans ce livre, nous utiliserons la notation `fun x -> sin (cos (sin x))` pour exprimer cette fonction.

Le terme `fun x -> sin (cos (sin x))` exprime une fonction. En revanche son sous-terme `sin x` n'exprime rien du tout : ni un réel, ni une fonction, car il contient une *variable libre* dont la valeur n'est pas connue.

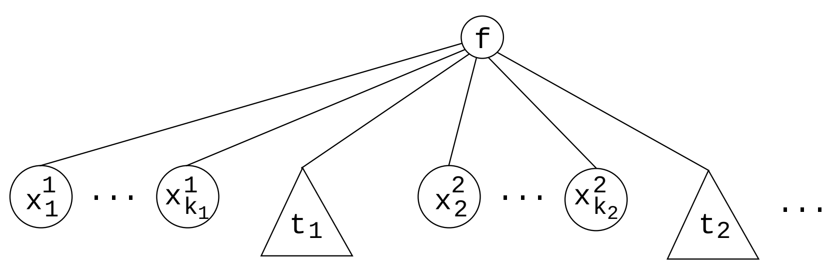
Pour avoir des variables liées dans les termes, il est nécessaire d'étendre la notion de terme en autorisant des variables libres, qui ne seront liées que plus tard. Cela demande aussi d'avoir des symboles, comme `fun` qui sont capables de lier des variables dans certains de leurs arguments. D'autres exemples de symboles lieurs sont le symbole `{ | }`, le symbole ∂/∂, le symbole $\int \mathtt{d}$, les symboles $\sum$ et $\prod$, les quantificateurs $\forall$ et $\exists$, ... Dans ce livre, nous utiliserons plusieurs symboles lieurs : le symbole `fun` ci-dessus, les symboles `fix`, `let`, `fixfun`...

L'arité d'un symbole `f` ne sera désormais plus un entier `n`, mais une suite finie d'entiers $(\mathtt{k}_1, \ldots, \mathtt{k}_\mathtt{n})$ qui indique que le symbole `f` lie $\mathtt{k}_1$ variables dans son premier argument, $\mathtt{k}_2$ variables dans le deuxième, ..., $\mathtt{k}_\mathtt{n}$ variables dans le $\mathtt{n}^{\text{ème}}$.

Ainsi, quand on s'est donné un langage — c'est-à-dire un ensemble de symboles munis d'une arité — et un ensemble infini dont les éléments sont appelés *variables*, on définit les termes inductivement par

- les variables sont des termes,
- si `f` est un symbole d'arité $(\mathtt{k}_1, \ldots, \mathtt{k}_\mathtt{n})$, $\mathtt{t}_1$, ..., $\mathtt{t}_\mathtt{n}$ sont des termes et $\mathtt{x}_1^1$, ..., $\mathtt{x}_{\mathtt{k}_1}^1$, ..., $\mathtt{x}_1^\mathtt{n}$, ..., $\mathtt{x}_{\mathtt{k}_\mathtt{n}}^\mathtt{n}$ sont des variables, alors $\mathtt{f}(\mathtt{x}_1^1 \ldots \mathtt{x}_{\mathtt{k}_1}^1\ \mathtt{t}_1, \ldots, \mathtt{x}_1^\mathtt{n} \ldots \mathtt{x}_{\mathtt{k}_\mathtt{n}}^\mathtt{n}\ \mathtt{t}_\mathtt{n})$ est un terme.

La notation $\mathtt{f}(\mathtt{x}_1^1 \ldots \mathtt{x}_{\mathtt{k}_1}^1\ \mathtt{t}_1, \ldots, \mathtt{x}_1^\mathtt{n} \ldots \mathtt{x}_{\mathtt{k}_\mathtt{n}}^\mathtt{n}\ \mathtt{t}_\mathtt{n})$ désigne l'arbre



Cette définition se comprend mieux en donnant un exemple. On construit un langage dans lequel les termes expriment des nombres réels et des fonctions réelles et qui comprend deux constantes `sin` et `cos` pour les fonctions sinus et cosinus, un symbole α, appelé *application*, tel que $\alpha(\mathtt{f,x})$ soit l'objet obtenu en appliquant la fonction `f` à l'objet `x` et un symbole `fun` pour construire des