Définition 1.10 (Fermeture réflexive-transitive)

Soit R est une relation binaire sur un ensemble E, la *fermeture réflexive-transitive* de la relation R est la relation R^* inductivement définie par les règles
- $t\ R^*\ t$,
- si $t\ R\ t'$ et $t'\ R^*\ t''$, alors $t\ R^*\ t''$.

Si $t\ R^*\ t'$, une dérivation du couple (t, t') est une suite finie $t_0, \ldots, t_n$, telle que $t_0 = t$, $t_n = t'$ et pour tout $i \leq n-1$, $t_i\ R\ t_{i+1}$.

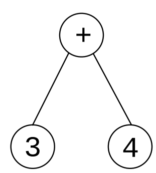
Si on voit R comme un graphe orienté, les dérivations sont les chemins de ce graphe et la relation R^* relie donc deux sommets, quand il y a un chemin qui mène de l'un à l'autre.

1.2 Les langages

1.2.1 Les langages sans variables

Maintenant que nous avons introduit la notion de définition inductive, nous allons l'utiliser pour définir la notion de *langage*. Nous allons dans un premier temps définir une notion très générale qui comprendra aussi bien les langages de programmation que les langages logiques. Puis nous définirons, dans un second temps, les langages de la logique des prédicats.

Nous cherchons, par ailleurs, à définir une notion de langage qui s'affranchit des conventions syntaxiques superficielles, par exemple de savoir si on écrit 3+4, $+(3, 4)$, ou encore 3 4 +. Cette expression sera plus abstraitement exprimée par un arbre



Chaque nœud de cet arbre est étiqueté par un symbole. Le nombre d'enfants d'un nœud de l'arbre dépend du symbole qui l'étiquette — 2 enfants si ce symbole est $+$, 0 si c'est 3 ou 4, ...

Un langage est donc un ensemble de symboles munis d'un entier appelé *arité*, ou plus simplement *nombre d'arguments*, de ce symbole. Les symboles sans arguments sont appelés des *constantes*.

L'ensemble des *expressions* de ce langage est l'ensemble d'arbres défini inductivement par la règle suivante.