# Semantic tree operations

D. Gueorguiev 1/6/2021

Notation:
$T^{(m)}$ – an $m$-ary tree
$\mathcal{V}(T)$ - the set of all nodes of $T$
$P(v, T)$ – path from root of $T$ to node $v$
$(k, v)$ – tree tuple of $T$
$k$ – tree factor, an integer
$\{\dot{k}_1, \dot{k}_2, \dots, \dot{k}_h\}$ - tuple of primitive factors
$\dot{k}$ – primitive factor, an integer
$\mathcal{K}(T)$ – the set of all primitive factor tuples of $T$
$\mathcal{A}(T)$ – the set of all arcs of $T$

## Summary and definitions

We are considering $m$-ary tree $T^{(m)}$ (abbreviated with $T$ from now on), which is a tree in which each node has at most $m$ children. Let us introduce the *tree tuple* $(k, v)$ where $v$ represents a *tree node* which is a semantic particle or a *subtree* of nodes (i.e. semantic particles). We will denote by $\mathcal{V}(T)$ the set of all nodes which belong to the tree $T$.
Here $k$ is a *tree factor* which encodes uniquely the position of the node or the root of the subtree $v$ in the parent tree $T$. More precisely, the factor $k$ encodes uniquely the path $P(v, T)$ from the root of the tree to the node $v$ or the root of subtree associated with $k$. Each tree factor $k$ representing a node other than the root of $T$ can be decomposed into a *tuple* $t$ of *primitive factors* $\{\dot{k}_1, \dot{k}_2, \dots, \dot{k}_h\}$ where $h$ is smaller or equal to the height of the tree. Here with dot-accented $\dot{k}$ we denote a primitive factor i.e. a factor which cannot be represented by any other combination of primitive factors.
We will denote with $\mathcal{K}(T)$ the set of all tuples of primitive factors associated with tree $T$. The position in $t$ of each of those primitive factors and their value encodes an *arc* from the set of arcs forming the path $P(v, T)$.

**Definition**: *Arc* of semantic tree
Let us define a tree with $N$ nodes as:

$$T = \sum_{i=0}^{N} (k_i, v_i) \qquad (1)$$

We denote with $t_i$ the tuple of the primitive factors for $k_i$:

$$t_i = \{\dot{k}_{1,i}, \dot{k}_{2,i}, \dots, \dot{k}_{h,i}\}\,. \qquad (2)$$

Every arc of $T$ is represented by a tuple $t = \{\dot{k}_1, \dot{k}_2, \dots, \dot{k}_l\}$ such that $t \in \mathcal{K}(T)$. However not every tuple $t \in \mathcal{K}(T)$ represents an arc - the tuple $\{\dot{k}_0\}$ associated with the root of $T$ does not correspond to an arc of $T$. We will denote with $\mathcal{A}(T)$ the set of all arcs of $T$. Obviously, $|\mathcal{A}(T)| = |\mathcal{K}(T)|$-1 $=$ $|\mathcal{V}(T)|$-1.

If we substitute (2) in (1) we come up with the following simplified notation for $T$ represented in terms of its primitive factors which will be used throughout this discussion:

$T \ = \sum_{i=0}^{N} (\dot{k}_{1,i}, \dot{k}_{2,i}, \ldots, \dot{k}_{h,i}, v_i)$ (3)

*Example*

$$
\begin{array}{ccccccc}
 & & & V_3 & & & \\
 & & / & | & \backslash & & \\
 & / & & | & & \backslash & \\
 & V_2 & & V_4 & & V_7 & \\
/ & & / & & \backslash & & \backslash \\
V_1 & & V_5 & & V_6 & & V_8
\end{array}
$$

The root node $V_3$ is given with the tuple $(k_0, v_3)$. Its children are given with $(k_1, v_2)$, $(k_2, v_4)$, and $(k_3, v_7)$. Their children are accordingly $(k_4, v_1)$, $(k_5, v_5)$, $(k_6, v_6)$, and $(k_7, v_8)$. In a later paragraph we will introduce rules which will allow us to show that the following expansions take place:

$k_1 = \dot{k}_1 \dot{k}_0$
$k_2 = \dot{k}_2 \dot{k}_0$
$k_3 = \dot{k}_3 \dot{k}_0$
$k_4 = \dot{k}_1 \dot{k}_1$
$k_5 = \dot{k}_2 \dot{k}_1$
$k_6 = \dot{k}_2 \dot{k}_2$
$k_7 = \dot{k}_3 \dot{k}_1$

Thus, the tree $T$ can be written as:
T = $(\dot{k}_0, v_3) + \ (\dot{k}_1 \dot{k}_0, v_2) + \ (k_2 k_0, v_4) + \ (\dot{k}_3 \dot{k}_0, v_7) + \ (\dot{k}_1 \dot{k}_1, v_1) + (\dot{k}_2 \dot{k}_1, v_5) + (\dot{k}_2 \dot{k}_2, v_6) +$ $(\dot{k}_3 \dot{k}_1, v_8)$.

Notice that for the root of the tree we always have:

$k_0 = \dot{k}_0$ (4)

Definition: Weighted semantic tree
Let us assume that the two trees are weighted so that for each tree there is weight function which maps each tuple factor to a real number which is the weight corresponding the arc associated with the specified tuple factor. Let us denote by $f$ and $f^*$ the two weight functions corresponding to $T$ and $T^*$. Generally, $f(k_i) \neq f^*(k_i)$. If $f \equiv f^*$ then $T$ and $T^*$ have the same weights on their matching arcs.

**Definition**: *Semantic significance vector of a semantic tree*
//TODO

## Semantic subtree expansion and node comparison

The following operations are defined for tree factors:

## Multiplication operation for semantic tuple factors

One possible implementation for the primitive factors $\dot{k}_1, \dot{k}_2, \ldots\ \dot{k}_m$ is to define them as the $m$ digits greater than 0 of $(m+1)$-nary number system such that $0 = \dot{k}_0 < \dot{k}_1 < \dot{k}_2 < \ \ldots < \dot{k}_m$. We define an operation `$*$` denoting digit concatenation $\dot{k}_i * \dot{k}_j = (m+1)\dot{k}_i + \dot{k}_j$. Obviously,
$\dot{k}_i * \dot{k}_j > \dot{k}_i$ for any pair $i, j = 1..m$
Note that the latter implies that

$\dot{k}_{i_1} * \dot{k}_{i_2} * \ldots * \dot{k}_{i_n} > \dot{k}_{j_1} * \dot{k}_{j_2} * \ldots * \dot{k}_{j_{n-1}}$ for any tuple where $i_p, j_q = 1..m, p = 1..n, q = 1..n-1$

$$\left(\dot{k}_i, \left(\dot{k}_j, v_j\right)\right) = \left(\dot{k}_i * \dot{k}_j, v_j\right)$$

Encoding a complete $m$-ary tree $T$ of height $h$ with the algebraic notation above:
$T = \left(\dot{k}_0, v_0^0\right)$ . Further we will assume that $\dot{k}_0 = 0$.
$v_0^0 = \left(\dot{k}_0, v_0^1\right) + \left(\dot{k}_1, v_1^1\right) + \left(\dot{k}_2, v_2^1\right) + \ \ldots\ + \left(\dot{k}_m, v_m^1\right)$

In general, we have:
$v_q^p = \left(\dot{k}_0, v_{(q-1)m}^{p+1}\right) + \left(\dot{k}_1, v_{(q-1)m+1}^{p+1}\right) + \left(\dot{k}_2, v_{(q-1)m+2}^{p+1}\right) + \ \ldots\ + \left(\dot{k}_m, v_{qm}^{p+1}\right)$
where $q = 1..m^h, p = 1..h$
Obviously, we have at most $\frac{(m^{h+1}-1)}{m-1}$ distinct terms $v_q^p$ which represent nodes i.e. semantic values.

## Tuple factor comparison operator

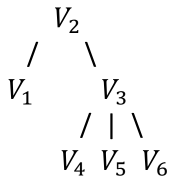
The expression for the tree also can be written as:
$T = \sum_{i=0}^{N}(k_i, v_i)$ where $N \le \frac{(m^{h+1}-1)}{m-1}$ and $k_i$ are the *node factors* given with $k_i = \dot{k}_{i_1} * \dot{k}_{i_2} * \ldots * \dot{k}_{i_n}; n \le h$. The node values $v_i$ are the values $v_q^p$ ordered in increasing order of $k_i$. This order corresponds to *level order traversal* of the $m$-ary tree. Note that with appropriately defined comparison operation `<` we can model different ways of traversing the $m$-ary tree. For instance, if we define `<` as the comparison for the values of $\left(m+1-\dot{k}_{i_1}\right)m^{n-1} + \left(m+1-\dot{k}_{i_2}\right)m^{n-2} + \cdots + \left(m+1-\dot{k}_{i_n}\right)$ we will have ordering which corresponds to the *preorder traversal* of the tree.

<u>*Example*</u>
*Peter is Dimitar's son.*
*Dimitar's son has a friend in the neighborhood and his friend's name is James.*

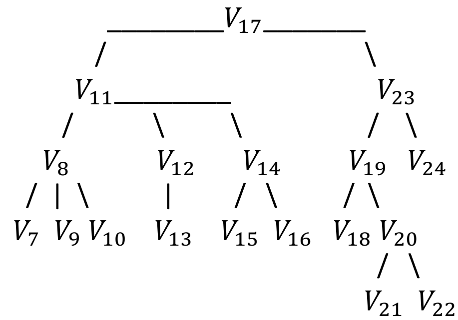
- ⇨ *James is Peter's friend*

*Peter is the son of Dimitar.*
$V_1$ $V_2$ $V_3$ $V_4$ $V_5$ $V_6$
*The son of Dimitar has a friend in the neighborhood and the name of his friend is James.*
$V_7$ $V_8$ $V_9$ $V_{10}$ $V_{11}$ $V_{12}$ $V_{13}$ $V_{14}$ $V_{15}$ $V_{16}$ $V_{17}$ $V_{18}$ $V_{19}$ $V_{20}$ $V_{21}$ $V_{22}$ $V_{23}$ $V_{24}$

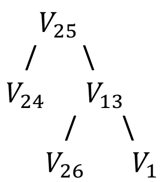
$T_1$




$T_2$




$T_3$




Expressing $T_1$ with the algebraic notation discussed earlier:

$$T_1 = (k_0, v_2) + (k_1, v_1) + \left(k_2, \left((k_0, v_3) + (k_1, v_4) + (k_2, v_5) + (k_3, v_6)\right)\right)$$

which is expanded to:

$$T_1 = (k_0, v_2) + (k_1, v_1) + (k_2 k_0, v_3) + (k_2 k_1, v_4) + (k_2 k_2, v_5) + (k_2 k_3, v_6)$$

Expressing $T_2$ with the algebraic notation yields:

$$T_2 = (k_0, v_{17}) + \Bigg(k_1, \bigg((k_0, v_{11}) + \Big(k_1, \big((k_0, v_8) + (k_1, v_7) + (k_2, v_9) + (k_3, v_{10})\big)\Big) + \Big(k_2, \big((k_0, v_{12}) + (k_1, v_{13})\big)\Big) + \Big(k_3, \big((k_0, v_{14}) + (k_1, v_{15}) + (k_2, v_{16})\big)\Big)\bigg)\Bigg) + \Bigg(k_2, \bigg((k_0, v_{23}) + \Big(k_1, \big((k_0, v_{19}) + (k_1, v_{18}) + \big(k_2, ((k_0, v_{20}) + (k_1, v_{21}) + (k_2, v_{22}))\big)\big)\Big) + (k_2, v_{24})\bigg)\Bigg)$$

which is expanded to:

$$\begin{aligned} T_2 = {} & (k_0, v_{17}) + (k_1 k_0, v_{11}) + (k_1 k_1 k_0, v_8) + (k_1 k_1 k_1, v_7) + (k_1 k_1 k_2, v_9) + (k_1 k_1 k_3, v_{10}) \\ & + (k_1 k_2 k_0, v_{12}) + (k_1 k_2 k_1, v_{13}) + (k_1 k_3 k_0, v_{14}) + (k_1 k_3 k_1, v_{15}) + (k_1 k_3 k_2, v_{16}) \\ & + (k_2 k_0, v_{23}) + (k_2 k_1 k_0, v_{19}) + (k_2 k_1 k_1, v_{18}) + (k_2 k_1 k_2 k_0, v_{20}) + (k_2 k_1 k_2 k_1, v_{21}) \\ & + (k_2 k_1 k_2 k_2, v_{22}) + (k_2 k_2, v_{24}) \end{aligned}$$

## Semantic Tree Difference

Let us have two trees represented as:

$T = \sum_{i=0}^{N} (k_i, v_i)$ and $T^* = \sum_{i=0}^{N} (k_i^*, v_i)$

Here $v_i, i = 1..N$ denote the nodes of the two trees (not subtrees) which are semantic particles with signatures $ssig(v_i)$. The sequences $k_i$ and $k_i^*$ denote the sequences of tuple factors which encode the position of each node $v_i$ in each of the two trees.

**Definition**: *Matching arcs of trees*
Let us assume that the two trees are weighted so that for each tree there is weight function which maps each tuple factor to a real number which is the weight corresponding the arc associated with the specified tuple factor. Let us denote by $f$ and $f^*$ the two weight functions corresponding to $T$ and $T^*$. Generally, $f(k_i) \neq f^*(k_i)$. If $f \equiv f^*$ then $T$ and $T^*$ have the same weights on their matching arcs.

We have the same set of semantic particles but they are arranged differently in two trees. We would like to define metric how different are the two trees.

//TODO