Algorithm 1 Débinariation

```
function Binaire2Naire($tree : list$)
    $tmp \leftarrow$ list(; )
    $bin \leftarrow \downarrow$;
    for $feuille \in tree$ do
        if isinstance($feuille, list$) then
            if $bin \in feuille[0]$ then
                $tmp.extend$(unbin(feuille[1 :]));
            else
                $tmp.append$(unbin(feuille))
        else
            $tmp.append(feuille)$
    return tmp
function Lexicale2Unaire($tree$)
    if $|tree| == 3$ then
        $head, leftchild, rightchild \leftarrow tree$;
        if $\uparrow \in head$ then
            $gauche, droite \leftarrow$ split(bin, head, 1)
            return [$gauche$, generalUnbin([droite, leftchild, rightchild])]
        else
            return [$head$, Lexicale2Unaire(leftchild), Lexicale2Unaire(rightchild)]
    else if $|tree| == 2$ then
        $head, child \leftarrow tree$;
        if $\uparrow \in head$ then
            $gauche, droite \leftarrow$ split(bin, head, 1)
            return [gauche, Lexicale2Unaire([droite, child])]
        else
            return [head,child]
function generalUnbin(tree)
    $firststep \leftarrow$ Lexicale2Unaire(tree)
    return generalUnbin(firststep)
```