Declaratieve Talen
Prolog 1

1 Balanced trees

A tree can either be empty (`nil`) or exist of a node (`t`) with a value and two subtrees.

$$\text{Tree} := \text{nil} \mid \text{t}(\text{Tree}, \text{Value}, \text{Tree})$$

Hence, the expression `t(t(nil,2,nil),3,nil)` represents a tree. A tree is balanced if the depths of the left and right subtree differ by at most one, and both subtrees are balanced as well.

Implement a predicate `balanced/1` that succeeds if a given tree is balanced and fails in all other cases.

```
?- balanced(nil).
true.

?- balanced(t(nil,3,nil)).
true.

?- balanced(t(nil,3,t(nil,4,nil))).
true.

?- balanced(t(nil,3,t(nil,4,t(nil,2,nil)))).
false.
```

Implement a predicate `add_to/3` that adds an element to a balanced tree, and ensures that the tree remains balanced. The tree does not have to be sorted.

```
?- add_to(t(t(nil,3,nil),2,nil),4,Tree).
Tree = t(t(nil,3,nil),2,t(nil,4,nil))
```

Extra: Perform all required changes to store in each node both a value and the depth of the tree at that point. Adapt the `add_to/3` predicate such that its complexity decreases.