Algorithm 1 3DO refinement

Require: $\forall\ v \in graph \quad v.updated \leftarrow$ True
Require: $\forall\ v \in graph \quad v.unambiguousFinal \neq$ NULL

$order \leftarrow \mathcal{S}(V)$
$graph \leftarrow (V, E)$
$maxTimes \leftarrow$??
$metaconvergence \leftarrow$ False
$stateStack \leftarrow$ empty
$times \leftarrow 0$
repeat
    $convergence \leftarrow$ True
    for $v \in graph$ **do**
        if v is a leaf node **then**
            continue
        else if $\nexists\ n \in v.neighbors\ S.T.\ n.updated = True$ **then**
            continue
        else
            $prevValue \leftarrow v.unambiguousFinal$
                ▷ remember current value for later comparison
            $(v.unambiguousFinal, v.ambiguousFinal) \leftarrow$
                $3DO(neighbor1.unambiguousFinal$
                , $neighbor2.unambiguousFinal$
                , $neighbor3.unambiguousFinal)$
            if $prevValue = v.unambiguousFinal$ **then**
                $v.updated \leftarrow False$
            else
                $v.updated \leftarrow$ True
                $convergence \leftarrow$ False
        $times \leftarrow time + 1$
    if convergence **then**
        textbfbreak
    else
        stateStack.push (graph)
        if prefix cycle in stateStack **then** ▷ Metaconvergence defined as cycle through one or more previous states
            $metaconvergence \leftarrow$ True
        else
            $metaconvergence \leftarrow$ False
until $times = maxTimes$ **or** $metaconvergence$