Warn: *Incorect*. “No Dead *Transitions*” cannot currently be validated.

We need to assert that something is true in at least 1 possible behavior. *TLC* does not have the ability to make this type of assertion (I think). I think we would need to implement this one manually outside of TLA+ managing states which ruins the elegance of the current specification.

```
──────────────── MODULE WFNet_Example ────────────────

LOCAL INSTANCE TLC

 \* Simple case
 Places ≜ {“source”, “p1”, “sink”}
 Transitions ≜ {“t1”, “t2”}
 Arcs ≜ [
    source ↦ {“t1”},
    p1 ↦ {“t2”},

    t1 ↦ {“p1”},
    t2 ↦ {“sink”}
 ]

 \* Net that breaks NoDeadTransitions. We can’t check across all behaviors
 so this cannot be enforced correctly
 Places ≜ {“source”, “p1”, “p2”, “sink”}
 Transitions ≜ {“t1”, “t2”, “t3”, “t4”}
 Arcs ≜ [
    source ↦ {“t1”, “t2”},
    p1 ↦ {“t3”},
    p2 ↦ {“t4”},

    t1 ↦ {“p1”},
    t2 ↦ {“p2”},
    t3 ↦ {“sink”},
    t4 ↦ {“sink”}
 ]

 Requires strong fairness to show “option to complete”
Places ≜ {“source”, “p1”, “sink”}
Transitions ≜ {“t1”, “t2”, “t3”}
Arcs ≜ [
    source ↦ {“t1”, “t2”},
    p1 ↦ {“t3”},

    t1 ↦ {“sink”},
    t2 ↦ {“p1”},
    t3 ↦ {“source”}
]
SourcePlace ≜ “source”
SinkPlace ≜ “sink”
```

$\textsc{variable}\ \mathit{Marking}$

$\mathit{WFN} \triangleq \textsc{instance}\ \mathit{WFNet}$

$\mathit{Spec} \triangleq \mathit{WFN}\,!\,\mathit{Spec}$

$\mathit{Invariants} \triangleq \mathit{WFN}\,!\,\mathit{Invariants}$

$\mathit{FinalMarking} \triangleq \mathit{WFN}\,!\,\mathit{FinalMarking}([\mathit{sink} \mapsto 1])$

We have access to *Marking* here NOTE: we get a warning that *Marking* is not a part of the next-state relation of *WFNet* but it is threaded through to the underlying *PetriNet* module correctly.

$\mathit{FinalMarkingManual} \triangleq \Diamond\Box(\mathit{Marking} = [\mathit{sink} \mapsto 1]\ @@\ [p \in \mathit{Places} \mapsto 0])$

$\mathit{ClassicallySound} \triangleq \mathit{WFN}\,!\,\mathit{ClassicallySound}$