Warn: Incomplete. "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"
```

VARIABLE $Marking$

$WFN \triangleq$ INSTANCE $WFNet$

$Spec \triangleq WFN\,!\,Spec$

$Invariants \triangleq WFN\,!\,Invariants$

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

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.

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

$ClassicallySound \triangleq WFN\,!\,ClassicallySound$