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*

From "Soundness of workflow nets: Classification, decidability, and analysis" by *WMP* van der *Aalst*.

Definition of a *Workflow* Net (or WF-Net)
1. There is a single source place i
2. There is a single sink place o
3. Every node is on a path from i to o
4. There is no reset arc connected to the sink place

Classical soundness
- a. Option to complete
- b. Proper completion
- c. No dead transitions

I am assuming a conventional restriction: only 1 source token, "safe", and no arc weights.

Instantiate *WFNet* with (*Places* , *Transitions* , *Arcs* , *SourcePlace* , *SinkPlace*) constants and (*Marking*) variable. *Marking* variable should be declared but not assigned by users of this module.

CONSTANTS $Places$, $Transitions$, $Arcs$, $SourcePlace$, $SinkPlace$
$ArcWeights \triangleq \langle\rangle$ *PetriNet* arc weight not supported.
$InitialMarking \triangleq [p \in \{SourcePlace\} \mapsto 1]$

VARIABLE $Marking$
$vars \triangleq \langle Marking \rangle$

$PN \triangleq$ INSTANCE $PetriNet$

Invariants

1. There is a single source place i.
2. There is a single sink place o.

$$SourceSinkInvariant \triangleq \land SourcePlace \in Places \land SinkPlace \in Places \land SourcePlace \neq SinkPlace$$

4. There is no reset arc connected to the sink place.

$$NoResetArcInvariant \triangleq \neg\exists\, k \in \text{DOMAIN } Arcs : k = SinkPlace$$

$$Invariants \triangleq \land SourceSinkInvariant \land NoResetArcInvariant \land PN!Invariants$$ From *PetriNet*

Operators

From *PetriNet*

$M^* \triangleq PN!^*(M)$
$Inputs(t) \triangleq PN!Inputs(t)$
$Outputs(t) \triangleq PN!Outputs(t)$
$Enabled(t) \triangleq PN!Enabled(t)$

Properties

a. Option to complete. "b. Proper completion" is implied.
Note: this requires strong fairness on firing!

$OptionToComplete \triangleq \Diamond\Box(Marking = [p \in \{SinkPlace\} \mapsto 1]^*)$

c. No dead transitions (and no dead places from "3. Every node is on a path from i to o").
TODO: This is wrong!! 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.
$NoDeadTransitions \triangleq \forall\, t \in Transitions: \neg\Box(\neg Enabled(t)) \setminus *$ This is wrong : (

$ClassicallySound \triangleq \land\, OptionToComplete$
$\land\, NoDeadTransitions$

From *PetriNet*

$Reachable(x) \triangleq PN!Reachable(x)$
$FinalMarking(x) \triangleq PN!FinalMarking(x)$
$Bound(x) \triangleq PN!Bound(x)$
$IsStateMachine \triangleq PN!IsStateMachine$
$IsMarkedGraph \triangleq PN!IsMarkedGraph$
$IsFreeChoiceNet \triangleq PN!IsFreeChoiceNet$

Spec

Strong fairness used in WF-Nets to allow for classical soundness.

$Init \triangleq Marking = InitialMarking^*$
$Next \triangleq \exists\, t \in Transitions : PN!Fire(t)$

$Spec \triangleq Init \land \Box[Next]_{vars} \land (\forall\, t \in Transitions : \mathrm{SF}_{vars}(PN!Fire(t)))$