

LOCAL INSTANCE *Utilities*

GIVEN the set of FA tasks:

CONSTANT *Tasks*

```
e.g. Tasks ==
{ [ name |-> "EVI", repeatable |-> FALSE, group |-> "non-destructive" ]
, [ name |-> "IVI", repeatable |-> FALSE, group |-> "non-destructive" ]
}
```

such that it adheres to the expected structure

ASSUME *IsFiniteSet*(*Tasks*)
 ASSUME $\forall task \in Tasks : (\forall id \in DOM(task) : id \in \{ "name", "repeatable", "group" \})$
 ASSUME $\forall task \in Tasks : task.name \in STRING$
 ASSUME $\forall task, otherTask \in Tasks : task.name = otherTask.name \Rightarrow task = otherTask$
 ASSUME $\forall task \in Tasks : task.repeatable \in BOOLEAN$
 ASSUME $\forall task \in Tasks : task.group \in \{ "destructive", "non-destructive", "both" \}$

with the following helper definitions

$TaskNames \triangleq \{ task.name : task \in Tasks \}$

$unknownTask \triangleq [name \mapsto "UNKNOWN", repeatable \mapsto FALSE, group \mapsto "both"]$

ASSUME $unknownTask \notin Tasks$

ASSUME $\forall task \in Tasks : task.name \neq unknownTask.name$

$Task(t) \triangleq ChooseOrDefault(Tasks, LAMBDA task : task.name = t, unknownTask)$

GIVEN the set of connections between FA tasks:

CONSTANT *Connections*

```
e.g. Connections ==
{ [ name |-> "has_successor", srcName |-> "EVI", dstName |-> "IVI" ]
}
```

such that it adheres to the expected structure

ASSUME *IsFiniteSet*(*Connections*)
 ASSUME $\forall conn \in Connections :$
 $\quad \forall id \in DOM(conn) : id \in \{ "name", "srcName", "dstName" \}$
 ASSUME $\forall conn \in Connections : conn.name \in$
 $\quad \{ "has_successor"$
 $\quad , "has_predecessor"$
 $\quad , "has_mandatory_predecessor"$
 $\quad , "has_mandatory_successor"$
 $\quad \}$
 ASSUME $\forall conn \in Connections : \exists task \in Tasks : task.name = conn.srcName$
 ASSUME $\forall conn \in Connections : \exists task \in Tasks : task.name = conn.dstName$

ASSUME $\forall conn \in Connections : conn.srcName \neq conn.dstName$

such that the induced binary relation satisfies some properties

$RequiresRel[x \in TaskNames, y \in TaskNames] \triangleq$

$\exists conn \in Connections :$

$\wedge conn.name = \text{"has_mandatory_successor"} \vee conn.name = \text{"has_mandatory_predecessor"}$

$\wedge conn.srcName = x$

$\wedge conn.dstName = y$

$ConRel[x \in TaskNames, y \in TaskNames] \triangleq$

$\vee \exists conn \in Connections :$

$\wedge conn.name = \text{"has_successor"} \vee conn.name = \text{"has_mandatory_successor"}$

$\wedge conn.srcName = x$

$\wedge conn.dstName = y$

$\vee \exists conn \in Connections :$

$\wedge conn.name = \text{"has_predecessor"} \vee conn.name = \text{"has_mandatory_predecessor"}$

$\wedge conn.srcName = y$

$\wedge conn.dstName = x$

$TransConRel \triangleq ReflexiveTransitiveClosure(ConRel, TaskNames)$

ASSUME $IsIrreflexive(ConRel, TaskNames)$

ASSUME $Is2Acyclic(TransConRel, TaskNames)$

with the following helper definitions

$unknownConnection \triangleq [name \mapsto \text{"UNKNOWN"}, srcName \mapsto \text{"UNKNOWN"}, dstName \mapsto \text{"UNKNOWN"}]$

ASSUME $unknownConnection \notin Connections$

ASSUME $\forall conn \in Connections : conn.name \neq unknownConnection.name$

$ErrorConn(s, d, CONSTRAINT(-, -)) \triangleq$

IF $\neg CONSTRAINT(s, d)$

THEN $\{[name \mapsto \text{"has_successor"}, srcName \mapsto s, dstName \mapsto d]\}$

ELSE $\{\}$

$ErrorConns(s, d, CONSTRAINT(-, -)) \triangleq \{conn \in Connections : \neg($

\vee

$\wedge conn.name = \text{"has_successor"}$

$\vee conn.name = \text{"has_mandatory_successor"}$

$\wedge conn.srcName = s \wedge conn.dstName = d$

\vee

$\wedge conn.name = \text{"has_predecessor"}$

$\vee conn.name = \text{"has_mandatory_predecessor"}$

$\wedge conn.srcName = d \wedge conn.dstName = s$

$\Rightarrow CONSTRAINT(s, d)$

$\})$

WHEN checking for errors in an input workflow

CONSTANT *Workflow*

e.g. *Workflow* == << "EVI", "IVI" >>

such that it adheres to the expected structure

ASSUME $DOM(Workflow) = 1 \dots Len(Workflow)$ a proper tuple

ASSUME $\forall t \in RAN(Workflow) : t \in \text{STRING}$