Premature Start / Premature finish phenotype (Zero-order)

Definition: Action starts before the earliest starting time

Scenario: User moves on to infusion before settings are properly accounted for.

Forcing function applied: Patching with conditionals/guards

Property: The system must follow a trace that includes all necessary setup actions.

|  |  |
| --- | --- |
| Without Patching | With Patching |
| The user could potentially move on to infusion without setting up system requirements.  This leads to a dangerous assumption where the user might think infusion is happening when it isn’t.  Safety check  Analysing...  Depth 2 -- States: 2 Transitions: 2 Memory used: 6882K  Trace to property violation in SKIPPED:  turn\_on  confirm\_settings  Analysed in: 0ms | All sequences of actions eventually take the user to their destination safely.  The user is given confirmation steps along the way, so they are aware of the sequence of actions that they are engaged in.  Safety check  Analysing...  Depth 8 -- States: 15 Transitions: 24 Memory used: 16981K  No deadlocks/errors  Analysed in: 5ms |

Forcing function:

CONFIRM\_PATH = P[0][0][0],

P[rate:0..1][capacity:0..1][lock:0..1] = **<- Conditionals for setting rate, capacity and locking**

(

when(rate == 0)

setting\_rate->P[1][capacity][0]

|

when(capacity == 0)

setting\_capacity->P[rate][1][0]

|

when(lock == 0 && capacity == 1 && rate == 1)

lock\_line -> P[1][1][1]

|

when(lock == 1 && capacity == 1 && rate == 1) **<- All three actions must happen to move on**

confirm\_settings -> P[1][1][1]

).

NOTE:  
This patch can also cover the Premature finish phenotype, as premature starting from the perspective of one part of the machine is a premature finish from the other.