

# Systems Design Laboratory

## Gas Tanks

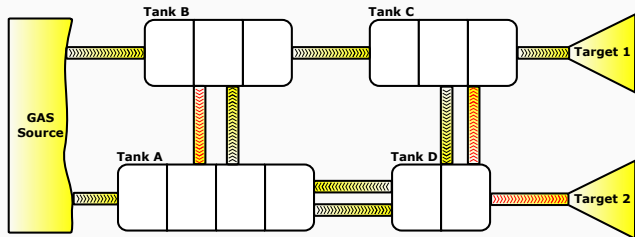
---

**Matteo Zavatteri**

<sup>1</sup>Department of Mathematics, University of Padova, ITALY

<sup>2</sup>Department of Computer Science, University of Verona, ITALY

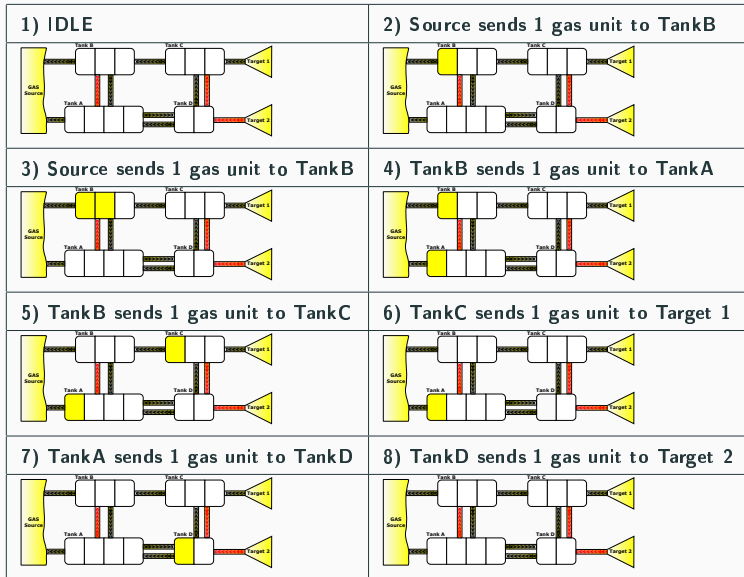
# Plant



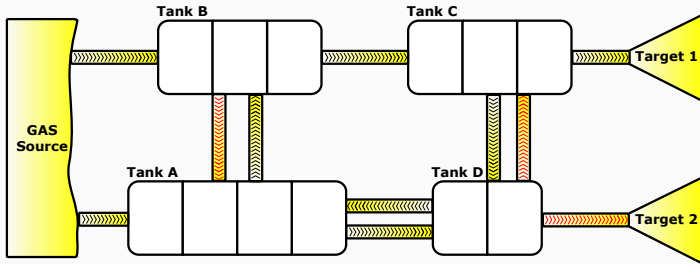
## Main components:

- An infinite source of gas
- 2 extraction points (leftmost pipes)
- 4 tanks of different capacity (given in units of gas, e.g.,  $m^3$ )
- 2 target points of infinite capacity
- Several other pipes to move single units of gas from one tank to another according to the direction of the arrows.
- Red pipes always allow to move gas (if any) from their source to their target.

# Usecase example



# The plant

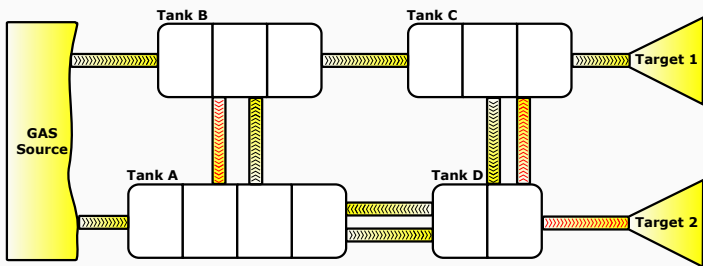


What should we model?



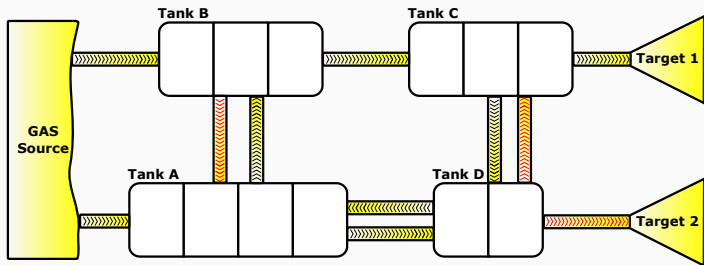
How many components?

# Tank A

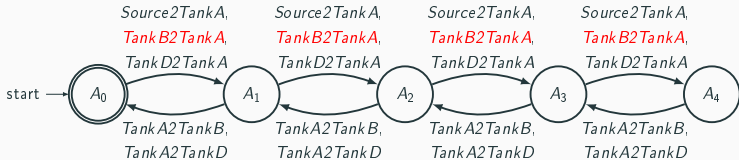
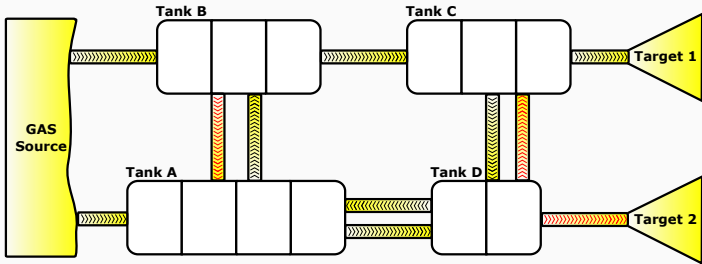


- States?
- Transitions?
- Event controllability?

# Tank A



# Tank A



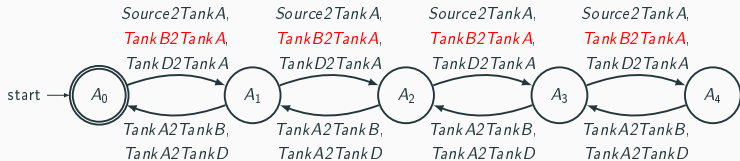
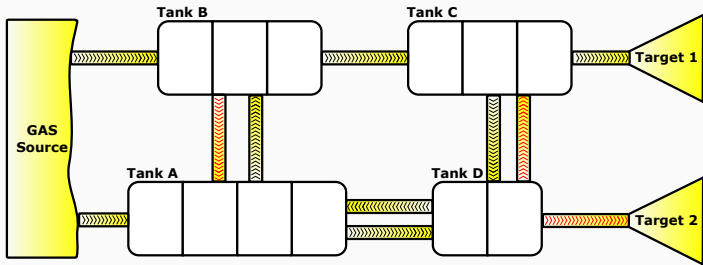
## Event modeling intuition:

*Source2TankA*: source supplies 1 gas unit to TankA

*TankB2TankA*: TankB supplies 1 gas unit, by removing it from itself, to TankA

...

# Tank A

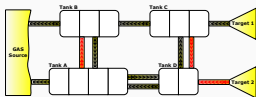


Realistic? Can't you spot a problem?

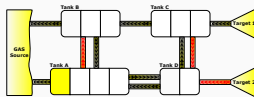


# Realistically, without control might be like...

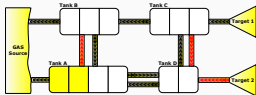
1) IDLE



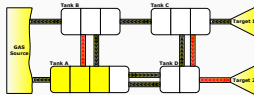
2) Source sends 1 gas unit to TankA



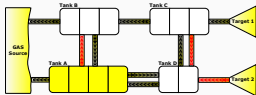
3) Source sends 1 gas unit to TankA



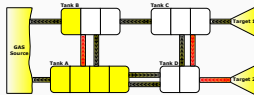
4) Source sends 1 gas unit to TankA



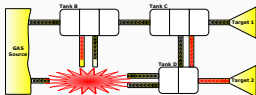
5) Source sends 1 gas unit to TankA



6) Source sends 1 gas unit to TankB

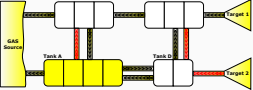
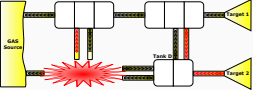
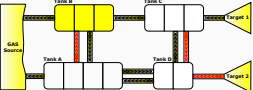
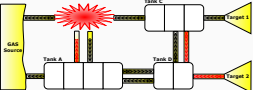
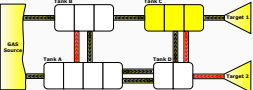
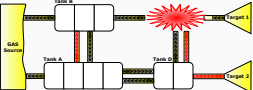
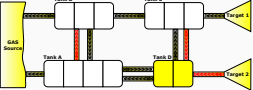
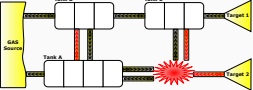


7) TankB sends 1 gas unit to TankA

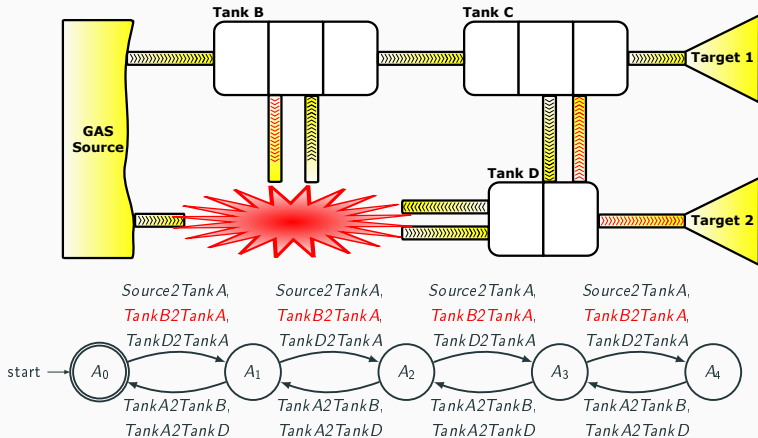


TankA blew up because it exceeded capacity. Note that *TankB2TankA* is uncontrollable.

# Problem: If a tank exceeds its capacity, it blows up

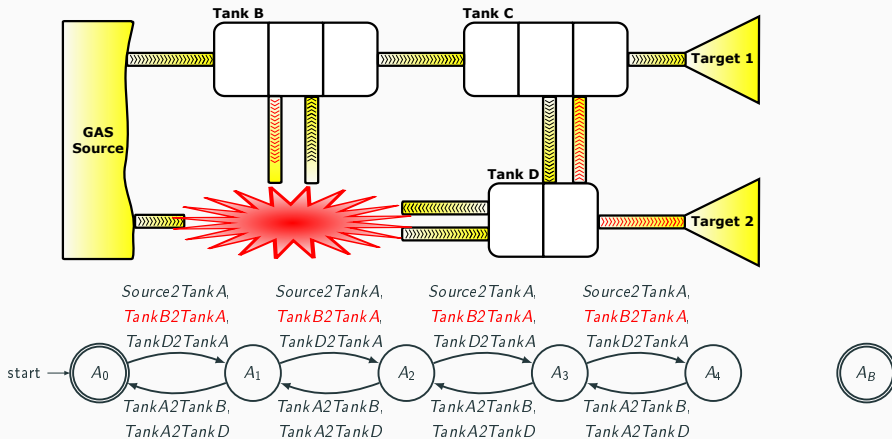
Critical state	Condition(s) leading to explosion
<p data-bbox="216 219 299 246"><b>TankA</b></p>  <p>The diagram shows a gas source on the left connected to four tanks: Tank A, Tank B, Tank C, and Tank D. Tank A is highlighted in yellow, indicating it is full. Tank B is connected to Tank A and Tank C. Tank C is connected to Tank B and Tank D. Tank D is connected to Tank C and has two outlets labeled Target 1 and Target 2.</p>	<p data-bbox="613 219 1155 246"><b>If Source/TankB/TankD supplies TankA...</b></p>  <p>The diagram shows the same setup as the critical state, but with a red starburst explosion at Tank A, indicating an explosion has occurred because it is being supplied from multiple sources while already being full.</p>
<p data-bbox="216 423 299 450"><b>TankB</b></p>  <p>The diagram shows the same setup, but Tank B is highlighted in yellow, indicating it is full. Tank A is connected to Tank B and Tank D. Tank D is connected to Tank B and Tank C. Tank C is connected to Tank D and has two outlets labeled Target 1 and Target 2.</p>	<p data-bbox="613 423 1059 450"><b>If Source/TankA supplies TankB...</b></p>  <p>The diagram shows the same setup, but with a red starburst explosion at Tank B, indicating an explosion has occurred because it is being supplied from multiple sources while already being full.</p>
<p data-bbox="216 627 299 654"><b>TankC</b></p>  <p>The diagram shows the same setup, but Tank C is highlighted in yellow, indicating it is full. Tank B is connected to Tank A and Tank C. Tank C is connected to Tank B and Tank D. Tank D is connected to Tank C and has two outlets labeled Target 1 and Target 2.</p>	<p data-bbox="613 627 1059 654"><b>If TankB/TankD supplies TankC...</b></p>  <p>The diagram shows the same setup, but with a red starburst explosion at Tank C, indicating an explosion has occurred because it is being supplied from multiple sources while already being full.</p>
<p data-bbox="216 831 299 858"><b>TankD</b></p>  <p>The diagram shows the same setup, but Tank D is highlighted in yellow, indicating it is full. Tank A is connected to Tank B and Tank D. Tank D is connected to Tank B and Tank C. Tank C is connected to Tank D and has two outlets labeled Target 1 and Target 2.</p>	<p data-bbox="613 831 1059 858"><b>If TankA/TankC supplies TankD...</b></p>  <p>The diagram shows the same setup, but with a red starburst explosion at Tank D, indicating an explosion has occurred because it is being supplied from multiple sources while already being full.</p>

# Tank A - Modeling the blow up situation



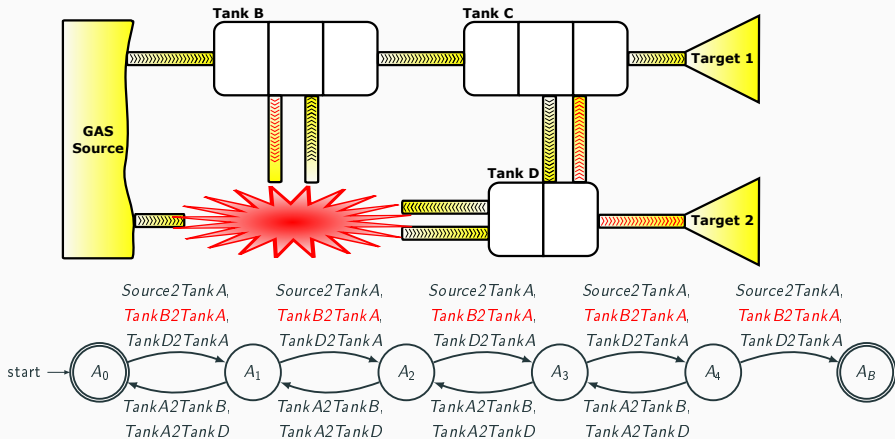
How can we model the fact that Tank A blows up?

# Tank A



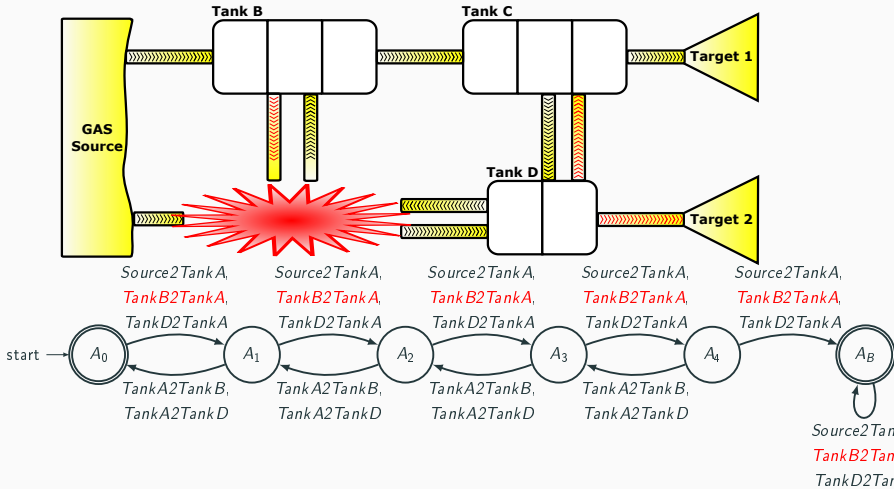
**A new marked state to model "TankA has blown up"**

# Tank A



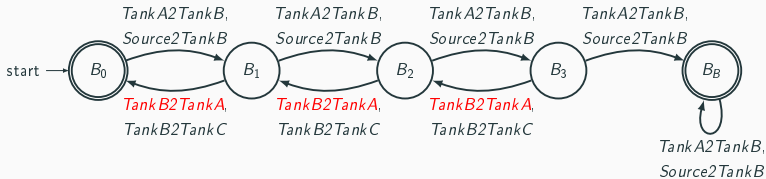
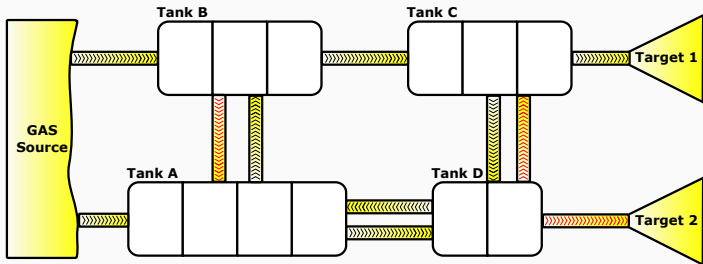
$A_B$  is reachable if TankA exceeds its capacity.

# Tank A

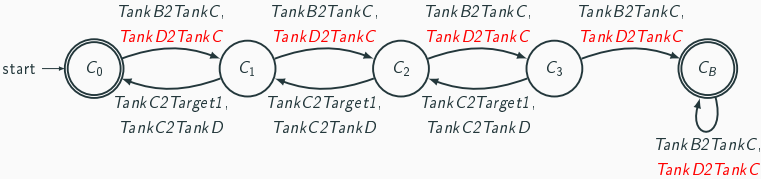
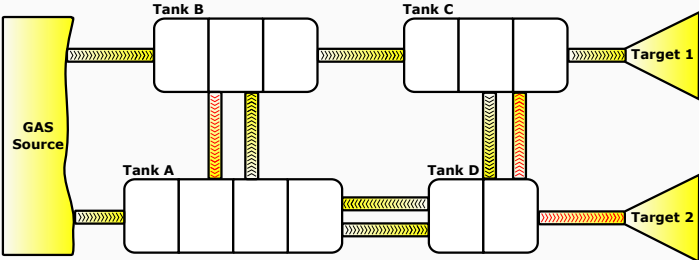


Once Tank A blows up, the pipes sending gas to it can still send gas. If this happens, the shipped units of gas are clearly lost.

# Tank B

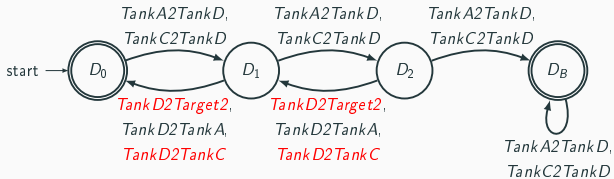
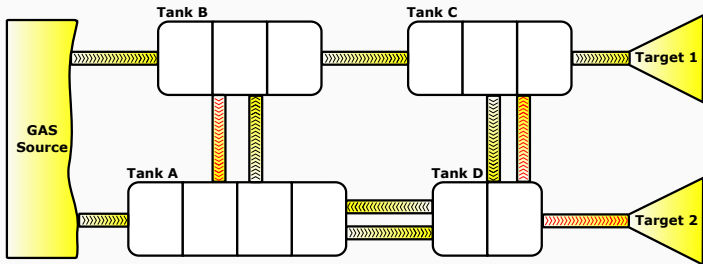


# Tank C

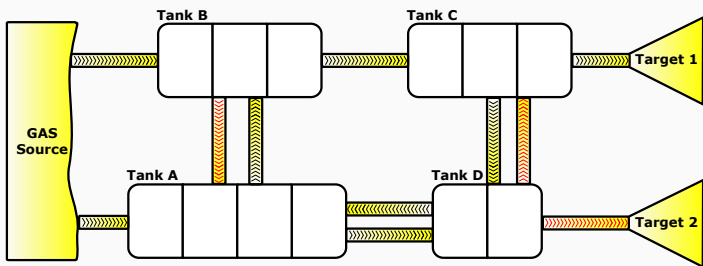




# Tank D

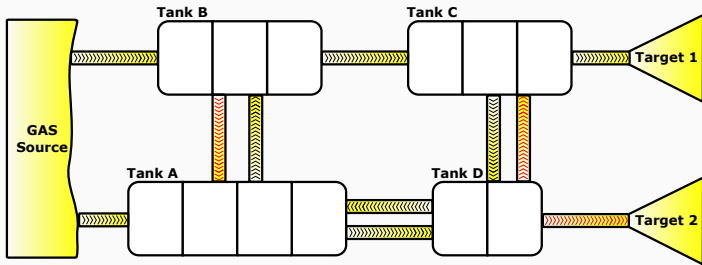


# Source and Targets



- States?
- Transitions?
- Event controllability?

# Source and Targets



Source

Target 1

Target 2

*Source2TankA, Source2TankB*



*TankC2Target1*



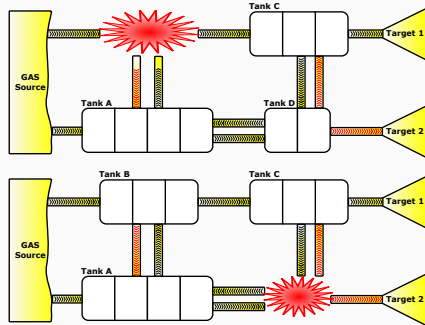
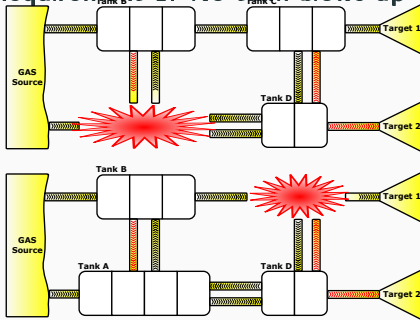
*TankD2Target2*



Not wrong, but quite useless.

# Requirement 1

## Requirement 1: No tank blows up



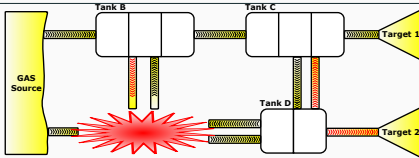
*We may start from the parallel composition of all tanks ... (don't!)*

Think: How many (separate) situations do we need to address?

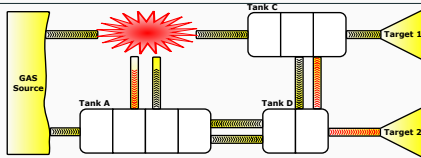
# Requirement 1 - Decomposition

**Requirement 1: No tank blows up**

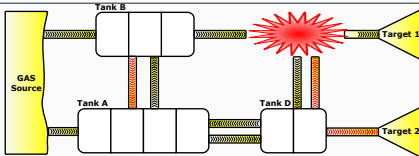
$R_{1A}$ : Tank A does not blow up



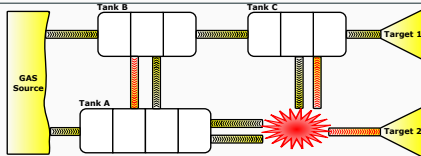
$R_{1B}$ : Tank B does not blow up



$R_{1C}$ : Tank C does not blow up



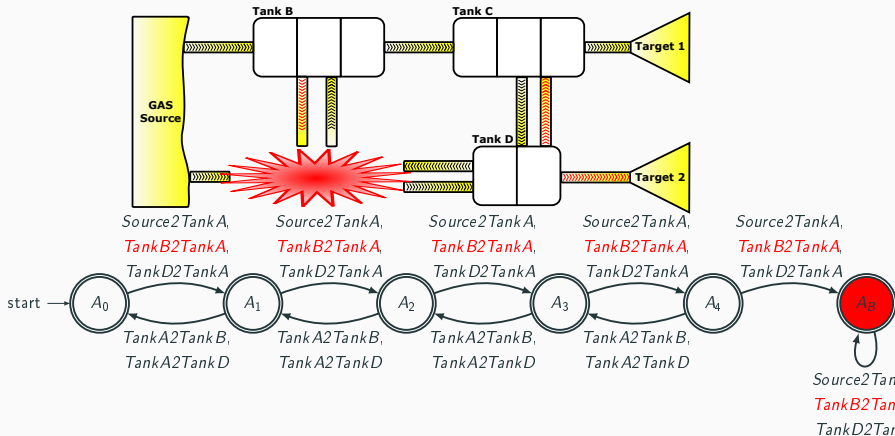
$R_{1D}$ : Tank D does not blow up





# Requirement 1A

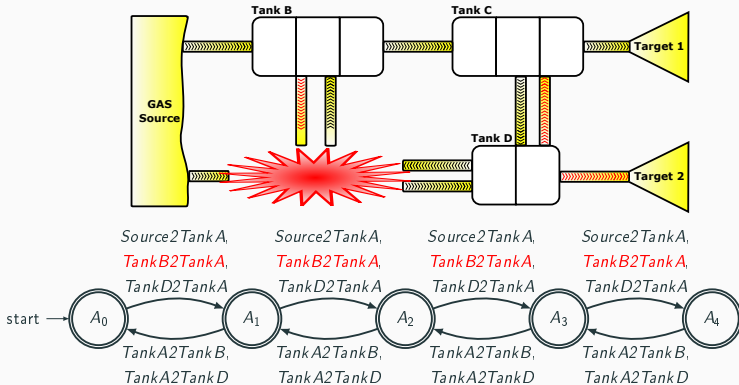
## Requirement 1A: TankA does not blow up



Step 2: Detect and remove the illegal state  $A_B$ .

# Requirement 1A

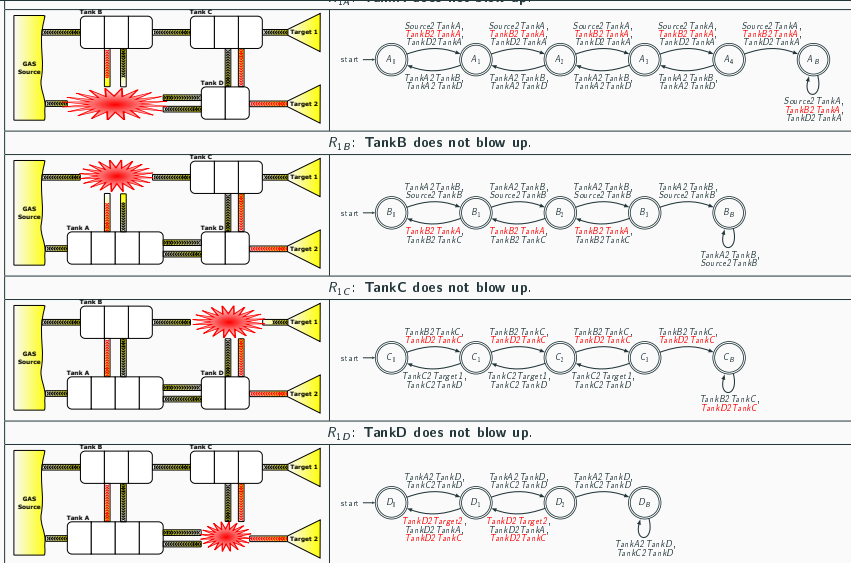
## Requirement 1A: TankA does not blow up



Final requirement.

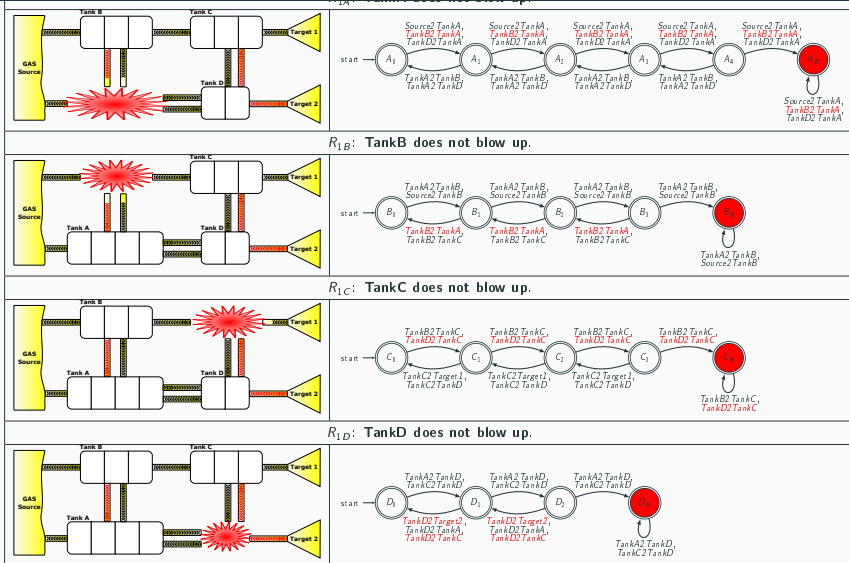


# Requirements 1A, 1B, 1C, and 1D - Formalization



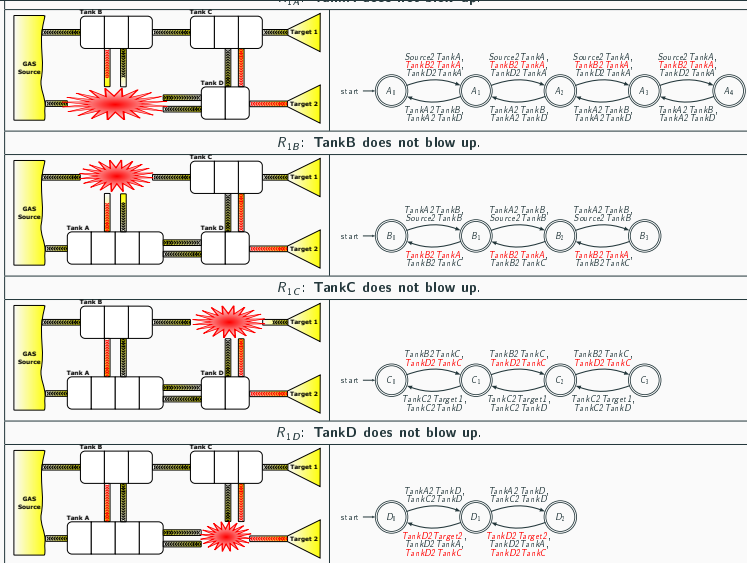
Step 1: Start with marked copies of TankA, TankB, TankC, and TankD

# Requirements 1A, 1B, 1C, and 1D - Formalization



Step 2: Identify and remove the illegal states  $A_B$ ,  $B_B$ ,  $C_B$ , and  $D_B$ .

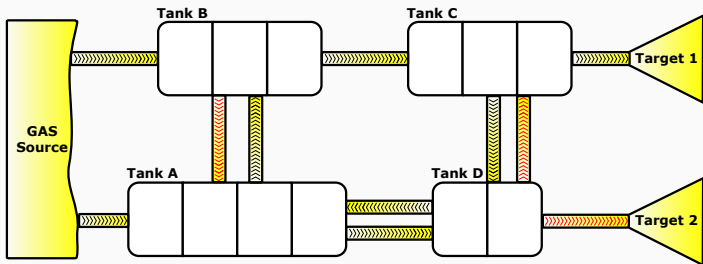
# Requirements 1A, 1B, 1C, and 1D - Formalization



Final sub-requirements.  $R_1 := R_{1A} \parallel R_{1B} \parallel R_{1C} \parallel R_{1D}$ .

## Requirement 2

Requirement 2: The plant must deliver at least 1 gas unit to each Target



How many automata?

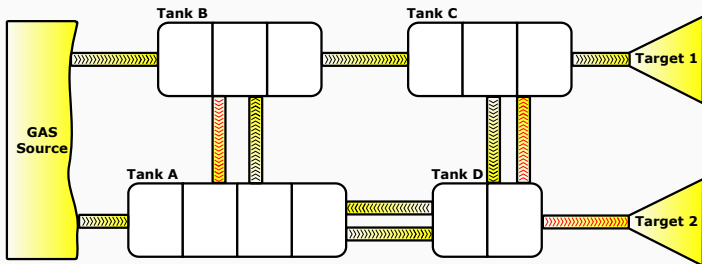
## Requirement 2 - Decomposition

Requirement 2: The plant must deliver at least 1 gas unit to each Target



$R_{2A}$  : The plant must deliver at least 1 gas unit to Target 1

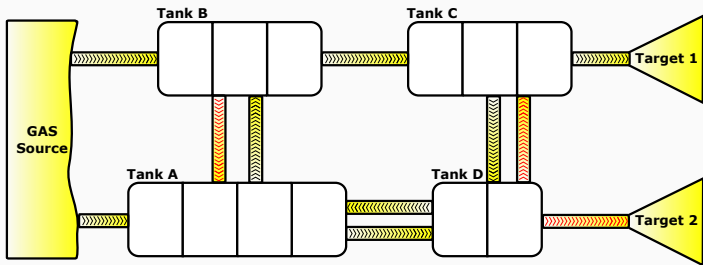
$R_{2B}$  : The plant must deliver at least 1 gas unit to Target 2



Hint: some task must be completed

## Requirement 2 - Target 1

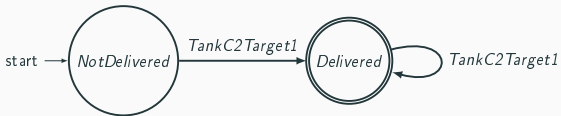
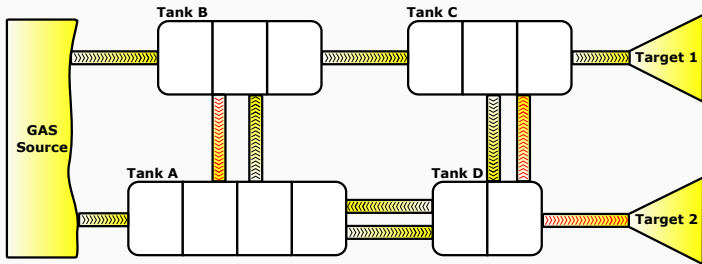
Requirement  $R_{2A}$ : The plant must deliver at least 1 gas unit to Target 1



- States?
- Marking?
- Transitions?
- Event controllability?

## Requirement 2 - Target 1

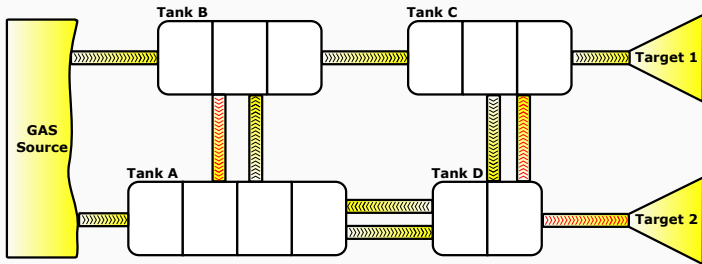
Requirement  $R_{2A}$ : The plant must deliver at least 1 gas unit to Target 1



Take a moment to reason on the marking.

## Requirement 2 - Target 2

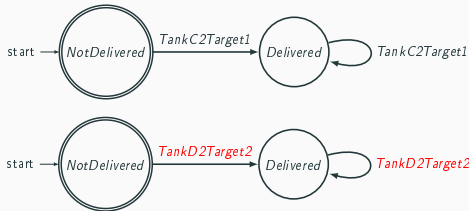
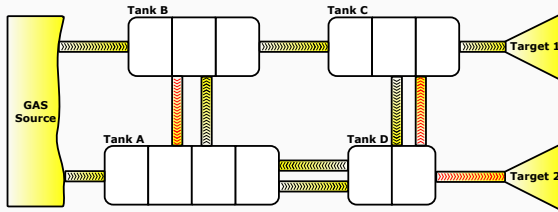
Requirement  $R_{2B}$ : The plant must deliver at least 1 gas unit to Target 2



So,  $R_2 := R_{2A} || R_{2B}$



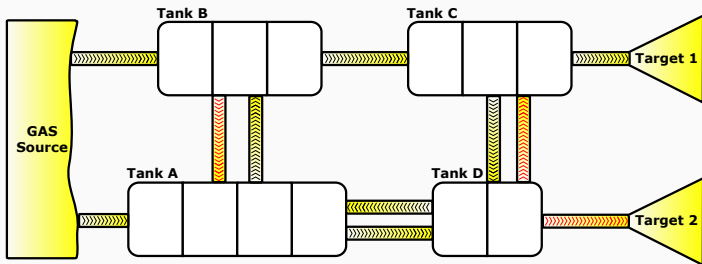
# What if we flip the marking?



What do these requirements say? Consider them instead of the previous  $R_{2A}$  and  $R_{2B}$  (initially one at a time and then together). What effect do they have on the synthesized supervisor?

## Requirement 3

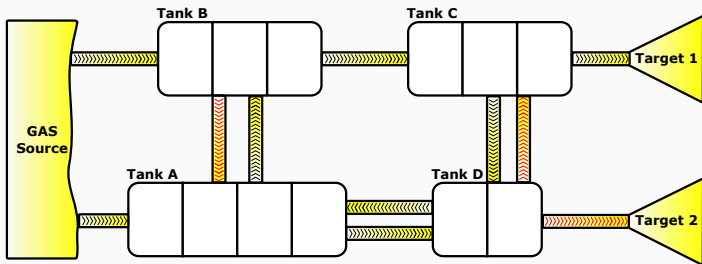
Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate



Straightforward but there is something to pay attention to... what?

## Requirement 3

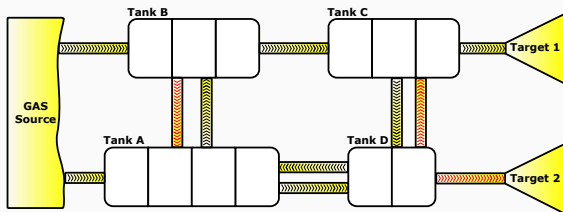
Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate



Which target will be supplied first? That's up to you.. but the requirement doesn't actually impose such a condition...So?

## Requirement 3 - Attempt 1 - Non-deterministic

Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate



$R_{3A}$ : If Target 1 is supplied first...

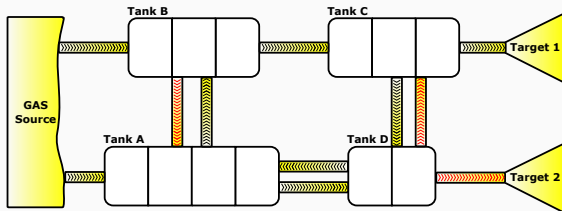
???

$R_{3B}$ : If Target 2 is supplied first...

???

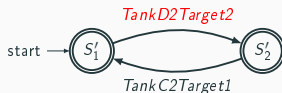
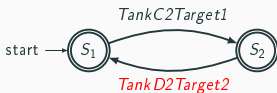
# Requirement 3 - Attempt 1 - Non-deterministic

Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate



$R_{3A}$ : If Target 1 is supplied first...

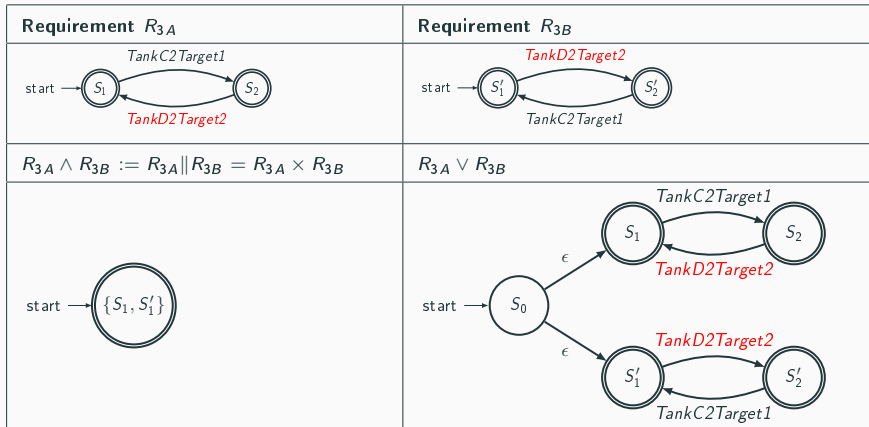
$R_{3B}$ : If Target 2 is supplied first...



Not certainly an AND of the two automata.  
We need the UNION of these two automata!

# Requirement 3 - Attempt 1 - Nondeterministic

**Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate**

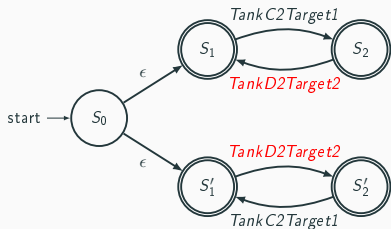


**Homework: synthesize a supervisor that (also) takes into consideration requirement  $R_{3A} \wedge R_{3B}$ . What effect does it have on the plant?**

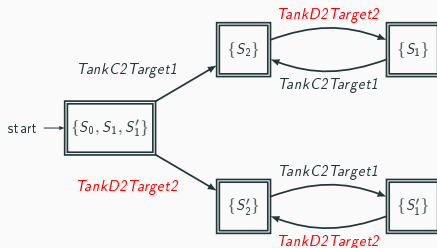
# Requirement 3 - Attempt 1 - Nondeterministic

Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate

NFA



DFA



## Requirement 3 - Attempt 2 - Deterministic

**Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate**



**3A) If Target 1 is supplied, then Target 2 must be supplied at least once before Target 1 is supplied again.**

**3B) If Target 2 is supplied, then Target 1 must be supplied at least once before Target 2 is supplied again.**

...

...

**If Target  $i = 1, 2$  is supplied, then Target  $(i \bmod 2) + 1$  must be supplied at least once before Target  $i$  is supplied again.**



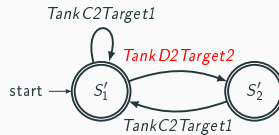
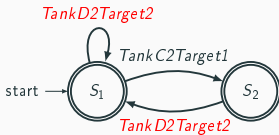
# Requirement 3 - Attempt 2 - Deterministic

**Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate**



**3A) If Target 1 is supplied, then Target 2 must be supplied at least once before Target 1 is supplied again.**

**3B) If Target 2 is supplied, then Target 1 must be supplied at least once before Target 2 is supplied again.**



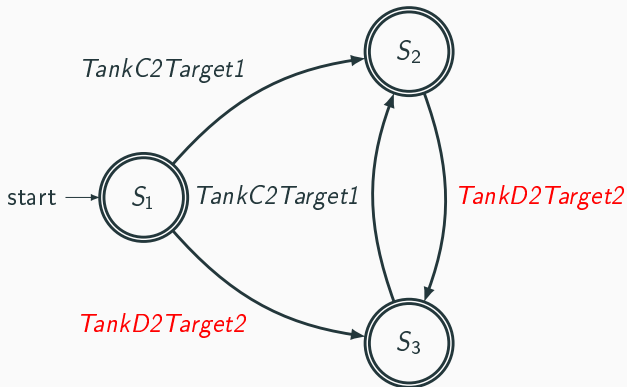
**If Target  $i = 1, 2$  is supplied, then Target  $(i \bmod 2) + 1$  must be supplied at least once before Target  $i$  is supplied again.**

**Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate**

**1 automaton only? (3 states)**

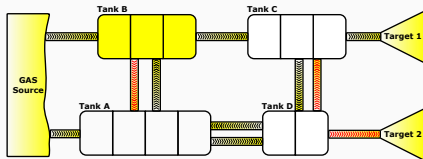
## Requirement 3 - Attempt 3 - Deterministic

Requirement 3: Gas deliveries to Target 1 and Target 2 must always alternate



# Requirement 4

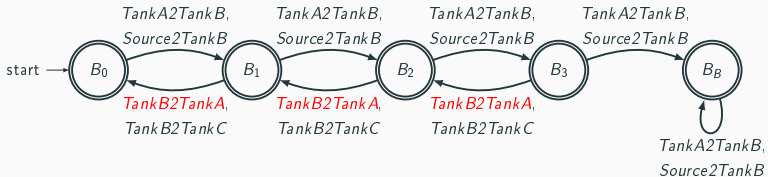
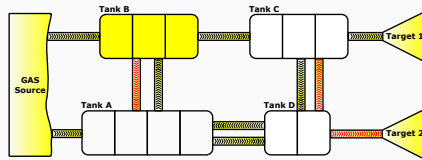
Requirement 4: TankB can supply TankC only if TankB is full



- States?
- Transitions?
- Event controllability?

# Requirement 4

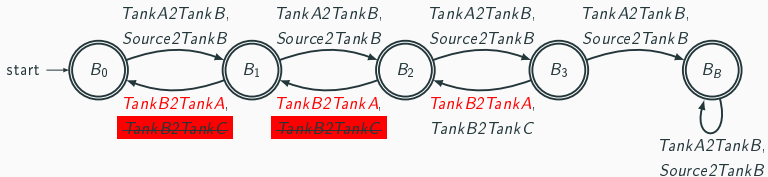
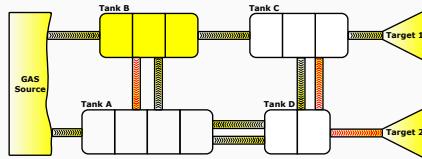
Requirement 4: TankB can supply TankC only if TankB is full



Step 1: Start from a copy of the automaton for Tank B.

# Requirement 4

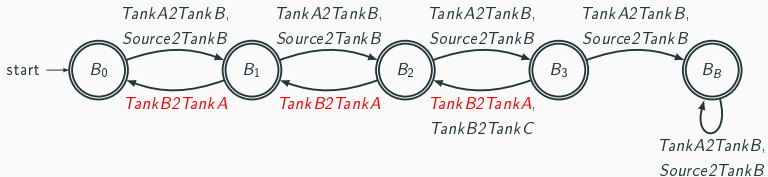
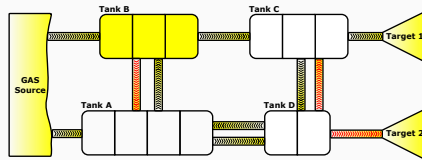
Requirement 4: TankB can supply TankC only if TankB is full



Step 2: Detect and remove the illegal transitions.

# Requirement 4

Requirement 4: TankB can supply TankC only if TankB is full



Final requirement.