

# Systems Design Laboratory

Teaser

---

**Matteo Zavatteri**

Department of Computer Science, University of Verona, ITALY



Systems Design Laboratory is  
not another theory course!



# Essential Information: Homepage and Academic Staff



# Essential Information: Homepage

The screenshot shows the homepage of the University of Verona for the Master's degree in Computer Engineering for Robotics and Smart Industry. The navigation menu includes Home, The programme, Studying at the University of Verona (selected), How to enrol, Services and opportunities, Dashboard, and Contacts. The main heading is 'Studying at the University of Verona', with a sub-heading 'Systems design laboratory (2021/2022)'. Below the heading, there are two buttons: 'Ritorna al piano didattico' and 'Ritorna agli insegnamenti per periodo'. The course details are displayed in a table-like format with two columns. The left column contains 'Teaching code' (4S009016), 'Coordinator' (Tiziano Villa), and 'Scientific Disciplinary Sector (SSD)' (INF/01 - INFORMATICS). The right column contains 'Academic staff' (Tiziano Villa, Luca Geretti, Matteo Zavatteri), 'Credits' (6), and 'Language' (English). There are also two buttons: 'Lessons timetable' and 'Seminars'. The page also includes a sidebar with 'Academic calendar', 'Teaching staff', 'Modules' (selected), 'Additional learning activities', and 'Career prospects'. At the bottom right, there is a partial URL starting with 'http'.

UNIVERSITÀ di VERONA

Master's degree in Computer Engineering for Robotics and Smart Industry  
Master's degree

Timetable Search

Home The programme ▾ Studying at the University of Verona ▾ How to enrol ▾ Services and opportunities Dashboard Contacts

## Studying at the University of Verona

Here you can find information on the organisational aspects of the Programme, lecture timetables, learning activities and useful contact details for your time at the University, from enrolment to graduation.

[Print / Share](#)

[A.A. 2021/2022](#)

Academic calendar

Teaching staff

**Modules**

Additional learning activities

Career prospects

← Ritorna al piano didattico

Ritorna agli insegnamenti per periodo →

### Systems design laboratory (2021/2022)

Teaching code 4S009016	Academic staff <a href="#">Tiziano Villa</a> , <a href="#">Luca Geretti</a> , <a href="#">Matteo Zavatteri</a>	<a href="#">Lessons timetable</a>
Coordinator <a href="#">Tiziano Villa</a>	Credits 6	<a href="#">Seminars</a>
Scientific Disciplinary Sector (SSD) INF/01 - INFORMATICS	Language English	

http

//www.corsi.univr.it/?ent=cs&aa=2021%2F2022&codiceCs=S81&codins=  
4S009016&discr=&discrCd=&id=954&menu=Studiare&tab=Insegnamenti&lang=en

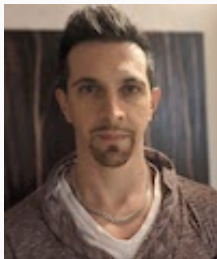
Tiziano Villa (coordinator)



- Formal models for discrete-event systems
- Boolean functions and networks
- Decision diagrams
- Multiple-valued and temporal logics
- Finite and  $\omega$ -automata

<https://www.di.univr.it/?ent=persona&id=3849&lang=en>

**Luca Geretti**



- Modeling with Hybrid Automata
- Static verification
- Dynamic verification

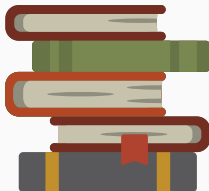
<https://www.di.univr.it/?ent=persona&id=6462&lang=en>

Matteo Zavatteri



- Modeling with Finite State Automata
- Supervisory Control
- ESCET software

## Essential Information: Educational Material



- Books and papers (some already suggested in previous classes)
- Lecture notes and classroom teaching material
- Reports on the analysis of case studies
- Online documentation and tutorials
- Example code

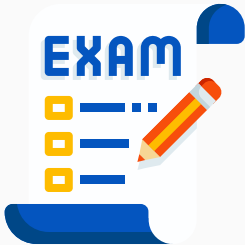


## Essential Information: Tentative Setup



**Fully Controllable Setting:** We are probably going to meet in a lab but you will do everything on your own computer.  
(Fewer problems, no UniVR/IT dependencies).

# Essential Information: Exam



**Mainly a project**

(possibly a short written test depending on the number of students)

### Systems Design Laboratory:



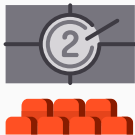
- is a “**hands-on**” modeling course
- provides you with **concrete skills**
- involves the use of several **software libraries**
- is about **solving concrete problems**



**Last but not least:** plenty of room for **theses**, research, and more (e.g., the ICE lab context)

<https://www.icelab.di.univr.it/?lang=en>

# Today's Teaser: Supervisory Control Part



Today you are going to see an example of  
concrete supervisory control application

- Modeling plants/specifications through **Finite State Automata**
- Synthesizing **controllers** for it with **ESCET**
- **Simulating** the controlled plants



# Eclipse Supervisory Control Engineering Toolkit (ESCET)

The screenshot shows the Eclipse ESCET website. At the top, there is a navigation bar with links for Home, About, Download, Documentation, Development, and Contact/Support. The main heading is "Eclipse ESCET™" with a sub-heading: "The Eclipse ESCET project provides a model-based approach and toolkit for the development of supervisory controllers." Below this is a "Learn more" button. The section "Languages and tools" features three items: CIF, Chi, and ToolDef, each with a brief description and a "Learn more" button.

Eclipse ESCET™ / Project ▾ Home About Download Documentation Development Contact/Support

Version: v0.4

## Eclipse ESCET™

The Eclipse ESCET project provides a model-based approach and toolkit for the development of supervisory controllers.

[Learn more](#)

### Languages and tools

- CIF**  
CIF is a modeling language and extensive toolset supporting the entire development process of supervisory controllers.  
[Learn more](#)
- Chi**  
Chi is a modeling language and toolset to analyze the performance of supervisory controllers.  
[Learn more](#)
- ToolDef**  
ToolDef is a cross-platform and machine-independent scripting language to automate CIF and Chi tools.  
[Learn more](#)

- **Modeling** of hybrid systems
- **Graphical user interface**
- **Simulation**
- **Controller synthesis** for (Extended) Finite State Automata
- **PLC code** generation
- Used in many **real-word case studies**

# Eclipse Supervisory Control Engineering Toolkit (ESCET)

4TC00 Home

Search docs

COURSE INFO

- Study guide
- Course schedule
- Downloads
- FAQ

MODULES

- 1: Tooling, Basics, Groups
- 2: Types 1, Data
- 3: Reuse 1, Time, SVS
- 4: Twin tooling, MSSE
- 5: Channels, Types Reuse 2
- 6: Functions, Stochastics
- 7: GII basics, GII teams
- 8: GII branches, GII advanced

EXAMINATIONS

- Intermediate exam
- Midterm assignment
- Final assignment
- Peer review
- Fraud

FESTO WORKSTATIONS

- Festo hardware manual
- Festo lab and videos
- Digital twins
- Twi-CAT setup
- Twi-CAT control of digital twin
- Twi-CAT control of Festo WS
- Twi-CAT tips and tricks

4TC00 Model-based systems engineering

## 4TC00 Model-based systems engineering

Watch on

4TC00 Model-based systems engineering is a third year bachelor course given by the *Control Systems Technology Group*<sup>TM</sup> of the Mechanical Engineering department at the *Eindhoven University of Technology*<sup>TM</sup>. The course treats the development process of supervisory controllers, from specification, simulation-based validation and visualization, to code generation and real-time testing on *3D digital twins* and on real Festo workstations. The video at the top of the page gives a short, 2.5 minute overview of the course.

This website is freely accessible to anyone interested in model-based systems engineering. A small part of the course, including the quizzes and group creation, is available only via *Canvas*<sup>TM</sup>. This part is accessible only to students from the Eindhoven University of Technology who are officially taking this course.

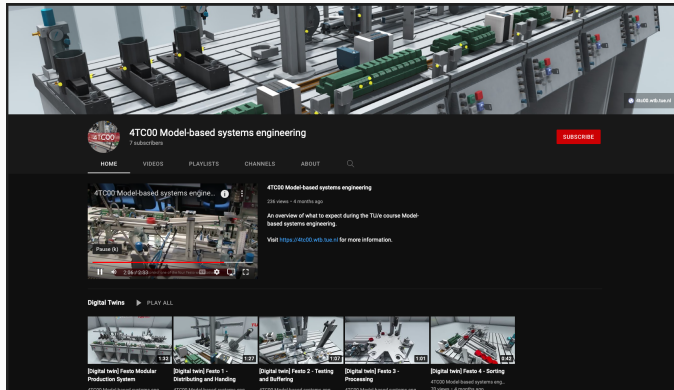
To ensure that you always see the latest version of each web page, it may be necessary to *refresh the page*<sup>TM</sup>, or to *clear the cache*<sup>TM</sup> (which refreshes all your webpages). When in doubt, you can find the date of each page at the bottom.

[STUDY GUIDE](#) → [SCHEDULE](#) → [MODULES](#) →

Introduced in the course 4TC00 Model-Based Systems Engineering  
(bachelor degree, 3rd year) Eindhoven University of Technology (TU/e)

<https://cstweb.wtb.tue.nl/4tc00/index.html>

# Eclipse Supervisory Control Engineering Toolkit (ESCET)

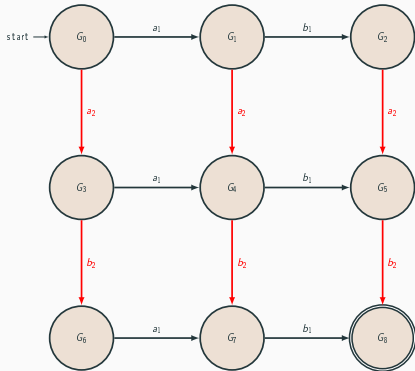


The image shows a screenshot of a YouTube channel page for '4TC00 Model-based systems engineering'. The channel has 7 subscribers and a 'SUBSCRIBE' button. The main video is titled '4TC00 Model-based systems engine...' and has 226 views, posted 4 months ago. The video description states: 'An overview of what to expect during the TU/e course Model-based systems engineering. Visit <https://4tc00.web.tue.nl/> for more information.' Below the main video, there is a 'Digital Twins' playlist with a 'PLAY ALL' button. The playlist contains five videos: 'Digital twin| Festo Modular Production System' (1:32), 'Digital twin| Festo 1 - Distributing and Handling' (1:37), 'Digital twin| Festo 2 - Testing and Buffering' (1:07), 'Digital twin| Festo 3 - Processing' (1:01), and 'Digital twin| Festo 4 - Sorting' (2:43).

Check out the youtube channel for videos, examples, and more

[https://www.youtube.com/channel/UC1lkrIkRkgtbYDul9BwI\\_Bw](https://www.youtube.com/channel/UC1lkrIkRkgtbYDul9BwI_Bw)

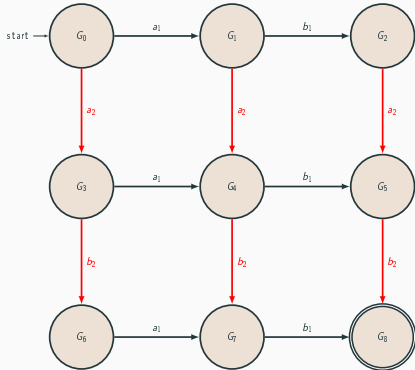
# The Database Concurrency Example



- Events  $a_1, b_1$  are controllable
- Events  $a_2, b_2$  are uncontrollable
- $G_0$  is the initial state
- $G_8$  is the marked state



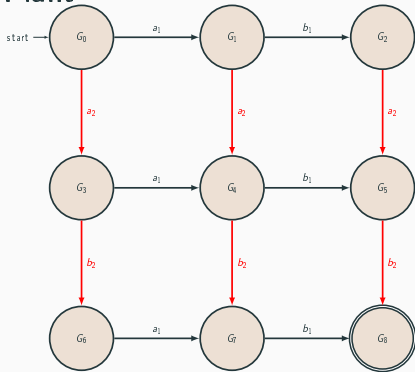
# The Database Concurrency Example: Requirement



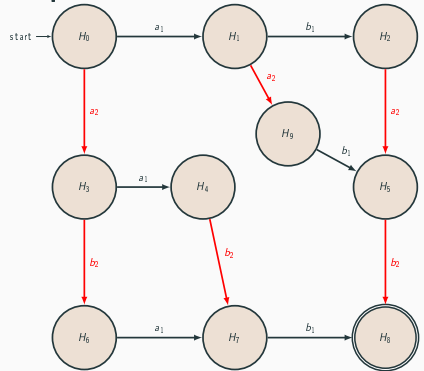
**Requirement:**  $a_1$  precedes  $b_1$  if and only if  $a_2$  precedes  $b_2$

# The Database Concurrency Example: Requirement

Plant

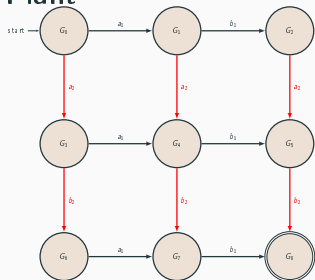


Requirement

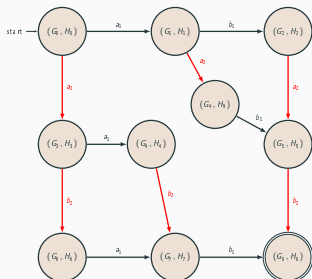
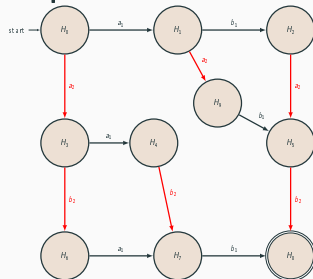


# The Database Concurrency Example: Controller Synthesis

## Plant



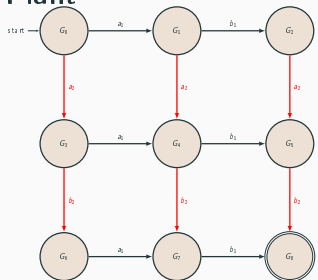
## Requirement



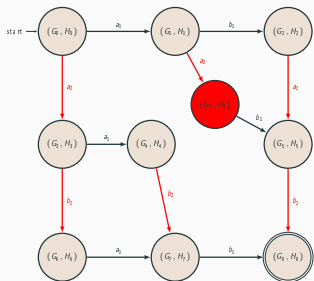
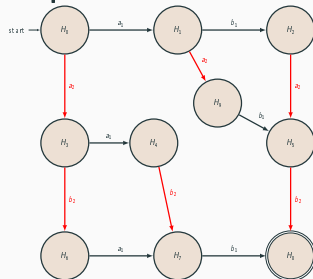
- $G \parallel H$

# The Database Concurrency Example: Controller Synthesis

## Plant



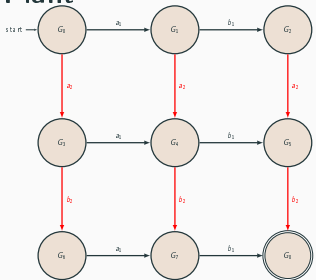
## Requirement



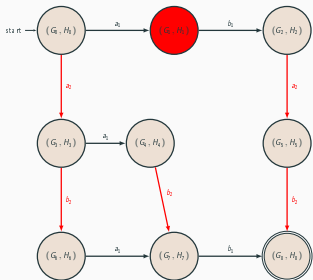
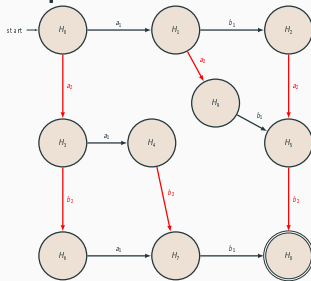
- $(G_4, H_9)$  is uncontrollable

# The Database Concurrency Example: Controller Synthesis

## Plant



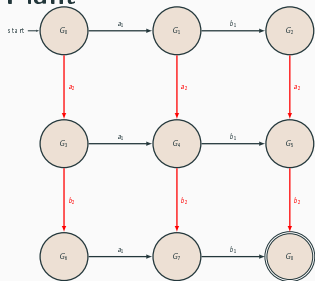
## Requirement



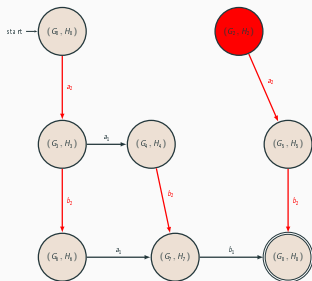
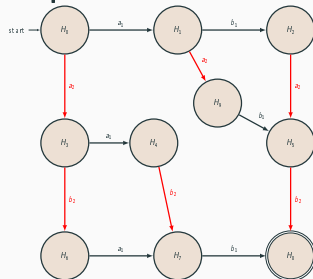
- $(G_1, H_1)$  is uncontrollable

# The Database Concurrency Example: Controller Synthesis

## Plant



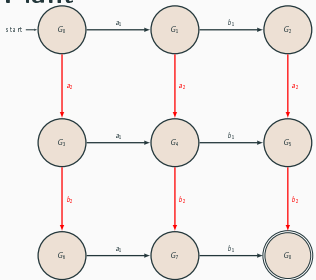
## Requirement



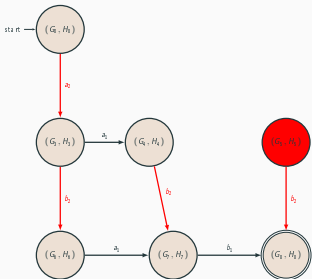
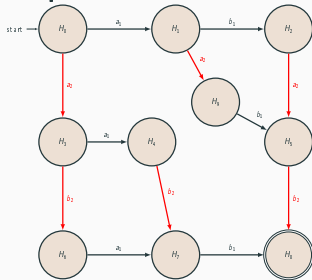
- $(G_2, H_2)$  is not accessible

# The Database Concurrency Example: Controller Synthesis

## Plant



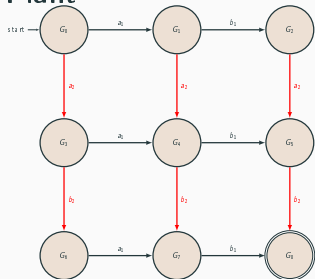
## Requirement



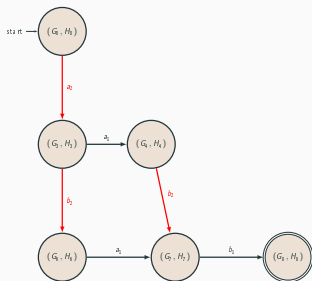
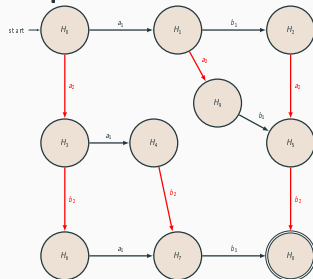
- $(G_5, H_5)$  is not accessible

# The Database Concurrency Example: Controller Synthesis

## Plant



## Requirement



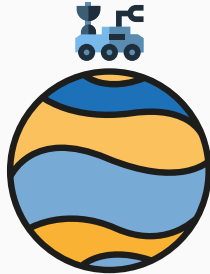
## Control Policy:

- At the beginning  $S$  disables  $a_1$
- When the **plant**  $G$  is in state  $G_4$ ,  $S$  disables  $b_1$ .



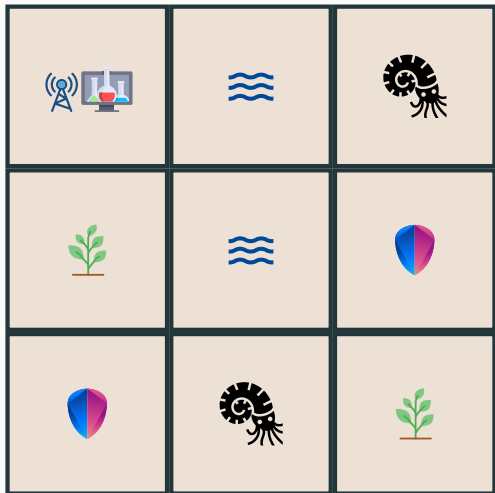
# A Sample-Collecting Problem







We sent a rover to some planet.



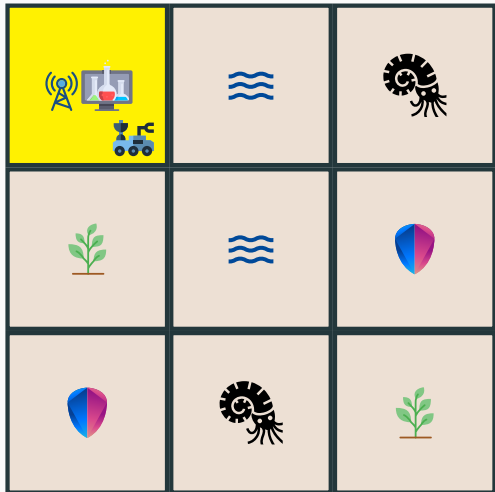
Our purpose is to use it for sample-collecting in the area where it landed.

# The Area



-  (transmission point)
-  (analysis point)
-  (liquid point)
-  (mineral point)
-  (flora point)
-  (fossil point)

## Details on Rover Free Behavior



Possible actions:

- analyze the collected samples
- data transmission
- move right
- move down

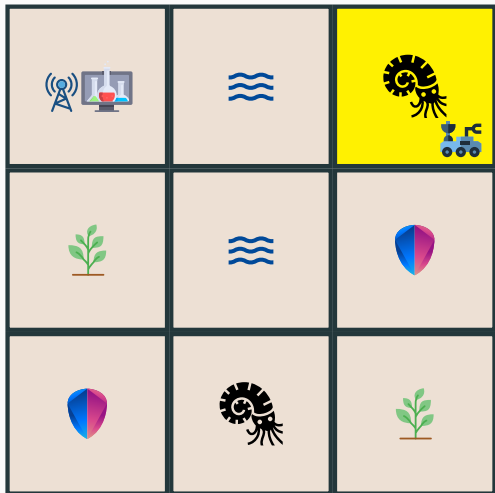
## Details on Rover Free Behavior



Possible actions:

- collect a liquid sample
- move left
- move right
- move down

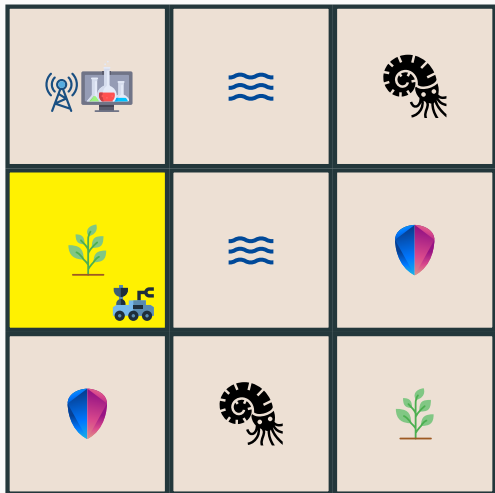
## Details on Rover Free Behavior



Possible actions:

- collect a fossil sample
- move left
- move down

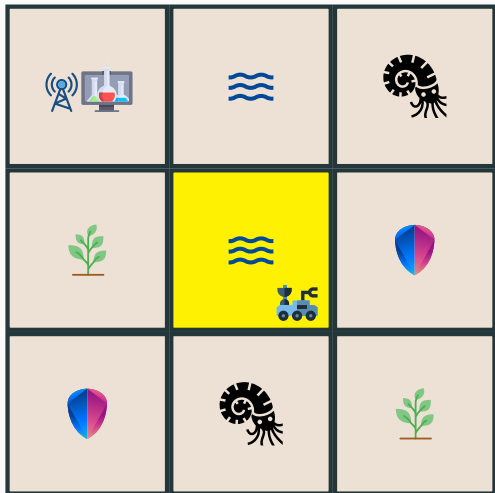
## Details on Rover Free Behavior



Possible actions:

- collect a flora sample
- move right
- move up
- move down

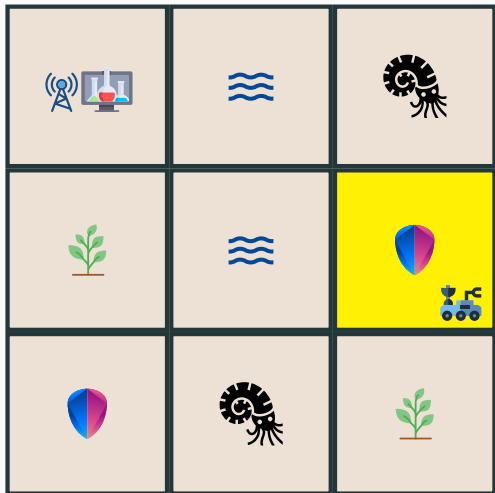
## Details on Rover Free Behavior



Possible actions:

- collect a liquid sample
- move left
- move right
- move down
- move up

## Details on Rover Free Behavior

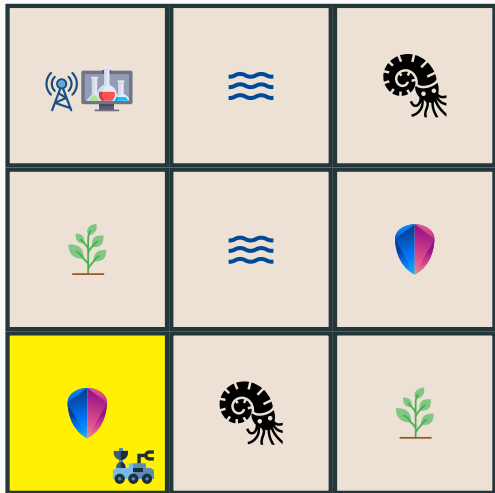


Possible actions:

- collect a mineral sample
- move up
- move left
- move down



## Details on Rover Free Behavior: Fossil points



Possible actions:

- collect a mineral sample
- move up
- move right

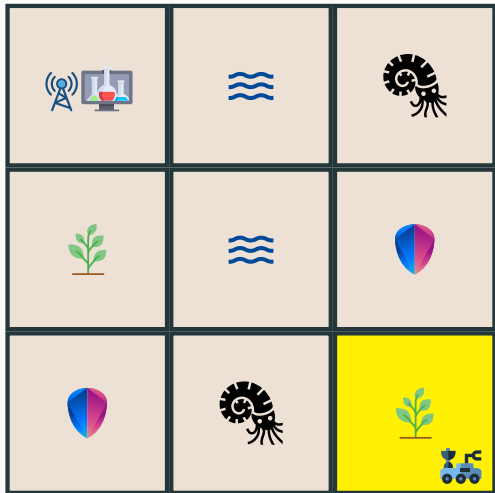
## Details on Rover Free Behavior: Fossil points



Possible actions:

- collect a fossil sample
- move up
- move left
- move right

## Details on Rover Free Behavior: Fossil points



Possible actions:

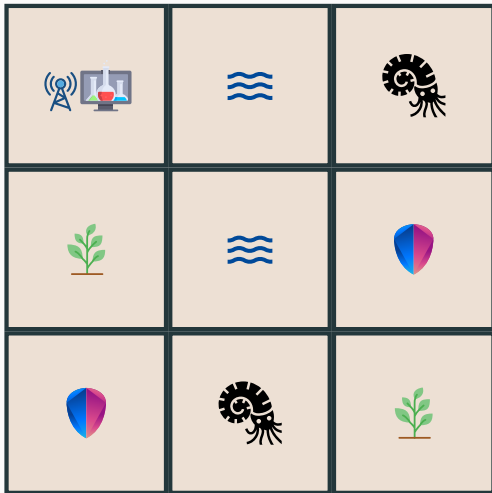
- collect a flora sample
- move up
- move left




What about states?



# Formalizing Plant States



## Formalizing Plant States

(1,1) 	(1,2) 	(1,3) 
(2,1) 	(2,2) 	(2,3) 
(3,1) 	(3,2) 	(3,3) 

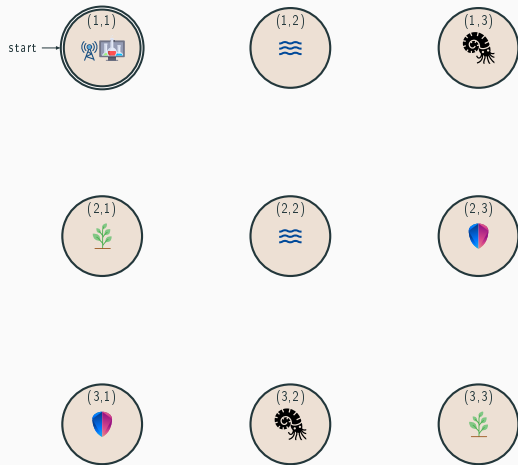
So, it makes sense to introduce a notion of **grid** on which the rover moves.

# Formalizing Plant States



Does it ring a bell?

# Formalizing Plant States





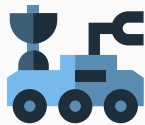
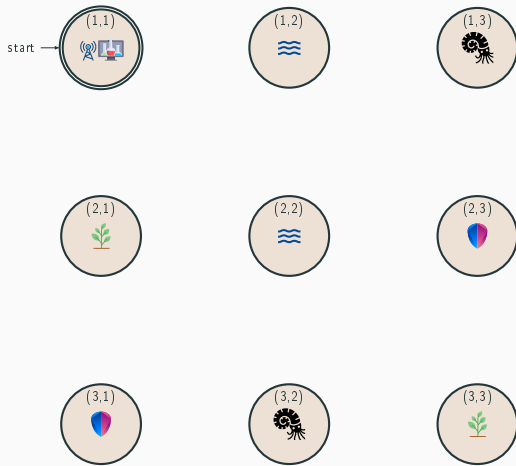
# Formalizing Plant Transitions



What about transitions?

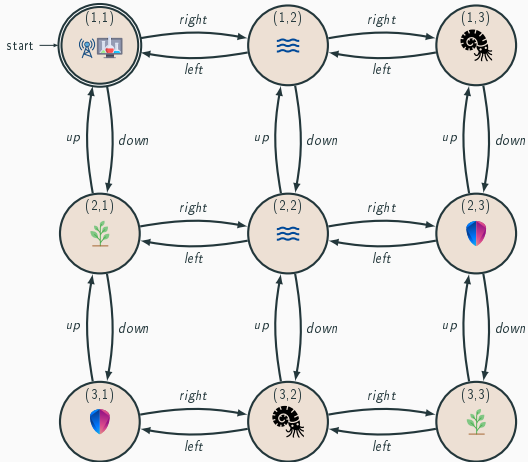


# Formalizing Plant Transitions



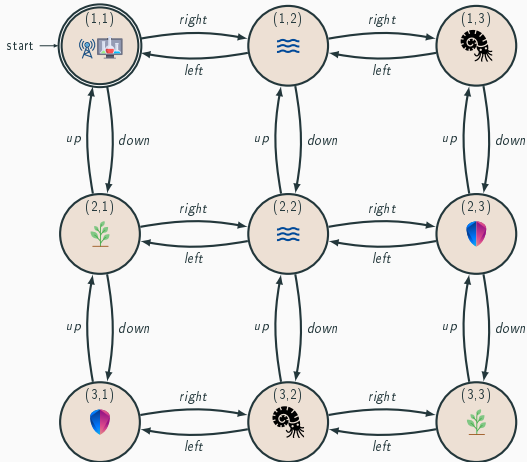
Movements?

# Formalizing Plant Transitions



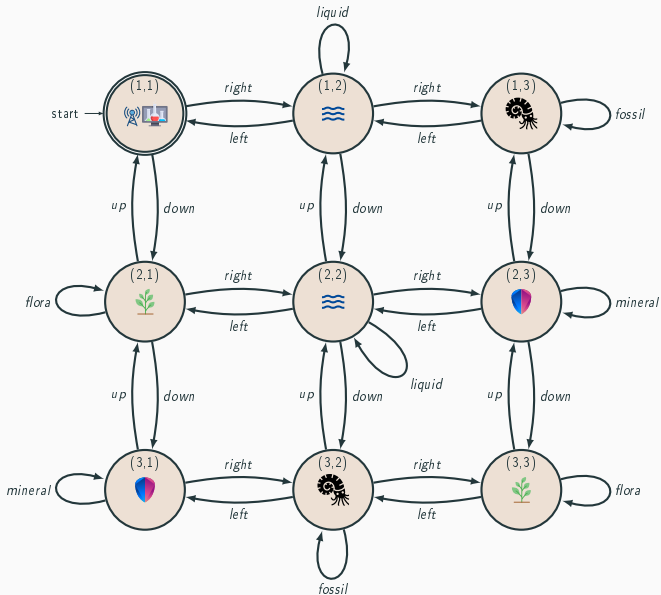
Movements

# Formalizing Plant Transitions



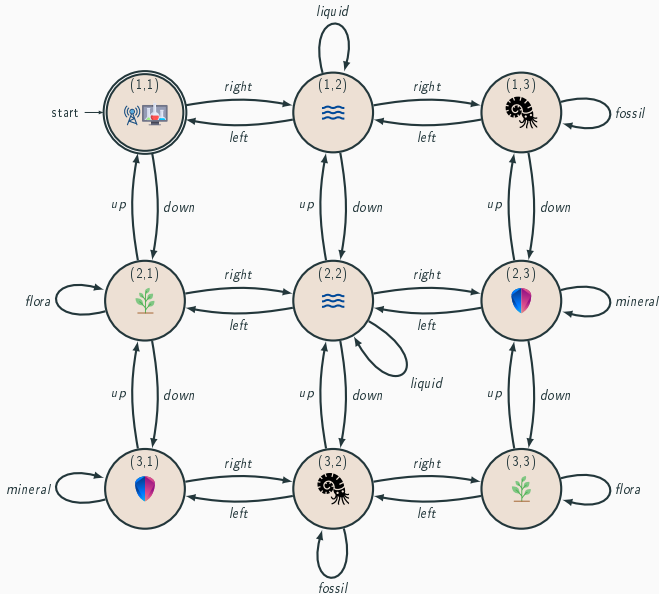
Sample collection?

# Formalizing Plant Transitions



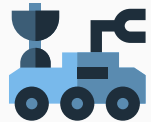
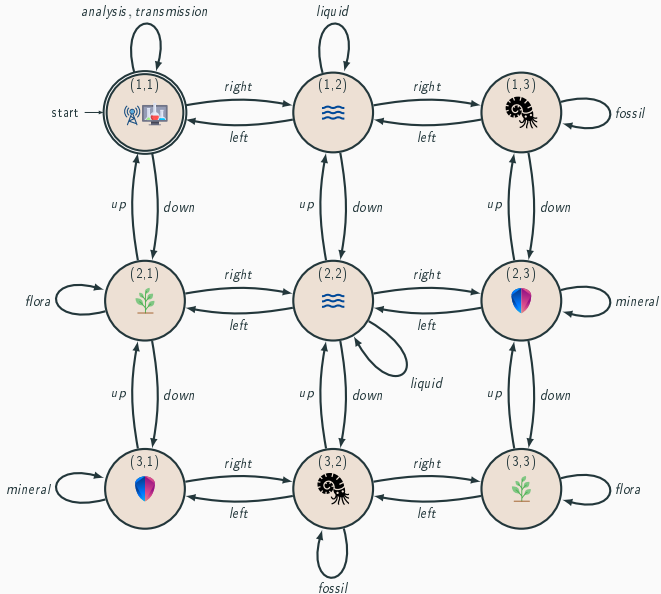
Sample collection

# Formalizing Plant Transitions



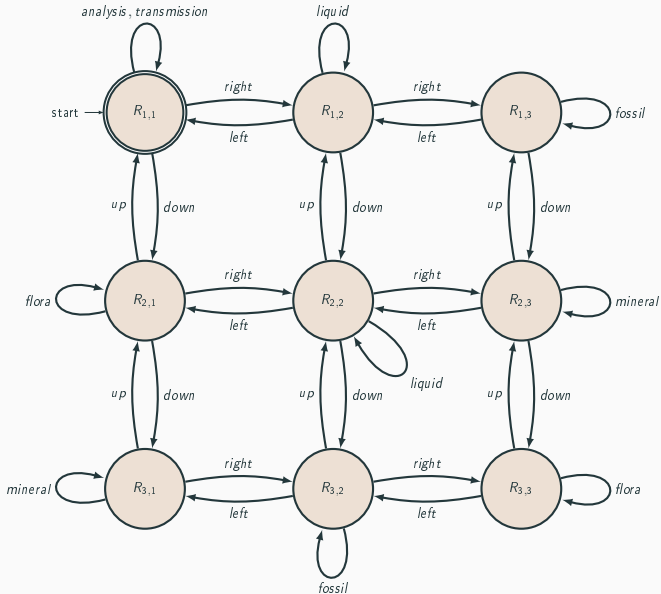
Analysis and  
Transmission?

# Formalizing Plant Transitions



Analysis and  
Transmission

# Finalizing Plant Formalization: State Names



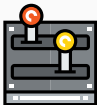
Uncontrolled Rover  
Behavior



# Formalizing Requirements

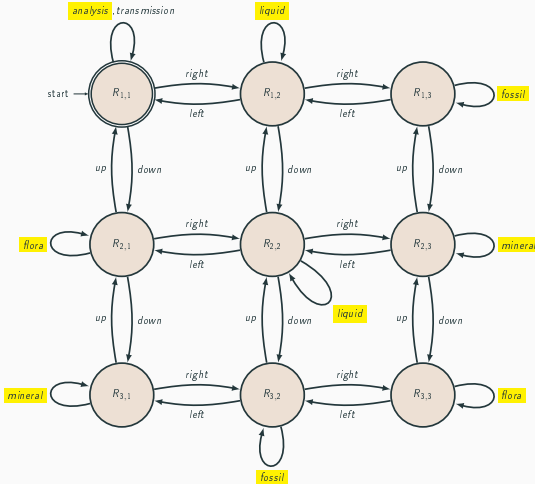


What about supervisory control?



# Requirement 1

## Plant



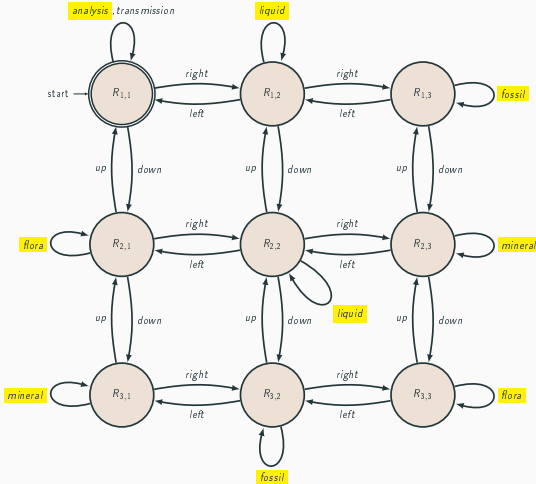
## Requirement

- States?
- Events?
- Marking?

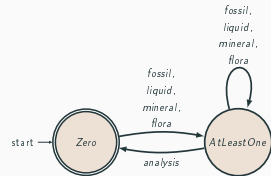
At least 1 sample must be collected before analysis

# Requirement 1

## Plant



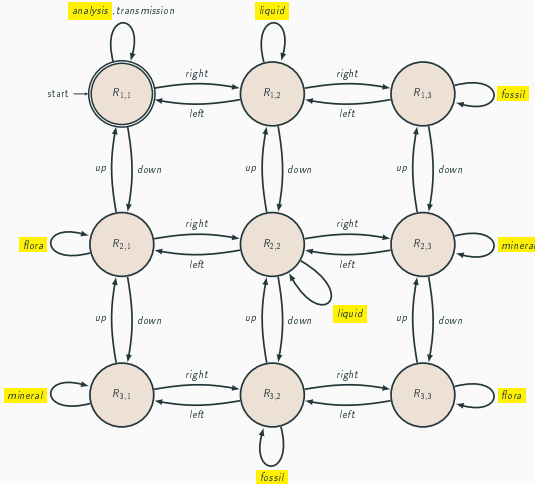
## Requirement



At least 1 sample must be collected before analysis

# Requirement 2

## Plant



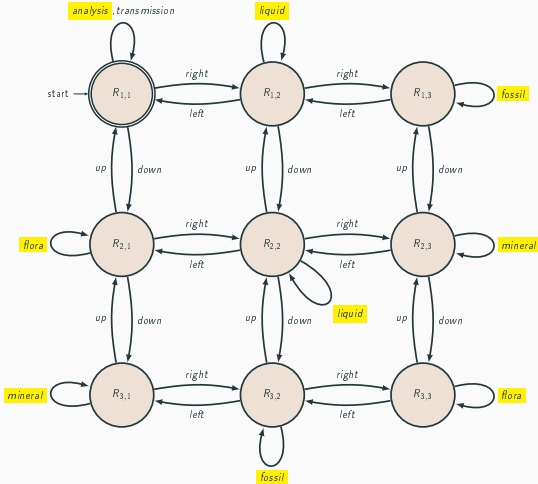
## Requirement

- States?
- Events?
- Marking?

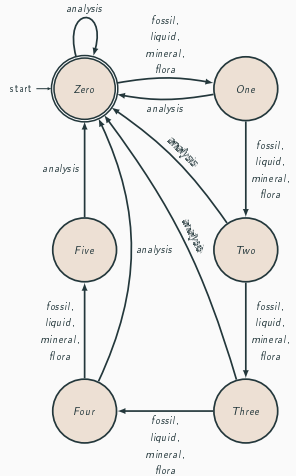
At most 5 samples must be collected before analysis

# Requirement 2

## Plant



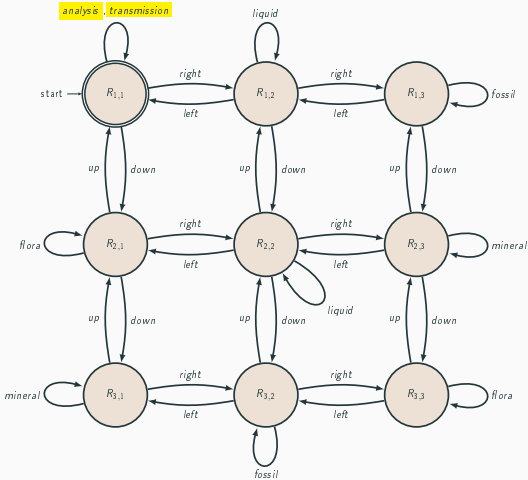
## Requirement



At most 5 samples must be collected before analysis

# Requirement 3

## Plant



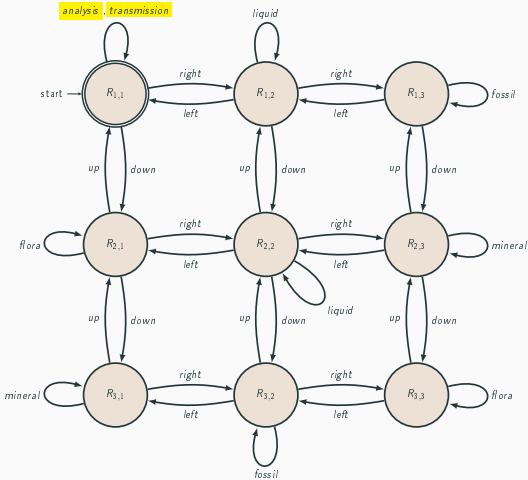
## Requirement

- States?
- Events?
- Marking?

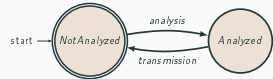
Every analysis must be transmitted exactly once

# Requirement 3

## Plant



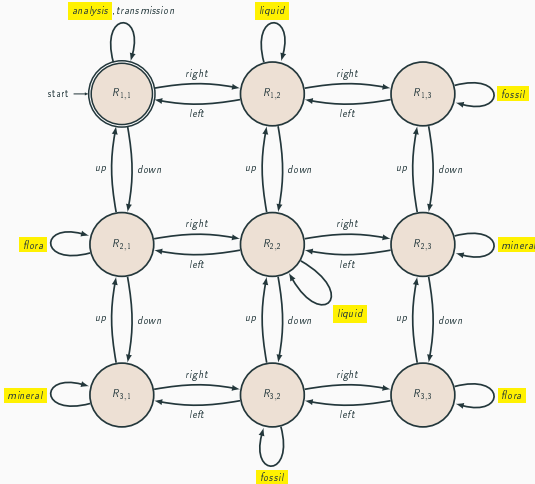
## Requirement



Every **analysis** must be **transmitted** exactly once

# Requirement 4

## Plant



## Requirement

- States?
- Events?
- Marking?

Exactly 1 liquid, at least 2 fossils, at most 3 florae, from 1 to 3 minerals must be collected before analysis:



## Requirement 4: Decomposition

Exactly 1 liquid, at least 2 fossils, at most 3 florae, from 1 to 3 minerals must be collected before analysis :



4a) Exactly 1 liquid must be collected before analysis

4b) At least 2 fossils must be collected before analysis

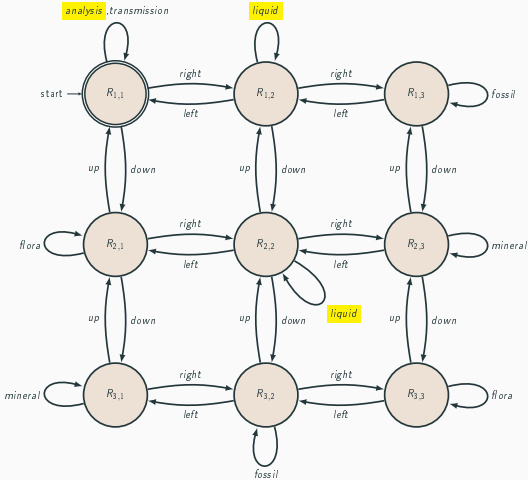
4c) At most 3 florae must be collected before analysis

4d) From 1 to 3 minerals must be collected before analysis

*Assumption: Every analysis removes all collected samples from the rover*

# Requirement 4a

## Plant



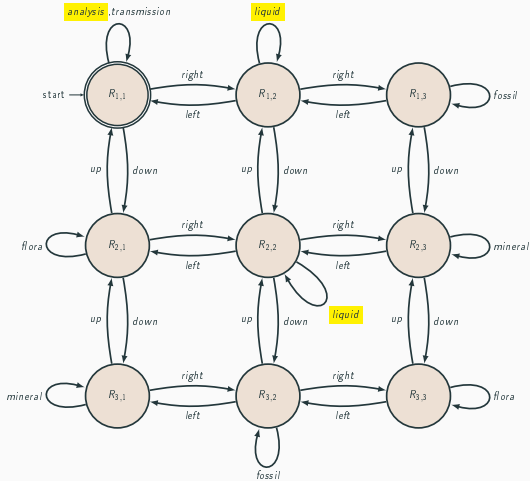
## Requirement

- States?
- **Events?**
- Marking?

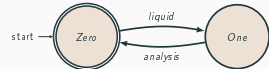
Exactly 1 **liquid** must be collected before **analysis**

# Requirement 4a

## Plant



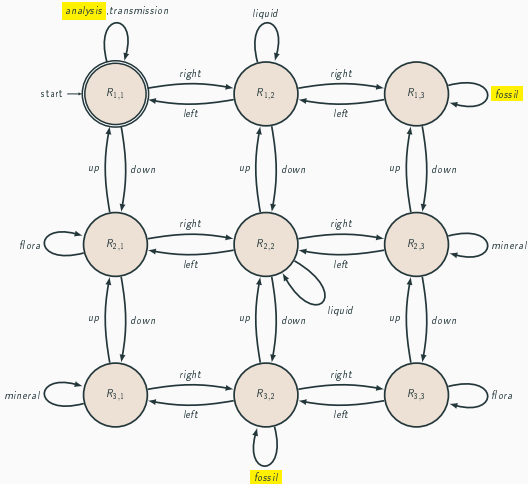
## Requirement



Exactly 1 liquid must be collected before analysis

# Requirement 4b

## Plant



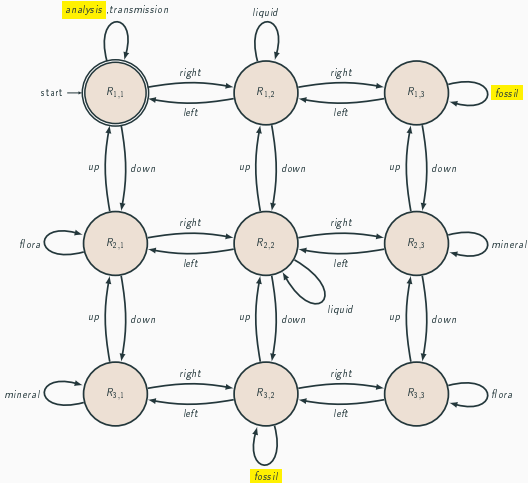
## Requirement

- States?
- Events?
- Marking?

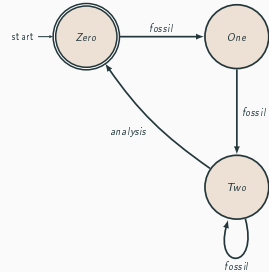
At least 2 fossils must be collected before analysis

# Requirement 4b

## Plant



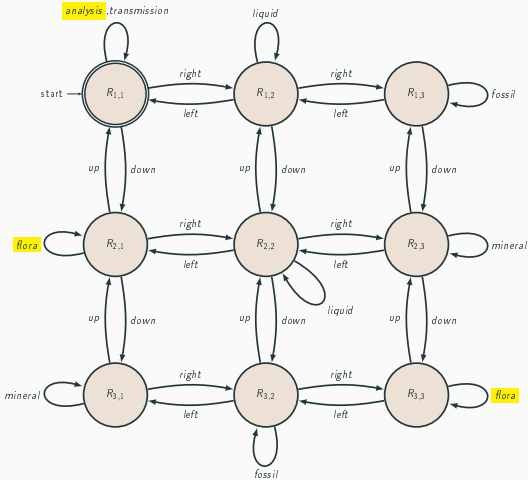
## Requirement



At least 2 fossils must be collected before analysis

# Requirement 4c

## Plant



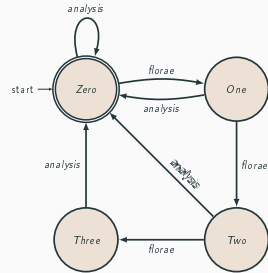
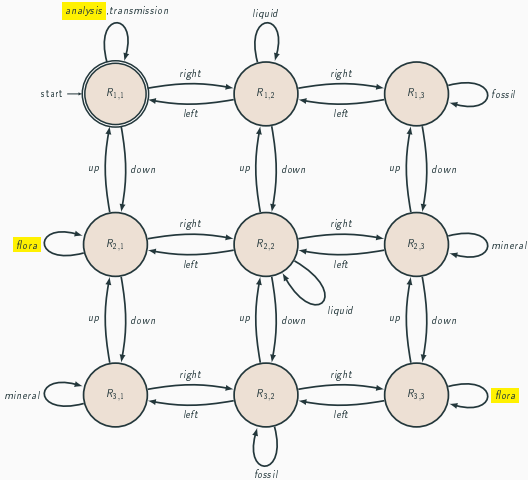
## Requirement

- States?
- Events?
- Marking?

At most 3 floras must be collected before analysis

# Requirement 4c

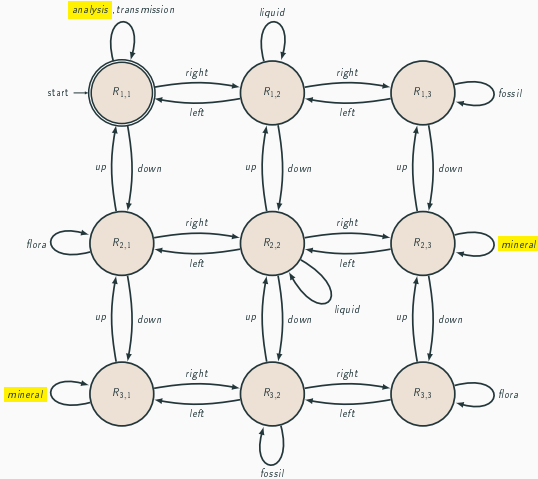
## Plant



At most 3 flora must be collected before analysis

# Requirement 4d

## Plant



## Requirement

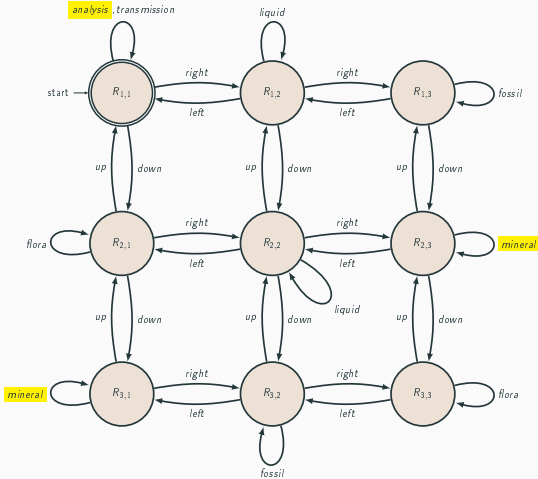
- States?
- Events?
- Marking?

From 1 to 3 minerals must be collected before analysis

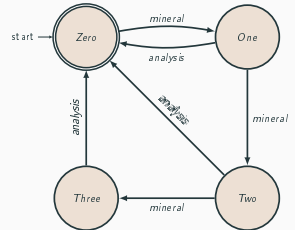


# Requirement 4d

## Plant



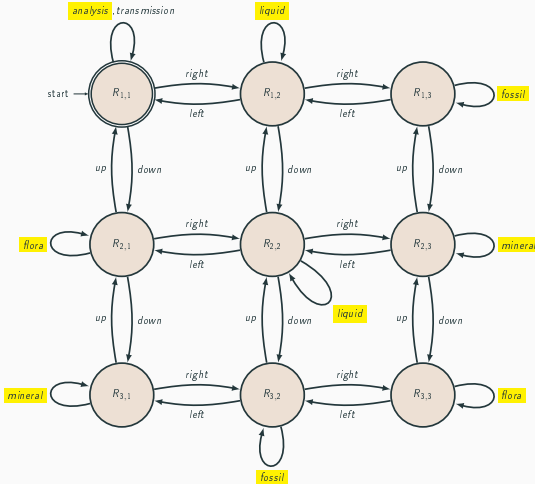
## Requirement



From 1 to 3 **minerals** must be collected before **analysis**

# Requirement 5

## Plant



## Requirement

- States?
- Events?
- Marking?

At most 1 sample from each subarea containing a fossil must be collected before analysis

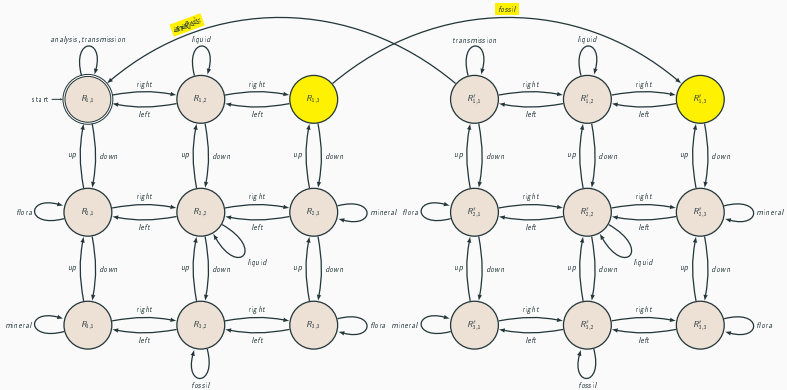
## Requirement 5: Decomposition

**At most 1 sample from each subarea containing a fossil must be collected before analysis**



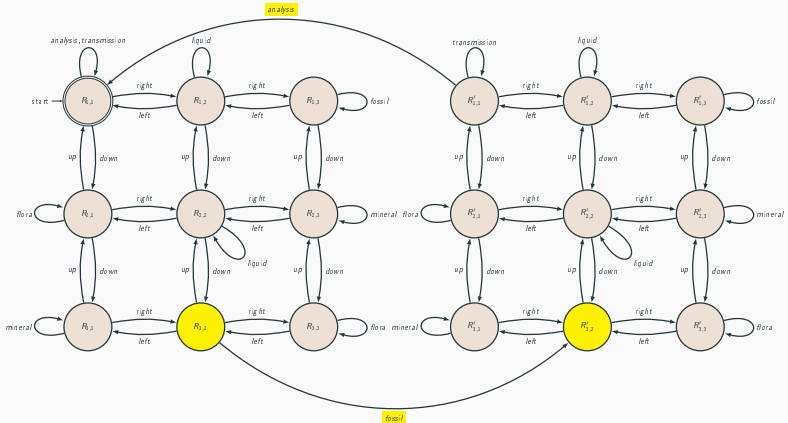
- 5a) **At most 1 sample from subarea (1, 3) containing a fossil must be collected before analysis**
- 5b) **At most 1 sample from subarea (3, 2) containing a fossil must be collected before analysis**

# Requirement 5a



**At most 1 sample from subarea (1, 3) containing a fossil must be collected before analysis**

# Requirement 5b



**At most 1 sample from subarea (3, 2) containing a fossil must be collected before analysis**

## Size of controller(s) (number of states)



Requirements	Generated States	Removed States	Total States
$R_1$	18	0	18
$R_1, R_2$	54	0	54
$R_1, R_2, R_3$	108	0	108
$R_1, R_2, R_3, R_4$	1458	846	612
$R_1, R_2, R_3, R_4, R_5$	1620	900	720