ALLOY 6

A MATTER OF TIME

Authors:

Luca Padalino

Francesca Pia Panaccione

Francesco Santambrogio

To understand how Alloy 5 deals with **dynamic modeling**

To understand which are the **limitations** of the **dynamic modeling** in Alloy 5 and why Alloy needed a **new version**

To understand which are the **new features** introduced in **Alloy 6**

STATIC

VS

DYNAMIC

STATIC

- Represents something that **does not change** over time
- Allows to describe a **legal state** of a dynamic system

```
abstract sig Person {
    father: lone Man,
    mother: lone Woman
}

sig Man extends Person {
    wife: lone Woman
}

sig Woman extends Person {
    husband: lone Man
}
```

STATIC

- Represents something that **does not change** over time
- Allows to describe a **legal state** of a dynamic system

STATIC MODEL INSTANCES

```
Person = {John, Sarah}
Man = {John}
Woman = {Sarah}
Married = {}
```

```
Person = {John, Sarah}
Man = {John}
Woman = {Sarah}
Married = {John, Sarah}
```

DYNAMIC

- Represents something **changing** over time

- Allows to describe possible **transitions between states** of the system

```
Person = {John, Sarah}
Man = {John}
Woman = {Sarah}
Married = {}
```

DYNAMIC TRANSITION

```
Person = {John, Sarah}
Man = {John}
Woman = {Sarah}
Married = {John, Sarah}
```

Until Alloy 6: **no** predefined **notion of time** and of state transition

BUT two ways to model dynamic aspects of a system:

1. By placing an **ordering** on some signatures
2. By introducing a **Time signature** expressing time

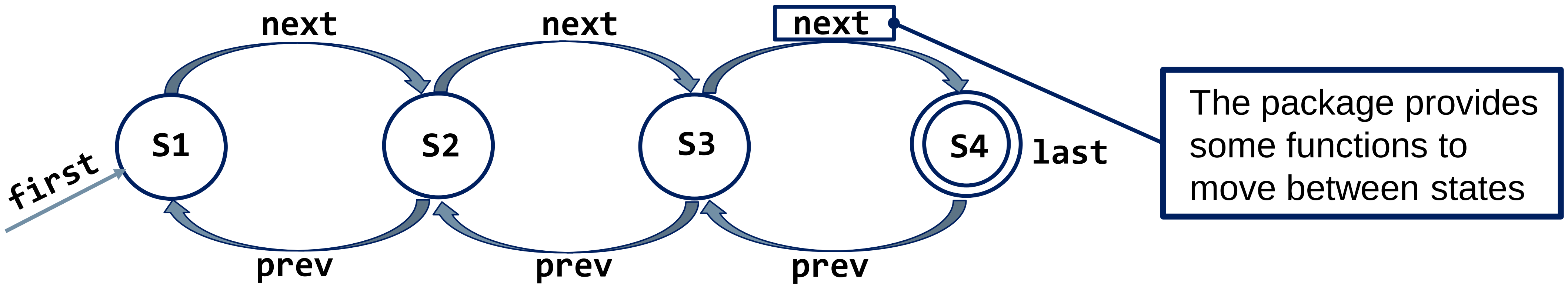
By placing an **ordering** on some signatures

```
open util/ordering[S]
sig S{}
```

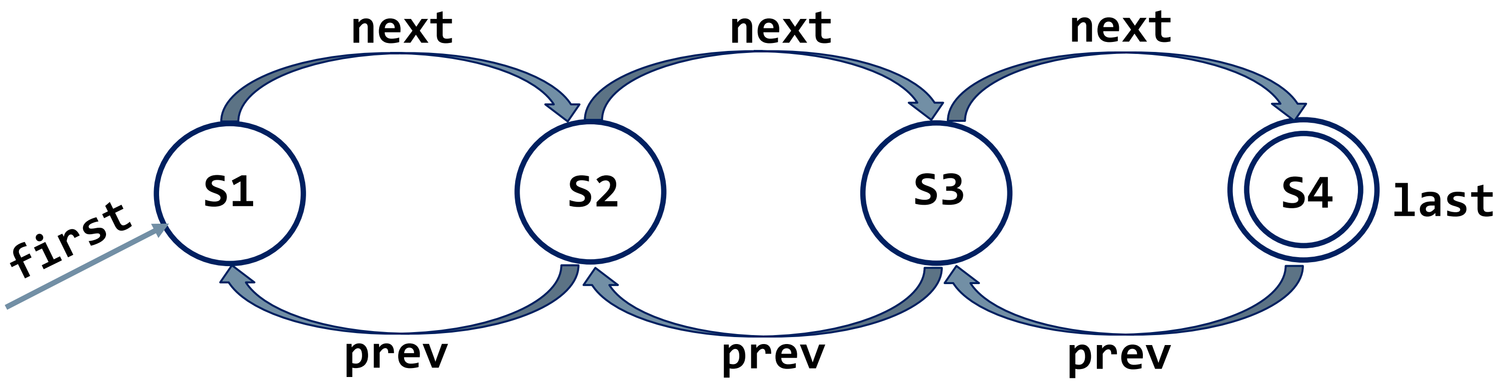
Creates a **single linear ordering** over signature S

- There are **multiple “S” atoms** representing the **same physical element** but **at different points in time**

- It is like creating a **finite state machine** on atoms of “S”



TRACE (Alloy 5): a fact that describes how the system will evolve by constraining "valid models" to ones where the system evolves properly




```
open util/ordering[S] as ord
…
fact traces {
    ord/first=S1
    all s: ord-ord/last |
    let s' = s.next |
    // general operations
    op1[s, s'] or … or opN[s, s']
}
```

The ordering method is really **hard to use** when we have **multiple** signatures that are changing or multiple properties that can change:

- We should **place the order on each signature** that can change over time
- → the code is not optimized

...WE CAN DO BETTER!

Until Alloy 6: **no** predefined **notion of time** and of state transition

BUT two ways to model dynamic aspects of a system:

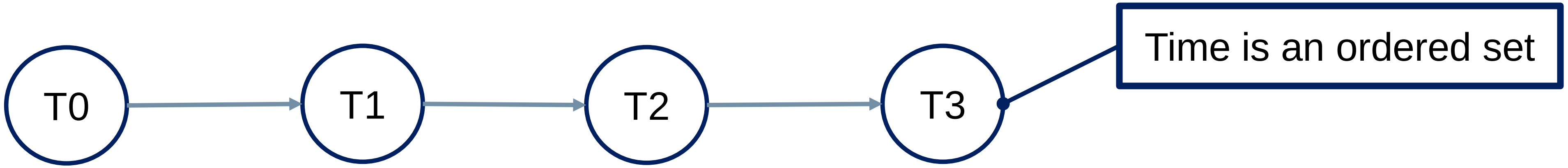
1. By placing an **ordering** on some signatures
2. By introducing a **Time signature** expressing time

by introducing a Time signature expressing time

```
open util/ordering[Time]
sig Time {}
```

Creates a **Time signature** that internally uses the ordering module

- The **linear ordering** is placed **on Time**



- We have to add a **time component** to each **relation** that **changes** over time

DYNAMIC

- Represents something **changing** over time

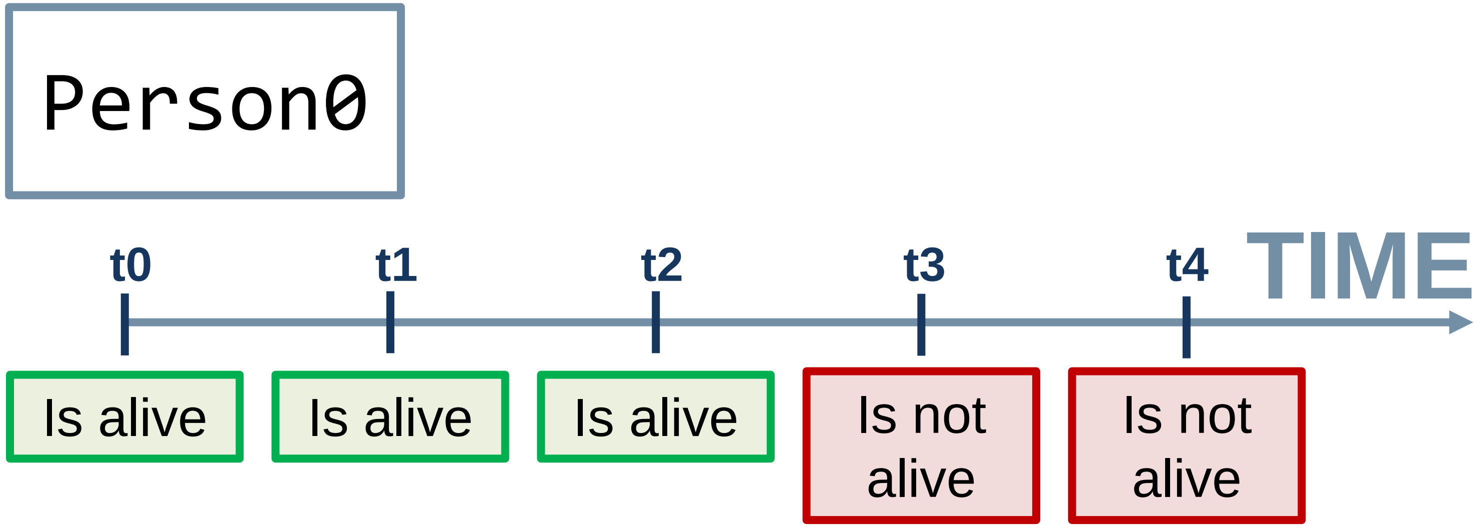
- Allows to describe possible **transitions between states** of the system

```
sig Time {}

abstract sig Person {
    father: Man
    mother: Woman
    alive: set Time
}

sig Man extends Person {
    wife: Woman lone -> Time
}

sig Woman extends Person {
    husband: Man lone -> Time
}
```

If the **relation** that changes over time is **BOOLEAN**:

```
sig Person {
   alive: set Time
}
```

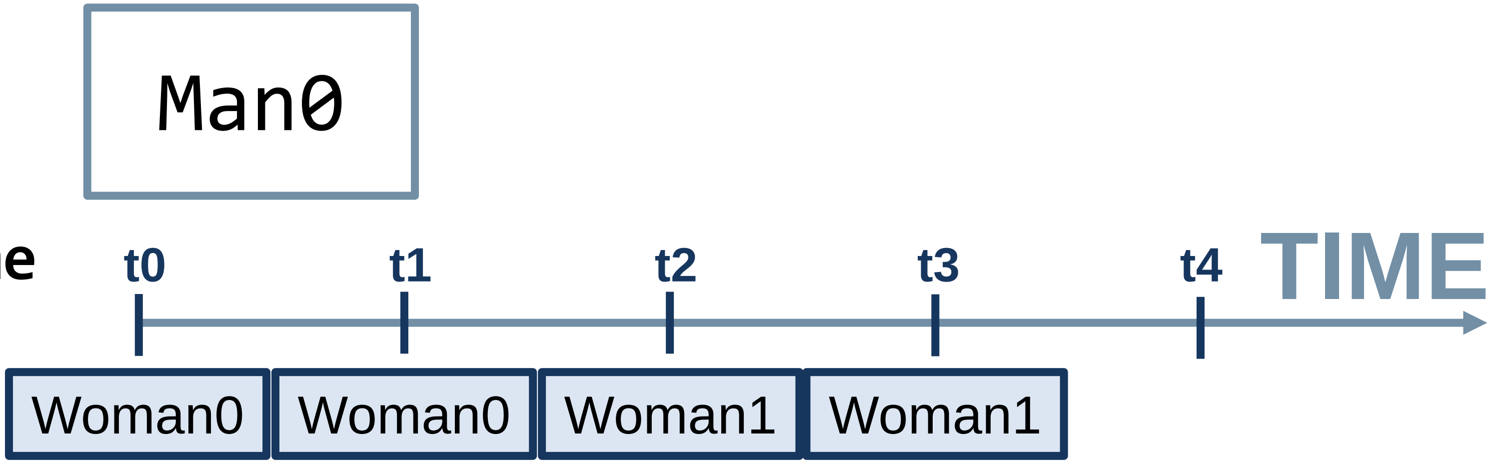
times where the field is true



If the **relation** that changes over time is **ARBITRARY**:

```
sig Man extends Person {
   wife: Woman lone -> Time
}
```

Multirelation with time:
for every Man and Time,
there is at most one Woman



There are **some limitations** to what we can model in a dynamic system:

- Import a package and try to **emulate** time without dealing with a **real** notion of **time**
- Alloy cannot test that some property is guaranteed to happen in infinite time (**liveness)**

...WE CAN DO BETTER!

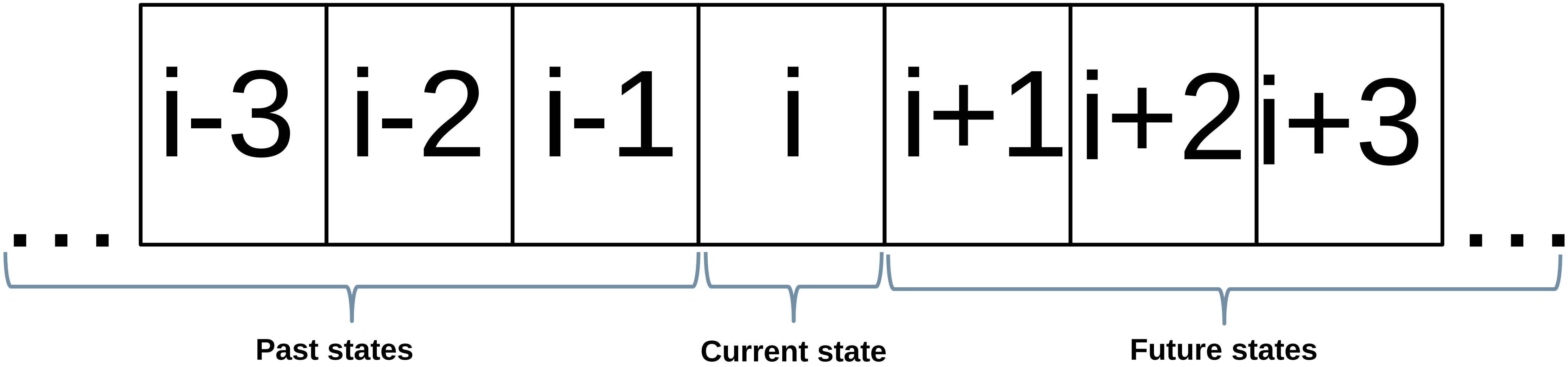
ALLOY 6

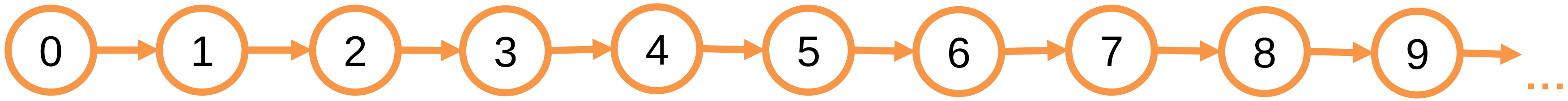
https://github.com/AlloyTools/org.alloytools.alloy/releases

Alloy 6: there is an **implicit**, built-in notion of **(discrete) time**

1. Linear temporal logic
2. Mutable signatures and fields
3. Temporal operators
4. Time horizon
5. New visualizer
6. Concurrency

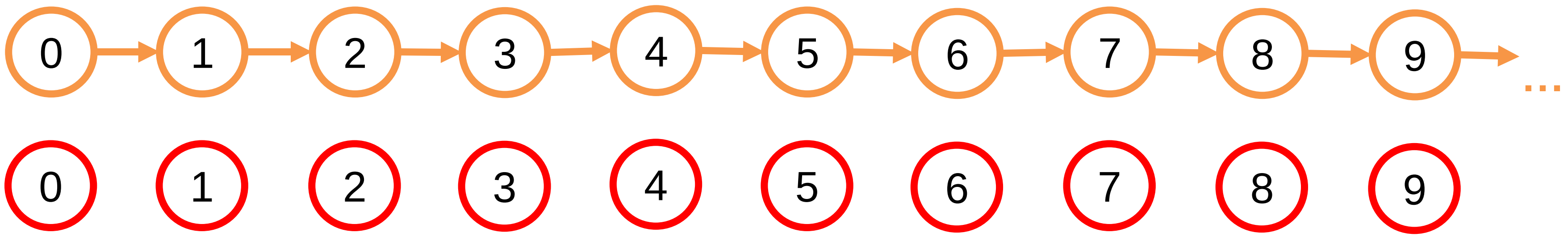
LINEAR TEMPORAL LOGIC (LTL): *«an infinite sequence of states where each point in time has a unique successor, based on a linear-time perspective»*





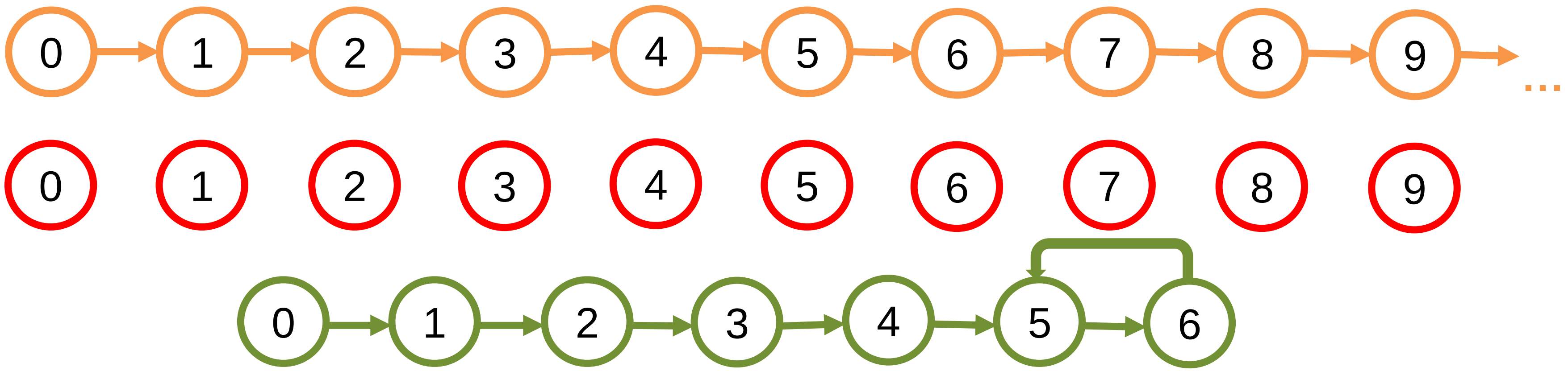
Instance = **TRACE**: infinite sequence of states

LINEAR TEMPORAL LOGIC (LTL)
State



Instance = **TRACE**: infinite sequence of states

STATE: a valuation for signatures and fields



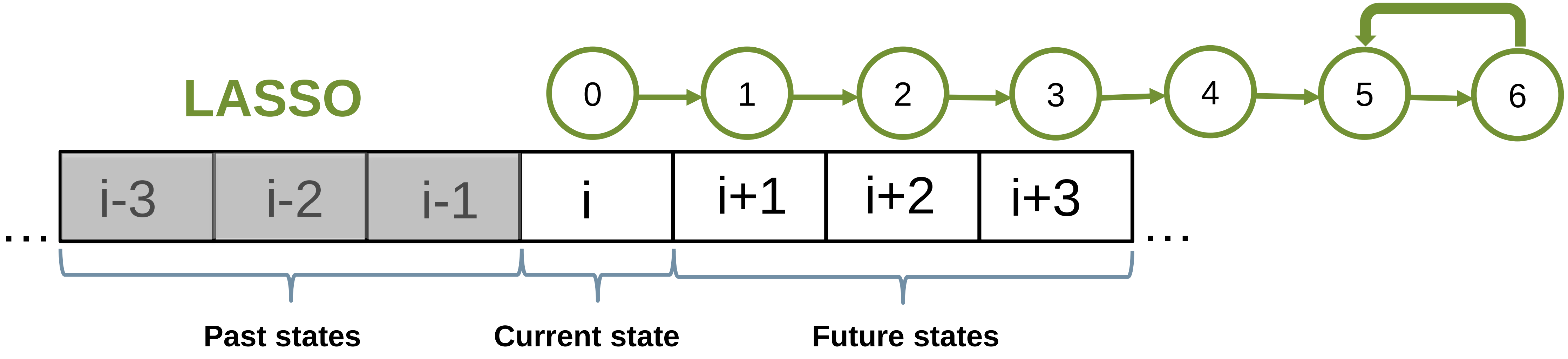
Instance = **TRACE**: infinite sequence of states

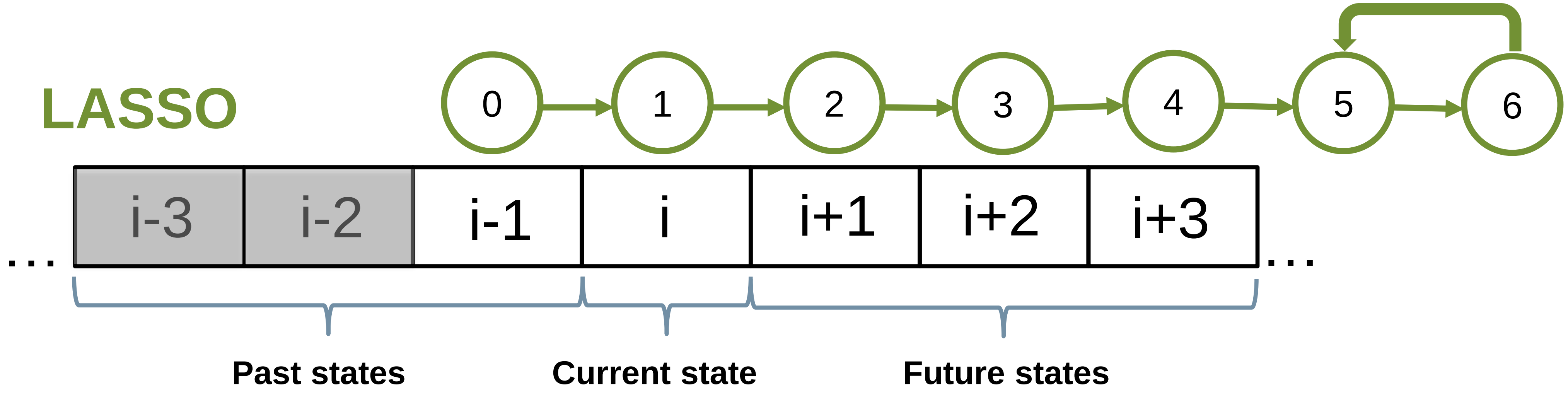
STATE: a valuation for signatures and fields

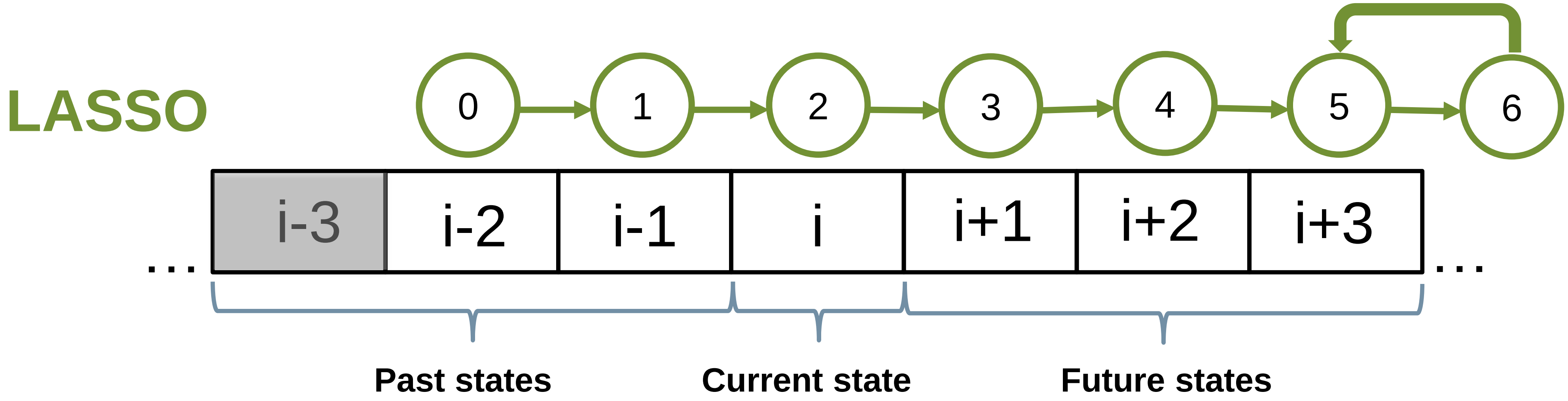
LASSO: a sequence of a finite number of states that loops back to a former state

LINEAR TEMPORAL LOGIC (LTL)

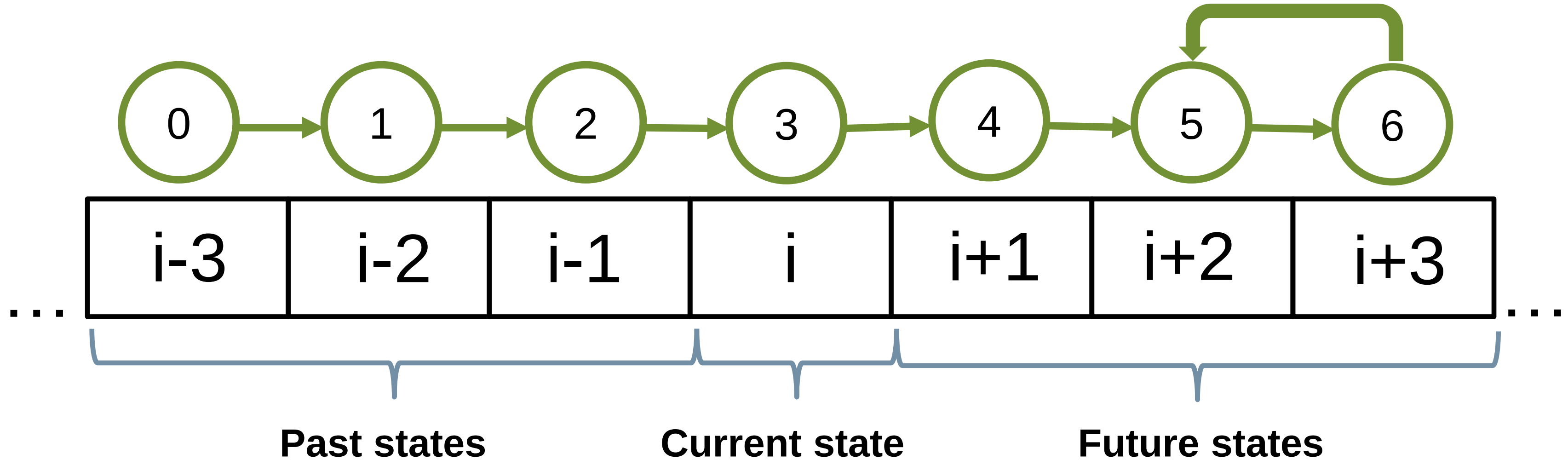
Trace, States, Lasso



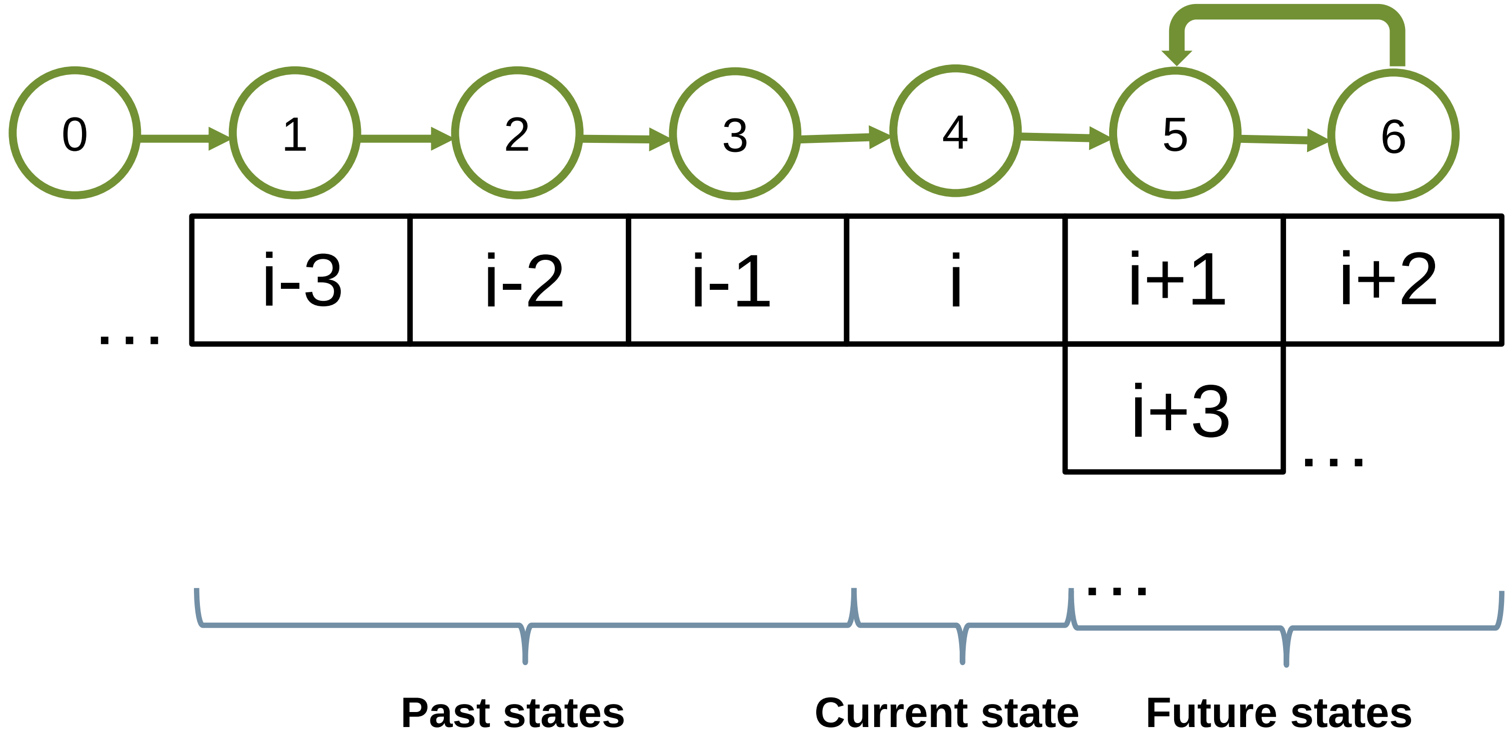

LASSO
0
1
2
3
4
5
6
…
i-3
i-2
i-1
i
i+1
i+2
i+3
…
Past states
Current state
Future states


LASSO
0
1
2
3
4
5
6
i-3
i-2
i-1
i
i+1
i+2
i+3
…
…
Past states
Current state
Future states

LASSO


0
1
2
3
4
5
6
…
i-3
i-2
i-1
i
i+1
i+2
i+3
…
Past states
Current state
Future states

LASSO



0
1
2
3
4
5
6
i-3
i-2
i-1
i
i+1
i+2
i+3
…
…
…
Past states
Current state
Future states

Alloy 6: an **implicit**, built-in notion of **(discrete) time**

1. Linear temporal logic
2. Mutable signatures and fields
3. Temporal operators
4. Time horizon
5. New visualizer
6. Concurrency

VAR

- A signature or field proceeded by **var** is said to be **mutable**
- A signature or field **not** proceeded by **var** is said to be **static** and assumed to be **constant** over time

```
enum Liveness {Alive, Dead, Unborn}

abstract sig Person {
    father: lone Man
    mother: lone Woman
    var liveness: Liveness
}

sig Man extends Person {
    var wife: lone Woman
}

sig Woman extends Person {
    var husband: lone Man
}
```

Alloy 6: an **implicit**, built-in notion of **(discrete) time**

1. Linear temporal logic
2. Mutable signatures and fields
3. Temporal operators
4. Time horizon
5. New visualizer
6. Concurrency

FUTURE	PAST
ALWAYS	HISTORICALLY
EVENTUALLY	ONCE
AFTER	BEFORE
UNTIL	SINCE
RELEASES	TRIGGERERD
;	(NO DUAL)

…FOR MORE INFO: (link)

Flipped Classroom

Video on Temporal Operators

Link Video

10 min.

Quiz 1

Dynamic modeling in Alloy 5

https://forms.office.com/e/9bjmhZTQ0j

5 min.

Quiz solutions

1. What is a dynamic model?
   - ❑ A model that represents something static.
   - ❑ A model that represents something changing over time.
   - ❑ A model that has a first-class notion of time.
2. How is time emulated in Alloy 5?
   - ❑ By using utility macros.
   - ❑ By placing an ordering on some signature.
   - ❑ By encoding it in the signature fields.
3. What is a trace in Alloy 5?
   - ❑ A fact that describes how the system will evolve.
   - ❑ A module that helps to model time.
   - ❑ A predicate that relates each state to the next state in the sequence.
4. What is the purpose of a time signature?
   - ❑ To represent complex specifications with multiple changing entities or properties.
   - ❑ To represent simple boolean properties that change over time.
   - ❑ To encode arbitrary properties with multirelations.

Quiz 2

LTL and Mutable Signatures and Fields

https://forms.office.com/e/G3MzQugLb5

5 min.

LTL & MUTABLE SIGNATURES AND FIELDS

Quiz solutions

1. What does the 'var' keyword do in Alloy 6?
    - ❑ Specifies that a signature or field is constant over time
    - ❑ Specifies that a signature or field is mutable
    - ❑ Specifies that a signature or field is a trace
    - ❑ Specifies that a signature or field is a lasso trace
2. What is a static signature or field in Alloy 6?
    - ❑ A signature or field that is constant over time
    - ❑ A signature or field that is a trace
    - ❑ A signature or field that is a lasso trace
    - ❑ A signature or field that is mutable
3. What is linear-time temporal logic used for in Alloy 6?
    - ❑ Reasoning about future and past states along a trace
    - ❑ Reasoning about constant values
    - ❑ Reasoning about mutable values
    - ❑ Reasoning about lasso traces

Quiz 3

Temporal Operators

https://forms.office.com/e/d5Himvahqs

10 min.

TEMPORAL OPERATORS
Quiz solutions

1. What is the condition for the expression "F until G" to be true in state i?
   - ❑ G is true in some state j ≥ i and F is true in every state k such that i ≤ k < j
   - ❑ G is true in every state ≥ i up to and including a state k in which F is true
   - ❑ F is true in state i and G is true in state i + 1
2. What is the condition for the expression "F ; G" to be true in state i?
   - ❑ G is true in some state j ≥ i and F is true in every state k such that i ≤ k < j
   - ❑ G is true in every state ≥ i up to and including a state k in which F is true
   - ❑ F is true in state i and G is true in state i + 1
3. What is the condition for the expression "always F" to be true in state i?
   - ❑ F is true in some state ≥ i
   - ❑ F is true in every state ≥ i
   - ❑ F is true in state i + 1
4. What is the condition for the expression "eventually F" to be true in state i?
   - ❑ F is true in some state ≥ i
   - ❑ F is true in every state ≥ i
   - ❑ F is true in state i + 1

TEMPORAL OPERATORS
Quiz solutions

5. What is the condition for the expression "after F" to be true in state i?
   - ❑ F is true in some state ≤ i
   - ❑ F is true in every state ≤ i
   - ❑ F is true in state i + 1
6. What is the condition for the expression "before F" to be true in state i?
   - ❑ F is true in some state ≤ i
   - ❑ F is true in every state ≤ i
   - ❑ F is true in state i – 1
7. What is the condition for the expression "historically F" to be true in state i?
   - ❑ F is true in some state ≥ i
   - ❑ F is true in every state ≤ i
   - ❑ F is true in state i + 1
8. What is the condition for the expression "once F" to be true in state i?
   - ❑ F is true in some state ≤ i
   - ❑ F is true in every state ≤ i
   - ❑ F is true in state i + 1

Alloy 6: an **implicit**, built-in notion of **(discrete) time**

1. Linear temporal logic
2. Mutable signatures and fields
3. Temporal operators
4. Time horizon
5. New visualizer
6. Concurrency

TIME HORIZON
Number of steps

TIME HORIZON: the possible number of transitions of lasso traces to explore

default	#steps = 10	`for 10 steps`
N	1 <= #steps <= N	`for N steps`
M .. N	M <= #steps <= N	`for M .. N steps`
1..	1 <= #steps	`for 1 .. steps`

TIME HORIZON
Model-checking

TIME HORIZON: the possible number of transitions of lasso traces to explore

default	#steps = 10	**BOUNDED MODEL-CHECKING**
N	1 <= #steps <= N	
M .. N	M <= #steps <= N	
1..	1 <= #steps	**COMPLETE MODEL-CHECKING**

Quiz 4

Time Horizon

https://forms.office.com/e/SXfQ5ByiNJ

5 min.

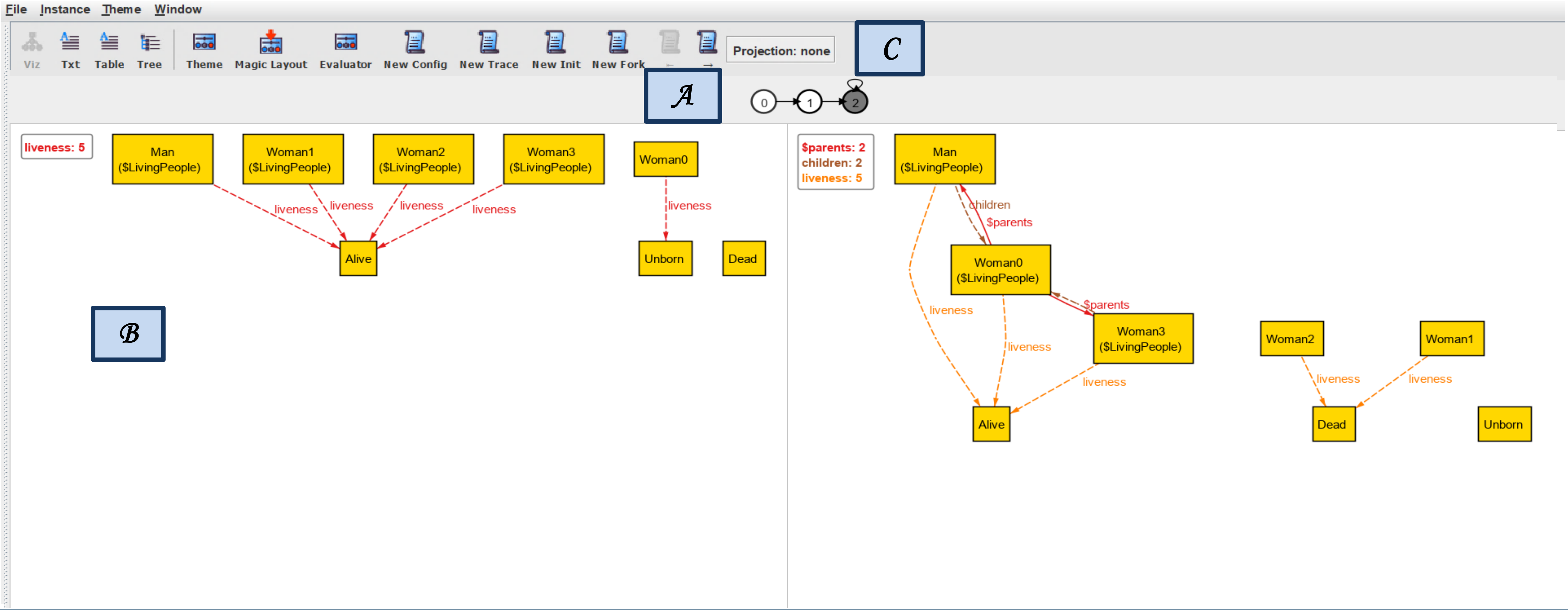
Quiz solutions

1. What is the time horizon in Alloy used for?
   - ❑ To specify the upper bound on the number of transitions in a lasso trace
   - ❑ To specify the lower bound on the number of transitions in a lasso trace
   - ❑ To specify the exact number of transitions in a lasso trace
   - ❑ To specify the type signature names in plain scopes
2. What is a lasso trace?
   - ❑ An infinite and non-repeating sequence of transitions
   - ❑ A finite and non-repeating sequence of transitions
   - ❑ An infinite and periodic sequence of transitions
   - ❑ A finite and periodic sequence of transitions
3. What is the purpose of the steps keyword in Alloy?
   - ❑ To specify the upper bound on the number of transitions in a lasso trace
   - ❑ To specify the lower bound on the number of transitions in a lasso trace
   - ❑ To specify the exact number of transitions in a lasso trace
   - ❑ To specify the type signature names in plain scopes
4. What is the difference between complete model-checking and bounded model checking?
   - ❑ Complete model-checking checks over all possible traces without bounding them upfront, while bounded model checking only checks a subset of possible traces with an upper bound on the number of transitions.
   - ❑ Complete model-checking checks only a subset of possible traces with an upper bound on the number of transitions, while bounded model checking checks over all possible traces without bounding them upfront.
   - ❑ Complete model-checking checks only the first and last states in a lasso trace, while bounded model checking checks all states in a lasso trace.
   - ❑ Complete model-checking and bounded model checking are the same thing.

Alloy 6: an **implicit**, built-in notion of **(discrete) time**

1. Linear temporal logic
2. Mutable signatures and fields
3. Temporal operators
4. Time horizon
5. New visualizer
6. Concurrency

NEW VISUALIZER

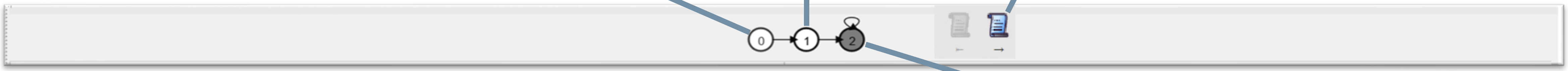


NEW VISUALIZER
Trace, states, lasso

Current state

Next state

To visualize the **next pair of states** in the trace (in this case 1-2)

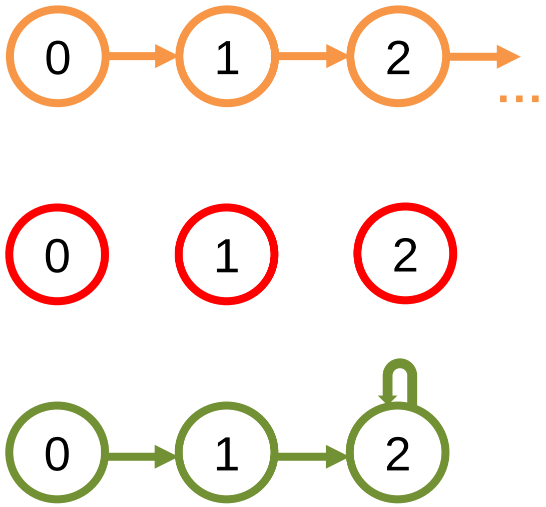


The state after the next state: it is in **gray** because not currently displayed

TRACE

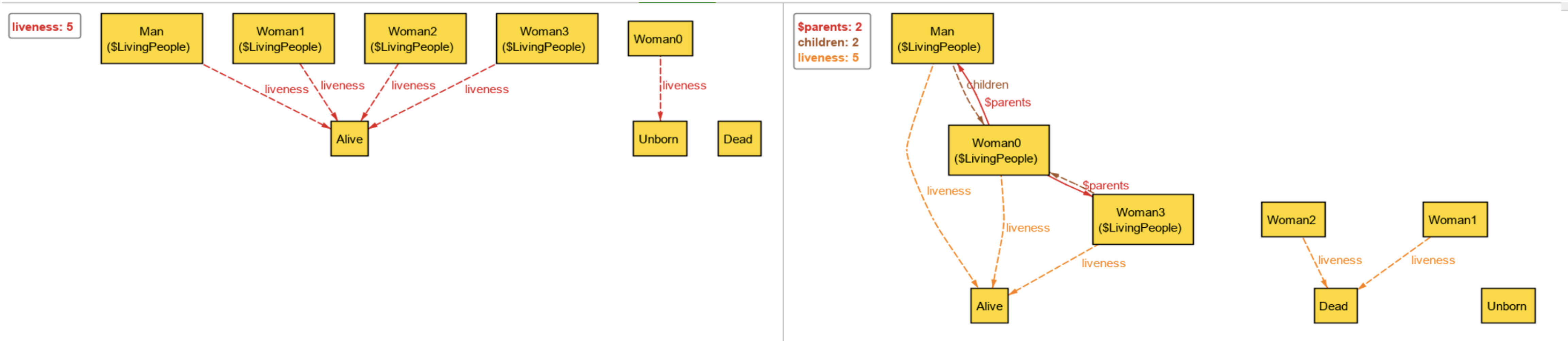
STATES

LASSO



𝓑

The visualizer shows a **step-pair**

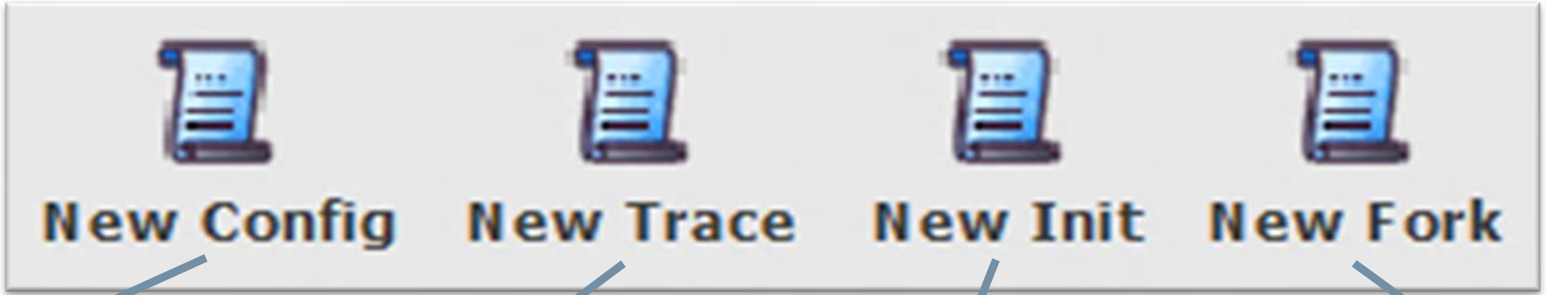


Current state **Next state**

NEW VISUALIZER

New generating options

C



yields a trace where the **configuration changed**

yields a **new trace** under the **same configuration**

yields a trace with a **different initial state**, under the **same configuration**

yields a **new trace** that is **similar to the current** one until the currently-displayed state but **differs afterwards**

Quiz 5

New Visualizer

https://forms.office.com/e/TWVpieWMCF

5 min.

Quiz solutions

1. What does the visualizer show?
   - ❑ All possible states of the system
   - ❑ The current state and the next state
   - ❑ All past states of the system.
   - ❑ The entire behavior of the system.
2. What is lasso in the visualizer?
   - ❑ A sequence of states that terminates.
   - ❑ A sequence of states that loops back to a previous state.
   - ❑ A sequence of states that terminates or loops back to a previous state.
   - ❑ A sequence of states that cannot be represented in the visualizer.
3. What does the "New Fork" option do?
   - ❑ Finds a new behavior trace from the existing configuration and initial state.
   - ❑ Finds a new initial state and behavior trace.
   - ❑ Fixes the present state and states before and finds a new next state.
   - ❑ Changes the immutable parts of the model and finds a new behavior trace.

NEW VISUALIZER
Quiz solutions

3. What is the purpose of the "New Init" option?
   - ❑ To find a new behavior trace from the existing configuration and initial state.
   - ❑ To change the immutable parts of the model and find a new behavior trace.
   - ❑ To fix the immutable relations and find a new initial state and behavior trace.
   - ❑ To fix the present state and states before and find a new next state.
4. What happens when you select the "New Config" option?
   - ❑ The visualizer finds a new initial state and behavior trace.
   - ❑ The visualizer shows a popup message.
   - ❑ The visualizer changes the immutable parts of the model and finds a new behavior trace.
5. When does the visualizer show the old layout with a "New Instance" button?
   - ❑ When the model is entirely dynamic.
   - ❑ When the model is entirely static.
   - ❑ When there are too many forks in the visualizer.
   - ❑ When the model has too many states to fit in memory.

Alloy 6: an **implicit**, built-in notion of **(discrete) time**

1. Linear temporal logic
2. Mutable signatures and fields
3. Temporal operators
4. Time horizon
5. New visualizer
6. Concurrency

CONCURRENCY: a property of a system in which multiple processes or threads can run simultaneously or appear to be running simultaneously.

Scenarios that deal with **CONCURRENCY**:

- distributed systems
- multi-threading/multi-tasking applications
- data migrations…

…EXAMPLE ON EXERCISE LECTURE

- Two ways to deal with dynamic modeling in Alloy 5:
  - **Odering module**
  - **Time signature**

To understand how Alloy 5 deals with **dynamic modeling**

- Limitations of dynamic modeling in Alloy 5:
  - Cannot tell **deadlocks**
  - No **liveness property**
  - No built-in **notion of time**

To understand which are the **limitations** of the **dynamic modeling** in Alloy 5 and why Alloy needed a **new version**

- New features introduced in Alloy 6:
  - **Linear temporal logic**
  - **Mutable signatures and fields**
  - **Temporal operators**
  - **Time horizon**
  - **New visualizer**
  - **Concurrency**

To understand which are the **new features** introduced in **Alloy 6**