ALLOY 6

A MATTER OF TIME

Authors:
Luca Padalino
Francesca Pia Panaccione
Francesco Santambrogio

ALLOY 6
Lesson objectives

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

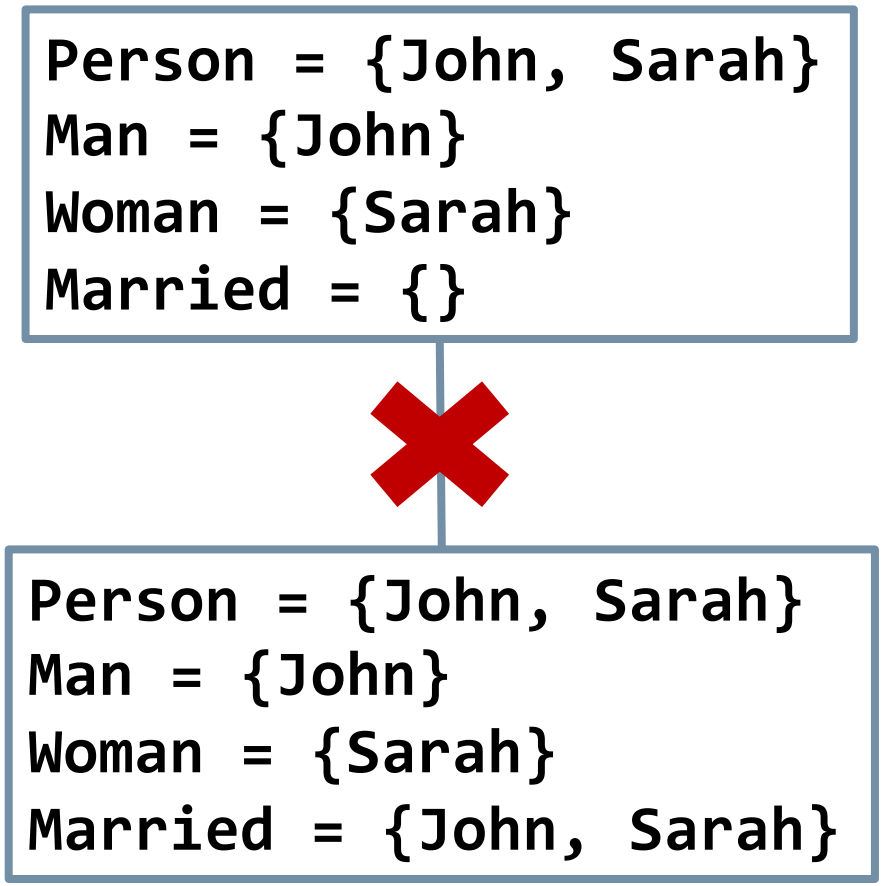
- Represents something that **does not change** over time
- Allows to describe a **legal state** of a dynamic system

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

sig Man extends Person {
    wife: lone Woman
}

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

STATIC

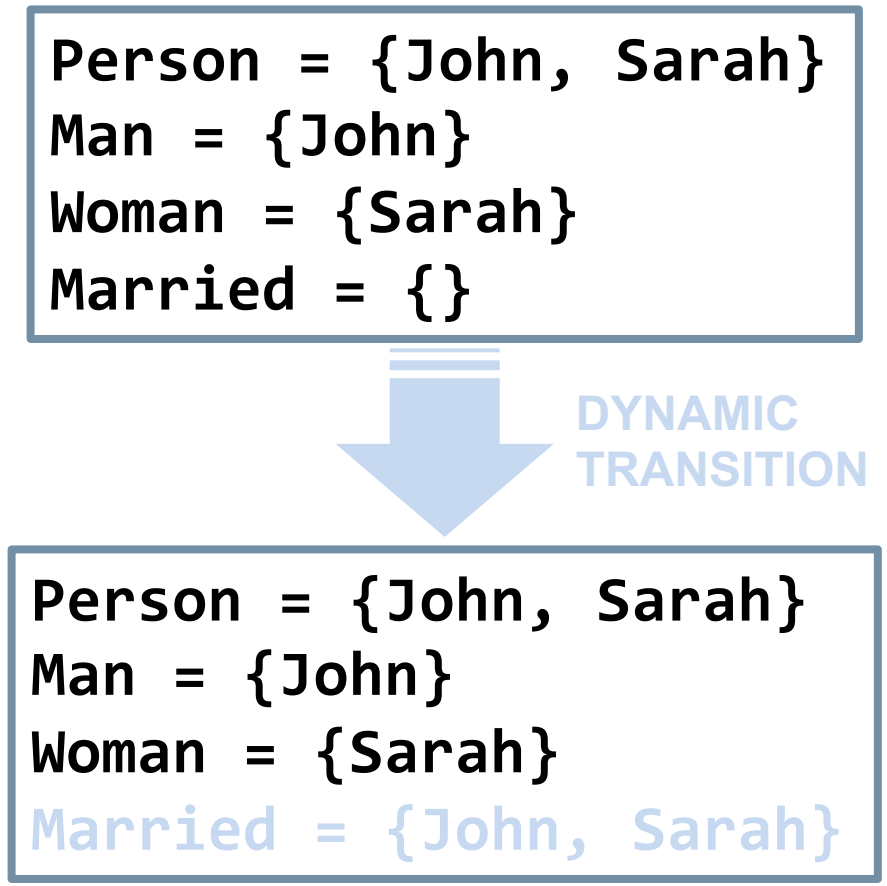
- Represents something that **does not change** over time
- Allows to describe a **legal state** of a dynamic system

STATIC MODEL INSTANCES



DYNAMIC

- Represents something **changing** over time
- Allows to describe possible **transitions between states** of the system



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

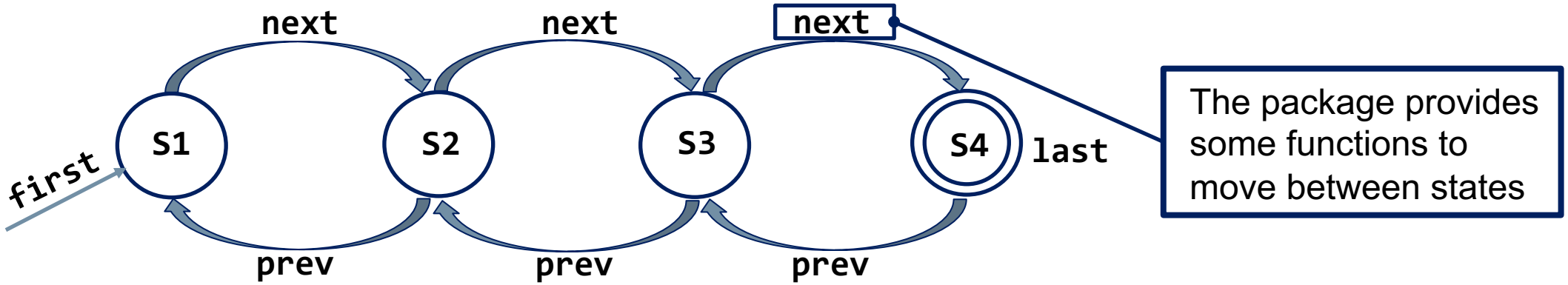
By placing an **ordering** on some signatures

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

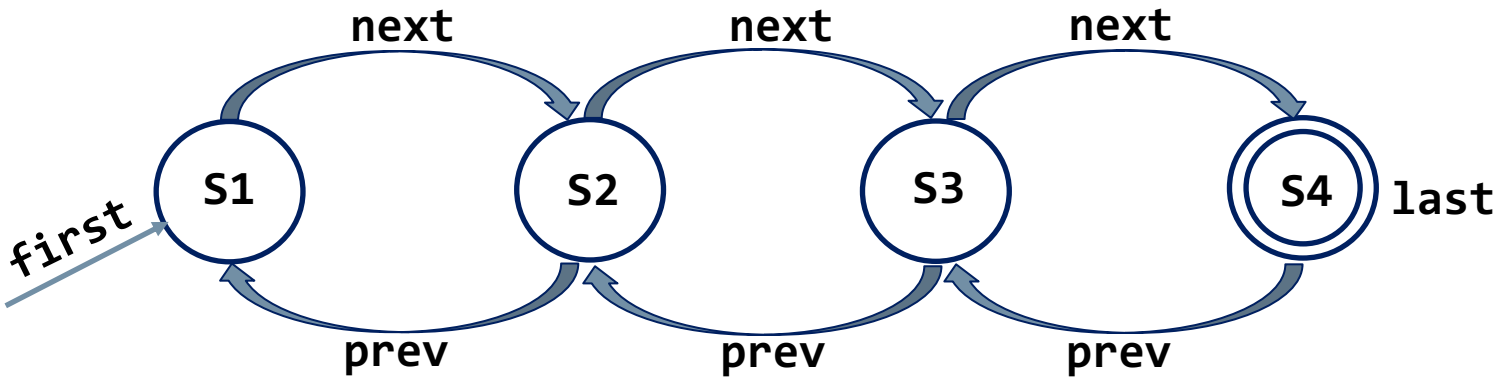
Creates a **single linear ordering** over signature S

- 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:

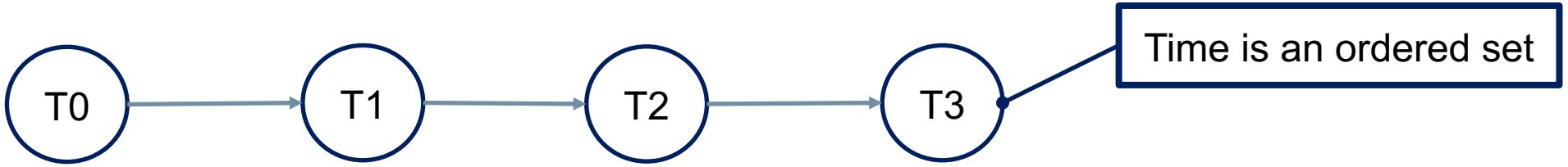
1. By placing an **ordering** on some signatures
2. By introducing a **Time signature** expressing time

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

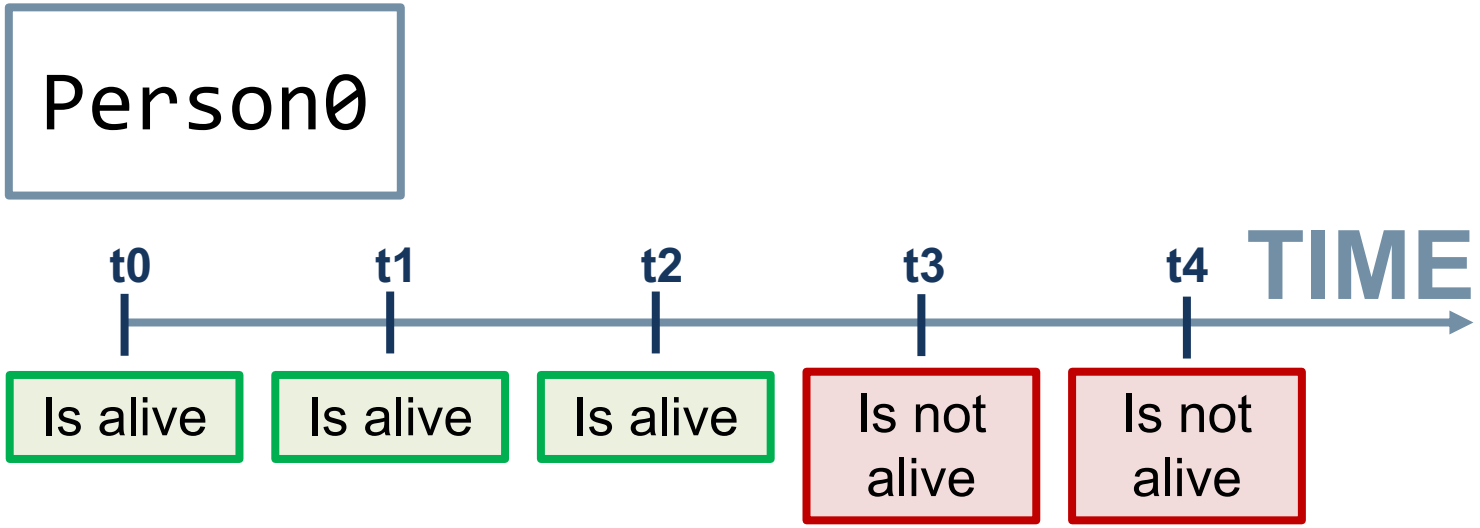
- Represents something **changing** over time
- Allows to describe possible **transitions between states** of the system

```
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
}
```

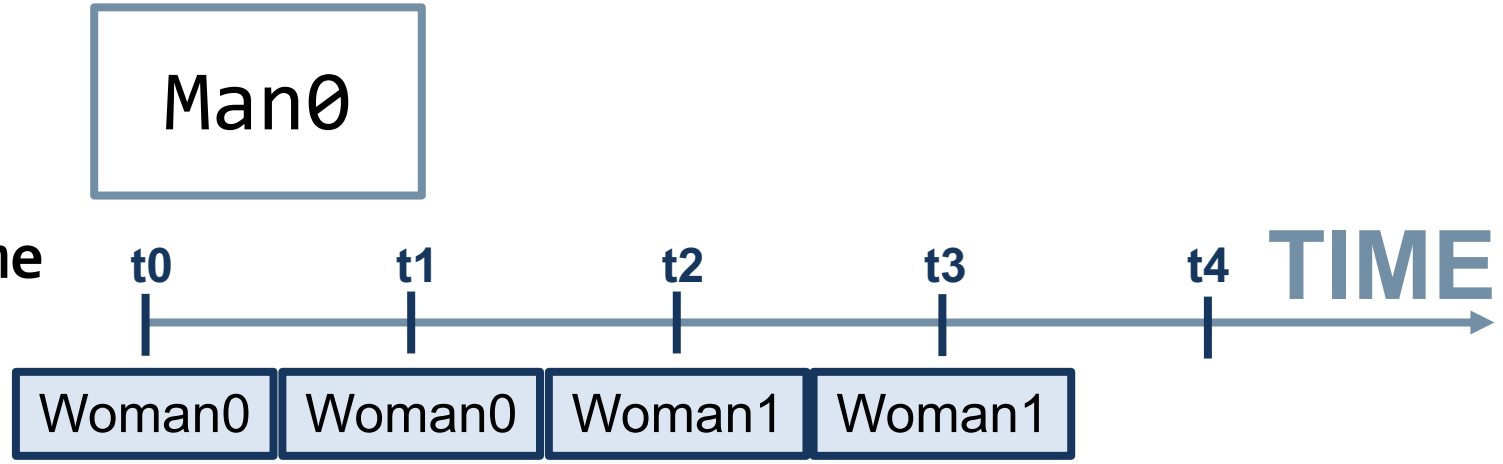
times where the field is true



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)**

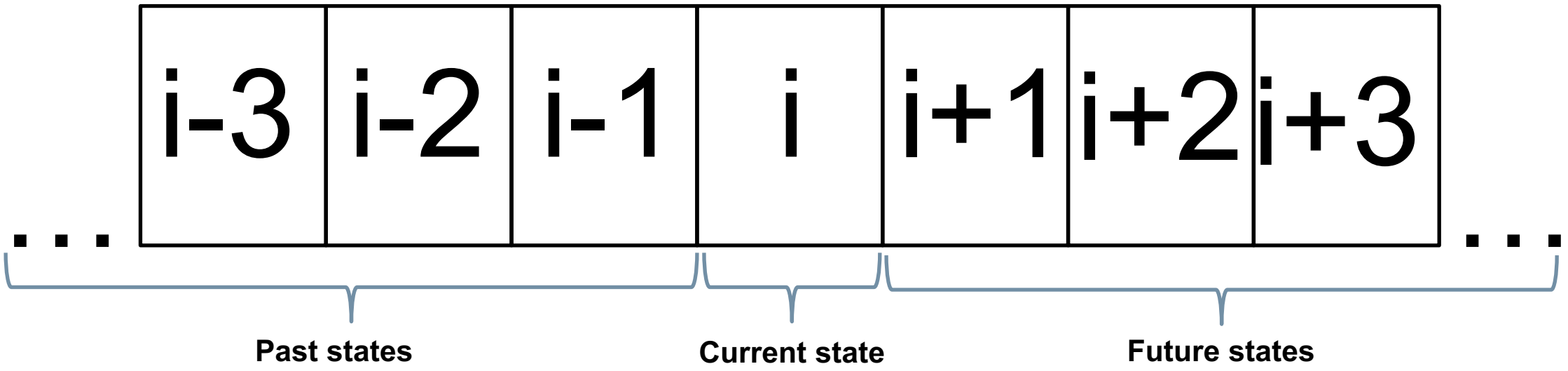
…WE CAN DO BETTER!

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

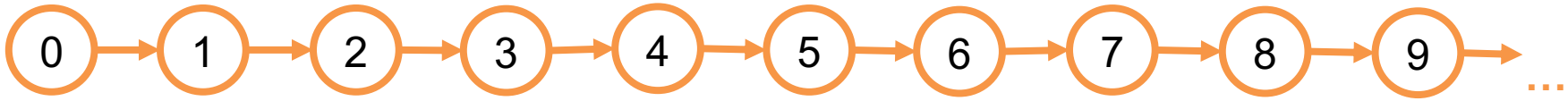
Alloy 6: there is an **implicit**, built-in notion of **(discrete) time**

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

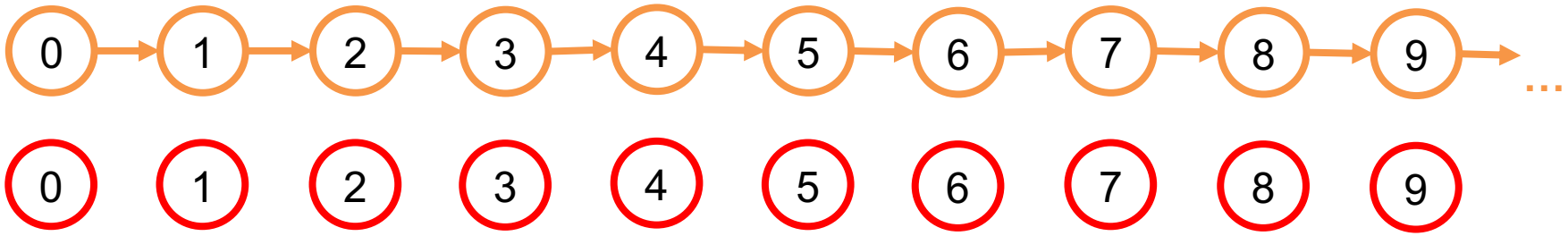
LINEAR TEMPORAL LOGIC (LTL): *«an infinite sequence of states where each point in time has a unique successor, based on a linear-time perspective»*



Trace

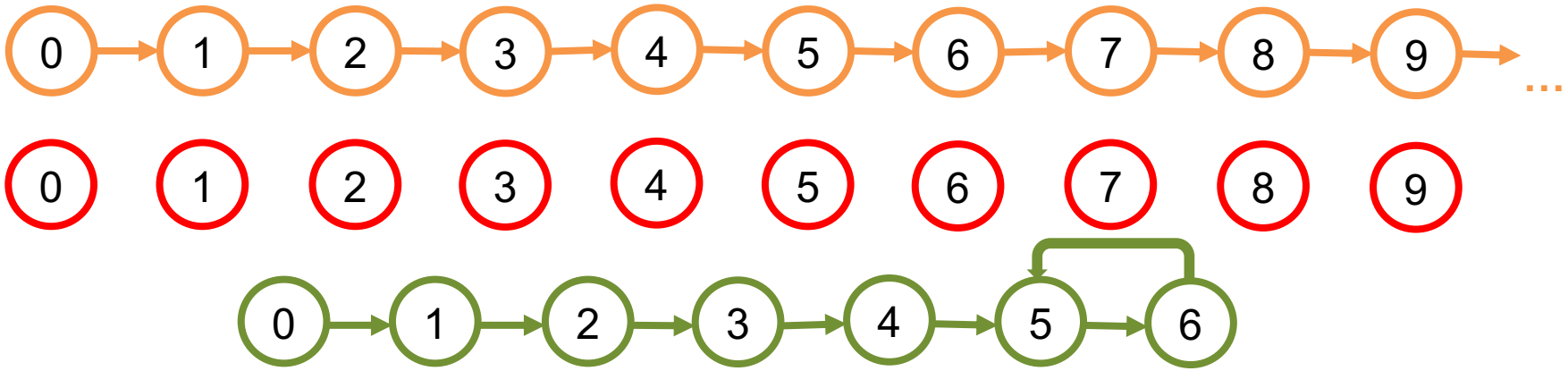


Instance = **TRACE**: infinite sequence of states



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

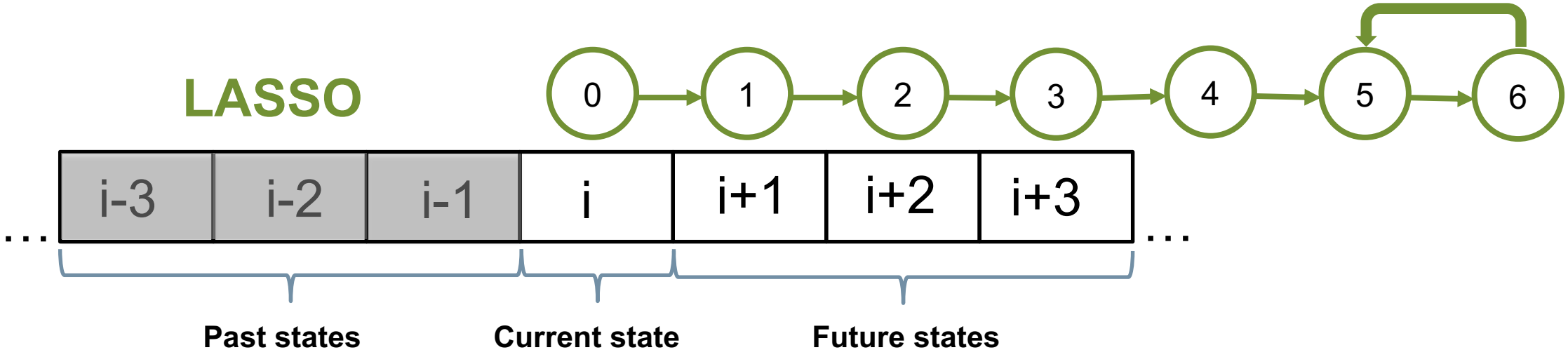
STATE: a valuation for signatures and fields

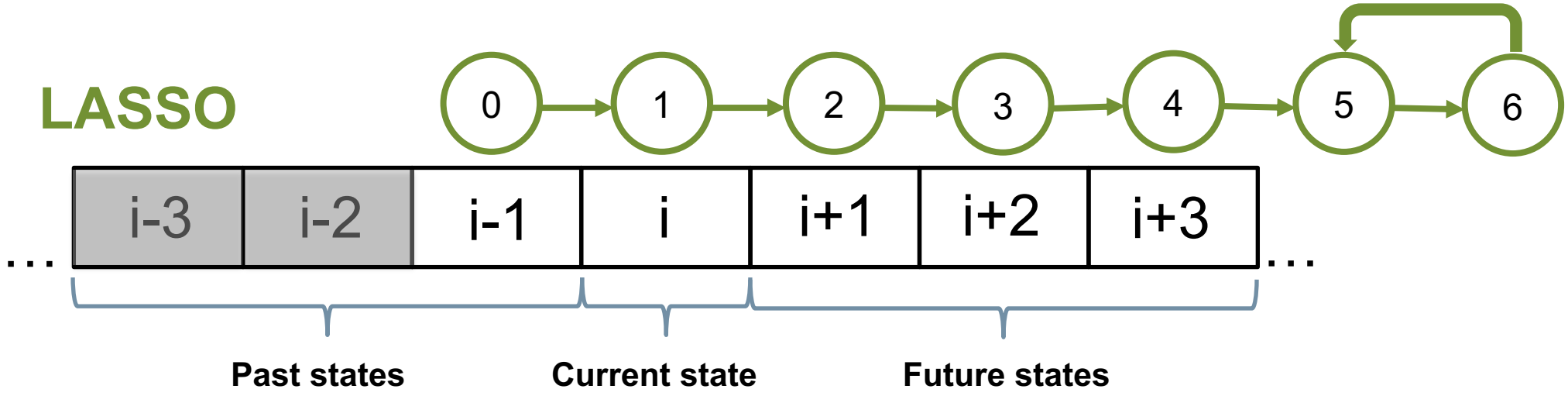


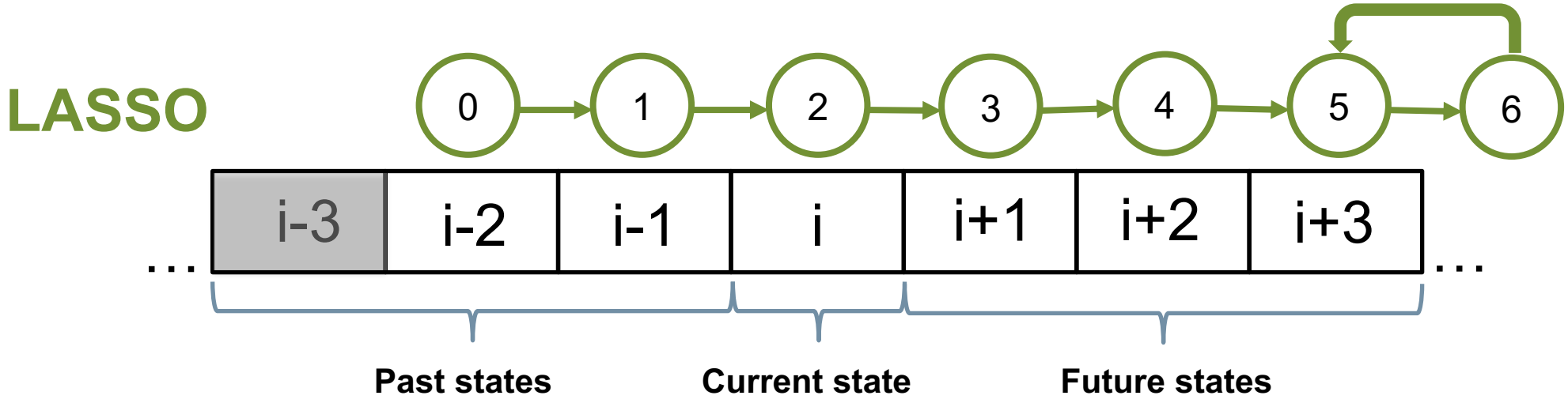
Instance = **TRACE**: infinite sequence of states

STATE: a valuation for signatures and fields

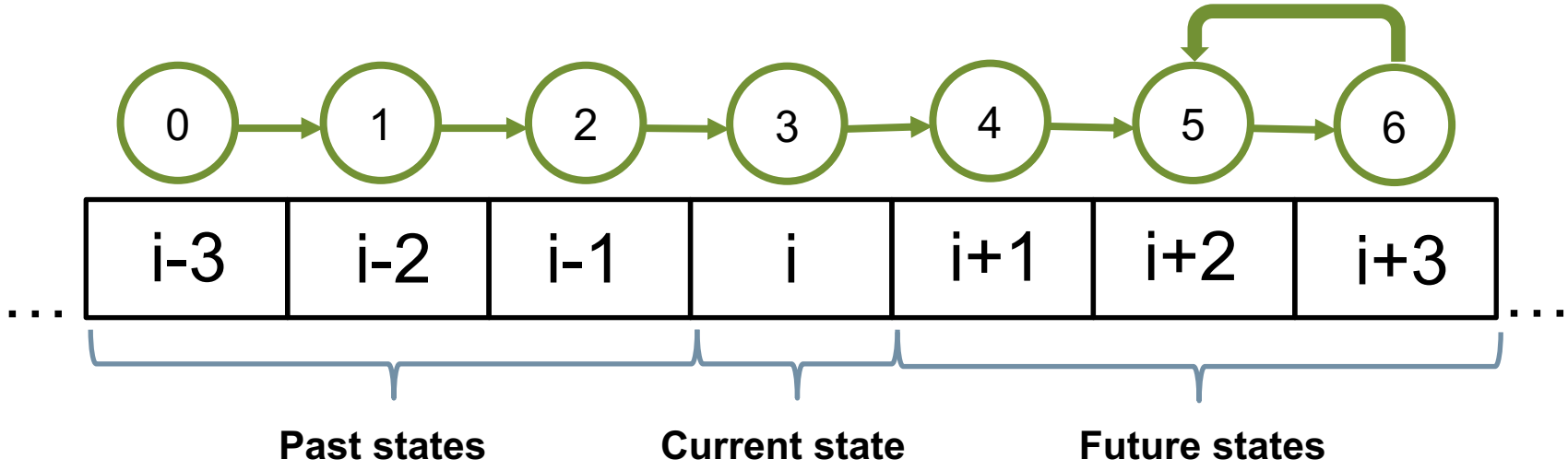
LASSO: a sequence of a finite number of states that loops back to a former state


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

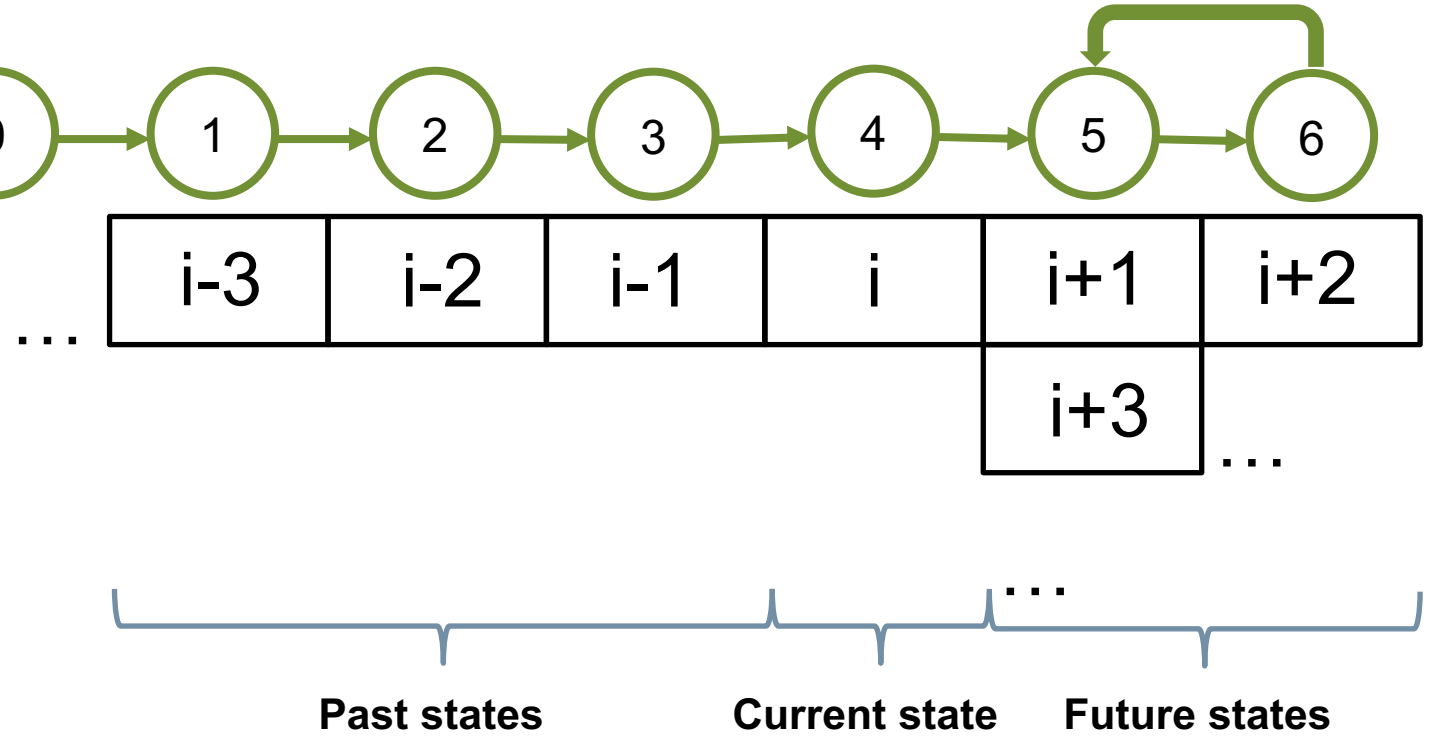

LASSO
0
1
2
3
4
5
6
i-3
i-2
i-1
i
i+1
i+2
i+3
Past states
Current state
Future states


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

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

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

TEMPORAL OPERATORS

Future and Past operators

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

…FOR MORE INFO: https://www.youtube.com/watch?v=GncsAmy29l0

Flipped Classroom

Video on Temporal Operators

Link Video

10 min.

Quiz 1

Dynamic modeling in Alloy 5

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

10 min.

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

10 min.

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.

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

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

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