Linear Temporal Logic

EECS 4315

www.eecs.yorku.ca/course/4315/

Linear temporal logic

Linear temporal logic (LTL) is a logic to reason about systems with nondeterminism.

The logic was introduced by Amir Pnueli.

A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 46–67. Providence, RI, USA, October/November 1977. IEEE.

Amir Pnueli (1941–2009)

- Recipient of the Turing Award (1996)
- Recipient of the Israel prize (2000)
- Foreign Associate of the U.S. National Academy of Engineering (1999)
- Fellow of the Association for Computing Machinery (2007)

Source: David Monniaux

Linear temporal logic

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \,\mathsf{U}\, f$$

where a is an atomic proposition.

An atomic proposition represents a basic property (such as the value of a particular variable being even or a particular method being invoked).

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \, \mathsf{U} \, f$$

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \,\mathsf{U}\, f$$

Question

Is $a \wedge \neg b$ is an LTL formula?

LTL

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \,\mathsf{U}\, f$$

Question

Is $a \wedge \neg b$ is an LTL formula?

Answer

Yes.

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \,\mathsf{U}\, f$$

Question

Is $a \wedge \bigcirc$ is an LTL formula?

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \, \mathrm{U} \, f$$

Question

Is $a \wedge \bigcirc$ is an LTL formula?

Answer

No.

LTL

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \, \mathsf{U} \, f$$

Question

Is $a \wedge \neg((\bigcirc b) \, \mathsf{U} \, c)$ is an LTL formula?

Definition

LTL is defined by the grammar

$$f ::= a \mid f \wedge f \mid \neg f \mid \bigcirc f \mid f \, \mathsf{U} \, f$$

Question

Is $a \wedge \neg((\bigcirc b) \, \mathsf{U} \, c)$ is an LTL formula?

Answer

Yes.

LTL

Given an execution path p, does it satisfy a particular LTL formula f?

An atomic proposition a is satisfied if a holds in the initial state of the execution path.

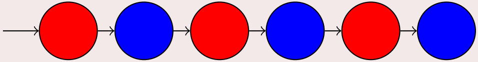
LTL

Given an execution path *p*, does it satisfy a particular LTL formula *f*?

An atomic proposition *a* is satisfied if a holds in the initial state of the execution path.

Question

Does the execution path



satisfy the atomic proposition red?

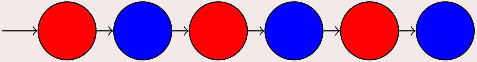
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

An atomic proposition a is satisfied if a holds in the initial state of the execution path.

Question

Does the execution path



satisfy the atomic proposition red?

Answer

Yes.

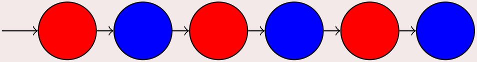
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

An atomic proposition a is satisfied if a holds in the initial state of the execution path.

Question

Does the execution path



satisfy the atomic proposition blue?

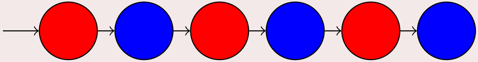
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

An atomic proposition a is satisfied if a holds in the initial state of the execution path.

Question

Does the execution path



satisfy the atomic proposition blue?

Answer

No.

LTL

Given an execution path p, does it satisfy a particular LTL formula f?

A formula $f \wedge g$ is satisfied if f holds and g holds.

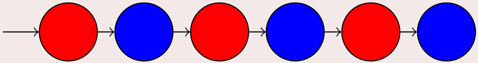
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

A formula $f \wedge g$ is satisfied if f holds and g holds.

Question

Does the execution path



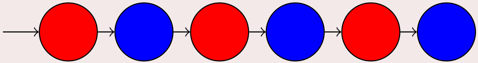
satisfy the formula red $\wedge$ blue?

Given an execution path p, does it satisfy a particular LTL formula f?

A formula $f \wedge g$ is satisfied if f holds and g holds.

Question

Does the execution path



satisfy the formula red $\wedge$ blue?

Answer

No.

LTL

Given an execution path p, does it satisfy a particular LTL formula f?

A formula $\neg f$ is satisfied if f does not hold.

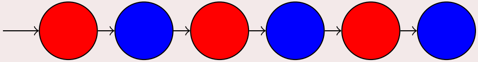
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

A formula $\neg f$ is satisfied if f does not hold.

Question

Does the execution path



satisfy the formula $\neg$ blue?

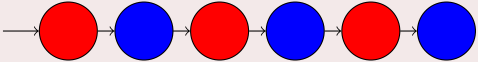
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

A formula $\neg f$ is satisfied if f does not hold.

Question

Does the execution path



satisfy the formula $\neg$ blue?

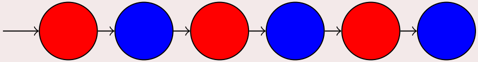
Answer

Yes.

Given an execution path p, does it satisfy a particular LTL formula f?

Question

Does the execution path

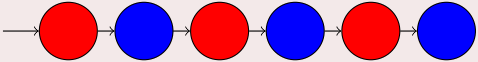


satisfy the formula red $\wedge \neg$ blue?

Given an execution path p, does it satisfy a particular LTL formula f?

Question

Does the execution path



satisfy the formula red $\wedge \neg$ blue?

Answer

Yes.

LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $\bigcirc a$ (pronounced as next a) is satisfied if a holds in the next state of the execution path (that is, the second state).

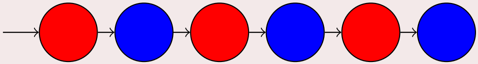
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $\bigcirc a$ (pronounced as next a) is satisfied if a holds in the next state of the execution path (that is, the second state).

Question

Does the execution path



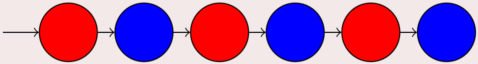
satisfy the formula $\bigcirc$red?

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $\bigcirc a$ (pronounced as next a) is satisfied if a holds in the next state of the execution path (that is, the second state).

Question

Does the execution path



satisfy the formula $\bigcirc$red?

Answer

No.

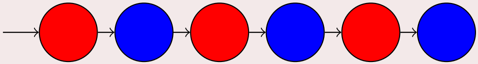
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $\bigcirc a$ (pronounced as next a) is satisfied if a holds in the next state of the execution path (that is, the second state).

Question

Does the execution path



satisfy the formula $\bigcirc$blue?

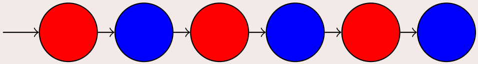
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $\bigcirc a$ (pronounced as next a) is satisfied if a holds in the next state of the execution path (that is, the second state).

Question

Does the execution path



satisfy the formula $\bigcirc$blue?

Answer

Yes.

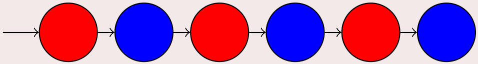
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $\bigcirc a$ (pronounced as next a) is satisfied if a holds in the next state of the execution path (that is, the second state).

Question

Does the execution path



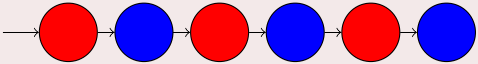
satisfy the formula $\bigcirc \bigcirc$ red?

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $\bigcirc a$ (pronounced as next a) is satisfied if a holds in the next state of the execution path (that is, the second state).

Question

Does the execution path



satisfy the formula $\bigcirc \bigcirc$ red?

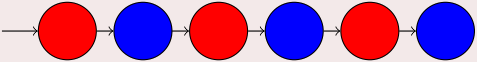
Answer

Yes.

Given an execution path p, does it satisfy a particular LTL formula f?

Question

Does the execution path

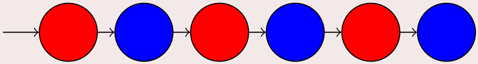


satisfy the formula red $\wedge \bigcirc$blue?

Given an execution path p, does it satisfy a particular LTL formula f?

Question

Does the execution path



satisfy the formula red $\wedge \bigcirc$blue?

Answer

Yes.

LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula $f \,\mathrm{U}\, g$ (pronounced as f until g) is satisfied if g holds in some state of the execution path and f holds in all states before that state.

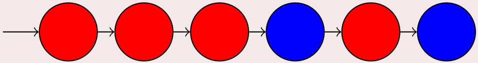
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula f U g (pronounced as f until g) is satisfied if g holds in some state of the execution path and f holds in all states before that state.

Question

Does the execution path



satisfy the atomic proposition red U blue?

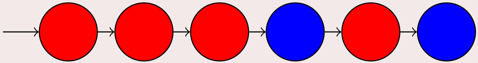
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula f U g (pronounced as f until g) is satisfied if g holds in some state of the execution path and f holds in all states before that state.

Question

Does the execution path



satisfy the atomic proposition red U blue?

Answer

Yes.

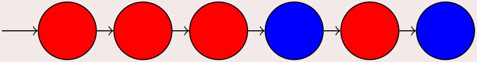
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula f U g (pronounced as f until g) is satisfied if g holds in some state of the execution path and f holds in all states before that state.

Question

Does the execution path



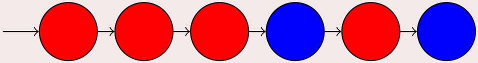
satisfy the atomic proposition blue U red?

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula f U g (pronounced as f until g) is satisfied if g holds in some state of the execution path and f holds in all states before that state.

Question

Does the execution path



satisfy the atomic proposition blue U red?

Answer

Yes![a]

[a] All states before the first red state are blue.

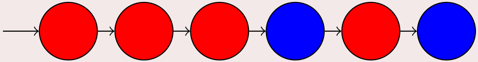
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula f U g (pronounced as f until g) is satisfied if g holds in some state of the execution path and f holds in all states before that state.

Question

Does the execution path



satisfy the atomic proposition red U ◯blue?

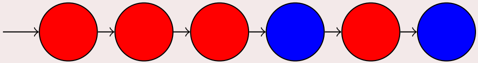
LTL

Given an execution path p, does it satisfy a particular LTL formula f?

The LTL formula f U g (pronounced as f until g) is satisfied if g holds in some state of the execution path and f holds in all states before that state.

Question

Does the execution path



satisfy the atomic proposition red U $\bigcirc$blue?

Answer

Yes.

Syntactic sugar

As usual

$$\begin{aligned} \text{true} &= a \vee \neg a \\ \text{false} &= \neg \text{true} \\ f \vee g &= \neg(\neg f \wedge \neg g) \\ f \Rightarrow g &= \neg f \vee g \end{aligned}$$

Also

$$\Diamond f = \text{true}\ \mathrm{U}\ f \qquad \text{(eventually } f\text{)}$$

$$\Box f = \neg \Diamond \neg f \qquad \text{(always } f\text{)}$$

Alternative syntax

$$\mathrm{X}f : \bigcirc f$$

$$\mathrm{F}f : \Diamond f$$

$$\mathrm{G}f : \Box f$$

> We introduce two basic tense operators, F and G.

A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, pages 46–67. Providence, RI, USA, October/November 1977. IEEE.

Transition system

Definition

A transition system is a tuple $\langle S, L, I, \rightarrow, \ell \rangle$ consisting of

- a set S of states,
- a set L of labels,
- a set $I \subseteq S$ of initial states,
- a transition relation $\rightarrow \subseteq S \times S$ such that for all $s \in S$ there exists $t \in S$ such that $s \rightarrow t$, and
- a labelling function $\ell : S \rightarrow 2^L$.

2^L denotes the set of subsets of L.

Powerset

Question

What is $2^{\{1,2,3\}}$?

Powerset

Question

What is $2^{\{1,2,3\}}$?

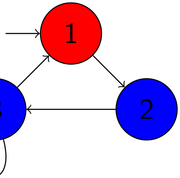
Answer

$\{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$

Transition system

Question

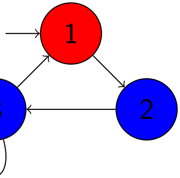
Formally define the transition system for the following system.



Transition system

Question

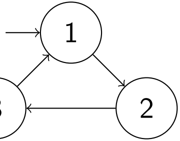
Formally define the transition system for the following system.



Answer

$\langle \{1, 2, 3\}, \{\text{red}, \text{blue}\}, \{1\}, \{1 \to 2, 2 \to 3, 3 \to 1, 3 \to 3\}, \{1 \mapsto \{\text{red}\}, 2 \mapsto \{\text{blue}\}, 3 \mapsto \{\text{blue}\}\}\rangle$

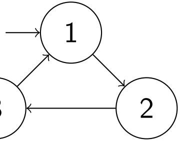
Execution paths



Definition

A path is an infinite sequence of states. $Paths(s)$ is the set of path starting in state s.

Execution paths



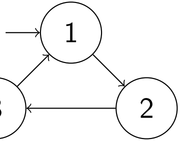
Definition

A path is an infinite sequence of states. *Paths*(s) is the set of path starting in state s.

Question

What is *Paths*(2)?

Execution paths



Definition

A path is an infinite sequence of states. $\mathit{Paths}(s)$ is the set of path starting in state s.

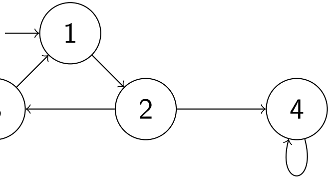
Question

What is $\mathit{Paths}(2)$?

Answer

$\mathit{Paths}(2) = \{231231231\ldots\}$

Execution paths



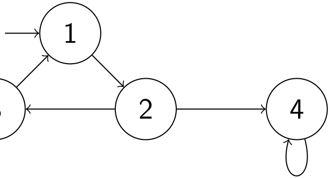
Definition

A path is an infinite sequence of states. $Paths(s)$ is the set of path starting in state s.

Question

What is $Paths(2)$?

Execution paths



Definition

A path is an infinite sequence of states. $Paths(s)$ is the set of path starting in state s.

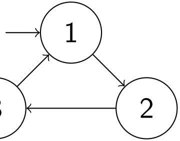
Question

What is $Paths(2)$?

Answer

$Paths(2) =$
$\{2444\ldots, 2312444\ldots, 2312312444\ldots, \ldots, 231231\ldots\}$

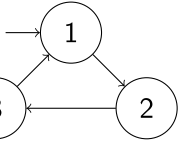
Execution paths



Definition

Let $p \in Paths(s)$ and $n \geq 0$. Then $p[n]$ is the $(n+1)$th state of the path p.

Execution paths



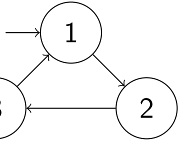
Definition

Let $p \in Paths(s)$ and $n \geq 0$. Then $p[n]$ is the $(n+1)$th state of the path p.

Question

Let $p = 123123\ldots$. What is $p[3]$?

Execution paths



Definition

Let $p \in \mathit{Paths}(s)$ and $n \geq 0$. Then $p[n]$ is the $(n+1)$th state of the path p.

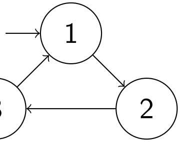
Question

Let $p = 123123\ldots$. What is $p[3]$?

Answer

$p[3] = 1$.

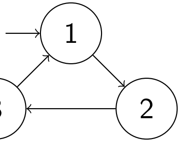
Execution paths



Definition

Let $p \in \mathit{Paths}(s)$ and $n \geq 0$. Then $p[n..]$ is the suffix starting with the $(n+1)$th state of the path p, that is, removing the first n states.

Execution paths



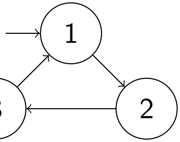
Definition

Let $p \in Paths(s)$ and $n \geq 0$. Then $p[n..]$ is the suffix starting with the $(n+1)$th state of the path p, that is, removing the first n states.

Question

Let $p = 123123\ldots$. What is $p[2..]$?

Execution paths



Definition

Let $p \in Paths(s)$ and $n \geq 0$. Then $p[n..]$ is the suffix starting with the $(n+1)$th state of the path p, that is, removing the first n states.

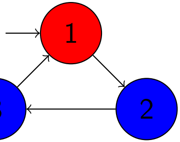
Question

Let $p = 123123\ldots$. What is $p[2..]$?

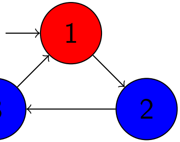
Answer

$p[2..] = 312312\ldots$.

Semantics of LTL



$p \models f$ denotes that path p satisfies LTL formula f.

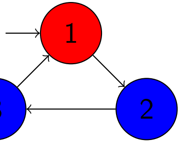


$p \models f$ denotes that path p satisfies LTL formula f.

Question

$123123\ldots \models$ blue?

Semantics of LTL



$p \models f$ denotes that path p satisfies LTL formula f.

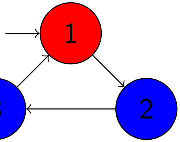
Question

$123123\ldots \models$ blue?

Answer

No.

Semantics of LTL

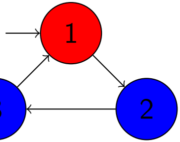


$p \models f$ denotes that path p satisfies LTL formula f.

Question

$123123\ldots \models \bigcirc \text{blue}$?

Semantics of LTL



$p \models f$ denotes that path p satisfies LTL formula f.

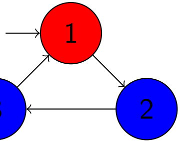
Question

$123123\ldots \models \bigcirc$blue?

Answer

Yes.

Semantics of LTL

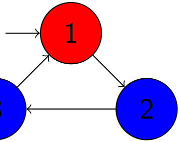


$p \models f$ denotes that path p satisfies LTL formula f.

Question

$123123\ldots \models \text{red} \land \bigcirc \text{blue}$?

Semantics of LTL



$p \models f$ denotes that path p satisfies LTL formula f.

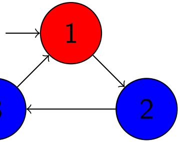
Question

$123123\ldots \models$ red $\wedge \bigcirc$blue?

Answer

Yes.

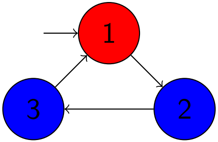
Semantics of LTL



$p \models f$ denotes that path p satisfies LTL formula f.

Question

$123123\ldots \models \neg$blue?



$p \models f$ denotes that path p satisfies LTL formula f.

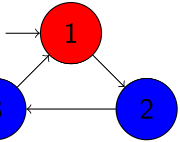
Question

$123123\ldots \models \neg$blue?

Answer

Yes.

Semantics of LTL

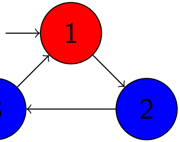


$p \models f$ denotes that path p satisfies LTL formula f.

Question

$123123\ldots \models$ red U blue?

Semantics of LTL



$p \models f$ denotes that path p satisfies LTL formula f.

Question

$123123\ldots \models$ red U blue?

Answer

Yes.

Semantics of LTL

Definition

$$
\begin{aligned}
p \models a &\text{ iff } a \in \ell(p[0]) \\
p \models f \wedge g &\text{ iff } p \models f \wedge p \models g \\
p \models \neg f &\text{ iff } p \not\models f \\
p \models \bigcirc f &\text{ iff } p[1..] \models f \\
p \models f \mathbin{\mathrm{U}} g &\text{ iff } \exists i \geq 0 : p[i..] \models g \wedge \forall 0 \leq j < i : p[j..] \models f
\end{aligned}
$$

Semantics of LTL

Question

How can we express $p \models \Diamond f$ in terms of $\cdots \models f$?

Semantics of LTL

Question

How can we express $p \models \Diamond f$ in terms of $\cdots \models f$?

Answer

$$
\begin{aligned}
& p \models \Diamond f \\
& \qquad \text{iff } p \models \text{true U } f \\
& \qquad \text{iff } \exists i \geq 0 : p[i..] \models f \wedge \forall 0 \leq j < i : p[j..] \models \text{true} \\
& \qquad \text{iff } \exists i \geq 0 : p[i..] \models f
\end{aligned}
$$

Semantics of LTL

Question

How can we express $p \models \Box f$ in terms of $\cdots \models f$?

Semantics of LTL

Question

How can we express $p \models \Box f$ in terms of $\cdots \models f$?

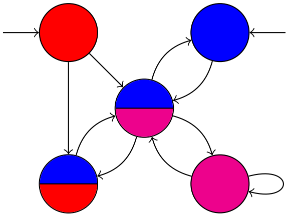
Answer

$$
\begin{aligned}
& p \models \Box f \\
& \qquad \text{iff } p \models \neg \Diamond \neg f \\
& \qquad \text{iff } \neg(\exists i \geq 0 : p[i..] \models \neg f) \\
& \qquad \text{iff } \forall i \geq 0 : p[i..] \models f
\end{aligned}
$$

Semantics of LTL

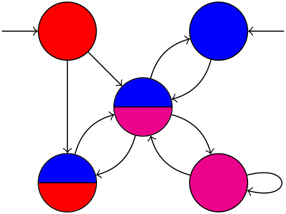
Let $TS = \langle S, L, I, \rightarrow, \ell \rangle$ be a transition system. Then

$$TS \models f \text{ iff } \forall s \in I : \forall p \in Paths(s) : p \models f$$



Question

$TS \models$ blue?



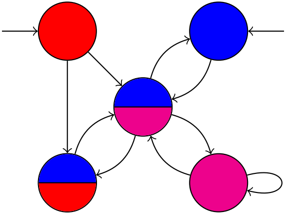
Question

$TS \models$ blue?

Answer

No.

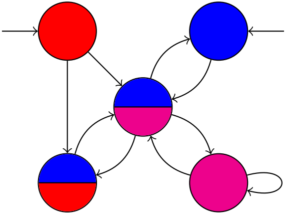
Semantics of LTL



Question

$TS \models \text{red} \vee \text{blue}$?

Semantics of LTL



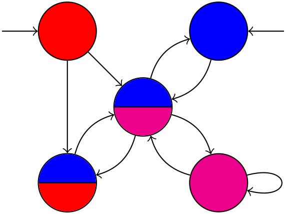
Question

$TS \models$ red $\vee$ blue?

Answer

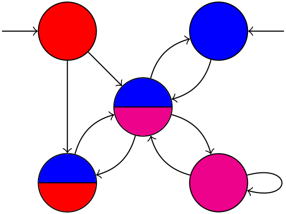
Yes.

Semantics of LTL



Question

$TS \models \bigcirc$blue?

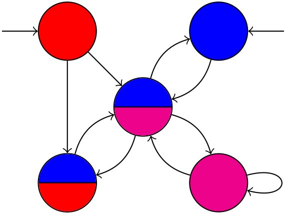


Question

$TS \models \bigcirc$blue?

Answer

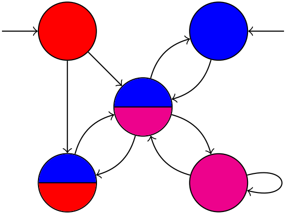
Yes.



Question

$TS \models$ red U blue?

Semantics of LTL



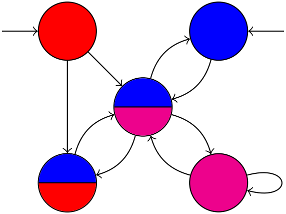
Question

$TS \models$ red U blue?

Answer

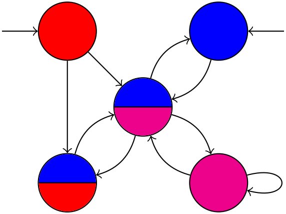
Yes.

Semantics of LTL



Question

$TS \models \Diamond$magenta?



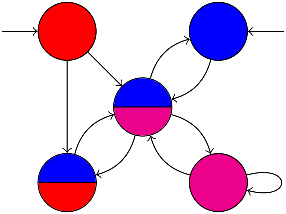
Question

$TS \models \Diamond \text{magenta}$?

Answer

Yes.

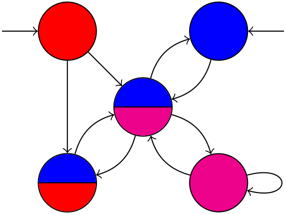
Semantics of LTL



Question

$TS \models \Box\Diamond \text{blue}$?

Semantics of LTL



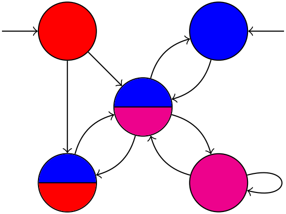
Question

$TS \models \Box\Diamond \text{blue}$?

Answer

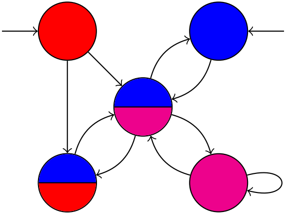
No.

Semantics of LTL



Question

$TS \models \Box(\neg\text{blue} \Rightarrow \bigcirc(\text{magenta} \vee \text{red}))$?



Question

$TS \models \square(\neg \text{blue} \Rightarrow \bigcirc(\text{magenta} \vee \text{red}))$?

Answer

Yes.

LTL and JPF

Atomic propositions may be used to express properties of JPF's virtual machine's state, such as

- initial states,
- final states,
- the values of attributes (for example, a boolean attribute being true, or an integer attribute being positive),
- the values of local variables (for example, a boolean local variable being true, or an integer local variable being positive)
- method invocations,
- method returns,
- etc.

LTL and JPF

Atomic propositions may be used to express properties of JPF's virtual machine's state, such as

- initial states,
- final states,
- the values of attributes (for example, a boolean attribute being true, or an integer attribute being positive),
- the values of local variables (for example, a boolean local variable being true, or an integer local variable being positive)
- method invocations,
- method returns,
- etc.

The extensions bitbucket.org/petercipov/jpf-ltl and bitbucket.org/michelelombardi/jpf-ltl of JPF support LTL, but neither is stable.

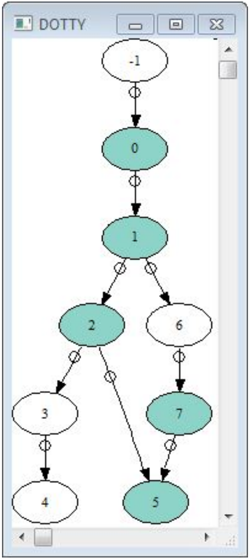
LTL and JPF

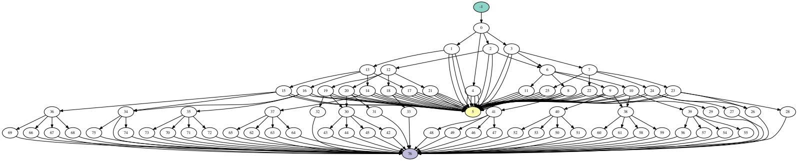
The extension jpf-label provides an easy way to label states. For example, consider the following app.

```
import java.util.Random;

public class Field {
  private static boolean value = true;

  public static void main(String[] args) {
    Random random = new Random();
    if (random.nextBoolean()) {
      Field.value = false;
      Field.value = true;
    } else {
      Field.value = random.nextBoolean();
    }
  }
}
```

```
target = Field
classpath = <path to the directory containing Field.class>
cg.enumerate_random = true

@using jpf-label
listener = label.StateSpaceDot
label.class = label.BooleanStaticField
label.BooleanStaticField.field = Field.value
```

jpf-label




-1
0
1
2
3
13
12
6
7
15
16
19
20
14
18
17
21
4
11
25
8
22
9
10
24
23
36
34
35
37
32
30
31
33
5
41
40
38
39
29
27
26
28
69
66
67
68
75
74
73
70
71
72
65
62
63
64
43
44
45
42
48
49
46
47
52
53
50
51
60
61
58
59
56
57
54
55
76

jpf-label

The following classes to label states. For some classes an additional property needs to be specified as indicated below.

- `Initial`: labels the initial state.
- `End`: labels the final states.
- `AllDifferent`: labels each state with a different label.
- `BooleanStaticField`: labels those states in which the static boolean field specified by the property `label.StaticBooleanField.field` is true.
- `PositiveIntegerLocalVariable`: labels those states in which the local integer variable specified by the property `label.LocalPositiveIntegerVariable.variable` is positive.
- `InvokedStaticMethod`: labels those states in which the method specified by the property `label.InvokedStaticMethod.method` is invoked.

jpf-label

- `ReturnedVoidMethod`: labels those states in which the void method specified by the property `label.ReturnedVoidMethod.method` has returned.
- `ThrownException`: labels those states in which an exception of the type specified by the property `label.ThrownException.type` has been thrown.
- `SynchronizedStaticMethod`: labels those states in which the synchronized method specified by the property `label.SynchronizedStaticMethod.method` acquires and has released the lock.

Project suggestions

- Implement additional properties for jpf-label.
- Improve error messages when class cannot be found on `classpath` or `native_classpath`.
- Improve a listener as in the sample project proposal (choose a listener that has not been considered in the past).