4507/ 6507 Software and Hardware Verification
Introduction to LTL

Andrei Popescu

University of Sheffield

These slides contain material from Denisa Diaconescu, Georg Struth and Traian Florin Șerbănuță

LTL

LTL = Linear(-time) Temporal Logic

Introduced into computer science by Amir Pnueli in 1977

A logic for reasoning about execution paths of systems

One of the most important logics for software and hardware verification

Overview

Syntax: LTL formulas

Semantics: labeled transition systems

Practical specification patterns

Formula equivalence

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
        else x++;
}
```

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
           else x++;
}
```

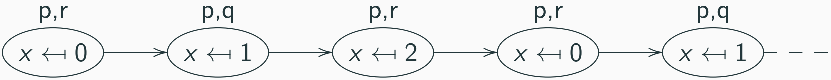
Let p be "prints hello",
q be "prints hi",
r be "x is even".
Say we start in a state where x is 0.

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
        else x++;
}
```

Let p be “prints hello”,
q be “prints hi”,
r be “x is even”.
Say we start in a state where x is 0.

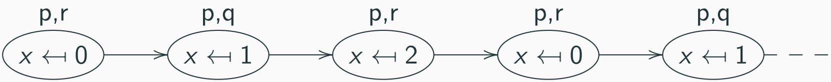


Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
            else x++;
}
```

Let p be "prints hello",
q be "prints hi",
r be "x is even".
Say we start in a state where x is 0.



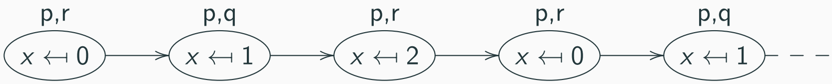
Always p holds.

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
          else x++;
}
```

Let p be "prints hello",
q be "prints hi",
r be "x is even".
Say we start in a state where x is 0.



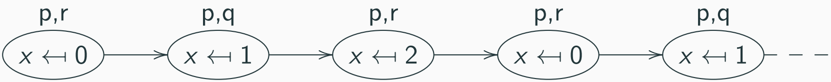
Always p holds. Always [p implies (q or r)].

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
        else x++;
}
```

Let p be "prints hello",
q be "prints hi",
r be "x is even".
Say we start in a state where x is 0.



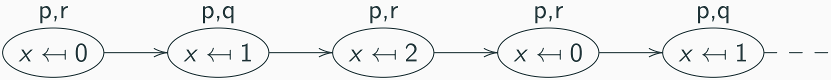
Always p holds. Always [p implies (q or r)].
Never (q and r) holds.

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
            else x++;
}
```

Let p be "prints hello",
q be "prints hi",
r be "x is even".
Say we start in a state where x is 0.



Always p holds. Always [p implies (q or r)].

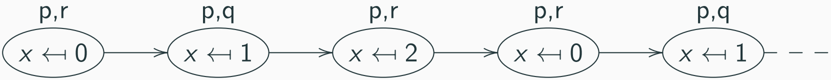
Never (q and r) holds. Always eventually q holds.

Basic Intuition

- Consider execution paths of a system into the future.
- Label states with atomic propositions $p, q, r, \ldots$ that hold along paths at various points in time.
- LTL formulas can express regular patterns about these propositions as execution proceeds.

```
while (x < 3) {
    print("hello");
    if (x == 1) print("hi");
    if (x == 2) x = 0;
          else x++;
}
```

Let p be "prints hello",
q be "prints hi",
r be "x is even".
Say we start in a state where x is 0.



Always p holds. Always [p implies (q or r)].

Never (q and r) holds. Always eventually q holds.

Can you think of other patterns?

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

LTL formulas, usually denoted φ, ψ etc., are defined as follows:

$$\varphi \;::=\; p \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi \,\mathsf{U}\, \psi$$

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

LTL formulas, usually denoted φ, ψ etc., are defined as follows:

$$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi \,\mathsf{U}\, \psi$$

Examples: $\Box(p \,\mathsf{U}\, (q \,\mathsf{U}\, r))$

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

LTL formulas, usually denoted φ, ψ etc., are defined as follows:

$$\varphi \;::=\; p \mid \neg\,\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \bigcirc\varphi \mid \Diamond\,\varphi \mid \Box\,\varphi \mid \varphi \,\mathsf{U}\, \psi$$

Examples: $\Box(p\,\mathsf{U}\,(q\,\mathsf{U}\,r))$ $\qquad \Box\Diamond p$

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

LTL formulas, usually denoted φ, ψ etc., are defined as follows:

$$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi \,\mathsf{U}\, \psi$$

Examples: $\Box(p \,\mathsf{U}\, (q \,\mathsf{U}\, r))$ $\qquad \Box\Diamond p$

$\neg$, $\vee$, $\wedge$, $\rightarrow$ are propositional connectives: "not", "or", "and", "implies".

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

LTL formulas, usually denoted φ, ψ etc., are defined as follows:

$$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi \, \mathsf{U} \, \psi$$

Examples: $\Box(p \, \mathsf{U} \, (q \, \mathsf{U} \, r))$ $\Box\Diamond p$

$\neg$, $\vee$, $\wedge$, $\rightarrow$ are propositional connectives: "not", "or", "and", "implies".

$\bigcirc$, $\Diamond$, $\Box$, U are temporal connectives: "next", "eventually". "always", "until".

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

LTL formulas, usually denoted φ, ψ etc., are defined as follows:

$$\varphi \ ::= \ p \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi \, \mathsf{U} \, \psi$$

Examples: $\Box(p \, \mathsf{U} \, (q \, \mathsf{U} \, r))$ $\qquad \Box\Diamond p$

$\neg$, $\vee$, $\wedge$, $\rightarrow$ are propositional connectives: "not", "or", "and", "implies".

$\bigcirc$, $\Diamond$, $\Box$, U are temporal connectives: "next", "eventually". "always", "until".

Pronunciation:

- $\bigcirc\varphi$ – Next φ
- $\Diamond\varphi$ – Eventually φ
- $\Box\varphi$ – Always φ
- $\varphi \, \mathsf{U} \, \psi$ – φ Until ψ

Syntax

Assume some set *Atoms* of atomic propositions (atoms for short) usually denoted p, q, r etc.

LTL formulas, usually denoted φ, ψ etc., are defined as follows:

$$\varphi \ ::= \ p \mid \neg \varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \bigcirc \varphi \mid \Diamond \varphi \mid \Box \varphi \mid \varphi \, \mathsf{U} \, \psi$$

Examples: $\Box(p \, \mathsf{U} \, (q \, \mathsf{U} \, r))$ $\Box\Diamond p$

$\neg$, $\vee$, $\wedge$, $\rightarrow$ are propositional connectives: "not", "or", "and", "implies".

$\bigcirc$, $\Diamond$, $\Box$, U are temporal connectives: "next", "eventually". "always", "until".

Pronunciation:

- $\bigcirc\varphi$ – Next φ
- $\Diamond\varphi$ – Eventually φ
- $\Box\varphi$ – Always φ
- $\varphi \, \mathsf{U} \, \psi$ – φ Until ψ

The unary connectives $\neg, \bigcirc, \Diamond, \Box$ have higher precedence than the binary connectives $\wedge, \vee, \rightarrow, \mathsf{U}$. E.g., $\Box\varphi \vee \psi$ is the same as $(\Box\varphi) \vee \psi$.

Syntax – Examples and Non-Examples

The following are LTL formulas:

- $(\Diamond p \wedge \Box q) \rightarrow (p \; \mathsf{U} \; r)$
- $\Diamond(p \rightarrow \Box r) \vee (\neg q \; \mathsf{U} \; p)$
- $p \; \mathsf{U} \; (q \; \mathsf{U} \; r)$
- $\Box \Diamond p \rightarrow \Diamond(q \vee s)$

Syntax – Examples and Non-Examples

The following are LTL formulas:

- $(\Diamond p \wedge \Box q) \rightarrow (p \; \mathsf{U} \; r)$
- $\Diamond(p \rightarrow \Box r) \vee (\neg q \; \mathsf{U} \; p)$
- $p \; \mathsf{U} \; (q \; \mathsf{U} \; r)$
- $\Box\Diamond p \rightarrow \Diamond(q \vee s)$

The following are not LTL formulas:

- $\mathsf{U} \; r$
- $q \; \Box \; p$

Syntax – Examples and Non-Examples

The following are LTL formulas:

- $(\Diamond p \wedge \Box q) \rightarrow (p \; \mathrm{U} \; r)$
- $\Diamond(p \rightarrow \Box r) \vee (\neg q \; \mathrm{U} \; p)$
- $p \; \mathrm{U} \; (q \; \mathrm{U} \; r)$
- $\Box\Diamond p \rightarrow \Diamond(q \vee s)$

The following are not LTL formulas:

- $\mathrm{U} \; r$
- $q \; \Box \; p$

Exercise. 1. Give five more examples of correctly constructed formulas. Include a formula that contains five atoms p, q, r, u, v, and a formula that contains three occurrences of $\Diamond$, one occurrence of $\Box$ and two occurrences of U. Read aloud the formulas that you have constructed.

Syntax – Examples and Non-Examples

The following are LTL formulas:

- $(\Diamond p \wedge \Box q) \rightarrow (p \; \mathrm{U} \; r)$
- $\Diamond(p \rightarrow \Box r) \vee (\neg q \; \mathrm{U} \; p)$
- $p \; \mathrm{U} \; (q \; \mathrm{U} \; r)$
- $\Box\Diamond p \rightarrow \Diamond(q \vee s)$

The following are not LTL formulas:

- $\mathrm{U} \; r$
- $q \; \Box \; p$

Exercise. 1. Give five more examples of correctly constructed formulas. Include a formula that contains five atoms p, q, r, u, v, and a formula that contains three occurrences of $\Diamond$, one occurrence of $\Box$ and two occurrences of U. Read aloud the formulas that you have constructed.

2. Give two examples of incorrectly constructed formulas that do not contain U or $\Box$.

Informal Semantics

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

LTL formulas are evaluated along this path, looking into the future:

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

LTL formulas are evaluated along this path, looking into the future:

- An atomic proposition p holds if p is true at the current point in time.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

LTL formulas are evaluated along this path, looking into the future:

- An atomic proposition p holds if p is true at the current point in time.
- The propositional connectives $\neg, \vee, \wedge \rightarrow$ have their usual meanings, e.g., $\varphi \wedge \psi$ holds if φ holds and ψ holds.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

LTL formulas are evaluated along this path, looking into the future:

- An atomic proposition *p* holds if *p* is true at the current point in time.
- The propositional connectives $\neg, \vee, \wedge \rightarrow$ have their usual meanings, e.g., $\varphi \wedge \psi$ holds if φ holds and ψ holds.
- Meaning of temporal connectives:
  - $\bigcirc\varphi$ holds if φ holds next, i.e., at the next point in time.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

LTL formulas are evaluated along this path, looking into the future:

- An atomic proposition p holds if p is true at the current point in time.
- The propositional connectives $\neg, \vee, \wedge \rightarrow$ have their usual meanings, e.g., $\varphi \wedge \psi$ holds if φ holds and ψ holds.
- Meaning of temporal connectives:
    - $\bigcirc\varphi$ holds if φ holds next, i.e., at the next point in time.
    - $\Diamond\varphi$ holds if φ holds eventually, i.e., now or at some future point in time.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

LTL formulas are evaluated along this path, looking into the future:

- An atomic proposition *p* holds if *p* is true at the current point in time.
- The propositional connectives $\neg, \vee, \wedge \rightarrow$ have their usual meanings, e.g., $\varphi \wedge \psi$ holds if φ holds and ψ holds.
- Meaning of temporal connectives:
  - $\bigcirc\varphi$ holds if φ holds next, i.e., at the next point in time.
  - $\Diamond\varphi$ holds if φ holds eventually, i.e., now or at some future point in time.
  - $\Box\varphi$ holds if φ holds always, i.e., now and at all future points in time.

Informal Semantics

We model time as the stream of natural numbers: $0, 1, 2, \ldots$.

We consider an infinite execution path, which at every point in time reaches a given state.

For every state on the path, we assume to know which atomic propositions are true in that state.

LTL formulas are evaluated along this path, looking into the future:

- An atomic proposition *p* holds if *p* is true at the current point in time.
- The propositional connectives $\neg, \vee, \wedge \rightarrow$ have their usual meanings, e.g., $\varphi \wedge \psi$ holds if φ holds and ψ holds.
- Meaning of temporal connectives:
  - $\bigcirc\varphi$ holds if φ holds next, i.e., at the next point in time.
  - $\Diamond\varphi$ holds if φ holds eventually, i.e., now or at some future point in time.
  - $\Box\varphi$ holds if φ holds always, i.e., now and at all future points in time.
  - $\varphi \, \mathsf{U} \, \psi$ holds if φ holds until ψ holds; i.e., ψ holds now or at some point in the future, and φ holds continuously until then.

Informal Semantics – Examples

We assume that *enabled*, *read*, *write*, etc. are all atoms.

By “further up in the future” we will mean “at the current time or later”.

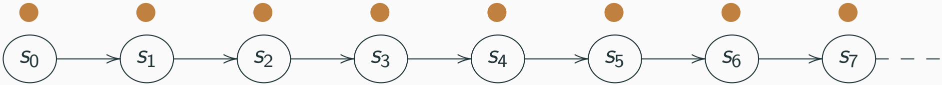
Informal Semantics – Examples

We assume that *enabled*, *read*, *write*, etc. are all atoms.

By "further up in the future" we will mean "at the current time or later".

$\Box$ *enabled* means:

enabled holds always, i.e., now and at all points in the future.



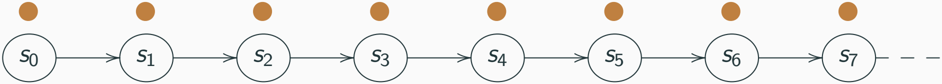
Informal Semantics – Examples

We assume that *enabled*, *read*, *write*, etc. are all atoms.

By "further up in the future" we will mean "at the current time or later".

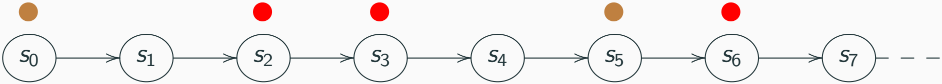
$\Box$ *enabled* means:

enabled holds always, i.e., now and at all points in the future.



$\Box \neg(read \wedge write)$ means:

Always (i.e., now and at all points in the future), it is not the case that *read* and *write* hold. In other words: It is never the case that *read* and *write* hold at the same time.



Informal Semantics – Examples

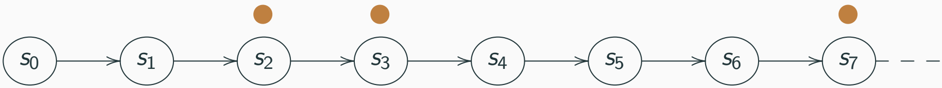
$\Box\Diamond$*enabled* means:

Always eventually *enabled* holds. In other words: Now and for all future points, there is a point further up in the future where *enabled* holds. Another way to say this: *enabled* holds infinitely often.

Informal Semantics – Examples

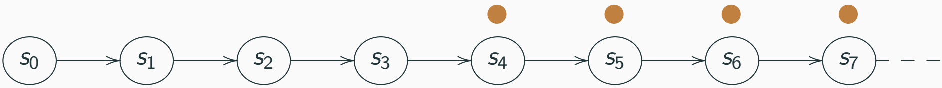
$\Box\Diamond$*enabled* means:

Always eventually *enabled* holds. In other words: Now and for all future points, there is a point further up in the future where *enabled* holds. Another way to say this: *enabled* holds infinitely often.



$\Diamond\Box$ *enabled* means:

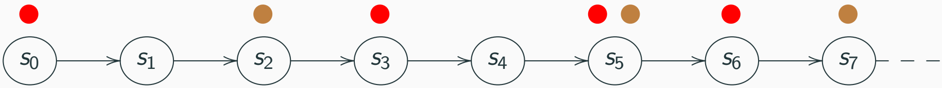
Eventually always *enabled* holds. In other words: Starting now or from a future point, *enabled* will hold continuously for all points in the future.



Informal Semantics – Examples

$\Box\,(request \rightarrow \Diamond grant)$ means:

Always [*request* implies eventually *grant*]. In other words: Always (i.e., now and at all points in the future), if *request* holds then eventually *grant* holds (i.e., there exists a point further up in the future where *grant* holds).



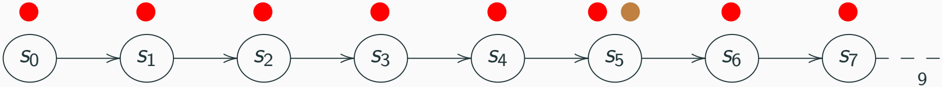
Informal Semantics – Examples

$\Box\,(request \rightarrow \Diamond grant)$ means:

Always [*request* implies eventually *grant*]. In other words: Always (i.e., now and at all points in the future), if *request* holds then eventually *grant* holds (i.e., there exists a point further up in the future where *grant* holds).

$(\Box\, request) \rightarrow (\Diamond grant)$ means:

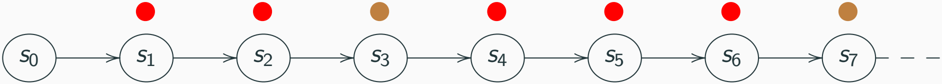
[Always *request*] implies [eventually *grant*]. In other words: If *request* holds at all points in time, then *grant* holds at some point in time.



Informal Semantics – Examples

$\square\,(request \rightarrow (request \,\mathsf{U}\, grant))$ means:

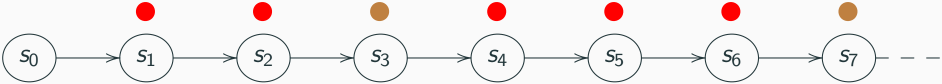
Always, *request* implies [*request* until *grant*]. In other words: At every point in the future, if *request* holds than here exists a point further up in the future where *grant* holds, and *request* holds continuously until that point.



Informal Semantics – Examples

$\Box\,(request \rightarrow (request\,\mathsf{U}\,grant))$ means:

Always, *request* implies [*request* until *grant*]. In other words: At every point in the future, if *request* holds than here exists a point further up in the future where *grant* holds, and *request* holds continuously until that point.



Exercise. Consider the following LTL formulas:

(a) $\Box\,(request\,\mathsf{U}\,grant)$

(b) $\Box\,\Diamond\,(request \rightarrow grant)$

(c) $\Box\,\Diamond\,request \rightarrow \Box\,\Diamond\,grant$

(d) $\Box\,\Diamond\,\Box\,enabled$

1. What is the correct way to parenthesize the point (c) formula, based on the operator precedence?

2. Depict graphically the meaning of these formulas. What is the difference between the point (d) formula and $\Diamond\,\Box\,enabled$?

Informal Semantics – Examples

Exercise. Consider the following LTL formulas:

(a) $\Box\,(\mathit{request}\ \mathsf{U}\ \mathit{grant})$

(b) $\Box\,\Diamond\,(\mathit{request} \rightarrow \mathit{grant})$

(c) $\Box\,\Diamond\,\mathit{request} \rightarrow \Box\,\Diamond\,\mathit{grant}$

(d) $\Box\,\Diamond\,\Box\,\mathit{enabled}$

1. What is the correct way to parenthesize the point (c) formula, based on the operator precedence?

2. Depict graphically the meaning of these formulas. What is the difference between the point (d) formula and $\Diamond\,\Box\,\mathit{enabled}$?

Practical Specification Patterns

- A process is always active in its starting state:

$$\Box\,(\mathit{start} \rightarrow \mathit{active})$$

Practical Specification Patterns

- A process is always active in its starting state:

$$\Box\,(\mathit{start} \rightarrow \mathit{active})$$

- It is always the case that requests are eventually granted:

$$\Box\,(\mathit{request} \rightarrow \Diamond \mathit{grant})$$

Practical Specification Patterns

- A process is always active in its starting state:

$$\square(\mathit{start} \rightarrow \mathit{active})$$

- It is always the case that requests are eventually granted:

$$\square(\mathit{request} \rightarrow \lozenge \mathit{grant})$$

- A given process will be enabled infinitely often:

$$\square\lozenge \mathit{enabled}$$

Practical Specification Patterns

- A process is always active in its starting state:

$$\Box\,(start \rightarrow active)$$

- It is always the case that requests are eventually granted:

$$\Box\,(request \rightarrow \Diamond grant)$$

- A given process will be enabled infinitely often:

$$\Box\Diamond enabled$$

- If a process is enabled infinitely often, then it will run infinitely often:

$$\Box\Diamond enabled \rightarrow \Box\Diamond run$$

Practical Specification Patterns

- A process will never become permanently inactive:

$$\neg \Diamond \Box \neg \mathit{active}$$

Practical Specification Patterns

- A process will never become permanently inactive:

$$\neg\Diamond\Box\neg\, \mathit{active}$$

- It is always the case that, when a lift is at the 2nd floor, travels upwards and the 5th floor is requested, it will not change direction until the 5th floor is reached:

Practical Specification Patterns

- A process will never become permanently inactive:

$$\neg\Diamond\Box\neg\, \mathit{active}$$

- It is always the case that, when a lift is at the 2nd floor, travels upwards and the 5th floor is requested, it will not change direction until the 5th floor is reached:

$$\Box(@2 \wedge \mathsf{upgoing} \wedge \mathsf{pressed5} \rightarrow (\mathsf{upgoing}\,\mathsf{U}\,@5))$$

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Let π be an infinite sequence of states $s_0 s_1 s_2 \ldots$. We think of $L(s_i)$ as the set of all atoms true at point i in time on π.

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Let π be an infinite sequence of states $s_0 s_1 s_2 \ldots$. We think of $L(s_i)$ as the set of all atoms true at point i in time on π.

For each i, we write π^i for the i'th suffix of π, namely $s_i s_{i+1} s_{i+2} \ldots$.

E.g., π^1 is $s_1 s_2 s_3 \ldots$ and π^2 is $s_2 s_3 s_4 \ldots$

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Let π be an infinite sequence of states $s_0 s_1 s_2 \ldots$. We think of $L(s_i)$ as the set of all atoms true at point i in time on π.

For each i, we write π^i for the i'th suffix of π, namely $s_i s_{i+1} s_{i+2} \ldots$.

E.g., π^1 is $s_1 s_2 s_3 \ldots$ and π^2 is $s_2 s_3 s_4 \ldots$

For an LTL formula φ, we define $\pi \models_L \varphi$, read *"π satisfies φ w.r.t. labeling L"* or *"φ holds for π w.r.t. labeling L"* by structural recursion on φ:

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Let π be an infinite sequence of states $s_0 s_1 s_2 \ldots$. We think of $L(s_i)$ as the set of all atoms true at point i in time on π.

For each i, we write π^i for the i'th suffix of π, namely $s_i s_{i+1} s_{i+2} \ldots$.

E.g., π^1 is $s_1 s_2 s_3 \ldots$ and π^2 is $s_2 s_3 s_4 \ldots$

For an LTL formula φ, we define $\pi \models_L \varphi$, read *"π satisfies φ w.r.t. labeling L"* or *"φ holds for π w.r.t. labeling L"* by structural recursion on φ:

$\pi \models_L p$ iff $p \in L(s_0)$

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Let π be an infinite sequence of states $s_0 s_1 s_2 \ldots$. We think of $L(s_i)$ as the set of all atoms true at point i in time on π.

For each i, we write π^i for the i'th suffix of π, namely $s_i s_{i+1} s_{i+2} \ldots$.

E.g., π^1 is $s_1 s_2 s_3 \ldots$ and π^2 is $s_2 s_3 s_4 \ldots$

For an LTL formula φ, we define $\pi \models_L \varphi$, read *"π satisfies φ w.r.t. labeling L"* or *"φ holds for π w.r.t. labeling L"* by structural recursion on φ:

$\pi \models_L p$	iff	$p \in L(s_0)$
$\pi \models_L \varphi \wedge \psi$	iff	$\pi \models_L \varphi$ and $\pi \models_L \psi$

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Let π be an infinite sequence of states $s_0 s_1 s_2 \ldots$. We think of $L(s_i)$ as the set of all atoms true at point i in time on π.

For each i, we write π^i for the i'th suffix of π, namely $s_i s_{i+1} s_{i+2} \ldots$.

E.g., π^1 is $s_1 s_2 s_3 \ldots$ and π^2 is $s_2 s_3 s_4 \ldots$

For an LTL formula φ, we define $\pi \models_L \varphi$, read *"π satisfies φ w.r.t. labeling L"* or *"φ holds for π w.r.t. labeling L"* by structural recursion on φ:

$\pi \models_L p$	iff	$p \in L(s_0)$
$\pi \models_L \varphi \wedge \psi$	iff	$\pi \models_L \varphi$ and $\pi \models_L \psi$
$\pi \models_L \varphi \vee \psi$	iff	$\pi \models_L \varphi$ or $\pi \models_L \psi$

Formal Semantics

Let S be a set of states and $L : S \to \mathcal{P}(Atoms)$ be a labeling function associating to each state s a set $L(s)$ of all atoms that are true in that state.
Note: $\mathcal{P}(Atoms)$ is the powerset (i.e., set of all subsets) of $Atoms$.

Let π be an infinite sequence of states $s_0 s_1 s_2 \ldots$. We think of $L(s_i)$ as the set of all atoms true at point i in time on π.

For each i, we write π^i for the i'th suffix of π, namely $s_i s_{i+1} s_{i+2} \ldots$.

E.g., π^1 is $s_1 s_2 s_3 \ldots$ and π^2 is $s_2 s_3 s_4 \ldots$

For an LTL formula φ, we define $\pi \models_L \varphi$, read *"π satisfies φ w.r.t. labeling L"* or *"φ holds for π w.r.t. labeling L"* by structural recursion on φ:

$\pi \models_L p$	iff	$p \in L(s_0)$
$\pi \models_L \varphi \wedge \psi$	iff	$\pi \models_L \varphi$ and $\pi \models_L \psi$
$\pi \models_L \varphi \vee \psi$	iff	$\pi \models_L \varphi$ or $\pi \models_L \psi$
$\pi \models_L \varphi \to \psi$	iff	$\pi \models_L \varphi$ implies $\pi \models_L \psi$

Formal Semantics

$$\pi \models_L \bigcirc \varphi \qquad \text{iff} \quad \pi^1 \models_L \varphi$$

Formal Semantics

$\pi \models_L \bigcirc\varphi$	iff	$\pi^1 \models_L \varphi$
$\pi \models_L \Diamond\varphi$	iff	there exists $i \geq 0$ such that $\pi^i \models_L \varphi$

Formal Semantics

$\pi \models_L \bigcirc\varphi$	iff	$\pi^1 \models_L \varphi$
$\pi \models_L \Diamond\varphi$	iff	there exists $i \geq 0$ such that $\pi^i \models_L \varphi$
$\pi \models_L \Box\varphi$	iff	for all $i \geq 0$ we have $\pi^i \models_L \varphi$

Formal Semantics

$$
\begin{array}{lll}
\pi \models_L \bigcirc\varphi & \text{iff} & \pi^1 \models_L \varphi \\
\pi \models_L \Diamond\varphi & \text{iff} & \text{there exists } i \geq 0 \text{ such that } \pi^i \models_L \varphi \\
\pi \models_L \Box\varphi & \text{iff} & \text{for all } i \geq 0 \text{ we have } \pi^i \models_L \varphi \\
\pi \models_L \varphi \,\mathsf{U}\, \psi & \text{iff} & \text{there exists } i \geq 0 \text{ such that } \pi^i \models_L \psi \text{ and} \\
 & & \text{for all } j \in \{0, \ldots, i-1\} \text{ we have } \pi^j \models_L \varphi
\end{array}
$$

Formal Semantics

$\pi \models_L \bigcirc\varphi$	iff	$\pi^1 \models_L \varphi$
$\pi \models_L \Diamond\varphi$	iff	there exists $i \geq 0$ such that $\pi^i \models_L \varphi$
$\pi \models_L \Box\varphi$	iff	for all $i \geq 0$ we have $\pi^i \models_L \varphi$
$\pi \models_L \varphi \, \mathsf{U} \, \psi$	iff	there exists $i \geq 0$ such that $\pi^i \models_L \psi$ and for all $j \in \{0, \ldots, i-1\}$ we have $\pi^j \models_L \varphi$

$\models$ is called the satisfaction relation. It is a relation between formulas and infinite sequences of states in the presence of a state labeling with atom sets.

Formal Semantics

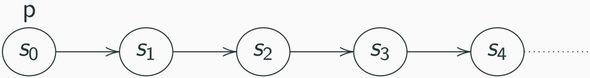
$\pi \models_L \bigcirc\varphi$	iff	$\pi^1 \models_L \varphi$
$\pi \models_L \Diamond\varphi$	iff	there exists $i \geq 0$ such that $\pi^i \models_L \varphi$
$\pi \models_L \Box\varphi$	iff	for all $i \geq 0$ we have $\pi^i \models_L \varphi$
$\pi \models_L \varphi \,\mathsf{U}\, \psi$	iff	there exists $i \geq 0$ such that $\pi^i \models_L \psi$ and for all $j \in \{0, \ldots, i-1\}$ we have $\pi^j \models_L \varphi$

$\models$ is called the satisfaction relation. It is a relation between formulas and infinite sequences of states in the presence of a state labeling with atom sets.

When the labeling L is fixed, we can write $\pi \models \varphi$ instead of $\pi \models_L \varphi$.

Semantics of Atoms Illustrated

$\pi \models p$



Semantics of "Next" Illustrated

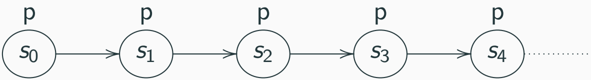
$\pi \models \bigcirc p$

Semantics of "Eventually" Illustrated

$\pi \models \lozenge p$

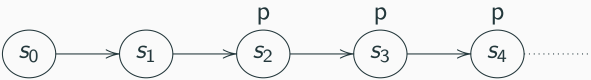
Semantics of "Always" Illustrated

$\pi \models \Box p$



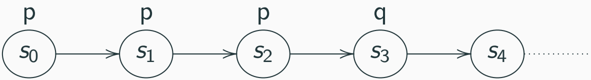
Combined Semantics of "Eventually" and "Always" Illustrated

$\pi \models \Diamond \Box p$



Semantics of "Until" Illustrated

$\pi \models p \, \mathsf{U} \, q$



Exercises

Transition Systems and Paths

Transition Systems and Paths

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

Transition Systems and Paths

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states

Transition Systems and Paths

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states
- $\rightarrow \subseteq S \times S$ a transition relation

Transition Systems and Paths

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states
- $\rightarrow \subseteq S \times S$ a transition relation
- $L : S \rightarrow \mathcal{P}(Atoms)$ a labeling function

Transition Systems and Paths

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states
- $\rightarrow \subseteq S \times S$ a transition relation
- $L : S \rightarrow \mathcal{P}(Atoms)$ a labeling function

such that every state has an outward transition, i.e., for all $s_1 \in S$ there exists $s_2 \in S$ with $s_1 \rightarrow s_2$.

Transition Systems and Paths

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states
- $\rightarrow \subseteq S \times S$ a transition relation
- $L : S \rightarrow \mathcal{P}(Atoms)$ a labeling function

such that every state has an outward transition, i.e., for all $s_1 \in S$ there exists $s_2 \in S$ with $s_1 \rightarrow s_2$.

A path π in an LTS $\mathcal{M} = (S, \rightarrow, L)$ is an infinite sequence of states $s_0 s_1 s_2 \dots$ such that for all $i \geq 0$, $s_i \rightarrow s_{i+1}$.

Transition Systems and Paths

A labeled transition system (LTS for short) is a triple $\mathcal{M} = (S, \rightarrow, L)$ consisting of:

- S a finite set of states
- $\rightarrow \subseteq S \times S$ a transition relation
- $L : S \rightarrow \mathcal{P}(Atoms)$ a labeling function

such that every state has an outward transition, i.e., for all $s_1 \in S$ there exists $s_2 \in S$ with $s_1 \rightarrow s_2$.

A path π in an LTS $\mathcal{M} = (S, \rightarrow, L)$ is an infinite sequence of states $s_0 s_1 s_2 \ldots$ such that for all $i \geq 0$, $s_i \rightarrow s_{i+1}$.

Paths are written as $\pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$

Transition Systems and Paths – Example

Recall the example with two parallel processes, where, for $i \in \{1, 2\}$:

- n_i denotes "process i **n**ot in critical section"
- r_i denotes "process i **r**equesting to enter critical section"
- c_i denotes "process i in **c**ritical section"

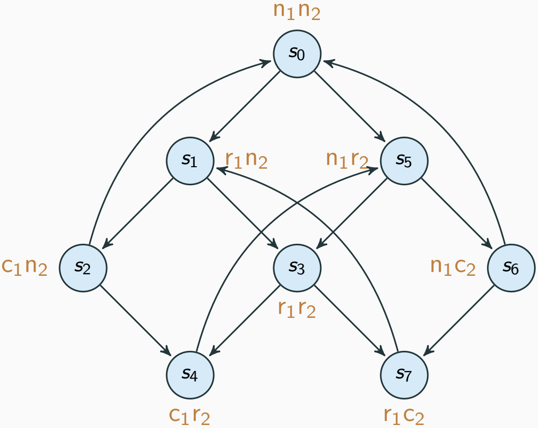
$Atoms = \{n_1, n_2, r_1, r_2, c_1, c_2\}$

Transition Systems and Paths – Example

Recall the example with two parallel processes, where, for $i \in \{1, 2\}$:

- n_i denotes "process i **n**ot in critical section"
- r_i denotes "process i **r**equesting to enter critical section"
- c_i denotes "process i in **c**ritical section"

$Atoms = \{n_1, n_2, r_1, r_2, c_1, c_2\}$

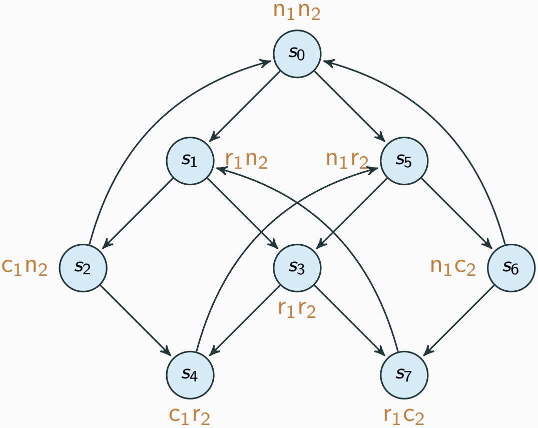


Transition Systems and Paths – Example

Recall the example with two parallel processes, where, for $i \in \{1, 2\}$:

- n_i denotes "process i **n**ot in critical section"
- r_i denotes "process i **r**equesting to enter critical section"
- c_i denotes "process i in **c**ritical section"

$Atoms = \{n_1, n_2, r_1, r_2, c_1, c_2\}$



$\mathcal{M} = (S, \rightarrow, L)$ where

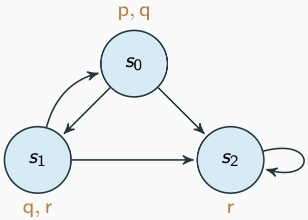
- $S = \{s_0, s_1, \ldots, s_7\}$
- $\rightarrow = \{(s_0, s_1), (s_0, s_5), \ldots\}$
- $L(s_0) = \{n_1, n_2\}$
- $L(s_1) = \{r_1, n_2\}$
- ...

Unwinding a Transition System

Visualise all paths from a given state s_0 by unwinding the LTS to obtain an infinite tree.

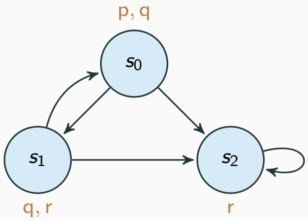
Unwinding a Transition System

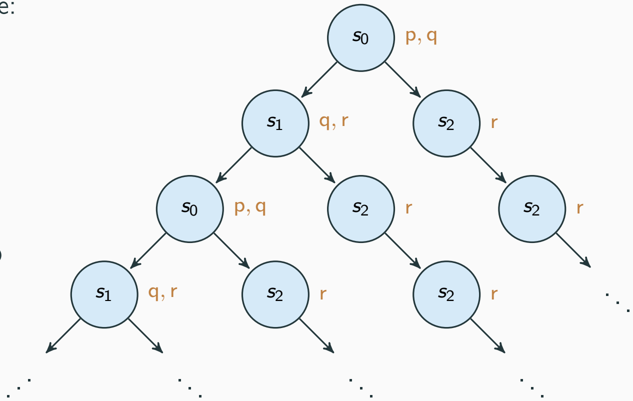
Visualise all paths from a given state s_0 by unwinding the LTS to obtain an infinite tree. For example:



Unwinding a Transition System

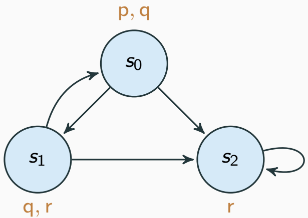
Visualise all paths from a given state s_0 by unwinding the LTS to obtain an infinite tree. For example:

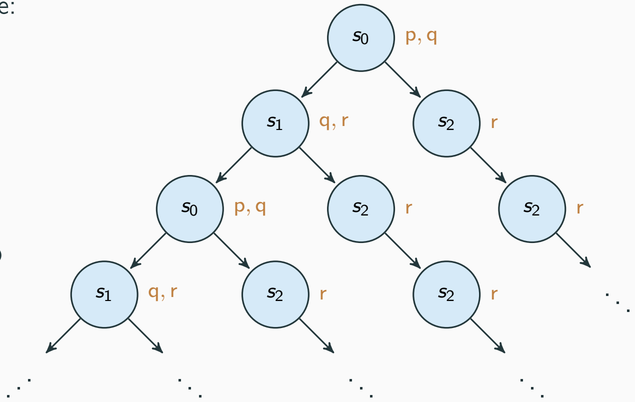




Unwinding a Transition System

Visualise all paths from a given state s_0 by unwinding the LTS to obtain an infinite tree. For example:

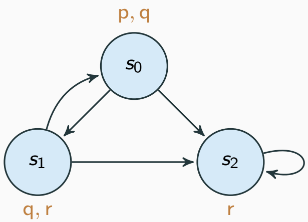


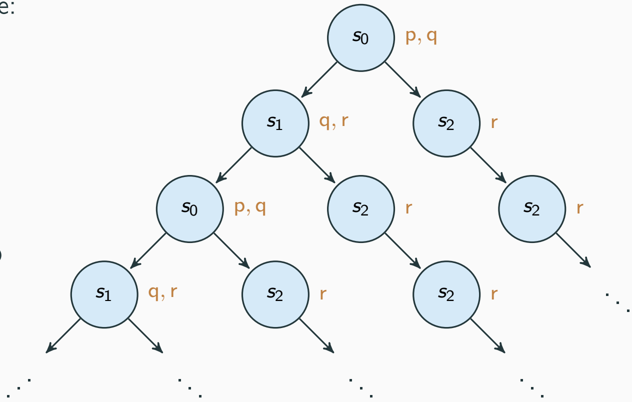


All possible paths starting in s_0:

Unwinding a Transition System

Visualise all paths from a given state s_0 by unwinding the LTS to obtain an infinite tree. For example:



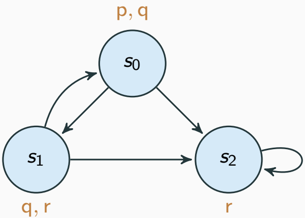


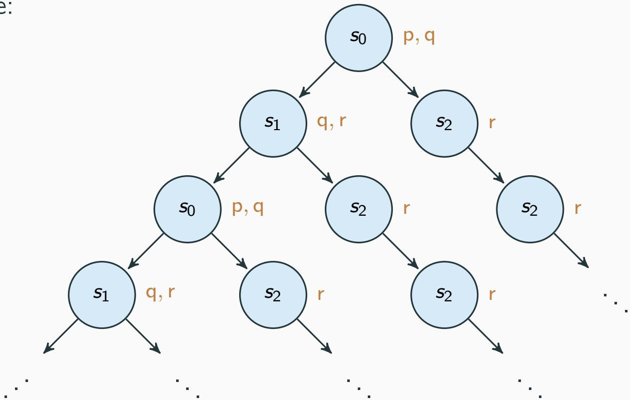
All possible paths starting in s_0:

$$(s_0 \to s_1 \to)^\infty$$

Unwinding a Transition System

Visualise all paths from a given state s_0 by unwinding the LTS to obtain an infinite tree. For example:





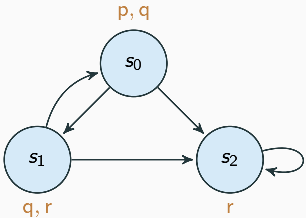
All possible paths starting in s_0:

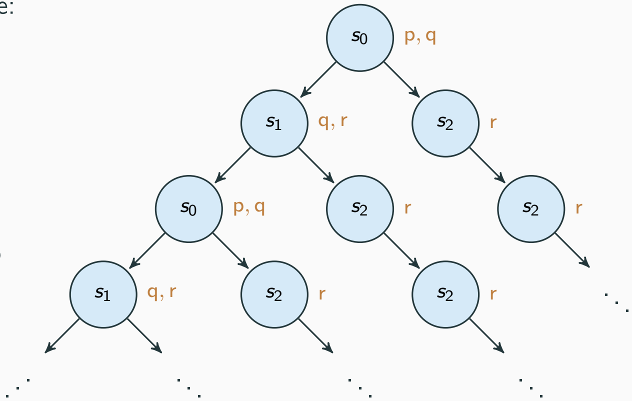
$(s_0 \to s_1 \to)^\infty$

$(s_0 \to s_1 \to)^n \, (s_2 \to)^\infty$ for $n \geq 1$

Unwinding a Transition System

Visualise all paths from a given state s_0 by unwinding the LTS to obtain an infinite tree. For example:





All possible paths starting in s_0:

$$(s_0 \to s_1 \to)^\infty$$

$$(s_0 \to s_1 \to)^n \, (s_2 \to)^\infty \quad \text{for } n \geq 1$$

$$(s_0 \to s_1 \to)^n \, s_0 \to (s_2 \to)^\infty \quad \text{for } n \geq 0$$

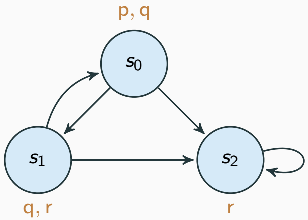
Formal Semantics Continued: Satisfaction Relation for LTSs

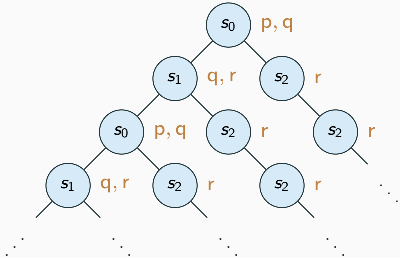
Let $\mathcal{M} = (S, \to, L)$ be an LTS and φ be an LTL formula.

We extend the satisfaction relation from infinite sequences to LTSs as follows:

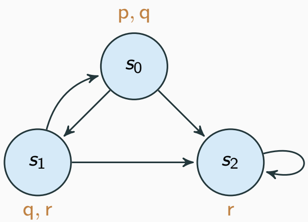
For a state $s \in S$, we define $\mathcal{M}, s \models \varphi$, read $\mathcal{M}$ satisfies φ in state s or φ holds for $\mathcal{M}$ in state s, to mean that $\pi \models_L \varphi$ for every path π of $\mathcal{M}$ starting at state s.

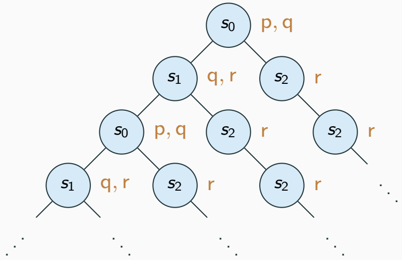
Satisfaction Relation for LTSs – Example





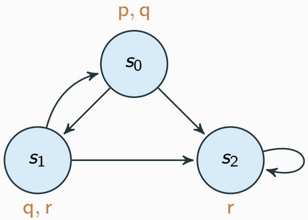
Satisfaction Relation for LTSs – Example

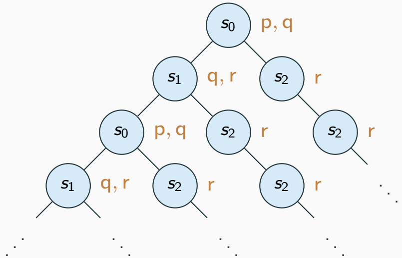




1. $\mathcal{M}, s_0 \models p \wedge q$

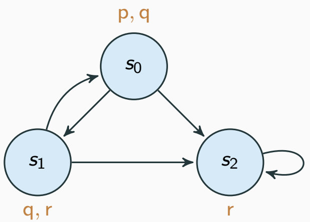
Satisfaction Relation for LTSs – Example

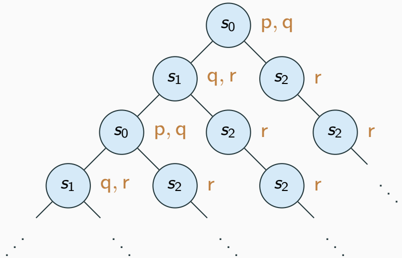




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$

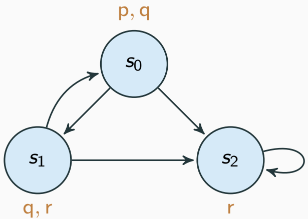
Satisfaction Relation for LTSs – Example

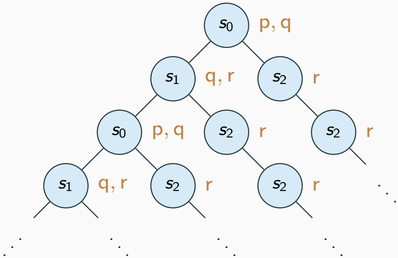




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$

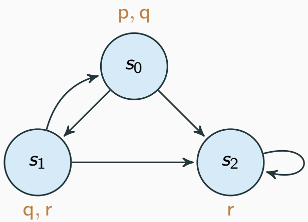
Satisfaction Relation for LTSs – Example

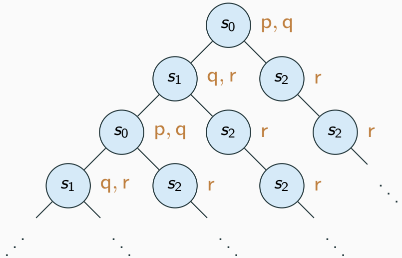




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$
4. $\mathcal{M}, s_0 \not\models \bigcirc (q \wedge r)$

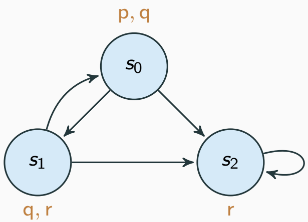
Satisfaction Relation for LTSs – Example

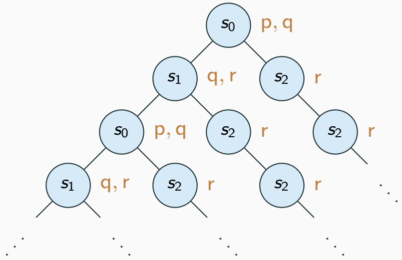




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$
4. $\mathcal{M}, s_0 \not\models \bigcirc (q \wedge r)$
5. $\mathcal{M}, s_0 \models \Box \neg (p \wedge r)$

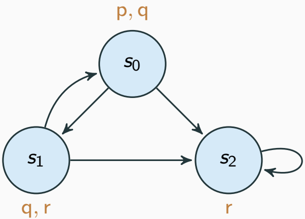
Satisfaction Relation for LTSs – Example

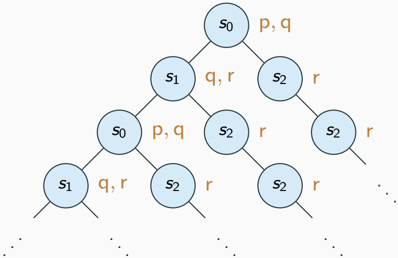




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$
4. $\mathcal{M}, s_0 \not\models \bigcirc(q \wedge r)$
5. $\mathcal{M}, s_0 \models \Box\neg(p \wedge r)$
6. $\mathcal{M}, s_2 \models \Box r$

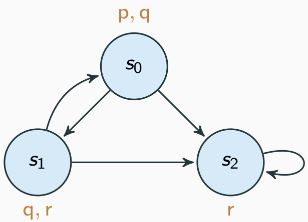
Satisfaction Relation for LTSs – Example

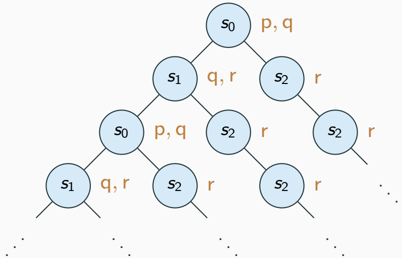




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$
4. $\mathcal{M}, s_0 \not\models \bigcirc(q \wedge r)$
5. $\mathcal{M}, s_0 \models \Box\neg(p \wedge r)$
6. $\mathcal{M}, s_2 \models \Box r$
7. $\mathcal{M}, s_0 \models$ $\Diamond(\neg q \wedge r) \rightarrow \Diamond\Box r$

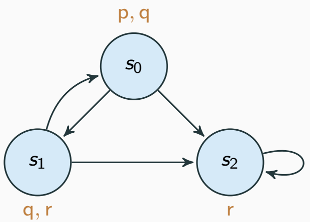
Satisfaction Relation for LTSs – Example

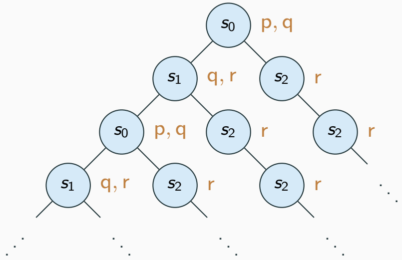




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$
4. $\mathcal{M}, s_0 \not\models \bigcirc(q \wedge r)$
5. $\mathcal{M}, s_0 \models \Box\neg(p \wedge r)$
6. $\mathcal{M}, s_2 \models \Box r$
7. $\mathcal{M}, s_0 \models$ $\Diamond(\neg q \wedge r) \rightarrow \Diamond\Box r$
8. $\mathcal{M}, s_0 \not\models \Box\Diamond p$

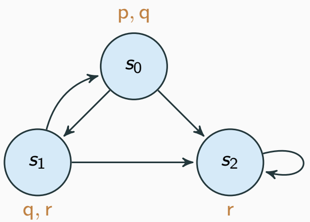
Satisfaction Relation for LTSs – Example

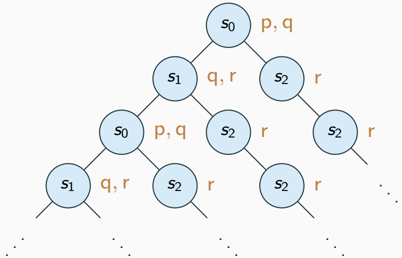




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$
4. $\mathcal{M}, s_0 \not\models \bigcirc(q \wedge r)$
5. $\mathcal{M}, s_0 \models \Box\neg(p \wedge r)$
6. $\mathcal{M}, s_2 \models \Box r$
7. $\mathcal{M}, s_0 \models \Diamond(\neg q \wedge r) \rightarrow \Diamond\Box r$
8. $\mathcal{M}, s_0 \not\models \Box\Diamond p$
9. $\mathcal{M}, s_0 \models \Box\Diamond p \rightarrow \Box\Diamond r$

Satisfaction Relation for LTSs – Example

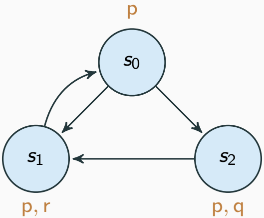




1. $\mathcal{M}, s_0 \models p \wedge q$
2. $\mathcal{M}, s_0 \models \neg r$
3. $\mathcal{M}, s_0 \models \bigcirc r$
4. $\mathcal{M}, s_0 \not\models \bigcirc (q \wedge r)$
5. $\mathcal{M}, s_0 \models \Box \neg (p \wedge r)$
6. $\mathcal{M}, s_2 \models \Box r$
7. $\mathcal{M}, s_0 \models \Diamond(\neg q \wedge r) \rightarrow \Diamond \Box r$
8. $\mathcal{M}, s_0 \not\models \Box \Diamond p$
9. $\mathcal{M}, s_0 \models \Box \Diamond p \rightarrow \Box \Diamond r$
10. $\mathcal{M}, s_0 \not\models \Box \Diamond r \rightarrow \Box \Diamond p$

Homework Exercise 1

Consider the LTS shown in the picture:



1. Write down the mathematical definitions of its components S, $\to$ and L.
2. Draw its unwinding tree.
3. Describe all its possible paths that start at state s_0.
4. Determine which of the following are true, and explain why or why not:

$s_1 \models p \wedge r$ $\qquad$ $s_0 \models \bigcirc r$

$s_0 \models \bigcirc(p \vee r)$ $\qquad$ $s_2 \models \Box p$

$s_0 \models (p \vee q) \,\mathrm{U}\, r$ $\qquad$ $s_1 \models (p \wedge \neg r) \,\mathrm{U}\, q$

5. Give your own examples of LTL formulas and states such that the formula holds or does not hold in the given state, and in each case explain why.

Homework Exercise 2

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold.

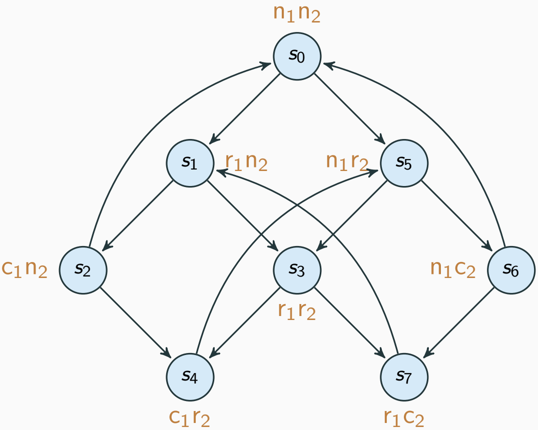
- The safety property: Only one process may execute critical section code at any point
- The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.
- The non-blocking property: A process can always request to enter its critical section.

Transition Systems and Paths – Example

Recall the example with two parallel processes, where, for $i \in \{1, 2\}$:

- n_i denotes "process i **n**ot in critical section"
- r_i denotes "process i **r**equesting to enter critical section"
- c_i denotes "process i in **c**ritical section"

$Atoms = \{n_1, n_2, r_1, r_2, c_1, c_2\}$



$\mathcal{M} = (S, \rightarrow, L)$ where

- $S = \{s_0, s_1, \ldots, s_7\}$
- $\rightarrow = \{(s_0, s_1), (s_0, s_5), \ldots\}$
- $L(s_0) = \{n_1, n_2\}$
- $L(s_1) = \{r_1, n_2\}$
- ...

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box\,(\neg(c_1 \wedge c_2))$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \to, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \square\,(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$
means (by the semantics in an LTS)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \varphi$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$
means (by the semantics in an LTS)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \varphi$
which means (by the semantics of $\Box$)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, $\pi^i \models_L \neg(c_1 \wedge c_2)$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \land c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$
means (by the semantics in an LTS)
for all $\pi \in Paths_{s_0}(\mathcal{M}), \ \pi \models_L \varphi$
which means (by the semantics of $\Box$)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, $\pi^i \models_L \neg(c_1 \land c_2)$
which means (by the semantics of the propositional connectives and atoms)
for all $\pi = t_0 t_1 t_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, not $(c_1 \in L(t_i)$ and $c_2 \in L(t_i))$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0). Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$
means (by the semantics in an LTS)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \varphi$
which means (by the semantics of $\Box$)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, $\pi^i \models_L \neg(c_1 \wedge c_2)$
which means (by the semantics of the propositional connectives and atoms)
for all $\pi = t_0 t_1 t_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, not ($c_1 \in L(t_i)$ and $c_2 \in L(t_i)$)
which is implied by
for all $s \in S$, not ($c_1 \in L(s)$ and $c_2 \in L(s)$)

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$
means (by the semantics in an LTS)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \varphi$
which means (by the semantics of $\Box$)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, $\pi^i \models_L \neg(c_1 \wedge c_2)$
which means (by the semantics of the propositional connectives and atoms)
for all $\pi = t_0 t_1 t_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, not $(c_1 \in L(t_i)$ and $c_2 \in L(t_i))$
which is implied by
for all $s \in S$, not $(c_1 \in L(s)$ and $c_2 \in L(s))$
which is true – can be checked by inspecting the system.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \to, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$
means (by the semantics in an LTS)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \varphi$
which means (by the semantics of $\Box$)
for all $\pi \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, $\pi^i \models_L \neg(c_1 \wedge c_2)$
which means (by the semantics of the propositional connectives and atoms)
for all $\pi = t_0 t_1 t_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, not ($c_1 \in L(t_i)$ and $c_2 \in L(t_i)$)
which is implied by
for all $s \in S$, not ($c_1 \in L(s)$ and $c_2 \in L(s)$)
which is true – can be checked by inspecting the system.

We conclude that $\mathcal{M}, s_0 \models \varphi$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

Safety property: Only one process may execute critical section code at any point.

An LTL formula expressing this is $\varphi = \Box(\neg(c_1 \wedge c_2))$. Let's prove that $\mathcal{M}, s_0 \models \varphi$.

$\mathcal{M}, s_0 \models \varphi$

means (by the semantics in an LTS)

for all $\pi \in Paths_{s_0}(\mathcal{M})$, $\pi \models_L \varphi$

which means (by the semantics of $\Box$)

for all $\pi \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, $\pi^i \models_L \neg(c_1 \wedge c_2)$

which means (by the semantics of the propositional connectives and atoms)

for all $\pi = t_0 t_1 t_2 \ldots \in Paths_{s_0}(\mathcal{M})$, for all $i \geq 0$, not $(c_1 \in L(t_i)$ and $c_2 \in L(t_i))$

which is implied by

for all $s \in S$, not $(c_1 \in L(s)$ and $c_2 \in L(s))$

which is true – can be checked by inspecting the system.

We conclude that $\mathcal{M}, s_0 \models \varphi$.

This was backwards reasoning, reducing the goal to something true.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

An LTL formula expressing this is $\varphi = \Box\,((r_1 \rightarrow \Diamond c_1) \land (r_2 \rightarrow \Diamond c_2))$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

An LTL formula expressing this is $\varphi = \Box\,((r_1 \rightarrow \Diamond c_1) \wedge (r_2 \rightarrow \Diamond c_2))$.

Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

An LTL formula expressing this is $\varphi = \Box\,((r_1 \rightarrow \Diamond c_1) \wedge (r_2 \rightarrow \Diamond c_2))$.

Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.

By the semantics in an LTS, it suffices to find one $\pi \in Path_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

An LTL formula expressing this is $\varphi = \Box((r_1 \rightarrow \Diamond c_1) \wedge (r_2 \rightarrow \Diamond c_2))$.

Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.

By the semantics in an LTS, it suffices to find one $\pi \in Path_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$.

We take $\pi = s_0(s_1 s_3 s_7)^\infty$.

$\pi \models_L \varphi$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \to, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.
An LTL formula expressing this is $\varphi = \Box((r_1 \to \Diamond c_1) \land (r_2 \to \Diamond c_2))$.
Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.
By the semantics in an LTS, it suffices to find one $\pi \in Path_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$.
We take $\pi = s_0(s_1 s_3 s_7)^\infty$.

$\pi \models_L \varphi$
implies, by the semantics of $\Box$ and $\land$
$(s_1 s_3 s_7)^\infty \models_L r_1 \to \Diamond c_1$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

An LTL formula expressing this is $\varphi = \Box((r_1 \rightarrow \Diamond c_1) \wedge (r_2 \rightarrow \Diamond c_2))$.

Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.

By the semantics in an LTS, it suffices to find one $\pi \in Path_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$.

We take $\pi = s_0(s_1 s_3 s_7)^\infty$.

$\pi \models_L \varphi$

implies, by the semantics of $\Box$ and $\wedge$

$(s_1 s_3 s_7)^\infty \models_L r_1 \rightarrow \Diamond c_1$

which implies, by the semantics of $\rightarrow$ and atoms (since $r_1 \in L(s_1)$)

$(s_1 s_3 s_7)^\infty \models_L \Diamond c_1$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

An LTL formula expressing this is $\varphi = \Box\,((r_1 \rightarrow \Diamond c_1) \wedge (r_2 \rightarrow \Diamond c_2))$.

Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.

By the semantics in an LTS, it suffices to find one $\pi \in Path_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$.

We take $\pi = s_0(s_1 s_3 s_7)^\infty$.

$\pi \models_L \varphi$

implies, by the semantics of $\Box$ and $\wedge$

$(s_1 s_3 s_7)^\infty \models_L r_1 \rightarrow \Diamond c_1$

which implies, by the semantics of $\rightarrow$ and atoms (since $r_1 \in L(s_1)$)

$(s_1 s_3 s_7)^\infty \models_L \Diamond c_1$

which implies, by the semantics of $\Diamond$ and atoms

$c_1 \in L(s_1)$ or $c_1 \in L(s_3)$ or $c_1 \in L(s_7)$

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.

An LTL formula expressing this is $\varphi = \Box((r_1 \rightarrow \Diamond c_1) \wedge (r_2 \rightarrow \Diamond c_2))$.

Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.

By the semantics in an LTS, it suffices to find one $\pi \in Path_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$.

We take $\pi = s_0(s_1 s_3 s_7)^\infty$.

$\pi \models_L \varphi$

implies, by the semantics of $\Box$ and $\wedge$

$(s_1 s_3 s_7)^\infty \models_L r_1 \rightarrow \Diamond c_1$

which implies, by the semantics of $\rightarrow$ and atoms (since $r_1 \in L(s_1)$)

$(s_1 s_3 s_7)^\infty \models_L \Diamond c_1$

which implies, by the semantics of $\Diamond$ and atoms

$c_1 \in L(s_1)$ or $c_1 \in L(s_3)$ or $c_1 \in L(s_7)$

which is false – as can be seen by inspecting the system.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The liveness property: Whenever a process requests to enter its critical section, it will eventually be allowed to do so.
An LTL formula expressing this is $\varphi = \Box\,((r_1 \rightarrow \Diamond c_1) \wedge (r_2 \rightarrow \Diamond c_2))$.
Let's prove that $\mathcal{M}, s_0 \not\models \varphi$.
By the semantics in an LTS, it suffices to find one $\pi \in Path_{s_0}(\mathcal{M})$ such that $\pi \not\models_L \varphi$.
We take $\pi = s_0(s_1 s_3 s_7)^\infty$.

$\pi \models_L \varphi$
implies, by the semantics of $\Box$ and $\wedge$
$(s_1 s_3 s_7)^\infty \models_L r_1 \rightarrow \Diamond c_1$
which implies, by the semantics of $\rightarrow$ and atoms (since $r_1 \in L(s_1)$)
$(s_1 s_3 s_7)^\infty \models_L \Diamond c_1$
which implies, by the semantics of $\Diamond$ and atoms
$c_1 \in L(s_1)$ or $c_1 \in L(s_3)$ or $c_1 \in L(s_7)$
which is false – as can be seen by inspecting the system.
Since the assumption $\pi \models_L \varphi$ leads to a contradiction, we conclude $\pi \not\models_L \varphi$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The non-blocking property: A process can always request to enter its critical section (provided it is not there already).

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The non-blocking property: A process can always request to enter its critical section (provided it is not there already).

Let's call a state t reachable from a state s if there is a finite path in $\mathcal{M}$ from s to t.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The non-blocking property: A process can always request to enter its critical section (provided it is not there already).

Let's call a state t reachable from a state s if there is a finite path in $\mathcal{M}$ from s to t.

We can express the non-blocking property for process 1 as follows:

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).

Let $\mathcal{M} = (S, \to, L)$ be that transition system.

The non-blocking property: A process can always request to enter its critical section (provided it is not there already).

Let's call a state t reachable from a state s if there is a finite path in $\mathcal{M}$ from s to t.

We can express the non-blocking property for process 1 as follows:

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

And similarly NB2 for process 2. Our property is therefore NB = "NB1 and NB2".

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The non-blocking property: A process can always request to enter its critical section (provided it is not there already).

Let's call a state t reachable from a state s if there is a finite path in $\mathcal{M}$ from s to t.

We can express the non-blocking property for process 1 as follows:
NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.
And similarly NB2 for process 2. Our property is therefore NB = "NB1 and NB2".

The properties NB1 and NB2 (hence NB as well) are true about the system $\mathcal{M}$.
This can be routinely checked by:
- looking at all the states s reachable from s_0 such that $c_1 \notin L(s)$
- and, for of them, finding a state t reachable from s such that $r_1 \in L(t)$.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The non-blocking property: A process can always request to enter its critical section (provided it is not there already).

Let's call a state t reachable from a state s if there is a finite path in $\mathcal{M}$ from s to t.

We can express the non-blocking property for process 1 as follows:
NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.
And similarly NB2 for process 2. Our property is therefore NB = "NB1 and NB2".

The properties NB1 and NB2 (hence NB as well) are true about the system $\mathcal{M}$.
This can be routinely checked by:
- looking at all the states s reachable from s_0 such that $c_1 \notin L(s)$
- and, for of them, finding a state t reachable from s such that $r_1 \in L(t)$.

But NB1, NB2 and NB are not expressible as LTL formulas.

Homework Exercise 2 – Solution

In the example with the two processes executed in parallel, determine whether the following properties are expressible in LTL; and if yes, whether they hold (for s_0).
Let $\mathcal{M} = (S, \rightarrow, L)$ be that transition system.

The non-blocking property: A process can always request to enter its critical section (provided it is not there already).

Let's call a state t reachable from a state s if there is a finite path in $\mathcal{M}$ from s to t.

We can express the non-blocking property for process 1 as follows:
NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.
And similarly NB2 for process 2. Our property is therefore NB = "NB1 and NB2".

The properties NB1 and NB2 (hence NB as well) are true about the system $\mathcal{M}$.
This can be routinely checked by:
- looking at all the states s reachable from s_0 such that $c_1 \notin L(s)$
- and, for of them, finding a state t reachable from s such that $r_1 \in L(t)$.

But NB1, NB2 and NB are not expressible as LTL formulas.
Can we prove this? Hmm... what does it even mean?

Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \rightarrow, L)$ and states $s_0 \in S$, $\underline{\text{NB1 is true for } \mathcal{M} \text{ and } s_0}$ iff $\underline{\mathcal{M}, s_0 \models \varphi}$.

Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \rightarrow, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

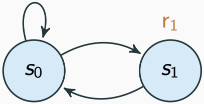
Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above.

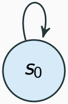
Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \rightarrow, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above. Let $\mathcal{M} = (S, \rightarrow, L)$ and $\mathcal{M}' = (S', \rightarrow', L')$ be the LTSs shown on the left and on the right, respectively.



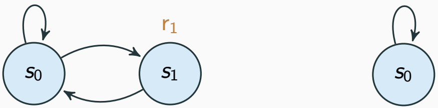


Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \rightarrow, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above. Let $\mathcal{M} = (S, \rightarrow, L)$ and $\mathcal{M}' = (S', \rightarrow', L')$ be the LTSs shown on the left and on the right, respectively.

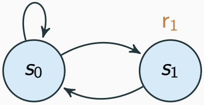


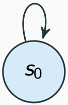
Clearly, NB1 is true for $\mathcal{M}$ and s_0, but NB1 is not true for $\mathcal{M}'$ and s_0.

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \to, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above. Let $\mathcal{M} = (S, \to, L)$ and $\mathcal{M}' = (S', \to', L')$ be the LTSs shown on the left and on the right, respectively.





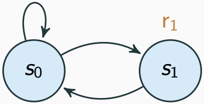
Clearly, NB1 is true for $\mathcal{M}$ and s_0, but NB1 is not true for $\mathcal{M}'$ and s_0. Then, by the choice of φ, we have $\mathcal{M}, s_0 \models \varphi$.

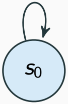
Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \to, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above.
Let $\mathcal{M} = (S, \to, L)$ and $\mathcal{M}' = (S', \to', L')$ be the LTSs shown on the left and on the right, respectively.





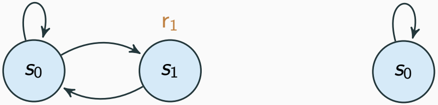
Clearly, NB1 is true for $\mathcal{M}$ and s_0, but NB1 is not true for $\mathcal{M}'$ and s_0.
Then, by the choice of φ, we have $\mathcal{M}, s_0 \models \varphi$.
And since $Path_{s_0}(\mathcal{M}') \subseteq Path_{s_0}(\mathcal{M})$ and $L(s_0) = L'(s_0)$ ($\mathcal{M}'$ is a subsystem of $\mathcal{M}$), from the above we have $\mathcal{M}', s_0 \models \varphi$.

Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \rightarrow, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above.
Let $\mathcal{M} = (S, \rightarrow, L)$ and $\mathcal{M}' = (S', \rightarrow', L')$ be the LTSs shown on the left and on the right, respectively.



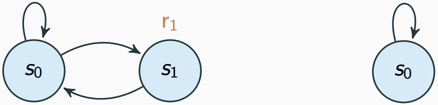
Clearly, NB1 is true for $\mathcal{M}$ and s_0, but NB1 is not true for $\mathcal{M}'$ and s_0.
Then, by the choice of φ, we have $\mathcal{M}, s_0 \models \varphi$.
And since $Path_{s_0}(\mathcal{M}') \subseteq Path_{s_0}(\mathcal{M})$ and $L(s_0) = L'(s_0)$ ($\mathcal{M}'$ is a subsystem of $\mathcal{M}$), from the above we have $\mathcal{M}', s_0 \models \varphi$.
Hence, by the choice of φ, NB1 is true for $\mathcal{M}'$ and s_0, which yields a contradiction.

Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \rightarrow, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above.
Let $\mathcal{M} = (S, \rightarrow, L)$ and $\mathcal{M}' = (S', \rightarrow', L')$ be the LTSs shown on the left and on the right, respectively.



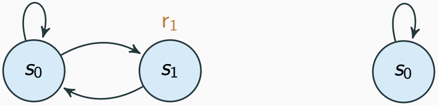
Clearly, NB1 is true for $\mathcal{M}$ and s_0, but NB1 is not true for $\mathcal{M}'$ and s_0.
Then, by the choice of φ, we have $\mathcal{M}, s_0 \models \varphi$.
And since $Path_{s_0}(\mathcal{M}') \subseteq Path_{s_0}(\mathcal{M})$ and $L(s_0) = L'(s_0)$ ($\mathcal{M}'$ is a subsystem of $\mathcal{M}$), from the above we have $\mathcal{M}', s_0 \models \varphi$.
Hence, by the choice of φ, NB1 is true for $\mathcal{M}'$ and s_0, which yields a contradiction.
We've reached a contradiction, meaning our assumption is false. So NB1 is not expressible in LTL.

Expressibility in LTL

NB1: For all states s reachable from s_0 such that $c_1 \notin L(s)$, there exists a state t reachable from s such that $r_1 \in L(t)$.

NB1 expressible in LTL means: There exists an LTL formula φ such that, for all LTSs $\mathcal{M} = (S, \rightarrow, L)$ and states $s_0 \in S$, NB1 is true for $\mathcal{M}$ and s_0 iff $\mathcal{M}, s_0 \models \varphi$.

Let's assume NB1 expressible in LTL, and let φ be an LTL formula as above.
Let $\mathcal{M} = (S, \rightarrow, L)$ and $\mathcal{M}' = (S', \rightarrow', L')$ be the LTSs shown on the left and on the right, respectively.



Clearly, NB1 is true for $\mathcal{M}$ and s_0, but NB1 is not true for $\mathcal{M}'$ and s_0.
Then, by the choice of φ, we have $\mathcal{M}, s_0 \models \varphi$.
And since $Path_{s_0}(\mathcal{M}') \subseteq Path_{s_0}(\mathcal{M})$ and $L(s_0) = L'(s_0)$ ($\mathcal{M}'$ is a subsystem of $\mathcal{M}$), from the above we have $\mathcal{M}', s_0 \models \varphi$.
Hence, by the choice of φ, NB1 is true for $\mathcal{M}'$ and s_0, which yields a contradiction.
We've reached a contradiction, meaning our assumption is false. So NB1 is not expressible in LTL.

Homework: Modify the proof to show that NB is not expressible in LTL.

Formula Equivalence

Two formulas φ and ψ are equivalent, denoted $\varphi \equiv \psi$, if they are satisfied by (i.e., hold for) exactly the same state labelings and infinite sequences of states: Given any labeling $L : S \to \mathcal{P}(Atoms)$ and any infinite sequence of states π, we have that $\pi \models_L \varphi$ iff $\pi \models_L \psi$

Formula Equivalence

Two formulas φ and ψ are equivalent, denoted $\varphi \equiv \psi$, if they are satisfied by (i.e., hold for) exactly the same state labelings and infinite sequences of states: Given any labeling $L : S \to \mathcal{P}(Atoms)$ and any infinite sequence of states π, we have that $\pi \models_L \varphi$ iff $\pi \models_L \psi$; in other words:

(1) $\pi \models_L \varphi$ implies $\pi \models_L \psi$
and
(2) $\pi \models_L \psi$ implies $\pi \models_L \varphi$.

Formula Equivalence

Two formulas φ and ψ are equivalent, denoted $\varphi \equiv \psi$, if they are satisfied by (i.e., hold for) exactly the same state labelings and infinite sequences of states: Given any labeling $L : S \to \mathcal{P}(Atoms)$ and any infinite sequence of states π, we have that $\pi \models_L \varphi$ iff $\pi \models_L \psi$; in other words:

(1) $\pi \models_L \varphi$ implies $\pi \models_L \psi$
and
(2) $\pi \models_L \psi$ implies $\pi \models_L \varphi$.

Note. If $\varphi \equiv \psi$, then φ and ψ will also be satisfied by the same LTSs in the same states: Given any LTS $\mathcal{M} = (S, \to, L)$ and any $s \in S$, we have that $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}, s \models \psi$.

Formula Equivalence

Two formulas φ and ψ are equivalent, denoted $\varphi \equiv \psi$, if they are satisfied by (i.e., hold for) exactly the same state labelings and infinite sequences of states: Given any labeling $L : S \to \mathcal{P}(Atoms)$ and any infinite sequence of states π, we have that $\pi \models_L \varphi$ iff $\pi \models_L \psi$; in other words:

(1) $\pi \models_L \varphi$ implies $\pi \models_L \psi$
and
(2) $\pi \models_L \psi$ implies $\pi \models_L \varphi$.

Note. If $\varphi \equiv \psi$, then φ and ψ will also be satisfied by the same LTSs in the same states: Given any LTS $\mathcal{M} = (S, \to, L)$ and any $s \in S$, we have that $\mathcal{M}, s \models \varphi$ iff $\mathcal{M}, s \models \psi$.

Homework Exercise 3: Explain why this is the case.

Some Formula Equivalences

Propositional tautologies:

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi \qquad \neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$$

Some Formula Equivalences

Propositional tautologies:

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi \qquad \neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$$

Duality laws:

$$\neg \bigcirc \varphi \equiv \bigcirc\neg\varphi \qquad \neg\Box\varphi \equiv \Diamond\neg\varphi \qquad \neg\Diamond\varphi \equiv \Box\neg\varphi$$

Some Formula Equivalences

Propositional tautologies:

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi \qquad \neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$$

Duality laws:

$$\neg \bigcirc \varphi \equiv \bigcirc\neg\varphi \qquad \neg\Box\varphi \equiv \Diamond\neg\varphi \qquad \neg\Diamond\varphi \equiv \Box\neg\varphi$$

Distributive laws:

$$\Box(\varphi \wedge \psi) \equiv \Box\varphi \wedge \Box\psi \qquad \Diamond(\varphi \vee \psi) \equiv \Diamond\varphi \vee \Diamond\psi \qquad \bigcirc(\varphi \,\mathsf{U}\, \psi) \equiv \bigcirc\varphi \,\mathsf{U}\, \bigcirc\psi$$

Some Formula Equivalences

Propositional tautologies:

$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi \qquad \neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi$$

Duality laws:

$$\neg \bigcirc \varphi \equiv \bigcirc\neg\varphi \qquad \neg \Box\varphi \equiv \Diamond\neg\varphi \qquad \neg \Diamond\varphi \equiv \Box\neg\varphi$$

Distributive laws:

$$\Box(\varphi \wedge \psi) \equiv \Box\varphi \wedge \Box\psi \qquad \Diamond(\varphi \vee \psi) \equiv \Diamond\varphi \vee \Diamond\psi \qquad \bigcirc(\varphi \,\mathsf{U}\, \psi) \equiv \bigcirc\varphi \,\mathsf{U}\, \bigcirc\psi$$

Note:

$$\Box(\varphi \vee \psi) \not\equiv \Box\varphi \vee \Box\psi \qquad \Diamond(\varphi \wedge \psi) \not\equiv \Diamond\varphi \wedge \Diamond\psi$$

Some Formula Equivalences

Inter-definability laws:

$$\Diamond\varphi \equiv \neg\Box\neg\varphi \qquad \Box\varphi \equiv \neg\Diamond\neg\varphi \qquad \Diamond\varphi \equiv \top \,\mathsf{U}\, \varphi$$

where $\top$ (read "True") is an abbreviation for $p \rightarrow p$ for some atom p

Some Formula Equivalences

Inter-definability laws:

$$\Diamond\varphi \equiv \neg\Box\neg\varphi \qquad \Box\varphi \equiv \neg\Diamond\neg\varphi \qquad \Diamond\varphi \equiv \top \,\mathsf{U}\, \varphi$$

where $\top$ (read "True") is an abbreviation for $p \to p$ for some atom p

Idempotency laws:

$$\Diamond\Diamond\varphi \equiv \Diamond\varphi \qquad \Box\Box\varphi \equiv \Box\varphi \qquad (\varphi \,\mathsf{U}\, \psi) \,\mathsf{U}\, \psi \equiv \varphi \,\mathsf{U}\, \psi \qquad \varphi \,\mathsf{U}\, (\varphi \,\mathsf{U}\, \psi) \equiv \varphi \,\mathsf{U}\, \psi$$

Some Formula Equivalences

Inter-definability laws:

$$\Diamond\varphi \equiv \neg\Box\neg\varphi \qquad \Box\varphi \equiv \neg\Diamond\neg\varphi \qquad \Diamond\varphi \equiv \top \,\mathsf{U}\, \varphi$$

where $\top$ (read "True") is an abbreviation for $p \to p$ for some atom p

Idempotency laws:

$$\Diamond\Diamond\varphi \equiv \Diamond\varphi \qquad \Box\Box\varphi \equiv \Box\varphi \qquad (\varphi \,\mathsf{U}\, \psi) \,\mathsf{U}\, \psi \equiv \varphi \,\mathsf{U}\, \psi \qquad \varphi \,\mathsf{U}\, (\varphi \,\mathsf{U}\, \psi) \equiv \varphi \,\mathsf{U}\, \psi$$

Absorption laws:

$$\Box\Diamond\Box\varphi \equiv \Diamond\Box\varphi \qquad \Diamond\Box\Diamond\varphi \equiv \Box\Diamond\varphi$$

Some Formula Equivalences

Inter-definability laws:

$$\Diamond\varphi \equiv \neg\Box\neg\varphi \qquad \Box\varphi \equiv \neg\Diamond\neg\varphi \qquad \Diamond\varphi \equiv \top\,\mathsf{U}\,\varphi$$

where $\top$ (read "True") is an abbreviation for $p \rightarrow p$ for some atom p

Idempotency laws:

$$\Diamond\Diamond\varphi \equiv \Diamond\varphi \qquad \Box\Box\varphi \equiv \Box\varphi \qquad (\varphi\,\mathsf{U}\,\psi)\,\mathsf{U}\,\psi \equiv \varphi\,\mathsf{U}\,\psi \qquad \varphi\,\mathsf{U}\,(\varphi\,\mathsf{U}\,\psi) \equiv \varphi\,\mathsf{U}\,\psi$$

Absorption laws:

$$\Box\Diamond\Box\varphi \equiv \Diamond\Box\varphi \qquad \Diamond\Box\Diamond\varphi \equiv \Box\Diamond\varphi$$

Expansion laws:

$$\Diamond\varphi \equiv \varphi \vee \bigcirc\Diamond\varphi \qquad \Box\varphi \equiv \varphi \wedge \bigcirc\Box\varphi \qquad \varphi\,\mathsf{U}\,\psi \equiv \psi \vee (\varphi \wedge \bigcirc(\varphi\,\mathsf{U}\,\psi))$$

Proving Formula Equivalences

Let us prove the following equivalence:

$$\Diamond\varphi \equiv \neg\Box\neg\varphi$$

Proving Formula Equivalences

Let us prove the following equivalence:

$$\Diamond\varphi \equiv \neg\Box\neg\varphi$$

Fix a labeling function $L : S \to \mathcal{P}(Atoms)$ and let π be an infinite sequence $s_0 s_1 s_2 \ldots$.

Proving Formula Equivalences

Let us prove the following equivalence:

$$\Diamond \varphi \equiv \neg \Box \neg \varphi$$

Fix a labeling function $L : S \to \mathcal{P}(Atoms)$ and let π be an infinite sequence $s_0 s_1 s_2 \ldots$. We must prove two things:

(1) $\pi \models \Diamond \varphi$ implies $\pi \models \neg \Box \neg \varphi$.

(2) $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Assume $\pi \models \Diamond\varphi$.

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Assume $\pi \models \Diamond\varphi$.

Hence, by semantics of $\Diamond$, there exists an i such that $\pi^i \models \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Assume $\pi \models \Diamond\varphi$.

Hence, by semantics of $\Diamond$, there exists an i such that $\pi^i \models \varphi$.

Hence, by logic, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Assume $\pi \models \Diamond\varphi$.

Hence, by semantics of $\Diamond$, there exists an i such that $\pi^i \models \varphi$.

Hence, by logic, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \models \neg\varphi$.

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Assume $\pi \models \Diamond\varphi$.

Hence, by semantics of $\Diamond$, there exists an i such that $\pi^i \models \varphi$.

Hence, by logic, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \models \neg\varphi$.

Hence, by semantics of $\Box$, it is not the case that $\pi \models \Box\neg\varphi$.

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Assume $\pi \models \Diamond\varphi$.

Hence, by semantics of $\Diamond$, there exists an i such that $\pi^i \models \varphi$.

Hence, by logic, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \models \neg\varphi$.

Hence, by semantics of $\Box$, it is not the case that $\pi \models \Box\neg\varphi$.

In other words, $\pi \not\models \Box\neg\varphi$.

Proving Formula Equivalences

Proving that $\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$:

Assume $\pi \models \Diamond\varphi$.

Hence, by semantics of $\Diamond$, there exists an i such that $\pi^i \models \varphi$.

Hence, by logic, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \models \neg\varphi$.

Hence, by semantics of $\Box$, it is not the case that $\pi \models \Box\neg\varphi$.

In other words, $\pi \not\models \Box\neg\varphi$.

Hence, by semantics of $\neg$, we have $\pi \models \neg\Box\neg\varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Hence, by semantics of $\neg$, we have $\pi \not\models \Box \neg \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Hence, by semantics of $\neg$, we have $\pi \not\models \Box \neg \varphi$.

In other words, it is not the case that $\pi \models \Box \neg \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Hence, by semantics of $\neg$, we have $\pi \not\models \Box \neg \varphi$.

In other words, it is not the case that $\pi \models \Box \neg \varphi$.

Hence, by semantics of $\Box$, it is not the case that: for all i, $\pi^i \models \neg \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Hence, by semantics of $\neg$, we have $\pi \not\models \Box \neg \varphi$.

In other words, it is not the case that $\pi \models \Box \neg \varphi$.

Hence, by semantics of $\Box$, it is not the case that: for all i, $\pi^i \models \neg \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Hence, by semantics of $\neg$, we have $\pi \not\models \Box \neg \varphi$.

In other words, it is not the case that $\pi \models \Box \neg \varphi$.

Hence, by semantics of $\Box$, it is not the case that: for all i, $\pi^i \models \neg \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Hence, by logic, there exists an i such that $\pi^i \models \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Hence, by semantics of $\neg$, we have $\pi \not\models \Box \neg \varphi$.

In other words, it is not the case that $\pi \models \Box \neg \varphi$.

Hence, by semantics of $\Box$, it is not the case that: for all i, $\pi^i \models \neg \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Hence, by logic, there exists an i such that $\pi^i \models \varphi$.

Hence, by semantics of $\Diamond$, we have $\pi \models \Diamond \varphi$.

Proving Formula Equivalences

Proving that $\pi \models \neg \Box \neg \varphi$ implies $\pi \models \Diamond \varphi$:

Assume $\pi \models \neg \Box \neg \varphi$.

Hence, by semantics of $\neg$, we have $\pi \not\models \Box \neg \varphi$.

In other words, it is not the case that $\pi \models \Box \neg \varphi$.

Hence, by semantics of $\Box$, it is not the case that: for all i, $\pi^i \models \neg \varphi$.

Hence, by semantics of $\neg$, it is not the case that: for all i, $\pi^i \not\models \varphi$.

Hence, by logic, there exists an i such that $\pi^i \models \varphi$.

Hence, by semantics of $\Diamond$, we have $\pi \models \Diamond \varphi$.

Note. The proof of "$\pi \models \neg\Box\neg\varphi$ implies $\pi \models \Diamond\varphi$" is the reverse of the proof of "$\pi \models \Diamond\varphi$ implies $\pi \models \neg\Box\neg\varphi$". So we could have proved directly "$\pi \models \Diamond\varphi$ iff $\pi \models \neg\Box\neg\varphi$" by a chain of equivalent (iff-related) statements.

Homework Exercise 4

Choose from the previous two slides any three laws (except for the propositional tautologies) and prove them.

Hint. Take the approach shown above, using the semantics of formulas and logical reasoning.

Summary of the Discussed Concepts

- LTL = Linear Temporal Logic
- Syntax = formulas built from
  - atoms
  - propositional connectives
  - temporal connectives
- LTL can express some practical specification patterns
- Semantics = the satisfaction relation
  - between infinite sequences and formulas
  - between LTSs and formulas
- Formula equivalence

Further Reading

Sections 5.1.1–5.1.4 of Baier & Katoen's "Principles of Model Checking" (MIT Press 2008)

Section 3.2 of Huth & Ryan's "Logic in Computer Science: Modelling and Reasoning about Systems" (Cambridge University Press 2004)

Note. Uses another (standard) notation for the temporal connectives:

- X instead of $\bigcirc$
- F instead of $\Diamond$ (think "in the Future")
- G instead of $\Box$ (think "Globally")