Specification and Analysis of Contracts
Lectures 3 and 4
Background: Modal Logics

Gerardo Schneider
gerardo@ifi.uio.no
http://folk.uio.no/gerardo/

Department of Informatics,
University of Oslo

SEFM School, Oct. 27 - Nov. 7, 2008
Cape Town, South Africa
Plan of the Course

1. Introduction
2. Components, Services and Contracts
3. Background: Modal Logics 1
4. Background: Modal Logics 2
5. Deontic Logic
6. Challenges in Defining a Good Contract language
7. Specification of 'Deontic' Contracts ($CL$)
8. Verification of 'Deontic' Contracts
9. Conflict Analysis of 'Deontic' Contracts
10. Other Analysis of 'Deontic' Contracts and Summary
Modal Logics

- **Modal logic** is the logic of **possibility** and **necessity**
  - $\square \varphi$: $\varphi$ is necessarily true.
  - $\Diamond \varphi$: $\varphi$ is possibly true.

- Not a single system but many different systems depending on application

- Good to reason about causality and situations with incomplete information

- Different interpretation for the modalities: belief, knowledge, provability, etc.
Modal Logics

- **Modal logic** is the logic of **possibility** and **necessity**
  - $\Box \varphi$: $\varphi$ is necessarily true.
  - $\Diamond \varphi$: $\varphi$ is possibly true.

- Not a single system but many different systems depending on application

- Good to reason about causality and situations with incomplete information

- Different interpretation for the modalities: belief, knowledge, provability, etc.

- Depending on the semantics, we can interpret $\Box \varphi$ differently
  - **temporal**: $\varphi$ will always hold
  - **doxastic**: I believe $\varphi$
  - **epistemic**: I know $\varphi$
  - **deontic**: It ought to be the case that $\varphi$
Modal logic is good to reason in **dynamic** situations
- Truth values may vary over time (classical logic is **static**)

Sentences in classical logic are interpreted over a single structure or **world**

In modal logic, interpretation consists of a collection $K$ of **possible worlds or states**
- If states change, then truth values can also change

**Dynamic interpretation of modal logic**
- Temporal logic
  - Linear time
  - Branching time
- Dynamic logic
Modal logic is good to reason in **dynamic** situations

- Truth values may vary over time (classical logic is **static**)

Sentences in classical logic are interpreted over a single structure or world

In modal logic, interpretation consists of a collection $K$ of possible worlds or states

- If states change, then truth values can also change

Dynamic interpretation of modal logic

- Temporal logic
  - Linear time
  - Branching time
- Dynamic logic
We will see

In the rest of this and next lecture (2 hours):

- Temporal logic
- Propositional modal logic
- Multimodal logic
- Dynamic logic
- $\mu$-calculus
- Real-time logics

In the following lecture (1 hour):

- Deontic logic
Plan

1. Temporal Logic
2. Propositional Modal Logic
3. Multimodal Logic
4. Dynamic Logic
5. Mu-calculus
6. Real-Time Logics
Plan

1. Temporal Logic
2. Propositional Modal Logic
3. Multimodal Logic
4. Dynamic Logic
5. Mu-calculus
6. Real-Time Logics
Temporal Logic

Introduction

- **Temporal logic** is the logic of **time**
- There are different ways of modeling time
  - linear time vs. branching time
  - time instances vs. time intervals
  - discrete time vs. continuous time
  - past and future vs. future only
In **Linear Temporal Logic (LTL)** we can describe such properties as, if $i$ is *now*,

- $p$ holds in $i$ and every following point (the future)
- $p$ holds in $i$ and every preceding point (the past)

We will only be concerned with the future.
In **Linear Temporal Logic (LTL)** we can describe such properties as, if $i$ is *now*,

- $p$ holds in $i$ and every following point (the future)
- $p$ holds in $i$ and every preceding point (the past)

We will only be concerned with the future
In **Linear Temporal Logic (LTL)** we can describe such properties as, if \( i \) is *now*,

- \( p \) holds in \( i \) and every following point (the future)
- \( p \) holds in \( i \) and every preceding point (the past)

We will only be concerned with the future

\[
\cdots \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \cdots \\
\text{i−2} \quad \text{i−1} \quad \text{i} \quad \text{i+1} \quad \text{i+2} \quad \text{i+3} \\
\]

\[
\cdots \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \bullet \quad \cdots \\
\text{i−2} \quad \text{i−1} \quad \text{i} \quad \text{i+1} \quad \text{i+2} \quad \text{i+3} \\
\]

Temporal Logic

Introduction

We extend the first-order language $\mathcal{L}$ to a temporal language $\mathcal{L}_T$ by adding the temporal operators $\Box$, $\Diamond$, $\lozenge$, $U$, $R$ and $W$.

**Interpretation**

- $\Box \varphi$ \hspace{1cm} $\varphi$ will *always* (in every state) hold
- $\Diamond \varphi$ \hspace{1cm} $\varphi$ will *eventually* (in some state) hold
- $\lozenge \varphi$ \hspace{1cm} $\varphi$ will hold at the *next* point in time
- $\varphi U \psi$ \hspace{1cm} $\psi$ will eventually hold, and *until* that point $\varphi$ will hold
- $\varphi R \psi$ \hspace{1cm} $\psi$ holds until (incl.) the point (if any) where $\varphi$ holds (*release*).
- $\varphi W \psi$ \hspace{1cm} $\varphi$ will hold until $\psi$ holds (*weak until* or *waiting for*).
We define **LTL formulae** as follows:

- $\mathcal{L} \subseteq \mathcal{L}_T$: first-order formulae are also LTL formulae
- If $\varphi$ is an LTL formula, so are $\Box \varphi$, $\Diamond \varphi$, $\bigcirc \varphi$ and $\neg \varphi$
- If $\varphi$ and $\psi$ are LTL formulae, so are $\varphi \bigcirc \psi$, $\varphi R \psi$, $\varphi W \psi$, $\varphi \lor \psi$, $\varphi \land \psi$, $\varphi \Rightarrow \psi$ and $\varphi \equiv \psi$
A path is an infinite sequence of states

\[ \sigma = s_0, s_1, s_2, \ldots \]

\( \sigma^k \) denotes the path \( s_k, s_{k+1}, s_{k+2}, \ldots \)

\( \sigma_k \) denotes the state \( s_k \)

All computations are paths, but not vice versa
We define the notion that an LTL formula $\varphi$ is true (false) relative to a path $\sigma$, written $\sigma \models \varphi$ ($\sigma \not\models \varphi$) as follows.

- $\sigma \models \varphi$ iff $\sigma_0 \models \varphi$ when $\varphi \in \mathcal{L}$
- $\sigma \models \neg \varphi$ iff $\sigma \not\models \varphi$
- $\sigma \models \varphi \lor \psi$ iff $\sigma \models \varphi$ or $\sigma \models \psi$
- $\sigma \models \Box \varphi$ iff $\sigma^k \models \varphi$ for all $k \geq 0$
- $\sigma \models \Diamond \varphi$ iff $\sigma^k \models \varphi$ for some $k \geq 0$
- $\sigma \models \bigcirc \varphi$ iff $\sigma^1 \models \varphi$

(cont.)
Linear Temporal Logic
Semantics

Definition

(Cont.)

\[ \sigma \models \varphi U \psi \quad \text{iff} \quad \sigma^k \models \psi \text{ for some } k \geq 0, \text{ and} \]
\[ \sigma^i \models \varphi \text{ for every } i \text{ such that } 0 \leq i < k \]

\[ \sigma \models \varphi R \psi \quad \text{iff} \quad \text{for every } j \geq 0, \]
\[ \text{if for every } i < j \sigma^i \not\models \varphi \text{ then } \sigma^j \models \psi \]

\[ \sigma \models \varphi W \psi \quad \text{iff} \quad \sigma \models \varphi U \psi \text{ or } \sigma \models \Box \varphi \]
Temporal Logic
Semantics

**Definition**

- If $\sigma \models \varphi$ for all paths $\sigma$, we say that $\varphi$ is *(temporally) valid* and write
  
  \[ \models \varphi \]
  
  *(Validity)*

- If $\models \varphi \equiv \psi$ (ie. $\sigma \models \varphi$ iff $\sigma \models \psi$, for all $\sigma$), we say that $\varphi$ and $\psi$ are *equivalent* and write
  
  \[ \varphi \sim \psi \]
  
  *(Equivalence)*
Temporal Logic

Semantics

Definition

- If $\sigma \models \varphi$ for all paths $\sigma$, we say that $\varphi$ is \textit{(temporally) valid} and write

  $\models \varphi$ (Validity)

- If $\models \varphi \equiv \psi$ (ie. $\sigma \models \varphi$ iff $\sigma \models \psi$, for all $\sigma$), we say that $\varphi$ and $\psi$ are \textit{equivalent} and write

  $\varphi \sim \psi$ (Equivalence)
\( \sigma \models \Box p \)

\[
\begin{array}{cccccc}
p & p & p & p & p & p \\
\bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\
0 & 1 & 2 & 3 & 4 & \\
\end{array}
\]
Temporal Logic

Semantics

\[ \sigma \models \Box p \]

\[
\begin{array}{cccccc}
p & p & p & p & p & p \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\
0 & 1 & 2 & 3 & 4 & \\
\end{array}
\]

\[ \sigma \models \Diamond p \]

\[
\begin{array}{cccccc}
p \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\
0 & 1 & 2 & 3 & 4 & \\
\end{array}
\]
Temporal Logic
Semantics

$\sigma \models \square p$

$\sigma \models \Diamond p$

$\sigma \models \Diamond p$

Gerardo Schneider (UiO)
Temporal Logic

Semantics

$\sigma \models p U q$ – The sequence of $p$ is finite

$$
\begin{array}{cccc}
  p & p & p & q \\
  \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\
  0 & 1 & 2 & 3 & 4
\end{array}
$$
Temporal Logic
Semantics

\( \sigma \models pU q \) – The sequence of \( p \) is finite

\[
\begin{array}{cccc}
p & p & p & q \\
\bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\
0 & 1 & 2 & 3 & 4
\end{array}
\]

\( \sigma \models pR q \) – The sequence of \( q \) may be infinite

\[
\begin{array}{cccc}
q & q & q & q, p \\
\bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\
0 & 1 & 2 & 3 & 4
\end{array}
\]
Temporal Logic

Semantics

\[ \sigma \models p \mathcal{U} q \quad \text{– The sequence of } p \text{ is finite} \]

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & \\
p & p & p & q & & \\
\bullet & \bullet & \bullet & \bullet & \bullet & \ldots
\end{array}
\]

\[ \sigma \models p \mathcal{R} q \quad \text{– The sequence of } q \text{ may be infinite} \]

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & \\
q & q & q & q & p & \\
\bullet & \bullet & \bullet & \bullet & \bullet & \ldots
\end{array}
\]

\[ \sigma \models p \mathcal{W} q \quad \text{– The sequence of } p \text{ may be infinite } (p \mathcal{W} q \equiv (p \mathcal{U} q) \lor \Box p) \]

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & \\
p & p & p & q & & \\
\bullet & \bullet & \bullet & \bullet & \bullet & \ldots
\end{array}
\]
Example (Response)

\( \Box (\varphi \Rightarrow \Diamond \psi) \)
Example (Response)

\[ \Box (\varphi \Rightarrow \Diamond \psi) \]

Every \( \varphi \)-position coincides with or is followed by a \( \psi \)-position

<table>
<thead>
<tr>
<th></th>
<th>( \varphi )</th>
<th>( \psi )</th>
<th>( \varphi, \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>1</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>2</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>3</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>4</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>5</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>6</td>
<td>•</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Example (Response)

$\Box(\varphi \Rightarrow \Diamond \psi)$

Every $\varphi$-position coincides with or is followed by a $\psi$-position

\[
\begin{array}{cccccc}
\varphi & \psi & \varphi, \psi \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & \ldots \\
\end{array}
\]

This formula will also hold in every path where $\varphi$ never holds

\[
\begin{array}{cccccc}
\neg \varphi & \neg \varphi & \neg \varphi & \neg \varphi & \neg \varphi \\
0 & 1 & 2 & 3 & 4 & \ldots \\
\end{array}
\]
It can be difficult to correctly formalize informally stated requirements in temporal logic.

**Example**

How does one formalize the informal requirement “ϕ implies ψ”?

- ϕ ⇒ ψ?
- □(ϕ ⇒ ψ)?
- ϕ ⇒ ◻ψ?
- □(ϕ ⇒ ◻ψ)?
It can be difficult to correctly formalize informally stated requirements in temporal logic

**Example**

How does one formalize the informal requirement “$\phi$ implies $\psi$”?  
- $\phi \Rightarrow \psi$?
- $\Box (\phi \Rightarrow \psi)$?
- $\phi \Rightarrow \Diamond \psi$?
- $\Box (\phi \Rightarrow \Diamond \psi)$?
Temporal Logic
Formalization

It can be difficult to correctly formalize informally stated requirements in temporal logic

Example

How does one formalize the informal requirement “$\varphi$ implies $\psi$”?  

- $\varphi \Rightarrow \psi$? $\varphi \Rightarrow \psi$ holds in the initial state
- $\Box(\varphi \Rightarrow \psi)$?
- $\varphi \Rightarrow \Diamond \psi$?
- $\Box(\varphi \Rightarrow \Diamond \psi)$?
Temporal Logic
Formalization

It can be difficult to correctly formalize informally stated requirements in temporal logic

Example

How does one formalize the informal requirement “\( \varphi \) implies \( \psi \)”?

- \( \varphi \Rightarrow \psi \)?

- \( \Box(\varphi \Rightarrow \psi) \)?
Temporal Logic

Formalization

It can be difficult to correctly formalize informally stated requirements in temporal logic.

Example

How does one formalize the informal requirement “φ implies ψ”?

- φ ⇒ ψ? φ ⇒ ψ holds in the initial state
- □(φ ⇒ ψ)? φ ⇒ ψ holds in every state
- φ ⇒ ♦ψ?
- □(φ ⇒ ♦ψ)?
Temporal Logic

Formalization

It can be difficult to correctly formalize informally stated requirements in temporal logic

Example

How does one formalize the informal requirement “φ implies ψ”?

- φ ⇒ ψ? φ ⇒ ψ holds in the initial state
- □(φ ⇒ ψ)? φ ⇒ ψ holds in every state
- φ ⇒ ◇ψ?
- □(φ ⇒ ◇ψ)?
It can be difficult to correctly formalize informally stated requirements in temporal logic.

**Example**

How does one formalize the informal requirement “ϕ implies ψ”?

- \( \varphi \Rightarrow \psi \)? \( \varphi \Rightarrow \psi \) holds in the initial state
- \( \square (\varphi \Rightarrow \psi) \)? \( \varphi \Rightarrow \psi \) holds in every state
- \( \varphi \Rightarrow \Diamond \psi \)? If \( \varphi \) holds in the initial state, \( \psi \) will hold in some state
- \( \square (\varphi \Rightarrow \Diamond \psi) \)?
It can be difficult to correctly formalize informally stated requirements in temporal logic.

**Example**

How does one formalize the informal requirement “ϕ implies ψ”?

- ϕ ⇒ ψ? ϕ ⇒ ψ holds in the initial state
- □(ϕ ⇒ ψ)? ϕ ⇒ ψ holds in every state
- ϕ ⇒ ◇ψ? If ϕ holds in the initial state, ψ will hold in some state
- □(ϕ ⇒ ◇ψ)?
It can be difficult to correctly formalize informally stated requirements in temporal logic

Example

How does one formalize the informal requirement "ϕ implies ψ"?

- ϕ ⇒ ψ? ϕ ⇒ ψ holds in the initial state
- □(ϕ ⇒ ψ)? ϕ ⇒ ψ holds in every state
- ϕ ⇒ ◻ψ? If ϕ holds in the initial state, ψ will hold in some state
- □(ϕ ⇒ ◻ψ)? As above, but iteratively
Temporal Logic
Duals

- For a binary boolean connective $\circ$ (such as $\land$), a binary boolean connective $\bullet$ is its dual if $\neg(\varphi \circ \psi)$ is equivalent to $(\neg\varphi \bullet \neg\psi)$
- Similarly for unary connectives; $\bullet$ is the dual of $\circ$ if $\neg\circ\varphi$ is equivalent to $\bullet\neg\varphi$.
- Duality is symmetrical; if $\bullet$ is the dual of $\circ$ then $\circ$ is the dual of $\bullet$, thus we may refer to two connectives as dual
- $\land$ and $\lor$ are duals; $\neg(\varphi \land \psi)$ is equivalent to $(\neg\varphi \lor \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$?
- Any other?
Temporal Logic

Duals

- For a binary boolean connective $\circ$ (such as $\land$), a binary boolean connective $\bullet$ is its dual if $\neg(\varphi \circ \psi)$ is equivalent to $(\neg \varphi \bullet \neg \psi)$
- Similarly for unary connectives; $\bullet$ is the dual of $\circ$ if $\neg \circ \varphi$ is equivalent to $\bullet \neg \varphi$.
- Duality is symmetrical; if $\bullet$ is the dual of $\circ$ then $\circ$ is the dual of $\bullet$, thus we may refer to two connectives as dual.
- $\land$ and $\lor$ are duals; $\neg(\varphi \land \psi)$ is equivalent to $(\neg \varphi \lor \neg \psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$?

- Any other?
Temporal Logic

Duals

- For a binary boolean connective $\circ$ (such as $\land$), a binary boolean connective $\bullet$ is its dual if $\neg(\varphi \circ \psi)$ is equivalent to $(\neg \varphi \bullet \neg \psi)$
- Similarly for unary connectives; $\bullet$ is the dual of $\circ$ if $\neg \circ \varphi$ is equivalent to $\bullet \neg \varphi$.
- Duality is symmetrical; if $\bullet$ is the dual of $\circ$ then $\circ$ is the dual of $\bullet$, thus we may refer to two connectives as dual
- $\land$ and $\lor$ are duals; $\neg(\varphi \land \psi)$ is equivalent to $(\neg \varphi \lor \neg \psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?
- Any other?
For a binary boolean connective $\circ$ (such as $\land$), a binary boolean connective $\bullet$ is its dual if $\neg (\varphi \circ \psi)$ is equivalent to $(\neg \varphi \bullet \neg \psi)$.

Similarly for unary connectives; $\bullet$ is the dual of $\circ$ if $\neg \circ \varphi$ is equivalent to $\bullet \neg \varphi$.

Duality is symmetrical; if $\bullet$ is the dual of $\circ$ then $\circ$ is the dual of $\bullet$, thus we may refer to two connectives as dual.

$\land$ and $\lor$ are duals; $\neg (\varphi \land \psi)$ is equivalent to $(\neg \varphi \lor \neg \psi)$.

$\neg$ is its own dual.

What is the dual of $\square$? And of $\Diamond$?

Any other?
Temporal Logic
Duals

- For a binary boolean connective $\circ$ (such as $\land$), a binary boolean connective $\bullet$ is its dual if $\neg(\varphi \circ \psi)$ is equivalent to $(\neg \varphi \bullet \neg \psi)$
- Similarly for unary connectives; $\bullet$ is the dual of $\circ$ if $\neg \circ \varphi$ is equivalent to $\bullet \neg \varphi$.
- Duality is symmetrical; if $\bullet$ is the dual of $\circ$ then $\circ$ is the dual of $\bullet$, thus we may refer to two connectives as dual
- $\land$ and $\lor$ are duals; $\neg(\varphi \land \psi)$ is equivalent to $(\neg \varphi \lor \neg \psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?
- $\Box$ and $\Diamond$ are duals: $\neg \Box \varphi \sim \Diamond \neg \varphi$, $\neg \Diamond \varphi \sim \Box \neg \varphi$
- Any other?
Temporal Logic

Duals

- For a binary boolean connective $\circ$ (such as $\land$), a binary boolean connective $\bullet$ is its dual if $\neg(\varphi \circ \psi)$ is equivalent to $(\neg \varphi \bullet \neg \psi)$
- Similarly for unary connectives; $\bullet$ is the dual of $\circ$ if $\neg \circ \varphi$ is equivalent to $\bullet \neg \varphi$.
- Duality is symmetrical; if $\bullet$ is the dual of $\circ$ then $\circ$ is the dual of $\bullet$, thus we may refer to two connectives as dual
- $\land$ and $\lor$ are duals; $\neg(\varphi \land \psi)$ is equivalent to $(\neg \varphi \lor \neg \psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?
- $\Box$ and $\Diamond$ are duals: $\neg \Box \varphi \sim \Diamond \neg \varphi$, $\neg \Diamond \varphi \sim \Box \neg \varphi$
- Any other?
Temporal Logic

Duals

- For a binary boolean connective ◦ (such as ∧), a binary boolean connective • is its dual if ¬(ϕ ◦ ψ) is equivalent to (¬ϕ • ¬ψ)
- Similarly for unary connectives; • is the dual of ◦ if ¬◦ ϕ is equivalent to •¬ϕ.
- Duality is symmetrical; if • is the dual of ◦ then ◦ is the dual of •, thus we may refer to two connectives as dual
- ∧ and ∨ are duals; ¬(ϕ ∧ ψ) is equivalent to (¬ϕ ∨ ¬ψ)
- ¬ is its own dual
- What is the dual of □? And of ◊?
- □ and ◊ are duals: ¬□ ϕ ∼ ◊¬ϕ, ¬◊ ϕ ∼ □¬ϕ
- Any other?
- U and R are duals:
  \[ ¬(ϕ U ψ) ∼ (¬ϕ) R (¬ψ) \]
  \[ ¬(ϕ R ψ) ∼ (¬ϕ) U (¬ψ) \]
Temporal Logic
Classification of Properties

Classification

We can classify a number of properties expressible in LTL:

- safety: $\Box \varphi$
- liveness: $\Diamond \varphi$
- obligation: $\Box \varphi \lor \Diamond \psi$
- recurrence: $\Box \Diamond \varphi$
- persistence: $\Diamond \Box \varphi$
- reactivity: $\Box \Diamond \varphi \lor \Diamond \Box \psi$
We can classify a number of properties expressible in LTL:

- **safety**: $\square \varphi$
- **liveness**: $\lozenge \varphi$
- **obligation**: $\square \varphi \lor \lozenge \psi$
- **recurrence**: $\square \lozenge \varphi$
- **persistence**: $\lozenge \square \varphi$
- **reactivity**: $\square \lozenge \varphi \lor \lozenge \square \psi$
We can classify a number of properties expressible in LTL:

- **safety**: $\square \varphi$
- **liveness**: $\Diamond \varphi$
- **obligation**: $\square \varphi \lor \Diamond \psi$
- **recurrence**: $\square \Diamond \varphi$
- **persistence**: $\Diamond \square \varphi$
- **reactivity**: $\square \Diamond \varphi \lor \Diamond \square \psi$
Temporal Logic
Classification of Properties

Classification

We can classify a number of properties expressible in LTL:

- safety: $\square \varphi$
- liveness: $\Diamond \varphi$
- obligation: $\square \varphi \lor \Diamond \psi$
- recurrence: $\square \Diamond \varphi$
- persistence: $\Diamond \square \varphi$
- reactivity: $\square \Diamond \varphi \lor \Diamond \square \psi$
Temporal Logic

Classification of Properties

Classification

We can classify a number of properties expressible in LTL:

- **safety**  $\Box \varphi$
- **liveness**  $\Diamond \varphi$
- **obligation**  $\Box \varphi \lor \Diamond \psi$
- **recurrence**  $\Box \Diamond \varphi$
- **persistence**  $\Diamond \Box \varphi$
- **reactivity**  $\Box \Diamond \varphi \lor \Diamond \Box \psi$
Temporal Logic
Classification of Properties

Classification

We can classify a number of properties expressible in LTL:

- **safety** $\Box \varphi$
- **liveness** $\Diamond \varphi$
- **obligation** $\Box \varphi \lor \Diamond \psi$
- **recurrence** $\Box \Diamond \varphi$
- **persistence** $\Diamond \Box \varphi$
- **reactivity** $\Box \Diamond \varphi \lor \Diamond \Box \psi$
Temporal Logic
Classification of Properties

Classification

We can classify a number of properties expressible in LTL:

- safety \( \square \varphi \)
- liveness \( \Diamond \varphi \)
- obligation \( \square \varphi \lor \Diamond \psi \)
- recurrence \( \square \Diamond \varphi \)
- persistence \( \Diamond \square \varphi \)
- reactivity \( \square \Diamond \varphi \lor \Diamond \square \psi \)
Plan

1. Temporal Logic
2. Propositional Modal Logic
3. Multimodal Logic
4. Dynamic Logic
5. Mu-calculus
6. Real-Time Logics
Propositional Modal Logic

- The logic of **possibility** and **necessity**
  - $\Box \varphi$: $\varphi$ is “necessarily true”, or “$\varphi$ holds in all possible worlds”
  - $\Diamond \varphi$: $\varphi$ is “possibly true”, or “there is a possible world that realizes $\varphi$”

- The modalities are **dual**
  - $\Diamond \varphi \overset{\text{def}}{=} \neg \Box \neg \varphi$
Propositional Modal Logic
Semantics: Kripke Frames

Definition

A Kripke frame $\mathcal{M}$ is a structure $(W, R, \nu)$ where

- $W$ is a finite non-empty set of states (or worlds) — $W$ is called the universe of $\mathcal{M}$
- $R \subseteq W \times W$ is an accessibility relation between states (transition relation)
- $\nu : \mathcal{P} \rightarrow 2^K$ determines the truth assignment to the atomic propositional variables in each state
Definition

We define the notion that a modal formula $\varphi$ is true in the world $w$ in the model $M$, written $M, w \models \varphi$ as follows:

- $M, w \models p$ iff $w \in \nu(p)$
- $M, w \models \neg \varphi$ iff $M, w \not\models \varphi$
- $M, w \models \varphi_1 \lor \varphi_2$ iff $M, w \models \varphi_1$ or $M, w \models \varphi_2$
- $M, w \models \Box \varphi$ iff $M, w' \models \varphi$ for all $w'$ such that $(w, w') \in R$
- $M, w \models \Diamond \varphi$ iff $M, w' \models \varphi$ for some $w'$ such that $(w, w') \in R$
Propositional Modal Logic

Example (Logic T)

- $R$ reflexive
- $M, w \models \Box \neg p$

\[ \square \neg p \rightarrow \neg p \rightarrow \neg p \]
Example (Logic T)

- $R$ reflexive
- $M, w \models \Box \neg p$

Example (Logic S4)

- $R$ reflexive and transitive
- $M, w \models \Box \neg p$
Remarks

- The semantics is alternatively called relational semantics, frame semantics, world semantics, possible world semantics, Kripke semantics/frame/structure.
- There are different variations of the definition of Kripke semantics.
- Sometimes a Kripke frame is defined to be a structure \((W, R)\), and then the triple \((W, R, \nu)\) is called a Kripke model.
- The Kripke model may be defined as \((W, R, \models)\) instead.
- Sometimes a set of starting states \(W_0 \subseteq W\) is added to the definition.
- In other cases a valuation function \(V : K \rightarrow 2^P\) is given instead of \(\nu\).
- The semantics of \(\Box\) and \(\Diamond\) depend on the properties of \(R\).
  - \(R\) can be reflexive, transitive, euclidean, etc.
  - Axioms and theorems will be determined by \(R\) (or vice-versa!)
Remarks

- The semantics is alternatively called relational semantics, frame semantics, world semantics, possible world semantics, Kripke semantics/frame/structure.
- There are different variations of the definition of Kripke semantics.
- Sometimes a Kripke frame is defined to be a structure \((W, R)\), and then the triple \((W, R, \nu)\) is called a Kripke model.
- The Kripke model may be defined as \((W, R, |\models|)\) instead.
- Sometimes a set of starting states \(W_0 \subseteq W\) is added to the definition.
- In other cases a valuation function \(V : K \to 2^P\) is given instead of \(\nu\).
- The semantics of \(\Box\) and \(\Diamond\) depend on the properties of \(R\):
  - \(R\) can be reflexive, transitive, euclidean, etc.
  - Axioms and theorems will be determined by \(R\) (or vice-versa!).
Remarks

- The semantics is alternatively called relational semantics, frame semantics, world semantics, possible world semantics, Kripke semantics/frame/structure.
- There are different variations of the definition of Kripke semantics.
- Sometimes a Kripke frame is defined to be a structure $(W, R)$, and then the triple $(W, R, \nu)$ is called a Kripke model.
- The Kripke model may be defined as $(W, R, |\models|$ instead.
- Sometimes a set of starting states $W_0 \subseteq W$ is added to the definition.
- In other cases a valuation function $V : K \rightarrow 2^P$ is given instead of $\nu$.
- The semantics of $\Box$ and $\Diamond$ depend on the properties of $R$.
  - $R$ can be reflexive, transitive, euclidean, etc.
  - Axioms and theorems will be determined by $R$ (or vice-versa!).
Plan

1. Temporal Logic
2. Propositional Modal Logic
3. Multimodal Logic
4. Dynamic Logic
5. Mu-calculus
6. Real-Time Logics
A multimodal logic contains a set $A = \{a, \ldots\}$ of modalities.

We can augment propositional logic with one modality for each $a \in A$.

If $\varphi$ is a formula and $a \in A$, then $[a] \varphi$ is a formula.

We also define $\langle a \rangle \varphi \overset{\text{def}}{=} \neg [a] \neg \varphi$.

The semantics of $\langle a \rangle$ and $[a]$ are defined as for $\Diamond a$ and $\Box a$, but “labelling” the transition with $a$. 
Definition

A Kripke frame now is a structure $\mathcal{M} = (W, R, \nu)$ where

- $W$ is a finite non-empty set of states (or worlds) – $W$ is called the universe of $\mathcal{M}$
- $R(a) \subseteq W \times W$ is the accessibility relation between states (transition relation), associating each modality in $a \in A$ to a transition
  - We get a labelled Kripke frame
- $\nu : P \rightarrow 2^K$ determines the truth assignment to the atomic propositional variables in each state
Example

- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$?
- What about $M, w_2 \models [b] \neg p$?
Example

\[ M, w_1 \models [a]p \]

\[ M, w_1 \models \langle a \rangle p \]

\[ M, w_1 \models \langle b \rangle p, \text{ and also } M, w_1 \models [b]p \]

What about \( M, w_2 \models \langle b \rangle \neg p \)?

What about \( M, w_2 \models [b]\neg p \)?
Example

- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$

- What about $M, w_2 \models \langle b \rangle \neg p$?
- What about $M, w_2 \models [b] \neg p$?
Example

- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$?
- What about $M, w_2 \models [b]\neg p$?
Multimodal Logic
Examples

Example

- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$? NO
- What about $M, w_2 \models [b]\neg p$?
Example

- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$? **NO**
- What about $M, w_2 \models [b] \neg p$?
Example

\[ \neg p \quad p \quad p \]

\[ a \quad b \quad a \quad a \]

\[ w_1 \quad w_2 \quad w_3 \]

- \( M, w_1 \models [a]p \)
- \( M, w_1 \models \langle a \rangle p \)
- \( M, w_1 \models \langle b \rangle p \), and also \( M, w_1 \models [b]p \)
- What about \( M, w_2 \models \langle b \rangle \neg p ? \) NO
- What about \( M, w_2 \models [b]\neg p ? \) YES
Plan

1 Temporal Logic
2 Propositional Modal Logic
3 Multimodal Logic
4 Dynamic Logic
5 Mu-calculus
6 Real-Time Logics
The dynamic aspect of modal logic fits well the framework of program execution

- $K$: universe of all possible execution states of a program
- With any program $\alpha$, define a relation $R$ over $K$ s.t. $(s, t) \in R$ iff $t$ is a possible final state of the program $\alpha$ with initial state $s$
  - “possible” since programs may be non-deterministic

Syntactically, each program gives rise to a modality of a multimodal logic

- $\langle \alpha \rangle \varphi$: it is possible to execute $\alpha$ and halt in a state satisfying $\varphi$
- $[\alpha] \varphi$: whenever $\alpha$ halts, it does so in a state satisfying $\varphi$

**Dynamic logic (PDL)** is more than just multimodal logic applied to programs

- It uses various calculi of programs, together with predicate logic, giving rise to a reasoning system for interacting programs

Dynamic logic subsumes Hoare logic
Propositional Dynamic Logic (PDL)

- The dynamic aspect of modal logic fits well the framework of program execution
  - $K$: universe of all possible execution states of a program
  - With any program $\alpha$, define a relation $R$ over $K$ s.t. $(s, t) \in R$ iff $t$ is a possible final state of the program $\alpha$ with initial state $s$
    - “possible” since programs may be non-deterministic
  - Syntactically, each program gives rise to a modality of a multimodal logic
    - $\langle \alpha \rangle \varphi$: it is possible to execute $\alpha$ and halt in a state satisfying $\varphi$
    - $[\alpha] \varphi$: whenever $\alpha$ halts, it does so in a state satisfying $\varphi$

- Dynamic logic (PDL) is more than just multimodal logic applied to programs
  - It uses various calculi of programs, together with predicate logic, giving rise to a reasoning system for interacting programs
  - Dynamic logic subsumes Hoare logic
The dynamic aspect of modal logic fits well the framework of program execution

- $K$: universe of all possible execution states of a program
- With any program $\alpha$, define a relation $R$ over $K$ s.t. $(s, t) \in R$ iff $t$ is a possible final state of the program $\alpha$ with initial state $s$
  - “possible” since programs may be non-deterministic

Syntactically, each program gives rise to a modality of a multimodal logic

- $\langle \alpha \rangle \varphi$: it is possible to execute $\alpha$ and halt in a state satisfying $\varphi$
- $[\alpha] \varphi$: whenever $\alpha$ halts, it does so in a state satisfying $\varphi$

**Dynamic logic (PDL)** is more than just multimodal logic applied to programs

- It uses various calculi of programs, together with predicate logic, giving rise to a reasoning system for interacting programs

Dynamic logic subsumes Hoare logic
Propositional Dynamic Logic (PDL)

- The dynamic aspect of modal logic fits well the framework of program execution
  - $K$: universe of all possible execution states of a program
  - With any program $\alpha$, define a relation $R$ over $K$ s.t. $(s, t) \in R$ iff $t$ is a possible final state of the program $\alpha$ with initial state $s$
    - “possible” since programs may be non-deterministic
  - Syntactically, each program gives rise to a modality of a multimodal logic
    - $\langle \alpha \rangle \varphi$: it is possible to execute $\alpha$ and halt in a state satisfying $\varphi$
    - $[\alpha] \varphi$: whenever $\alpha$ halts, it does so in a state satisfying $\varphi$

Dynamic logic (PDL) is more than just multimodal logic applied to programs
- It uses various calculi of programs, together with predicate logic, giving rise to a reasoning system for interacting programs
- Dynamic logic subsumes Hoare logic
Propositional Dynamic Logic

Syntax

- **PDL** contains syntax constructs from:
  - Propositional logic
  - Modal logic
  - Algebra of regular expressions

- **Expressions** are of two sorts
  - Propositions and formulas: $\varphi, \psi, \ldots$
  - Programs: $\alpha, \beta, \gamma, \ldots$
Propositional Dynamic Logic
Syntax

Definition

Programs and propositions of regular PDL are built inductively using the following operators

- Propositional operators
  - $\rightarrow$ implication
  - $0$ falsity

- Program operators
  - $;$ composition
  - $\cup$ choice
  - $*$ iteration

- Mixed operators
  - $[\ ]$ necessity
  - $?\ $ test
Propositional Dynamic Logic

Intuitive Meaning

- $[\alpha]\varphi$: It is necessary that after executing $\alpha$, $\varphi$ is true (necessity)
- $\alpha \cup \beta$: Choose either $\alpha$ or $\beta$ non-deterministically and execute it (choice)
- $\alpha; \beta$: Execute $\alpha$, then execute $\beta$ (concatenation, sequencing)
- $\alpha^*$: Execute $\alpha$ a non-deterministically chosen finite of times –zero or more (Kleene star)
- $\varphi?$: Test $\varphi$; proceed if true, fail if false (test)

We define $\langle \alpha \rangle \varphi \overset{\text{def}}{=} \neg [\alpha] \neg \varphi$
[\alpha]\varphi: It is necessary that after executing \alpha, \varphi is true (necessity)

\alpha \cup \beta: Choose either \alpha or \beta non-deterministically and execute it (choice)

\alpha; \beta: Execute \alpha, then execute \beta (concatenation, sequencing)

\alpha^*: Execute \alpha a non-deterministically chosen finite of times –zero or more (Kleene star)

\varphi?: Test \varphi; proceed if true, fail if false (test)

We define \langle \alpha \rangle \varphi \overset{\text{def}}{=} \neg [\alpha] \neg \varphi
Propositional Dynamic Logic

Intuitive Meaning

- $[\alpha] \varphi$: It is necessary that after executing $\alpha$, $\varphi$ is true (necessity)
- $\alpha \cup \beta$: Choose either $\alpha$ or $\beta$ non-deterministically and execute it (choice)
- $\alpha; \beta$: Execute $\alpha$, then execute $\beta$ (concatenation, sequencing)
- $\alpha^*$: Execute $\alpha$ a non-deterministically chosen finite of times –zero or more (Kleene star)
- $\varphi?$: Test $\varphi$; proceed if true, fail if false (test)

We define $\langle \alpha \rangle \varphi \overset{\text{def}}{=} \neg [\alpha] \neg \varphi$
- $[\alpha] \varphi$: It is necessary that after executing $\alpha$, $\varphi$ is true (necessity)
- $\alpha \cup \beta$: Choose either $\alpha$ or $\beta$ non-deterministically and execute it (choice)
- $\alpha; \beta$: Execute $\alpha$, then execute $\beta$ (concatenation, sequencing)
- $\alpha^*$: Execute $\alpha$ a non-deterministically chosen finite number of times – zero or more (Kleene star)
- $\varphi?$: Test $\varphi$; proceed if true, fail if false (test)

We define $\langle \alpha \rangle \varphi \overset{\text{def}}{=} \neg [\alpha] \neg \varphi$
Propositional Dynamic Logic

Intuitive Meaning

- $[\alpha]\varphi$: It is necessary that after executing $\alpha$, $\varphi$ is true (necessity)
- $\alpha \cup \beta$: Choose either $\alpha$ or $\beta$ non-deterministically and execute it (choice)
- $\alpha; \beta$: Execute $\alpha$, then execute $\beta$ (concatenation, sequencing)
- $\alpha^*$: Execute $\alpha$ a non-deterministically chosen finite of times –zero or more (Kleene star)
- $\varphi?$: Test $\varphi$; proceed if true, fail if false (test)

We define $\langle \alpha \rangle \varphi \overset{\text{def}}{=} \neg [\alpha] \neg \varphi$
- $[\alpha]\varphi$: It is necessary that after executing $\alpha$, $\varphi$ is true (necessity)
- $\alpha \cup \beta$: Choose either $\alpha$ or $\beta$ non-deterministically and execute it (choice)
- $\alpha; \beta$: Execute $\alpha$, then execute $\beta$ (concatenation, sequencing)
- $\alpha^*$: Execute $\alpha$ a non-deterministically chosen finite of times –zero or more (Kleene star)
- $\varphi?$: Test $\varphi$; proceed if true, fail if false (test)

We define $\langle \alpha \rangle \varphi \overset{\text{def}}{=} \neg [\alpha] \neg \varphi$
Propositional Dynamic Logic

Additional Programs

\[
\begin{align*}
\text{skip} & \quad \text{def} \quad 1? \\
\text{fail} & \quad \text{def} \quad 0? \\
\text{if } \varphi_1 \rightarrow \alpha_1 \mid \ldots \mid \varphi_n \rightarrow \alpha_n \text{ fi} & \quad \text{def} \quad \varphi_1?; \alpha_1 \cup \ldots \cup \varphi_n?; \alpha_n \\
\text{do } \varphi_1 \rightarrow \alpha_1 \mid \ldots \mid \varphi_n \rightarrow \alpha_n \text{ od} & \quad \text{def} \quad (\varphi_1?; \alpha_1 \cup \ldots \cup \varphi_n?; \alpha_n)^*; (\neg \varphi_1 \wedge \ldots \wedge \neg \varphi_n) \\
\text{if } \varphi \text{ then } \alpha \text{ else } \beta & \quad \text{def} \quad \text{if } \varphi \rightarrow \alpha \mid \neg \varphi \rightarrow \beta \text{ fi} \\
& \quad \quad = \varphi?; \alpha \cup \neg \varphi?; \beta \\
\text{while } \varphi \text{ do } \alpha & \quad \text{def} \quad \text{do } \varphi \rightarrow \alpha \text{ od} \\
& \quad \quad = (\varphi?; \alpha)^*; \neg \varphi? \\
\text{repeat } \alpha \text{ until } \varphi & \quad \text{def} \quad \alpha; \text{while } \neg \varphi \text{ do } \alpha \text{ od} \\
& \quad \quad = \alpha; (\neg \varphi?; \alpha)^*; \varphi? \\
\{ \varphi \} \alpha \{ \psi \} & \quad \text{def} \quad \varphi \rightarrow [\alpha] \psi
\end{align*}
\]
Remark

- It is possible to reason about programs by using PDF proof system
- We will not see the semantics here
- The semantics of PDL comes from that from modal logic
  - Kripke frames
- We will see its application in our contract language
- **µ-calculus** is a powerful language to express properties of transition systems by using least and greatest fixpoint operators
  - \( \nu \) is the greatest fixpoint meaning **looping**
  - \( \mu \) is the least fixpoint meaning **finite looping**
- Many temporal and program logics can be encoded into the \( \mu \)-calculus
- Efficient model checking algorithms
- Formulas are interpreted relative to a transition system
  - The Kripke structure needs to be slightly modified
$\mu$-calculus: Syntax

- Let $\text{Var} = \{Z, Y, \ldots\}$ be an (infinite) set of \textit{variable names}
- Let $\text{Prop} = \{P, Q, \ldots\}$ be a set of \textit{atomic propositions}
- Let $L = \{a, b, \ldots\}$ be a set of \textit{labels} (or \textit{actions})

**Definition**

The set of $\mu$-calculus formulae (w.r.t. $(\text{Var}, \text{Prop}, L)$) is defined as follows:

- $P$ is a formula
- $Z$ is a formula
- If $\phi_1$ and $\phi_2$ are formulae, so is $\phi_1 \land \phi_2$
- If $\phi$ is a formula, so is $[a]\phi$
- If $\phi$ is a formula, so is $\neg\phi$
- If $\phi$ is a formula, then $\nu Z.\phi$ is a formula
  - Provided every \textit{free} occurrence of $Z$ in $\phi$ occurs positively (within the scope of an even number of negations)
  - $\nu$ is the only binding operator
\( \mu \)-calculus: Syntax

- If \( \phi(Z) \), then the subsequent writing \( \phi(\psi) \) means \( \phi \) with \( \psi \) substituted for all free occurrences of \( Z \)
- The positivity requirement syntactically guarantees monotonicity in \( Z \)
  - Unique minimal and maximal fixpoint
- Derived operators
  - \( \phi_1 \lor \phi_2 \overset{\text{def}}{=} \neg(\neg \phi_1 \land \neg \phi_2) \)
  - \( \langle a \rangle \phi \overset{\text{def}}{=} \neg[a] \neg \phi \)
  - \( \mu Z. \phi(Z) \overset{\text{def}}{=} \neg \nu Z. \neg \phi(\neg Z) \)
If $\phi(Z)$, then the subsequent writing $\phi(\psi)$ means $\phi$ with $\psi$ substituted for all free occurrences of $Z$.

- The positivity requirement syntactically guarantees monotonicity in $Z$.
  - Unique minimal and maximal fixpoint

- Derived operators
  - $\phi_1 \lor \phi_2 \overset{\text{def}}{=} \neg(\neg\phi_1 \land \neg\phi_2)$
  - $\langle a \rangle \phi \overset{\text{def}}{=} \neg[a] \neg\phi$
  - $\mu Z.\phi(Z) \overset{\text{def}}{=} \neg\nu Z.\neg\phi(\neg Z)$
μ-calculus: Syntax

- If $\phi(Z)$, then the subsequent writing $\phi(\psi)$ means $\phi$ with $\psi$ substituted for all free occurrences of $Z$.
- The positivity requirement syntactically guarantees monotonicity in $Z$.
  - Unique minimal and maximal fixpoint.
- Derived operators:
  - $\phi_1 \lor \phi_2 \overset{\text{def}}{=} \neg(\neg\phi_1 \land \neg\phi_2)$
  - $\langle a \rangle \phi \overset{\text{def}}{=} \neg[a] \neg\phi$
  - $\mu Z. \phi(Z) \overset{\text{def}}{=} \neg \nu Z. \neg\phi(\neg Z)$
A labelled transition system (LTS) is a triple \( M = (S, T, L) \), where:

- \( S \) is a nonempty set of states
- \( L \) is a set of labels (actions) as defined before
- \( T \subseteq S \times L \times S \) is a transition relation

A modal \( \mu \)-calculus structure \( T \) (over \( Prop \) and \( L \)) is a LTS \( (S, T, L) \) together with an interpretation \( \mathcal{V}_{prop} : Prop \rightarrow 2^S \) for the atomic propositions
\( \mu \)-calculus

Semantics

**Definition**

Given a structure \( \mathcal{T} \) and an interpretation \( \mathcal{V} : \text{Var} \rightarrow 2^S \) of the variables, the set \( \| \phi \|_{\mathcal{T}}^\mathcal{V} \) is defined as follows:

\[
\begin{align*}
\| P \|_{\mathcal{T}}^\mathcal{V} & = \mathcal{V}_{\text{Prop}}(P) \\
\| Z \|_{\mathcal{T}}^\mathcal{V} & = \mathcal{V}(Z) \\
\| \neg \phi \|_{\mathcal{T}}^\mathcal{V} & = S - \| \phi \|_{\mathcal{T}}^\mathcal{V} \\
\| \phi_1 \land \phi_2 \|_{\mathcal{T}}^\mathcal{V} & = \| \phi_1 \|_{\mathcal{T}}^\mathcal{V} \cap \| \phi_2 \|_{\mathcal{T}}^\mathcal{V} \\
\| [a] \phi \|_{\mathcal{T}}^\mathcal{V} & = \{ s \mid \forall t.(s, a, t) \in T \Rightarrow t \in \| \phi \|_{\mathcal{T}}^\mathcal{V} \} \\
\| \nu Z. \phi \|_{\mathcal{T}}^\mathcal{V} & = \bigcup \{ S \subseteq S \mid S \subseteq \| \phi \|_{\mathcal{T}}^\mathcal{V[Z:=S]} \}
\end{align*}
\]

where \( \mathcal{V}[Z := S] \) is the valuation mapping \( Z \) to \( S \) and otherwise agrees with \( \mathcal{V} \).
If we consider only positive formulae, we may add the following derived operators

**Interpretation**

\[
\begin{align*}
\| \phi_1 \lor \phi_2 \|_V &= \| \phi_1 \|_V \cup \| \phi_2 \|_V \\
\| \langle a \rangle \phi \|_V &= \{ s \mid \exists t. (s, a, t) \in T \land t \in \| \phi \|_V \} \\
\| \mu Z. \phi \|_V &= \bigcap \{ S \subseteq S \mid S \supseteq \| \phi \|_V[Z:=s] \}
\end{align*}
\]
\( \mu \)-calculus

Example

- \( \mu \) is liveness
  - “On all length \( a \)-path, \( P \) eventually holds”
    \[ \mu Z. (P \lor [a]Z) \]
  - “On some \( a \)-path, \( P \) holds until \( Q \) holds”
    \[ \mu Z. (Q \lor (P \land \langle a \rangle Z)) \]

- \( \nu \) is safety
  - “\( P \) is true along every \( a \)-path”
    \[ \nu Z. (P \land [a]Z) \]
  - “On every \( a \)-path \( P \) holds while \( Q \) fails”
    \[ \nu Z. (Q \lor (P \land [a]Z)) \]
Plan

1. Temporal Logic
2. Propositional Modal Logic
3. Multimodal Logic
4. Dynamic Logic
5. Mu-calculus
6. Real-Time Logics
Temporal logic (TL) is concerned with the **qualitative** aspect of temporal system requirements

- Invariance, responsiveness, etc

TL cannot refer to metric time: Not suitable for the specification of **quantitative** temporal requirements

There are many ways to extend a temporal logic with **real-time**

1. Replace the unrestricted temporal operators with **time-bounded** versions
2. Extend temporal logic with explicit references to the times of temporal contexts (**freeze quantification**)
3. Add an explicit clock variable
Real-time Logics

- Temporal logic (TL) is concerned with the **qualitative** aspect of temporal system requirements
  - Invariance, responsiveness, etc
- TL cannot refer to metric time: Not suitable for the specification of **quantitative** temporal requirements
- There are many ways to extend a temporal logic with **real-time**
  1. Replace the unrestricted temporal operators with **time-bounded versions**
  2. Extend temporal logic with explicit references to the times of temporal contexts (**freeze quantification**)
  3. Add an explicit clock variable
Real-time Logics

- Temporal logic (TL) is concerned with the **qualitative** aspect of temporal system requirements
  - Invariance, responsiveness, etc
- TL cannot refer to metric time: Not suitable for the specification of **quantitative** temporal requirements
- There are many ways to extend a temporal logic with **real-time**
  1. Replace the unrestricted temporal operators with **time-bounded** versions
  2. Extend temporal logic with explicit references to the times of temporal contexts (**freeze quantification**)
  3. Add an explicit **clock variable**
Real-time Logics
1. Bounded Temporal Operators

Example of a R-T logic with bounded temporal operators

\[ \varphi := p \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \mathcal{U}_I \varphi \]

where \( p \) is a propositional variable, and \( I \) is a rational interval

- Informally, \( \varphi_1 \mathcal{U}_I \varphi_2 \) holds at time \( t \) in a timed observation sequence iff
  - There is a later time \( t' \in t + I \) s.t. \( \varphi_2 \) holds at time \( t' \) and \( \varphi_1 \) holds through the interval \((t, t')\)

- Derived operators
  - \( \Diamond_I \varphi \overset{\text{def}}{=} \text{true} \mathcal{U}_I \varphi \): time-bounded eventually
  - \( \Box_I \varphi \overset{\text{def}}{=} \neg \Diamond_I \neg \varphi \): time-bounded always
Example of a R-T logic with bounded temporal operators

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi U_I \varphi \]

where \( p \) is a propositional variable, and \( I \) is a rational interval

- Informally, \( \varphi_1 U_I \varphi_2 \) holds at time \( t \) in a timed observation sequence iff
  - There is a later time \( t' \in t + I \) s.t. \( \varphi_2 \) holds at time \( t' \) and \( \varphi_1 \) holds through the interval \((t, t')\)

- Derived operators
  - \( \Diamond_I \varphi \overset{\text{def}}{=} \text{true} U_I \varphi \): time-bounded eventually
  - \( \Box_I \varphi \overset{\text{def}}{=} \neg \Diamond_I \neg \varphi \): time-bounded always
Example of a R-T logic with bounded temporal operators

\[ \varphi := p | \neg \varphi | \varphi \land \varphi | \varphi \mathcal{U}_I \varphi \]

where \( p \) is a propositional variable, and \( I \) is a rational interval

- Informally, \( \varphi_1 \mathcal{U}_I \varphi_2 \) holds at time \( t \) in a timed observation sequence iff
  - There is a later time \( t' \in t + I \) s.t. \( \varphi_2 \) holds at time \( t' \) and \( \varphi_1 \) holds through the interval \( (t, t') \)
- Derived operators
  - \( \Diamond_I \varphi \overset{\text{def}}{=} \text{true} \mathcal{U}_I \varphi \): time-bounded eventually
  - \( \Box_I \varphi \overset{\text{def}}{=} \neg \Diamond_I \neg \varphi \): time-bounded always
Real-time Logics
1. Bounded Temporal Operators

Example of a R-T logic with bounded temporal operators

\[ \varphi := p \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi U_I \varphi \]

where \( p \) is a propositional variable, and \( I \) is a rational interval

- Informally, \( \varphi_1 U_I \varphi_2 \) holds at time \( t \) in a timed observation sequence iff
  - There is a later time \( t' \in t + I \) s.t. \( \varphi_2 \) holds at time \( t' \) and \( \varphi_1 \) holds through the interval \((t, t')\)

- Derived operators
  - \( \Diamond_I \varphi \overset{\text{def}}{=} \text{true} U_I \varphi \): time-bounded eventually
  - \( \Box_I \varphi \overset{\text{def}}{=} \neg \Diamond_I \neg \varphi \): time-bounded always

Example

- \( \Box_{[2,4]} p \) means “\( p \) holds at all times within 2 to 4 time units”
- \( \Box (p \Rightarrow \Diamond_{[0,3]} q) \): “every stimulus \( p \) is followed by a response \( q \) within 3 time units”
Real-time Logics

2. Freeze Quantification

- Bounded-operator cannot express non-local timing requirements
  - Ex: “every stimulus $p$ is followed by a response $q$, followed by another response $r$, such that $r$ is within 3 time units of $p$”
- Need to have explicit references to time of temporal contexts
- The freeze quantifier $x.$ binds $x$ to the time of the current temporal context
  - $x.\varphi(x)$ holds at time $t$ iff $\varphi(t)$ does
- A logic with freeze quantifier is called half-order
Real-time Logics
2. Freeze Quantification

- Bounded-operator cannot express *non-local* timing requirements
  - Ex: “every stimulus $p$ is followed by a response $q$, followed by another response $r$, such that $r$ is within 3 time units of $p$”

- Need to have explicit references to time of temporal contexts

- The freeze quantifier $x.$ binds $x$ to the time of the current temporal context
  - $x.\varphi(x)$ holds at time $t$ iff $\varphi(t)$ does

- A logic with freeze quantifier is called *half-order*
Real-time Logics
2. Freeze Quantification

- Bounded-operator cannot express *non-local* timing requirements
  - Ex: “every stimulus \( p \) is followed by a response \( q \), followed by another response \( r \), such that \( r \) is within 3 time units of \( p \)”

- Need to have explicit references to time of temporal contexts

- The **freeze quantifier** \( x. \) binds \( x \) to the time of the current temporal context
  - \( x.\phi(x) \) holds at time \( t \) iff \( \phi(t) \) does

- A logic with freeze quantifier is called **half-order**
Real-time Logics
2. Freeze Quantification

- Bounded-operator cannot express *non-local* timing requirements
  - Ex: “every stimulus $p$ is followed by a response $q$, followed by another response $r$, such that $r$ is within 3 time units of $p$”

- Need to have explicit references to time of temporal contexts
- The **freeze quantifier** $x.$ binds $x$ to the time of the current temporal context
  - $x.\varphi(x)$ holds at time $t$ iff $\varphi(t)$ does

- A logic with freeze quantifier is called **half-order**

---

**Example of a R-T logic with freeze quantification**

\[
\varphi := p \mid \pi \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi U \varphi \mid x.\varphi
\]

- $V$ is a set of time variables
- $\pi \in \Pi(V)$ represents atomic timing constraints with free variables from $V$ (e.g., $z \leq x + 3$)
Real-time Logics

2. Freeze Quantification

Example

“Every stimulus $p$ is followed by a response $q$ within 3 time units”

$$\Box x. (p \Rightarrow \Diamond y. (q \land y \leq x + 3))$$
Example

- “Every stimulus $p$ is followed by a response $q$ within 3 time units”
  \[
  \Box x. (p \Rightarrow \Diamond y. (q \land y \leq x + 3))
  \]

- “Every stimulus $p$ is followed by a response $q$, followed by another response $r$, such that $r$ is within 3 time units of $p$”
  \[
  \Box x. (p \Rightarrow \Diamond (q \land \Diamond z. (r \land z \leq x + 3)))
  \]
Real-time Logics

3. Explicit Clock Variable

- It uses a dynamic state variable $T$ (the clock variable), and
- A first-order quantification for global (rigid) variables over time
Real-time Logics
3. Explicit Clock Variable

- It uses a dynamic state variable $T$ (the clock variable), and
- A first-order quantification for global (rigid) variables over time

Example of a R-T logic with explicit clocks

$$\varphi ::= p \mid \pi \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \cup \varphi \mid \exists x. \varphi$$

- $x \in V$, with $V$ a set of (global) time variables
- $\pi \in \prod(V \cup \{T\})$ represents atomic timing constraints over the variables from $V \cup \{T\}$ (e.g., $T \leq x + 3$)

The freeze quantifier $x.\varphi$ is equivalent to $\exists x.(T = x \land \varphi)$
Real-time Logics
3. Explicit Clock Variable

- It uses a dynamic state variable \( T \) (the clock variable), and
- A first-order quantification for global (rigid) variables over time

### Example of a R-T logic with explicit clocks

\[
\varphi ::= p \mid \pi \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi U \varphi \mid \exists x. \varphi
\]

- \( x \in V \), with \( V \) a set of (global) time variables
- \( \pi \in \Pi(V \cup \{ T \}) \) represents atomic timing constraints over the variables from \( V \cup \{ T \} \) (e.g., \( T \leq x + 3 \))

The freeze quantifier \( x.\varphi \) is equivalent to \( \exists x.(T = x \land \varphi) \)

### Example

- “Every stimulus \( p \) is followed by a response \( q \) within 3 time units”

\[
\forall x. \Box((p \land T = x) \Rightarrow \diamond(q \land T \leq x + 3))
\]
Real-time Logics

Examples of Real-Time Logics

Linear-time:

- **MTL** (metric temporal logic)
  - A propositional bounded-operator logic
- **TPTL** (timed temporal logic)
  - A propositional half-order logic using only the future operators *until* and *next*
- **RTTL** (real-time temporal logic)
  - A first-order explicit-clock logic
- **XCTL** (explicit-clock temporal logic)
  - A propositional explicit-clock logic with a rich timing constraints (comparison and addition)
  - Does not allow explicit quantification over time variables (implicit universal quantification)
- **MITL** (metric interval temporal logic)
  - A propositional linear-time with an interval-based strictly-monotonic real-time semantics
  - Does not allow equality constraints
Real-time Logics

Examples of Real-Time Logics

Branching-time:

- **RTCTL** (real-time computation tree logic)
  - A propositional branching-time logic for synchronous systems
  - Bounded-operator extension of CTL with a point-based strictly-monotonic integer-time semantics

- **TCTL** (timed computation tree logic)
  - A propositional branching-time logic with less restricted semantics
  - Bounded-operator extension of CTL with an interval-based strictly-monotonic real-time semantics
Final Remarks

For most of the presented logics, there is an axiomatic system, and/or a Natural Deduction system.

Though important, it is not needed for the rest of the tutorial.

- Our contract language will use the syntax of some of the presented logics.
- We will focus on the semantics (Kripke models, semantic encoding into other logic).
Further Reading

Modal and Temporal Logics


Dynamic Logic


$\mu$-calculus:

- J. Bradfield and C. Stirling. *Modal logics and $\mu$-calculi: an introduction*

Real-time logics: