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 ($\mathcal{CL}$)
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
  - $\Box\varphi$: φ is necessarily true.
  - $\Diamond\varphi$: φ 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
  - $\Box\varphi$: φ is necessarily true.
  - $\Diamond\varphi$: φ 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	φ will always hold
doxastic	I believe φ
epistemic	I know φ
deontic	It ought to be the case that φ

- 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

Modal Logics

We will see

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

- Temporal logic
- Propositional modal logic
- Multimodal logic
- Dynamic logic
- μ-calculus
- Real-time logics

In the following lecture (1 hour):

- Deontic logic

Plan

Plan

- 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*,

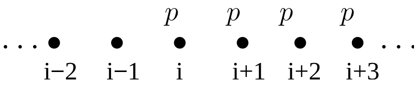
- p holds in i and every following point (the future)
- p holds in i and every preceding point (the past)

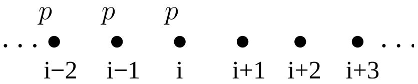
We will only be concerned with the future

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





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

Interpretation

$\Box\,\varphi$	φ will *always* (in every state) hold
$\Diamond\,\varphi$	φ will *eventually* (ins some state) hold
$\bigcirc\,\varphi$	φ will hold at the *next* point in time
$\varphi\,\mathcal{U}\,\psi$	ψ will eventually hold, and *until* that point φ will hold
$\varphi\,R\,\psi$	ψ holds until (incl.) the point (if any) where φ holds (*release*)
$\varphi\,W\,\psi$	φ will hold until ψ holds (*weak until* or *waiting for*)

Definition

We define LTL formulae as follows:

- $\mathcal{L} \subseteq \mathcal{L}_{\mathrm{T}}$: first-order formulae are also LTL formulae
- If φ is an LTL formulae, so are

$$\Box\,\varphi,\ \Diamond\,\varphi,\ \bigcirc\,\varphi \text{ and } \neg\varphi$$

- If φ and ψ are LTL formulae, so are

$$\varphi\,\mathcal{U}\,\psi,\ \varphi\,R\,\psi,\ \varphi\,W\,\psi,\ \varphi\vee\psi,\ \varphi\wedge\psi,\ \varphi \Rightarrow \psi \text{ and } \varphi\equiv\psi$$

Definition

- A path is an infinite sequence of states

$$\sigma = s_0, s_1, s_2, \ldots$$

- σ^k denotes the *path* $s_k, s_{k+1}, s_{k+2}, \ldots$
- σ_k denotes the *state* s_k
- All computations are paths, but not vice versa

Definition

We define the notion that an LTL formula φ is true (false) relative to a path σ, written $\sigma \models \varphi$ ($\sigma \not\models \varphi$) as follows.

$$
\begin{array}{lll}
\sigma \models \varphi & \text{iff} & \sigma_0 \models \varphi \text{ when } \varphi \in \mathcal{L} \\
\sigma \models \neg\varphi & \text{iff} & \sigma \not\models \varphi \\
\sigma \models \varphi \vee \psi & \text{iff} & \sigma \models \varphi \text{ or } \sigma \models \psi \\
 & & \\
\sigma \models \Box\varphi & \text{iff} & \sigma^k \models \varphi \text{ for all } k \geq 0 \\
\sigma \models \Diamond\varphi & \text{iff} & \sigma^k \models \varphi \text{ for some } k \geq 0 \\
\sigma \models \bigcirc\varphi & \text{iff} & \sigma^1 \models \varphi
\end{array}
$$

(cont.)

Definition

(cont.)

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

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

$$\sigma \models \varphi \, W \, \psi \quad \text{iff} \quad \sigma \models \varphi \,\mathcal{U}\, \psi \text{ or } \sigma \models \Box \varphi$$

Definition

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

$$\models \varphi \qquad \text{(Validity)}$$

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

$$\varphi \sim \psi \qquad \text{(Equivalence)}$$

Definition

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

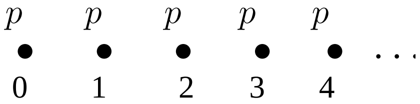
$$\models \varphi \qquad \text{(Validity)}$$

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

$$\varphi \sim \psi \qquad \text{(Equivalence)}$$

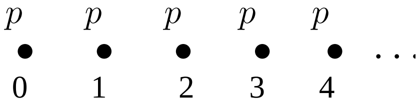
Temporal Logic
Semantics

$\sigma \models \Box p$

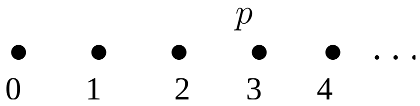


Temporal Logic
Semantics

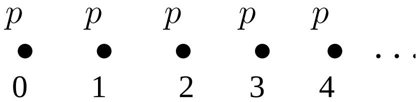
$\sigma \models \Box\, p$



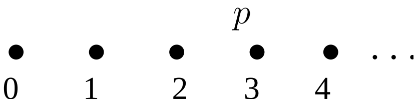
$\sigma \models \Diamond\, p$



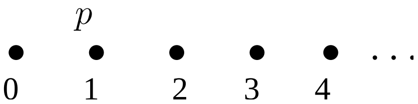
$\sigma \models \Box\, p$



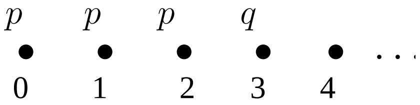
$\sigma \models \Diamond\, p$



$\sigma \models \bigcirc\, p$



$\sigma \models p\,\mathcal{U}\,q$ – The sequence of p is finite



Temporal Logic
Semantics

$\sigma \models p \, \mathcal{U} \, q$ – The sequence of p is finite

p	p	p	q		
$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\ldots$
0	1	2	3	4	

$\sigma \models p \, R \, q$ – The sequence of q may be infinite

q	q	q	q, p		
$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\ldots$
0	1	2	3	4	

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

$$\begin{array}{ccccccc} p & p & p & q & & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\ 0 & 1 & 2 & 3 & 4 & \end{array}$$

$\sigma \models p R q$ – The sequence of q may be infinite

$$\begin{array}{ccccccc} q & q & q & q,p & & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\ 0 & 1 & 2 & 3 & 4 & \end{array}$$

$\sigma \models p W q$ – The sequence of p may be infinite ($p W q \equiv (p \mathcal{U} q) \vee \Box p$)

$$\begin{array}{ccccccc} p & p & p & q & & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\ 0 & 1 & 2 & 3 & 4 & \end{array}$$

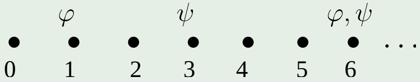
Example (Response)

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

Example (Response)

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

Every φ-position coincides with or is followed by a ψ-position



Example (Response)

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

Every φ-position coincides with or is followed by a ψ-position

$$\begin{array}{cccccccc} & \varphi & & \psi & & & \varphi,\psi & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & \end{array}$$

This formula will also hold in every path where φ never holds

$$\begin{array}{cccccc} \neg\varphi & \neg\varphi & \neg\varphi & \neg\varphi & \neg\varphi & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \ldots \\ 0 & 1 & 2 & 3 & 4 & \end{array}$$

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"?

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

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

Example

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

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

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

Example

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

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

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

Example

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

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

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

Example

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

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

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

Example

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

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

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

Example

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

- $\varphi \Rightarrow \psi$? $\varphi \Rightarrow \psi$ holds in the initial state
- $\Box(\varphi \Rightarrow \psi)$? $\varphi \Rightarrow \psi$ holds in every state
- $\varphi \Rightarrow \Diamond\psi$? If φ holds in the initial state, ψ will hold in some state
- $\Box(\varphi \Rightarrow \Diamond\psi)$?

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

Example

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

- $\varphi \Rightarrow \psi$? $\varphi \Rightarrow \psi$ holds in the initial state
- $\Box(\varphi \Rightarrow \psi)$? $\varphi \Rightarrow \psi$ holds in every state
- $\varphi \Rightarrow \Diamond\psi$? If φ holds in the initial state, ψ will hold in some state
- $\Box(\varphi \Rightarrow \Diamond\psi)$?

Temporal Logic
Formalization

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

Example

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

- $\varphi \Rightarrow \psi$? $\varphi \Rightarrow \psi$ holds in the initial state
- $\Box(\varphi \Rightarrow \psi)$? $\varphi \Rightarrow \psi$ holds in every state
- $\varphi \Rightarrow \Diamond\psi$? If φ holds in the initial state, ψ will hold in some state
- $\Box(\varphi \Rightarrow \Diamond\psi)$? As above, but iteratively

- For a binary boolean connective $\circ$ (such as $\wedge$), 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
- $\wedge$ and $\vee$ are duals; $\neg(\varphi \wedge \psi)$ is equivalent to $(\neg\varphi \vee \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$?

- Any other?

Temporal Logic

Duals

- For a binary boolean connective $\circ$ (such as $\wedge$), 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
- $\wedge$ and $\vee$ are duals; $\neg(\varphi \wedge \psi)$ is equivalent to $(\neg\varphi \vee \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$?
- Any other?

Temporal Logic
Duals

- For a binary boolean connective $\circ$ (such as $\wedge$), 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
- $\wedge$ and $\vee$ are duals; $\neg(\varphi \wedge \psi)$ is equivalent to $(\neg\varphi \vee \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?
- Any other?

- For a binary boolean connective $\circ$ (such as $\wedge$), 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
- $\wedge$ and $\vee$ are duals; $\neg(\varphi \wedge \psi)$ is equivalent to $(\neg\varphi \vee \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?

- Any other?

Temporal Logic

Duals

- For a binary boolean connective $\circ$ (such as $\wedge$), 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
- $\wedge$ and $\vee$ are duals; $\neg(\varphi \wedge \psi)$ is equivalent to $(\neg\varphi \vee \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?
- $\Box$ and $\Diamond$ are duals: $\neg \Box \varphi \sim \Diamond \neg\varphi, \quad \neg \Diamond \varphi \sim \Box \neg\varphi$
- Any other?

- For a binary boolean connective $\circ$ (such as $\wedge$), 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
- $\wedge$ and $\vee$ are duals; $\neg(\varphi \wedge \psi)$ is equivalent to $(\neg\varphi \vee \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?
- $\Box$ and $\Diamond$ are duals: $\neg \Box \varphi \sim \Diamond \neg\varphi, \quad \neg \Diamond \varphi \sim \Box \neg\varphi$
- Any other?

Temporal Logic

Duals

- For a binary boolean connective $\circ$ (such as $\wedge$), 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
- $\wedge$ and $\vee$ are duals; $\neg(\varphi \wedge \psi)$ is equivalent to $(\neg\varphi \vee \neg\psi)$
- $\neg$ is its own dual
- What is the dual of $\Box$? And of $\Diamond$?
- $\Box$ and $\Diamond$ are duals: $\neg \Box \varphi \sim \Diamond \neg\varphi$, $\quad \neg \Diamond \varphi \sim \Box \neg\varphi$
- Any other?
- U and R are duals:

$$\neg(\varphi \mathcal{U} \psi) \sim (\neg\varphi)\, R\, (\neg\psi)$$
$$\neg(\varphi\, R\, \psi) \sim (\neg\varphi) \mathcal{U} (\neg\psi)$$

Classification

We can classify a number of properties expressible in LTL:

safety	$\square \varphi$
liveness	$\diamond \varphi$
obligation	$\square \varphi \vee \diamond \psi$
recurrence	$\square \diamond \varphi$
persistence	$\diamond \square \varphi$
reactivity	$\square \diamond \varphi \vee \diamond \square \psi$

Classification

We can classify a number of properties expressible in LTL:

safety	$\Box\,\varphi$
liveness	$\Diamond\,\varphi$
obligation	$\Box\,\varphi \vee \Diamond\,\psi$
recurrence	$\Box\Diamond\,\varphi$
persistence	$\Diamond\Box\,\varphi$
reactivity	$\Box\Diamond\,\varphi \vee \Diamond\Box\,\psi$

Classification

We can classify a number of properties expressible in LTL:

safety	$\Box \varphi$
liveness	$\Diamond \varphi$
obligation	$\Box \varphi \vee \Diamond \psi$
recurrence	$\Box \Diamond \varphi$
persistence	$\Diamond \Box \varphi$
reactivity	$\Box \Diamond \varphi \vee \Diamond \Box \psi$

Classification

We can classify a number of properties expressible in LTL:

safety	$\Box \varphi$
liveness	$\Diamond \varphi$
obligation	$\Box \varphi \vee \Diamond \psi$
recurrence	$\Box \Diamond \varphi$
persistence	$\Diamond \Box \varphi$
reactivity	$\Box \Diamond \varphi \vee \Diamond \Box \psi$

Classification

We can classify a number of properties expressible in LTL:

safety	$\Box\,\varphi$
liveness	$\Diamond\,\varphi$
obligation	$\Box\,\varphi \vee \Diamond\,\psi$
recurrence	$\Box\,\Diamond\,\varphi$
persistence	$\Diamond\,\Box\,\varphi$
reactivity	$\Box\,\Diamond\,\varphi \vee \Diamond\,\Box\,\psi$

Classification

We can classify a number of properties expressible in LTL:

safety	$\Box\,\varphi$
liveness	$\Diamond\,\varphi$
obligation	$\Box\,\varphi \vee \Diamond\,\psi$
recurrence	$\Box\Diamond\,\varphi$
persistence	$\Diamond\Box\,\varphi$
reactivity	$\Box\Diamond\,\varphi \vee \Diamond\Box\,\psi$

Classification

We can classify a number of properties expressible in LTL:

safety	$\Box \varphi$
liveness	$\Diamond \varphi$
obligation	$\Box \varphi \vee \Diamond \psi$
recurrence	$\Box \Diamond \varphi$
persistence	$\Diamond \Box \varphi$
reactivity	$\Box \Diamond \varphi \vee \Diamond \Box \psi$

Plan

Propositional Modal Logic

- The logic of possibility and necessity
  - $\Box\varphi$: φ is "necessarily true", or "φ holds in all possible worlds"
  - $\Diamond\varphi$: φ is "possibly true", or "there is a possible world that realizes φ"
- The modalities are dual
  - $\Diamond\varphi \stackrel{\text{def}}{=} \neg\Box\neg\varphi$

Definition

A Kripke frame $\mathcal{M}$ is a structure (W, R, ν) 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)
- $\nu : \mathbb{P} \longrightarrow 2^K$ determines the truth assignment to the atomic propositional variables in each state

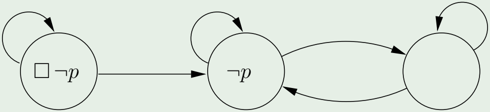
Definition

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

$$
\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in \nu(p) \\
\\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models \varphi_1 \vee \varphi_2 & \text{iff} & \mathcal{M}, w \models \varphi_1 \text{ or } \mathcal{M}, w \models \varphi_2 \\
\\
\mathcal{M}, w \models \Box\,\varphi & \text{iff} & \mathcal{M}, w' \models \varphi \text{ for all } w' \text{ such that } (w, w') \in R \\
\mathcal{M}, w \models \Diamond\,\varphi & \text{iff} & \mathcal{M}, w' \models \varphi \text{ for some } w' \text{ such that } (w, w') \in R
\end{array}
$$

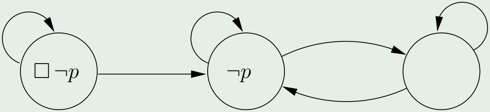
Example (Logic T)

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



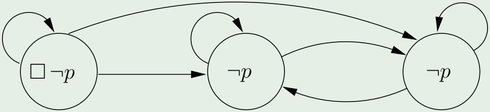
Example (Logic T)

- 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, ν) 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 $V : K \to 2^{\mathbb{P}}$ is given instead of ν
- 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, ν) 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 $V : K \to 2^{\mathbb{P}}$ is given instead of ν
- 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, ν) 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 $V : K \to 2^{\mathbb{P}}$ is given instead of ν
- 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

Multimodal Logic

- 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 φ is a formula and $a \in A$, then $[a]\varphi$ is a formula
- We also define $\langle a \rangle \varphi \stackrel{\text{def}}{=} \neg [a] \neg \varphi$
- The semantics of $\langle a \rangle$ and $[a]$ are defined as for $\Diamond\, a$ and $\Box\, a$, but "labelling" the transition with a

Multimodal Logic

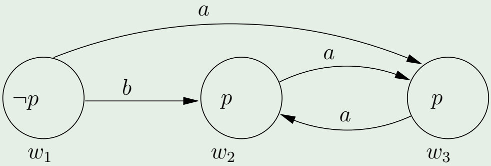
Definition

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
- $\nu : \mathbb{P} \longrightarrow 2^K$ determines the truth assignment to the atomic propositional variables in each state

Multimodal Logic
Examples

Example

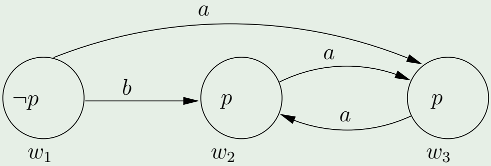


- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$?
- What about $M, w_2 \models [b] \neg p$?

Multimodal Logic

Examples

Example

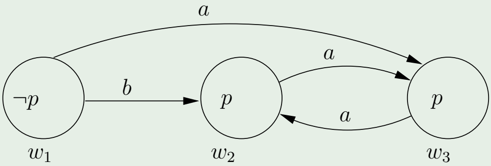


- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$?
- What about $M, w_2 \models [b] \neg p$?

Multimodal Logic

Examples

Example

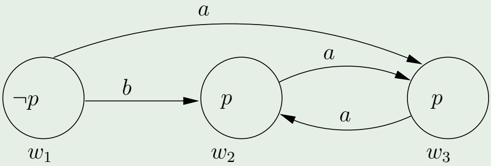


- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$?
- What about $M, w_2 \models [b] \neg p$?

Multimodal Logic

Examples

Example

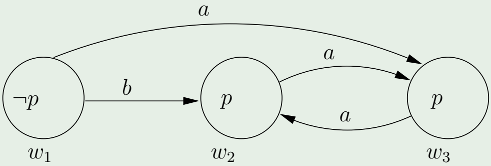


- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a \rangle p$
- $M, w_1 \models \langle b \rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b \rangle \neg p$?
- What about $M, w_2 \models [b] \neg p$?

Multimodal Logic

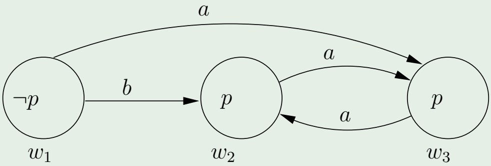
Examples

Example



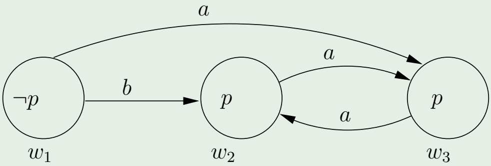
- $M, w_1 \models [a]p$
- $M, w_1 \models \langle a\rangle p$
- $M, w_1 \models \langle b\rangle p$, and also $M, w_1 \models [b]p$
- What about $M, w_2 \models \langle b\rangle \neg p$? NO
- What about $M, w_2 \models [b]\neg p$?

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



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

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 α, define a relation R over K s.t. $(s, t) \in R$ iff t is a possible final state of the program α 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 α and halt in a state satisfying φ
  - $[\alpha]\varphi$: whenever α halts, it does so in a state satisfying φ
- 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 α, define a relation R over K s.t. $(s,t) \in R$ iff t is a possible final state of the program α 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 α and halt in a state satisfying φ
  - $[\alpha]\varphi$: whenever α halts, it does so in a state satisfying φ
- 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 α, define a relation R over K s.t. $(s, t) \in R$ iff t is a possible final state of the program α 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 α and halt in a state satisfying φ
  - $[\alpha]\varphi$: whenever α halts, it does so in a state satisfying φ
- 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 α, define a relation R over K s.t. $(s, t) \in R$ iff t is a possible final state of the program α 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 α and halt in a state satisfying φ
    - $[\alpha]\varphi$: whenever α halts, it does so in a state satisfying φ
- 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

- 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: $\alpha, \beta, \gamma, \ldots$

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

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

- We define $\langle\alpha\rangle\varphi \stackrel{\text{def}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive Meaning

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

- We define $\langle\alpha\rangle\varphi \stackrel{\text{def}}{=} \neg[\alpha]\neg\varphi$

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

- We define $\langle\alpha\rangle\varphi \stackrel{\mathrm{def}}{=} \neg[\alpha]\neg\varphi$

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

- We define $\langle\alpha\rangle\varphi \stackrel{\mathrm{def}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive Meaning

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

- We define $\langle\alpha\rangle\varphi \stackrel{\text{def}}{=} \neg[\alpha]\neg\varphi$

Propositional Dynamic Logic

Intuitive Meaning

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

- We define $\langle\alpha\rangle\varphi \stackrel{\text{def}}{=} \neg[\alpha]\neg\varphi$

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

Propositional Dynamic Logic

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

Plan

μ-calculus

- μ-calculus is a powerful language to express properties of transition systems by using least and greatest fixpoint operators
  - ν is the greatest fixpoint meaning **looping**
  - μ is the least fixpoint meaning **finite looping**
- Many temporal and program logics can be encoded into the μ-calculus
- Efficient model checking algorithms
- Formulas are interpreted relative to a transition system
  - The Kripke structure needs to be slightly modified

μ-calculus: Syntax

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

Definition

The set of μ-calculus formulae (w.r.t. $(Var, Prop, L)$) is defined as follows:

- P is a formula
- Z is a formula
- If ϕ_1 and ϕ_2 are formulae, so is $\phi_1 \wedge \phi_2$
- If ϕ is a formula, so is $[a]\phi$
- If ϕ is a formula, so is $\neg\phi$
- If ϕ is a formula, then $\nu Z.\phi$ is a formula
  - Provided every *free* occurrence of Z in ϕ occurs positively (within the scope of an even number of negations)
  - ν is the only binding operator

μ-calculus: Syntax

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

μ-calculus: Syntax

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

μ-calculus: Syntax

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

μ-calculus: Semantics

Definition

A labelled transition system (LTS) is a triple $M = (\mathcal{S}, T, L)$, where:

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

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

Definition

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

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

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

$$
\begin{array}{rcl}
\|\phi_1 \vee \phi_2\|^{\mathcal{T}}_{\mathcal{V}} & = & \|\phi_1\|^{\mathcal{T}}_{\mathcal{V}} \cup \|\phi_2\|^{\mathcal{T}}_{\mathcal{V}} \\
\|\langle a \rangle \phi\|^{\mathcal{T}}_{\mathcal{V}} & = & \{s \mid \exists t.(s, a, t) \in T \wedge t \in \|\phi\|^{\mathcal{T}}_{\mathcal{V}} \\
\|\mu Z.\phi\|^{\mathcal{T}}_{\mathcal{V}} & = & \bigcap \{S \subseteq \mathcal{S} \mid S \supseteq \|\phi\|^{\mathcal{T}}_{\mathcal{V}[Z:=S]}\}
\end{array}
$$

μ-calculus

Example

- μ is liveness
  - "On all length a-path, P eventually holds"

$$\mu Z.(P \vee [a]Z)$$

  - "On some a-path, P holds until Q holds"

$$\mu Z.(Q \vee (P \wedge \langle a \rangle Z)$$

- ν is safety
  - "P is true along every a-path"

$$\nu Z.(P \wedge [a]Z)$$

  - "On every a-path P holds while Q fails"

$$\nu Z.(Q \vee (P \wedge [a]Z))$$

Plan

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

Example of a R-T logic with bounded temporal operators

$$\varphi \ := \ p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi\,\mathcal{U}_I\,\varphi$$

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. φ_2 holds at time t' and φ_1 holds through the interval (t, t')
- Derived operators
  - $\Diamond_I \varphi \stackrel{\text{def}}{=} \mathit{true}\,\mathcal{U}_I\,\varphi$: time-bounded eventually
  - $\Box_I \varphi \stackrel{\text{def}}{=} \neg\Diamond_I\neg\varphi$: time-bounded always

Example of a R-T logic with bounded temporal operators

$$\varphi \ := \ p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi\, \mathcal{U}_I\, \varphi$$

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. φ_2 holds at time t' and φ_1 holds through the interval (t, t')
- Derived operators
  - $\Diamond_I \varphi \stackrel{\text{def}}{=} \mathit{true}\, \mathcal{U}_I\, \varphi$: time-bounded eventually
  - $\Box_I \varphi \stackrel{\text{def}}{=} \neg\Diamond_I \neg\varphi$: time-bounded always

Example of a R-T logic with bounded temporal operators

$$\varphi \ := \ p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi\,\mathcal{U}_I\,\varphi$$

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. φ_2 holds at time t' and φ_1 holds through the interval (t, t')
- Derived operators
  - $\Diamond_I\varphi \stackrel{\text{def}}{=} \mathit{true}\,\mathcal{U}_I\,\varphi$: time-bounded eventually
  - $\Box_I\varphi \stackrel{\text{def}}{=} \neg\Diamond_I\neg\varphi$: time-bounded always

Example of a R-T logic with bounded temporal operators

$$\varphi \ := \ p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathcal{U}_I \, \varphi$$

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. φ_2 holds at time t' and φ_1 holds through the interval (t, t')
- Derived operators
  - $\Diamond_I \varphi \stackrel{\text{def}}{=} \mathit{true} \, \mathcal{U}_I \, \varphi$: time-bounded eventually
  - $\Box_I \varphi \stackrel{\text{def}}{=} \neg \Diamond_I \neg \varphi$: 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"

- 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**

- 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**

- 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**

- 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

$$\varphi \ := \ p \mid \pi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathcal{U} \, \varphi \mid x.\varphi$$

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

Example

- "Every stimulus p is followed by a response q within 3 time units"

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

Example

- "Every stimulus p is followed by a response q within 3 time units"

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

- "Every stimulus p is followed by a response q, followed by another response r, such that r is within 3 time units of p"

$$\Box x.(p \Rightarrow \Diamond(q \wedge \Diamond z.(r \wedge z \leq x + 3)))$$

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

- 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

$$\varphi \ := \ p \mid \pi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi\,\mathcal{U}\,\varphi \mid \exists x.\varphi$$

- $x \in V$, 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 $\exists x.(T = x \wedge \varphi)$

- 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

$$\varphi \;:=\; p \mid \pi \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi\,\mathcal{U}\,\varphi \mid \exists x.\varphi$$

- $x \in V$, 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 $\exists x.(T = x \wedge \varphi)$

Example

- "Every stimulus p is followed by a response q within 3 time units"

$$\forall x.\Box((p \wedge T = x) \Rightarrow \Diamond(q \wedge T \leq x + 3))$$

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

Branching-time:

- RTCTL (real-time computation tree logic)
  - A propositional branching-time logic for synchronouys 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

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

- M. Fitting. **Basic Modal Logic.** Handbook of Logic in Artificial Intelligence and Logic Programming, vol. 1, 1993
- C. Stirling. **Modal and Temporal Logics**. Handbook of Logic in Computer Science, vol. 2, 1992

Dynamic Logic

- D. Harel, D. Kozen and J. Tiuryn. **Dynamic Logic.** MIT, 2003

μ-calculus:

- J. Bradfield and C. Stirling. **Modal logics and μ-calculi: an introduction**

Real-time logics:

- R. Alur and T. Henzinger. **Logics and Models of Real time: A Survey**. LNCS 600, pp. 74-106, 1992