Proof Theory of Temporal Logics

Bahareh Afshari

University of Gothenburg, Sweden

ANU Logic Summer School, December 2023

References

- *Logic in Computer Science: Modelling and reasoning about systems.* M. Huth and M. Ryan. CUP 2004.

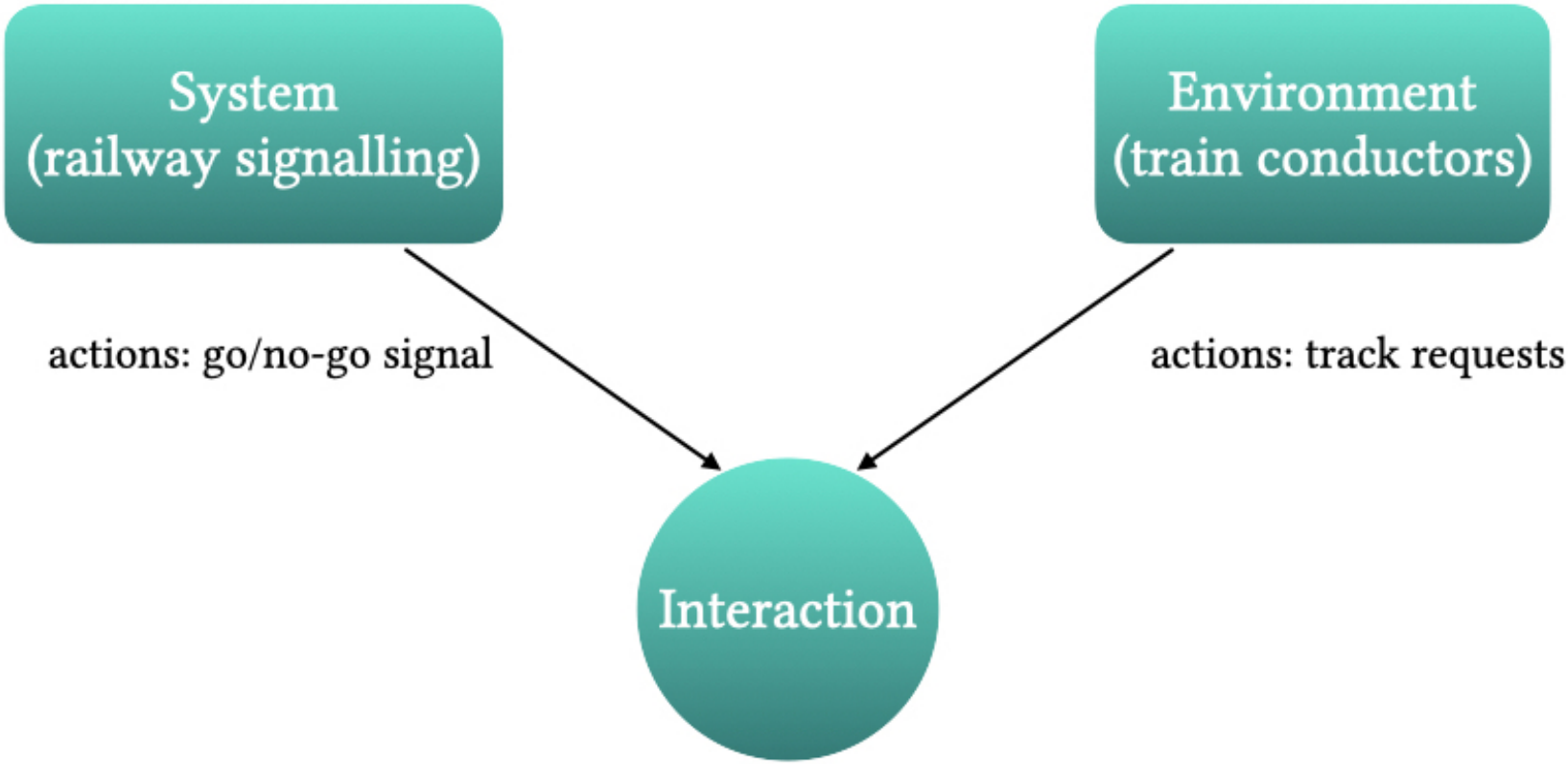
- *Temporal Logics in Computer Science: Finite-state systems.* S. Demri, V. Goranko, and M. Lange. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016.

Outline

Section 1

Motivating example

Railway signalling



Questions Is the system operating correctly? Are there any faults?

Reactive systems

Main characteristics:

- interact with the environment
- safety-critical
- run indefinitely

Other examples: *air traffic control systems* or *automated stock trading*

Want to verify

- **Liveness Good things eventually happen**
- **Safety Bad things never happen**

An example in railway signalling

- Let p_i denote "Train #i is granted permission to enter the signal block."
- Requirements:
  1. Eventually p_1 (liveness)
  2. Eventually p_2 (liveness)
  3. Never(p_1 and p_2) (safety)

Liveness "Something good will eventually happen."
Safety "Something bad will never happen."

Question How can we express these properties in a logical language?

- $\mathbf{1} \equiv p_1$ or **Next**(**1**)
- **3** $\equiv$ not(p_1 and p_2) and **Next**(**3**)
- x $\equiv \varphi$(x) where φ is a statement using finite vocabulary such as **Next** and logical connectives.

Questions of intrest

1. Are there logics suitable to express such temporal properties?
2. What is the cost of checking liveness, safety and other fairness constraints?
3. Can practical algorithms be developed for real-world examples?

Section 2

Modal Logic

Basic modal logic

Propositional logic:

- propositional variables p, q, ...
  connectives $\neg$, $\wedge$, $\vee$, $\rightarrow$
- Syntax $\quad p \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi$
- Semantics: formulas are either *true* or *false*

One can extend propositional logic by

- adding two unary predicates:

  $\Box$: box; necessarily

  $\Diamond$: diamond; possibly

- If φ is a formula so are $\Box\varphi$ and $\Diamond\varphi$.
- Examples: $\Box(p \vee \Diamond q)$, $\Box\Diamond\neg p$

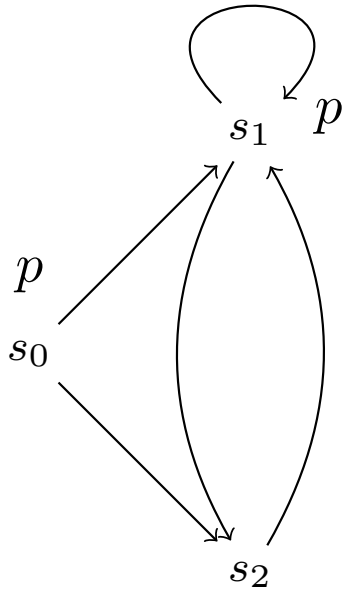
Models of modal logic

A model of modal logic, called a **transition system** (or Kripke structure), is a triple $T = (S, \to, \lambda)$ with a

- set of states S;
- transition relation $\to \subseteq S \times S$, writing also $s \to t$ for $(s, t) \in \to$
- function $\lambda : \textsc{Prop} \to \mathcal{P}(S)$ interpreting atomic propositions

Note that a transition system is simply a **directed graph** whose nodes are labelled by **finite sets of propositions**.

Example



Transition system with states $\{s_0, s_1, s_2\}$

Semantics of modal logic

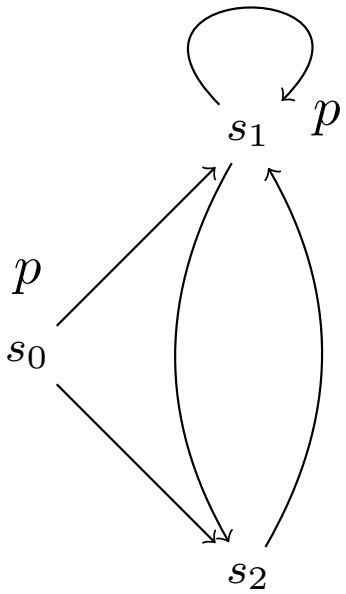
We define what it means for a transition system T to model/satisfy a formula φ at a world s, written $T, s \models \varphi$, by structural induction:

- $T, s \models p$ iff $p \in \lambda(s)$
- $T, s \models \neg\varphi$ iff $T, s \not\models \varphi$
- $T, s \models \varphi \wedge \psi$ iff $T, s \models \varphi$ and $T, s \models \psi$
- $T, s \models \varphi \vee \psi$ iff $T, s \models \varphi$ or $T, s \models \psi$
- $T, s \models \Box\varphi$ iff for every $s \to t$ we have $T, t \models \varphi$
- $T, s \models \Diamond\varphi$ iff for some $s \to t$ we have $T, t \models \varphi$

We write $\|\varphi\|^T$ to denote the set of all states s of transition system T that satisfy the formula φ. In other words,

$$s \in \|\varphi\|^T \text{ if and only if } T, s \models \varphi.$$

Examples



Transition system T with states $\{s_0, s_1, s_2\}$

- $\|p\|^T = \{s_0, s_1, s_2\}$
- $s_0 \models^T \Box p$
- $\|q\|^T = \{s_2\}$
- $\|\Diamond q\|^T = \{s_0, s_1\}$
- $\|\Diamond\Diamond q\|^T = \{s_0, s_1, s_2\}$
- $\|\Box\Diamond\neg q\|^T = ?$

Satisfiability via tableaux

We will look at a **syntactic** way of constructing models for modal logics.

Semantic methods:

- selection
- canonical model construction
- filteration

Syntactic methods:

- tableaux
- automata
- games

Remark Tableaux are often designed for satisfiability but one can also talk about tableaux for validity. As the two notions are dual if you have a tableau system for one you can define an analogous one for the other.

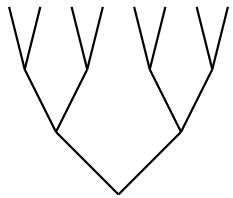
Yet another model construction technique

Questions

1. Can we get finite models? Often, but not always.
2. Is it a robust technique like canonical model construction? No!
3. Can one use tableaux for decidability? Yes, sometimes.
4. When should we use tableaux? Seeking computational properties.

How does a tableau look like?

Our tableaux are going to look like (upwards growing) trees:



Given a formula φ we want to check if it satisfiable. The idea is to systematically build a model from **using the logical structure** of φ:

- Start the tree by putting φ at the **root**;
- From the root make new **branches** (unary, binary or more);
- The idea of these new nodes is to reduce checking satisfiability of φ to checking **satisfiability of its constituents**;
- To check satisfiability of $\varphi_1 \wedge \varphi_2$ we check both φ_1 and φ_2;
- Continue breaking down φ until only **literals** remain.

Preliminaries: Negation Normal Form

To make our tableaux construction as simple as possible we are going to work with modal formulas in negation normal form.

Definition

A formula of modal logic is said to be in Negation Normal Form (NNF) iff it can be constructed by $\Box$, $\Diamond$, $\wedge$, $\vee$ from propositions and negated propositions (**literals**).

Examples:

- $\neg\Diamond(\neg p \wedge q)$ is not in NNF but $\Box(p \vee \neg q)$ is.
- What about $\Box\neg(p \wedge q)$?

NNF is not a restriction on expressibility:

Lemma (Exercise)

Every formula of modal logic is equivalent to one in NNF. The equivalence is provable in **K**.

Motivating example

Consider the formula $\varphi = (\Box\Box p \wedge \Diamond q) \vee (\neg p \wedge \Box\neg q)$. We aim to find a model of φ, namely some $M = (W, \to, V)$ and $w_0 \in W$ such that $M, w_0 \models \varphi$.

$$w_0 \models (\Box\Box p \wedge \Diamond q) \vee (\neg p \wedge \Box\neg q)$$
$$w_0 \models (\Box\Box p \wedge \Diamond q)$$
$$w_0 \models \{\Box\Box p, \Diamond q\}$$
$$\exists w_1 : w_0 \to w_1 \qquad w_1 \models \{\Box p, q\}$$

This "model search" naturally gives rise to the structure M given by:

$$\begin{cases} W = \{w_0, w_1\} \\ w_0 \to w_1 \\ V(p) = \varnothing, V(q) = \{w_1\} \end{cases}$$

This search for satisfiability (or soundness) can be formalised using tableaux, a **tree** where each node is labelled by a **subset of** $Sub(\varphi)$ (the set of all subformulas of φ).

Trees

A **tree** (over Σ) is a triple $t = \langle S, \to, \lambda \rangle$ with a distinguished node $\rho \in S$ which satisfies the following conditions.

- $(S, \to)$ is a connected directed graph.
- There are no transitions into ρ.
- For every $s \in S \setminus \{\rho\}$ there is exactly one $s_0 \in S$ such that $s_0 \to s$.
- $\lambda \colon S \to \mathcal{P}(\Sigma)$ is called the **labelling function** of t.

The node ρ is referred to as the **root** of the tree and any node without outgoing edges is a **leaf**.

Remark. A tree is a special case of a Kripke structure.

A **path** through tree t is a function $\mathbb{P} \colon \mathbb{N} \to S$ such that

- $\mathbb{P}(0) = \rho$,
- For every n, if $\mathbb{P}(n)$ is not a leaf, then $\mathbb{P}(n) \to \mathbb{P}(n+1)$.

Notation

- From now on, all formulas considered are in **NNF**.
- We use $\varphi, \psi, \delta, \gamma, \ldots$ (also with indices) for denoting formulas.
- Finite sets of formulas are denoted by $\Gamma, \Delta, \Theta, \Lambda, \ldots$

- PROP a (possibly infinite) set of propositions.
- A **literal** is an element of **LIT** $=$ PROP $\cup \{\neg p \mid p \in \text{PROP}\}$.
- A set $X \subseteq$ LIT is **inconsistent** if $\{p, \neg p\} \subseteq X$ for some $p \in$ PROP.

- $\Box\Gamma = \{\, \Box\varphi \mid \varphi \in \Gamma \,\}$ and $\Diamond\Gamma = \{\, \Diamond\varphi \mid \varphi \in \Gamma \,\}$.
- Γ, φ means $\Gamma \cup \{\varphi\}$, and Γ, Δ denotes $\Gamma \cup \Delta$.

Pre-tableaux

A **pre-tableau** for φ is a tree $t = \langle S, \to, \lambda \rangle$ over $Sub(\varphi)$ such that

1. $\lambda(\rho) = \{\varphi\}$,
2. every leaf $l \in S$ is labelled by a sequent of the form $\Box\Delta, \Diamond\Lambda, \Theta$ where $\Theta \subseteq \textsc{Lit}$ and either
   - $\Lambda = \varnothing$, or
   - Θ is inconsistent.
3. every non-leaf of t is related to its immediate successors by one of the rules below.

$$\wedge \ \frac{\Delta, \varphi_0, \varphi_1}{\Delta, \varphi_0 \wedge \varphi_1} \qquad \vee_0 \ \frac{\Delta, \varphi_0}{\Delta, \varphi_0 \vee \varphi_1} \qquad \vee_1 \ \frac{\Delta, \varphi_1}{\Delta, \varphi_0 \vee \varphi_1}$$

$$\text{mod} \ \frac{\Delta, \delta_1 \qquad \Delta, \delta_2 \qquad \cdots \qquad \Delta, \delta_n}{\Box\Delta, \Diamond\{\delta_1, \cdots, \delta_n\}, \Theta} \ \Theta \subseteq \textsc{Lit} \text{ is consistent}$$

Rules for **K** pre-tableaux

Remarks

- A pre-tableau is a finitely branching tree.
- Branching occurs only mod-rules.
- The rules are read bottom-up (from root towards leaves).
- For each rule the distinguished formulas in the lower and upper sequents are called respectively the **principal** and **residual** formulas of the rule.
- In the mod-rule all formulas are considered distinguished, including those in Θ.
- The mod-rule can be applied only if no other rule is applicable.
- For modal logic **K** tableaux are well-founded trees. In other words, every path is finite. **Why?**

Examples of pre-tableaux

The formula $(\Box\Box p \wedge \Diamond q) \vee (\neg p \wedge \Box\neg q)$ has two pre-tableaux:

$$\vee_0 \dfrac{\wedge \dfrac{\mathrm{mod} \dfrac{\Box p, q}{\Box\Box p, \Diamond q}}{\Box\Box p \wedge \Diamond q}}{(\Box\Box p \wedge \Diamond q) \vee (\neg p \wedge \Box\neg q)} \qquad \vee_1 \dfrac{\wedge \dfrac{\neg p, \Box\neg q}{\neg p \wedge \Box\neg q}}{(\Box\Box p \wedge \Diamond q) \vee (\neg p \wedge \Box\neg q)}$$

Question How many pre-tableaux does $(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)$ have?

Examples of pre-tableaux

$$\dfrac{\dfrac{\dfrac{\dfrac{\neg p, p}{\neg p, p \vee \Box\Box\neg q}\vee_0}{\neg p \vee \Diamond\Diamond q, p \vee \Box\Box\neg q}\vee_0}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}\wedge$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{q, \neg q}{\Diamond q, \Box\neg q}\text{mod}}{\Diamond\Diamond q, \Box\Box\neg q}\text{mod}}{\Diamond\Diamond q, p \vee \Box\Box\neg q}\vee_1}{\neg p \vee \Diamond\Diamond q, p \vee \Box\Box\neg q}\vee_1}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}\wedge$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\neg p, \Box\Box\neg q}{\neg p, p \vee \Box\Box\neg q}\vee_1}{\neg p \vee \Diamond\Diamond q, p \vee \Box\Box\neg q}\vee_0}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}\wedge$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{q}{\Diamond q}\text{mod}}{\Diamond\Diamond q, p}\text{mod}}{\Diamond\Diamond q, p \vee \Box\Box\neg q}\vee_0}{\neg p \vee \Diamond\Diamond q, p \vee \Box\Box\neg q}\vee_1}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}\wedge$$

Tableaux definition

A **tableau** is a pre-tableau where the sequent (finite set of formulas) at each leaf has the form $\Box\Gamma, \Theta$ where $\Theta \subseteq \textsc{Lit}$ is consistent.

Question Which of these are tableaux?

$$\wedge\,\dfrac{\vee_0\,\dfrac{\vee_0\,\dfrac{\neg p, p}{\neg p, p \vee \Box\Box\neg q}}{\neg p \vee \Diamond\Diamond q, p \vee \Box\Box\neg q}}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}$$

$$\wedge\,\dfrac{\vee_1\,\dfrac{\mathrm{mod}\,\dfrac{\mathrm{mod}\,\dfrac{q, \neg q}{\Diamond q, \Box\neg q}}{\Diamond\Diamond q, \Box\Box\neg q}}{\Diamond\Diamond q, p \vee \Box\Box\neg q}}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}$$

$$\wedge\,\dfrac{\vee_0\,\dfrac{\vee_1\,\dfrac{\neg p, \Box\Box\neg q}{\neg p, p \vee \Box\Box\neg q}}{\neg p \vee \Diamond\Diamond q, p \vee \Box\Box\neg q}}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}$$

$$\wedge\,\dfrac{\vee_1\,\dfrac{\vee_0\,\dfrac{\mathrm{mod}\,\dfrac{\mathrm{mod}\,\dfrac{q}{\Diamond q}}{\Diamond\Diamond q, p}}{\Diamond\Diamond q, p \vee \Box\Box\neg q}}{\neg p \vee \Diamond\Diamond q, p \vee \Box\Box\neg q}}{(\neg p \vee \Diamond\Diamond q) \wedge (p \vee \Box\Box\neg q)}$$

Properties

Observation

Every formula of modal logic has at least one (and finitely many) pre-tableaux.

Question How many pre-tableaux does a formula have?

Lemma (Exercise)

Let $t = \langle S, \to, \lambda \rangle$ be a tableau and suppose $s \in S$. Then for all $p \in \textsc{Prop}$ we have $\{p, \neg p\} \not\subseteq \lambda(s)$.

Soundness

Lemma

If φ has a tableau then φ is satisfiable.

Let $t = \langle S, \to, \lambda \rangle$ be a tableau for φ.

Define a structure $M = (W, R, V)$ and a map $\tau \colon S \to W$ such that

1. $\tau(\rho) = w_0 \in W$.
2. If $s_0 \to s_1$ and the tableau rule applied at s is mod-rule then $(\tau(s_0), \tau(s_1)) \in R$, otherwise $\tau(s_0) = \tau(s_1)$.
3. $w \in V(p)$ iff there exists $s \in S$ such that $\tau(s) = w$ and $p \in \lambda(s)$.

Exercise Complete the proof by first proving

- Show that for every $s \in S$ and $\psi \in \lambda(s)$ we have

$$\psi \in \lambda(s) \qquad \text{iff} \qquad M, \tau(s) \models \psi$$

Hint: By double induction on i) complexity of ψ and ii) distance of s from the leaves.

How to extract the tree-model?

Both of the following are tableaux.

$$\vee_0 \dfrac{\mathrm{mod}\dfrac{\wedge\dfrac{\Box p, q}{\Box\Box p, \Diamond q}}{\Box\Box p \wedge \Diamond q}}{(\Box\Box p \wedge \Diamond q) \vee (\neg p \wedge \Box\neg q)} \qquad \vee_0 \dfrac{\wedge\dfrac{\neg p, \Box\neg q}{\neg p \wedge \Box\neg q}}{(\Box\Box p \wedge \Diamond q) \vee (\neg p \wedge \Box\neg q)}$$

The models they give us are respectively:

$$\begin{cases} W = \{w_0, w_1, \} \\ R = \{(w_0, w_1))\} \\ V(p) = \varnothing, V(q) = \{w_1\} \end{cases} \qquad \begin{cases} W = \{w_0\} \\ R = \varnothing \\ V(p) = \varnothing \end{cases}$$

Completeness

Lemma

If φ is satisfiable then φ has a tableau.

Proof. Let $T = (S, \to, \lambda)$ be a transition system with distinguished node s_0 such that $T, s_0 \models \varphi$.

- Use the model T as a guiding tool to construct a tableau for φ.
- More specifically, build a pre-tableau such that every sequent is satisfied by some $s \in S$.
- Every path in the pre-tableau will correspond to a path in T.
- Every leaf in the pre-tableau will be of the form $\Box\Gamma, \Theta$ where Θ is consistent.

Exercise. Fill in the details of the proof.

Corollary

Modal logic has the finite model property.

Exercises

Consider the formulas φ and ψ given by

$$\varphi := \Box p \to \Box\Box p$$
$$\psi := \Diamond(p \to q) \to (\Box p \to \Diamond q)$$

1. Give an equivalent NNF formula for each
2. Using tableaux check if they are valid/satisfiable

Notes

- If you are claiming it is satisfiable then you should provide a tableau and describe the model it gives rise to.
- If you are claiming it is not satisfiable then you must argue that it has no tableaux; you can do this by listing all pre-tableaux and showing none satisfy the tableaux condition.
- Validity of φ can be deduced from satisfiability of $\neg\varphi$.

Expressive limitation of modal logic

Suppose we are interested to express the following property

there is a finite path that reaches a node labelled by p

We can try (and fail) using the modal syntax: $p \vee \Diamond p \vee \Diamond\Diamond p \vee \cdots$

Modal language does not allow us to express properties along arbitrary (finite or infinite) paths in a structure. Can we do better? Yes!

Temporal Logics

Temporal logics provide us with a language that can express path quantification:

- "along every path ..."
- "there exists a path along which ..."

and much more complex properties.

Section 3

Linear Time Temporal Logic

Linear-time Temporal Logic

LTL-formulas, over atomic propositions $p_1, \cdots, p_n$, are defined

$$\begin{array}{rcll} \varphi & ::= & p_i & \text{atomic proposition} \\ & | & \neg\varphi & \text{negation} \\ & | & \varphi \wedge \psi & \text{conjunction} \\ & | & \varphi \vee \psi & \text{disjunction} \\ & | & \mathbf{X}\,\varphi & \text{next} \\ & | & \varphi\,\mathbf{U}\,\psi & \text{until} \end{array}$$

Intended meaning

$\mathbf{X}\,\varphi$ "φ is true at the **next** time-step"

$\varphi\,\mathbf{U}\,\psi$ "φ is true **until** ψ is true (and ψ holds eventually)"

Two additional constructs

$\mathbf{F}\,\varphi$ "φ is **eventually** true"
φ is true at **some** point in the future

$\mathbf{G}\,\varphi$ "φ is **always** true"
φ is true at **every** point in the future (including the present)

They are expressible in LTL by

$$\begin{aligned} \mathbf{F}\,\varphi &:= \mathsf{true}\,\mathbf{U}\,\varphi \\ \mathbf{G}\,\varphi &:= \neg(\mathbf{F}\,\neg\varphi) \end{aligned}$$

Examples of properties expressible in LTL

- **Recurrence**: "p_1 holds infinitely often"

$$\mathbf{G}\,(\mathbf{F}\,p_1)$$

- **Periodicity**: "p_1 is true initially and precisely at every third moment"

$$p_1 \;\wedge\; \mathbf{X}\,\neg p_1 \;\wedge\; \mathbf{X}\,\mathbf{X}\,\neg p_1 \;\wedge\; \mathbf{G}\,(p_1 \leftrightarrow \mathbf{X}\,\mathbf{X}\,\mathbf{X}\,p_1)$$

- **Request-response**: "It is *always* the case that whenever p_1 holds, p_2 will hold sometime later"

$$\mathbf{G}\,(p_1 \rightarrow \mathbf{X}\,\mathbf{F}\,p_2)$$

- **Fairness**: "If p_1 is true infinitely often, then so is p_2"

$$\mathbf{G}\,\mathbf{F}\,p_1 \rightarrow \mathbf{G}\,\mathbf{F}\,p_2$$

Kripke structures revisited

A **(pointed) Kripke structure** over a set of atomic propositions $\{ p_1, \ldots, p_n \}$ is a quadruple (S, R, λ, s_0) with

- a finite state-set S
- an initial/distinguished state $s_0 \in S$
- a transition relation $R \subseteq S \times S$, and
- a labelling function $\lambda \colon S \longrightarrow \mathcal{P}(\{ p_1, \cdots, p_n \})$, associating to each $s \in S$ the set $\lambda(s)$ of those p_i that are satisfied at s.

Notation

We write $\lambda(s)$ as a bit vector $\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{B}^n$ such that $b_i = 1$ iff $p_i \in \lambda(s)$.

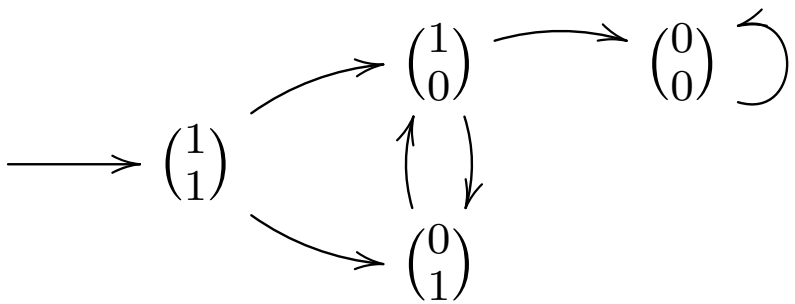
Paths and label sequences

An infinite **path** $s_0 s_1 s_2 \cdots$ through a Kripke structure (S, R, λ, s_0) induces an infinite sequencee of "labels" over the alphabet $\mathbb{B}^n$

$$\lambda(s_0)\,\lambda(s_1)\,\lambda(s_2)\cdots$$

Example

A Kripke structure over $\{p_1, p_2\}$.



$\binom{1}{1}\binom{1}{0}\binom{0}{1}\binom{1}{0}\binom{0}{0}\binom{0}{0}\cdots$

$\binom{1}{1}\binom{0}{1}\binom{1}{0}\binom{0}{1}\binom{1}{0}\binom{0}{0}\binom{0}{0}\cdots$

Semantics of LTL

LTL-formulas over atomic propositions $p_1, \cdots, p_n$ are interpreted as sets of ω-words α over the alphabet $\mathbb{B}^n$.

Notation. Let $\alpha = \alpha(0)\,\alpha(1)\,\cdots \in (\mathbb{B}^n)^\omega$:

1. α^i stands for $\alpha(i)\alpha(i+1)\cdots$, the ***i*-suffix** of α; so $\alpha = \alpha^0$.
2. $(\alpha(i))_j$ is the j-th component of the vector $\alpha(i)$.

Satisfaction relation. Let $i \geq 0$. Define $\alpha^i \vDash \varphi$ by recursion over φ:

$$
\begin{array}{lll}
\alpha^i \vDash p_j & \text{iff} & (\alpha(i))_j = 1 \\
\alpha^i \vDash \neg\varphi & \text{iff} & (\alpha^i \nvDash \varphi) \\
\alpha^i \vDash \varphi \vee \psi & \text{iff} & \alpha^i \vDash \varphi \;\text{ or }\; \alpha^i \vDash \psi \\
\alpha^i \vDash \varphi \wedge \psi & \text{iff} & \alpha^i \vDash \varphi \;\text{ and }\; \alpha^i \vDash \psi \\
\alpha^i \vDash \mathbf{X}\,\varphi & \text{iff} & \alpha^{i+1} \vDash \varphi \\
\alpha^i \vDash \varphi\,\mathbf{U}\,\psi & \text{iff} & \exists j \geq i : \Big(\alpha^j \vDash \psi \;\wedge\; \forall i \leq k < j : \alpha^k \vDash \varphi\Big)
\end{array}
$$

We say α **satisfies** φ, denoted $\alpha \vDash \varphi$, if $\alpha^0 \vDash \varphi$.

On meta-theory of LTL

1. *Linear vs Branching*
   - **Linear-time properties** set the same conditions on every infinite path through a system modelled by a Kripke structure.
   - **Branching-time properties** are conditions on the structure of the tree formed by all paths through a Kripke structure

See e.g. *Branching vs. linear time: Final showdown*, M.Y. Vardi, TACAS 2001:1–22.

2. *LTL vs FOL*

   Kamp showed in his influential doctoral thesis (1868) that LTL-definable languages are first-order definable:

See e.g. *A proof of Kamp's theorem.* A. Rabinovich: CSL'12:516–527.

Axiomatisation of LTL

- We work in **two-sided** sequent calculus so we can handle negation.
- Rules for the propositional part is as usual with left and right **intro/outro rules** for each connective.
- The main task is to deal with $\varphi \,\mathbf{U}\, \psi$.
- We treat this new connective by exploiting the following equivalence

$$\varphi \,\mathbf{U}\, \psi \equiv \psi \vee (\varphi \wedge \mathbf{X}\,(\varphi \,\mathbf{U}\, \psi))$$

$$X\ \frac{\Gamma \Rightarrow \Delta}{\Sigma, \mathbf{X}\,\Gamma \Rightarrow \mathbf{X}\,\Delta, \Lambda}\ \text{with } \Sigma, \Lambda \text{ arbitrary sequents}$$

$$U_R\ \frac{\Gamma \Rightarrow \Delta, \varphi, \psi \qquad \Gamma \Rightarrow \psi, \mathbf{X}\,(\varphi \,\mathbf{U}\, \psi)}{\Gamma \Rightarrow \Delta, \varphi \,\mathbf{U}\, \psi}$$

$$U_L\ \frac{\Gamma, \psi \Rightarrow \Delta, \qquad \Gamma, \varphi, \mathbf{X}\,(\varphi \,\mathbf{U}\, \psi) \Rightarrow \Delta}{\Gamma, \varphi \,\mathbf{U}\, \psi \Rightarrow \Delta}$$

Non-wellfounded proofs

Pre-proofs are finite branching but can have infinitely long branches

We need to know the following notions for a pre-proof

- path
- trace

Proofhood condition every infinite path carries an infinite trace on the left/right of $\Rightarrow$

Remark Notice:

- every infinite trace stabilises on the left or right of $\Rightarrow$
- every infinite path take the right premise of U_L/U_R infinitely often

LTL finitary axiomatisation

We need to add an **induction** axiom to handle the until operator:

$$\text{ind}\ \frac{\psi \vee (\varphi \wedge \mathbf{X}\,\delta) \to \delta}{\varphi\ \mathbf{U}\ \psi \to \delta}$$

Or equivalently,

$$\text{ind}\ \frac{\psi \to \delta \qquad \varphi \wedge \mathbf{X}\,\delta \to \delta}{\varphi\ \mathbf{U}\ \psi \to \delta}$$

Section 4

Propositional Dynamic Logic

PDL

Idea: extend modal logic with a modality for every program

Let $\pi, \pi', \ldots$ denote **programs**. We allow ourself an infinite collection of diamonds of the form $\langle\pi\rangle\varphi$. The intended interpretation is

- $\langle\pi\rangle\varphi$ meaning some terminating execution of the program π from the current state leads to a state satisfying φ.
- $[\pi]\varphi$ meaning every terminating execution of π from the current state leads to a state satisfying φ.

So far there is nothing interesting about viewing labels as programs. But by imposing a **structure** on our programs we can capture the essence of program execution/verification.

What structure?

Suppose we have our list $a, b, c, \ldots$ of **basic programs**, those that cannot be broken down to smaller ones.

The following four operations allow us to build **complex programs** from basic ones:

- **Sequence**: If π_1 and π_2 are programs then $\pi_1; \pi_2$ is a program. The program $\pi_1; \pi_2$ executes π_1 followed by π_2.
- **Choice**: If π_1 and π_2 are programs then $\pi_1 \cup \pi_2$ is a program. The program $\pi_1 \cup \pi_2$ will (non-deterministically) choose one of π_1 or π_2 and execute it.
- **Iteration**: If π is a program then π^* is program. The program π^* executes π a finite number of times.
- **Test**: If φ is a formula then $\varphi?$ is a program. Program $\varphi?$ tests if formula φ is true; if the answer is yes it continues otherwise it fails.

Examples

$$p \rightarrow [\pi]q$$

if p then after we have executed program π we are guaranteed to have q.

$$p \rightarrow [\pi_1 \cup \pi_2]q$$

if p then after executing either program π_1 or program π_2 we have q.

$$(p?; \pi_1) \cup (\neg p?; \pi_2)$$

if p then execute π_1 else execute π_2

$$(p?; \pi)^*; \neg p?$$

while p do π

Note, the first two are formulas and second two programs. Programs are used to define formulas. But formulas can also be used to build new programs which in turn can be applied to create more complex formulas.

Syntax of PDL

Formally, the programs and formulas, with respect to a set B of basic programs and a set **Prop** of propositions, are defined by mutual induction:

$$\pi := a \mid \pi;\pi \mid \pi^* \mid \pi \cup \pi \mid \varphi?$$
$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid [\pi]\varphi$$

where $a \in \mathsf{B}$ and $p \in \textsc{Prop}$.

Some remarks

PDL is a widely used in both theoretical and applied Computer Science (e.g. Description Logics, Game Logics and Linguistics).

As you already see, there are two kinds of syntax:

- **state formulas** evaluated on states e.g. $\langle \pi \rangle p$
- **program formulas** evaluated on pairs of states e.g. $[p?]q$

In a language that allows for infinite disjunctions the iteration operators can be written as

$$\langle \pi^* \rangle \varphi = \bigvee_{n \in \mathbb{N}} \langle \pi \rangle^n \varphi$$

Semantics of PDL

A **model** M for PDL is a triple $(W, \{R_\pi\}_{\pi \in \mathsf{B}}, V)$ where

- W is a set of worlds;
- R_π is an accessibility relation for each *basic* program π
- $V : \textsc{Prop} \to 2^W$ is a valuation of propositions

The satisfaction relation is analogous to that of normal modal logic:

$$
\begin{array}{lll}
M, w \models p & \text{iff} & w \in V(p) \\
M, w \models \neg\varphi & \text{iff} & M, w \not\models \varphi \\
M, w \models \varphi \vee \psi & \text{iff} & M, w \models \varphi \text{ or } M, w \models \psi \\
M, w \models [\pi]\varphi & \text{iff} & \text{for all } w' \text{ such that } (w, w') \in R_\pi \text{ then } M, w' \models \varphi
\end{array}
$$

Notice that for the definition above we are invoking the accessibility relation R_π where π can be a non-basic program.

Accessibility for complex programs

$$
\begin{aligned}
R_{\pi_1 \cup \pi_2} &:= R_{\pi_1} \cup R_{\pi_2} \\
R_{\pi_1;\pi_2} &:= R_{\pi_1} \circ R_{\pi_2} = \{(u,w) \mid \exists v.(u,v) \in R_{\pi_1} \wedge (v,w) \in R_{\pi_2}\} \\
R_{\pi^*} &:= (R_\pi)^* \quad \text{i.e. the reflexive transitive closure of } R_\pi \\
R_{\varphi?} &:= \{(w,w) \mid M, w \models \varphi\}
\end{aligned}
$$

These capture the intended program semantics that we discussed earlier and frames that satisfy these are called **regular frames**.

Failure of compactness

Consider the infinite set S consisting of formulas

$$\langle \pi^* \rangle p,\ \neg p,\ [\pi]\neg p,\ [\pi][\pi]\neg p,\ [\pi][\pi][\pi]\neg p,\ \ldots$$

It is easy to satisfy every finite subset of S but the entire set is unsatisfiable.

Finitry axiomatisation

Axioms:

1. propositional tautologies
2. $[\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$
3. $[\pi](\varphi \wedge \psi) \leftrightarrow ([\pi]\varphi \wedge [\pi]\psi)$
4. $[\pi_1 \cup \pi_2]\varphi \leftrightarrow ([\pi_1]\varphi \wedge [\pi_2]\varphi)$
5. $[\pi_1;\pi_2]\varphi \leftrightarrow ([\pi_1][\pi_2]\varphi)$
6. $[\varphi?]\psi \leftrightarrow (\varphi \to \psi)$
7. $\big(\varphi \wedge [\pi][\pi^*]\varphi\big) \leftrightarrow [\pi^*]\varphi$
8. $\big(\varphi \wedge [\pi^*](\varphi \to [\pi]\varphi)\big) \to [\pi^*]\varphi$

Rules:

MP from $\vdash \varphi$ and $\vdash \varphi \to \psi$ infer $\vdash \psi$

Nec from $\vdash \varphi$ we infer $\vdash [\pi]\varphi$

Test-free PDL

The axiom

$$[\varphi?]\psi \leftrightarrow (\varphi \rightarrow \psi)$$

seems to suggest that for every formula of PDL there may be a test-free formula equivalent to it.

But test-free PDL is less expressive than PDL. The following PDL formula does not have a test-free equivalent:

$$\langle (p?;\pi)^*;\neg p?\rangle\langle\pi\rangle p$$

Completness

Theorem

PDL is sound and complete for regular frames.

Soundness is immediate (Axiom 8 needs a little work).

Completeness via canonical model construction is problematic as the canonical model cannot be made regular. A strengthening of the filtration method is employed.

In fact filtration will give us more:

Theorem

PDL has the small model property: a satisfiable PDL formula φ is guaranteed to have a model with size no more than $2^{\mathsf{size}(\varphi)}$ where $\mathsf{size}(\varphi)$ is the number of symbols in φ.

Some words on filtration method

Filtration is model construction techniques that allows us to

(1) build finite models (if they exists)

(2) accommodate for logics that lack compactness

For the filtration method one generally uses the subformulas of a formula φ to define "filters". In the case of PDL, we need a more general notion of subformulas called **Fischer-Ladner Closure** of formulas.

Fischer-Ladner closure of PDL formulas

A set of PDL-formulas Σ is **closed under FL** if

1. $\varphi \vee \psi \in \Sigma$ implies $\varphi \in \Sigma$ and $\psi \in \Sigma$
2. $\neg\varphi \in \Sigma$ implies $\varphi \in \Sigma$
3. for basic program a basic: $[a]\varphi \in \Sigma$ implies $\varphi \in \Sigma$
4. $[\pi_1 \cup \pi_2]\varphi$ implies $[\pi_1]\varphi \in \Sigma$ and $[\pi_2]\varphi \in \Sigma$
5. $[\pi_1;\pi_2]\varphi \in \Sigma$ implies $[\pi_1][\pi_2]\varphi \in \Sigma$ and $[\pi_2]\varphi \in \Sigma$
6. $[\pi^*]\varphi \in \Sigma$ implies $[\pi;\pi^*]\varphi \in \Sigma$
7. $[\varphi?]\psi \in \Sigma$ implies $\varphi \in \Sigma$

Filtration model

We take

- the singleton set $\{\varphi\}$ and generate the FL-closed set Σ containing φ
- the Kripke structure $\mathfrak{M}$ (which satisfies φ but may be infinite) and quotient it by the relation

$$u \equiv v \quad \text{iff} \quad \textbf{for every } \varphi \in \Sigma \text{ we have } \mathfrak{M}, u \models \varphi \text{ iff } \mathfrak{M}, v \models \varphi$$

Define a filtration model $\mathfrak{M}' = (W', R', V')$ by setting

(i) $W' = \{[w] : w \in W\}$

(ii) for basic program a:

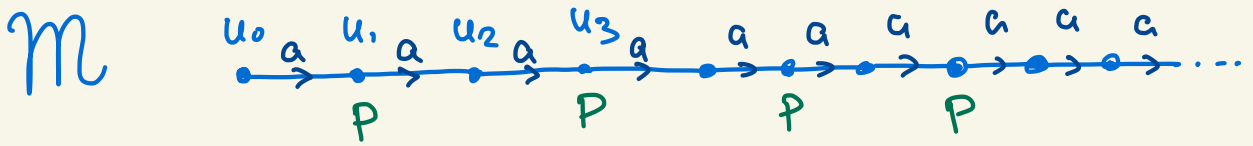
$$[u]R'_a[v] \qquad \text{iff} \qquad \exists u_0 \in [u] \text{ and } \exists v_0 \in [v] \text{ s.t. } u_0 R_a v_0$$

(iii) $V'(p) = \{[w] : \mathfrak{M}, w \models p\}$ for every $p \in \Sigma$.

To complete the proof we need to establish that for every $\psi \in \Sigma$

$$\mathfrak{M}', [w] \models \psi \qquad \text{iff} \qquad \mathfrak{M}, w \models \psi$$

Example 1



$\Sigma = \{p, \langle a \rangle p\}$

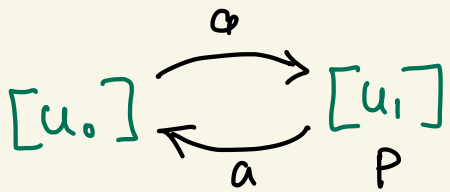
Two equivalent classes

$[u_0] = \{u_{2i} : \text{for every } i\}$

$[u_1] = \{u_{2i+1} : \text{for every } i\}$

$u_{2i} R_a u_{2i+1}$ iff $[u_{2i}] R' [u_{2i+1}]$

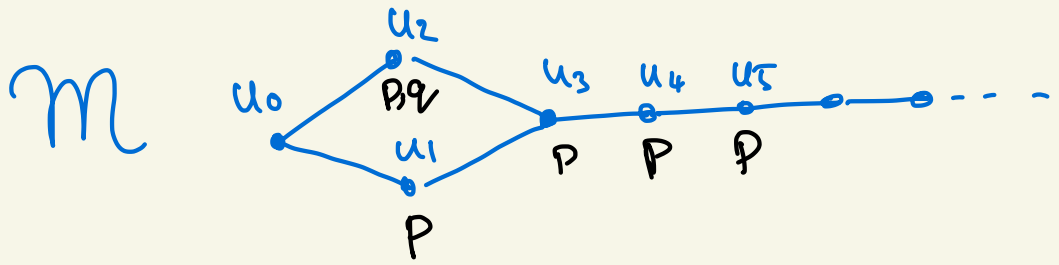
$u_{2i+1} R_a u_{2i+2}$ iff $[u_{2i+1}] R' [u_{2i+2}]$



EXC: a b a b

$a \rightsquigarrow a^*$

Example 2



$$\Sigma = \{p, \Diamond p\}$$

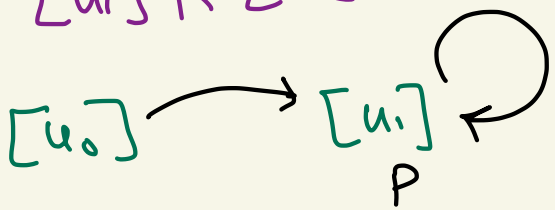
Two equiv. classes

$[u_0] = \{u_0\}$

$[u_1] = \{u_1, u_2, \dots\}$

$u_0 R u_1$ iff $[u_0] R' [u_1]$

$u_1 R u_3$ iff $[u_1] R' [u_1]$



Exercise

1. The Fischer-Ladner closure of a formula $\{\varphi\}$, denoted $FL(\varphi)$ is the smallest set containing the formula φ and closed under 1–7. Show $FL(\varphi)$ is a finite set.
2. Let $\mathfrak{M} = (W, \{R_\pi\}_{\pi \in \Pi})$ be a frame. Prove for $\pi_1, \pi_2, \pi \in \Pi$
   1. $\mathfrak{M} \models \langle \pi_1 ; \pi_2 \rangle \varphi \leftrightarrow \langle \pi_1 \rangle \langle \pi_2 \rangle \varphi$ if and only if $R_{\pi_1;\pi_2} = R_{\pi_1} \circ R_{\pi_2}$
   2. $\mathfrak{M} \models \langle \pi_1 \cup \pi_2 \rangle \varphi \leftrightarrow \langle \pi_1 \rangle \varphi \vee \langle \pi_2 \rangle \varphi$ if and only if $R_{\pi_1 \cup \pi_2} = R_{\pi_1} \cup R_{\pi_2}$
   3. If $R_{\pi^*} = (R_\pi)^*$ then $\mathfrak{M} \models \varphi \vee \langle \pi \rangle \langle \pi^* \rangle \varphi \leftrightarrow \langle \pi^* \rangle \varphi$
   4. If $R_{\pi^*} = (R_\pi)^*$ then $\mathfrak{M} \models [\pi^*](\varphi \to [\pi]\varphi) \to (\varphi \to [\pi^*]\varphi)$
   5. If $\mathfrak{M} \models \varphi \vee \langle \pi \rangle \langle \pi^* \rangle \varphi \to \langle \pi^* \rangle \varphi$ then $(R_\pi)^* \subseteq R_{\pi^*}$.
3. Work out what **Segerberg's axiom** $[\pi^*](\varphi \to [\pi]\varphi) \to (\varphi \to [\pi^*]\varphi)$ is stating.

Section 5

Cyclic and Ill-founded Proofs

Proof Systems à la Hilbert

A proof of a sequent Γ in a given system $\mathscr{S}$ is a **finite tree** in which leaves are labelled by axioms, the labelling of inner nodes respect the inference rules of $\mathscr{S}$, and the root is labelled by Γ.

But these 'concrete' proofs are hard to produce because of

- non-determinism
- induction 'axioms'

Cyclic Proofs: the core idea

Geared towards *proof search*, a cyclic proof formalises **an infinite decent argument**.

$$\sqrt{2} = p/q \quad \rightarrow \quad p^2 = 2q^2 \quad \rightarrow \quad p(p-q) = q(2q-p)$$

$$\sqrt{2} = p/q = (2q-p)/(p-q)$$

$$\sqrt{2} = p/q = p'/q' = p''/q'' = \cdots \qquad \text{with } q > q' > q'' > \cdots$$

Example

A circular proof of "every natural number is either even or odd".

$$\dfrac{\dfrac{\dfrac{E(0)}{E(0)\vee O(0)}}{x=0\to E(x)\vee O(x)} \qquad \dfrac{\dfrac{E(y)\vee O(y) \qquad \dfrac{\vdots}{O(y)\to E(y+1)} \qquad \dfrac{\vdots}{E(y)\to O(y+1)}}{E(y+1)\vee O(y+1)}}{x=y+1\to E(x)\vee O(x)}}{E(x)\vee O(x)}\ \text{case dist.}$$

$$\dfrac{E(x)\vee O(x)}{\forall x(E(x)\vee O(x))}$$

Predicates used:

$$E(0)$$
$$E(x) := \exists y(x = 2y)$$
$$O(x) := \exists y(x = 2y + 1)$$

Non-example

$$\dfrac{\dfrac{\dfrac{}{\bot}\,\text{weak.}}{\bot, \bot}\,\text{contr.}}{\bot}$$

A very general definition

Definition. A cyclic proof of Γ in $\mathscr{S}$ is a derivation tree which can have both axiomatic and **non-axiomatic** leaves as long as the latter satisfy the property:

1. if the non-axiomatic leaf is labelled by Δ then there is another node labelled by Δ which occurs earlier in the derivation tree
2. between the two occurrences of Δ **progress** is made

What is progress? How is it determined?

Cyclic proofs for modal logic K4

Formulas $p \mid \overline{p} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi$

Sequents $\Gamma, \Delta, \ldots$ finite sets of formulas treated as disjunction

Axioms $\Gamma, \varphi, \bar{\varphi}$

Inference rules

$$\frac{\Gamma, \bot}{\Gamma}\,\bot \qquad \frac{\Gamma, \varphi \qquad \Gamma, \psi}{\Gamma, \varphi \wedge \psi}\,\wedge \qquad \frac{\Gamma, \varphi, \psi}{\Gamma, \varphi \vee \psi}\,\vee \qquad \frac{\Gamma, \Diamond\Gamma, \varphi}{\Diamond\Gamma, \Box\varphi, \Delta}\,4$$

where $\Diamond\Gamma$ abbreviates $\{\Diamond\varphi_1, \Diamond\varphi_2, \ldots, \Diamond\varphi_n\}$ for $\Gamma = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$.

- **axiom 4:** $\Box\varphi \to \Box\Box\varphi$
- **Löb's axiom:** $\Box(\Box p \to p) \to \Box p$.

A cyclic proof of Löb's axiom

$$\dfrac{\dfrac{\dfrac{\dfrac{\Box p \wedge \bar{p}, \Diamond(\Box p \wedge \bar{p}), p}{\Box p, \Diamond(\Box p \wedge \bar{p}), p}\,4 \qquad \bar{p}, \Diamond(\Box p \wedge \bar{p}), p}{\Box p \wedge \bar{p}, \Diamond(\Box p \wedge \bar{p}), p}\,\wedge}{\Diamond(\Box p \wedge \bar{p}), \Box p}\,4}{\Diamond(\Box p \wedge \bar{p}) \vee \Box p}\,\vee$$

Theorem (Shamkanov 2014)

K4 + circularity = GL

where GL is Gödel-Löb logic axiomatisable by

Boolean tautologies + $\Box$-distribution + Löb axiom + MP

An example involving fixed points

- p is reachable $\quad R =_\mu p \vee \Diamond R$
- p is reachable in an even number of steps $\quad E =_\mu p \vee \Diamond\Diamond E$
- p is reachable in an odd number of steps $\quad O =_\mu \Diamond(p \vee \Diamond O)$

The following is valid:

$$R \to E \vee O$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{R \to E, O}{\Diamond R \to p, \Diamond E, \Diamond O}\,\Diamond \qquad p \to \cdots p \cdots}{(\Diamond R \vee p) \to p, \Diamond E, \Diamond O}\,\vee_l}{R \to \Diamond E, (p \vee \Diamond O)}\,R, \vee_r}{\Diamond R \to p, \Diamond\Diamond E, \Diamond(p \vee \Diamond O)}\,\Diamond}{\Diamond R \to p, \Diamond\Diamond E, O}\,O \qquad p \to p, \Diamond\Diamond E, O}{(p \vee \Diamond R) \to (p \vee \Diamond\Diamond E), O}\,\vee_l, \vee_r}{R \to E, O}\,R,E$$

Proof by induction vs cyclic proofs

- principle of induction
- local
- non-algorithmic

- principle of infinite decent
- global
- algorithmic
- symmetric treatment of inductive and co-inductive properties

Section 6

Modal μ-calculus

Modal Logic revisited

Syntax: $p \mid \bar{p} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Box\varphi \mid \Diamond\varphi$

Semantics: For directed labelled graph $T = (S, \to, \lambda)$:

$$\|p\|^T = \{u \in S \mid p \in \lambda(u)\}$$
$$\|\varphi \wedge \psi\|^T = \|\varphi\|^T_V \cap \|\psi\|^T$$
$$\|\Box\varphi\|^T = \{u \in S \mid \text{for all } v, \text{ if } u \to v \text{ then } v \in \|\varphi\|^T\ \}$$

Examples: $q \vee \Diamond q \vee \Diamond\Diamond q$; $\underbrace{\Box \cdots \Box}_{n} \Diamond p$.

Expressive limitation: properties along arbitrary (finite or infinite) paths in a structure cannot be presented

Infinite path quantification:

1. $s \models \bigvee_n \Diamond^n q$ – q is reachable from s
2. $s \models \bigwedge_n \Box^n \Diamond p$ – All paths from s always satisfy $\Diamond p$

Fixed point quantifiers

1. $s \models q \vee \Diamond q \vee \Diamond\Diamond q \vee \cdots$ iterating $\mathsf{x} \mapsto q \vee \Diamond \mathsf{x}$ on $\bot$

Given $\varphi(\mathsf{x})$ (positive in x) we introduce

- $\mu \mathsf{x}\, \varphi =$ least fixed point of $\mathsf{x} \mapsto \varphi(\mathsf{x})$ Liveness **inductive**
- $\nu \mathsf{x}\, \varphi =$ greatest fixed point of $\mathsf{x} \mapsto \varphi(\mathsf{x})$ Safety **co-inductive**

A note on expressiveness

$$\mu \mathsf{x}(p \vee \Box \mathsf{x}) : \text{all paths contain a } p$$
$$\mu \mathsf{x}(p \vee \Box \mathsf{x}) \neq p \vee \Box p \vee \Box\Box p \vee \cdots$$

$$\mu \mathsf{x}(p \vee \Box \mathsf{x}) = p \vee \Box p \vee \Box\Box p \vee \cdots \vee \Box^{\omega} p \vee \cdots$$

Modal μ-calculus

Syntax $p \mid \overline{p} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mathsf{x} \mid \mu\mathsf{x}\,\varphi \mid \nu\mathsf{x}\,\varphi$

Semantics Fix $T = (S, \to, \lambda)$. Given $\varphi(\mathsf{x})$ define

$$f_\varphi \colon 2^S \to 2^S$$
$$U \mapsto \|\varphi(U)\|^T$$

$$\|\mu\mathsf{x}\,\varphi\|^T = \text{least fixed point of } f_\varphi = \bigcap \{U \subseteq S \mid f_\varphi(U) \subseteq U\}$$

$$\|\nu\mathsf{x}\,\varphi\|^T = \text{greatest fixed point of } f_\varphi = \bigcup \{U \subseteq S \mid U \subseteq f_\varphi(U)\}$$

Duality Define $\overline{\varphi}$ as the De Morgan dual of φ:

$$\overline{\mu\mathsf{x}\varphi(\mathsf{x})} = \nu\mathsf{x}\overline{\varphi}(\mathsf{x}) \qquad \|\overline{\varphi}\|^T = S \setminus \|\varphi\|^K$$

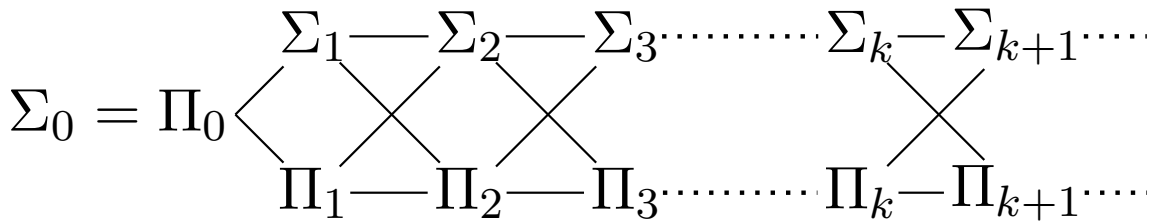
Slogans and more examples

ν is **looping/safety** and μ is **finite looping/liveness**

- $\nu \mathsf{x}(\Box \mathsf{x} \wedge \varphi)$: **always** φ
- $\mu \mathsf{x}(\Diamond \mathsf{x} \vee \varphi)$: **reachable** φ
- $\mu \mathsf{x}(\psi \vee (\varphi \wedge \Box \mathsf{x}))$: φ **until** ψ
- $\nu \mathsf{x} \Box(\varphi \wedge \mathsf{x})$: φ is **common knowledge**
- $\nu \mathsf{x} \mu \mathsf{y}\big((p \vee \Diamond \mathsf{y}) \wedge \Diamond \mathsf{x}\big)$: a path along which p **holds infinitely often**
- $\nu \mathsf{x}\big(\mu \mathsf{y}(p \vee \Diamond \mathsf{y}) \wedge \Diamond \mathsf{x}\big)$: a path along which p **is always reachable**

Hierarchies in μ-calculus

- Count alternations between the two quantifiers **Simple hierarchy**
- Count the *genuine* alternations between least and greatest fixed point quantifiers (**depth**) **Alternation hierarchy**



This hierarchy is strict (Bradfield 1998).

Connexion:

- LTL can be captured by μ-calculus formulas of depth 2
- CTL, PDL can be captured by alternation-free μ-calculus
- CTL*: depth 3

Nice properties

- Expressive: LTL, PDL, CTL, CTL*
- Robustly decidability: model checking, satisfiability
- Closed under bisimulation — but logic is not compact.
- Equivalent to: alternating parity automata, parity games
- Tree Model Property: every satisfiable formula has a tree model.
- Finite Model Property: every satisfiable formula has a finite model.

Section 7

Proof Systems

Sequent calculus Koz for μ-calculus

$$\frac{}{\varphi, \overline{\varphi}}\ \varphi \qquad \frac{\Gamma}{\Gamma, \varphi}\ \text{weak}$$

$$\frac{\Gamma, \varphi, \psi}{\Gamma, \varphi \vee \psi}\ \vee \qquad \frac{\Gamma, \varphi \qquad \Gamma, \psi}{\Gamma, \varphi \wedge \psi}\ \wedge \qquad \frac{\Gamma, \varphi}{\Diamond\Gamma, \Box\varphi}\ \text{mod}$$

$$\frac{\Gamma, \varphi(\sigma \mathsf{x} \varphi(\mathsf{x}))}{\Gamma, \sigma \mathsf{x} \varphi}\ \sigma \qquad \frac{\Gamma, \varphi(\overline{\Gamma})}{\Gamma, \nu \mathsf{x} \varphi}\ \text{ind} \qquad \frac{\Gamma, \overline{\varphi} \qquad \Gamma, \varphi}{\Gamma}\ \text{cut}$$

Abbreviation.

- " , " read as disjunction
- $\Gamma = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ finite set of formulas
- $\Diamond\Gamma := \{\Diamond\varphi_1, \Diamond\varphi_2, \ldots, \Diamond\varphi_n\}$
- $\overline{\Gamma} := \overline{\varphi}_1 \wedge \overline{\varphi}_2 \wedge \ldots \wedge \overline{\varphi}_n$
- $\sigma \in \{\mu, \nu\}$

Validity and proofs

A closed formula φ is **valid** iff $\|\varphi\|^T = S$ for every $T = (S, \to, \lambda)$.

Theorem (Kozen 1983; Walukiewicz 2000)

For every closed φ we have $\models \varphi$ *iff* $\mathsf{Koz} \vdash \varphi$.

Soundness Proved by Kozen (1983).
Completeness aconjuntive fragment; full μ-calculus

1. Completeness of disjunctive formulas: tableaux.
2. Provable equivalence between disjunctive and μ-formuas: tableaux, games, automata.

Modal μ-calculus revisited

μ-formulas

$$p \mid \overline{p} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \mathsf{x} \mid \mu\mathsf{x}\,\varphi \mid \nu\mathsf{x}\,\varphi$$

μ-quantifier: least fixed point

- $\varphi(\mu\mathsf{x}\varphi) \to \mu\mathsf{x}\varphi$
- $\varphi(\psi) \to \psi \vdash \mu\mathsf{x}\varphi \to \psi$

ν-quantifier: greatest fixed point

- $\nu\mathsf{x}\varphi \to \varphi(\nu\mathsf{x}\varphi)$
- $\psi \to \varphi(\psi) \vdash \psi \to \nu\mathsf{x}\varphi$

ν-regenration

$$\frac{\Gamma, \varphi(\nu\mathsf{x}\varphi(\mathsf{x}))}{\Gamma, \nu\mathsf{x}\varphi}\ \nu$$

ν-induction

$$\frac{\Gamma, \varphi(\overline{\Gamma})}{\Gamma, \nu\mathsf{x}\varphi}\ \text{ind}$$

Axioms and rules of proof system **Fix**

$$\frac{}{\varphi, \bar{\varphi}} \qquad \frac{\Gamma}{\Gamma, \varphi}\ \text{weak}$$

$$\frac{\Gamma, \varphi, \psi}{\Gamma, \varphi \vee \psi}\ \vee \qquad \frac{\Gamma, \varphi \qquad \Gamma, \psi}{\Gamma, \varphi \wedge \psi}\ \wedge \qquad \frac{\Gamma, \varphi}{\Diamond\Gamma, \Box\varphi}\ \text{mod}$$

$$\frac{\Gamma, \varphi(\mu \mathrm{x} \varphi)}{\Gamma, \mu \mathrm{x} \varphi}\ \mu \qquad \frac{\Gamma, \varphi(\nu \mathrm{x} \varphi)}{\Gamma, \nu \mathrm{x} \varphi}\ \nu$$

A **pre-tableau** is a finitely branching tree in axioms and rules of Fix.

Principal and **residual** formulas of a rule are defined analogous to that of modal logic.

Examples of pre-tableaux

The formula $(\nu x \Diamond x) \vee (\mu y \Box y)$ has (essentially) one pre-tableau:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\nu x \Diamond x, \mu y \Box y}{\Diamond \nu x \Diamond x, \Box \mu y \Box y}\,\text{mod}}{\Diamond \nu x \Diamond x, \mu y \Box y}\,\mu}{\nu x \Diamond x, \mu y \Box y}\,\nu}{\nu x \Diamond x \vee \mu y \Box y}\,\vee$$

Traces

Fix a pre-tableau $t = \langle S, \to, \lambda \rangle$ for Γ and a path $\mathbb{P}$ through t. A finite sequence of formulas $\varphi_0, \varphi_1, \ldots, \varphi_n$ is a **trace** through $\mathbb{P}$ if

- $\varphi_i \in \lambda(\mathbb{P}(i))$ for each $i \leq n$;
- $\varphi_{i+1} = \varphi_i$ if φ_i is not principal in the rule from $\mathbb{P}(i)$ to $\mathbb{P}(i+1)$, otherwise φ_{i+1} is the residual subformula of φ_i in the label of $\mathbb{P}(i+1)$.

An infinite seq. of formulas $(\varphi_i)_i$ is a **trace** if every initial seq. is a trace.

Lemma

*For every infinite trace there exists a variable that appears infinitely often in the trace and **subsumes** all other infinitely occurring variables.*

The **subsumption ordering** on variables appearing in a formula is the order of the the associated fixed point quantifier read from left to right.

Example in the formula $\mu \mathsf{x} \nu \mathsf{y}.(\Diamond \mathsf{x} \vee p) \wedge \Diamond \mathsf{y}$ the variable x subsumes y, in notation $\mathsf{x} \leq \mathsf{y}$.

Example of traces

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\nu x \Diamond x, \mu y \Box y}{\Diamond \nu x \Diamond x, \Box \mu y \Box y}\,\text{mod}}{\Diamond \nu x \Diamond x, \mu y \Box y}\,\mu}{\nu x \Diamond x, \mu y \Box y}\,\nu}{\nu x \Diamond x \vee \mu y \Box y}\,\vee$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\nu x \Diamond x, \mu y \Box y}{\Diamond \nu x \Diamond x, \Box \mu y \Box y}\,\text{mod}}{\Diamond \nu x \Diamond x, \mu y \Box y}\,\mu}{\nu x \Diamond x, \mu y \Box y}\,\nu}{\nu x \Diamond x \vee \mu y \Box y}\,\vee$$

μ-traces and ν-traces

In each infinite trace the unique variable identified by the lemma will be referred to as the "**most significant variable**" of the trace.

We call an infinite trace

- a **μ-trace** if its most significant variable is a μ-variable
- a **ν-trace** if its most significant variable is a ν-variable

A pre-tableau is a **tableau** if

1. the sequent at each leaf is an axiom, and
2. every infinite path contains a **ν-trace** (progressive thread).

Examples of μ and ν traces

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\nu x \Diamond x, \mu y \Box y}{\Diamond \nu x \Diamond x, \Box \mu y \Box y}\,\text{mod}}{\Diamond \nu x \Diamond x, \mu y \Box y}\,\mu}{\nu x \Diamond x, \mu y \Box y}\,\nu}{\nu x \Diamond x \vee \mu y \Box y}\,\vee \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\nu x \Diamond x, \mu y \Box y}{\Diamond \nu x \Diamond x, \Box \mu y \Box y}\,\text{mod}}{\Diamond \nu x \Diamond x, \mu y \Box y}\,\mu}{\nu x \Diamond x, \mu y \Box y}\,\nu}{\nu x \Diamond x \vee \mu y \Box y}\,\vee$$

Question What is the subsumption order? What kind of traces do we have here?

Another example

It is not hard to check (semantically) that the following formula is valid.

$$\mu \mathsf{x} \nu \mathsf{y} \varphi(\mathsf{x}, \mathsf{y}) \to \nu \mathsf{y} \mu \mathsf{x} \varphi(\mathsf{x}, \mathsf{y})$$

Let $\psi_1 := \nu \mathsf{x} \mu \mathsf{y} \overline{\varphi}(\mathsf{x}, \mathsf{y})$ and $\psi_2 := \nu \mathsf{z} \mu \mathsf{w} \varphi(\mathsf{z}, \mathsf{w})$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mu \mathsf{y} \overline{\varphi}(\psi_1, \mathsf{y}), \mu \mathsf{w} \varphi(\mathsf{w}, \psi_2)}{\psi_1, \mu \mathsf{w} \varphi(\mathsf{w}, \psi_2)}\, \nu_{\mathsf{x}} \qquad \mu \mathsf{y} \overline{\varphi}(\psi_1, \mathsf{y}), \psi_2}{\vdots}}{\overline{\varphi}(\psi_1, \mu \mathsf{y} \overline{\varphi}(\psi_1, \mathsf{y})), \varphi(\mu \mathsf{w} \varphi(\mathsf{w}, \psi_2), \psi_2)}\, \mu_{\mathsf{y}}, \mu_{\mathsf{w}}}{\mu \mathsf{y} \overline{\varphi}(\psi_1, \mathsf{y}), \mu \mathsf{w} \varphi(\mathsf{z}, \psi_2)}\, \nu_{\mathsf{z}}}{\mu \mathsf{y} \overline{\varphi}(\psi_1, y), \psi_2}\, \nu_{\mathsf{x}}}{\psi_1, \psi_2}$$

Question What is the subsumption order? What kind of traces do we have here? $\mathsf{x} \leq \mathsf{y} \leq \mathsf{z} \leq \mathsf{w} \leq \cdots$.

Tableaux characterisation of validity

Theorem (Niwinski, Walukiewicz 1996)

For every closed formula φ, there exists a tableau-proof for φ iff φ is valid.

Theorem (Jungteerapanich, 2009; Stirling, 2014)

φ is valid if and only if there is a ***regular/cyclic*** *proof (in the axioms and rules of* Fix*).*

Some words on regularisation ...

Virtues of cyclic approach

- proof search
- decidability
- interpolation
- applications in other areas such as databases

$\vdots$

State-of-the-art

Cyclic proofs have been investigated for many logics including:

- **First-order logic with inductive definitions** (Brotherston, 2005; Brotherston and Simpson, 2011; Berardi and Tatsuta, 2019)
- **Arithmetic** (Simpson, 2017; Berardi and Tatsuta, 2017; Das, 2019)
- **Linear logic** (Baelde et al., 2016; De and Saurin, 2019)
- **Modal and dynamic logics** (Sprenger and Dam, 2003; Jungteerapanich, 2009; Shamkanov, 2014; Stirling, 2014; Kokkinis and Studer, 2016; Afshari and Leigh, 2017; Enqvist et al., 2019; Afshari and Leigh, 2020)
- **Program semantics** (Santocanale, 2002; Docherty and Rowe, 2019)
- **Automated theorem proving** (Brotherston, Gorogiannis, et al., 2012; Rowe and Brotherston, 2017; Tellez and Brotherston, 2020)

Possible research questions

- cyclic proofs for extension of modal logic with
  - converse modalities
  - nominals
  - counting
- cyclic proofs of fragment guarded fixed point logics
- cyclic proofs for PDL
- intuitionistic cycles
- higher type cycles
- higher order cycles

What sort of questions?

- designing sound (and complete) axiomatisation
- devising proof sreach algorithms
- investigating metalogical properties such as consistency, interpolation, cut-elimination