Expressiveness of Temporal Logic

Course in Automatic system Verification: Theory and Applications

Borelli Roberto
borelli.roberto@spes.uniud.it

University of Udine

July 3, 2023

Abstract

In this work, I consider the expressive power of various temporal logics. First, I recall some basic results about expressiveness of first order logic. Then I consider the case of LTL and I show a theorem that can be used to prove that the concept of parity is not definable in this context. I discuss a counterexample that proves that the mentioned theorem doesn't directly apply to $LTL{+}P$ and I briefly highlight how a possible investigation may lead to a generalization of the theorem to the $LTL{+}P$ case. Next, I relate first order definable languages with LTL ones and I present an extension to LTL which allows us to increase the expressive power and capture regular languages without changing the complexity of the decision procedure. Finally, I move to the more interesting case of interval logic. I introduce the notion of bisimulation and its use in modal logic and, in particular, I show how to apply it to prove that the logic $A\overline{A}$ is strictly more expressive than its future fragment A.

Contents

Preliminaries

Model theory

Model theory offers a way to measure expressive power [Stu00]

Model theory

Model theory offers a way to measure expressive power [Stu00]

Two fundamental problems:

- Which kind of structures are definable in a logic $\mathcal{L}$?
- Which pairs of structures are distinguishable by means of formulae of logic $\mathcal{L}$?

Model theory

Model theory offers a way to measure expressive power [Stu00]

Two fundamental problems:

- Which kind of structures are definable in a logic $\mathcal{L}$?
- Which pairs of structures are distinguishable by means of formulae of logic $\mathcal{L}$?

Basic definitions:

- Let $\mathcal{L}$ be a logic, let σ be a signature, and let C be a class of σ-structures

Model theory

Model theory offers a way to measure expressive power [Stu00]

Two fundamental problems:

- Which kind of structures are definable in a logic $\mathcal{L}$?
- Which pairs of structures are distinguishable by means of formulae of logic $\mathcal{L}$?

Basic definitions:

- Let $\mathcal{L}$ be a logic, let σ be a signature, and let C be a class of σ-structures
- We say C is $\mathcal{L}$-**elementary** if there exists a possibly infinite set of formulae $\tau = \{\tau_1, \tau_2, \tau_3, \dots\}$ such that for each σ-structure s it holds $s \in C \iff s \models \tau$
- If τ is finite we say C is $\mathcal{L}$-**basic elementary**

Expressive power of FO Logic

- As seen in the first part of the course we have several tools to prove that properties are not expressible in first order logic:
  - Ehrenfeucht-Fraïssé games
  - 0/1 laws
  - Locality of first order formulas (Hanf/Gaiffman theorems)
  - Compactness theorem (just for the infinite setting)

FO^k = FO with at most k different variables' names

Expressive power of FO Logic

- As seen in the first part of the course we have several tools to prove that properties are not expressible in first order logic:
  - Ehrenfeucht-Fraïssé games
  - 0/1 laws
  - Locality of first order formulas (Hanf/Gaiffman theorems)
  - Compactness theorem (just for the infinite setting)
- Moving to other logics we have other tools and theorems, as a matter of fact we can use Pebble games dealing with FO^k logic

FO^k = FO with at most k different variables' names

Expressive power of FO Logic

- As seen in the first part of the course we have several tools to prove that properties are not expressible in first order logic:
    - Ehrenfeucht-Fraïssé games
    - 0/1 laws
    - Locality of first order formulas (Hanf/Gaiffman theorems)
    - Compactness theorem (just for the infinite setting)
- Moving to other logics we have other tools and theorems, as a matter of fact we can use Pebble games dealing with FO^k logic
- Temporal logics ??

FO^k = FO with at most k different variables' names

Even in FO: the finite case

- The concept of *even* is not capturable by *FO* definable languages

$[k]$ is a shortcut for $\{0, \ldots, k\}$

Even in FO: the finite case

- The concept of *even* is not capturable by *FO* definable languages
- The language $(aa)^*$ is regular but not star-free or equivalently not first order definable. Two ways to prove the result:
  - The minimum DFA has a counter
  - EF games: consider the set of structures $EVEN_{aa} = \{([2k+1], +1, <, Q_a) \ : \ k >= 0, Q_a = [2k+1]\}$

$[k]$ is a shortcut for $\{0, \ldots, k\}$

Even in FO: the finite case

- The concept of *even* is not capturable by *FO* definable languages
- The language $(aa)^*$ is regular but not star-free or equivalently not first order definable. Two ways to prove the result:
  - The minimum DFA has a counter
  - EF games: consider the set of structures $EVEN_{aa} = \{([2k+1], +1, <, Q_a) \ : \ k >= 0, Q_a = [2k+1]\}$
- The language $(ab)^*$ is star-free. In fact:
  - The minimum DFA doesn't have a counter
  - We can build a generalized regular expression with star height 0 $(ab)^* = \overline{b\Sigma^* \cup \Sigma^* a \cup \Sigma^* aa\Sigma^* \cup \Sigma^* bb\Sigma^*} = \overline{b\overline{\emptyset} \cup \overline{\emptyset}a \cup \overline{\emptyset}aa\overline{\emptyset} \cup \overline{\emptyset}bb\overline{\emptyset}}$
- Notice that in $(ab)^*$,we are not counting the number of a's

$[k]$ is a shortcut for $\{0, \ldots, k\}$

Even in FO: the infinite case

- Considering ω-words we need a different notion of parity, in fact we have $(pp)^\omega = (p)^\omega$
- $\Sigma = \{p, \neg p\}$
  $EVEN_{POSITIONS} = (p(p + \neg p))^\omega$
- As can be shown with similar arguments $EVEN_{POSITIONS}$ is not ω-star-free
- On the contrary $(p \cdot \neg p)^\omega$ is ω-star-free
  - $(p \ \neg p)^\omega = \emptyset \cdot ((p \cdot \neg p)^*)^\omega$
  - $\emptyset$ and $(p \cdot \neg p)^*$ are star-free
  - $(p \cdot \neg p)^* \cdot (p \cdot \neg p)^* \subseteq (p \cdot \neg p)^*$

An ω-language L is ω-star-free if it can be written as $L = \cup_{i \in I} U_i \cdot V_i^\omega$ with U, V star-free and $V \cdot V \subseteq V$

Expressive power of LTL

P. Wolper. "Temporal logic can be more expressive". In: *Information and Control* 56.1 (1983), pp. 72–99. ISSN: 0019-9958

Even in LTL

- $EVEN_{COMPUTATIONS}(p) =$
  $\{\sigma : \sigma$ is a computation on which p is true on all even states$\}$
- $EVEN_{COMPUTATIONS}(p)$ is not expressible in LTL

Even in LTL

- $EVEN_{COMPUTATIONS}(p) =$
  $\{\sigma : \ \sigma$ is a computation on which p is true on all even states$\}$
- $EVEN_{COMPUTATIONS}(p)$ is not expressible in LTL
- What about $\psi = p \wedge G(p \rightarrow X\neg p) \wedge G(\neg p \rightarrow Xp)$?
- It is not satisfied by the model where p is always true

Even in LTL

- $EVEN_{COMPUTATIONS}(p) =$
  $\{\sigma : \sigma$ is a computation on which p is true on all even states$\}$
- $EVEN_{COMPUTATIONS}(p)$ is not expressible in LTL
- What about $\psi = p \wedge G(p \rightarrow X\neg p) \wedge G(\neg p \rightarrow Xp)$?
- It is not satisfied by the model where p is always true
- What about $\psi' = \psi \vee Gp$?
- Still doesn't capture the notion of $EVEN_{COMPUTATIONS}(p)$
- There are infinite computations in $EVEN_{COMPUTATIONS}(p)$ which are not model ψ

Even in LTL

- $EVEN_{COMPUTATIONS}(p) =$
  $\{\sigma : \ \sigma \text{ is a computation on which } p \text{ is true on all even states}\}$

- $EVEN_{COMPUTATIONS}(p)$ is not expressible in LTL

- What about $\psi = p \wedge G(p \rightarrow X\neg p) \wedge G(\neg p \rightarrow Xp)$?

- It is not satisfied by the model where p is always true

- What about $\psi' = \psi \vee Gp$?

- Still doesn't capture the notion of $EVEN_{COMPUTATIONS}(p)$

- There are infinite computations in $EVEN_{COMPUTATIONS}(p)$ which are not model ψ

- As a matter of fact we can use infinite formulas of finite length
  $\tau = \{p, XXp, XXXXp, \dots\}$

Even in LTL

To prove the previous result we consider the sequence $p^i(\neg p)p^\omega$

0	1	2	$\cdots$	$i-1$	i	$i+1$	$i+2$	$\cdots$
p	p	p	$\dots$	p	$\neg p$	p	p	$\dots$

Theorem

Let $f(p)$ be an LTL formula,
Let n denote the number of X operators in f
Every sequence $p^i(\neg p)p^\omega$ where $i > n$ has the same truth value on f:
$p^{(n+1)}(\neg p)p^\omega \models f(p) \iff p^{(n+2)}(\neg p)p^\omega \models f(p) \iff \dots$

Even in LTL

Proof.

Denote by $eval_i(f)$ the truth value of f on the sequence $p^i(\neg p)p^\omega$, we want to prove that $eval_{n+1}(f(p)) = eval_{n+2}(f(p)) = eval_{n+3}(f(p)) = \ldots$ In other words we want to prove that $eval_i(f(p))$ is independent of $i > n$.

Even in LTL

Proof.

Denote by $eval_i(f)$ the truth value of f on the sequence $p^i(\neg p)p^\omega$, we want to prove that $eval_{n+1}(f(p)) = eval_{n+2}(f(p)) = eval_{n+3}(f(p)) = \dots$ In other words we want to prove that $eval_i(f(p))$ is independent of $i > n$. Consider $j > n$, we proceed by induction on the structure of the formula $f(p)$ (we suppose the theorem holds on all the sub-formulas):

Even in LTL

Proof.

Denote by $eval_i(f)$ the truth value of f on the sequence $p^i(\neg p)p^\omega$, we want to prove that $eval_{n+1}(f(p)) = eval_{n+2}(f(p)) = eval_{n+3}(f(p)) = \ldots$ In other words we want to prove that $eval_i(f(p))$ is independent of $i > n$. Consider $j > n$, we proceed by induction on the structure of the formula $f(p)$ (we suppose the theorem holds on all the sub-formulas):

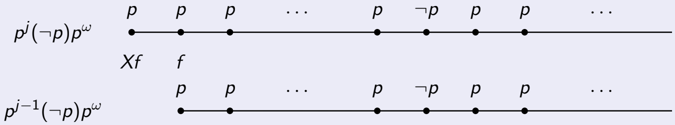
- Propositional cases are straightforward.

Even in LTL

Proof.

Denote by $eval_i(f)$ the truth value of f on the sequence $p^i(\neg p)p^\omega$, we want to prove that $eval_{n+1}(f(p)) = eval_{n+2}(f(p)) = eval_{n+3}(f(p)) = \dots$ In other words we want to prove that $eval_i(f(p))$ is independent of $i > n$. Consider $j > n$, we proceed by induction on the structure of the formula $f(p)$ (we suppose the theorem holds on all the sub-formulas):

- Propositional cases are straightforward.
- Case $f(p) = Xf$. We have $eval_j(Xf) = eval_{j-1}(f)$. f contains $n-1$ X operators so we have $j-1 > n-1$ and hence by the inductive hypothesis the value of $eval_{j-1}(f)$ is independent of j.



Even in LTL

Proof.

- Case $f(p) = f_1 \ U \ f_2$. From
  - $(f_1 \ U \ f_2) = f_2 \vee (f_1 \wedge X(f_1 \ U \ f_2))$
  - $eval_j(Xf) = eval_{j-1}(f)$

  we have $eval_j(f_1 \ U \ f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_1 \ U \ f_2))$

Even in LTL

Proof.

- Case $f(p) = f_1 \ U \ f_2$. From
  - $(f_1 \ U \ f_2) = f_2 \vee (f_1 \wedge X(f_1 \ U \ f_2))$
  - $eval_j(Xf) = eval_{j-1}(f)$

  we have $eval_j(f_1 \ U \ f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_1 \ U \ f_2))$ and unfolding n times:
  $eval_j(f_1 \ U \ f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_2) \vee (eval_{j-1}(f_1) \wedge \cdots \wedge (eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_n(f_1 \ U \ f_2))) \ldots))$
  By the inductive hypothesis
  $eval_j(f_k) = eval_{j-1}(f_k) = \cdots = eval_{n+1}(f_k)$ for $k = 1, 2$

Even in LTL

Proof.

- Case $f(p) = f_1 \ U \ f_2$. From
  - $(f_1 \ U \ f_2) = f_2 \vee (f_1 \wedge X(f_1 \ U \ f_2))$
  - $eval_j(Xf) = eval_{j-1}(f)$

  we have $eval_j(f_1 \ U \ f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_1 \ U \ f_2))$ and unfolding n times:
  $eval_j(f_1 \ U \ f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_2) \vee (eval_{j-1}(f_1) \wedge \cdots \wedge (eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_n(f_1 \ U \ f_2))) \ldots))$
  By the inductive hypothesis
  $eval_j(f_k) = eval_{j-1}(f_k) = \cdots = eval_{n+1}(f_k)$ for $k = 1, 2$
  so we have:
  $eval_j(f_1 \ U \ f_2) = eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_n(f_1 \ U \ f_2))$

Even in LTL

Proof.

- Case $f(p) = f_1 \; U \; f_2$. From
  - $(f_1 \; U \; f_2) = f_2 \vee (f_1 \wedge X(f_1 \; U \; f_2))$
  - $eval_j(Xf) = eval_{j-1}(f)$

  we have $eval_j(f_1 \; U \; f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_1 \; U \; f_2))$
  and unfolding n times:
  $eval_j(f_1 \; U \; f_2) = eval_j(f_2) \vee (eval_j(f_1) \wedge eval_{j-1}(f_2) \vee (eval_{j-1}(f_1) \wedge \cdots \wedge (eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_n(f_1 \; U \; f_2))) \ldots))$
  By the inductive hypothesis
  $eval_j(f_k) = eval_{j-1}(f_k) = \cdots = eval_{n+1}(f_k)$ for $k = 1, 2$
  so we have:
  $eval_j(f_1 \; U \; f_2) = eval_{n+1}(f_2) \vee (eval_{n+1}(f_1) \wedge eval_n(f_1 \; U \; f_2))$ and hence
  $eval_{n+1}(f_1 \; U \; f_2) = eval_{n+2}(f_1 \; U \; f_2) = eval_{n+3}(f_1 \; U \; f_2) = \ldots$

$\square$

Even in LTL

Corollary

For each $m \geq 2$ *the set* $m_{COMPUTATIONS}(p) =$
$\{\sigma.\ \sigma$ *is a computation on which* p *is true at each state multiple of* $m\}$ *is not expressible in LTL*

Even in LTL

Corollary

For each $m \geq 2$ the set $m_{COMPUTATIONS}(p) = \{\sigma.\ \sigma$ is a computation on which p is true at each state multiple of $m\}$ is not expressible in LTL

Proof.

Suppose that the formula $f(p)$ with n X operators captures $m_{COMPUTATIONS}(p)$ for a given m.

Even in LTL

Corollary

For each $m \geq 2$ *the set* $m_{COMPUTATIONS}(p) = \{\sigma.\ \sigma$ *is a computation on which* p *is true at each state multiple of* $m\}$ *is not expressible in LTL*

Proof.

Suppose that the formula $f(p)$ with n X operators captures $m_{COMPUTATIONS}(p)$ for a given m. Choose k such that $km > n$. Then we have

- $p^{km+1}(\neg p)p^{\omega} \in m_{COMPUTATIONS}(p)$
- $p^{km}(\neg p)p^{\omega} \notin m_{COMPUTATIONS}(p)$

but $eval_{km+1}(f(p)) = eval_{km}(f(p))$ □

LTL and FO

The previous result was not a case, in fact we have an equivalence between LTL and FO theory of discrete linear orders [Kam68]

LTL and FO

The previous result was not a case, in fact we have an equivalence between LTL and FO theory of discrete linear orders [Kam68]

Theorem

$(LTL + P)[XU, XS] \subseteq FO^3[<]$

XU = Strong Strict Until, XS = Strong Strict Since

LTL and FO

The previous result was not a case, in fact we have an equivalence between LTL and FO theory of discrete linear orders [Kam68]

Theorem

$(LTL + P)[XU, XS] \subseteq FO^3[<]$

XU = Strong Strict Until, XS = Strong Strict Since

Theorem

$FO[<] \subseteq (LTL + P)[XU, XS]$

LTL and FO

The previous result was not a case, in fact we have an equivalence between LTL and FO theory of discrete linear orders [Kam68]

Theorem

$(LTL + P)[XU, XS] \subseteq FO^3[<]$

XU = Strong Strict Until, XS = Strong Strict Since

Theorem

$FO[<] \subseteq (LTL + P)[XU, XS]$

Corollary

$FO[<] = FO^3[<]$

Express Even in LTL

To express EVEN in LTL, [Wol83] proposed 2 solutions

1. Add quantifiers to *LTL* and use the the formula
   $\psi(p) = \exists q(q \wedge G(q \rightarrow X\neg q) \wedge G(\neg q \rightarrow Xq) \wedge G(q \rightarrow p))$
   The logic obtained in this way is called *QLTL* and its satisfiability problem is non-elementary [KPV01]
2. Add the operator $even(p)$ directly in the syntax of LTL

Express Even in LTL

To express EVEN in LTL, [Wol83] proposed 2 solutions

1. Add quantifiers to *LTL* and use the the formula
   $\psi(p) = \exists q(q \wedge G(q \rightarrow X\neg q) \wedge G(\neg q \rightarrow Xq) \wedge G(q \rightarrow p))$
   The logic obtained in this way is called *QLTL* and its satisfiability problem is non-elementary [KPV01]
2. Add the operator $even(p)$ directly in the syntax of LTL

Regarding point 2, which kind of operators can we add to LTL to increase its expressive power while keeping its decision problem PSPACE-complete?

Express Even in LTL

To express EVEN in LTL, [Wol83] proposed 2 solutions

1. Add quantifiers to *LTL* and use the the formula
   $\psi(p) = \exists q(q \wedge G(q \rightarrow X\neg q) \wedge G(\neg q \rightarrow Xq) \wedge G(q \rightarrow p))$
   The logic obtained in this way is called *QLTL* and its satisfiability problem is non-elementary [KPV01]
2. Add the operator $even(p)$ directly in the syntax of LTL

Regarding point 2, which kind of operators can we add to LTL to increase its expressive power while keeping its decision problem PSPACE-complete?

$\rightarrow$ We can add operators corresponding to right-linear grammars

The logic obtained in this way is called **ETL** (Extended Temporal Logic)

ETL: Extended Temporal Logic

- Grammar $G = (V, T, P, S)$ with $n = |V|$ and $m = |T|$
  Variables: $V = \{V_1, \ldots, V_n\}$
  Terminals: $T = \{t_1, \ldots, t_m\}$
  Productions of the form $V_i \rightarrow t_{ij}$ or $V_i \rightarrow t_{ij} V_{ij}$

ETL: Extended Temporal Logic

- Grammar $G = (V, T, P, S)$ with $n = |V|$ and $m = |T|$
  Variables: $V = \{V_1, \ldots, V_n\}$
  Terminals: $T = \{t_1, \ldots, t_m\}$
  Productions of the form $V_i \rightarrow t_{ij}$ or $V_i \rightarrow t_{ij} V_{ij}$

- To each variable V_i we associate an m-ary operator $\mathcal{G}_i(f_1, \ldots, f_m)$

- **ETL syntax** $\phi ::= AP \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid X\phi \mid \phi U \phi \mid \mathcal{G}_i(\phi, \ldots, \phi)$

ETL: Extended Temporal Logic

- Grammar $G = (V, T, P, S)$ with $n = |V|$ and $m = |T|$
  Variables: $V = \{V_1, \dots, V_n\}$
  Terminals: $T = \{t_1, \dots, t_m\}$
  Productions of the form $V_i \rightarrow t_{ij}$ or $V_i \rightarrow t_{ij} V_{ij}$
- To each variable V_i we associate an m-ary operator $\mathcal{G}_i(f_1, \dots, f_m)$
- **ETL syntax** $\phi ::= AP \mid \neg\phi \mid \phi \wedge \phi \mid \phi \vee \phi \mid X\phi \mid \phi U \phi \mid \mathcal{G}_i(\phi, \dots, \phi)$
- **ETL semantics** Given an LTL structure $M = (S, x, L)$ we say

$$M, x \models \mathcal{G}_i(f_1, \dots, f_m)$$
$$\Longleftrightarrow$$
$$\exists w \in L(G[S = V_i]) \text{ of the form } w = t_{w_0} t_{w_1} t_{w_2} \dots (1 \leq w_j \leq m) \text{ s.t. }$$
$$\forall j \geq 0 \;\; M, x^j \models f_{w_j}$$

where the j-th argument of $\mathcal{G}_i$, f_j, is the temporal formula related to the terminal t_j

Even in ETL

Consider the grammar G with S the initial symbol and with the following productions:

$S \to aR$

$R \to bS$

Even in ETL

Consider the grammar G with S the initial symbol and with the following productions:

$S \to aR$

$R \to bS$

We have $L(G) = (ab)^\omega$

Even in ETL

Consider the grammar G with S the initial symbol and with the following productions:

$S \rightarrow aR$

$R \rightarrow bS$

We have $L(G) = (ab)^{\omega}$

Consider the operator $\mathcal{G}_S(f_a, f_b)$

$\mathcal{G}_S(p, \neg p \vee p)$ expresses the property *even*(p)

Even in ETL

Consider the grammar G with S the initial symbol and with the following productions:

$S \to aR$

$R \to bS$

We have $L(G) = (ab)^\omega$

Consider the operator $\mathcal{G}_S(f_a, f_b)$

$\mathcal{G}_S(p, \neg p \vee p)$ expresses the property *even*(p)

$$
\begin{aligned}
M, x \models & \mathcal{G}_S(p, \neg p \vee p) \\
& \iff \forall j \; ((j \text{ is even} \to M, x^j \models f_a) \wedge (j \text{ is odd} \to M, x^j \models f_b)) \\
& \iff \forall j \; ((j \text{ is even} \to M, x^j \models p) \wedge (j \text{ is odd} \to M, x^j \models \neg p \vee p)) \\
& \iff \forall j \; (j \text{ is even} \to M, x^j \models p)
\end{aligned}
$$

ETL: Complexity

Theorem

The satisfiability problem for ETL is PSPACE-hard

The proof is a reduction from *finite automation inequivalence*. Given two DFA A and B we construct an ETL formula which is satisfiable iff $\mathcal{L}(A) \neq \mathcal{L}(B)$

If we allowed Context-free Grammar operators ETL would be undecidable.

ETL: Complexity

Theorem

The satisfiability problem for ETL is PSPACE-hard

The proof is a reduction from *finite automation inequivalence*. Given two DFA A and B we construct an ETL formula which is satisfiable iff $\mathcal{L}(A) \neq \mathcal{L}(B)$

If we allowed Context-free Grammar operators ETL would be undecidable.

A decision procedure can be obtained by extending the tableau method proposed by Manna and Pnueli (1981).

Theorem

The satisfiability problem for ETL is in PSPACE

Expressive power of PNL

D. Della Monica, A. Montanari, and P. Sala. "The Importance of the Past in Interval Temporal Logics: The Case of Propositional Neighborhood Logic". In: *Logic Programs, Norms and Action: Essays in Honor of Marek J. Sergot on the Occasion of His 60th Birthday*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 79–102. ISBN: 978-3-642-29414-3

Interval temporal logic

- Linearly-ordered domain: $\mathbb{D} = (\mathcal{D}, <)$
- The basic time-unit are strict intervals: $[i, j], i, j \in \mathcal{D}, i < j$
- Interval structure: $\mathbb{I}(\mathbb{D})$ the set of intervals over $\mathbb{D}$
- Interval model: $M = (\mathbb{I}(\mathbb{D}), V)$ where V is a labeling function $V : AP \to 2^{\mathbb{I}(\mathbb{D})}$

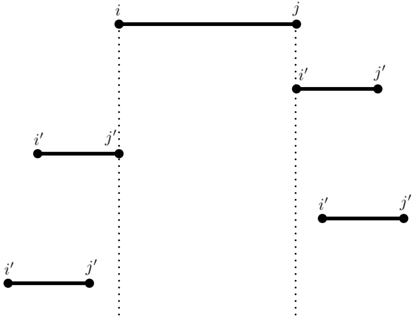
Interval temporal logic

- Linearly-ordered domain: $\mathbb{D} = (\mathcal{D}, <)$
- The basic time-unit are strict intervals: $[i, j], i, j \in \mathcal{D}, i < j$
- Interval structure: $\mathbb{I}(\mathbb{D})$ the set of intervals over $\mathbb{D}$
- Interval model: $M = (\mathbb{I}(\mathbb{D}), V)$ where V is a labeling function $V : AP \rightarrow 2^{\mathbb{I}(\mathbb{D})}$
- There are 13 $(6 \times 2 + 1)$ relations between pairs of intervals (Allen's interval relations)
- These relations are treated as accessibility relation in the Kripke structure
- To each relation R_X we associate a unary modal operator $\langle X \rangle$

Some of the Allen's relations

Among all 13 relations (which are mutually exclusive and jointly exhaustive), consider *after*, *meets* and their inverses

- $\langle A \rangle$: $[i, j] \ R_A \ [i', j'] \iff j = i'$
- $\langle \overline{A} \rangle$: $[i', j'] \ R_{\overline{A}} \ [i, j] \iff j' = i$
- $\langle L \rangle$: $[i, j] \ R_L \ [i', j'] \iff j < i'$
- $\langle \overline{L} \rangle$: $[i', j'] \ R_{\overline{L}} \ [i, j] \iff j' < i$



The logic $\mathcal{HS}$ and related fragments

The logic $\mathcal{HS}$:

- **syntax** $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle X \rangle \phi$
  where $\langle X \rangle \phi$ is any modal operator related to the Allen's relation R_X

- **semantics**
  $M, [i,j] \models \langle X \rangle \phi \iff \exists [i',j']$ s.t. $[i,j] \; R_X \; [i',j']$ and $M, [i',j'] \models \phi$
  where M is an interval model

The logic $\mathcal{HS}$ and related fragments

The logic $\mathcal{HS}$:

- **syntax** $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle X \rangle \phi$
  where $\langle X \rangle \phi$ is any modal operator related to the Allen's relation R_X
- **semantics**
  $M, [i,j] \models \langle X \rangle \phi \iff \exists [i',j']$ s.t. $[i,j]\ R_X\ [i',j']$ and $M, [i',j'] \models \phi$
  where M is an interval model

There are 4096 fragments of $\mathcal{HS}$ and about 90% of them are undecidable [Del11]

The logic $\mathcal{HS}$ and related fragments

The logic $\mathcal{HS}$:

- **syntax** $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle X \rangle \phi$
  where $\langle X \rangle \phi$ is any modal operator related to the Allen's relation R_X

- **semantics**
  $M, [i,j] \models \langle X \rangle \phi \iff \exists [i',j']$ s.t. $[i,j]\ R_X\ [i',j']$ and $M, [i',j'] \models \phi$
  where M is an interval model

There are 4096 fragments of $\mathcal{HS}$ and about 90% of them are undecidable [Del11]

We consider the following fragments:

- The logic RPNL A: $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle A \rangle \phi$
- The logic PNL $A\overline{A}$: $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle A \rangle \phi \mid \langle \overline{A} \rangle \phi$
- The logic $A\overline{L}$: $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \langle A \rangle \phi \mid \langle \overline{L} \rangle \phi$

$\mathcal{F}$-Bisimulation: definition

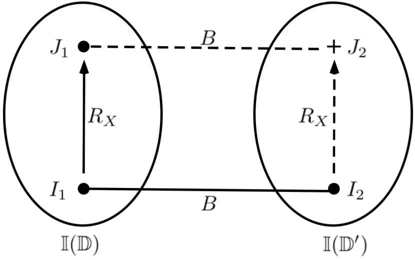
Let $\mathcal{F}$ be a logic, $M = (\mathbb{I}(\mathbb{D}), V)$ and $M' = (\mathbb{I}(\mathbb{D}'), V')$ two models
The binary relation $B \subseteq \mathbb{I}(\mathbb{D}) \times \mathbb{I}(\mathbb{D}')s$ is an $\mathcal{F}$-**bisimulation** if

1. $(I_1, I_2) \in B \implies$ the intervals I_1, I_2 satisfy the same propositional letters

$\mathcal{F}$-Bisimulation: definition

Let $\mathcal{F}$ be a logic, $M = (\mathbb{I}(\mathbb{D}), V)$ and $M' = (\mathbb{I}(\mathbb{D}'), V')$ two models
The binary relation $B \subseteq \mathbb{I}(\mathbb{D}) \times \mathbb{I}(\mathbb{D}')s$ is an $\mathcal{F}$-**bisimulation** if

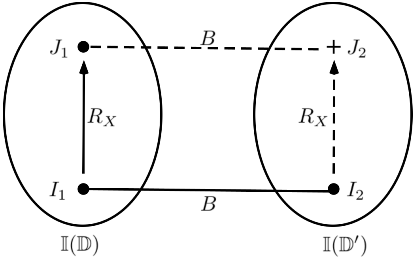
1. $(I_1, I_2) \in B \implies$ the intervals I_1, I_2 satisfy the same propositional letters
2. $(I_1, I_2) \in B \wedge (I_1, J_1) \in R_X \implies \exists J_2\ (I_2, J_2) \in R_X \wedge (J_1, J_2) \in B$



$\mathcal{F}$-Bisimulation: definition

Let $\mathcal{F}$ be a logic, $M = (\mathbb{I}(\mathbb{D}), V)$ and $M' = (\mathbb{I}(\mathbb{D}'), V')$ two models
The binary relation $B \subseteq \mathbb{I}(\mathbb{D}) \times \mathbb{I}(\mathbb{D}')s$ is an $\mathcal{F}$-**bisimulation** if

1. $(I_1, I_2) \in B \implies$ the intervals I_1, I_2 satisfy the same propositional letters
2. $(I_1, I_2) \in B \wedge (I_1, J_1) \in R_X \implies \exists J_2\ (I_2, J_2) \in R_X \wedge (J_1, J_2) \in B$



3. $(I_1, I_2) \in B \wedge (I_2, J_2) \in R_X \implies \exists J_1\ (I_1, J_1) \in R_X \wedge (J_1, J_2) \in B$

where I_1, I_2, J_1, J_2 are intervals and $\langle \overline{X} \rangle$ is an operator in the logic $\mathcal{F}$

$\mathcal{F}$-Bisimulation: invariance result

Theorem

Let M and M' be two interval models, $I \in \mathbb{I}(\mathbb{D})$ and $J \in \mathbb{J}(\mathbb{D}')$ two intervals. **If** there exists a $\mathcal{F}$-bisimulation B between M and M' with $(I, J) \in B$ **then** $M, I \equiv M', J$

$\mathcal{F}$-Bisimulation: invariance result

Theorem

Let M and M' be two interval models, $I \in \mathbb{I}(\mathbb{D})$ and $J \in \mathbb{J}(\mathbb{D}')$ two intervals. **If** there exists a $\mathcal{F}$-bisimulation B between M and M' with $(I, J) \in B$ **then** $M, I \equiv M', J$

- In other words, any $\mathcal{F}$-Bisimulation preserves the truth of all formulas in $\mathcal{F}$
- Sketch proof. Consider the B such that $(I, J) \in B$. Suppose that $M, I \models \phi$, show that $M', J \models \phi$ by structural induction on ϕ. Propositional cases are straightforward. Modal cases are dealt by using the forward condition of bisimulation
- The theorem is an instantiation of a general result on modal logics [BRV01]
- In general the converse of the theorem does not hold

Expressiveness results in $\mathcal{HS}$ fragments

$\mathcal{F}_1$ is *strictly less expressive than* $\mathcal{F}_2$ ($\mathcal{F}_1 \prec \mathcal{F}_2$) if all operators $\langle X \rangle$ in $\mathcal{F}_1$ are definable in $\mathcal{F}_2$ but not viceversa

Expressiveness results in $\mathcal{HS}$ fragments

$\mathcal{F}_1$ is *strictly less expressive than* $\mathcal{F}_2$ ($\mathcal{F}_1 \prec \mathcal{F}_2$) if all operators $\langle X \rangle$ in $\mathcal{F}_1$ are definable in $\mathcal{F}_2$ but not viceversa

To prove $\langle X \rangle$ is not definable in $\mathcal{F}$

- Find models M, M' and a $\mathcal{F}$-bisimulation B
- Find $\mathcal{F}$-bisimilar intervals I and J and...
- Show that $M, I \models \langle X \rangle p$ and $M', J \not\models \langle X \rangle p$

Expressiveness results in $\mathcal{HS}$ fragments

$\mathcal{F}_1$ is *strictly less expressive than* $\mathcal{F}_2$ ($\mathcal{F}_1 \prec \mathcal{F}_2$) if all operators $\langle X \rangle$ in $\mathcal{F}_1$ are definable in $\mathcal{F}_2$ but not viceversa

To prove $\langle X \rangle$ is not definable in $\mathcal{F}$

- Find models M, M' and a $\mathcal{F}$-bisimulation B
- Find $\mathcal{F}$-bisimilar intervals I and J and...
- Show that $M, I \models \langle X \rangle p$ and $M', J \not\models \langle X \rangle p$

Notice the similarity between the use of EF Games in FO and LTL theorem seen before

Past operators in A: the $\langle \overline{L} \rangle$ modality

Theorem

The modality $\langle \overline{L} \rangle$ is not definable in A over $\mathbb{N}$

Past operators in A: the $\langle \overline{L} \rangle$ modality

Theorem

The modality $\langle \overline{L} \rangle$ is not definable in A over $\mathbb{N}$

Proof.

- Consider $M = (\mathbb{I}(\mathbb{N}), V)$ and $M' = (\mathbb{I}(\mathbb{N}), V)$
- $AP = \{p\}$ and $V(p) = \{[i, i+1] : i \geq 0\}$

Past operators in A: the $\langle \overline{L} \rangle$ modality

Theorem

The modality $\langle \overline{L} \rangle$ is not definable in A over $\mathbb{N}$

Proof.

- Consider $M = (\mathbb{I}(\mathbb{N}), V)$ and $M' = (\mathbb{I}(\mathbb{N}), V)$
- $AP = \{p\}$ and $V(p) = \{[i, i+1] : i \geq 0\}$
- A-Bisimulation
  $B = \{([i,j],[i,j]) : 0 \leq i < j\} \cup \{([i,j],[i+1,j+1]) : 0 \leq i < j\}$
- Intervals $[1,2]$ and $[2,3]$ are A-bisimilar but $M,[1,2] \not\models \langle \overline{L} \rangle p$ while $M',[2,3] \models \langle \overline{L} \rangle p$

$\square$

Past operators in A: the $\langle \bar{L} \rangle$ modality

Corollary

A is strictly less expressive than $A\bar{L}$. $\square$

Past operators in A: the $\langle \overline{L} \rangle$ modality

Corollary

A is strictly less expressive than $A\overline{L}$. □

Corollary

A is strictly less expressive than $A\overline{A}$.

Proof.

Note that the $\langle \overline{L} \rangle$ modality is definable in $A\overline{A}$.
It holds that $\langle \overline{L} \rangle \phi = \langle \overline{A} \rangle \langle \overline{A} \rangle \phi$ □

Past operators in A: the $\langle\overline{L}\rangle$ modality

Corollary

A is strictly less expressive than $A\overline{L}$. □

Corollary

A is strictly less expressive than $A\overline{A}$.

Proof.

Note that the $\langle\overline{L}\rangle$ modality is definable in $A\overline{A}$.
It holds that $\langle\overline{L}\rangle\phi = \langle\overline{A}\rangle\langle\overline{A}\rangle\phi$ □

Which type of relation between $A\overline{A}$ and $A\overline{L}$??

Past operators in $A\overline{L}$: the $\langle \overline{A} \rangle$ modality

Theorem

The modality $\langle \overline{A} \rangle$ is not definable in $A\overline{L}$ over $\mathbb{N}$

Past operators in $A\overline{L}$: the $\langle\overline{A}\rangle$ modality

Theorem

The modality $\langle\overline{A}\rangle$ is not definable in $A\overline{L}$ over $\mathbb{N}$

Proof.

- Consider $M = (\mathbb{I}(\mathbb{N}), V)$ and $M' = (\mathbb{I}(\mathbb{N}), V)$
- $AP = \{p\}$ and $V(p) = \mathbb{I}(\mathbb{N})$

Past operators in $A\overline{L}$: the $\langle\overline{A}\rangle$ modality

Theorem

The modality $\langle\overline{A}\rangle$ is not definable in $A\overline{L}$ over $\mathbb{N}$

Proof.

- Consider $M = (\mathbb{I}(\mathbb{N}), V)$ and $M' = (\mathbb{I}(\mathbb{N}), V)$
- $AP = \{p\}$ and $V(p) = \mathbb{I}(\mathbb{N})$
- $A\overline{L}$-Bisimulation $B = \{([i,j],[i,j]) : 0 \leq i < j\} \cup \{([0,2],[1,2])\}$
  $\rightarrow$ 2 types of forward and backward conditions need to be checked
- Intervals $[0,2]$ and $[1,2]$ are $A\overline{L}$-bisimilar but $M,[0,2] \not\models \langle\overline{A}\rangle$ *true* while $M',[1,2] \models \langle\overline{A}\rangle$ *true*

$\square$

Past operators in $A\overline{L}$: the $\langle \overline{A} \rangle$ modality

Corollary

$A \prec A\overline{L} \prec A\overline{A}$ $\square$

Past operators in $A\overline{L}$: the $\langle\overline{A}\rangle$ modality

Corollary

$A \prec A\overline{L} \prec A\overline{A}$ $\square$

- In this setting past modalities increase the expressive power (in contrast to LTL)
- As an example $A\overline{A}$ is able to separate $\mathbb{R}$ and $\mathbb{Q}$ while A is not
- The previous proofs can be generalized to work with $\mathbb{R}, \mathbb{Q}, \mathbb{Z}$ and finite prefixes of $\mathbb{N}$

Conclusions

Expressiveness in a picture

$$LTL \equiv LTL + P \equiv FO[<] \equiv \text{Counter-free automata}$$

Expressiveness in a picture

$$LTL \equiv LTL + P \equiv FO[<] \equiv \text{Counter-free automata}$$

$$\cap$$

$$ETL \equiv QLTL \equiv MSO[<] \equiv \text{Büchi automata}$$

Expressiveness in a picture

$$LTL \equiv LTL + P \equiv FO[<] \equiv \text{Counter-free automata}$$

$$\cap$$

$$ETL \equiv QLTL \equiv MSO[<] \equiv \text{Büchi automata}$$

$$A \prec A\overline{L} \prec A\overline{A}$$

Concluding remarks

- The concept of *even* is not *FO*-definable
- We can use bisimulation dealing with modal logic
- Adding past modalities may or may not increase the expressive power (but other properties like succinctness or complexity may change)
- Not always the increase in expressive power comes at the cost of an increase in complexity for the decision problem
  - *LTL* and *ETL* are both *PSPACE*-complete
  - $A, A\overline{L}$ and $A\overline{A}$ are *NEXPTIME*-complete

Future investigations

- The considered *LTL* theorem (slide 9) doesn't directly apply to $LTL{+}P$
  Consider $\psi_k = pU(\neg p \wedge XGp) \wedge F(\neg p \wedge Y^k true \wedge Z^{k+1} false)$
  - For a computation σ, $\sigma \models pU(\neg p \wedge XGp)$ iff $\sigma = p^i(\neg p)p^\omega$ for some i
  - $p^i(\neg p)p^\omega \models \psi_k$ iff $i = k$

  $\rightarrow$ We have $p^k(\neg p)p^\omega \models \psi_k$ but $p^{k+1}(\neg p)p^\omega \not\models \psi_k$ contradicting the LTL theorem

Future investigations

- The considered *LTL* theorem (slide 9) doesn't directly apply to *LTL+P*
  Consider $\psi_k = pU(\neg p \wedge XGp) \wedge F(\neg p \wedge Y^k true \wedge Z^{k+1} false)$
  - For a computation σ, $\sigma \models pU(\neg p \wedge XGp)$ iff $\sigma = p^i(\neg p)p^\omega$ for some i
  - $p^i(\neg p)p^\omega \models \psi_k$ iff $i = k$

  $\rightarrow$ We have $p^k(\neg p)p^\omega \models \psi_k$ but $p^{k+1}(\neg p)p^\omega \not\models \psi_k$ contradicting the LTL theorem
- How can we extend the *LTL* theorem to *LTL + P*?
  - If we take an algorithm to translate *LTL* formula into *LTL+P*, can we say something about the number of next operators?
  - Forgetting about translation algorithms, can we directly generalize the theorem taking into account the *LTL+P* semantics and some other properties, like the number of *yesterday* operators?

Bibliography I

[BRV01] P. Blackburn, M. d. Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001. DOI: 10.1017/CBO9781107050884.

[Del11] D. Della Monica. "Expressiveness, decidability, and undecidability of Interval Temporal Logic". PhD thesis. University of Udine, 2011.

[DG08] V. Diekert and P. Gastin. "First-order definable languages". In: Jan. 2008, pp. 261–306.

Bibliography II

[DMS12] D. Della Monica, A. Montanari, and P. Sala. "The Importance of the Past in Interval Temporal Logics: The Case of Propositional Neighborhood Logic". In: *Logic Programs, Norms and Action: Essays in Honor of Marek J. Sergot on the Occasion of His 60th Birthday*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 79–102. ISBN: 978-3-642-29414-3.

[Kam68] H. Kamp. "Tense Logic and the Theory of Linear Order". PhD thesis. Ucla, 1968.

[KPV01] O. Kupferman, N. Piterman, and M. Y. Vardi. "Extended Temporal Logic Revisited". In: *CONCUR 2001 — Concurrency Theory*. Ed. by K. G. Larsen and M. Nielsen. Springer Berlin Heidelberg, 2001, pp. 519–535. ISBN: 978-3-540-44685-9.

Bibliography III

[Stu00] H. Sturm. "Elementary Classes in Basic Modal Logic". In: *Studia Logica: An International Journal for Symbolic Logic* 64.2 (2000), pp. 193–213. (Visited on 06/04/2023).

[Wol83] P. Wolper. "Temporal logic can be more expressive". In: *Information and Control* 56.1 (1983), pp. 72–99. ISSN: 0019-9958.