Temporal Modalities in Answer Set Programming

Pedro Cabalar

University of Corunna (Spain)
cabalar@udc.es

September 23rd, 2020
TIME 2020, Bozen-Bolzano, Italy
Joint work with

Felicidad Aguado
U. Corunna

Gilberto Pérez
U. Corunna

Concepción Vidal
U. Corunna

Martín Diéguez
U. Angers

Torsten Schaub
U. Potsdam

Anna Schuhmann
U. Potsdam
Other collaborators:

- Jorge Fandinno, Roland Kaminski, François Laferriere, Philip Morkisch (U. Potsdam, DE)
- Philippe Balbiani, Luis Fariñas del Cerro (IRIT, Toulouse, FR)
- David Pearce (U. P. Madrid, ES)
- Stephane Demri (CNRS, ENS Paris-Saclay, FR)
- Laura Bozzelli (U. Naples, IT)
1 Introduction

2 Definitions and examples

3 Automata-based computation

4 Temporal Logic Programming

5 Conclusions and open topics
Motivation

- Context: temporal reasoning with transition systems.
- Typical reasoning problems: simulation, explanation, planning, diagnosis, verification.
Motivation

- Context: temporal reasoning with transition systems.
- Typical reasoning problems:
  simulation, explanation, planning, diagnosis, verification.
- ... but focusing on Knowledge Representation (KR)
Motivation

- Context: temporal reasoning with transition systems.

- Typical reasoning problems: simulation, explanation, planning, diagnosis, verification.

- ... but focusing on Knowledge Representation (KR)
Motivation

- Context: temporal reasoning with transition systems.
- Typical reasoning problems: simulation, explanation, planning, diagnosis, verification.
- ... but focusing on Knowledge Representation (KR)
Which are the desirable properties of a good KR?

1. Simplicity

2. Natural understanding: clear semantics

3. Allows automated reasoning methods that:
   - are efficient
   - or at least, their complexity can be assessed

Elaboration tolerance [McCarthy98]

Small changes in the problem ⇒ small changes in specification

Typical problems of lack of elaboration:
frame, ramification, qualification

Example: automata satisfy everything, but lack elaboration

Pedro Cabalar (University of Corunna (Spain) cabalar@udc.es)
Which are the desirable properties of a good KR?

1. Simplicity

2. Natural understanding: clear semantics

3. Allows automated reasoning methods that:
   - are efficient
   - or at least, their complexity can be assessed

4. Elaboration tolerance [McCarthy98]

Small changes in the problem ⇒ small changes in specification
Which are the desirable properties of a good KR?

1. Simplicity

2. Natural understanding: clear semantics

3. Allows automated reasoning methods that:
   ▶ are efficient
   ▶ or at least, their complexity can be assessed

4. Elaboration tolerance [McCarthy98]
   Small changes in the problem ⇒ small changes in specification

Typical problems of lack of elaboration:
frame, ramification, qualification
Which are the desirable properties of a good KR?

1. Simplicity
2. Natural understanding: clear semantics
3. Allows automated reasoning methods that:
   - are efficient
   - or at least, their complexity can be assessed
4. Elaboration tolerance [McCarthy98]
   Small changes in the problem $\Rightarrow$ small changes in specification

Typical problems of lack of elaboration: frame, ramification, qualification

Example: automata satisfy everything, but lack elaboration
Elaboration tolerance for action domains:

*Representing Action and Change by Logic Programs* [Gelfond & Lifschitz 93] use Answer Set Programming (ASP)
Temporal ASP

Elaboration tolerance for action domains:

*Representing Action and Change by Logic Programs*

[Gelfond & Lifschitz 93] use Answer Set Programming (ASP)

However, ASP has no temporal constructs
Example

- Initially, a lamp switch can be *up* or *down*.

```prolog
\text{up}(0); \text{down}(0).
```

```
time(0..n).
up(T+1) :- up(T), not down(T+1), time(T).
down(T+1) :- down(T), not up(T+1), time(T).
\{up(T)\} :- time(T).
```
Example

- Initially, a lamp switch can be *up* or *down*.
- By default, the switch state persists by inertia,

```
time(0..n).
up(0); down(0).
up(T+1) :- up(T), not down(T+1), time(T).
down(T+1) :- down(T), not up(T+1), time(T).
```
Example

- Initially, a lamp switch can be *up* or *down*.
- By default, the switch state persists by inertia, but we can arbitrarily close it at any moment.

```
time(0..n).
up(0); down(0).
up(T+1) :- up(T), not down(T+1), time(T).
down(T+1) :- down(T), not up(T+1), time(T).
{up(T)} :- time(T).
```
Examples of problems that cannot be solved in ASP:

- Is there a reachable state with *up* and *down* false?
Examples of problems that cannot be solved in ASP:

- Is there a reachable state with $up$ and $down$ false?
- Once $up$ becomes true, does it remain so forever?
Examples of problems that cannot be solved in ASP:

- Is there a reachable state with *up* and *down* false?
- Once *up* becomes true, does it remain so forever?
- The switch cannot be closed infinitely often without eventually damaging the lamp
Modal Temporal Logic

These topics typically covered by (Modal) Temporal Logics
Modal Temporal Logic

These topics typically covered by (Modal) Temporal Logics

A simple and well-known case

**Linear-time Temporal Logic (LTL)**

\(\square\) (forever), \(\Diamond\) (eventually), \(\circ\) (next), \(U\) (until)
Modal Temporal Logic

These topics typically covered by (Modal) Temporal Logics

A simple and well-known case

Linear-time Temporal Logic (LTL)

- □ (forever), ♦ (eventually), ○ (next), U (until)

✓ Decidable inference methods. Satisfiability: PSPACE-complete
Modal Temporal Logic

These topics typically covered by (Modal) Temporal Logics

A simple and well-known case

**Linear-time Temporal Logic (LTL)**

- $\square$ (forever), $\diamond$ (eventually), $\circ$ (next), $U$ (until)

✓ Decidable inference methods. Satisfiability: PSPACE-complete

✓ Relation to other mathematical models:
  algebra, automata, formal languages
Modal Temporal Logic

These topics typically covered by (Modal) Temporal Logics

A simple and well-known case

**Linear-time Temporal Logic (LTL)**

\(\square\) (forever), \(\Diamond\) (eventually), \(\circ\) (next), \(\mathbf{U}\) (until)

✓ Decidable inference methods. Satisfiability: PSPACE-complete

✓ Relation to other mathematical models:
   algebra, automata, formal languages

✓ Fragment of First-Order Logic: [Kamp 68] LTL = Monadic FO (<)
Modal Temporal Logic

These topics typically covered by (Modal) Temporal Logics

A simple and well-known case

Linear-time Temporal Logic (LTL)

- □ (forever), ◊ (eventually), ○ (next), U (until)

- Decidable inference methods. Satisfiability: PSPACE-complete

- Relation to other mathematical models: algebra, automata, formal languages

- Fragment of First-Order Logic: [Kamp 68] LTL = Monadic FO (≤)

- Model checking and verification of reactive systems
Modal Temporal Logic

These topics typically covered by (Modal) Temporal Logics

A simple and well-known case

**Linear-time Temporal Logic (LTL)**

\(\square\) (forever), \(\Diamond\) (eventually), \(\circ\) (next), \(\mathbf{U}\) (until)

✓ Decidable inference methods. Satisfiability: PSPACE-complete

✓ Relation to other mathematical models:
  algebra, automata, formal languages

✓ Fragment of First-Order Logic: [Kamp 68] LTL = Monadic FO (<)

✓ Model checking and verification of reactive systems

✓ Many uses in AI: planning, ontologies, multi-agent systems, . . .
Modal Temporal Logic

A simple and well-known example

Linear-time Temporal Logic (LTL)

\( \square, \diamond, \circ, \mathbf{U} \ldots \)

\( \times \) Monotonic: action domain representations manifest frame problem
Modal Temporal Logic

A simple and well-known example

**Linear-time Temporal Logic (LTL)**

\[ \square, \Diamond, \circ, \mathbf{U} \ldots \]

**\( \times \) Monotonic:** action domain representations manifest **frame problem**

In model checking no worry on this:
usually, logical description of **automaton states**
A simple and well-known example

**Linear-time Temporal Logic (LTL)**

\( \Box, \Diamond, \circ, U \ldots \)

*Monotonic*: action domain representations manifest frame problem

In model checking no worry on this:
usually, logical description of **automaton states**
even worse! nothing less elaboration tolerant than an automaton
Modal Temporal Logic

A simple and well-known example

Linear-time Temporal Logic (LTL)

\[ \Box, \Diamond, \lozenge, \mathbf{U} \ldots \]

\[\times\] Monotonic: action domain representations manifest frame problem

In model checking no worry on this:
usually, logical description of automaton states
even worse! nothing less elaboration tolerant than an automaton

\[\times\] NMR attempts for LTL: limited syntax, only for queries, control
rules, etc. Not really embodied in LTL
Our proposal

Temporal Equilibrium Logic (TEL) [C_&Pérez 07]

TEL = ASP + LTL

- **ASP**: logical characterisation Equilibrium Logic [Pearce 96]
- **LTL**: We add temporal operators $\Box$, $\Diamond$, $\circ$, $U$, $R$.

Result: Temporal Stable Models for any arbitrary LTL theory.
Initially, a lamp switch can be closed \((p)\) or open \((q)\).

By default, the switch state persists by inertia, but we can arbitrarily close it at any moment.

- \(\text{time}(0..n)\).
- \(\text{up}(0), \text{down}(0)\).
- \(\text{up}(T+1) :- \text{up}(T), \neg \text{down}(T+1), \text{time}(T)\).
- \(\text{down}(T+1) :- \text{down}(T), \neg \text{up}(T+1), \text{time}(T)\).
- \(\{\text{up}(T)\} :- \text{time}(T)\).
Initially, a lamp switch can be closed ($p$) or open ($q$).

By default, the switch state persists by inertia, but we can arbitrarily close it at any moment.

- $up \lor down$
- Initially:
  - $\square(\Diamond up \leftarrow up \land \neg \Diamond down)$
  - $\square(\Diamond down \leftarrow down \land \neg \Diamond up)$
- Inertia
- Choice:
  - $\square(up \lor \neg up)$

Idea: LTL syntax, but keeping ASP semantics.
Introduction

Definitions and examples

Automata-based computation

Temporal Logic Programming

Conclusions and open topics
Equilibrium Logic \cite{Pearce96}: generalises stable models for arbitrary propositional theories.
Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

1. A non-classical monotonic (intermediate) logic called Here-and-There (HT)
Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

1. A non-classical monotonic (intermediate) logic called Here-and-There (HT)

\[
\text{HT models} \quad \quad \quad \quad \quad \text{Classical models}
\]

2. A selection of (certain) minimal models that yields nonmonotonicity
Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

1. A non-classical monotonic (intermediate) logic called Here-and-There (HT)

2. A selection of (certain) minimal models that yields nonmonotonicity
Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

1. A non-classical monotonic (intermediate) logic called Here-and-There (HT)

2. A selection of (certain) minimal models that yields nonmonotonicity
Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

1. A non-classical monotonic (intermediate) logic called Here-and-There (HT)

2. A selection of (certain) minimal models that yields nonmonotonicity
Here-and-There

- Interpretation = pairs $\langle H, T \rangle$ of sets of atoms $H \subseteq T$

Example: $H = \{p, q\}$, $T = \{p, q, r, s\}$.

Intuition:

- There = perhaps true
- Here = proved
- Not there = false

When $H = T$ we have a classical model.

Pedro Cabalar (University of Corunna (Spain) cabalar@udc.es)

Temporal Modalities in ASP

TIME 2020 16 / 58
Here-and-There

- Interpretation = pairs $\langle H, T \rangle$ of sets of atoms $H \subseteq T$
- Example: $H = \{p, q\}$, $T = \{p, q, r, s\}$. 

Intuition:
- There = perhaps true
- Here = proved
- Not there = false

When $H = T$ we have a classical model.
Interpretation = pairs \( \langle H, T \rangle \) of sets of atoms \( H \subseteq T \)

Example: \( H = \{ p, q \}, T = \{ p, q, r, s \} \). Intuition:

\[
\begin{align*}
\text{Not there} &= \text{false} \\
\text{Here} &= \text{proved} \\
\text{There} &= \text{perhaps true}
\end{align*}
\]

When \( H = T \) we have a classical model.
**Here-and-There**

**Satisfaction of formulas**

\[ \langle H, T \rangle \models \varphi \iff \text{“} \varphi \text{ is proved”} \]
Here-and-There

Satisfaction of formulas

\[
\langle H, T \rangle \models \varphi \iff \text{“} \varphi \text{ is proved”}
\]

\[
\langle T, T \rangle \models \varphi \iff \text{“} \varphi \text{ potentially true”} \iff T \models \varphi \text{ classically}
\]
Here-and-There

Satisfaction of formulas

\[ \langle H, T \rangle \models \varphi \iff \text{“\varphi is proved”} \]
\[ \langle T, T \rangle \models \varphi \iff \text{“\varphi potentially true”} \iff T \models \varphi \text{ classically} \]

\[ \langle H, T \rangle \models p \text{ if } p \in H \text{ (for any atom } p) \]
Satisfaction of formulas

\[ \langle H, T \rangle \models \varphi \iff \text{“} \varphi \text{ is proved”} \]
\[ \langle T, T \rangle \models \varphi \iff \text{“} \varphi \text{ potentially true”} \iff T \models \varphi \text{ classically} \]

- \[ \langle H, T \rangle \models p \text{ if } p \in H \] (for any atom \( p \))

- \( \land, \lor \) as always
Here-and-There

Satisfaction of formulas

\[ \langle H, T \rangle \models \varphi \iff \text{"\varphi is proved"} \]
\[ \langle T, T \rangle \models \varphi \iff \text{"\varphi potentially true"} \iff T \models \varphi \text{ classically} \]

- \[ \langle H, T \rangle \models p \text{ if } p \in H \] (for any atom \( p \))
- \( \land, \lor \) as always
- \[ \langle H, T \rangle \models \varphi \rightarrow \psi \text{ if both} \]
  - \( T \models \varphi \rightarrow \psi \text{ classically} \)
  - \[ \langle H, T \rangle \models \varphi \text{ implies } \langle H, T \rangle \models \psi \]
Here-and-There

Satisfaction of formulas

\[ \langle H, T \rangle \models \varphi \iff \text{“} \varphi \text{ is proved”} \]
\[ \langle T, T \rangle \models \varphi \iff \text{“} \varphi \text{ potentially true”} \iff T \models \varphi \text{ classically} \]

- \( \langle H, T \rangle \models p \) if \( p \in H \) (for any atom \( p \))

- \( \land, \lor \) as always

- \( \langle H, T \rangle \models \varphi \rightarrow \psi \) if both
  - \( T \models \varphi \rightarrow \psi \) classically
  - \( \langle H, T \rangle \models \varphi \) implies \( \langle H, T \rangle \models \psi \)

- Negation \( \neg F \) is defined as \( F \rightarrow \bot \)

- \( \langle H, T \rangle \models \varphi \) implies \( T \models \varphi \) (proved implies potentially true)
Equilibrium models

Definition (Equilibrium/stable model)

A model \( \langle T, T \rangle \) of \( \Gamma \) is an equilibrium model iff

\[
\text{there is no } H \subset T \text{ such that } \langle H, T \rangle \models \Gamma.
\]

When this holds, \( T \) is called a stable model.
A model $\langle T, T \rangle$ of $\Gamma$ is an equilibrium model iff

there is no $H \subset T$ such that $\langle H, T \rangle \models \Gamma$.

When this holds, $T$ is called a stable model.

In other words, all assumptions $T$ are eventually proved $H$.
(Linear) Temporal Equilibrium Logic

**Syntax** = propositional plus

- □φ = “forever” φ
- ◊φ = “eventually” φ
- ◯φ = “next moment” φ
- φ U ψ = φ “until eventually” ψ
- φ R ψ = φ “release” ψ
- φ W ψ = φ “while” ψ

As we had with Equilibrium Logic:

1. A monotonic underlying logic: Temporal Here-and-There (THT)
2. An ordering among models. Select minimal models.
(Linear) Temporal Equilibrium Logic

- **Syntax** = propositional plus
  - □φ = “forever” φ
  - ◊φ = “eventually” φ
  - ◯φ = “next moment” φ
  - φ U ψ = φ “until eventually” ψ
  - φ R ψ = φ “release” ψ
  - φ W ψ = φ “while” ψ

- As we had with Equilibrium Logic:
  1. A monotonic underlying logic: Temporal Here-and-There (THT)
  2. An ordering among models. Select minimal models.
In standard LTL, interpretations are $\infty$ sequences of sets of atoms:

\[
\begin{array}{cccccc}
\{p, q\} & \{p\} & \{q\} & \{\}\{p, q\} & \ldots \\
0 & 1 & 2 & 3 & 4 
\end{array}
\]
In standard LTL, interpretations are $\infty$ sequences of sets of atoms

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

In THT we will have $\infty$ sequences of HT interpretations

\[
\begin{array}{cccccc}
\quad & \quad & \quad & \quad & \quad & \ldots \\
0 & 1 & 2 & 3 & 4 \\
\end{array}
\]
We define an ordering among sequences $H \leq T$ when

$$
T_0 \rightarrow T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_i \rightarrow \ldots
$$

$$
U \mid U \mid U \mid U \mid
$$

$$
H_0 \rightarrow H_1 \rightarrow H_2 \rightarrow \ldots \rightarrow H_i \rightarrow \ldots
$$
We define an ordering among sequences $H \leq T$ when

$$
T_0 \rightarrow T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_i \rightarrow \ldots \\
U \upharpoonright \ldots U \upharpoonright \ldots U \upharpoonright \ldots U \upharpoonright

H_0 \rightarrow H_1 \rightarrow H_2 \rightarrow \ldots \rightarrow H_i \rightarrow \ldots
$$

**Definition (THT-interpretation)**
is a pair of sequences of sets of atoms $\langle H, T \rangle$ with $H \leq T$.  

Sequences

We define an ordering among sequences $H < T$ when

\[
\begin{align*}
&T_0 \rightarrow T_1 \rightarrow T_2 \rightarrow \ldots \rightarrow T_i \rightarrow \ldots \\
&\mathcal{U} \mid \mathcal{U} \mid \mathcal{U} \mid \mathcal{U} \\
&H_0 \rightarrow H_1 \rightarrow H_2 \rightarrow \ldots \rightarrow H_i \rightarrow \ldots 
\end{align*}
\]

Definition (THT-interpretation)
is a pair of sequences of sets of atoms $\langle H, T \rangle$ with $H \leq T$. 

Pedro Cabalar (University of Corunna (Spain)
cabalar@udc.es

Temporal Modalities in ASP

TIME 2020
Temporal Here-and-There (THT)

\[ \langle H, T \rangle, i \models \varphi \iff \text{“} \varphi \text{ is proved at } i \text{“} \]
Temporal Here-and-There (THT)

\[ \langle H, T \rangle, i \models \varphi \iff \text{“\varphi is proved at } i\text{”} \]
\[ \langle T, T \rangle, i \models \varphi \iff \text{“\varphi potentially true at } i\text{”} \iff \mathbf{T}, i \models \varphi \text{ in LTL} \]
Temporal Here-and-There (THT)

$\langle H, T \rangle, i \models \varphi \iff \text{“} \varphi \text{ is proved at } i \text{“}$

$\langle T, T \rangle, i \models \varphi \iff \text{“} \varphi \text{ potentially true at } i \text{“} \iff T, i \models \varphi$ in LTL

- An interpretation $M = \langle H, T \rangle$ satisfies $\alpha$ at situation $i$, written $M, i \models \alpha$

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$M, i \models \alpha$ when ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>an atom $p$</td>
<td>$p \in H_i$</td>
</tr>
<tr>
<td>$\land, \lor$</td>
<td>as usual</td>
</tr>
<tr>
<td>$\varphi \rightarrow \psi$</td>
<td>$T, i \models \varphi \rightarrow \psi$ in LTL and $\langle H, T \rangle, i \models \varphi$ implies $\langle H, T \rangle, i \models \psi$</td>
</tr>
</tbody>
</table>
Temporal Here-and-There (THT)

\[ \langle H, T \rangle, i \models \varphi \iff \text{“} \varphi \text{ is proved at } i \text{”} \]
\[ \langle T, T \rangle, i \models \varphi \iff \text{“} \varphi \text{ potentially true at } i \text{”} \iff T, i \models \varphi \text{ in LTL} \]

- An interpretation \( M = \langle H, T \rangle \) satisfies \( \alpha \) at situation \( i \), written \( M, i \models \alpha \)

<table>
<thead>
<tr>
<th>( \alpha )</th>
<th>( M, i \models \alpha ) when . . .</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Diamond \varphi )</td>
<td>( (M, i+1) \models \varphi )</td>
</tr>
<tr>
<td>( \Box \varphi )</td>
<td>( \forall j \geq i, \ M, j \models \varphi )</td>
</tr>
<tr>
<td>( \Diamond \varphi )</td>
<td>( \exists j \geq i, \ M, j \models \varphi )</td>
</tr>
<tr>
<td>( \varphi \ U \psi )</td>
<td>( \exists j \geq i, \ M, j \models \psi ) and ( \forall k \text{ s.t. } i \leq k &lt; j, \ M, k \models \varphi )</td>
</tr>
<tr>
<td>( \varphi \ R \psi )</td>
<td>( \forall j \geq i, \ M, j \models \psi ) or ( \exists k, i \leq k &lt; j, \ M, k \models \varphi )</td>
</tr>
</tbody>
</table>

- \( M \) is a model of a theory \( \Gamma \) when \( M, 0 \models \alpha \) for all \( \alpha \in \Gamma \)
(Linear) Temporal Equilibrium Logic

\[ \circ \varphi \]\n
\[ \varphi \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \ldots \]
(Linear) Temporal Equilibrium Logic

- $\diamond \varphi$

- $\square \varphi$
(Linear) Temporal Equilibrium Logic

\[ \diamond \varphi \]

\[ \square \varphi \]

\[ \lozenge \varphi \]
(Linear) Temporal Equilibrium Logic

\[ \varphi \ U \psi = \text{repeat } \varphi \text{ until (mandatorily) } \psi \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
\( \varphi \ U \psi = \text{repeat } \varphi \text{ until (mandatorily) } \psi \)
$(\text{Linear})$ Temporal Equilibrium Logic

$$\varphi \mathbf{U} \psi = \text{repeat } \varphi \text{ until (mandatorily) } \psi$$

\[ \varphi \quad \varphi \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
(Linear) Temporal Equilibrium Logic

\[ \varphi \ U \psi = \text{repeat } \varphi \text{ until (mandatorily) } \psi \]

\[ \varphi \quad \varphi \quad \varphi \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
(Linear) Temporal Equilibrium Logic

\[ \varphi \mathbin{U} \psi = \text{repeat } \varphi \text{ until (mandatorily) } \psi \]

\[ \varphi \quad \varphi \quad \varphi \quad \varphi \]

\[ \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \ldots \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \ldots \]
\( \varphi \ U \psi = \text{repeat } \varphi \text{ until (mandatorily) } \psi \)
$(\text{Linear})$ Temporal Equilibrium Logic

$\varphi \mathbb{R} \psi = \text{disjunction of two cases}$

$\psi \mathbb{U} (\psi \land \varphi)$

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
\( \varphi \ R \ \psi = \text{disjunction of two cases} \)

\[ \psi \ U (\psi \land \varphi) \]

\[ \psi \]

\[ \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \ldots \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \ldots \]
(Linear) Temporal Equilibrium Logic

\( \varphi \ R \ \psi = \text{disjunction of two cases} \)

\[ \psi \ U \ (\psi \land \varphi) \]

\( \psi \quad \psi \)

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
\( \varphi \mathcal{R} \psi = \text{disjunction of two cases} \)

\[ \psi \mathcal{U} (\psi \land \varphi) \]

\[ \psi \quad \psi \quad \psi \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
$(\text{Linear})$ Temporal Equilibrium Logic

$\varphi \ R \ \psi = \text{disjunction of two cases}$

$\psi \ U \ (\psi \land \varphi)$

\[
\begin{array}{cccc}
\psi & \psi & \psi & \psi \\
\bullet & \rightarrow & \bullet & \rightarrow \\
\end{array}
\]

$\psi \land \varphi$

\[
\begin{array}{cccc}
\psi & \psi & \psi & \psi \\
\bullet & \rightarrow & \bullet & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
\psi & \psi & \psi & \psi \\
\bullet & \rightarrow & \bullet & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
\psi & \psi & \psi & \psi \\
\bullet & \rightarrow & \bullet & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
\psi & \psi & \psi & \psi \\
\bullet & \rightarrow & \bullet & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
\psi & \psi & \psi & \psi \\
\bullet & \rightarrow & \bullet & \rightarrow \\
\end{array}
\]

\[
\begin{array}{cccc}
\psi & \psi & \psi & \psi \\
\bullet & \rightarrow & \bullet & \rightarrow \\
\end{array}
\]

...
(Linear) Temporal Equilibrium Logic

\( \varphi \mathrel{R} \psi = \text{disjunction of two cases} \)

- \( \psi \mathrel{U} (\psi \land \varphi) \)

[Diagram of temporal sequence with \( \varphi \) preceding \( \psi \) and \( \psi \land \varphi \) following]
(Linear) Temporal Equilibrium Logic

ϕ \text{ R } ψ = \text{ disjunction of two cases}

ψ \text{ U } (ψ \land ϕ)

ψ \land ϕ

□ψ
(Linear) Temporal Equilibrium Logic

$ϕ \text{ W } ψ = \text{ do } ϕ \text{ while } ψ$

$ϕ$

$\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots$
(Linear) Temporal Equilibrium Logic

$$\varphi \mathbf{W} \psi = \text{do } \varphi \text{ while } \psi$$

$$\psi?$$

\[
\begin{array}{c}
\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots
\end{array}
\]
\( W \psi = \text{do } \varphi \text{ while } \psi \)

\[ \psi ? \]

\[ \varphi \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
(Linear) Temporal Equilibrium Logic

$\varphi \mathbf{W} \psi = \text{do } \varphi \text{ while } \psi$

\[ \psi? \quad \psi? \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
\( \varphi \mathcal{W} \psi = \text{do } \varphi \text{ while } \psi \)

\[ \psi \rightarrow \psi \rightarrow \varphi \rightarrow \ldots \]
(Linear) Temporal Equilibrium Logic

\[ \varphi \text{ W } \psi = \text{do } \varphi \text{ while } \psi \]

\[ \psi? \quad \psi? \quad \psi? \]

\[ \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
(Linear) Temporal Equilibrium Logic

$\varphi \xrightarrow{W} \psi = \text{do } \varphi \text{ while } \psi$

$\psi ? \psi ? \psi ?$

$\varphi$

$\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \cdots$
Some valid THT formulas:

\[ \Diamond \varphi \leftrightarrow \top \mathbf{U} \varphi \]
\[ \Box \varphi \leftrightarrow \bot \mathbf{R} \varphi \]
\[ \circ (\varphi \otimes \psi) \leftrightarrow \circ \varphi \otimes \circ \psi \]
\[ \varphi \mathbf{U} \psi \leftrightarrow \psi \lor (\varphi \land \circ (\varphi \mathbf{U} \psi)) \]
\[ \varphi \mathbf{R} \psi \leftrightarrow \psi \land (\varphi \lor \circ (\varphi \mathbf{R} \psi)) \]
\[ \varphi \mathbf{W} \psi \leftrightarrow \varphi \land (\psi \rightarrow \circ (\varphi \mathbf{W} \psi)) \]
\[ \neg (\varphi \mathbf{U} \psi) \leftrightarrow \neg \varphi \mathbf{R} \neg \psi \]
\[ \circ \neg \varphi \leftrightarrow \neg \circ \varphi \]
\[ \neg (\varphi \mathbf{R} \psi) \leftrightarrow \neg \varphi \mathbf{U} \neg \psi \]

For \( \otimes = \land, \lor, \rightarrow, \mathbf{U}, \mathbf{R} \).
Temporal Here-and-There (THT)

- Some valid THT formulas:

  \[ \Diamond \phi \leftrightarrow \top \land \phi \]
  \[ \Box \phi \leftrightarrow \bot \lor \phi \]
  \[ \circ \left( \phi \land \psi \right) \leftrightarrow \circ \phi \land \circ \psi \]
  \[ \phi \lor \psi \leftrightarrow \psi \lor \left( \phi \land \circ \left( \phi \lor \psi \right) \right) \]
  \[ \phi \land \psi \leftrightarrow \psi \land \left( \phi \lor \circ \left( \phi \land \psi \right) \right) \]
  \[ \neg \left( \phi \lor \psi \right) \leftrightarrow \neg \phi \land \neg \psi \]
  \[ \circ \neg \phi \leftrightarrow \neg \circ \phi \]
  \[ \neg \left( \phi \land \psi \right) \leftrightarrow \neg \phi \lor \neg \psi \]

- For \( \otimes = \land, \lor, \rightarrow, \lnot, \top, \bot \).

- Axiomatization of THT [Balbiani & Diéguez 16]
Definition (Temporal Equilibrium Model)

of a theory $\Gamma$ is a model $M = \langle T, T \rangle$ of $\Gamma$ such that there is no $H < T$ satisfying $\langle H, T \rangle, 0 \models \Gamma$. 

Temporal Equilibrium Logic (TEL) is the logic induced by temporal equilibrium models.
Definition (Temporal Equilibrium Model)

of a theory $\Gamma$ is a model $M = \langle T, T \rangle$ of $\Gamma$ such that there is no $H < T$ satisfying $\langle H, T \rangle, 0 \models \Gamma$.

Temporal Equilibrium Logic (TEL) is the logic induced by temporal equilibrium models.

Definition (Temporal Stable Model)

$T$ is a temporal stable model of a theory $\Gamma$ iff $\langle T, T \rangle$ is a temporal equilibrium model of $\Gamma$. 
Some examples

Example 1: TEL models of $\square(\neg p \rightarrow \circ p)$. It’s like an infinite program:

$\neg p \rightarrow \circ p$

$\neg \circ p \rightarrow \circ^2 p$

$\neg \circ^2 p \rightarrow \circ^3 p$

$\vdots$
Some examples

- Example 1: TEL models of $\square(\neg p \rightarrow o p)$. It’s like an infinite program:

$$\neg p \rightarrow o p$$
$$\neg o p \rightarrow o^2 p$$
$$\neg o^2 p \rightarrow o^3 p$$
$$...$$

- TEL models have the form

$$\emptyset \quad p \quad \emptyset \quad p \quad \emptyset$$

$$\bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow ...$$

corresponding to LTL models of $\neg p \land \square(\neg p \leftrightarrow o p)$. 

Pedro Cabalar (University of Corunna (Spain) cabalar@udc.es)
Some examples

- Example 2: consider TEL models of $\lozenge p$
Example 2: consider TEL models of $\lozenge p$

is like $p \lor \lozenge p \lor \lozenge \lozenge p \lor \ldots$
Some examples

Example 2: consider TEL models of $\Diamond p$

is like $p \lor \Diamond p \lor \Diamond \Diamond p \lor \ldots$

TEL models have the form

$\emptyset \quad \emptyset \quad \emptyset \quad p \quad \emptyset$

$\bullet \longrightarrow \bullet \longrightarrow \ldots \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \bullet \longrightarrow \ldots$
Example 2: consider TEL models of $\Diamond p$

is like $p \lor \lozenge p \lor \lozenge \lozenge p \lor \ldots$

TEL models have the form

$$
\emptyset \quad \emptyset \quad \emptyset \quad p \quad \emptyset
$$

$$
\bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots
$$

corresponding to LTL models of $\neg p \mathbf{U} (p \land \lozenge \Box \neg p)$
In ASP terms, how can we represent temporal stable models? 

infinitely long! infinitely many!

Answer: using Büchi automata. An infinite-length word is accepted iff it visits some acceptance state infinitely often.
In ASP terms, how can we represent temporal stable models? infinitely long! infinitely many!

Answer: using Büchi automata. An infinite-length word is accepted iff it visits some acceptance state infinitely often.
Some examples

- Example 3: consider TEL models of $\Box\Diamond p$
- In LTL this means $p$ occurs infinitely often.
Example 3: consider TEL models of $\lozenge\Diamond p$
- In LTL this means $p$ occurs infinitely often.
- So take any LTL model $T$ like that, i.e., $\langle T, T \rangle$ is a total THT model.
Some examples

- Example 3: consider TEL models of $\square \diamond p$
- In LTL this means $p$ occurs infinitely often.
- So take any LTL model $T$ like that, i.e., $\langle T, T \rangle$ is a total THT model.
- Now build some $H < T$ by removing one $p$ at some point. But then $\langle H, T \rangle$ is also a model since $H$ contains $\infty - 1 = \infty$ $p$'s yet!
Some examples

- Example 3: consider TEL models of $\Box \Diamond p$
- In LTL this means $p$ occurs infinitely often.
- So take any LTL model $T$ like that, i.e., $\langle T, T \rangle$ is a total THT model.
- Now build some $H < T$ by removing one $p$ at some point. But then $\langle H, T \rangle$ is also a model since $H$ contains $\infty - 1 = \infty$ $p$'s yet!
- Therefore, $\Box \Diamond p$ alone has no TEL models.
Some examples

- We can still express \textit{infinitely often} by disabling minimality of $p$
Some examples

- We can still express infinitely often by disabling minimality of $p$
- This can be done adding the (excluded middle) axiom

$$\square(p \lor \neg p) \quad (EM)$$

(a choice rule in ASP)
Some examples

- We can still express infinitely often by disabling minimality of \( p \)
- This can be done adding the (excluded middle) axiom
  \[ \square(p \lor \neg p) \] (EM)
  (a choice rule in ASP)
- In fact, if we add (EM) for all atoms, TEL collapses into LTL
Some examples

Example 4: consider TEL models of the pair of formulas

$$\square(\neg \diamond p \rightarrow p)$$
$$\square(\diamond p \rightarrow p)$$
Example 4: consider TEL models of the pair of formulas

$$\square (\neg \Diamond p \rightarrow p)$$

$$\square (\Diamond p \rightarrow p)$$

Curiosity: implications go backwards in time
Some examples

- Example 4: consider TEL models of the pair of formulas

\[ \Box (\neg \Diamond p \rightarrow p) \]
\[ \Box (\Diamond p \rightarrow p) \]

- Curiosity: implications go \textit{backwards in time}
- This is LTL-equivalent to:

\[ \Box ((\neg \Diamond p \rightarrow p) \land (\Diamond p \rightarrow p)) \]
Some examples

- Example 4: consider TEL models of the pair of formulas

  \( \Box (\neg \circ p \rightarrow p) \)
  \( \Box (\circ p \rightarrow p) \)

- Curiosity: implications go backwards in time

- This is LTL-equivalent to:

  \( \Box ((\neg \circ p \rightarrow p) \land (\circ p \rightarrow p)) \)
  \( \equiv \Box (\neg \circ p \lor \circ p \rightarrow p) \)
  \( \equiv \Box p \)
Some examples

- Example 4: consider TEL models of the pair of formulas

\[ \Box(\neg \bigcirc p \rightarrow p) \]
\[ \Box(\bigcirc p \rightarrow p) \]

- So LTL models make \( p \) true forever,
Some examples

- Example 4: consider TEL models of the pair of formulas

\[ \square(\neg \circ p \rightarrow p) \]
\[ \square(\circ p \rightarrow p) \]

- So LTL models make \( p \) true forever, but we won’t get TEL models!
Some examples

Example 4: consider TEL models of the pair of formulas

\[ \Box (\neg \Diamond p \rightarrow p) \]
\[ \Box (\Diamond p \rightarrow p) \]

So LTL models make \( p \) true forever, but we won’t get TEL models!

We can build a strictly smaller model with \( H \) where from some point on \( T \), \( p \) becomes false forever

\[
\begin{array}{cccccc}
T & p & p & p & p & p \\
\| & \| & \| & \| & U & U \\
H & p & p & p & \emptyset & \emptyset \\
\end{array}
\]

\[ \bullet \rightarrow \bullet \rightarrow \ldots \rightarrow \bullet \rightarrow \bullet \rightarrow \bullet \rightarrow \ldots \]
Some examples

Example 5: lamp switch again

\[ \Box (up \land \neg \Diamond down) \rightarrow \Diamond up \]  
Initially

\[ \Box (down \land \neg \Diamond up) \rightarrow \Diamond down \]  
Inertia

\[ \Box (up \lor \neg up) \]  
Choice

We never get \( up \land \neg down \) Once \( up \) is true, it remains so forever

Pedro Cabalar (University of Corunna (Spain) cabalar@udc.es)

Temporal Modalities in ASP

TIME 2020 37 / 58
Some examples

Example 5: lamp switch again

\[ \square (up \land \neg \circ down) \rightarrow \circ up \] Initially
\[ \square (down \land \neg \circ up) \rightarrow \circ down \] Inertia
\[ \square (up \lor \neg up) \] Choice

We never get \( up \land down \)
Once \( up \) is true, it remains so forever
Some properties

- Reasonable behavior when theories “look like” logic programs

But what happens with arbitrary temporal formulas?

\[ \text{e.g.} \quad \diamond p \land (\neg \Box \diamond q \rightarrow \diamond (p \lor q)) \]

Kamp's translation into MFO(\textless) is applicable to TEL!

Example: Kamp \[\Box (\neg p \rightarrow \diamond p)\] amounts to:

\[ \forall x (\neg p(x) \rightarrow \exists y (y = x + 1 \land p(y))) \]

Temporal equilibrium models of \( \phi \) are in one-to-one relation to equilibrium models of the first-order formula Kamp \[\phi\].

FO-Equilibrium Logic is the most general logical characterisation of ASP.

Pedro Cabalar (University of Corunna (Spain) cabalar@udc.es [10pt])

Temporal Modalities in ASP

TIME 2020 38 / 58
Some properties

- Reasonable behavior when theories “look like” logic programs

- But what happens with arbitrary temporal formulas?
  e.g. $\Diamond p \land (\neg \Box \Diamond q \rightarrow \Diamond(p \lor q))$
Some properties

- Reasonable behavior when theories “look like” logic programs
- But what happens with arbitrary temporal formulas? e.g. \( \diamond p \land (\neg \Box \diamond q \rightarrow \diamond (p \cup q)) \)
- Kamp’s translation into MFO(\(<\)) is applicable to TEL!
Some properties

- Reasonable behavior when theories “look like” logic programs.
- But what happens with arbitrary temporal formulas?
  e.g. $\Diamond p \land (\neg \Box \Diamond q \rightarrow \Diamond (p \mathbin{\text{U}} q))$
- Kamp’s translation into MFO($<$) is applicable to TEL!
- Example $Kamp[\Box(\neg p \rightarrow \Diamond p)]$ amounts to:
  \[
  \forall x \ (\neg p(x) \rightarrow \exists y \ (y = x + 1 \land p(y)))
  \]
Some properties

- Reasonable behavior when theories “look like” logic programs
- But what happens with arbitrary temporal formulas? e.g. $\Diamond p \land (\neg \square \Diamond q \rightarrow \Diamond (p \mathbin{U} q))$
- Kamp’s translation into MFO($<$) is applicable to TEL!
- Example $Kamp[\square (\neg p \rightarrow \Diamond p)]$ amounts to:
  $$\forall x (\neg p(x) \rightarrow \exists y (y = x + 1 \land p(y)))$$
- Temporal equilibrium models of $\varphi$ are in one-to-one relation to Equilibrium Models of the first order formula $Kamp[\varphi]$
Some properties

- Reasonable behavior when theories “look like” logic programs

- But what happens with arbitrary temporal formulas?
  e.g. $\diamond p \land (\neg \Box \Box q \rightarrow \diamond (p \lor q))$

- Kamp’s translation into MFO(\prec) is applicable to TEL!

- Example $Kamp[\Box(\neg p \rightarrow \circ p)]$ amounts to:

  $$\forall x (\neg p(x) \rightarrow \exists y (y = x + 1 \land p(y)))$$

- Temporal equilibrium models of $\varphi$ are in one-to-one relation to Equilibrium Models of the first order formula $Kamp[\varphi]$

- FO-Equilibrium Logic is the most general logical characterisation of ASP
1. Introduction

2. Definitions and examples

3. Automata-based computation

4. Temporal Logic Programming

5. Conclusions and open topics
1. Encoding THT into LTL

- THT can be encoded into LTL, adding auxiliary atoms using the same translation of $\rightarrow$ from HT to classical logic.

- Intuition: $p$ will represent $p \in T$ whereas $p'$ will mean $p \in H$. 

---

Pedro Cabalar (University of Corunna (Spain))
cabalar@udc.es

Temporal Modalities in ASP

TIME 2020 40 / 58
1. Encoding THT into LTL

- THT can be encoded into LTL, adding auxiliary atoms using the same translation of $\rightarrow$ from HT to classical logic.

- Intuition: $p$ will represent $p \in T$ whereas $p'$ will mean $p \in H$.

**Example**

<table>
<thead>
<tr>
<th>THT</th>
<th>LTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Box (\downarrow \land \neg \Diamond \uparrow \rightarrow \Diamond \downarrow)$</td>
<td>$\Box (\uparrow' \rightarrow \uparrow) \land \Box (\downarrow' \rightarrow \downarrow)$</td>
</tr>
<tr>
<td>$\land \Box (\downarrow \land \neg \Diamond \uparrow \rightarrow \Diamond \downarrow)$</td>
<td>$\land \Box (\downarrow' \land \neg \Diamond \uparrow \rightarrow \Diamond \downarrow')$</td>
</tr>
</tbody>
</table>
1. Encoding THT into LTL

- **Warning**: this does not mean that we can encode Temporal Stable Models as models of an LTL theory!
Warning: this does not mean that we can encode Temporal Stable Models as models of an LTL theory!

This is an open question
(failed attempt [C_ & Diéguez, ASPOCP’14])

We know it holds for some fragments (splittable temporal programs)
1. Encoding THT into LTL

- **Warning**: this does not mean that we can encode Temporal Stable Models as models of an LTL theory!

- This is an open question
  (failed attempt [C_ & Diéguez, ASPOCP'14])

We know it holds for some fragments (splittable temporal programs)

- THT-satisfiability = \( \text{PSpace-complete} \) [C_ & Demri 11]
- TEL-satisfiability = \( \text{ESpace-complete} \) [Bozzelli & Pearce 15]
Automata-based methods

[C_ & Demri 2011]

Definition (Automata Based Computation Method)

LTL (i.e. total) models which do not have a strictly smaller $\langle H, T \rangle$

Intuition: $A_{\varphi}'$ captures the $\langle H, T \rangle$ satisfying $H < T$
Automata-based methods

[C & Demri 2011]

**Definition (Automata Based Computation Method)**

LTL (i.e. total) models which do not have a strictly smaller $\langle H, T \rangle$

$A_\varphi \otimes h(A_{\varphi'})$

- **Intuition:** $A_{\varphi'}$ captures the $\langle H, T \rangle$ satisfying $H < T$

- We use the $\varphi^*$ translation and force non-LTL models.

Example: if $\varphi = \diamond up$ then

$\varphi' = \diamond up' \land \Box(up' \rightarrow up) \land \diamond(up \land \neg up')$
Automata-based methods

[C. & Demri 2011]

**Definition (Automata Based Computation Method)**

LTL (i.e. total) models which do not have a strictly smaller $\langle H, T \rangle$

$A_\varphi \otimes h(A_\varphi')$

- **Intuition:** $A_\varphi'$ captures the $\langle H, T \rangle$ satisfying $H < T$

- We use the $\varphi^*$ translation and force non-LTL models.

  **Example:** if $\varphi = \diamond up$ then

  $\varphi' = \diamond up' \land \Box (up' \rightarrow up) \land \diamond (up \land \neg up')$

- Operation $h(A_\varphi')$ filters out the auxiliary atoms $p'$

Pedro Cabalar

Temporal Modalities in ASP

TIME 2020 42 / 58
Automata-based methods

Definition (Automata Based Computation Method)

LTL (i.e., total) models which do not have a strictly smaller \( \langle H, T \rangle \)

\[
A_{\varphi} \otimes h(A_{\varphi'})
\]

Intuition: \( A_{\varphi'} \) captures the \( \langle H, T \rangle \) satisfying \( H < T \)

- We use the \( \varphi^{*} \) translation and force non-LTL models.
  
  **Example:** if \( \varphi = \Diamond up \) then
  
  \[
  \varphi' = \Diamond up' \land \Box(up' \rightarrow up) \land \Diamond(up \land \neg up')
  \]

- Operation \( h(A_{\varphi'}) \) filters out the auxiliary atoms \( p' \)

- Büchi automata are closed w.r.t. complementation and intersection
Example

$\text{up} \lor \text{down}.
\square (\text{up} \land \neg \Diamond \text{down} \rightarrow \Diamond \text{up}).
\square (\text{down} \land \neg \Diamond \text{up} \rightarrow \Diamond \text{down}).
\square (\text{up} \lor \neg \text{up})$

\[\text{down} \land \neg \text{up} \rightarrow \neg \text{down} \land \text{up} \rightarrow \text{down} \land \text{up}\]
Example

\[ up \lor down. \]

\[ \Box (up \land \neg \lozenge down \rightarrow \lozenge up). \]

\[ \Box (down \land \neg \lozenge up \rightarrow \lozenge down). \]

\[ \Box (up \lor \neg up) \]

\[ \lozenge \Box up \rightarrow \Box stuck. \]
Example

\( \text{up} \lor \text{down} \).

\[ \Box (\text{up} \land \neg \Diamond \text{down} \rightarrow \Diamond \text{up}) \]

\[ \Box (\text{down} \land \neg \Diamond \text{up} \rightarrow \Diamond \text{down}) \]

\[ \Box (\text{up} \lor \neg \text{up}) \]

\[ \Diamond \Box \text{up} \rightarrow \Box \text{stuck} \]
1. Introduction

2. Definitions and examples

3. Automata-based computation

4. Temporal Logic Programming

5. Conclusions and open topics
Finite traces and past operators

- \( \text{LTL}_f = \text{LTL for finite traces} \) [De Giacomo & Vardi 13]
Finite traces and past operators

- $\text{LTL}_f = \text{LTL for finite traces} [\text{De Giacomo & Vardi 13}]

- Closer to ASP problem solving strategy:
  solutions for planning, diagnosis, explanation, \ldots are finite
Finite traces and past operators

- LTL$_f$ = LTL for finite traces [De Giacomo & Vardi 13]

- Closer to ASP problem solving strategy: solutions for planning, diagnosis, explanation, ... are finite

- Natural representation of dynamic rules: [Gabbay 87]

  declarative past $\rightarrow$ imperative future
Finite traces and past operators

- $\text{LTL}_f = \text{LTL}$ for finite traces [De Giacomo & Vardi 13]
- Closer to ASP problem solving strategy: solutions for planning, diagnosis, explanation, ... are finite
- Natural representation of dynamic rules: [Gabbay 87]
  declarative past $\rightarrow$ imperative future
- Adding past operators to LTL: same expressiveness but
Finite traces and past operators

- $\text{LTL}_f = \text{LTL for finite traces}$ [De Giacomo & Vardi 13]

- Closer to ASP problem solving strategy: solutions for planning, diagnosis, explanation, . . . are finite

- Natural representation of dynamic rules: [Gabbay 87]
  declarative past $\rightarrow$ imperative future

- Adding past operators to LTL: same expressiveness but exponentially more succinct [Markey 03]
Finite traces and past operators

<table>
<thead>
<tr>
<th>Past</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>●</td>
<td>○</td>
</tr>
<tr>
<td>S</td>
<td>U</td>
</tr>
<tr>
<td>T</td>
<td>R</td>
</tr>
</tbody>
</table>

for |

previous
since
trigger
next
until
release
while
Finite traces and past operators

<table>
<thead>
<tr>
<th>Past</th>
<th>Future</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\bullet) for previous</td>
<td>(\circ) for next</td>
</tr>
<tr>
<td>(S) for since</td>
<td>(U) for until</td>
</tr>
<tr>
<td>(T) for trigger</td>
<td>(R) for release</td>
</tr>
<tr>
<td>(W) for while</td>
<td></td>
</tr>
</tbody>
</table>

\(\begin{align*}
\square \varphi & \overset{\text{def}}{=} \perp T \varphi & \text{always before} \\
\lozenge \varphi & \overset{\text{def}}{=} T S \varphi & \text{eventually before} \\
\ulcorner \varphi \urcorner & \overset{\text{def}}{=} \neg \bullet T & \text{initial} \\
\hat{\bullet} \varphi & \overset{\text{def}}{=} \bullet \varphi \lor I & \text{weak previous} \\
\end{align*}
\)

\(\begin{align*}
\square \varphi & \overset{\text{def}}{=} \perp R \varphi & \text{always after} \\
\lozenge \varphi & \overset{\text{def}}{=} T U \varphi & \text{eventually after} \\
F & \overset{\text{def}}{=} \neg T & \text{final} \\
\hat{\circ} \varphi & \overset{\text{def}}{=} \varphi \lor F & \text{weak next}
\end{align*}\)
Satisfaction of formulas introduces conditions on **trace limits** on the past \((i \geq 0)\) and the future \((i < \lambda)\)

<table>
<thead>
<tr>
<th>(\alpha)</th>
<th>(M, i \models \alpha) when ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\Box \varphi)</td>
<td>(i + 1 &lt; \lambda) and ((M, i+1) \models \varphi)</td>
</tr>
<tr>
<td>(\Diamond \varphi)</td>
<td>(i + 1 = \lambda) or ((M, i+1) \models \varphi)</td>
</tr>
<tr>
<td>(\varphi \ U \ \psi)</td>
<td>(\exists j : i \leq j &lt; \lambda, \ M, j \models \psi) and (\forall k) s.t. (i \leq k &lt; j, \ M, k \models \varphi)</td>
</tr>
<tr>
<td>(\bullet \varphi)</td>
<td>(i &gt; 0) and ((M, i-1) \models \varphi)</td>
</tr>
<tr>
<td>(\hat{\bullet} \varphi)</td>
<td>(i = 0) or ((M, i-1) \models \varphi)</td>
</tr>
<tr>
<td>(\varphi \ S \ \psi)</td>
<td>(\exists j : 0 \leq j \leq i, \ M, j \models \psi) and (\forall k) s.t. (j &lt; k \leq i, \ M, k \models \varphi)</td>
</tr>
</tbody>
</table>
 Finite traces and past operators

- Satisfaction of formulas introduces conditions on trace limits on the past ($i \geq 0$) and the future ($i < \lambda$)

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th>$M, i \models \alpha$ when ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Diamond \varphi$</td>
<td>$i + 1 &lt; \lambda$ and $(M, i+1) \models \varphi$</td>
</tr>
<tr>
<td>$\Diamond^* \varphi$</td>
<td>$i + 1 = \lambda$ or $(M, i+1) \models \varphi$</td>
</tr>
<tr>
<td>$\varphi \mathbf{U} \psi$</td>
<td>$\exists j : i \leq j &lt; \lambda$, $M, j \models \psi$ and $\forall k$ s.t. $i \leq k &lt; j$, $M, k \models \varphi$</td>
</tr>
<tr>
<td>$\Diamond \varphi$</td>
<td>$i &gt; 0$ and $(M, i-1) \models \varphi$</td>
</tr>
<tr>
<td>$\Diamond^* \varphi$</td>
<td>$i = 0$ or $(M, i-1) \models \varphi$</td>
</tr>
<tr>
<td>$\varphi \mathbf{S} \psi$</td>
<td>$\exists j : 0 \leq j \leq i$, $M, j \models \psi$ and $\forall k$ s.t. $j &lt; k \leq i$, $M, k \models \varphi$</td>
</tr>
</tbody>
</table>

- When $\lambda = \omega$ we get (infinite-traces) TEL as before
Temporal theories can be reduced to a normal form closer to logic programs.
Normal form

- Temporal theories can be reduced to a normal form closer to logic programs
- temporal literals = \{a, ¬a, \bullet a, ¬\bullet a \mid a \in Atoms\}

Definition (Temporal rule)

A temporal rule is either:

- an initial rule \( B \rightarrow A \)
- a dynamic rule \( \widehat{\circ}(B \rightarrow A) \)
- a fulfillment rule \( \Box(\Box p \rightarrow q) \) or \( \Box(p \rightarrow \Diamond q) \)

where \( B = b_1 \land \cdots \land b_n \) with \( n \geq 0 \), \( A = a_1 \lor \cdots \lor a_m \) with \( m \geq 0 \)

\( b_i, a_j \) = temporal literals for dynamic rules
\( b_i, a_j \) = regular literals \( a, \neg a \) for initial rules
\( p, q \) = atoms
Temporal theories can be reduced to a normal form closer to logic programs

temporal literals = \{a, \neg a, \lozenge a, \neg \lozenge a \mid a \in Atoms\}

Definition (Temporal rule)

A temporal rule is either:
- an initial rule \( B \rightarrow A \)
- a dynamic rule \( \lozenge \Box (B \rightarrow A) \)
- a final rule \( \Box (F \rightarrow (B \rightarrow A)) \) when traces are finite

where \( B = b_1 \land \cdots \land b_n \) with \( n \geq 0 \), \( A = a_1 \lor \cdots \lor a_m \) with \( m \geq 0 \)

\( b_i, a_j \) = temporal literals for dynamic rules

\( b_i, a_j \) = regular literals \( a, \neg a \) for initial rules

\( p, q \) = atoms

A temporal logic program is a set of temporal rules.
An interesting fragment are present-centered programs

- initial rule $B \rightarrow A$
- dynamic rule $\circ \Box (B \rightarrow A)$
- final rule $\Box (F \rightarrow (B \rightarrow A))$

present-centered = $A$ does not contain temporal operators
An interesting fragment are present-centered programs

- initial rule \( B \rightarrow A \)
- dynamic rule \( \Diamond \Box (B \rightarrow A) \)
- final rule \( \Box (F \rightarrow (B \rightarrow A)) \)

present-centered = \( A \) does not contain temporal operators

Example of present-centered program:

\[
\begin{align*}
\text{Initially} & \quad up \lor down \\
\text{Inertia} & \quad \Diamond \Box (\lozenge up \land \neg down \rightarrow up) \\
\text{Inertia} & \quad \Diamond \Box (\lozenge down \land \neg up \rightarrow down) \\
\text{Choice} & \quad \Box (up \lor \neg up)
\end{align*}
\]
Tool telingo [C., Kaminski, Morkisch & Schaub 19]
Temporal extension of ASP solver clingo
Tool telingo [C_, Kaminski, Morkisch & Schaub 19]
Temporal extension of ASP solver clingo

#program initial.
up;down.

#program dynamic.
up :- 'up, not down.
down :- 'down, not up.

#program always.
{up}.

Pedro Cabalar ( University of Corunna (Spain)
cabalar@udc.es

Temporal Modalities in ASP
TIME 2020 51 / 58
We can use a more general syntactic fragment
\[ \text{past-future} = \alpha \rightarrow \beta \]
where
- \( \alpha \) may only contain past operators
- \( \beta \) may only contain future operators

and none of them contains \( \rightarrow \)
We can use a more general syntactic fragment
\[ \text{past-future} = \alpha \rightarrow \beta \]
where
- \( \alpha \) may only contain past operators
- \( \beta \) may only contain future operators

and none of them contains \( \rightarrow \)

Example: the integrity constraint

\[ \text{shoot} \land \Box \text{unloaded} \land \Diamond \Diamond \text{shoot} \rightarrow \bot \]

can be expressed in \text{telingo} as:
We can use a more general syntactic fragment
\[ \text{past-future} = \alpha \rightarrow \beta \]
where
- \( \alpha \) may only contain past operators
- \( \beta \) may only contain future operators

and none of them contains \( \rightarrow \)

Example: the integrity constraint

\[ \text{shoot} \land \Box \text{unloaded} \land \Diamond \Diamond \text{shoot} \rightarrow \bot \]

can be expressed in telingo as:

\[ :- \text{shoot}, \landtel \{ \landstar \text{unloaded} \land < <? \text{shoot} \}. \]
Beyond LTL

- In [Bosser et al. 18, C_ et al. 19] we extend TEL and TEL$_f$ to the syntax of Linear Dynamic Logic (LDL) [De Giacomo & Vardi 13]

- DEL = LDL + ASP.
Beyond LTL

- In [Bosser et al. 18, C_ et al. 19] we extend TEL and $TEL_f$ to the syntax of Linear Dynamic Logic (LDL) [De Giacomo & Vardi 13]

- $DEL = LDL + ASP$. Example:
Beyond LTL

- In [Bosser et al. 18, C_ et al. 19] we extend TEL and TEL\textsubscript{f} to the syntax of Linear Dynamic Logic (LDL) [De Giacomo & Vardi 13]

- DEL = LDL + ASP. Example:

  \[ \perp \neg \leftrightarrow \langle ( (up^* + down^*); ready?; serve )^*; wait^* \rangle F \]

  elevator moving in a unique direction until the call is served
Beyond LTL

- In [Bosser et al. 18, C_ et al. 19] we extend TEL and TEL\(_f\) to the syntax of Linear Dynamic Logic (LDL) [De Giacomo & Vardi 13]

- DEL = LDL + ASP. Example:
  \[
  \perp \leftarrow \langle ( (up^* + down^*); ready?; serve )^*; wait^* \rangle F
  \]
  elevator moving in a unique direction until the call is served

- [C_ et al, ECAI 20] LDL operators implemented in telingo (only in constraints)

```prolog
#program initial.
:- not &del{ *( (*up + *down) ;; ?ready ;; serve) ;; *wait .>? &final }.
```
In [C_ et al, ICLP 20] (Tomorrow 18:15) we introduce metric operators

\( \square (\text{red} \land \text{green} \rightarrow \bot) \)
\( \square (\neg \text{green} \rightarrow \text{red}) \)
\( \square (\text{push} \rightarrow \Diamond_3 \square_4 \text{green}) \)

The traffic light is red by default
Beyond LTL

In [C et al, ICLP 20] (Tomorrow 18:15) we introduce metric operators

\[ \square (\text{red} \land \text{green} \rightarrow \bot) \]
\[ \square (\neg \text{green} \rightarrow \text{red}) \]
\[ \square (\text{push} \rightarrow \Diamond 3 \square 4 \text{green}) \]

The traffic light is red by default when we push it, it takes at most 3 steps to stay green for 4 steps.
Beyond LTL

In [C_ et al, ICLP 20] we introduce metric operators

\( \Box ( \text{red} \land \text{green} \rightarrow \bot) \)
\( \Box (\neg \text{green} \rightarrow \text{red}) \)
\( \Box (\text{push} \rightarrow \Diamond_3 \Box_4 \text{green}) \)

The traffic light is red by default when we push it, it takes at most 3 steps to stay green for 4 steps.

We extended this for intervals (only discrete by now).
1 Introduction

2 Definitions and examples

3 Automata-based computation

4 Temporal Logic Programming

5 Conclusions and open topics
Forthcoming survey: [Aguado et al. 2020] (under review)
TPLP 20th anniversary special issue
Conclusions

👍 Forthcoming survey: [Aguado et al. 2020] (under review)

TPLP 20th anniversary special issue

- **TEL = suitable framework** for temporal reasoning + ASP

- Simple semantics thanks to just **merging two logical** formalisms: Equilibrium Logic + LTL.

- **Implementations:** telingo, abstem, stelp

- It constitutes a new **open field.** Many open topics . . .
Open topics

- Open theoretical problems:
  - Kamp’s theorem: monadic EL(\(<\)) can be transformed into THT? (possibly not)
  - Interdefinability of operators
  - Can temporal stable models be captured by LTL?

- Finite traces: axiomatisation, automata-based methods, grounding

- New syntactic subclasses with satisfiability lower than EXPSPACE [Bozzelli & Pearce 15]

- Planning tool. Compare to planners using LTL control knowledge like TLPlan [Bacchus & Kabanza 00].

- Encoding action languages
Temporal Modalities in Answer Set Programming

Pedro Cabalar

Thank you for your attention!

September 23rd, 2020
TIME 2020
Bozen-Bolzano, Italy