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)

Motivation

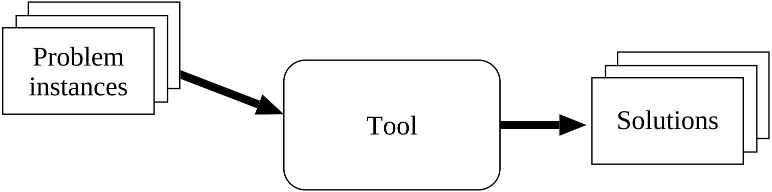
- Context: temporal reasoning with transition systems.
- Typical reasoning problems:
  simulation, explanation, planning, diagnosis, verification.

Motivation

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

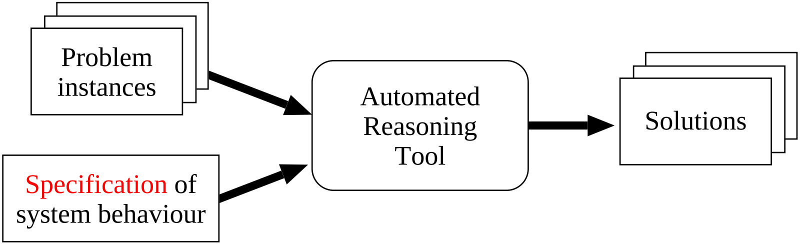
Motivation

- 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)



Keypoint: representation

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

Keypoint: representation

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

Keypoint: representation

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

Keypoint: representation

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

Temporal ASP

- 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:

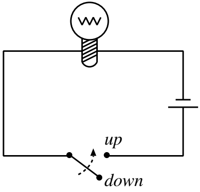
  Representing Action and Change by Logic Programs
  [Gelfond & Lifschitz 93] use Answer Set Programming (ASP)

- However, ASP has no temporal constructs

Transition systems in ASP

Example

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

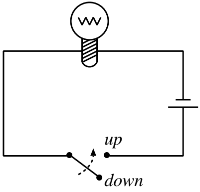



```
time(0..n).
up(0);down(0).
```

Transition systems in ASP

Example

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

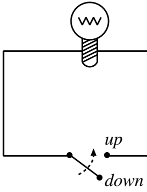



```
time(0..n).
up(0);down(0).

up(T+1)   :- up(T), not down(T+1), time(T).
down(T+1) :- down(T), not up(T+1), time(T).
```

Transition systems in ASP

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).
```

An example

Examples of problems that cannot be solved in ASP:

- Is there a reachable state with *up* and *down* false?

An example

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?

An example

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), $\mathbf{U}$ (until)

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

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)

$\Box$ (forever), $\Diamond$ (eventually), $\circ$ (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 (<)

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
- ✓ 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)

$\Box$ (forever), $\Diamond$ (eventually), $\circ$ (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
- ✓ Many uses in AI: planning, ontologies, multi-agent systems, . . .

Modal Temporal Logic

A simple and well-known example

Linear-time Temporal Logic (LTL)

$\Box$, $\Diamond$, $\circ$, **U** ...

✗ Monotonic: action domain representations manifest frame problem

Modal Temporal Logic

A simple and well-known example

Linear-time Temporal Logic (LTL)

$\square$, $\Diamond$, $\circ$, **U** . . .

✗ Monotonic: action domain representations manifest frame problem

In model checking no worry on this:
usually, logical description of automaton states

Modal Temporal Logic

A simple and well-known example

Linear-time Temporal Logic (LTL)

$\square$, $\Diamond$, $\circ$, **U** ...

✗ 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)

$\Box$, $\Diamond$, $\circ$, $\mathbf{U}$ ...

- ✗ 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

- ✗ 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

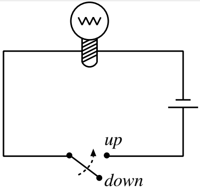
- ASP: logical characterisation Equilibrium Logic [Pearce 96]
- LTL: We add temporal operators $\square$, $\Diamond$, $\circ$, **U**, **R**.

Result: Temporal Stable Models for any arbitrary LTL theory.

Example

Example

- 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.

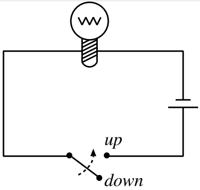



```
time(0..n).
up(0),down(0).

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).
```

Example

Example

- 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.



$up \vee down$	Initially
$\Box(\circ up \leftarrow up \wedge \neg\circ down)$	Inertia
$\Box(\circ down \leftarrow down \wedge \neg\circ up)$	Inertia
$\Box(up \vee \neg up)$	Choice

Idea: LTL syntax, but keeping ASP semantics

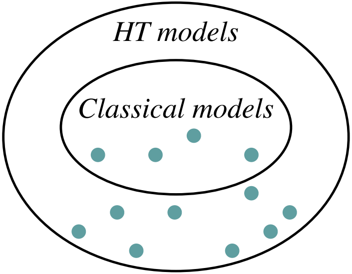
Equilibrium Logic

Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories.

Equilibrium Logic

Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

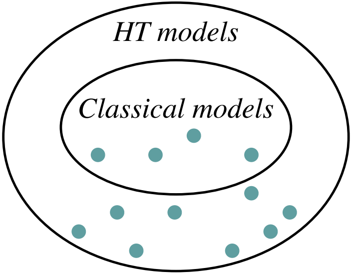
1. A non-classical monotonic (intermediate) logic called Here-and-There (HT)



Equilibrium Logic

Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

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

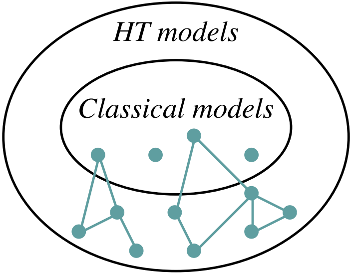


2. A selection of (certain) minimal models that yields nonmonotonicity

Equilibrium Logic

Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

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

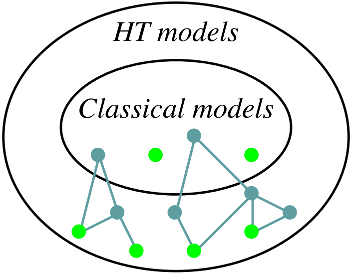


2. A selection of (certain) minimal models that yields nonmonotonicity

Equilibrium Logic

Equilibrium Logic [Pearce96]: generalises stable models for arbitrary propositional theories. Consists of:

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

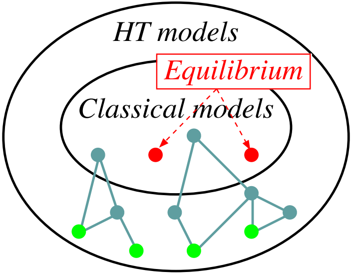


2. A selection of (certain) minimal models that yields nonmonotonicity

Equilibrium Logic

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

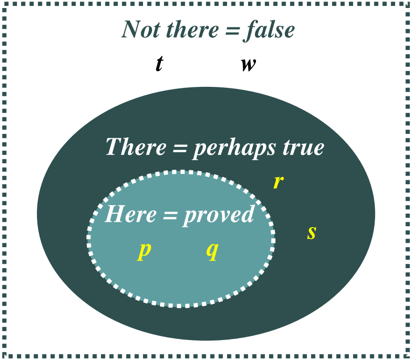
- Interpretation = pairs $\langle H, T \rangle$ of sets of atoms $H \subseteq T$

Here-and-There

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

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:



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

Here-and-There

Satisfaction of formulas

$\langle H, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ is proved"

Here-and-There

Satisfaction of formulas

$\langle H, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ is proved"

$\langle T, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true" $\quad \Leftrightarrow \quad T \models \varphi$ classically

Here-and-There

Satisfaction of formulas

$\langle H, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ is proved"

$\langle T, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true" $\quad \Leftrightarrow \quad T \models \varphi$ classically

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

Here-and-There

Satisfaction of formulas

$\langle H, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ is proved"
$\langle T, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true" $\quad \Leftrightarrow \quad T \models \varphi$ classically

- $\langle H, T \rangle \models p$ if $p \in H$ (for any atom p)
- $\wedge, \vee$ as always

Here-and-There

Satisfaction of formulas

$\langle H, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ is proved"

$\langle T, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true" $\quad \Leftrightarrow \quad T \models \varphi$ classically

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

Here-and-There

Satisfaction of formulas

$\langle H, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ is proved"

$\langle T, T \rangle \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true" $\quad \Leftrightarrow \quad T \models \varphi$ classically

- $\langle H, T \rangle \models p$ if $p \in H$ (for any atom p)
- $\wedge, \vee$ 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 Γ 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.

Equilibrium models

Definition (Equilibrium/stable model)

A model $\langle T, T \rangle$ of Γ 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

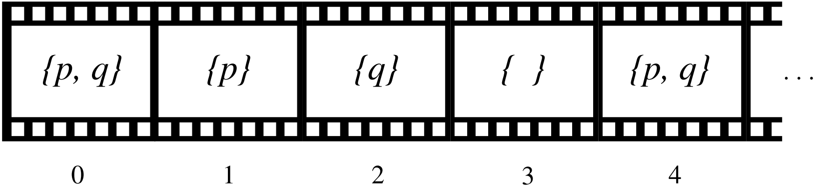
- Syntax = propositional plus
  - $\Box\varphi$ = "forever" φ
  - $\Diamond\varphi$ = "eventually" φ
  - $\circ\varphi$ = "next moment" φ
  - φ **U** ψ = φ "until eventually" ψ
  - φ **R** ψ = φ "release" ψ
  - φ **W** ψ = φ "while" ψ

(Linear) Temporal Equilibrium Logic

- Syntax = propositional plus
  - $\Box\varphi$ = "forever" φ
  - $\Diamond\varphi$ = "eventually" φ
  - $\circ\varphi$ = "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.

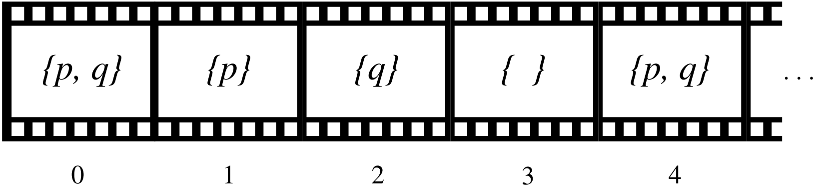
Sequences

- In standard LTL, interpretations are ∞ sequences of sets of atoms

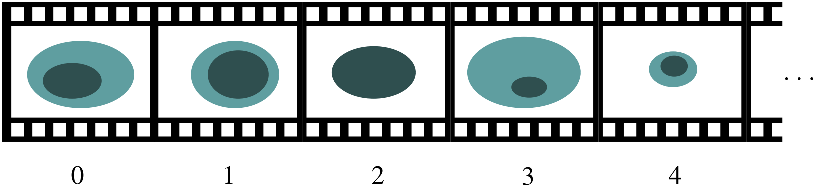


Sequences

- In standard LTL, interpretations are ∞ sequences of sets of atoms



- In THT we will have ∞ sequences of HT interpretations



Sequences

- We define an ordering among sequences $\mathbf{H} \leq \mathbf{T}$ when

$$\begin{array}{ccccccccccc} T_0 & \longrightarrow & T_1 & \longrightarrow & T_2 & \longrightarrow & \dots & \longrightarrow & T_i & \longrightarrow & \dots \\ \cup\vert & & \cup\vert & & \cup\vert & & & & \cup\vert & & \\ H_0 & \longrightarrow & H_1 & \longrightarrow & H_2 & \longrightarrow & \dots & \longrightarrow & H_i & \longrightarrow & \dots \end{array}$$

Sequences

- We define an ordering among sequences $\mathbf{H} \leq \mathbf{T}$ when

$$
\begin{array}{ccccccccccc}
T_0 & \longrightarrow & T_1 & \longrightarrow & T_2 & \longrightarrow & \dots & \longrightarrow & T_i & \longrightarrow & \dots \\
\cup\vert & & \cup\vert & & \cup\vert & & & & \cup\vert & & \\
H_0 & \longrightarrow & H_1 & \longrightarrow & H_2 & \longrightarrow & \dots & \longrightarrow & H_i & \longrightarrow & \dots
\end{array}
$$

Definition (THT-interpretation)

is a pair of sequences of sets of atoms $\langle \mathbf{H}, \mathbf{T} \rangle$ with $\mathbf{H} \leq \mathbf{T}$. □

Sequences

- We define an ordering among sequences $\mathbf{H} < \mathbf{T}$ when

$$
\begin{array}{ccccccccccc}
T_0 & \longrightarrow & T_1 & \longrightarrow & T_2 & \longrightarrow & \dots & \longrightarrow & T_i & \longrightarrow & \dots \\
\cup\vert & & \cup\vert & & \cup & & & & \cup\vert & & \\
H_0 & \longrightarrow & H_1 & \longrightarrow & H_2 & \longrightarrow & \dots & \longrightarrow & H_i & \longrightarrow & \dots
\end{array}
$$

Definition (THT-interpretation)

is a pair of sequences of sets of atoms $\langle \mathbf{H}, \mathbf{T} \rangle$ with $\mathbf{H} \leq \mathbf{T}$. □

Temporal Here-and-There (THT)

$\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi \quad \Leftrightarrow \quad$ "φ is proved at i"

Temporal Here-and-There (THT)

$\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi \quad \Leftrightarrow \quad$ "φ is proved at i"

$\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true at i" $\quad \Leftrightarrow \quad \mathbf{T}, i \models \varphi$ in LTL

Temporal Here-and-There (THT)

$\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi \quad \Leftrightarrow \quad$ "φ is proved at i"

$\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true at i" $\quad \Leftrightarrow \quad \mathbf{T}, i \models \varphi$ in LTL

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

α	$M, i \models \alpha$ when ...
an atom p	$p \in H_i$
$\wedge, \vee$	as usual
$\varphi \rightarrow \psi$	$\mathbf{T}, i \models \varphi \rightarrow \psi$ in LTL and $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi$ implies $\langle \mathbf{H}, \mathbf{T} \rangle, i \models \psi$

Temporal Here-and-There (THT)

$\langle \mathbf{H}, \mathbf{T} \rangle, i \models \varphi \quad \Leftrightarrow \quad$ "φ is proved at i"

$\langle \mathbf{T}, \mathbf{T} \rangle, i \models \varphi \quad \Leftrightarrow \quad$ "φ potentially true at i" $\quad \Leftrightarrow \quad \mathbf{T}, i \models \varphi$ in LTL

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

α	$M, i \models \alpha$ when ...
$\circ\varphi$	$(M, i{+}1) \models \varphi$
$\Box\varphi$	$\forall j \geq i, \quad M, j \models \varphi$
$\Diamond\varphi$	$\exists j \geq i, \quad M, j \models \varphi$
$\varphi \ \mathbf{U} \ \psi$	$\exists j \geq i, \quad M, j \models \psi$ and $\forall k$ s.t. $i \leq k < j, \quad M, k \models \varphi$
$\varphi \ \mathbf{R} \ \psi$	$\forall j \geq i, \quad M, j \models \psi$ or $\exists k, i \leq k < j, \quad M, k \models \varphi$

- M is a model of a theory Γ when $M, 0 \models \alpha$ for all $\alpha \in \Gamma$

(Linear) Temporal Equilibrium Logic

- $\circ\varphi$

(Linear) Temporal Equilibrium Logic

- $\bigcirc\varphi$

- $\square\varphi$

(Linear) Temporal Equilibrium Logic

- $\bigcirc\varphi$

- $\square\varphi$

- $\Diamond\varphi$

(Linear) Temporal Equilibrium Logic

- φ **U** ψ = repeat φ until (mandatorily) ψ

- φ **U** ψ = repeat φ until (mandatorily) ψ

(Linear) Temporal Equilibrium Logic

- φ **U** ψ = repeat φ until (mandatorily) ψ

- φ **U** ψ = repeat φ until (mandatorily) ψ

(Linear) Temporal Equilibrium Logic

- φ **U** ψ = repeat φ until (mandatorily) ψ

(Linear) Temporal Equilibrium Logic

- φ **U** ψ = repeat φ until (mandatorily) ψ

(Linear) Temporal Equilibrium Logic

φ **R** ψ = disjunction of two cases

- ψ **U** $(\psi \wedge \varphi)$

(Linear) Temporal Equilibrium Logic

φ **R** ψ = disjunction of two cases

- ψ **U** $(\psi \wedge \varphi)$

(Linear) Temporal Equilibrium Logic

φ **R** ψ = disjunction of two cases

- ψ **U** $(\psi \wedge \varphi)$

(Linear) Temporal Equilibrium Logic

φ **R** ψ = disjunction of two cases

- ψ **U** $(\psi \wedge \varphi)$

(Linear) Temporal Equilibrium Logic

φ **R** ψ = disjunction of two cases

- ψ **U** $(\psi \wedge \varphi)$

φ **R** ψ = disjunction of two cases

- ψ **U** $(\psi \wedge \varphi)$

(Linear) Temporal Equilibrium Logic

φ **R** ψ = disjunction of two cases

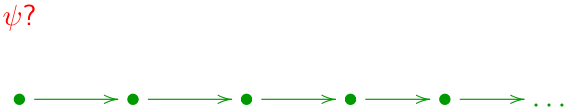
- ψ **U** $(\psi \wedge \varphi)$

- $\Box\psi$

φ **W** ψ = do φ while ψ

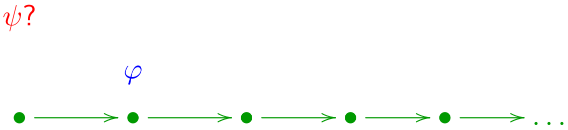
(Linear) Temporal Equilibrium Logic

φ **W** ψ = do φ while ψ

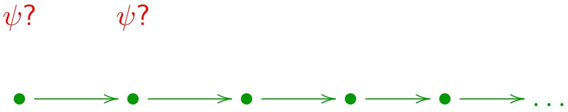


(Linear) Temporal Equilibrium Logic

φ **W** ψ = do φ while ψ

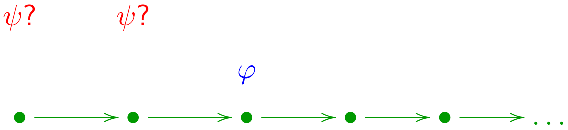


φ **W** ψ = do φ while ψ



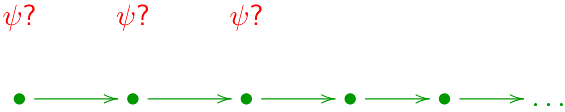
(Linear) Temporal Equilibrium Logic

φ **W** ψ = do φ while ψ

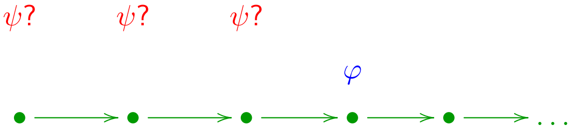


(Linear) Temporal Equilibrium Logic

φ **W** ψ = do φ while ψ



φ **W** ψ = do φ while ψ



Temporal Here-and-There (THT)

- Some valid THT formulas:

$$
\begin{array}{rcl}
\Diamond\varphi & \leftrightarrow & \top \;\mathbf{U}\; \varphi \\
\Box\varphi & \leftrightarrow & \bot \;\mathbf{R}\; \varphi \\
\circ(\varphi \otimes \psi) & \leftrightarrow & \circ\varphi \otimes \circ\psi \\
\varphi \;\mathbf{U}\; \psi & \leftrightarrow & \psi \vee (\varphi \wedge \circ(\varphi \;\mathbf{U}\; \psi)) \\
\varphi \;\mathbf{R}\; \psi & \leftrightarrow & \psi \wedge (\varphi \vee \circ(\varphi \;\mathbf{R}\; \psi)) \\
\varphi \;\mathbf{W}\; \psi & \leftrightarrow & \varphi \wedge (\psi \rightarrow \circ(\varphi \;\mathbf{W}\; \psi)) \\
\neg(\varphi \;\mathbf{U}\; \psi) & \leftrightarrow & \neg\varphi \;\mathbf{R}\; \neg\psi \\
\circ\neg\varphi & \leftrightarrow & \neg\circ\varphi \\
\neg(\varphi \;\mathbf{R}\; \psi) & \leftrightarrow & \neg\varphi \;\mathbf{U}\; \neg\psi
\end{array}
$$

For $\otimes = \wedge, \vee, \rightarrow, \mathbf{U}, \mathbf{R}$.

Temporal Here-and-There (THT)

- Some valid THT formulas:

$$\begin{array}{rcl}
\Diamond\varphi & \leftrightarrow & \top \ \mathbf{U} \ \varphi \\
\Box\varphi & \leftrightarrow & \bot \ \mathbf{R} \ \varphi \\
\circ(\varphi \otimes \psi) & \leftrightarrow & \circ\varphi \otimes \circ\psi \\
\varphi \ \mathbf{U} \ \psi & \leftrightarrow & \psi \vee (\varphi \wedge \circ(\varphi \ \mathbf{U} \ \psi)) \\
\varphi \ \mathbf{R} \ \psi & \leftrightarrow & \psi \wedge (\varphi \vee \circ(\varphi \ \mathbf{R} \ \psi)) \\
\varphi \ \mathbf{W} \ \psi & \leftrightarrow & \varphi \wedge (\psi \rightarrow \circ(\varphi \ \mathbf{W} \ \psi)) \\
\neg(\varphi \ \mathbf{U} \ \psi) & \leftrightarrow & \neg\varphi \ \mathbf{R} \ \neg\psi \\
\circ\neg\varphi & \leftrightarrow & \neg\circ\varphi \\
\neg(\varphi \ \mathbf{R} \ \psi) & \leftrightarrow & \neg\varphi \ \mathbf{U} \ \neg\psi
\end{array}$$

For $\otimes = \wedge, \vee, \rightarrow, \mathbf{U}, \mathbf{R}$.

- Axiomatization of THT [Balbiani & Diéguez 16]

Temporal Equilibrium Models

Definition (Temporal Equilibrium Model)

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

Temporal Equilibrium Models

Definition (Temporal Equilibrium Model)

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

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

Definition (Temporal Stable Model)

$\mathbf{T}$ is a temporal stable model of a theory Γ iff $\langle \mathbf{T}, \mathbf{T} \rangle$ is a temporal equilibrium model of Γ. □

Some examples

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

$$\begin{array}{rcl} \neg p & \rightarrow & \circ p \\ \neg \circ p & \rightarrow & \circ^2 p \\ \neg \circ^2 p & \rightarrow & \circ^3 p \\ & \vdots & \end{array}$$

Some examples

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

$$\begin{array}{rcl} \neg p & \rightarrow & \circ p \\ \neg \circ p & \rightarrow & \circ^2 p \\ \neg \circ^2 p & \rightarrow & \circ^3 p \\ & \vdots & \end{array}$$

- TEL models have the form

$$\begin{array}{cccccc} \emptyset & p & \emptyset & p & \emptyset & \\ \bullet \longrightarrow & \bullet \longrightarrow & \bullet \longrightarrow & \bullet \longrightarrow & \bullet \longrightarrow & \ldots \end{array}$$

corresponding to LTL models of $\neg p \wedge \square(\neg p \leftrightarrow \circ p)$.

Some examples

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

Some examples

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

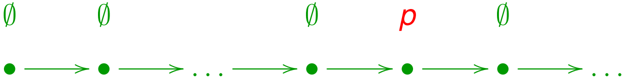
  is like $p \vee \circ p \vee \circ\circ p \vee \ldots$

Some examples

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

  is like $p \vee \circ p \vee \circ\circ p \vee \ldots$

  TEL models have the form



Some examples

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

  is like $p \vee \circ p \vee \circ\circ p \vee \dots$

  TEL models have the form

$$\begin{array}{ccccccccccc} \emptyset & & \emptyset & & & & \emptyset & & p & & \emptyset \\ \bullet & \longrightarrow & \bullet & \longrightarrow & \dots & \longrightarrow & \bullet & \longrightarrow & \bullet & \longrightarrow & \bullet \longrightarrow \dots \end{array}$$

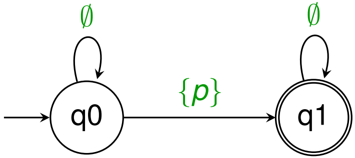
corresponding to LTL models of $\neg p \ \mathbf{U} \ (p \wedge \circ\Box\neg p)$

An example

- In ASP terms, how can we represent temporal stable models? infinitely long! infinitely many!

An example

- 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.

Some examples

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

Some examples

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

Some examples

- We can still express 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

$$\Box(p \vee \neg p) \tag{EM}$$

(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

$$\Box(p \vee \neg p) \qquad \text{(EM)}$$

  (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

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

Some examples

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

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

- Curiosity: implications go backwards in time

Some examples

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

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

- Curiosity: implications go backwards in time
- This is LTL-equivalent to:

$$\Box\big((\neg \circ p \rightarrow p) \wedge (\circ p \rightarrow p)\big)$$

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:

$$\begin{aligned} & \Box\big((\neg\circ p \rightarrow p) \wedge (\circ p \rightarrow p)\big) \\ \equiv\ & \Box\big(\underbrace{\neg\circ p \vee \circ p}_{\top} \rightarrow p\big) \\ \equiv\ & \Box p \end{aligned}$$

Some examples

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

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

- So LTL models make p true forever,

Some examples

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

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

- 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

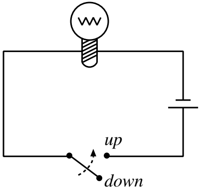
$$\square(\neg\circ p \;\rightarrow\; p)$$
$$\square(\circ p \;\rightarrow\; p)$$

- So LTL models make p true forever, but we won't get TEL models!
- We can build a strictly smaller model with **H** where from some point on **T**, p becomes false forever

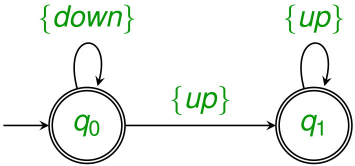
T	p	p		p	p	p
	$\parallel$	$\parallel$		$\parallel$	$\cup$	$\cup$
H	p	p		p	$\emptyset$	$\emptyset$
	$\bullet \longrightarrow$	$\bullet \longrightarrow$	$\ldots \longrightarrow$	$\bullet \longrightarrow$	$\bullet \longrightarrow$	$\bullet \longrightarrow \ldots$

Some examples

- Example 5: lamp switch again

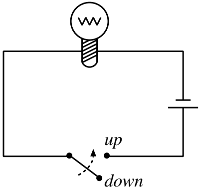


	$up \vee down$	Initially
$\Box(up \wedge \neg \circ down \rightarrow$	$\circ up)$	Inertia
$\Box(down \wedge \neg \circ up \rightarrow$	$\circ down)$	Inertia
	$\Box(up \vee \neg up)$	Choice

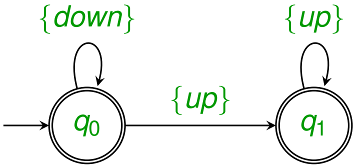


Some examples

- Example 5: lamp switch again



	$up \lor down$	Initially
$\Box(up \land \neg\circ down$	$\rightarrow \circ up)$	Inertia
$\Box(down \land \neg\circ up$	$\rightarrow \circ down)$	Inertia
	$\Box(up \lor \neg up)$	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

Some properties

- Reasonable behavior when theories "look like" logic programs
- But what happens with arbitrary temporal formulas?
  e.g. $\Diamond p \wedge (\neg \Box \Diamond q \rightarrow \Diamond (p \ \mathbf{U} \ q))$

Some properties

- Reasonable behavior when theories "look like" logic programs
- But what happens with arbitrary temporal formulas?
  e.g. $\Diamond p \wedge (\neg\Box\Diamond q \rightarrow \Diamond(p \ \mathbf{U} \ q))$
- 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. $\Diamond p \wedge (\neg \Box \Diamond q \rightarrow \Diamond (p \ \mathbf{U} \ q))$
- Kamp’s translation into MFO(<) is applicable to TEL!
- Example $Kamp[\ \Box(\neg p \rightarrow \circ p)\ ]$ amounts to:

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

Some properties

- Reasonable behavior when theories "look like" logic programs
- But what happens with arbitrary temporal formulas?
  e.g. $\Diamond p \wedge (\neg\Box\Diamond q \rightarrow \Diamond(p \ \mathbf{U} \ q))$
- Kamp's translation into MFO(<) is applicable to TEL!
- Example $Kamp[\ \Box(\neg p \rightarrow \circ p)\ ]$ amounts to:

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

- Temporal equilibrium models of φ 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. $\Diamond p \wedge (\neg \Box \Diamond q \rightarrow \Diamond (p \ \mathbf{U} \ q))$
- Kamp's translation into MFO(<) is applicable to TEL!
- Example $Kamp[\ \Box(\neg p \rightarrow \circ p)\ ]$ amounts to:

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

- Temporal equilibrium models of φ 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. Enconding 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 $p \in H$

1. Enconding 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

THT		**LTL**
$\Box(down \wedge \neg\circ up \rightarrow \circ down)$	$\equiv$	$\Box(up' \rightarrow up) \wedge \Box(down' \rightarrow down)$
		$\wedge\Box(down \wedge \neg\circ up \rightarrow \circ down)$
		$\wedge\Box(down' \wedge \neg\circ up \rightarrow \circ down')$

1. Encoding THT into LTL

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

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)

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 = PSPACE-complete [C_ & Demri 11]
  TEL-satisfiability = EXPSPACE-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 $\langle H, T \rangle$

$$\mathcal{A}_\varphi \quad \otimes \quad \overline{h(\mathcal{A}_{\varphi'})}$$

- Intuition: $\mathcal{A}_{\varphi'}$ captures the $\langle \mathbf{H}, \mathbf{T} \rangle$ satisfying $\mathbf{H} < \mathbf{T}$

Automata-based methods

[C_ & Demri 2011]

Definition (Automata Based Computation Method)

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

$$\mathcal{A}_\varphi \otimes \overline{h(\mathcal{A}_{\varphi'})}$$

- Intuition: $\mathcal{A}_{\varphi'}$ captures the $\langle \mathbf{H}, \mathbf{T} \rangle$ satisfying $\mathbf{H} < \mathbf{T}$
- We use the φ^* translation and force non-LTL models.
  Example: if $\varphi = \Diamond up$ then

  $\varphi' = \Diamond up' \wedge \Box(up' \rightarrow up) \wedge \Diamond(up \wedge \neg up')$

Automata-based methods

[C_ & Demri 2011]

Definition (Automata Based Computation Method)

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

$$\mathcal{A}_\varphi \quad \otimes \quad \overline{h(\mathcal{A}_{\varphi'})}$$

- Intuition: $\mathcal{A}_{\varphi'}$ captures the $\langle \mathbf{H}, \mathbf{T} \rangle$ satisfying $\mathbf{H} < \mathbf{T}$
- We use the φ^* translation and force non-LTL models.
  Example: if $\varphi = \Diamond up$ then

  $$\varphi' = \Diamond up' \wedge \Box(up' \rightarrow up) \wedge \Diamond(up \wedge \neg up')$$
- Operation $h(\mathcal{A}_{\varphi'})$ filters out the auxiliary atoms p'

Automata-based methods

[C_ & Demri 2011]

Definition (Automata Based Computation Method)

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

$$\mathcal{A}_\varphi \quad \otimes \quad \overline{h(\mathcal{A}_{\varphi'})}$$

- Intuition: $\mathcal{A}_{\varphi'}$ captures the $\langle \mathbf{H}, \mathbf{T}\rangle$ satisfying $\mathbf{H} < \mathbf{T}$
- We use the φ^* translation and force non-LTL models.
  Example: if $\varphi = \Diamond up$ then

  $\varphi' = \Diamond up' \wedge \Box(up' \rightarrow up) \wedge \Diamond(up \wedge \neg up')$
- Operation $h(\mathcal{A}_{\varphi'})$ filters out the auxiliary atoms p'
- Büchi automata are closed w.r.t. complementation and intersection

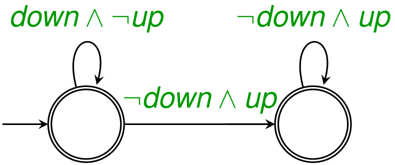
Example

$up \vee down.$

$\square\,(up \wedge \neg\circ down \rightarrow \circ up)\,.$

$\square\,(down \wedge \neg\circ up \rightarrow \circ down)\,.$

$\square\,(up \vee \neg up)$



Example

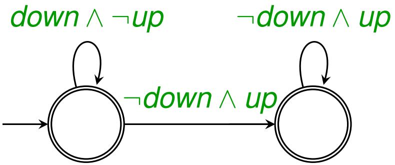
$up \vee down.$

$\Box\,(up \wedge \neg \circ down \rightarrow \circ up)\,.$

$\Box\,(down \wedge \neg \circ up \rightarrow \circ down)\,.$

$\Box\,(up \vee \neg up)$

$\Diamond\Box up \rightarrow \Box\ stuck.$



Example

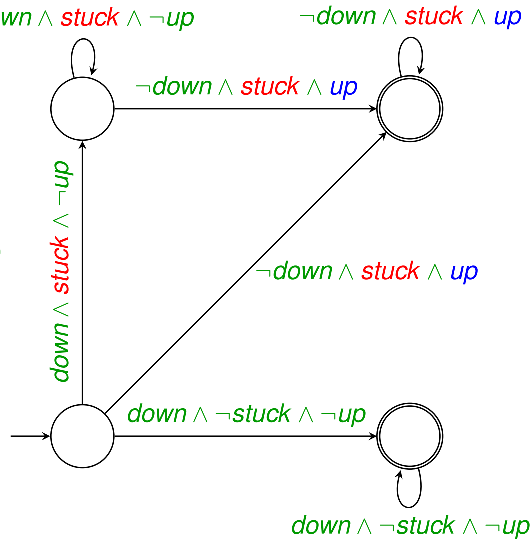
$up \vee down.$

$\Box\,(up \wedge \neg\circ down \rightarrow \circ up)$

$\Box\,(down \wedge \neg\circ up \rightarrow \circ down)$

$\Box\,(up \vee \neg up)$

$\Diamond\Box up \rightarrow \Box stuck$



Finite traces and past operators

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

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

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

- 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

- Adding past operators to LTL: same expressiveness but

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

- Adding past operators to LTL: same expressiveness but exponentially more succint [Markey 03]

Finite traces and past operators

Past	$\bullet$	for *previous*	*Future*	$\circ$	for *next*
	S	for *since*		**U**	for *until*
	T	for *trigger*		**R**	for *release*
				W	for *while*

Finite traces and past operators

Past	$\bullet$	for *previous*		*Future*	$\circ$	for *next*
	$\mathbf{S}$	for *since*			$\mathbf{U}$	for *until*
	$\mathbf{T}$	for *trigger*			$\mathbf{R}$	for *release*
					$\mathbf{W}$	for *while*

$\blacksquare\varphi$	$\stackrel{def}{=}$	$\bot\,\mathbf{T}\,\varphi$	*always before*	$\Box\varphi$	$\stackrel{def}{=}$	$\bot\,\mathbf{R}\,\varphi$	*always after*
$\blacklozenge\varphi$	$\stackrel{def}{=}$	$\top\,\mathbf{S}\,\varphi$	*eventually before*	$\Diamond\varphi$	$\stackrel{def}{=}$	$\top\,\mathbf{U}\,\varphi$	*eventually after*
$\mathbf{I}$	$\stackrel{def}{=}$	$\neg\bullet\top$	*initial*	$\mathbf{F}$	$\stackrel{def}{=}$	$\neg\top$	*final*
$\widehat{\bullet}\varphi$	$\stackrel{def}{=}$	$\bullet\varphi\vee\mathbf{I}$	*weak previous*	$\widehat{\circ}\varphi$	$\stackrel{def}{=}$	$\varphi\vee\mathbf{F}$	*weak next*

Finite traces and past operators

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

α	$M, i \models \alpha$ when ...
$\circ\varphi$	$i + 1 < \lambda$ and $(M, i{+}1) \models \varphi$
$\widehat{\circ}\varphi$	$i + 1 = \lambda$ or $(M, i{+}1) \models \varphi$
φ **U** ψ	$\exists j : i \leq j < \lambda,\quad M, j \models \psi$ and $\forall k$ s.t. $i \leq k < j,\quad M, k \models \varphi$
$\bullet\varphi$	$i > 0$ and $(M, i{-}1) \models \varphi$
$\widehat{\bullet}\varphi$	$i = 0$ or $(M, i{-}1) \models \varphi$
φ **S** ψ	$\exists j : 0 \leq j \leq i,\quad M, j \models \psi$ and $\forall k$ s.t. $j < k \leq i,\quad M, k \models \varphi$

Finite traces and past operators

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

α	$M, i \models \alpha$ when ...
$\circ\varphi$	$i+1 < \lambda$ and $(M, i{+}1) \models \varphi$
$\widehat{\circ}\varphi$	$i+1 = \lambda$ or $(M, i{+}1) \models \varphi$
$\varphi \ \mathbf{U} \ \psi$	$\exists j : i \leq j < \lambda, \quad M, j \models \psi$ and $\forall k$ s.t. $i \leq k < j, \quad M, k \models \varphi$
$\bullet\varphi$	$i > 0$ and $(M, i{-}1) \models \varphi$
$\widehat{\bullet}\varphi$	$i = 0$ or $(M, i{-}1) \models \varphi$
$\varphi \ \mathbf{S} \ \psi$	$\exists j : 0 \leq j \leq i, \quad M, j \models \psi$ and $\forall k$ s.t. $j < k \leq i, \quad M, k \models \varphi$

- When $\lambda = \omega$ we get (infinite-traces) TEL as before

Normal form

- 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 = $\{a, \neg a, \bullet a, \neg\bullet a \mid a \in Atoms\}$

Definition (Temporal rule)

A temporal rule is either:

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

where $B = b_1 \wedge \cdots \wedge b_n$ with $n \geq 0$, $A = a_1 \vee \cdots \vee a_m$ 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

Normal form

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

Definition (Temporal rule)

A temporal rule is either:

- an initial rule $B \to A$
- a dynamic rule $\widehat{\circ}\,\square(B \to A)$
- a final rule $\square(\,\mathbf{F} \to (B \to A)\,)$ when traces are finite

where $B = b_1 \wedge \cdots \wedge b_n$ with $n \geq 0$, $A = a_1 \vee \cdots \vee a_m$ 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.

Syntactic Fragment

- An interesting fragment are present-centered programs
  - initial rule $B \rightarrow A$
  - dynamic rule $\widehat{\circ}\,\square(B \rightarrow A)$
  - final rule $\square(\,\mathbf{F} \rightarrow (B \rightarrow A)\,)$

  present-centered = A does not contain temporal operators

Syntactic Fragment

- An interesting fragment are present-centered programs
  - initial rule $B \rightarrow A$
  - dynamic rule $\widehat{\circ}\,\Box(B \rightarrow A)$
  - final rule $\Box(\,\mathbf{F} \rightarrow (B \rightarrow A)\,)$

  present-centered = A does not contain temporal operators

- Example of present-centered program:

	$up \vee down$	Initially
$\widehat{\circ}\Box(\bullet up \wedge \neg down \;\rightarrow$	$up)$	Inertia
$\widehat{\circ}\Box(\bullet down \wedge \neg up \;\rightarrow$	$down)$	Inertia
	$\Box(up \vee \neg up)$	Choice

Syntactic Fragment

- Tool `telingo` [C_, Kaminski, Morkisch & Schaub 19]
  Temporal extension of ASP solver `clingo`

Syntactic Fragment

- 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}.
```

Syntactic Fragment

- We can use a more general syntactic fragment past-future = $\alpha \rightarrow \beta$ where
  - α may only contain past operators
  - β may only contain future operators

  and none of them contains $\rightarrow$

Syntactic Fragment

- We can use a more general syntactic fragment past-future = $\alpha \rightarrow \beta$ where
  - α may only contain past operators
  - β may only contain future operators

  and none of them contains $\rightarrow$
- Example: the integrity constraint

$$shoot \wedge \blacksquare unloaded \wedge \bullet\blacklozenge shoot \rightarrow \bot$$

  can be expressed in `telingo` as:

Syntactic Fragment

- We can use a more general syntactic fragment past-future = $\alpha \to \beta$ where
  - α may only contain past operators
  - β may only contain future operators

  and none of them contains $\to$
- Example: the integrity constraint

$$\mathit{shoot} \wedge \blacksquare \mathit{unloaded} \wedge \bullet\blacklozenge \mathit{shoot} \to \bot$$

can be expressed in `telingo` as:

```
:- shoot, &tel { <* unloaded & < <? 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_f to the syntax of Linear Dynamic Logic (LDL) [De Giacomo & Vardi 13]
- DEL = LDL + ASP. Example:

  $\bot \neg \leftarrow \langle\ ((up^* + down^*); ready?; serve\ )^*; wait^* \rangle \mathbf{F}$

  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:

  $\bot \neg \leftarrow \langle\ (\ (up^* + down^*); ready?; serve\ )^*; wait^* \rangle \mathbf{F}$

  elevator moving in a unique direction until the call is served
- [C_ et al, ECAI 20] LDL operators implemented in `telingo` (only in constraints)

```
#program initial.
:- not &del{ *( (*up + *down) ;; ?ready ;; serve)
               ;; *wait .>? &final }.
```

Beyond LTL

- ☞ In [C_ et al, ICLP 20] (🕘Tomorrow 18:15) we introduce metric operators

$$\Box(red \wedge green \rightarrow \bot)$$
$$\Box(\neg green \rightarrow red)$$
$$\Box(push \rightarrow \Diamond_3 \Box_4 green)$$

The traffic light is red by default

Beyond LTL

- ☞ In [C_ et al, ICLP 20] (🕘Tomorrow 18:15) we introduce metric operators

$$\Box(red \wedge green \rightarrow \bot)$$
$$\Box(\neg green \rightarrow red)$$
$$\Box(push \rightarrow \Diamond_3 \Box_4 green)$$

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] (🕘Tomorrow 18:15) we introduce metric operators

$$\Box(red \wedge green \rightarrow \bot)$$
$$\Box(\neg green \rightarrow red)$$
$$\Box(push \rightarrow \Diamond_3 \Box_4 green)$$

  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).

Conclusions

☞ 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