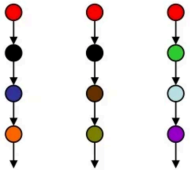
Linear Temporal Logic on Finite Traces as a Specification Language

Giuseppe De Giacomo

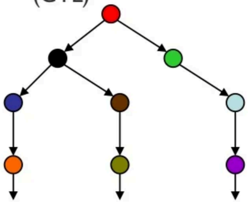
Motivation: AI

Temporal logic

- Linear Time
  - Every moment has a unique successor
  - Infinite sequences (words)
  - Linear Time Temporal Logic (LTL)



- Branching Time
  - Every moment has several successors
  - Infinite tree
  - Computation Tree Logic (CTL)



Courtesy of Carlo Gezzi

Motivation: AI

Artificial Intelligence and in particular the Knowledge Representation and Planning community well aware of temporal logics since a long time:

- Temporally extended goals [BacchusKabanza96]
- Temporal constraints on trajectories [GereviniHslumLongSaettiDimopoulos09 - PDDL3.0 2009]
- Declarative control knowledge on trajectories [BaierMcIlraith06]
- Procedural control knowledge on trajectories [BaierFrizMcIlraith07]
- Temporal specification in planning domains [CalvaneseDeGiacomoVardi02]
- Planning via model checking
  - Branching time (CTL)[CimattiGiunchigliaGiunchigliaTraverso97]
  - Linear time (LTL) [DeGiacomoVardi99]

Motivation: AI

Temporal extended goals and constraints in AI

Foundations borrowed from temporal logics studied in CS, in particular:
Linear Temporal Logic (LTL) [Pnueli77].

However:

- Often, LTL is interpreted on finite trajectories/traces.
- Often, distinction between interpreting LTL on infinite or on finite traces is blurred.

- Temporally extended goals [BacchusKabanza96] - infinite/finite
- Temporal constraints on trajectories [GereviniHslumLongSaettiDimopoulos09 - PDDL3.0 2009] - finite
- Declarative control knowledge on trajectories [BaierMcIlraith06] - finite
- Procedural control knowledge on trajectories [BaierFrizMcIlraith07] - finite
- Temporal specification in planning domains [CalvaneseDeGiacomoVardi02] - infinite
- Planning via model checking - infinite
    - Branching time (CTL) [CimattiGiunchigliaGiunchigliaTraverso97]
    - Linear time (LTL) [DeGiacomoVardi99]

Motivation: AI

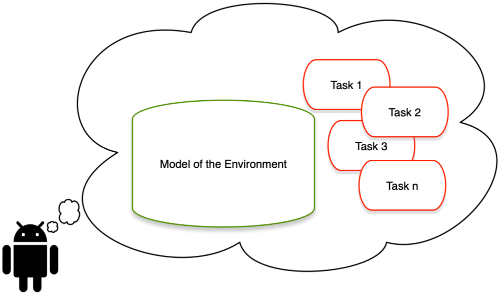
Planning in AI:

- Is all about having a task specification or "goal" and producing a "plan" (or strategy or policy) to satisfy the task in the environment model.

- Which tasks?
  - A **task that terminates!**
  - Typically, just reaching a certain state in the environment

Why tasks that terminate?

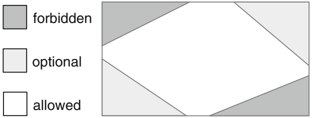
- Because it is the agent that is planning/reasoning
- If the task would not terminate, the agent would be stuck into doing the same task forever
- But then, why bother with equipping it with a model of the environment and of the task at all?
- Note it is the agent, NOT the designer, who has such a model



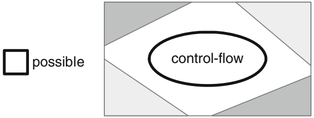
Motivation: BPM

Business Process Management community has proposed a declarative approach to business process modeling based on LTL on finite traces: DECLARE

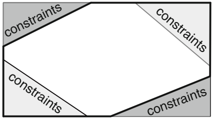
Basic idea: Drop explicit representation of processes, and LTL formulas specify the allowed finite traces. [VanDerAalstPesic06] [PesicBovsnavkiDraganVanDerAalst10].



(a) forbidden, optional and allowed in business processes



(b) procedural workflow



(c) declarative workflow

LTL over finite traces

LTL$_f$: the language (in symbols)

Same syntax as standard LTL but interpreted over finite traces

$$\varphi ::= A \mid \quad \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \quad \bigcirc\varphi \mid \bullet\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi_1 \,\mathcal{U}\, \varphi_2$$

- A: atomic propositions
- $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, $\varphi_1 \supset \varphi_2$: boolean connectives
- $\bigcirc\varphi$: "(next step exists and) at next step (of the trace) φ holds"
- $\bullet\varphi$: "if next step exists then at next step φ holds" *(weak next)* ($\bullet\varphi \;\equiv\; \neg\bigcirc\neg\varphi$)
- $\Diamond\varphi$: "φ will eventually hold" ($\Diamond\varphi \;\equiv\; \mathtt{true}\,\mathcal{U}\,\varphi$)
- $\Box\varphi$: "from current till last instant φ will always hold" ($\Box\varphi \;\equiv\; \neg\Diamond\neg\varphi$)
- $\varphi_1 \,\mathcal{U}\, \varphi_2$:"eventually φ_2 holds, and φ_1 holds until φ_2 does"

LTL$_f$: the language (in words)

Note: we do not need fancy symbols we can use english words instead:

$$\varphi ::= A \mid \quad \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \supset \varphi_2 \mid \quad \mathit{next}\,\varphi \mid \mathit{wnext}\,\varphi \mid \mathit{eventually}\,\varphi \mid \mathit{always}\,\varphi \mid \varphi_1 \;\mathit{until}\,\varphi_2$$

LTL over finite traces

In symbols

$\Diamond A$	"eventually A"	*reachability*
$\Box A$	"always A"	*safety*
$\Box(A \supset \Diamond B)$	"always if A then eventually B"	*reactiveness*
$A \,\mathcal{U}\, B$	"A until B"	*strong until – stronger than English until*
$A \,\mathcal{U}\, B \vee \Box A$	"A until B"	*weak until – just like English until*

In words

eventually A	"eventually A"	*reachability*
always A	"always A"	*safety*
always$(A \supset$ *eventually* $B)$	"always if A then eventually B"	*reactiveness*
A *until* B	"A until B"	*strong until – stronger than English until*
A *until* $B \vee$ *always* A	"A until B"	*weak until – just like English until*

LTL$_f$ Semantics

Finite Traces

The semantics of LTL$_f$ is given in terms of finite traces denoting a finite sequence of consecutive instants of time.

- Finite traces are finite words π over the alphabet of $2^{\mathcal{P}}$, i.e., as alphabet we have all the possible propositional interpretations of the propositional symbols in $\mathcal{P}$.
- We denote the length of a trace π as $length(\pi)$.
- We denote the positions, i.e. instants, on the trace as π, i with $0 \leq i \leq last$, where $last = length(\pi) - 1$ is the last element of the trace.

LTL_f Semantics

LTL_f Semantics

Given a finite trace π, we inductively define when an LTL_f formula φ is true at an instant i (for $0 \leq i \leq last$), in symbols $\pi, i \models \varphi$, as follows:

- $\pi, i \models A$, for $A \in \mathcal{P}$ iff $A \in \pi(i)$.
- $\pi, i \models \neg\varphi$ iff $\pi, i \not\models \varphi$.
- $\pi, i \models \varphi_1 \wedge \varphi_2$ iff $\pi, i \models \varphi_1$ and $\pi, i \models \varphi_2$.
- $\pi, i \models \bigcirc\varphi$ iff $i{+}1 \leq last$ and $\pi, i{+}1 \models \varphi$.
- $\pi, i \models \bullet\varphi$ iff $i{+}1 \leq last$ implies $\pi, i{+}1 \models \varphi$.
- $\pi, i \models \Diamond\varphi$ iff for some j such that $i \leq j \leq last$, we have $\pi, j \models \varphi$.
- $\pi, i \models \Box\varphi$ iff for all j such that $i \leq j \leq last$, we have $\pi, j \models \varphi$.
- $\pi, i \models \varphi_1 \,\mathcal{U}\, \varphi_2$ iff for some j such that $i \leq j \leq last$, we have $\pi, j \models \varphi_2$ and for all k, $i \leq k < j$, we have $\pi, k \models \varphi_1$.

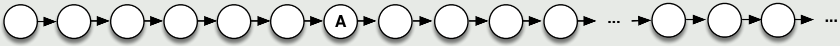
LTL$_f$ Examples

Example

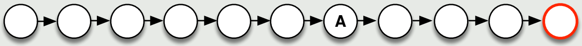
Consider the following formula:

$$\Diamond A$$

- On infinite traces:



- On finite traces:



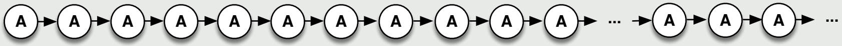
LTL$_f$ Examples

Example

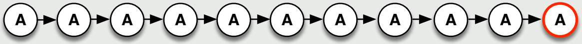
Consider the following formula:

$$\Box A$$

- On infinite traces:



- On finite traces:



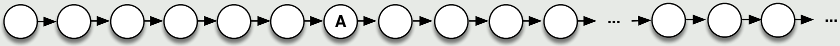
LTL_f Examples

Example

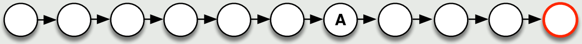
Consider the following formula:

$$\Diamond \bigcirc A$$

- On infinite traces:



- On finite traces:

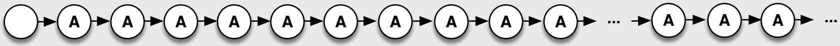


Example

Consider the following formula:

$$\Box \bigcirc A$$

- On infinite traces:



- On finite traces:

None!!!

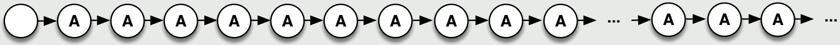
LTL$_f$ Examples

Example

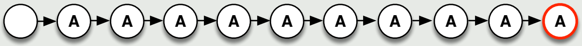
Consider the following formula:

$$\Box \bullet A$$

- On infinite traces (in LTL $\bullet A$ must be replaced by $\neg \bigcirc \neg A$ which is equivalent to $\bigcirc A$):



- On finite traces:



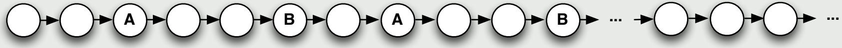
LTL$_f$ Examples

Example

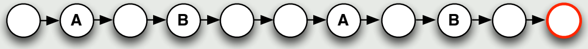
Consider the following formula:

$$\Box(A \supset \Diamond B)$$

- On infinite traces:



- On finite traces:



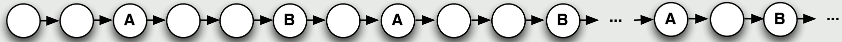
LTL_f Examples

Example

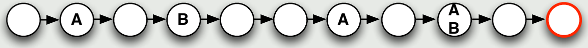
Consider the following formula:

$$\Box(A \supset \Diamond B) \land \Box(B \supset \Diamond A)$$

- On infinite traces:



- On finite traces:



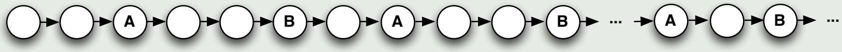
LTL$_f$ Examples

Example

Consider the following formula:

$$\Box(A \supset \bigcirc\Diamond B) \wedge \Box(B \supset \bigcirc\Diamond A)$$

- On infinite traces:



- On finite traces:

A **and** B **cannot appear at all!**

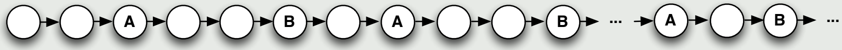
LTL$_f$ Examples

Example

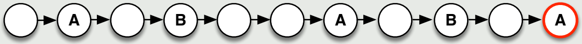
Consider the following formula:

$$\Box(A \supset \bullet\Diamond B) \wedge \Box(B \supset \bullet\Diamond A)$$

- On infinite traces (in LTL $\bullet A$ must be replaced by $\neg\circ\neg A$ which is equivalent to $\circ A$):



- On finite traces:



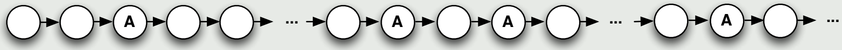
LTL$_f$ Examples

Example ("Fairness" does not make sense in LTL$_f$)

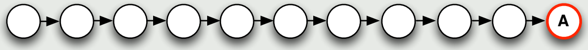
$$\Box\Diamond A$$

for any point in the trace there is a point later where A holds ("Fairness").

- On infinite traces:



- On finite traces becomes equivalent to last point in the trace satisfies A



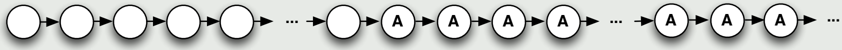
LTL$_f$ Examples

Example ("Stability" does not make sense in LTL$_f$)

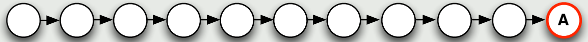
$$\Diamond \Box A$$

there exists a point in the trace such that from then on φ holds ("Stability").

- On infinite traces:



- On finite traces becomes equivalent to last point in the trace satisfies φ



*In other words, no direct nesting of **eventually** and **always** connectives is meaningful in* LTL$_f$*, this contrast what happens in* LTL *of infinite traces.*

Capturing STRIPS

Example (Capturing STRIPS Planning as LTL_f SAT)

- For each action $A \in Act$ with precondition φ and effects $\bigwedge_{F \in Add(A)} F \wedge \bigwedge_{F \in Del(A)} \neg F$
  - $\Box(\bigcirc A \supset \varphi)$: if next action A has occurred (denoted by a proposition A) then now precondition φ must be true;
  - $\Box(\bigcirc A \supset \bigcirc(\bigwedge_{F \in Add(A)} F \wedge \bigwedge_{F \in Del(A)} \neg F))$: when A occurs, its effects are true;
  - $\Box(\bigcirc A \supset \bigwedge_{F \notin Add(A) \cup Del(A)} (F \equiv \bigcirc F))$: everything not in add or delete list, remains unchanged.
- At every step one and only one action is executed: $\Box((\bigvee_{A \in Act} A) \wedge (\bigwedge_{A_i, A_j \in Act, A_i \neq A_j} A_i \supset \neg A_j))$.
- Initial situation is described as the conjunction of propositions $Init$ that are true/false at the beginning of the trace: $\bigwedge_{F \in Init} F \wedge \bigwedge_{F \notin Init} \neg F$.
- Finally goal φ_g eventually holds: $\Diamond \varphi_g$.

Thm: A plan exists iff the LTL_f formula is SAT.

Capturing SitCalc

Example (Propositional SitCalc Basic Action Theories in LTL_f)

- Successor state axiom (instantiated for each action A) $F(do(A,s)) \equiv \varphi^+(s) \vee (F(s) \wedge \neg\varphi^-(s))$ can be fully captured:

$$\Box(\bigcirc A \supset (\bigcirc F \equiv \varphi^+ \vee F \wedge \neg\varphi^-).$$

- Precondition axioms $Poss(A,s) \equiv \varphi_A(s)$ can only be captured in the part saying "if A happens then its precondition must be true":

$$\Box(\bigcirc A \supset \varphi_A).$$

*The part saying "if the precondition φ_A holds then action A is **possible**" cannot be expressed in linear time formalisms, since they talk about traces that actually happen not the ones that are possible.*

Questions

- Does $\neg\bigcirc\varphi \equiv \bigcirc\neg\varphi$ holds as in LTL$_{f}$?
- **No**, $\neg\bigcirc\varphi \equiv \bullet\neg\varphi$ holds in LTL$_f$!

- Can you denote the last element of the trace in LTL$_f$?
- **Yes** $Last \equiv \bullet false$

- Can you talk about the first element in the trace?
- **Yes** e.g. A

- Can you talk about the last element in the trace?
- **Yes** e.g. $\Diamond(Last \wedge A)$

- Is $\Box\bigcirc true$ equivalent to $true$ or to $false$?
- It is equivalent to $false$.

- Is $\Box\bullet true$ equivalent to $true$or to $false$?
- It is equivalent to $true$.

Other Examples

name of template	*LTL semantics*
responded existence(A, B)	$\Diamond A \Rightarrow \Diamond B$
co-existence(A, B)	$\Diamond A \Leftrightarrow \Diamond B$
response(A, B)	$\Box(A \Rightarrow \Diamond B)$
precedence(A, B)	$(\neg B\ U A) \vee \Box(\neg B)$
succession(A, B)	*response*$(A, B) \wedge$ *precedence*(A, B)
alternate response(A, B)	$\Box(A \Rightarrow \bigcirc(\neg A\ U B))$
alternate precedence(A, B)	*precedence*$(A, B) \wedge$ $\Box(B \Rightarrow \bigcirc(\mathit{precedence}(A, B)))$
alternate succession(A, B)	*alternate response*$(A, B) \wedge$ *alternate precedence*(A, B)
chain response(A, B)	$\Box(A \Rightarrow \bigcirc B)$
chain precedence(A, B)	$\Box(\bigcirc B \Rightarrow A)$
chain succession(A, B)	$\Box(A \Leftrightarrow \bigcirc B)$

name of template	*LTL semantics*
not co-existence(A, B)	$\neg(\Diamond A \wedge \Diamond B)$
not succession(A, B)	$\Box(A \Rightarrow \neg(\Diamond B))$
not chain succession(A, B)	$\Box(A \Rightarrow \bigcirc(\neg B))$

name of template	*LTL semantics*
$existence(1, A)$ $existence(2, A)$ ... $existence(n, A)$	$\Diamond A$ $\Diamond(A \wedge \bigcirc(existence(1, A)))$ ... $\Diamond(A \wedge \bigcirc(existence(n-1, A)))$
$absence(A)$	$\neg existence(1, A)$
$absence(2, A)$ $absence(3, A)$ ... $absence(n+1, A)$	$\neg existence(2, A)$ $\neg existence(3, A)$ ... $\neg existence(n+1, A)$
$init(A)$	A

Weak Until and Release in LTL_f

Weak Until

Weak Until, denoted by $\varphi \mathcal{W} \psi$ says that "φ holds until ψ holds, however it is fine for ψ not to hold at all, and in that case φ holds forever". Note this is the typical interpretation of the word "until" in English. Formally it is defined as:

$$\varphi_1 \mathcal{W} \varphi_2 \doteq (\varphi_1 \,\mathcal{U}\, \varphi_2) \vee \Box \varphi_1$$

Release

Release denoted by $\varphi \mathcal{R} \psi$ says that "φ releases ψ from holding forever". It can be defined as:

$$\varphi_1 \mathcal{R} \varphi_2 \doteq \varphi_1 \mathcal{W} (\varphi_1 \wedge \varphi_2)$$

The following holds:

- $\varphi_1 \mathcal{R} \varphi_1 \equiv \neg(\neg \varphi_1 \,\mathcal{U}\, \neg \varphi_2)$
- it also holds that
  $\varphi_1 \mathcal{U} \varphi_2 \equiv \neg(\neg \varphi_1 \mathcal{R} \neg \varphi_2)$

(Release is dual of Until)

Duals in LTL_f

- $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$ and equivalently $\varphi_1 \wedge \varphi_2 \equiv \neg(\neg\varphi_1 \vee \neg\varphi_2)$
- $\bullet\varphi \equiv \neg\circ\neg\varphi$ and equivalently $\circ\varphi \equiv \neg\bullet\neg\varphi$
- $\Box\varphi \equiv \neg\Diamond\neg\varphi$ and equivalently $\Diamond\varphi \equiv \neg\Box\neg\varphi$
- $\varphi_1 \mathcal{R} \varphi_1 \equiv \neg(\neg\varphi_1 \,\mathcal{U}\, \neg\varphi_2)$ and equivalently $\varphi_1 \,\mathcal{U}\, \varphi_1 \equiv \neg(\neg\varphi_1 \mathcal{R} \neg\varphi_2)$

Negation Normal Form for LTL_f

NNF

Negation Normal Form for LTL_f: for $a \in AP$

$$\varphi ::= \mathsf{true} \mid \mathsf{false} \mid A \mid \neg A \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \bigcirc\varphi \mid \bullet\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi\,\mathcal{U}\,\varphi \mid \varphi\mathcal{R}\varphi$$

Theorem

Each LTL_f formula φ admits an equivalent in NNF, denoted $nnf(\varphi)$, which is obtained in linear time in the size formula by pushing the negation all in exploiting duals.

Exploit duals through the follow equivalences to push negation all the way in:

- $\neg\neg\varphi \equiv \varphi$
- $\neg(\varphi_1 \land \varphi_2) \equiv \neg\varphi_1 \lor \neg\varphi_2$
- $\neg(\varphi_1 \lor \varphi_2) \equiv \neg\varphi_1 \land \neg\varphi_2$
- $\neg\bigcirc\varphi \equiv \bullet\neg\varphi$
- $\neg\bullet\varphi \equiv \bigcirc\neg\varphi$
- $\neg\Diamond\varphi \equiv \Box\neg\varphi$
- $\neg\Box\varphi \equiv \Diamond\neg\varphi$
- $\neg(\varphi_1\,\mathcal{U}\,\varphi_1) \equiv \neg\varphi_1\mathcal{R}\neg\varphi_2$
- $\neg(\varphi_1\mathcal{R}\varphi_1) \equiv \neg\varphi_1\,\mathcal{U}\,\neg\varphi_2$

Fixpoint Equivalences in LTL_f

Introduced since the early days of LTL in CS, for connection with fixpoint theory and tableaux expansion rules, [GabbayPnueliShelahStavi80],[Manna82],[Emerson90]

- $\Diamond\varphi \equiv \varphi \vee \bigcirc(\Diamond\varphi)$ –then choose lfp
- $\Box\varphi \equiv \varphi \wedge \bullet(\Box\varphi)$ –then choose gfp *(Note: in LTL_f, differently from LTL, $\Box\varphi \equiv \varphi \wedge \bigcirc(\Box\varphi)$ does **not** hold.)*
- $\varphi_1 \,\mathcal{U}\, \varphi_2 \equiv \varphi_2 \vee (\varphi_1 \wedge \bigcirc(\varphi_1 \,\mathcal{U}\, \varphi_2))$ –then choose lfp
- $\varphi_1 \mathcal{R} \varphi_2 \equiv \varphi_2 \wedge (\varphi_1 \vee \bullet(\varphi_1 \mathcal{R} \varphi_2))$ –then choose gfp

Fixpoint Equivalences in LTL_f and "next normal form"

By recursively applying fixpoint equivalences, considering as base case propositions and formulas prefixed with $\bigcirc$ or $\bullet$, i.e.:

$$
\begin{array}{rcl}
nextNF(A) & = & A \\
nextNF(\bigcirc\varphi) & = & \bigcirc\varphi \\
nextNF(\bullet\varphi) & = & \bullet\varphi \\
nextNF(\neg\varphi) & = & \neg nextNF(\varphi) \\
nextNF(\varphi_1 \wedge \varphi_2) & = & nextNF(\varphi_)\wedge nextNF(\varphi_2) \\
nextNF(\varphi_1 \vee \varphi_2) & = & nextNF(\varphi_)\vee nextNF(\varphi_2)
\end{array}
\qquad
\begin{array}{rcl}
nextNF(\Diamond\varphi & = & nextNF(\varphi)\vee\bigcirc(\Diamond\varphi) \\
nextNF(\Box\varphi) & = & nextNF(\varphi)\wedge\bullet(\Box\varphi) \\
nextNF(\varphi_1\,\mathcal{U}\,\varphi_2) & = & nextNF(\varphi_2)\vee(nextNF(\varphi_1)\wedge\bigcirc(\varphi_1\,\mathcal{U}\,\varphi_2)) \\
nextNF(\varphi_1\mathcal{R}\varphi_2) & = & nextNF(\varphi_2)\wedge(nextNF(\varphi_1)\vee\bullet(\varphi_1\mathcal{R}\varphi_2))
\end{array}
$$

we get that every formula φ in LTL_f (or LTL, LDL_f, Pure Past LTL) can be decomposed is equivalent to a formula of the form

$$\varphi \equiv Bool(A, \bigcirc\varphi, \bullet\varphi)$$

that is φ gets partitioned into a part that to be evaluated NOW and a part that to be evaluated NEXT.

This observation is at the base of many results, including, e.g.:

- translation of LTL into alternating automata [Vardi95]
- Bacchus&Kabanza's progression algorithm for LTL [BacchusKabanza96].
- Super-good algorithms for Pure-Past LTL [DeGiacomoFuggittiFavoritoRubin20],[BonassiDeGiacomoFuggittiGereviniScala22].
- State-of-the-art symbolic tablaux algorithms implemente in BLACK for Pure-Past LTL [GeattiGiganteMontanari21]

Test yourself!

https://brown.co1.qualtrics.com/jfe/form/SV_38fUSW6EHtaB3My

Curtesy of Ben Greenman – https://users.cs.utah.edu/~blg/publications/publications.html#gpdzdkmnz-fm-2024