Iowa State University

Laboratory for Temporal Logic

Formalizing Time: Temporal Logics and the Challenge of Visualizing MLTL

Laura Gamboa Guzman

May 1, 2025

Temporal logic

- From the Stanford Encyclopedia of Philosophy[1]:

"Temporal Logic covers all formal approaches to representing and reasoning about time and temporal information."

- These formalisms rely on instant-based models of time, instead of interval-based, and follow a modal-style approach.

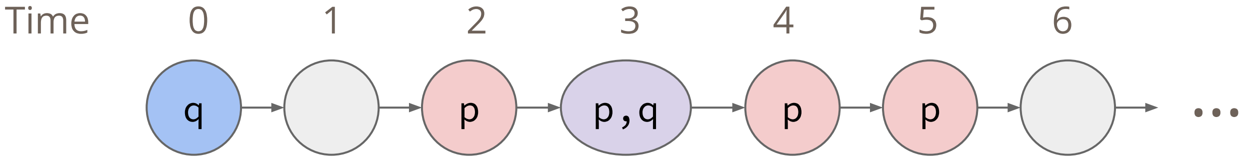
1. Goranko, Valentin and Antje Rumberg, "Temporal Logic", *The Stanford Encyclopedia of Philosophy* (Summer 2024 Edition), Edward N. Zalta & Uri Nodelman (eds.), https://plato.stanford.edu/archives/sum2024/entries/logic-temporal/.

LTL - Linear Temporal Logic

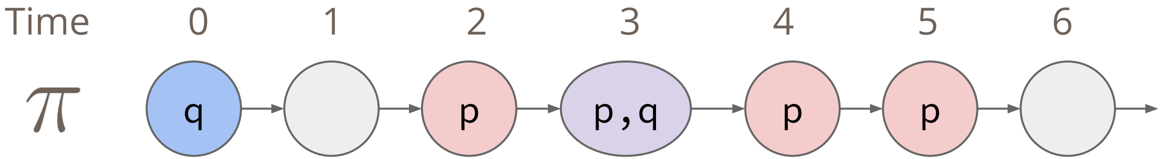
- The syntax is given by the following grammar, where p is any atomic proposition.

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid G\phi \mid F\phi \mid \phi U \psi \mid \phi R \psi \mid X\phi$$

- These formulas are interpreted over infinite time lines called traces, which are maps $\pi : \omega \rightarrow 2^{\mathcal{AP}}$.



LTL - Linear Temporal Logic



$$\pi, i \vDash p \text{ iff } p \in \pi(i)$$

$$\pi, i \vDash X\phi \text{ iff } \pi, i+1 \vDash \phi$$

$$\pi, i \vDash G\phi \text{ iff } \forall j \geq i \ (\pi, j \vDash \phi)$$

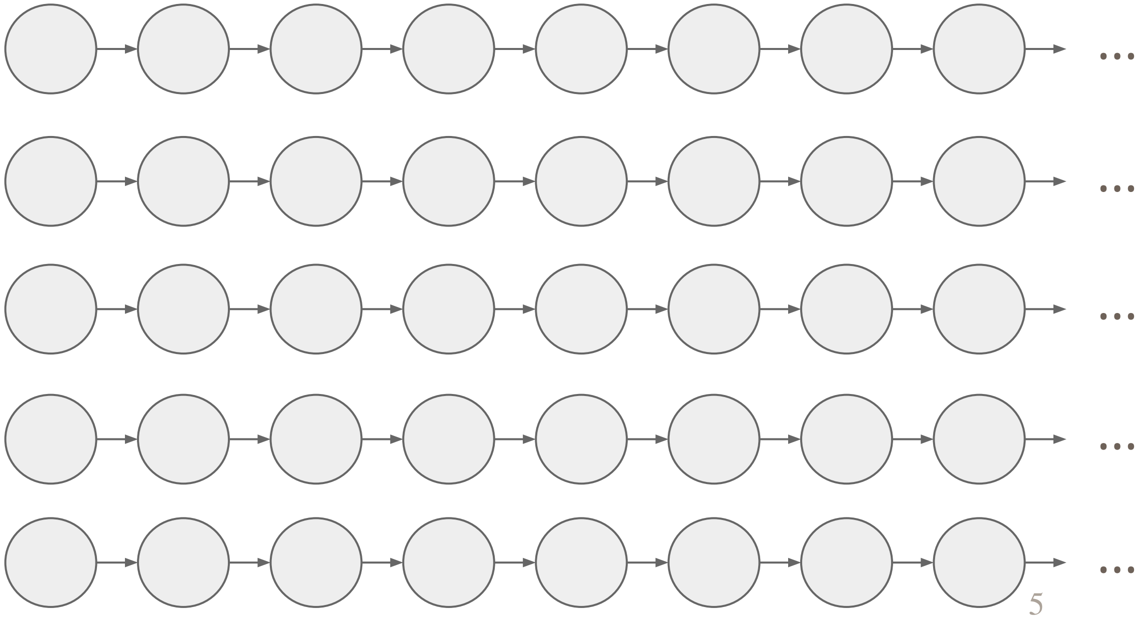
$$\pi, i \vDash F\phi \text{ iff } \exists j \geq i \ (\pi, j \vDash \phi)$$

$$\pi, i \vDash \phi U \psi \text{ iff } \exists j \geq i \ (\pi, j \vDash \psi \,\&\, \forall i \leq k < j \ (\pi, k \vDash \phi))$$

$$\pi, i \vDash \phi R \psi \text{ iff } \forall j \geq i \ (\pi, j \nvDash \psi \implies \exists k \ (i \leq k < j \,\&\, \pi, k \vDash \phi))$$

LTL - Linear Temporal Logic

Operator	Syntax	0	1	2	3	4	5	6	7
Next	$\mathbf{X}p$								
Globally	$\mathbf{G}p$								
in the **F**uture	$\mathbf{F}p$								
Until	$p\,\mathbf{U}q$								
Release	$p\,\mathbf{R}q$								



LTL expressiveness

- “Something bad (p) never happens”
- “It is always the case that something good (p) eventually happens”
- “At some point p will hold forever”

LTL expressiveness

- “Something bad (p) never happens” $\neg F p$
- “It is always the case that something good (p) eventually happens” GFp
- “At some point p will hold forever” FGp

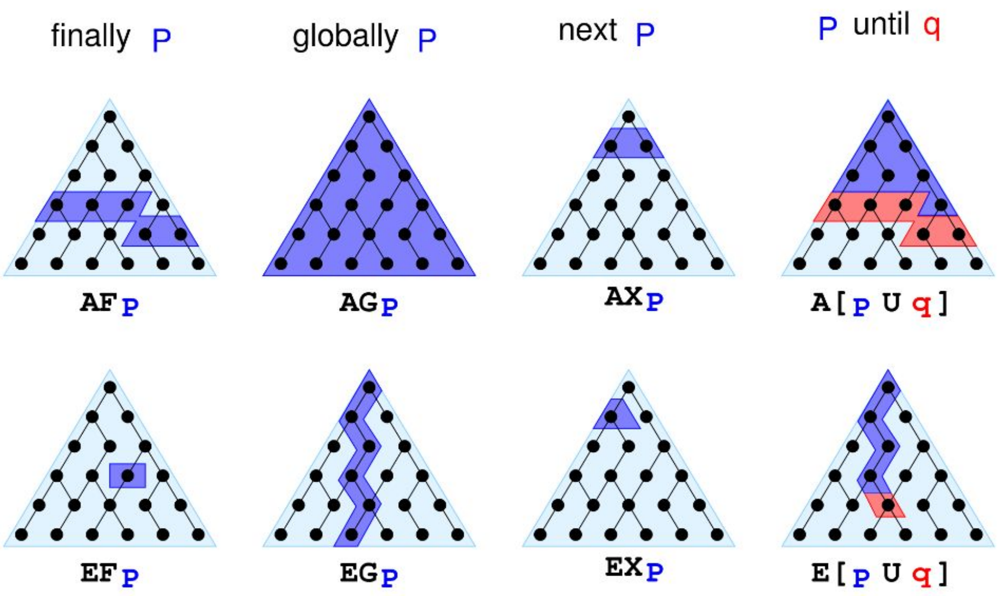
Fact: Propositional LTL over the naturals has exactly the expressive power of FO[<]. (Cabbay, Pnueli, Shelah, Stavi, 1980)

CTL - Computation Tree Logic

- Reasons about branching paths. The temporal connectives are always preceded by path quantifiers:

  A – for all paths,

  E – there exists a path.



CTL expressiveness

- “Something bad (p) never happens”
- “It is always the case that something good (p) eventually happens”
- “At some point p will hold forever”

CTL expressiveness

- "Something bad (p) never happens" $AG(\neg p)$
- "It is always the case that something good (p) eventually happens" $AG\,(AFp)$
- "At some point p will hold forever" Can't be done!

Fact: Although CTL is not less expressive than LTL, an example would be the formula $AG(EF(p))$, common CTL formulas have LTL equivalents. (M.Y. Vardi, Branching vs. linear time: Final showdown, in: TACAS, LNCS vol. 2031, Springer, 2001, pp. 122)

STL - Signal Temporal Logic

Signal temporal logic allows us to reason about signals with values in a continuous interval, allowing us to have predicates over real values and also reason about real time.

- STL predicates: Assume we have signals $x_1[t], x_2[t], \ldots, x_n[t]$, then atomic predicates are of the form

$$\mu = f(x_1[t], \ldots, x_n[t]) > 0$$

- STL formulas:

$$\phi ::= \top \mid \mu \mid \neg\phi \mid \phi \wedge \psi \mid \phi U_{[a,b]} \psi$$

- Semantics:

$$(x, t) \vDash \phi U \psi \text{ iff } \exists t' \in [t + a, t + b] \\ ((x, t') \vDash \psi \,\&\, \forall t'' \in [t, t'] \; ((x, t'') \vDash \phi))$$

STL - Signal Temporal Logic

Signal tempo
allows us to
about signals
values in a co
interval, allow
have predicate
real values and
reason about r

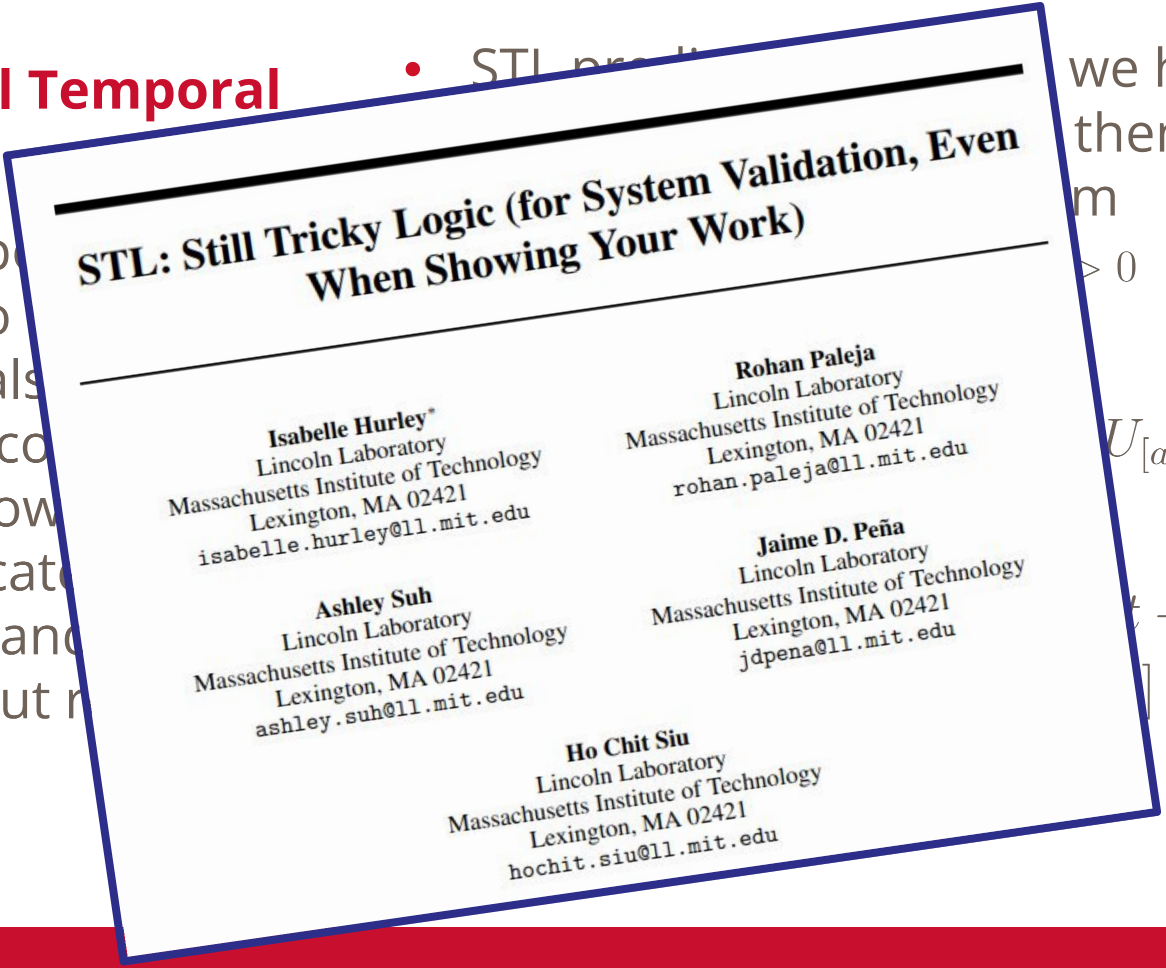
- STL predi... we have
then atomic
m

> 0

$U_{[a,b]}\psi$

$t + b]$

$] \; ((x, t'') \models \phi))$



STL: Still Tricky Logic (for System Validation, Even When Showing Your Work)

Isabelle Hurley*
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, MA 02421
isabelle.hurley@ll.mit.edu

Rohan Paleja
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, MA 02421
rohan.paleja@ll.mit.edu

Ashley Suh
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, MA 02421
ashley.suh@ll.mit.edu

Jaime D. Peña
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, MA 02421
jdpena@ll.mit.edu

Ho Chit Siu
Lincoln Laboratory
Massachusetts Institute of Technology
Lexington, MA 02421
hochit.siu@ll.mit.edu

MLTL - Mission-time Linear Temporal Logic

- In short: Finite version of LTL with (finite) interval bounds.

Operator	Syntax	0	1	2	3	4	5	6	7
Globally	$\mathbf{G}_{[2,5]}\, p$			p	p	p	p		
in the **F**uture	$\mathbf{F}_{[0,4]}\, p$					p			
Until	$p\, \mathbf{U}_{[1,6]}\, q$		p	p	p	p	q		
Release	$p\, \mathbf{R}_{[2,7]}\, q$			q	q	q	q	q	p, q

MLTL - Mission-time Linear Temporal Logic

- <u>In short</u>: Finite version of LTL with (finite) interval bounds.

$\pi \models \mathcal{F}_{[a,b]}\alpha$ iff $|\pi| > a$ and $\exists i \in [a,b]$ such that $\pi_i \models \alpha$

$\pi \models \mathcal{G}_{[a,b]}\alpha$ iff $|\pi| \leq a$ or $\forall i \in [a,b]\ \pi_i \models \alpha$

$\pi \models \alpha\ \mathcal{U}_{[a,b]}\beta$ iff $|\pi| > a$ and $\exists i \in [a,b]$ such that $\pi_i \models \beta$ and
$\quad \forall j \in [a, i-1]\ \pi_j \models \alpha$

$\pi \models \alpha\ \mathcal{R}_{[a,b]}\beta$ iff $|\pi| \leq a$ or $\forall i \in [a,b]\ \pi_i \models \beta$ or
$\quad \exists j \in [a, b-1]$ such that $\pi_j \models \alpha$ and $\forall a \leq k \leq j\ \pi_k \models \beta$

MLTL - expressiveness

- Technically speaking, MLTL is only as expressive as classical propositional logic!
- Computation length of a formula: it is the minimum length required for a trace to ensure that none of the intervals in are out of bounds.

$$\text{complen}(p_k) = \text{complen}(\neg p_k) = 1$$

$$\text{complen}(\alpha \wedge \beta) = \text{complen}(\alpha \vee \beta) = \max(\text{complen}(\alpha), \text{complen}(\beta))$$

$$\text{complen}(\mathcal{G}_{[a,b]}\alpha) = \text{complen}(\mathcal{F}_{[a,b]}\alpha) = b + \text{complen}(\alpha)$$

$$\text{complen}(\alpha \mathcal{U}_{[a,b]}\beta) = \text{complen}(\alpha \mathcal{R}_{[a,b]}\beta) = b + \max(\text{complen}(\alpha) - 1, \text{complen}(\beta))$$

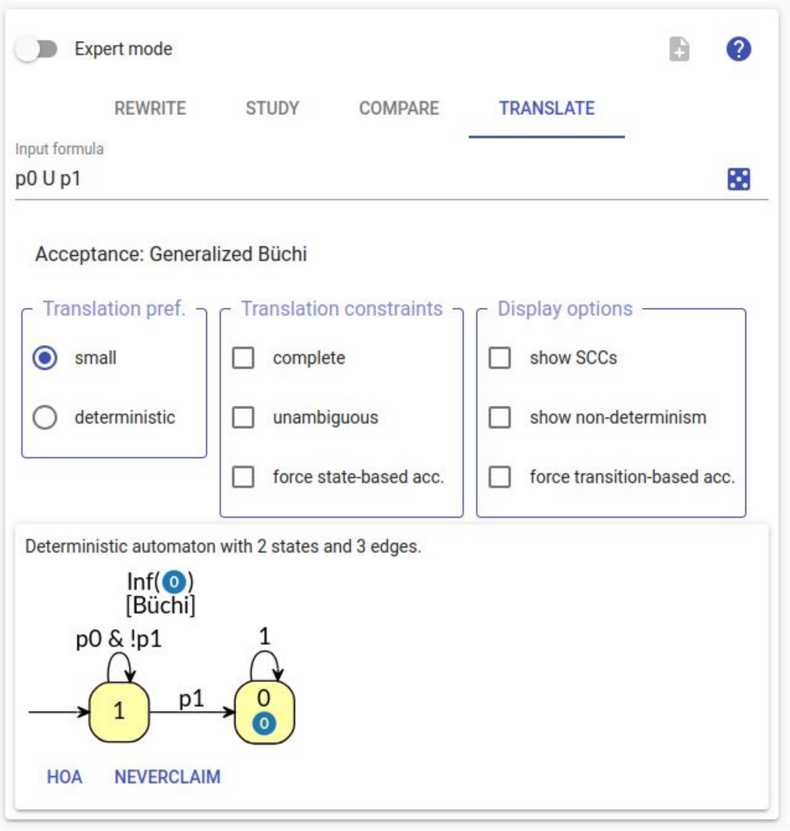
MLTL - expressiveness

- From now on, assume that $\mathcal{AP} = \{p_0, p_1, \ldots, p_{n-1}\}$.
- An MLTL formula ϕ over $\mathcal{AP}$ with computation length m can be encoded as a propositional formula over nm propositional variables.
- It can also be seen as a fragment of LTLf via the following map:

$$\phi U_{[a,b]} \psi \mapsto \bigvee_{i \in [a,b]} \left(X^i \psi \wedge \bigwedge_{j \in [a, i-1]} X^j \phi \right)$$

Visualizing Temporal Logics

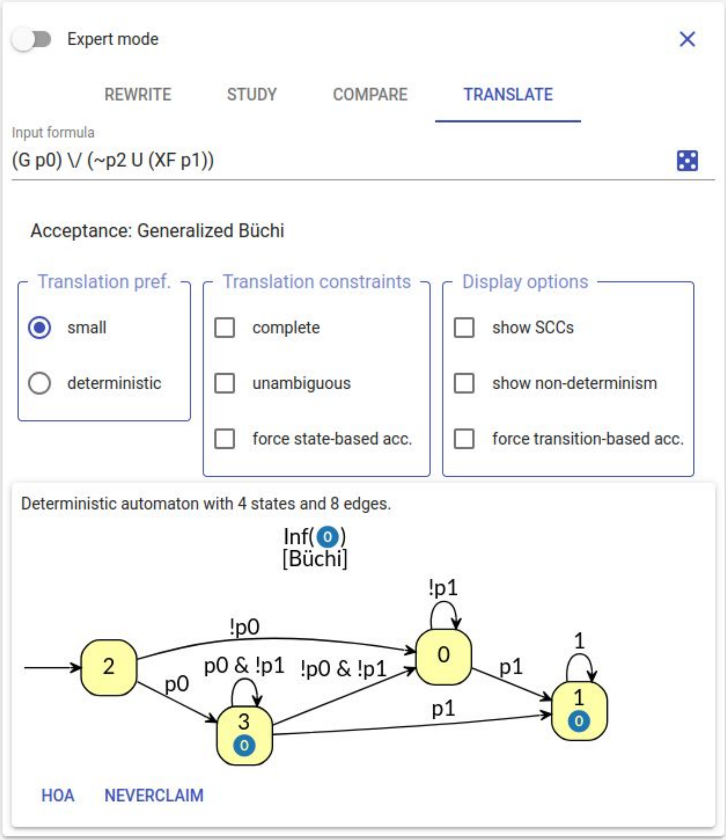
- In order to apply a formal technique to a system, designers need to **validate** the formula specifications.
- LTL: many algorithms have been developed to translate LTL formulas into Büchi Automata with similar size to the original formula.



https://spot.lre.epita.fr/index.html

Visualizing Temporal Logics

- In order to apply a formal technique to a system, designers need to **validate** the formula specifications.
- LTL: many algorithms have been developed to translate LTL formulas into Büchi Automata with similar size to the original formula.



https://spot.lre.epita.fr/index.html

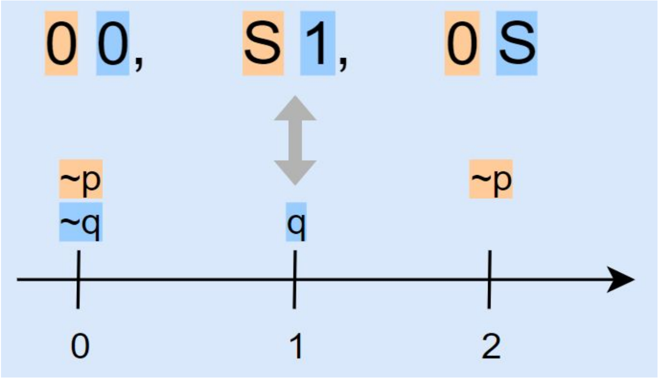
Visualizing MLTL

Goal 1: Visualize MLTL formulas, much like truth tables for boolean formulas.

p	q	~p	~q	p ∧ q	~(p ∧ q)	~p ∨ ~q
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

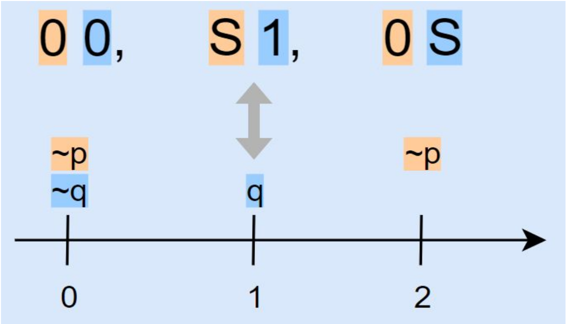
WEST: MLTL Regular Expressions

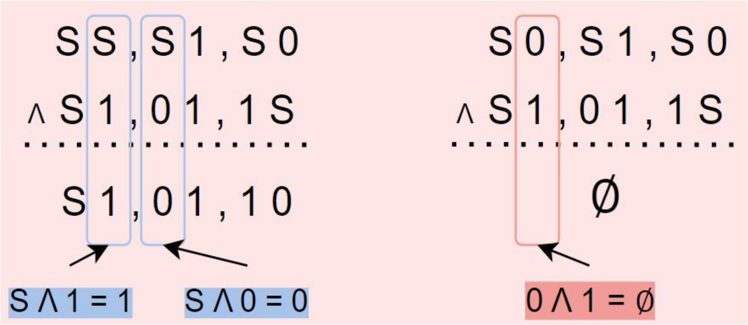
- We use Regular Expressions represent sets of traces.
- 1 = True, 0 = False,
- S = (0 | 1),
- Commas separate time steps.



WEST: MLTL Regular Expressions

- From now on, assume all formulas are in NNF using the usual equivalences.
- Given a formula with computation length m we compute a set of MLTL Regular Expressions that capture all satisfying traces of length m.





WEST: MLTL Regular Expressions

$$reg(\top) = S^n \qquad reg(\bot) = \varnothing$$

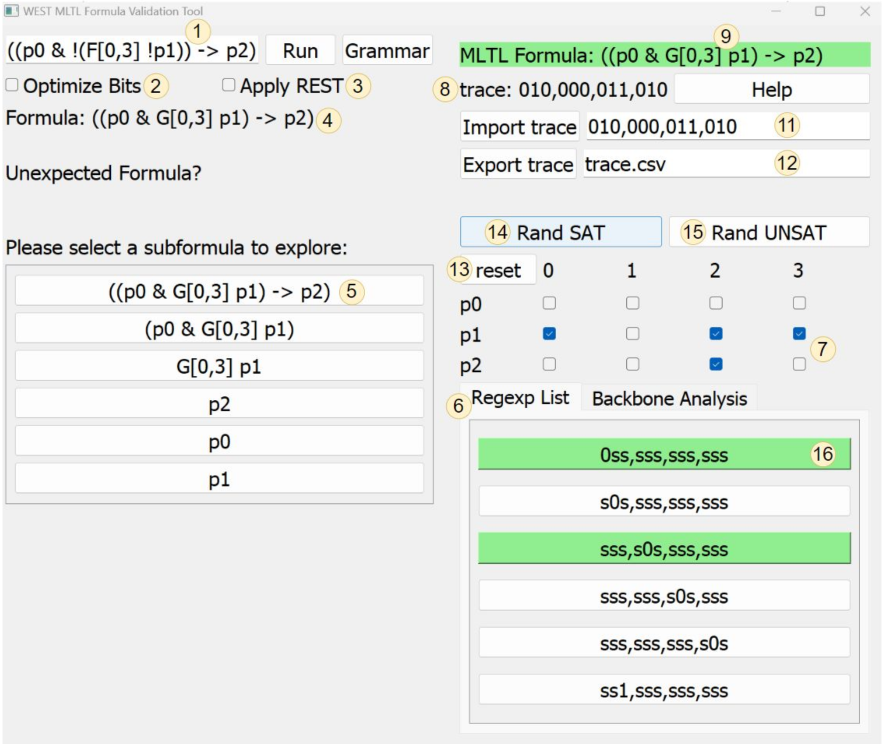
$$reg(p_k) = S^k 1 S^{n-k-1} \qquad reg(\neg p_k) = S^k 0 S^{n-k-1}$$

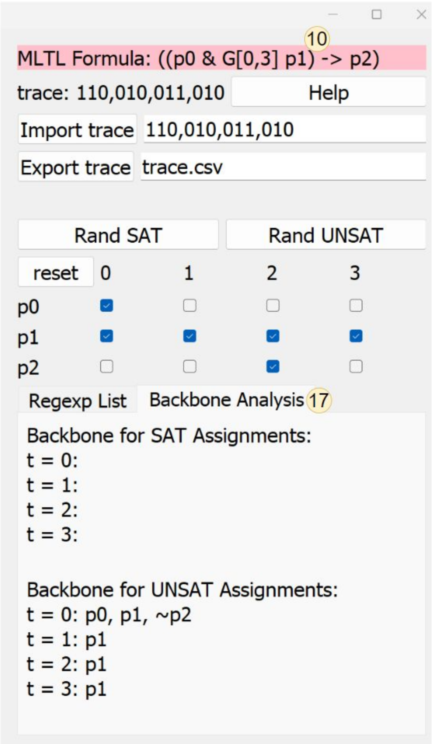
$$reg(\varphi \vee \psi) = reg(\varphi) \vee reg(\psi) \qquad reg(\varphi \wedge \psi) = reg(\varphi) \wedge reg(\psi)$$

$$reg(\mathcal{G}_{[a,b]}\varphi) = \bigwedge_{i=a}^{b} (S^n,)^i reg(\varphi) \qquad reg(\mathcal{F}_{[a,b]}\varphi) = \bigvee_{i=a}^{b} (S^n,)^i reg(\varphi)$$

Completeness and Correctness: For any well-formed MLTL formula φ in NNF, a trace π of length cplen(φ) satisfies φ if and only if π belongs to the regular language described by reg(φ).

WEST: MLTL Regular Expressions



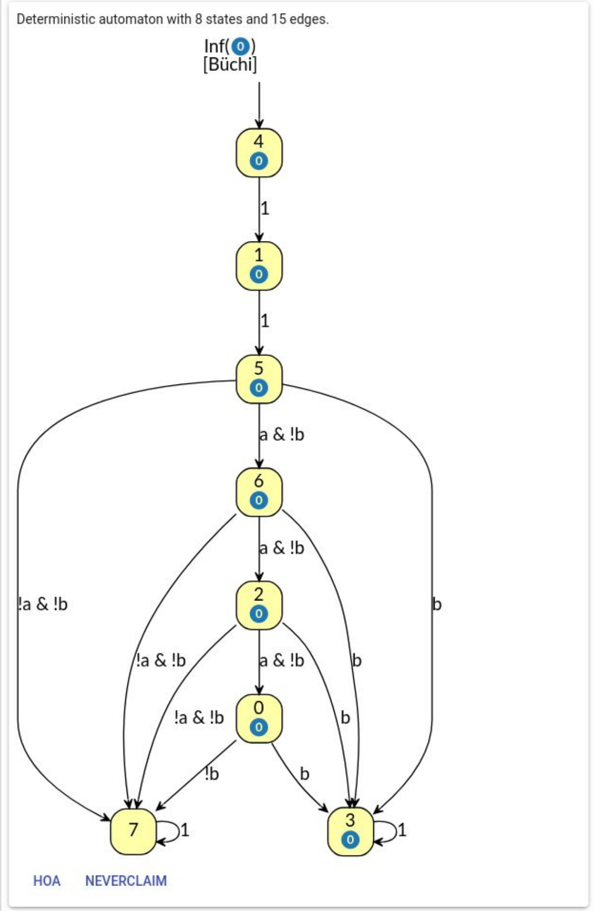


SCAN ME

What about MLTL to Automata?

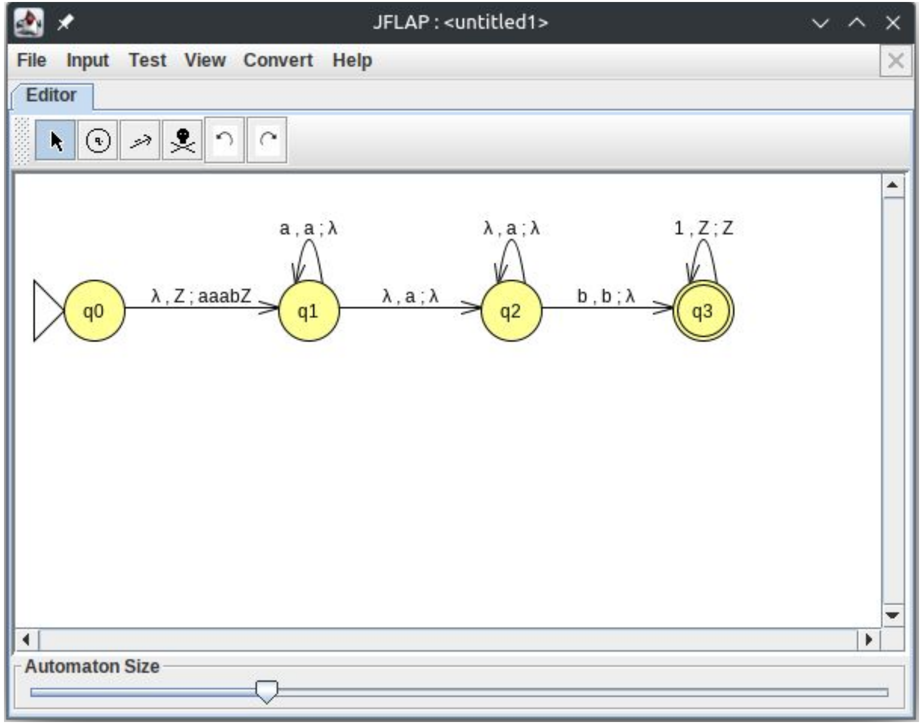
- Encoding MLTL into LTL/LTLf and then into an automata hiddens the succinctness of the language.
- Example: The formula $aU_{[2,5]}b$ can be encoded in LTLf as

$$X(X(b \vee (a \wedge X(b \vee (a \wedge X(b \vee (a \wedge Xb)))))))$$



Future work: MLTL to Automata!

- Encoding MLTL into LTL/LTLf and then into an automata hiddens the succinctness of the language.
- However, looking at the automata for the LTLf formula aUb, this suggests an approach using PDAs.



$$(\mathbf{x}, t) \models \varphi\, \mathcal{U}_{[a,b]}\, \psi \;\Leftrightarrow\; \exists t' \in [t+a, t+b] \text{ such that } (x, t') \models \psi \wedge \forall t'' \in [t, t'],\ (x, t'') \models \varphi\}$$

$$(x,t) \vDash \phi U \psi \text{ iff } \exists t' \in [t+a, t+b]$$

$$((x,t') \vDash \psi \,\&\, \forall t'' \in [t,t']\; ((x,t'') \vDash \phi))$$

$$\phi U_{[a,b]} \psi \mapsto \bigvee_{i \in [a,b]} \left(X^i \psi \wedge \bigwedge_{j \in [a, i-1]} X^j \phi \right)$$

1 = True, 0 = False,
S = (0 | 1),
Commas separate time steps.

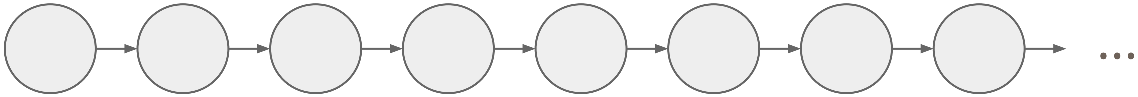
Operator	**Syntax**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Globally	$\mathbf{G}p$			p	p	p	p		
in the **F**uture	$\mathbf{F}p$					p			
Until	$p\,\mathbf{U}q$		p	p	p	p	q		
Release	$p\,\mathbf{R}q$			q	q	q	q	q	p,q

$$\pi, i \vDash \phi R \psi \text{ iff } \forall j \geq i \; (\pi, j \nvDash \psi \implies \exists k \; (i \leq k < j \,\&\, \pi, k \vDash \phi))$$

$$\pi, i \vDash X\phi \text{ iff } \pi, i+1 \vDash \phi$$

$$\pi, i \vDash G\phi \text{ iff } \forall j \geq i \; (\pi, j \vDash \phi)$$

$$\pi, i \vDash F\phi \text{ iff } \exists j \geq i \; (\pi, j \vDash \phi)$$



$$\phi ::= \top \mid \mu \mid \neg\phi \mid \phi \wedge \psi \mid \phi U_{[a,b]} \psi$$