

Formalizing Time: Temporal Logics and the Challenge of Visualizing MLTL

Laura Gamboa Guzman
May 1, 2025

Temporal logic

- From the Stanford Encyclopedia of Philosophy¹:

“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.

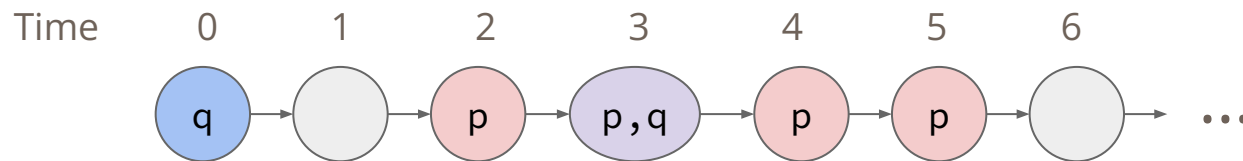
1. Goranko, Valentin and Antje Rumberg, "Temporal Logic", *The Stanford Encyclopedia of Philosophy* (Summer 2024 Edition), Edward N. Zalta & Uri Nodelman (eds.), <https://plato.stanford.edu/archives/sum2024/entries/logic-temporal/>.

LTL - Linear Temporal Logic

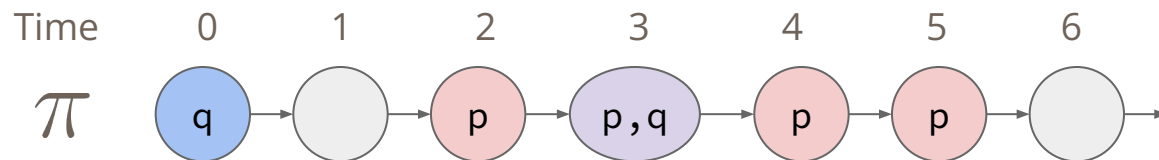
- The syntax is given by the following grammar, where p is any atomic proposition.

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

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



LTL - Linear Temporal Logic



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

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





























































$$\pi, i \models G\phi \text{ iff } \forall j \geq i (\pi, j \models \phi) \quad \pi, i \models F\phi \text{ iff } \exists j \geq i (\pi, j \models \phi)$$

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

$$\pi, i \models \phi R \psi \text{ iff } \forall j \geq i (\pi, j \not\models \psi \implies \exists k (i \leq k < j \ \& \ \pi, k \models \phi))$$

4

LTL - Linear Temporal Logic

Operator	Syntax	0	1	2	3	4	5	6	7					
Next	Xp													...
Globally	Gp													...
in the Future	Fp													...
Until	$p \mathbf{U} q$													...
Release	$p \mathbf{R} q$													...

5

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”

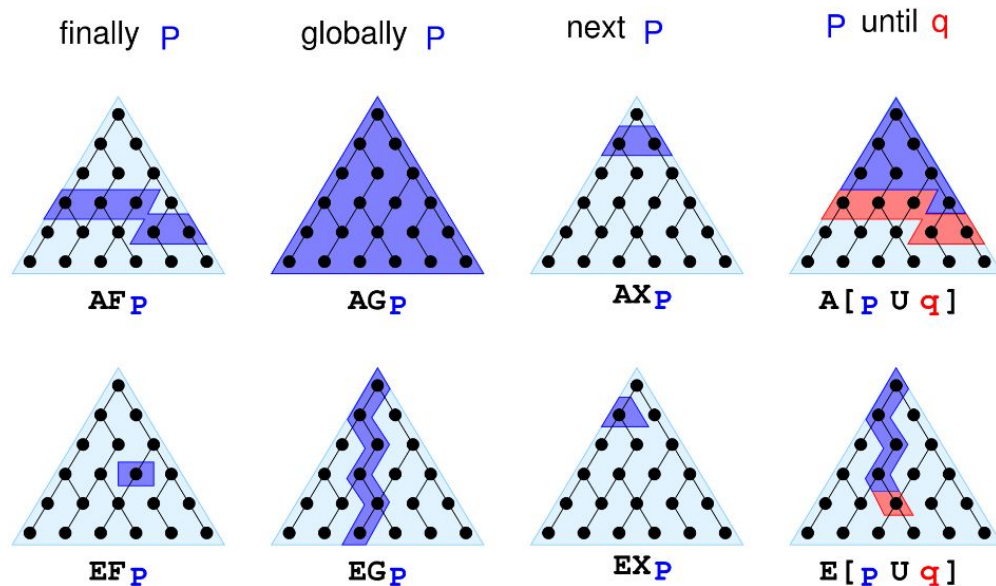
LTL expressiveness

- “Something bad (p) never happens” $\neg Fp$
- “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 $x_1[t], x_2[t], \dots, x_n[t]$, then atomic predicates are of the form

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

- STL formulas:

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

- Semantics:

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

STL - Signal Temporal Logic

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

- STL is a logic

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
jdpenna@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 in the F uture Until Release	$G_{[2,5]} p$			p	p	p	p		
	$F_{[0,4]} p$					p			
	$p U_{[1,6]} q$		p	p	p	p	q		
	$p R_{[2,7]} q$			q	q	q	q	q	p, q

13

MLTL - Mission-time Linear Temporal Logic

- In short: 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
 $\forall j \in [a, i - 1] \pi_j \models \alpha$

$\pi \models \alpha \mathcal{R}_{[a,b]} \beta$ iff $|\pi| \leq a$ or $\forall i \in [a, b] \pi_i \models \beta$ or
 $\exists j \in [a, b - 1]$ 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 $\mathcal{AP} = \{p_0, p_1, \dots, p_{n-1}\}$.
- 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.

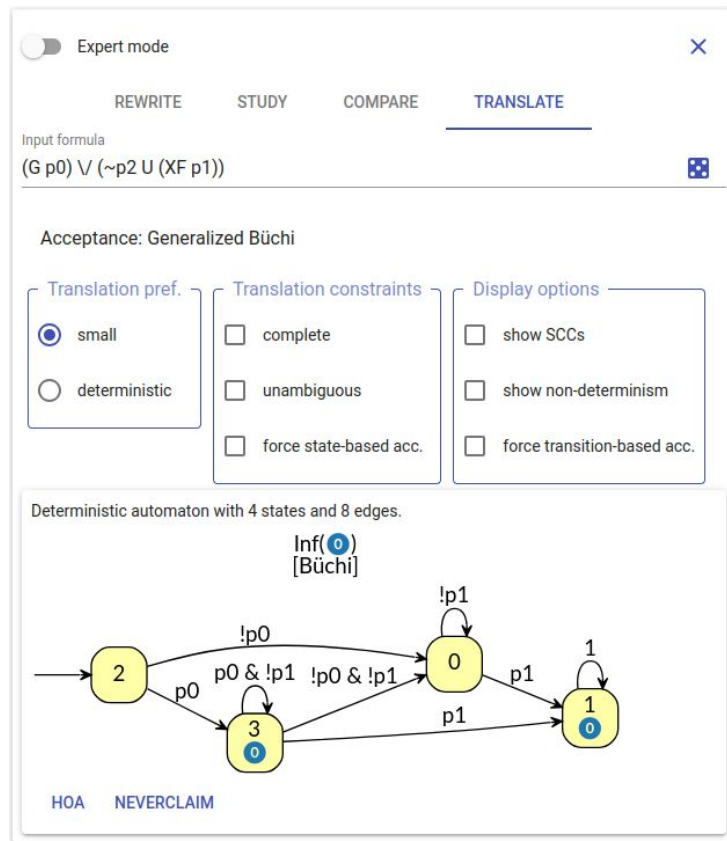
The screenshot shows the Spot web interface for translating LTL formulas. At the top, there's a navigation bar with 'REWRITE', 'STUDY', 'COMPARE', and 'TRANSLATE' (the active tab). Below the navigation bar, the 'Input formula' is 'p0 U p1'. The 'Acceptance' is set to 'Generalized Büchi'. There are three panels for configuration: 'Translation pref.' with 'small' selected, 'Translation constraints' with 'complete', 'unambiguous', and 'force state-based acc.' options, and 'Display options' with 'show SCCs', 'show non-determinism', and 'force transition-based acc.' options. The main area displays a 'Deterministic automaton with 2 states and 3 edges.' The automaton has two states: state 1 (yellow) and state 0 (blue). State 1 is the initial state and has a self-loop labeled 'p0 & !p1'. State 0 is an accepting state (double circle) and has a self-loop labeled '1'. There is a transition from state 1 to state 0 labeled 'p1'. Above the automaton, the text 'Inf(0) [Büchi]' is shown. At the bottom, there are links for 'HOA' and 'NEVERCLAIM'.

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

17

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>

18

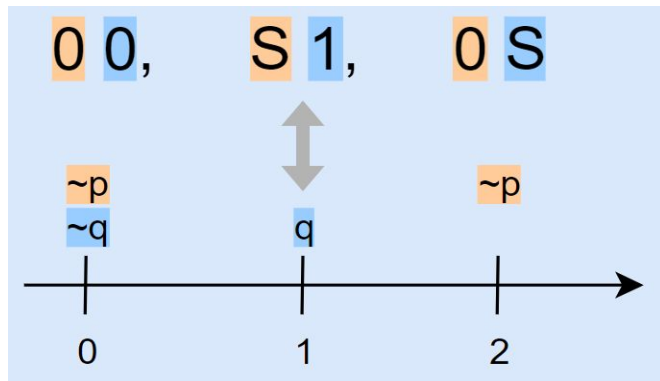
Visualizing MLTL

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

p	q	$\sim p$	$\sim q$	$p \wedge q$	$\sim(p \wedge q)$	$\sim p \vee \sim q$
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

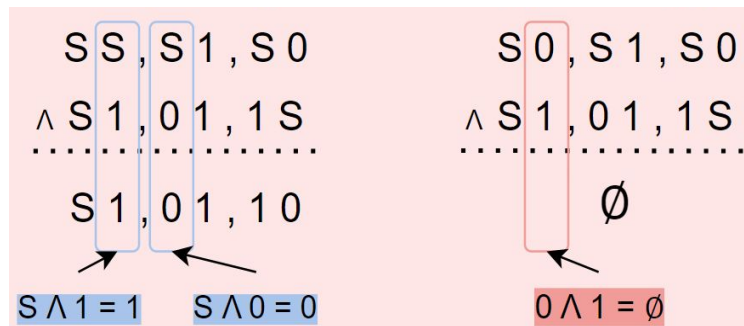
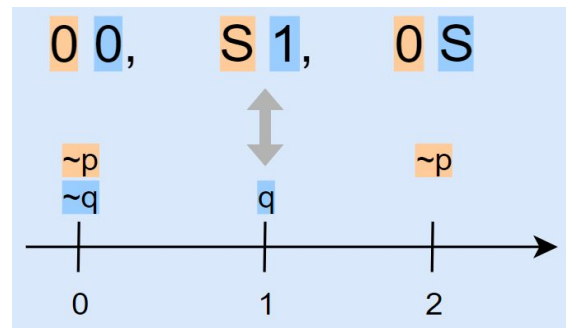
WEST: MLTL Regular Expressions

- We use Regular Expressions represent sets of traces.
- 1 = True, 0 = False,
- $S = (0 \mid 1)$,
- 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

$$\text{reg}(\top) = S^n$$

$$\text{reg}(\perp) = \emptyset$$

$$\text{reg}(p_k) = S^k 1 S^{n-k-1}$$

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

$$\text{reg}(\varphi \vee \psi) = \text{reg}(\varphi) \vee \text{reg}(\psi) \quad \text{reg}(\varphi \wedge \psi) = \text{reg}(\varphi) \wedge \text{reg}(\psi)$$

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

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

WEST: MLTL Regular Expressions

WEST MLTL Formula Validation Tool

MLTL Formula: $((p0 \ \& \ ! (F[0,3] \ ! p1)) \rightarrow p2)$

Optimize Bits Apply REST

Formula: $((p0 \ \& \ G[0,3] \ p1) \rightarrow p2)$

Unexpected Formula?

Please select a subformula to explore:

- $((p0 \ \& \ G[0,3] \ p1) \rightarrow p2)$
- $(p0 \ \& \ G[0,3] \ p1)$
- $G[0,3] \ p1$
- $p2$
- $p0$
- $p1$

trace: 010,000,011,010

Import trace 010,000,011,010

Export trace trace.csv

Rand SAT Rand UNSAT

reset	0	1	2	3
p0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
p1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
p2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Regexp List Backbone Analysis

0ss,sss,sss,sss

s0s,sss,sss,sss

sss,s0s,sss,sss

sss,sss,s0s,sss

sss,sss,sss,s0s

ss1,sss,sss,sss

MLTL Formula: $((p0 \ \& \ G[0,3] \ p1) \rightarrow p2)$

trace: 110,010,011,010

Import trace 110,010,011,010

Export trace trace.csv

Rand SAT Rand UNSAT

reset	0	1	2	3
p0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
p1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
p2	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Regexp List Backbone Analysis

Backbone for SAT Assignments:

t = 0:

t = 1:

t = 2:

t = 3:

Backbone for UNSAT Assignments:

t = 0: p0, p1, ~p2

t = 1: p1

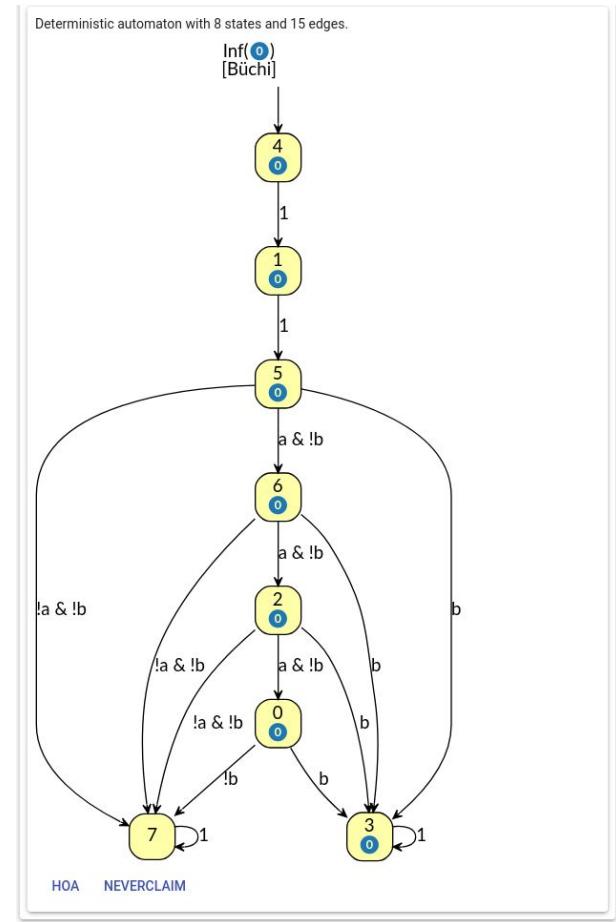
t = 2: p1

t = 3: p1



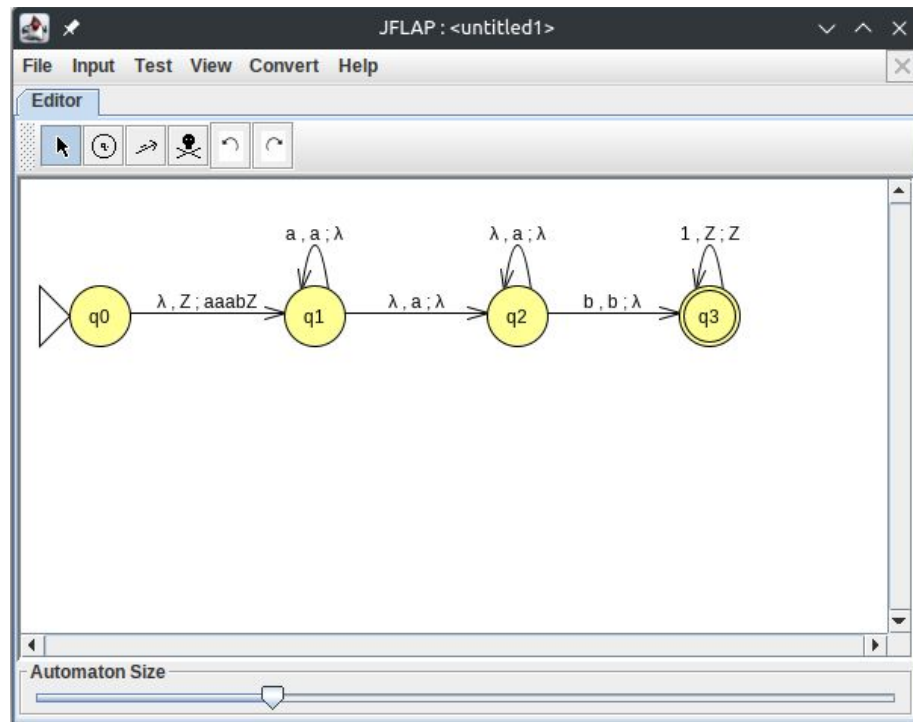
What about MLTL to Automata?

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



25

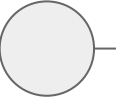
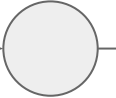
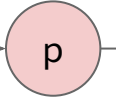
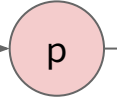
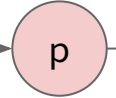
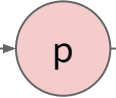
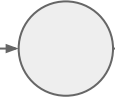
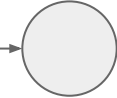
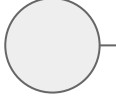
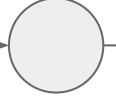
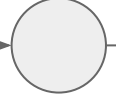
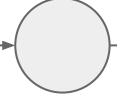
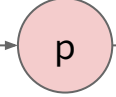
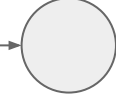
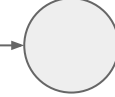
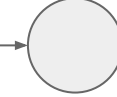
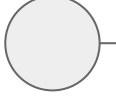
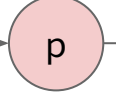
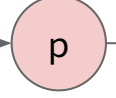
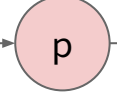
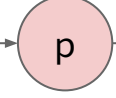
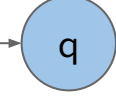
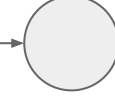
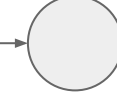
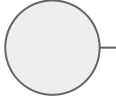
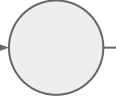
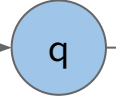
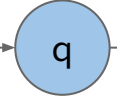
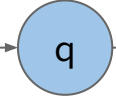
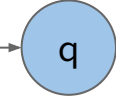
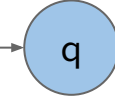
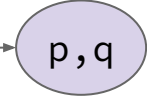
$$(\mathbf{x}, t) \models \varphi \mathcal{U}_{[a,b]} \psi \Leftrightarrow \exists t' \in [t + a, t + b] \text{ such that } (x, t') \models \psi \wedge \forall t'' \in [t, t'], (x, t'') \models \varphi$$

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

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

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

1 = True, 0 = False,
 $S = (0 \mid 1)$,
 Commas separate time
 steps.

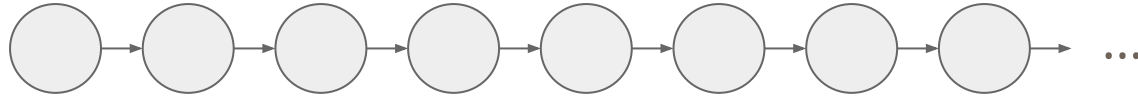
Operator	Syntax	0	1	2	3	4	5	6	7
Globally in the F uture Until Release	Gp								
	Fp								
	$p \mathbf{U} q$								
	$p \mathbf{R} q$								

$$\pi, i \models \phi R \psi \text{ iff } \forall j \geq i (\pi, j \not\models \psi \implies \exists k (i \leq k < j \ \& \ \pi, k \models \phi))$$

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

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

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



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