AP-Observation Automata for Abstraction-based Verification of Continuous-time Systems

Sasinee Pruekprasert and Clovis Eberhart

The University of Tokyo Tohoku University

International Colloquium on Theoretical Aspects of Computing (ICTAC) 2025

supported by JSPS KAKENHI Grant Numbers JP21K14191, JP22KK0155, and JP25H00446

Our goal

- Given

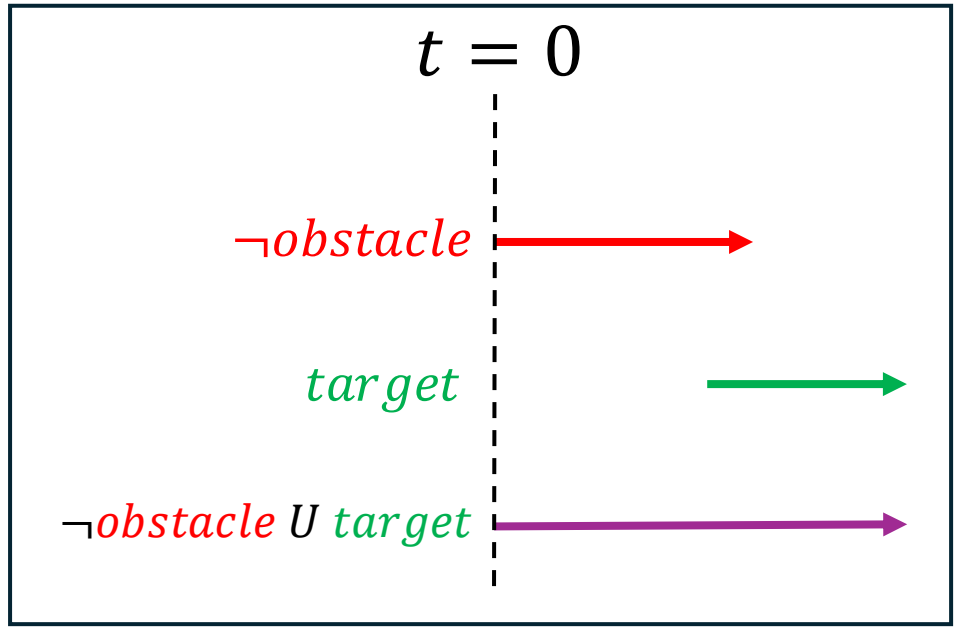
continuous-state
continuous-time
system Σ

LTL Specification
φ

- Problem: Does system Σ satisfy φ?
- Goal: An algorithm to verify if a given Σ satisfy a given φ

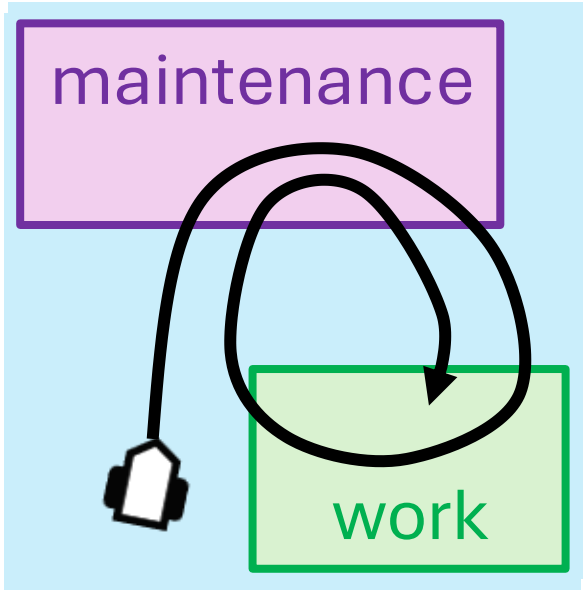
Linear Temporal Logic (LTL)

$$\varphi ::= \underset{\text{(true)}}{\top} \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi U \varphi$$



- $p \in AP$ is an atomic proprosition
  - E.g., "colliding with obstacle", "reaching target region"

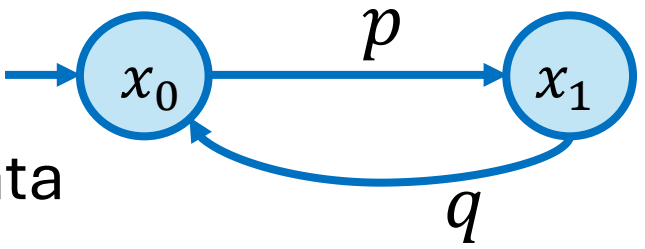
- $\neg obstacle\ U\ target$
- $\Diamond target \equiv \top\ U\ target$
- $\Box\neg obstacle \equiv \neg\Diamond obstacle$
- $\Box\Diamond maintenance\_station \wedge \Box\Diamond work\_station$



LTL as Verification/Control Specifications

Given

- Systems modeled by deterministic finite-state automata

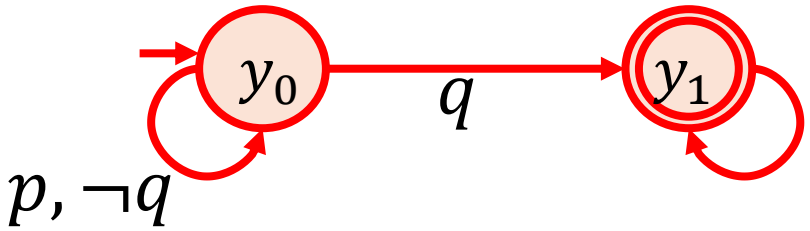


- LTL specification

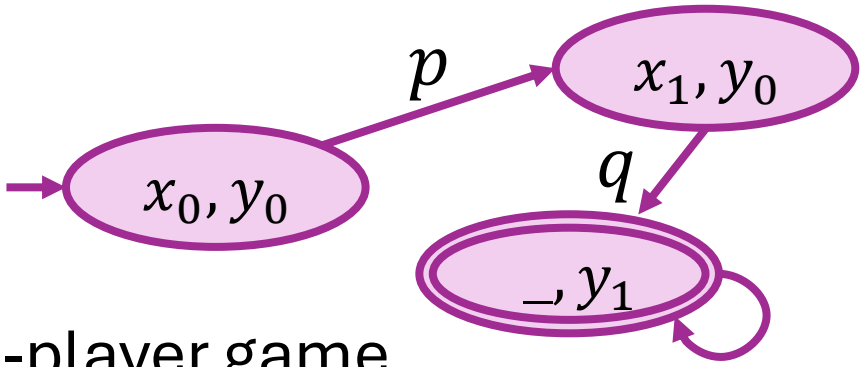
$$pUq$$

(p holds until q holds)

Büchi automaton (Vardi and Wolper, 1986)



System verification problem: by taking product



Controller synthesis problem: formulate as a two-player game

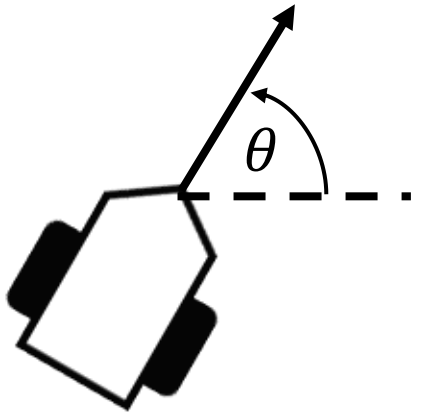
Continuous-state Continuous-time Systems

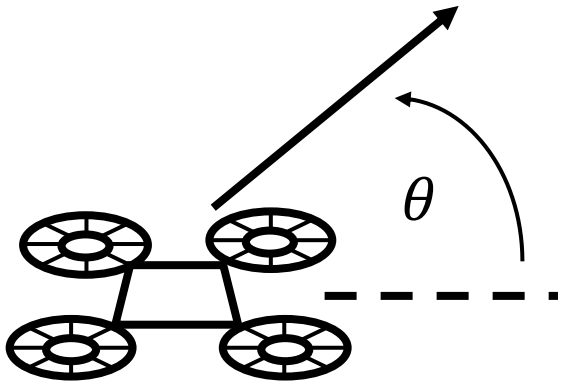
e.g.,

$$\dot{x}(t) = v\big(1 + \lambda(t)\big)\cos(\theta(t))$$

$$\dot{y}(t) = v\big(1 + \lambda(t)\big)\sin(\theta(t))$$

$\dot{\theta}(t)$ is the control input





$$\dot{x}(t) \in [v(t)\cos(\theta(t)) - \varepsilon, v(t)\cos(\theta(t)) + \varepsilon]$$

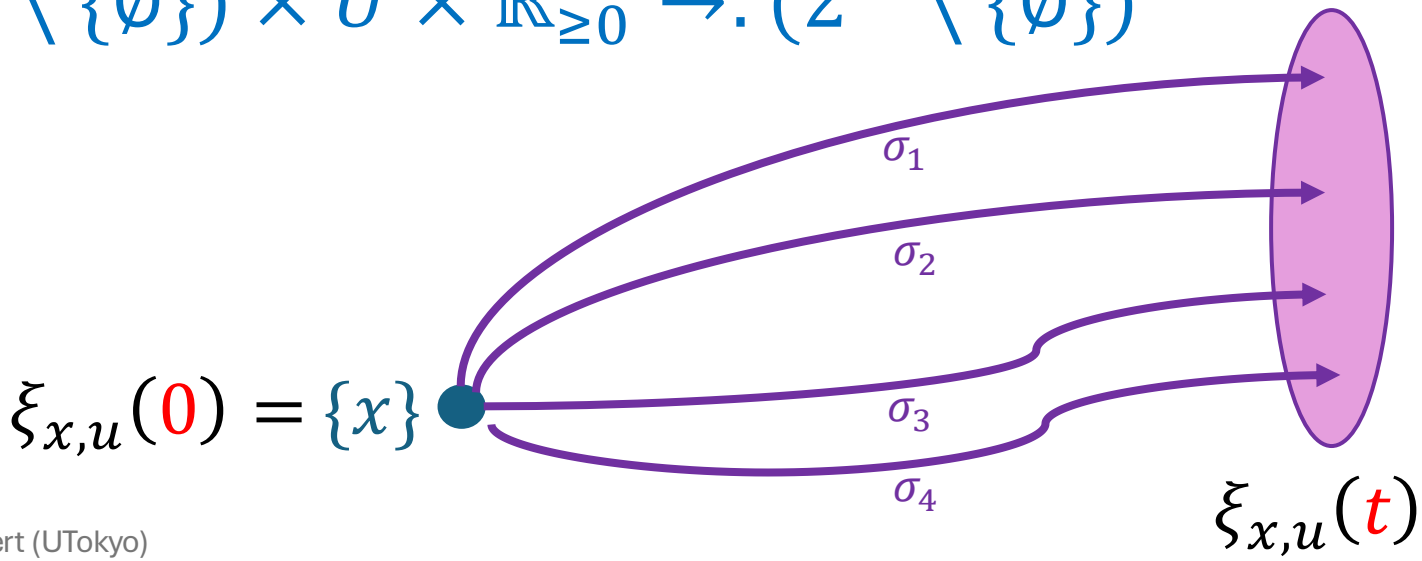
$$\dot{y}(t) \in [v(t)\sin(\theta(t)) - \varepsilon, v(t)\sin(\theta(t)) + \varepsilon]$$

$\theta(t)$ is the control input

Continuous-time Non-deterministic Systems

$$\Sigma = (X, x_{in}, U, \xi)$$

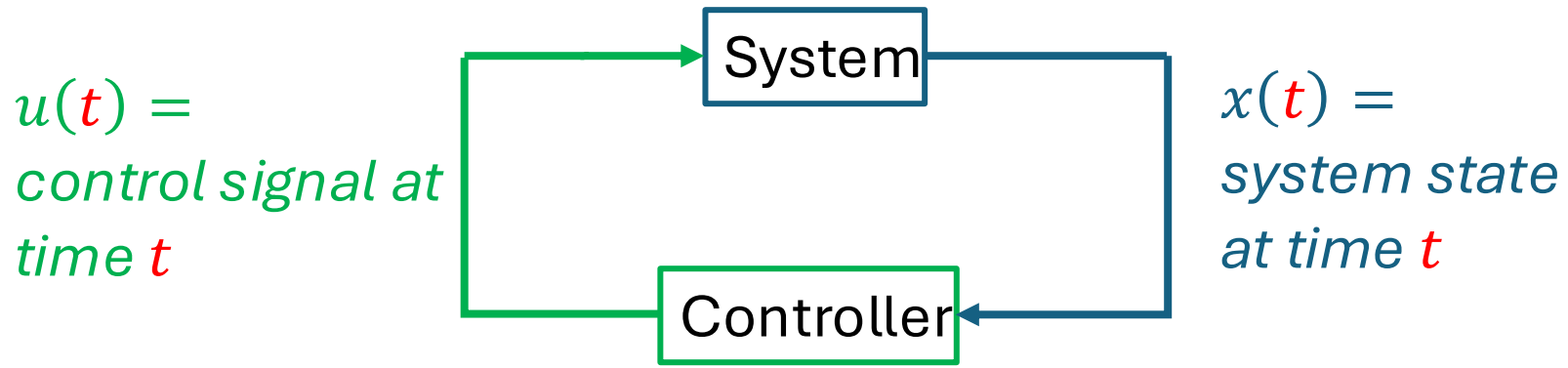
- X: Set of states, $x_{in} \in X$: the initial state
- U: Set of control input
- $\xi: (2^X \setminus \{\emptyset\}) \times U \times \mathbb{R}_{\geq 0} \rightarrow: (2^X \setminus \{\emptyset\})$



Question:

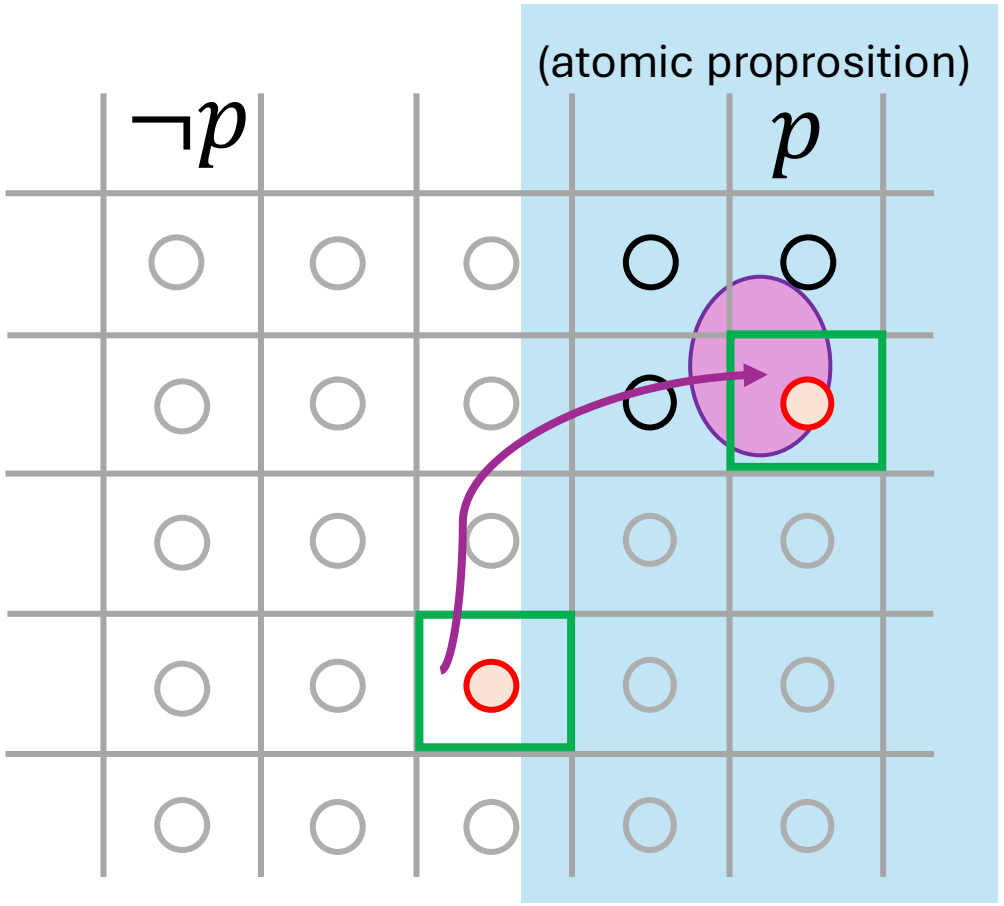
How to verify/control **continuous-time** system for LTL specs?

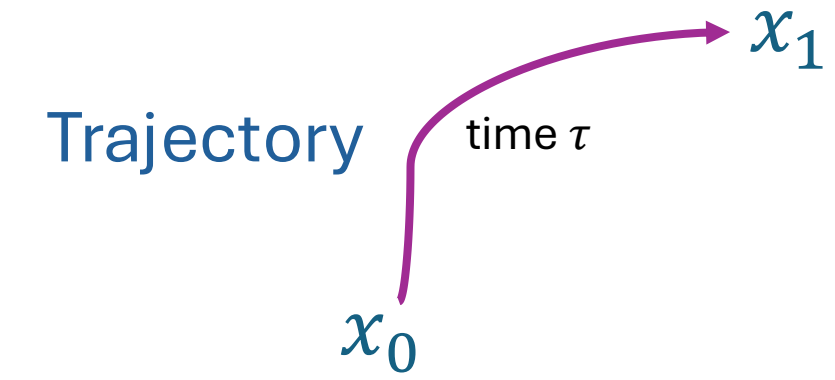
Our focus: Periodic-time Control Loop



$t \in \{0, \tau, 2\tau, \ldots\}, \qquad \tau \in \mathbb{R}_{\geq 0}$

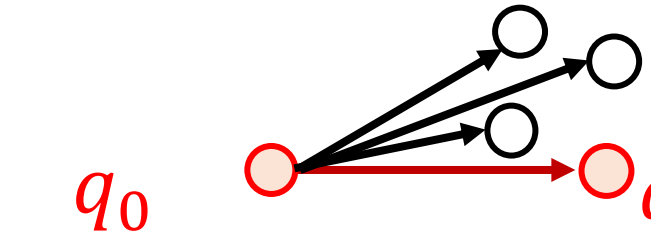
Abstraction: Continuous → Discrete System





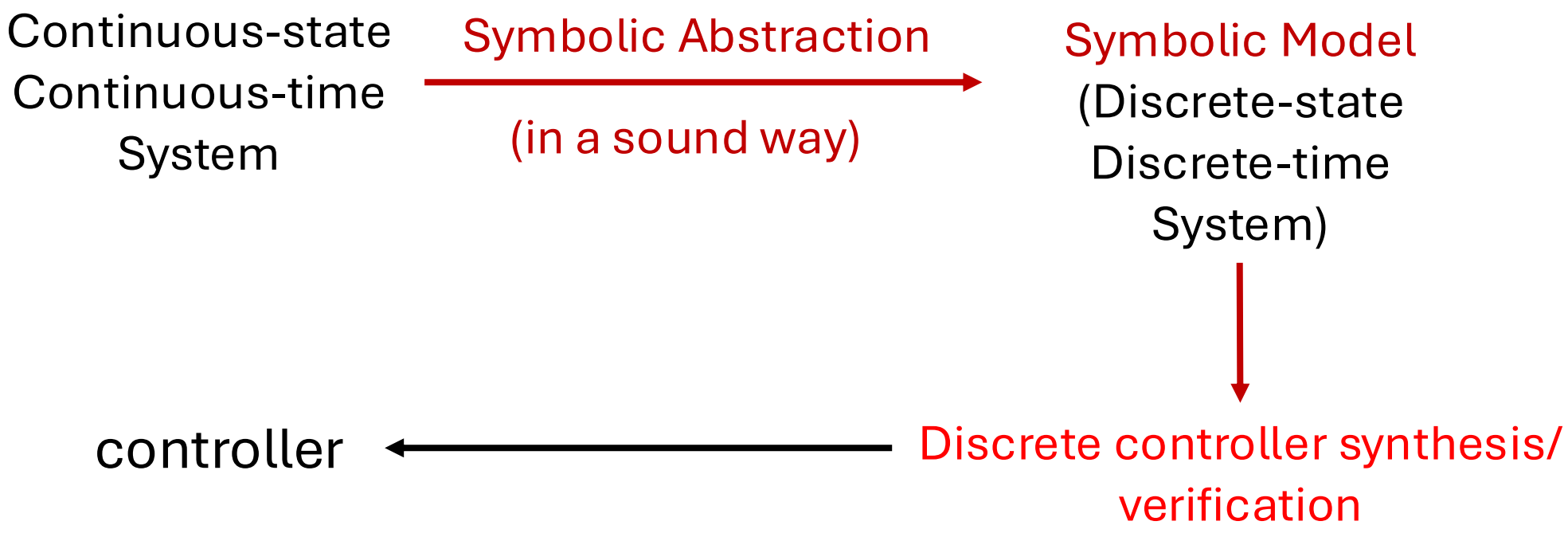
is represented by

discrete transitions



$= abs(x_0)$ $= abs(x_1)$

Abstraction-based Verification & Control



Verification/Control problem is reduced to a problem in a discrete-state space

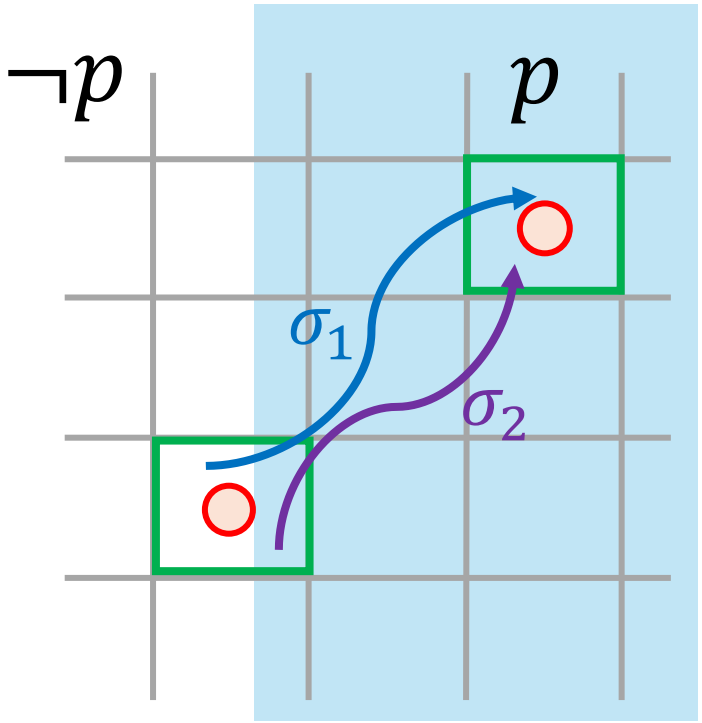
Symbolic Control

- Information loss from abstraction
  - Nondeterminism from the abstraction
  - **Information loss on the atomic propositions from time discretization**

Therefore,
symbolic control often gives sound but not complete solutions

- Previous work consider
  - subclasses of LTL specifications
  - restricted classes of (nonlinear) system
  - discrete-time systems and discrete-time semantics LTL

Information loss on APs from Time discretization



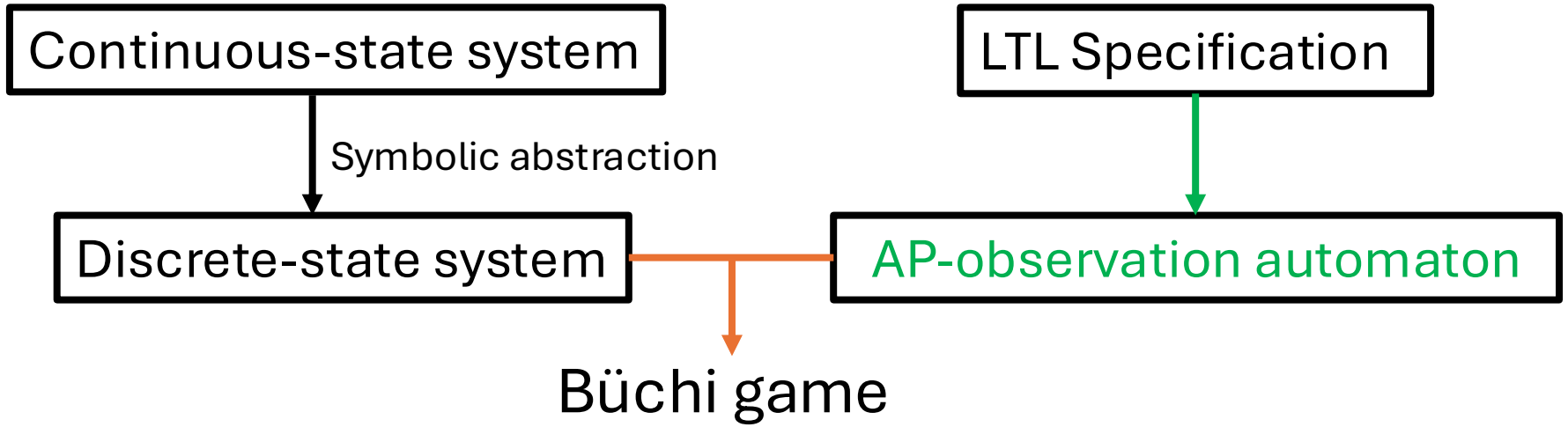
Spec: $\Box p \equiv \neg(TU\neg p)$
(p holds always)

Both trajectories σ_1 and σ_2
are abstracted to
the same transition

However, only σ_2 satisfies $\Box_{[0,\tau]}p$
(p always holds along the trajectory)

Our Contribution

- AP-Observation Automaton: a Büchi automaton tailored specifically for abstraction-based verification & control of continuous-time system



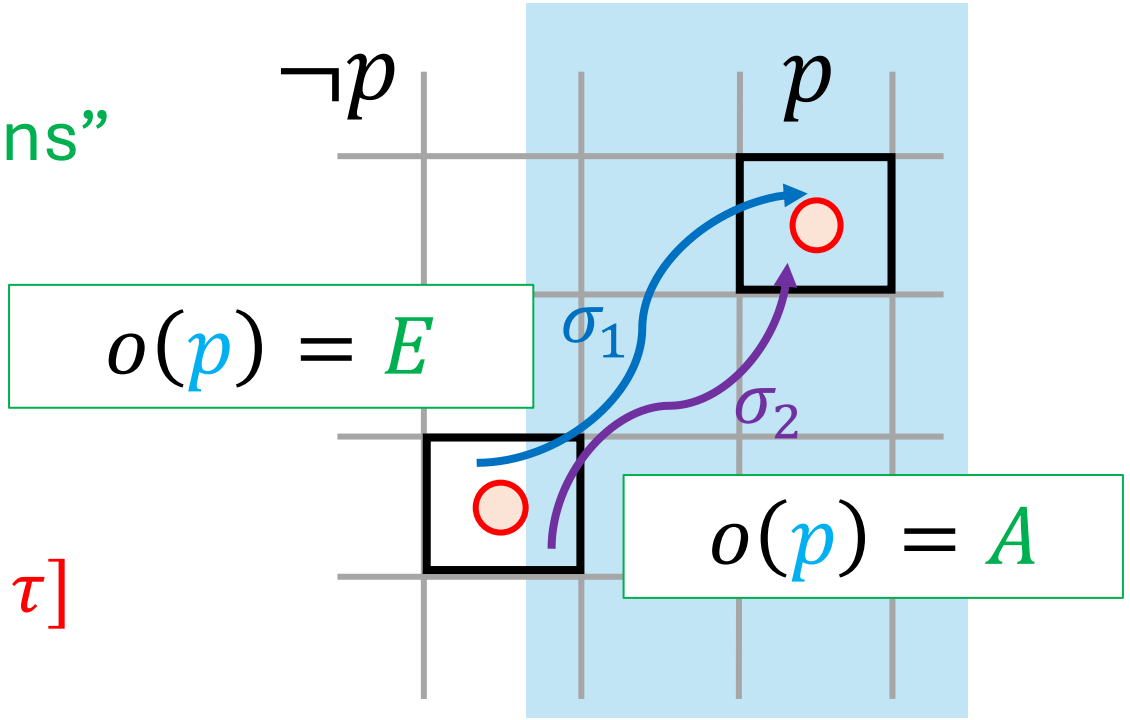
- Two-player game-based verification

Atomic Proposition (AP) Observations

We abstract the satisfaction of

$p \in AP$ along a trajectory $\sigma: [0, \tau] \to X$
into 4 possible "AP-observations"

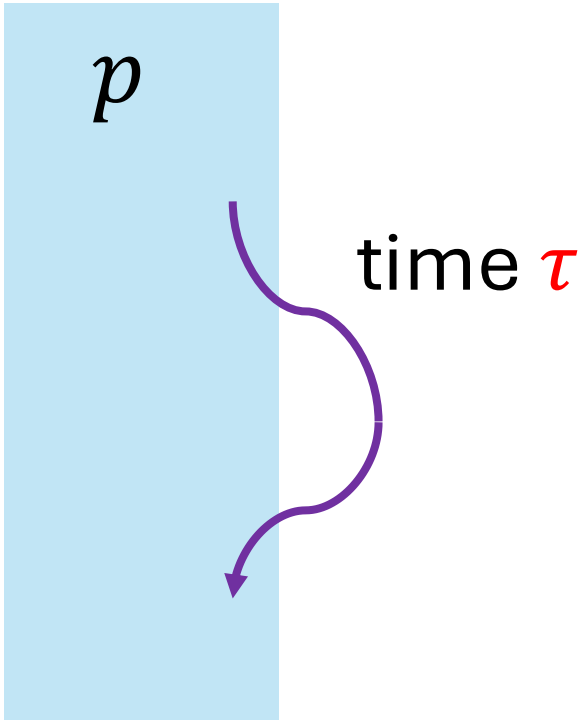
- A : p holds at All time $t \in [0, \tau]$
- N : p Never hold for all $t \in [0, \tau]$
- Z : $\exists t' \in (0, \tau)$,
  p holds for $[0, t')$, but not for $(t', \tau]$
  (holds at time Zero)
- E : $\exists t' \in (0, \tau)$,
  p holds for $(t', \tau]$, but not for $[0, t')$
  (holds at the End)



Atomic Proposition (AP) Observations

We abstract the satisfaction of

$p \in AP$ along a trajectory $\sigma: [0, \tau] \to X$
into 4 possible "AP-observations"



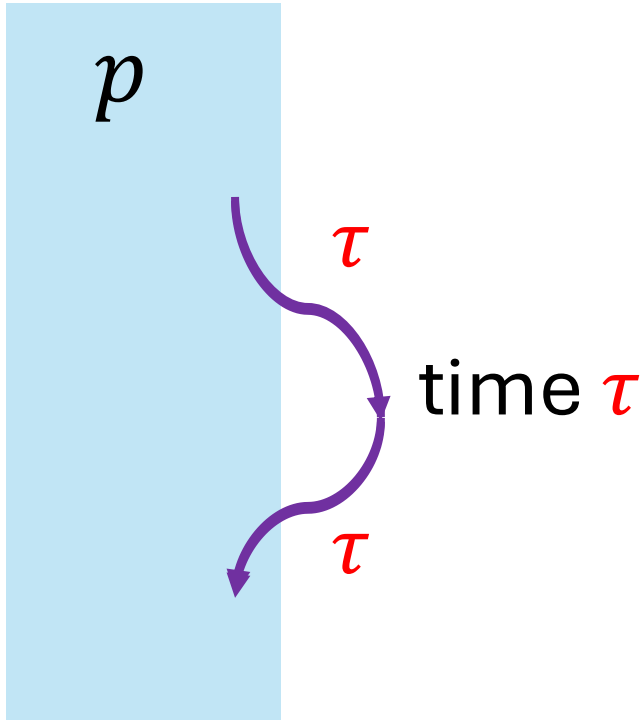
- A : p holds at All time $t \in [0, \tau]$
- N : p Never hold for all $t \in [0, \tau]$
- Z : $\exists t' \in (0, \tau)$,
  p holds for $[0, t')$, but not for $(t', \tau]$
  (holds at time Zero)
- E : $\exists t' \in (0, \tau)$,
  p holds for $(t', \tau]$, but not for $[0, t')$
  (holds at the End)

Assumption 1

Given $P: X \to 2^{AP}$ (assigns APs to each state).
$\forall \sigma: [0, \tau] \to X$, $\forall t \in [0, \tau]$, $\forall p \in AP$,

1. If $p \in P(\sigma(0)) \cap P(\sigma(\tau))$,

   then, $p \in P(\sigma(t))$, $\forall t' \in [0, \tau]$

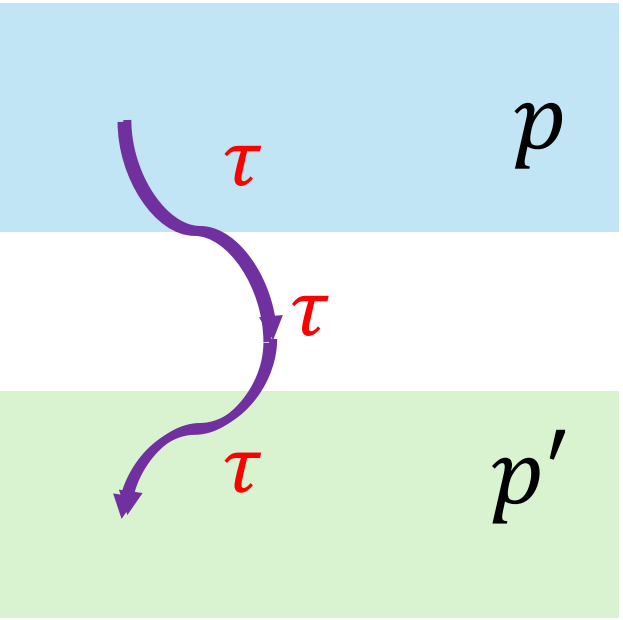


If p holds at the beginning and the end, p holds at all times
(σ cannot cross the border of each AP region twice within time τ)

Assumption 2

Given $P: X \to 2^{AP}$ (assignes APs to each state).
$\forall \sigma: [0, \tau] \to X$, $\forall t \in [0, \tau]$, $\forall p, p' \in AP$,

2. If $p \in P(\sigma(0)) \setminus P(\sigma(\tau))$
   and $p' \in P(\sigma(0)) \setminus P(\sigma(\tau))$,
   then, $p = p'$



(σ cross at most one AP region boundary within time τ)

A constraint for "small enough" τ is **given in Lemma 1**

Atomic Proposition (AP) Observations

We abstract $p \in AP$ along $\sigma: [0, \tau] \to X$ into

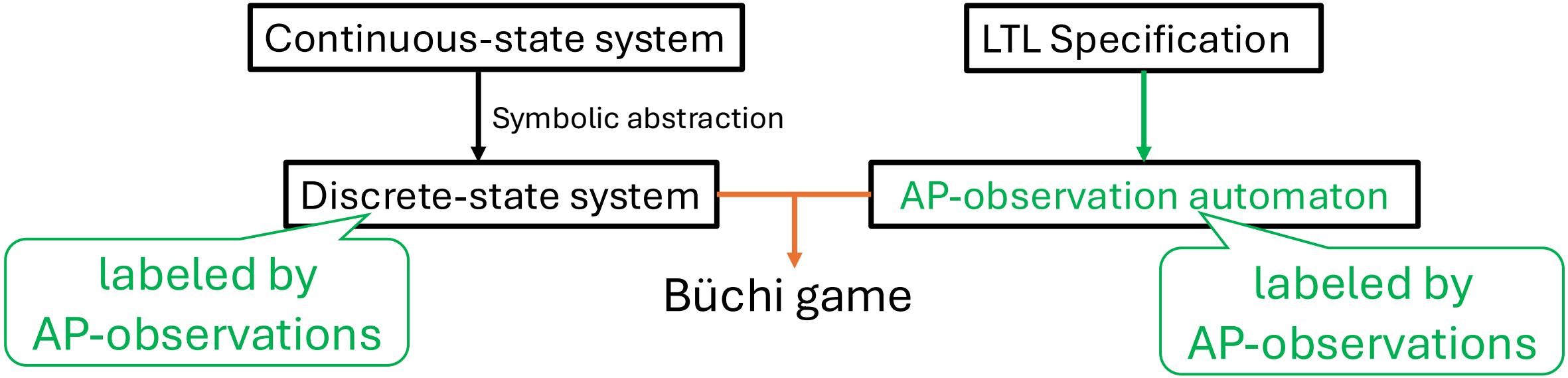
- A : p holds at All time $t \in [0, \tau]$
- N : p Never hold for all $t \in [0, \tau]$
- Z : $\exists t' \in (0, \tau), p$ holds for $[0, t']$, b ero)
- E : $\exists t' \in (0, \tau), p$ holds for $(t', \tau]$, b nd)

Labels of the discrete-state system transitions

By Assumption 1,
for all $\sigma: [0, \tau] \to X$, $o(p) \in \{A, N, Z, E\}$

Our Contribution

- AP-Observation Automaton: a Büchi automaton tailored specifically for abstraction-based verification & control of continuous-time system



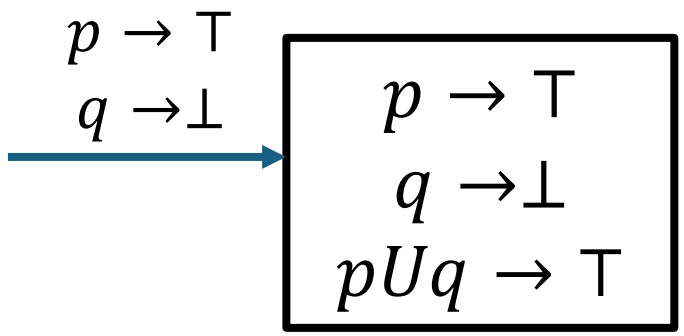
- Two-player game-based verification

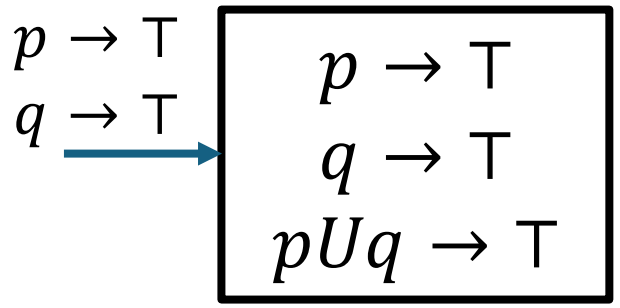
Review: LTL to Buchi automaton (Vardi and Wolper, 1986)

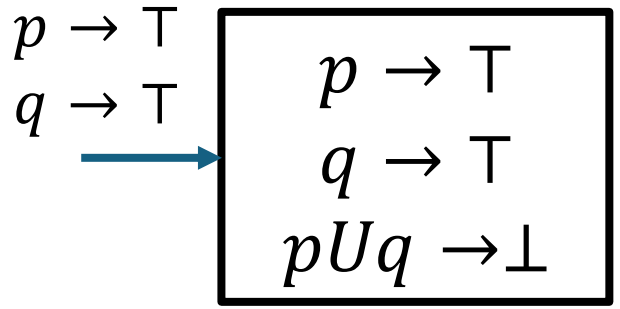
State: sub-formulas → $\{\top, \bot\}$

Transition labels: APs → $\{\top, \bot\}$

e.g., $\varphi = pUq$



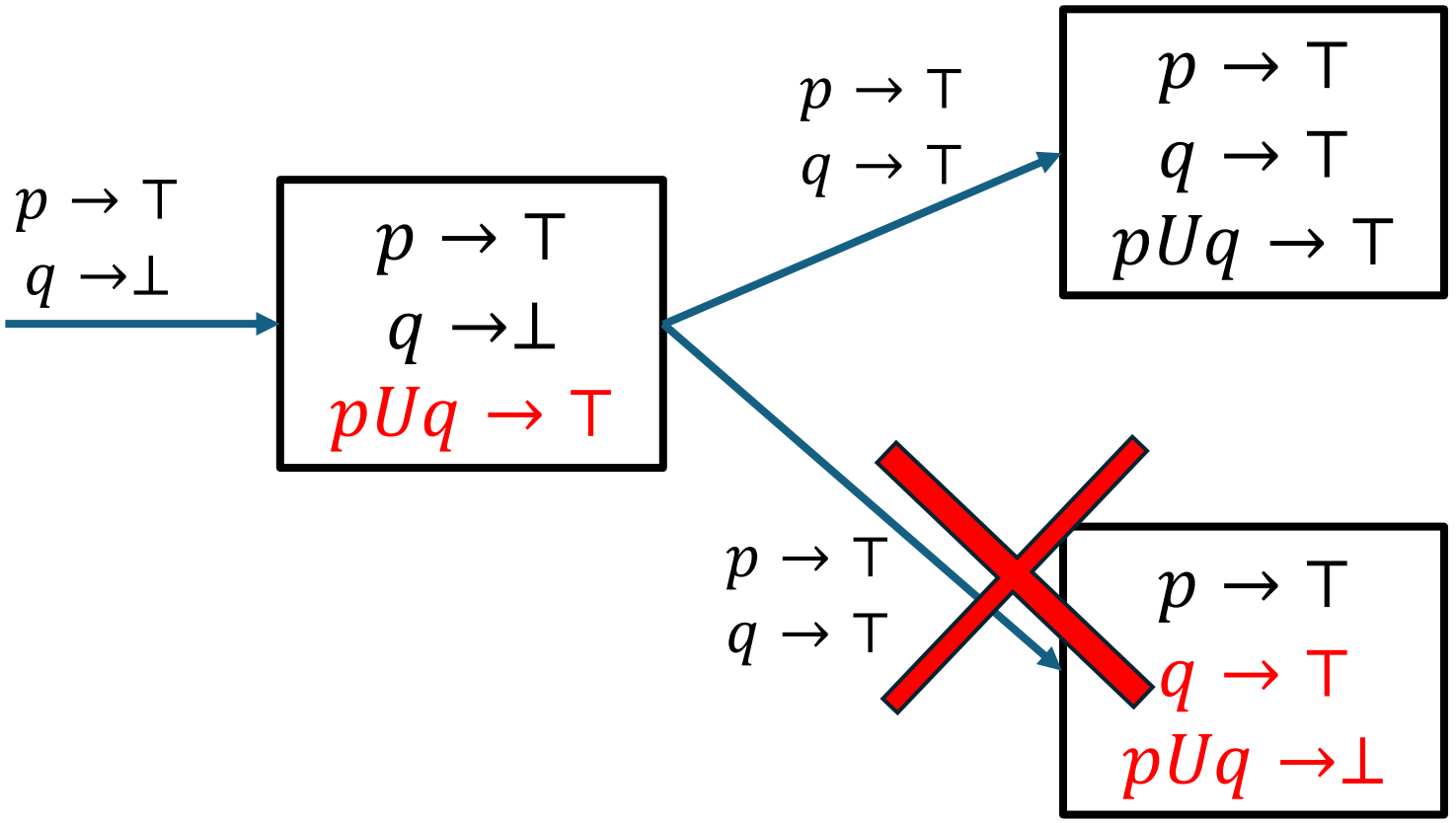


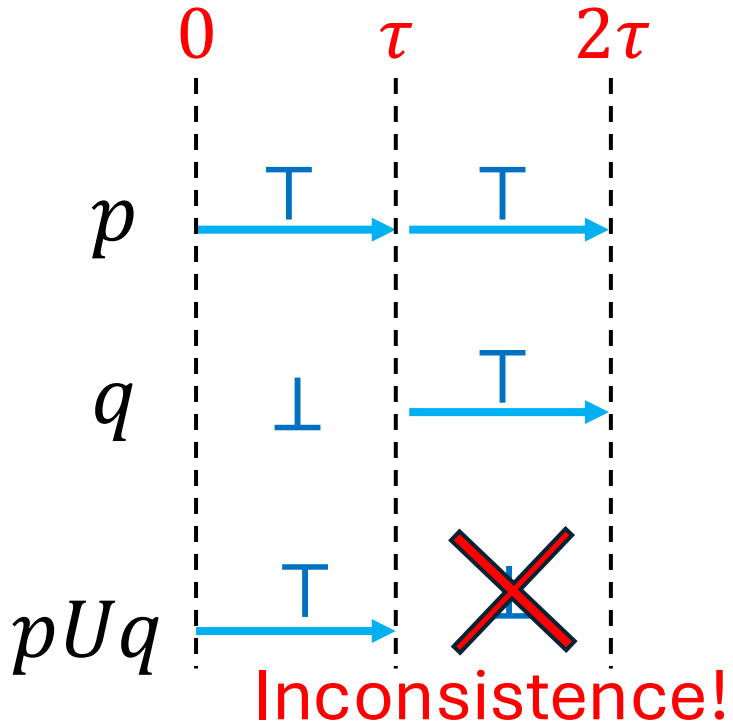


Rule for having transition $M \to M'$:

$pUq \to \top$ *at* M iff ($q \to \top$ *at* M

or ($p \to \top$ *at* M and $pUq \to \top$ *at* M'))

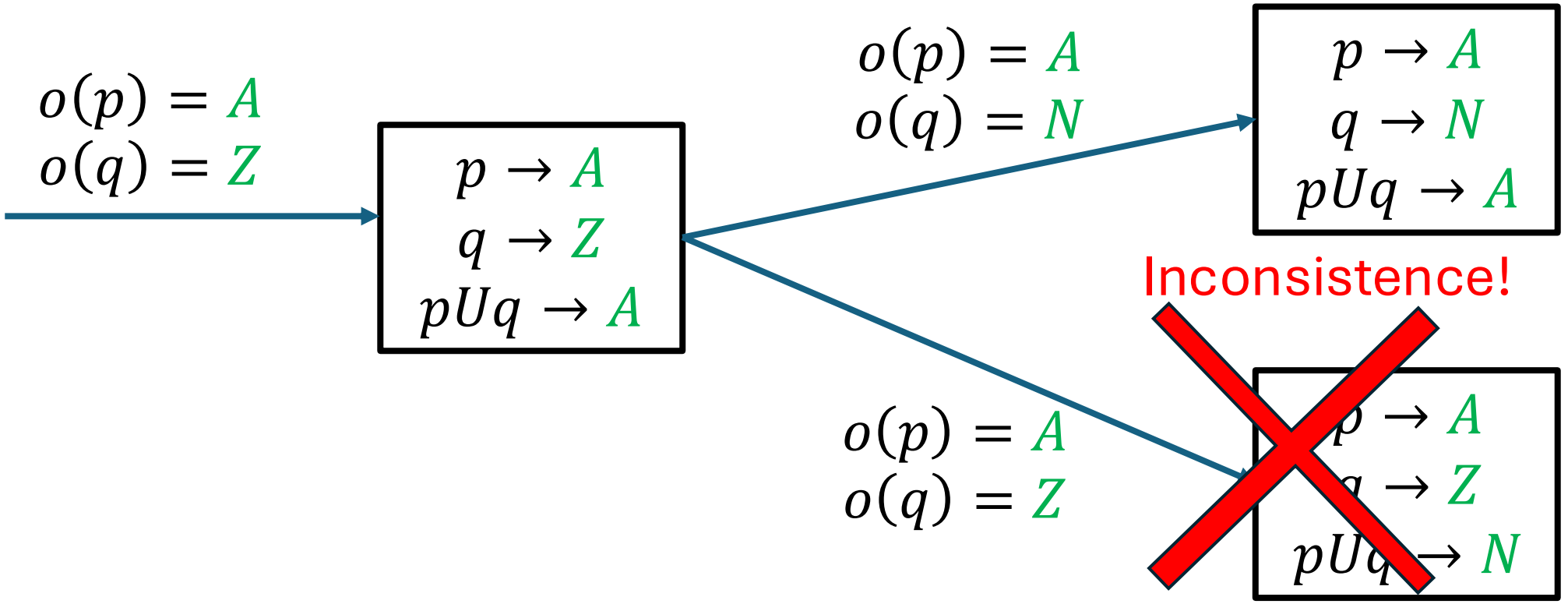




LTL to AP-Observation Automata

State: sub formulas → $\{A, Z, E, N\}$

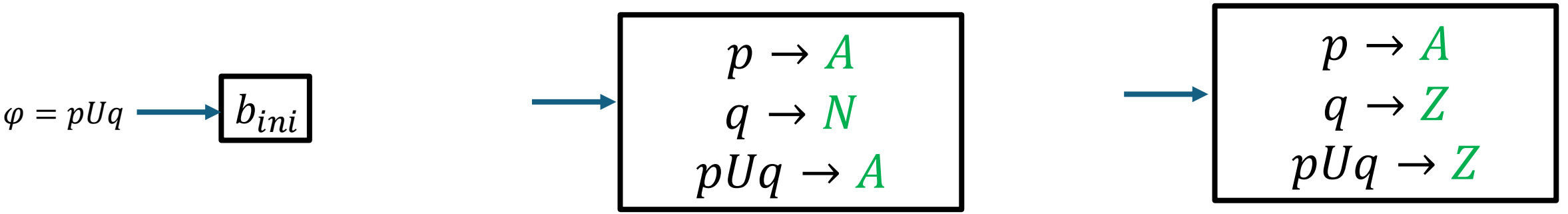
Transition labels: observations of each AP



LTL to AP-Observation Automata

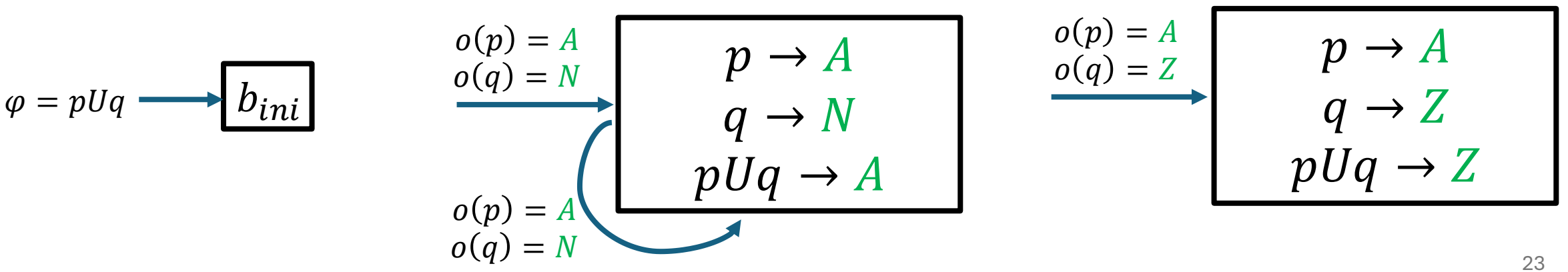
$$B_\varphi = (cs(\varphi) \cup \{b_{ini}\}, \delta_\varphi, F_\varphi)$$

- State set:
  - $cs(\varphi)$ is the set of "consistent subformula valuation"
  - b_{ini} is the initial state



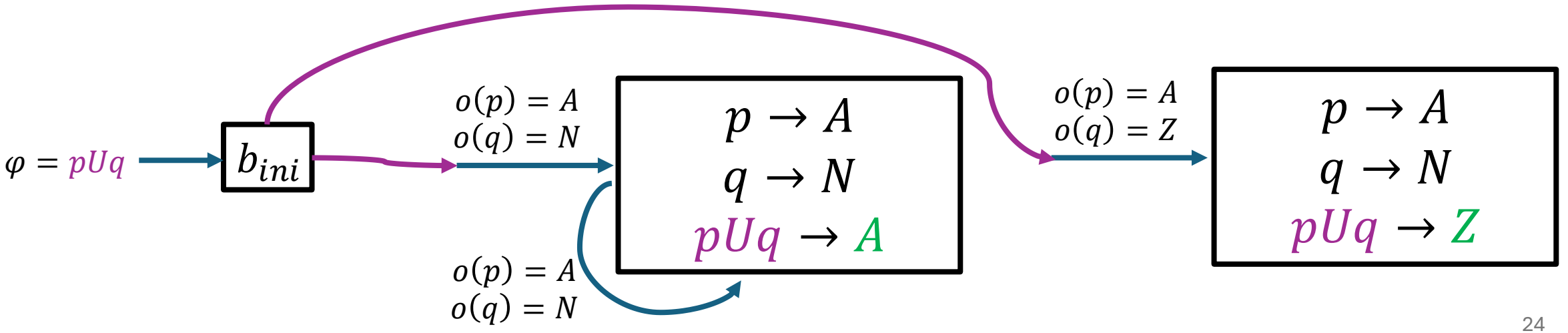
$$B_\varphi = (cs(\varphi) \cup \{b_{ini}\}, \delta_\varphi, F_\varphi)$$

- Transition function δ_φ is labeled by $o: AP \to \{A, Z, E, N\}$
- There exists $M \xrightarrow{o} M'$
  - Forall $a \in AP,\ o(a) = M'(a)$



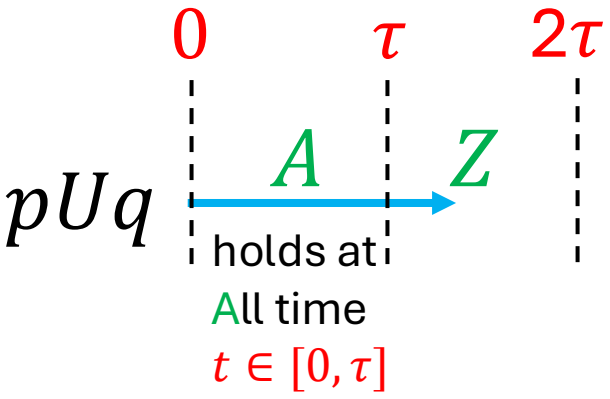
$$B_\varphi = (cs(\varphi) \cup \{b_{ini}\}, \delta_\varphi, F_\varphi)$$

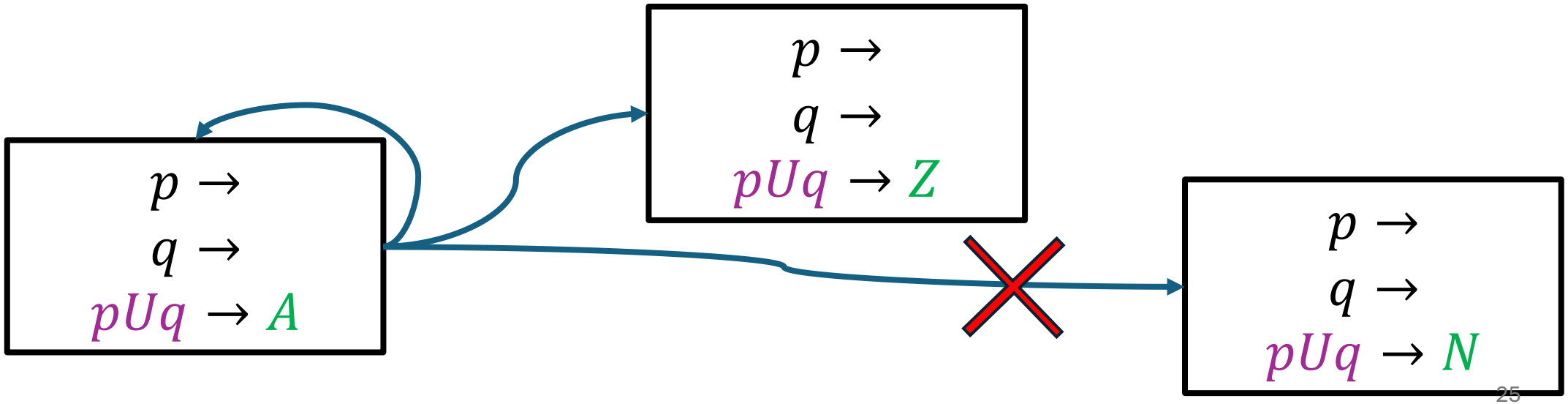
- There exists $M \xrightarrow{o} M'$
  - Forall $a \in AP,\ o(a) = M'(a)$
  - $b_{ini} \xrightarrow{o} M'$ if $M'(\varphi) \in \{A, Z\}$



$$B_\varphi = (cs(\varphi) \cup \{b_{ini}\}, \delta_\varphi, F_\varphi)$$

- There exists $M \xrightarrow{o} M'$
  - Forall $a \in AP,\ o(a) = M'(a)$
  - $b_{ini} \xrightarrow{o} M'$ if $M'(\varphi) \in \{A, Z\}$
- For all sub-formulas ψ, $M(\psi) \in \{A, E\}$ iff $M'(\psi) \in \{A, Z\}$





LTL to AP-Observation Automata

$$B_\varphi = (cs(\varphi) \cup \{b_{ini}\}, \delta_\varphi, F_\varphi)$$

- $cs(\varphi)$ is the set of "**consistent** subformula evaluation"

Which subformulas can hold in the same time interval

ψ_1	ψ_2	$\psi_1 \wedge \psi_2$	$\psi_1 \vee \psi_2$	$\psi_1 \,\mathrm{U}\, \psi_2$	$\psi_1 \,\mathrm{R}\, \psi_2$
A	A	A	A	A	A
	Z	Z	A	AZ	Z
	E	E	A	A	E
	N	N	A	**AN**	N
Z	A	Z	A	A	AZ
	Z	Z	Z	Z	Z
	E	N	A	A	EN
	N	N	Z	N	N

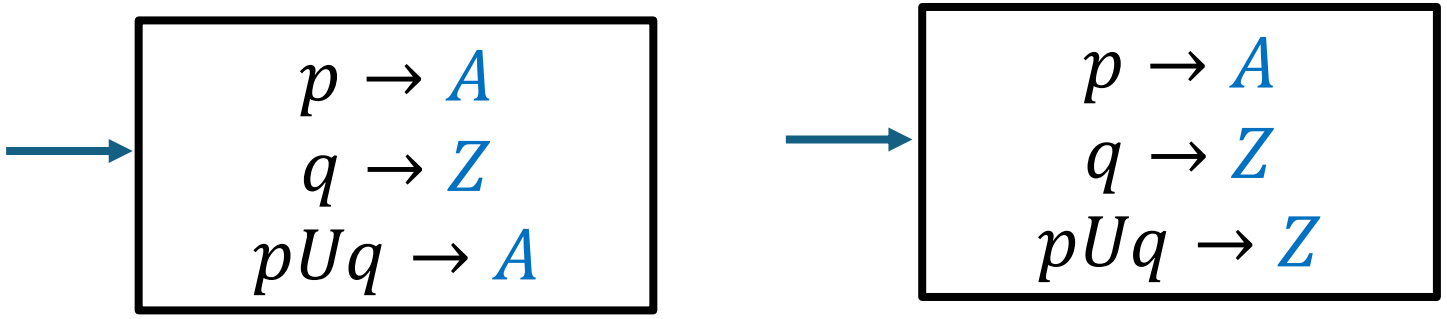
ψ_1	ψ_2	$\psi_1 \wedge \psi_2$	$\psi_1 \vee \psi_2$	$\psi_1 \,\mathrm{U}\, \psi_2$	$\psi_1 \,\mathrm{R}\, \psi_2$
E	A	E	A	A	A
	Z	N	A	AZ	N
	E	E	E	E	E
	N	N	E	EN	N
N	**A**	N	A	A	**AN**
	Z	N	Z	Z	N
	E	N	E	E	EN
	N	N	N	N	N

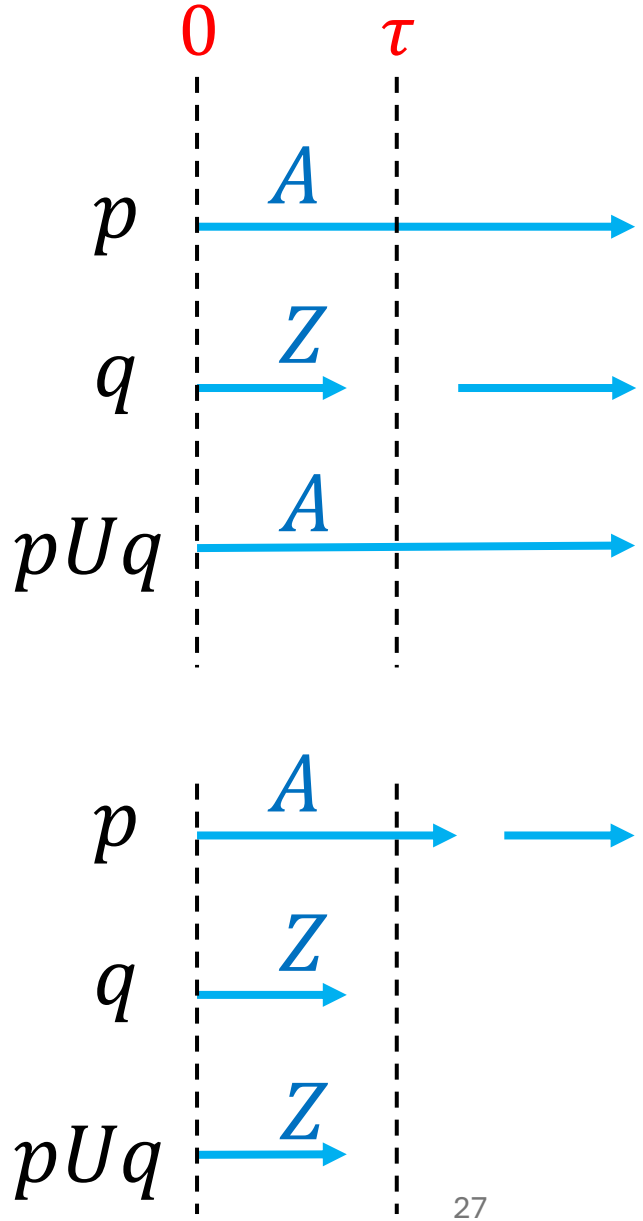
Consistency rules: (ψ_1, ψ_2 are sub-formulas of the specification φ)

LTL to AP-Observation Automata

- $cs(\varphi)$ is the set of "**consistent** subformula evaluation"

p	q			pUq
ψ_1	ψ_2	$\psi_1 \wedge \psi_2$	$\psi_1 \vee \psi_2$	$\psi_1 \mathrm{U} \psi_2$
A	A	A	A	A
	Z	Z	A	**AZ**
	E	E	A	A
	N	N	A	AN



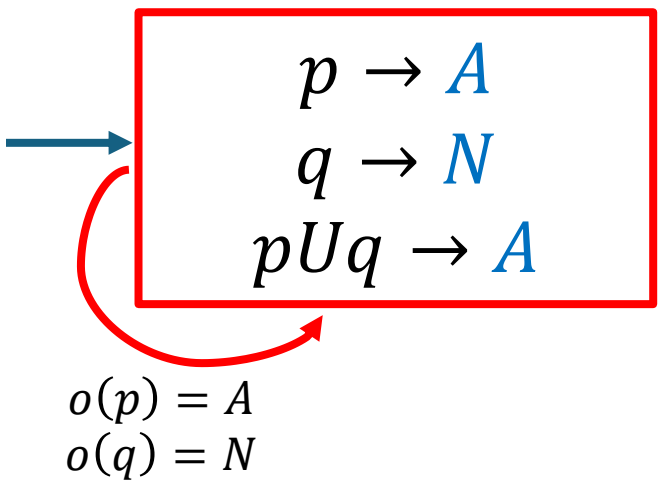


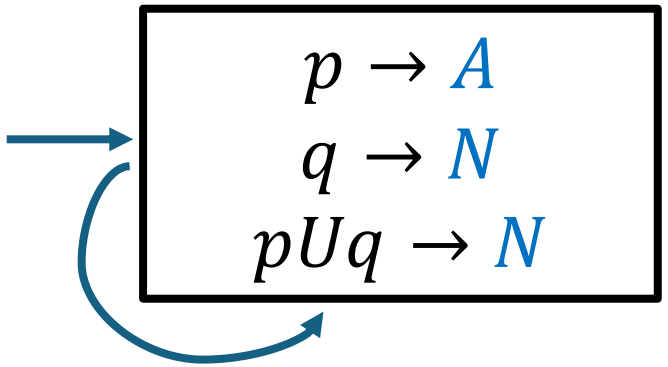
LTL to AP-Observation Automata

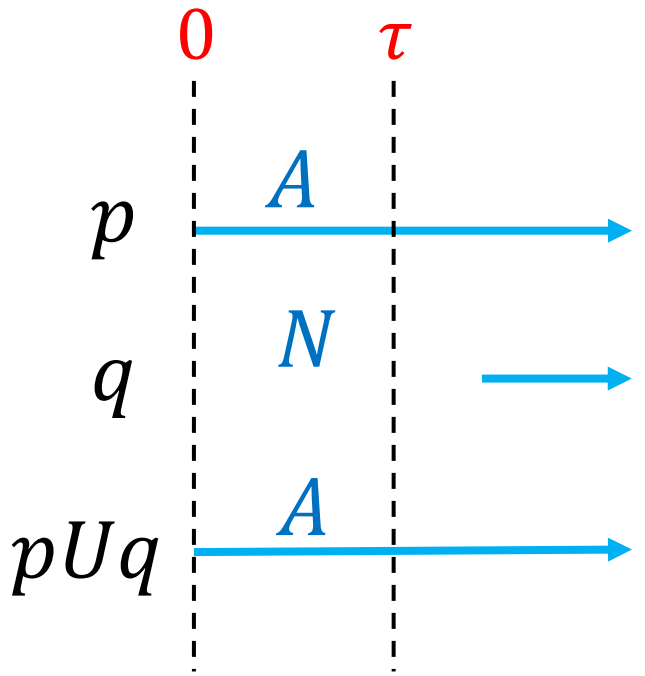
- $cs(\varphi)$ is the set of "**consistent** subformula evaluation"

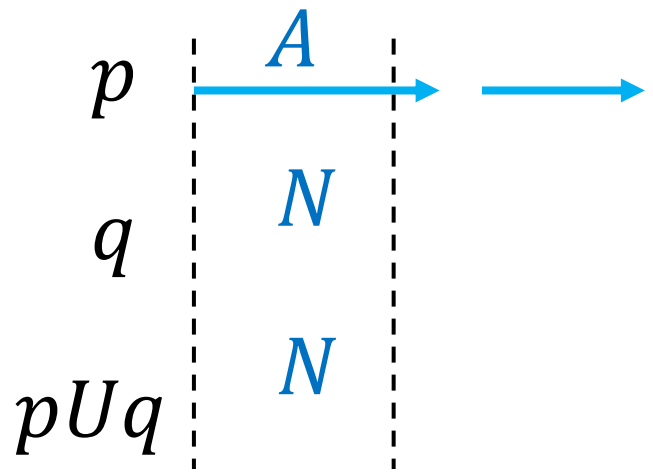
p q pUq

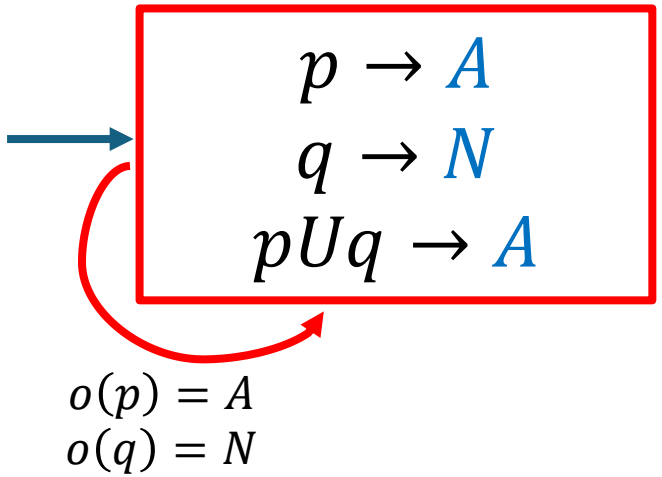
ψ_1	ψ_2	$\psi_1 \wedge \psi_2$	$\psi_1 \vee \psi_2$	$\psi_1 \mathrm{U} \psi_2$
A	A	A	A	A
	Z	Z	A	AZ
	E	E	A	A
	N	N	A	**AN**





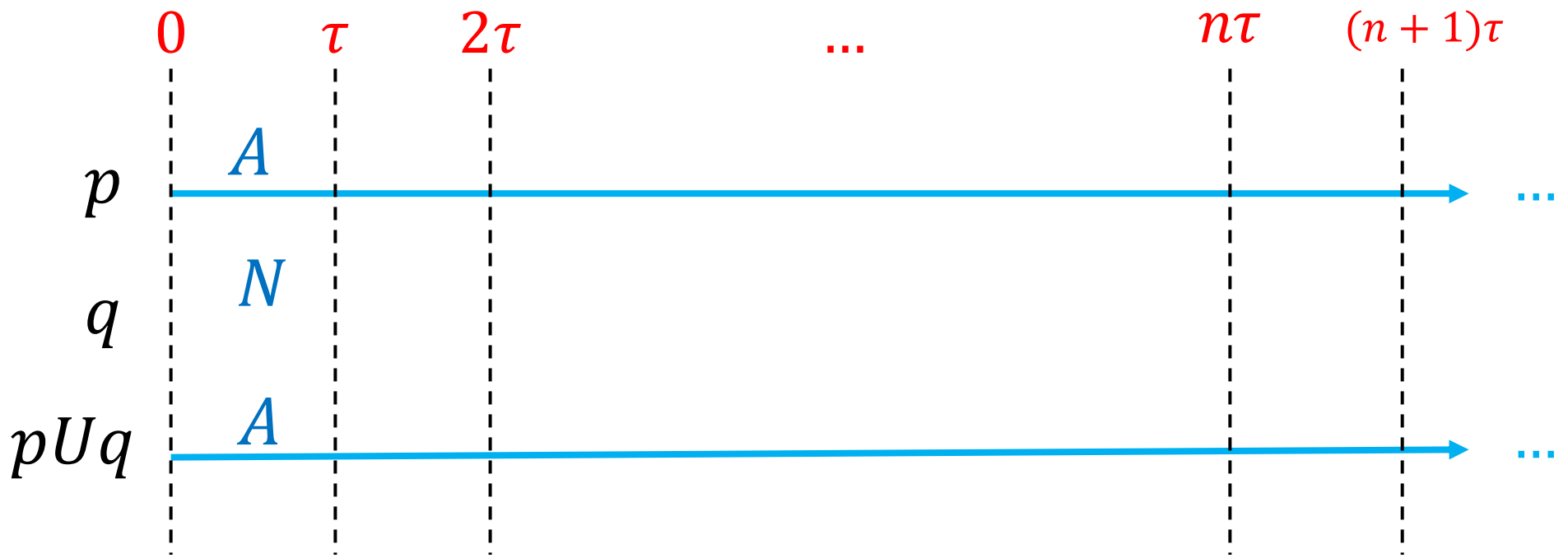






$v \in cs(\varphi)$ is a consistent valuation

$$F_{pUq} = \{v \mid v(q) \neq N \text{ and } v(pUq) \neq A\}$$

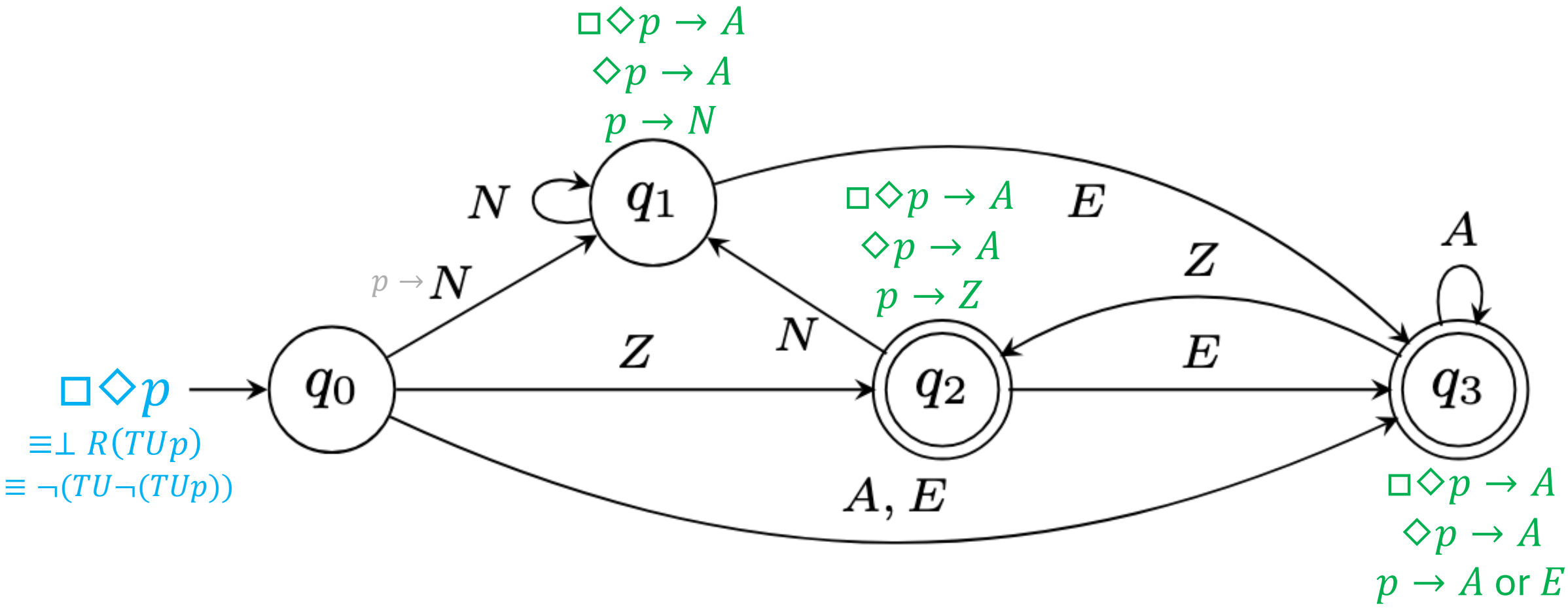


LTL to AP-Observation Automata

$$B_\varphi = (cs(\varphi) \cup \{b_{ini}\}, \delta_\varphi, F_\varphi)$$

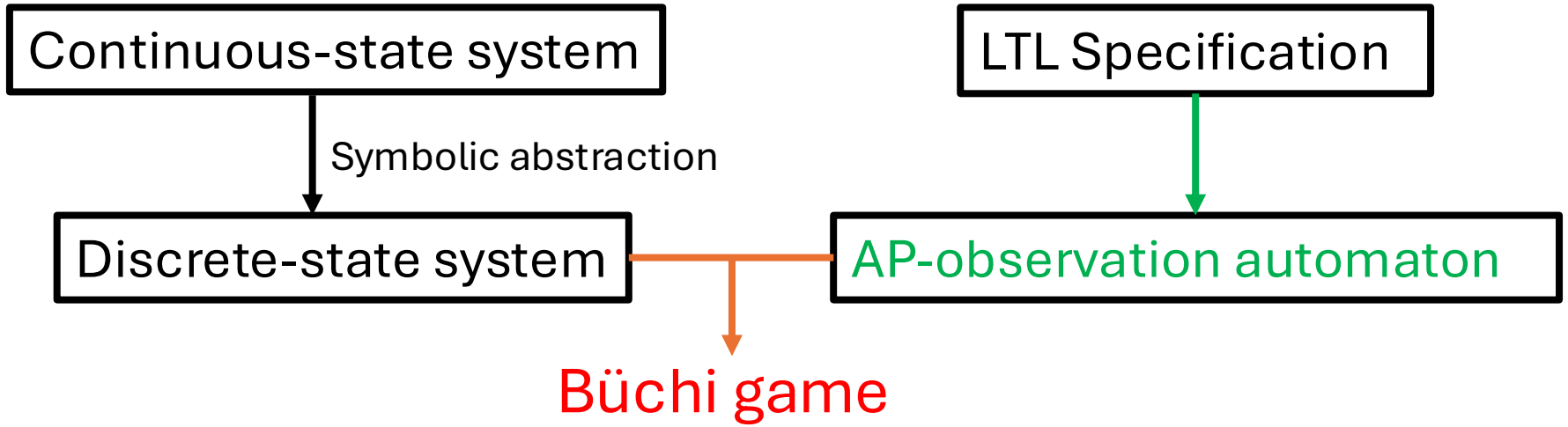
- State set:
  - $cs(\varphi)$ is the set of "consistent subformula evaluation"
  - b_{ini} is the initial state
- Transition function δ_φ is labeled by $o: AP \to \{A, Z, E, N\}$
  There exists $M \xrightarrow{o} M'$
  - Forall $a \in AP,\ o(a) = M'(a)$
  - For all sub-formula ψ, $M(\psi) \in A, E$ iff $M'(\psi) \in A, E$
  - $b_{ini} \xrightarrow{o} M'$ if $M'(\varphi) \in \{A, Z\}$
- Generalized Büchi accepting condition $\{F_{\phi_1}, F_{\phi_2}, \dots, F_{\phi_n}\}$,
  ϕ_i is an $\psi_1 U \psi_2$ or $\psi_1 R \psi_2$ subformula

LTL to AP-Observation Automata

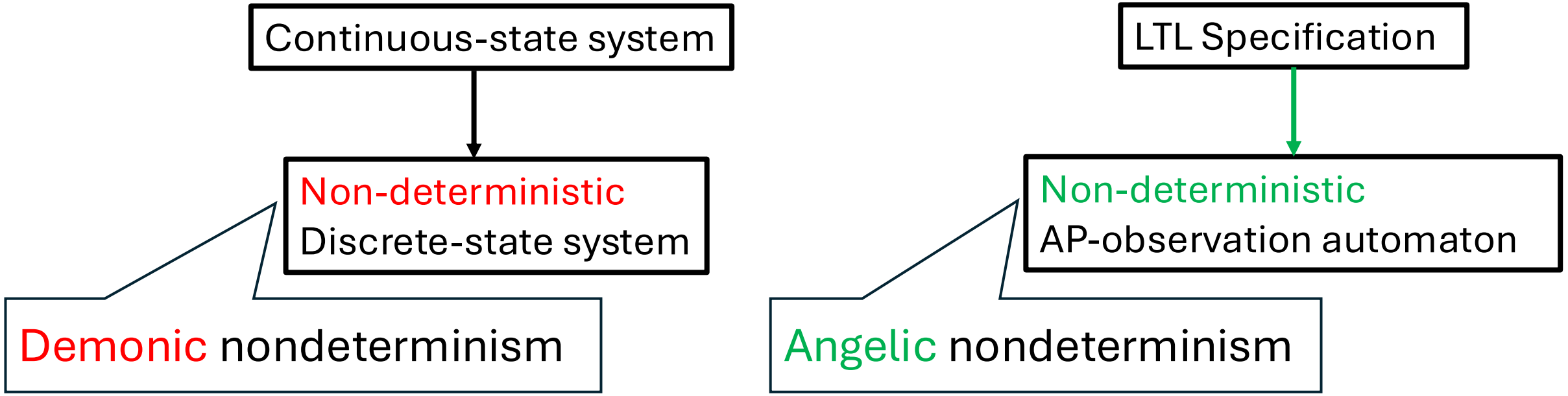


Our Contribution

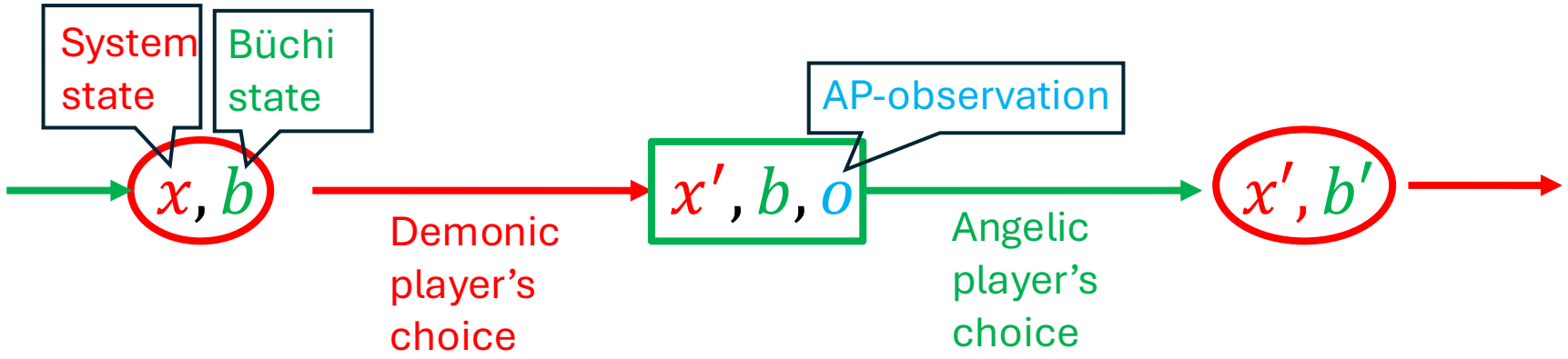
- AP-Observation Automaton: a Büchi automaton tailored specifically for abstraction-based verification & control of continuous-time system



- Two-player game-based verification

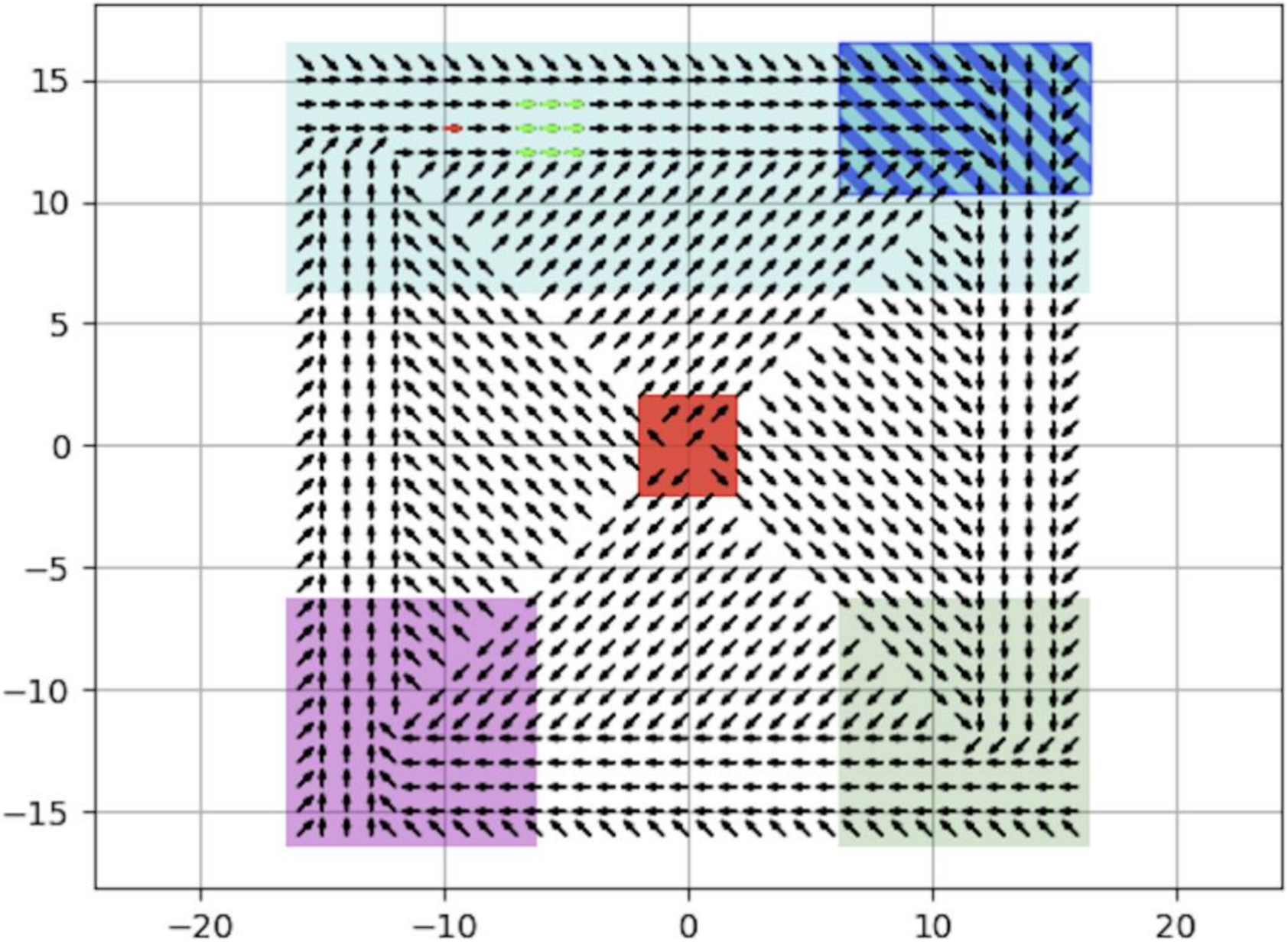


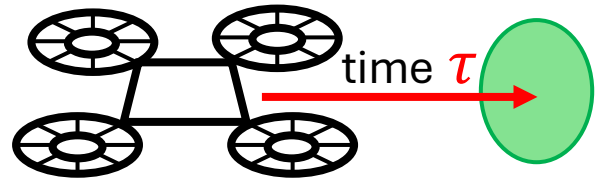
Two-player Büchi Game: Demonic VS Angelic nondeterminism



Theorem:
If the angelic player wins, the system satisfies the specification

Illustrative Example: A surveillance drone

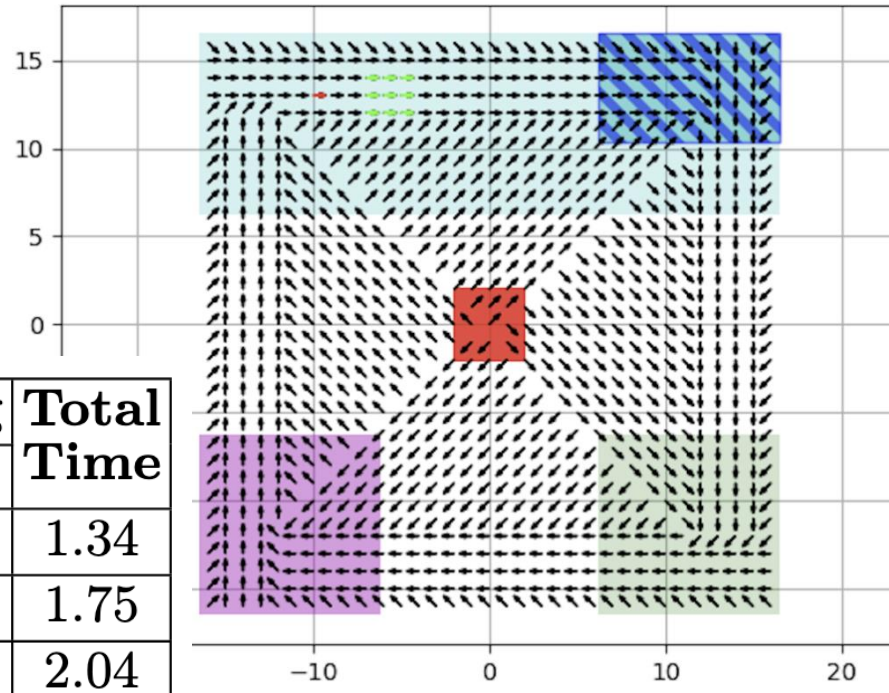




Speed $\in [3.9, 4.1]\ m/s$

Angle for heading direction may vary by up to 0.08 radians

Time interval $\tau = 1s$



(minimized)

Specification φ	$\mathcal{B}_\varphi$		Game construction		Game solving	Total Time
	Size	Time(s)	Size	Time(s)	Time(s)	
$\Box \neg r$	2	0.01	651 + 642	0.23	0.35	1.34
$\Diamond p$	5	0.01	757 + 680	0.52	0.46	1.75
$c \,\mathrm{U}\, b$	7	0.04	830 + 691	0.69	0.55	2.04
$b \,\mathrm{R}\, c$	7	0.04	814 + 685	0.69	0.53	2.03
$\Diamond\Box \neg r$	6	0.02	1,933+1,924	0.60	2.05	3.44
$\Box\Diamond g$	7	0.02	4,640+2,631	1.93	2.48	5.20
$\Diamond(g \wedge \Diamond p)$	33	0.40	4,856+2,687	8.98	3.05	13.19
$\Box \neg r \wedge (\Diamond p \wedge \Diamond c)$	46	51.39	7,723+4,144	29.66	7.54	89.36
$\Box \neg r \wedge \Diamond(g \wedge \Diamond p)$	49	53.86	7,279+4,030	25.49	7.06	87.18

Conclusion

- We consider abstraction-based verification and control of continuous-time continuous-state systems
- We propose
  - AP-Observation Automaton: a Büchi automaton tailored specifically for abstraction-based verification & control of continuous-time system
  - Two-player game-based Verification
- Future work
  - Symbolic control
  - Weaken the assumptions
    consider other type of AP-observations