Converting LTL to Buchi

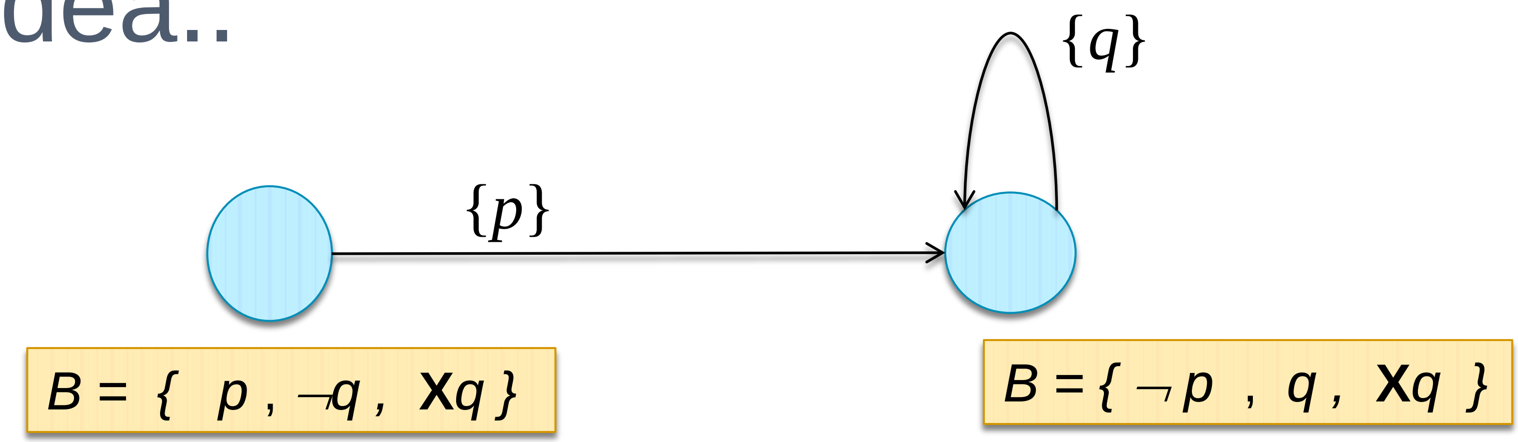
Wishnu Prasetya

wishnu@cs.uu.nl
www.cs.uu.nl/docs/vakken/pv

Converting LTL to Buchi

- Given an LTL formula φ, construct a Buchi automaton M that accepts the same sentences as φ.
  - Recall: “sentence” is a sequence of ‘something’, each is a set of propositions. Sentence = (abstract) execution.
- Steps:
  - Construct GNBA
  - Convert to NBA
  - Optimize

Idea..



To help us, each state *s* will be labeled with an “observation” *B*. It is a consistent set of formulas. Any infinite sequence starting from s must satisfy all formulas in *B*.

The set of candidate “observations” for a given φ is finite; and we can figure out how to connect them with arrows.

Restricting to X/U

- All LTL formulas can be expressed with just **X** and **U**.

<>φ	=	true **U** φ
[] φ	=	$\neg$ (**<>** φ)
φ **W** ψ	=	**[]** φ $\vee$ φ **U** ψ

- Let's assume that your input formula is expressed in this form of LTL.

Closure

- **closure**(φ) is the set of all
  - subformulas of φ (incl φ itself)
  - negations of subformulas

- Example: $\varphi = p \,\mathbf{U}\, q$

$$\boxed{\textbf{closure}(\varphi) \quad = \quad \{\ p, q, \neg p, \neg q, p \,\mathbf{U}\, q\ , \neg(p \,\mathbf{U}\, q)\ \}}$$

- Only the value of the formulas in the closure can affect the value of φ.

Observation

- Example: $\varphi = p \mathbf{U} q$

$$\textbf{closure}(\varphi) = \{ p, q, \neg p, \neg q, p \mathbf{U} q, \neg(p \mathbf{U} q) \}$$

- An ‘*observation*’ B is in principle a subset of the closure, but we want it to be ‘consistent’ and ‘maximal’.
  - $\{ p, q, p \mathbf{U} q \}$ → OK
  - $\{ p, \neg p \}$ → inconsistent
  - { p } → not maximal

Consistency of the *B's*

- An observation B must be consistent with respect to propositional logic:
  - f and $\neg f$ cannot be both in B
  - $f \wedge g \in B \Rightarrow f, g \in B$

- Locally consistent with respect to "until". For any f **U** $g \in$ **closure**(φ) :
  - $g \in B \Rightarrow f\ \mathbf{U}\ g \in B$
  - $f\ \mathbf{U}\ g \in B$ and $g \notin B \Rightarrow f \in B$

Maximality

- Every observation B should be *maximal* $\rightarrow$

For every $f \in$ **closure**(φ), either $f \in B$ or $\neg f \in B$.

- Ex. $\varphi = p \, \mathbf{U} \, q$

closure(φ) = $\{ p, q, \neg p, \neg q, p \, \mathbf{U} \, q, \neg(p \, \mathbf{U} \, q) \}$

8 maximal subsets, 3 are inconsistent.

$\{ p, q, p \cup q \}$

$\{ p, q, \neg(p \cup q) \}$

$\{ p, \neg q, p \cup q \}$

$\{ p, \neg q, \neg(p \cup q) \}$

$\{ \neg p, q, p \cup q \}$

$\{ \neg p, q, \neg(p \cup q) \}$

$\{ \neg p, \neg q, p \cup q \}$

$\{ \neg p, \neg q, \neg(p \cup q) \}$

Constructing the automaton A_φ

- **States**: observations from **closures**(φ)

- **Initial states**: all states that contain φ

- **Arrows**: for any pairs observations *B,C* add this arrow:

  $$B \;\text{---}\; V \longrightarrow\; C$$

  - If this arrow is 'consistent'
  - *V* = the set of propositions in *B*.

- **Acceptance** states?

The arrows

- $B \;\text{——}\; V \longrightarrow\; C$ is consistent if :

> - $\mathbf{X}f \in B \;\Rightarrow\; f \in C$
> - $f\,\mathbf{U}\,g \in B \;\Leftrightarrow\; g \in B$
>   or $f \in B$ and $f\,\mathbf{U}\,g \in C$

- Example:

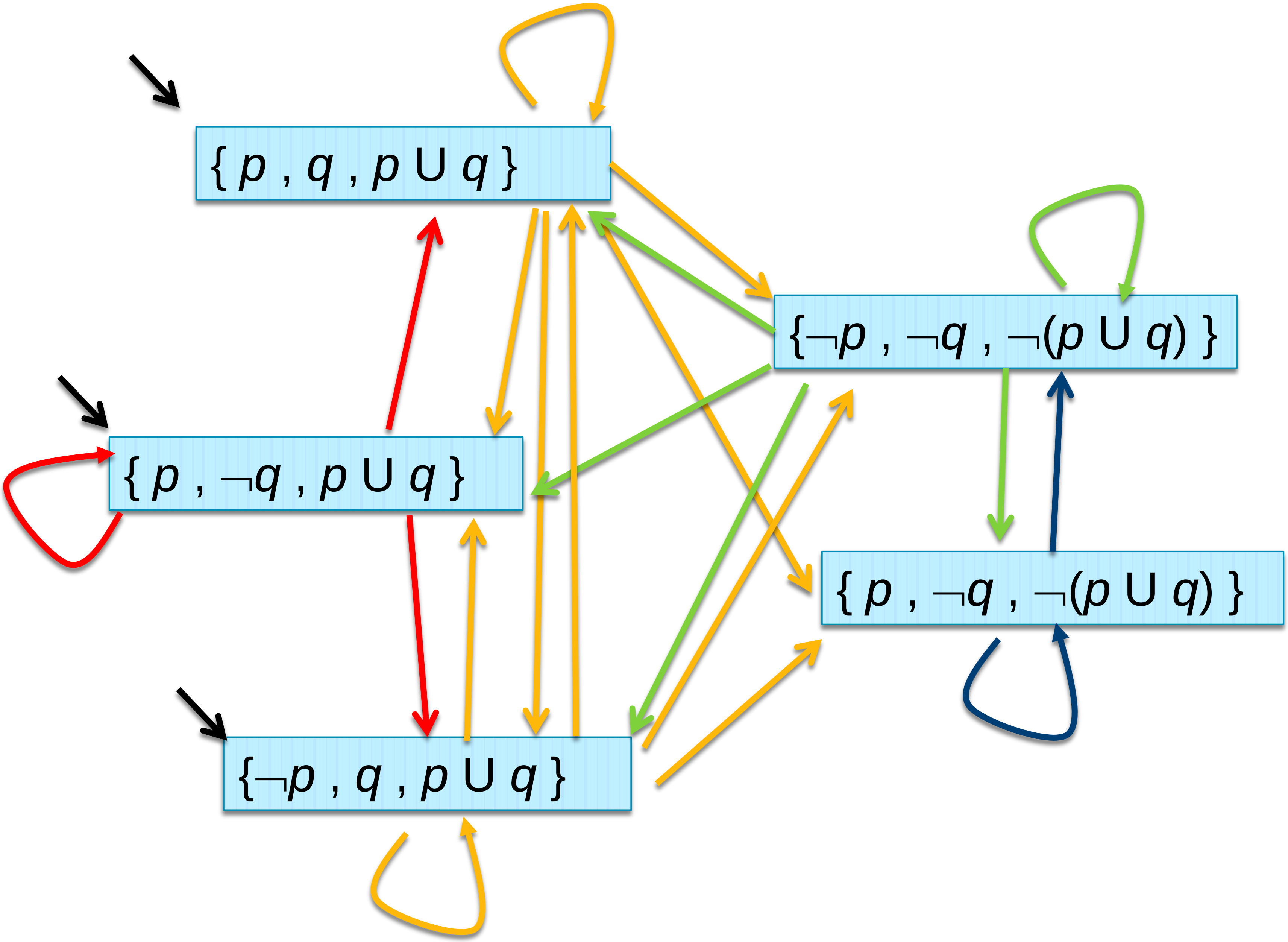
$\{ p , q , p \cup q \}$

$\{ p , \neg q , p \cup q \}$

$\{\neg p , q , p \cup q \}$

$\{ p , \neg q , \neg(p \cup q) \}$

$\{\neg p , \neg q , \neg(p \cup q) \}$


{ p , q , p U q }
{¬p , ¬q , ¬(p U q) }
{ p , ¬q , p U q }
{ p , ¬q , ¬(p U q) }
{¬p , q , p U q }

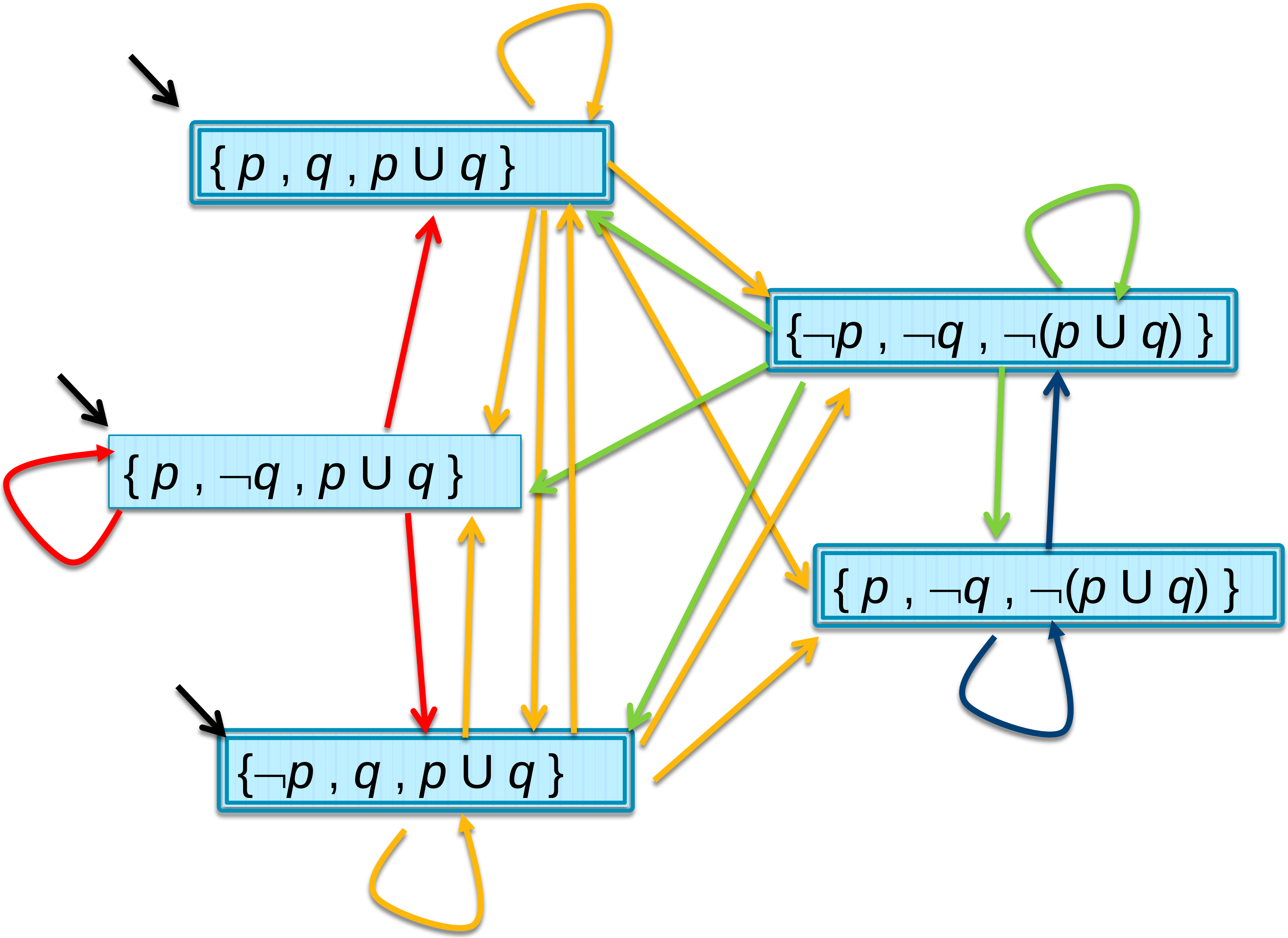
Enforcing eventuality

- For each $f\ \mathbf{U}\ g \in$ **closure**(φ), add an accepting group:

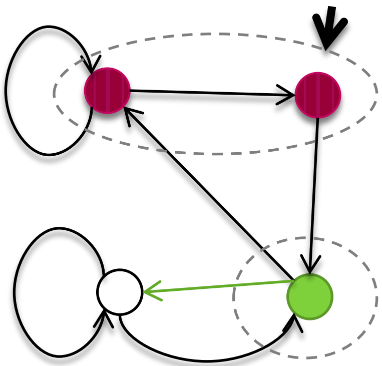
$$\mathbf{F}\ (f\ \mathbf{U}\ g) \;=\; \{\ B \mid B \in \mathbf{Q}\ \wedge\ g \in B\ \} \;\cup\; \{\ B \mid B \in \mathbf{Q}\ \wedge\ f\ \mathbf{U}\ g \notin B\ \}$$

where Q is the set of states of GNBA of φ that we are constructing.

(btw = the set of all ‘observations’)


{ p , q , p U q }
{¬p , ¬q , ¬(p U q) }
{ p , ¬q , p U q }
{ p , ¬q , ¬(p U q) }
{¬p , q , p U q }

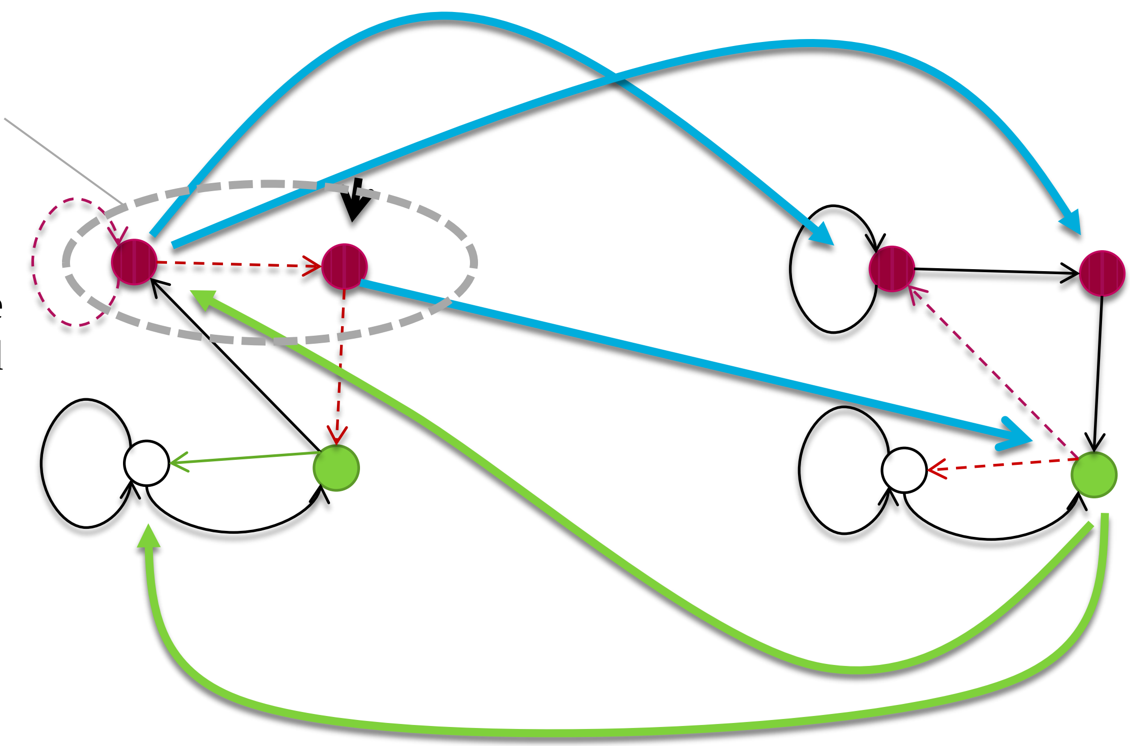
From GNBA to NBA



GNBA with 2x accepting groups.

single accentance
group of the new
automaton

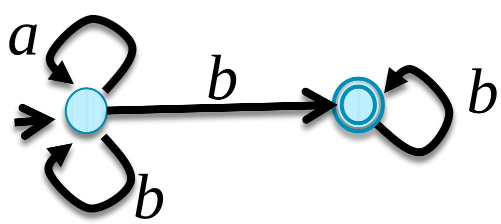
dashed-red arrows are
dropped



Can we make it deterministic?

- In ordinary automaton, DFA can be converted to an equivalent NDFA (equivalent = generating the same sentences).

- For Buchi?



No *deterministic* Buchi can generate the sentences of this Buchi

- NBA is really more powerful than DBA.

How big are they?

- NGBA generated by our procedure → $|M| = 2^{|\varphi|}$.
- Converting to GBA multiplies the number of states with C+1, where C is the number of **U** in φ
- There are LTL formulas of polynomial size, whose NBA will have at least exponential number of states.