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 ‘symbols’, each is a set of propositions. Sentence = (abstract) execution.
- Steps:
  - Construct GNBA
  - Convert to NBA
  - Optimize

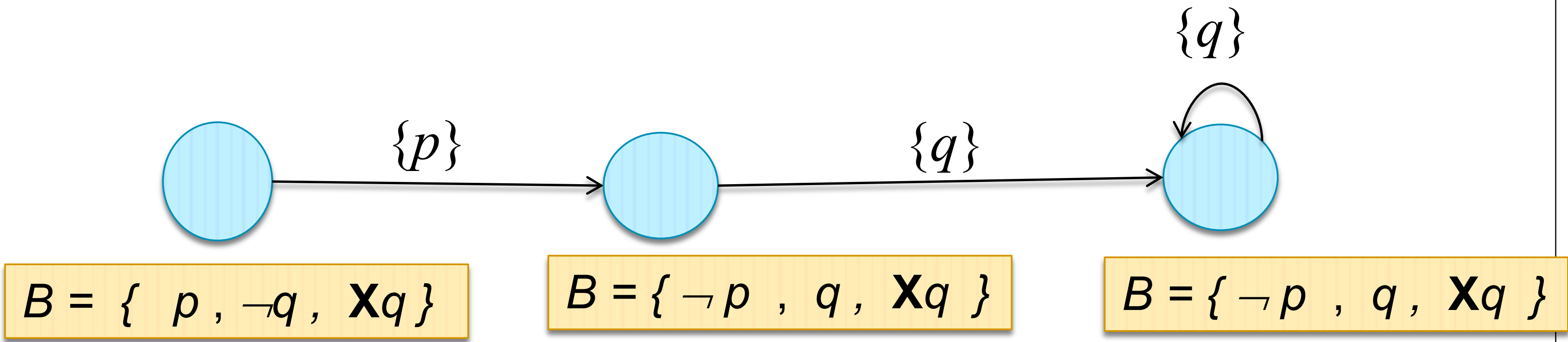
Restricting to X/U

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

$$<>\varphi \quad = \quad \text{true} \ \mathbf{U} \ \varphi$$
$$[]\ \varphi \quad = \quad \neg\, (\, <> \neg\, \varphi\,)$$
$$\varphi \ \mathbf{W} \ \psi \quad = \quad []\ \varphi \ \vee \ \varphi \ \mathbf{U} \ \psi$$

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

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.

Closure

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

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

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

- 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
  - $f \wedge g \in B \Rightarrow f, g \in B$

- Consistent with respect to "until". For any $f\, \mathbf{U}\, g \in$ **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$(\varphi) = \{ p, q, \neg p, \neg q, p \,\mathbf{U}\, q, \neg(p \,\mathbf{U}\, q) \}$

$\{ p, q, p \,\mathrm{U}\, q \}$

$\{ p, q, \neg(p \,\mathrm{U}\, q) \}$

$\{ p, \neg q, p \,\mathrm{U}\, q \}$

$\{ p, \neg q, \neg(p \,\mathrm{U}\, q) \}$

$\{ \neg p, q, p \,\mathrm{U}\, q \}$

$\{ \neg p, q, \neg(p \,\mathrm{U}\, q) \}$

$\{ \neg p, \neg q, p \,\mathrm{U}\, q \}$

$\{ \neg p, \neg q, \neg(p \,\mathrm{U}\, q) \}$

5 Observations (blue). Red ones may look like observations, but are inconsistent.

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:

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

  - *V* = the set of postive (not in negation) propositions in *B*.
  - If this arrow is 'consistent'

- **Acceptance** states?

The arrows

- B —–– V –→ C is consistent if (1) :

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

- Example:

$\{\, p , q , p \,\mathrm{U}\, q \,\}$

$\{p\}$

$\{\, p , \neg q , p \,\mathrm{U}\, q \,\}$

$\{p\}$

$\{\, p , \neg q , \neg(p \,\mathrm{U}\, q) \,\}$

$\{p\}$

$\{\neg p , q , p \,\mathrm{U}\, q \,\}$

$\{\neg p , \neg q , \neg(p \,\mathrm{U}\, q) \,\}$

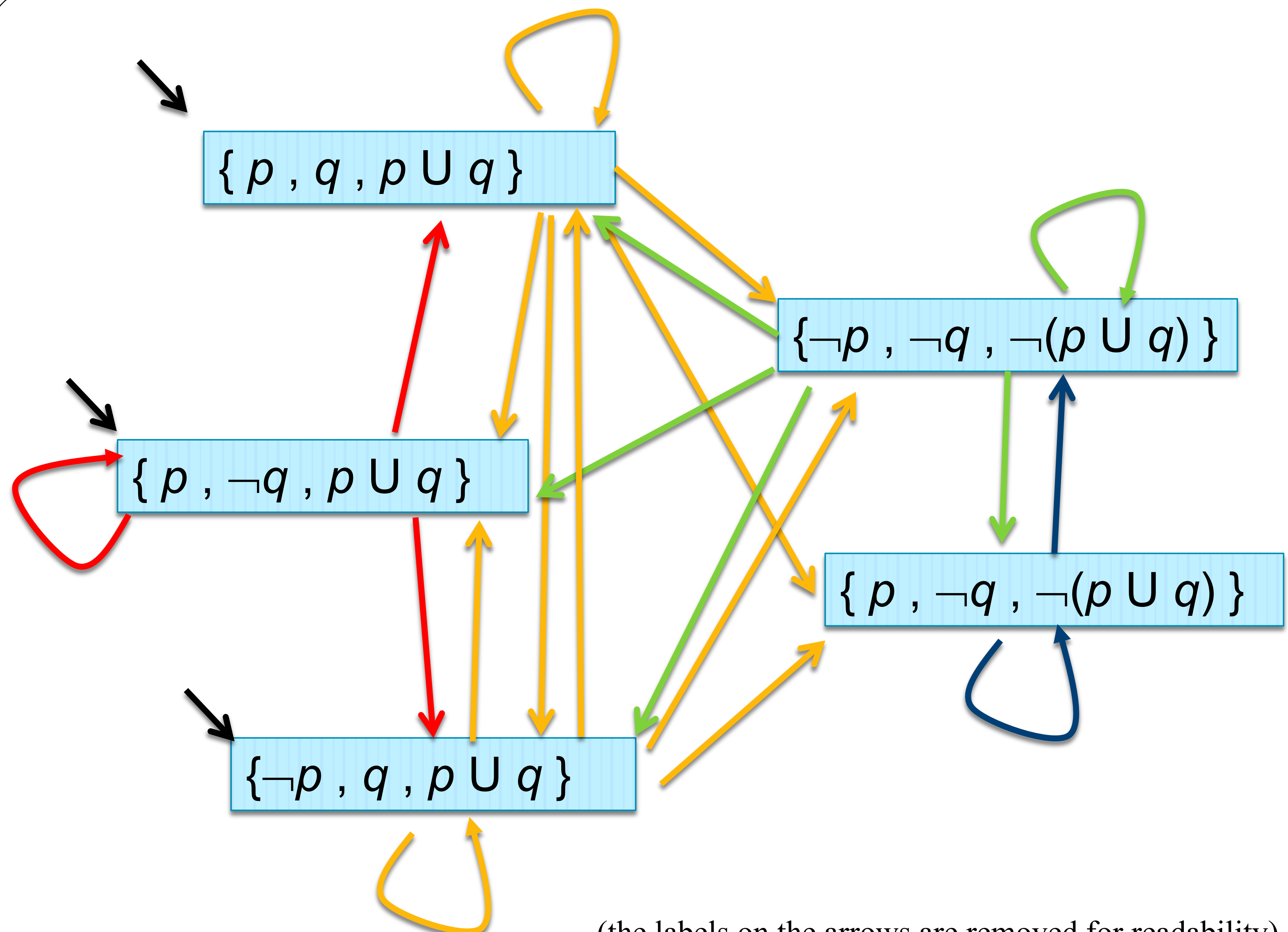
The arrows

- $B \; \text{---} \; V \longrightarrow \; C$ is consistent if (1) :

> • $\mathbf{X}f \in B \;\Rightarrow\; f \in C$
> • $f\,\mathbf{U}\,g \in B \;\Rightarrow\; g \in B$
> or $(\, f \in B$ and $f\,\mathbf{U}\,g \in C\,)$

- Furthermore (2) :

> •“$\neg\mathbf{X}f$” $\in B \;\Rightarrow$ “$\neg f$ ” $\in C$
> •“$\neg(f\,\mathbf{U}\,g)$” $\in B \;\Rightarrow$
> (“$\neg f$” $\in B$ and “$\neg g$” $\in B$)
> or
> ($f \in B$ and “$\neg g$” $\in B$ and “$\neg(f\,\mathbf{U}\,g)$” $\in C$))



(the labels on the arrows are removed for readability)

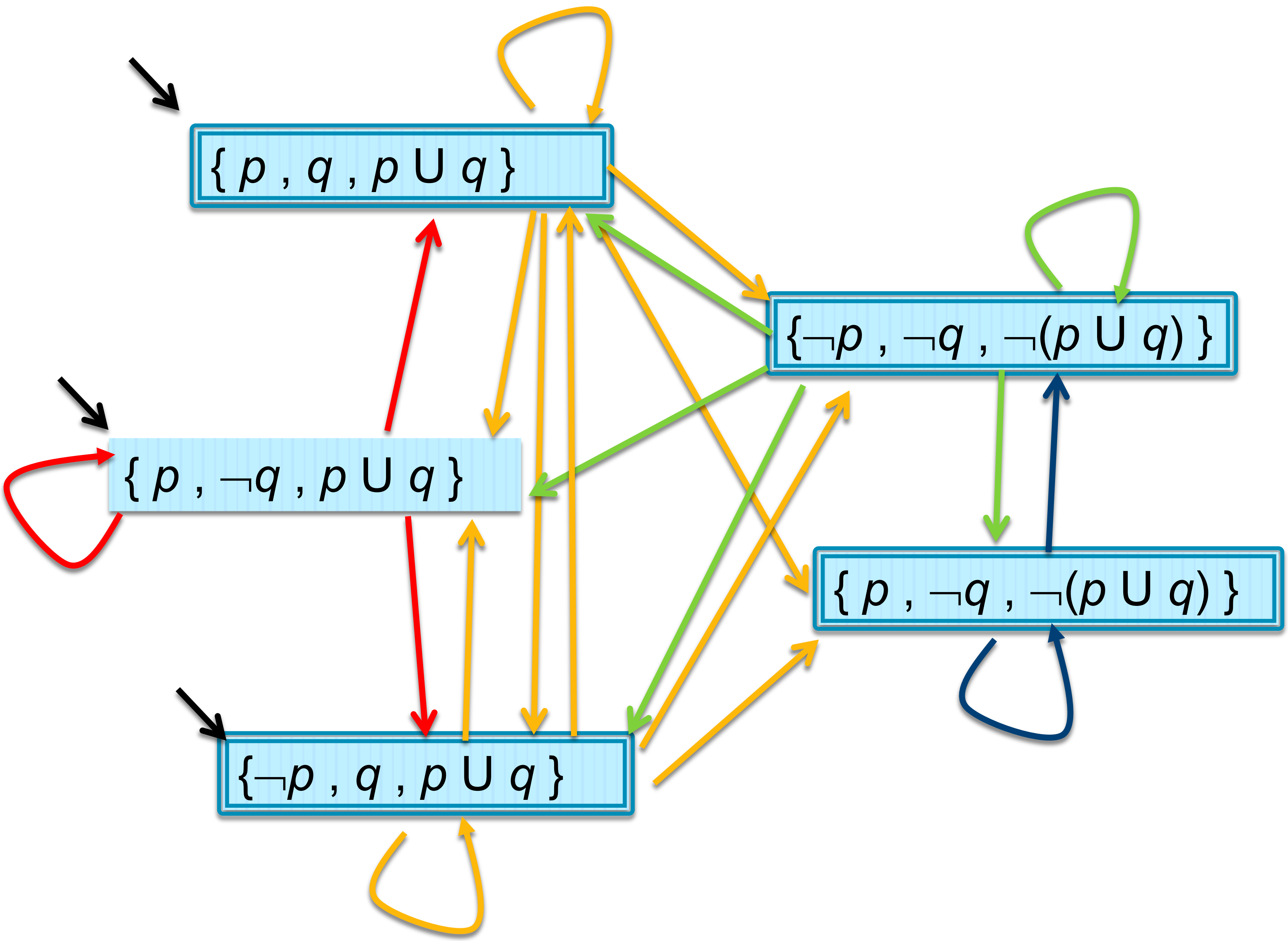
Enforcing eventuality

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

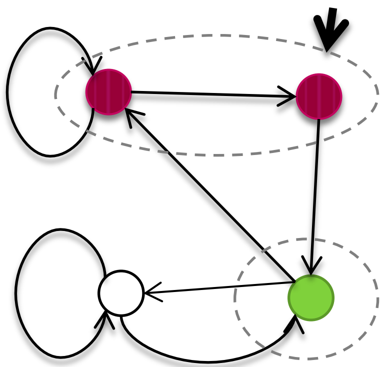
$$\mathbf{F}\ (f\ \mathbf{U}\ g) = \{\ B \mid B \in \mathbf{Q}\ \wedge\ g \in B\ \} \cup \{\ B \mid B \in \mathbf{Q} \wedge\ f\ \mathbf{U} g \notin B\ \}$$

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

(Q = 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 }

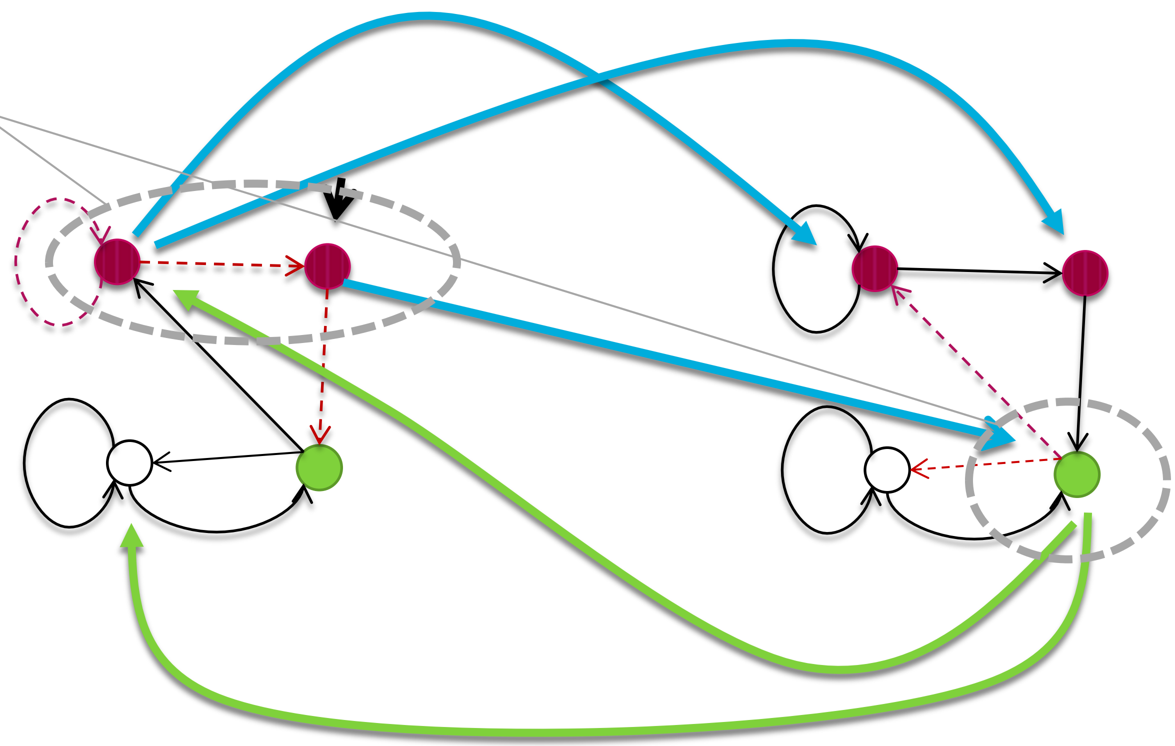
From GNBA to NBA



GNBA with 2x accepting groups.

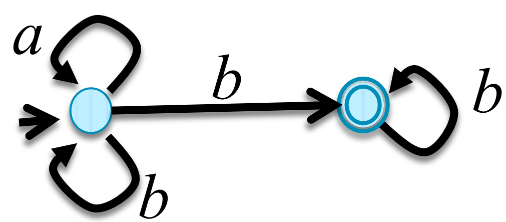
single accentance group of the new automaton

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?

- NDGBA generated by our procedure → $|M| = 2^{|\varphi|}$.
- Converting to NDBA multiplies the number of states with *C*, where *C* is the number of **U** in φ