UvA

## Multi-Agent Systems

# Alternating-Time Temporal Logic

Mina Young Pedersen
m.y.pedersen@uva.nl

Fall 2025

# What is Alternating-Time Temporal Logic (ATL)?

- Alternating-Time Temporal Logic (ATL) is a logic to reason about agents and their abilities over time
  - Temporal logic meets game theory
  - Or, Coalition Logic meets Computation-Tree Logic
    - The language of ATL has both operators $\langle\langle A \rangle\rangle \phi$ and temporal operators *X*, *G*, *F*, and *U*

# What is Alternating-Time Temporal Logic (ATL)?

- In Coalition Logic we could only reason about agents' outcomes in one step
- In ATL, we can reason in multiple steps, like in extensive form games
  - And we can express things like:
    - "Agents can achieve the goal": $\langle\langle agents \rangle\rangle F$goal
    - "Agents can enforce a safety property": $\langle\langle agents \rangle\rangle G$safe

# Concurrent Game Models

- The most common way of defining the semantics of Alternating-Time Temporal Logic is by using **concurrent game models**

- As before, we define a set $\mathsf{At}$ of atoms

# Concurrent Game Models

**Definition (Concurrent Game Structures and Models)**

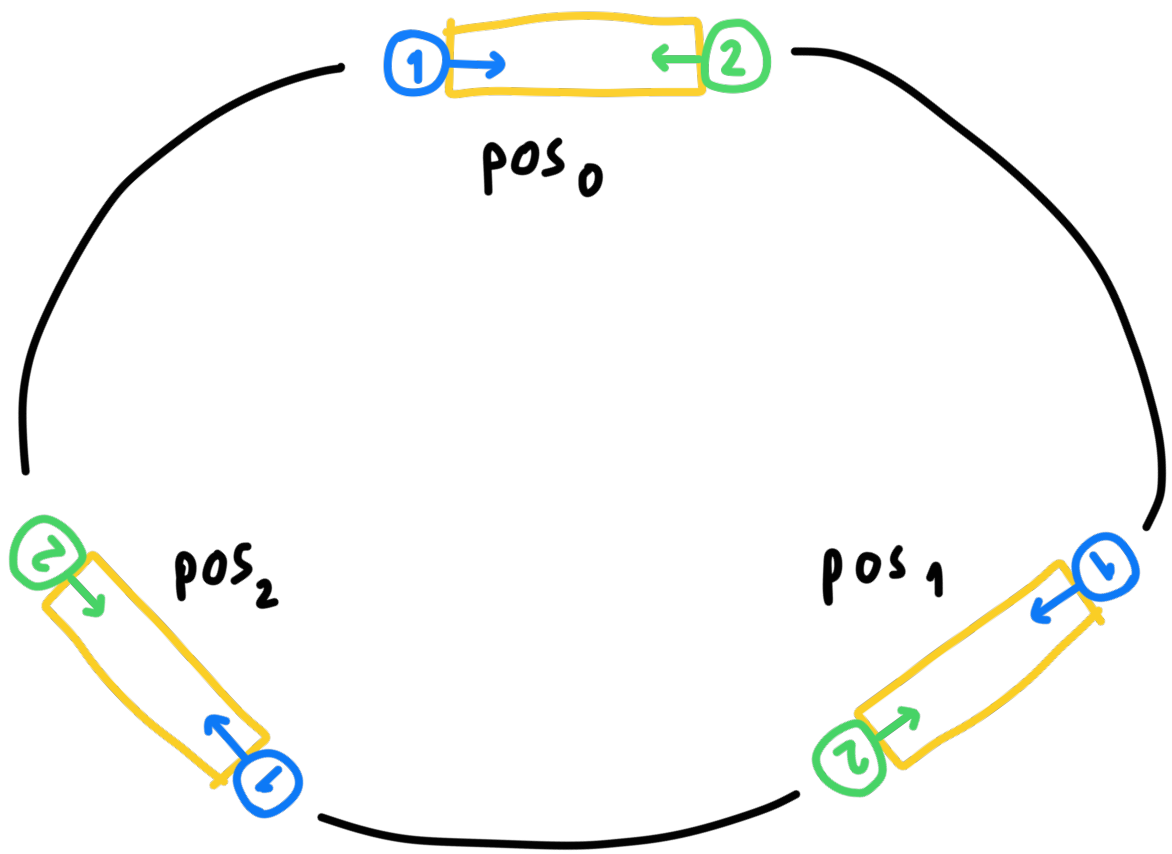
A concurrent game model (CGM) is a tuple $M = (\mathbb{A}gt, St, Act, d, o, V)$, where:

- $\mathbb{A}gt = \{1, \ldots, k\}$ is a non-empty finite set of **players**, or **agents;**
- $St$ is a non-empty set of **states;**
- $Act$ is a non-empty set of (atomic) **actions;**
- $d : (\mathbb{A}gt \times St) \to \mathscr{P}(Act) \setminus \{\emptyset\}$ is a function that defines the **set of actions available to each player at each state;**
- $o$ is a (deterministic) **transition function** that assigns a unique outcome state $o(q, \alpha_1, \ldots, \alpha_k)$ to each state $q$ and each tuple of actions $\langle \alpha_1, \ldots, \alpha_k \rangle$ that can be executed in $q$;
- $V$ : At $\to \mathscr{P}(St)$ is a valuation function.

A concurrent game structure (CGS) $S = (\mathbb{A}gt, St, Act, d, o)$ is a concurrent game model without a valuation.

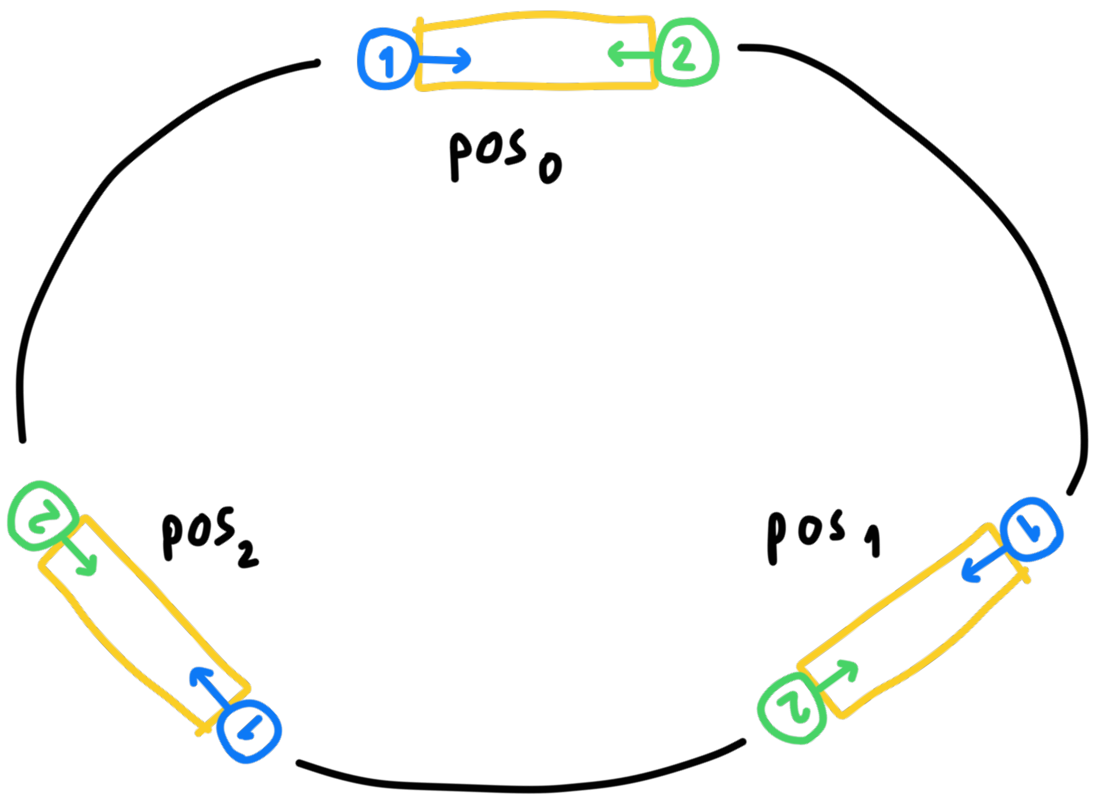
# Example: Robot and Carriage

Suppose you have two robots that push a carriage from opposite sides:




The carriage can move clockwise or anticlockwise, or it can remain in the same place

# Example: Robot and Carriage




Each robot can either push or refrain from pushing (wait)

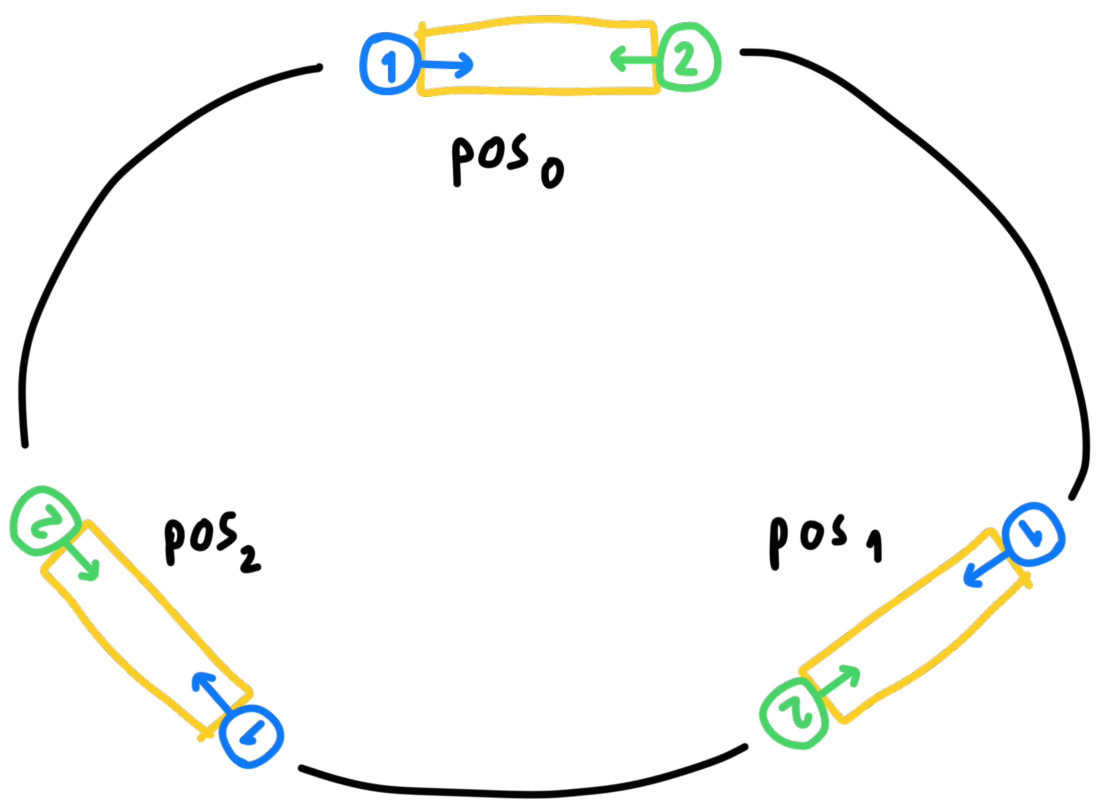
The robots use the same force:

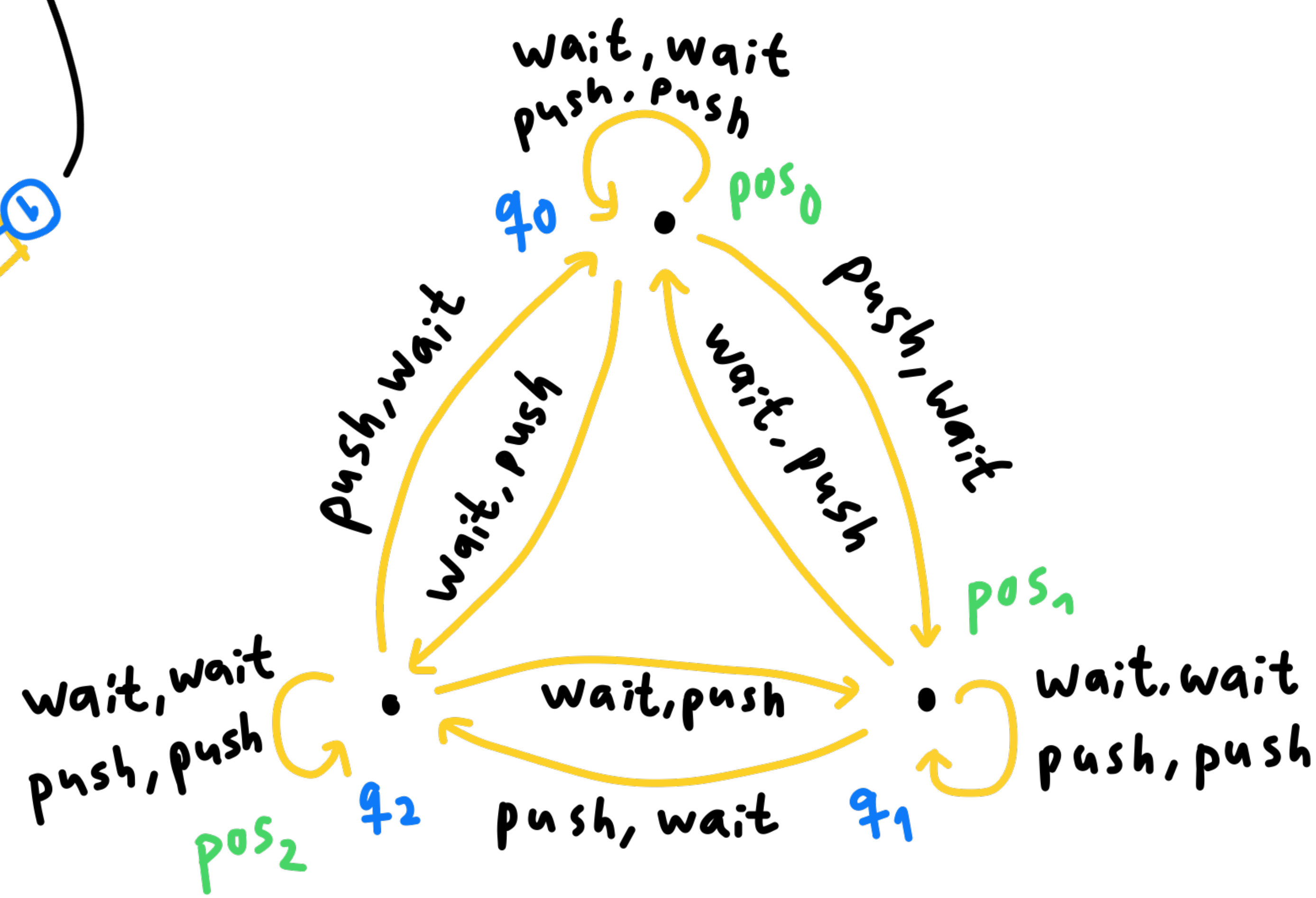
If they both push, or both wait, the carriage does not move

If Robot 1 pushes and Robot 2 waits, the carriage does clockwise, and vice versa

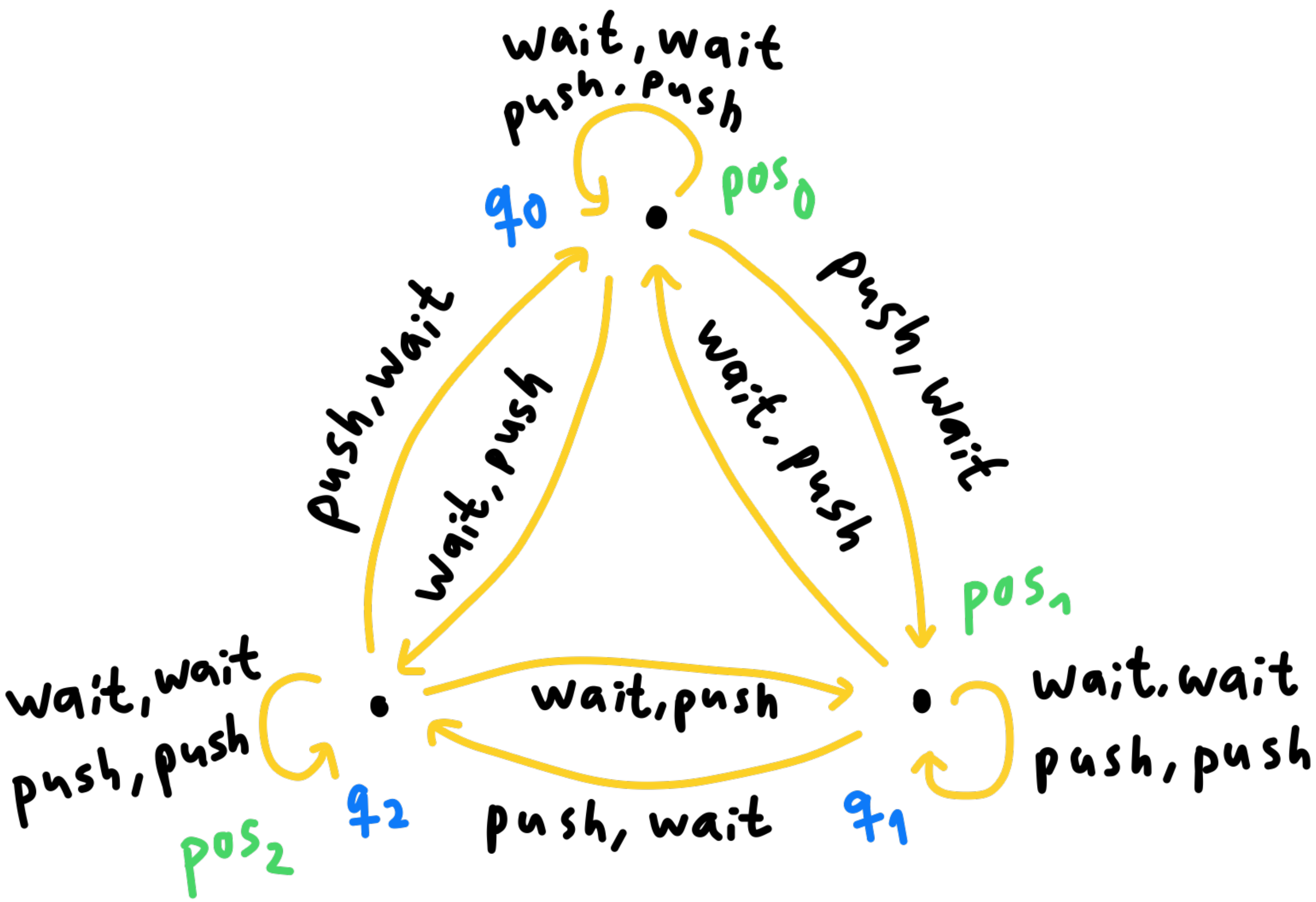
# Example: Robot and Carriage

Let's model this scenario as a concurrent game model



# Example: Robot and Carriage




$$V(pos_1) = \{q_1\}$$

**(…)**

$$\mathbb{A}gt = \{1,2\}$$

$$St = \{q_0, q_1, q_2\}$$

$$Act = \{\textsf{push}, \textsf{wait}\}$$

$$d(1,q_1) = \{\textsf{push}, \textsf{wait}\}$$

**(…)**

$$o(q_0, \textsf{push}, \textsf{wait}) = q_1$$

**(…)**

# Representing Games as Concurrent Game Models

- CGMs have a close relationship to strategic, repeated, and extensive-form games with perfect information
- The major difference is that CGMs lack the notion of payoffs
  - But those can be embedded in a natural way, using atomic propositions
  - Formally:
    - Consider any extensive-form game $G$ and let $U$ be the set of all possible payoffs in the game
    - For each value $v \in U$ and each agent $a \in \mathbb{A}gt$, we introduce a proposition $u_a^v$ and fix $q \in V(u_a^v)$ iff $a$ gets a payoff $v$ in $q$

# Prisoner's Dilemma

- Remember the Prisoner's Dilemma?

| | Stick to the story | Rat |
|---|---|---|
| Stick to the story | (-100,-100) | (-1000,0) |
| Rat | (0,-1000) | (-500,-500) |

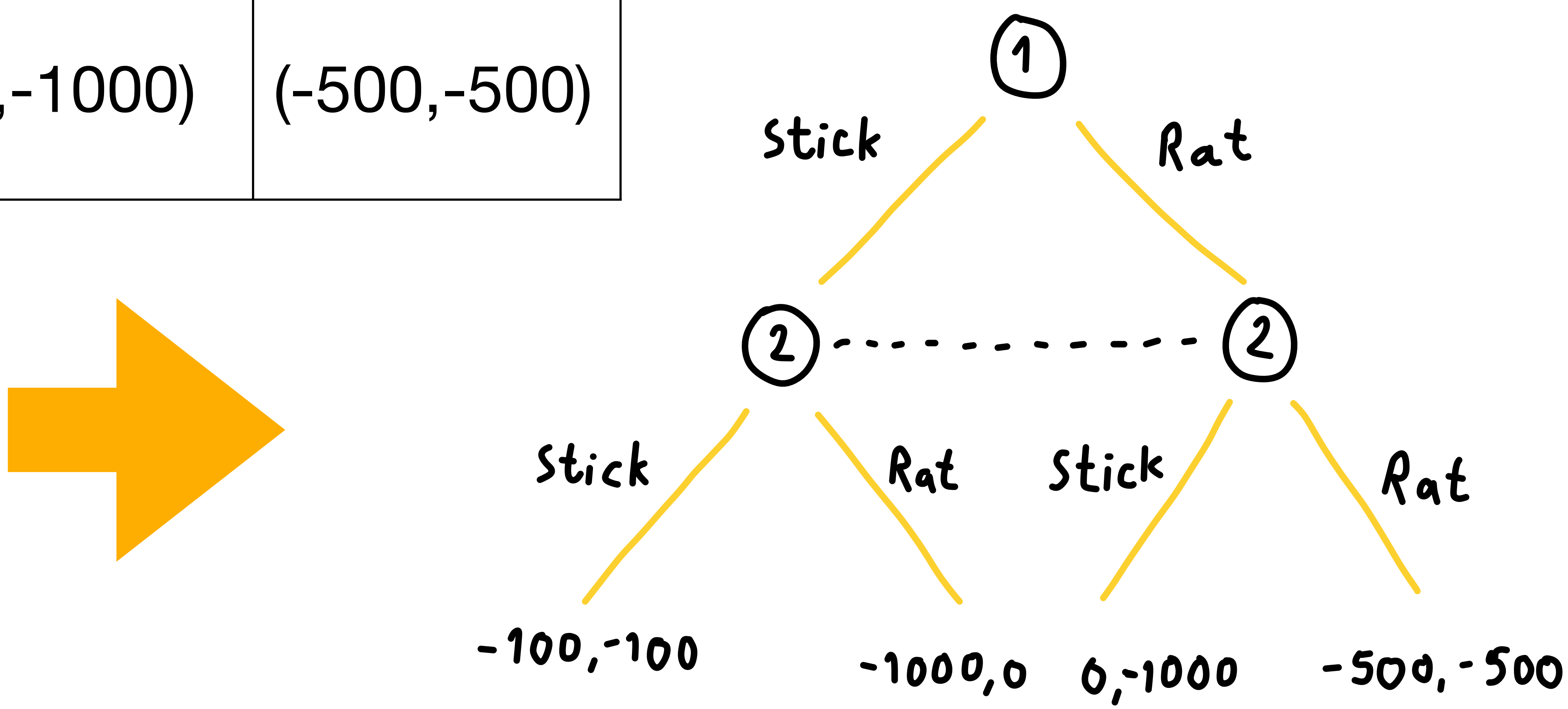
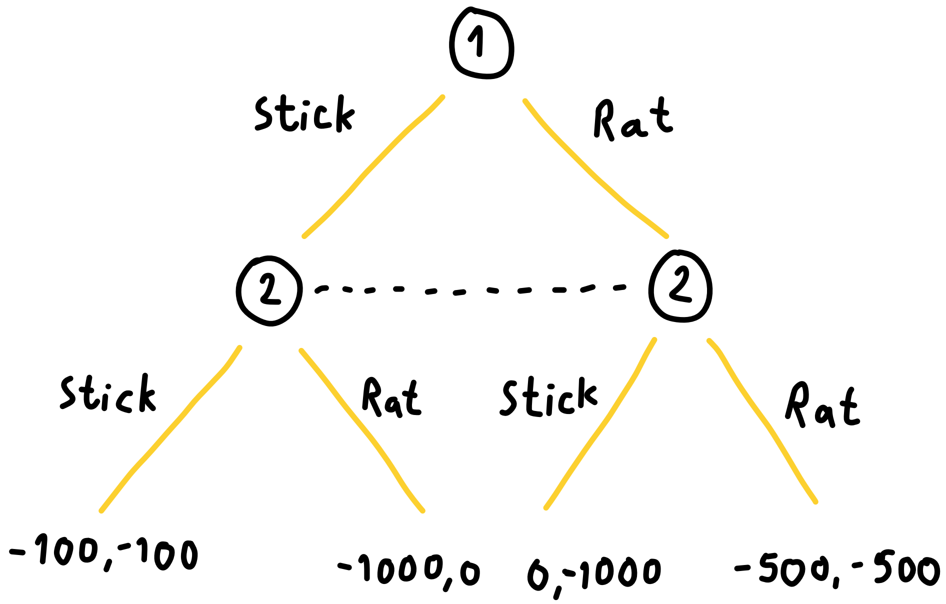
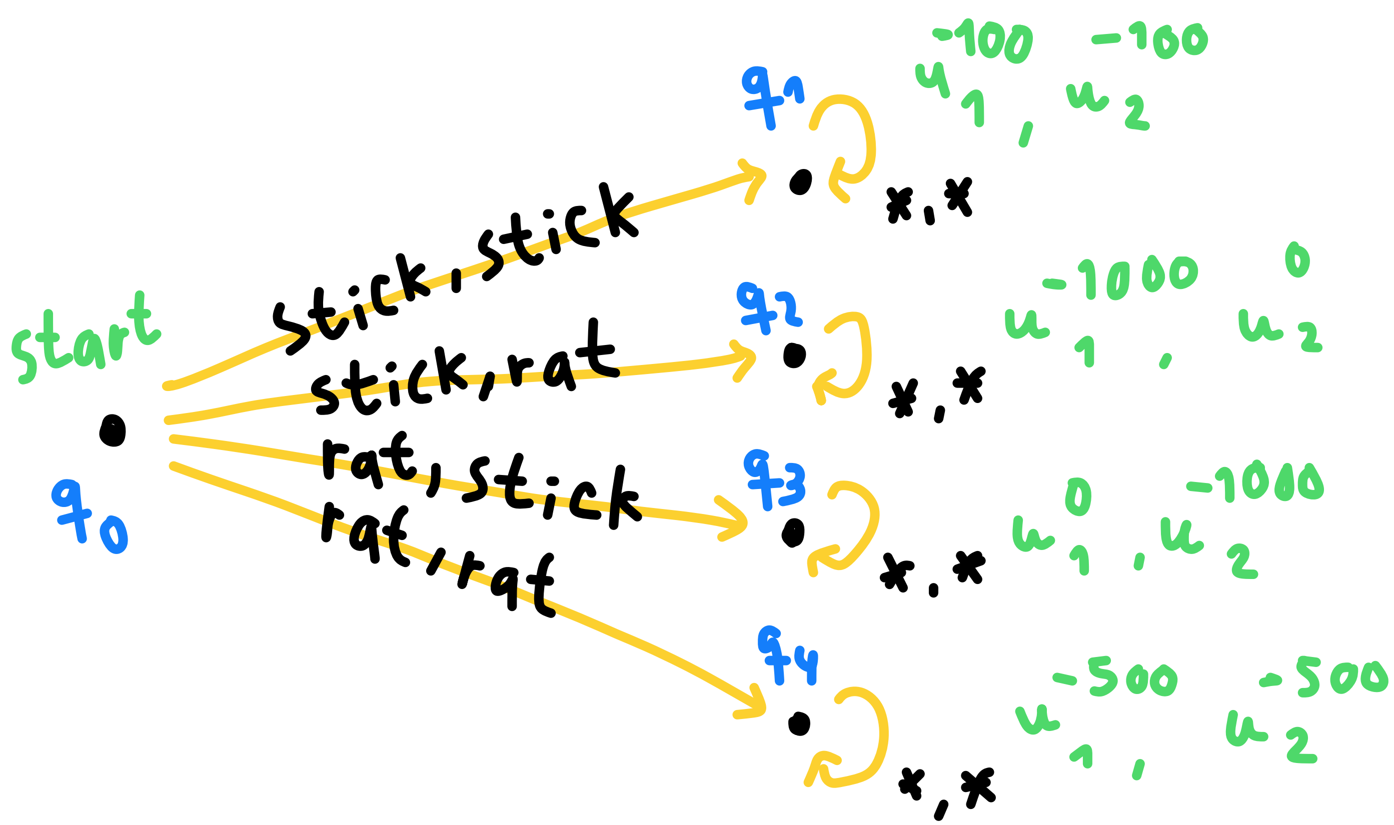



Image: clipartkey.com

# Prisoner's Dilemma as a CGM




$$M = (\mathbb{A}gt, St, Act, d, o, V)$$

$$\mathbb{A}gt = \{1,2\}$$

$$St = \{q_0, \ldots, q_4\}$$

$$Act = \{\textsf{stick, rat}\}$$

$$d(1,q_0) = \{\textsf{stick, rat}\}\ (\ldots)$$

$$o(q_0, \textsf{rat, rat}) = q_4\ \ (\ldots)$$

$$V(u_1^0) = \{q_3\}\ (\ldots)$$

Image: clipartkey.com

# Strategies

- A **strategy** of a player $a$ in a CGM is a conditional plan that specifies what $a$ should do in each possible situation

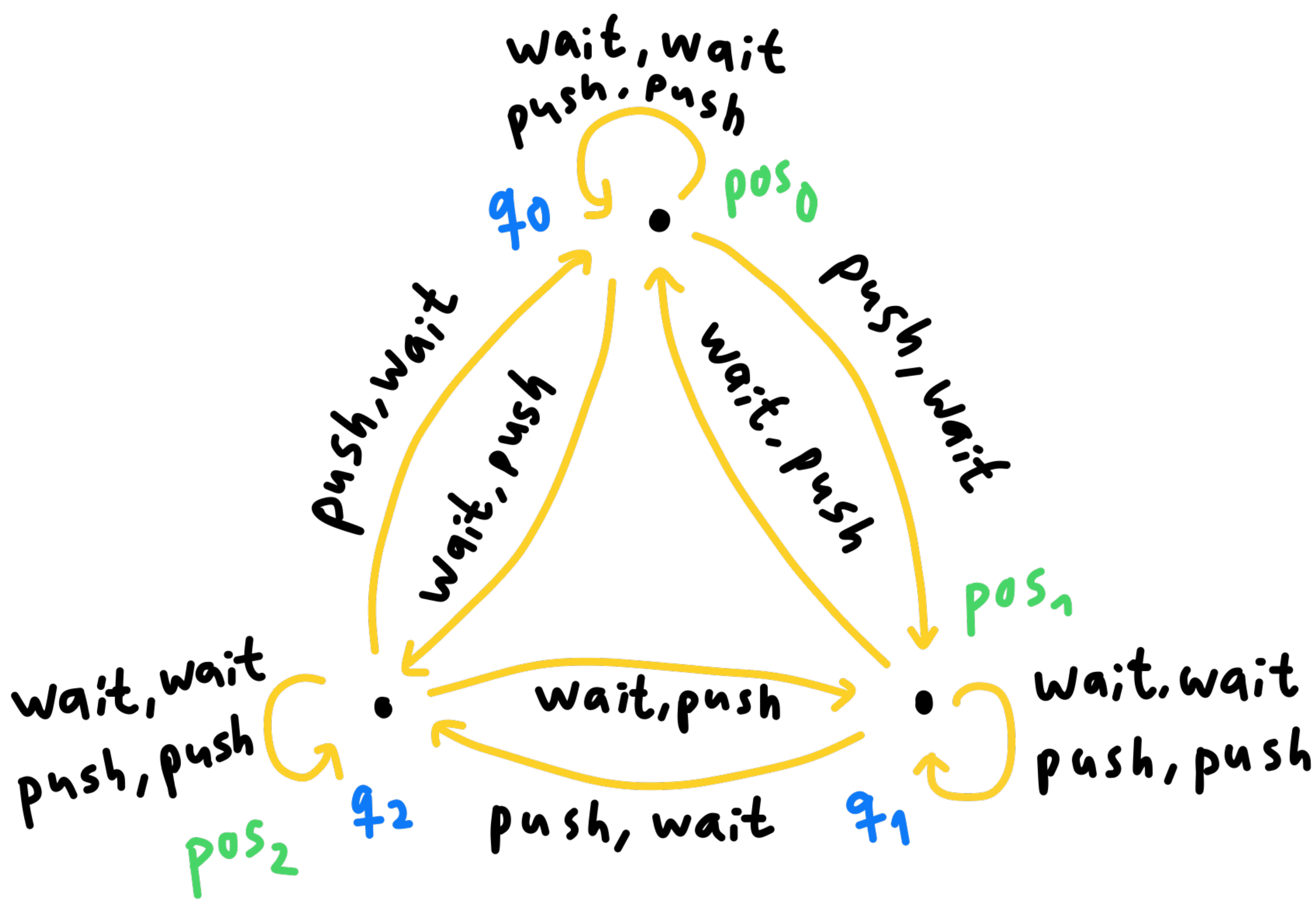
- In a memoryless strategy, the agent does not remember previous actions

**Definition (Strategy)**

**Let $M = (\mathbb{A}gt, St, Act, d, o, V)$ be a CGM. A (memoryless) strategy of a player $a \in \mathbb{A}gt = \{1,\ldots,k\}$ is a function $s_a : St \to Act$ such that $s_a(q) \in d(a,q)$, for $q \in St$.**

**A collective strategy of a coalition $A = \{a_1, \ldots, a_r\}$ is simply a tuple of strategies $S_A = (s_{a_1}, \ldots, s_{a_r})$, one per player in $A$.**

# Back to Robot and Carriage




Some examples of strategies:

$$s_1(q_0) = s_1(q_1) = s_1(q_2) = \text{wait}$$

Robot 1 will execute *wait* no matter what happens

$$s_1'(q_0) = s_1'(q_1) = \text{wait},\ s_1'(q_2) = \text{push}$$

Robot 1 will *wait* unless the carriage is in position 2; in that case, *push*

# Paths

- Remember the definition of *paths* that we used for the semantics of Computation-Tree Logic

**Definition (Path)**

**A path in a CGM $M = (\mathbb{A}gt, St, Act, d, o, V)$ is an infinite sequence of states $q_1, q_2, q_3, \ldots$ in $St$ such that, for each $i \leq 1$, $q_i \to q_{i+1}$. We write the path as $q_1 \to q_2 \to \ldots$.**

# Outcomes of Strategies

- We now formally define the outcomes of a collective strategy in a state
- For a collective strategy $s_A$, we denote agent $a$'s component of the strategy by $s_A[a]$
    - $s_A[a](q)$ is the *action* that $s_A[a]$ prescribes to player $a$ at state $q$

# Outcomes of Strategies

- The function $out(q, s_A)$ returns the set of all paths that may occur when agents in $A$ execute strategy $s_A$ from state $q$ onward

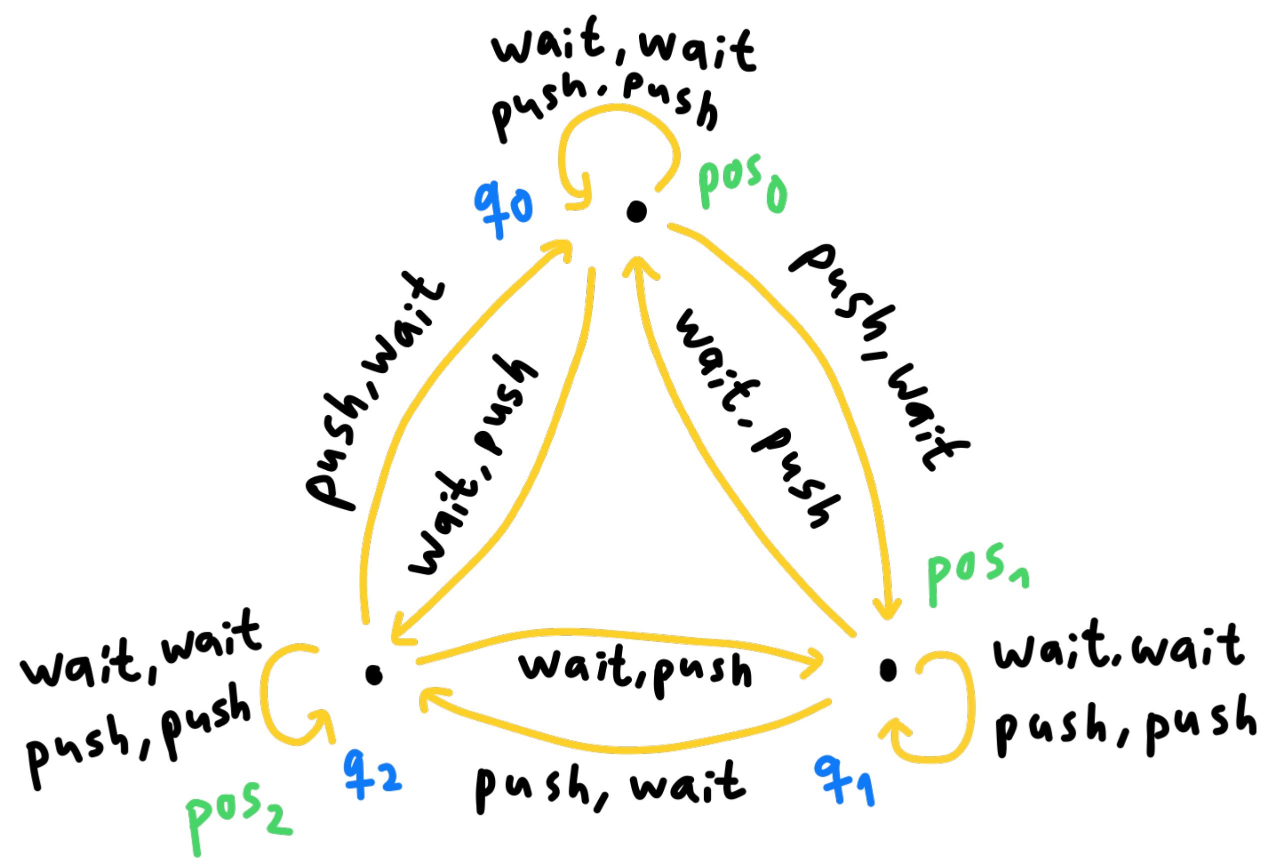
**Definition (Outcomes)**

Let $M = (\mathbb{A}gt, St, Act, d, o, V)$ be a CGM, $S_A = (s_{a_1}, \ldots, s_{a_r})$ be a collective strategy for a coalition $A \subseteq \mathbb{A}gt$, and $q \in St$ be a state. We define the function $out(q, s_A) = \{\pi = q_0 \to q_1 \to q_2 \to \ldots \mid q_0 = q$ and for each $i = 0,1,\ldots$ there exists $(\alpha^i_{a_1}, \ldots, \alpha^i_{a_k})$ such that $\alpha^i_a \in d(a, q_i)$ for every $a \in \mathbb{A}gt$, and $\alpha^i_a = s_A[a](q_i)$ for every $a \in A$ and $q_{i+1} = o(q_i, \alpha^i_{a_1}, \ldots, \alpha^i_{a_k})\}$.

# Back to Robot and Carriage

$s_1(q_0) = s_1(q_1) = s_1(q_2) =$ wait

Robot 1 will execute *wait* no matter what happens




$$q_0 \to q_0 \to q_0 \to \ldots \in out(q_0, s_1)$$

$$q_0 \to q_2 \to q_2 \to \ldots \in out(q_0, s_1)$$

$$q_0 \to q_2 \to q_1 \to q_1 \to \ldots \in out(q_0, s_1)$$

$$q_0 \to q_2 \to q_1 \to q_0 \to \ldots \in out(q_0, s_1)$$

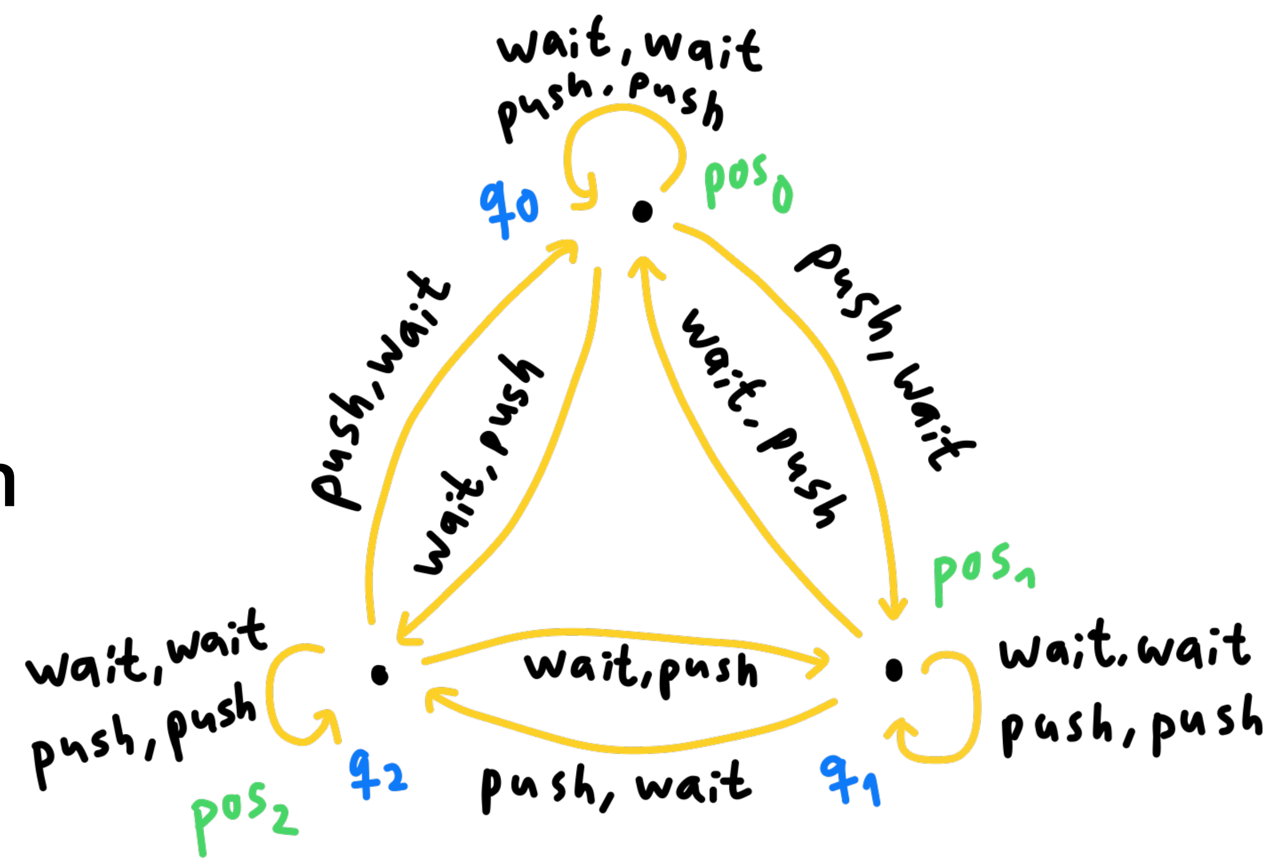
# Back to Robot and Carriage

$$s_1(q_0) = s_1(q_1) = s_1(q_2) = \text{wait}$$

Robot 1 will execute *wait* no matter what happens

$$s_2(q_0) = s_2(q_1) = s_2(q_2) = \text{push}$$

Robot 2 will execute *push* no matter what happens




$$out(q_0, (s_1, s_2)) = \{q_0 \to q_1 \to q_2 \to q_0 \to \dots\}$$

# Language of ATL

- With the language of ATL, we can use logical formulas to talk about concurrent game models

**Definition (Language of ATL)**

Let At be a set of atomic propositions. Formulas $\phi, \psi$ of the **ATL language** are given inductively as follows:

$$\phi, \psi ::= \mathsf{At} \mid \neg\phi \mid (\phi \wedge \psi) \mid \langle\langle A \rangle\rangle X\phi \mid \langle\langle A \rangle\rangle G\phi \mid \langle\langle A \rangle\rangle F\phi \mid \langle\langle A \rangle\rangle \phi U \psi$$

We define ( $\vee$ , $\rightarrow$ , $\leftrightarrow$ , $\top$ , $\bot$ ) as standard.

# Semantics of ATL

- We define the semantics of ATL based on the set $out(q, s_A)$ (all paths that may occur when agents in $A$ execute strategy $s_A$ from state $q$ onward)

**Definition (Semantic Interpretation/Truth, pt.1)**

**Let $M = (\mathbb{A}gt, St, Act, d, o, V)$ be a CGM and $q \in St$ be a state.**

**Truth of a ATL formula $\phi$ at state $q$ in $M$, written $M, q \Vdash \phi$, is defined inductively as follows:**

$M, q \Vdash p$ **iff** $q \in V(p)$

$M, q \Vdash \neg\phi$ **iff** $M, q \nVdash \phi$

$M, q \Vdash \phi \wedge \psi$ **iff** $M, q \Vdash \phi$ **and** $M, q \Vdash \psi$

$M, q \Vdash \langle\langle A \rangle\rangle X\phi$ **iff there is a collective strategy** $s_A$ **such that, for every path** $q \to q_1 \to q_2 \to \ldots \in out(q, s_A)$**, we have** $M, q_1 \Vdash \phi$

# Semantics of ATL

**Definition (Semantic Interpretation/Truth, pt.2)**

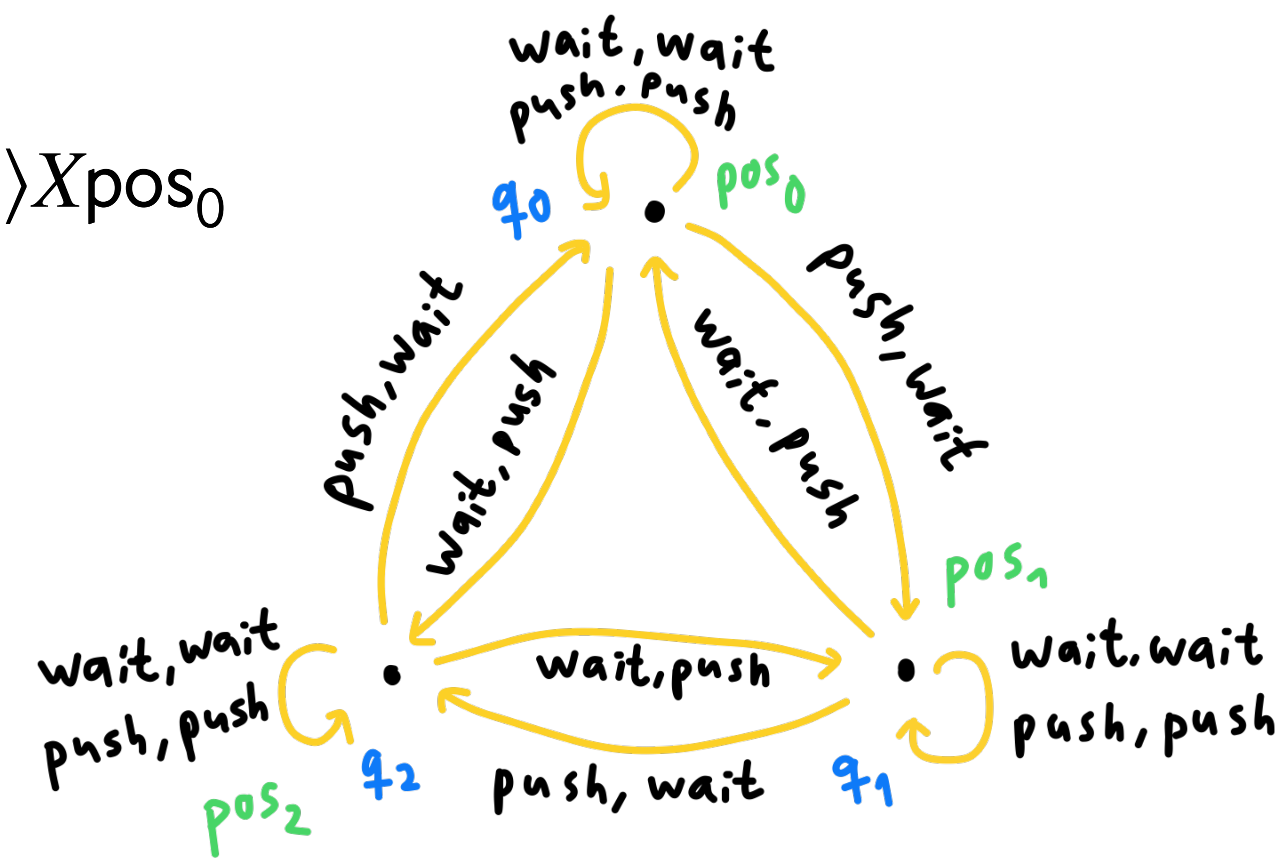
$M, q \Vdash \langle\langle A \rangle\rangle F\phi$ iff there is a collective strategy $s_A$ such that,
for every path $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow \ldots \in out(q, s_A)$, we have
$M, q_i \Vdash \phi$ for some $i \geq 0$

$M, q \Vdash \langle\langle A \rangle\rangle G\phi$ iff there is a collective strategy $s_A$ such that,
for every path $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow \ldots \in out(q, s_A)$, we have
$M, q_i \Vdash \phi$ for all $i \geq 0$

$M, q \Vdash \langle\langle A \rangle\rangle \phi U \psi$ iff there is a collective strategy $s_A$ such that,
for every path $q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow \ldots \in out(q, s_A)$, we have
$M, q_i \Vdash \psi$ for some $i \geq 0$ and $M, q_j \Vdash \phi$ for all $0 \leq j < i$

# Back to Robot and Carriage

$M, q_0 \Vdash \neg\langle\langle\{1\}\rangle\rangle X \mathsf{pos}_0 \wedge \neg\langle\langle\{2\}\rangle\rangle X \mathsf{pos}_0$




Neither robot can solely force the carriage to stay in place

$M, q_0 \Vdash \neg\langle\langle\{1\}\rangle\rangle F \mathsf{pos}_1$

$M, q_0 \Vdash \neg\langle\langle\{1\}\rangle\rangle G \mathsf{pos}_0$

$M, q_0 \Vdash \langle\langle\{1\}\rangle\rangle G \neg \mathsf{pos}_1$

with the strategy $s_1(q_0) = s_1(q_1) =$ wait, $s_1(q_2) =$ push

$M, q_i \Vdash \langle\langle\{1,2\}\rangle\rangle F \mathsf{pos}_0 \wedge \langle\langle\{1,2\}\rangle\rangle F \mathsf{pos}_1 \wedge \langle\langle\{1,2\}\rangle\rangle F \mathsf{pos}_2$

Together, the robots can move the carriage to any position they like

# Relationship Between ATL and CTL

- Recall the language of CTL:

$$\phi, \psi ::= \mathsf{At} \mid \neg\phi \mid (\phi \wedge \psi) \mid AX\phi \mid EX\phi \mid AF\phi \mid EF\phi \mid AG\phi \mid EG\phi \mid A[\phi U \psi] \mid E[\phi U \psi]$$

- We can think of the language of ATL as an extension of the language of CTL, by the following translation:
  - $A\gamma \equiv \langle\langle \emptyset \rangle\rangle \gamma$
  - $E\gamma \equiv \langle\langle \mathbb{A}gt \rangle\rangle \gamma$

# Relationship Between ATL and Coalition Logic

- Recall the language of Coalition Logic:

$$\phi, \psi ::= \mathsf{At} \mid \neg\phi \mid (\phi \wedge \psi) \mid \langle\langle C \rangle\rangle \phi$$

- We can think of the language of ATL as an extension of the language of CL, by the following translation:
    - $\langle\langle C \rangle\rangle \phi \equiv \langle\langle A \rangle\rangle X \phi$

# Some Validities in ATL

- We define validity in ATL precisely as in the case of the basic modal logic

**Definition (Modal Validity)**

A modal formula $\phi$ is **valid**, written as $\Vdash \phi$, if $M, q \Vdash \phi$ for all models and states.

# Some Validities in ATL

$$\langle\langle A\rangle\rangle F\phi \leftrightarrow \neg\langle\langle A\rangle\rangle G\neg\phi$$

note that this equivalent to saying $\langle\langle A\rangle\rangle F\phi \equiv \neg\langle\langle A\rangle\rangle G\neg\phi$

$$\langle\langle A\rangle\rangle F\phi \leftrightarrow \phi \vee \langle\langle A\rangle\rangle X\langle\langle A\rangle\rangle F\phi$$

$$\langle\langle A\rangle\rangle G\phi \leftrightarrow \phi \wedge \langle\langle A\rangle\rangle X\langle\langle A\rangle\rangle G\phi$$

$$\langle\langle A\rangle\rangle \phi_1 U\phi_2 \leftrightarrow \phi_2 \vee (\phi_1 \wedge \langle\langle A\rangle\rangle X\langle\langle A\rangle\rangle \phi_1 U\phi_2)$$

# Proving Validities in ATL?

Exercise:

Prove that

$$\langle\langle\{a\}\rangle\rangle Fp \rightarrow (p \vee \langle\langle\{a\}\rangle\rangle F \langle\langle\{a\}\rangle\rangle Xp)$$

is valid.

# Additional Reading Material

- For additional reading material, I advise:
    - Wojchiech Jamroga: *Logical Methods for Specification and Verification of Multi-Agent Systems*, Institute of Computer Science Polish Academy of Sciences, 2015. https://home.ipipan.waw.pl/w.jamroga/papers/jamroga15specifmas-published.pdf