Alternating-time Temporal Logic

Natasha Alechina Brian Logan

Utrecht University

n.a.alechina@uu.nl b.s.logan@uu.nl

ATL* and ATL

- Hope you all listened to Orna Kupferman's talk
- the topic of today's lecture is **ATL**: **Alternating-time Temporal Logic** (Alur et al. 1997-2002)
- the lecture will cover:
  - syntax and semantics of ATL*
  - syntax and semantics of ATL
  - properties of ATL that will be useful for model checking

Main idea behind ATL/ATL*

- temporal logic meets game theory
- main idea: cooperation modalities

$\langle\langle A \rangle\rangle \Phi$: coalition A has a collective strategy to enforce temporal property Φ

Example Formulae

- $\langle\langle agents \rangle\rangle F$ goal "agents can achieve the goal"
- $\langle\langle agents \rangle\rangle G$ safe "agents can enforce a safety property"
- $\langle\langle agent, env \rangle\rangle G$ (request $\rightarrow F$ granted) fairness: this is an ATL* property

Formal Syntax of ATL*

Syntax of ATL*

$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle\gamma,$
$\gamma ::= \varphi \mid \neg\gamma \mid \gamma \wedge \gamma \mid X\gamma \mid F\gamma \mid G\gamma \mid \gamma_1 \, U \, \gamma_2.$

As in LTL, “eventually” and “always” can be derived from “until”:

- $F\gamma \equiv true \, U \, \gamma$
- $G\gamma \equiv \neg F \neg\gamma$

Two main syntactic variants:

- “Vanilla” ATL: every temporal operator preceded by exactly one cooperation modality
- ATL*: no syntactic restrictions

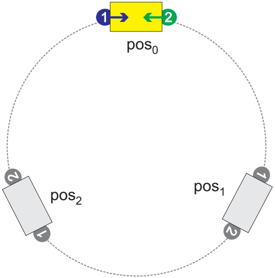
ATL Models: Concurrent Game Structures

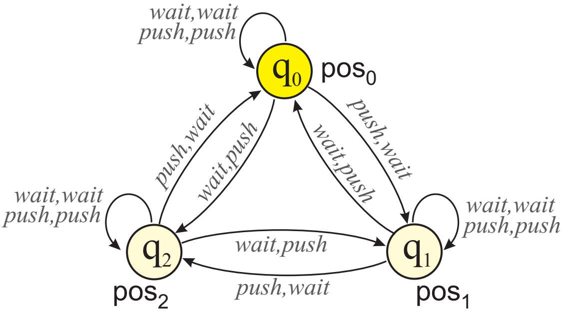
Definition (Concurrent Game Structure)

A **concurrent game structure** is a tuple $M = \langle Agt, St, \mathcal{V}, Act, d, o \rangle$, where:

- Agt: a finite set of all agents
- St: a set of states
- $\mathcal{V}$: a valuation of propositions
- Act: a finite set of (atomic) actions
- $d : Agt \times St \to 2^{Act}$ defines actions available to an agent in a state
- o: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, \ldots, \alpha_k)$ to states and tuples of actions

Example: Robots and Carriage





Strategies

Definition (Strategy)

A **strategy** is a conditional plan.

memoryless strategy: a function $s_a : St \to Act$

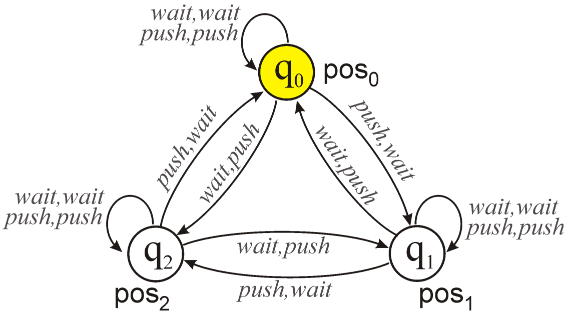
perfect recall strategy: a function $s_a : St^+ \to Act$

A collective strategy is simply a tuple of individual strategies.

Definition (Outcome of a strategy)

Function $out(q, s_A)$ returns the set of all paths that may result from agents A executing strategy s_A from state q onward.

Example: Robots and Carriage



example robot 1
strategy s_1
$s_1(q_0) = wait$
$s_1(q_2) = push$
$s_1(q_1) = wait$
$out(q_0, s_1) = \{$
$q_0 q_0 q_0 \ldots$ (2 waits)
$q_0 q_2 q_2 \ldots$ (2 pushes)
$q_0(q_0^* q_2^*)^*$ (mixed)
all paths avoiding q_1 $\}$

Semantics of ATL*

Definition (Semantics of ATL*: state formulae)

$M, q \models p$ iff q is in $\mathcal{V}(p)$;
$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \wedge \varphi_2$ iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;

Definition (Semantics of ATL*: path formulae)

Semantics of ATL*

Definition (Semantics of ATL*: state formulae)

$M, q \models p$	iff q is in $\mathcal{V}(p)$;
$M, q \models \neg\varphi$	iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \wedge \varphi_2$	iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;
$M, q \models \langle\langle A \rangle\rangle \Phi$	iff there is a collective strategy s_A such that, for every path $\lambda \in out(q, s_A)$, we have $M, \lambda \models \Phi$;

Definition (Semantics of ATL*: path formulae)

Semantics of ATL*

Definition (Semantics of ATL*: state formulae)

$M, q \models p$	iff q is in $\mathcal{V}(p)$;
$M, q \models \neg\varphi$	iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \wedge \varphi_2$	iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;
$M, q \models \langle\langle A \rangle\rangle \Phi$	iff there is a collective strategy s_A such that, for every path $\lambda \in out(q, s_A)$, we have $M, \lambda \models \Phi$;

Definition (Semantics of ATL*: path formulae)

Analogous to LTL and CTL*!

Semantics of ATL*

Definition (Semantics of ATL*: state formulae)

$M, q \models p$	iff q is in $\mathcal{V}(p)$;
$M, q \models \neg\varphi$	iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \wedge \varphi_2$	iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;
$M, q \models \langle\!\langle A \rangle\!\rangle \Phi$	iff there is a collective strategy s_A such that, for every path $\lambda \in out(q, s_A)$, we have $M, \lambda \models \Phi$;

Definition (Semantics of ATL*: path formulae)

$M, \lambda \models \varphi$	iff $M, \lambda[0] \models \varphi$, for a state formula φ;
$M, \lambda \models X\gamma$	iff $M, \lambda[1..\infty] \models \gamma$;
$M, \lambda \models F\gamma$	iff $M, \lambda[i..\infty] \models \gamma$ for some $i \geq 0$;
$M, \lambda \models G\gamma$	iff $M, \lambda[i..\infty] \models \gamma$ for all $i \geq 0$;
$M, \lambda \models \gamma_1\, U\, \gamma_2$	iff $M, \lambda[i..\infty] \models \gamma_2$ for some $i \geq 0$, and $M, \lambda[j..\infty] \models \gamma_1$ forall $0 \leq j < i$;

State-Based Semantics for ATL

The semantics of “vanilla” ATL can be given entirely in terms of states:

$M, q \models p$ iff p is in $\mathcal{V}(q)$;
$M, q \models \neg\varphi$ iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \wedge \varphi_2$ iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;

State-Based Semantics for ATL

The semantics of "vanilla" ATL can be given entirely in terms of states:

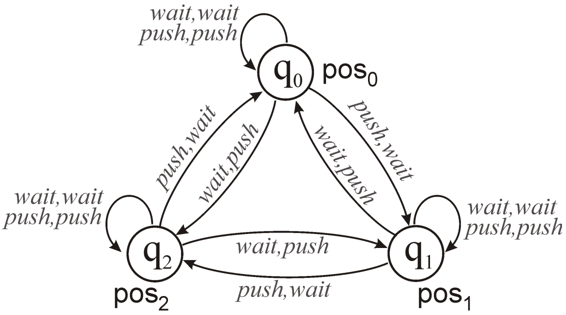
$M, q \models p$	iff p is in $\mathcal{V}(q)$;
$M, q \models \neg\varphi$	iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \wedge \varphi_2$	iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;
$M, q \models \langle\langle A \rangle\rangle X \varphi$	iff there is a collective strategy s_A such that, for every path $\lambda \in out(q, s_A)$, we have $M, \lambda[1] \models \varphi$;

State-Based Semantics for ATL

The semantics of “vanilla” ATL can be given entirely in terms of states:

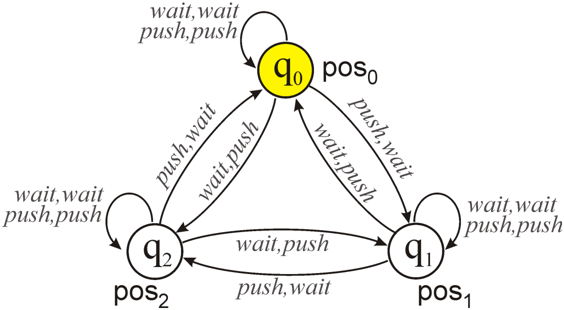
$M, q \models p$	iff p is in $\mathcal{V}(q)$;
$M, q \models \neg\varphi$	iff $M, q \not\models \varphi$;
$M, q \models \varphi_1 \wedge \varphi_2$	iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;
$M, q \models \langle\langle A\rangle\rangle X\varphi$	iff there is a collective strategy s_A such that, for every path $\lambda \in out(q, s_A)$, we have $M, \lambda[1] \models \varphi$;
$M, q \models \langle\langle A\rangle\rangle F\varphi$	iff there is s_A such that, for every $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi$ for some $i \geq 0$;
$M, q \models \langle\langle A\rangle\rangle G\varphi$	iff there is s_A such that, for every $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi$ for all $i \geq 0$;
$M, q \models \langle\langle A\rangle\rangle \varphi_1 \, U \, \varphi_2$	iff there is s_A such that, for every $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi_2$ for some $i \geq 0$ and $M, \lambda[j] \models \varphi_1$ for all $0 \leq j < i$.

Example: Robots and Carriage



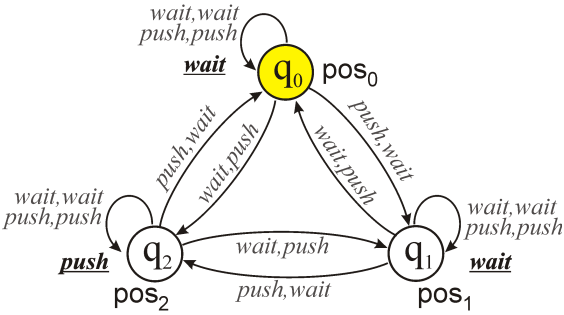
$\mathsf{pos}_0 \rightarrow \langle\langle 1 \rangle\rangle G \neg \mathsf{pos}_1$

Example: Robots and Carriage



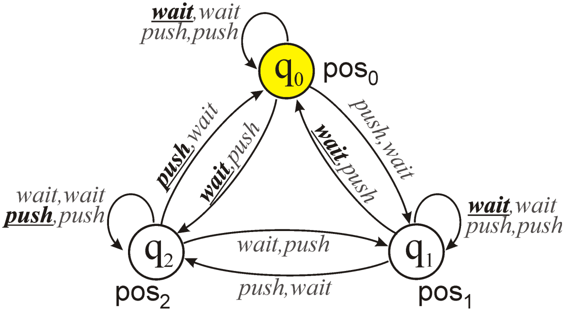
$pos_0 \rightarrow \langle\langle 1 \rangle\rangle G \neg pos_1$

Example: Robots and Carriage



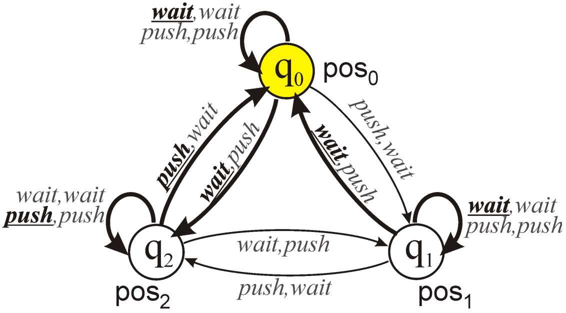
$\mathsf{pos}_0 \rightarrow \langle\!\langle 1 \rangle\!\rangle G \neg \mathsf{pos}_1$

Example: Robots and Carriage



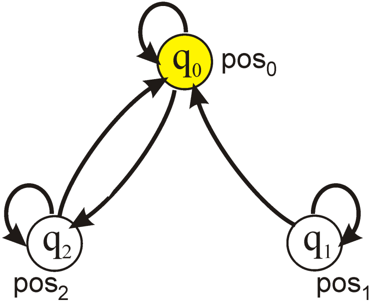
$\mathrm{pos}_0 \rightarrow \langle\langle 1 \rangle\rangle G \neg \mathrm{pos}_1$

Example: Robots and Carriage



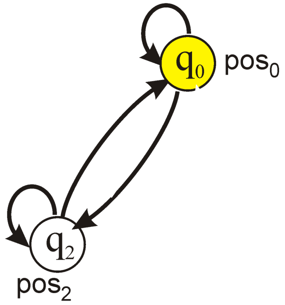
$\mathsf{pos}_0 \rightarrow \langle\!\langle 1 \rangle\!\rangle G\,\neg\mathsf{pos}_1$

Example: Robots and Carriage



$$pos_0 \rightarrow \langle\langle 1 \rangle\rangle G \neg pos_1$$

Example: Robots and Carriage



$$pos_0 \rightarrow \langle\langle 1 \rangle\rangle G \neg pos_1$$

Semantic Embedding of CTL in ATL

Temporal reasoning can be **semantically** embedded in strategic reasoning as follows:

- we take a transition system to be a concurrent game structure with a single agent ("the system" s)
- transitions are due to actions of the agent
- $E\gamma$ ("there is a path on which γ holds") can be then translated to $\langle\langle s \rangle\rangle\gamma$ ("the system can behave in a way that makes γ true")
- $A\gamma$ ("for all paths, γ holds") can be translated to $\langle\langle \emptyset \rangle\rangle\gamma$ ("γ is enforced whatever all the agents – i.e., the system – do")

Syntactic Embedding of CTL in ATL

Also, ATL extends the branching-time logic CTL by the following **syntactic** translation:

- $A\gamma \equiv \langle\langle\emptyset\rangle\rangle\gamma$ ("for all paths" = necessary outcomes)
- $E\gamma \equiv \langle\langle Agt\rangle\rangle\gamma$ ("there is a path" = outcomes obtainable by grand coalition)

Syntactic Embedding of CL in ATL

ATL extends Coalition Logic CL by the following **syntactic** translation:

- $[A]\varphi \equiv \langle\langle A \rangle\rangle X \varphi$

Memory Does not Influence Ability in “Vanilla” ATL

Let us discern between two definitions of the satisfaction relation:

- $\models_R$: perfect recall is assumed, strategies are of type $f : St^+ \to Act$
- $\models_r$: only memoryless strategies are allowed, i.e., $f : St \to Act$

Theorem

For any M, q and φ, we have:

$$M, q \models_r \varphi \quad \Leftrightarrow \quad M, q \models_R \varphi.$$

ATL* and Memory

For ATL* – contrary to “vanilla” ATL – memory matters:

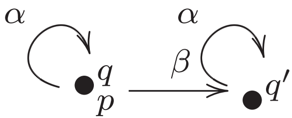
Theorem

There is a model M, a state q in M, and a formula φ, such that

$$M, q \models_r \varphi \quad \not\Leftrightarrow \quad M, q \models_R \varphi$$

Counterexample:

M:



$$\varphi = \langle\langle a \rangle\rangle (Xp \wedge XX\neg p)$$

Fixpoint Properties

Theorem

*The following formulae are **valid** in ATL:*

- $\langle\langle A\rangle\rangle G\,\varphi \quad\leftrightarrow\quad \varphi \wedge \langle\langle A\rangle\rangle X \langle\langle A\rangle\rangle G\,\varphi$
- $\langle\langle A\rangle\rangle \varphi_1\, U\, \varphi_2 \quad\leftrightarrow\quad \varphi_2 \vee \varphi_1 \wedge \langle\langle A\rangle\rangle X \langle\langle A\rangle\rangle \varphi_1\, U\, \varphi_2.$

Corollary

Strategy for *A* that achieves an objective specified in “vanilla” ATL can be **synthesized incrementally** (no backtracking is necessary).

References

R. Alur, T. A. Henzinger, and O. Kupferman.
Alternating-time Temporal Logic.
Journal of the ACM, 49:672–713, 2002.