CS 395T

Game-Based Verification of Contract Signing Protocols

Alternating Transition Systems

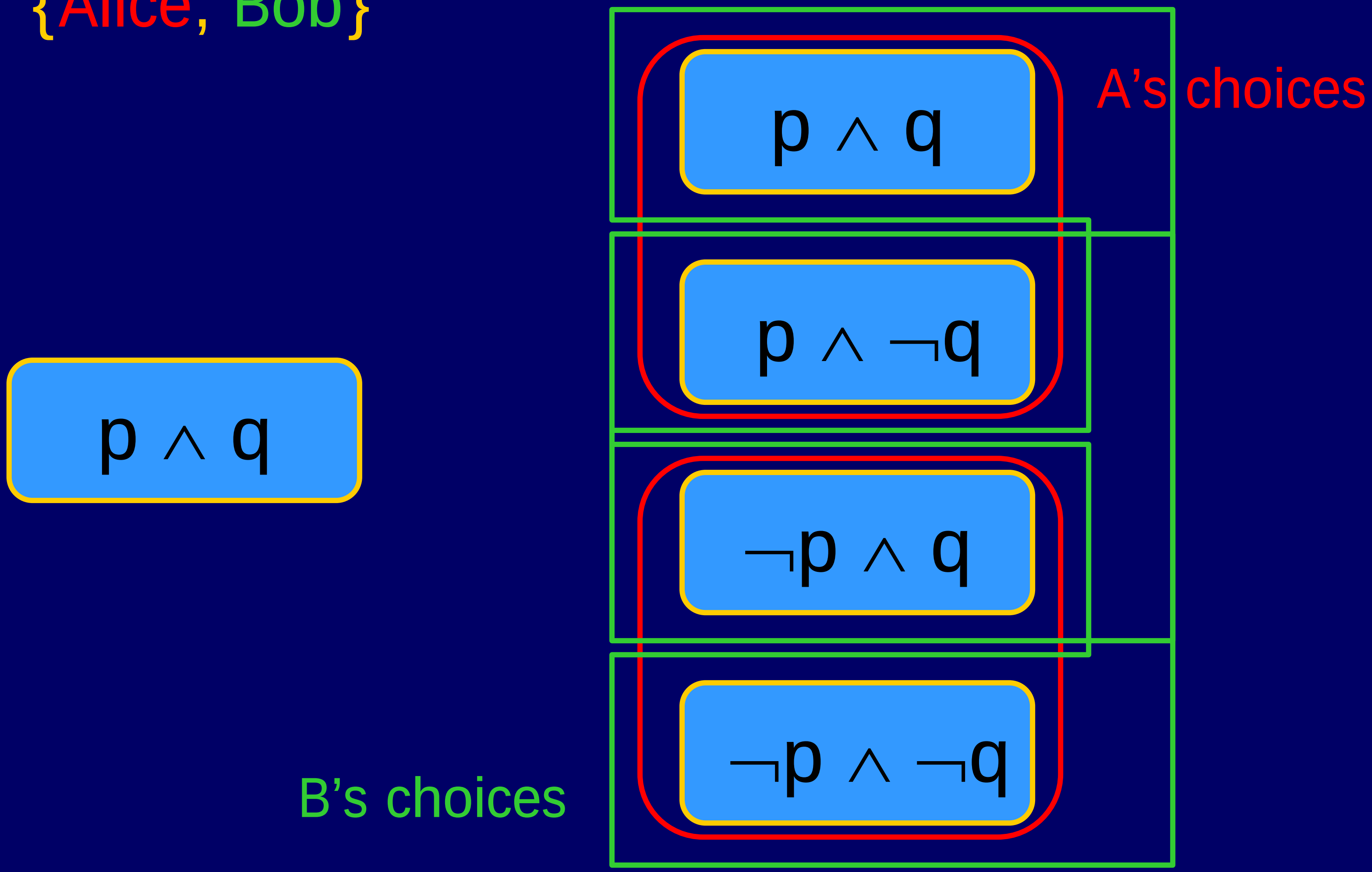
- Game variant of Kripke structures
  - *R. Alur, T. Henzinger, O. Kupferman. "Alternating-time temporal logic". FOCS 1997.*
- Start by defining state space of the protocol
  - Π is a set of propositions
  - Σ is a set of players
  - Q is a set of states
  - $Q_0 \subseteq Q$ is a set of initial states
  - $\pi: Q \rightarrow 2^{\Pi}$ maps each state to the set of propositions that are true in the state
- So far, this is very similar to Murφ

Transition Function

- δ: $Q \times \Sigma \rightarrow 2^{2^Q}$ maps a state and a player to a nonempty set of choices, where each choice is a set of possible next states
  - When the system is in state q, each player chooses a set $Q_a \in \delta(q,a)$
  - The next state is the intersection of choices made by all players $\bigcap_{a \in \Sigma} \delta(q,a)$
  - The transition function must be defined in such a way that the intersection contains a unique state
- Informally, a player chooses a set of possible next states, then his opponents choose one of them

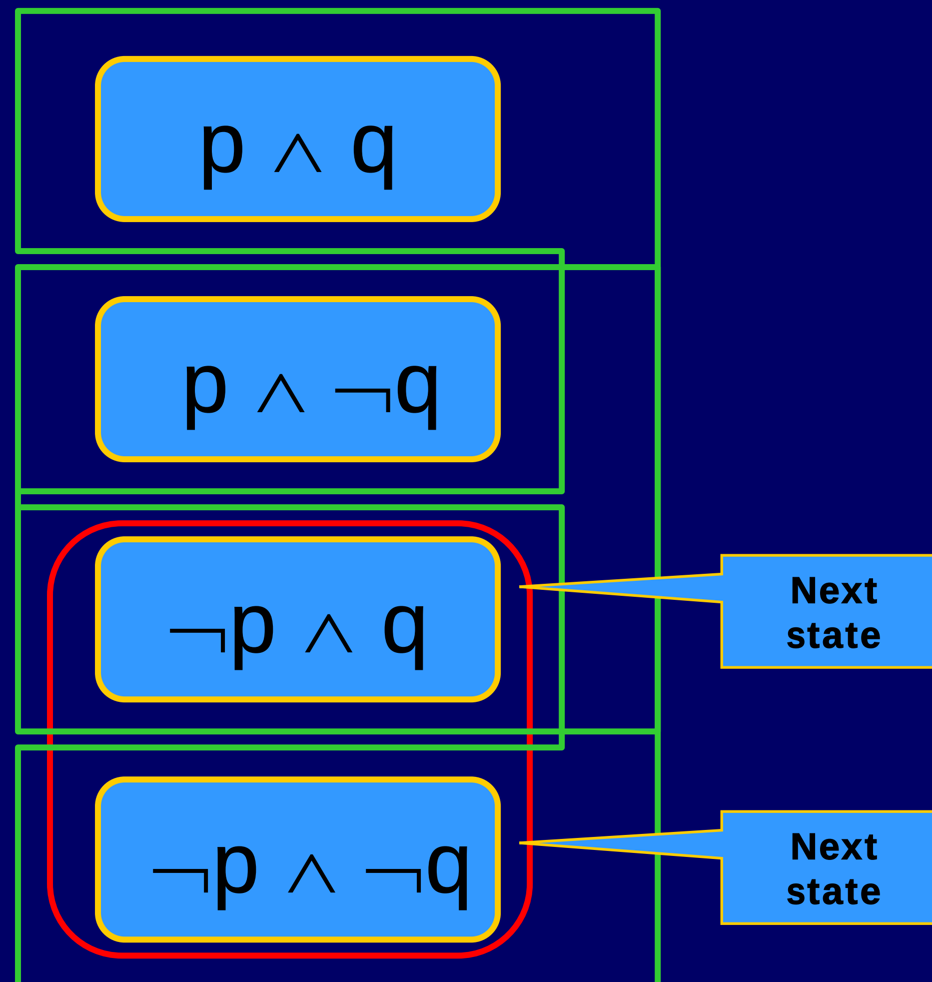
Example: Two-Player ATS

Σ = {Alice, Bob}



Example: Computing Next State

Σ = {Alice, Bob}



If A chooses this set...

... B can choose either state

Alternating-Time Temporal Logic

- Propositions $p \in \Pi$
- $\neg\varphi$ or $\varphi_1 \vee \varphi_2$ where $\varphi, \varphi_1, \varphi_2$ are ATL formulas
- $\langle\langle A\rangle\rangle \bigcirc \varphi$, $\langle\langle A\rangle\rangle \Box \varphi$, $\langle\langle A\rangle\rangle \varphi_1 U \varphi_2$ where $A \subseteq \Sigma$ is a set of players, $\varphi, \varphi_1, \varphi_2$ are ATL formulas
  - These formulas express the ability of coalition A to achieve a certain outcome
  - $\bigcirc$, $\Box$, U are standard temporal operators (similar to what we saw in PCTL)
- Define $\langle\langle A\rangle\rangle \Diamond \varphi$ as $\langle\langle A\rangle\rangle$ true U φ

Strategies in ATL

- A strategy for a player $a \in \Sigma$ is a mapping $f_a : Q^+ \rightarrow 2^Q$ such that for all prefixes $\lambda \in Q^*$ and all states $q \in Q$, $f_a(\lambda \cdot q) \in \delta(q,a)$
  - For each player, strategy maps any sequence of states to a set of possible next states
- Informally, the strategy tells the player in each state what to do next
  - Note that the player cannot choose the next state. He can only choose a <u>set</u> of possible next states, and opponents will choose one of them as the next state.

Temporal ATL Formulas (I)

- $\langle\langle A\rangle\rangle\bigcirc\varphi$ iff there exists a set F_a of strategies, one for each player in A, such that for all future executions $\lambda\in out(q,F_a)$ φ holds in first state $\lambda[1]$
  - Here $out(q,F_a)$ is the set of all future executions assuming the players follow the strategies prescribed by F_a, i.e., $\lambda=q_0q_1q_2\ldots\in out(q,F_a)$ if $q_0=q$ and $\forall i\ q_{i+1}\in \bigcap_{a\in A} f_a(\lambda[0,i])$
- Informally, $\langle\langle A\rangle\rangle\bigcirc\varphi$ holds if coalition A has a strategy such that φ always holds in the next state

Temporal ATL Formulas (II)

- $\langle\langle A\rangle\rangle\Box\varphi$ iff there exists a set F_a of strategies, one for each player in A, such that for all future executions $\lambda\in out(q,F_a)$ φ holds in all states
  - Informally, $\langle\langle A\rangle\rangle\Box\varphi$ holds if coalition A has a strategy such that φ holds in every execution state
- $\langle\langle A\rangle\rangle\Diamond\varphi$ iff there exists a set F_a of strategies, one for each player in A, such that for all future executions $\lambda\in out(q,F_a)$ φ eventually holds in some state
  - Informally, $\langle\langle A\rangle\rangle\Diamond\varphi$ holds if coalition A has a strategy such that φ is true at some point in every execution

Protocol Description Language

- Guarded command language
  - Very similar to PRISM input language (proposed by the same people)
- Each action described as [] guard $\rightarrow$ command
  - guard is a boolean predicate over state variables
  - command is an update predicate, same as in PRISM
  - Simple example:

```
[]SigM1B ∧ ¬SendM2 ∧ ¬StopB -> SendMrB1':=true;
```

Fairness in ATL

$$\neg\langle\langle B,Com\rangle\rangle\Diamond(contract_A\wedge\neg\langle\langle A_h\rangle\rangle\Diamond contract_B)$$

Bob in collaboration with communication channels

does not have a strategy

to reach a state in which

Bob has Alice's signature

but honest Alice does not have a strategy

to reach a state in which Alice has Bob's signature

Timeliness + Fairness in ATL

$$\langle\langle A_h \rangle\rangle \Diamond (\mathit{stop}_A \wedge (\neg \mathit{contract}_B \rightarrow \neg \langle\langle B, \mathit{Com} \rangle\rangle \Diamond \mathit{contract}_A))$$

Honest Alice always has a strategy to reach a state

in which she can stop the protocol and

if she does not have Bob's signature

then Bob does not have a strategy to obtain Alice's signature even if he controls communication channels

Abuse-Freeness in ATL

$$\neg\langle\langle A\rangle\rangle\Diamond(\text{proveToC} \wedge \langle\langle A\rangle\rangle\Diamond\text{contract}_B \wedge$$
$$\langle\langle A\rangle\rangle\Diamond(\text{aborted} \wedge \neg\langle\langle B_h\rangle\rangle\Diamond\text{contract}_A))$$

Alice doesn’t have a strategy to reach state in which
she can prove to Charlie that
she has a strategy to obtain Bob’s signature AND
a strategy to abort the protocol, i.e., reach a state where
Alice has received abort token and Bob doesn’t have
a strategy to obtain Alice’s signature

Modeling TTP and Communication

- Trusted third party is impartial
  - This is modeled by defining a unique TTP strategy
  - TTP has no choice: in every state, the next action is uniquely determined by its sole strategy
- Can model protocol under different assumptions about communication channels
  - Unreliable: infinite delay possible, order not guaranteed
    - Add "idle" action to the channel state machine
  - Resilient: finite delays, order not guaranteed
    - Add "idle" action + special constraints to ensure that every message is eventually delivered (rule out infinite delay)
  - Operational: immediate transmission

MOCHA Model Checker

- Model checker specifically designed for verifying alternating transition systems
  - System behavior specified as guarded commands
    - Essentially the same as PRISM input, except that transitions are nondeterministic (as in in Murφ), not probabilistic
  - Property specified as ATL formula
- Slang scripting language
  - Makes writing protocol specifications easier
- Try online implementation!
  - http://www-cad.eecs.berkeley.edu/~mocha/trial/