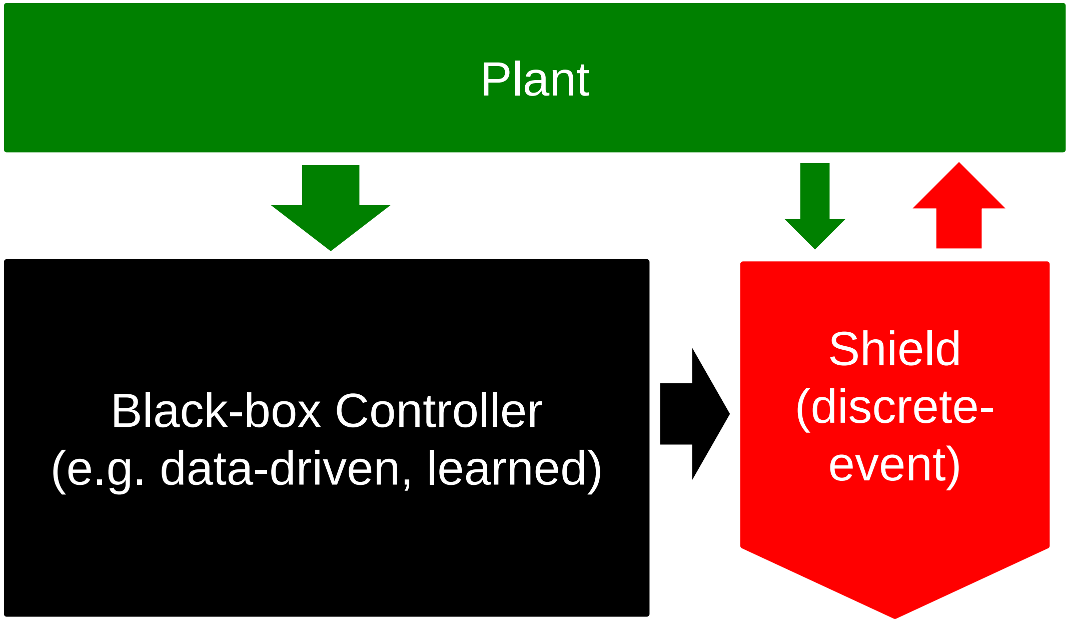
Temporal Logics for Multi-Agent Systems

Tom Henzinger

IST Austria

Joint work with Rajeev Alur, Guy Avni, Krish Chatterjee,
Luca de Alfaro, Orna Kupferman, and Nir Piterman.

Shielded Control



Shield can ensure safety and fairness (temporal-logic specification), performance (quantitative spec), and/or incremental regimes.

Multiple Agents
(e.g. plant, controller, shield; robotics)

A_1:

```
bool  x := 0
loop
    choice
        | x := 0
        | x := x+1 mod 2
    end choice
end loop
```

Φ_1: □ (x ¸ y)

A_2:

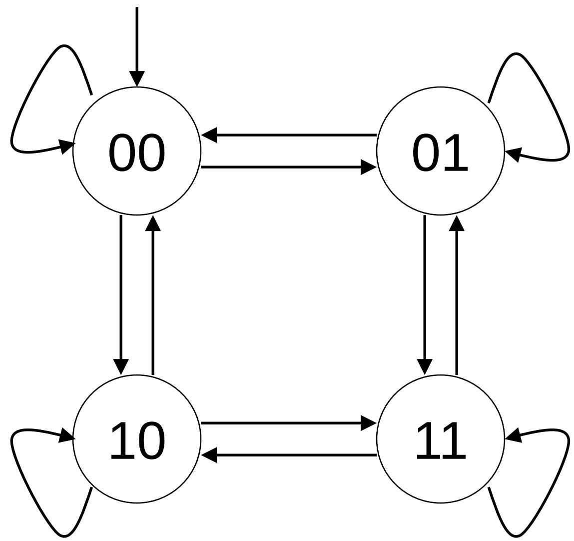
```
bool  y := 0
loop
    choice
        |  y := x
        |  y := x+1 mod 2
    end choice
end loop
```

Φ_2: □ (y = 0)

State Space as Graph

X 8 □ (x ¸ y)

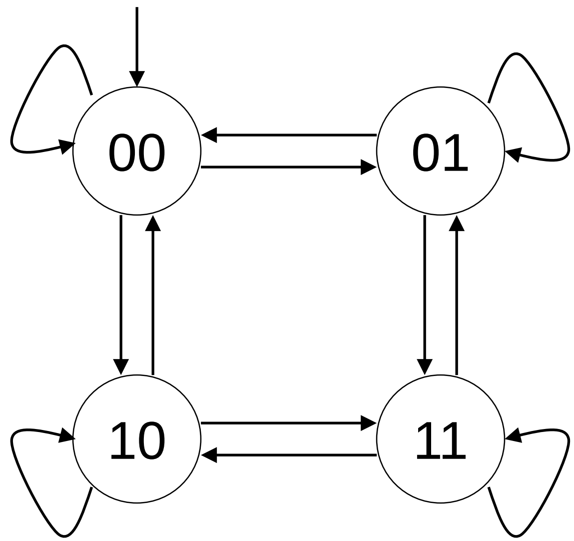
✓ 9 □ (x ¸ y)



State Space as Graph

X 8 □ (x ¸ y)

✓ 9 □ (x ¸ y)



hhA_1ii □ (x ¸ y)

hhA_2ii □ (y = 0)

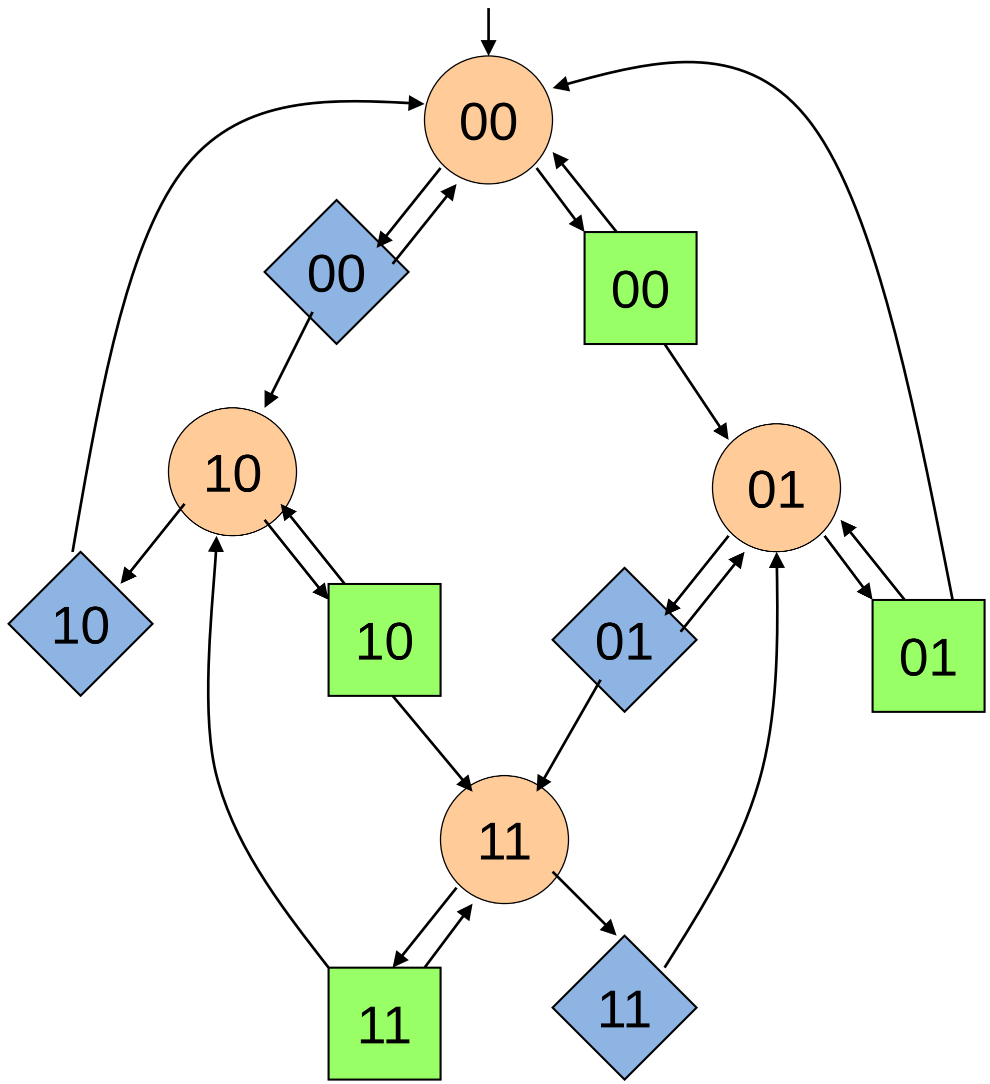
State Space as Game

X 8 □ (x ¸ y)

✓ 9 □ (x ¸ y)

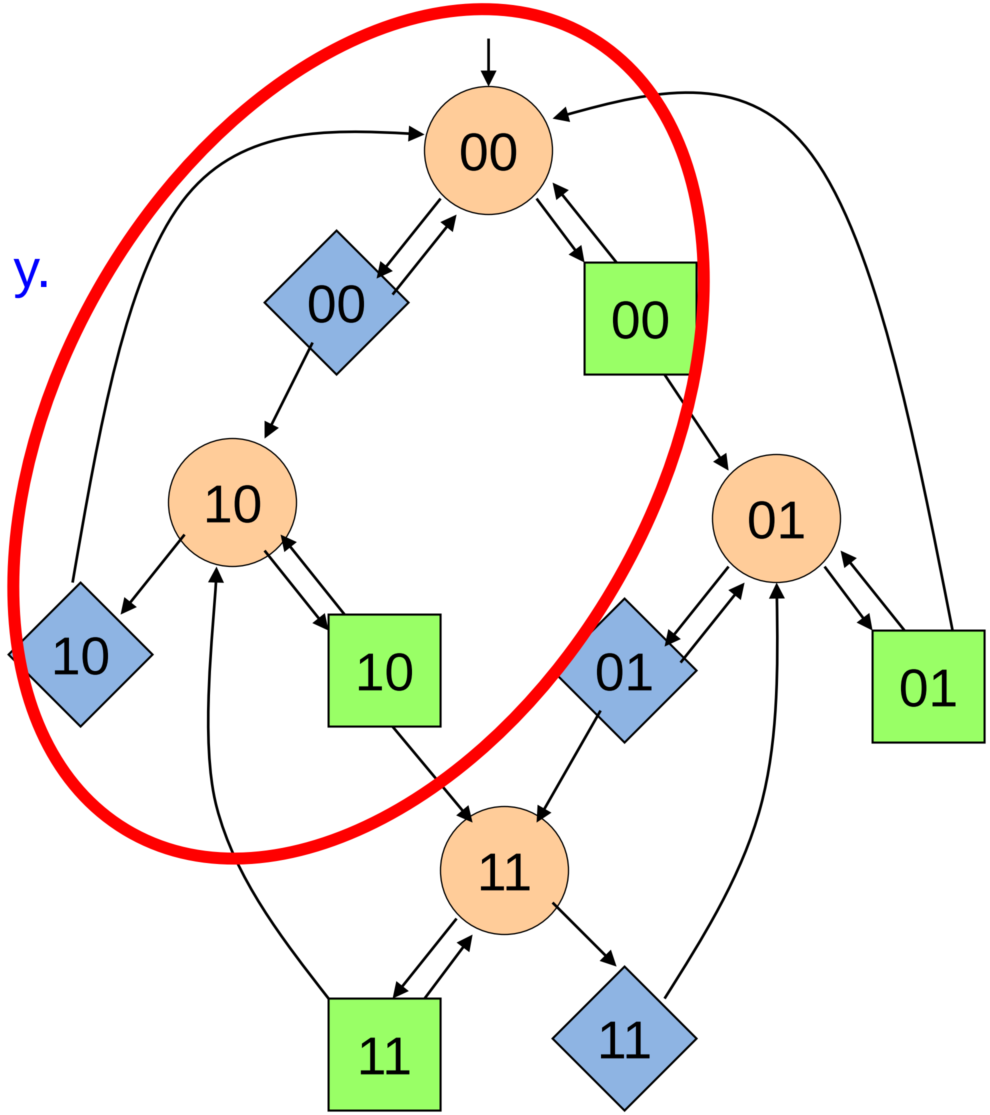
X hhA$_1$ii □ (x ¸ y)

✓ hhA$_2$ii □ (y = 0)



State Space as Game

- If A_2 keeps y = 0,
  then A_1 can keep x ¸ y.



Reactive Synthesis

Agent Synthesis (a.k.a. discrete-event control)

- Given: agent A, specification Φ, and environment E
- Find: refinement A' of A so that A'||E satisfies Φ
- Solution: A' = winning strategy in game A against E for objective Φ

Reactive Synthesis

Agent Synthesis (a.k.a. discrete-event control)

Given: agent A, specification Φ, and environment E

Find: refinement A' of A so that A'||E satisfies Φ

Solution: A' = winning strategy in game A against E for objective Φ

Multi-Agent Synthesis (e.g. shielded or distributed control)

Given:

-two agents A_1 and A_2

-specifications Φ_1 and Φ_2 for A_1 and A_2

Find:

refinements A'_1 and A'_2 of A_1 and A_2 so that

$A'_1||A'_2||S$ satisfies Φ_1 Æ Φ_2 for every fair scheduler S

Mutual Exclusion

```
while( true ) {
  flag[1] := true; turn := 2;

  choice
  | while( flag[1] ) nop;
  | while( flag[2] ) nop;
  | while( turn=1 ) nop;
  | while( turn=2 ) nop;
  | while( flag[1] & turn=2 ) nop;
  | while( flag[1] & turn=1 ) nop;
  | while( flag[2] & turn=1 ) nop;
  | while( flag[2] & turn=2 ) nop;
  end choice;

  CritSec; flag[1] := false;
  nonCritSec;
}
```

```
while( true ) {
  flag[2] := true; turn :=1;

  choice
  | while( flag[1] ) nop;
  | while( flag[2] ) nop;
  | while( turn=1 ) nop;
  | while( turn=2 ) nop;
  | while( flag[1] & turn=2 ) nop;
  | while( flag[1] & turn=1 ) nop;
  | while( flag[2] & turn=1 ) nop;
  | while( flag[2] & turn=2 ) nop;
  end choice;

   CritSec; flag[2] := false;
   nonCritSec;
}
```

Multi-Agent Synthesis Formulation 1

Do there exist refinements A'_1 and A'_2 so that

$[A'_1 \,||\, A'_2 \,||\, S]$ µ (Φ_1 Æ Φ_2)

for every fair scheduler S ?

Solution: game $A_1||A_2$ against S for objective Φ_1 Æ Φ_2

Too weak

(solution has A_1 and A_2 cooperate, e.g. alternate).

Multi-Agent Synthesis Formulation 2

Do there exist refinements A'_1 and A'_2 so that

1. $[A'_1 \mid\mid A_2 \mid\mid S]$ µ Φ_1
2. $[A_1 \mid\mid A'_2 \mid\mid S]$ µ Φ_2

for every fair scheduler S ?

Solution: two games A_1 against $A_2 \mid\mid S$ for objective Φ_1, and A_2 against $A_1 \mid\mid S$ for objective Φ_2

Too strong
(answer is NO, e.g. because agent may stay in CritSec).

Multi-Agent Synthesis Formulation 3

Do there exist refinements A'_1 and A'_2 so that

1. $[A'_1 \,||\, A_2 \,||\, S]$ µ $(\Phi_2\ )\quad \Phi_1)$
2. $[A_1 \,||\, A'_2 \,||\, S]$ µ $(\Phi_1\ )\quad \Phi_2)$
3. $[A'_1 \,||\, A'_2 \,||\, S]$ µ $(\Phi_1$ Æ $\Phi_2)$

for every fair scheduler S ?

Mutual Exclusion

```
while( true ) {
  flag[1] := true; turn := 2;

  while( flag[2] & turn=1 ) nop;

  CritSec; flag[1] := false;
  nonCritSec;
}
```

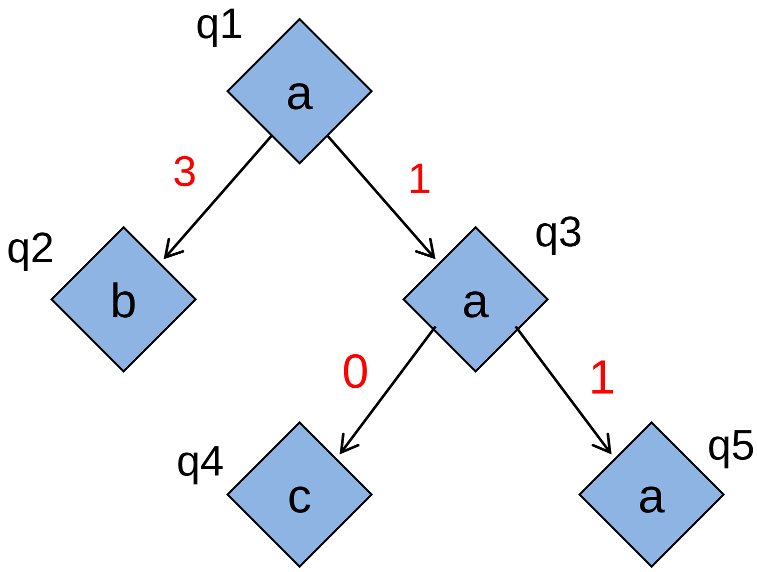
```
while( true ) {
  flag[2] := true; turn := 1;

  while( flag[1] & turn=2 ) nop;

  CritSec; flag[2] := false;
  nonCritSec;
}
```

Solution is exactly Peterson’s mutual-exclusion protocol.

Games on Labeled Graphs

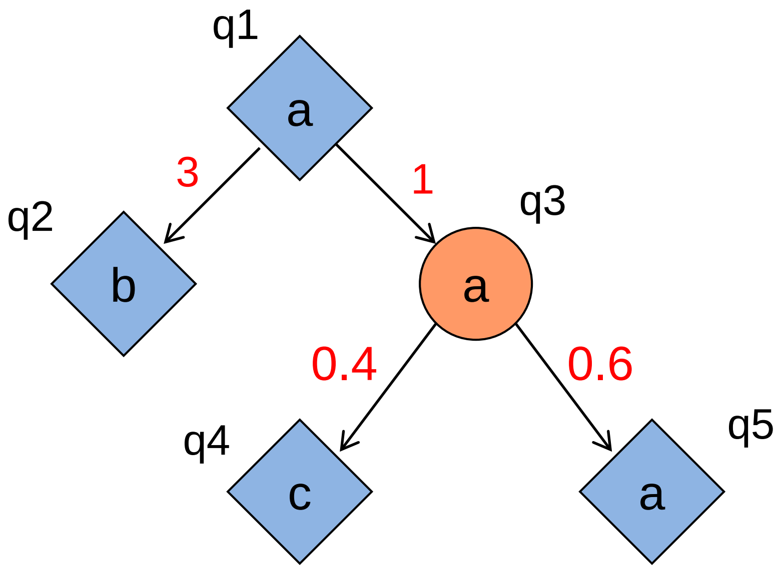
nodes	=	system states
node labels	=	observations
edges	=	state transitions
edge labels	=	transition costs
players	=	agents

Labeled Graph



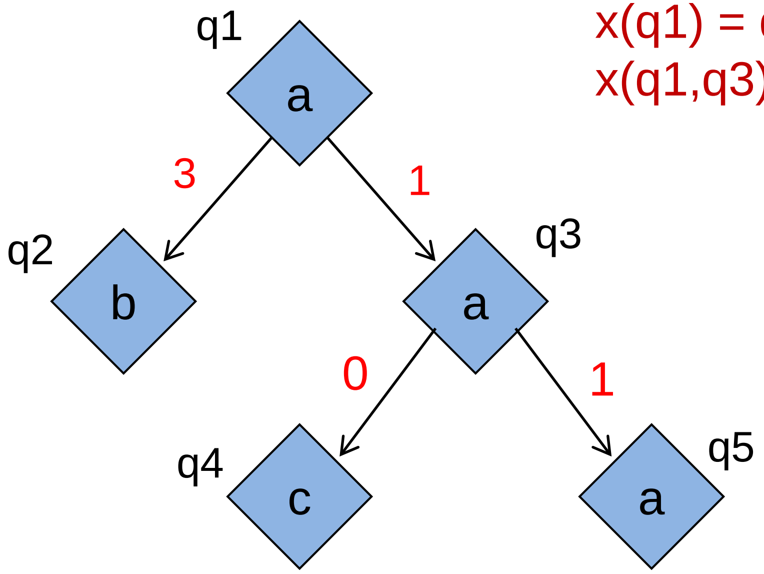
1-agent system without uncertainty.

Markov Decision Process



1-agent system with uncertainty.

Labeled Graph

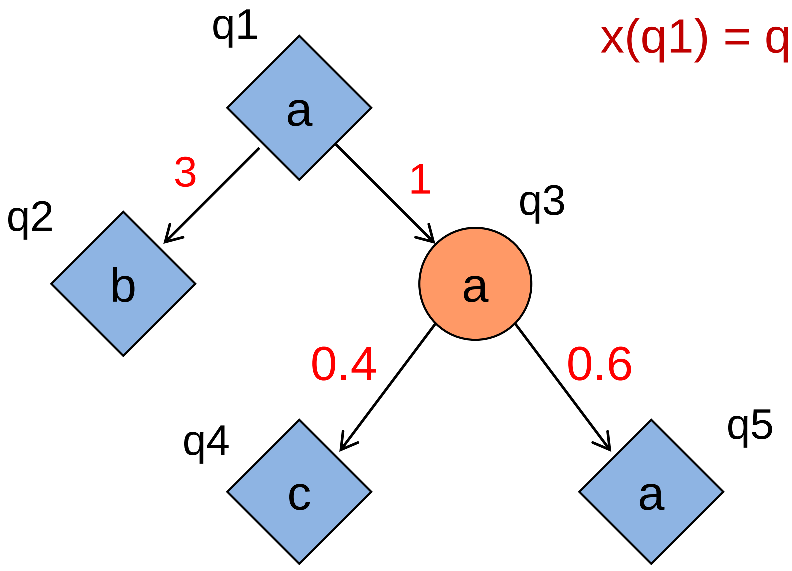


x(q1) = q3
x(q1,q3) = {q4: 0.4; q5: 0.6}

State q 2 Q
Strategy x: Q^* ! D(Q)
x@q: probability space on $Q^!$

} c (x)@q1 = 0.4
avg (x)@q1 = 0.8

Markov Decision Process

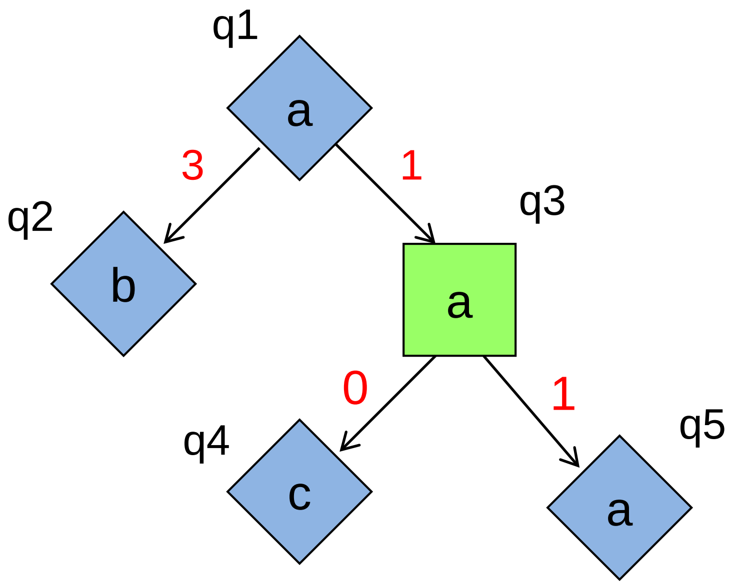


x(q1) = q3

State q 2 Q
Strategy x: Q* ! D(Q)
x@q: probability space on Q!

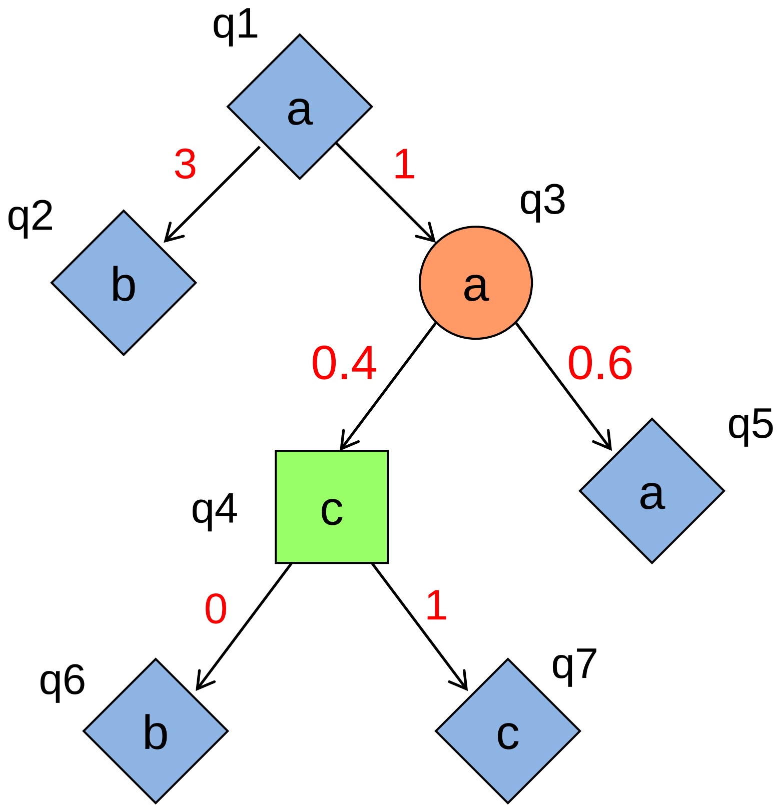
} c (x)@q1 = 0.4
avg (x)@q1 = 1

Turn-based Game



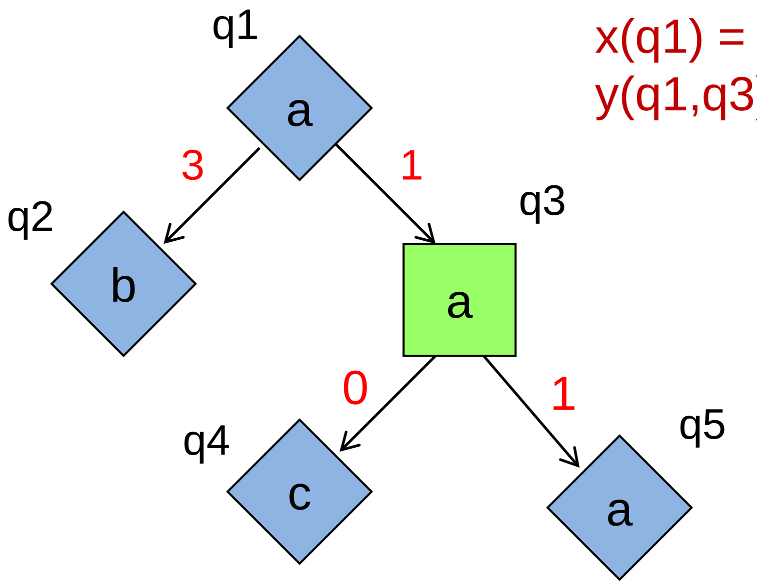
Asynchronous 2-agent system without uncertainty.

Stochastic Game



Asynchronous 2-agent system with uncertainty.

Turn-based Game

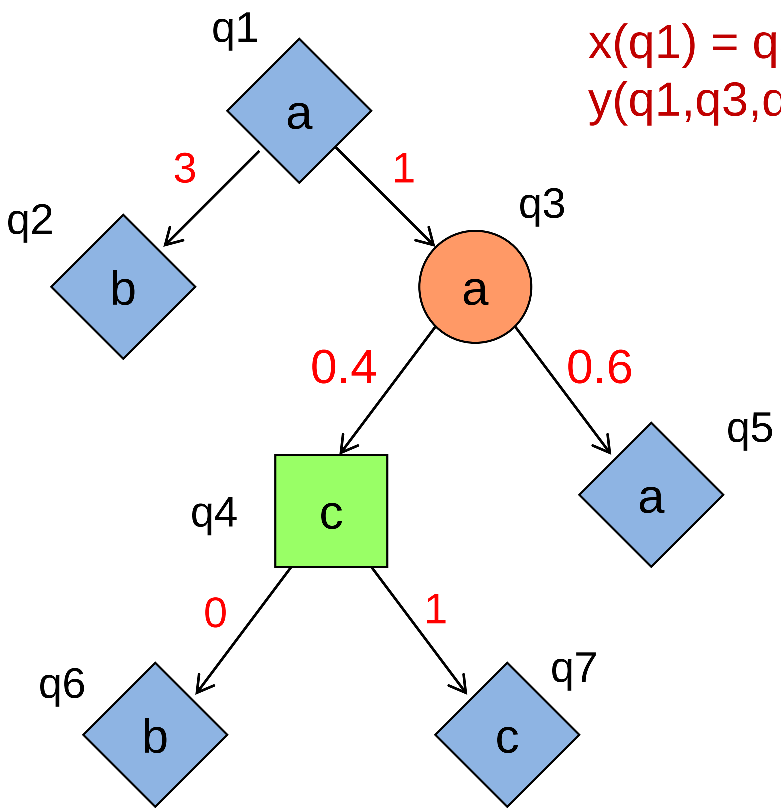


x(q1) = q3
y(q1,q3) = {q4: 0.4; q5: 0.6}

State q 2 Q
Strategies x,y: Q* ! D(Q)
(x,y)@q: probability space on Q!

} c (x,y)@q1 = 0.4
avg (x,y)@q1 = 0.8

Stochastic Game



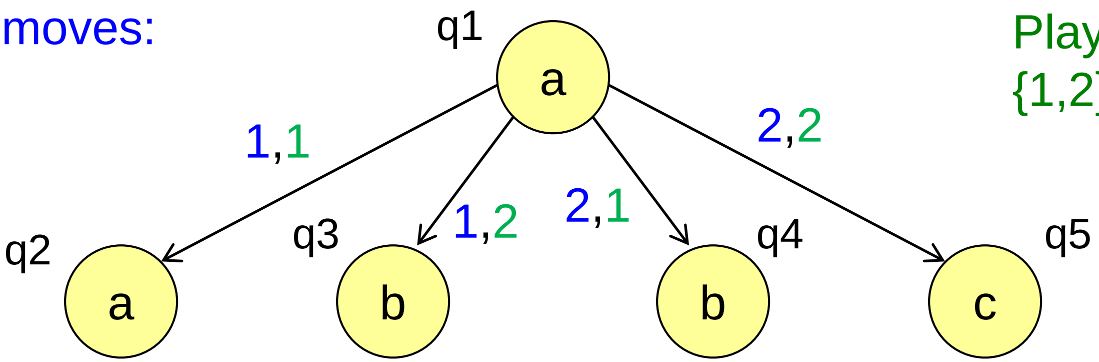
x(q1) = q3
y(q1,q3,q4) = {q6: 0.4; q7: 0.6}

State q 2 Q
Strategies x,y: Q^* ! D(Q)
(x,y)@q: probability space on $Q^!$

} c (x,y)@q1 = 0.4
avg (x,y)@q1 = 0.92

Concurrent Game

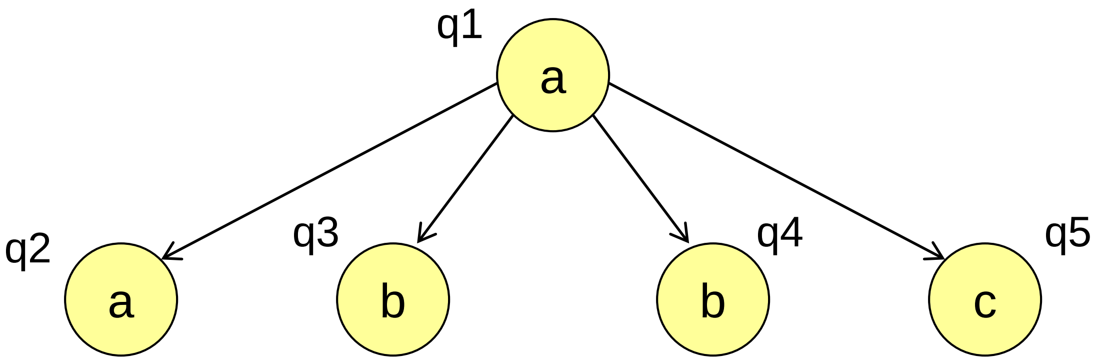
Player Left moves:
{1,2}



Player Right moves:
{1,2}

Synchronous 2-agent system without uncertainty.

Concurrent Stochastic Game



q1:	1	2
1	q2: 0.3 q3: 0.2 q4: 0.5 q5:	q2: 0.1 q3: 0.1 q4: 0.5 q5: 0.3
2	q2: q3: 0.2 q4: 0.1 q5: 0.7	q2: 1.0 q3: q4: q5:

Player Column moves:
{1,2}

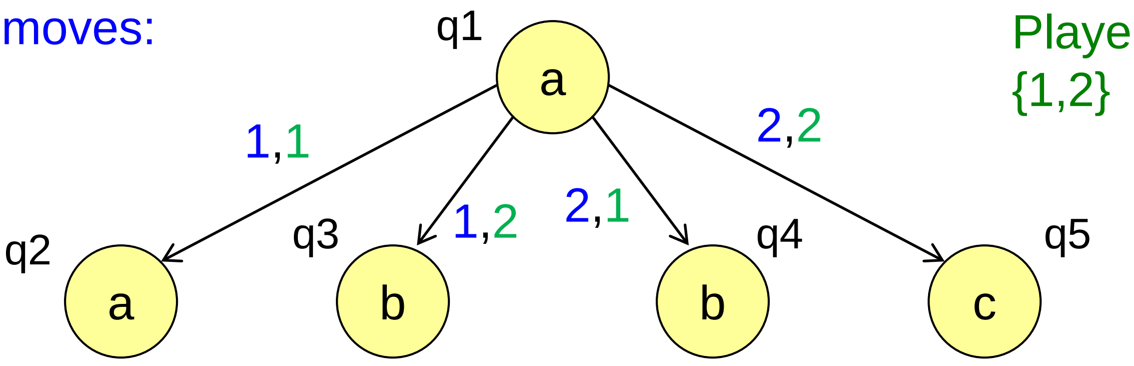
Player Row moves:
{1,2}

Matrix game
at each node.

Synchronous 2-agent system with uncertainty.

Concurrent Game

Player Left moves:
{1,2}



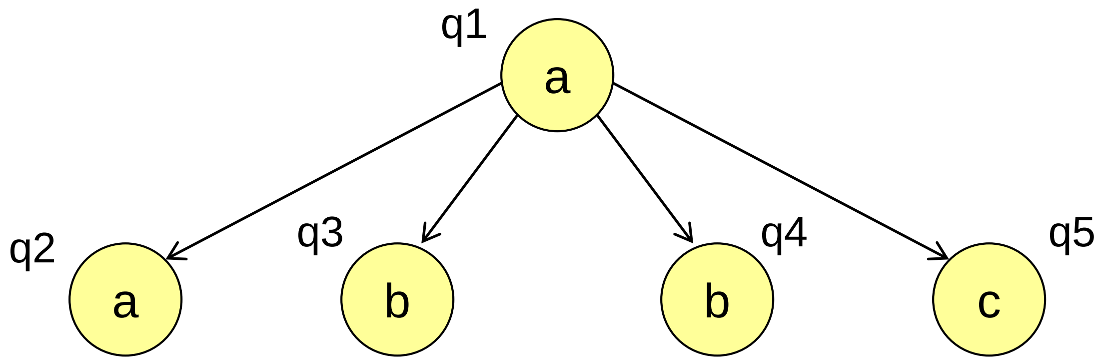
Player Right moves:
{1,2}

State q 2 Q
Strategies x,y: Q* ! D(Moves)
(x,y)@q: probability space on Q!

x(q1) = 2
y(q1) = {1: 0.4; 2: 0.6}

} c (x,y)@q1 = 0.6

Concurrent Stochastic Game



Player Column moves: {1,2}

Player Row moves: {1,2}

q1:

	1	2
1	q2: 0.3 q3: 0.2 q4: 0.5 q5:	q2: 0.1 q3: 0.1 q4: 0.5 q5: 0.3
2	q2: q3: 0.2 q4: 0.1 q5: 0.7	q2: 1.0 q3: q4: q5:

State q 2 Q

Strategies x,y: Q* ! D(Moves)

(x,y)@q: probability space on Q!

x(q1) = 2

y(q1) = {1: 0.4; 2: 0.6}

} c (x,y)@q1 = 0.28

Timed Games, Hybrid Games, etc.

Strategy Logic

1. first-order quantification over sorted strategies
2. linear temporal formulas over observation sequences
3. interpreted over states

q ² (9 x) (8 y) Á iff there exists a player-1 strategy x such that for all player-2 strategies y Á (x,y)@q = 1

Alternating-Time Temporal Logic

1. path quantifiers over sets of players
2. linear temporal formulas over observation sequences
3. interpreted over states

$q \models \langle\langle T \rangle\rangle \varphi$ iff if the game starts from state q
the players in set T can ensure that
the LTL formula φ holds with probability 1

Alternating-Time Temporal Logic

1. path quantifiers over sets of players
2. linear temporal formulas over observation sequences
3. interpreted over states

q ² hhTii Á iff if the game starts from state q
the players in set T can ensure that
the LTL formula Á holds with probability 1

hh;ii Á = 8 Á

hhUii Á = 9 Á where U is the set of all players

[[T]] Á = : hhU\Tii : Á “the players in U\T cannot prevent Á”

ATL* µ SL

$$hhTii\ \acute{A} = (9\ x_1,\ldots,x_m\ 2\ ¦\ _T)\ (8\ y_1,\ldots,y_n\ 2\ ¦\ _{U\backslash T})\ \acute{A}$$

ATL* (SL

Player 1 can ensure $Á_1$ if player 2 ensures $Á_2$:

$$(9\ x)(8\ y)\ (\ ((8\ x')\ Á_2(x',y))\ )\quad Á_1(x,y)\ )$$

ATL* (SL

Player 1 can ensure Á$_1$ if player 2 ensures Á$_2$:

(9 x)(8 y) (((8 x') Á$_2$(x',y)))   Á$_1$(x,y))

The strategy x dominates all strategies w.r.t. objective Á:

(8 x')(8 y) (Á(x',y))   Á(x,y))

ATL* (SL

Player 1 can ensure $Á_1$ if player 2 ensures $Á_2$:

(9 x)(8 y) (((8 x') $Á_2$(x',y))) $Á_1$(x,y))

The strategy x dominates all strategies w.r.t. objective Á:

(8 x')(8 y) (Á(x',y)) Á(x,y))

The strategy profile (x,y) is a secure Nash equilibrium:

(9 x)(9 y) (($Á_1$ Æ$Á_2$) (x,y)
Æ(8 y') ($Á_2$) $Á_1$) (x,y')
Æ(8 x') ($Á_1$) $Á_2$) (x',y))

ATL

ATL is the fragment of ATL* in which every temporal operator is preceded by a path quantifier:

$\langle\langle T \rangle\rangle \bigcirc a$ — single-shot game

$\langle\langle T \rangle\rangle \Diamond b$ — reachability game

$\langle\langle T \rangle\rangle \Box c$ — safety game

ATL

ATL is the fragment of ATL* in which every temporal operator is preceded by a path quantifier:

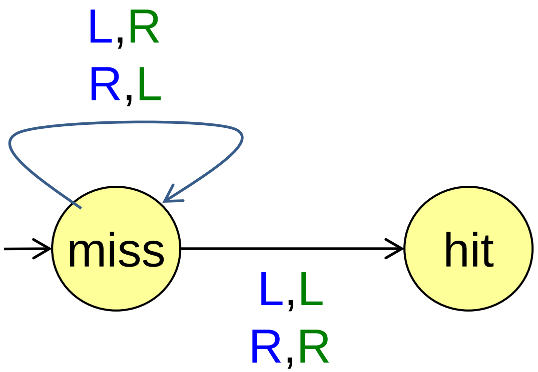
hhTii ° a	single-shot game
hhTii } b	reachability game
hhTii □ c	safety game

Not in ATL:

hhTii □} c	Buchi game
hhTii Á	! -regular (parity) game

Pure Winning

Player 1:
{moveL,moveR}



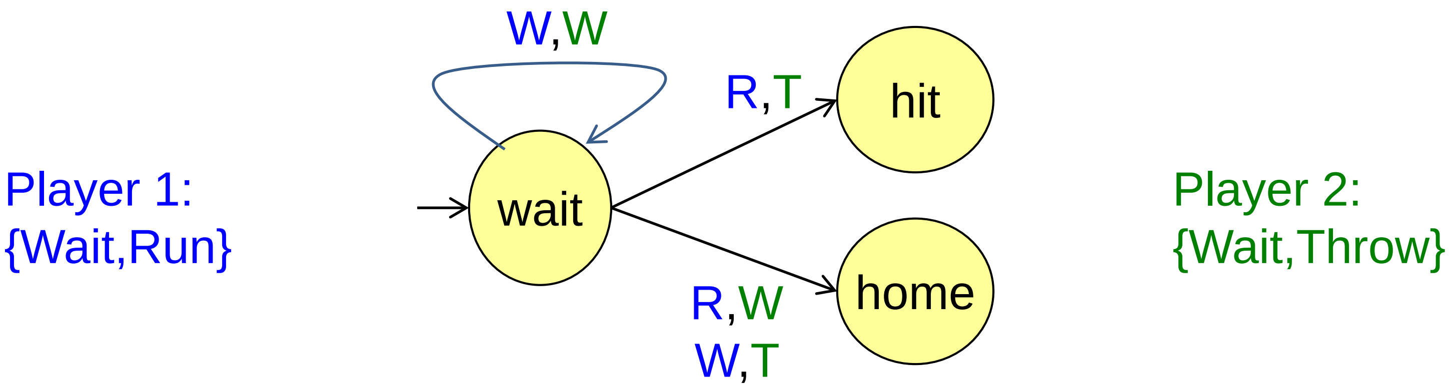
Player 2:
{throwL,throwR}

✗ $\langle\langle P2 \rangle\rangle_{pure} \Diamond\ hit$

✓ $\langle\langle P2 \rangle\rangle \Diamond\ hit$

Player 2 needs randomness to win.

Limit Winning



X hhP1ii } home

✓ hhP1ii $_{\text{limit}}$ } home

Player 1 can win with probability arbitrarily close to 1.

Quantitative ATL

hhP1ii Á = (9 x) (8 y) (Á(x,y) = 1)

hhP1ii $_{limit}$ Á = (sup_x inf_y Á(x,y)) = 1

Quantitative ATL

hhP1ii Á = (9 x) (8 y) (Á(x,y) = 1)

hhP1ii$_{limit}$ Á = (sup$_x$ inf$_y$ Á(x,y)) = 1

hhP1ii$_{val}$ Á = sup$_x$ inf$_y$ Á(x,y)

Complexity of Formula Evaluation (a.k.a. model checking)

CTL: linear in formula, linear/NLOGSPACE in graph
Pure ATL: linear in formula, linear/PTIME in graph
Quantitative ATL: linear in formula, quadratic in graph

CTL^*: PSPACE in formula (convert to word automaton)
ATL^*: 2EXPTIME in formula (convert to tree automaton)

SL: extra exponential for every quantifier elimination

Summary: Classification of Graph Games

1. Number of players: 1 (graph), 1.5 (MDP), 2 , 2.5, k agents
2. Alternation: turn-based or concurrent
3. Formulas: zero-sum (ATL) or equilibria (SL)
4. Strategies: pure or randomized; how much memory needed
5. Values: qualitative (boolean) or quantitative (real)
6. Objectives: Borel 1 (□), 2 (□}), 2.5 (! -regular), 3 (lim avg)

Summary: Classification of Graph Games

1. Number of players: 1 (graph), 1.5 (MDP), 2 , 2.5, k agents
2. Alternation: turn-based or concurrent
3. Formulas: zero-sum (ATL) or equilibria (SL)
4. Strategies: pure or randomized; how much memory needed
5. Values: qualitative (boolean) or quantitative (real)
6. Objectives: Borel 1 (□), 2 (□}), 2.5 (! -regular), 3 (lim avg)
7. Full or partial information (can be undecidable!)

Turn-based Games are Pleasant

-optimal strategies always exist [McIver/Morgan]

-in the non-stochastic case, pure finite-memory optimal strategies exist for ω-regular objectives [Gurevich/Harrington]

-for parity objectives, pure memoryless optimal strategies exist [Emerson/Jutla; Condon], hence NP Å coNP

Concurrent Games are Difficult

-determinacy for randomized but not for pure strategies

-optimal strategies may not exist and ε-close strategies may require infinite memory

-sup inf values may be irrational

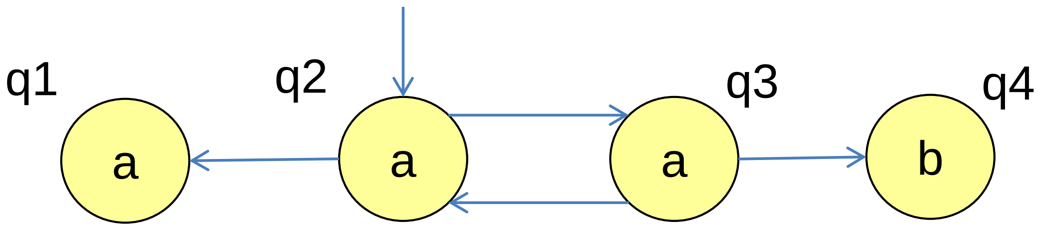
Bidding Game

Each player has a budget.
At each node, each player bids part of their budget.
The winning player chooses the transition.

Richman bidding: the winning bid goes to the losing player.
Poorman bidding: the winning bid goes to the "bank."
Recharging: the budgets are increased by transition weights.

Difficulty: infinitely many possible moves (bids).

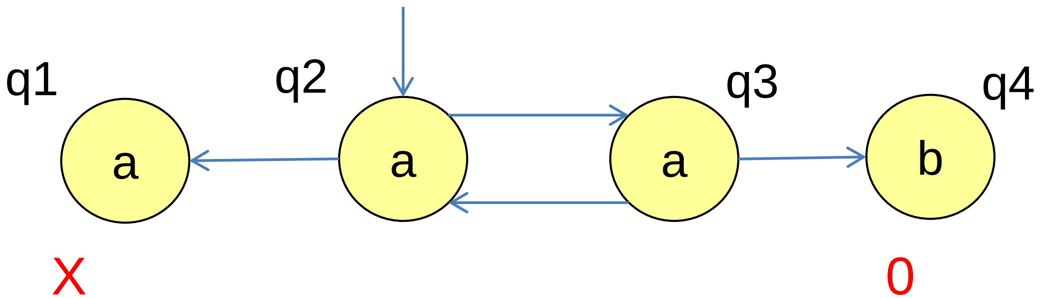
Richman Bidding



The sum of the budgets of players 1 and 2 is 1.

What is the threshold budget for player 1 to win } b ?

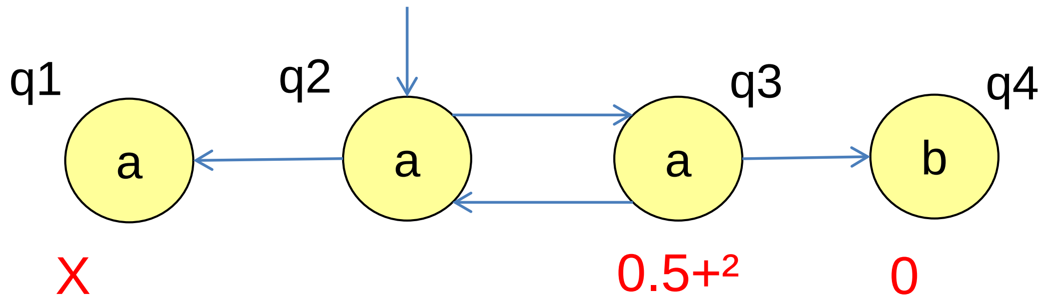
Richman Bidding



The sum of the budgets of players 1 and 2 is 1.

What is the threshold budget for player 1 to win } b ?

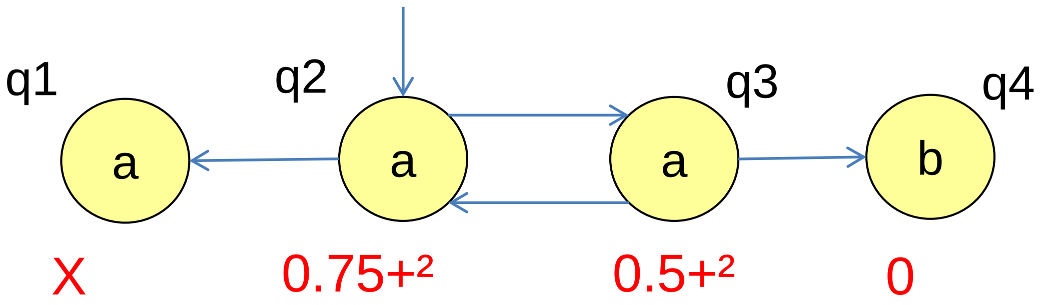
Richman Bidding



The sum of the budgets of players 1 and 2 is 1.

What is the threshold budget for player 1 to win } b ?

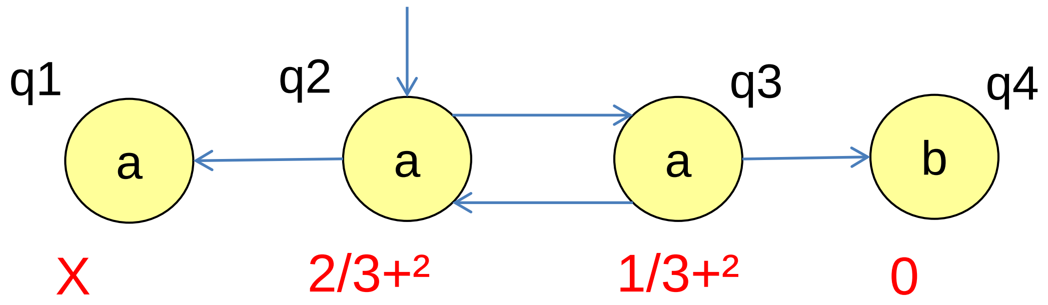
Richman Bidding



The sum of the budgets of players 1 and 2 is 1.

What is the threshold budget for player 1 to win } b ?

Richman Bidding



The sum of the budgets of players 1 and 2 is 1.

What is the threshold budget for player 1 to win } b ?

Some References

Alternating-time temporal logic: JACM 2002

Multi-agent (assume-guarantee) synthesis: TACAS 2007

Concurrent reachability games: TCS 2007

Strategy logic: Information & Computation 2010

Infinite-duration bidding games: CONCUR 2017