Bidding Games on Graphs: In Theory and in Practice

Suman Sadhukhan[1]

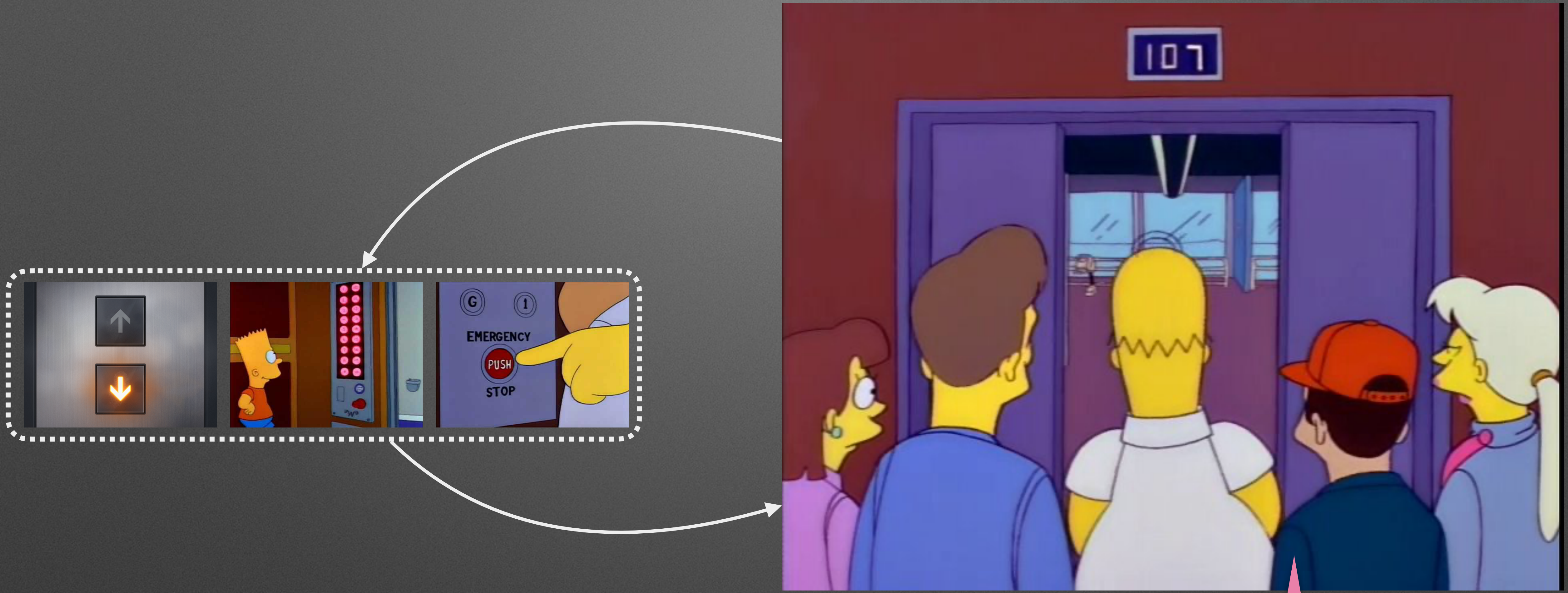
Talk at IARCS Verification Seminar

March 19, 2024

Work in collaboration with Guy Avni[1] and Kaushik Mallik[2]

[1]University of Haifa, [2]IST Austria

Games in Formal Verification

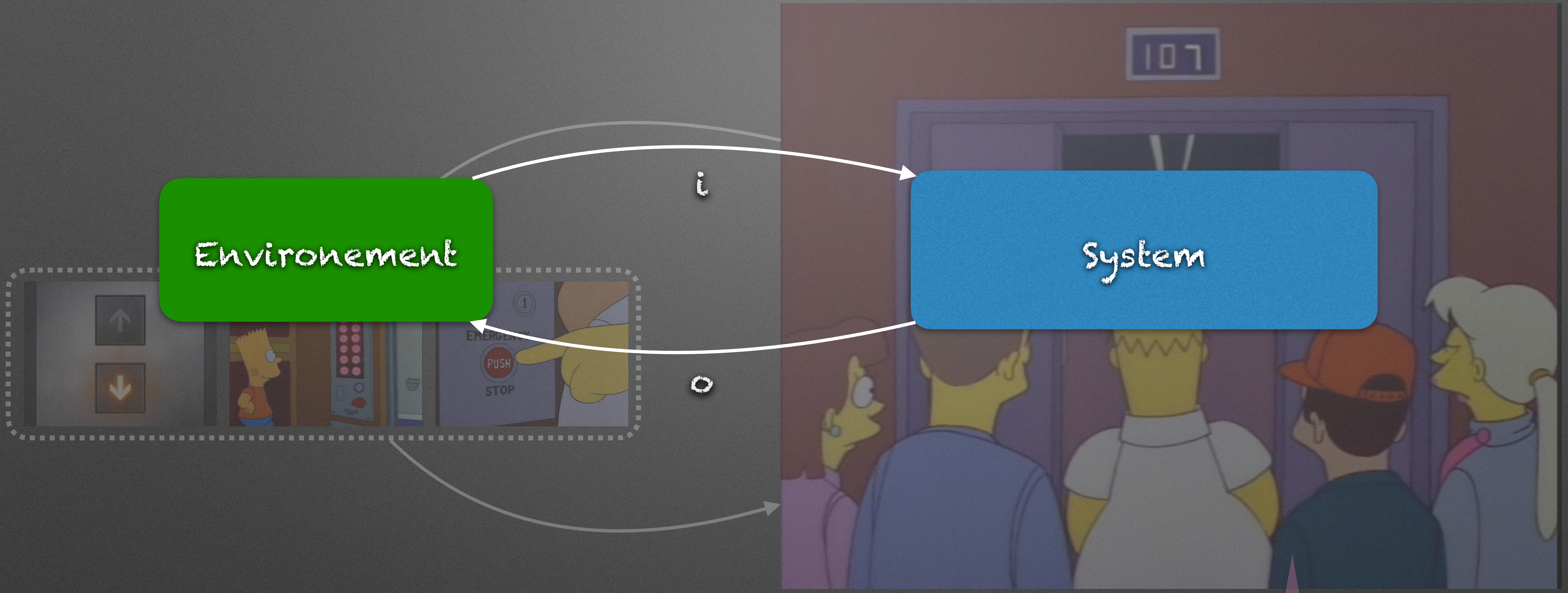


- Door opens iff the lift is at the correct level
- Stops when someone calls

EMERGENCY!!

....

Games in Formal Verification

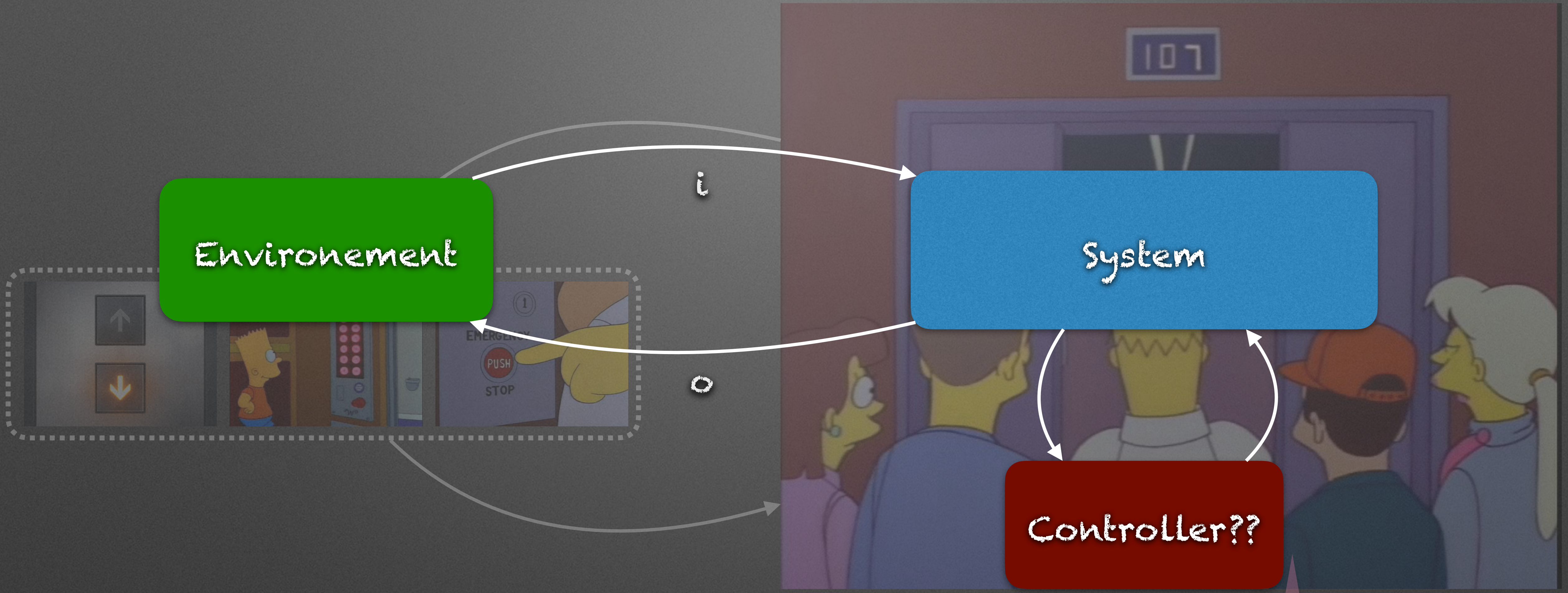


- Door opens iff the lift is at the correct level
- Stops when someone calls

EMERGENCY!!

....

Games in Formal Verification



Environement

System

Controller??

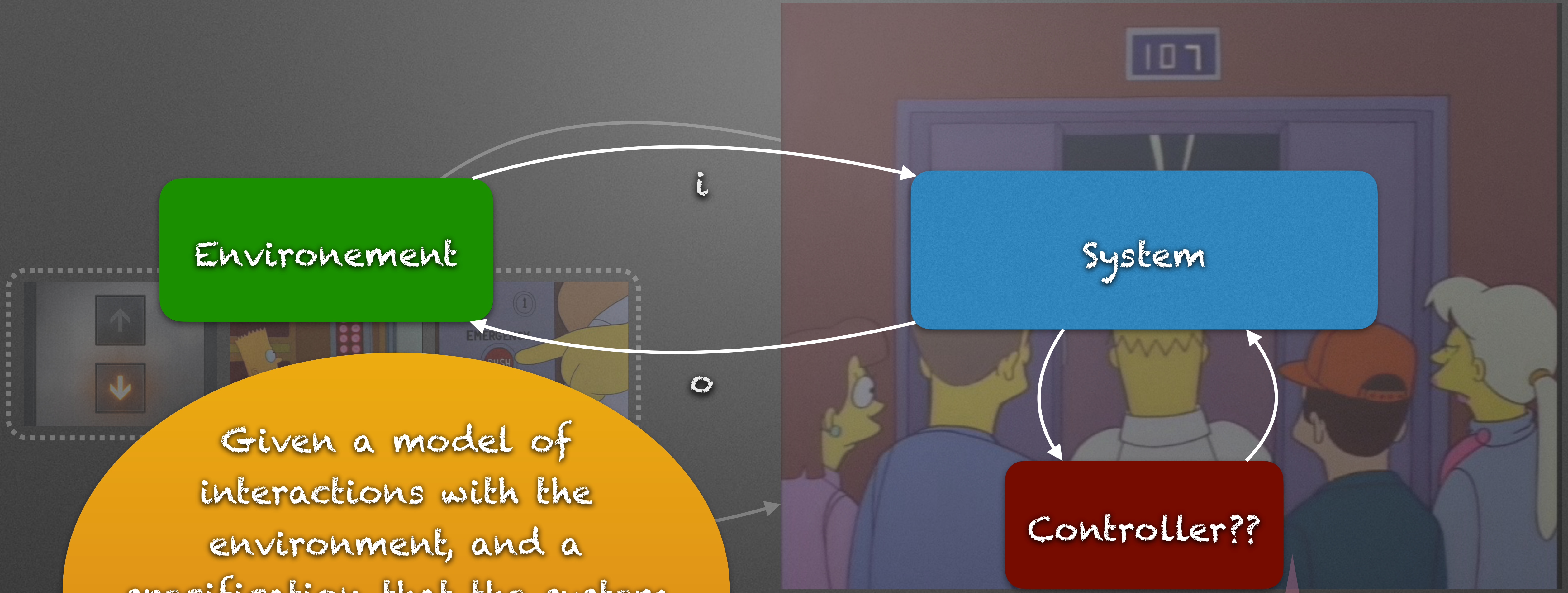
Specifications written in LTL etc....

- Doo at the correct level
- Stops when someone calls

EMERGENCY!!

....

Games in Formal Verification



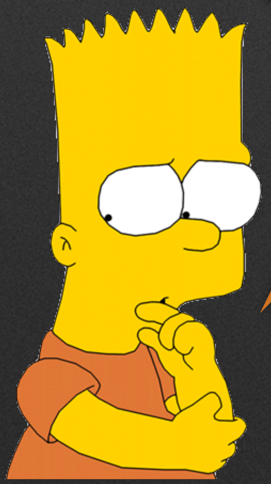
Given a model of interactions with the environment, and a specification that the system needs to satisfy, does there exist a controller who can guarantee that?

Specifications written in LTL etc....

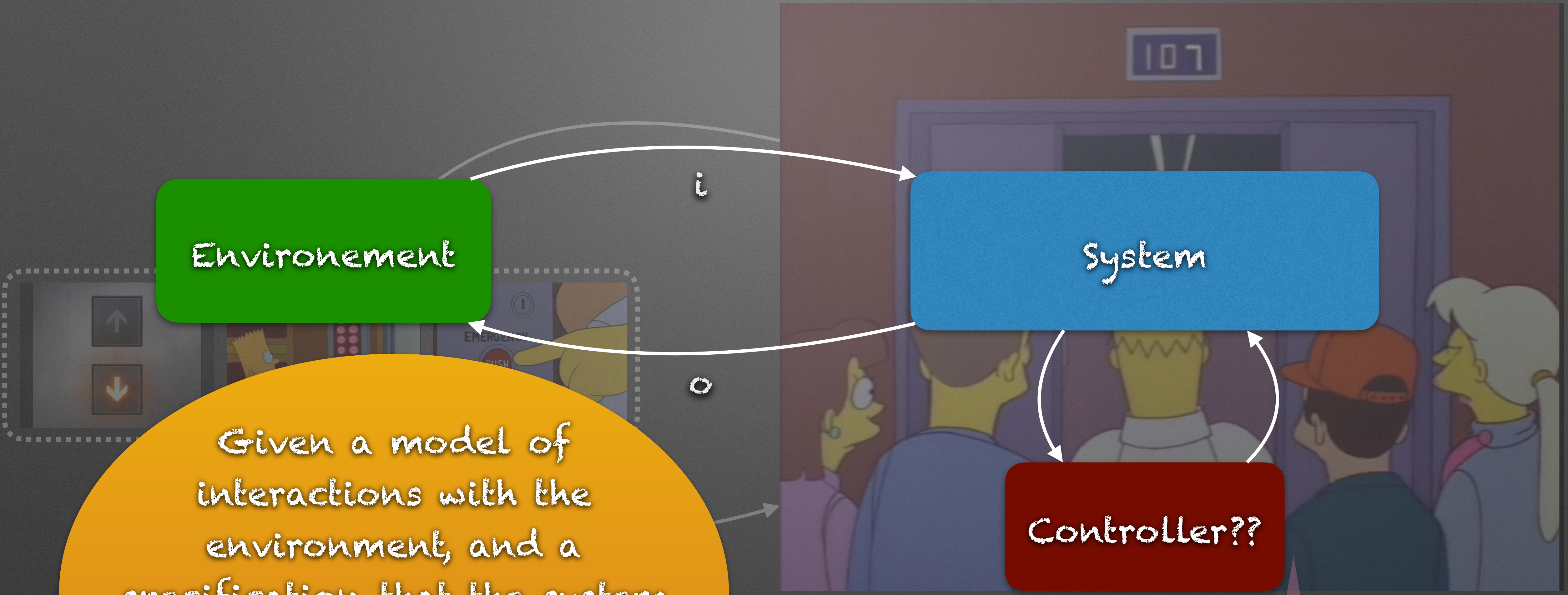
- Doo at the correct level
- Stops when someone calls

EMERGENCY!!

....

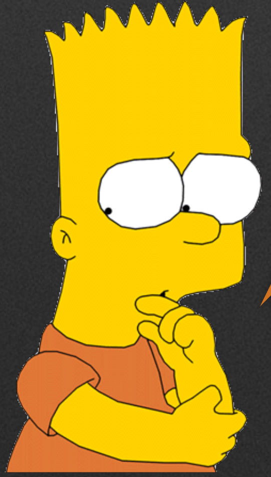


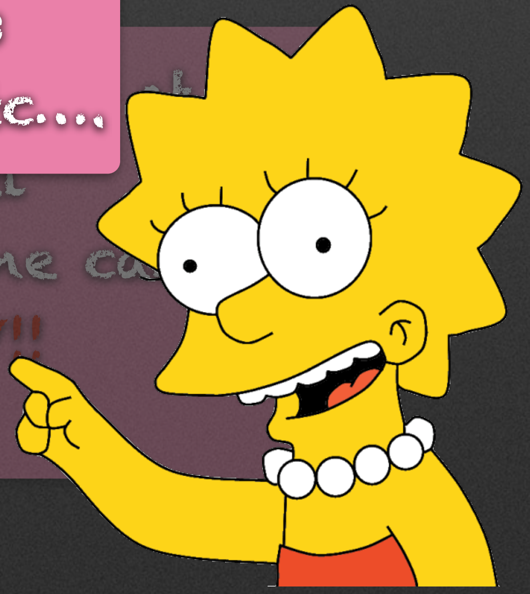
Games in Formal Verification



Given a model of interactions with the environment, and a specification that the system needs to satisfy, does there exist a controller who can guarantee that?

Specifications written in LTL etc…

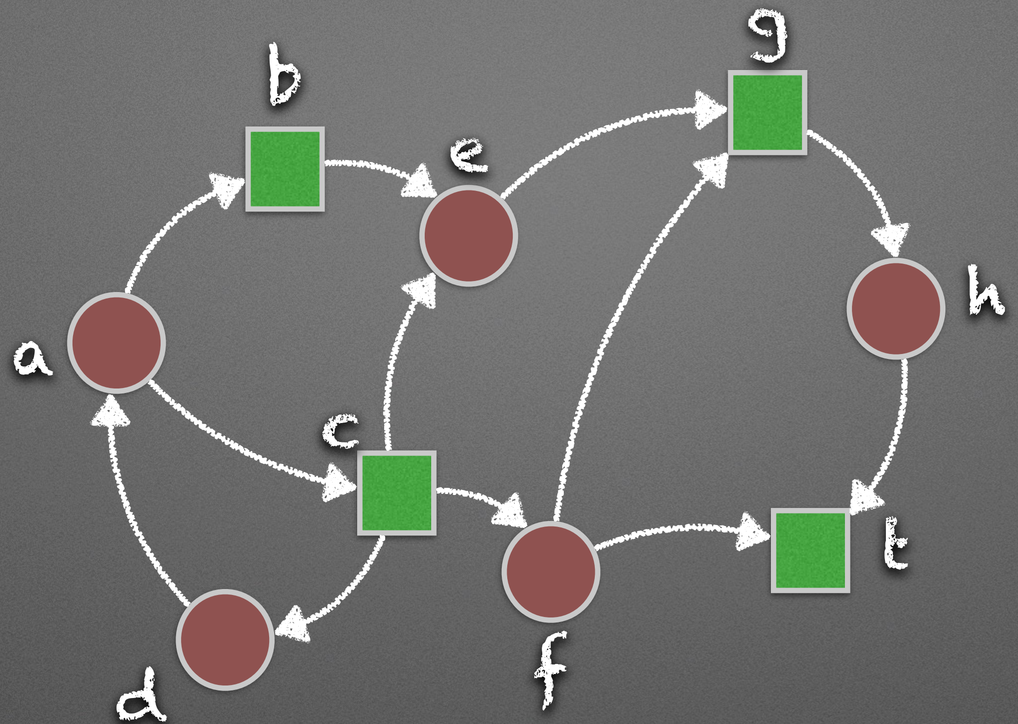




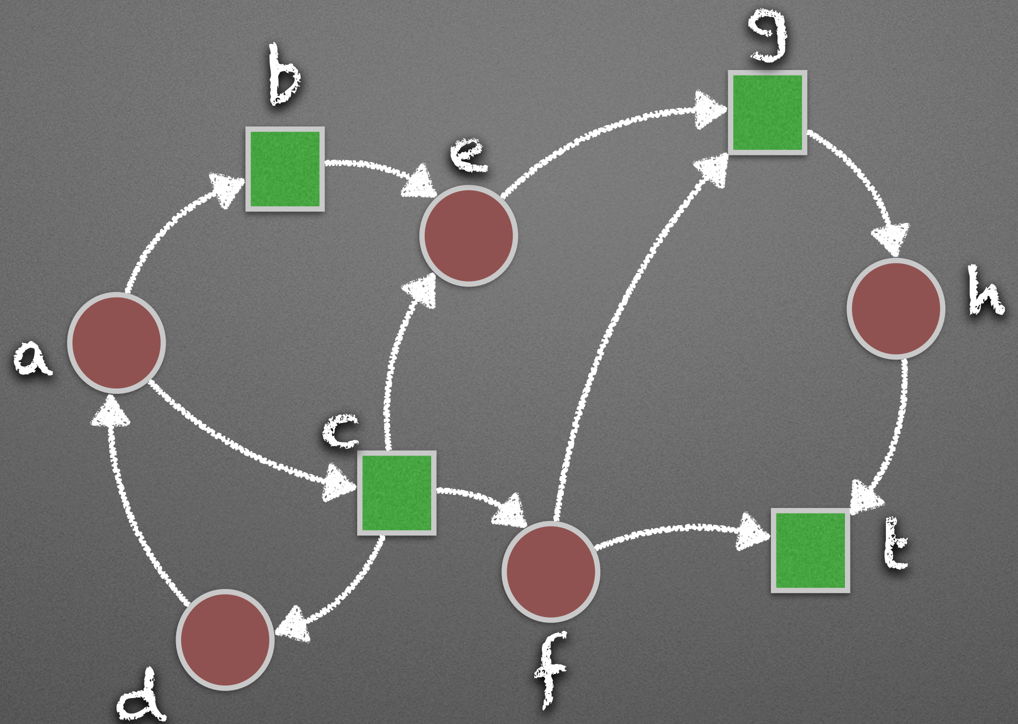
We construct a two-player zero-sum game which models the interactions, encode specs in winning conditions, and

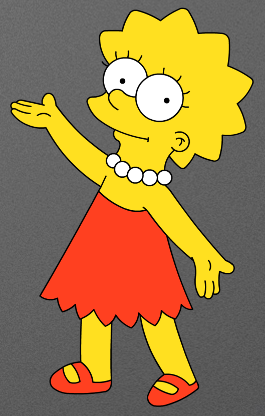
Winning strategy == correct-by-design controller

Background: Turn-based Graph Games

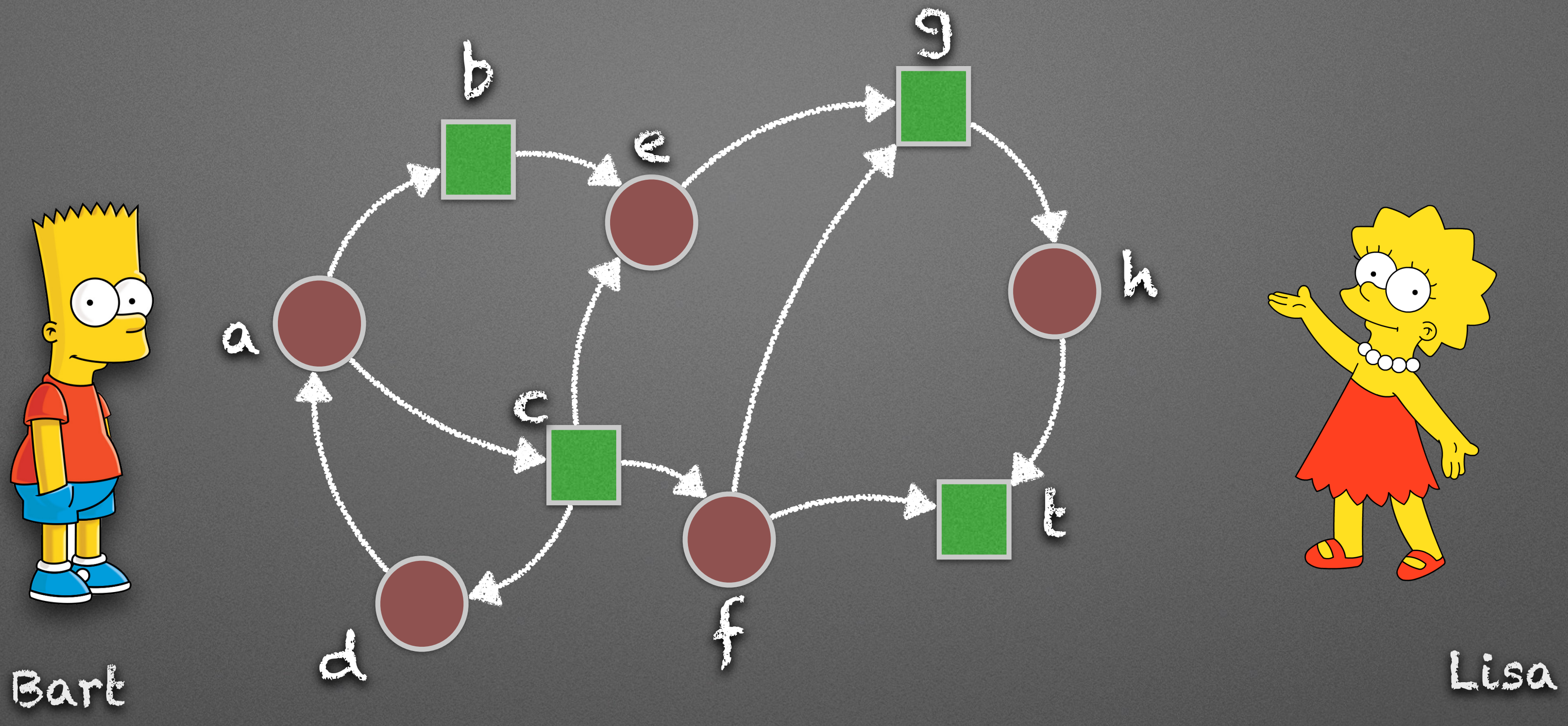


Background: Turn-based Graph Games

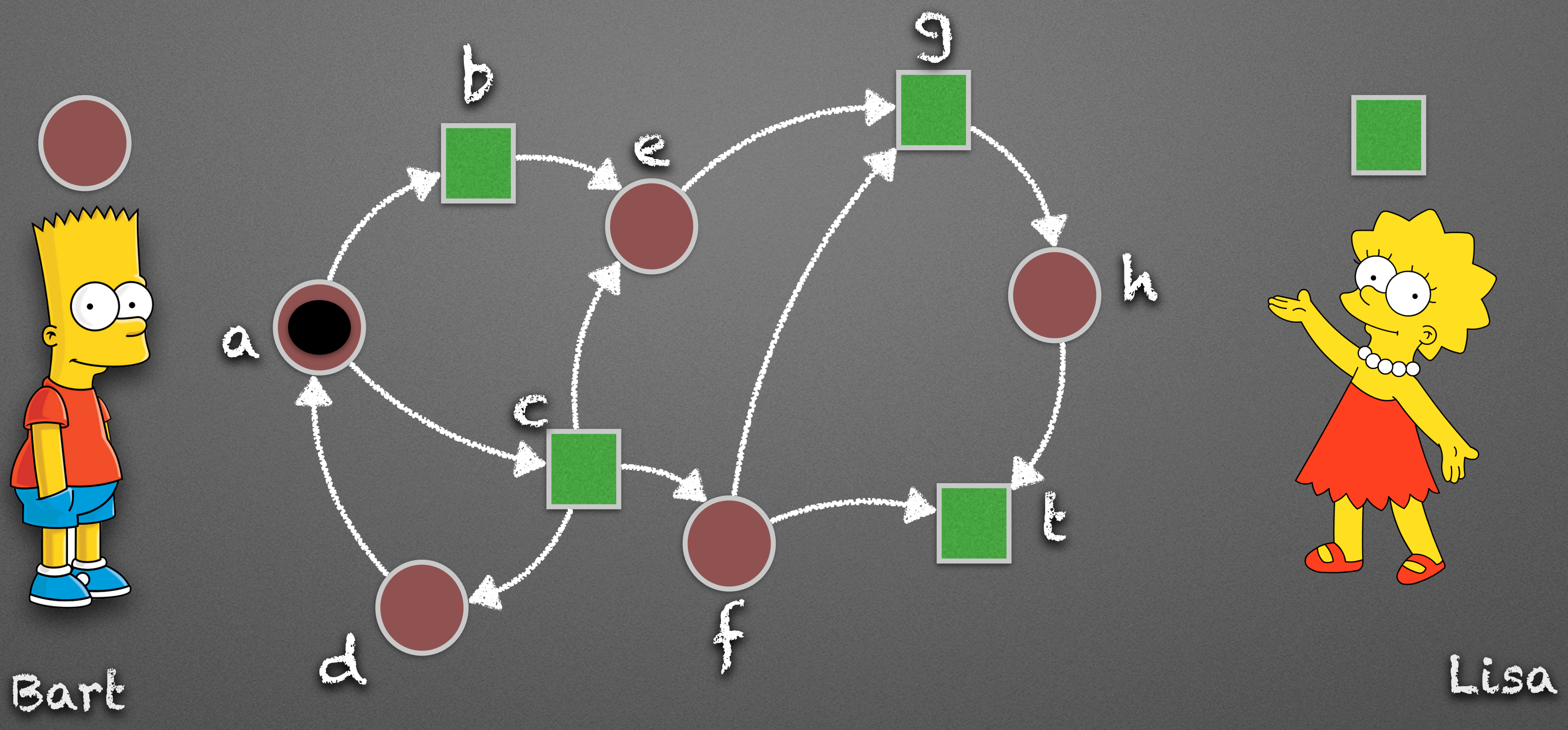




Background: Turn-based Graph Games

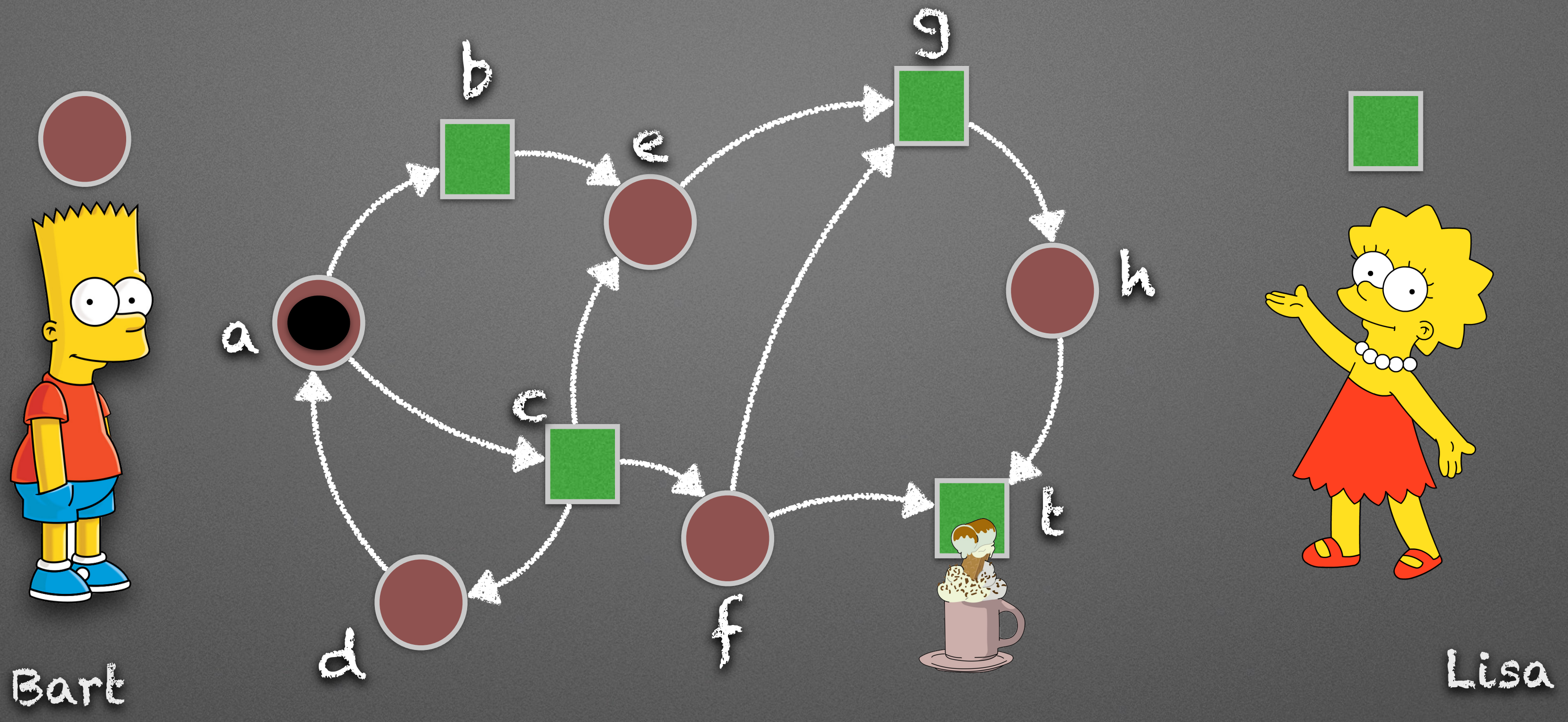


Background: Turn-based Graph Games



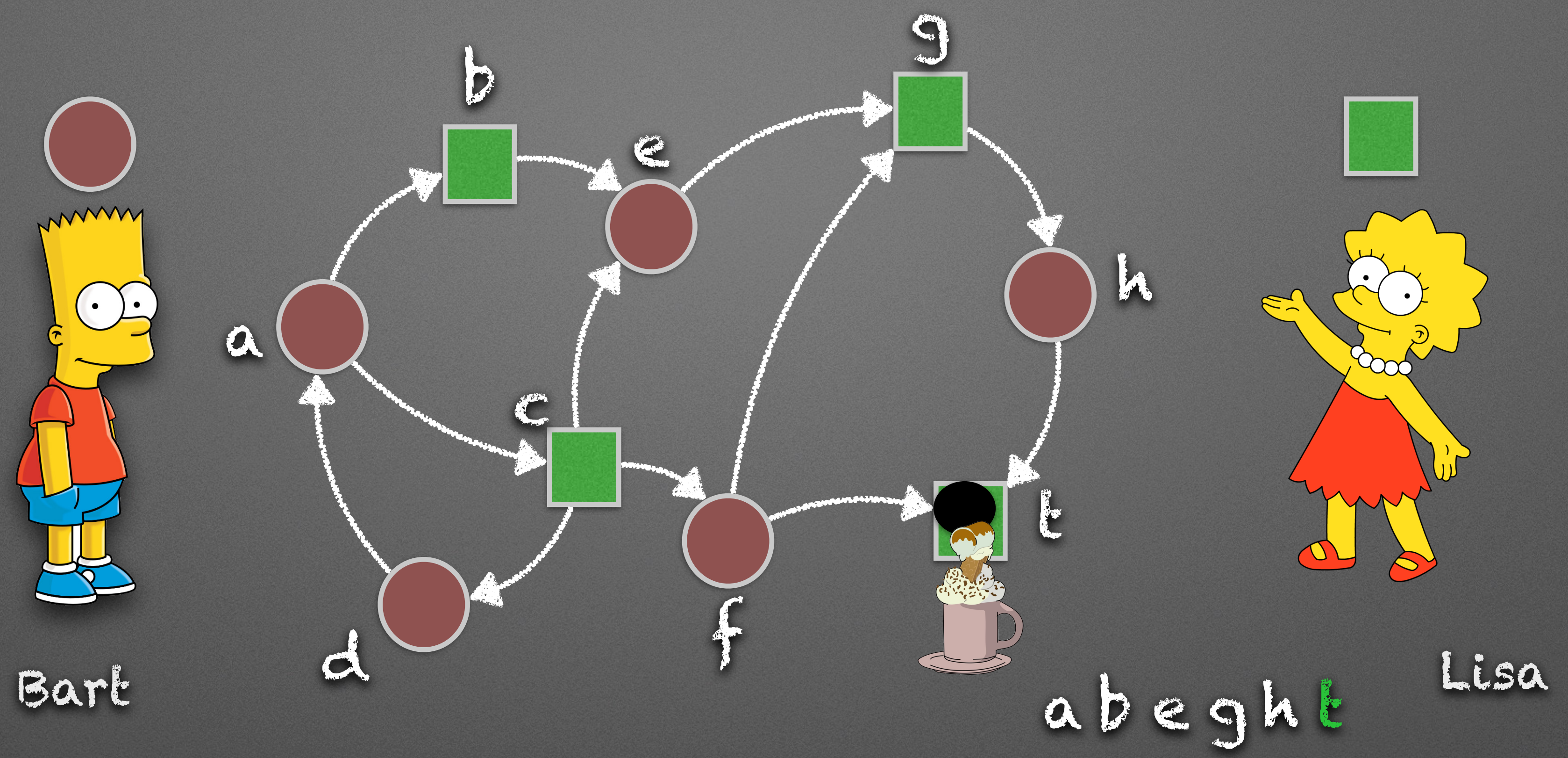
Turn-based: Players alternate turns in moving the token

Background: Turn-based Graph Games



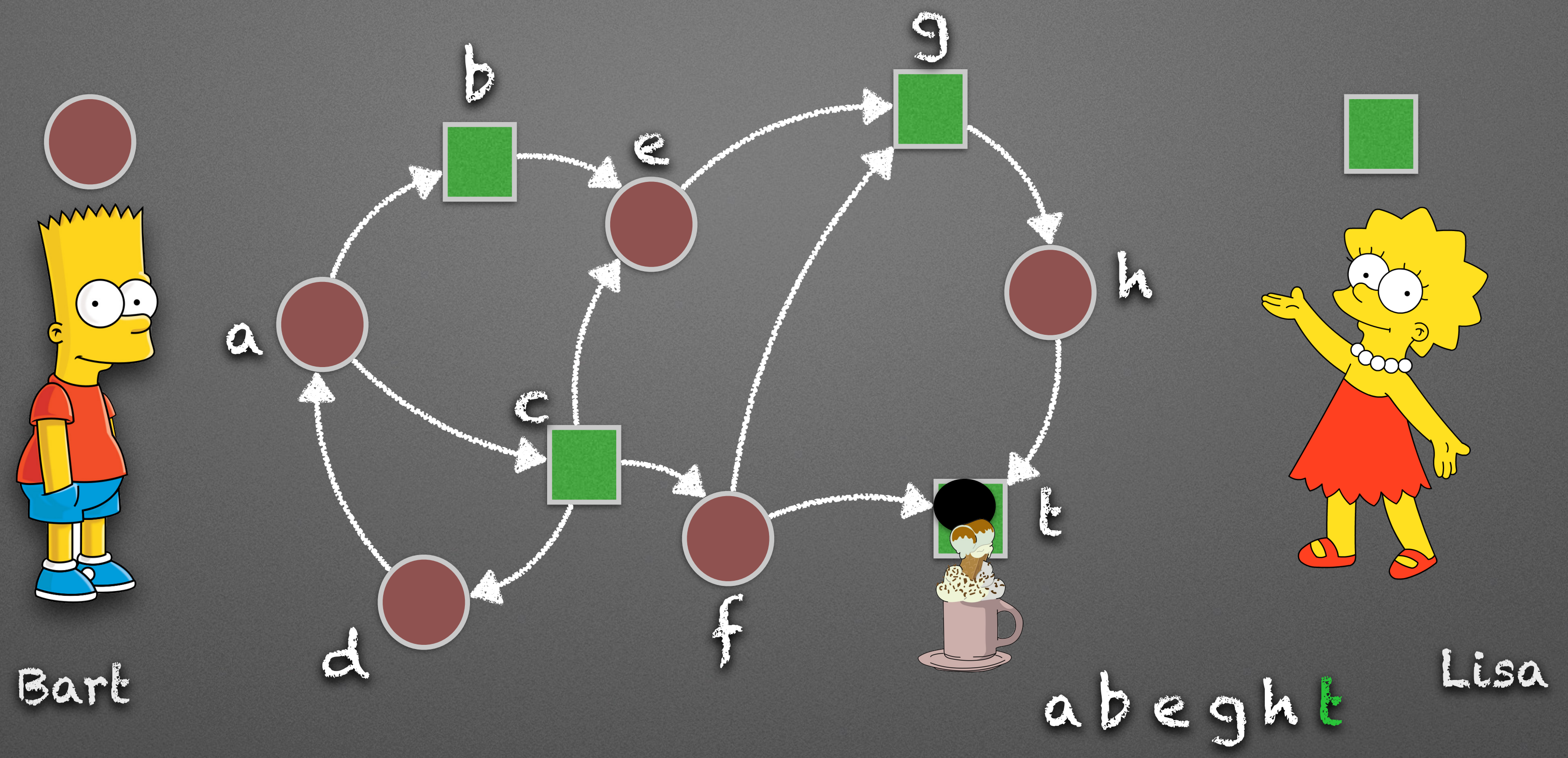
Turn-based: Players alternate turns in moving the token

Background: Turn-based Graph Games



Turn-based: Players alternate turns in moving the token

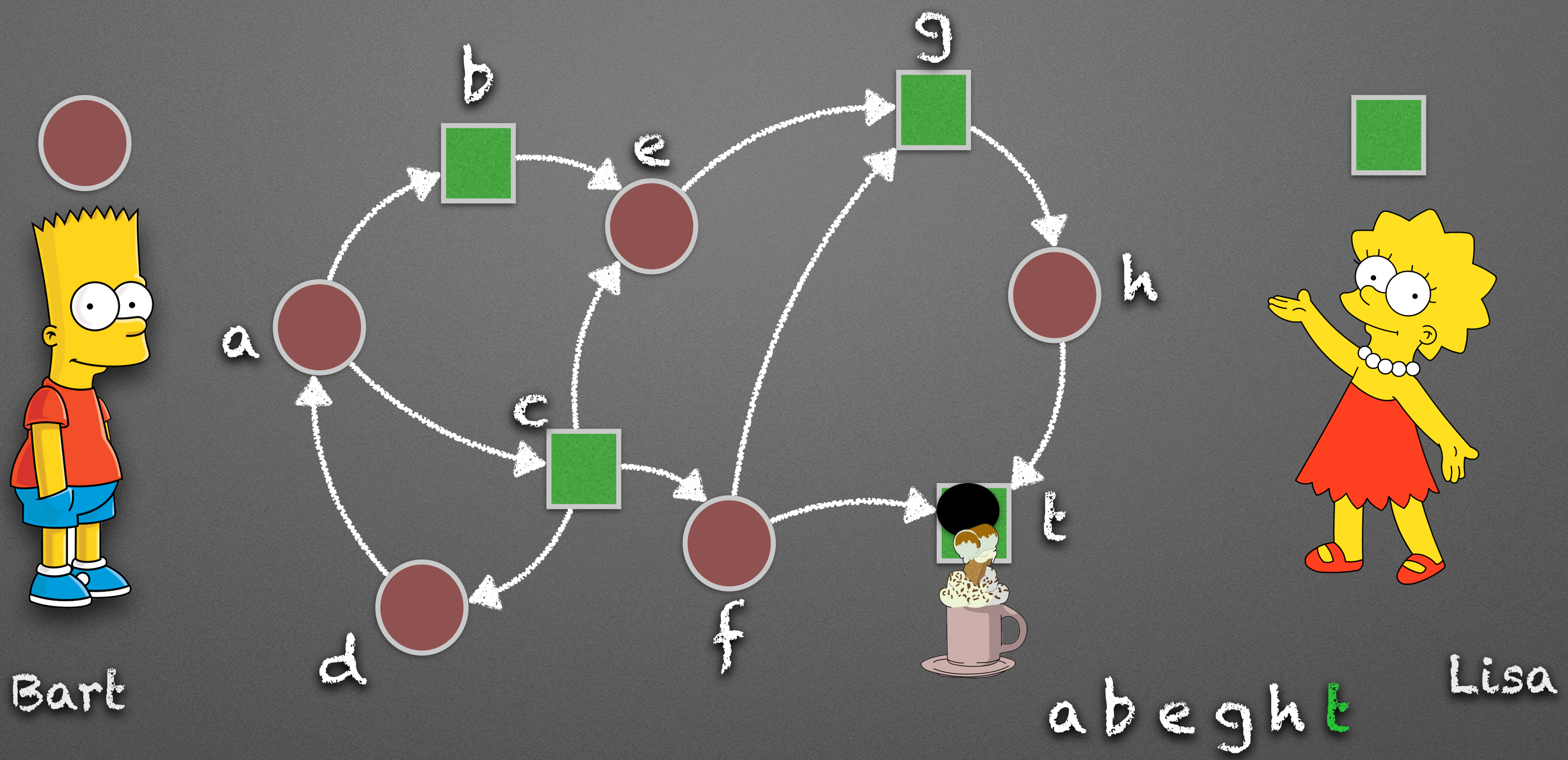
Background: Turn-based Graph Games



Turn-based: Players alternate turns in moving the token

Winning Conditions: Reachability, Buchi, Parity

Background: Turn-based Graph Games

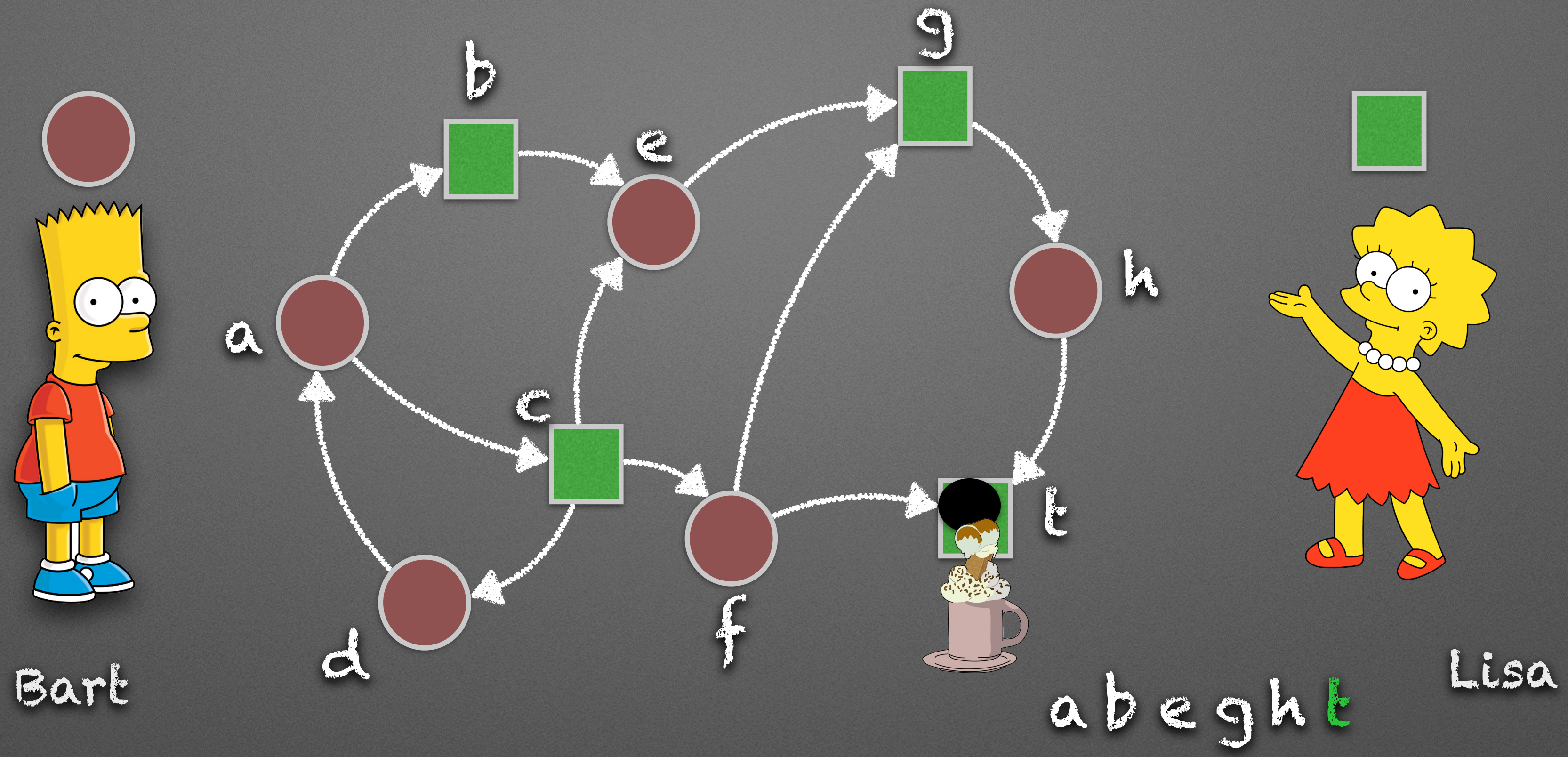


Turn-based: Players alternate turns in moving the token

Winning Conditions: Reachability, Buchi, Parity

Zero-sum games: Every infinite play has a winner

Background: Turn-based Graph Games



Turn-based: Players alternate turns in moving the token

Winning Conditions: Reachability, Buchi, Parity

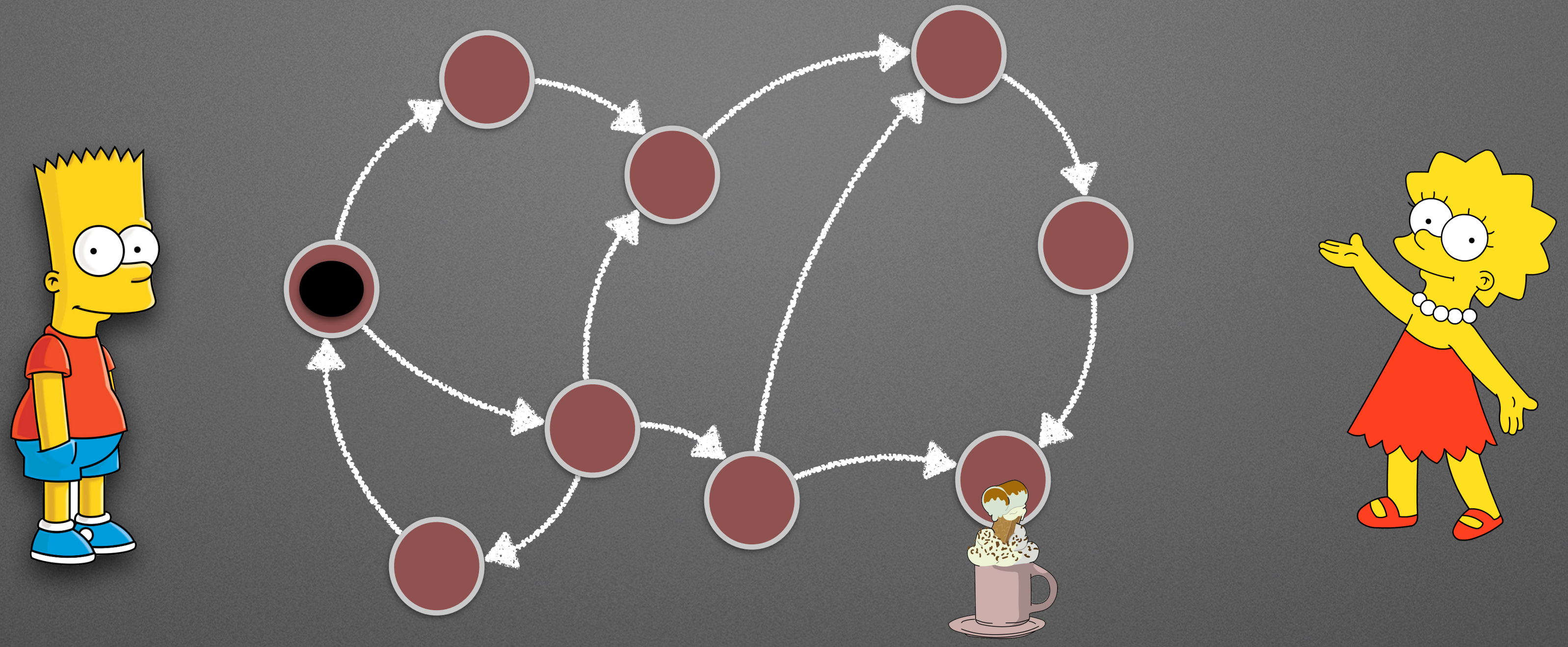
Zero-sum games: Every infinite play has a winner

Decision Problem:

Input: A game graph $\mathcal{G}$, a winning condition $\mathscr{W}$ for Bart, and initial configuration (vertex) v.

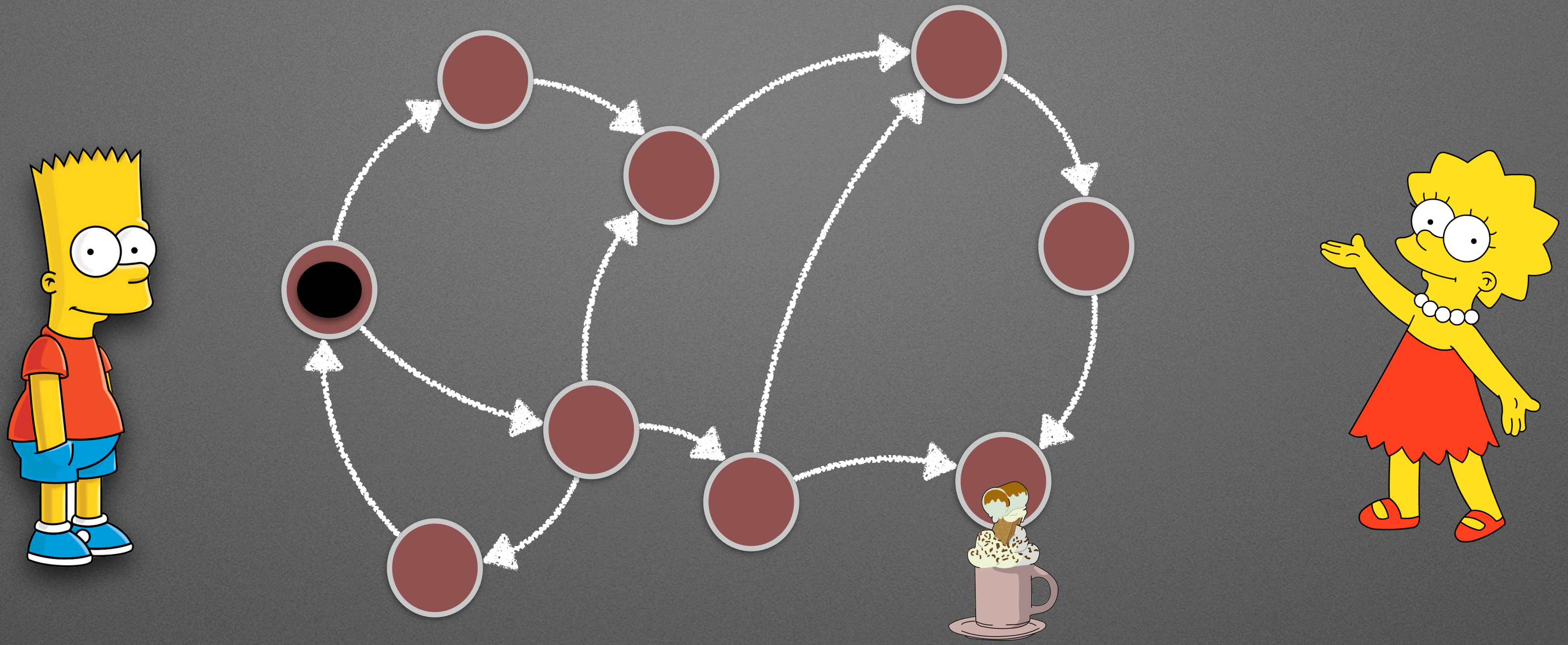
Output: Yes, iff Bart has a winning strategy for $\mathscr{W}$ from v in $\mathcal{G}$

Bidding Games on Graphs



Graph Games: Two-player zero-sum infinite-duration games

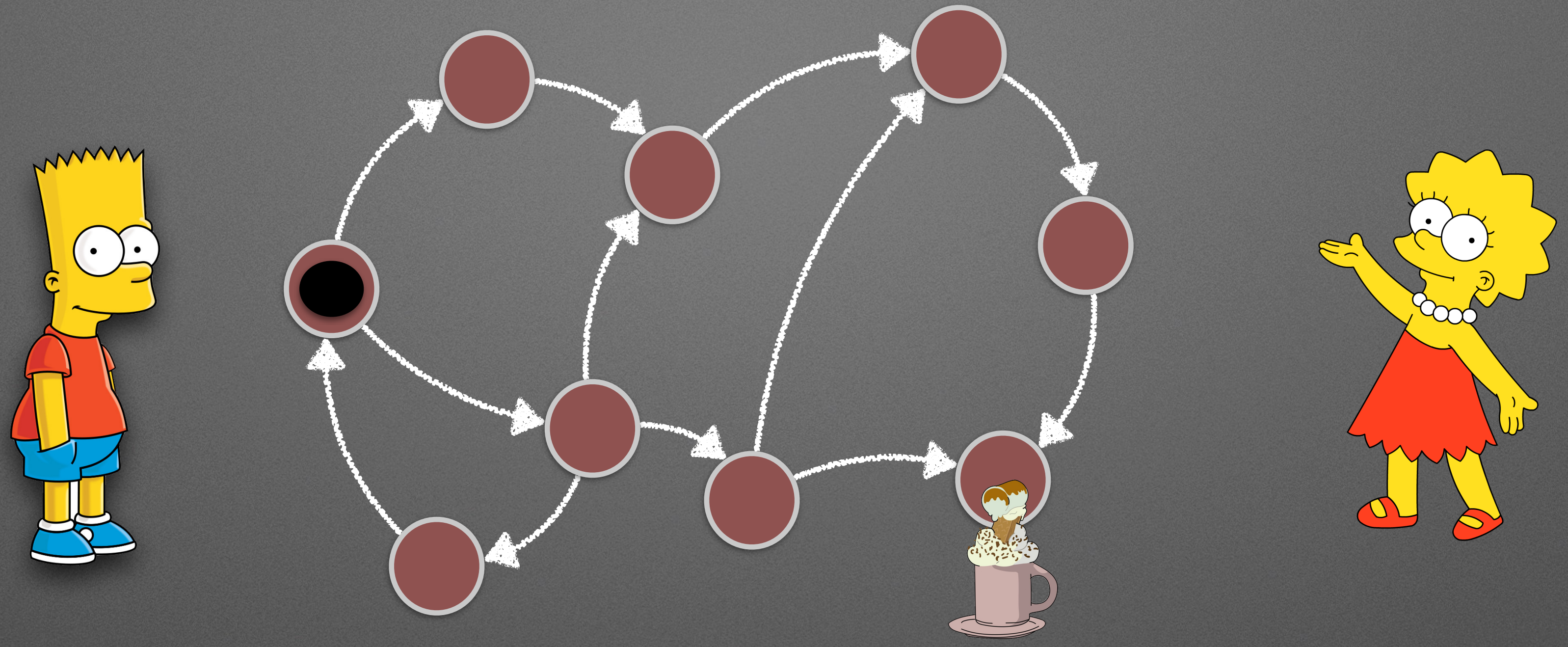
Bidding Games on Graphs



Graph Games: Two-player zero-sum infinite-duration games

Both players have budgets
In each turn, each player bids for getting the turn to move the token

Bidding Games on Graphs



Graph Games: Two-player zero-sum infinite-duration games

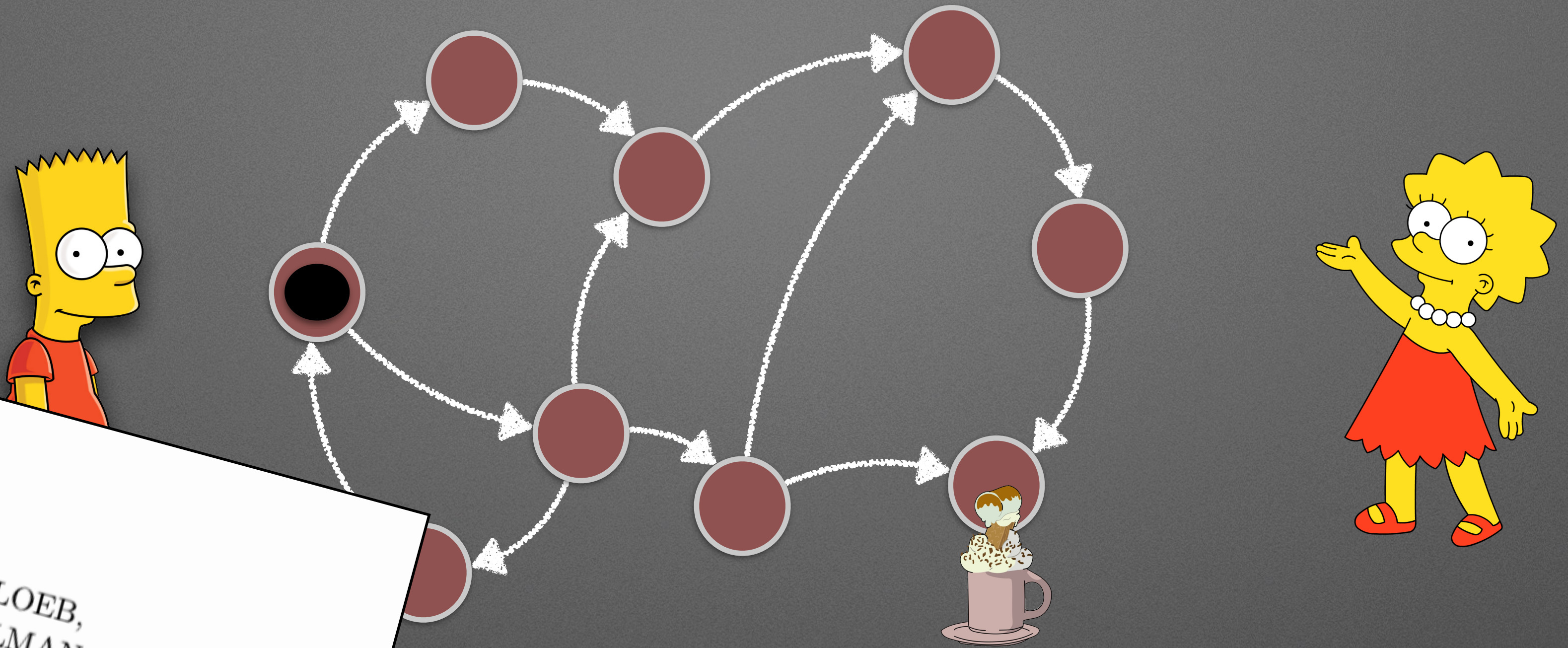
Both players have budgets
In each turn, each
player bids for getting
the turn to move the
token

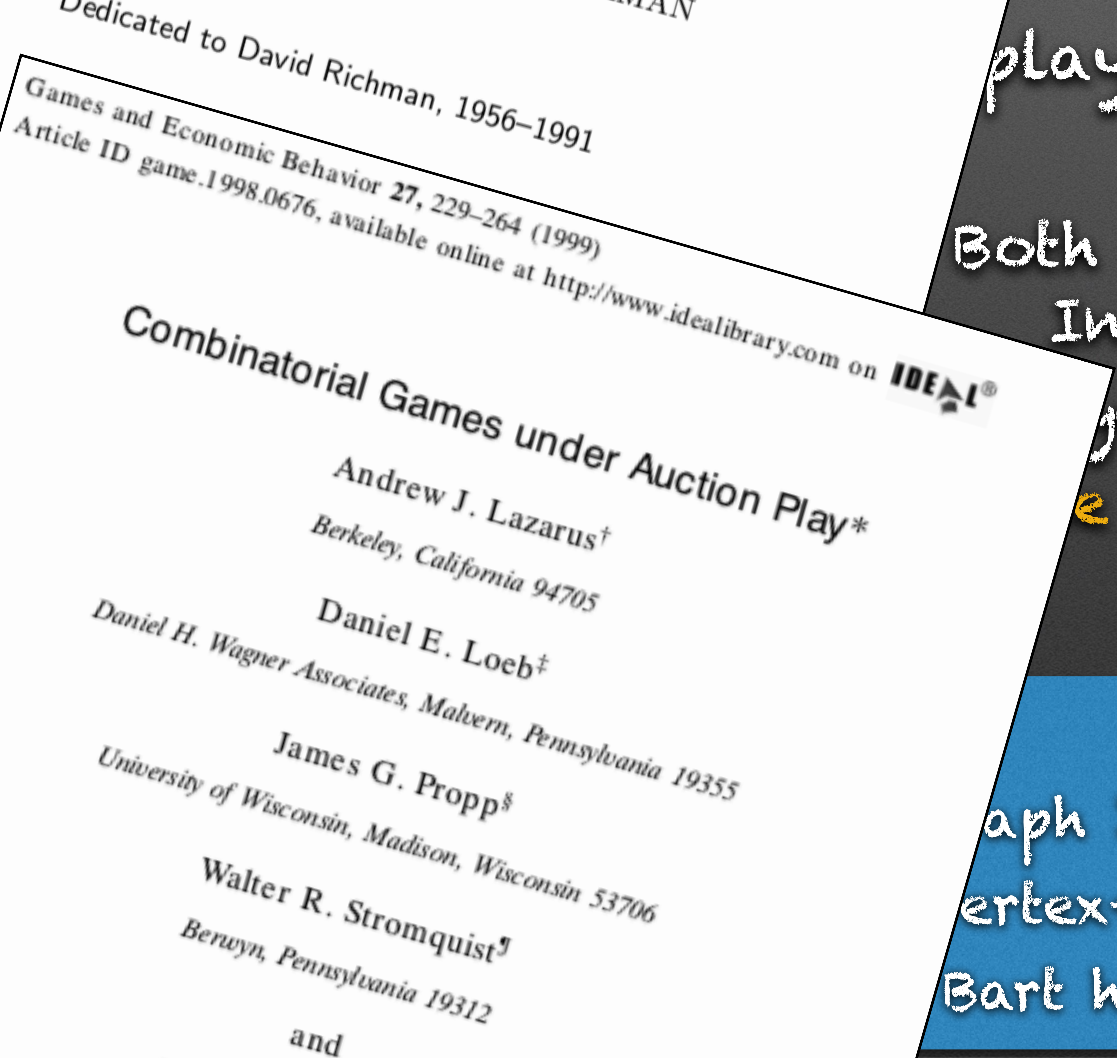
Decision Problem:

Input: A game graph $\mathcal{G}$, a winning condition $\mathscr{W}$ for Bart, and initial configuration (vertex+budget) c.

Output: Yes, iff Bart has a winning strategy for $\mathscr{W}$ from c in $\mathcal{G}$

Bidding Games on Graphs





Richman Games

ANDREW J. LAZARUS, DANIEL E. LOEB,
JAMES G. PROPP, AND DANIEL ULLMAN

Dedicated to David Richman, 1956–1991

Games and Economic Behavior **27**, 229–264 (1999)
Article ID game.1998.0676, available online at http://www.idealibrary.com on IDEAL®

Combinatorial Games under Auction Play*

Andrew J. Lazarus[†]
Berkeley, California 94705

Daniel E. Loeb[‡]
Daniel H. Wagner Associates, Malvern, Pennsylvania 19355

James G. Propp[§]
University of Wisconsin, Madison, Wisconsin 53706

Walter R. Stromquist[¶]
Berwyn, Pennsylvania 19312

and

player zero-sum infinite-duration games

Both players have budgets
In each turn, each
yer bids for getting
e turn to move the
token

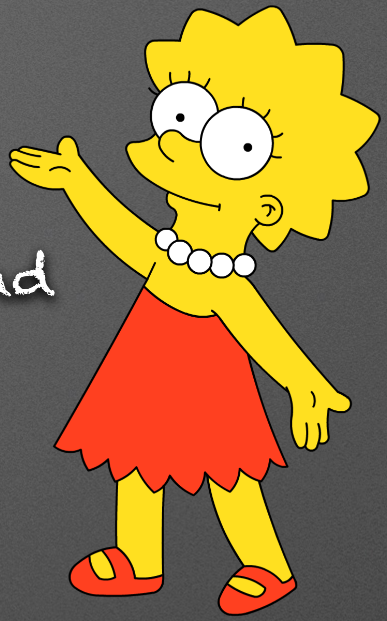
Decision Problem:

aph $\mathcal{G}$, a winning condition $\mathcal{W}$ for Bart, and initial
ertex+budget) c.
Bart has a winning strategy for $\mathcal{W}$ from c in $\mathcal{G}$

Bidding Mechanisms

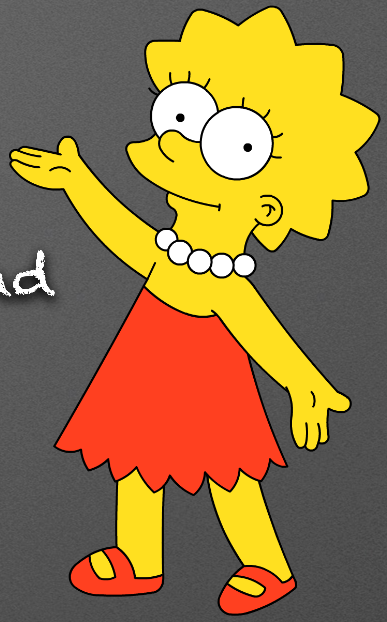
Bidding Mechanisms

In each turn, both players simultaneously submit “legal” bids, and the higher bidder moves the token.



Bidding Mechanisms

In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.



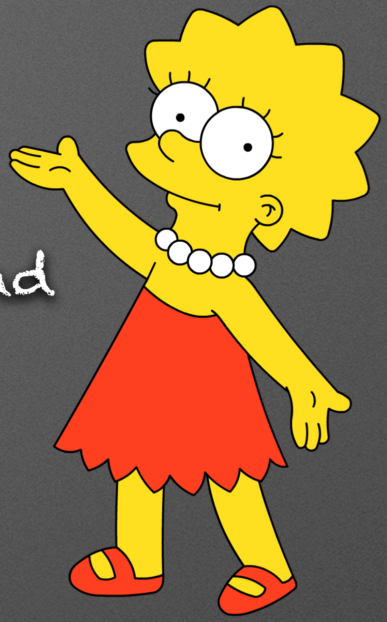
Who pays?

Where?

What?

Bidding Mechanisms

In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.



Who pays?

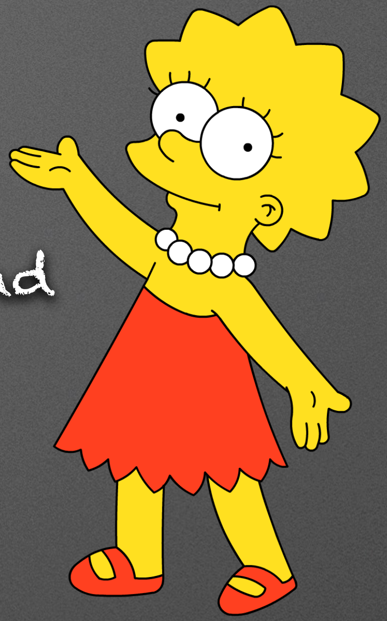
Where?

What?

{ first-price, all-pay }

Bidding Mechanisms

In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.

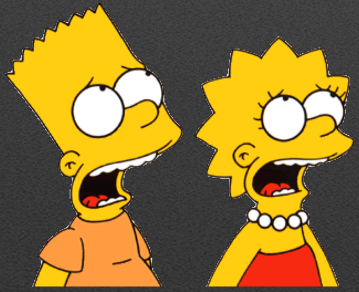


Who pays?

Where?

What?

{ first-price, all-pay }

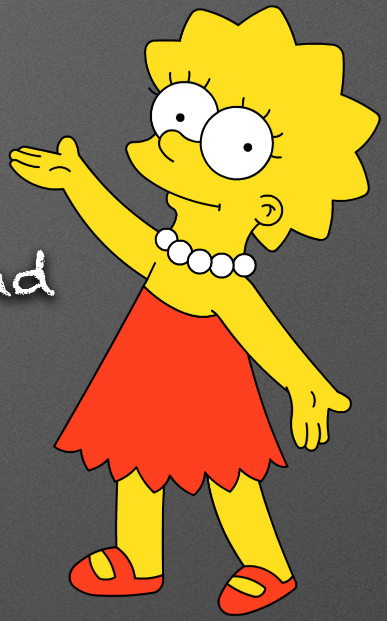


Higher bidder

Both

Bidding Mechanisms

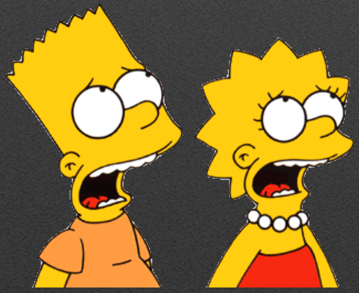
In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.



Who pays? Where? What?

$\{$first-price, all-pay$\}$ × $\{$Richman, Poorman$\}$

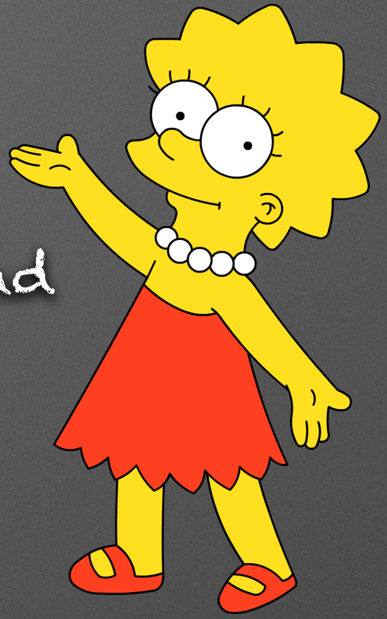
Higher bidder



Both

Bidding Mechanisms

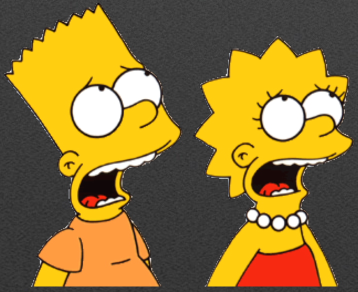
In each turn, both players simultaneously submit “legal” bids, and the higher bidder moves the token.



Who pays? Where? What?

{first-price, all-pay} × {Richman, Poorman}

Higher bidder

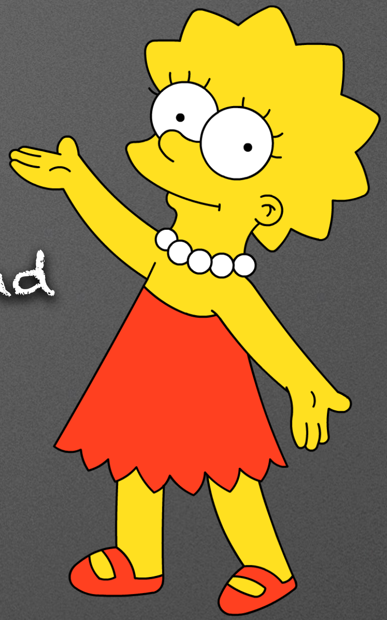


Both

pay the other bidder

Bidding Mechanisms

In each turn, both players simultaneously submit “legal” bids, and the higher bidder moves the token.



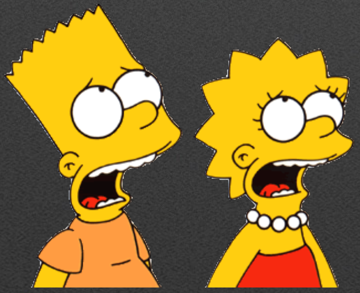
Who pays?

Where?

What?

{first-price, all-pay} × {Richman, Poorman}

Higher bidder



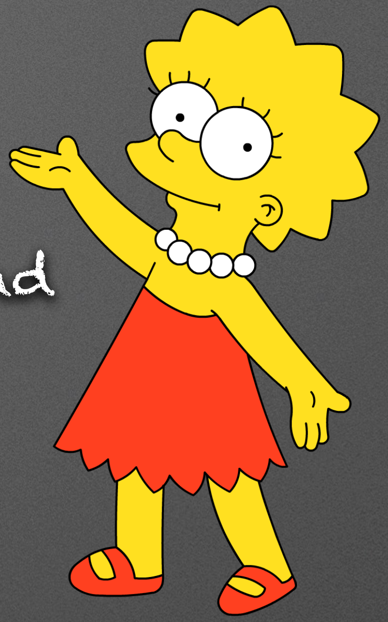
Both

pay the other bidder

pay the bank

Bidding Mechanisms

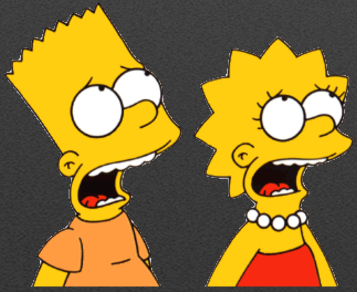
In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.



Who pays? Where? What?

{ first-price, all-pay } × { Richman, Poorman } × {continuous, discrete}

Higher bidder



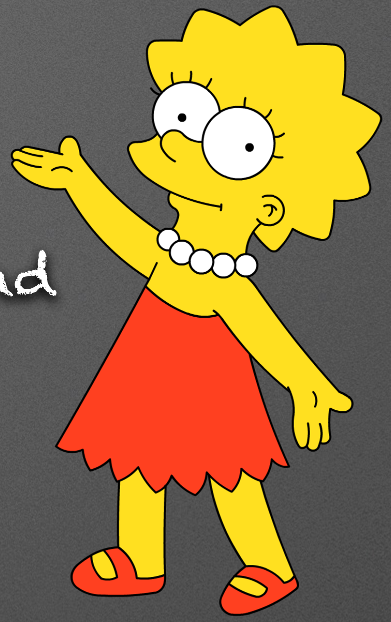
Both

pay the other bidder

pay the bank

Bidding Mechanisms

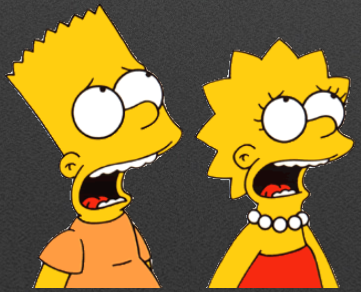
In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.



Who pays? Where? What?

{ first-price, all-pay } × { Richman, Poorman } × {continuous, discrete}

Higher bidder



Both

pay the other bidder

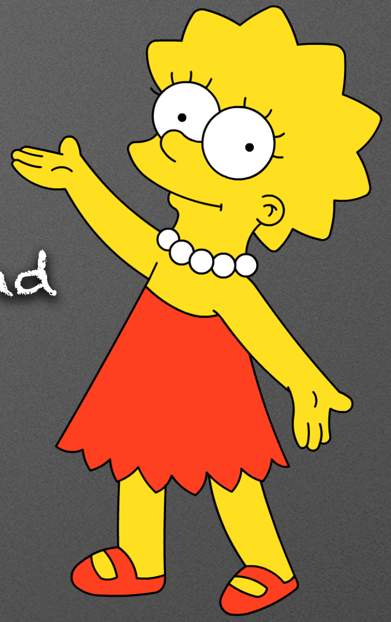
pay the bank

arbitrary

fixed granularity

Bidding Mechanisms

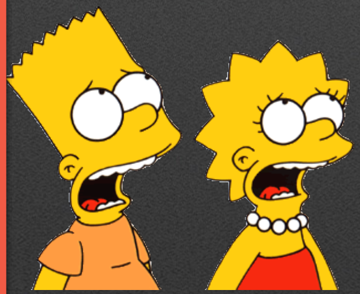
In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.



Who pays? Where? What?

{ first-price, all-pay } × { Richman, Poorman } × {continuous, discrete}

Higher bidder



Both

pay the other bidder

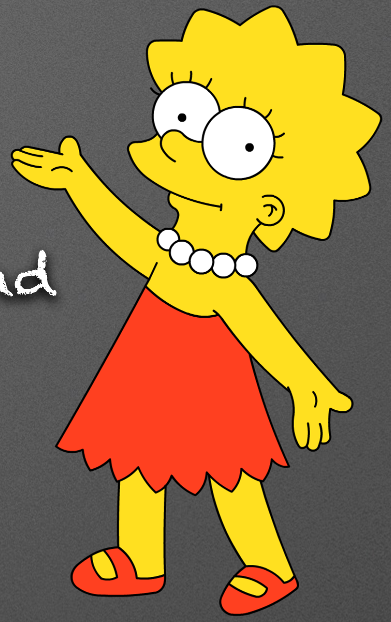
pay the bank

arbitrary

fixed granularity

Bidding Mechanisms

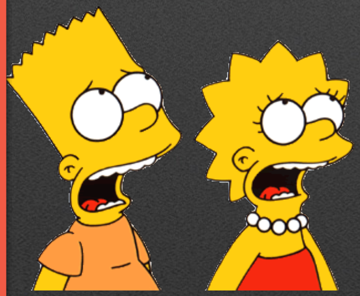
In each turn, both players simultaneously submit "legal" bids, and the higher bidder moves the token.



Who pays? Where? What?

{ first-price, all-pay } × { Richman, Poorman } × {continuous, discrete}

Higher bidder



Both

pay the other bidder

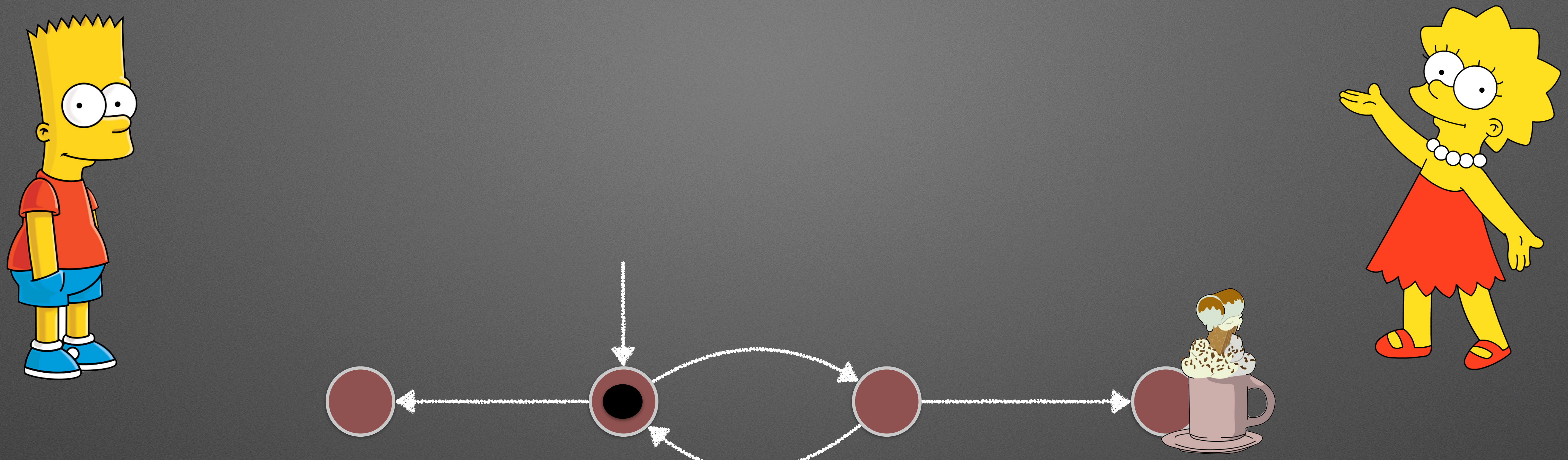
pay the bank

arbitrary

fixed granularity

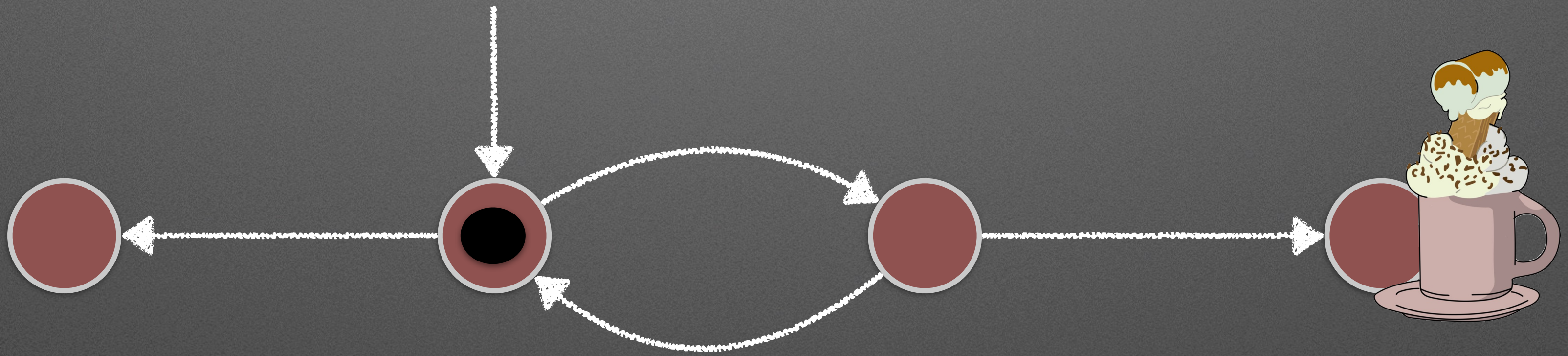
Reachability first-price Richman continuous

[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]



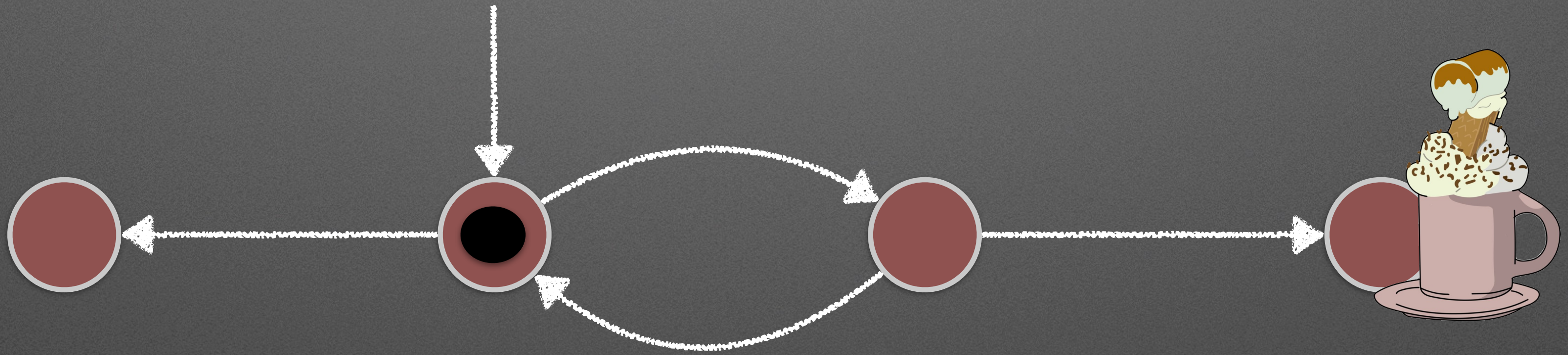
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Given a game and initial budgets, decide which player has a winning bidding strategy from a given vertex.



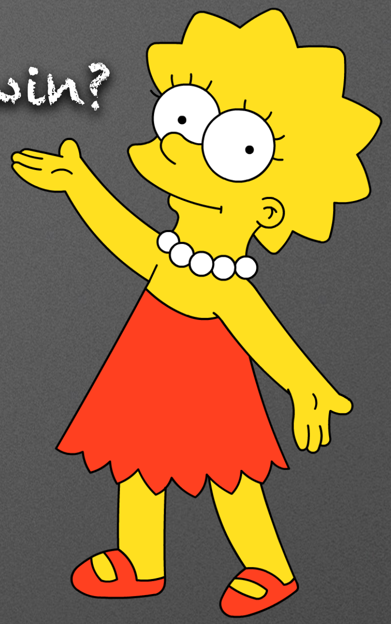
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

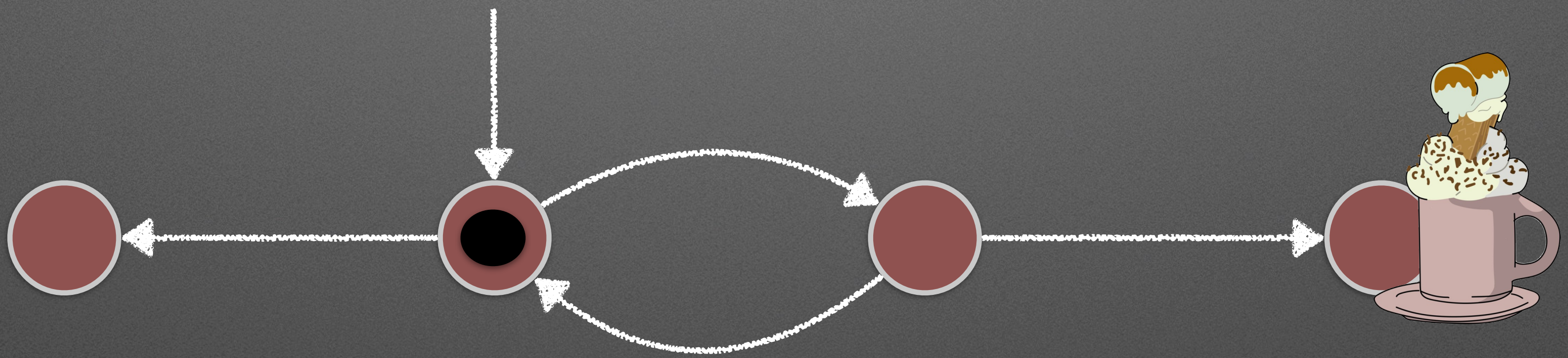
How much initial budget is necessary & sufficient for Bart to win?



Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?





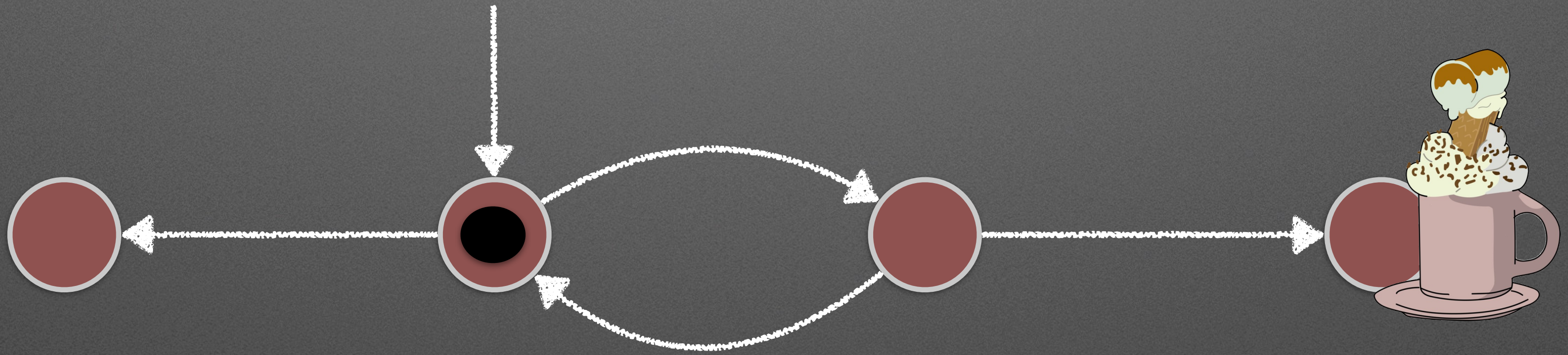
Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?

Easy:

$0.75 + \epsilon$



Total budget is normalised to 1

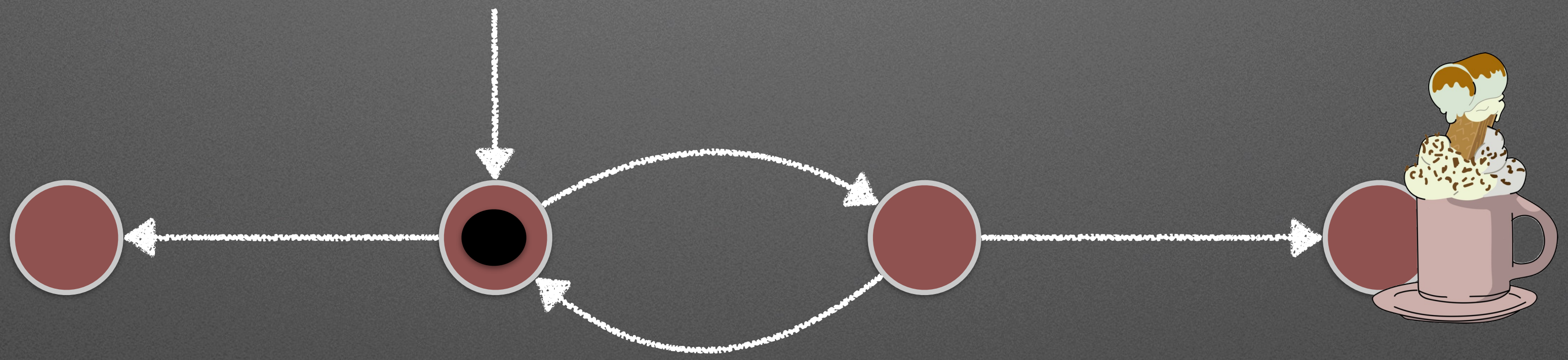
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?

Easy:

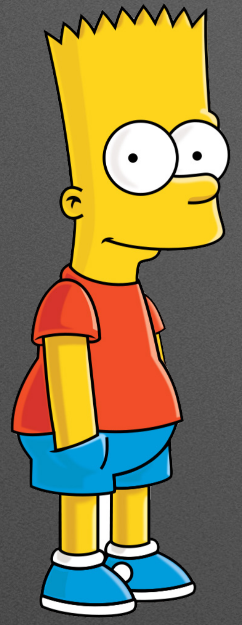
$0.75 + \epsilon$

$0.25 - \epsilon$

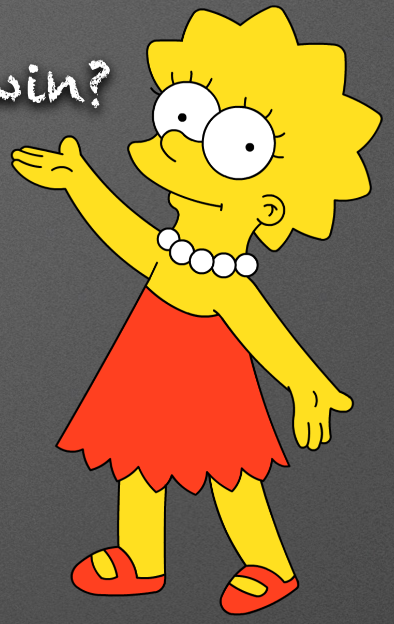


Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

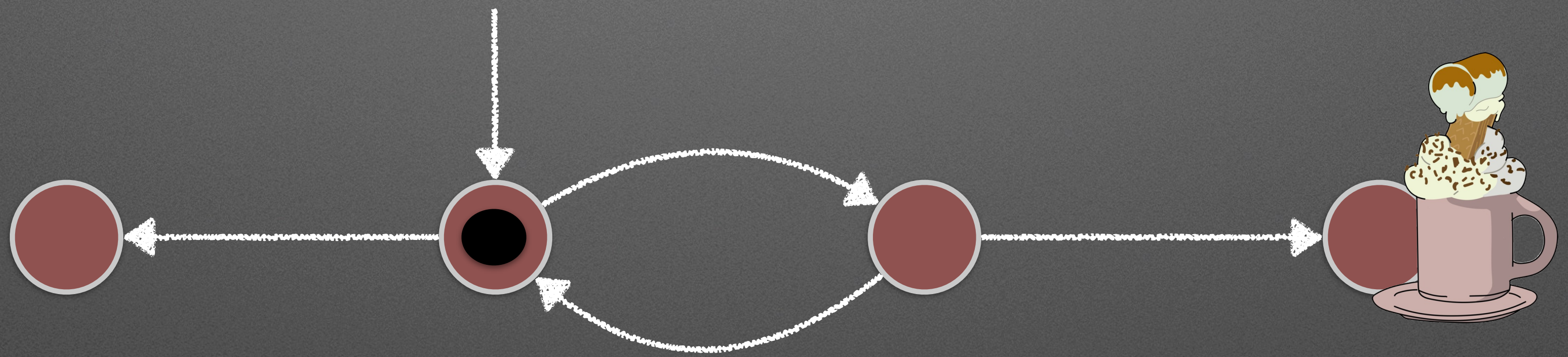


How much initial budget is necessary & sufficient for Bart to win?



Easy:

$0.75 + \epsilon$ 0.25 $0.25 - \epsilon$



Total budget is normalised to 1

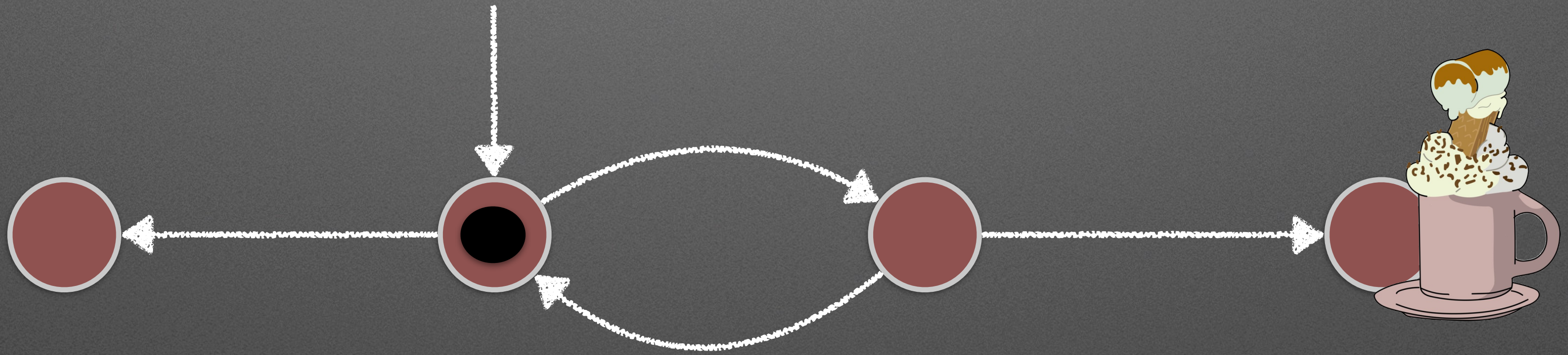
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?

Easy:

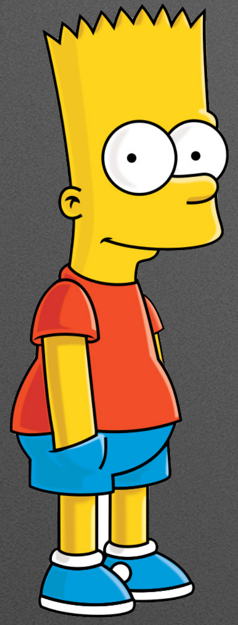
$0.75 + \epsilon$

$0.25 - \epsilon$



Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

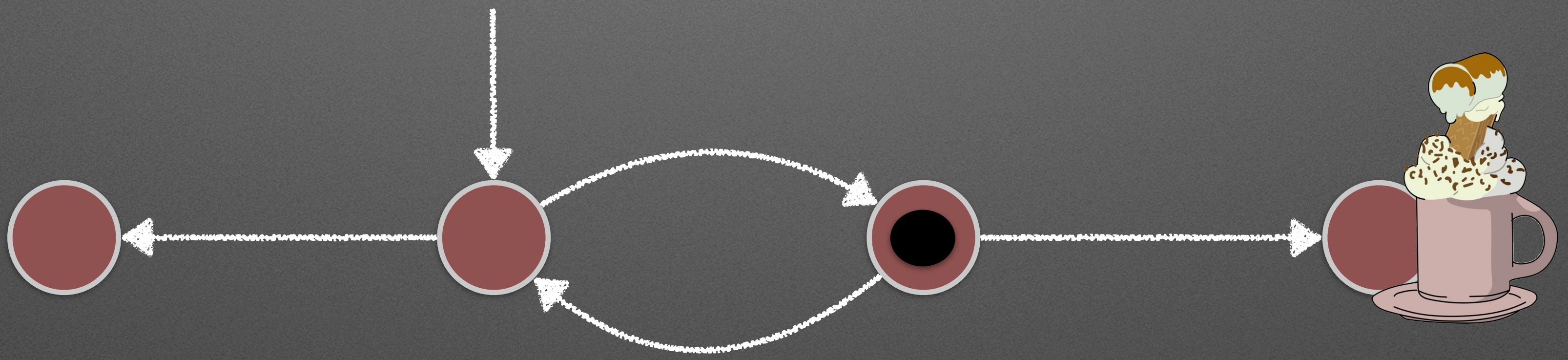


How much initial budget is necessary & sufficient for Bart to win?

Easy:

$0.5 + \epsilon$

$0.5 - \epsilon$



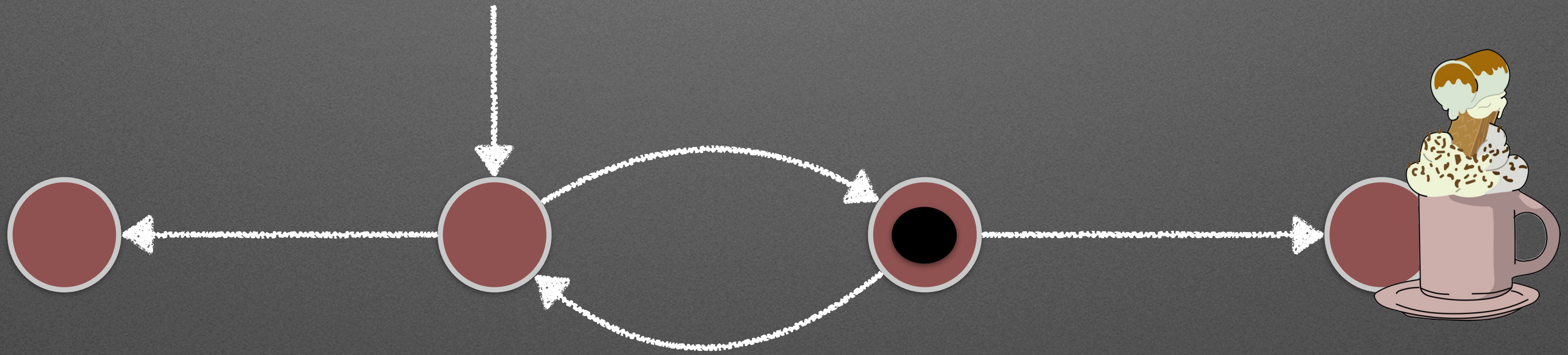
Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?

Easy:

$0.5 + \epsilon$ $0.5 + \epsilon$ $0.5 - \epsilon$

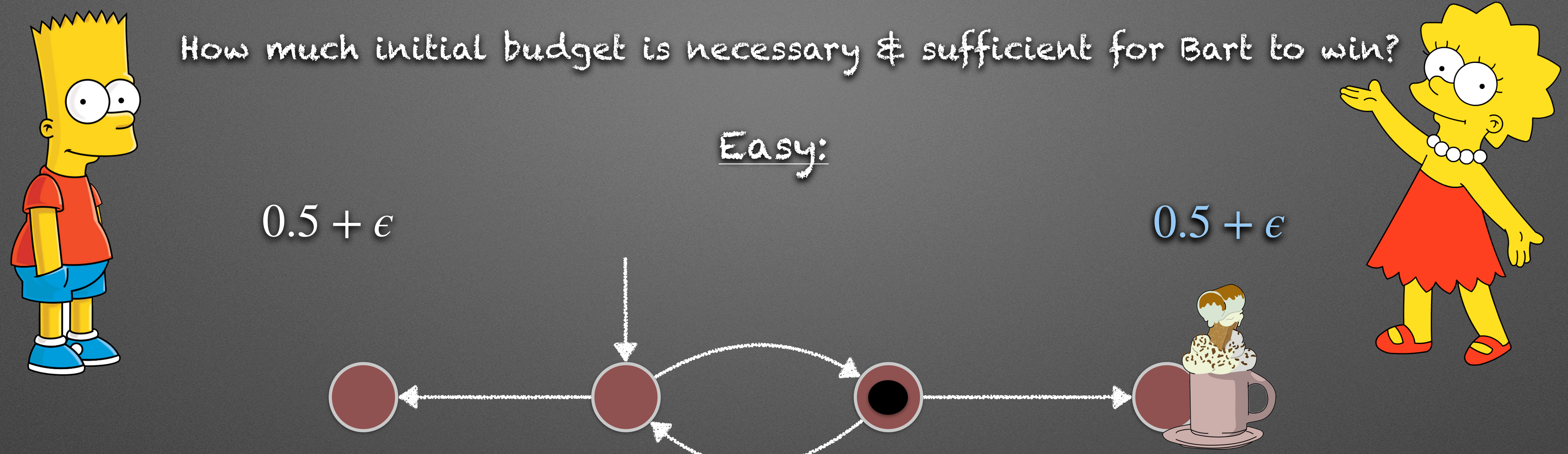


Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?

Easy:

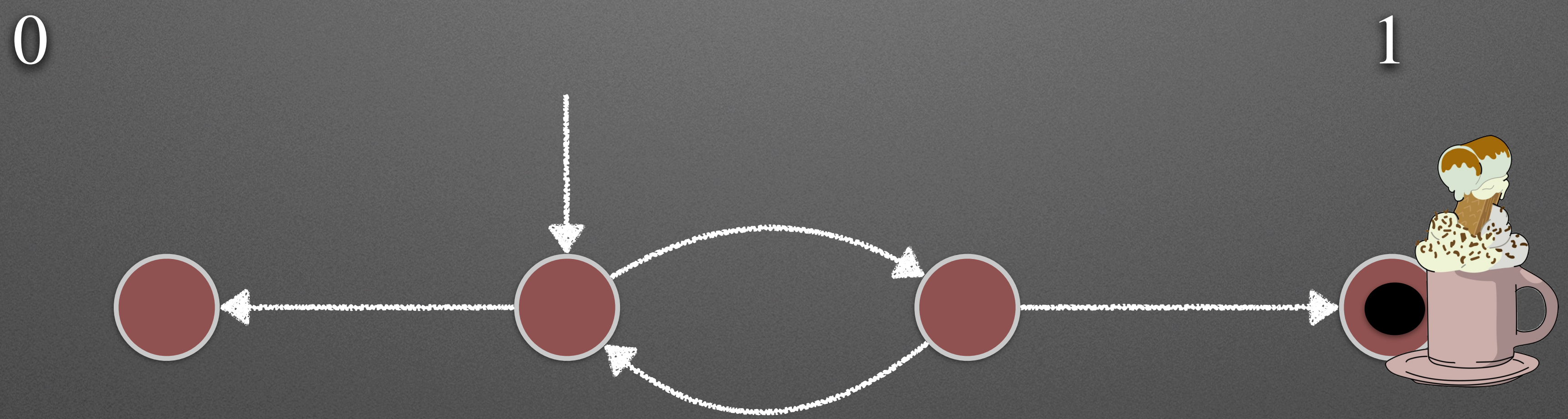


Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?

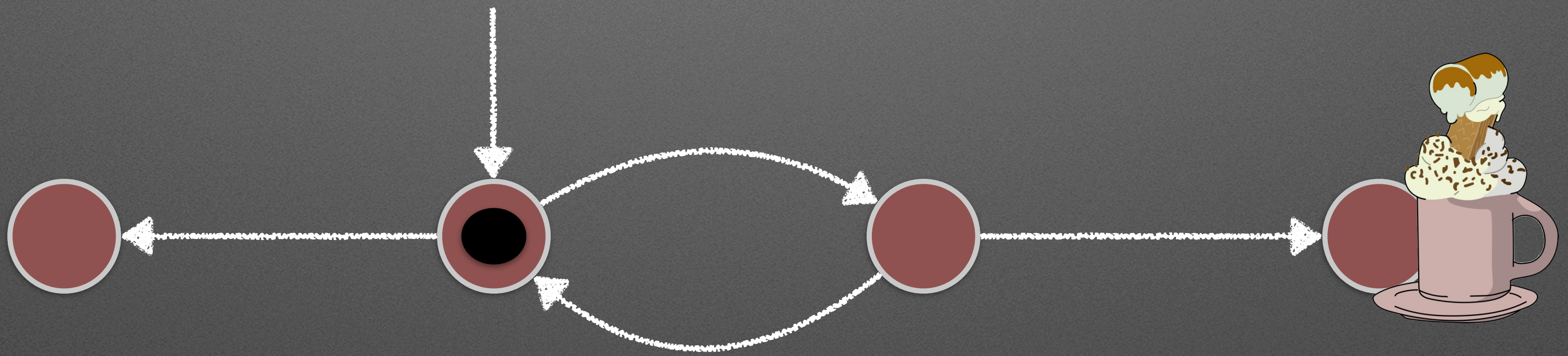
Easy:



Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

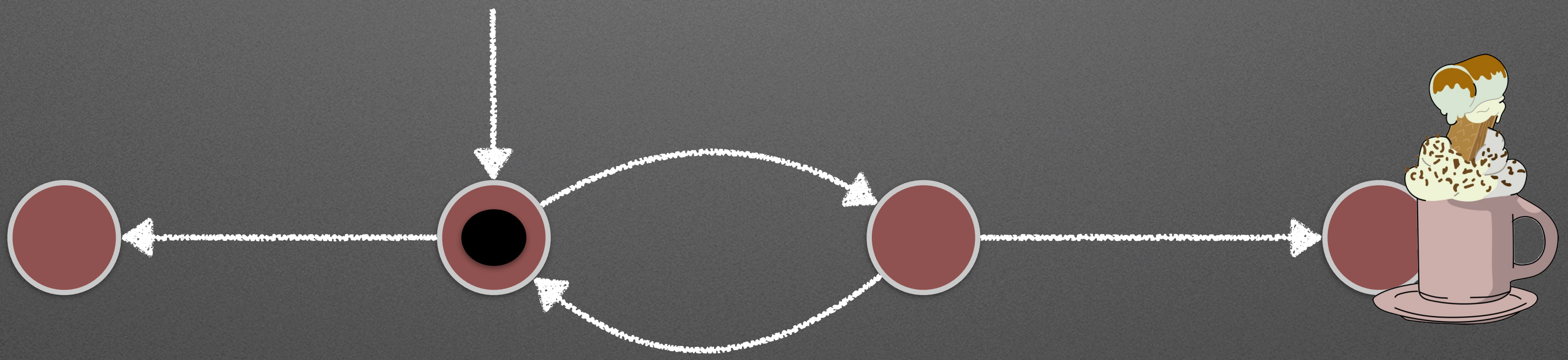


Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

Yes!



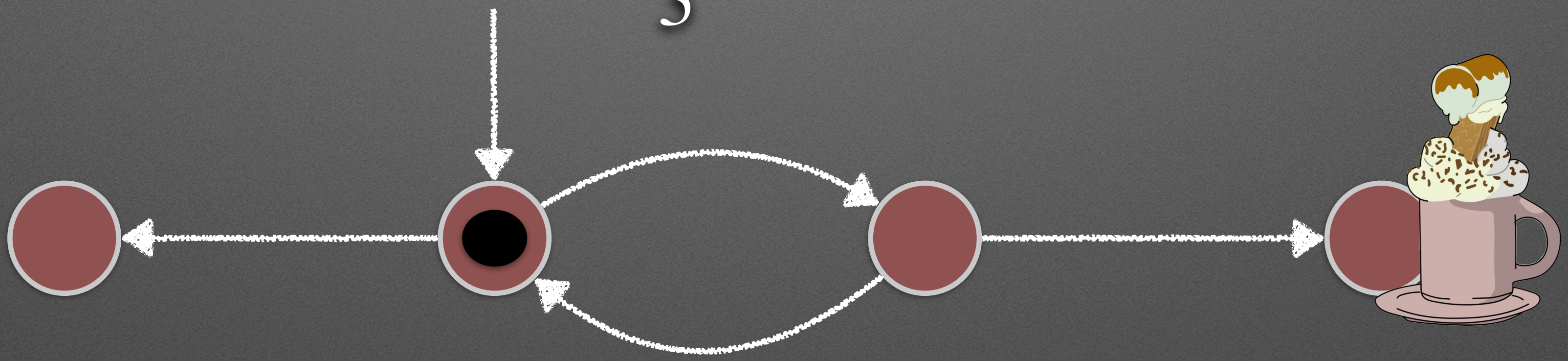
Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

Yes!

$$\frac{2}{3} + \epsilon$$



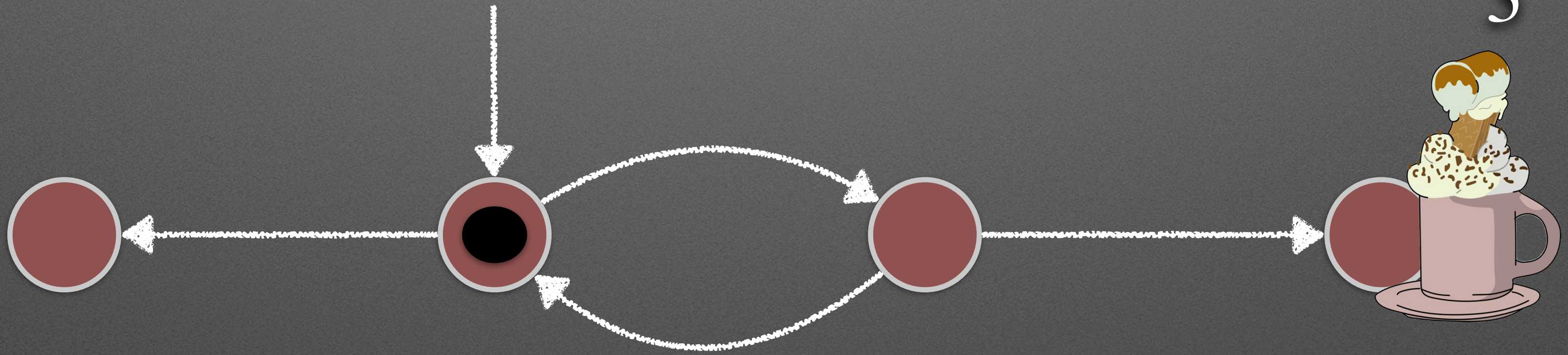
Total budget is normalised to 1

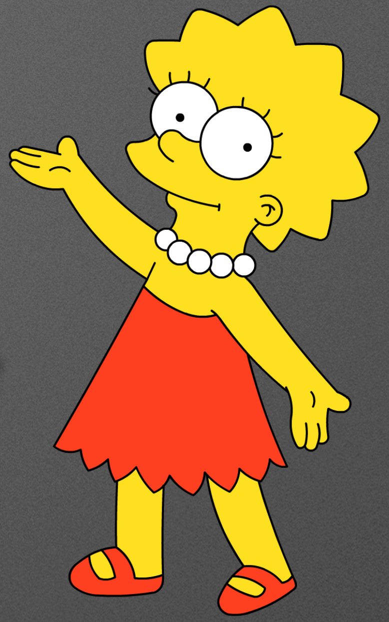
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

$\frac{2}{3} + \epsilon$

$\frac{1}{3} - \epsilon$

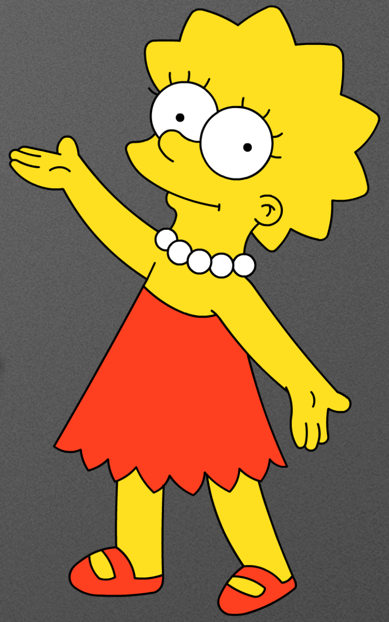




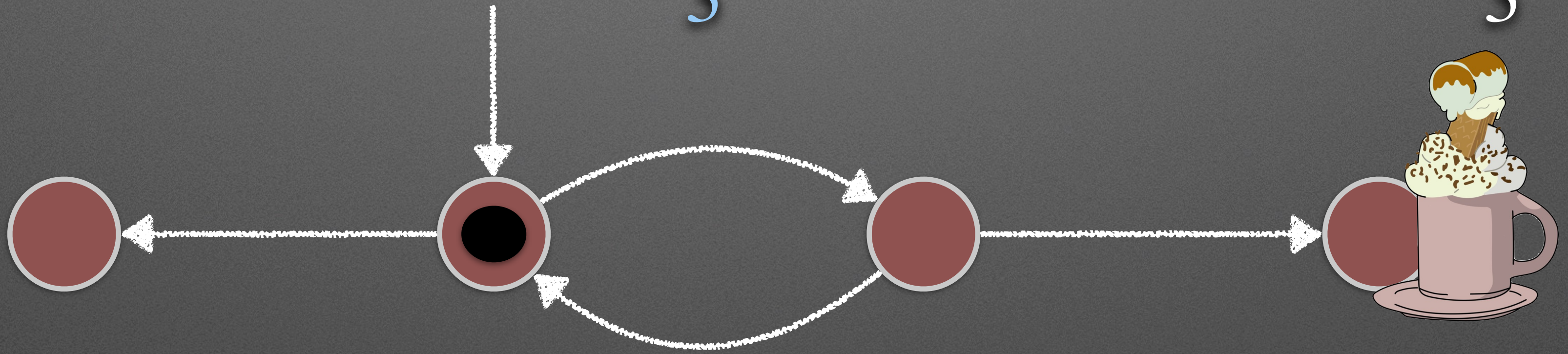
Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?



$\frac{2}{3} + \epsilon$ $\frac{1}{3}$ $\frac{1}{3} - \epsilon$

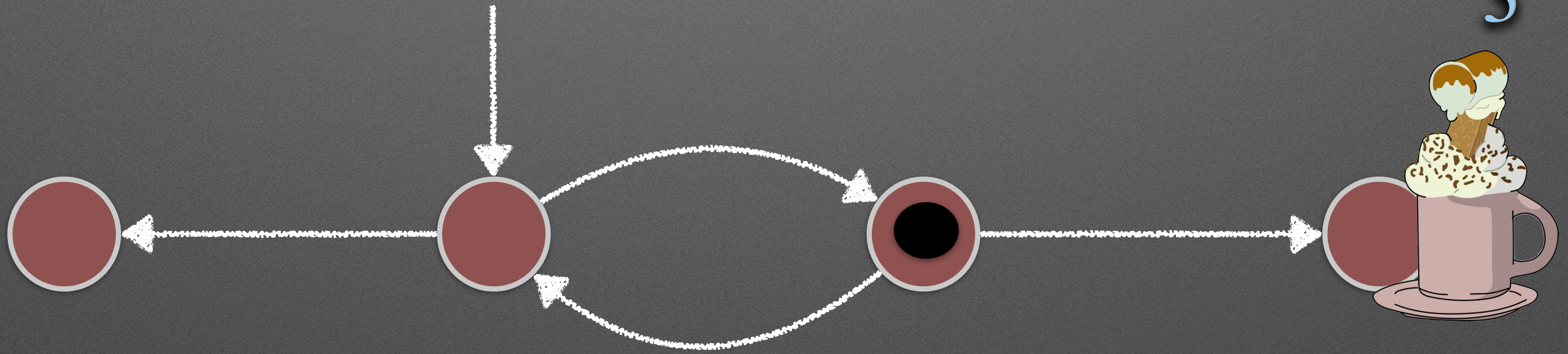


Total budget is normalised to 1

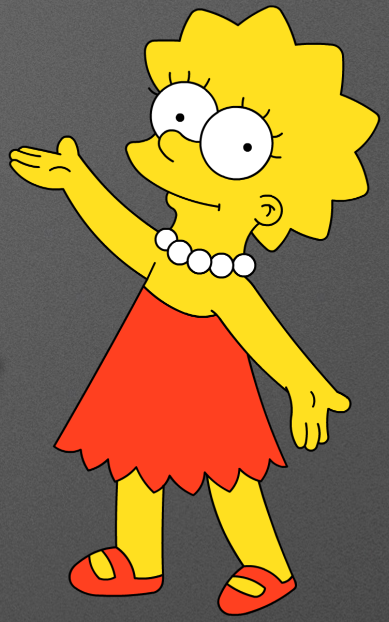
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

$\frac{2}{3} + \epsilon$



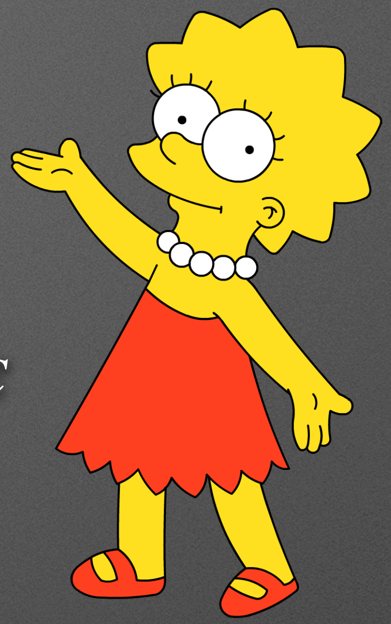
$\frac{1}{3} - \epsilon$



Total budget is normalised to 1

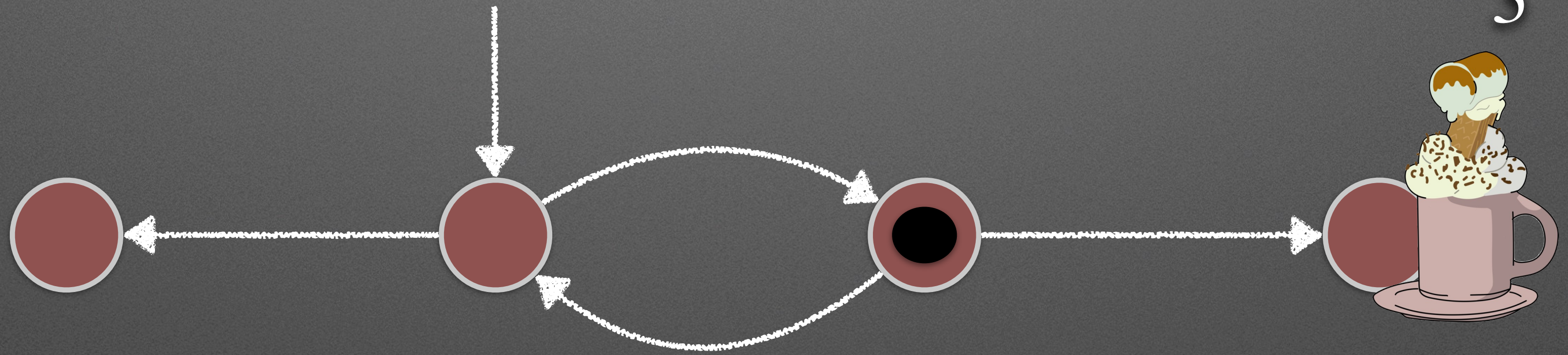
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?



$\frac{1}{3} + \epsilon$

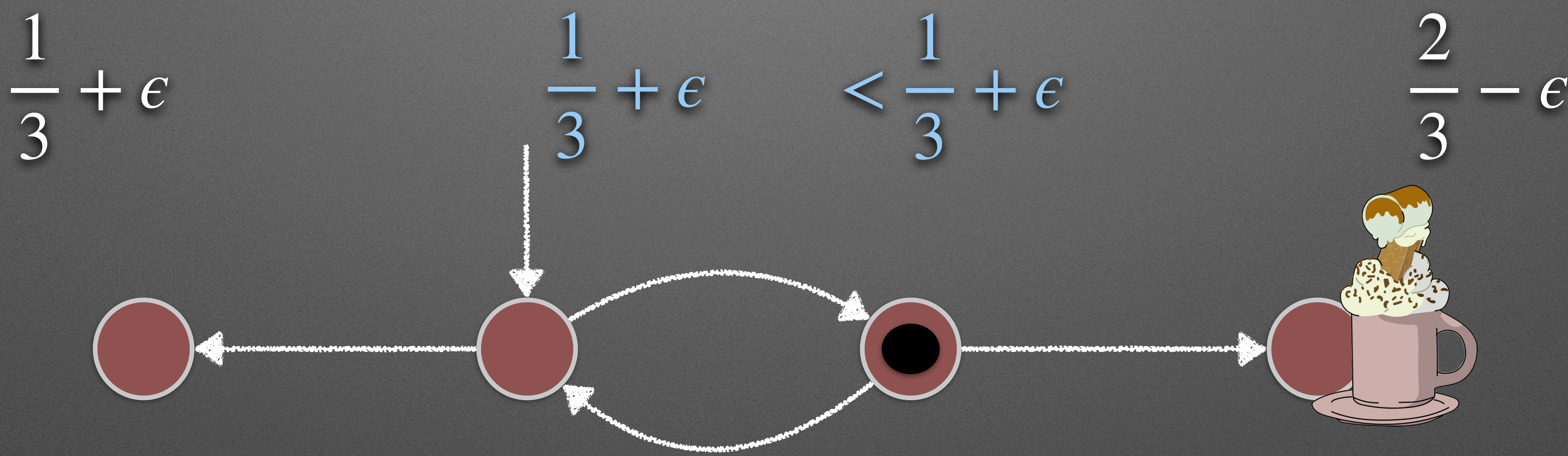
$\frac{2}{3} - \epsilon$



Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

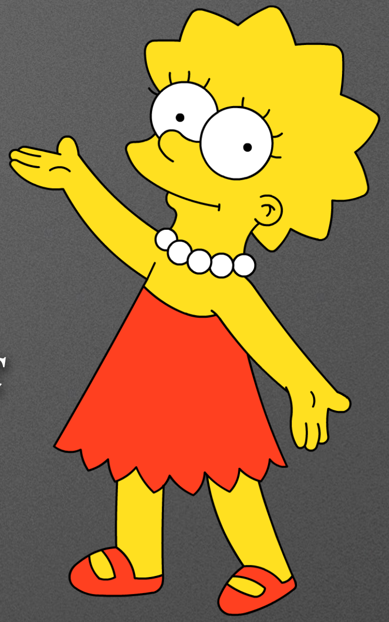


$\frac{1}{3}+\epsilon$

$\frac{1}{3}+\epsilon$

$< \frac{1}{3}+\epsilon$

$\frac{2}{3}-\epsilon$



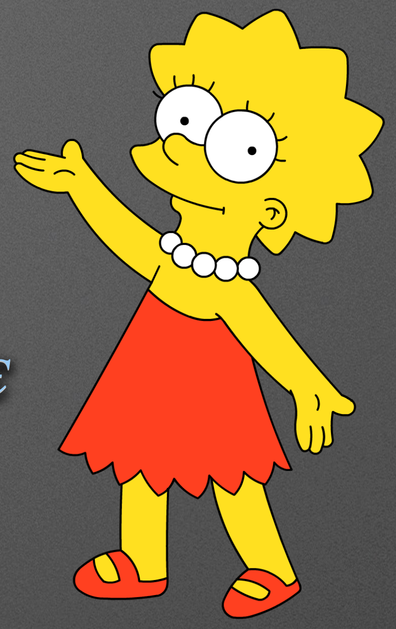
Total budget is normalised to 1

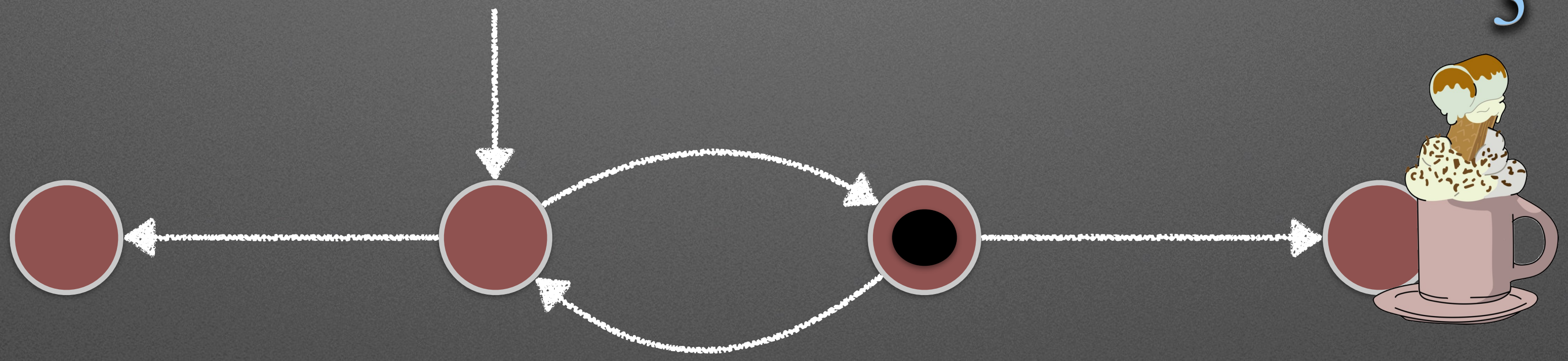
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

$\frac{1}{3}+\epsilon$

$\frac{2}{3}+\epsilon$

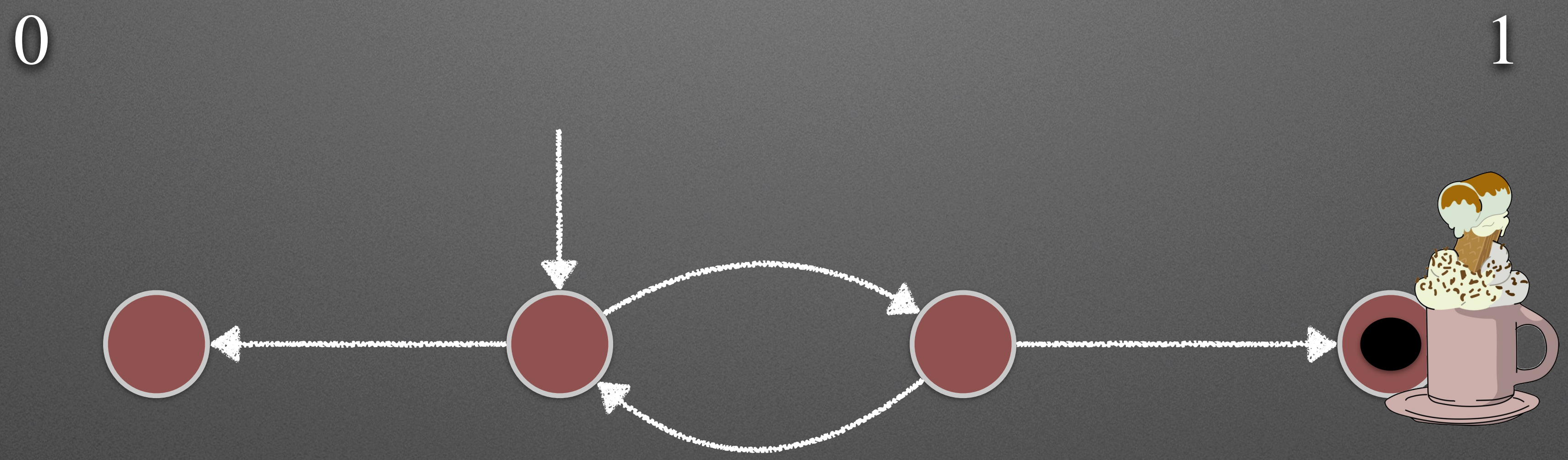




Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?



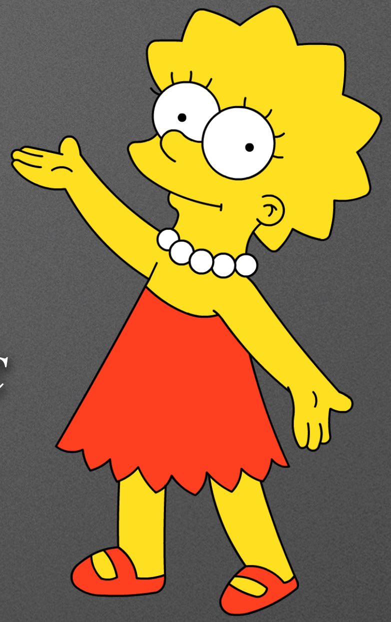
Total budget is normalised to 1

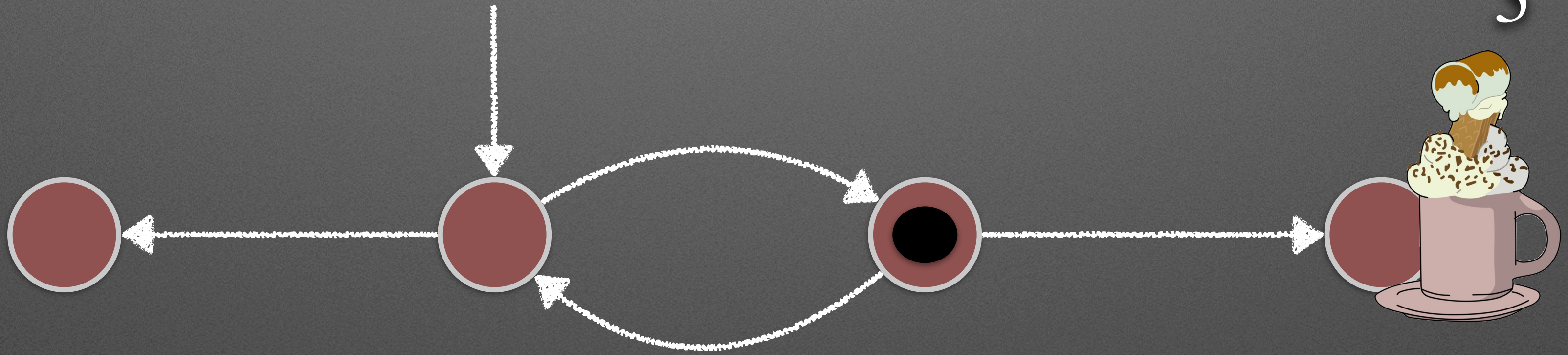
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman ’96,’99]

Can he do any better?

$\frac{1}{3} + \epsilon$

$\frac{2}{3} - \epsilon$

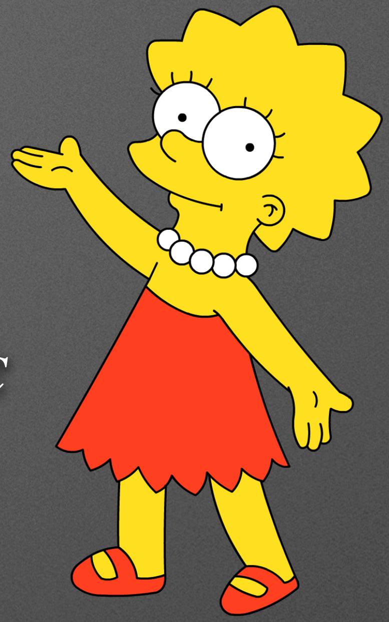




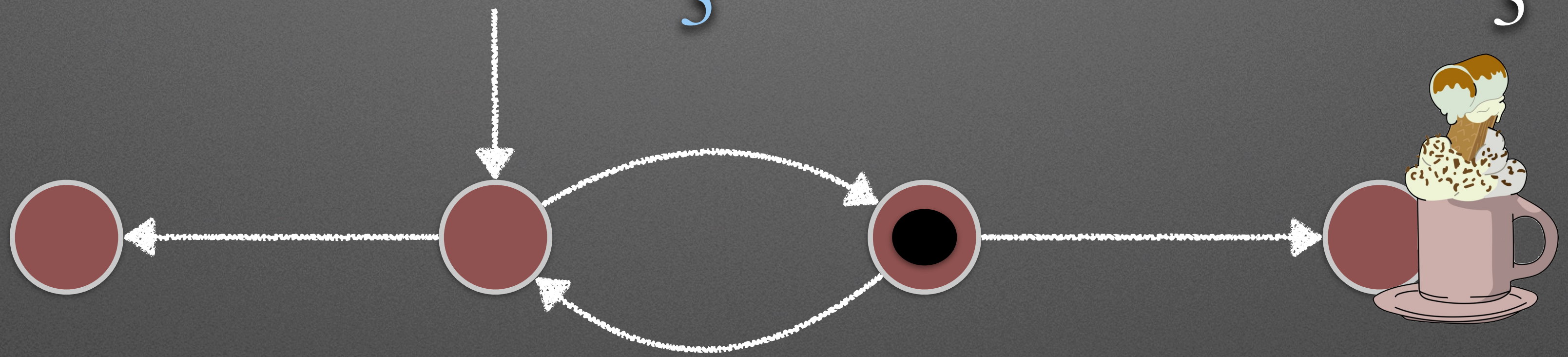
Total budget is normalised to 1

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?



$\frac{1}{3} + \epsilon$ $\quad > \frac{1}{3} + \epsilon$ $\quad \frac{2}{3} - \epsilon$



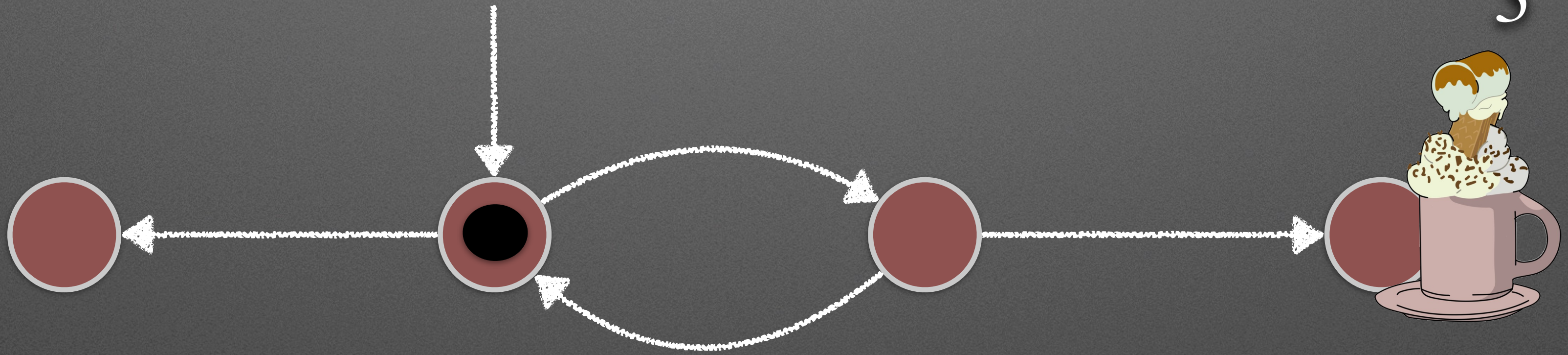
Total budget is normalised to 1

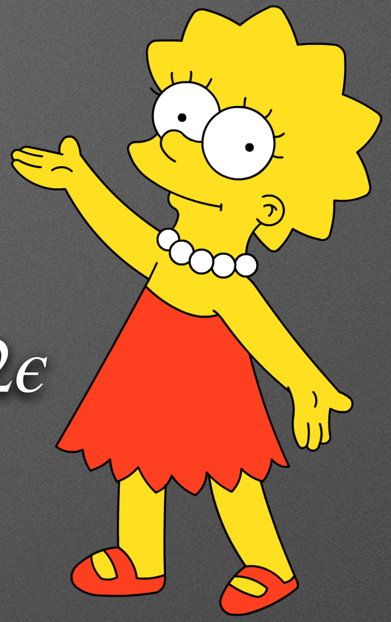
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

$\frac{2}{3}+2\epsilon$

$\frac{1}{3}-2\epsilon$





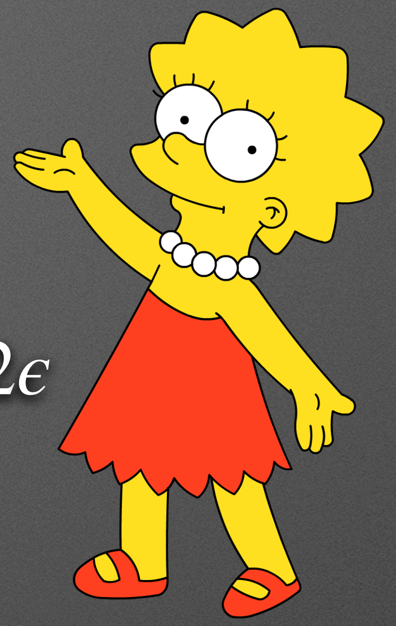
Total budget is normalised to 1

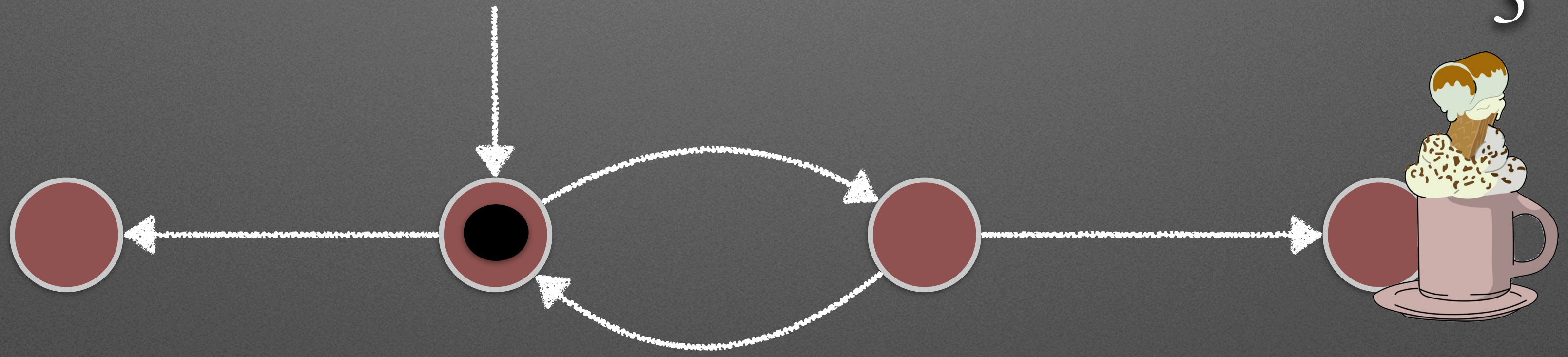
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

$\frac{2}{3} + \boxed{2\epsilon}$

$\frac{1}{3} - 2\epsilon$

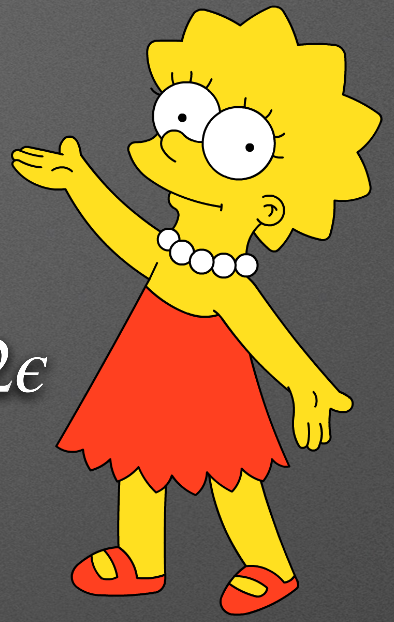




Total budget is normalised to 1

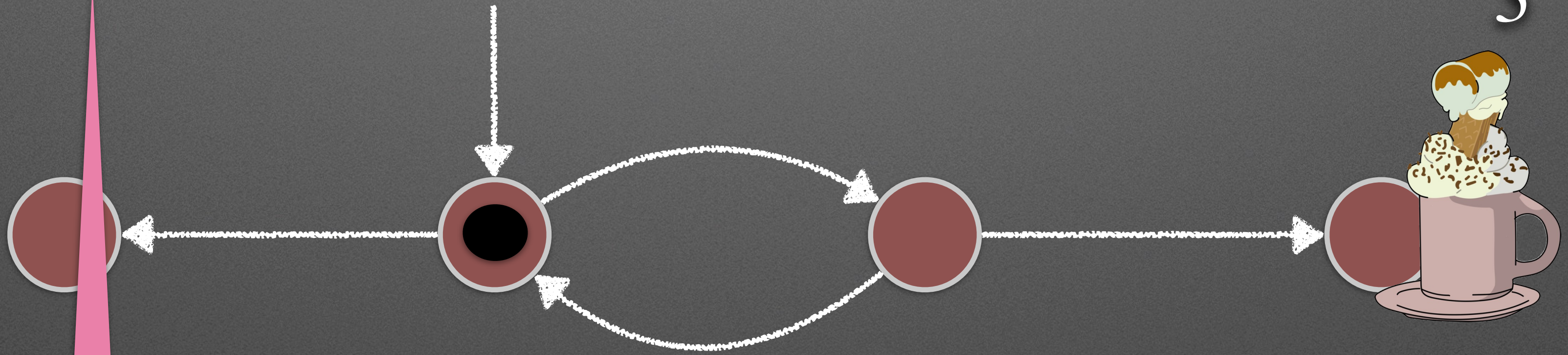
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?



$\frac{2}{3} + \boxed{2\epsilon}$

$\frac{1}{3} - 2\epsilon$



Total budget is normalised to 1

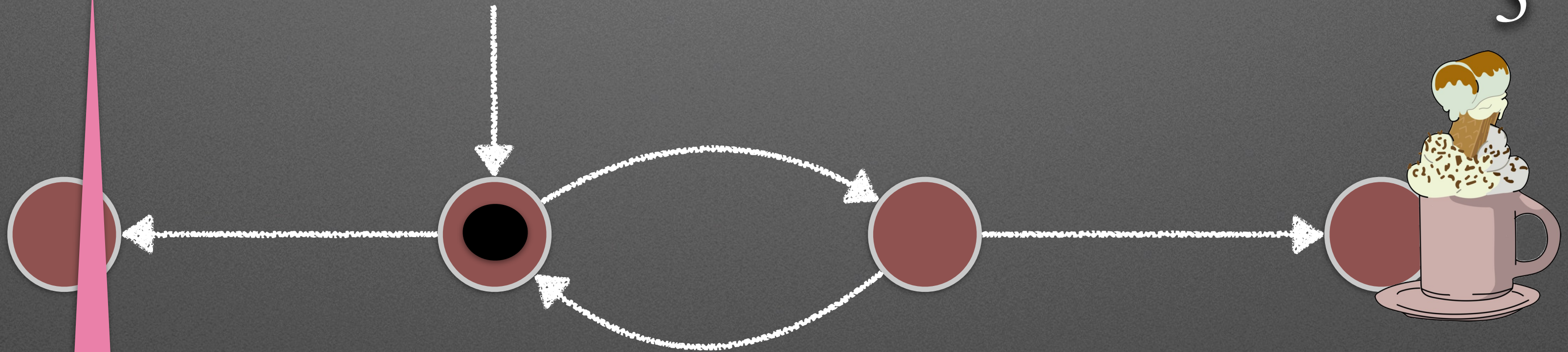
Continue untill he has budget > 0.75, then he wins

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better?

$\frac{2}{3} + \boxed{2\epsilon}$

$\frac{1}{3} - 2\epsilon$

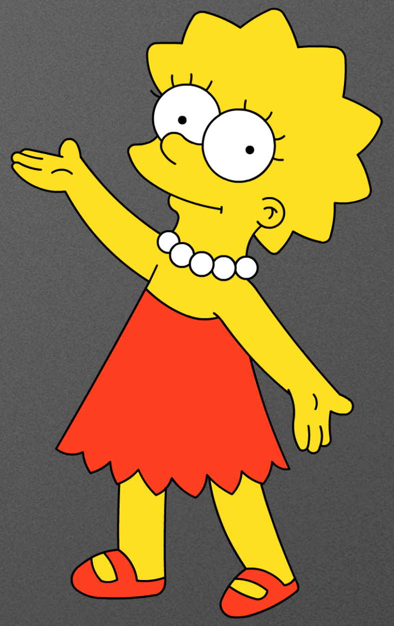


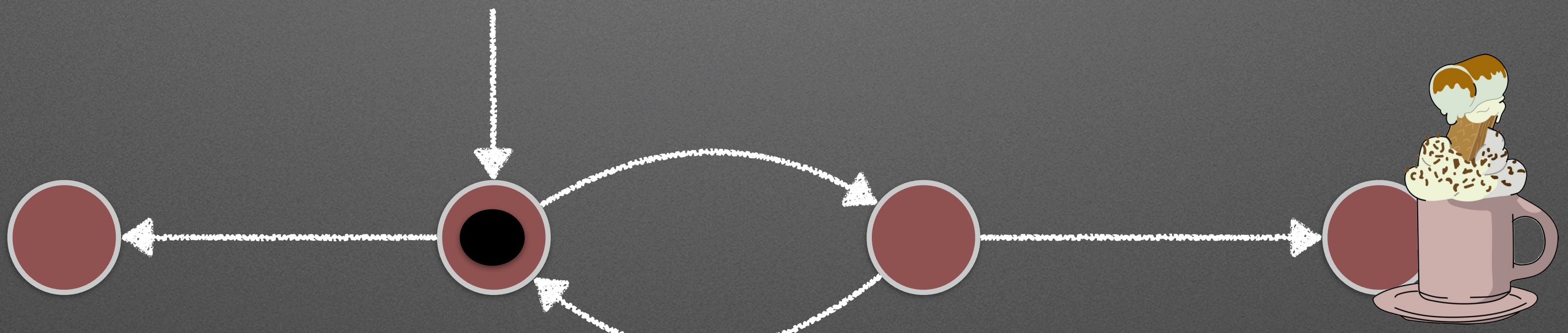
Total budget is normalised to 1

Continue untill he has budget > 0.75, then he wins

Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better than $\frac{2}{3}$?

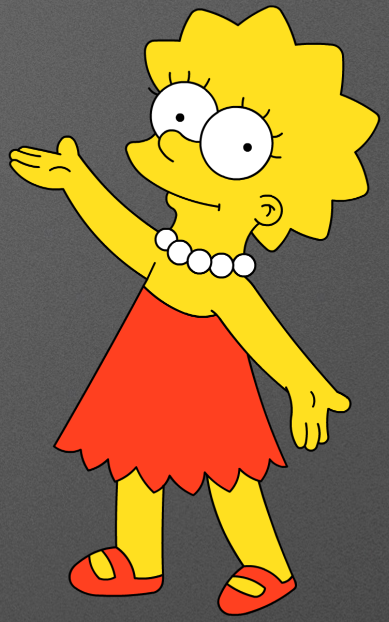


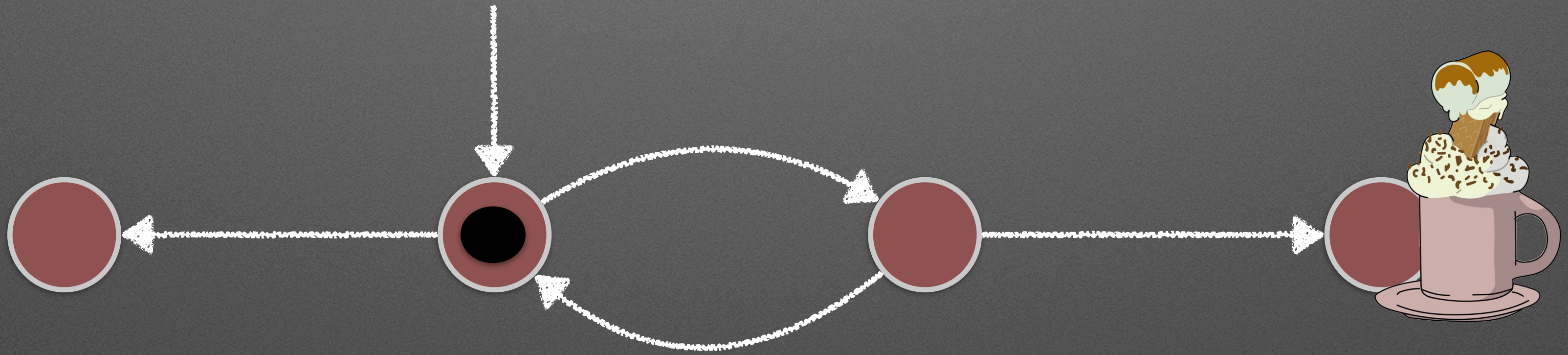


Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better than $\frac{2}{3}$?

The answer is No!





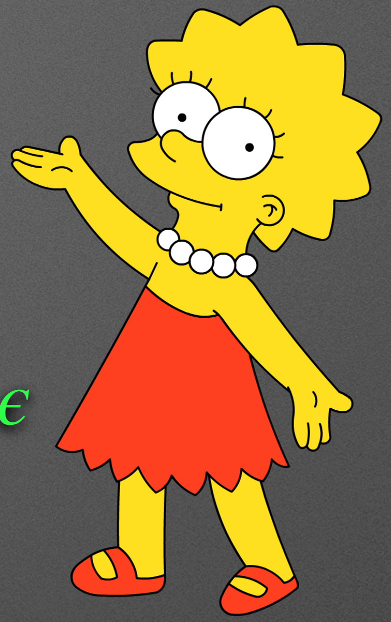
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

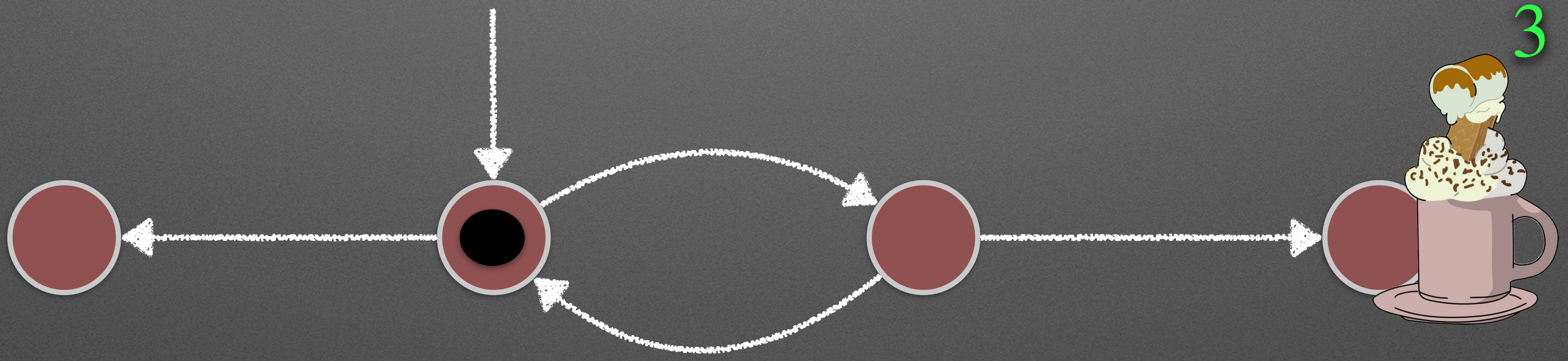
$\frac{2}{3} - \epsilon$

Can he do any better than $\frac{2}{3}$?

The answer is No!

$\frac{1}{3} + \epsilon$



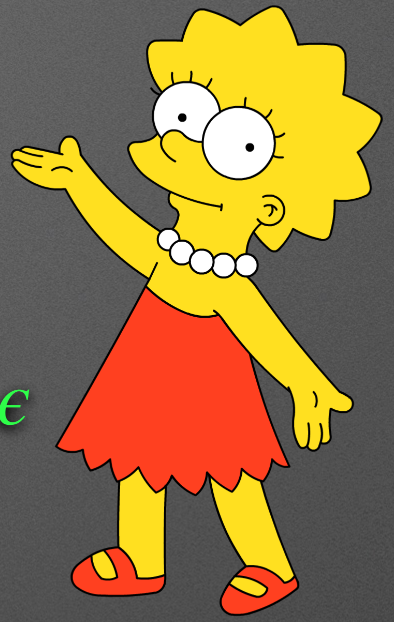


Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

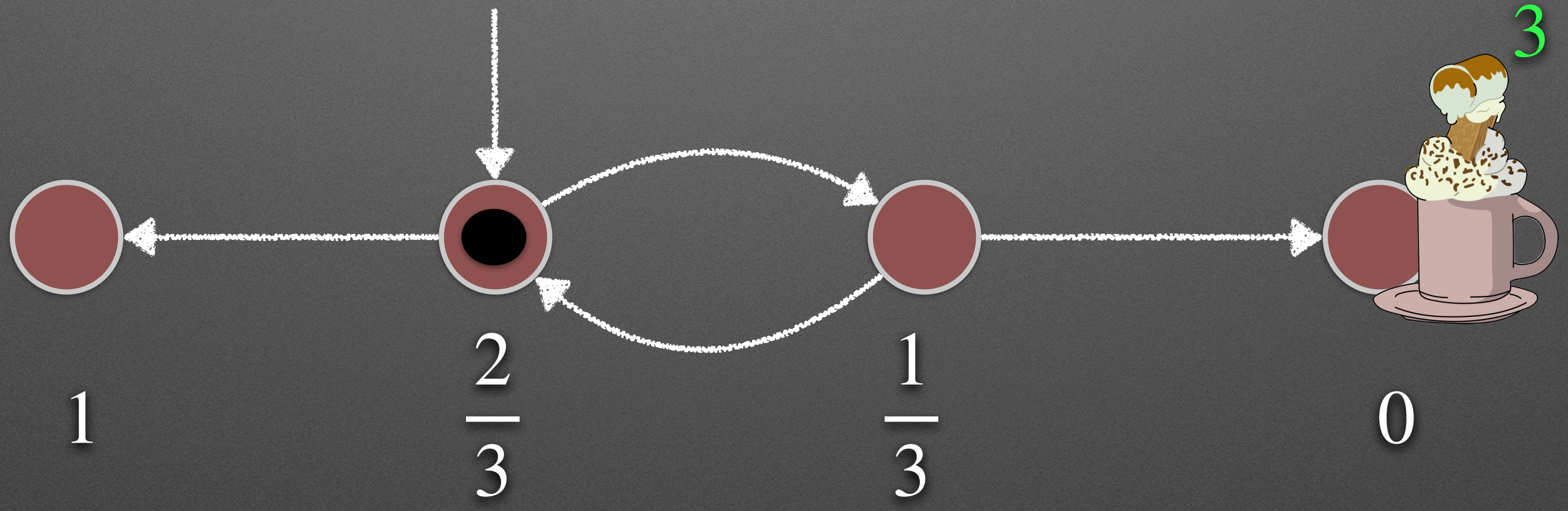
Can he do any better than $\frac{2}{3}$?

The answer is No!

$\frac{2}{3} - \epsilon$



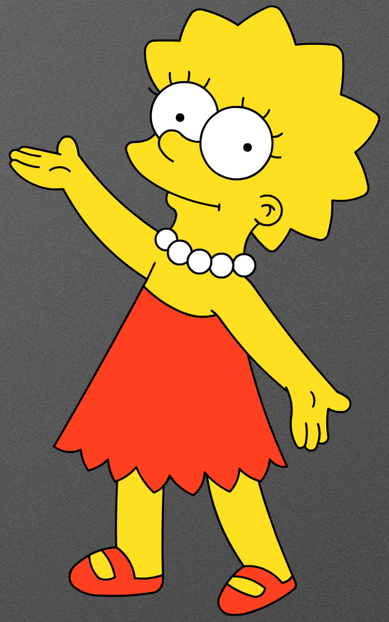
$\frac{1}{3} + \epsilon$

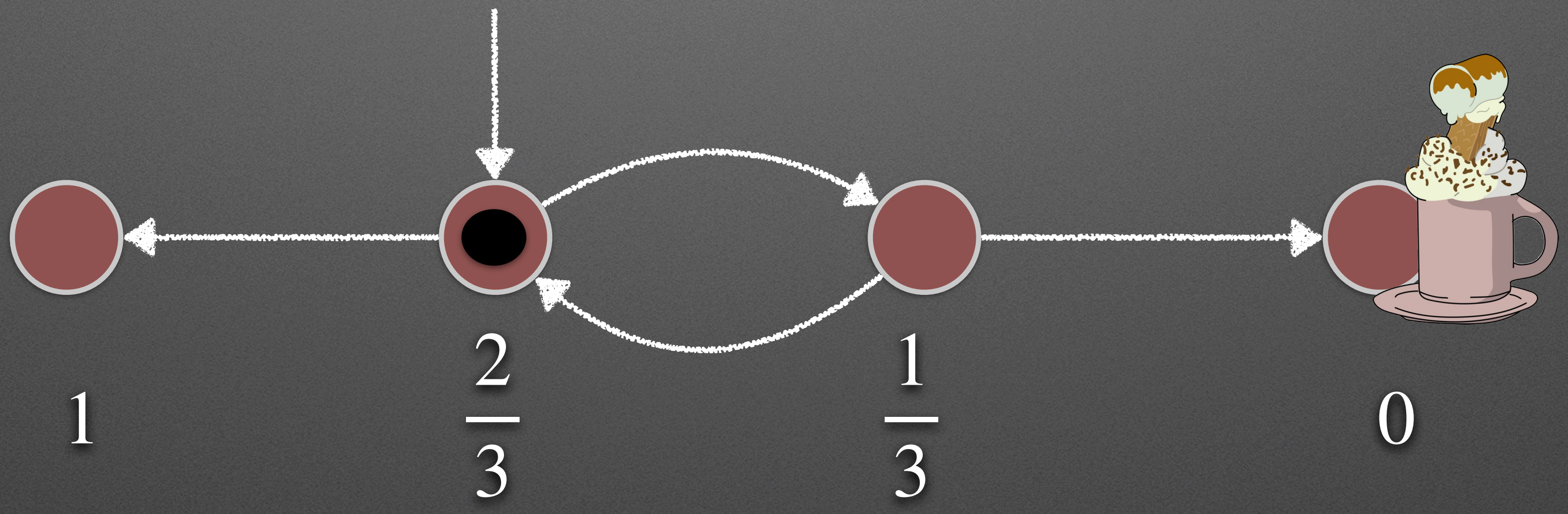


Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

Can he do any better than $\frac{2}{3}$?

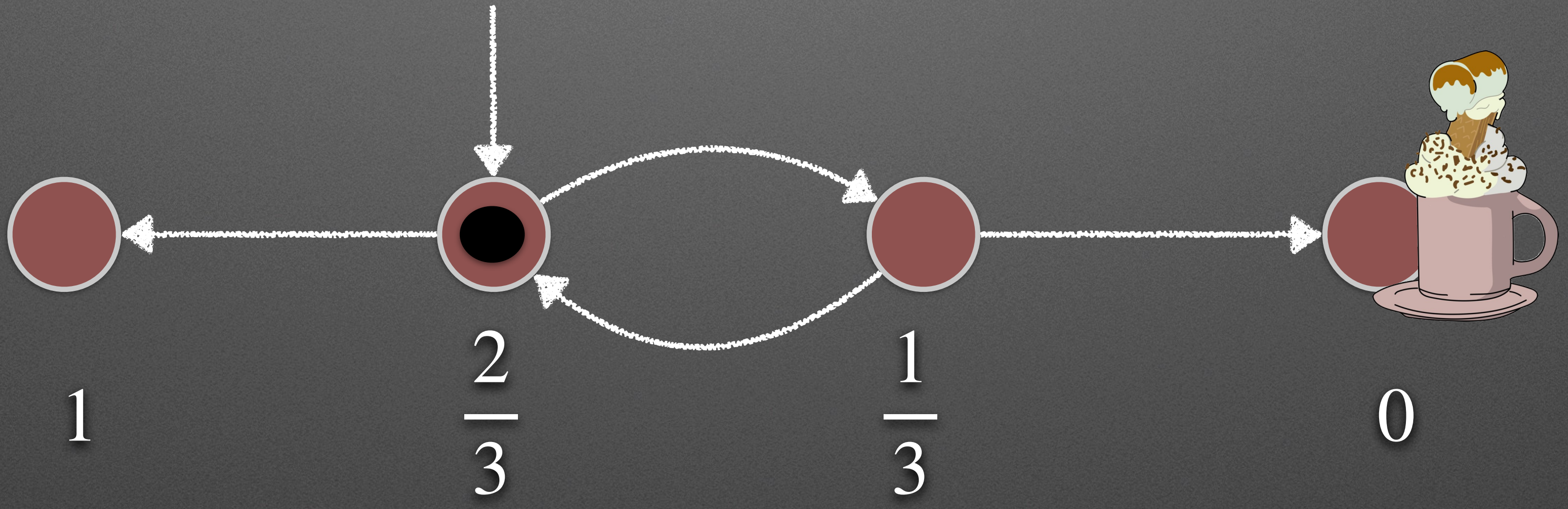
The answer is No!





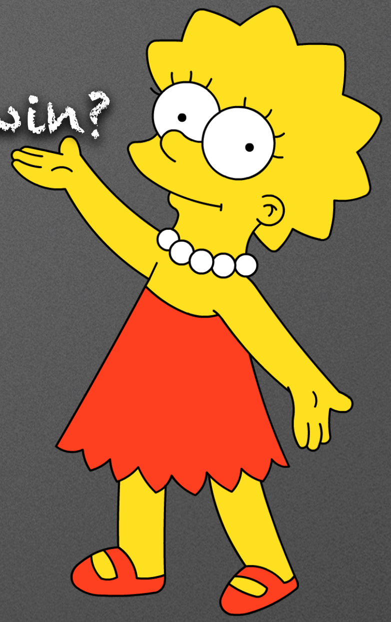
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?



Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

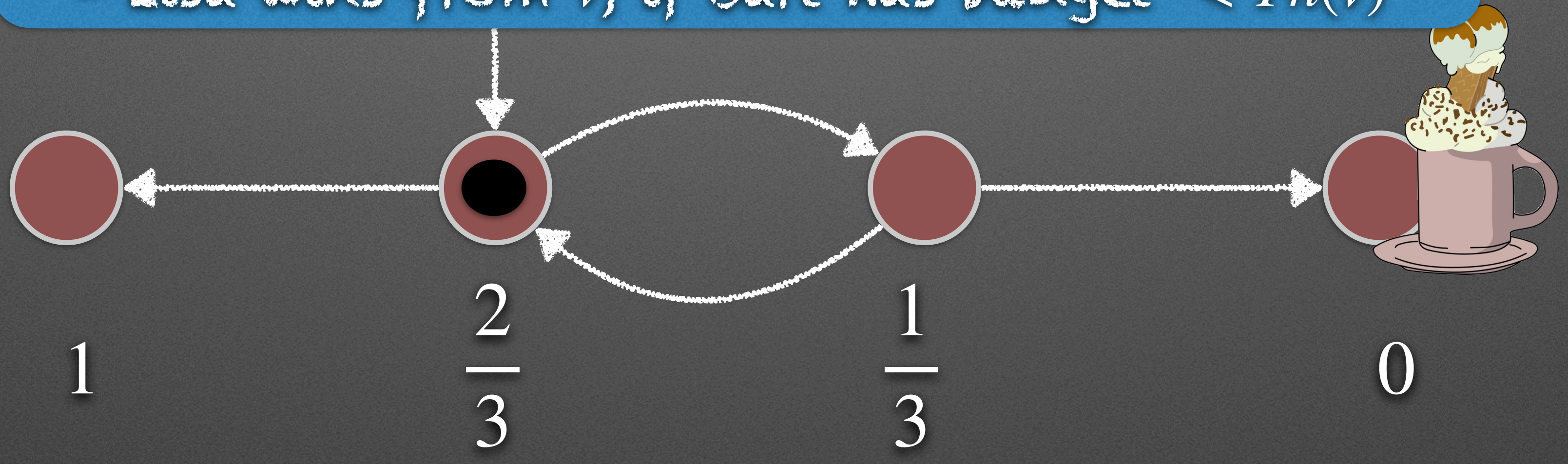
How much initial budget is necessary & sufficient for Bart to win?



Theorem:

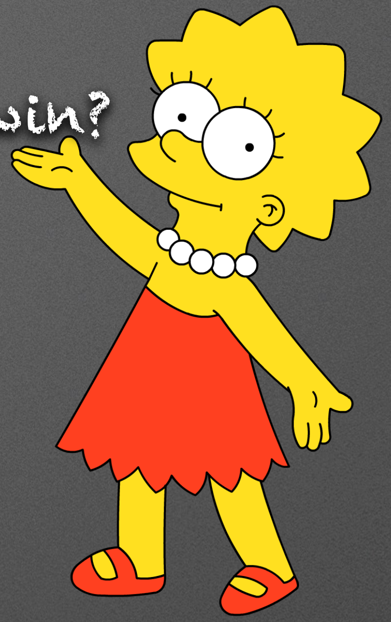
Bidding games are determined.

- Bart wins from v, if he has budget $> Th(v)$
- Lisa wins from v, if Bart has budget $< Th(v)$



Reachability first-price Richman continuous [Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

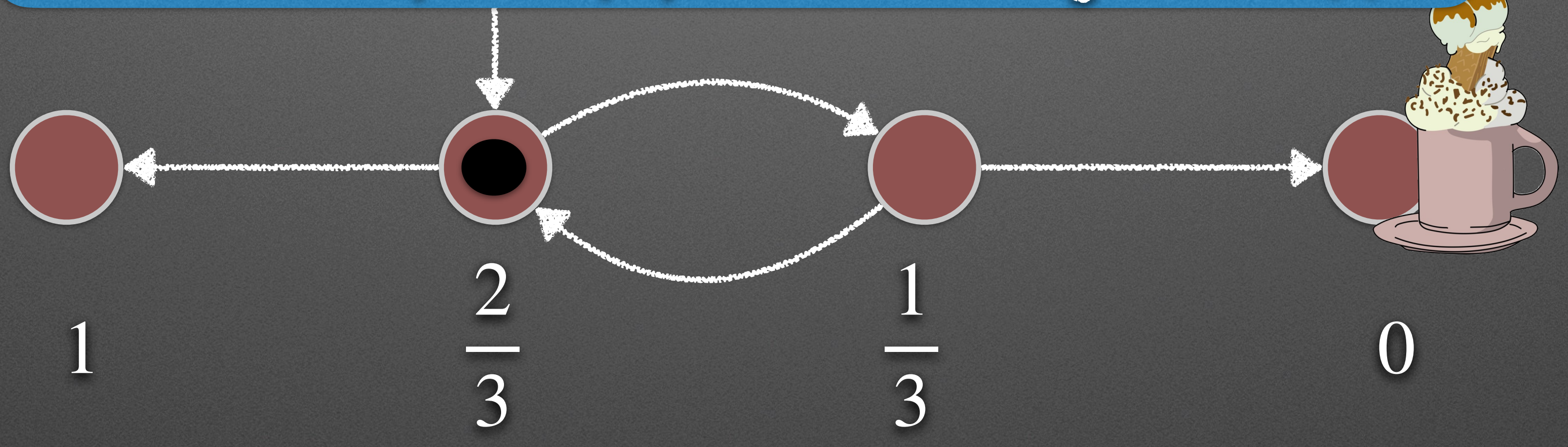
How much initial budget is necessary & sufficient for Bart to win?



Theorem:

Bidding games are determined.

- Bart wins from v, if he has budget $> Th(v)$
- Lisa wins from v, if Bart has budget $< Th(v)$



Theorem [LLPU96, LLPSU99]:

(1) Threshold budgets exist

(2) They satisfy an average property

(3) Optimal bids can be derived from the threshold budgets

(4) In NP ∩ co-NP via a (simple) reduction to stochastic games

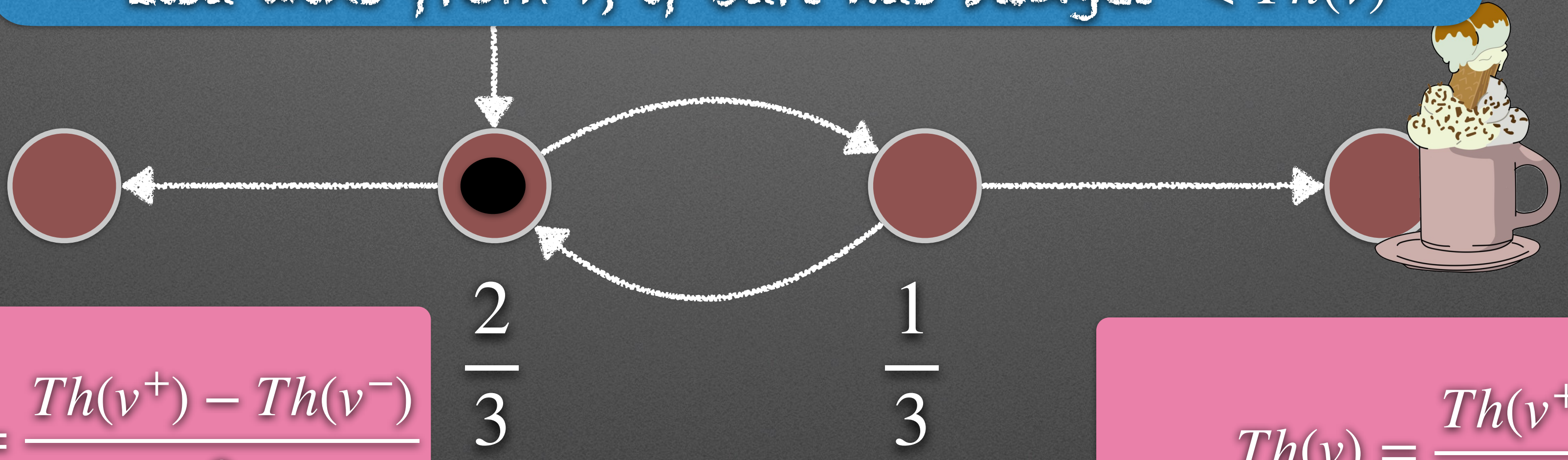
Reachability first-price Richman continuous
[Lazarus, Loeb, Propp, Stromquist, Ullman '96,'99]

How much initial budget is necessary & sufficient for Bart to win?

Theorem:

Bidding games are determined.

- Bart wins from v, if he has budget $> Th(v)$
- Lisa wins from v, if Bart has budget $< Th(v)$



Optimal bids: $b(v) = \dfrac{Th(v^+) - Th(v^-)}{2}$

$$Th(v) = \frac{Th(v^+) + Th(v^-)}{2}$$

Theorem [LLPU96, LLPSU99]:

(1) Threshold budgets exist

(2) They satisfy an average property

(3) Optimal bids can be derived from the threshold budgets

(4) In NP ∩ co-NP via a (simple) reduction to stochastic games

P.S. v^+ and v^- are the max/min neighbours wrt Th()

Part I (in Theory): Discrete Bidding Games

Reachability first-price Richman discrete

[Develin & Payne. 2009]

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E\rangle$

where $k \in \mathbb{N}$ is the total budget

A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

Discrete

At each turn, Player i bids an integer $b_i \leq B_i$

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Tie breaking (when $b_1 = b_2$)

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E\rangle$

where $k \in \mathbb{N}$ is the total budget

A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Tie breaking (when $b_1 = b_2$)

8

10

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E\rangle$

where $k \in \mathbb{N}$ is the total budget

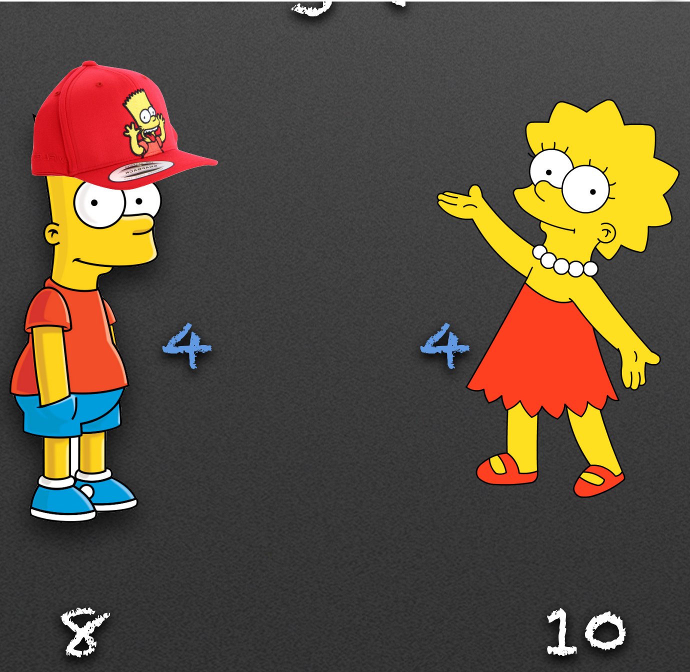
A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Tie breaking (when $b_1 = b_2$)



Reachability first-price Richman discrete

[Develin & Payne, 2009]

A bidding games is played on an arena: $\langle k, V, E\rangle$

where $k \in \mathbb{N}$ is the total budget

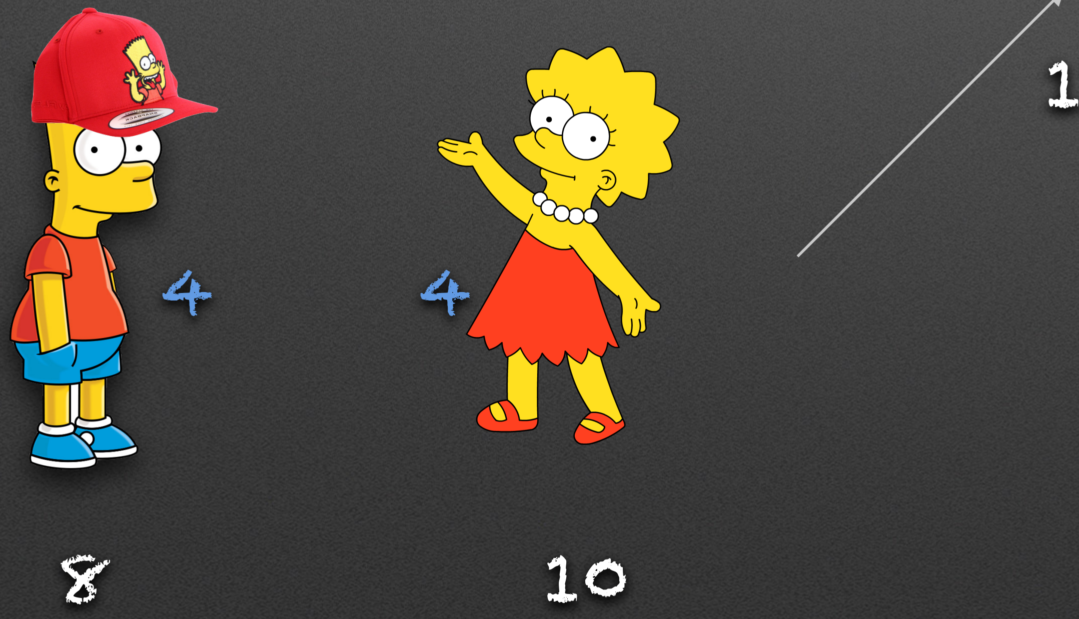
A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

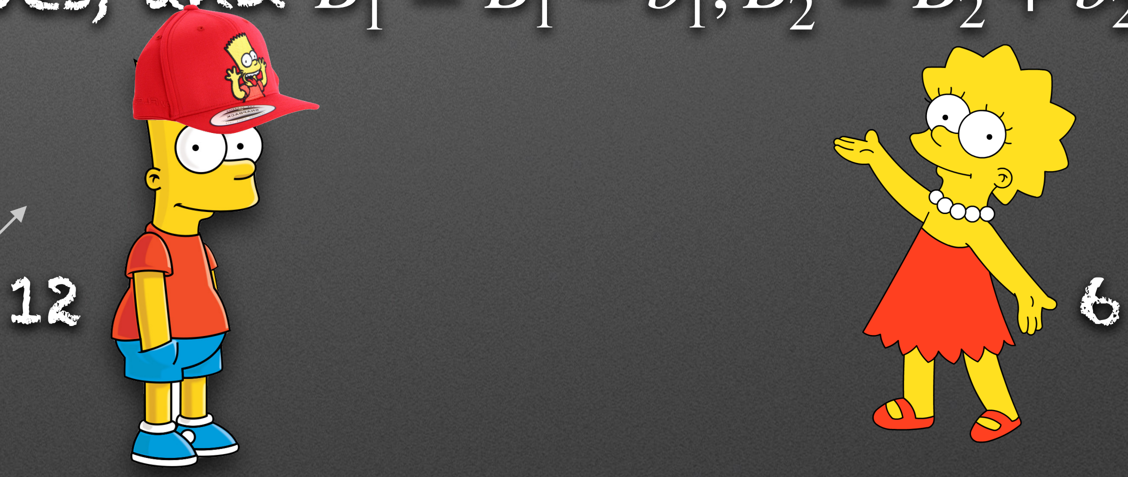
At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Tie breaking (when $b_1 = b_2$)





Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E\rangle$

where $k \in \mathbb{N}$ is the total budget

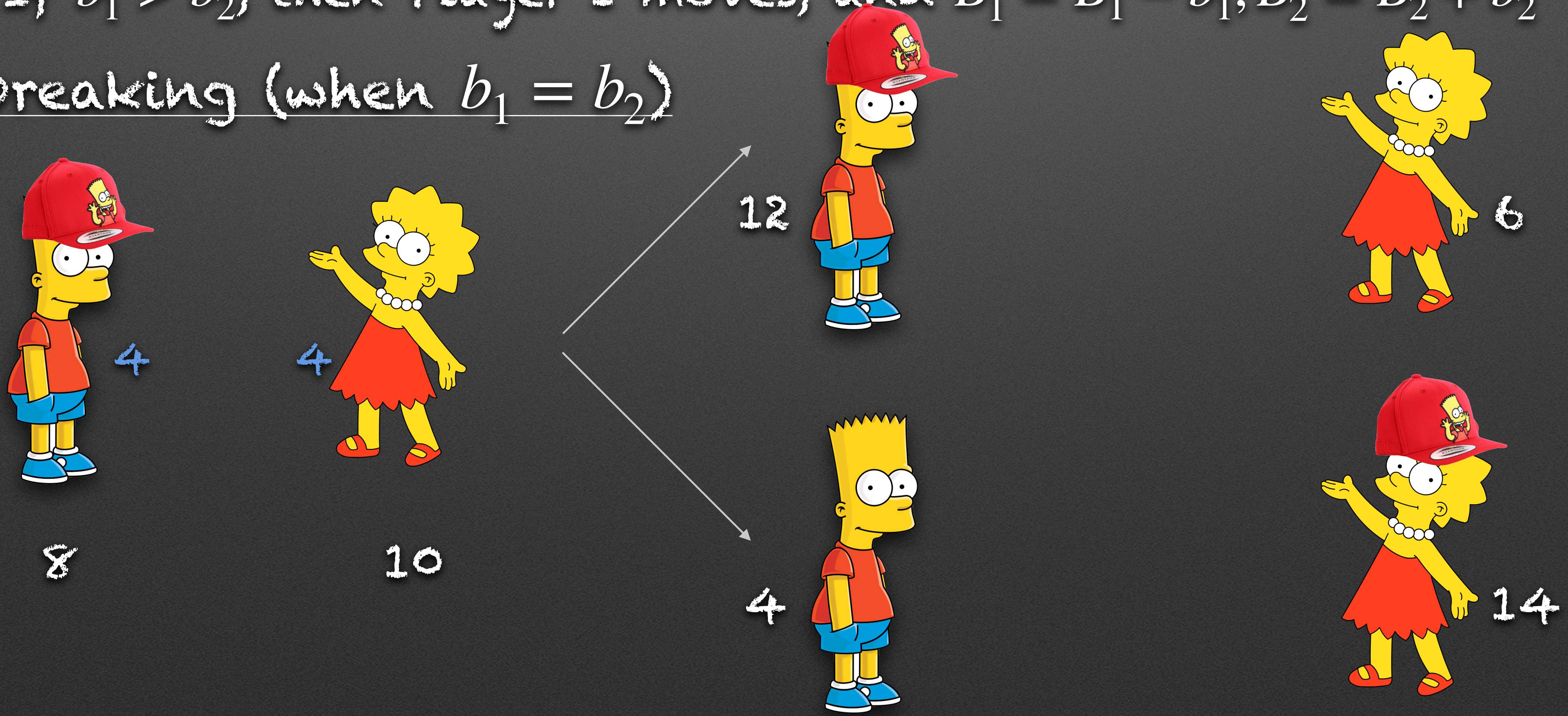
A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Tie breaking (when $b_1 = b_2$)



Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

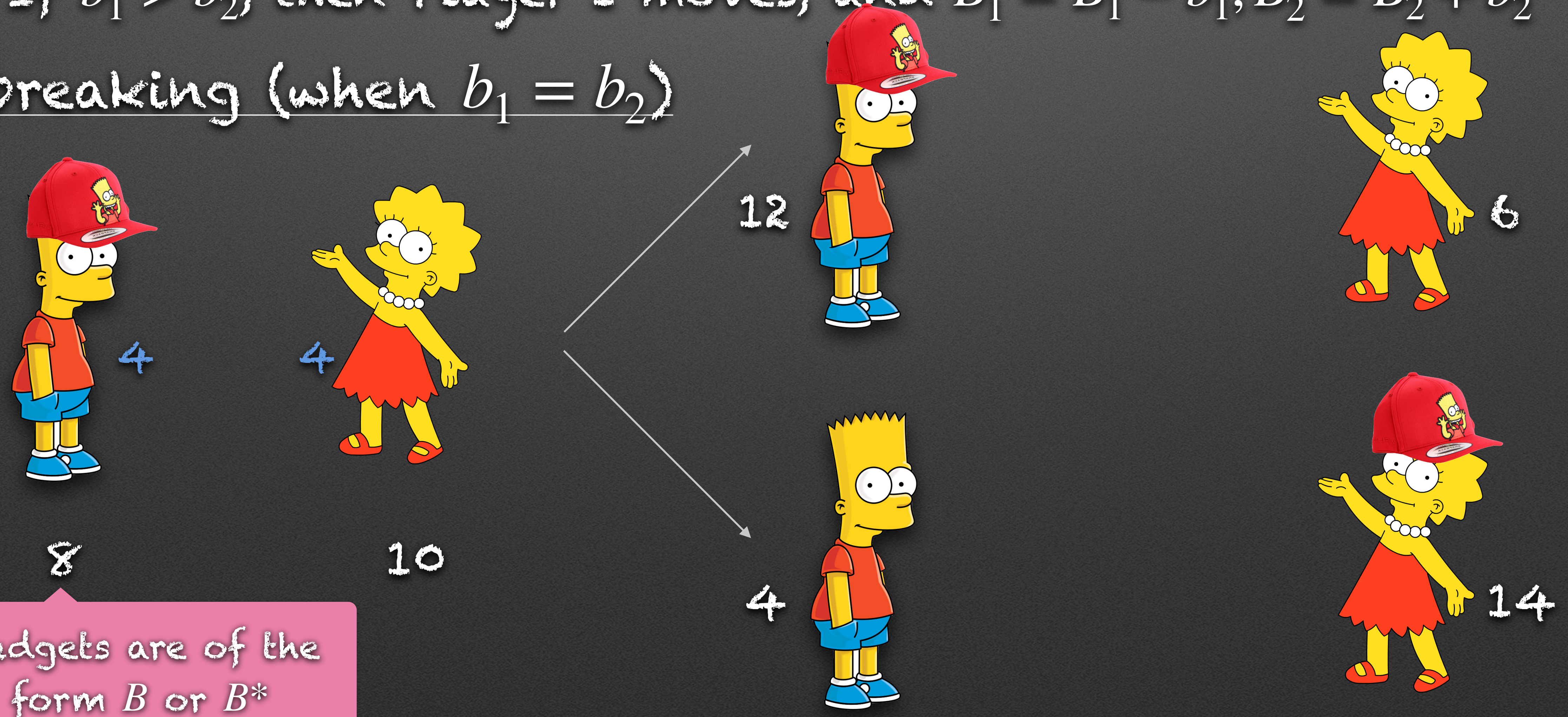
A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Tie breaking (when $b_1 = b_2$)



Budgets are of the form B or B^*

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

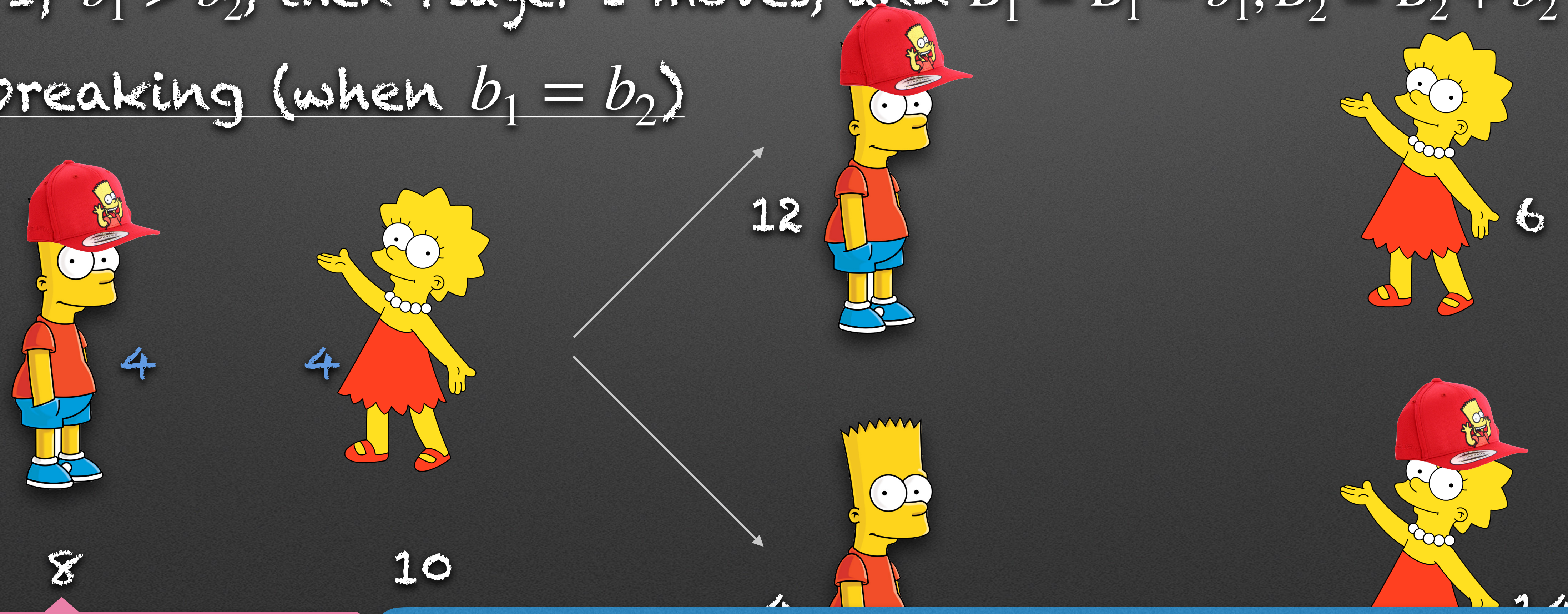
A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

Discrete

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Tie breaking (when $b_1 = b_2$)



Budgets are of the form B or B^*

The possible budgets are as follows:
$0 < 0^* < 1 < 1^* \ldots k < k^* < k + 1$

Reachability first-price Richman discrete

[Develin & Payne. 2009]

A bidding games is played on an arena: $\mathscr{A} = \langle k, V, E \rangle$

where $k \in \mathbb{N}$ is the total budget

A configuration is (v, B_1, B_2) with $B_1 + B_2 = k$

At each turn, Player i bids an integer $b_i \leq B_i$

If $b_1 > b_2$, then Player 1 moves, and $B_1' = B_1 - b_1, B_2' = B_2 + b_2$

Decision Problem:

Input: A game graph $\mathscr{G}$, total budget k (encoded in binary), Winning condition $\mathscr{W}$, and initial configuration $\langle v, B_1, k \ominus B_1 \rangle$, where $B_1 =$ Bart's initial budget

Output: Yes iff Bart wins the game from v with budget B_1

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Theorem[LLPU '96, LLPSU '99]:

(1) Threshold budgets exist

(2) Threshold budgets satisfy average property

(3) Bids are derived from the thresholds

(4) Computing threshold budgets is in NP ∩ coNP

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Theorem[LLPU '96, LLPSU '99]:

(1) Threshold budgets exist

(2) Threshold budgets satisfy average property

(3) Bids are derived from the thresholds

(4) Computing threshold budgets is in NP ∩ coNP

Theorem[Develin & Payne '09]:

A discrete version of (1)-(3) holds

EXPTIME Value iteration algorithm for computing thresholds.

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Theorem[LLPU '96, LLPSU '99]:

(1) Threshold budgets exist

(2) Threshold budgets satisfy average property

(3) Bids are derived from the thresholds

(4) Computing threshold budgets is in NP ∩ coNP

Theorem[Develin & Payne '09]:

A discrete version of (1)-(3) holds

EXPTIME Value iteration algorithm for computing thresholds.

Buchi

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Theorem[LLPU '96, LLPSU '99]:

(1) Threshold budgets exist
(2) Threshold budgets satisfy average property
(3) Bids are derived from the thresholds
(4) Computing threshold budgets is in NP ∩ coNP

Theorem[Develin & Payne '09]:

A discrete version of (1)-(3) holds

EXPTIME Value iteration algorithm for computing thresholds.

Buchi

Theorem[Avni, Henzinger, Chonev '19]:
(Easily) Reduce to reachability games

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Theorem[LLPU '96, LLPSU '99]:

(1) Threshold budgets exist
(2) Threshold budgets satisfy average property
(3) Bids are derived from the thresholds
(4) Computing threshold budgets is in NP ∩ coNP

Theorem[Develin & Payne '09]:

A discrete version of (1)-(3) holds

EXPTIME Value iteration algorithm for computing thresholds.

Buchi

Theorem[Avni, Henzinger, Chonev '19]:
(Easily) Reduce to reachability games

Theorem[Aghajohari, Avni, Henzinger '21]:
Muller games are determined

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Theorem[LLPU '96, LLPSU '99]:

(1) Threshold budgets exist

(2) Threshold budgets satisfy average property

(3) Bids are derived from the thresholds

(4) Computing threshold budgets is in NP ∩ coNP

Theorem[Develin & Payne '09]:

A discrete version of (1)-(3) holds

EXPTIME Value iteration algorithm for computing thresholds.

Buchi

Theorem[Avni, Henzinger, Chonev '19]:
(Easily) Reduce to reachability games

Theorem[Aghajohari, Avni, Henzinger '21]:
Muller games are determined

No known structure on the threshold budgets

- Do Threshold budgets satisfy the average property?
- Do threshold budgets give rise to bids?
- Best algorithm to compute them is exponential

Continuous vs Discrete Bidding

arbitrary

Reachability

fixed granularity

Theorem[LLPU '96, LLPSU '99]:

(1) Threshold budgets exist

(2) Threshold budgets satisfy average property

(3) Bids are derived from the thresholds

(4) Computing threshold budgets is in NP ∩ coNP

Theorem[Develin & Payne '09]:

A discrete version of (1)-(3) holds

EXPTIME Value iteration algorithm for computing thresholds.

Buchi

Theorem[Avni, Henzinger, Chonev '19]:
(Easily) Reduce to reachability games

Theorem[Aghajohari, Avni, Henzinger '21]:
Muller games are determined

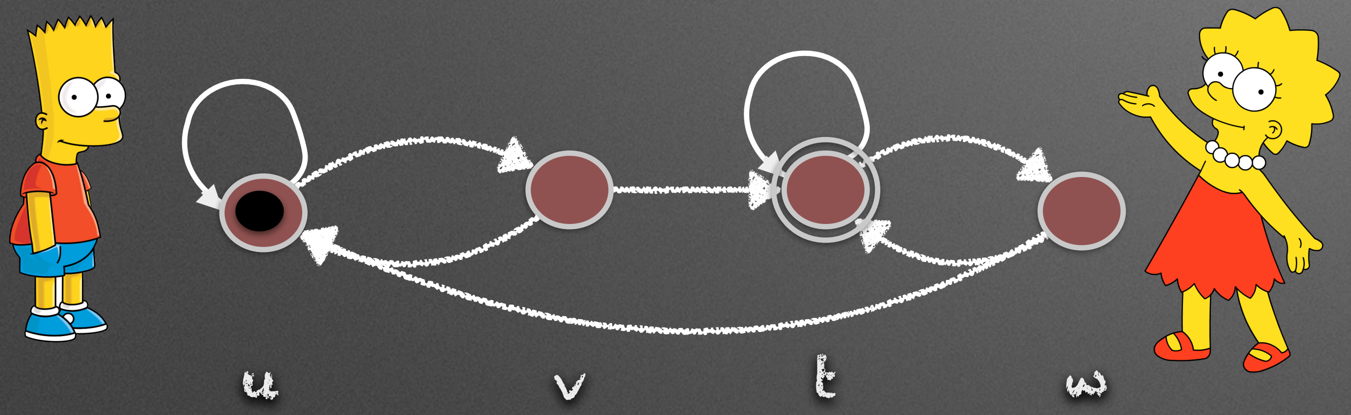
No known structure on the threshold budgets

- Do Threshold budgets satisfy the average property?
- Do threshold budgets give rise to bids?
- Best algorithm to compute them is exponential

Buchi Games

Buchi winning condition:

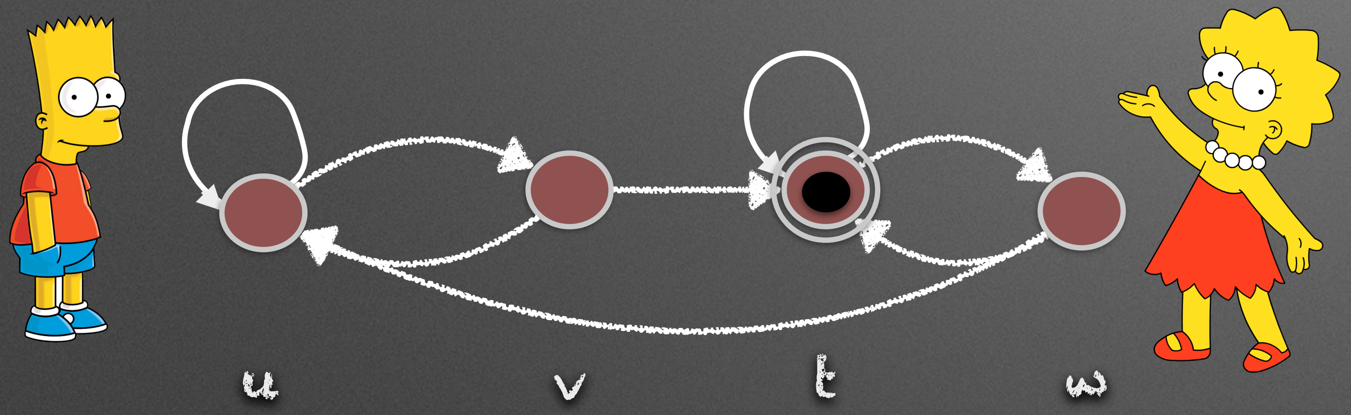
Bart wins iff the set of target vertices T is visited infinitely often



Buchi Games

Buchi winning condition:

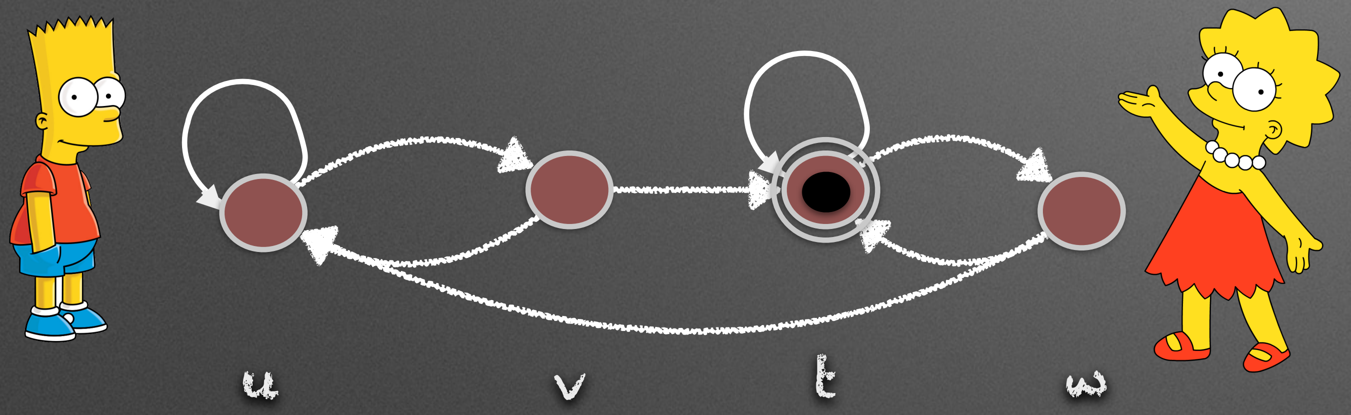
Bart wins iff the set of target vertices T is visited infinitely often



Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

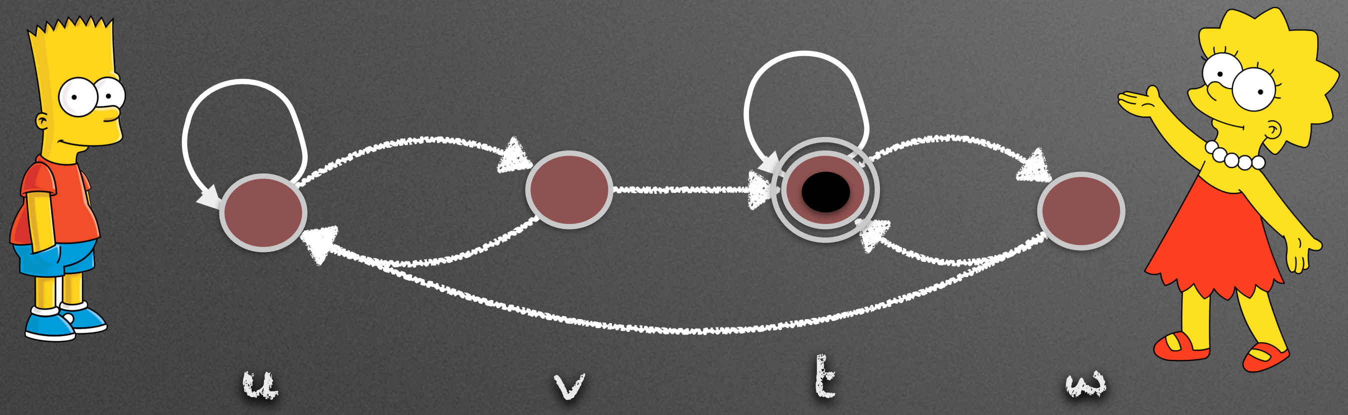
≡

Computing Lisa's coBuchi Threshold

Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

$\Downarrow$

Computing Bart's Buchi Threshold

$\equiv$

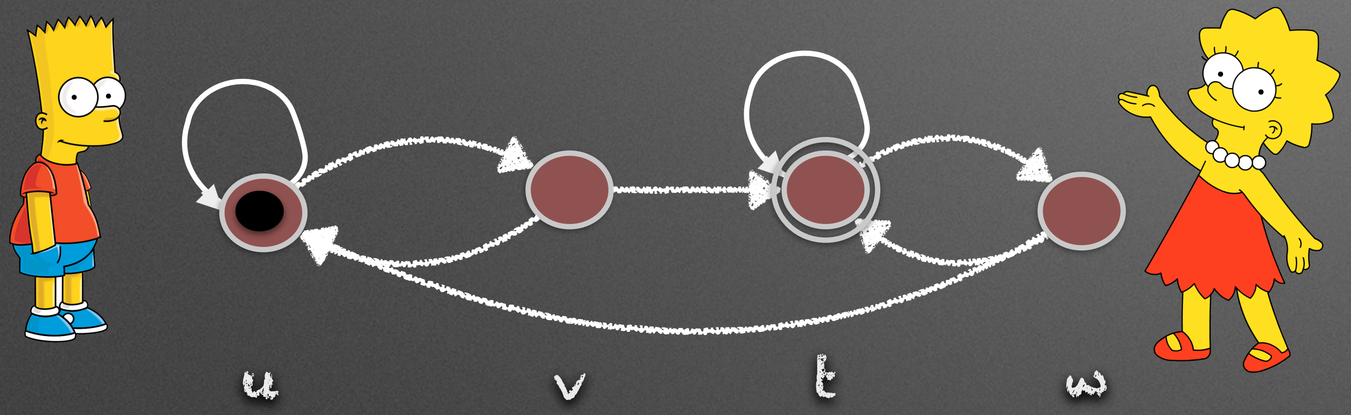
Computing Lisa's coBuchi Threshold

visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

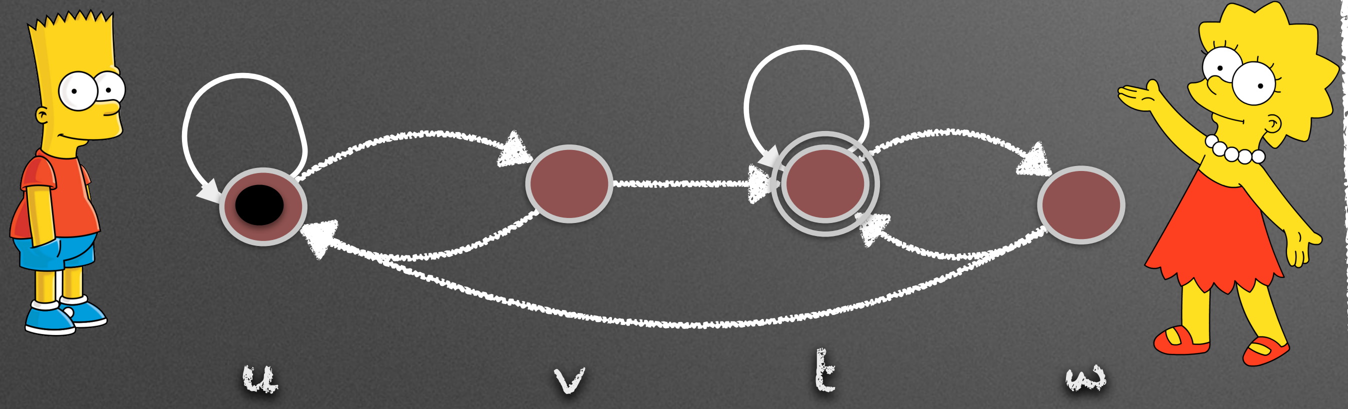
Computing Lisa's coBuchi Threshold

visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

$\Downarrow$

Computing Bart's Buchi Threshold

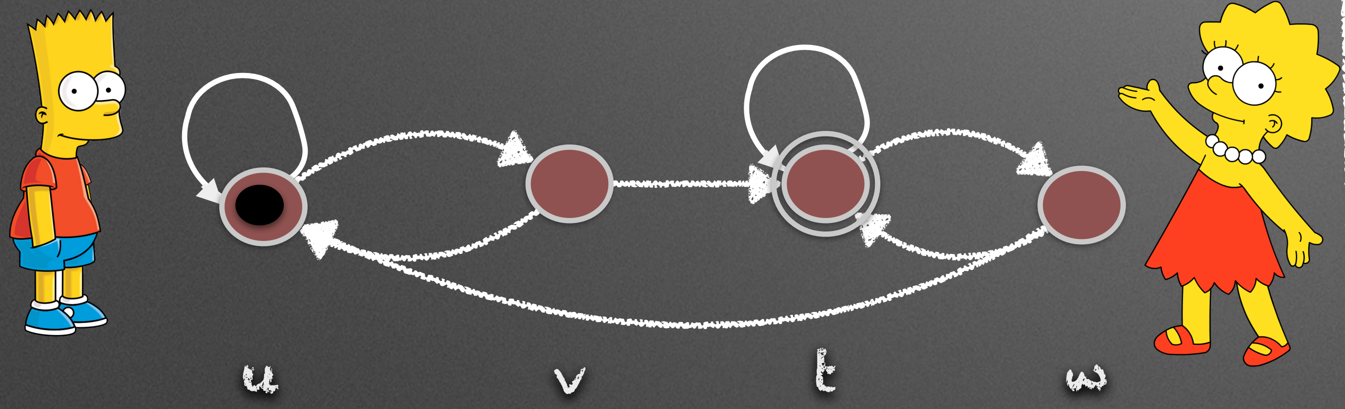
$\equiv$

Computing Lisa's coBuchi Threshold

visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:
Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

⇓

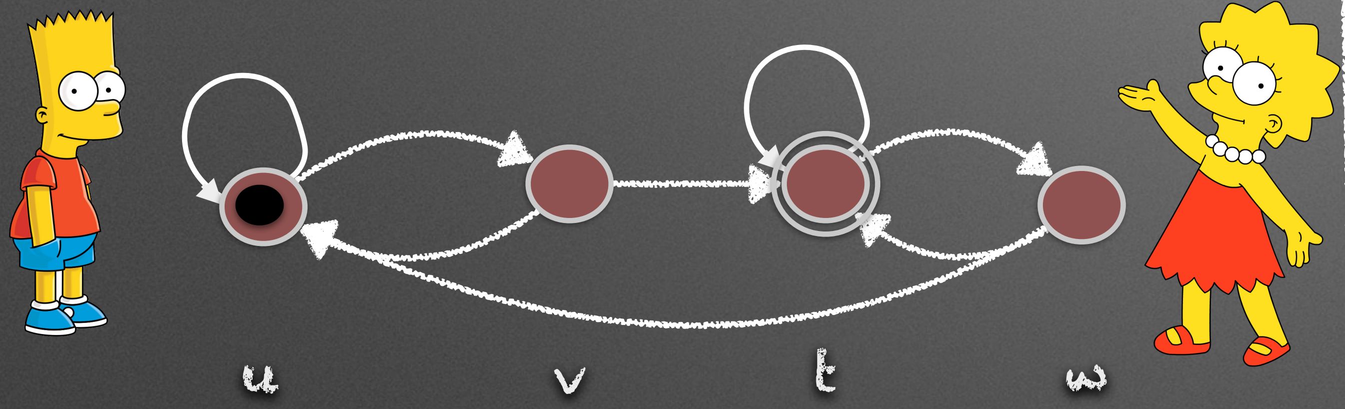
Computing Bart's Buchi Threshold
≡
Computing Lisa's coBuchi Threshold

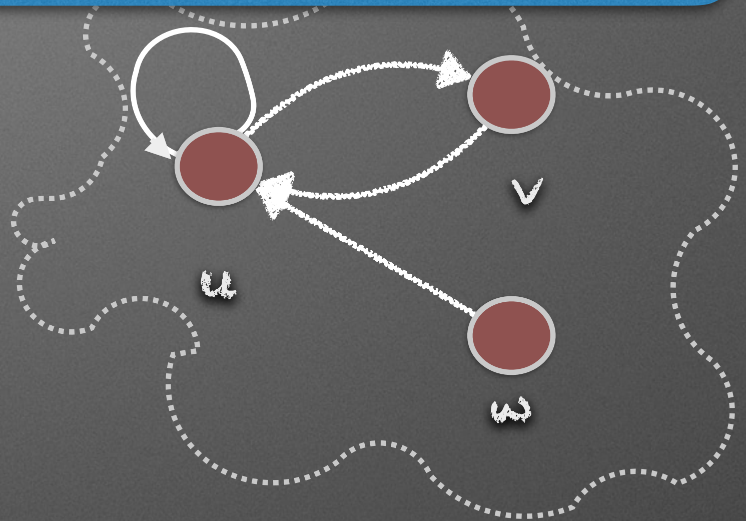
visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

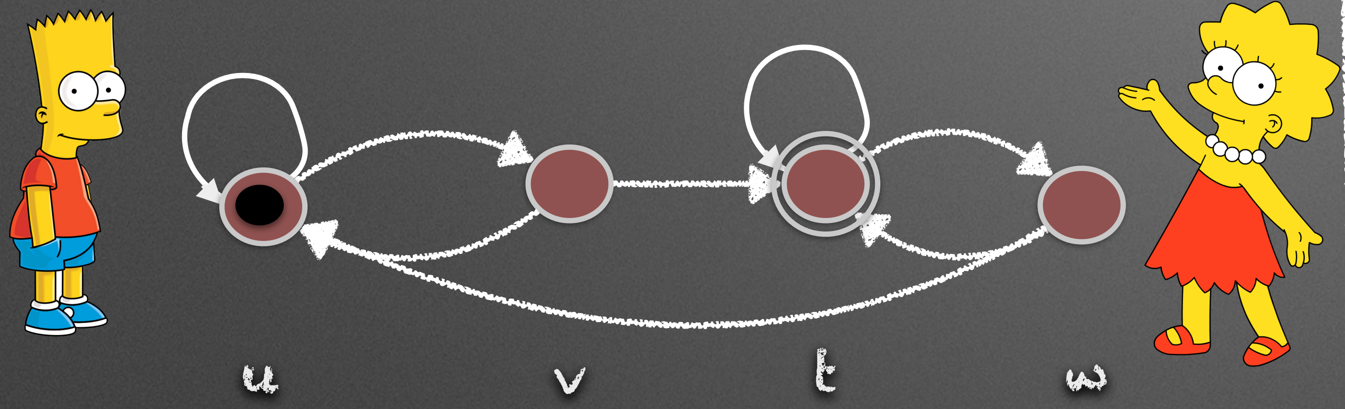
Computing Lisa's coBuchi Threshold

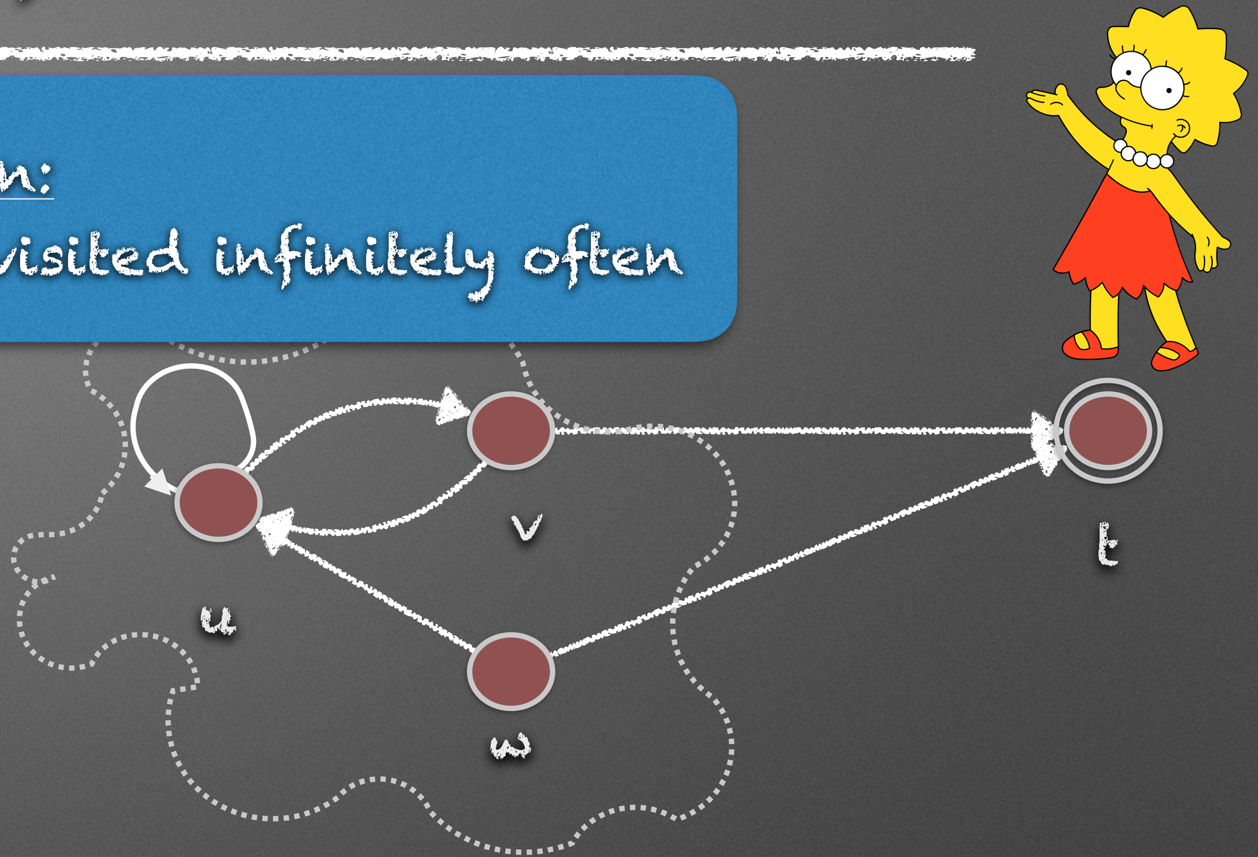
visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

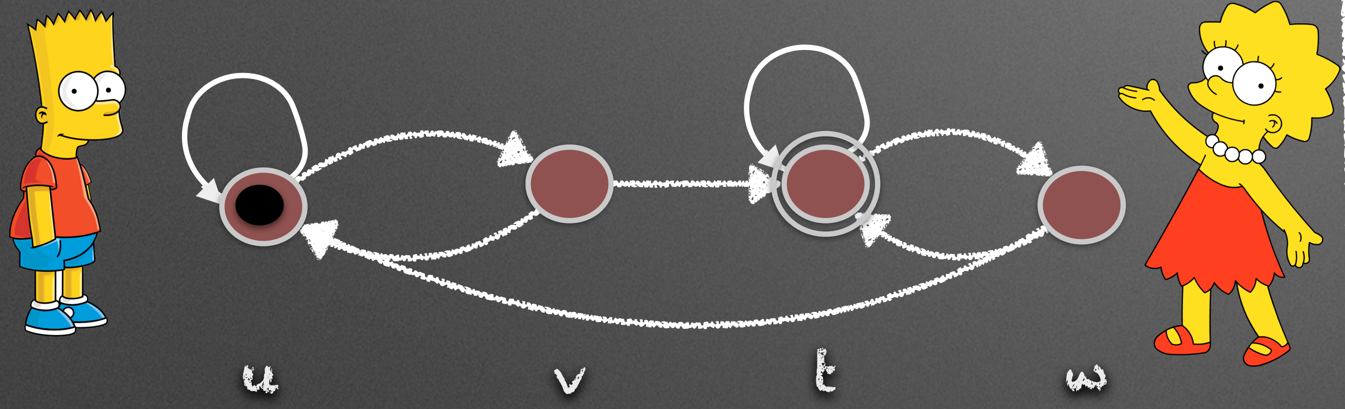
Computing Lisa's coBuchi Threshold

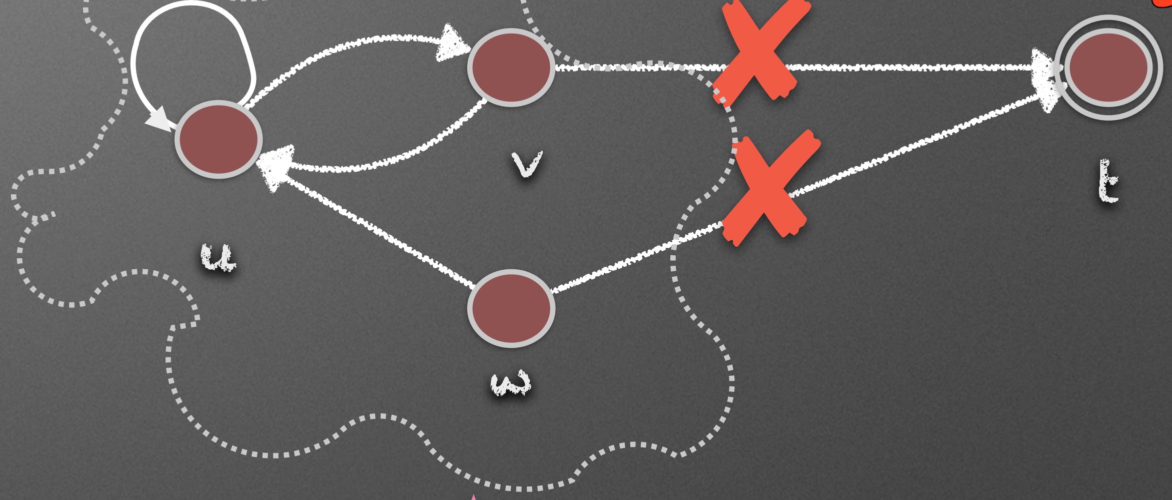
visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

Determinacy [Aghajohari et al'21]

$\Downarrow$

Computing Bart's Buchi Threshold

$\equiv$

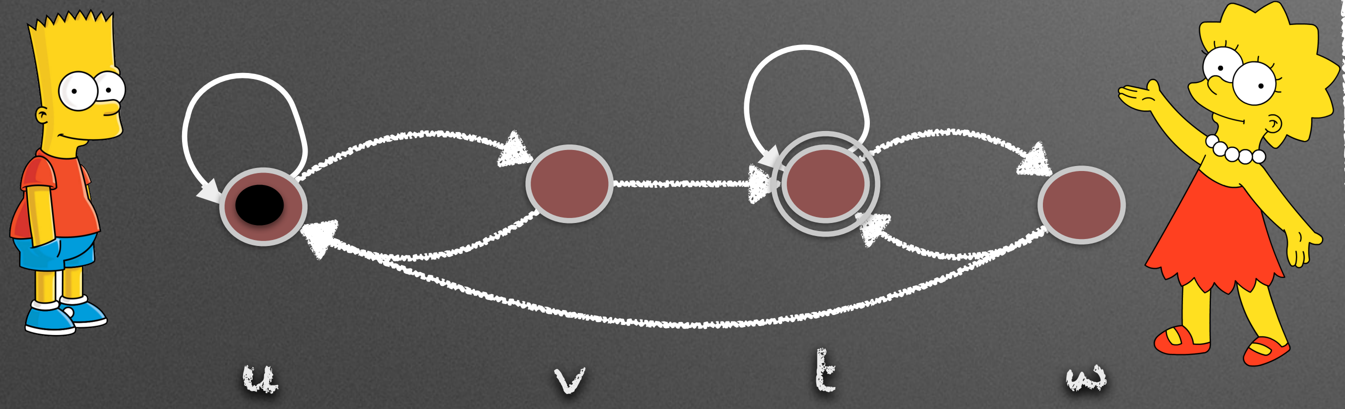
Computing Lisa's coBuchi Threshold

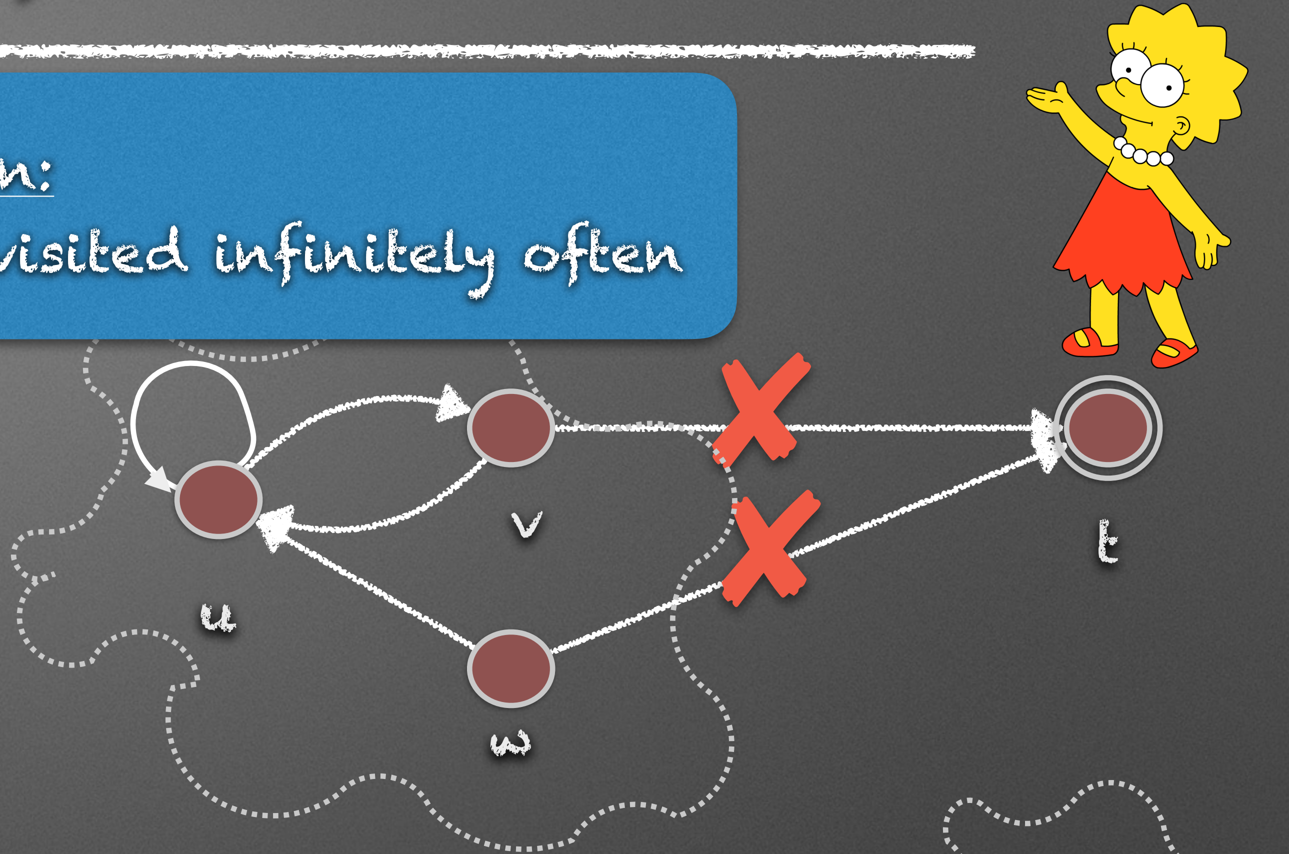
visit t finitely often

Fixed-point Algorithm for co-Buchi Games

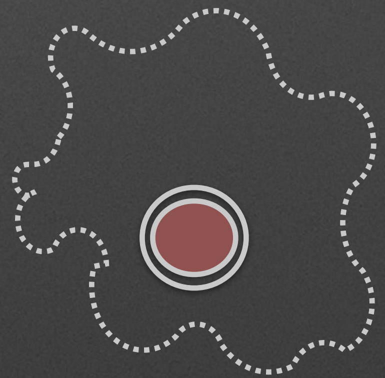
Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)



Determinacy [Aghajohari et al'21]

$\Downarrow$

Computing Bart's Buchi Threshold

$\equiv$

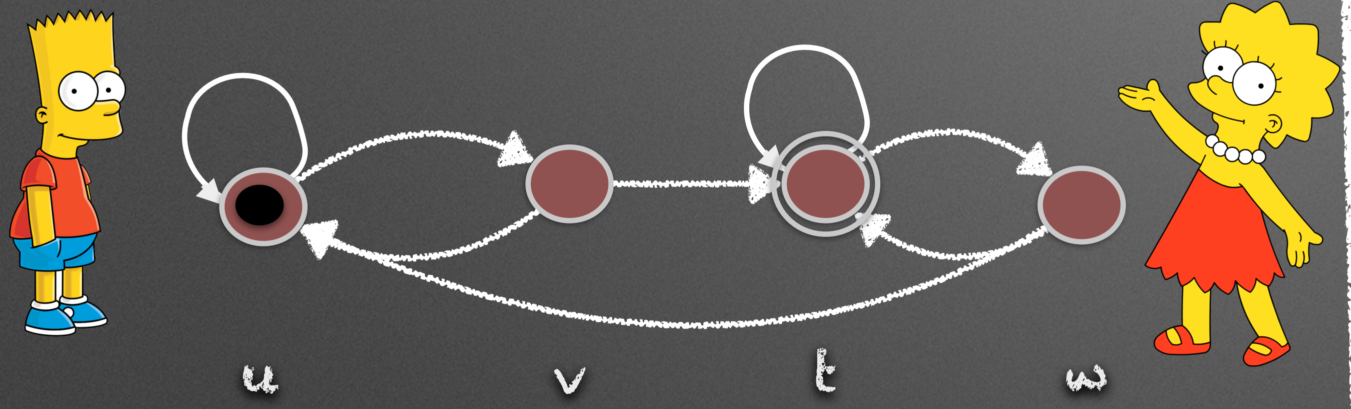
Computing Lisa's coBuchi Threshold

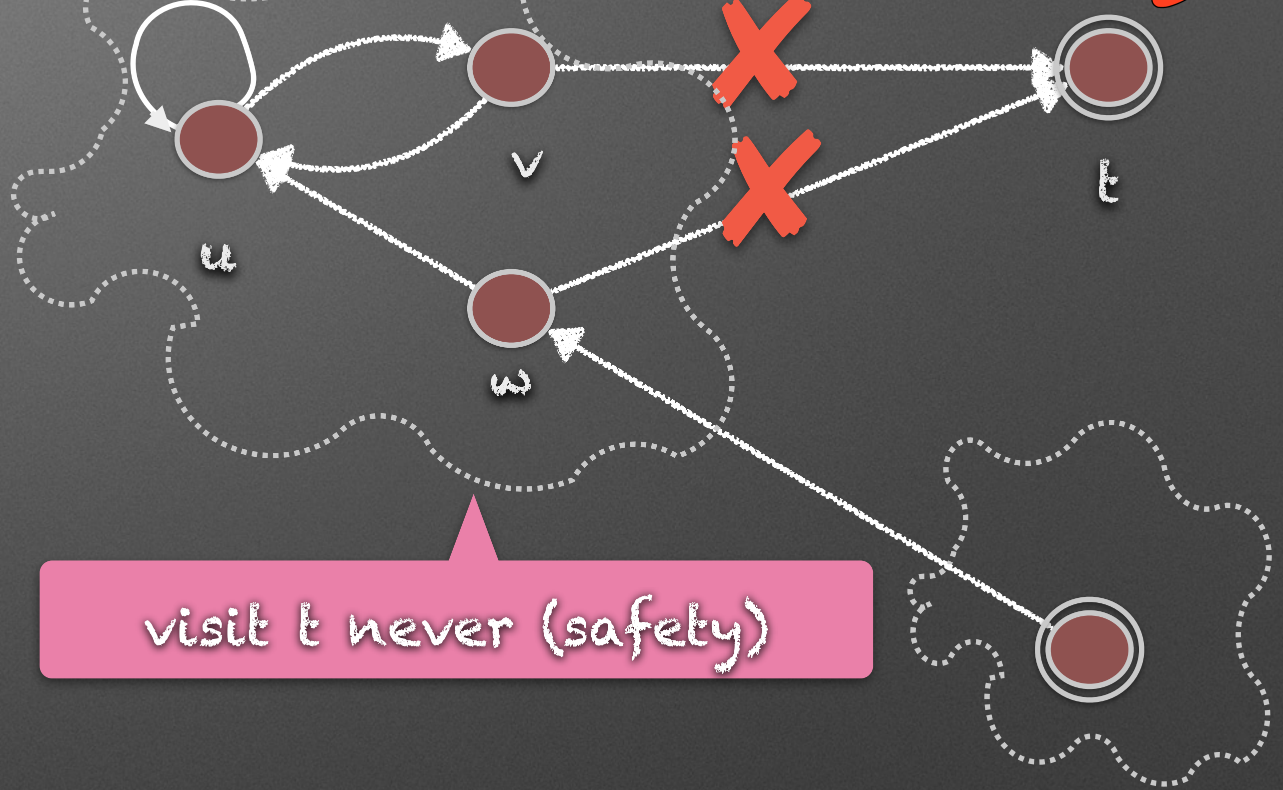
visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

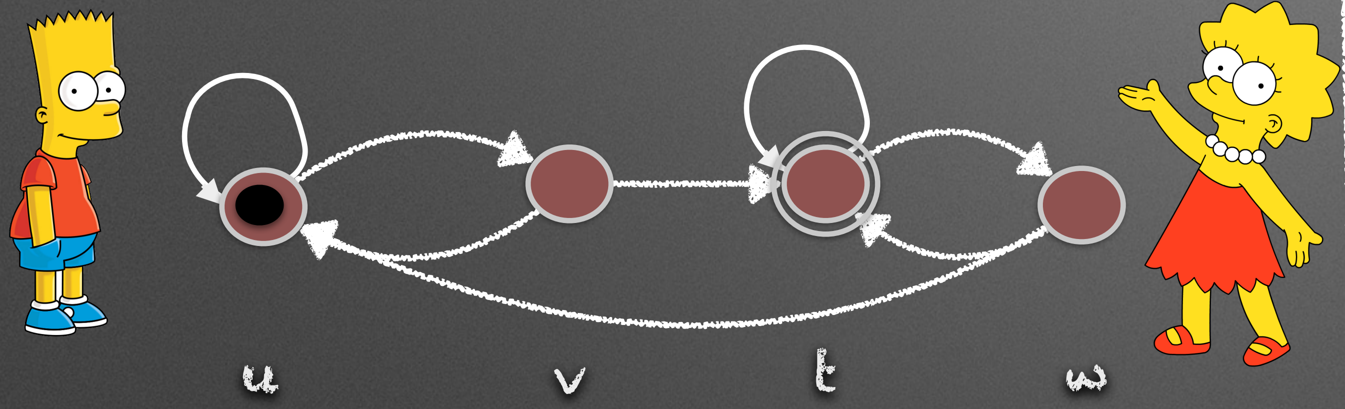
Computing Lisa's coBuchi Threshold

visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

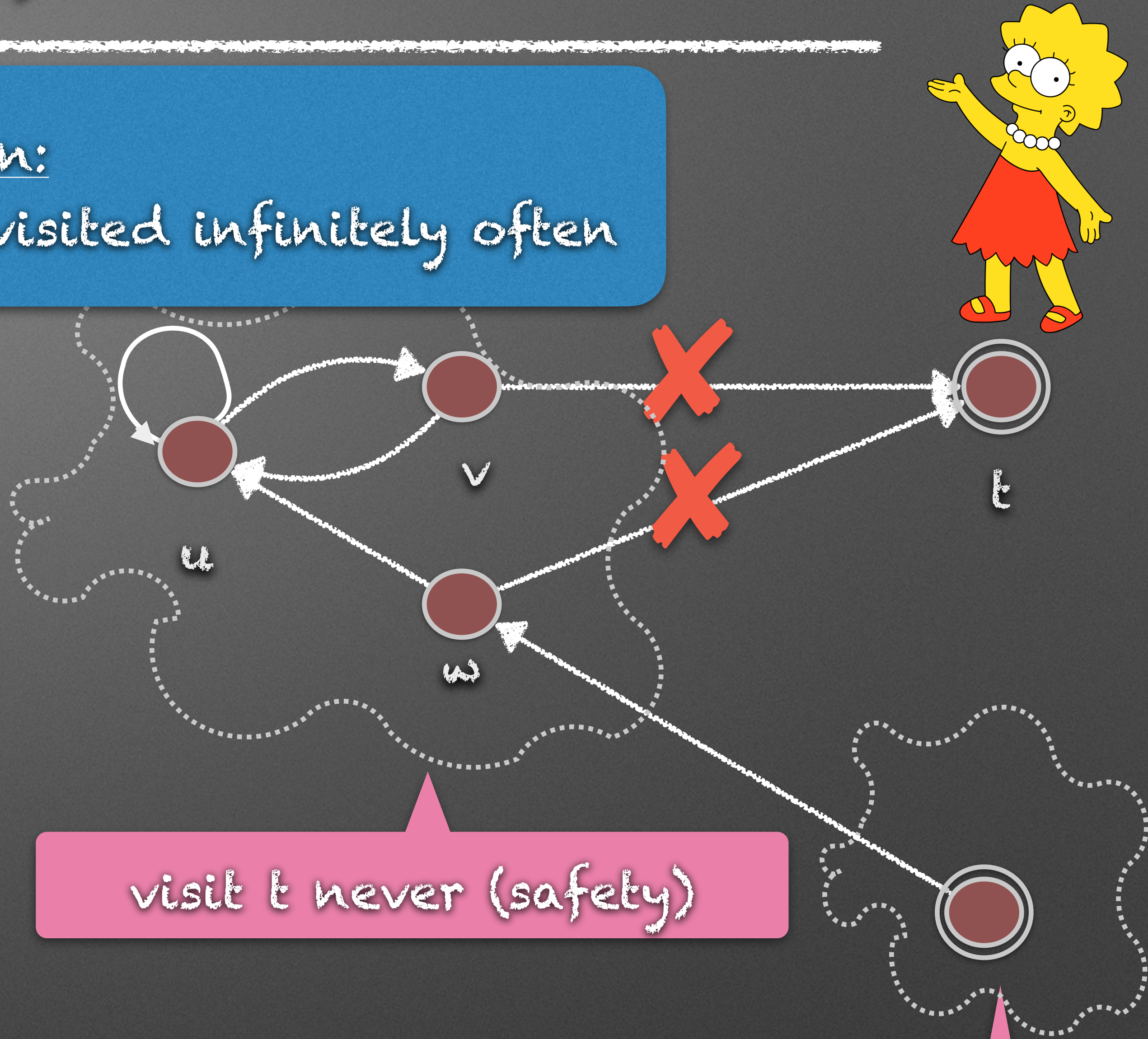
$\Downarrow$

Computing Bart's Buchi Threshold

$\equiv$

Computing Lisa's coBuchi Threshold

visit t finitely often



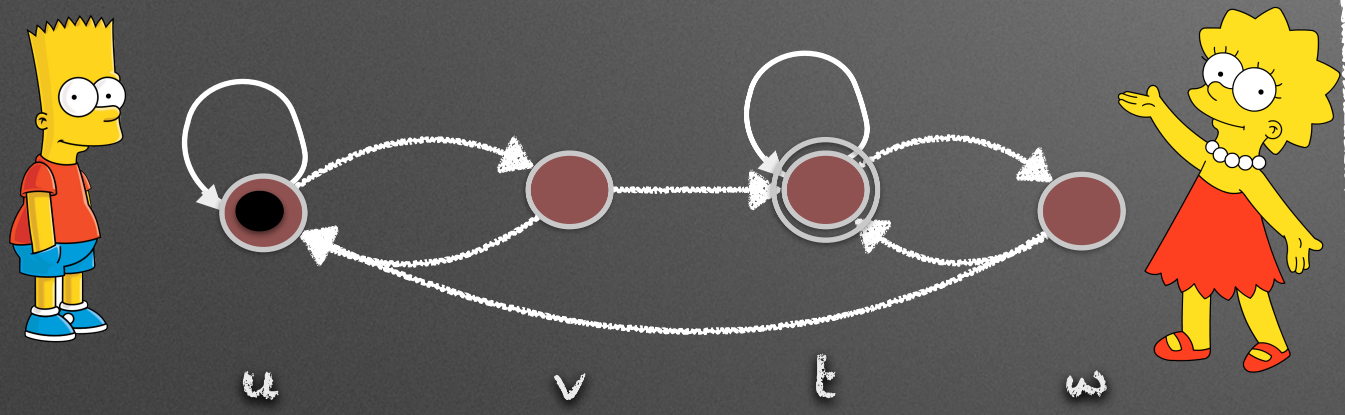
visit t never (safety)

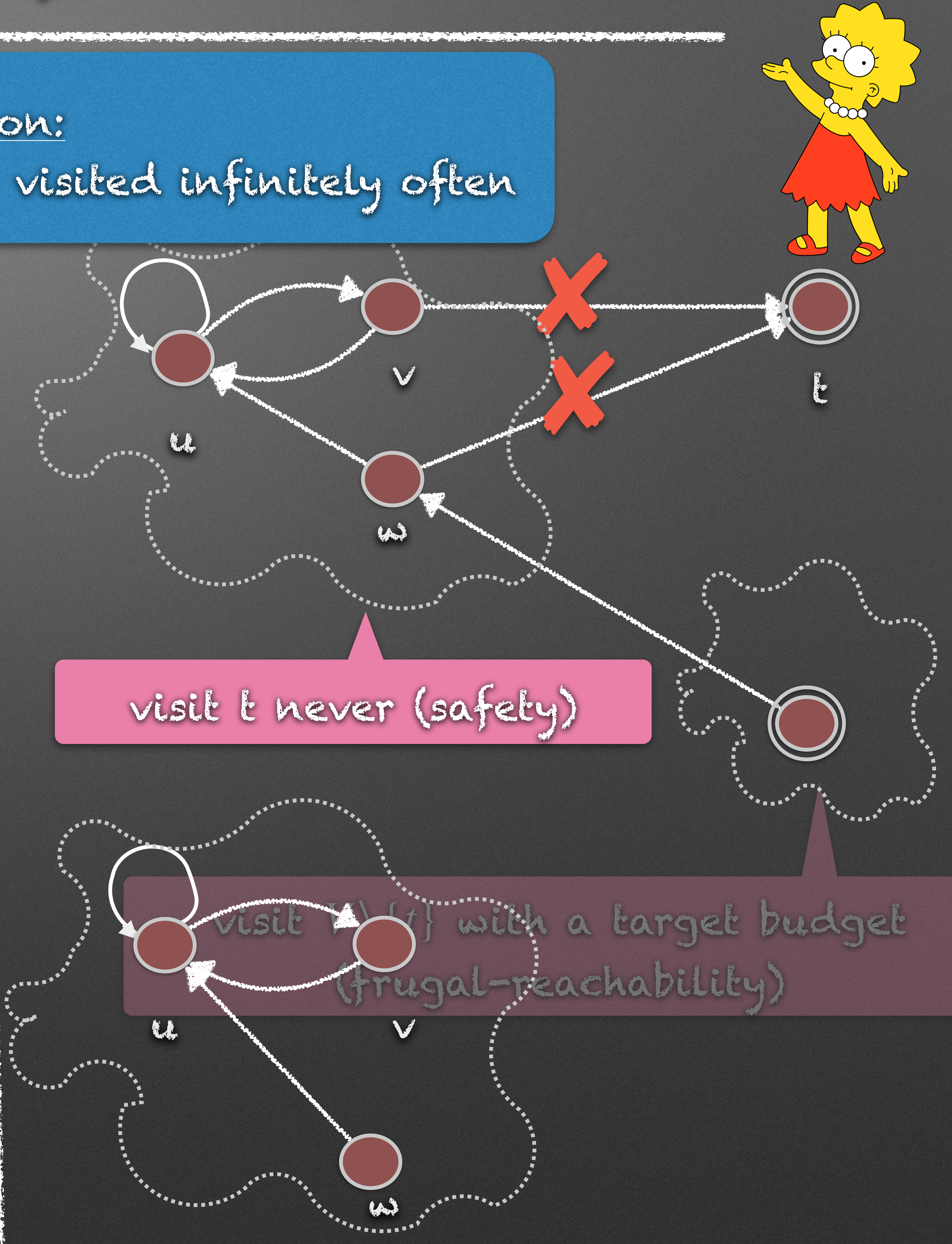
visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

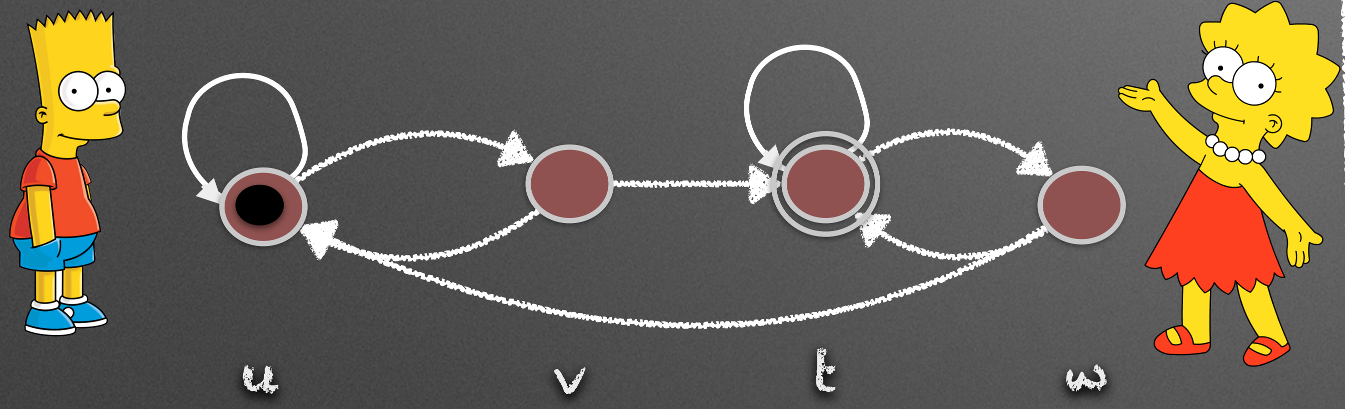
Computing Lisa's coBuchi Threshold

visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

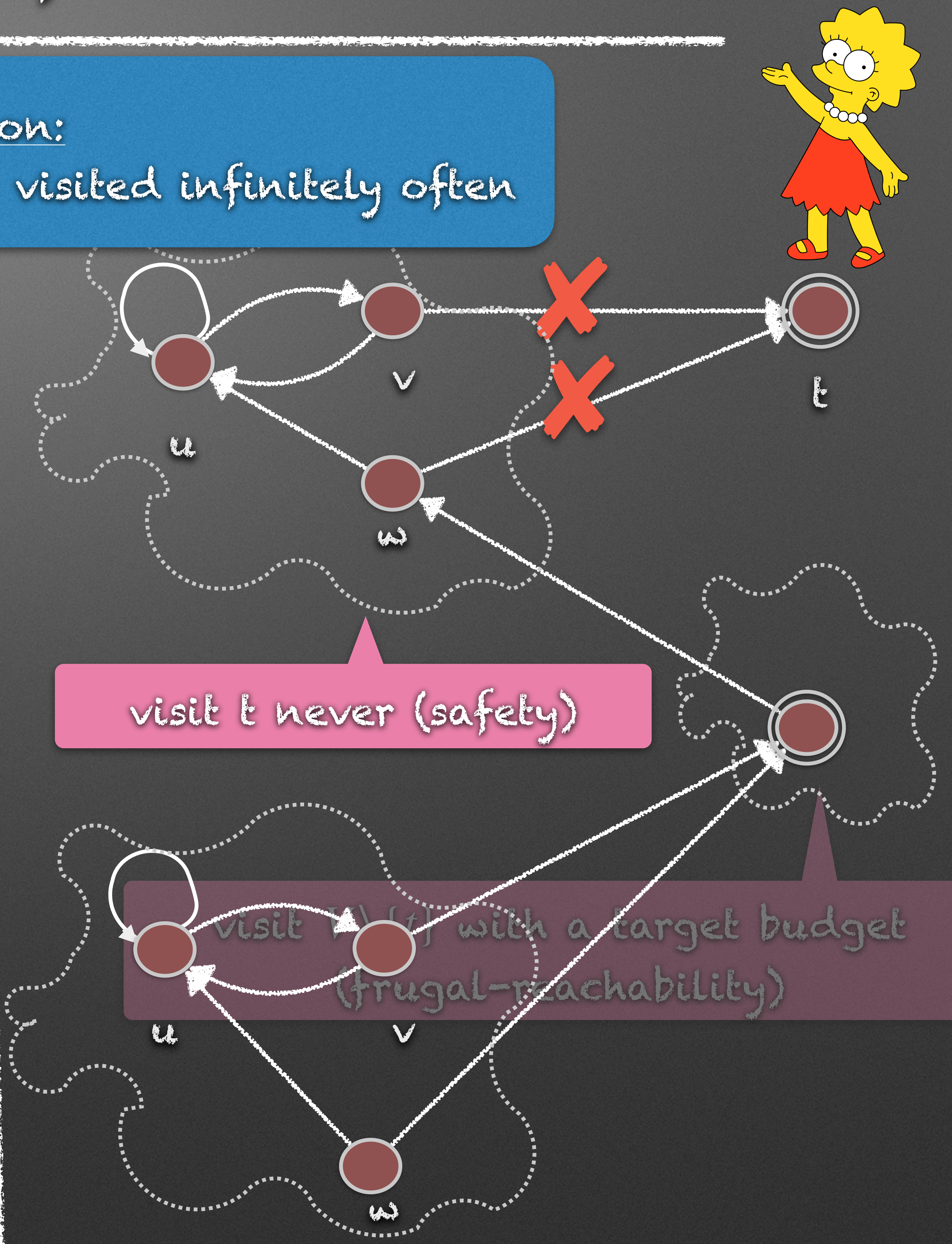
⇓

Computing Bart's Buchi Threshold

≡

Computing Lisa's coBuchi Threshold

visit t finitely often



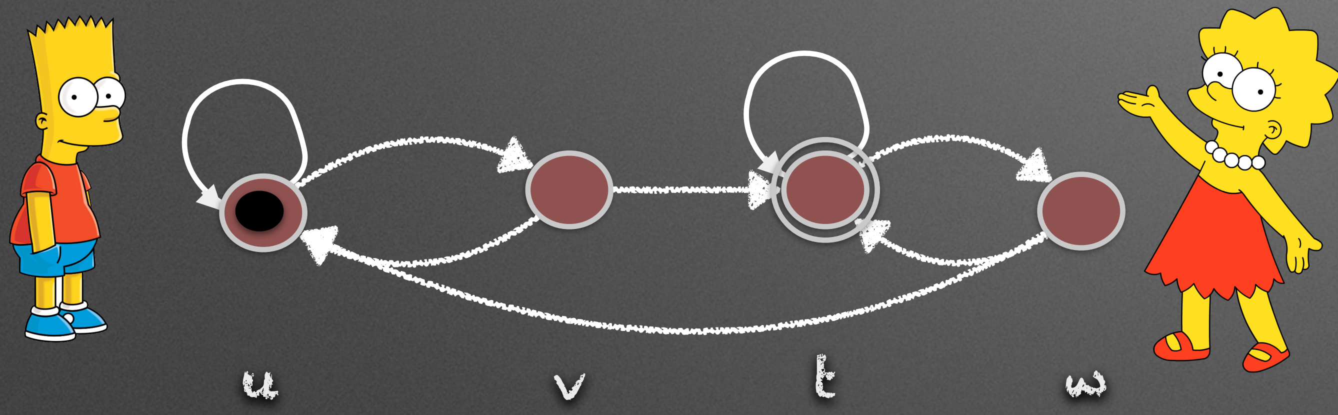
visit t never (safety)

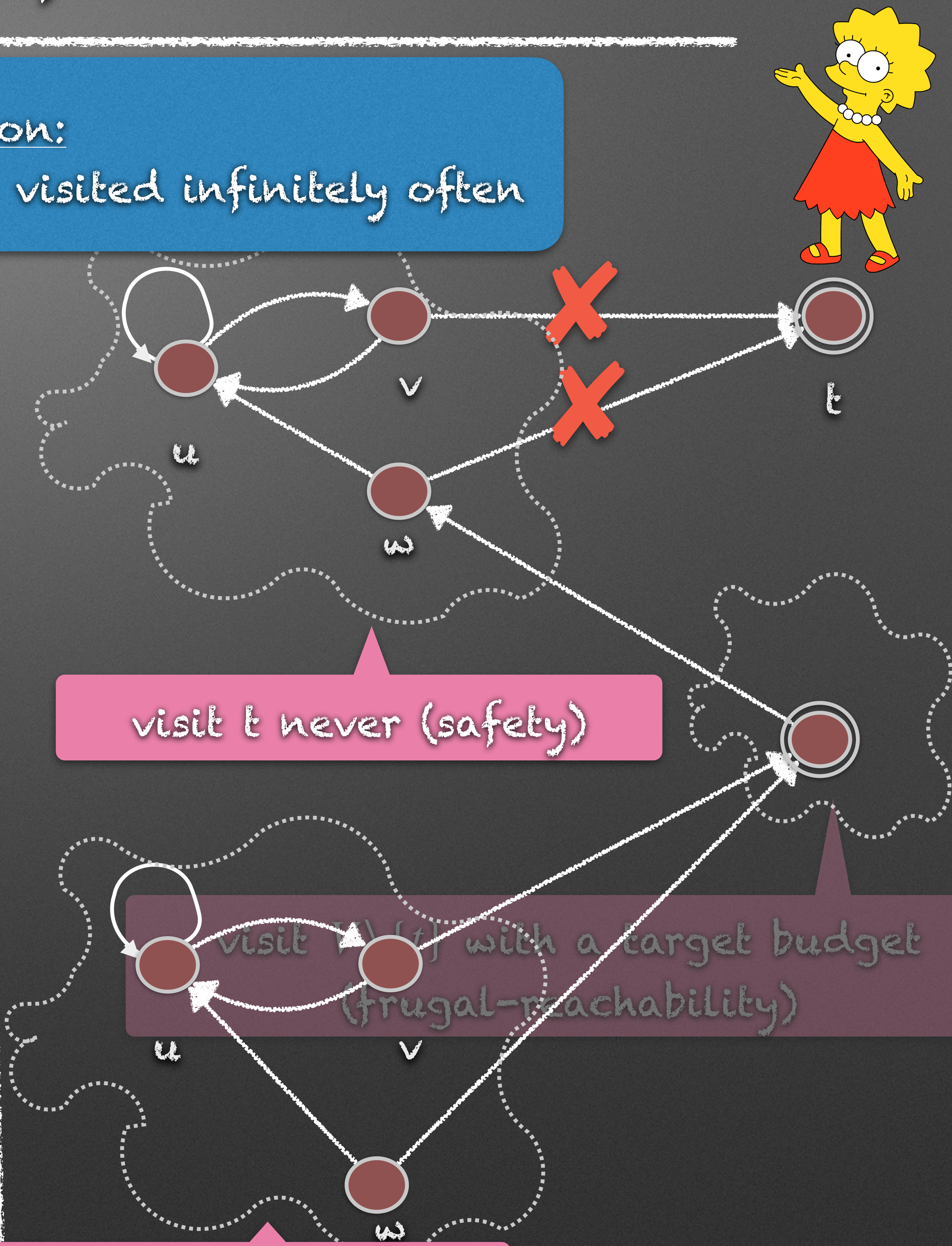
visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

Computing Lisa's coBuchi Threshold

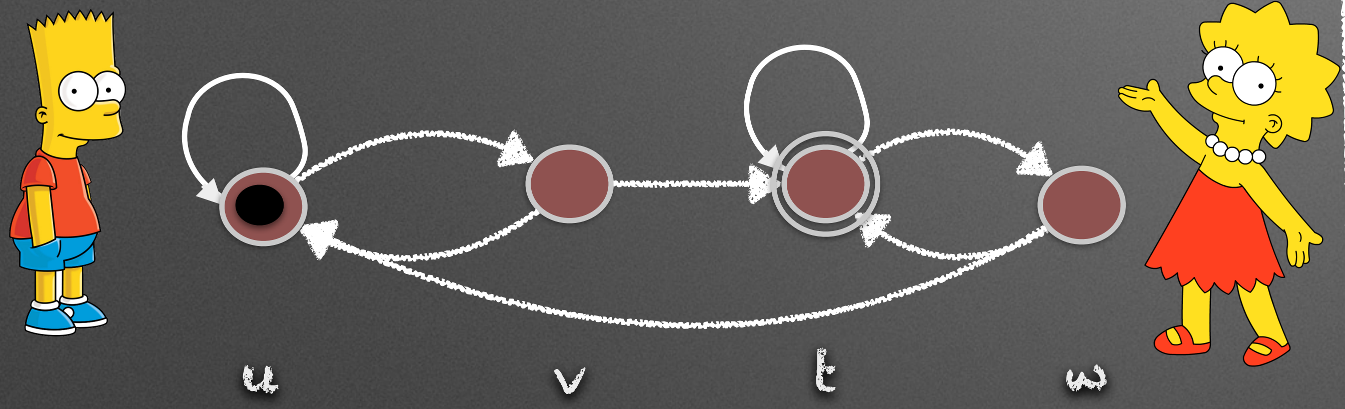
visit t finitely often

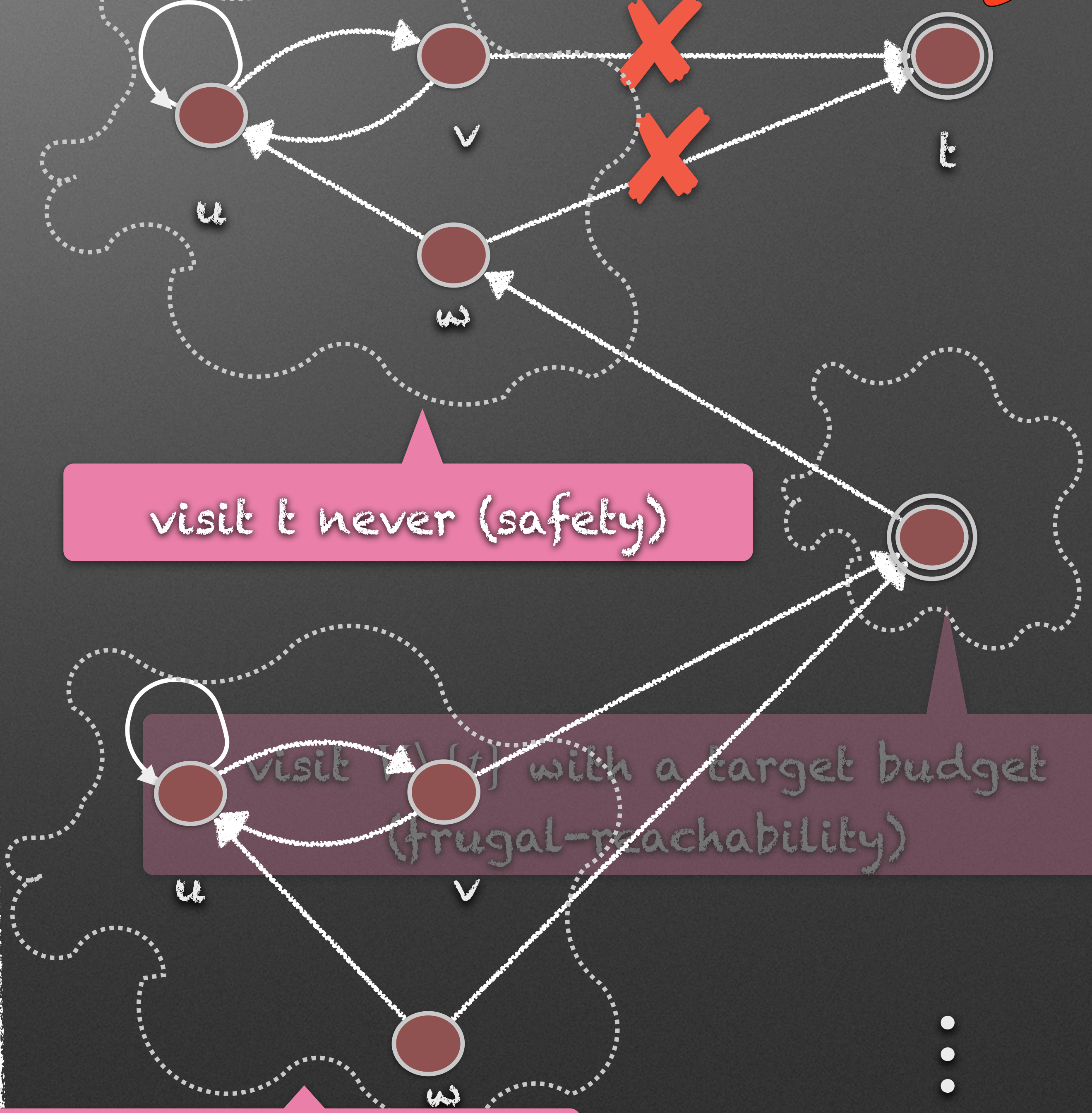
visit t with a target budget

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

Computing Lisa's coBuchi Threshold

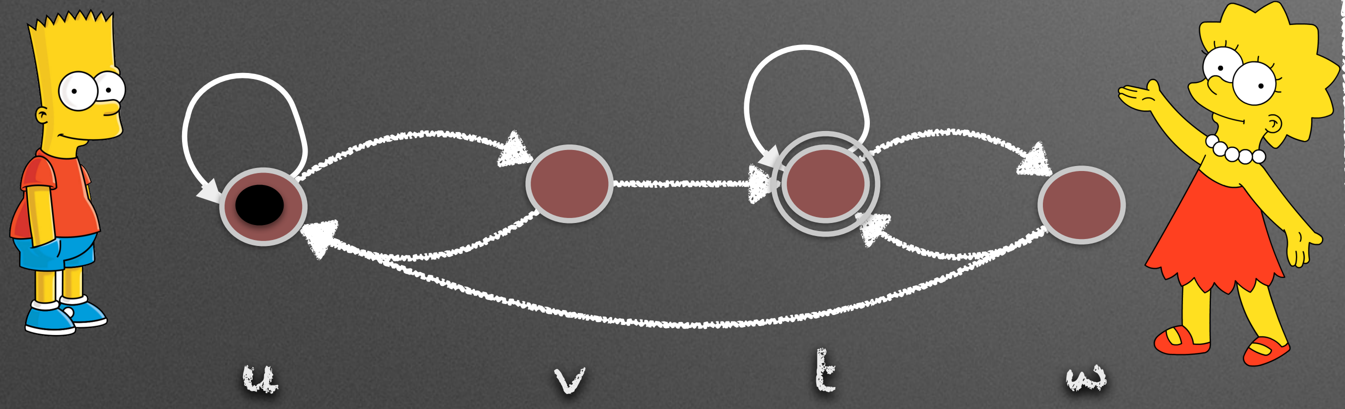
visit t finitely often

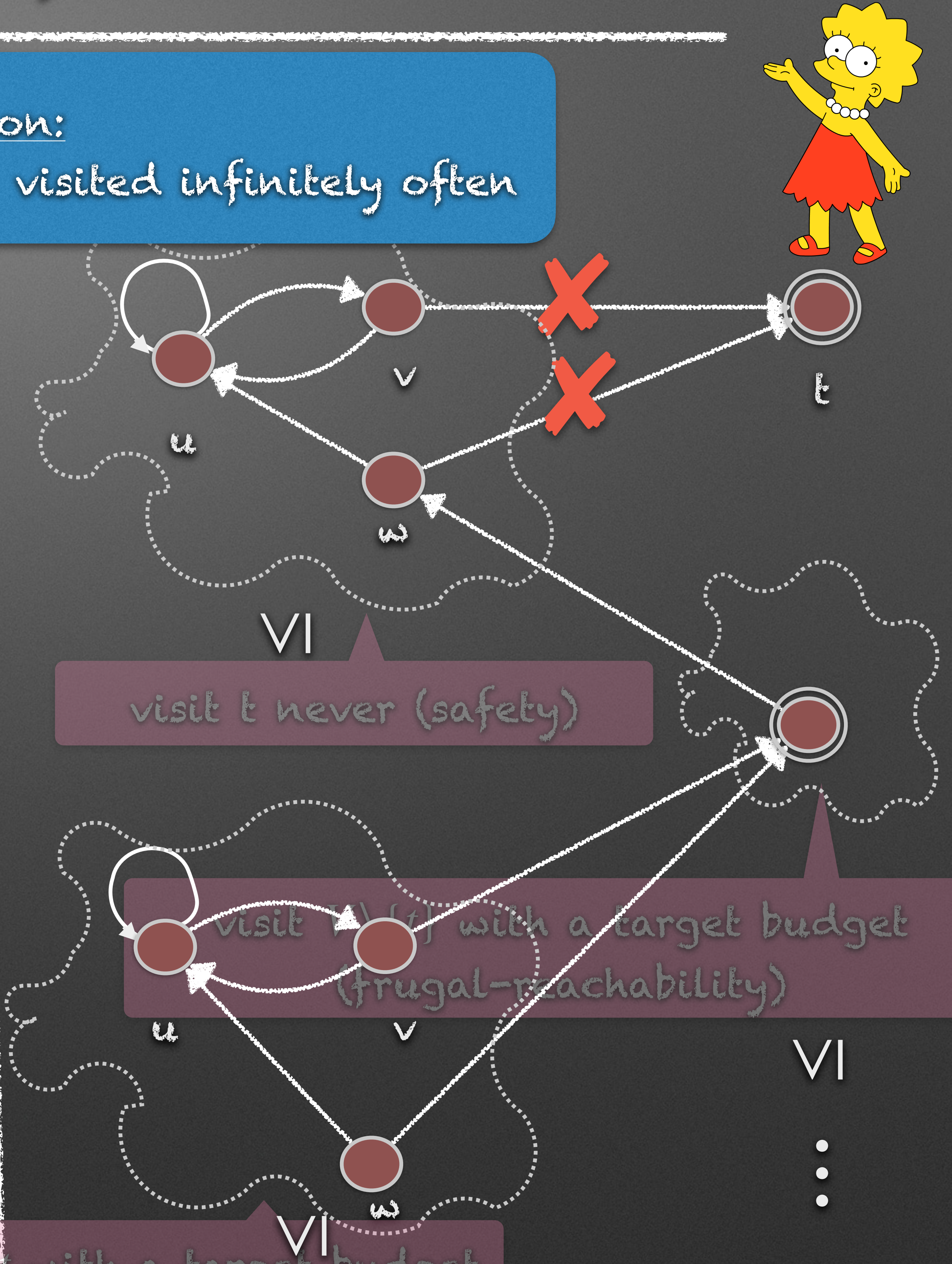
visit t with a target budget

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

VI

visit t with a target budget

Determinacy [Aghajohari et al'21]

⇓

Computing Bart's Buchi Threshold

≡

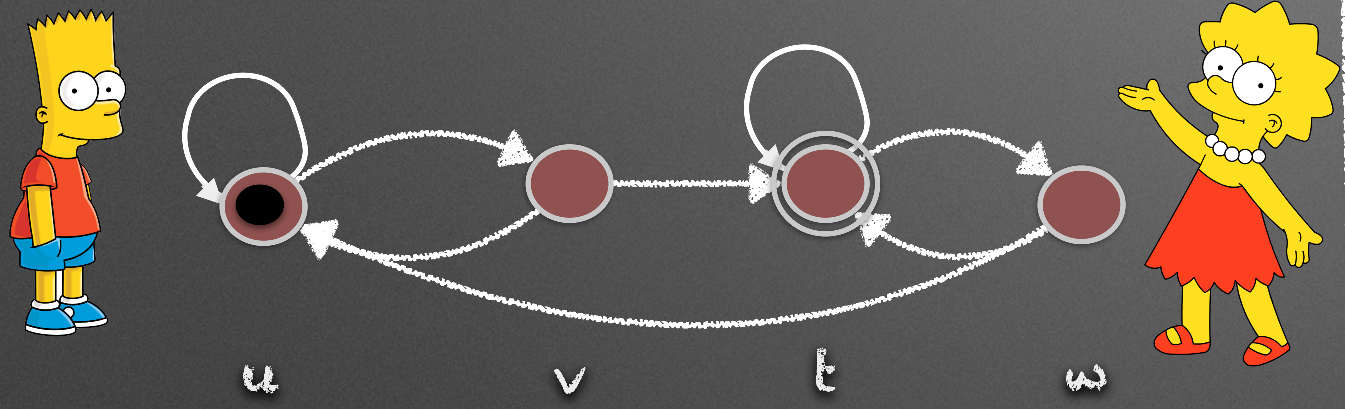
Computing Lisa's coBuchi Threshold

visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often



Determinacy [Aghajohari et al'21]

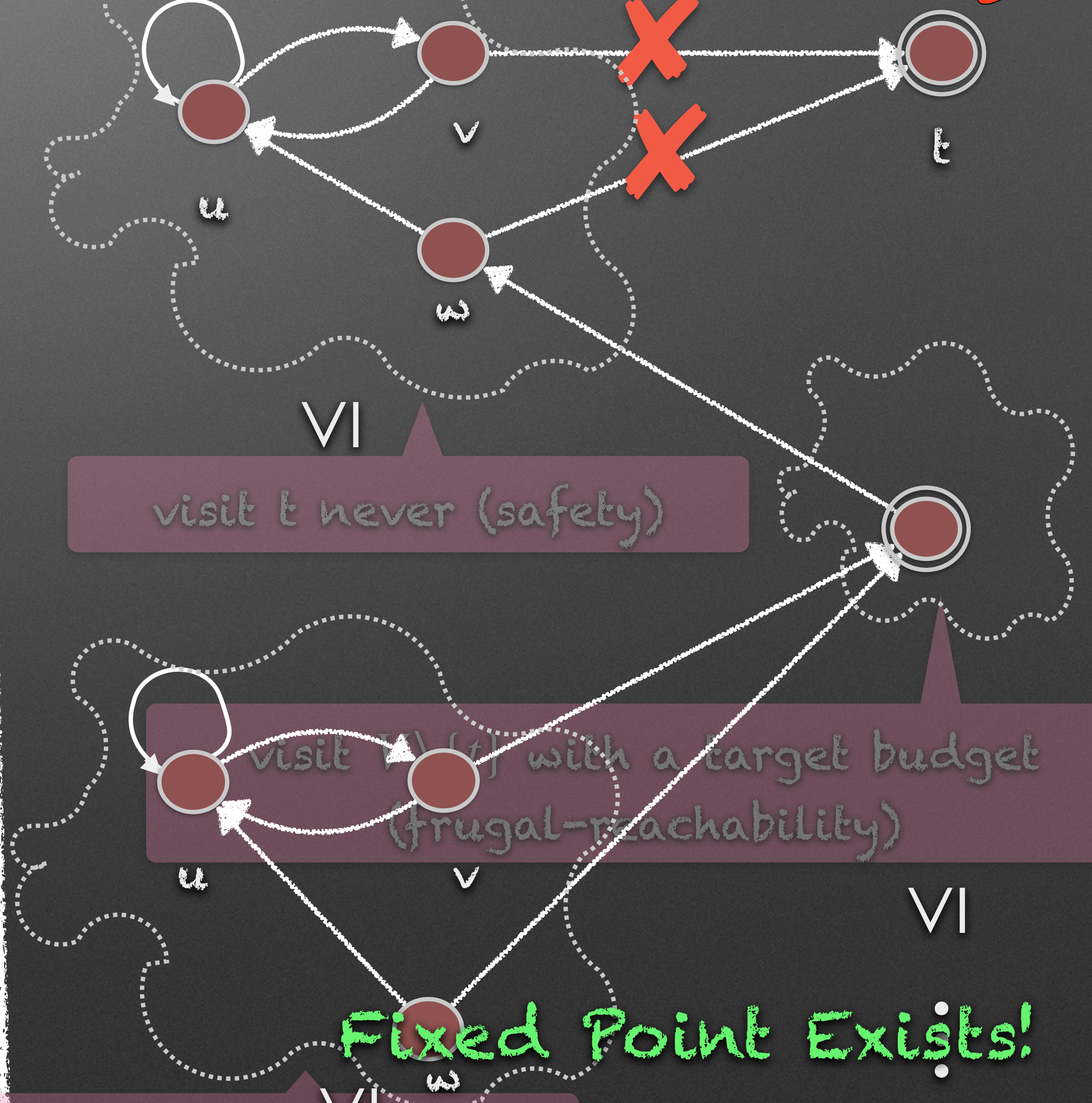
⇓

Computing Bart's Buchi Threshold

≡

Computing Lisa's coBuchi Threshold

visit t finitely often



visit t never (safety)

visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

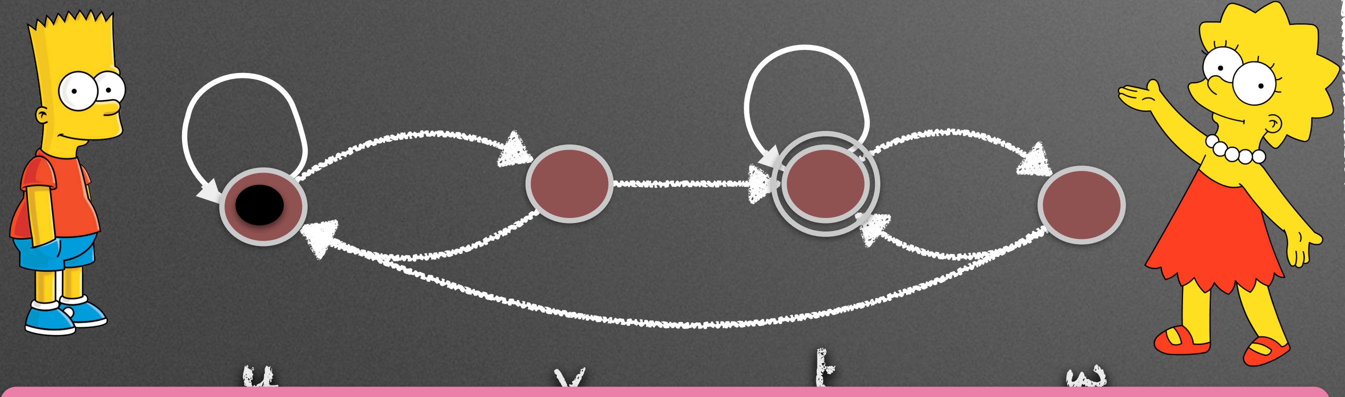
visit t with a target budget

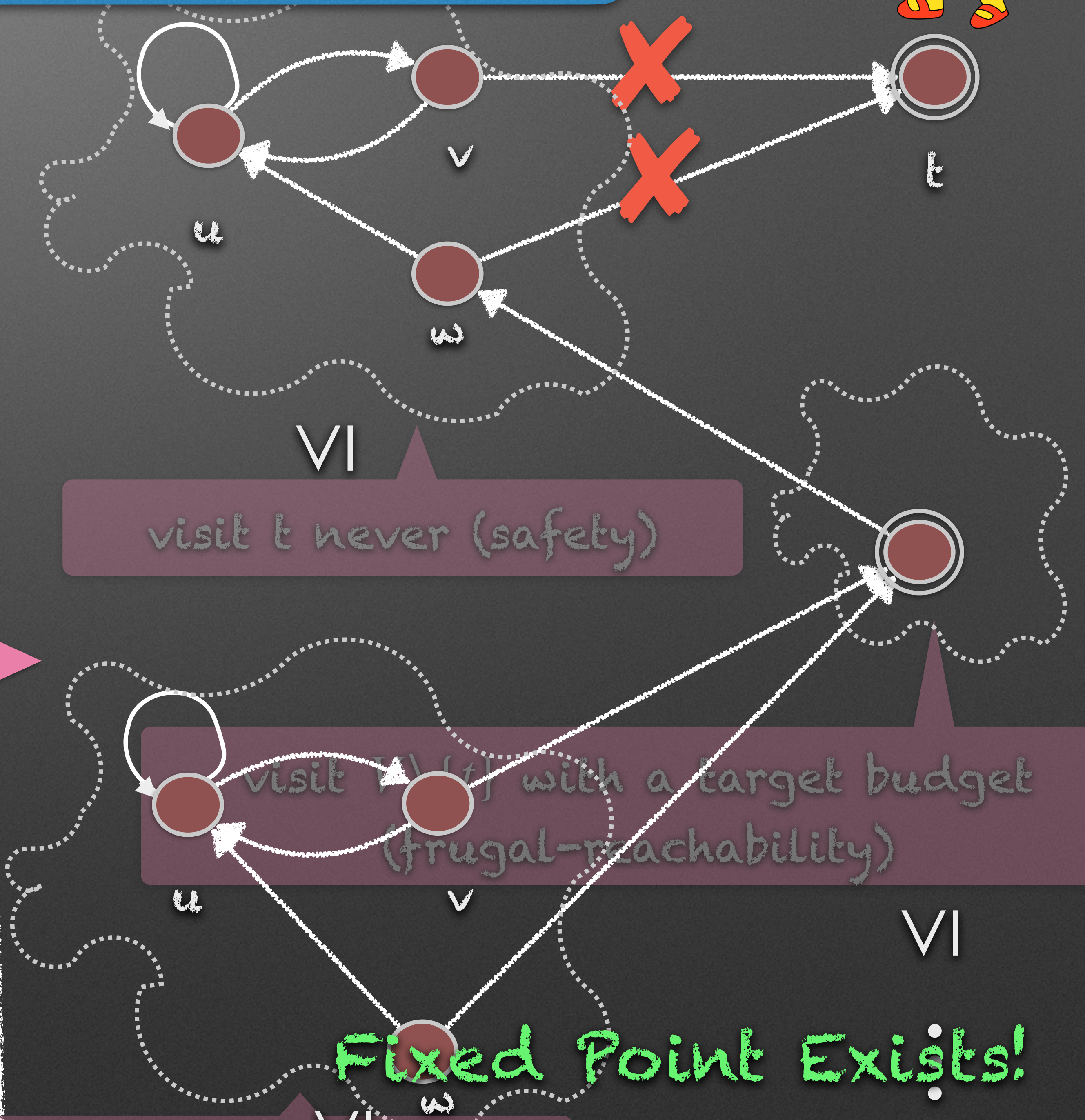
Fixed Point Exists!

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

visit $V \setminus \{t\}$ with a target budget (frugal-reachability)

visit t with a target budget

Fixed Point Exists!

No known structure on the threshold budgets

- ✓ Do Threshold budgets satisfy the average property?
- ✓ Do threshold budgets give rise to bids?
- Best algorithm to compute them is exponential

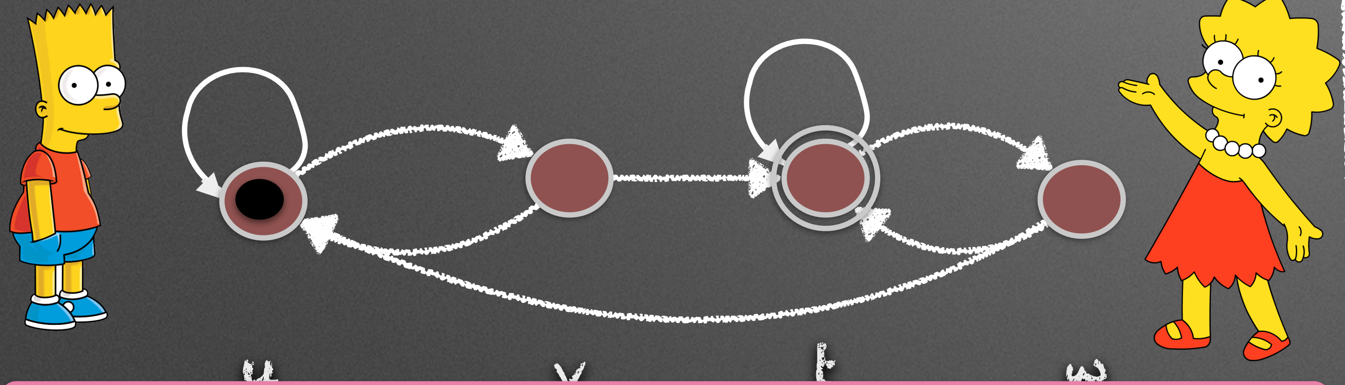
Computing Lisa's coBuchi Threshold

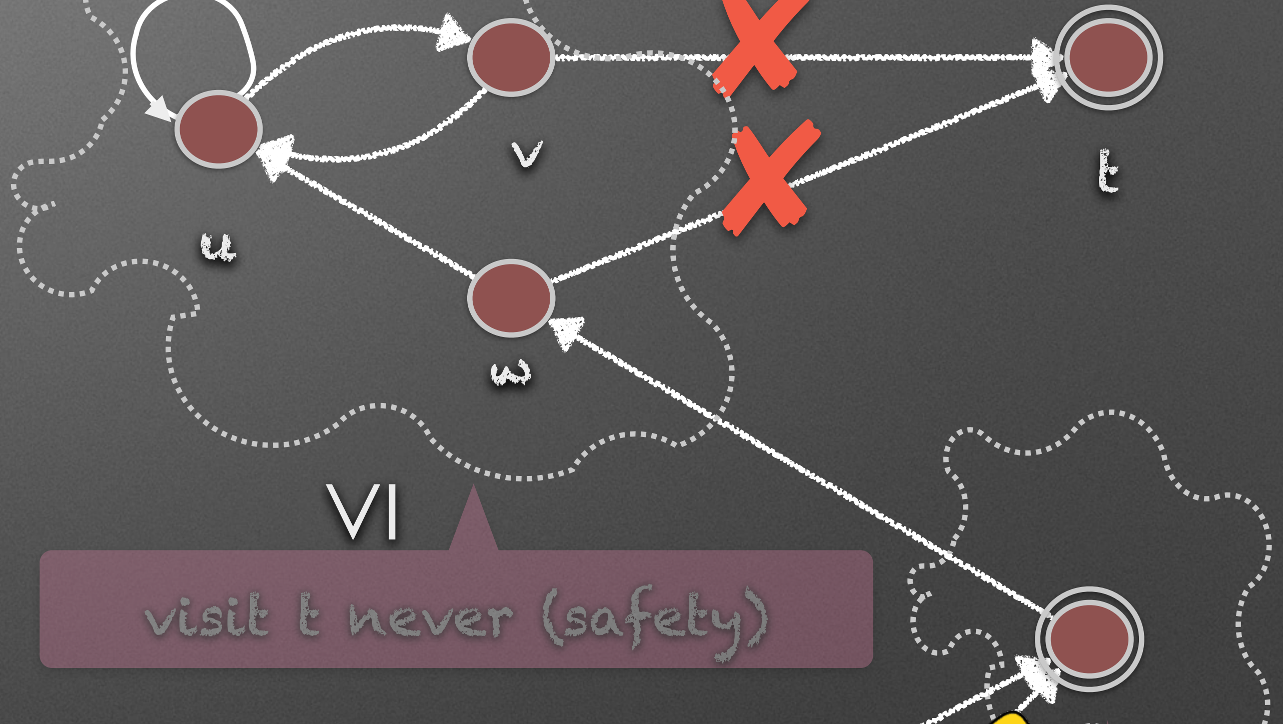
visit t finitely often

Fixed-point Algorithm for co-Buchi Games

Buchi winning condition:

Bart wins iff the set of target vertices T is visited infinitely often





visit t never (safety)

No known structure on the threshold budgets

- ✓ Do Threshold budgets satisfy the average property?
- ✓ Do threshold budgets give rise to bids?
- 🔍 Best algorithm to compute them is exponential

Computing Lis

visi

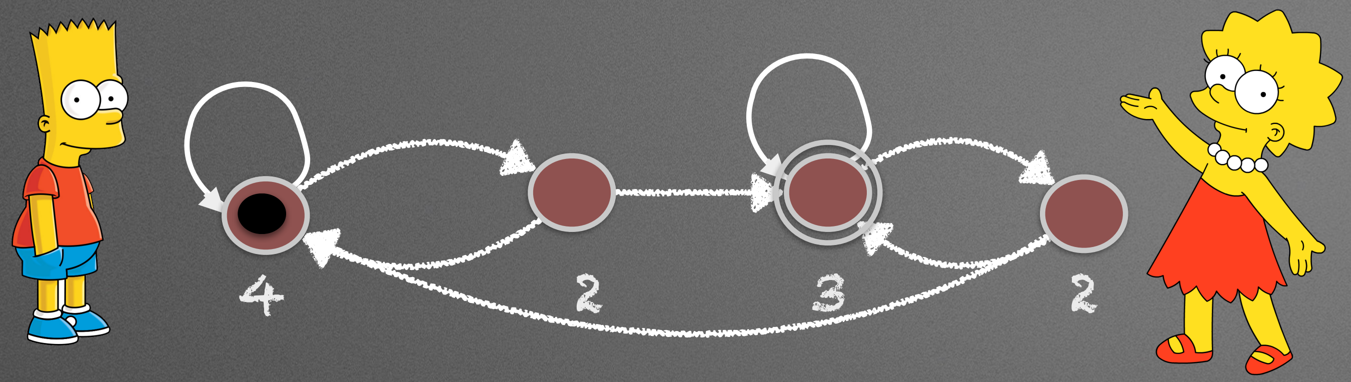
visit V\{t} with

Corollaries: [Avni & S.]

(1) Threshold budgets satisfy a discrete average proper

(2) Optimal bids arise from the thresholds

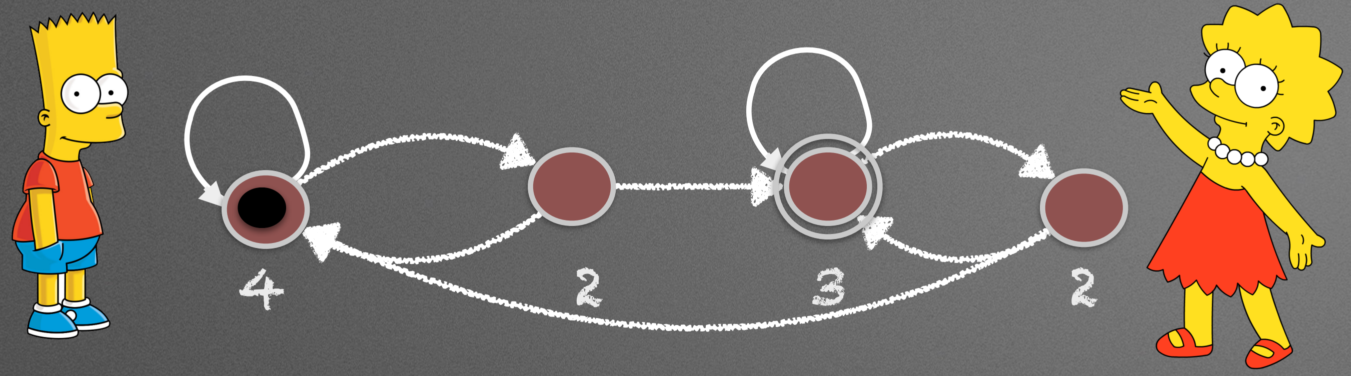
Fixed-point algorithm for Parity Games



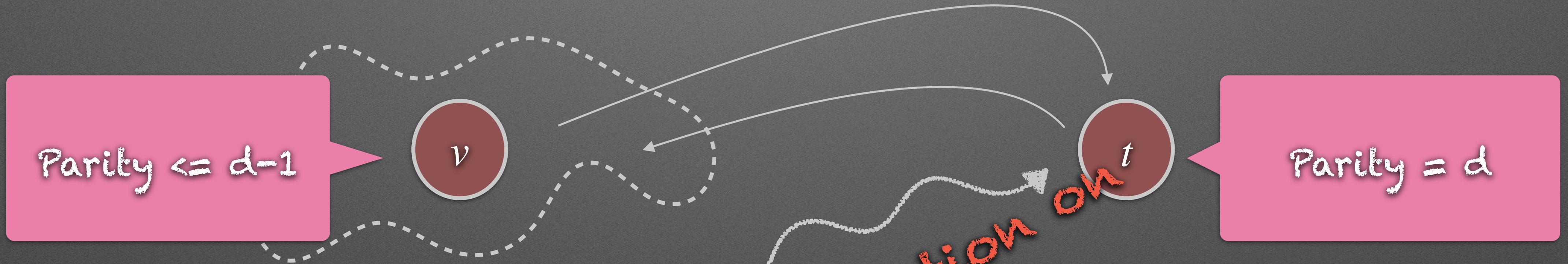
Parity Winning Condition:

Bart (Player 1) wins iff infinitely occurring max priority is odd.

Fixed-point algorithm for Parity Games



Parity Winning Condition:
Bart (Player 1) wins iff infinitely occurring max priority is odd.



Additionally: Induction on parities

$T_1 = \text{fr-parity}(t, k+1)$

$R_1 = \text{fr-Reach}(v, T_1)$

$T_2 = \text{fr-parity}(t, R_1)$

⋮

$T_n = \text{fr-parity}(t, R_{n-1})$

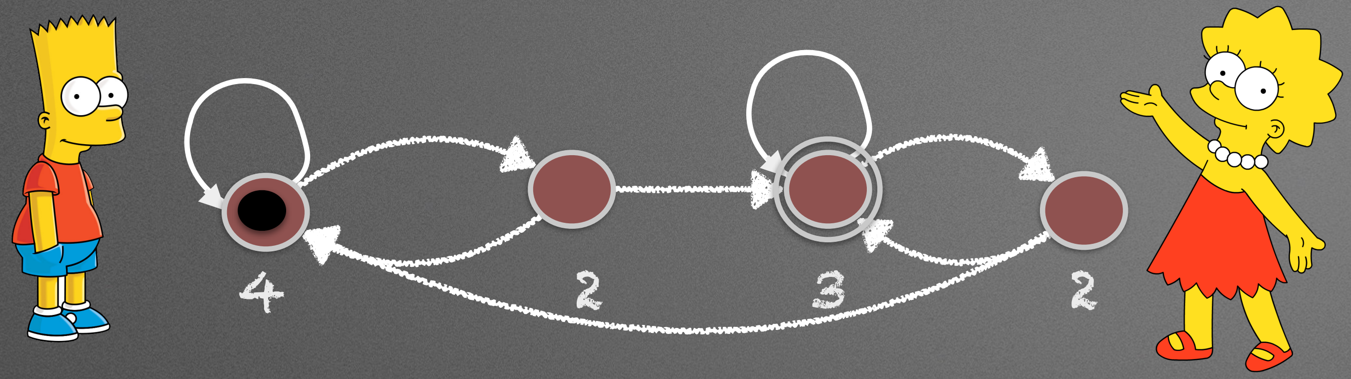
$R_n = \text{fr-Reach}(v, T_n)$

||

$T_{n+1} = \text{fr-Parity}(t, R_n)$

Theorem:
$Th_{Parity}(v) = T_n$

Fixed-point algorithm for Parity Games



Parity Winning Condition:

Bart (Player 1) wins iff infinitely occurring max priority is odd.

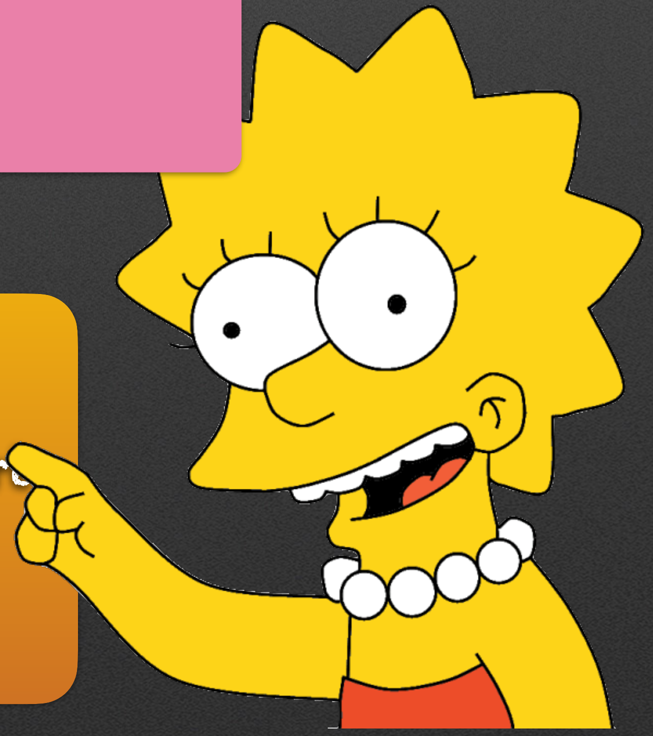
No known structure on the threshold budgets

- ✓ Do Threshold budgets satisfy the average property?
- ✓ Do threshold budgets give rise to bids?
- 🔍 Best algorithm to compute them is exponential

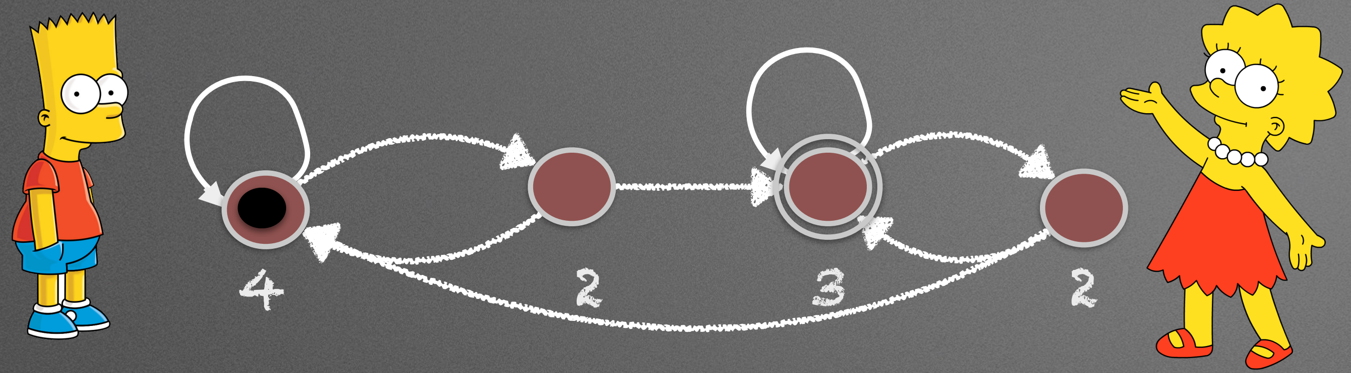
Corollaries: [Avni & S.]

(1) Threshold budgets satisfy a discrete average propert

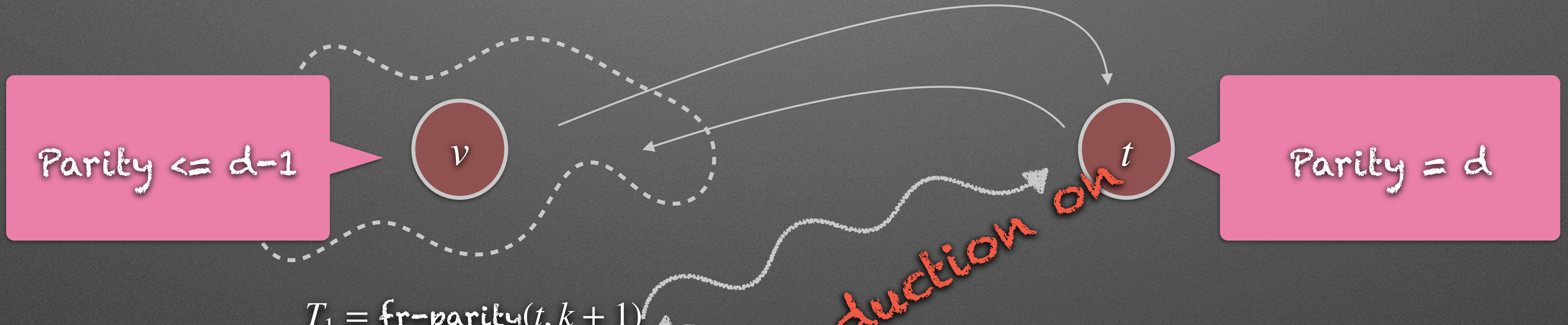
(2) Optimal bids arise from the thresholds



Fixed-point algorithm for Parity Games



Parity Winning Condition:
Bart (Player 1) wins iff infinitely occurring max priority is odd.



$T_1 = \text{fr-parity}(t, k+1)$

No known structure on the threshold budgets

- ✓ Do Threshold budgets satisfy the average property?
- ✓ Do threshold budgets give rise to bids?
- Best algorithm to compute them is exponential

1)

The

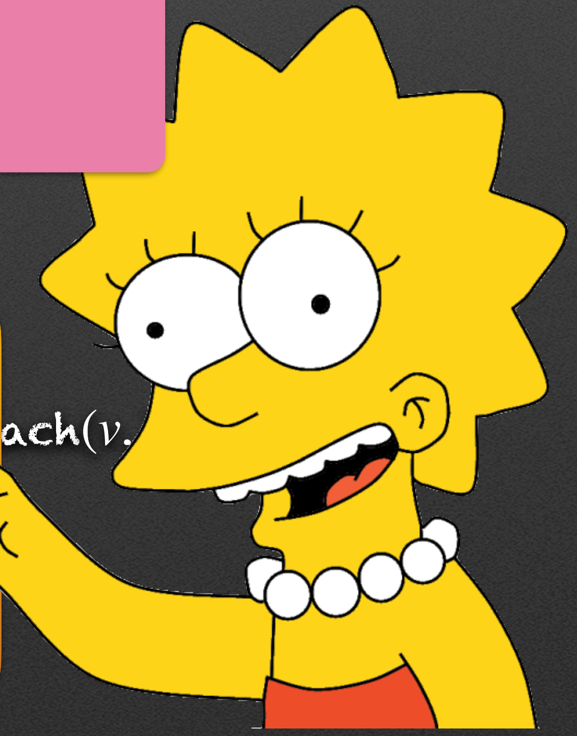
Th_{Parit}

ach(v.

Corollaries: [Avni & S.]

(1) Threshold budgets satisfy a discrete average propert

(2) Optimal bids arise from the thresholds



Computing Threshold budgets for Parity Games

Computing Threshold budgets for Parity Games

Guess a $T : V \to [k+1]$,
and check if it satisfies the discrete average property.

Computing Threshold budgets for Parity Games

Guess a $T : V \to [k+1]$,

and check if it satisfies the discrete average property.

$$T(v) = \lfloor \frac{|T(v^+)| + |T(v^-)|}{2} \rfloor + \varepsilon$$

such that $\varepsilon = 0, 1$, or $*$,

($*$ denotes that the tie-breaking advantage is needed)

Computing Threshold budgets for Parity Games

Guess a $T: V \to [k+1]$,

and check if it satisfies the discrete average property.

$$T(v) = \lfloor \frac{|T(v^+)| + |T(v^-)|}{2} \rfloor + \varepsilon$$

such that $\varepsilon = 0, 1,$ or $*$,

($*$ denotes that the tie-breaking advantage is needed)

Fails

Computing Threshold budgets for Parity Games

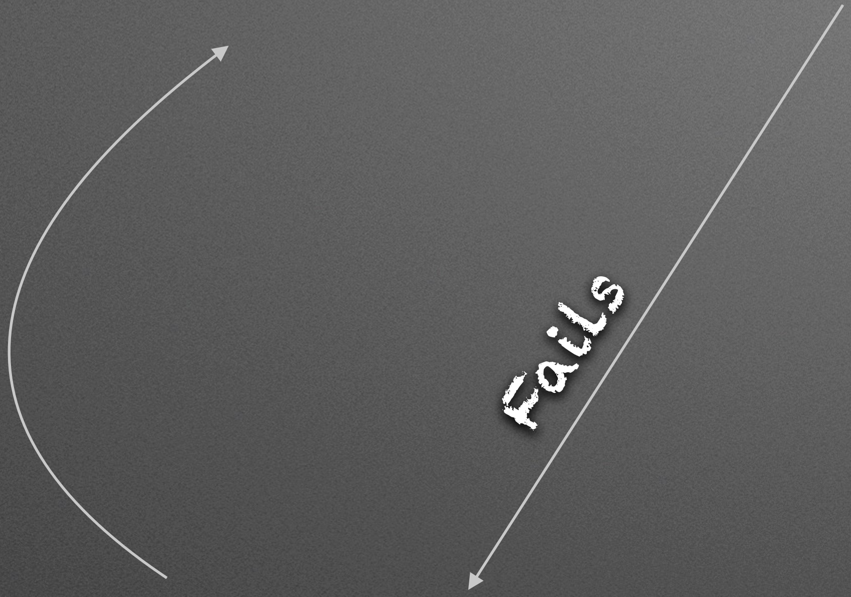
Guess a $T : V \to [k+1]$,

and check if it satisfies the discrete average property.

$$T(v) = \lfloor \frac{|T(v^+)| + |T(v^-)|}{2} \rfloor + \varepsilon$$

such that $\varepsilon = 0, 1,$ or $*$,

($*$ denotes that the tie-breaking advantage is needed)



Fails

Go back to Step 1

Computing Threshold budgets for Parity Games

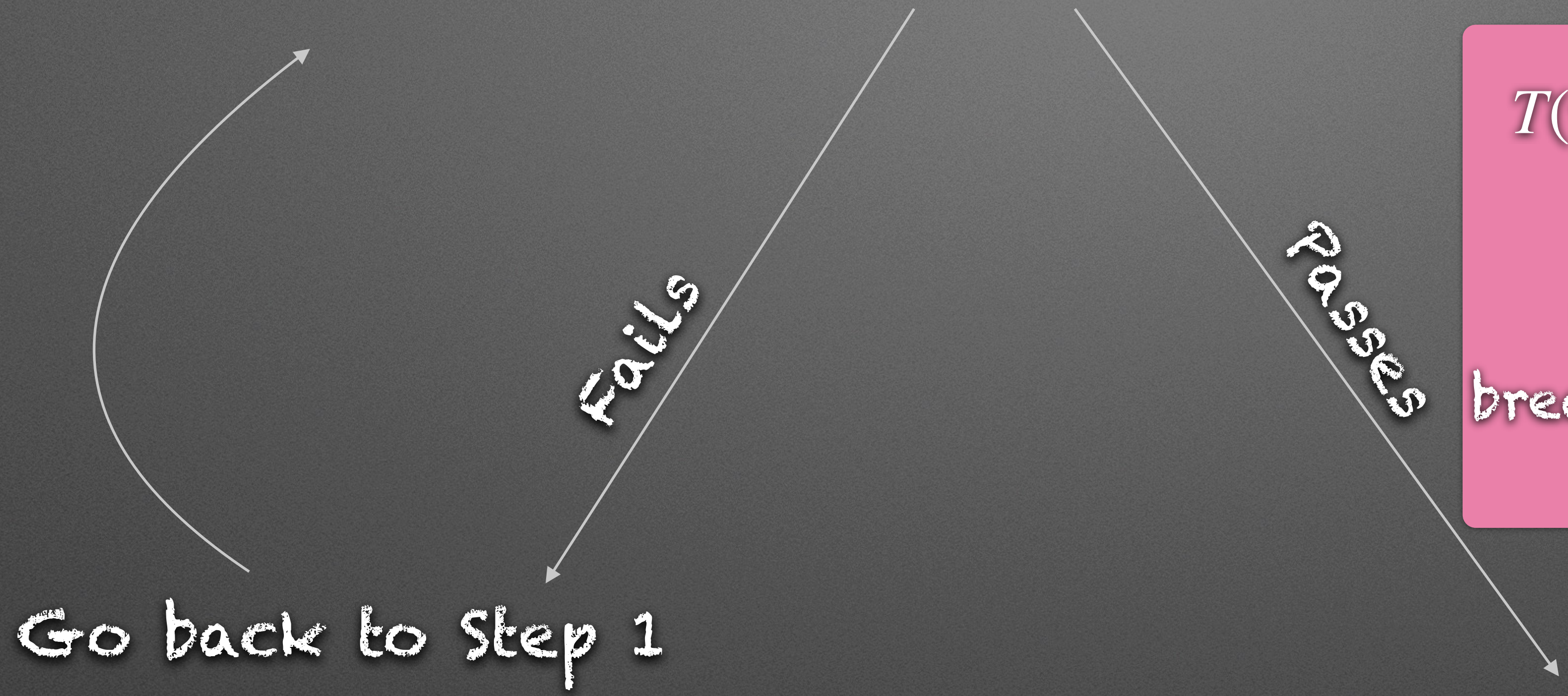
Guess a $T : V \to [k+1]$,

and check if it satisfies the discrete average property.

$$T(v) = \lfloor \frac{|T(v^+)| + |T(v^-)|}{2} \rfloor + \varepsilon$$

such that $\varepsilon = 0, 1$, or $*$,

($*$ denotes that the tie-breaking advantage is needed)

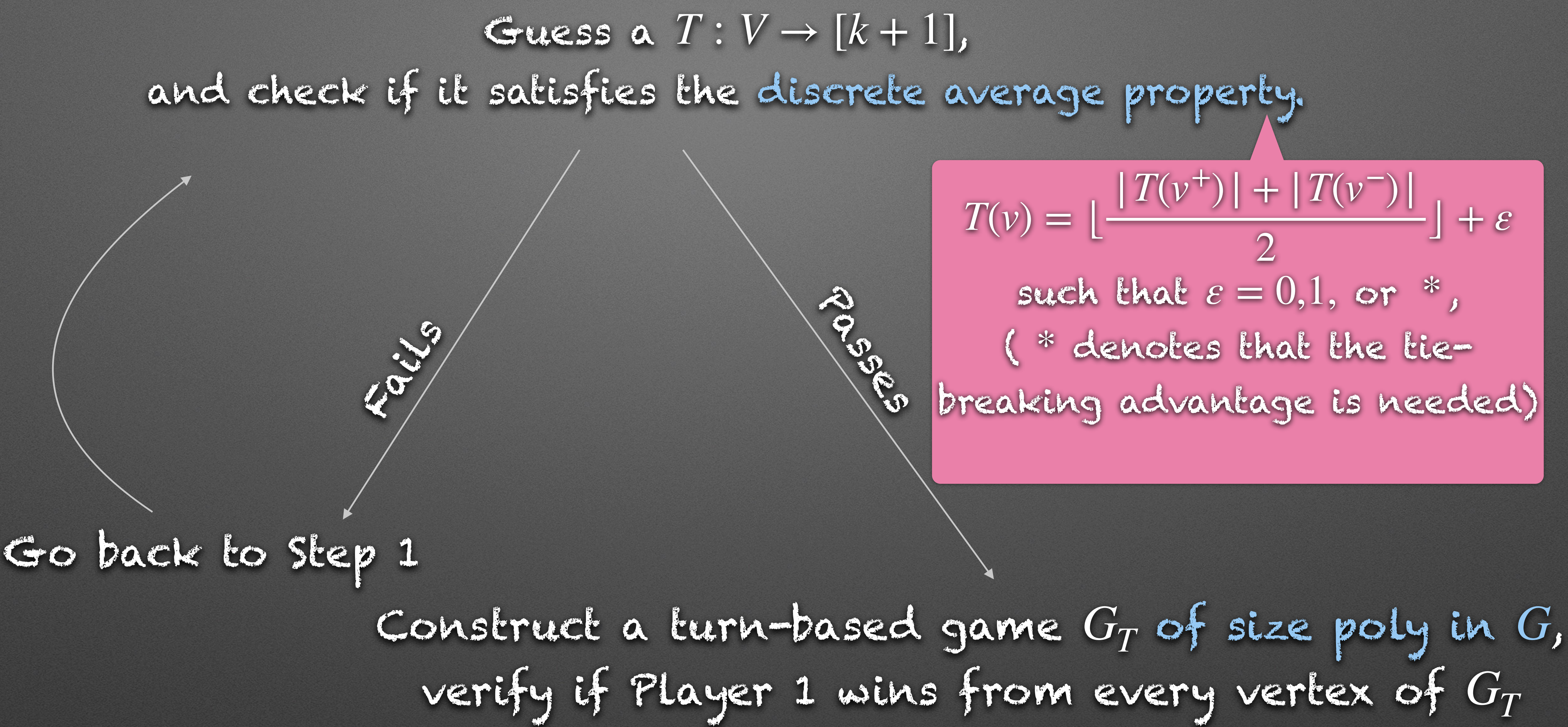


Fails

Passes

Go back to Step 1

Computing Threshold budgets for Parity Games



Guess a $T : V \rightarrow [k+1]$,

and check if it satisfies the discrete average property.

$$T(v) = \lfloor \frac{|T(v^+)| + |T(v^-)|}{2} \rfloor + \varepsilon$$

such that $\varepsilon = 0,1,$ or $*$,

($*$ denotes that the tie-breaking advantage is needed)

Fails

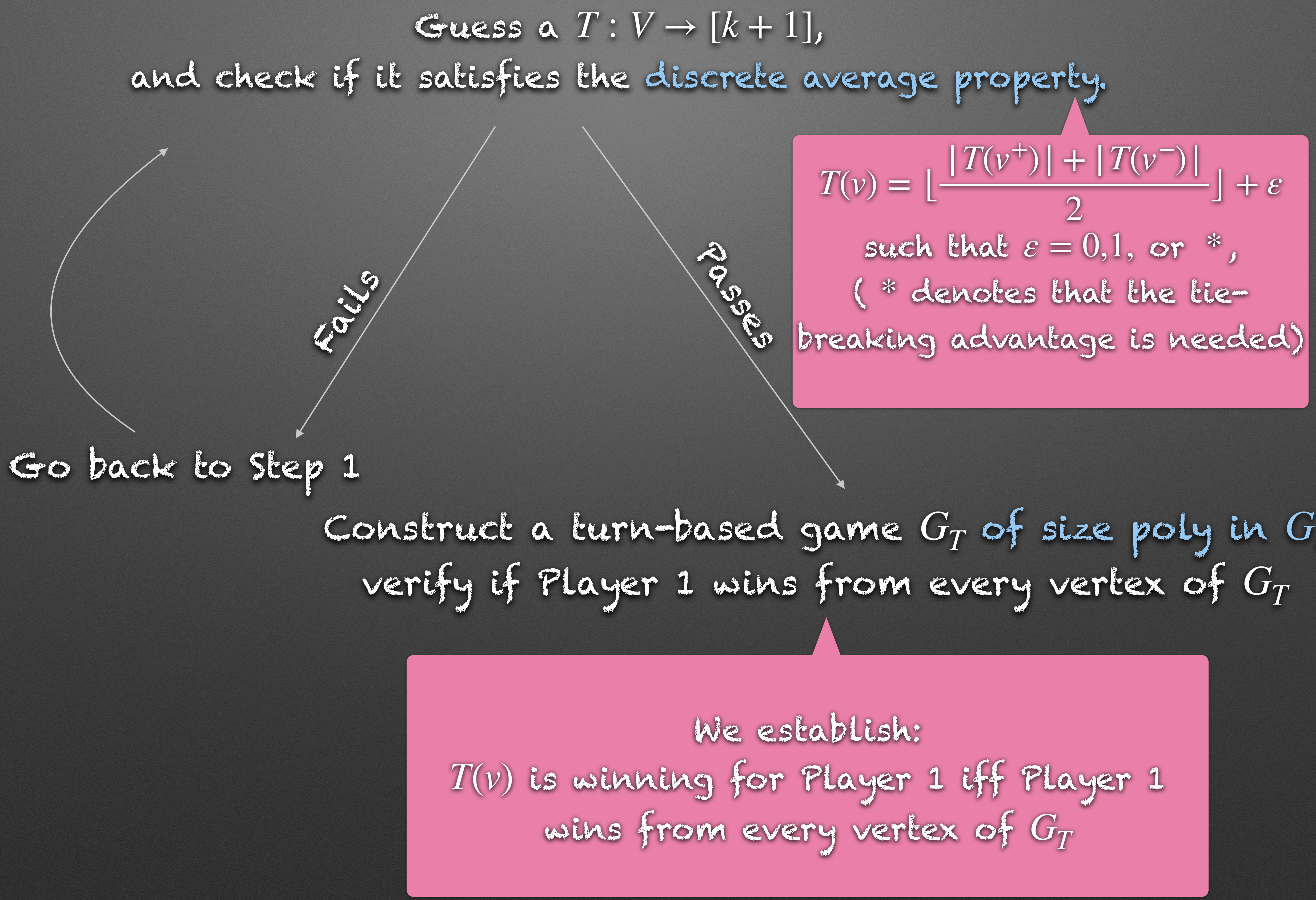
Go back to Step 1

Passes

Construct a turn-based game G_T of size poly in G,

verify if Player 1 wins from every vertex of G_T

Computing Threshold budgets for Parity Games



Guess a $T : V \rightarrow [k+1]$,

and check if it satisfies the discrete average property.

$$T(v) = \lfloor \frac{|T(v^+)| + |T(v^-)|}{2} \rfloor + \varepsilon$$

such that $\varepsilon = 0{,}1$, or $*$,

($*$ denotes that the tie-breaking advantage is needed)

Fails

Go back to Step 1

Passes

Construct a turn-based game G_T of size poly in G,

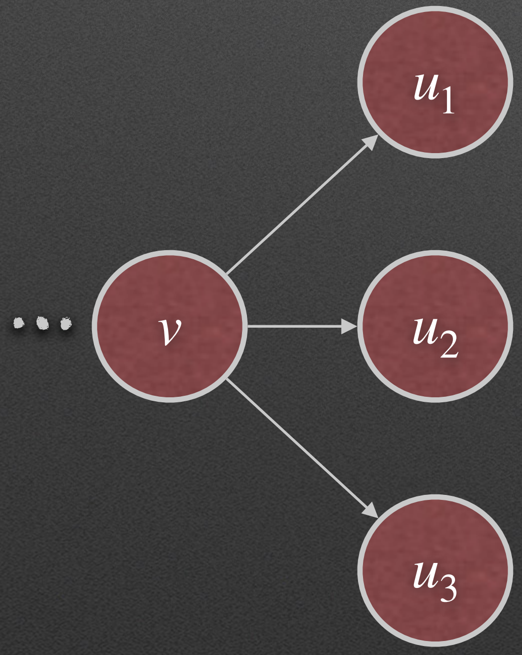
verify if Player 1 wins from every vertex of G_T

We establish:

$T(v)$ is winning for Player 1 iff Player 1 wins from every vertex of G_T

From bidding to turn-based games

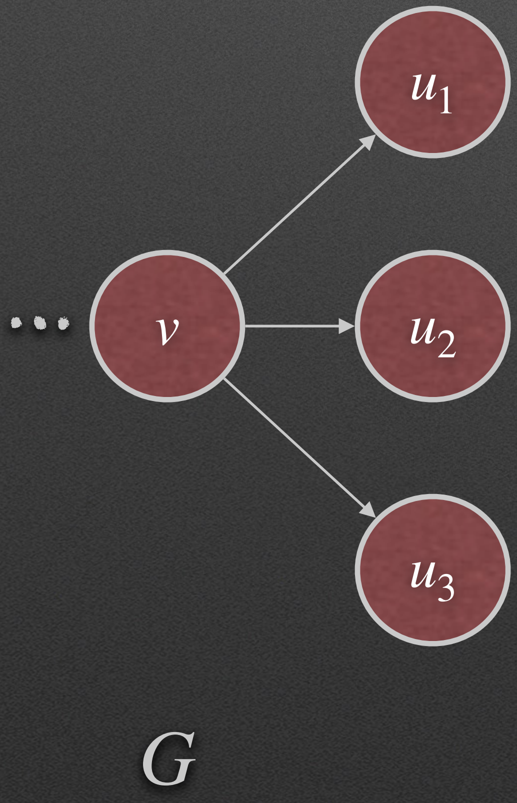
From bidding to turn-based games



G

From bidding to turn-based games

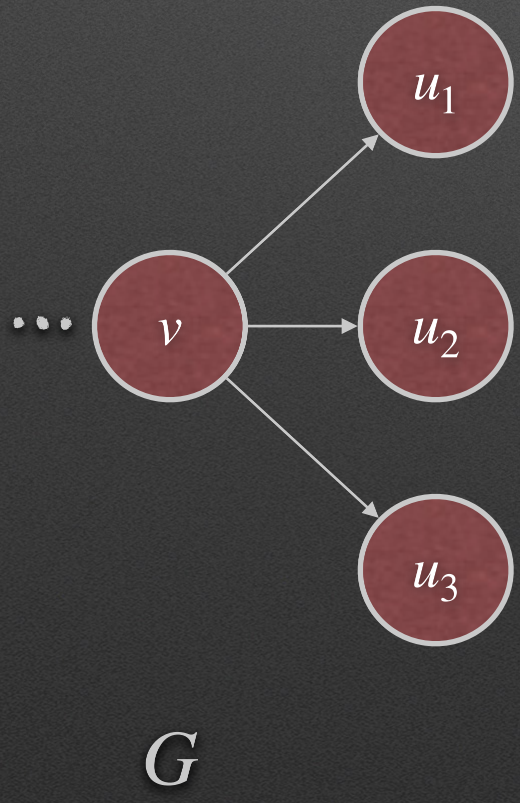
Vertices: $\langle v, T(v)\rangle$, $\langle v, T(v) \oplus 1\rangle$, $\langle v, \top \rangle$, $\langle v, T(v)\rangle$



From bidding to turn-based games

Vertices: $\langle v, T(v) \rangle$, $\langle v, T(v) \oplus 1 \rangle$, $\langle v, \top \rangle$, $\langle v, T(v) \rangle$

T gives rise to Player 1's bid: $b_T(v) \approx \dfrac{T(v^+) - T(v^-)}{2}$

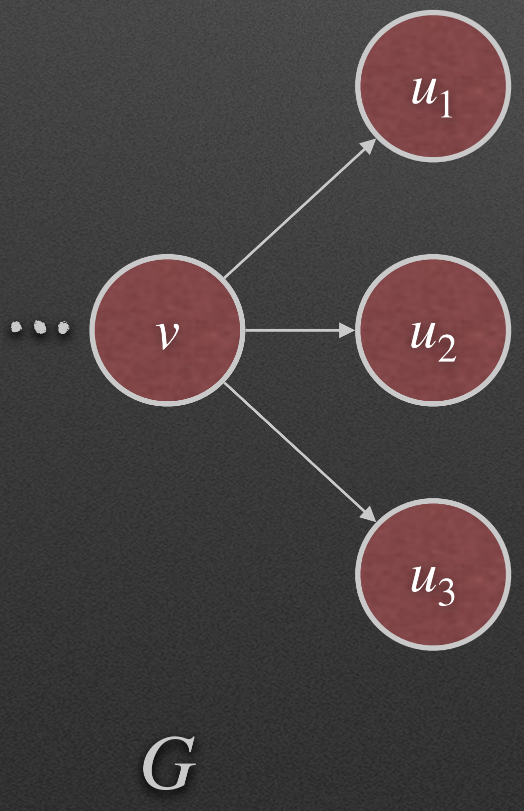


From bidding to turn-based games

Vertices: $\langle v, T(v) \rangle$, $\langle v, T(v) \oplus 1 \rangle$, $\langle v, \top \rangle$, $\langle v, T(v) \rangle$

T gives rise to Player 1's bid: $b_T(v) \approx \dfrac{T(v^+) - T(v^-)}{2}$

Player 2's two optimal responses: 0 or $b_T(v) \oplus 1$

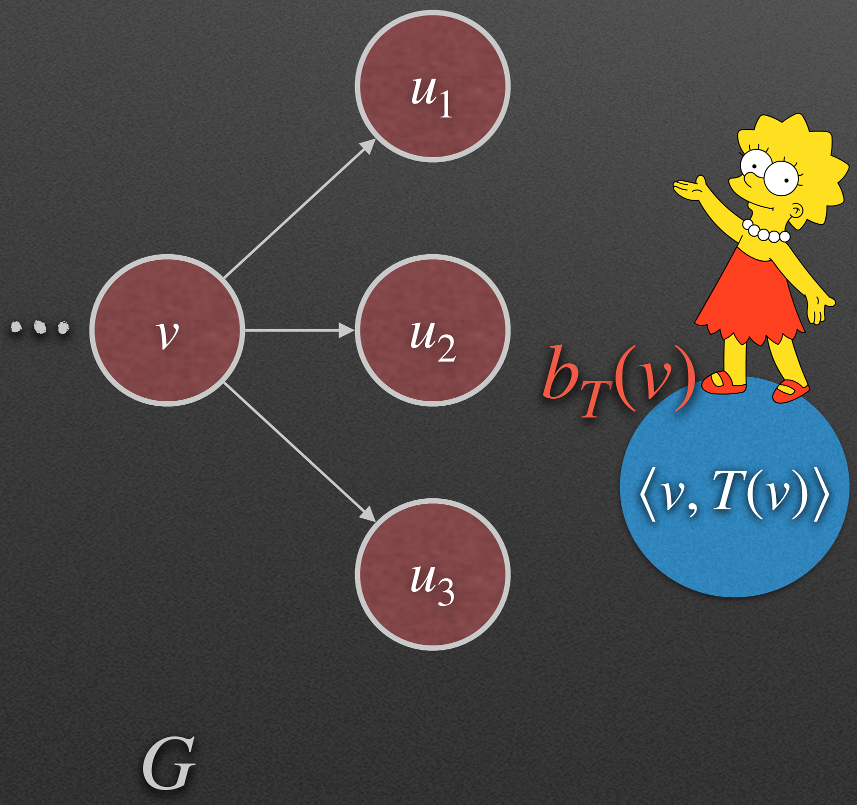


From bidding to turn-based games

Vertices: $\langle v, T(v)\rangle$, $\langle v, T(v) \oplus 1\rangle$, $\langle v, \mathrm{T} \rangle$, $\langle v, T(v)\rangle$

T gives rise to Player 1's bid: $b_T(v) \approx \dfrac{T(v^+) - T(v^-)}{2}$

Player 2's two optimal responses: 0 or $b_T(v) \oplus 1$

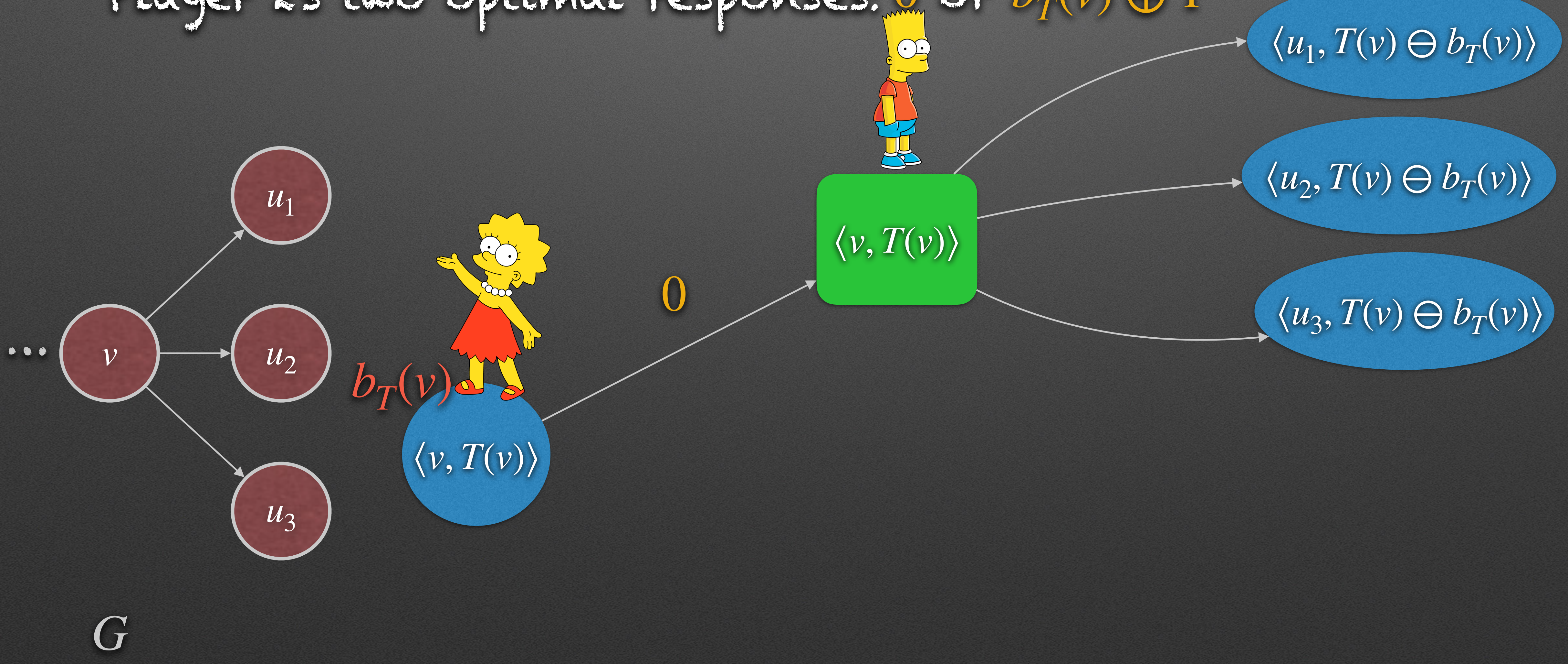


From bidding to turn-based games

Vertices: $\langle v, T(v) \rangle$, $\langle v, T(v) \oplus 1 \rangle$, $\langle v, \mathrm{T} \ \rangle$, $\langle v, T(v) \rangle$

T gives rise to Player 1's bid: $b_T(v) \approx \dfrac{T(v^+) - T(v^-)}{2}$

Player 2's two optimal responses: 0 or $b_T(v) \oplus 1$


$\langle u_1, T(v) \ominus b_T(v) \rangle$
$\langle u_2, T(v) \ominus b_T(v) \rangle$
$\langle u_3, T(v) \ominus b_T(v) \rangle$
u_1
$\cdots$
v
u_2
u_3
$\langle v, T(v) \rangle$
0
$b_T(v)$
$\langle v, T(v) \rangle$
G

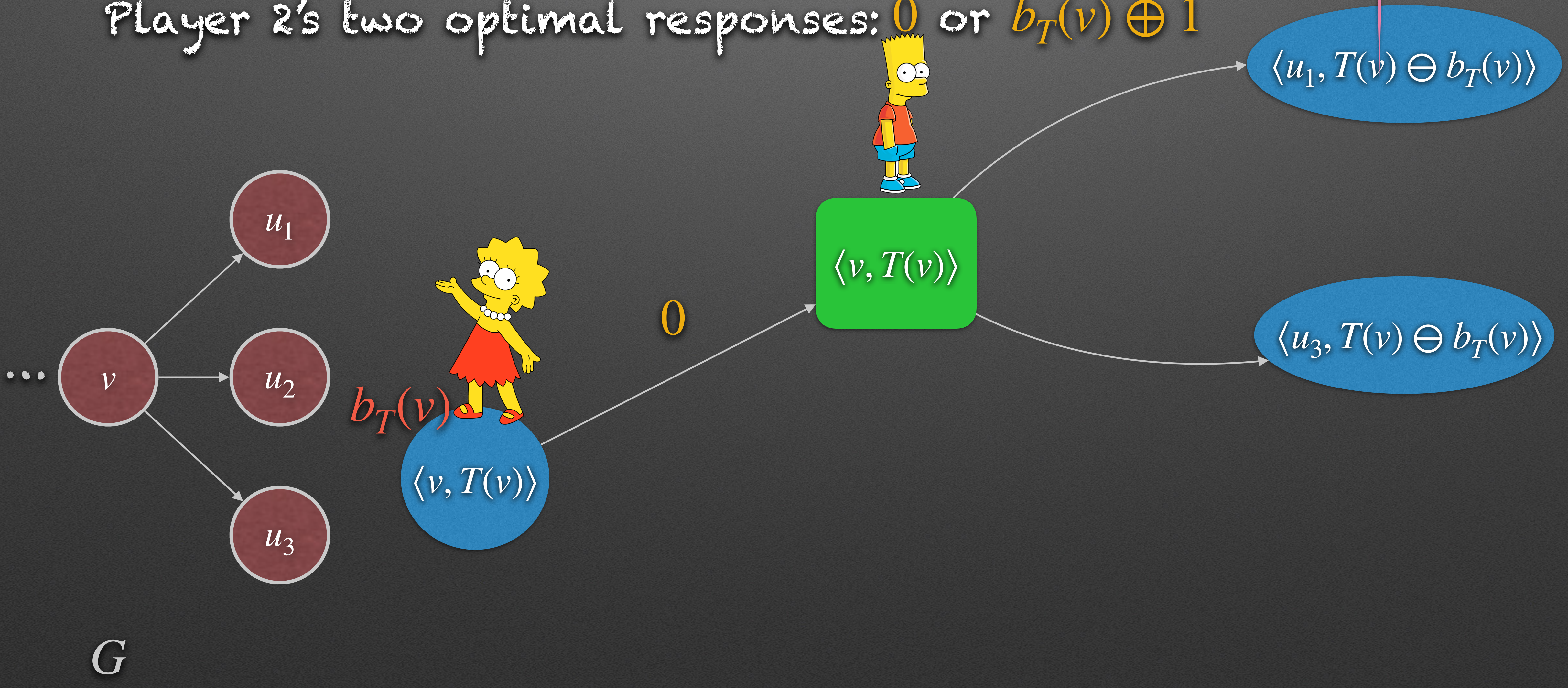
From bidding to turn-based games

Vertices: $\langle v, T(v) \rangle$, $\langle v, T(v) \oplus 1 \rangle$, $\langle v, \mathrm{T} \rangle$, $\langle v, T(v) \rangle$

T gives rise to Player 1's bid: $b_T(v) \approx \dfrac{T(v^+) - T(v^-)}{2}$

Player 2's two optimal responses: 0 or $b_T(v) \oplus 1$

Keep only u_i's such that these budgets are in $\{T(u_i), T(u_i) \oplus 1\}$



...
v
u_1
u_2
u_3
$b_T(v)$
$\langle v, T(v) \rangle$
0
$\langle v, T(v) \rangle$
$\langle u_1, T(v) \ominus b_T(v) \rangle$
$\langle u_3, T(v) \ominus b_T(v) \rangle$
G

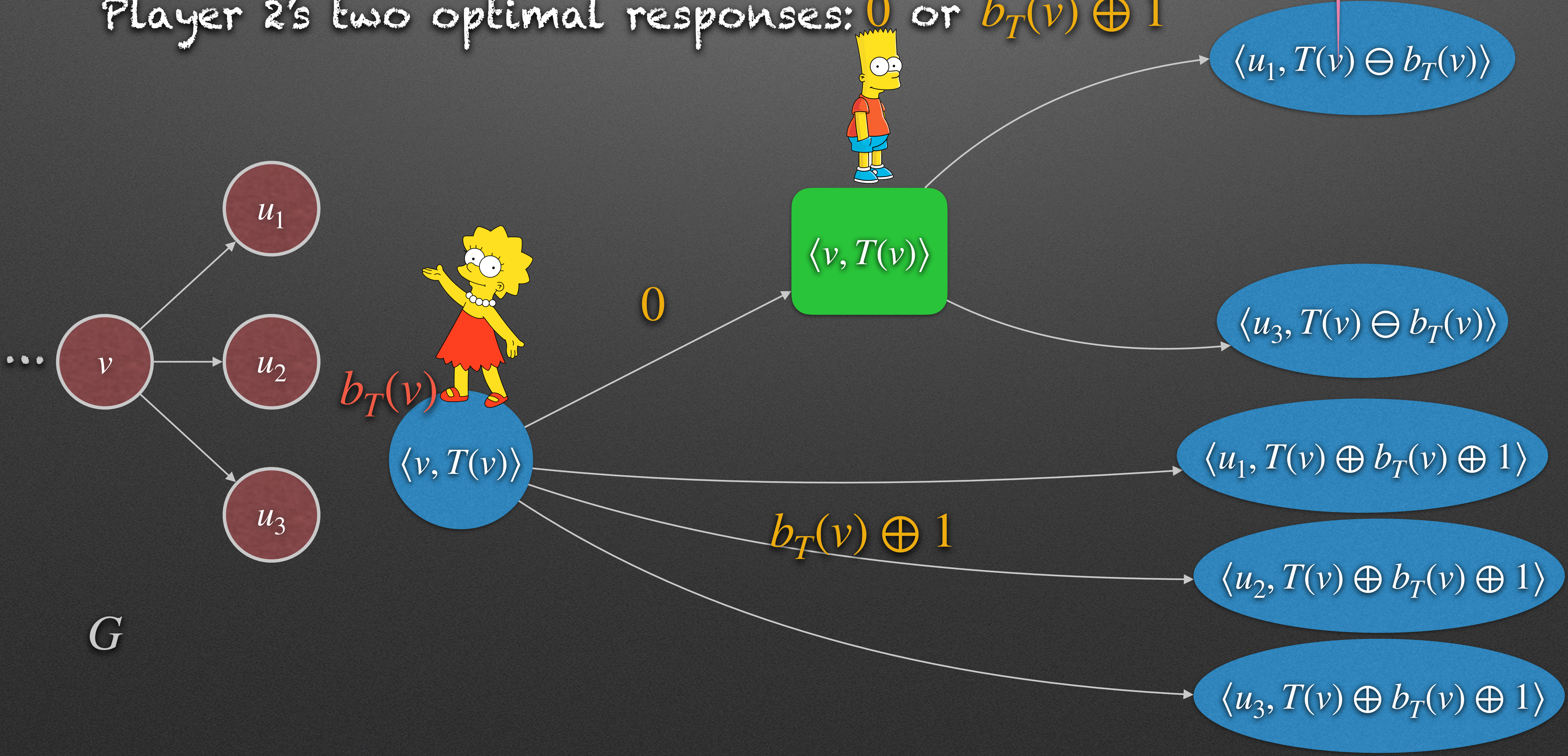
From bidding to turn-based games

Vertices: $\langle v, T(v)\rangle$, $\langle v, T(v)\oplus 1\rangle$, $\langle v, \mathsf{T} \rangle$, $\langle v, T(v)\rangle$

T gives rise to Player 1's bid: $b_T(v) \approx \dfrac{T(v^+) - T(v^-)}{2}$

Player 2's two optimal responses: 0 or $b_T(v) \oplus 1$

Keep only u_i's such that these budgets are in $\{T(u_i), T(u_i) \oplus 1\}$



...
v
u_1
u_2
u_3
G
$b_T(v)$
$\langle v, T(v)\rangle$
0
$\langle v, T(v)\rangle$
$b_T(v) \oplus 1$
$\langle u_1, T(v) \ominus b_T(v)\rangle$
$\langle u_3, T(v) \ominus b_T(v)\rangle$
$\langle u_1, T(v) \oplus b_T(v) \oplus 1\rangle$
$\langle u_2, T(v) \oplus b_T(v) \oplus 1\rangle$
$\langle u_3, T(v) \oplus b_T(v) \oplus 1\rangle$

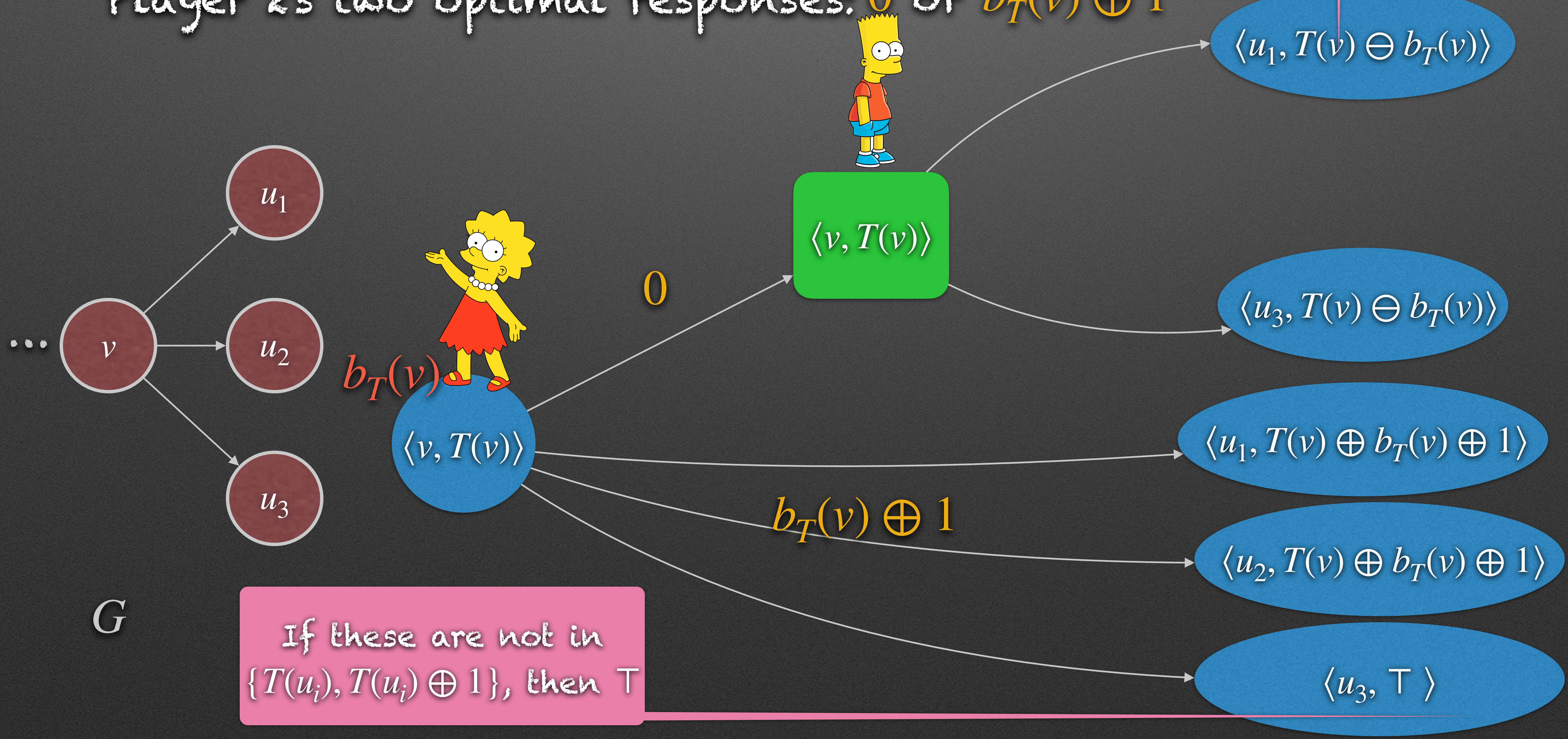
From bidding to turn-based games

Vertices: $\langle v, T(v)\rangle$, $\langle v, T(v) \oplus 1\rangle$, $\langle v, \top \rangle$, $\langle v, T(v)\rangle$

Keep only u_i's such that these budgets are in $\{T(u_i), T(u_i) \oplus 1\}$

T gives rise to Player 1's bid: $b_T(v) \approx \dfrac{T(v^+) - T(v^-)}{2}$

Player 2's two optimal responses: 0 or $b_T(v) \oplus 1$



$\langle u_1, T(v) \ominus b_T(v)\rangle$
u_1
$\langle v, T(v)\rangle$
0
$\langle u_3, T(v) \ominus b_T(v)\rangle$
…
v
u_2
$b_T(v)$
$\langle v, T(v)\rangle$
$\langle u_1, T(v) \oplus b_T(v) \oplus 1\rangle$
u_3
$b_T(v) \oplus 1$
$\langle u_2, T(v) \oplus b_T(v) \oplus 1\rangle$
G
$\langle u_3, \top \rangle$

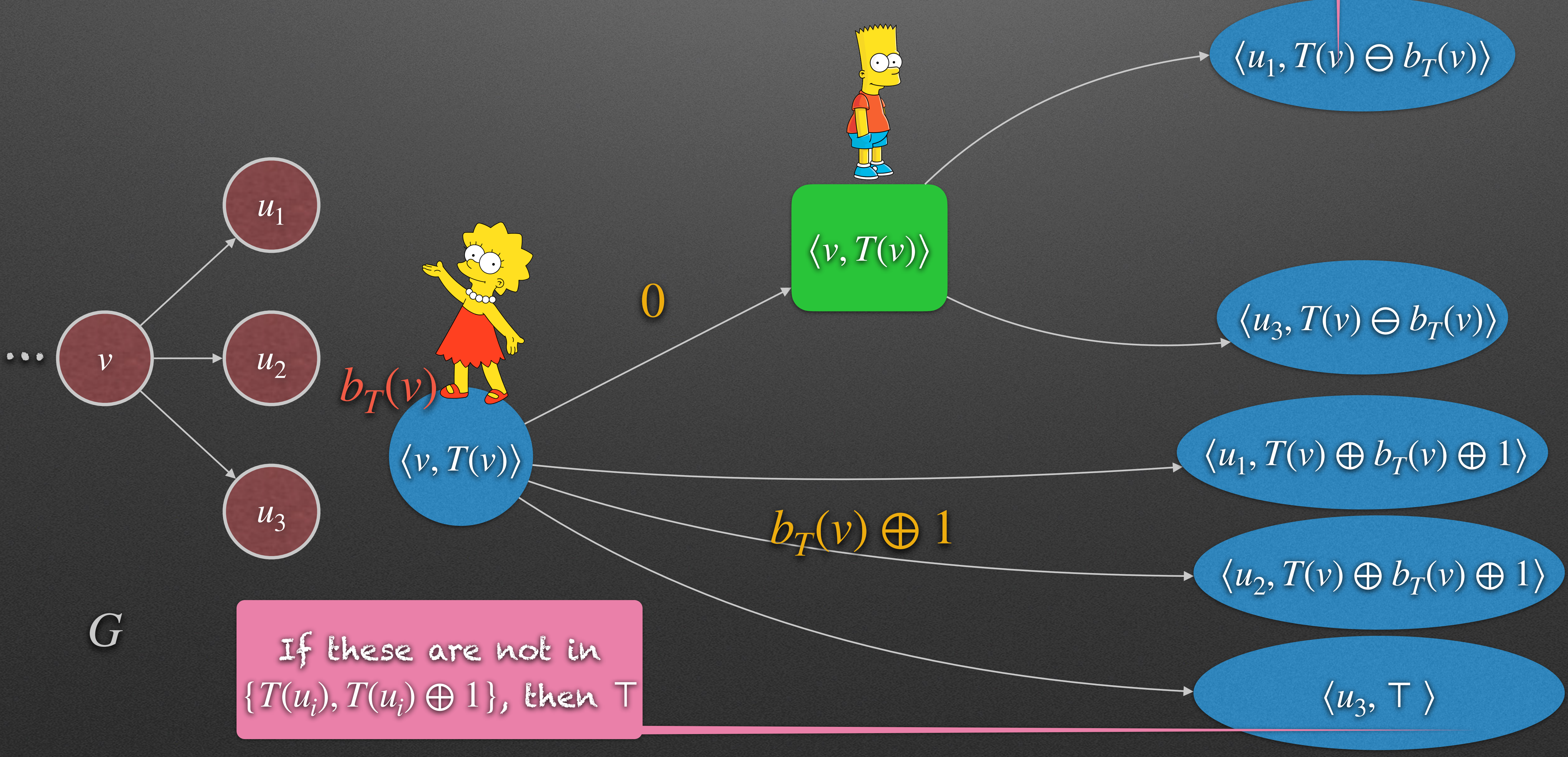
If these are not in $\{T(u_i), T(u_i) \oplus 1\}$, then $\top$

From bidding to turn-based games

Theorem: [Avni & S.]

1. If Player 1 wins from every vertex, then $T \geq Th_G$
2. If $T = Th_G$, then Player 1 wins from each vertex of G_T

Keep only u_i's such that these budgets are in $\{T(u_i), T(u_i) \oplus 1\}$



...
v
u_1
u_2
u_3
$b_T(v)$
$\langle v, T(v) \rangle$
0
$\langle v, T(v) \rangle$
$\langle u_1, T(v) \ominus b_T(v) \rangle$
$\langle u_3, T(v) \ominus b_T(v) \rangle$
$\langle u_1, T(v) \oplus b_T(v) \oplus 1 \rangle$
$b_T(v) \oplus 1$
$\langle u_2, T(v) \oplus b_T(v) \oplus 1 \rangle$
$\langle u_3, \top \rangle$
G

If these are not in $\{T(u_i), T(u_i) \oplus 1\}$, then ⊤

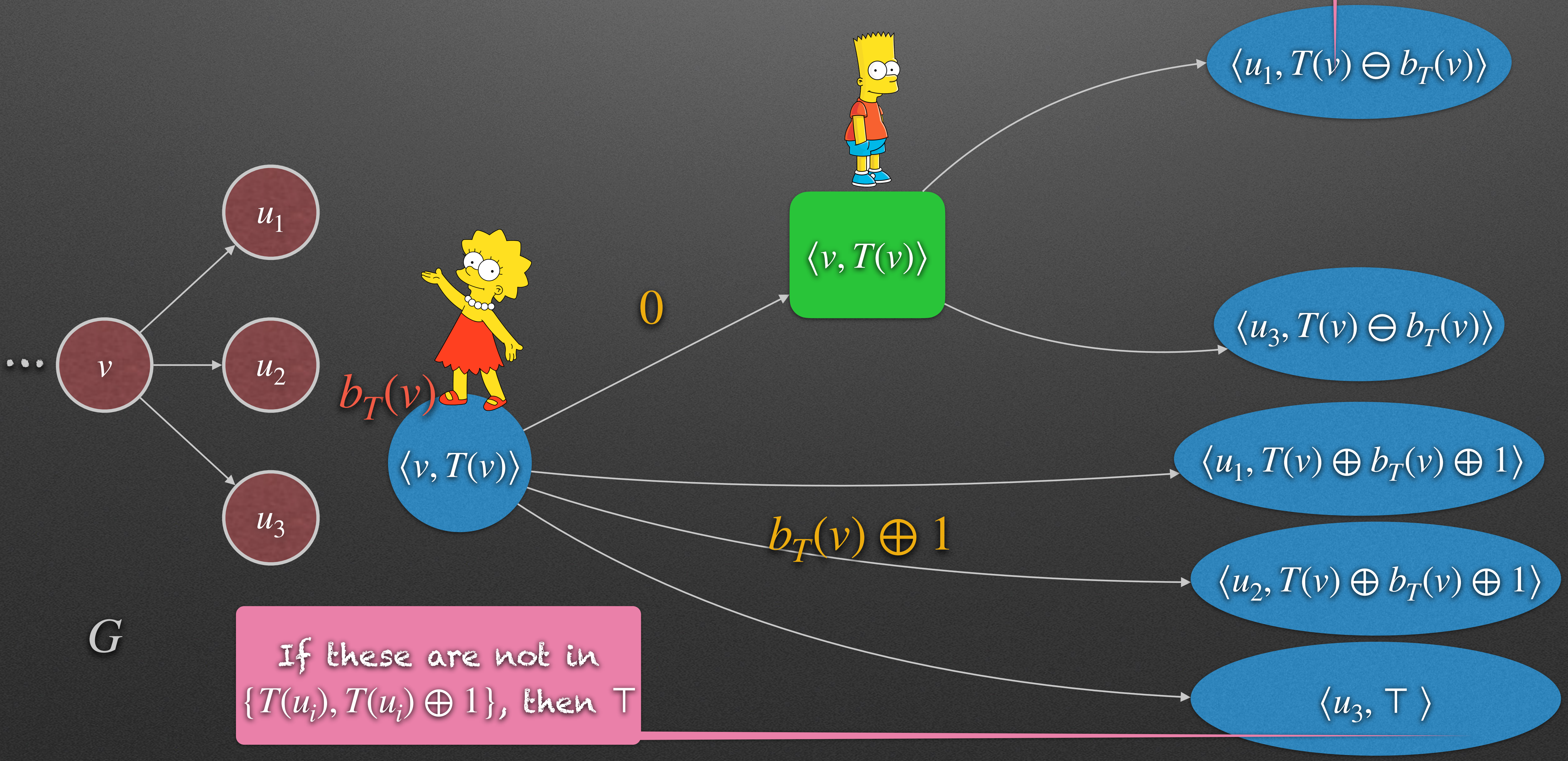
From bidding to turn-based games

Theorem: [Avni & S.]

1. If Player 1 wins from every vertex, then $T \geq Th_G$
2. If $T = Th_G$, then Player 1 wins from each vertex of G_T

Repeat the same with respect to Player 2 (Lisa)

such that ts are in $) \oplus 1\}$


$\langle u_1, T(v) \ominus b_T(v) \rangle$
$\langle v, T(v) \rangle$
0
$\langle u_3, T(v) \ominus b_T(v) \rangle$
u_1
…
v
u_2
$b_T(v)$
$\langle v, T(v) \rangle$
$\langle u_1, T(v) \oplus b_T(v) \oplus 1 \rangle$
u_3
$b_T(v) \oplus 1$
$\langle u_2, T(v) \oplus b_T(v) \oplus 1 \rangle$
G
$\langle u_3, \top \rangle$

If these are not in $\{T(u_i), T(u_i) \oplus 1\}$, then $\top$

From bidding to turn-based games

<u>Theorem: [Avni & S.]</u>

1. If Player 1 wins from every vertex, then $T \geq Th_G$
2. If $T = Th_G$, then Player 1 wins from each vertex of G_T

<u>Remark:</u>

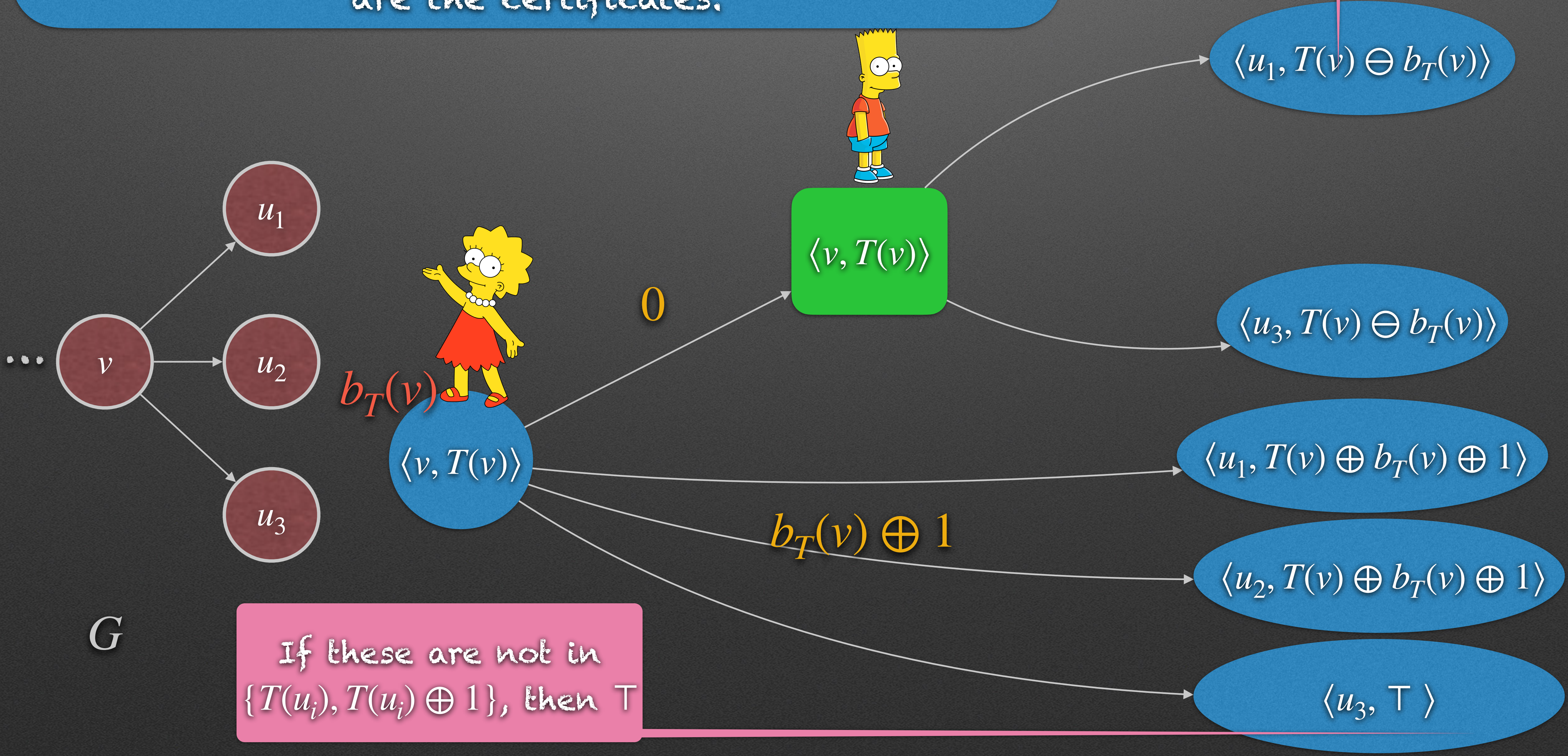
Both $T : V \to [k+1]$ and the winning strategy of G_T are the certificates.

such that

ts are in

$) \oplus 1\}$

Repeat the same with respect to Player 2 (Lisa)



$\langle u_1, T(v) \ominus b_T(v) \rangle$
u_1
$\langle v, T(v) \rangle$
0
$\langle u_3, T(v) \ominus b_T(v) \rangle$
...
v
u_2
$b_T(v)$
$\langle v, T(v) \rangle$
$\langle u_1, T(v) \oplus b_T(v) \oplus 1 \rangle$
u_3
$b_T(v) \oplus 1$
$\langle u_2, T(v) \oplus b_T(v) \oplus 1 \rangle$
G
$\langle u_3, \top \rangle$

If these are not in $\{T(u_i), T(u_i) \oplus 1\}$, then ⊤

From bidding to turn-based games

Theorem: [Avni & S.]

1. If Player 1 wins from every vertex, then $T \geq Th_G$
2. If $T = Th_G$, then Player 1 wins from each vertex of G_T

Repeat the same with respect to Player 2 (Lisa)

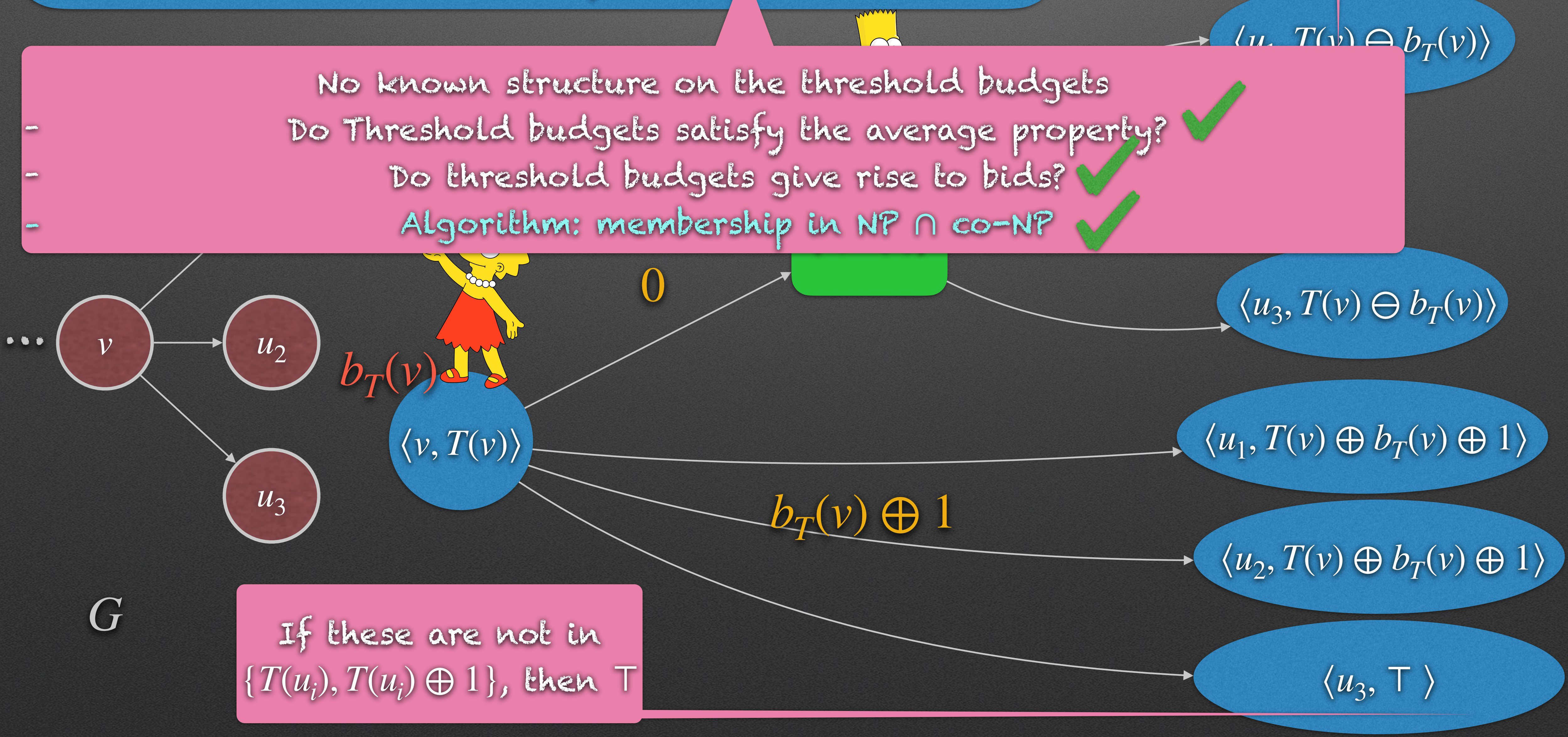
such that
ts are in
$) \oplus 1\}$

Remark:

Both $T : V \to [k+1]$ and the winning strategy of G_T are the certificates.

No known structure on the threshold budgets

- Do Threshold budgets satisfy the average property? ✓
- Do threshold budgets give rise to bids? ✓
- Algorithm: membership in NP ∩ co-NP ✓



$\langle u_1, T(v) \ominus b_T(v)\rangle$
…
v
u_2
u_3
$b_T(v)$
$\langle v, T(v)\rangle$
0
$\langle u_3, T(v) \ominus b_T(v)\rangle$
$\langle u_1, T(v) \oplus b_T(v) \oplus 1\rangle$
$b_T(v) \oplus 1$
$\langle u_2, T(v) \oplus b_T(v) \oplus 1\rangle$
$\langle u_3, \top \rangle$
G

If these are not in $\{T(u_i), T(u_i) \oplus 1\}$, then ⊤

From bidding to turn-based games

Theorem: [Avni & S.]

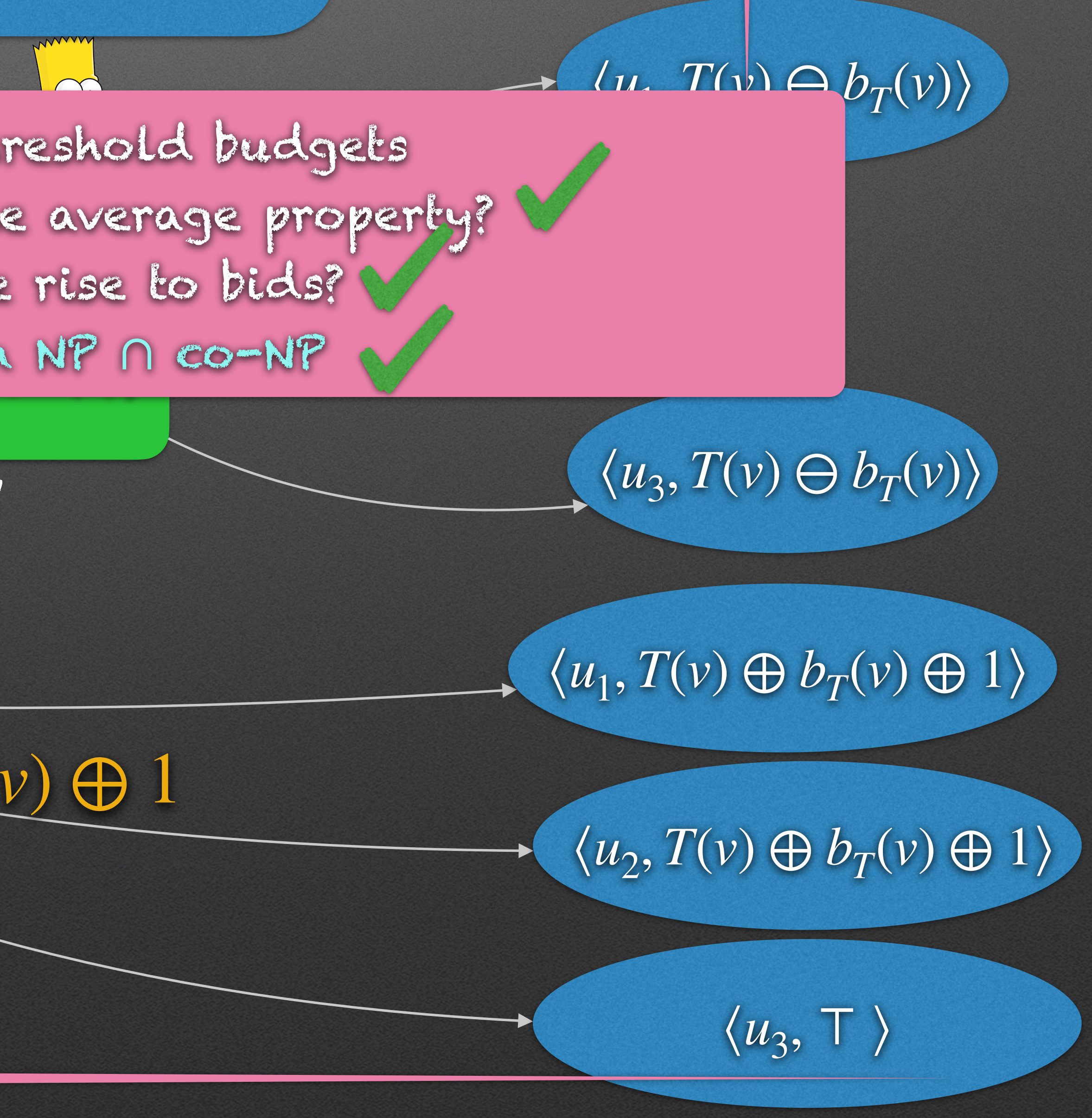
1. If Player 1 wins from every vertex, then $T \geq Th_G$
2. If $T = Th_G$, then Player 1 wins from each vertex of G_T

Repeat the same with respect to Player 2 (Lisa)

such that
ts are in
$) \oplus 1\}$

Remark:

Both $T : V \to [k+1]$ and the winning strategy of G_T are the certificates.



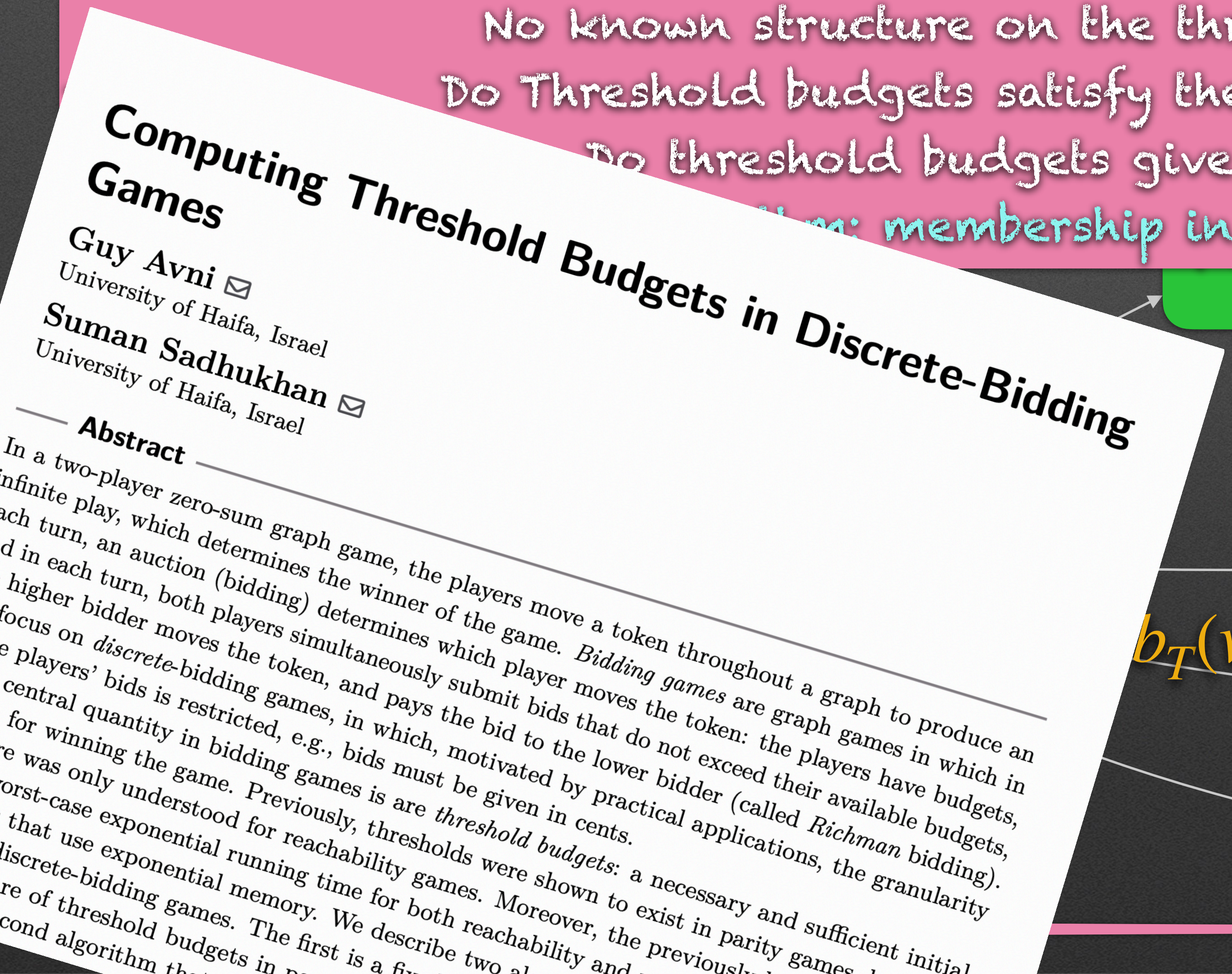
$\langle u_1, T(v) \ominus b_T(v) \rangle$
$\langle u_3, T(v) \ominus b_T(v) \rangle$
$\langle u_1, T(v) \oplus b_T(v) \oplus 1 \rangle$
$b_T(v) \oplus 1$
$\langle u_2, T(v) \oplus b_T(v) \oplus 1 \rangle$
$\langle u_3, \top \rangle$

No known structure on the threshold budgets

Do Threshold budgets satisfy the average property? ✓

Do threshold budgets give rise to bids? ✓

m: membership in NP ∩ co-NP ✓



Computing Threshold Budgets in Discrete-Bidding Games

Guy Avni ✉
University of Haifa, Israel

Suman Sadhukhan ✉
University of Haifa, Israel

Abstract

In a two-player zero-sum graph game, the players move a token throughout a graph to produce an infinite play, which determines the winner of the game. *Bidding games* are graph games in which in ach turn, an auction (bidding) determines which player moves the token: the players have budgets, d in each turn, both players simultaneously submit bids that do not exceed their available budgets, higher bidder moves the token, and pays the bid to the lower bidder (called *Richman* bidding). focus on *discrete*-bidding games, in which, motivated by practical applications, the granularity e players' bids is restricted, e.g., bids must be given in cents. central quantity in bidding games is are *threshold budgets*: a necessary and sufficient initial for winning the game. Previously, thresholds were shown to exist in parity games, b re was only understood for reachability games. Moreover, the previously orst-case exponential running time for both reachability and that use exponential memory. We describe two al iscrete-bidding games. The first is a fi re of threshold budgets in p cond algorithm th

Take Away - Part I

Theorem:

Finding Threshold budgets in parity discrete bidding games is NP ∩ co-NP.

Take Away - Part I

Theorem:

Finding Threshold budgets in parity discrete bidding games is NP ∩ co-NP.

Imrovement: Earlier only EXPTIME algorithm was known for discrete bidding (including Reachability)

Take Away - Part I

Theorem:
Finding Threshold budgets in parity discrete bidding games is NP ∩ co-NP.

Corollary:
Polynomial size winning strategies exist.

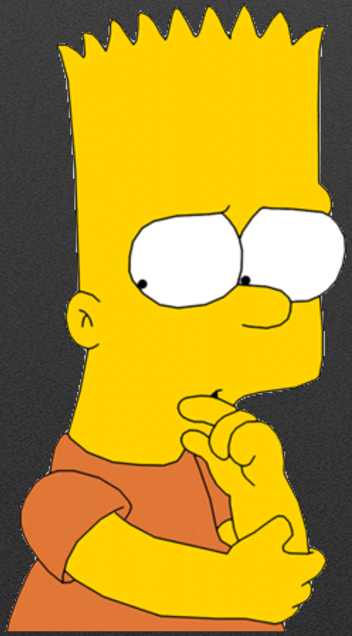
Imrovement: Earlier only EXPTIME algorithm was known for discrete bidding (including Reachability)

Take Away - Part I

Theorem:
Finding Threshold budgets in parity discrete bidding games is NP ∩ co-NP.

Corollary:
Polynomial size winning strategies exist.

Imrovement: Earlier only EXPTIME algorithm was known for discrete bidding (including Reachability)

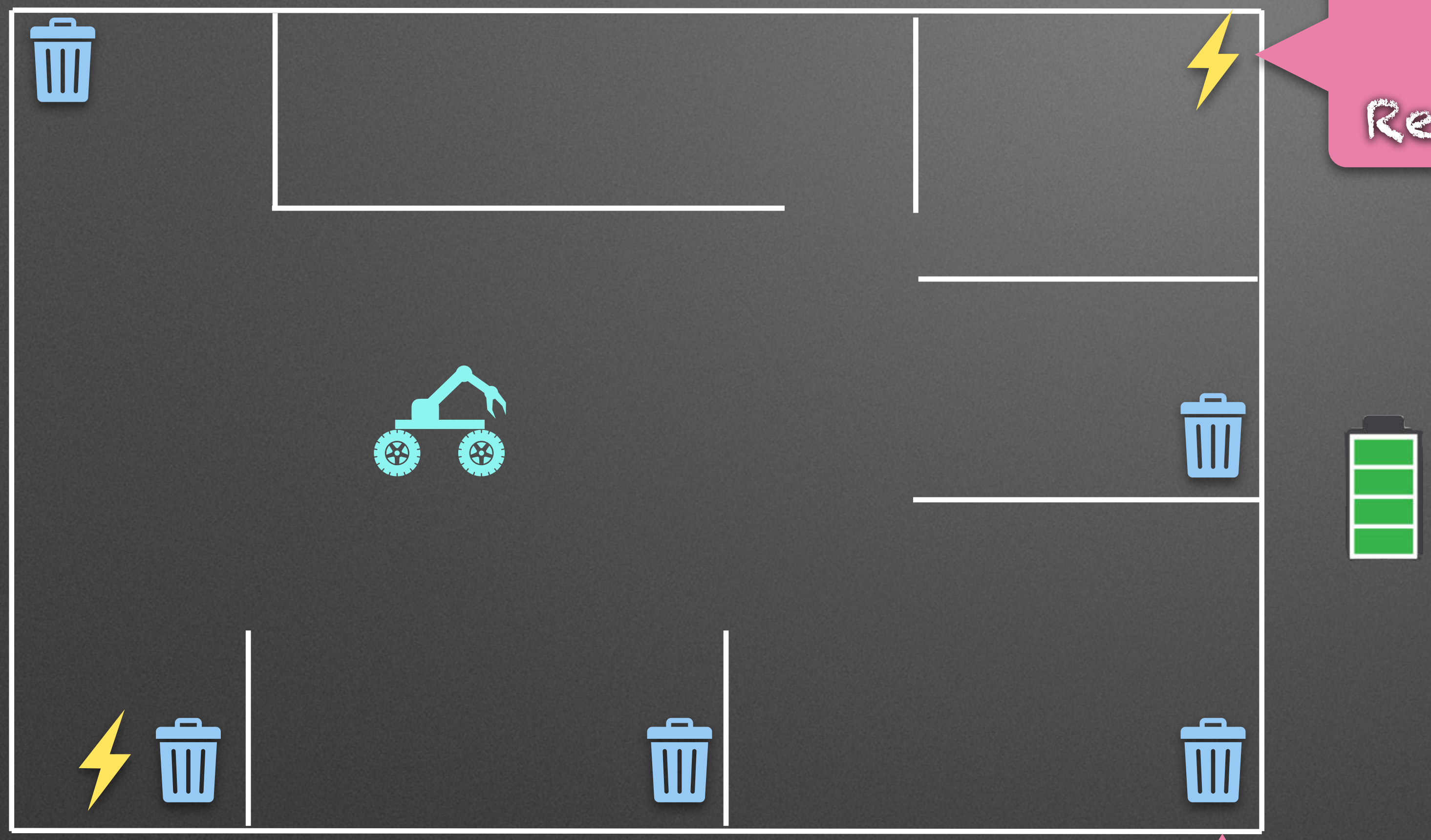


Food for thought?

- Turn-based parity games are in NP ∩ co-NP, but not known to be in P.
- Turn-based parity games -> discrete bidding games with fixed budgets
- Discrete bidding parity games with budgets in binary -> membership in NP ∩ co-NP

Part II (in Practice): Continuous Bidding Games in Multi-objective Decentralised Synthesis

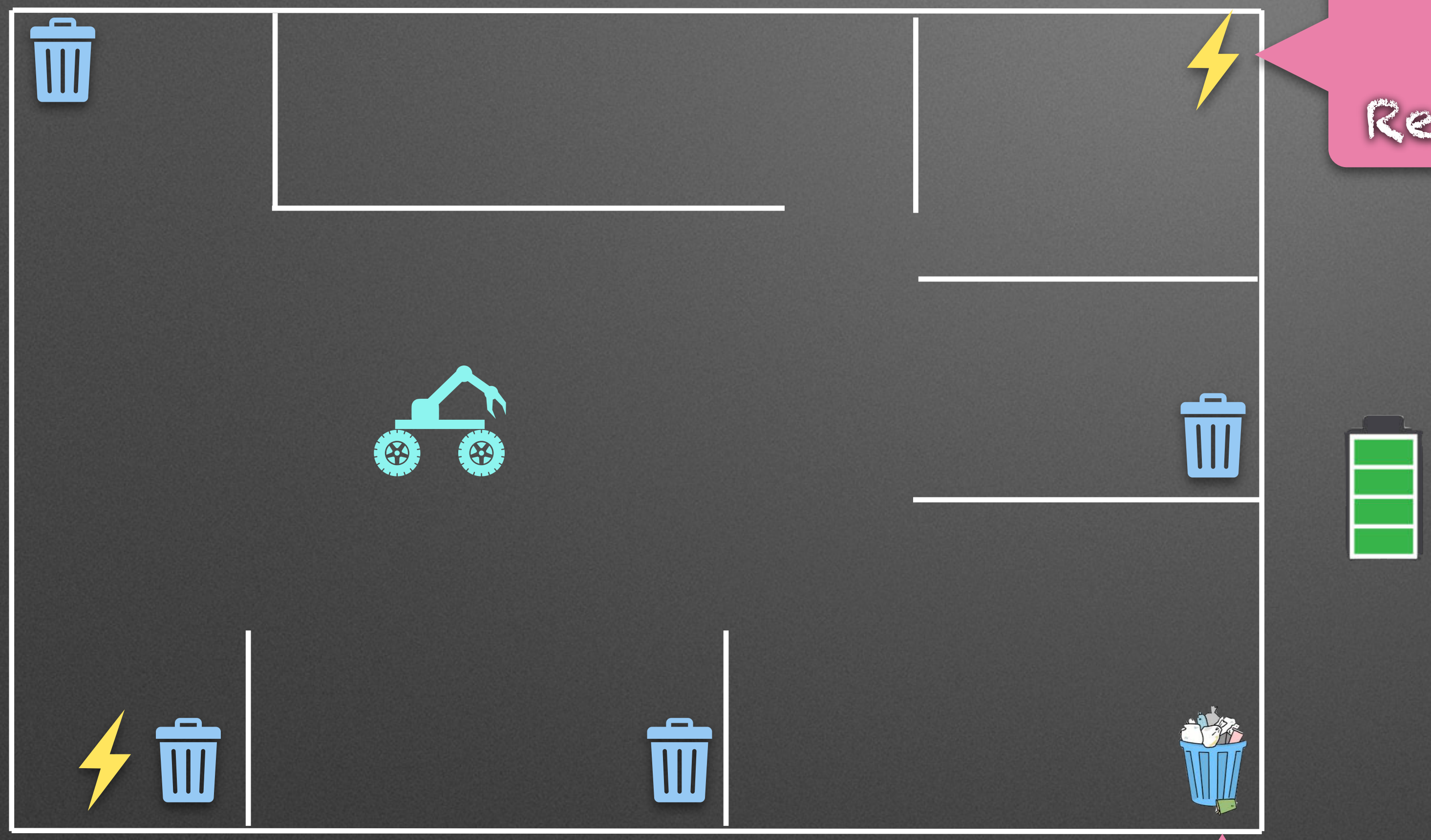
Multi-objective Control Problem



Objective 2 (ψ_2):
Recharge before battery runs out

Objective 1 (ψ_1):
Repeatedly empty all trash cans

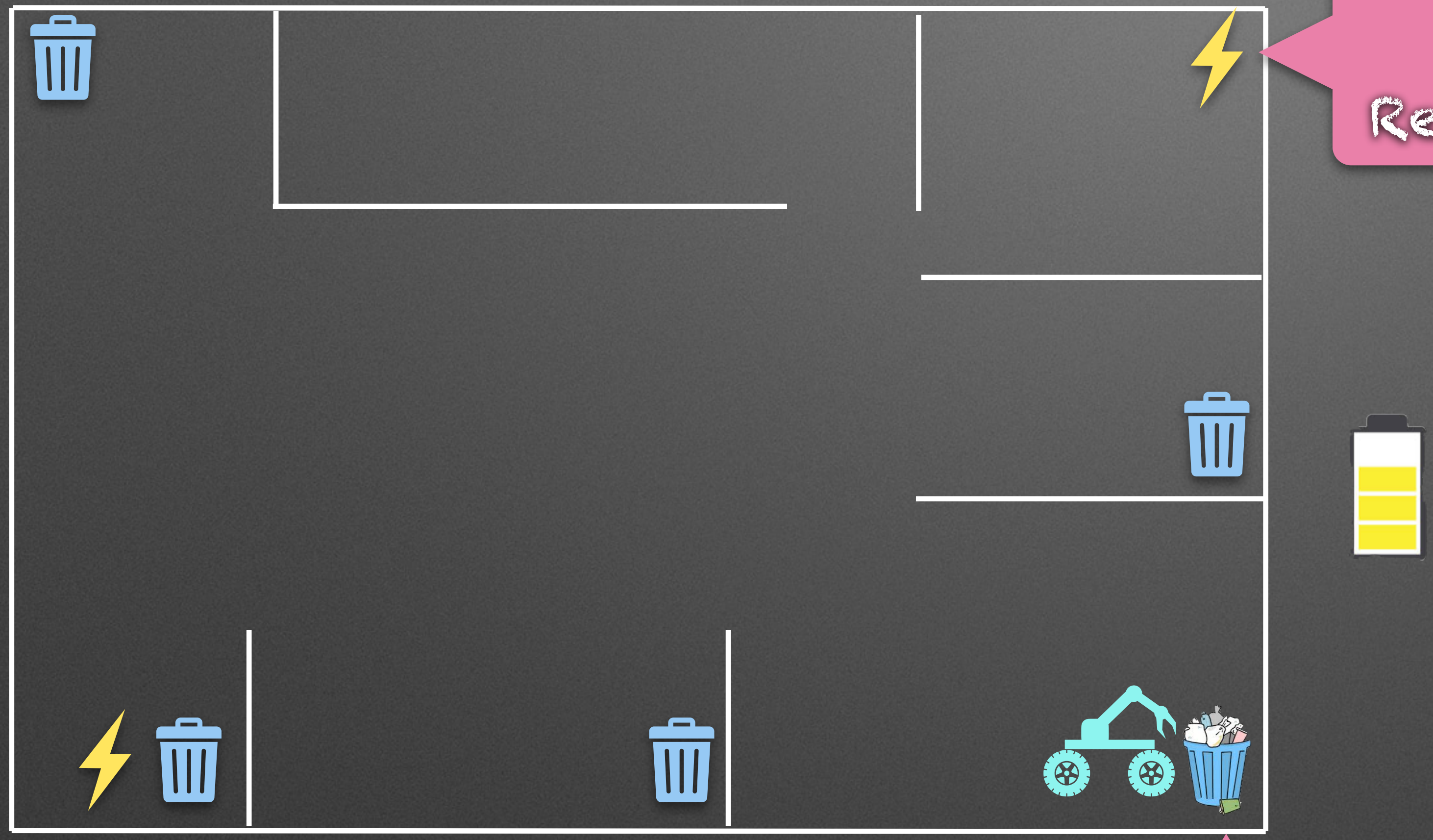
Multi-objective Control Problem



Objective 2 (ψ_2):
Recharge before battery runs out

Objective 1 (ψ_1):
Repeatedly empty all trash cans

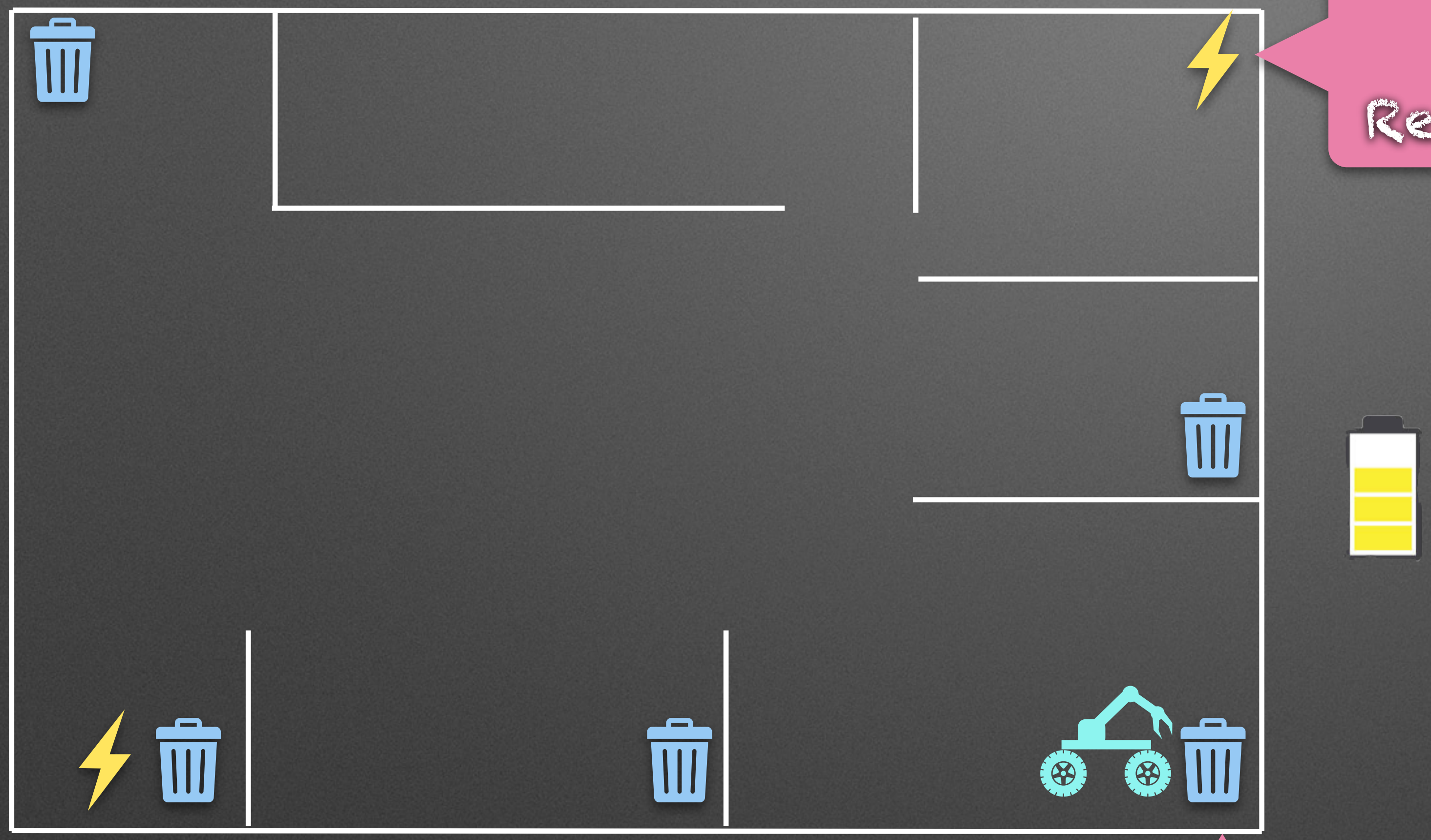
Multi-objective Control Problem



Objective 2 (ψ_2):
Recharge before battery runs out

Objective 1 (ψ_1):
Repeatedly empty all trash cans

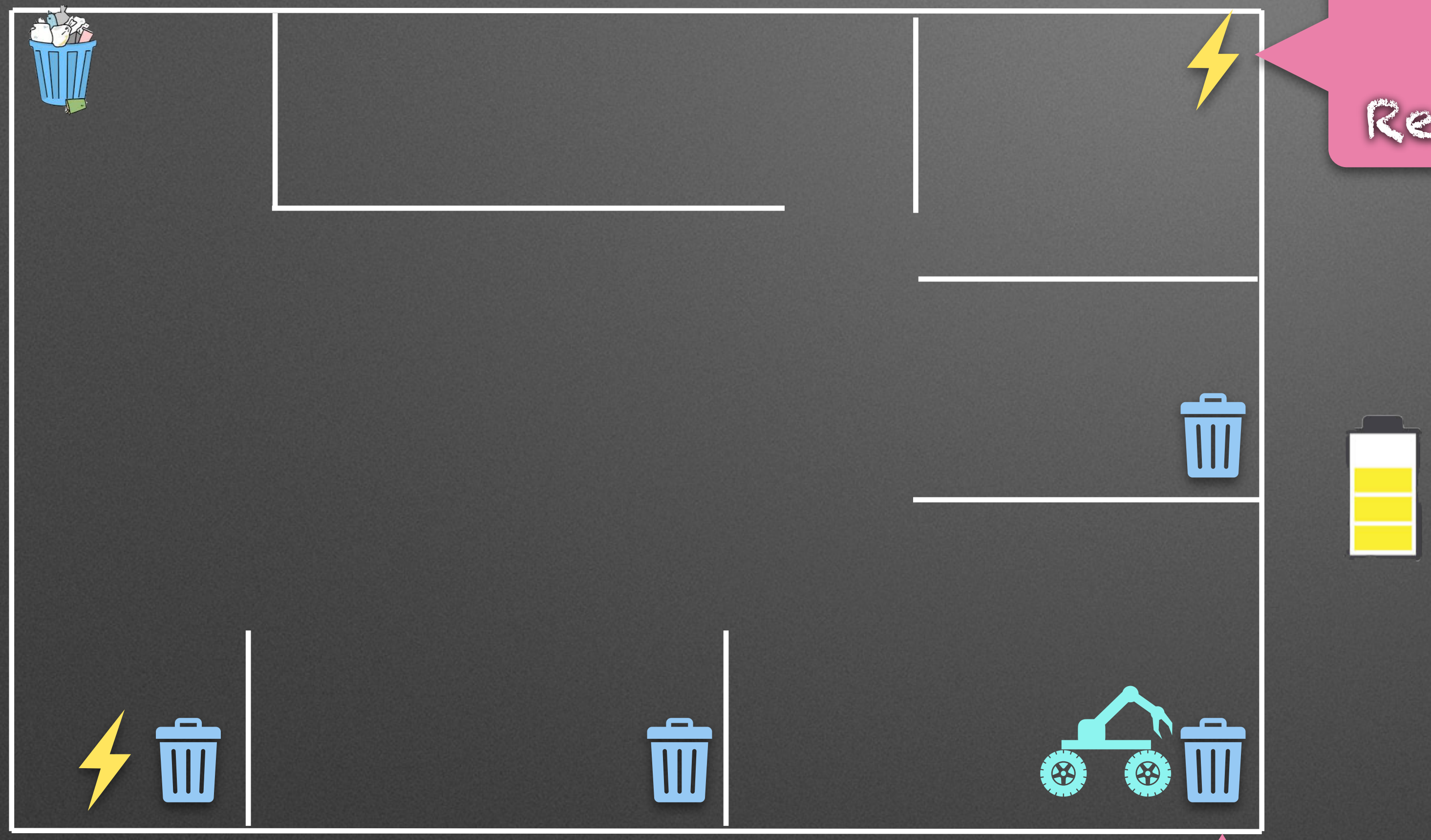
Multi-objective Control Problem



Objective 2 (ψ_2):
Recharge before battery runs out

Objective 1 (ψ_1):
Repeatedly empty all trash cans

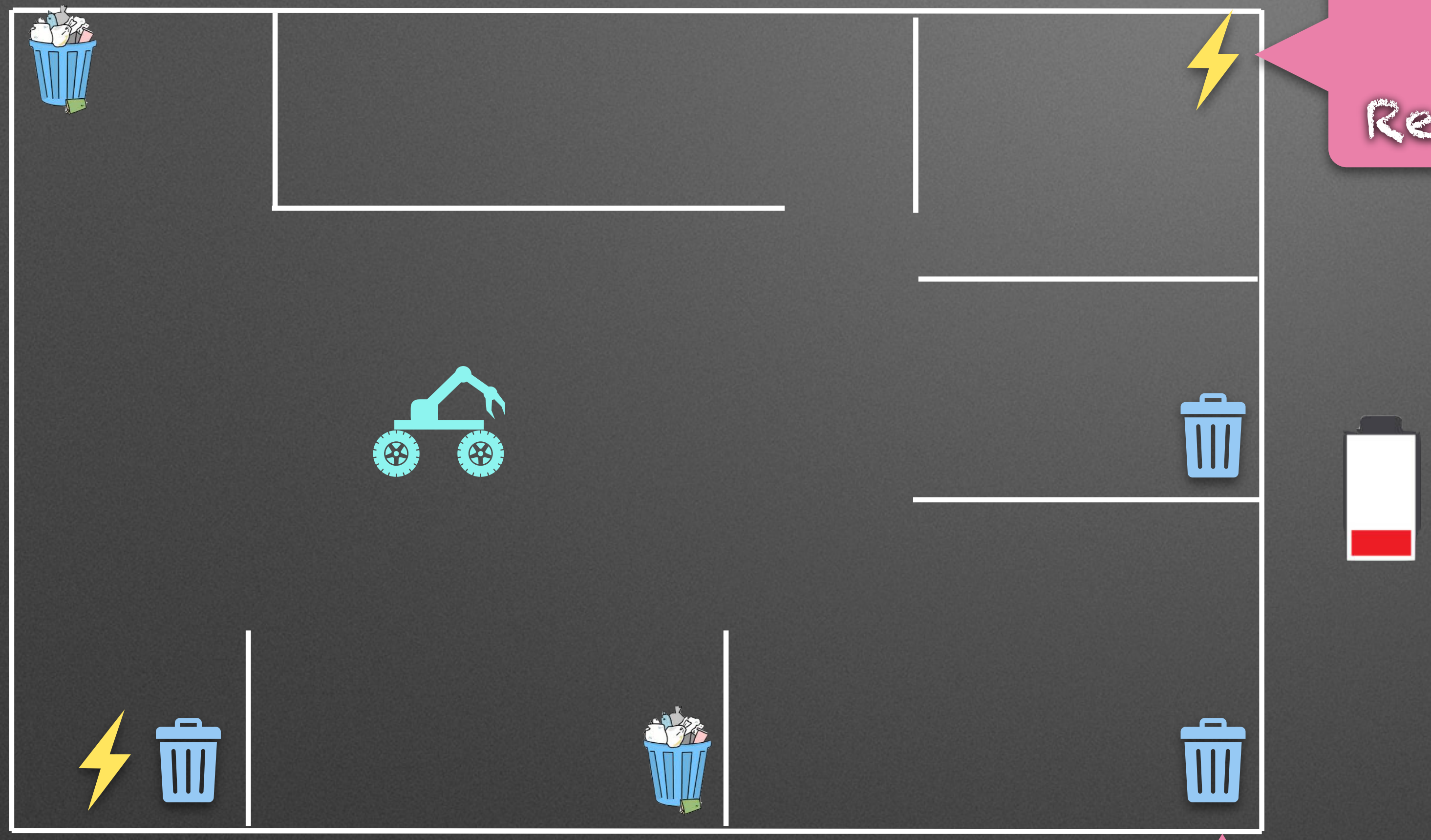
Multi-objective Control Problem



Objective 2 (ψ_2):
Recharge before battery runs out

Objective 1 (ψ_1):
Repeatedly empty all trash cans

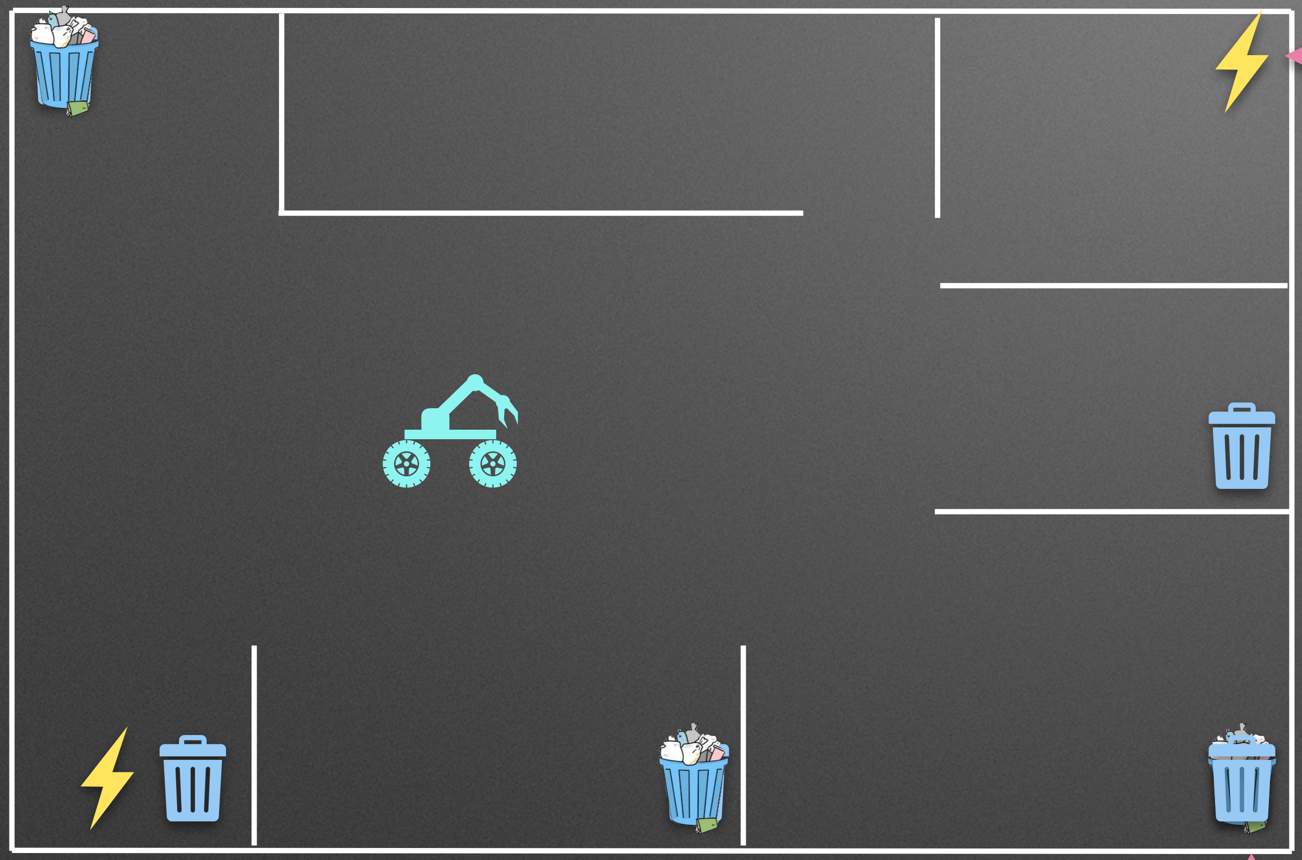
Multi-objective Control Problem

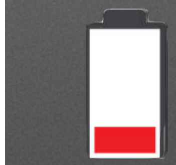


Objective 2 (ψ_2):
Recharge before battery runs out

Objective 1 (ψ_1):
Repeatedly empty all trash cans

Multi-objective Control Problem

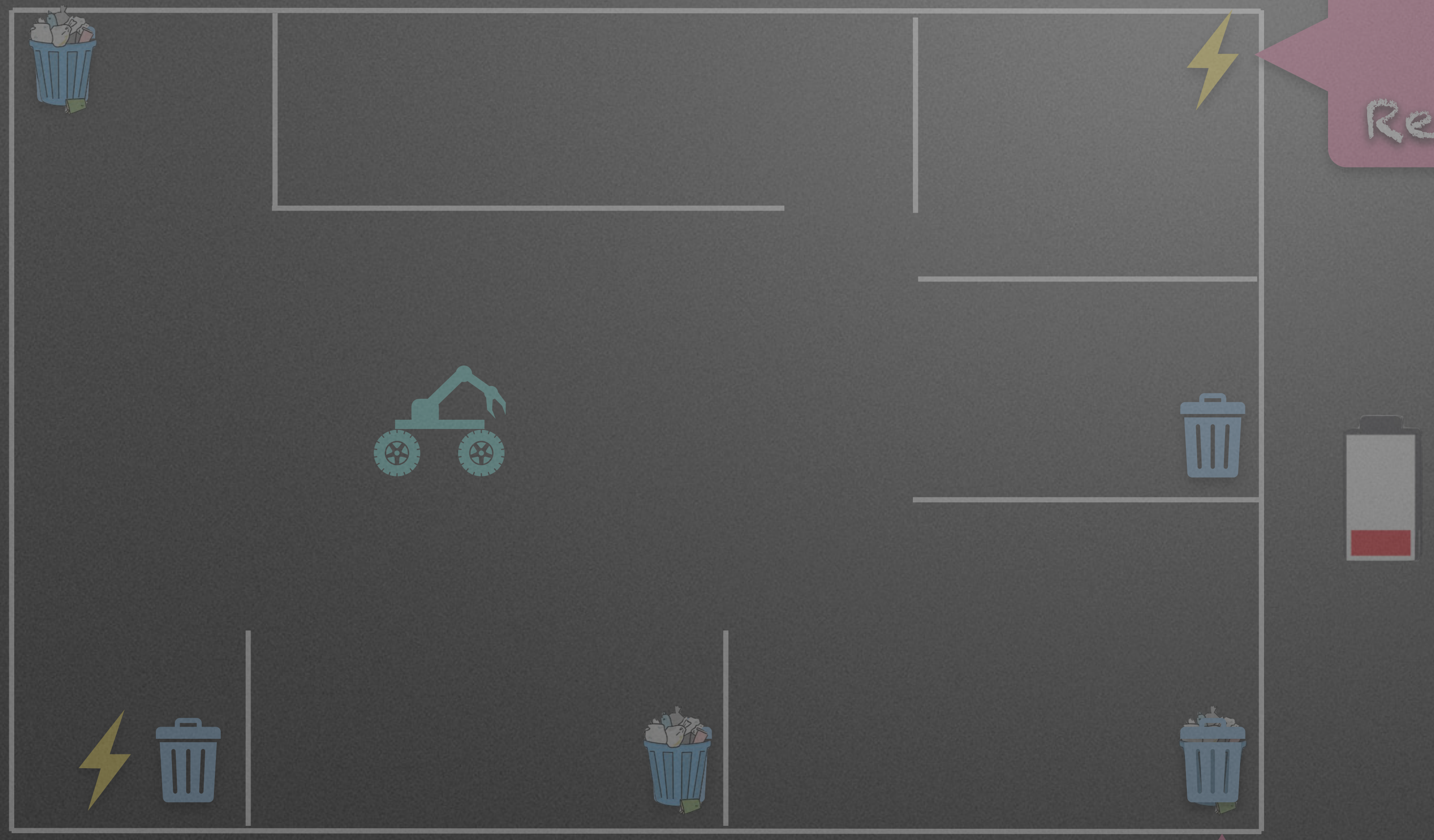




Objective 2 (ψ_2):
Recharge before battery runs out

Objective 1 (ψ_1):
Repeatedly empty all trash cans

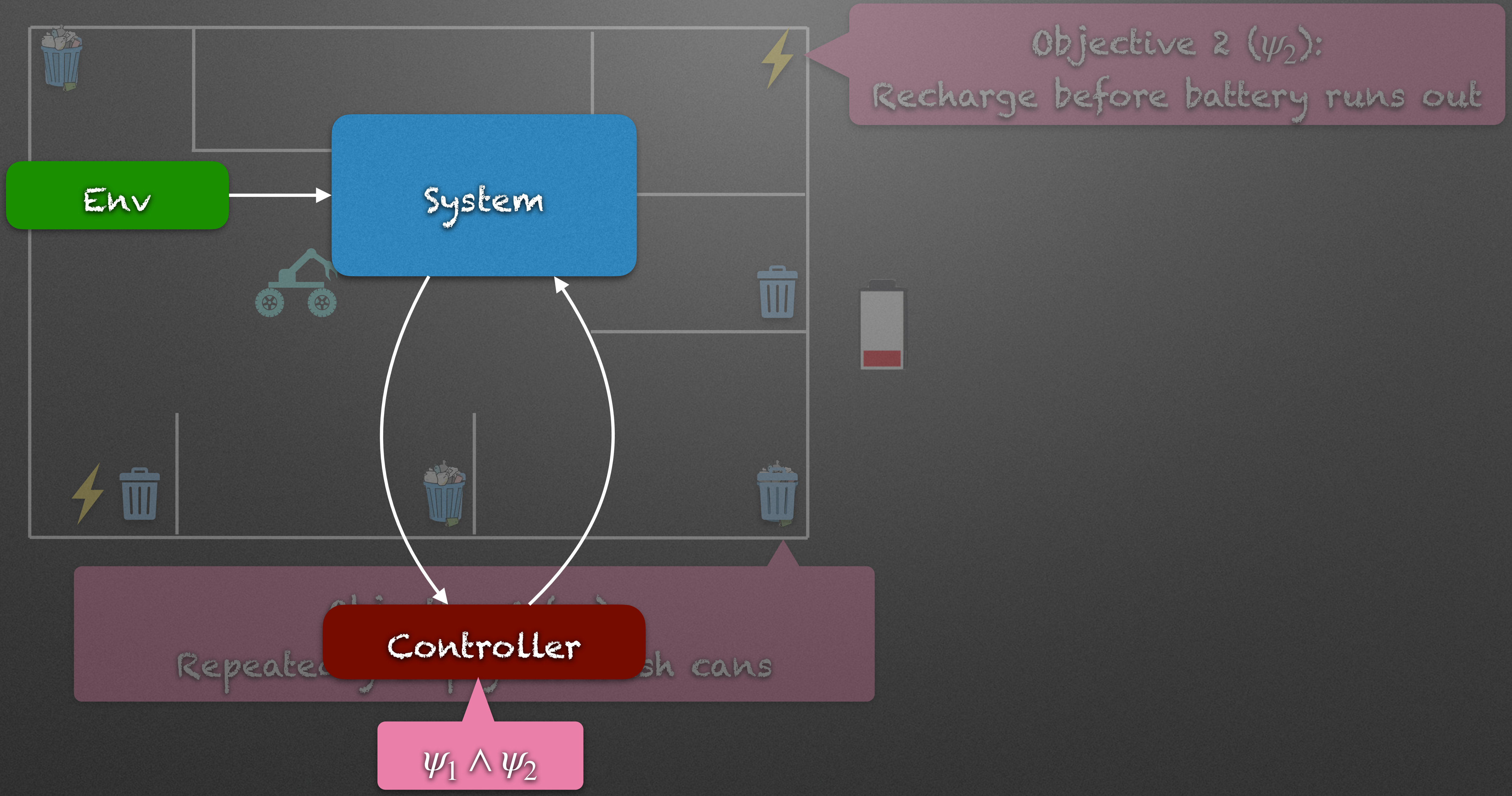
Multi-objective Control Problem



Objective 2 (ψ_2):
Recharge before battery runs out

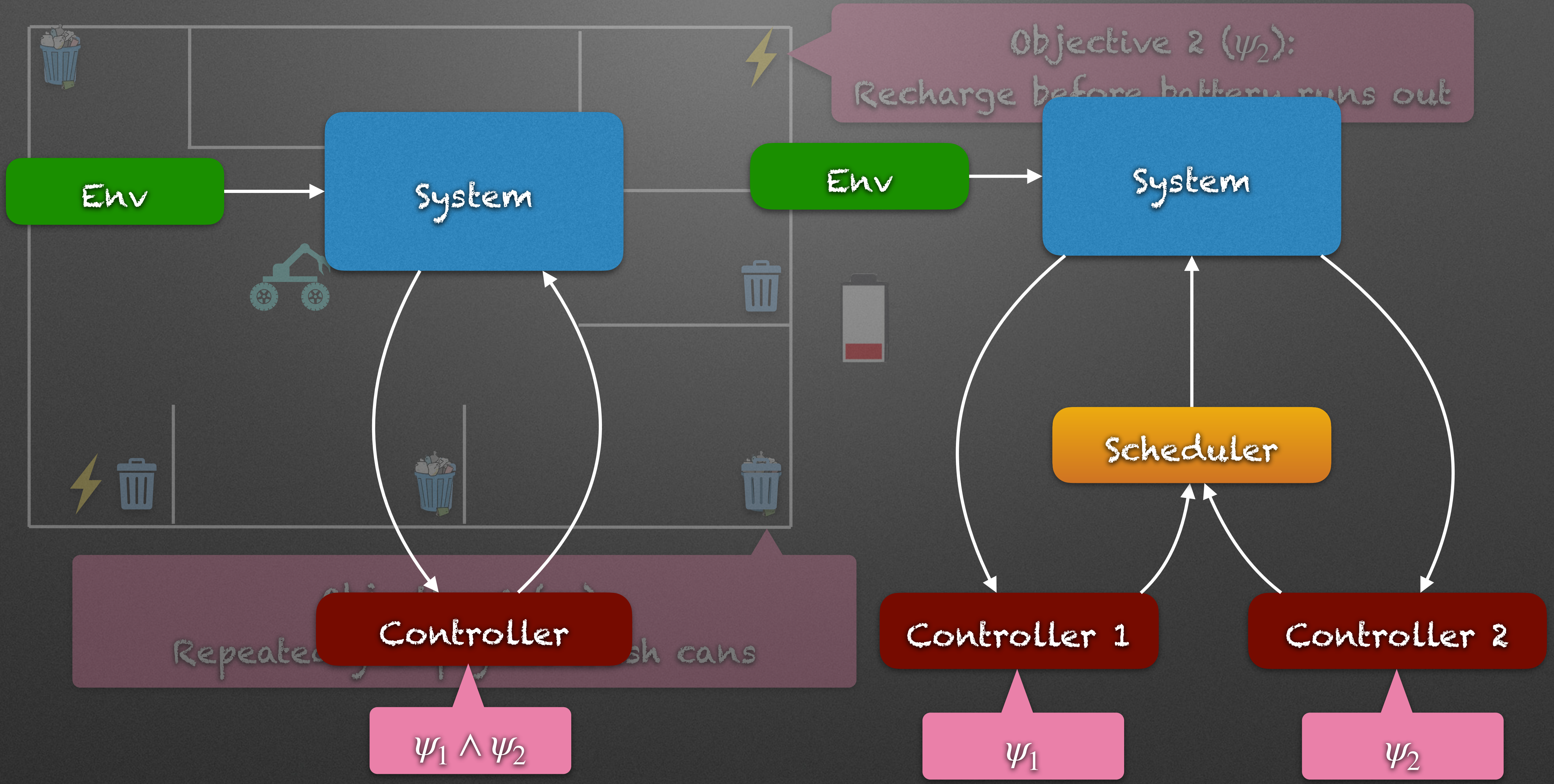
Objective 1 (ψ_1):
Repeatedly empty all trash cans

Multi-objective Control Problem



Centralised Controller Synthesis

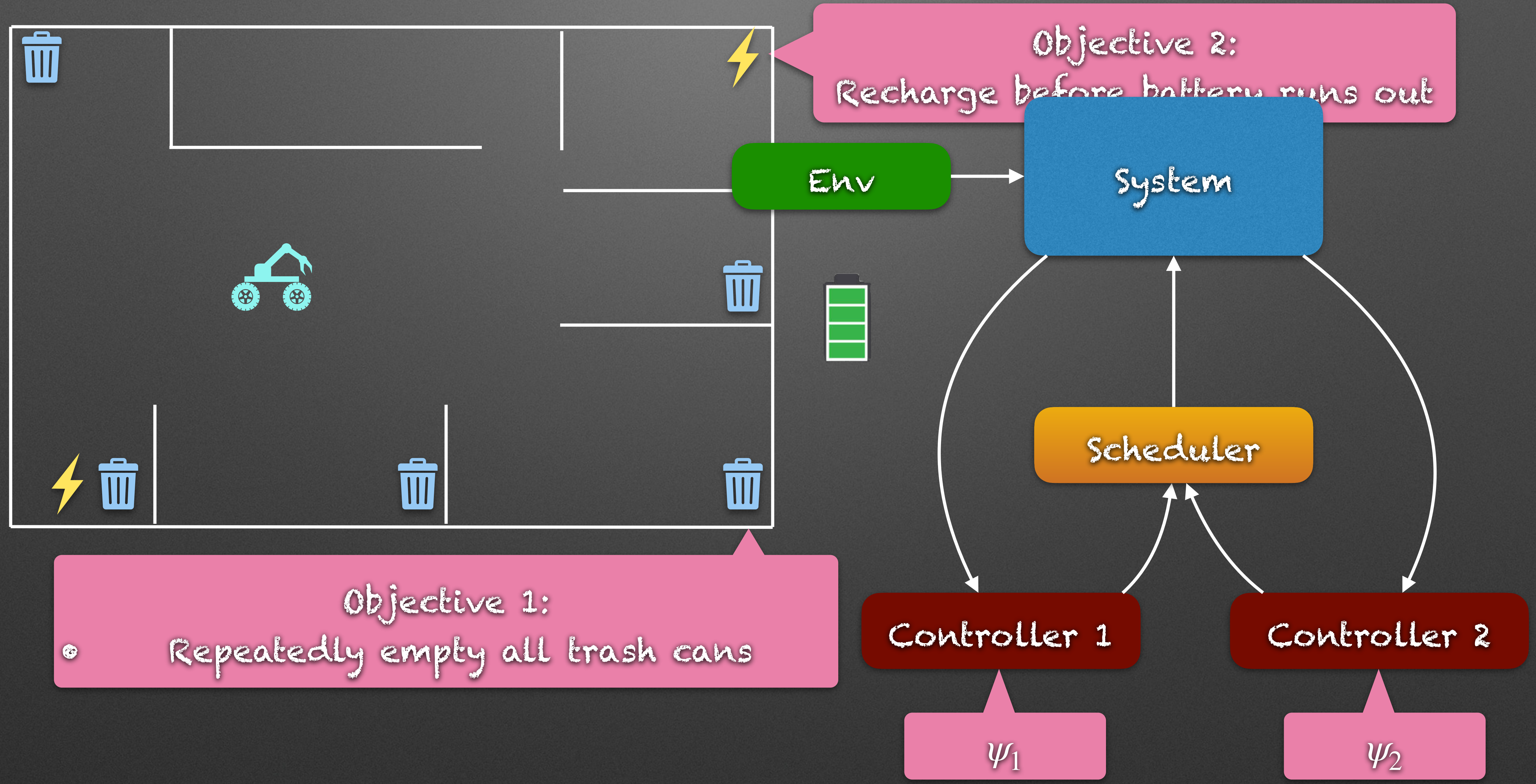
Multi-objective Control Problem



Centralised Controller Synthesis

De-centralised Controller Synthesis

Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Env

System

Scheduler

Controller 1

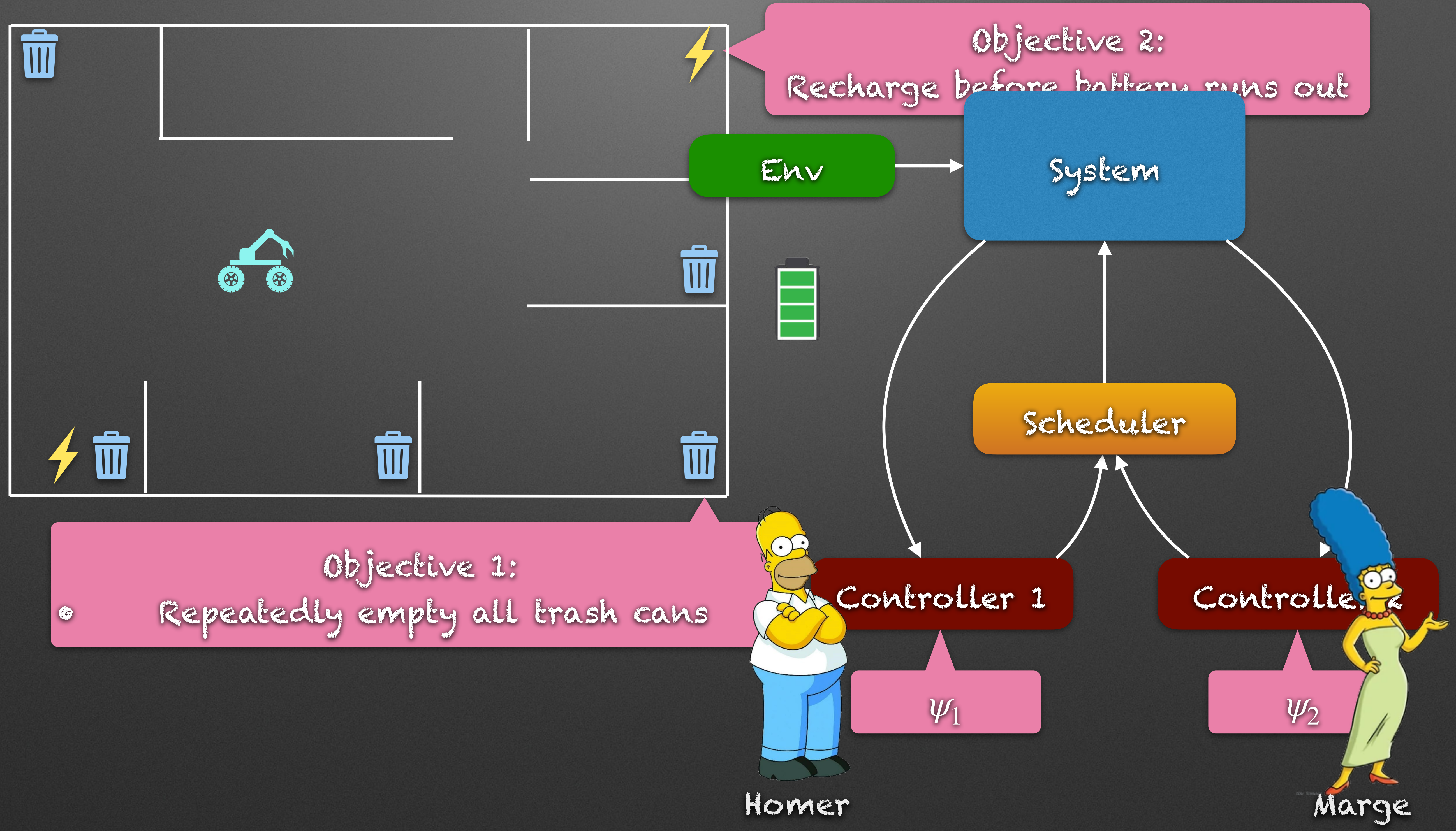
Controller 2

ψ_1

ψ_2

Objective 1:
Repeatedly empty all trash cans

Multi-objective Decentralised Controller Synthesis

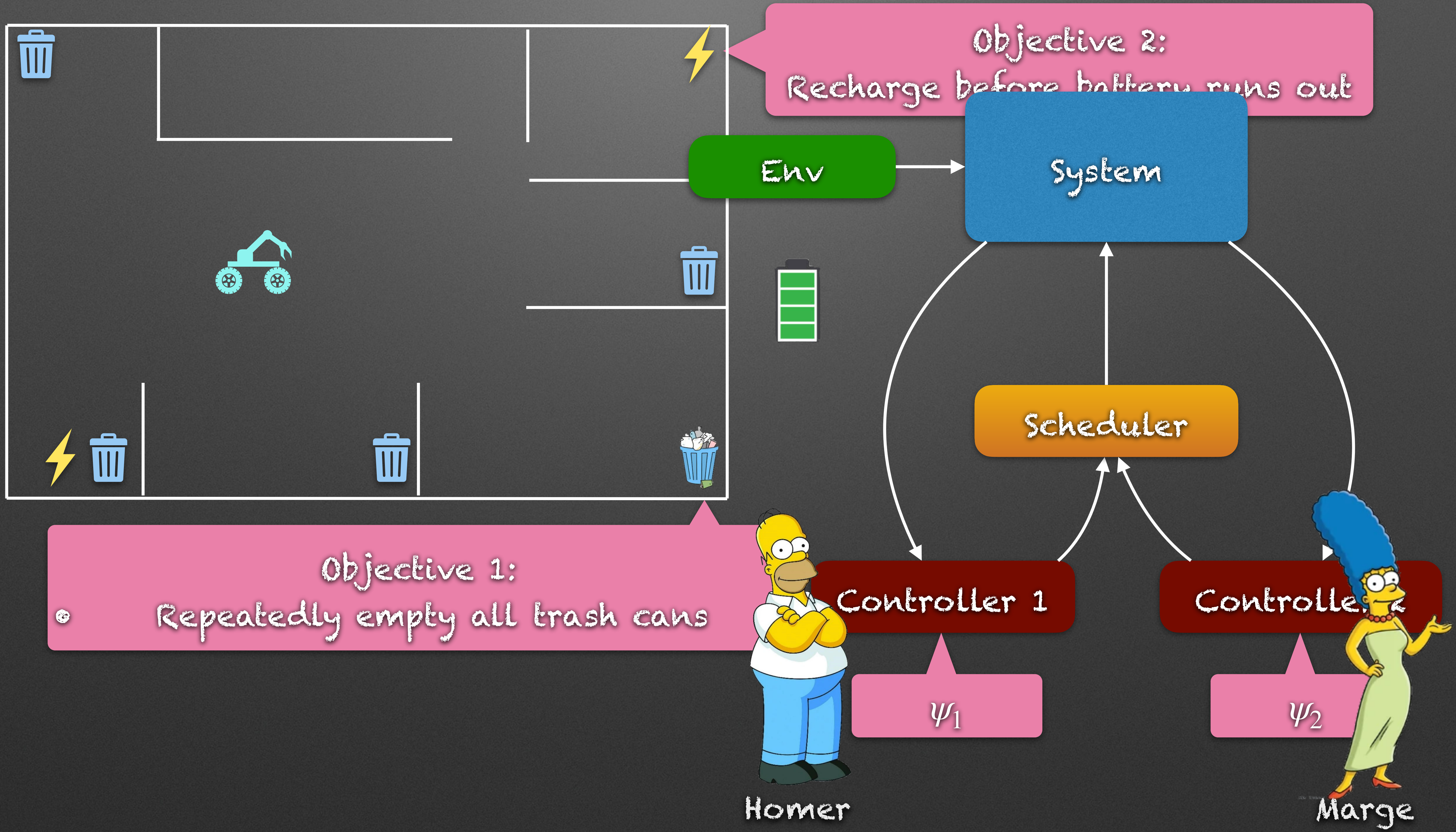


Objective 2:
Recharge before battery runs out

Objective 1:

- Repeatedly empty all trash cans

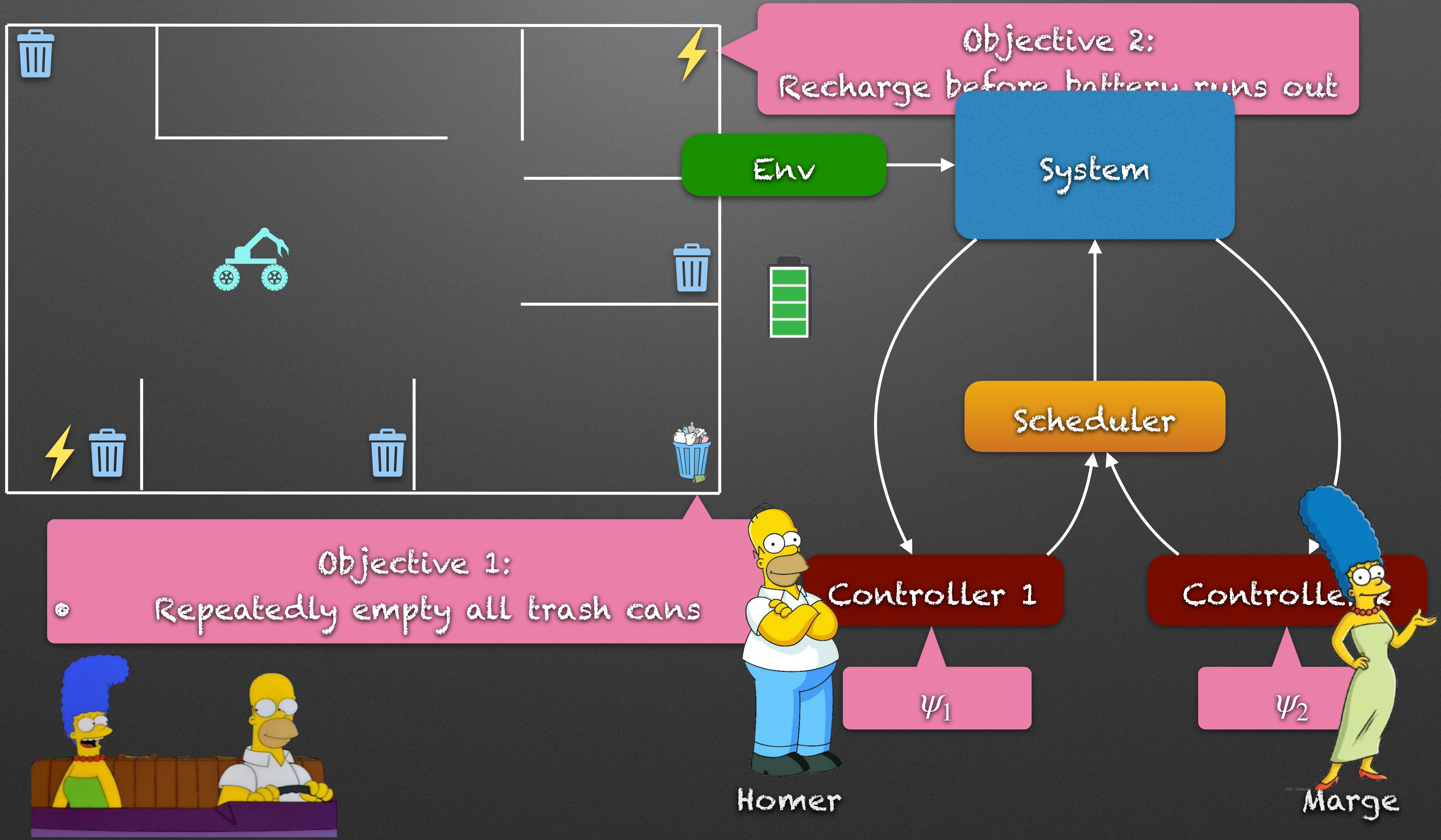
Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Objective 1:
- Repeatedly empty all trash cans

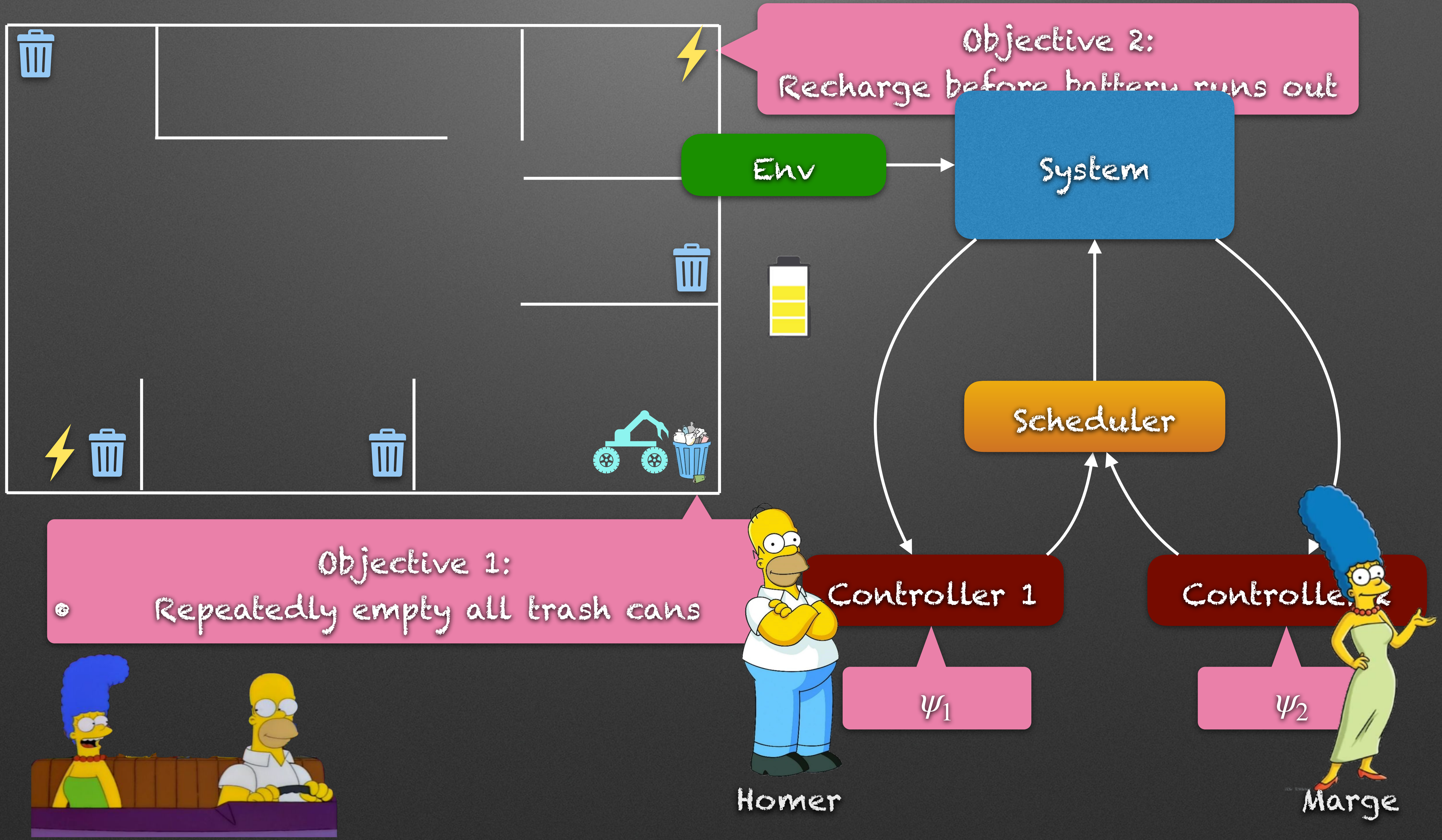
Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Objective 1:
Repeatedly empty all trash cans

Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Env

System

Scheduler

Controller 1

Controller 2

ψ_1

ψ_2

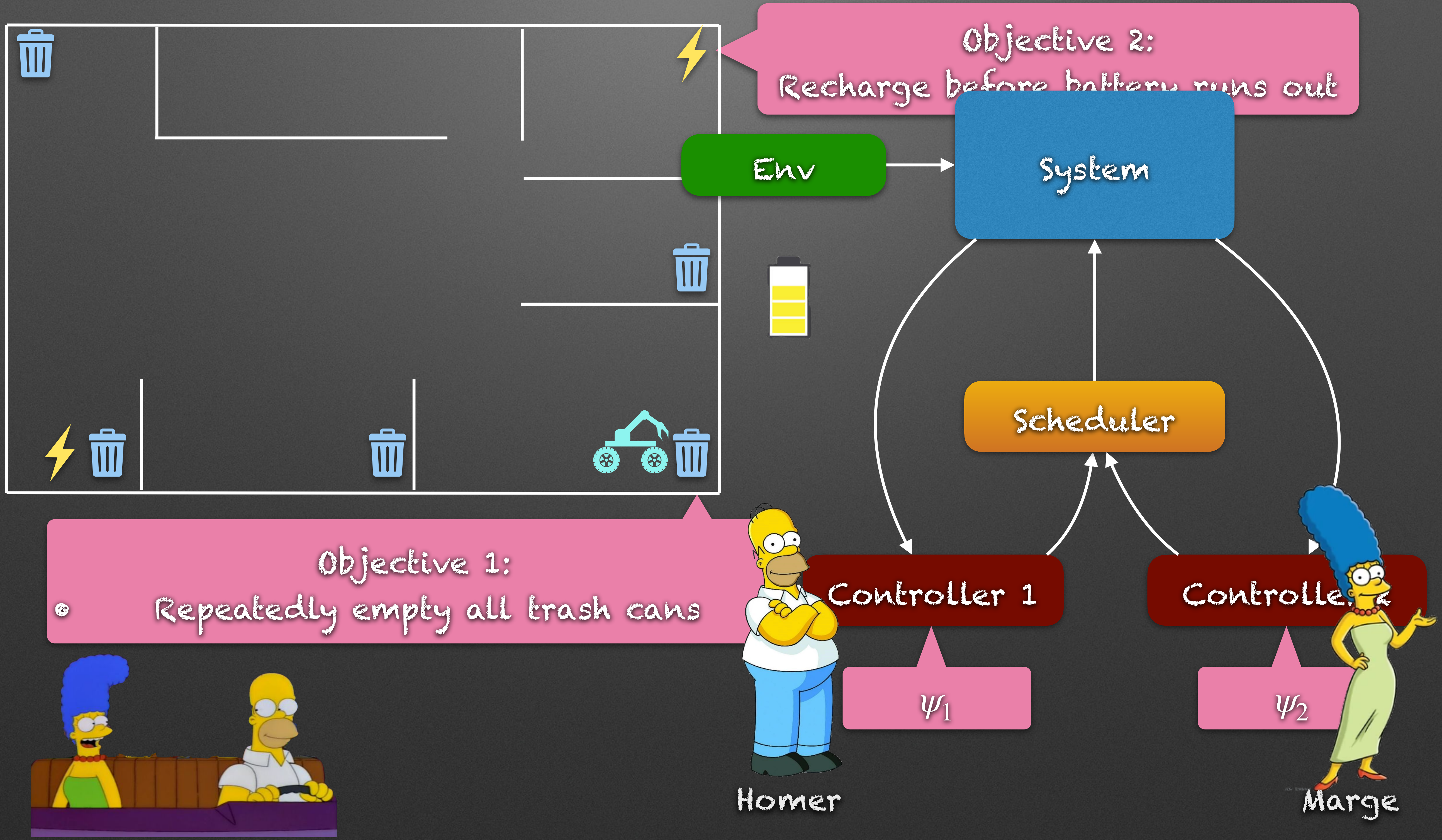
Objective 1:

- Repeatedly empty all trash cans

Homer

Marge

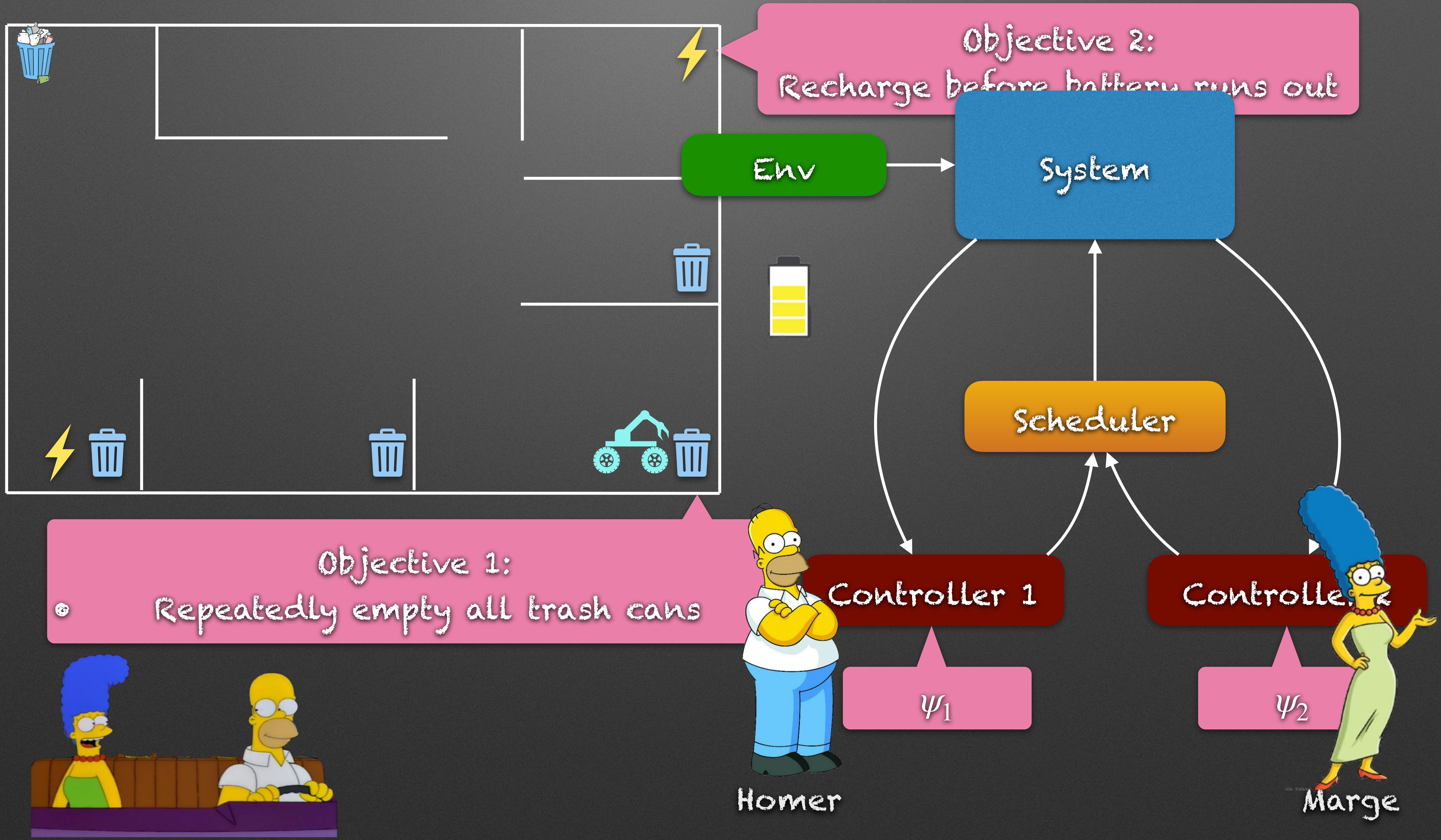
Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Objective 1:
Repeatedly empty all trash cans

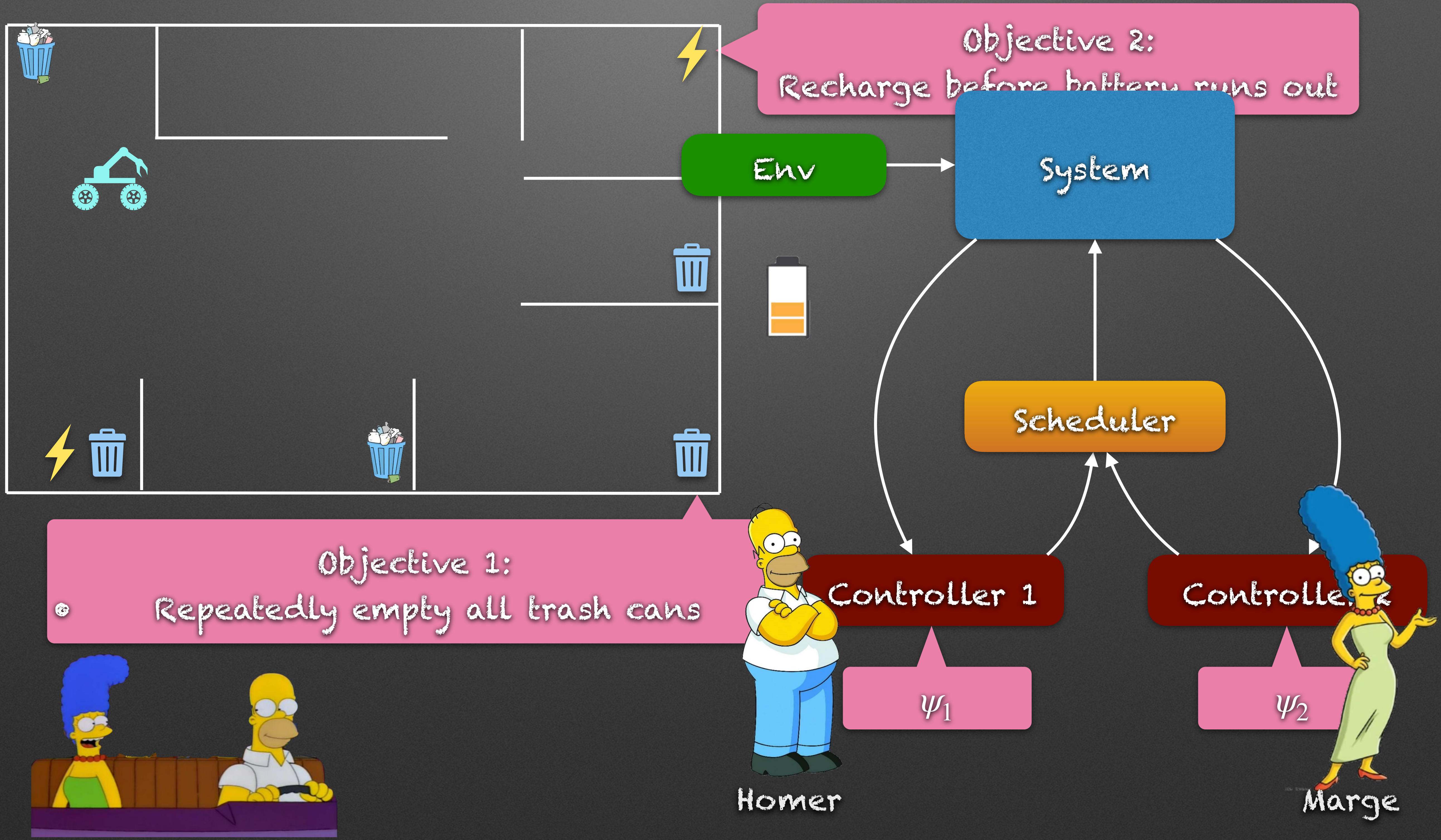
Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Objective 1:
- Repeatedly empty all trash cans

Multi-objective Decentralised Controller Synthesis

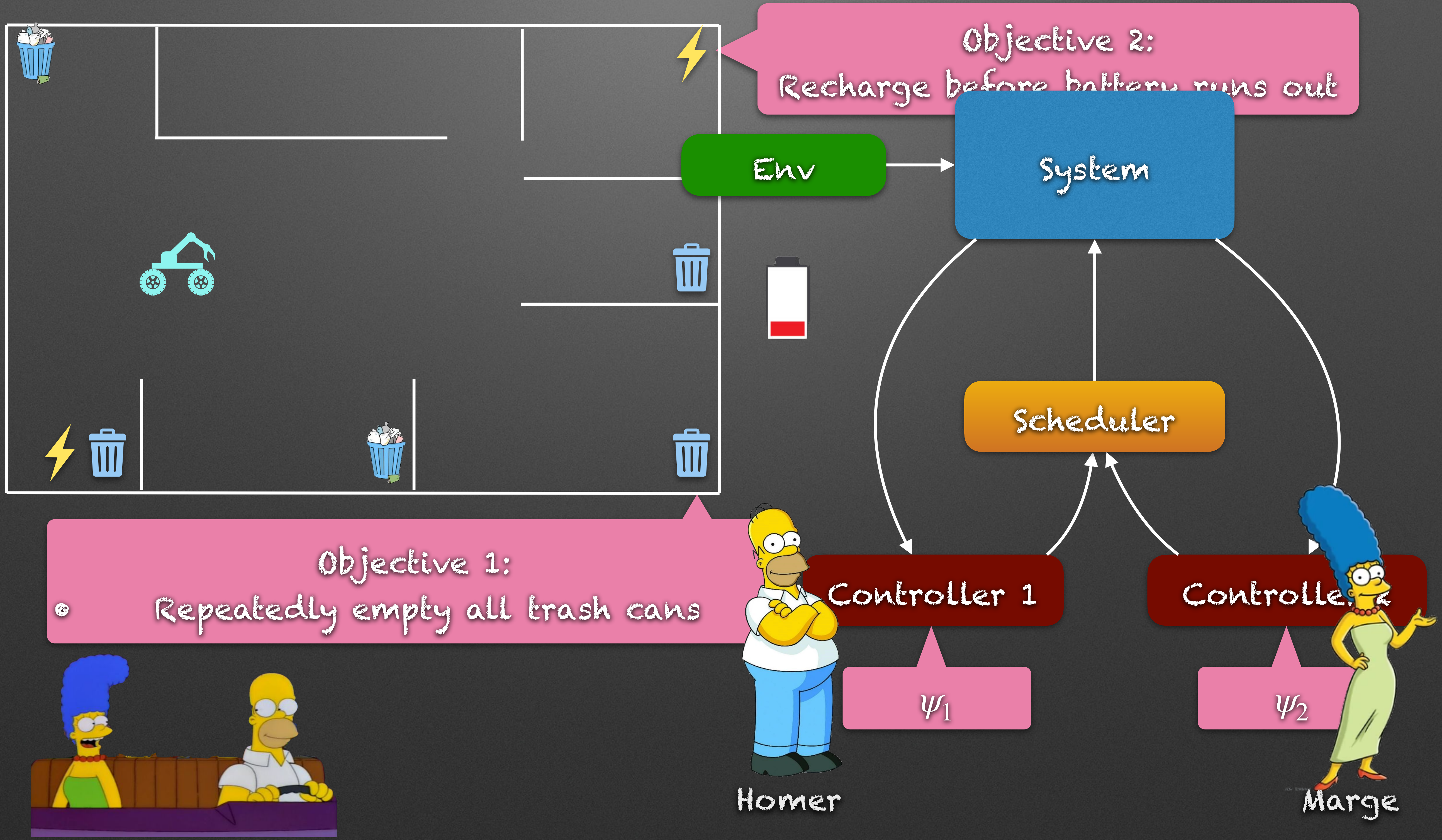


Objective 2:
Recharge before battery runs out

Objective 1:

- Repeatedly empty all trash cans

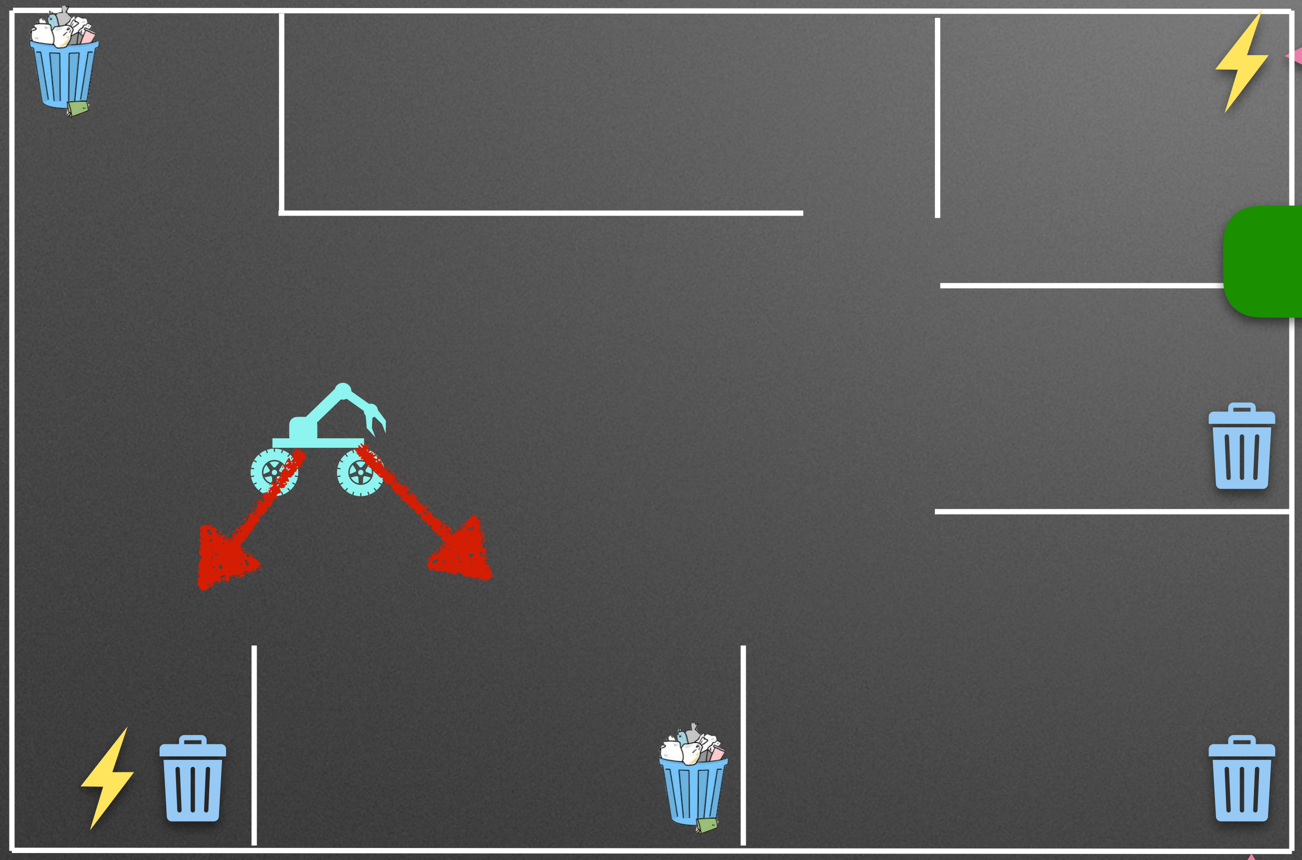
Multi-objective Decentralised Controller Synthesis



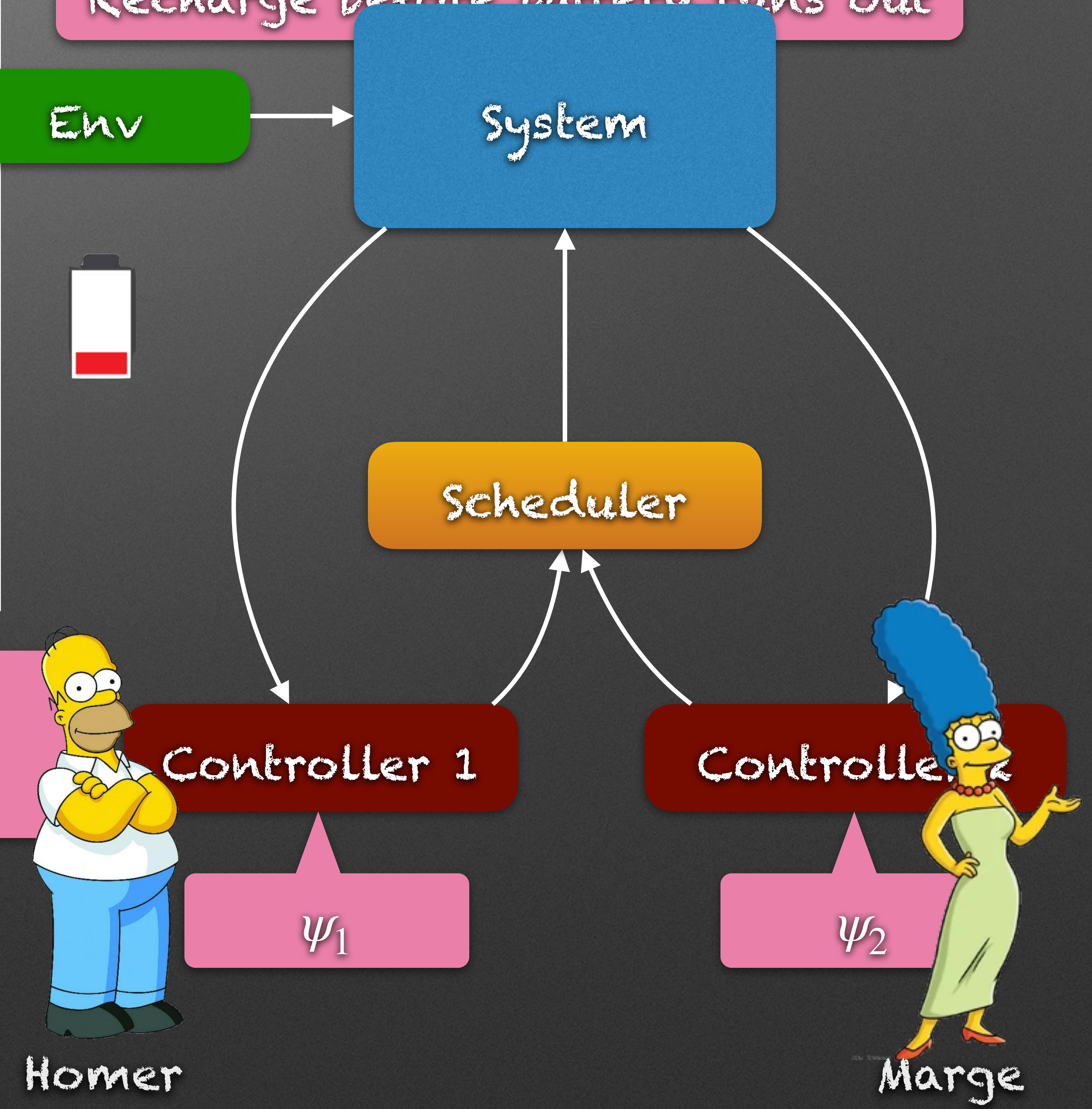
Objective 2:
Recharge before battery runs out

Objective 1:
- Repeatedly empty all trash cans

Multi-objective Decentralised Controller Synthesis



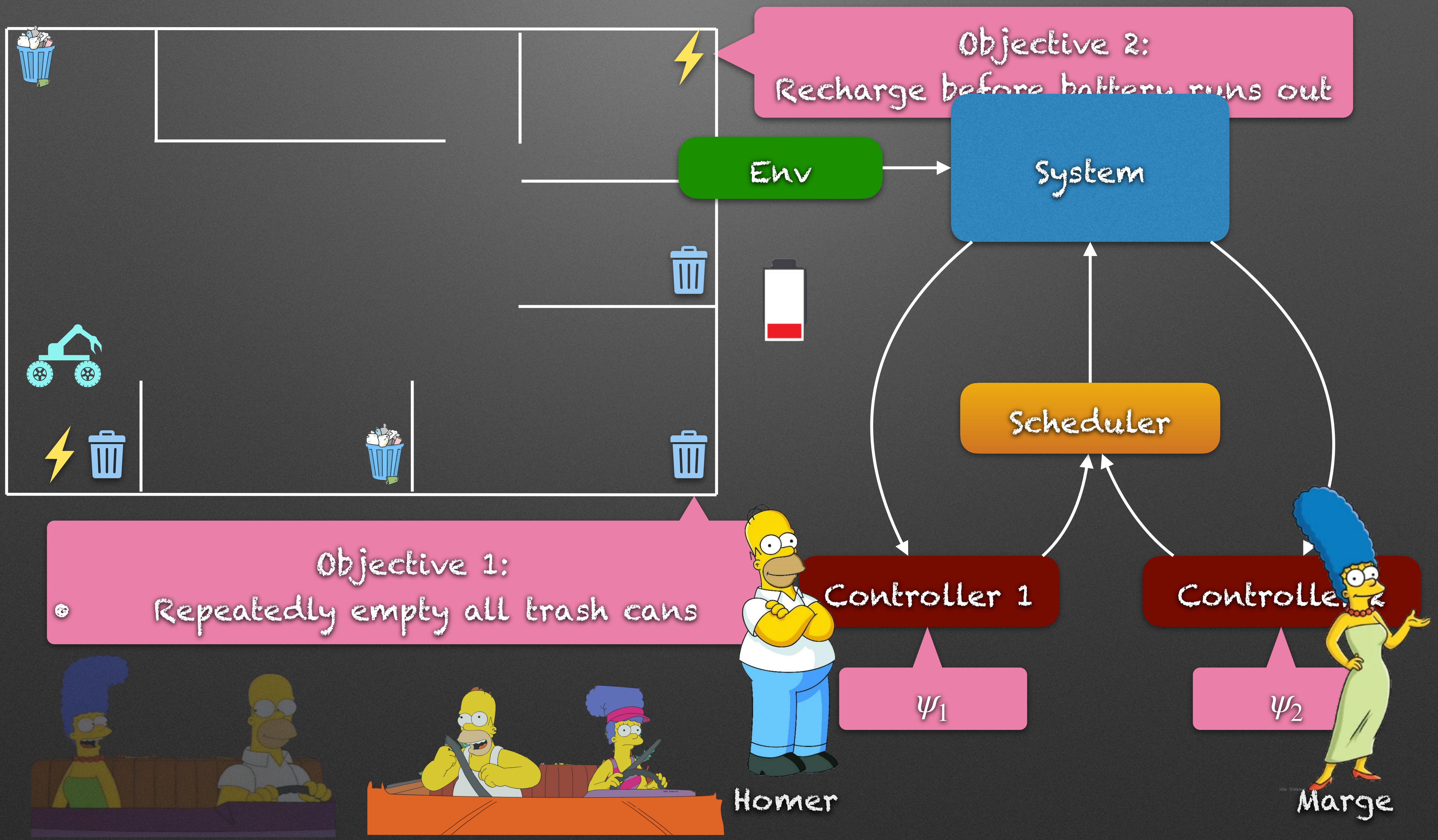
Objective 2:
Recharge before battery runs out



Objective 1:

- Repeatedly empty all trash cans

Multi-objective Decentralised Controller Synthesis

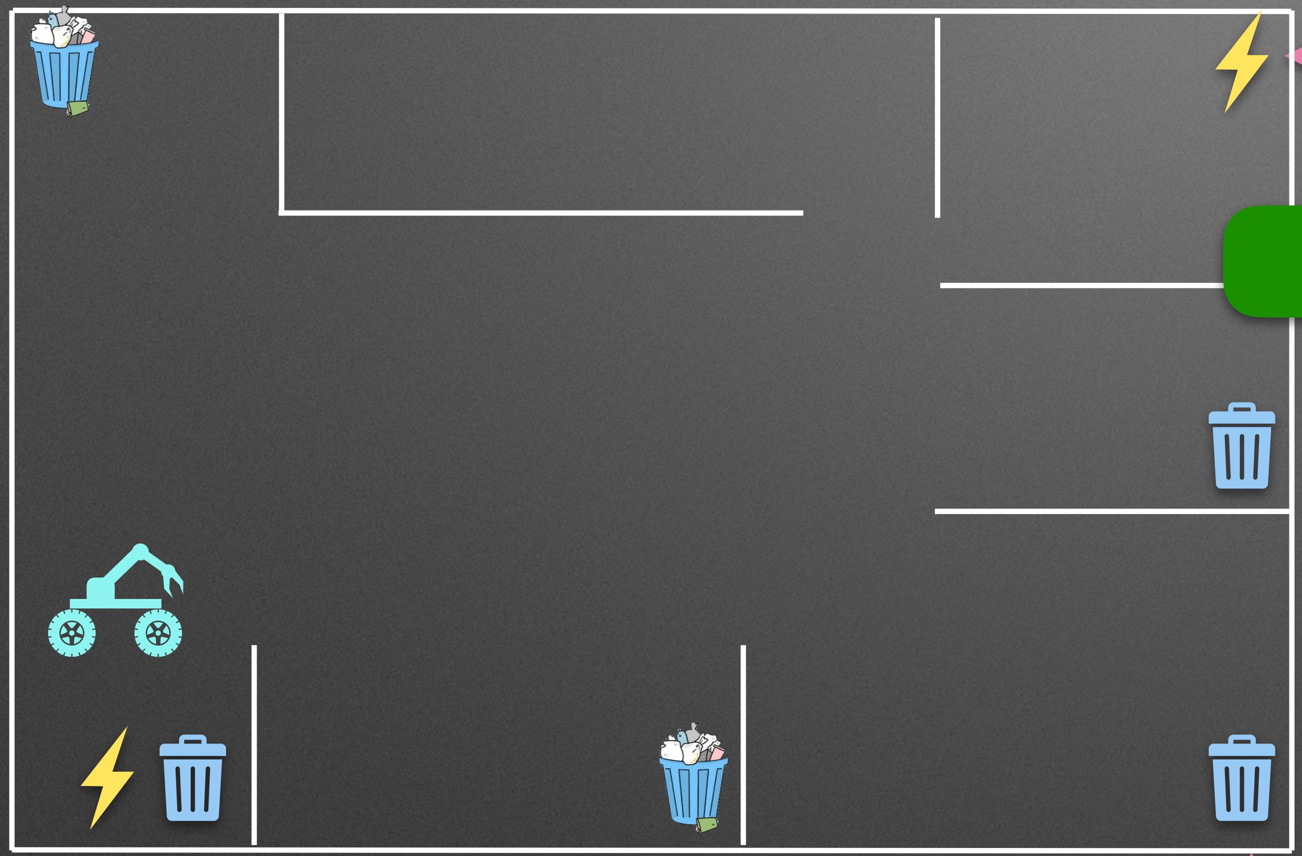


Objective 2:
Recharge before battery runs out

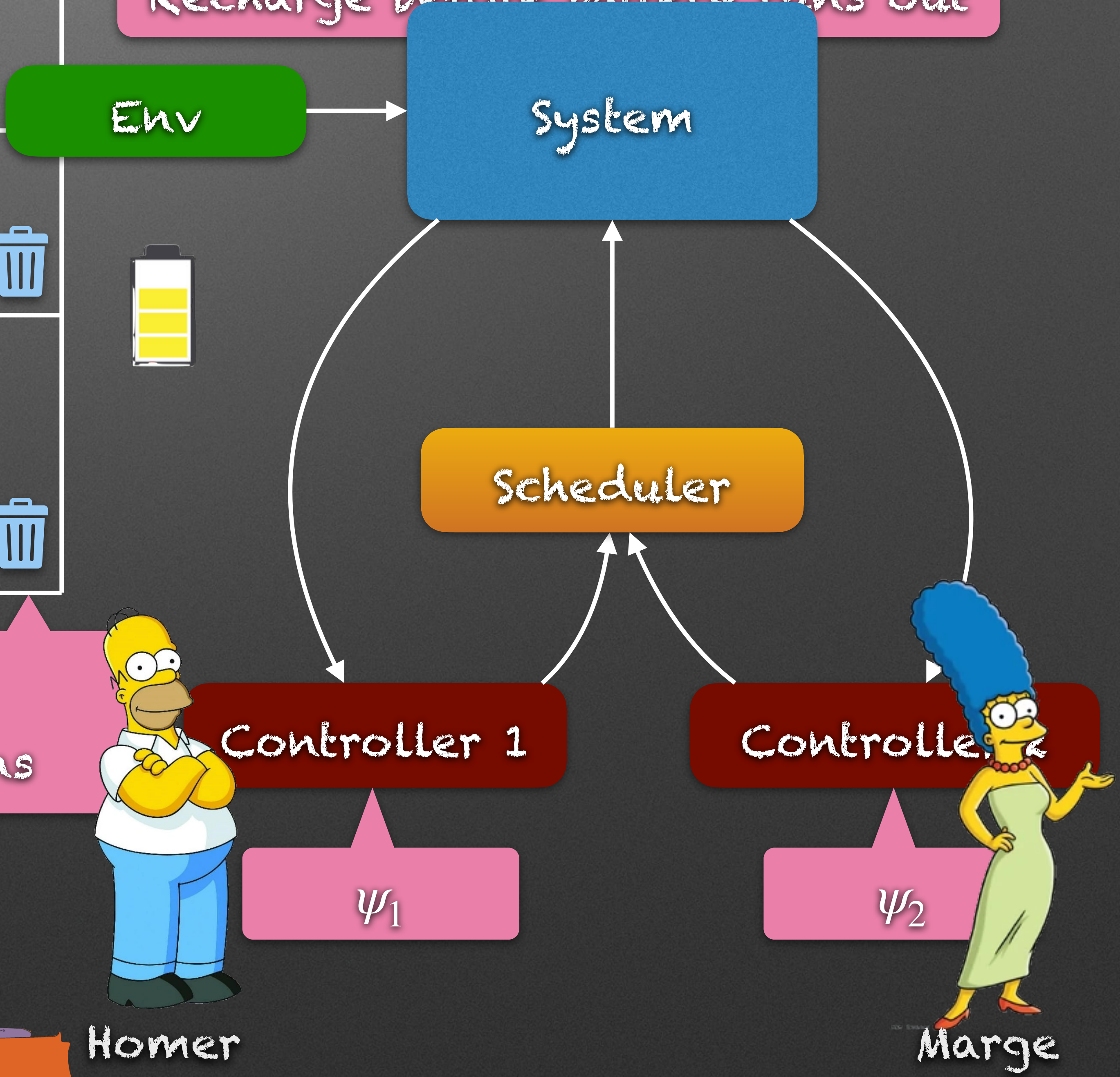
Objective 1:

- Repeatedly empty all trash cans

Multi-objective Decentralised Controller Synthesis

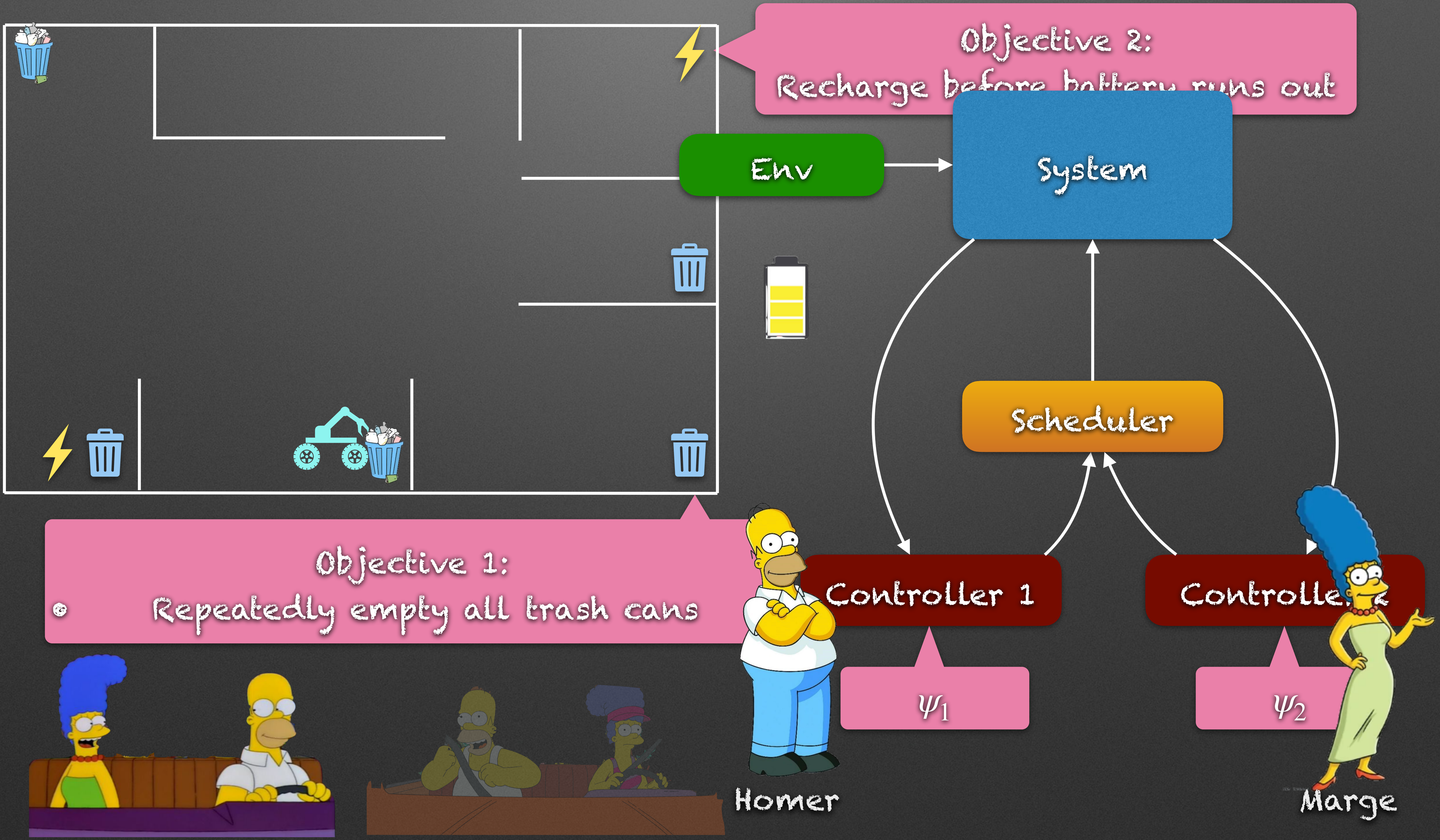


Objective 2:
Recharge before battery runs out



Objective 1:
- Repeatedly empty all trash cans

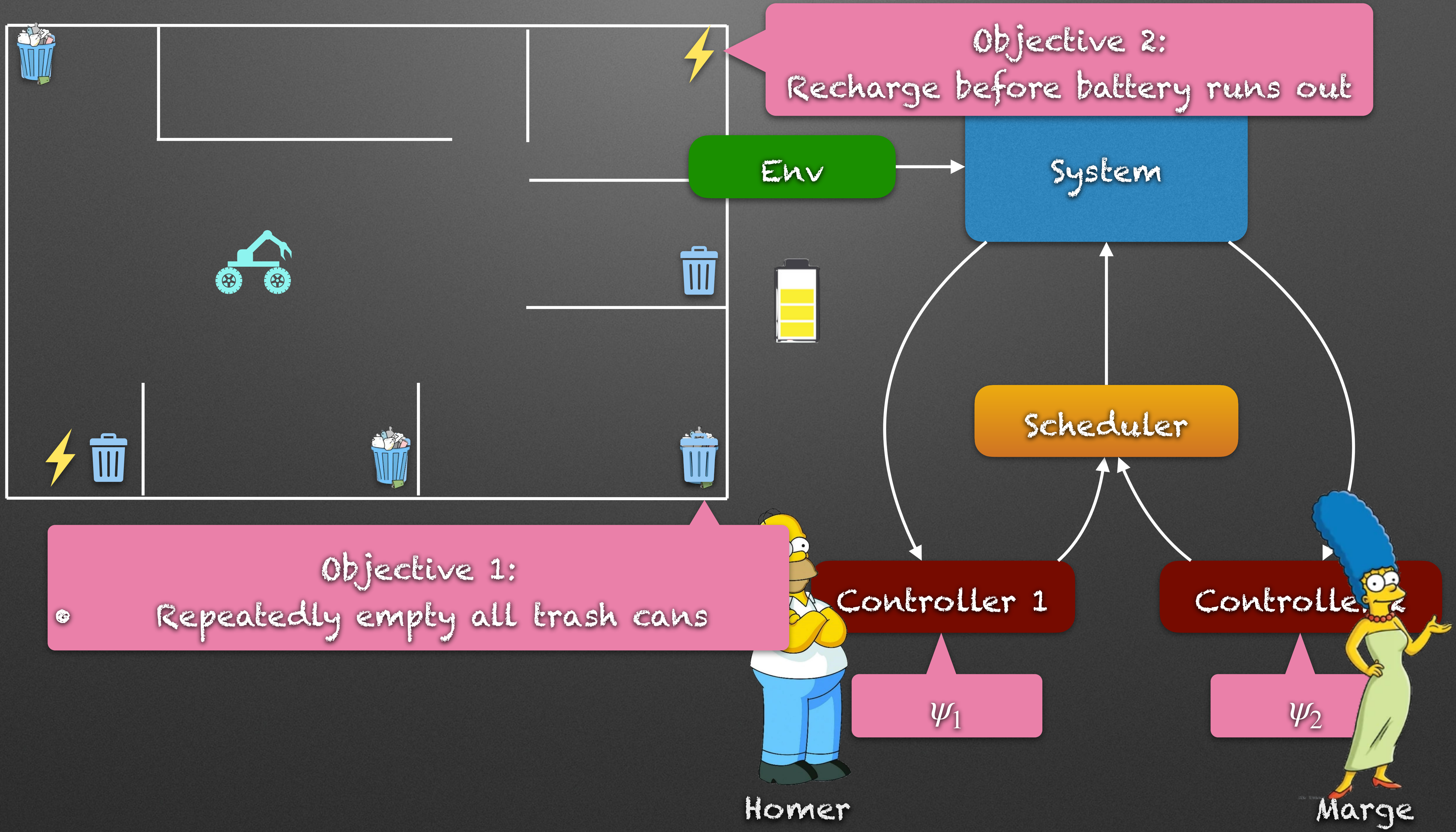
Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Objective 1:
- Repeatedly empty all trash cans

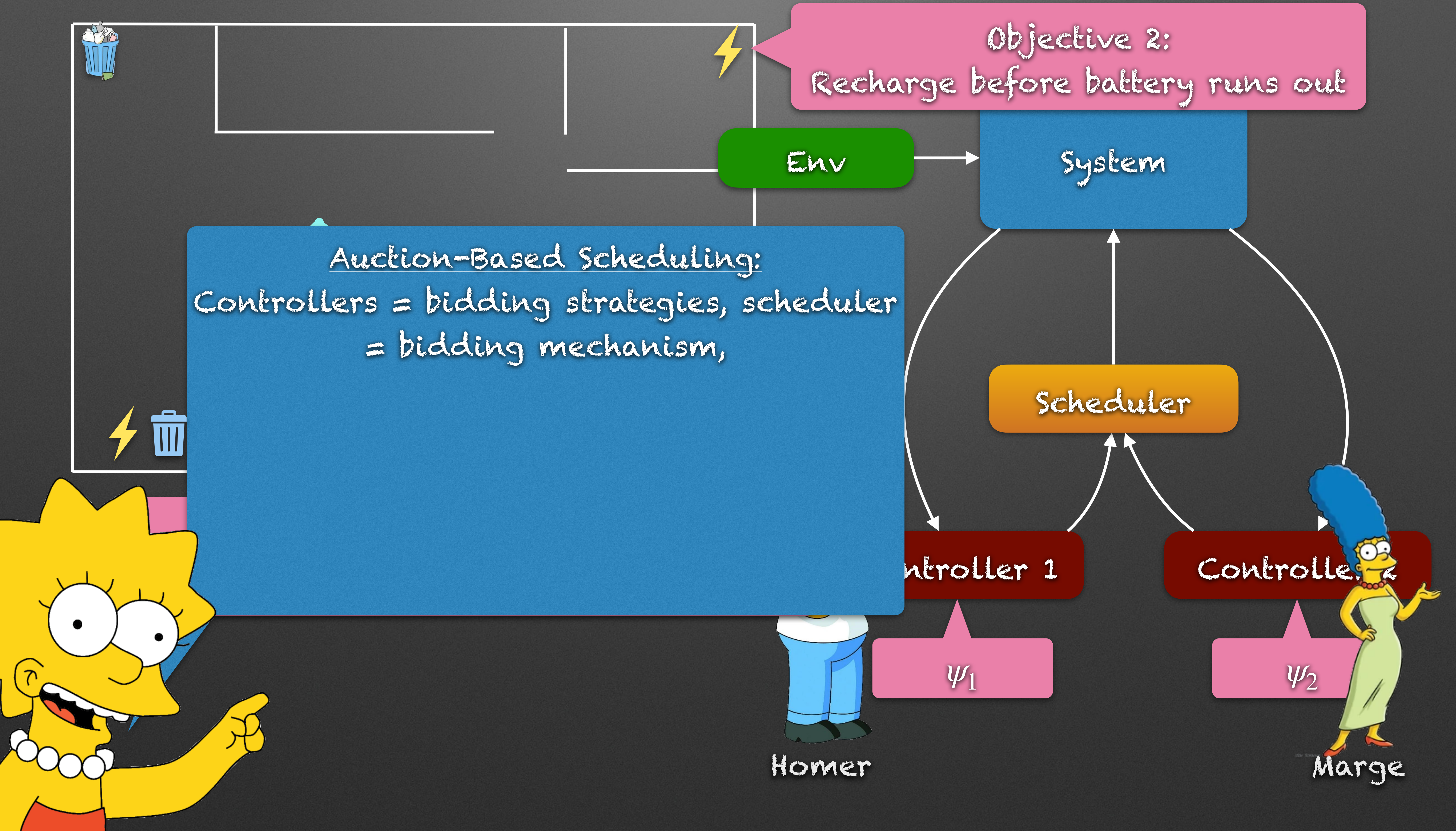
Multi-objective Decentralised Controller Synthesis



Objective 2:
Recharge before battery runs out

Objective 1:
- Repeatedly empty all trash cans

Multi-objective Decentralised Controller Synthesis

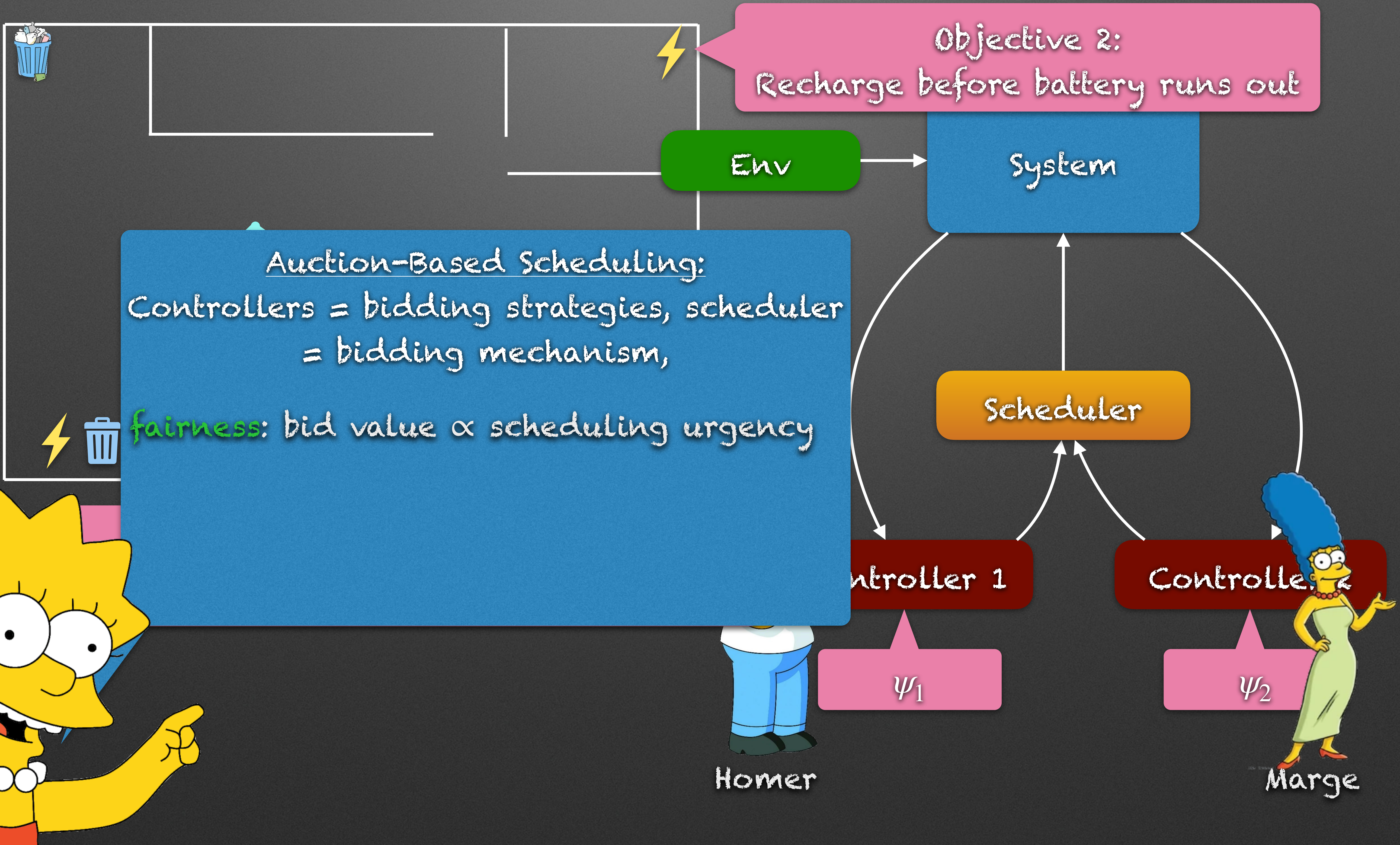


Objective 2:
Recharge before battery runs out

Auction-Based Scheduling:

Controllers = bidding strategies, scheduler = bidding mechanism,

Multi-objective Decentralised Controller Synthesis

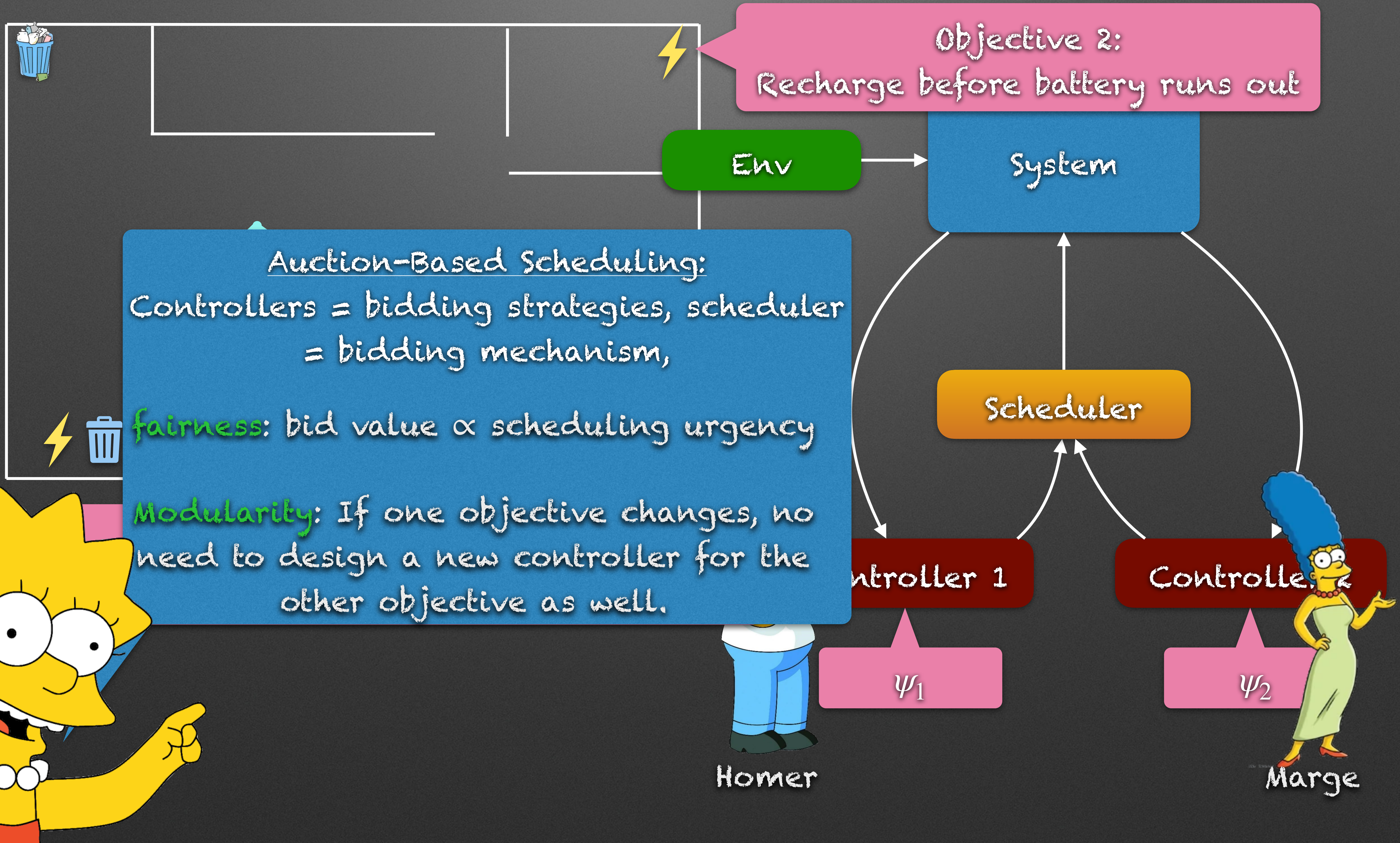


Auction-Based Scheduling:

Controllers = bidding strategies, scheduler = bidding mechanism,

fairness: bid value ∝ scheduling urgency

Multi-objective Decentralised Controller Synthesis



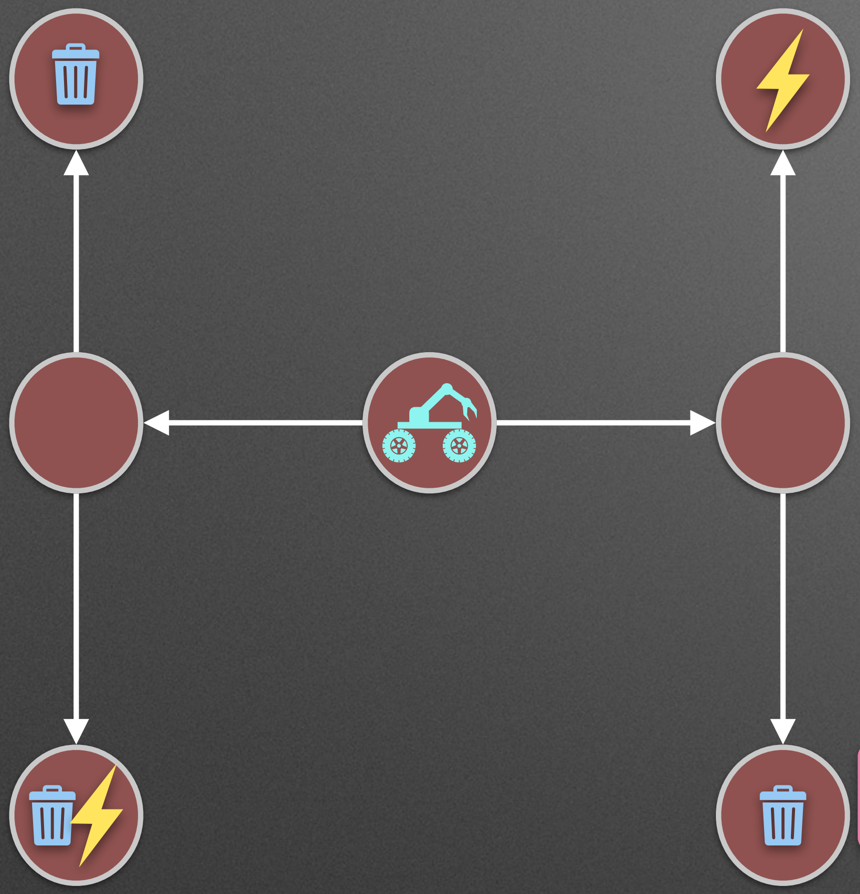
<u>Auction-Based Scheduling:</u>

Controllers = bidding strategies, scheduler = bidding mechanism,

fairness: bid value ∝ scheduling urgency

Modularity: If one objective changes, no need to design a new controller for the other objective as well.

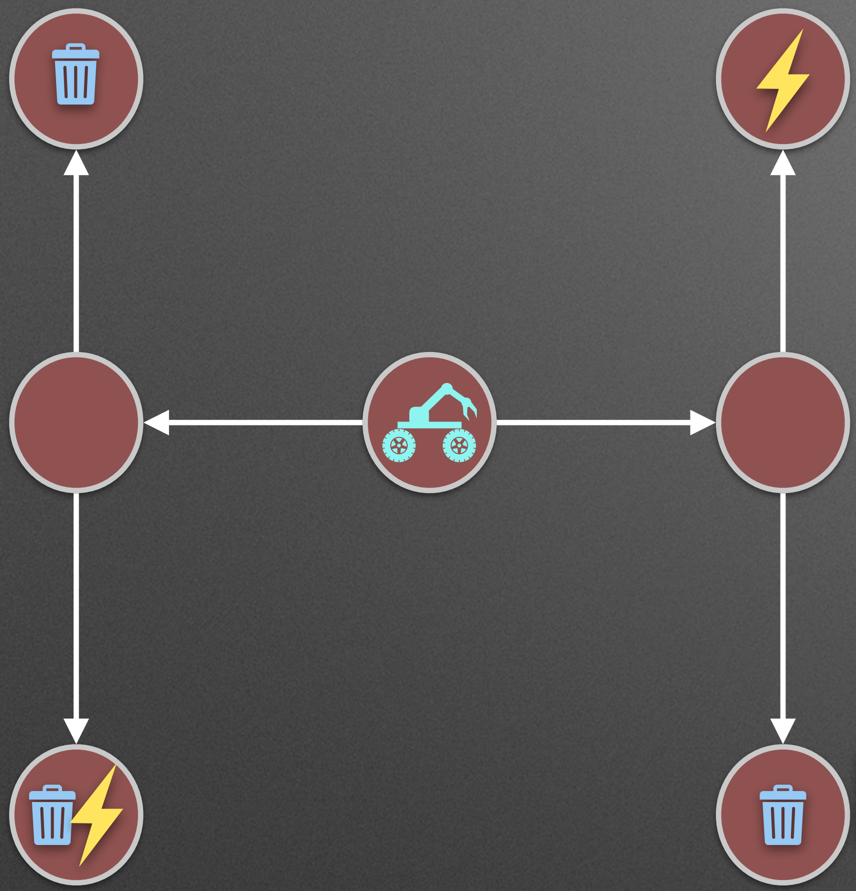
Auction-Based Scheduling



Wants to clear trash-cans

Wants to recharge the battery

Auction-Based Scheduling



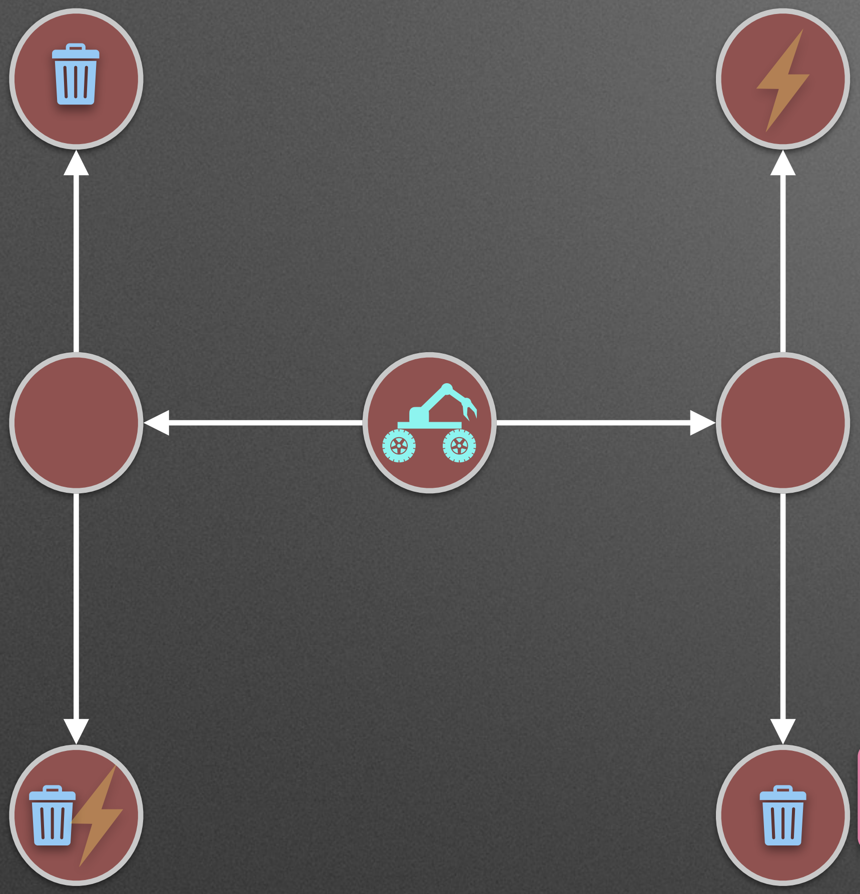
$\frac{1}{4} + \epsilon_1$

Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Auction-Based Scheduling



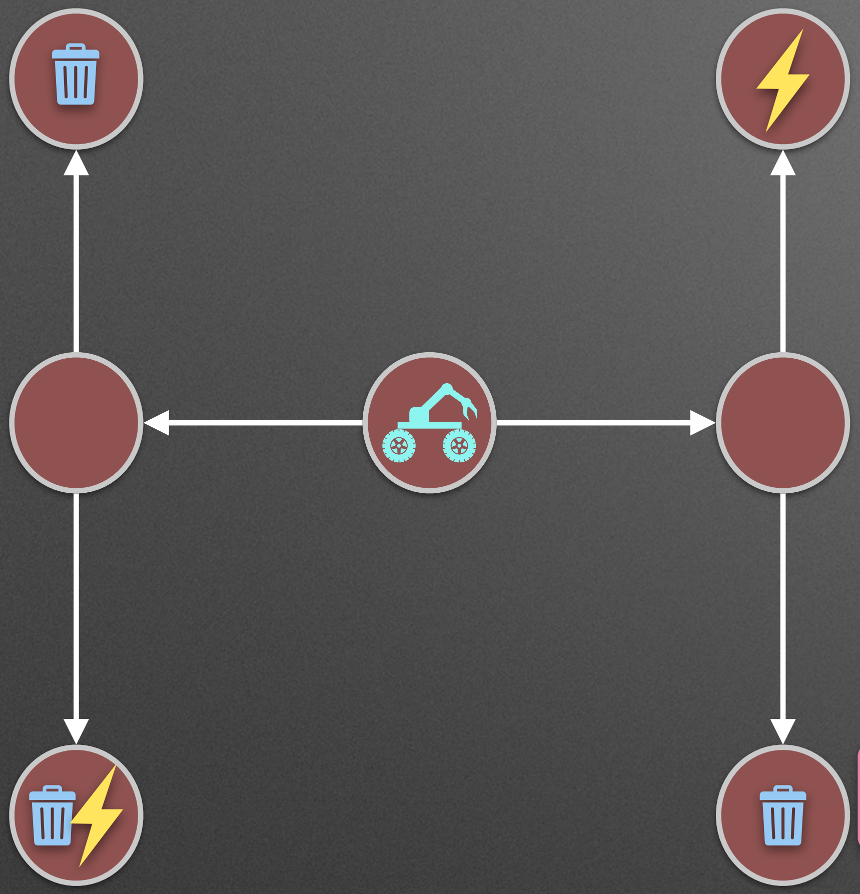
$\frac{1}{4} + \epsilon_1$

Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon_1$

$\frac{1}{2} + \epsilon_2$

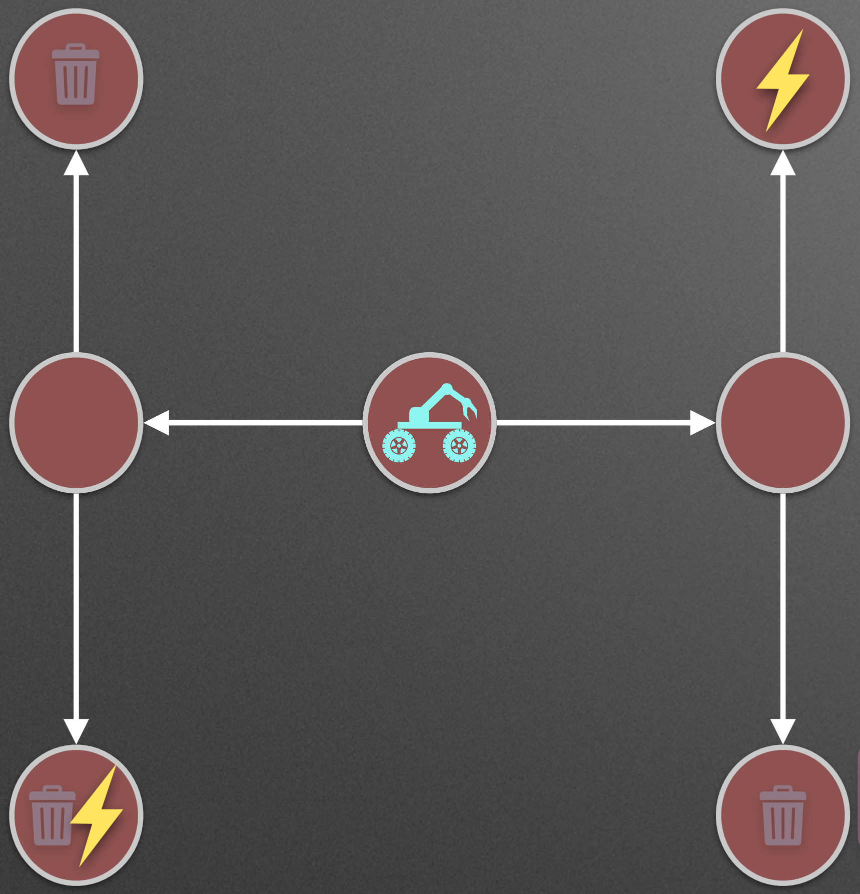
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon_1$

Wants to clear trash-cans

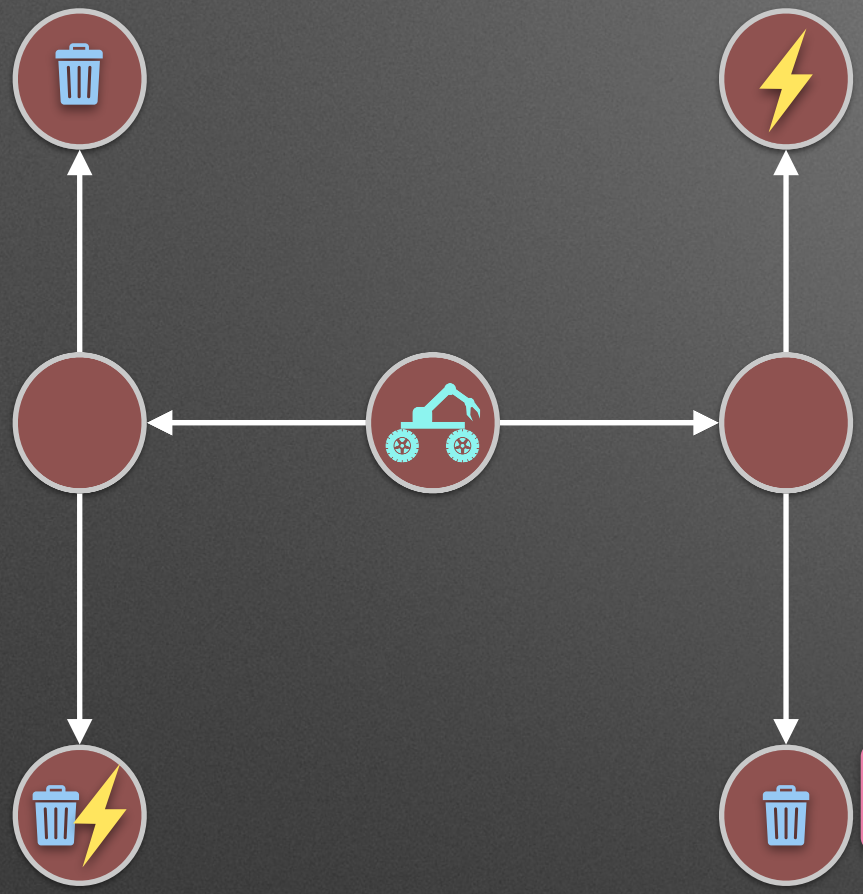
$\frac{1}{2} + \epsilon_2$

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon_1$

Wants to clear trash-cans

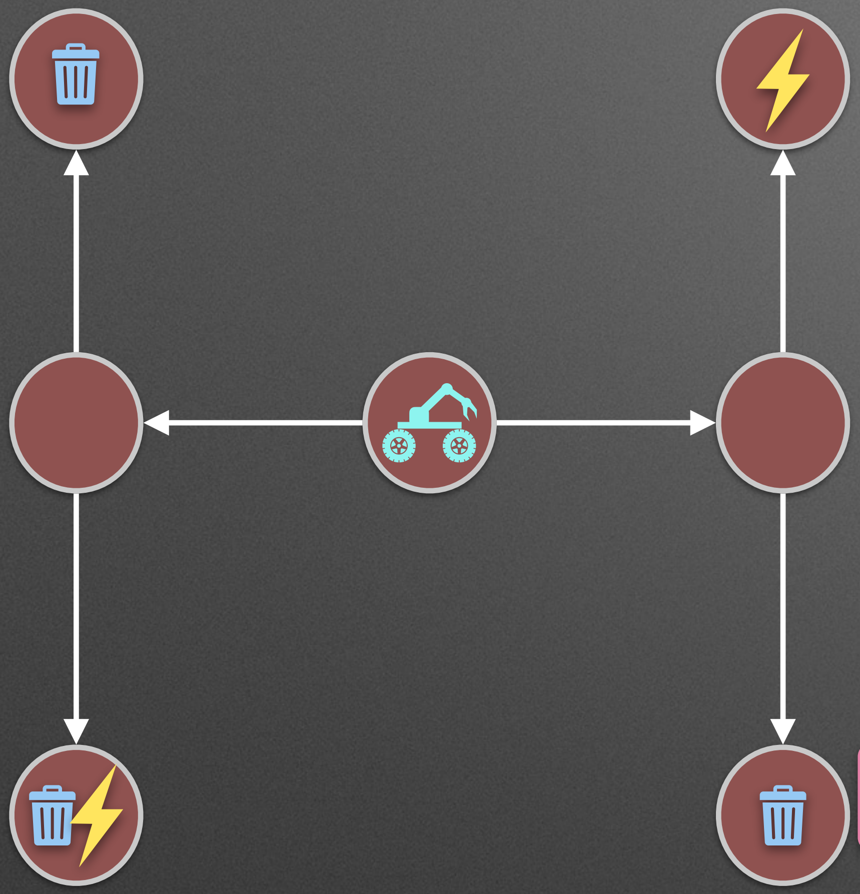
$\frac{1}{2} + \epsilon_2$

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon_1$

Sum of the budgets < 1

$\frac{1}{2} + \epsilon_2$

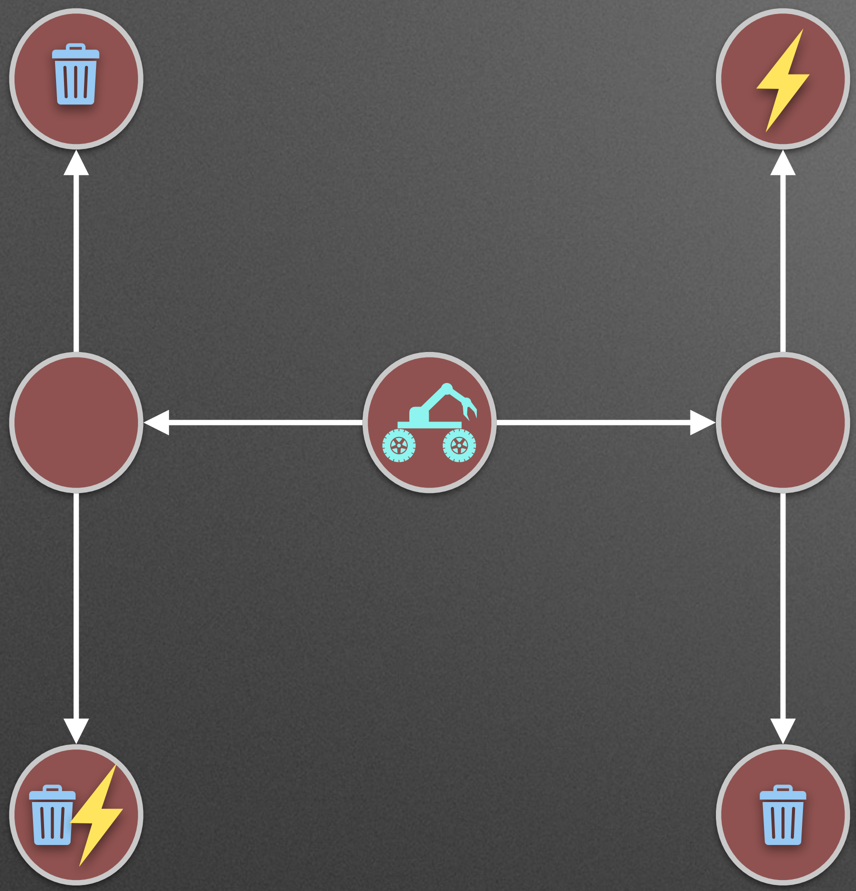
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon$

Wants to clear trash-cans

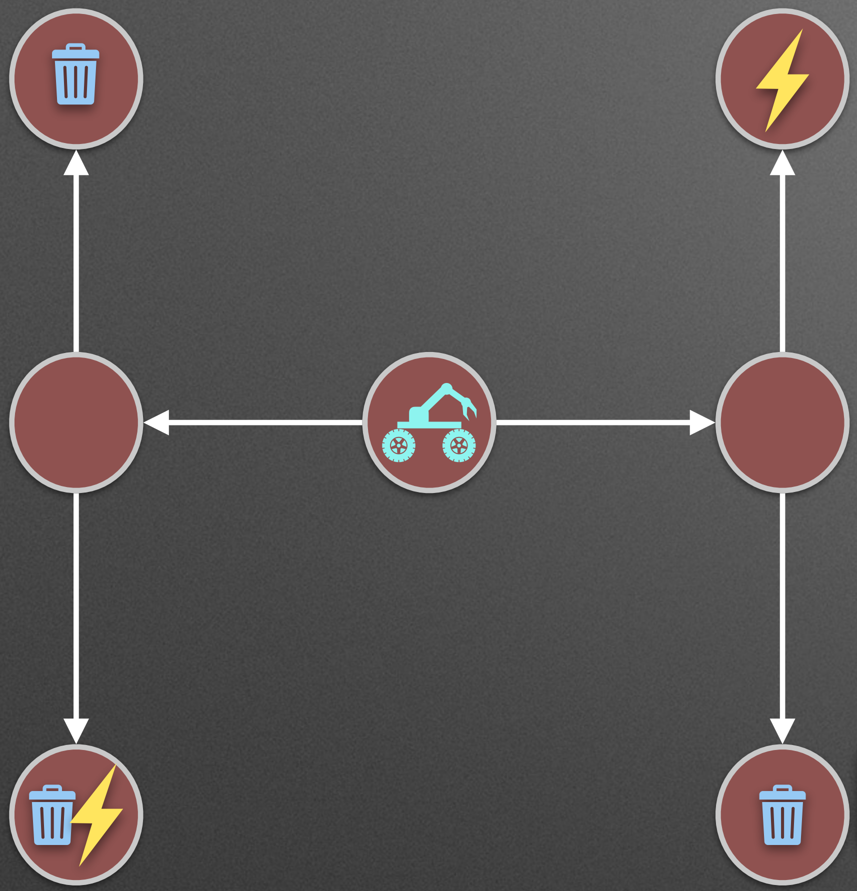
$\frac{3}{4} - \epsilon$

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon$

$\frac{3}{4} - \epsilon$

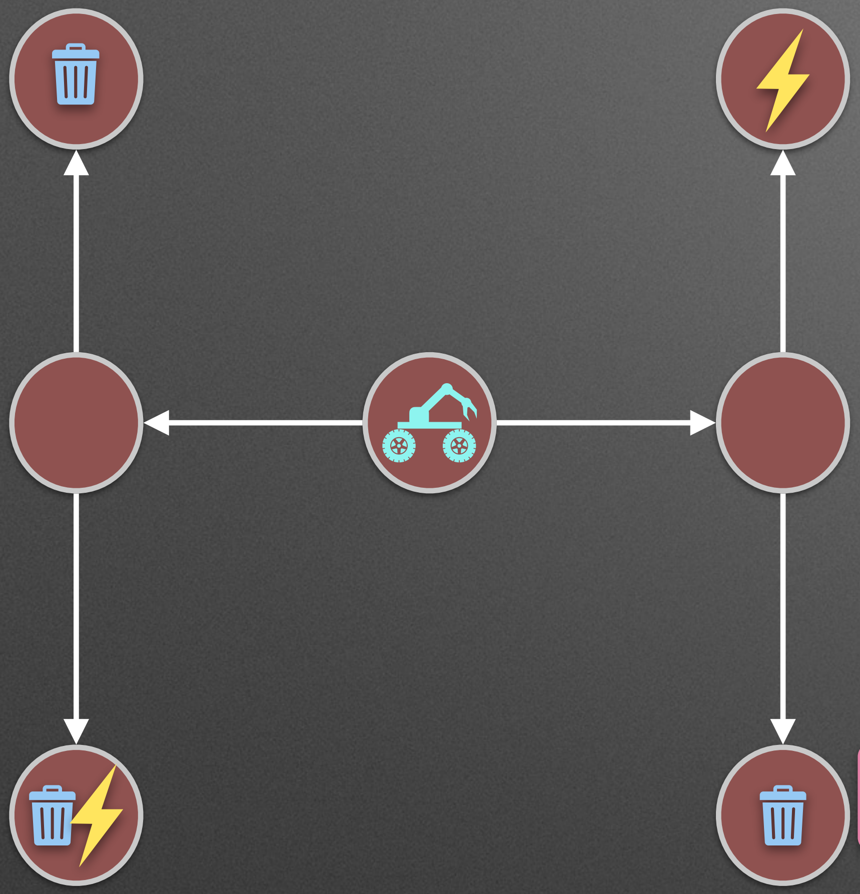
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\epsilon < \frac{1}{4}$

$\frac{1}{4} + \epsilon$

Wants to clear trash-cans

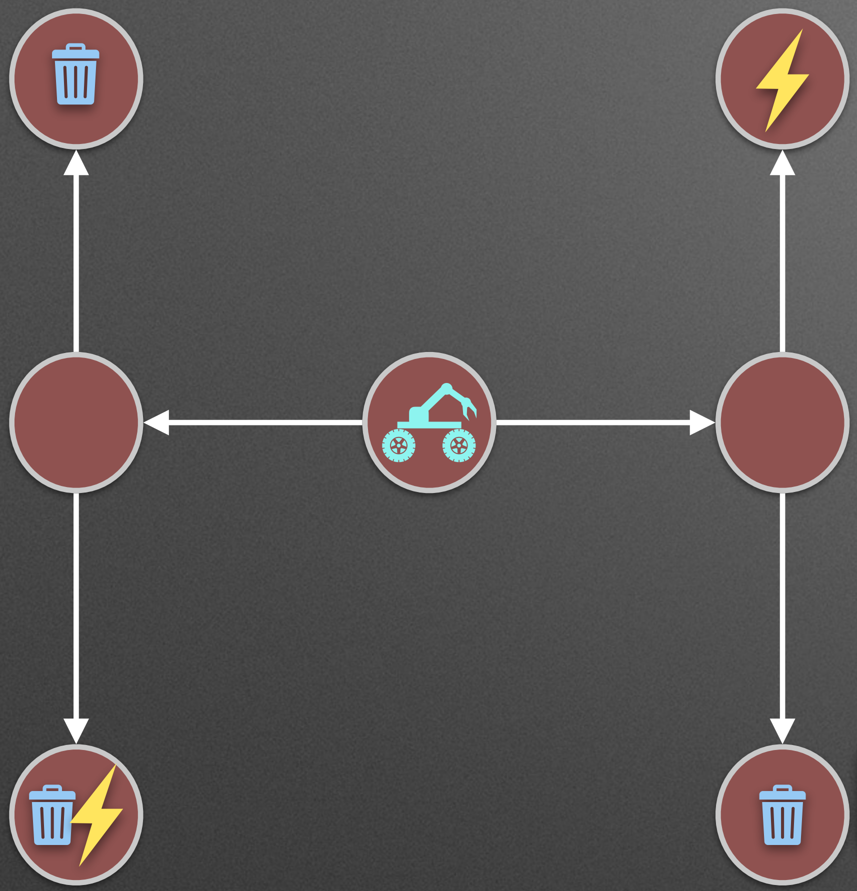
$\frac{3}{4} - \epsilon$

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon$ $\frac{1}{4}$ 0 $\frac{3}{4} - \epsilon$

$\epsilon < \frac{1}{4}$

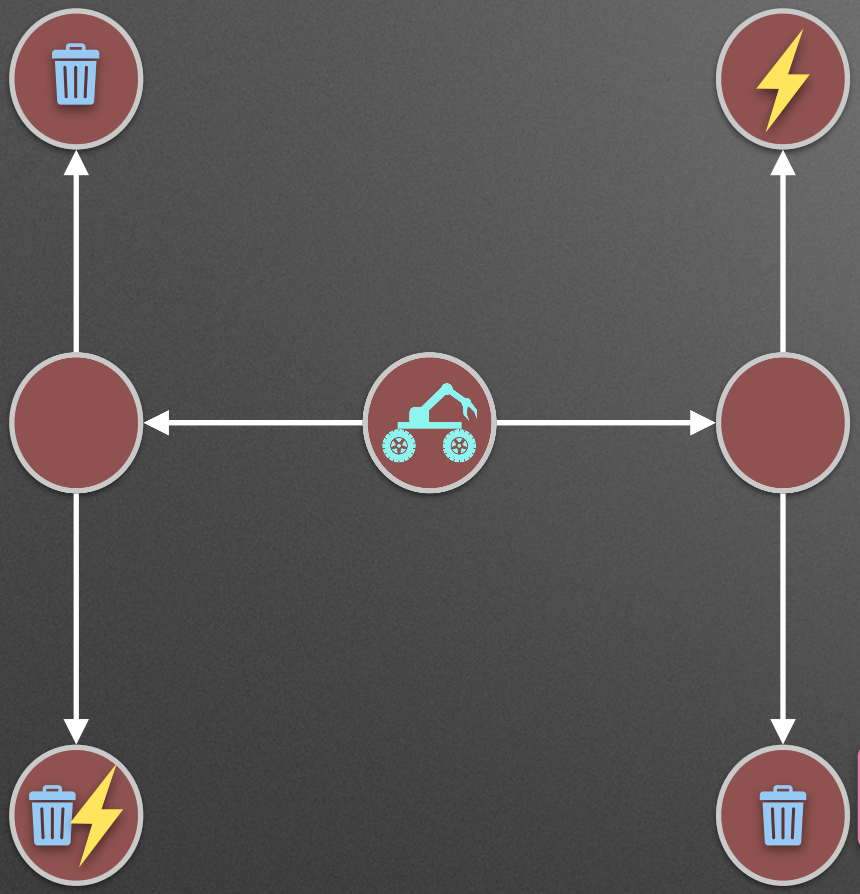
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



ϵ

Wants to clear trash-cans

$\epsilon < \frac{1}{4}$

$1 - \epsilon$

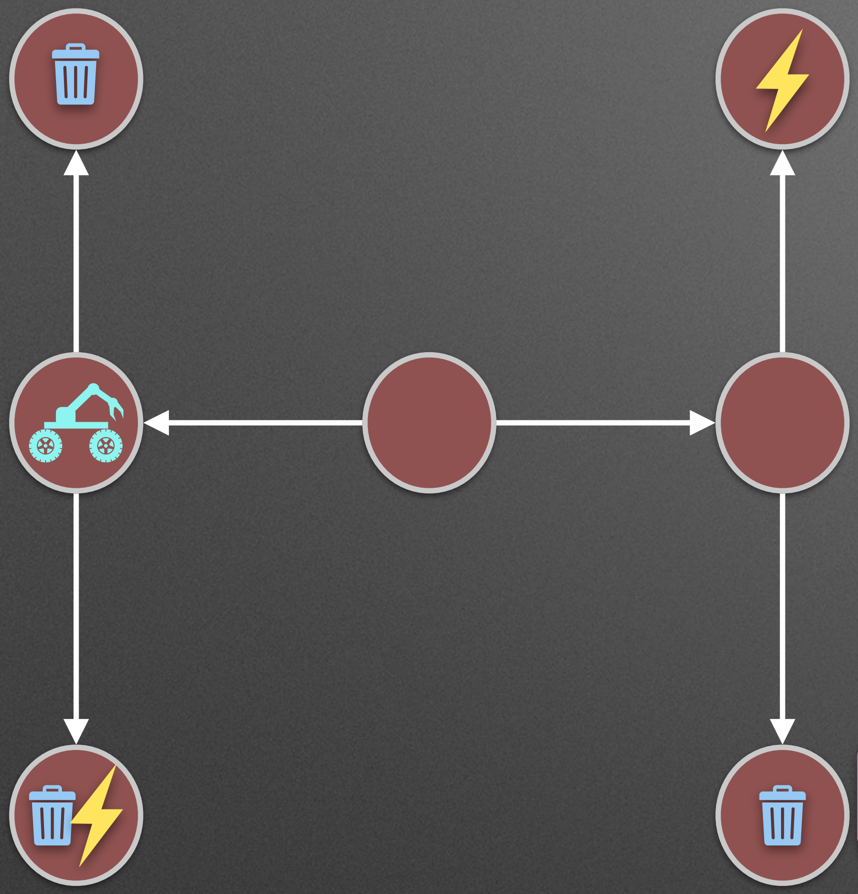
Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling

$\epsilon < \frac{1}{4}$



ϵ

Wants to clear trash-cans

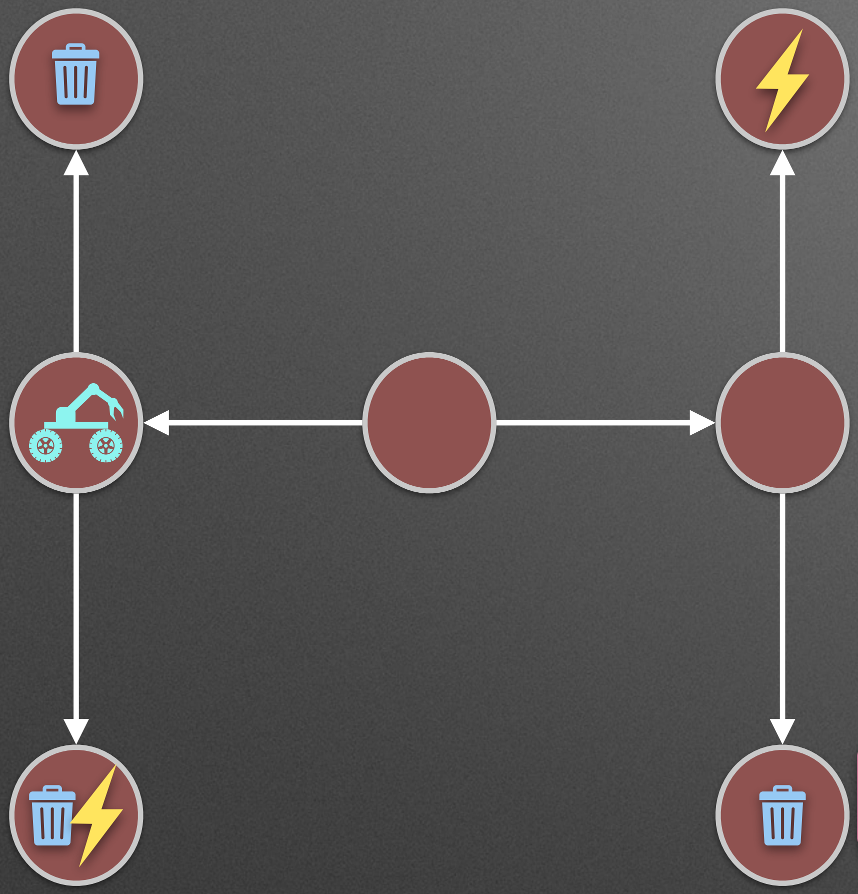
$1 - \epsilon$

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



ϵ ϵ $1-\epsilon$ $1-\epsilon$

$\epsilon < \frac{1}{4}$

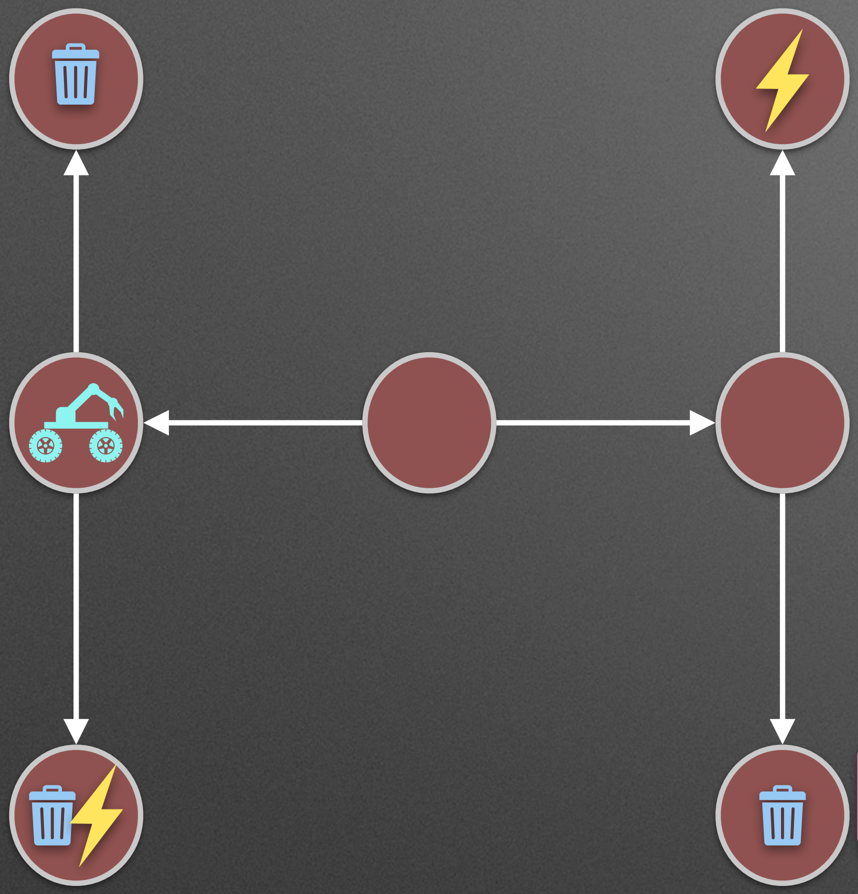
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



1

Wants to clear trash-cans

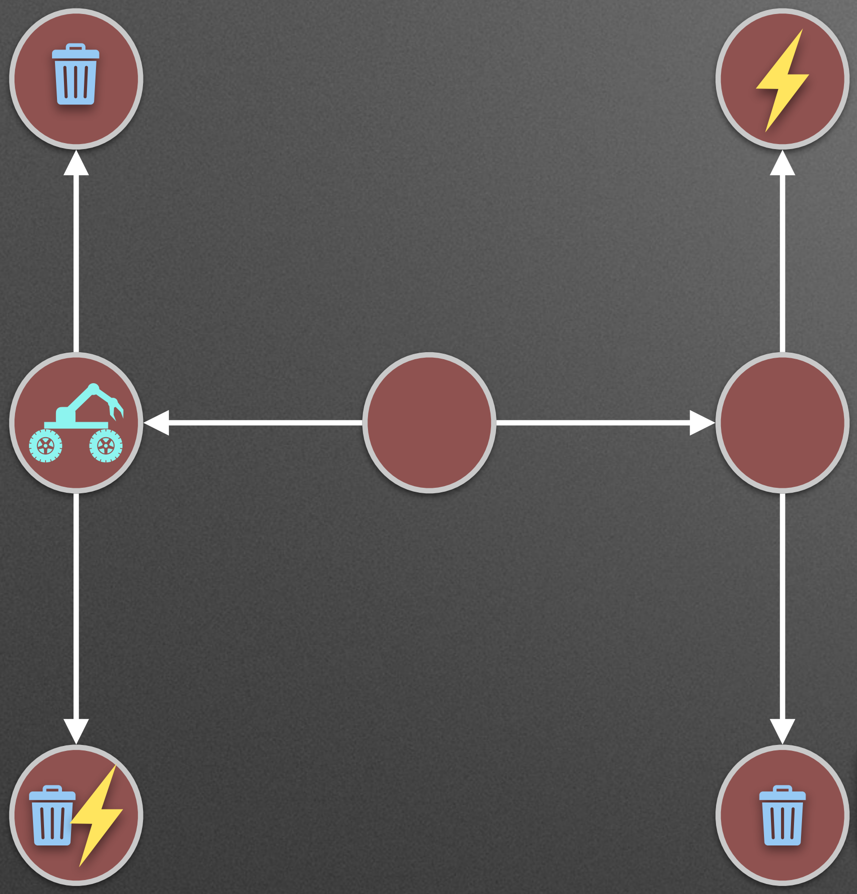
0

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



1

0

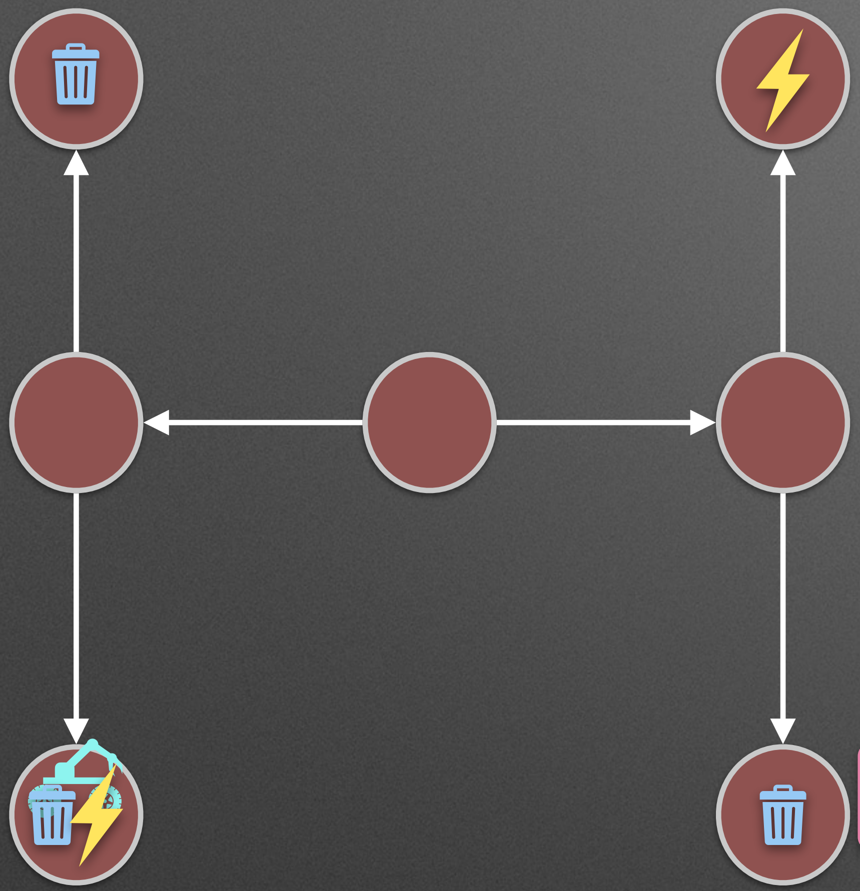
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



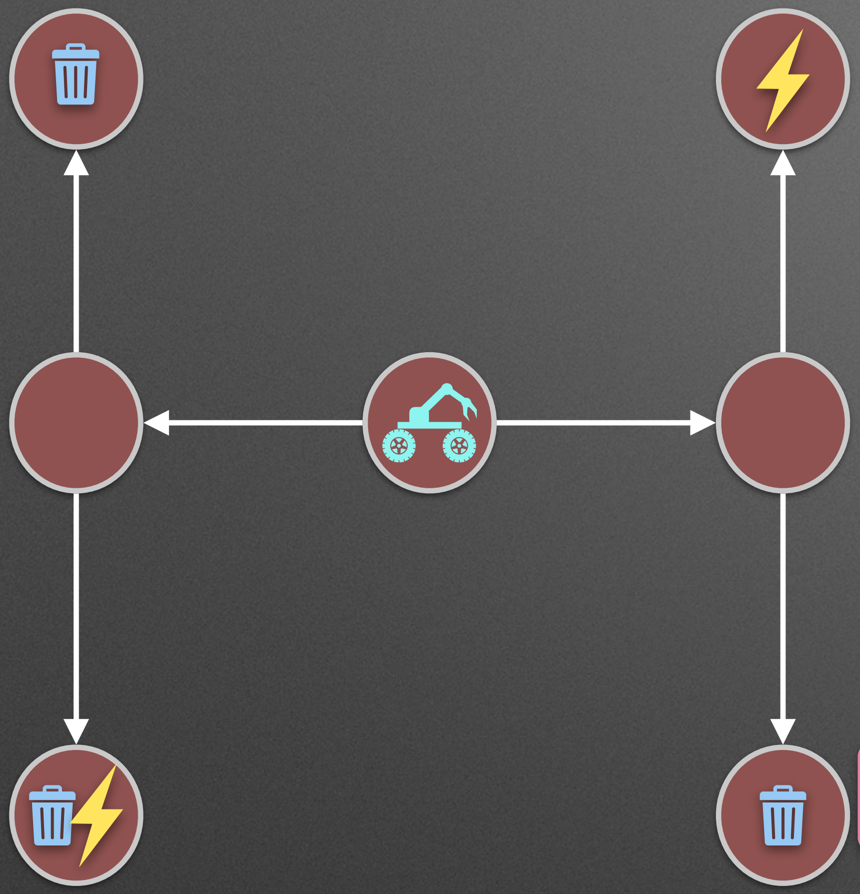
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon_1$

Sum of the budgets < 1

$\frac{1}{2} + \epsilon_2$

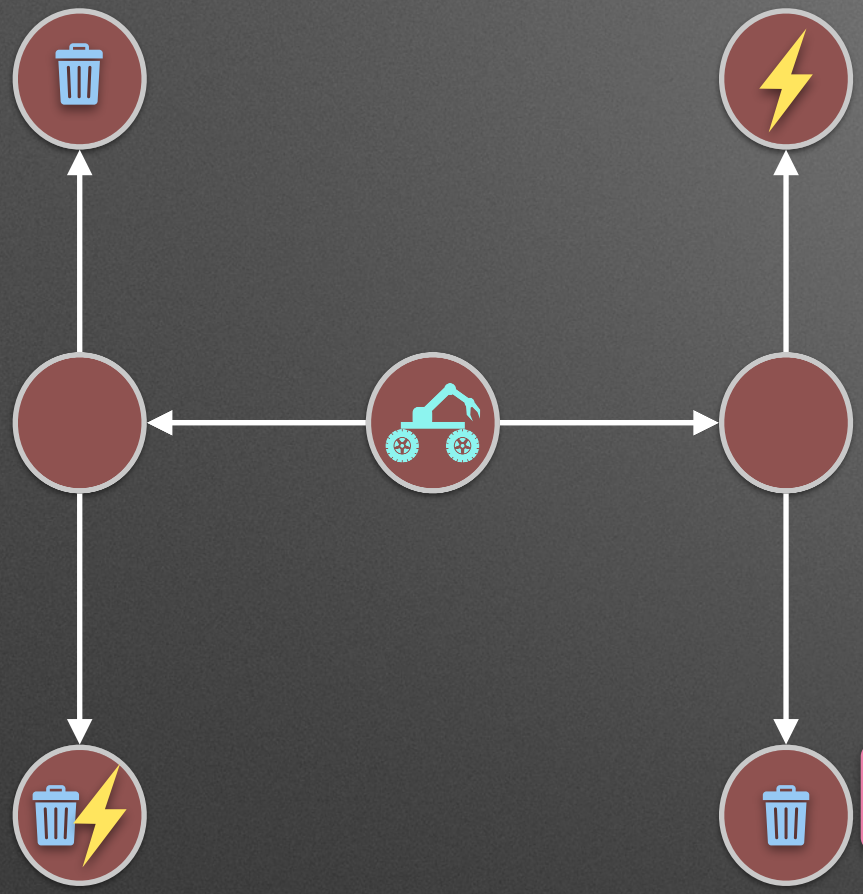
Wants to clear trash-cans

Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his objective with any budget of $> \frac{1}{2}$

Auction-Based Scheduling



$\frac{1}{4} + \epsilon_1$

Sum of the budgets < 1

$\frac{1}{2} + \epsilon_2$

Wants to clear trash-cans

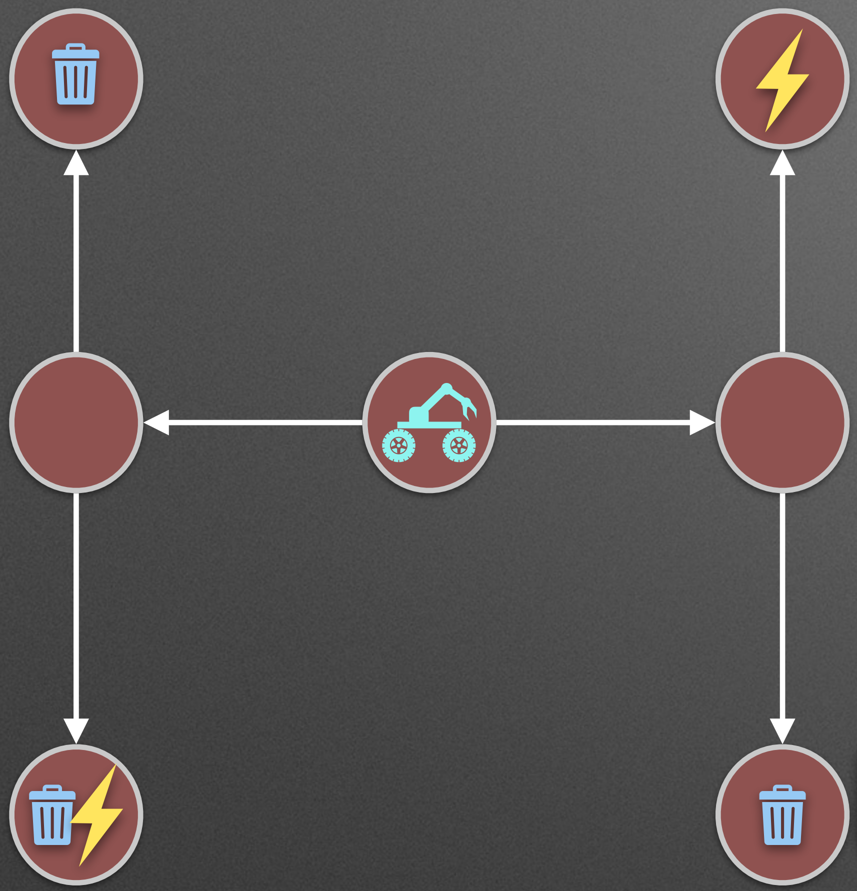
Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his o

"Strong" Synthesis: Winning strategies can be synthesized (in decentralised manner) AND composed without knowing other player's objective

Auction-Based Scheduling



$\frac{1}{4} + \epsilon_1$

Sum of the budgets < 1

$\frac{1}{2} + \epsilon_2$

But what if the sum is ≥ 1

Wants to clear trash-cans

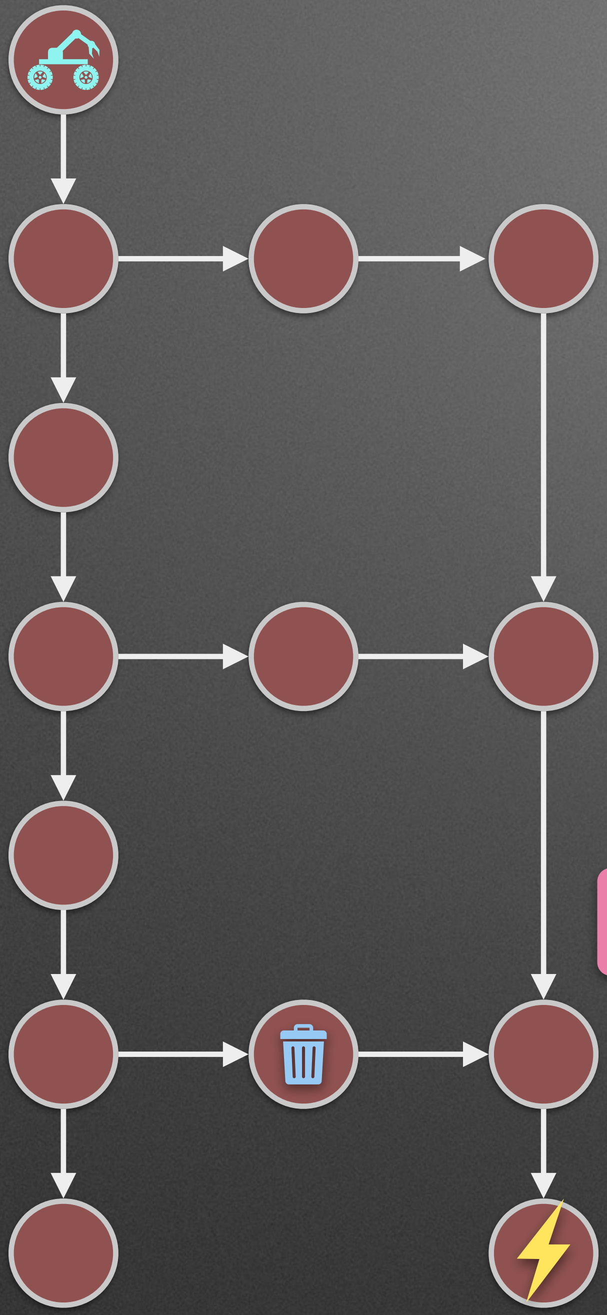
Wants to recharge the battery

Claim 1: Homer can fulfil his objective with any budget of $> \frac{1}{4}$

Claim 2: Marge can fulfil his o

"Strong" Synthesis: Winning strategies can be synthesized (in decentralised manner) AND composed without knowing other player's objective

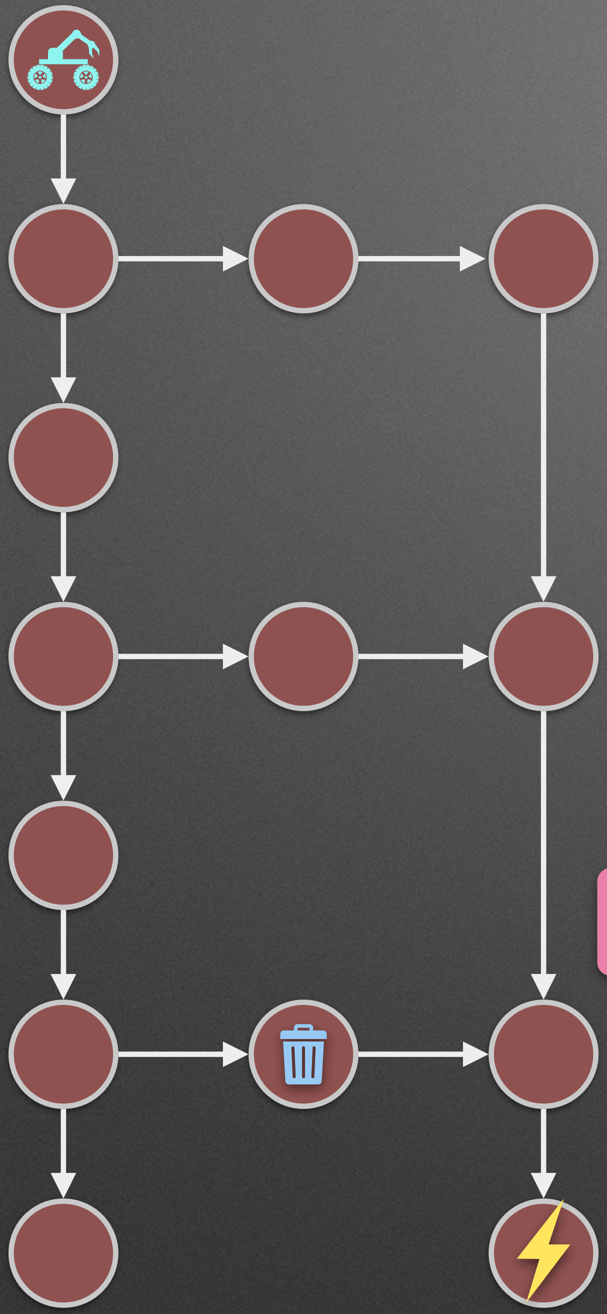
Classification based on Assumption



Wants to clear trash-cans

Wants to recharge the battery

Classification based on Assumption



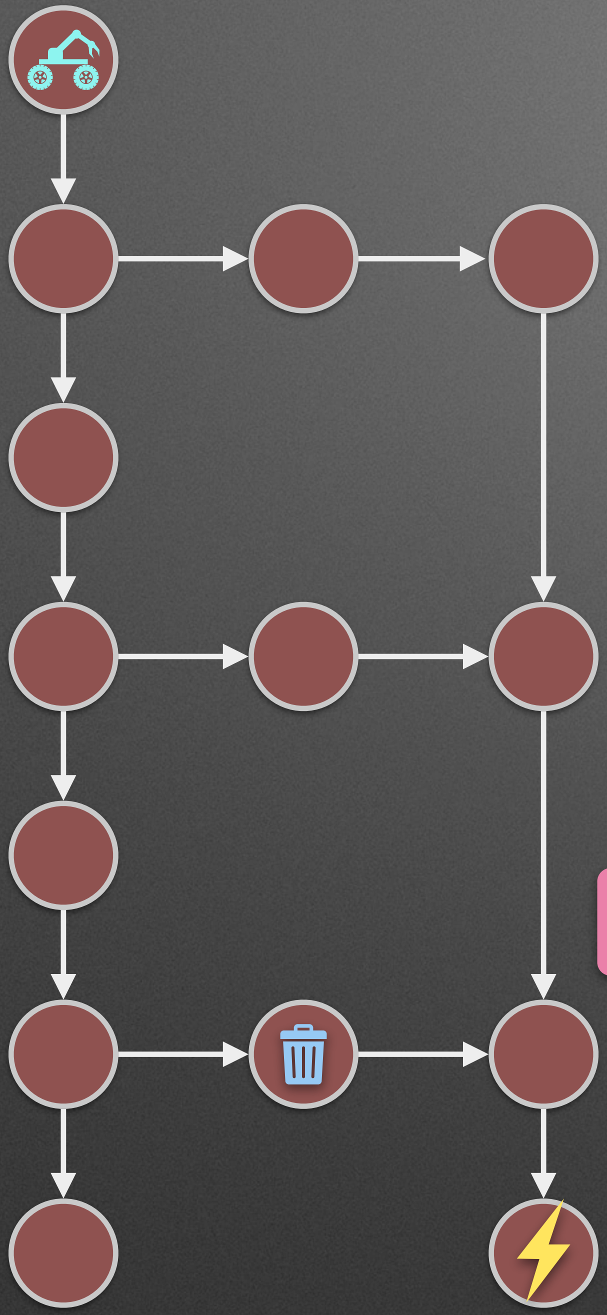
$\frac{7}{8} + \epsilon_1$

Wants to clear trash-cans

$\frac{1}{8} + \epsilon_2$

Wants to recharge the battery

Classification based on Assumption



$\frac{7}{8} + \epsilon_1$

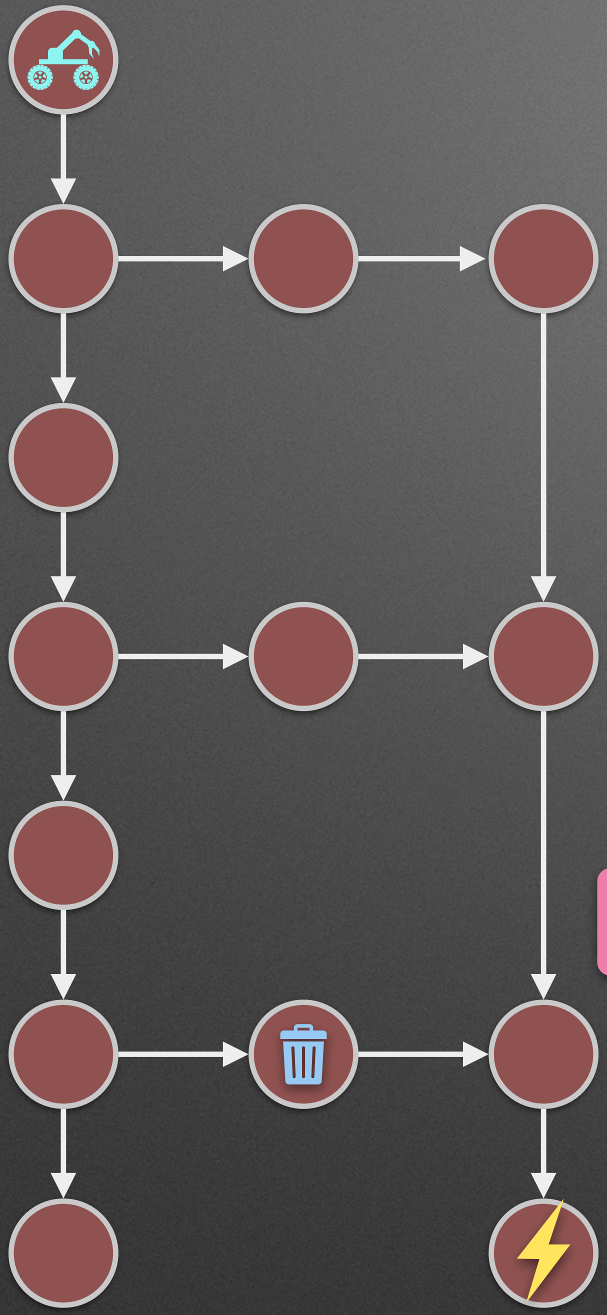
Sum of the budgets > 1

$\frac{1}{8} + \epsilon_2$

Wants to clear trash-cans

Wants to recharge the battery

Classification based on Assumption



$\frac{7}{8} + \epsilon_1$

Sum of the budgets > 1

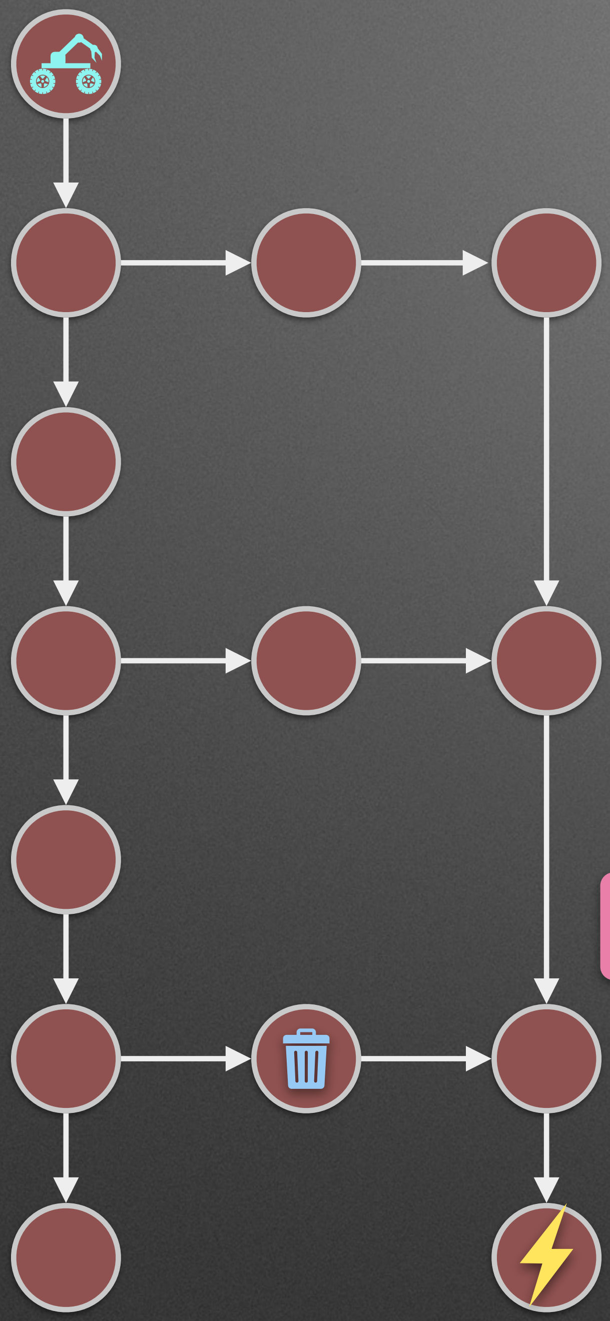
$\frac{1}{8} + \epsilon_2$

FAILS!!

Wants to clear trash-cans

Wants to recharge the battery

Classification based on Assumption



$$\frac{7}{8} + \epsilon_1$$

Sum of the budgets > 1

$$\frac{1}{8} + \epsilon_2$$

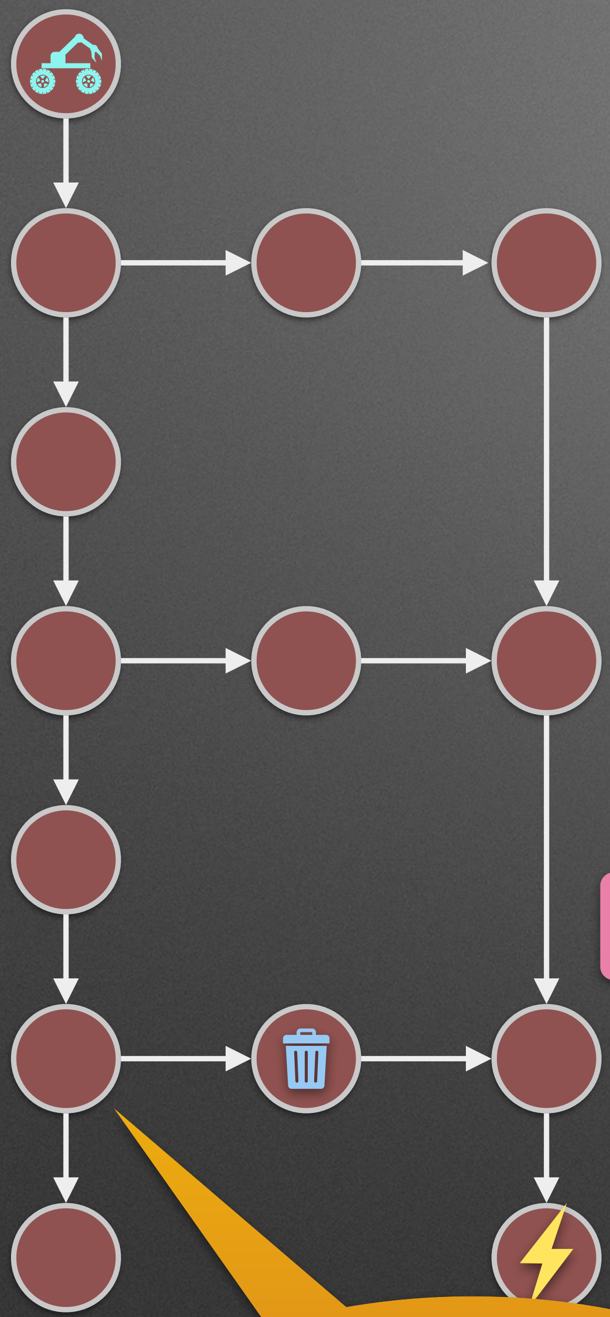
FAILS!!

Wants to clear trash-cans

Wants to recharge the battery

Solution? We let them know what the other one's objective is, and that they play "rationally"

Classification based on Assumption



Homer knows Marge wouldnt take this edge and vice-versa

$\frac{7}{8} + \epsilon_1$

Sum of the budgets > 1

$\frac{1}{8} + \epsilon_2$

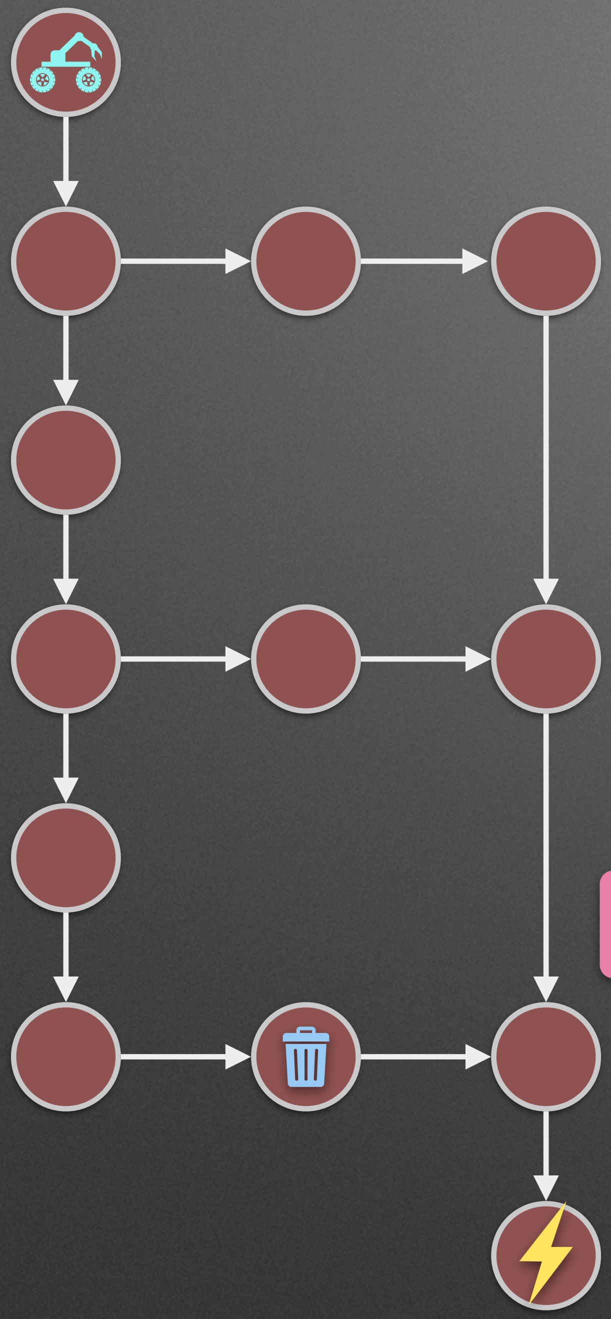
FAILS!!

Wants to clear trash-cans

Wants to recharge the battery

Solution? We let them know what the other one's objective is, and that they play "rationally"

Classification based on Assumption



$\frac{7}{8} + \epsilon_1$

Sum of the budgets > 1

$\frac{1}{8} + \epsilon_2$

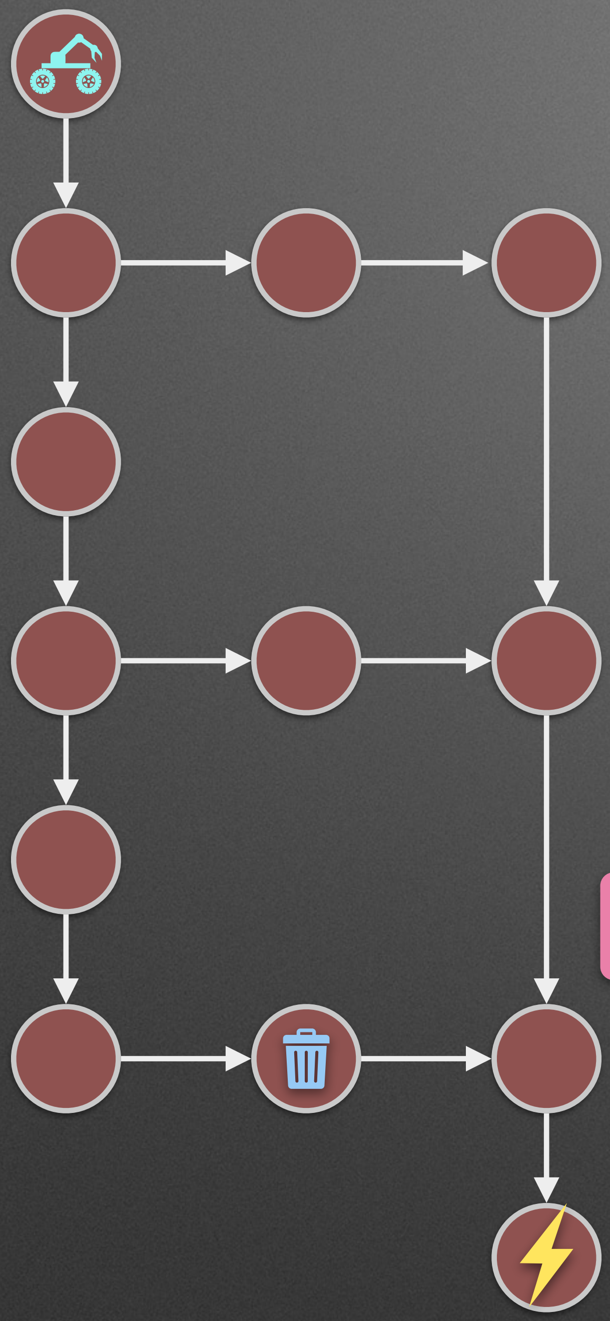
FAILS!!

Wants to clear trash-cans

Wants to recharge the battery

Solution? We let them know what the other one's objective is, and that they play "rationally"

Classification based on Assumption



$> \frac{3}{4}$

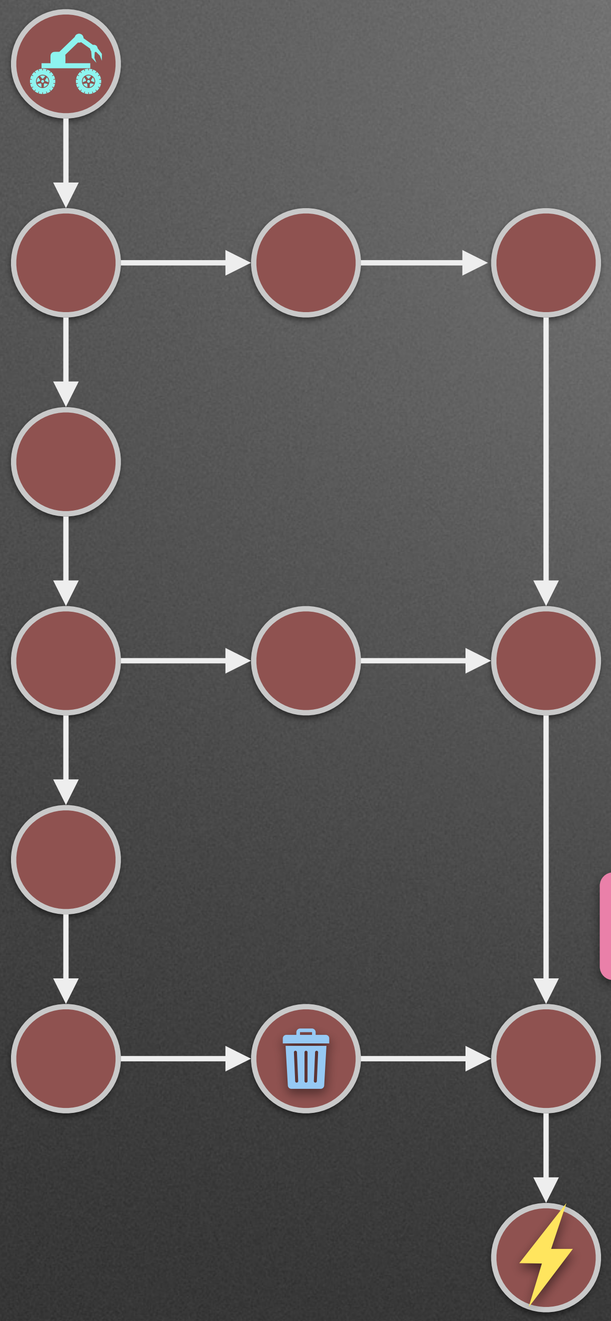
Wants to clear trash-cans

> 0

Wants to recharge the battery

Solution? We let them know what the other one's objective is, and that they play "rationally"

Classification based on Assumption



$> \frac{3}{4}$

Sum of the budgets < 1

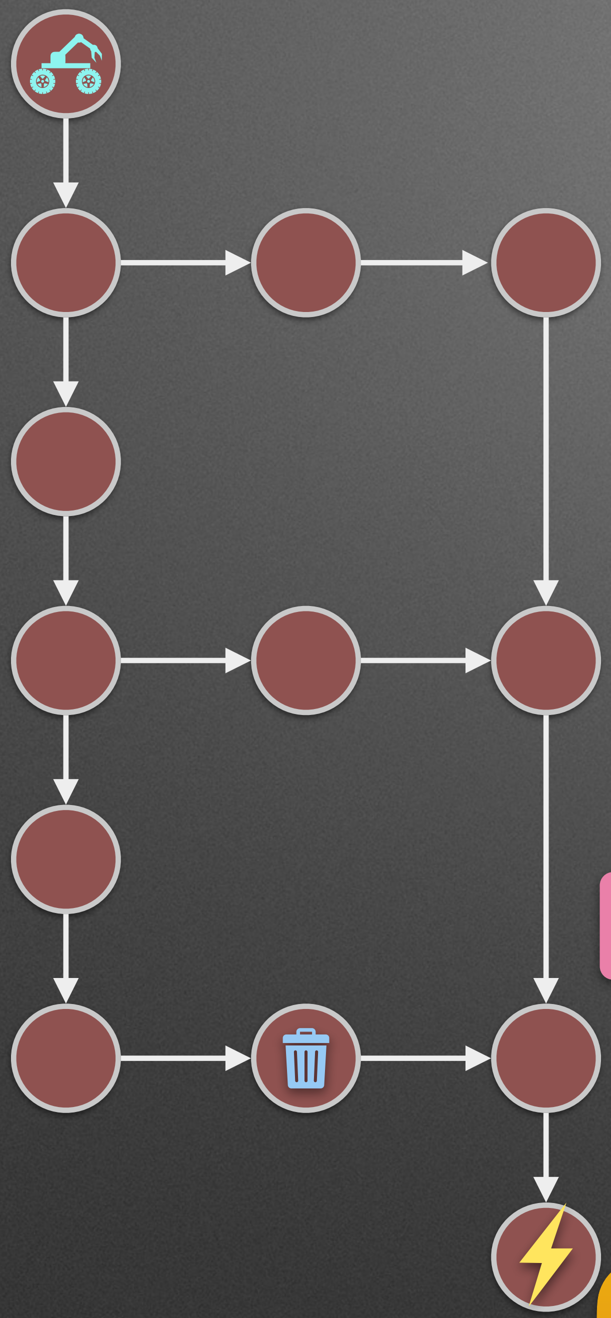
> 0

Wants to clear trash-cans

Wants to recharge the battery

Solution? We let them know what the other one's objective is, and that they play "rationally"

Classification based on Assumption



$> \frac{3}{4}$

Sum of the budgets < 1

> 0

Wants to clear trash-cans

Wants to recharge the battery

Solution? We let them know what the

"Assume-admissible" Synthesis: Winning strategies can be synthesized (in decentralised manner) AND composed after knowing the other player's objective, and their rational behaviour

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives

Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives
Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives

Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

Strong Synthesis:

- No knowledge about objective, no assumption about behaviour, quite flexible if solution exists
- Restricted solution space

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives
Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

Strong Synthesis:

- No knowledge about objective, no assumption about behaviour, quite flexible if solution exists
- Restricted solution space

Assume-admissible Synthesis:

- Knowledge about objective, rational behavioural assumption, less flexible
- Solution space expanded

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives

Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

Strong Synthesis:

- No knowledge about objective, no assumption about behaviour, quite flexible if solution exists
- Restricted solution space

Assume-admissible Synthesis:

- Knowledge about objective, rational behavioural assumption, less flexible
- Solution space expanded

Assume-guarantee synthesis:

- Knowledge about objective, exact behavioural assumption, least flexible
- Solution space expanded more

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives

Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

Tradeoff between knowledge/assumption/flexibility vs solution

Strong Synthesis:

- No knowledge about objective, no assumption about behaviour, quite flexible if solution exists
- Restricted solution space

Assume-admissible Synthesis:

- Knowledge about objective, rational behavioural assumption, less flexible
- Solution space expanded

Assume-guarantee synthesis:

- Knowledge about objective, exact behavioural assumption, least flexible
- Solution space expanded more

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives
Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

(Some of the) Theorems (Avni, Mallik, and S.' 24):

Tradeoff between knowledge/assumption/flexibility vs solution

Strong Synthesis:
- No knowledge about objective, no assumption about behaviour, quite flexible if solution exists
- Restricted solution space

Assume-admissible Synthesis:
- Knowledge about objective, rational behavioural assumption, less flexible
- Solution space expanded

Assume-guarantee synthesis:
- Knowledge about objective, exact behavioural assumption, least flexible
- Solution space expanded more

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives
Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

(Some of the) Theorems (Avni, Mallik, and S.' 24):

- Strong Synthesis can be solved in $NP \cap coNP$, and in PTIME for binary graphs. Moreover, for SCC and Buchi objectives, strong synthesis is always possible.

Tradeoff between knowledge/assumption/flexibility vs solution

Strong Synthesis:

- No knowledge about objective, no assumption about behaviour, quite flexible if solution exists
- Restricted solution space

Assume-admissible Synthesis:

- Knowledge about objective, rational behavioural assumption, less flexible
- Solution space expanded

Assume-guarantee synthesis:

- Knowledge about objective, exact behavioural assumption, least flexible
- Solution space expanded more

Auction-based Scheduling Problem - (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives
Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil their own objectives.

For a given set of assumptions about the other controller

(Some of the) Theorems (Avni, Mallik, and S.' 24):

- Strong Synthesis can be solved in $NP \cap coNP$, and in PTIME for binary graphs. Moreover, for SCC and Buchi objectives, strong synthesis is always possible.
- Strong-synthesised controller in redundant-vertex-removed graph (if exists) => Assume-admissible controller for original graph [Sound solution but not complete]

....

Tradeoff between knowledge/assumption/flexibility vs solution

Strong Synthesis:

- No knowledge about objective, no assumption about behaviour, quite flexible if solution exists
- Restricted solution space

Assume-admissible Synthesis:

- Knowledge about objective, rational behavioural assumption, less flexible
- Solution space expanded

Assume-guarantee synthesis:

- Knowledge about objective, exact behavioural assumption, least flexible
- Solution space expanded more

Auction-based Scheduling Problem – (somewhat) Formally

Input: A graph-arena G and two non-conflicting objectives

Output: Yes, if we can synthesise two controllers and schedule them via bidding so that they fulfil thei own objectives.

For a given set of assumptions about the other controller

(Some of the) Theorems (Avni, Mallik, and S.' 24):

- Synthesis can be solved in $NP \cap coNP$, and in PTIME for binary
  for SCC and Buchi objectives, strong synthesis is always possible.
- redundant-vertex-removed graph (if exists) =>
  graph [Sound solution but not complete]
  ..

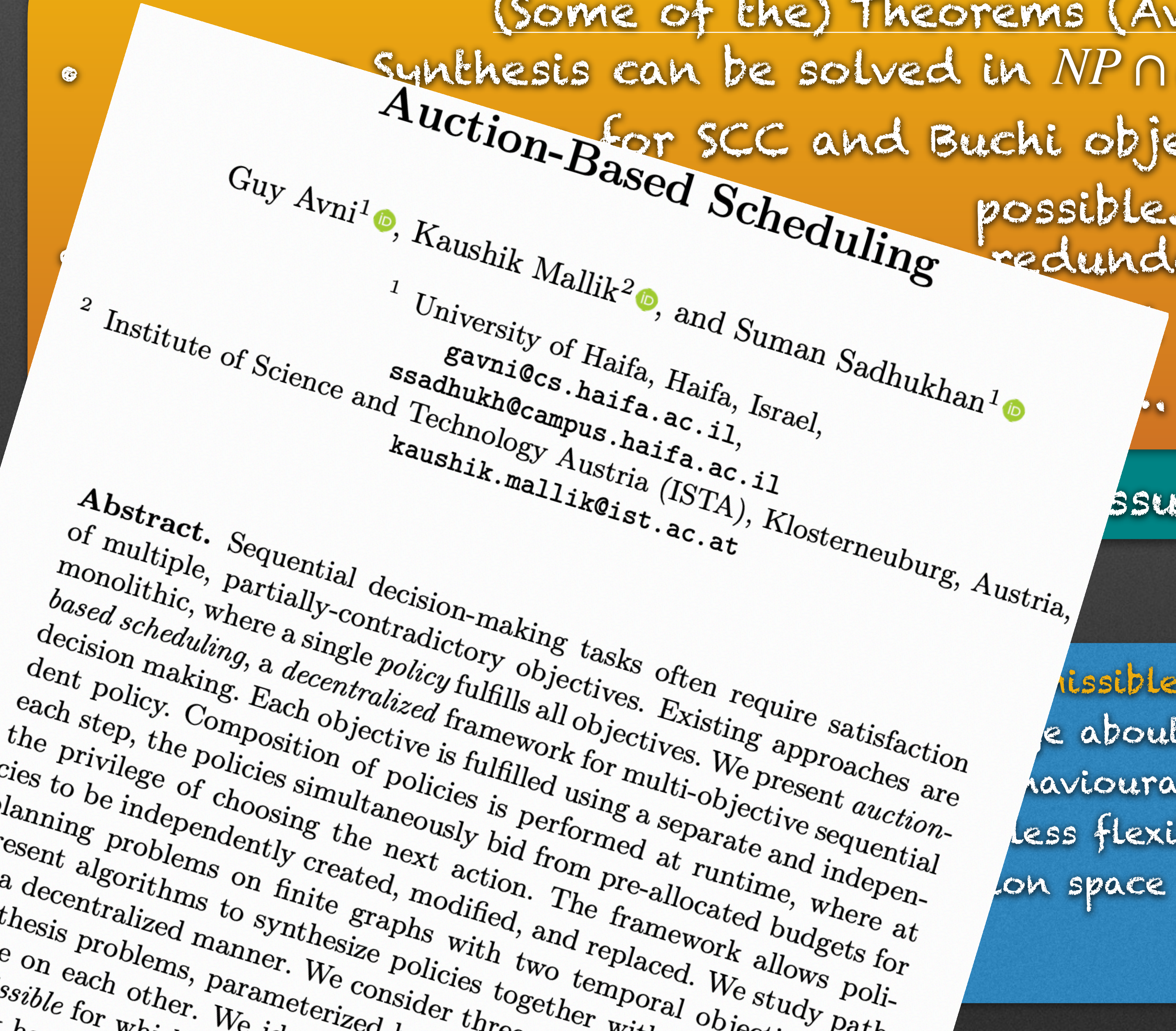
ssumption/flexibility vs solution

issible Synthesis:

e about objective,
aviour al assumption,
less flexible
on space expanded

Assume-guarantee synthesis:

- Knowledge about objective, exact behavioural assumption, least flexible
- Solution space expanded more



Auction-Based Scheduling

Guy Avni[1], Kaushik Mallik[2], and Suman Sadhukhan[1]

[1] University of Haifa, Haifa, Israel,
gavni@cs.haifa.ac.il,
ssadhukh@campus.haifa.ac.il

[2] Institute of Science and Technology Austria (ISTA), Klosterneuburg, Austria,
kaushik.mallik@ist.ac.at

Abstract. Sequential decision-making tasks often require satisfaction of multiple, partially-contradictory objectives. Existing approaches are monolithic, where a single policy fulfills all objectives. We present *auction-based scheduling*, a *decentralized* framework for multi-objective sequential decision making. Each objective is fulfilled using a separate and independent policy. Composition of policies is performed at runtime, where at each step, the policies simultaneously bid from pre-allocated budgets for the privilege of choosing the next action. The framework allows policies to be independently created, modified, and replaced. We study path lanning problems on finite graphs with two temporal objectives resent algorithms to synthesize policies together with a decentralized manner. We consider thre thesis problems, parameterized e on each other. We id *ssible* for whi

Take Away - Part II

Decentralised Synthesis Problem:

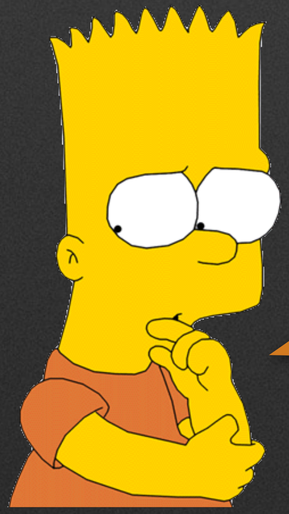
Given a game arena $\mathscr{G}$, two overlapping winning objectives, when can we synthesize two controllers and schedule them so that both the objectives are satisfied.

Take Away - Part II

Decentralised Synthesis Problem:

Given a game arena $\mathscr{G}$, two overlapping winning objectives, when can we synthesize two controllers and schedule them so that both the objectives are satisfied.

- We propose a solution using bidding games
- We identify where some solution always exists, and analyse complexity for finding solutions for qualitative objectives
- We show knowledge/assumption/flexibility tradeoff with solution space



Future work:

- Quantitative objectives?
- Complete solutions for assume-admissibility, assume-guarantee?
- Multi-player bidding?

Recap: Bidding Games on Graphs

In Theory

Studied Richman first-price discrete parity bidding games:

- Fixed-point algorithm gives nice structure to the threshold budgets, and optimal bids
- Showed membership in NP ∩ coNP by using that structure and algorithm for turn-based parity games

In Practice

Auction-Based Scheduling:

- Proposed a solution for decentralised synthesis problem using bidding for scheduling mechanism
- Studied where such solution always exists (graph arena, objectives), where it gives sound-but-incomplete solution , and complexity results
- Tradeoff between solution space and behavioural solution

Recap: Bidding Games on Graphs

In Theory

Studied Richman first-price discrete parity bidding games:

- Fixed-point algorithm gives nice structure to the threshold budgets, and optimal bids
- Showed membership in NP ∩ coNP by using that structure and algorithm for turn-based parity games

In Practice

Auction-Based Scheduling:

- Proposed a solution for decentralised synthesis problem using bidding for scheduling mechanism
- Studied where such solution always exists (graph arena, objectives), where it gives sound-but-incomplete solution , and complexity results
- Tradeoff between solution space and behavioural solution

Thanks!