

The Gittins Index:

A Design Principle for Decision Making Under Uncertainty

Ziv Scully

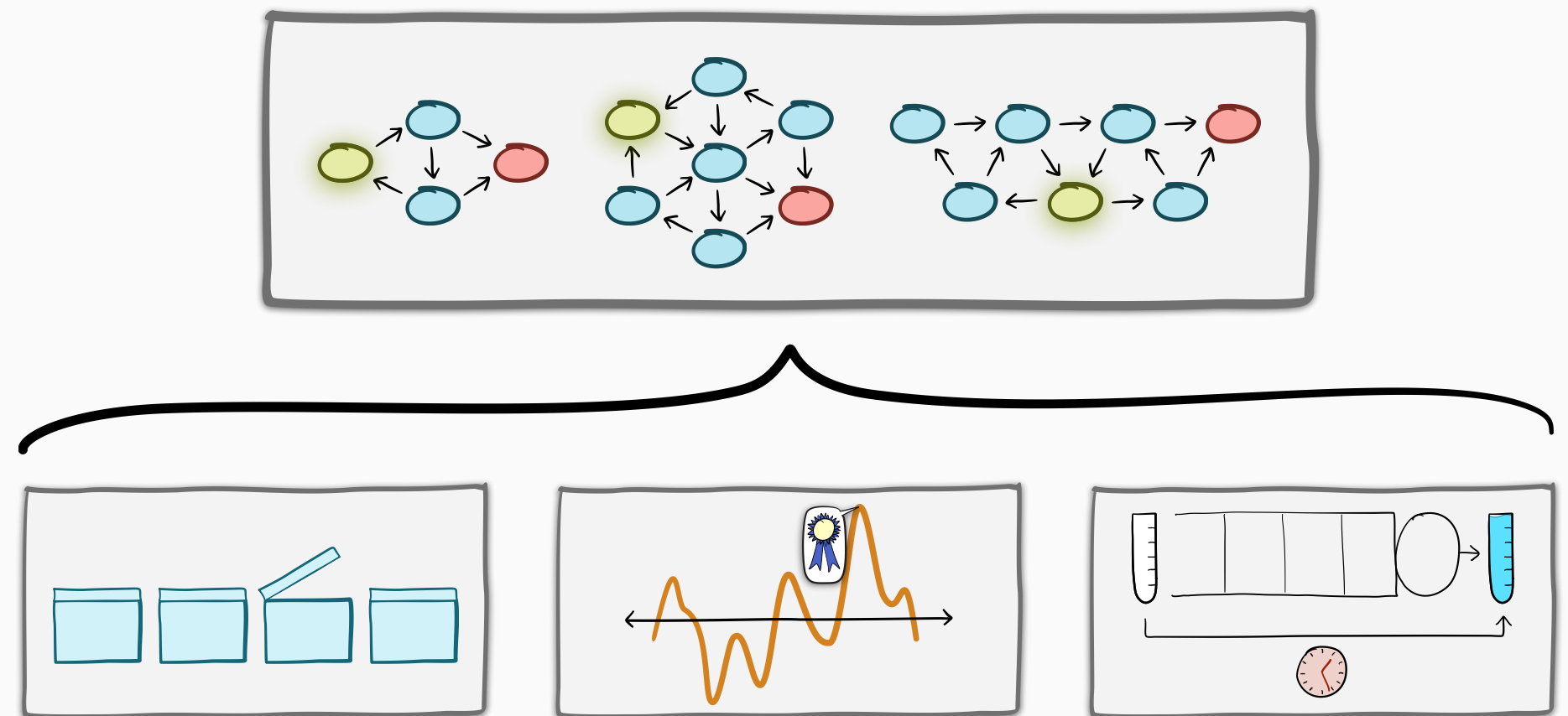
Cornell ORIE

<https://ziv.codes>

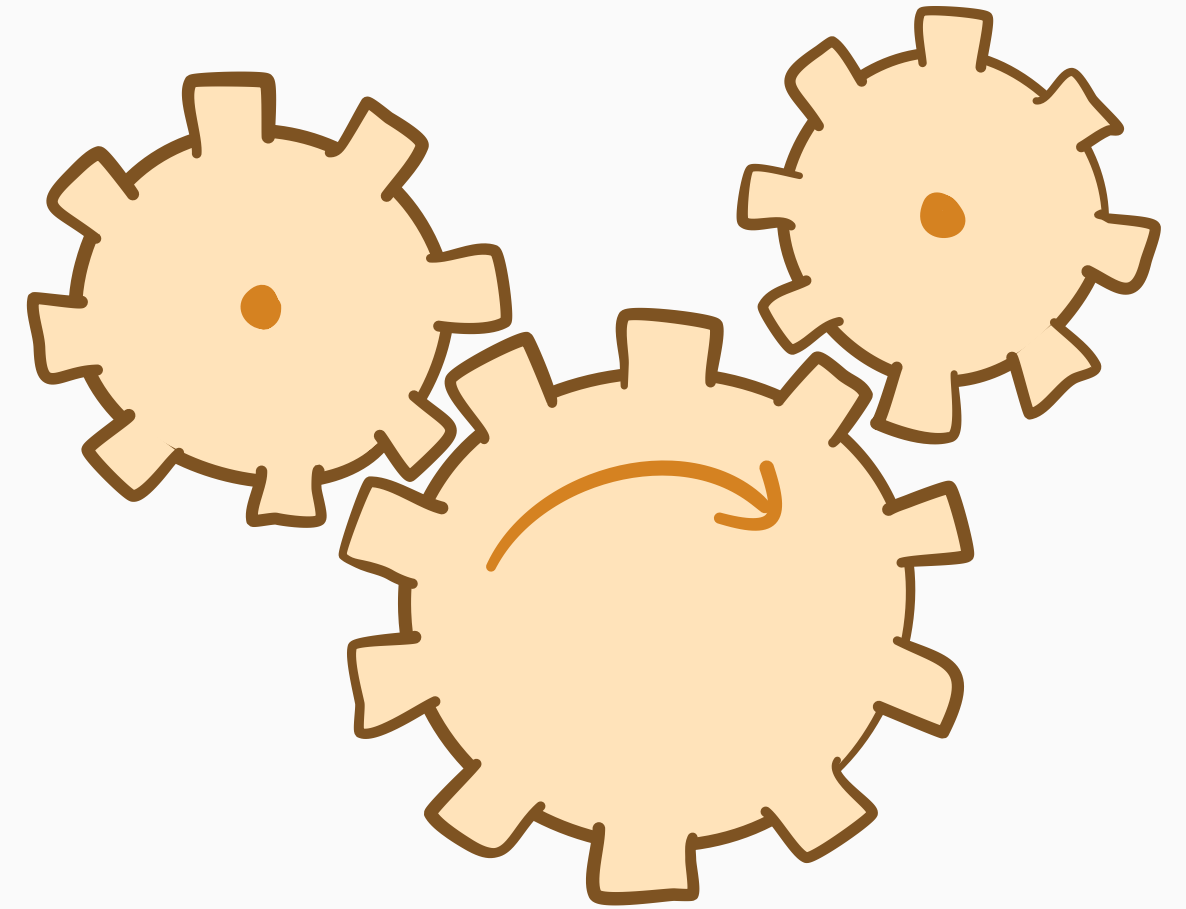
Alex Terenin

Cornell CDSSES

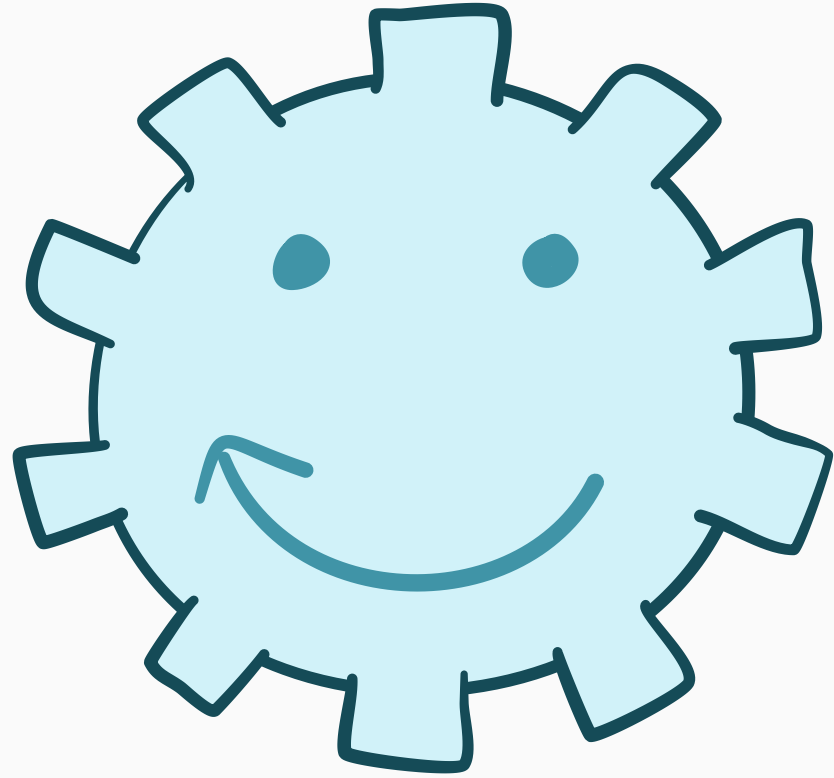
<https://avt.im>



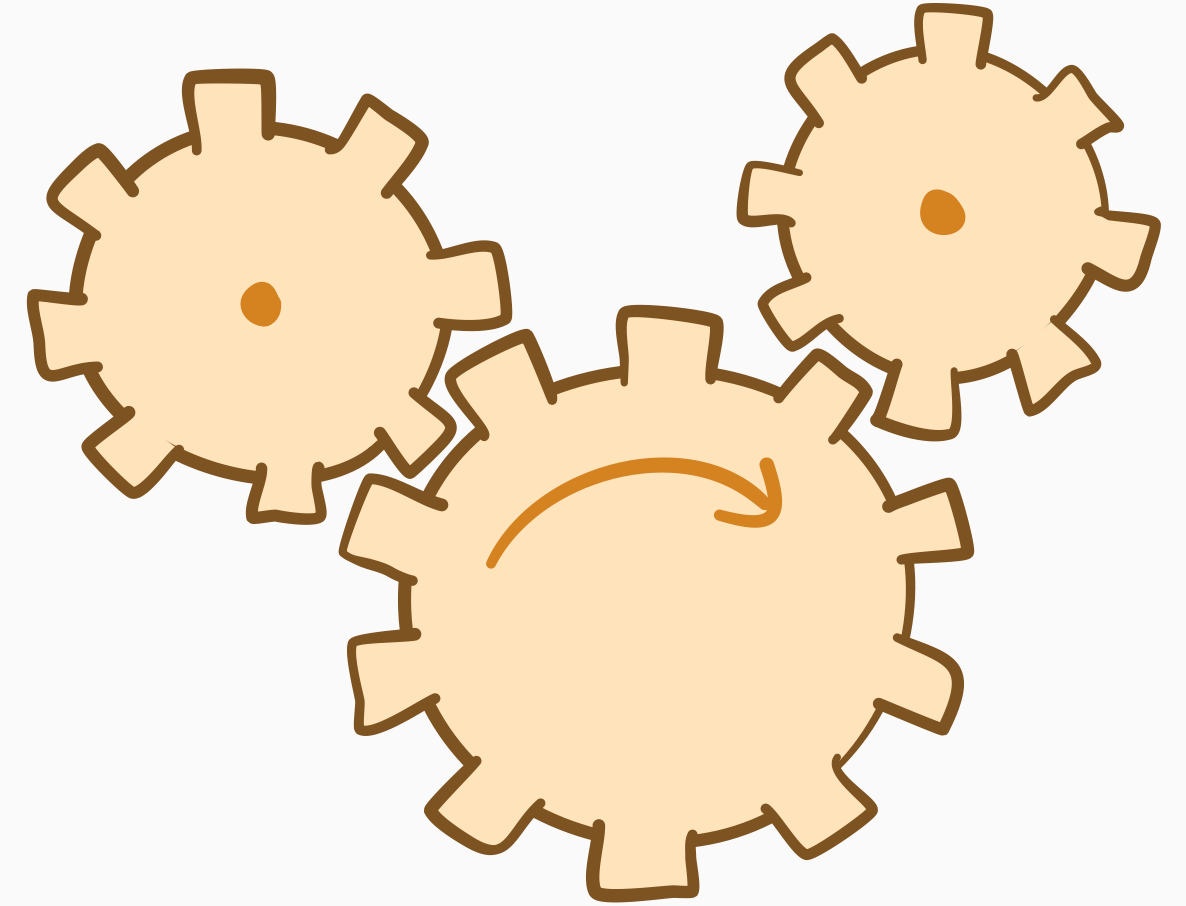
Practical problems



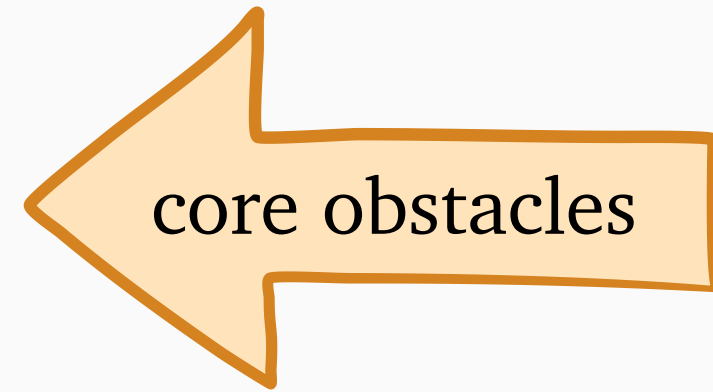
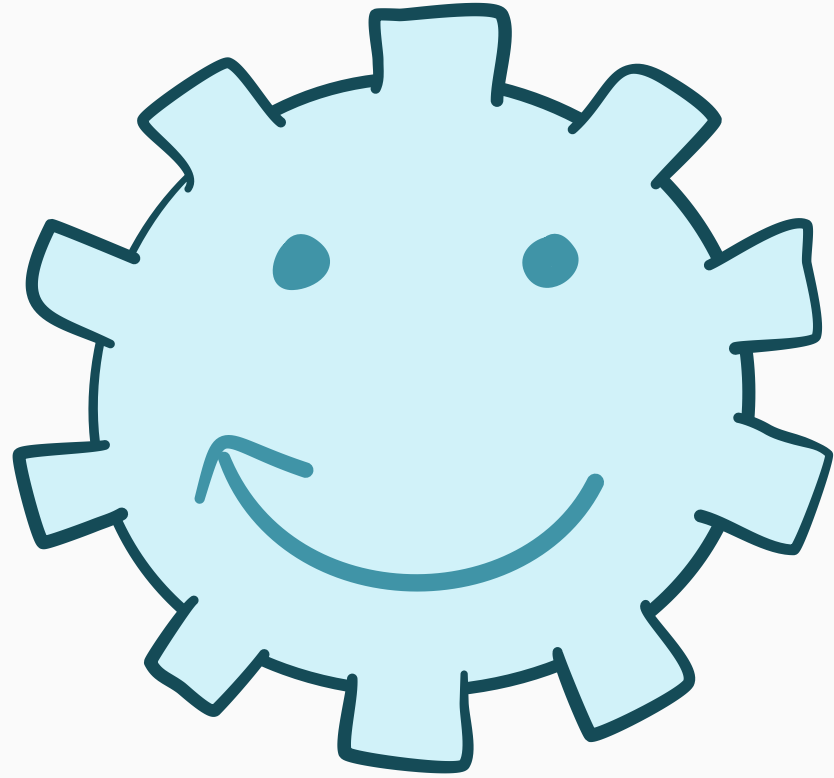
Theoretical problems



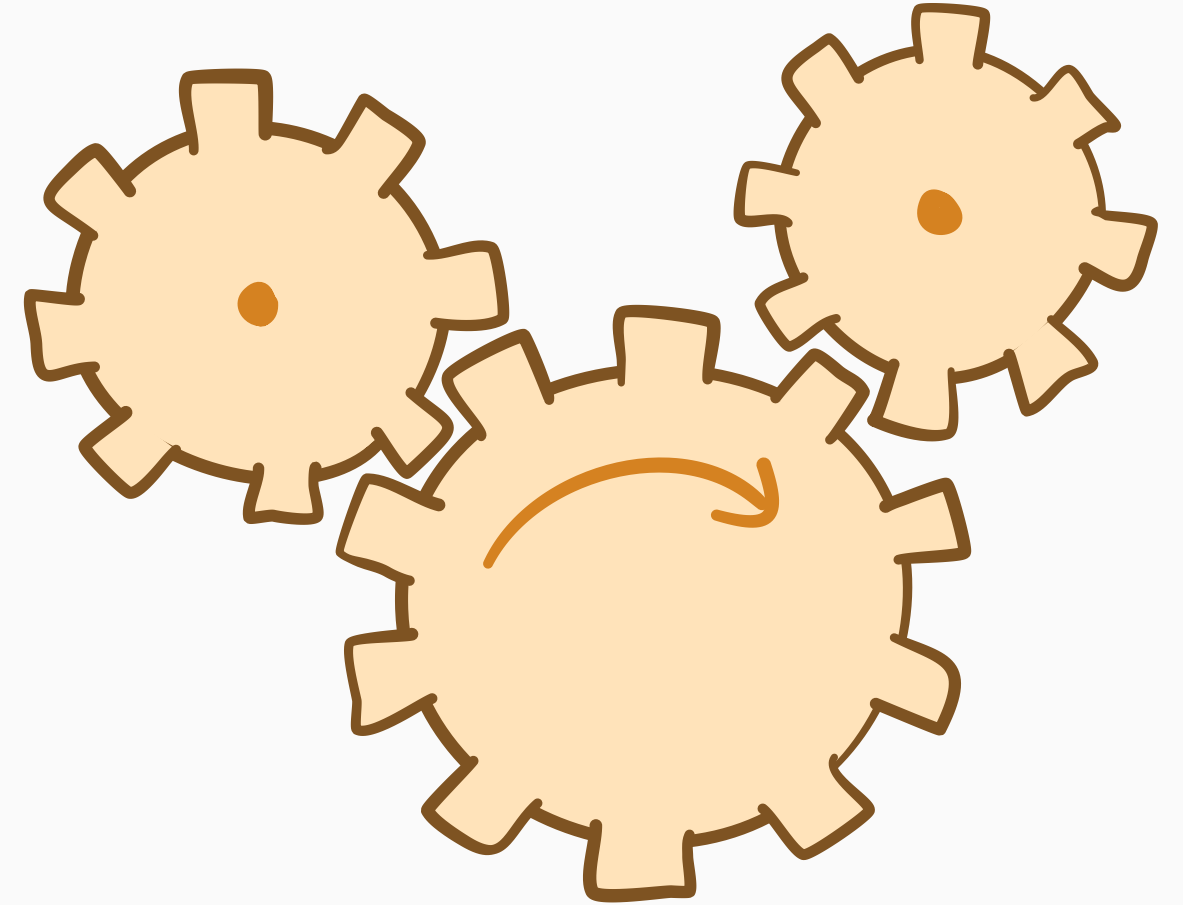
Practical problems



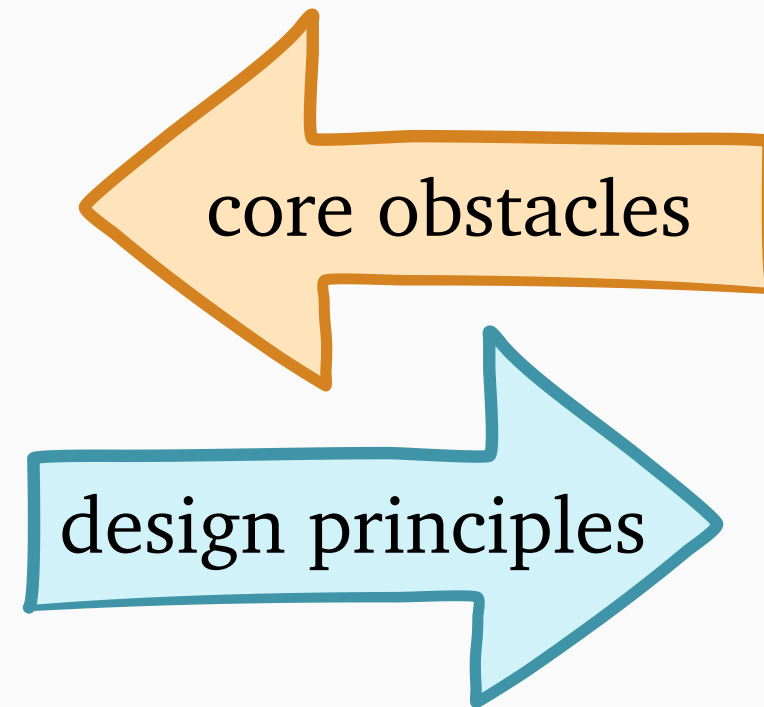
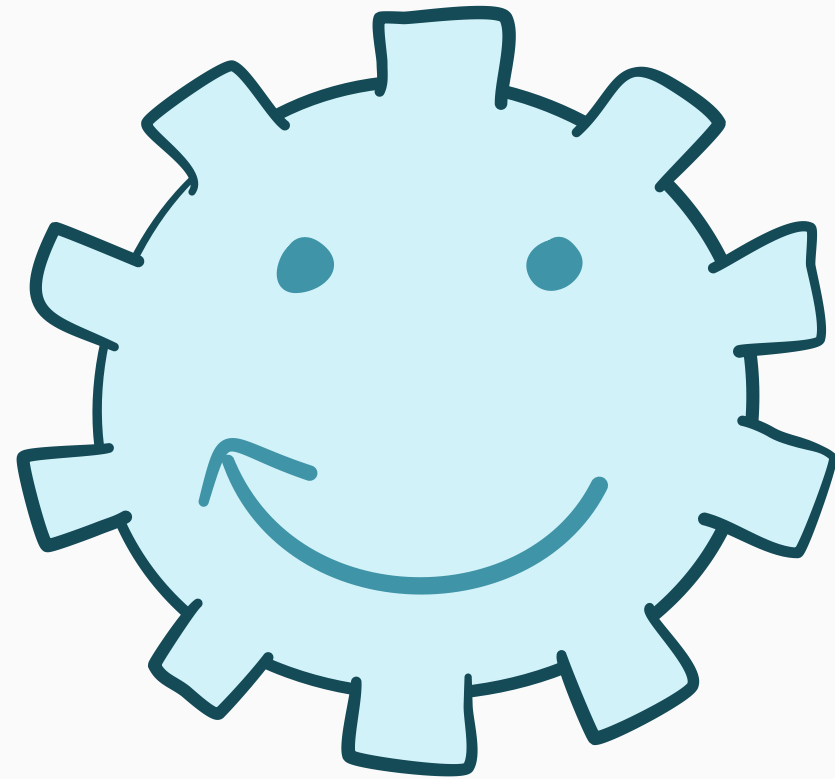
Theoretical problems



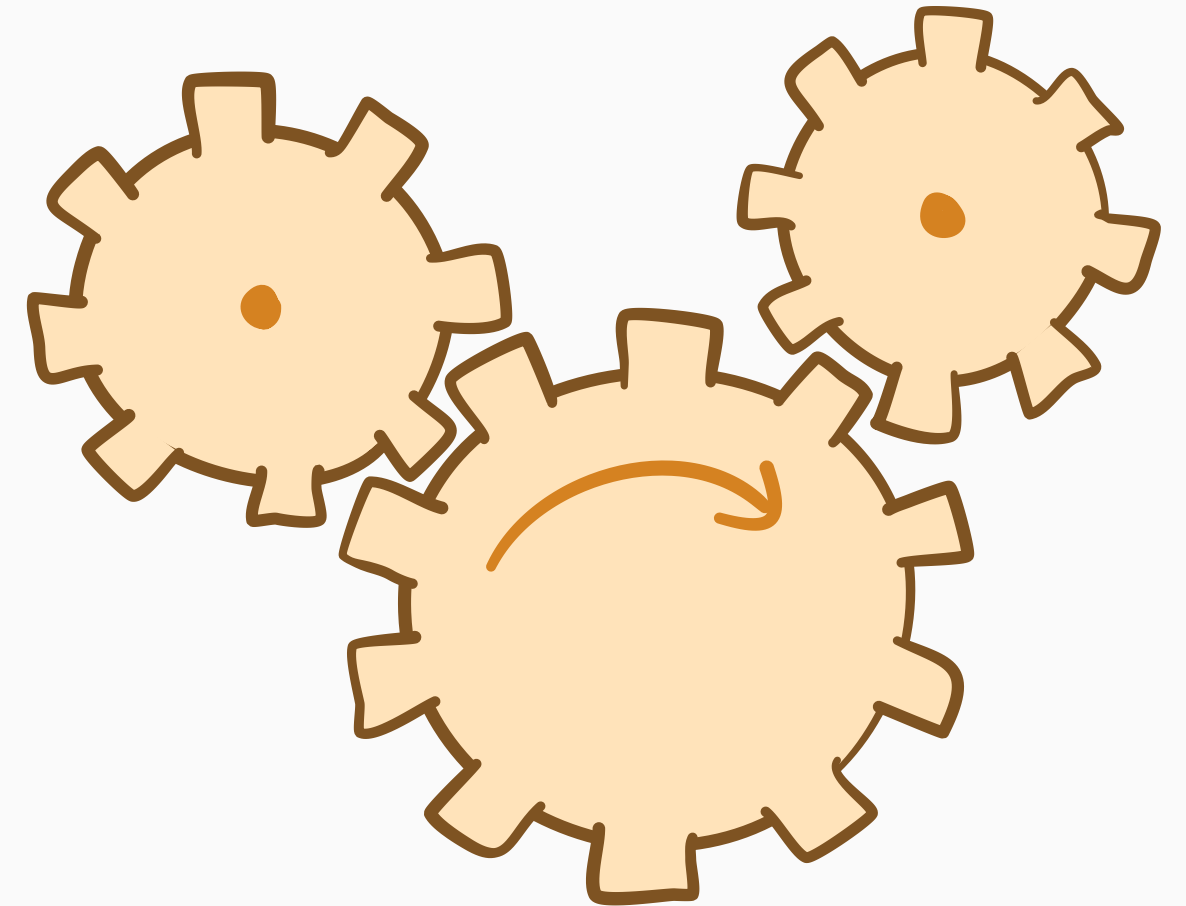
Practical problems



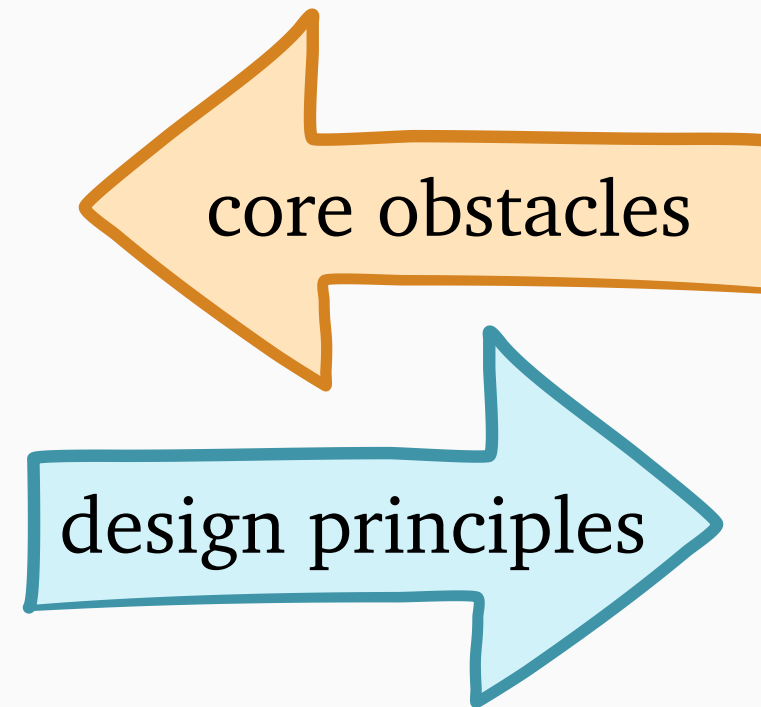
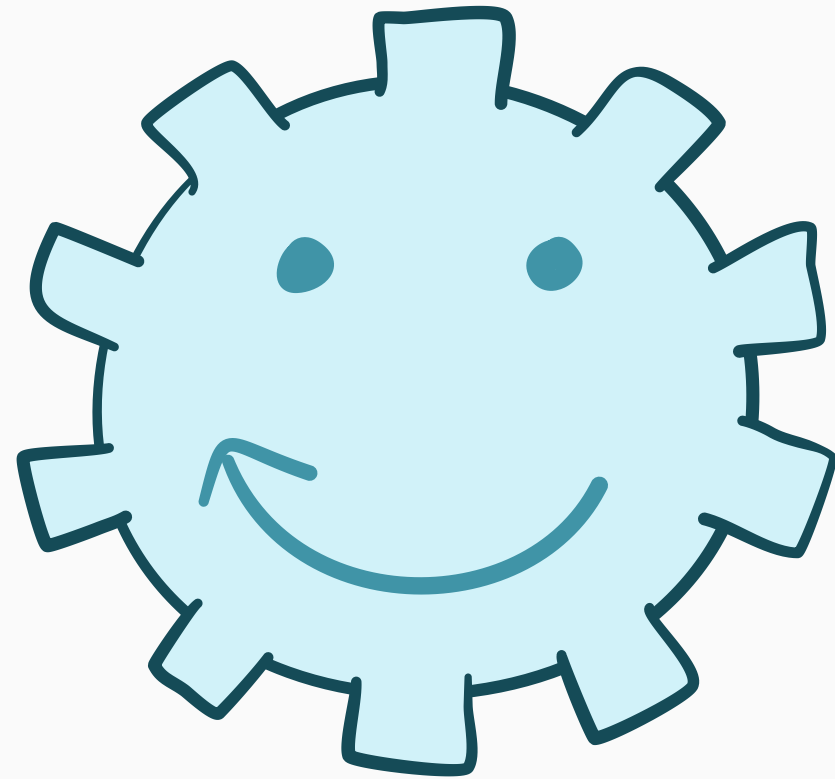
Theoretical problems



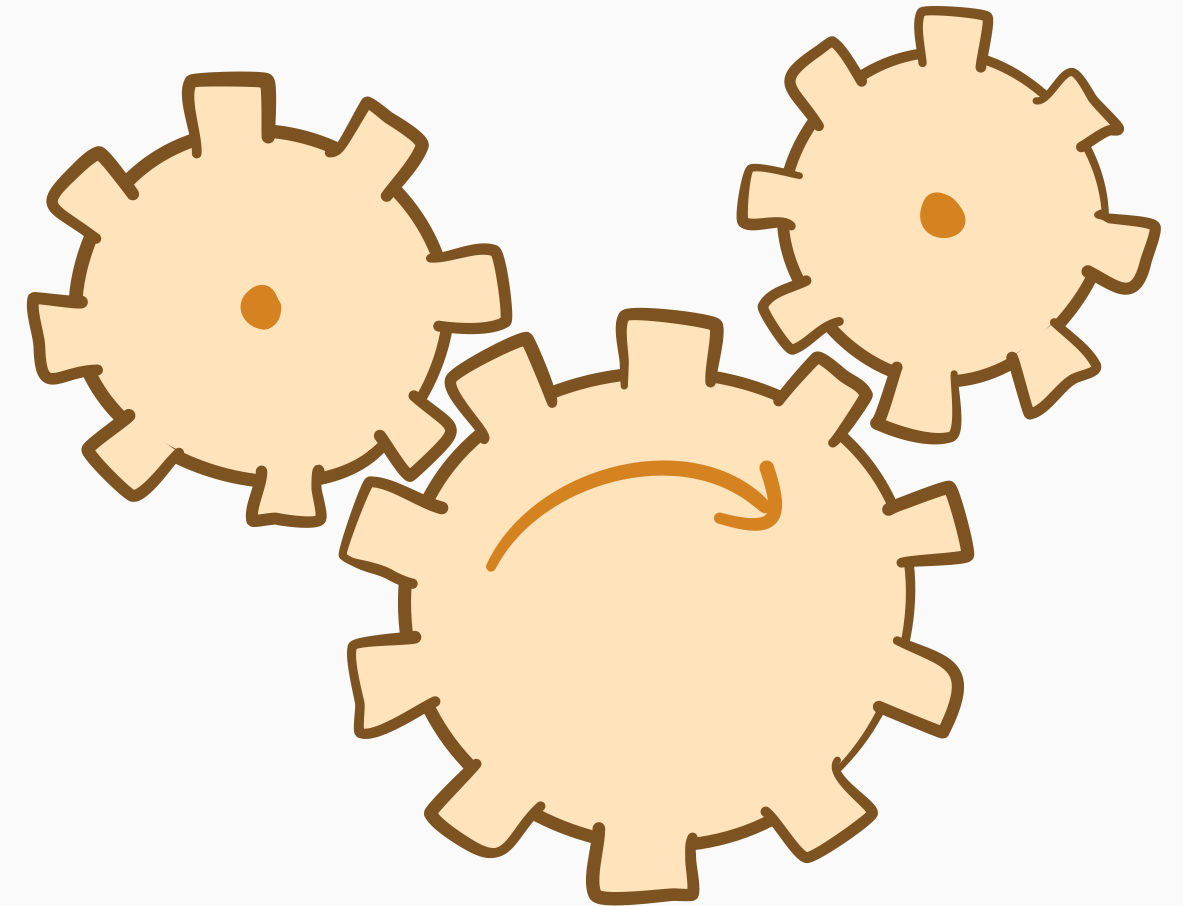
Practical problems



Theoretical problems

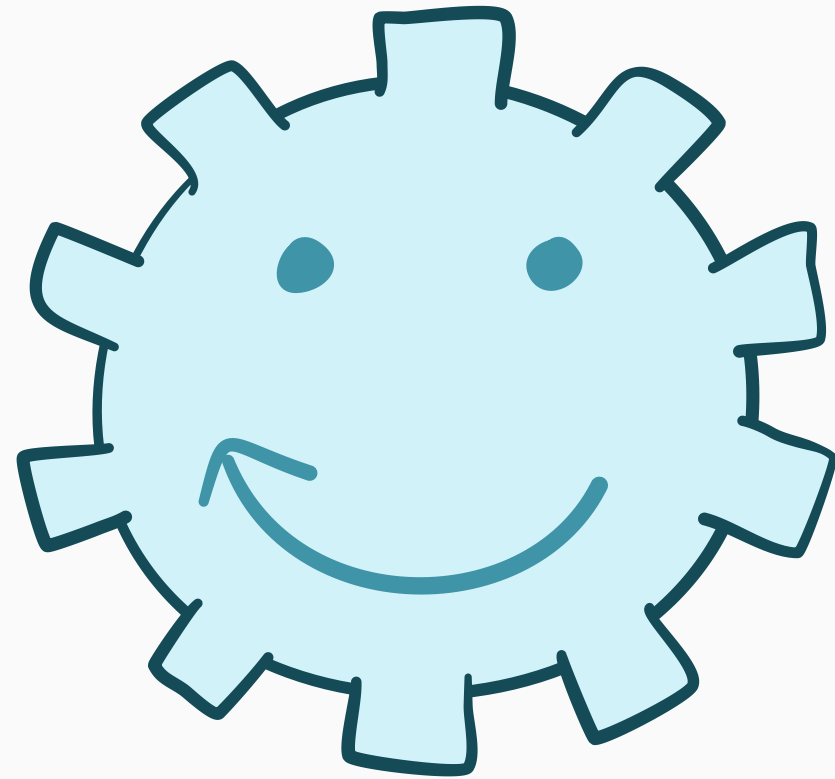


Practical problems

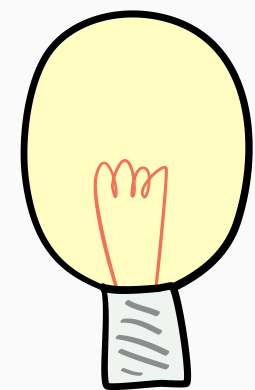
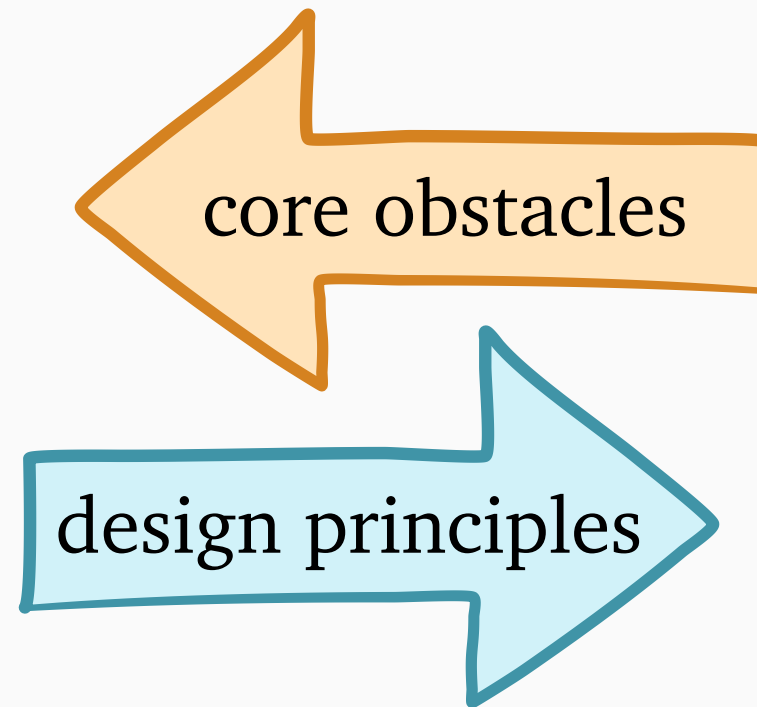
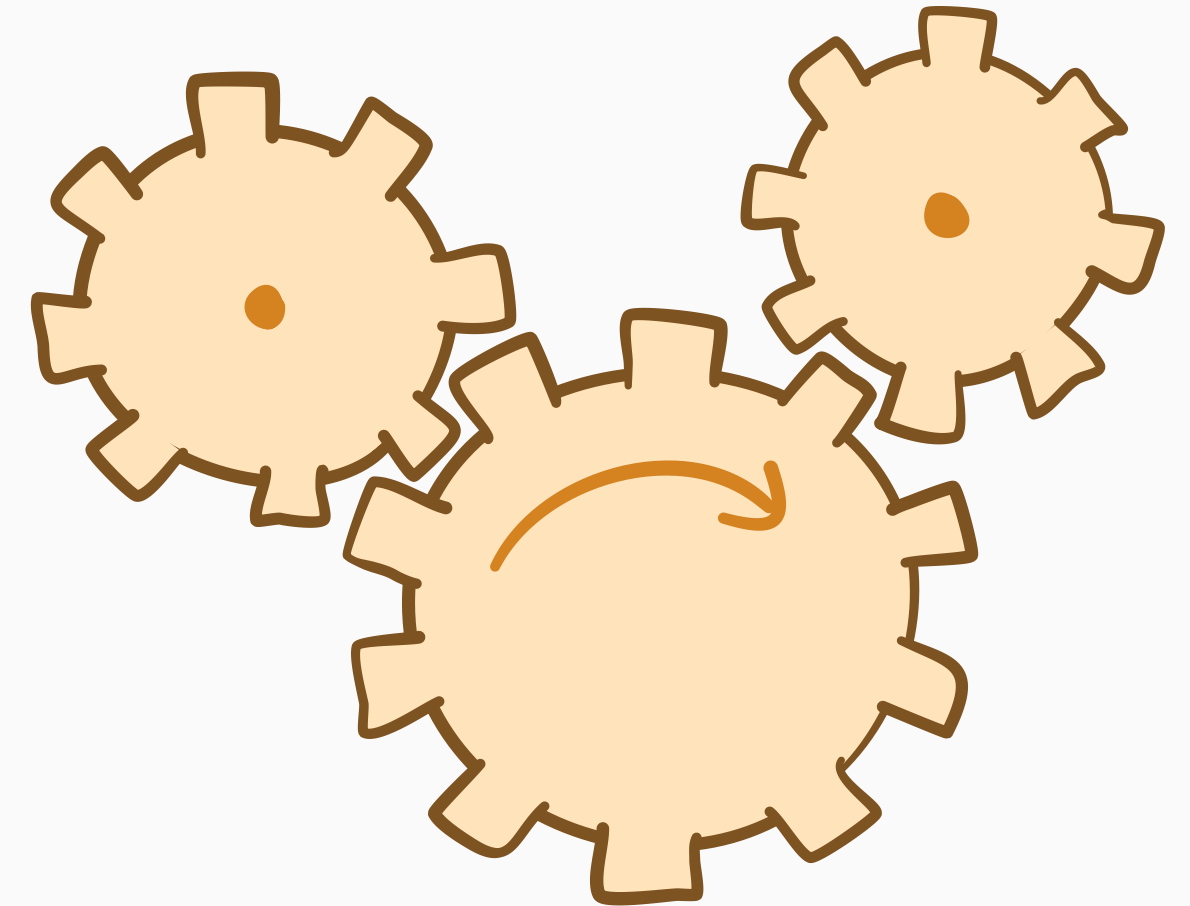


uncertainty

Theoretical problems



Practical problems

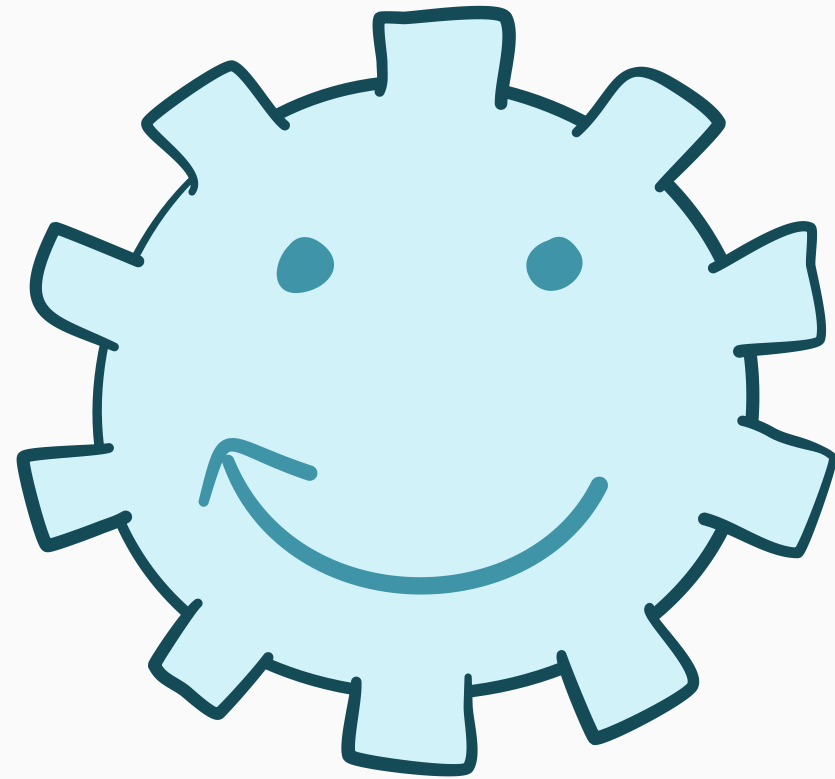


optimistic estimates
regularization
adaptive > nonadaptive
quantify information

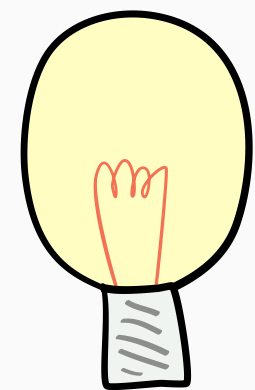
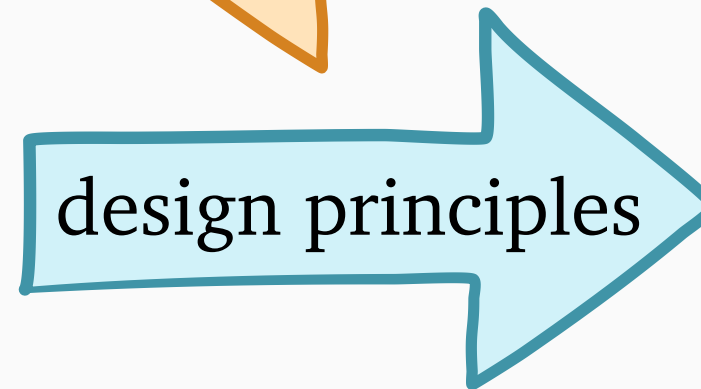
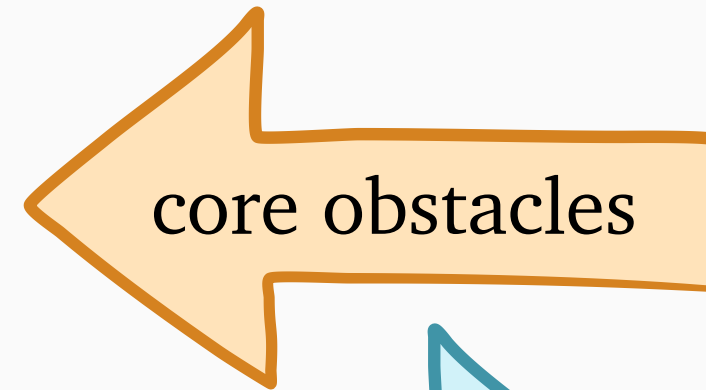
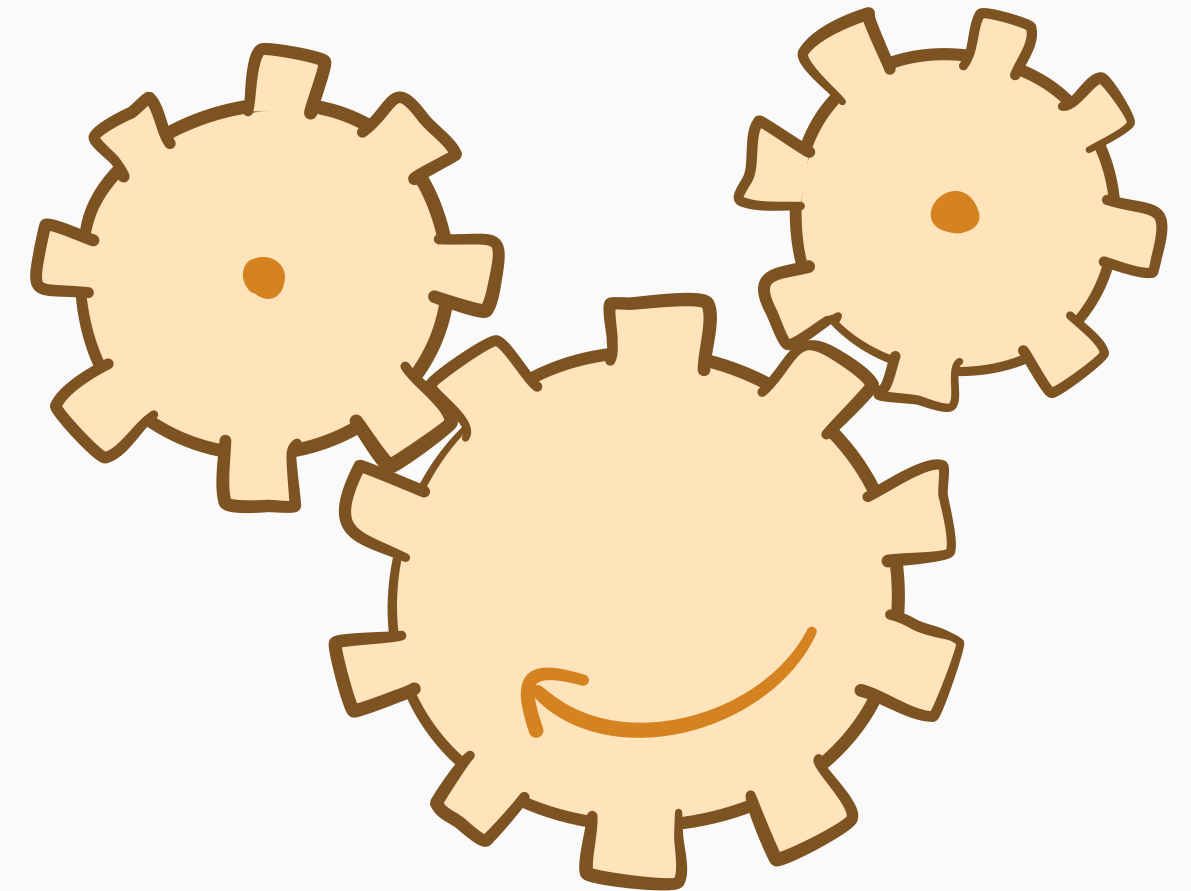


uncertainty

Theoretical problems



Practical problems

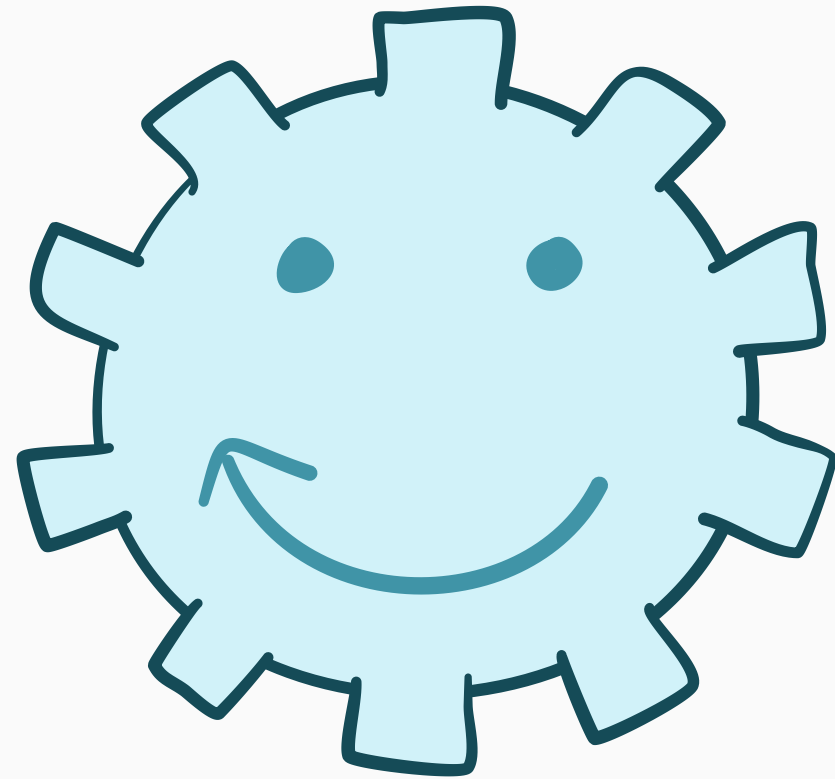


optimistic estimates
regularization
adaptive > nonadaptive
quantify information

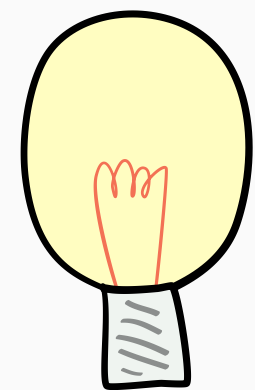
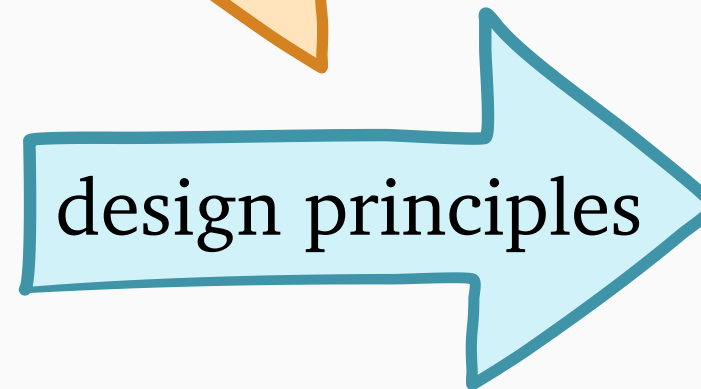
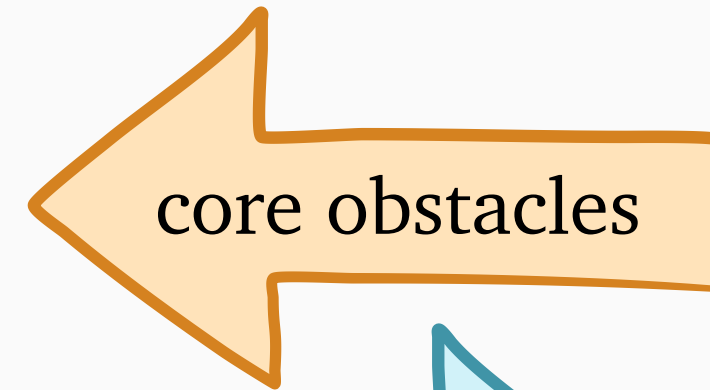
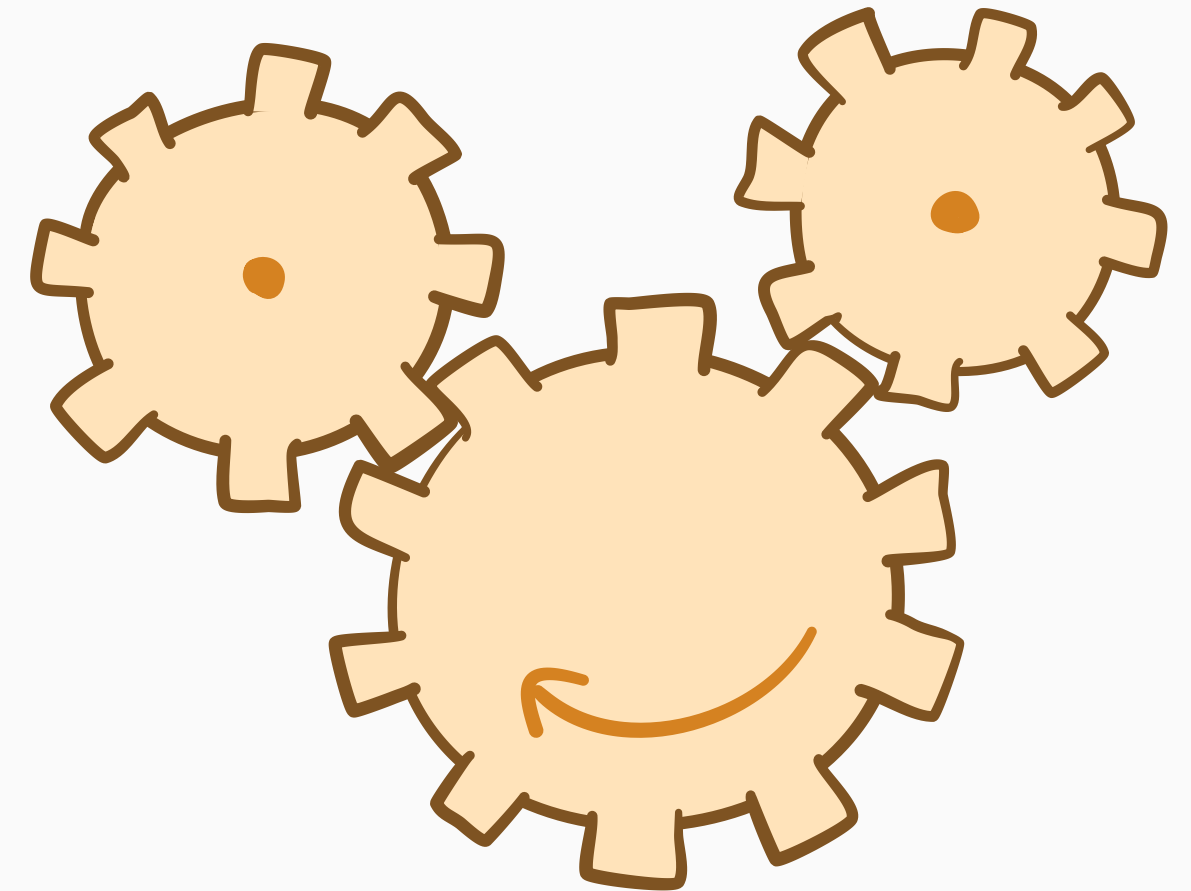


uncertainty

Theoretical problems



Practical problems

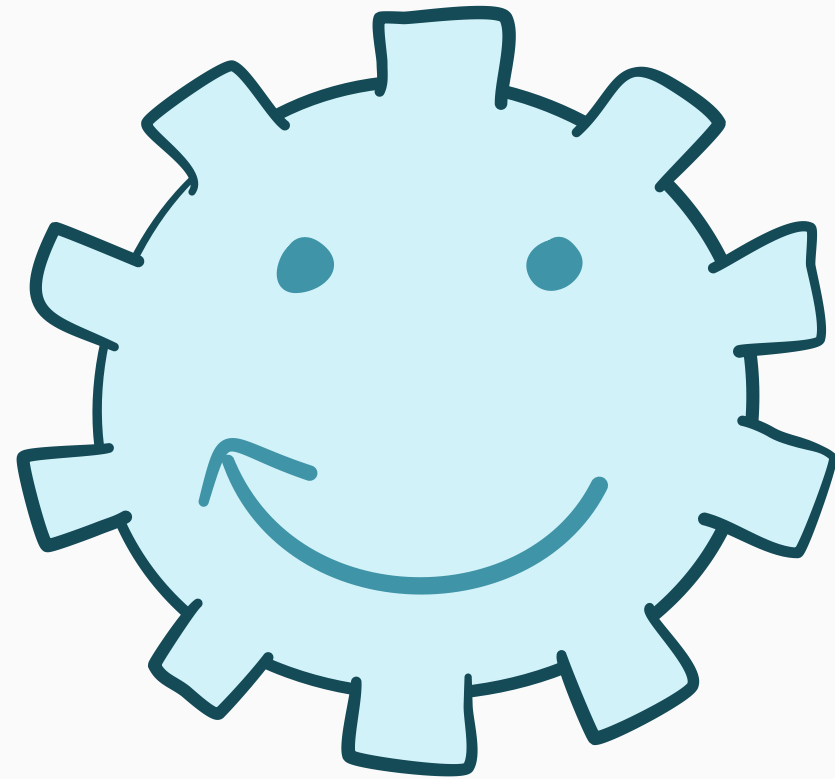


optimistic estimates
regularization
adaptive > nonadaptive
quantify information

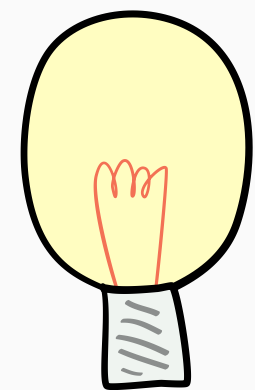
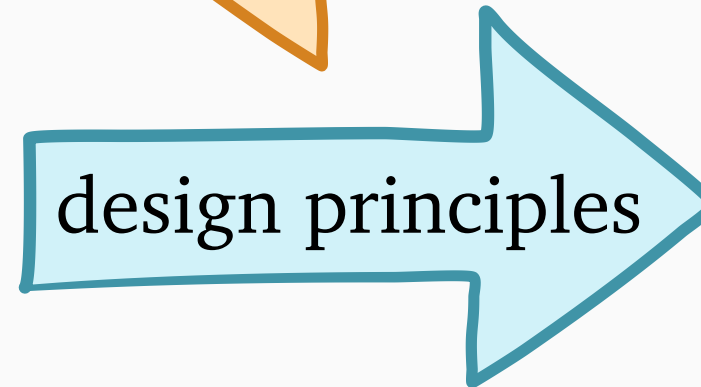
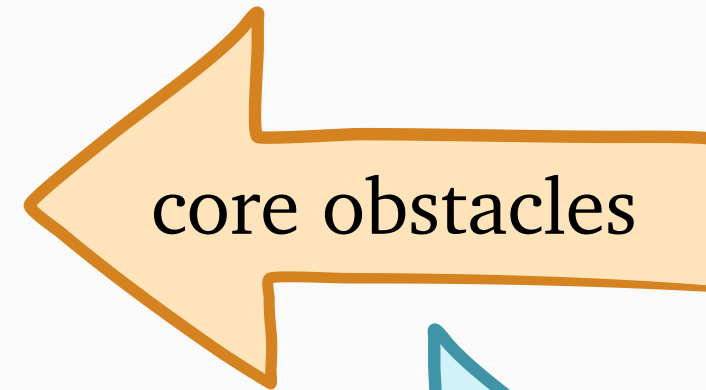
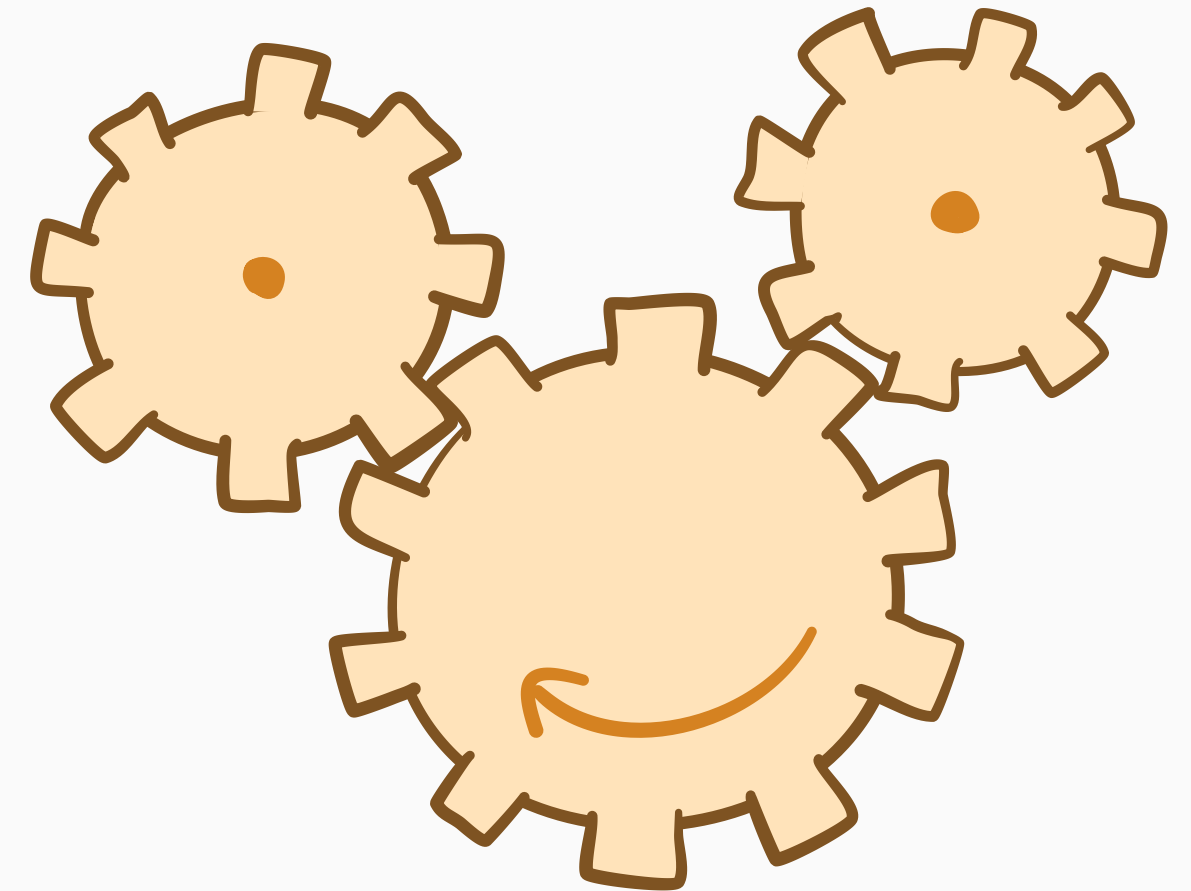


uncertainty with...

Theoretical problems



Practical problems

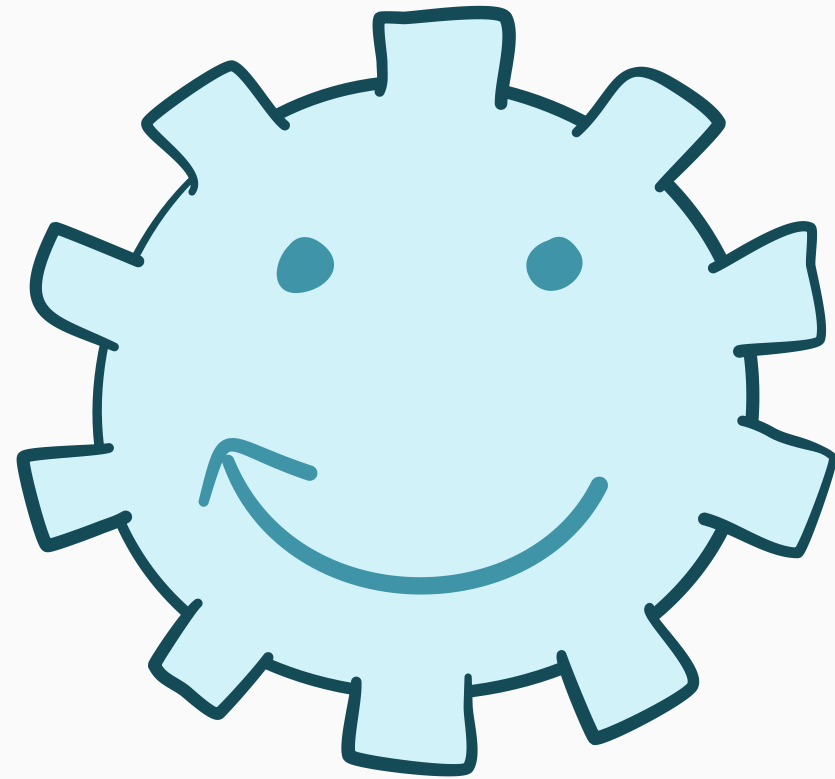


optimistic estimates
regularization
adaptive > nonadaptive
quantify information

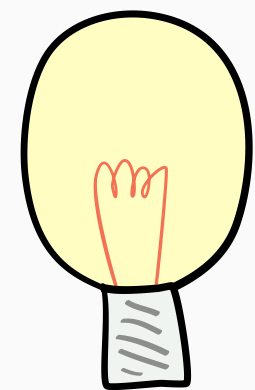
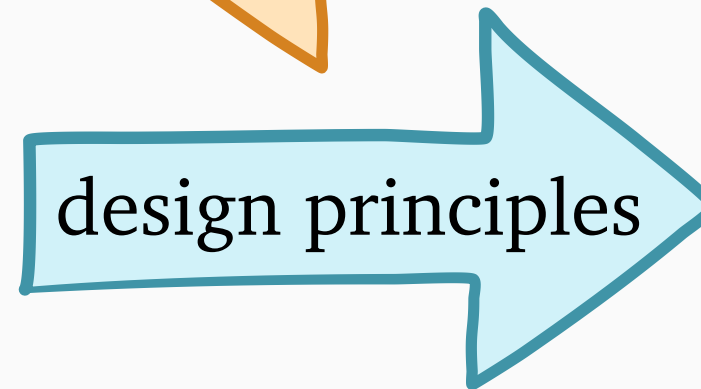
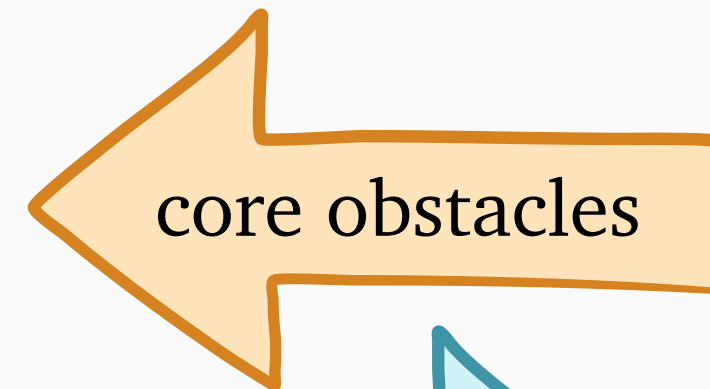
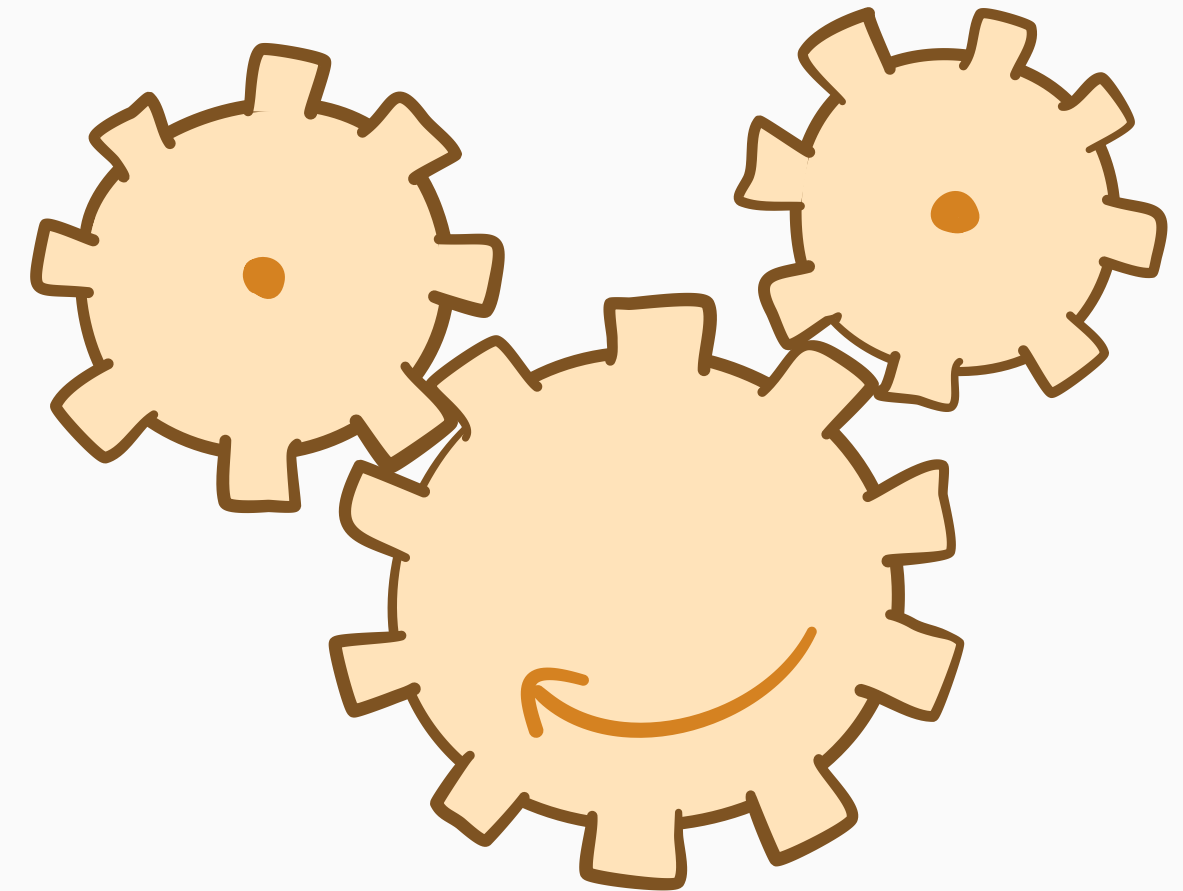


uncertainty with...
costly information

Theoretical problems



Practical problems

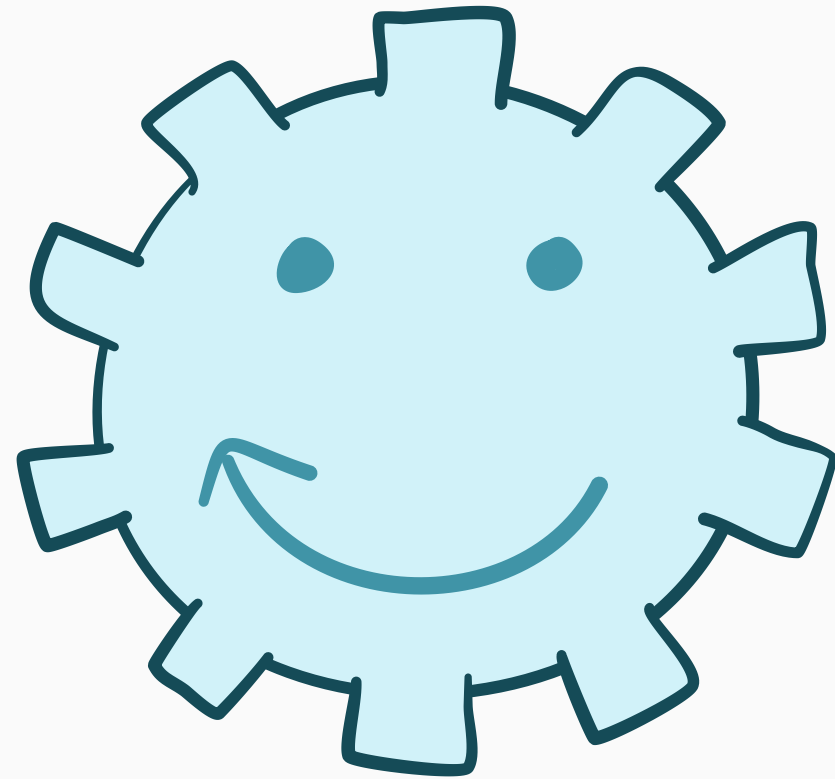


optimistic estimates
regularization
adaptive > nonadaptive
quantify information

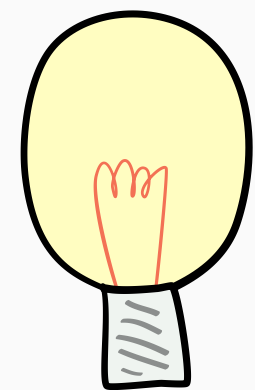
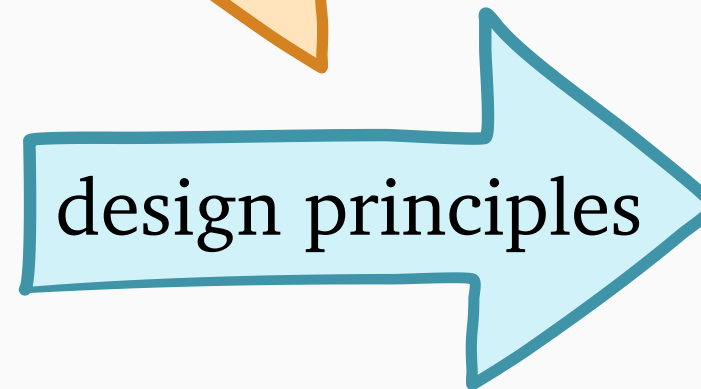
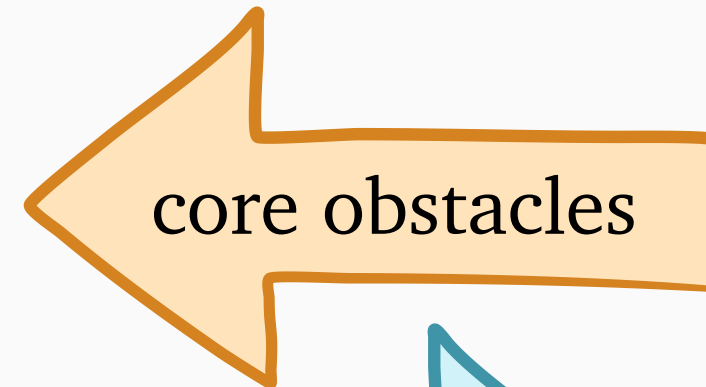
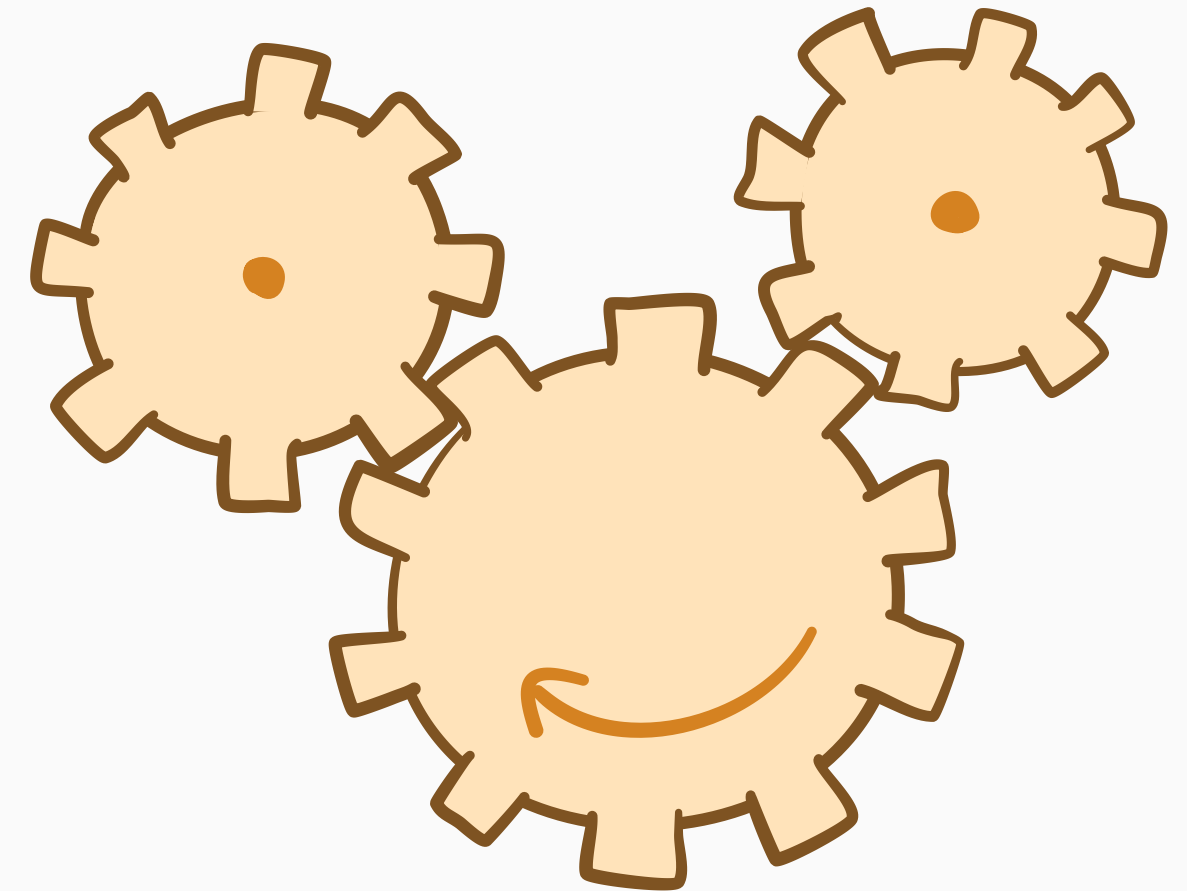


uncertainty with...
costly information
multi-stage uncertainty

Theoretical problems



Practical problems

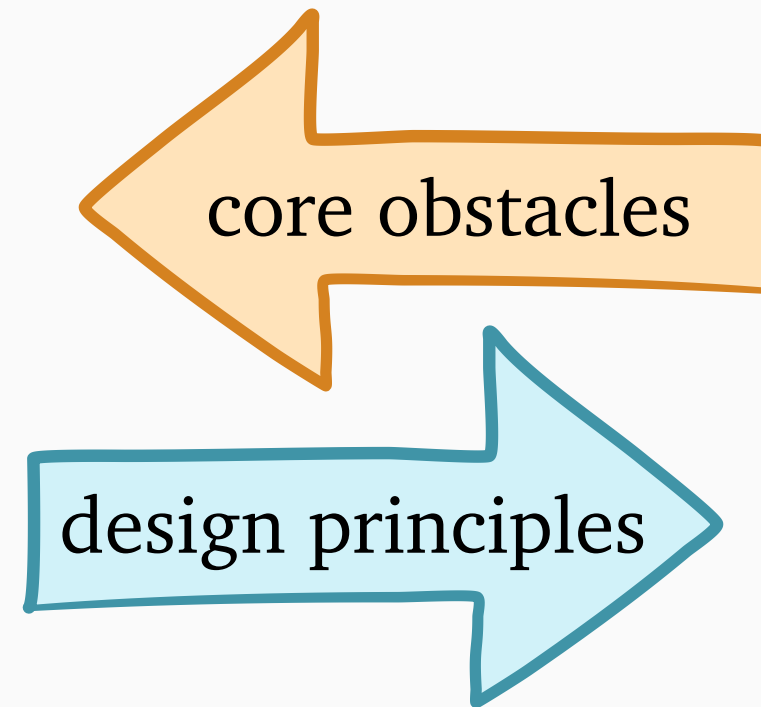
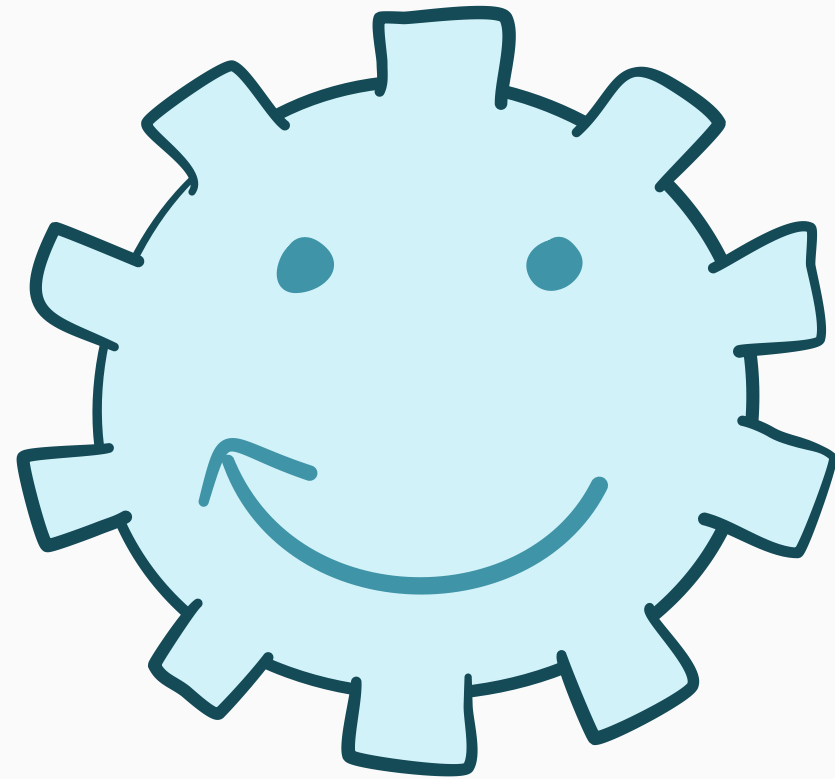


optimistic estimates
regularization
adaptive > nonadaptive
quantify information

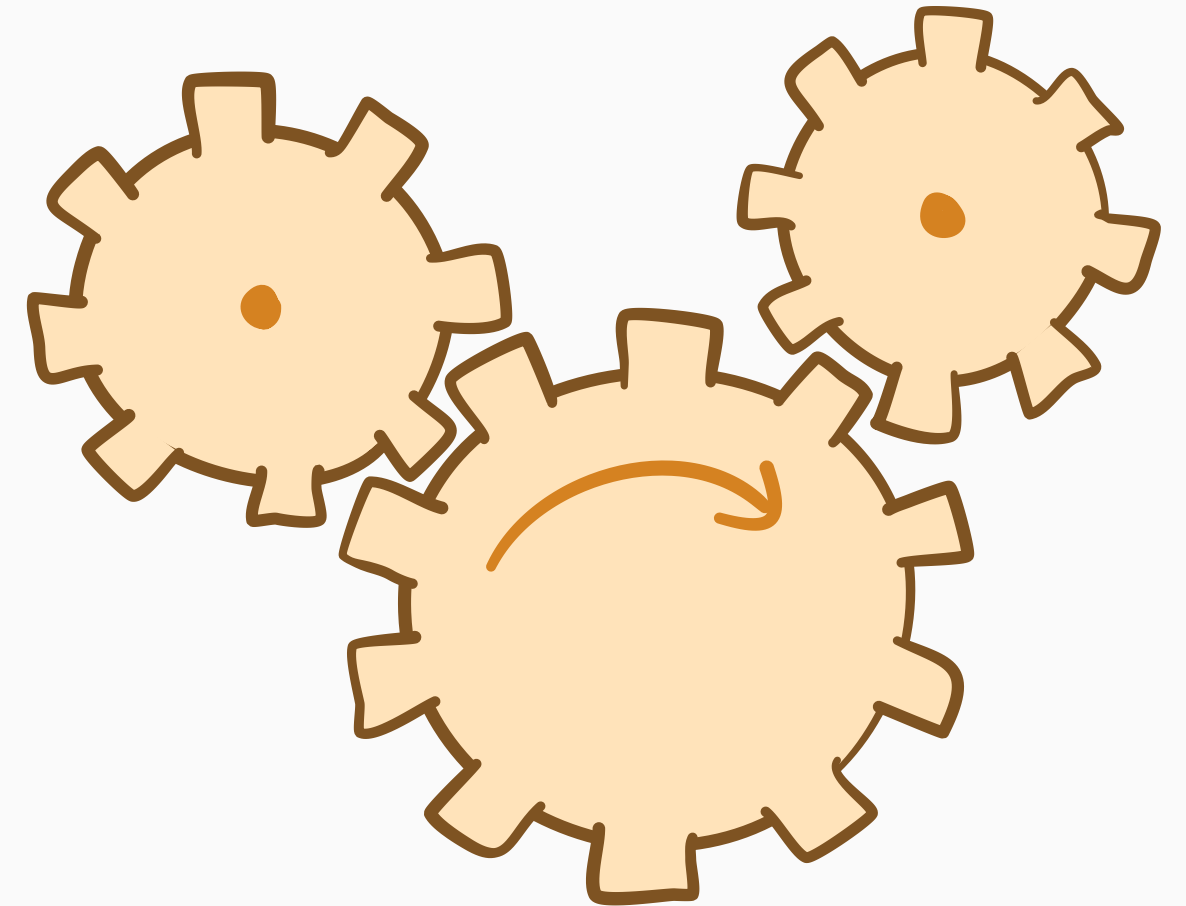


uncertainty with...
costly information
multi-stage uncertainty
heterogeneity

Theoretical problems

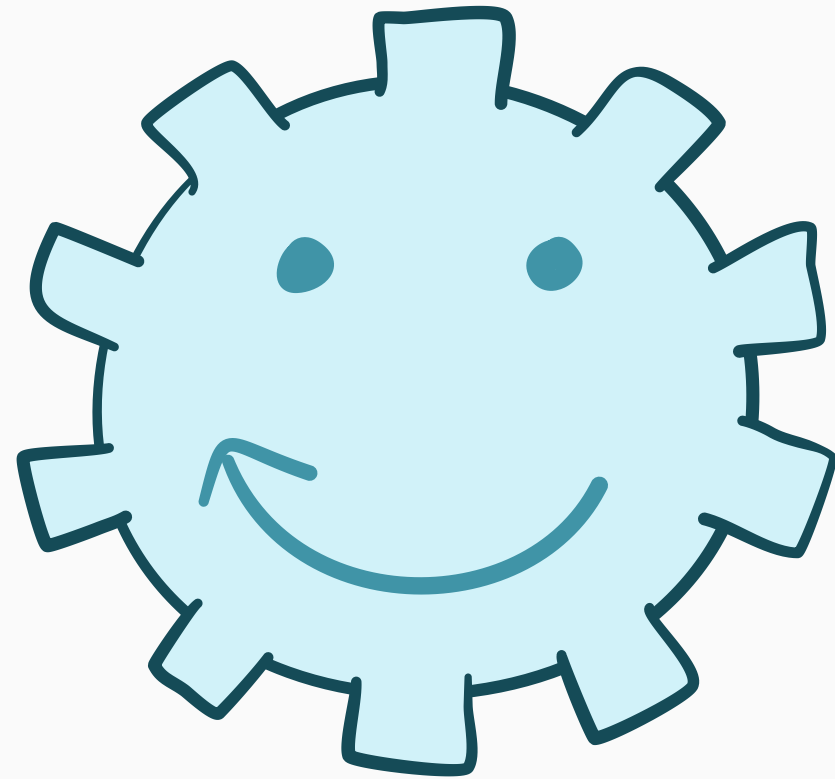


Practical problems

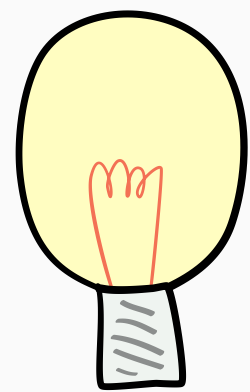
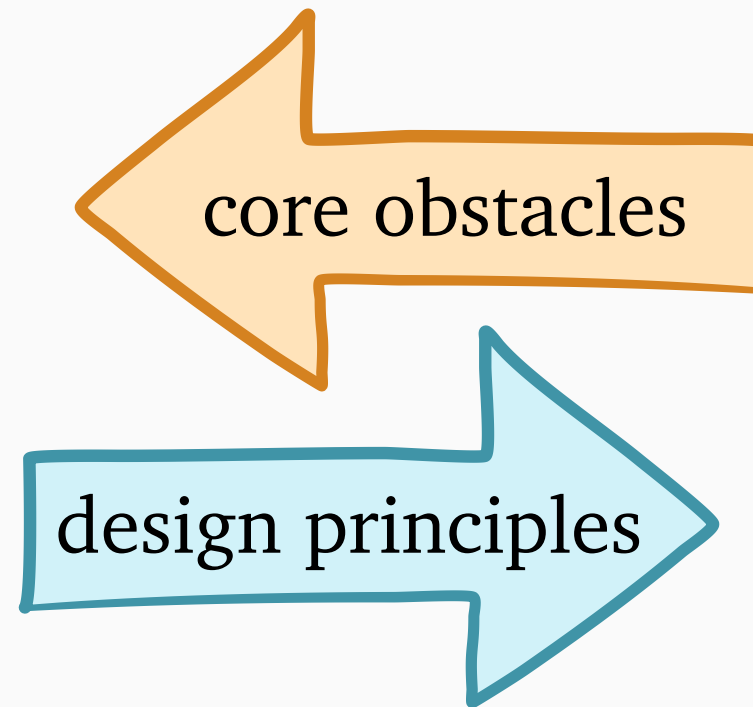
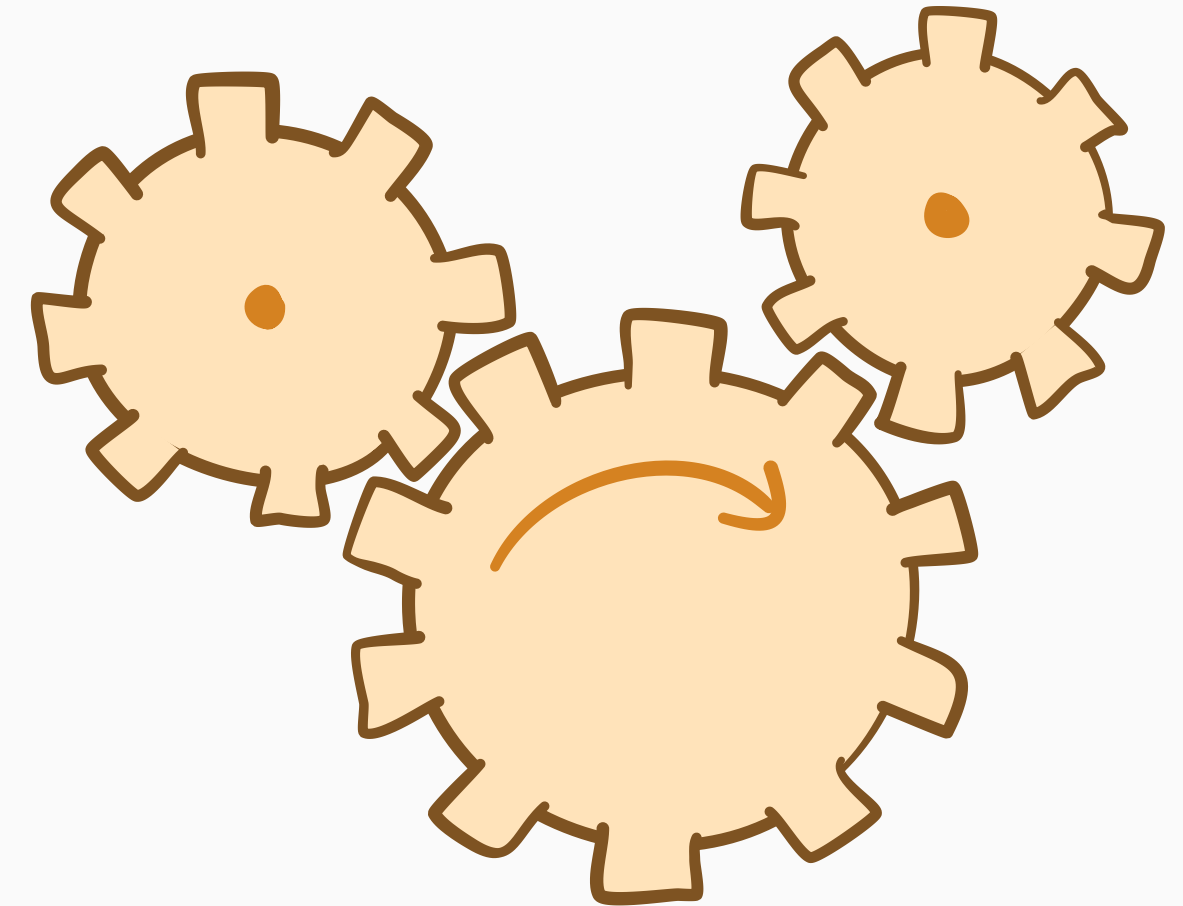


uncertainty with...
costly information
multi-stage uncertainty
heterogeneity

Theoretical problems



Practical problems

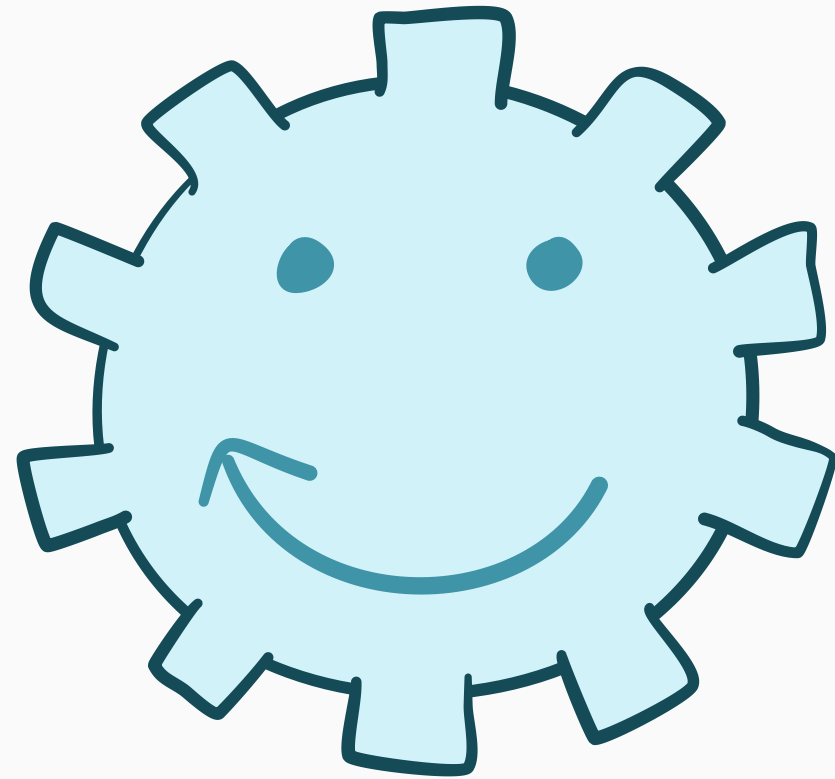


Gittins design principle:
evaluate stochastic options by
calibrating to deterministic options

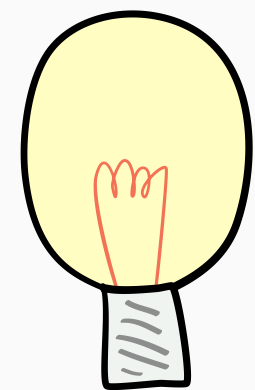
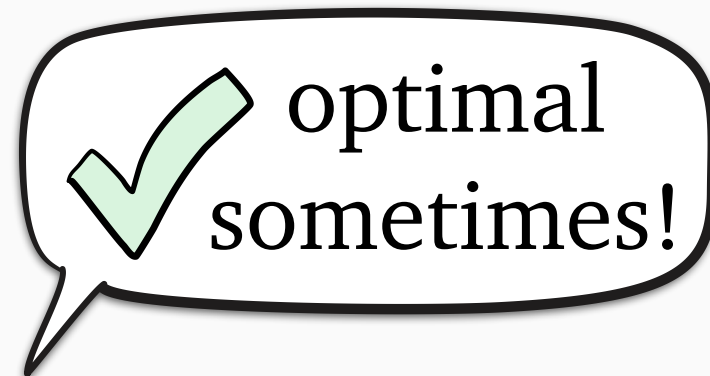
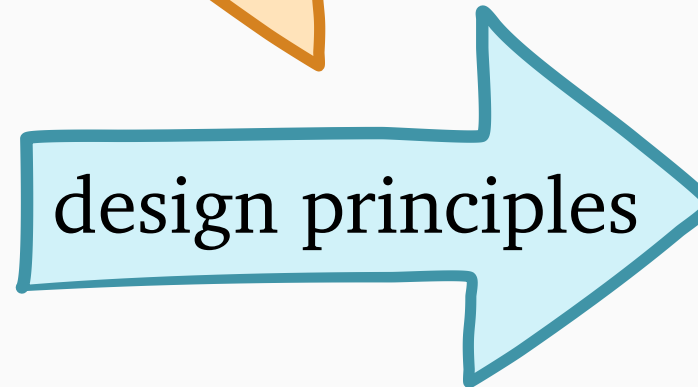
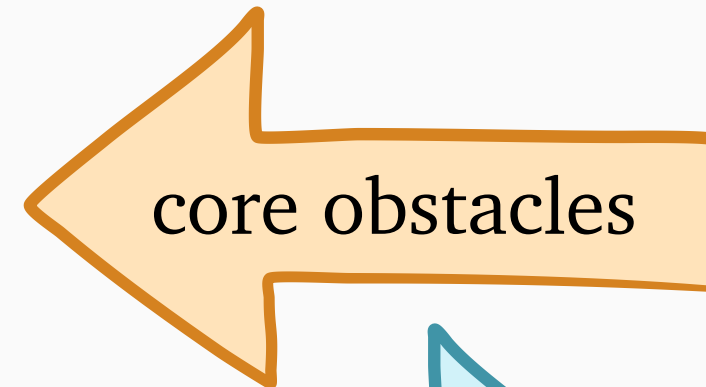
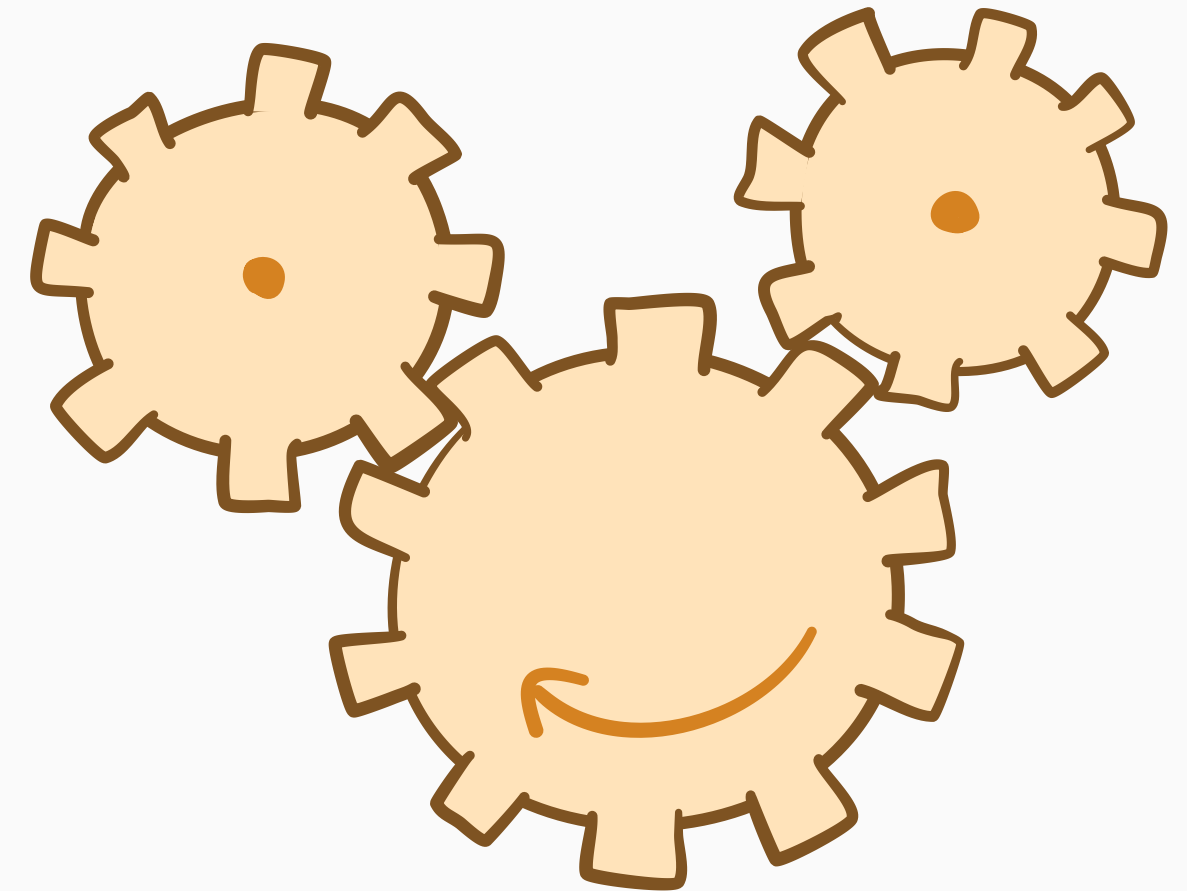


uncertainty with...
costly information
multi-stage uncertainty
heterogeneity

Theoretical problems



Practical problems

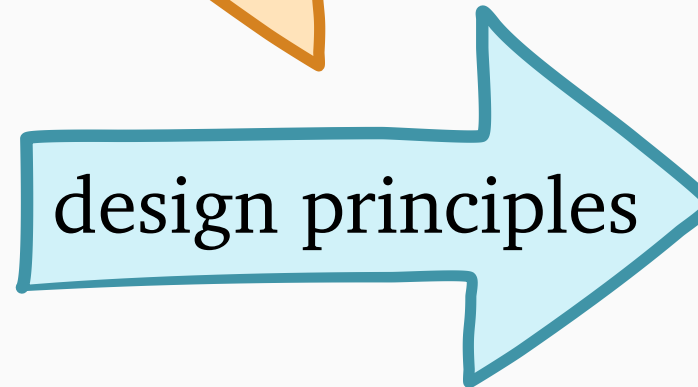
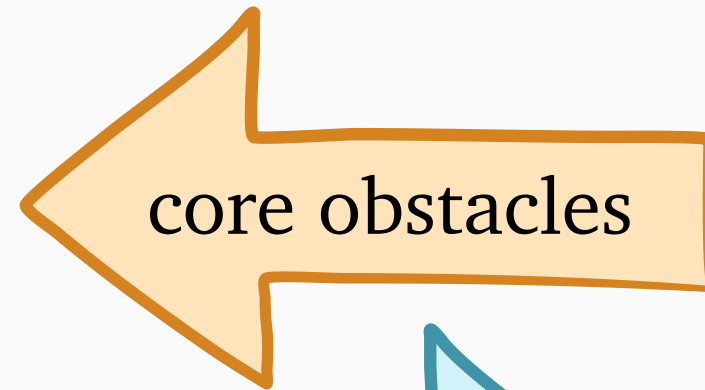
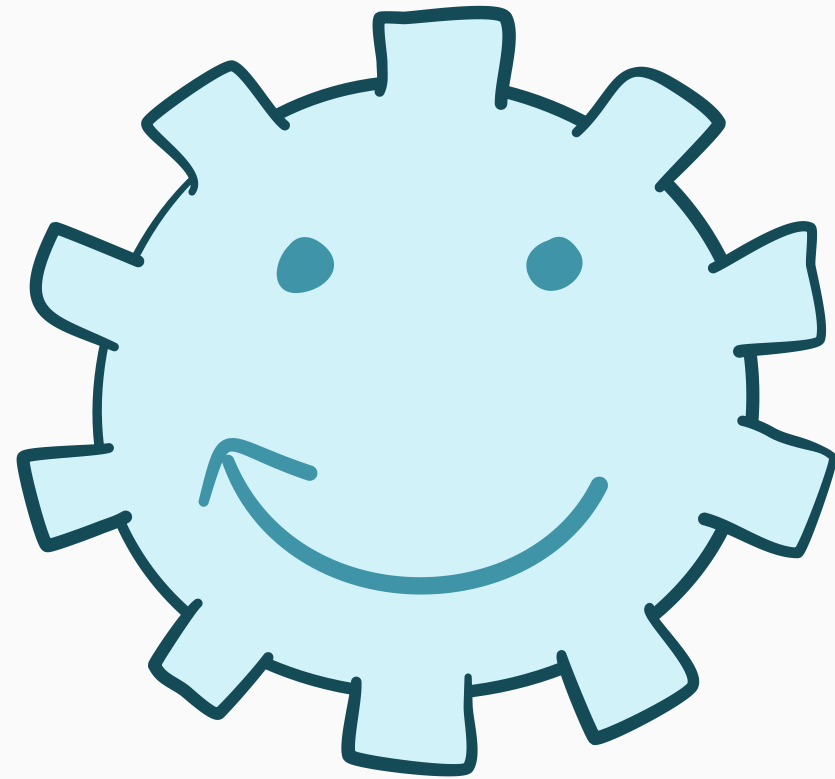


Gittins design principle:
evaluate stochastic options by
calibrating to deterministic options

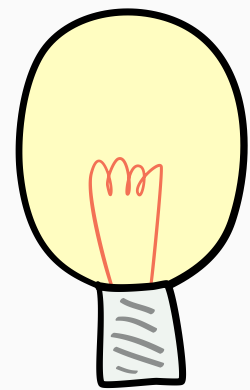
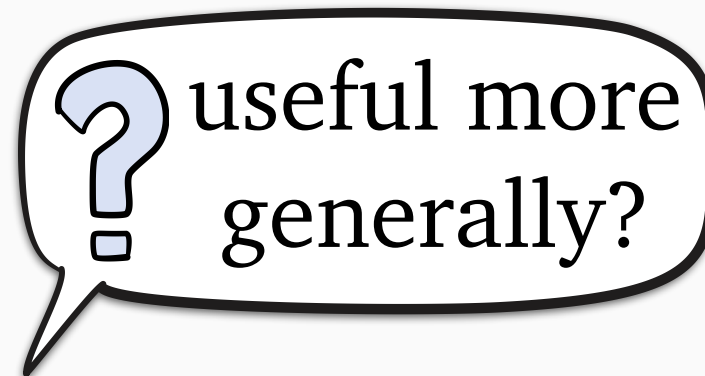
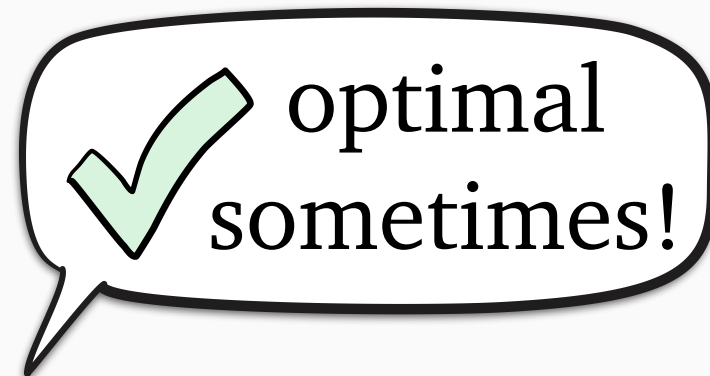
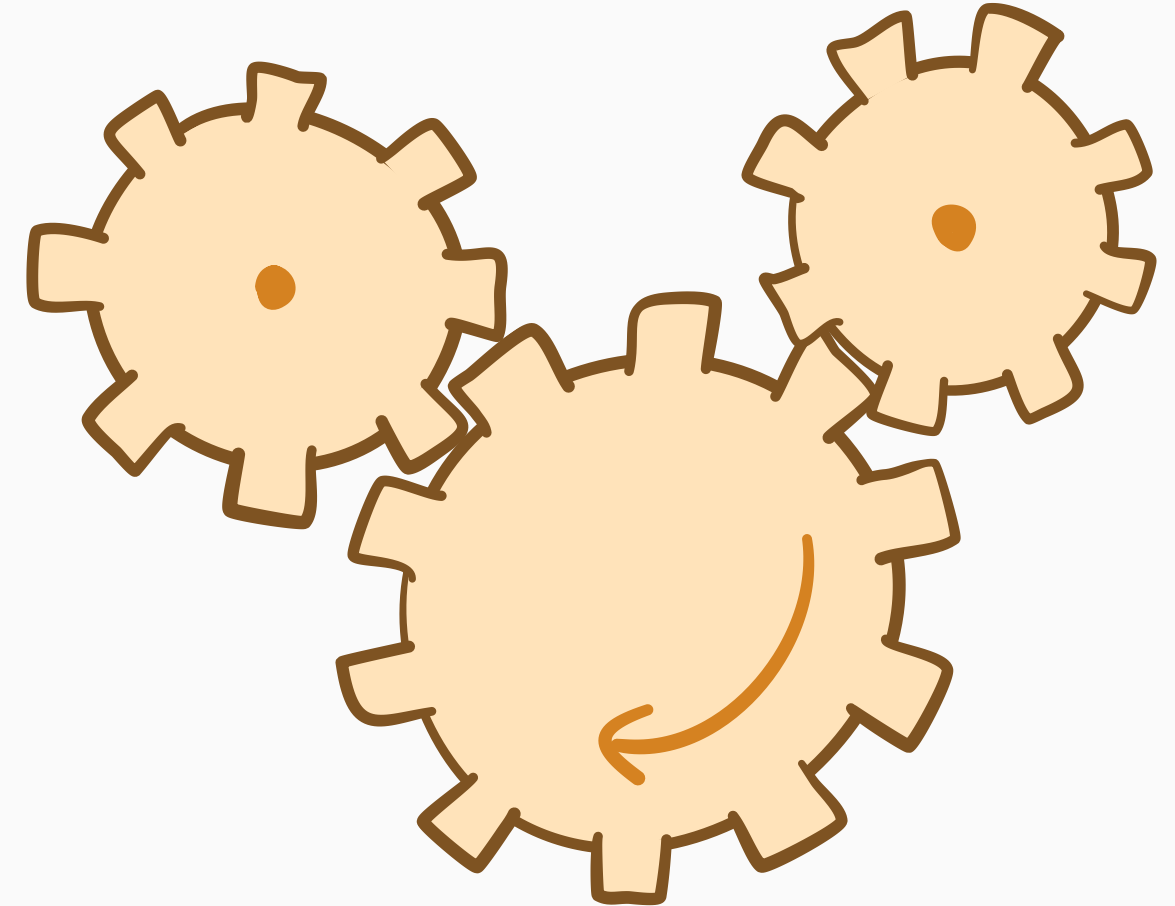


uncertainty with...
costly information
multi-stage uncertainty
heterogeneity

Theoretical problems



Practical problems



Gittins design principle:
evaluate stochastic options by
calibrating to deterministic options

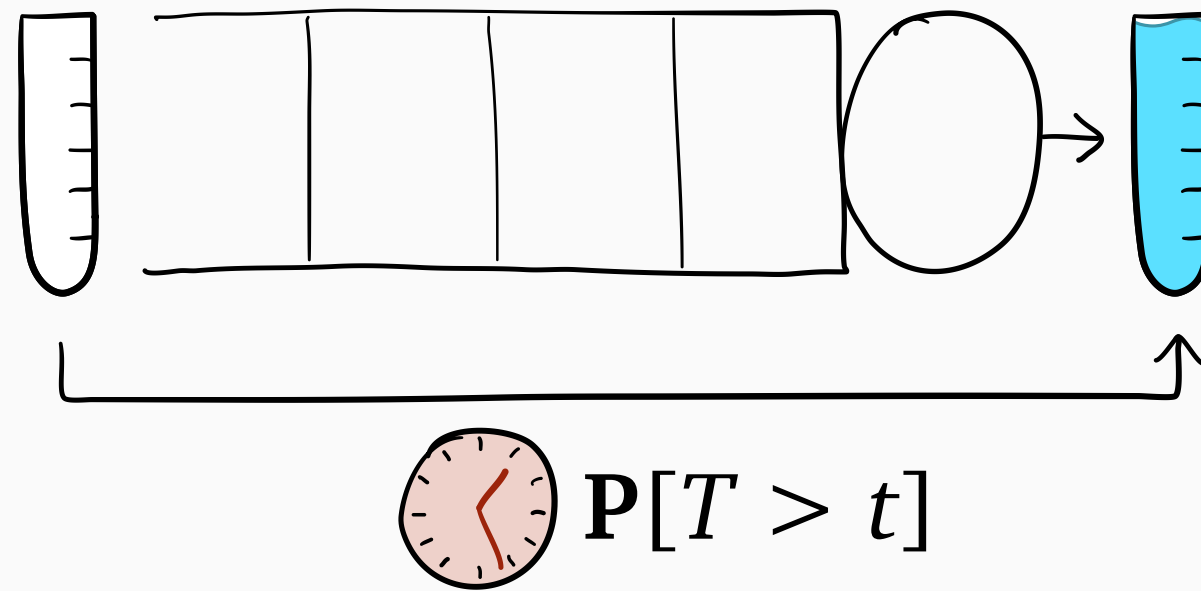


uncertainty with...
costly information
multi-stage uncertainty
heterogeneity

Two new **Gittins** applications

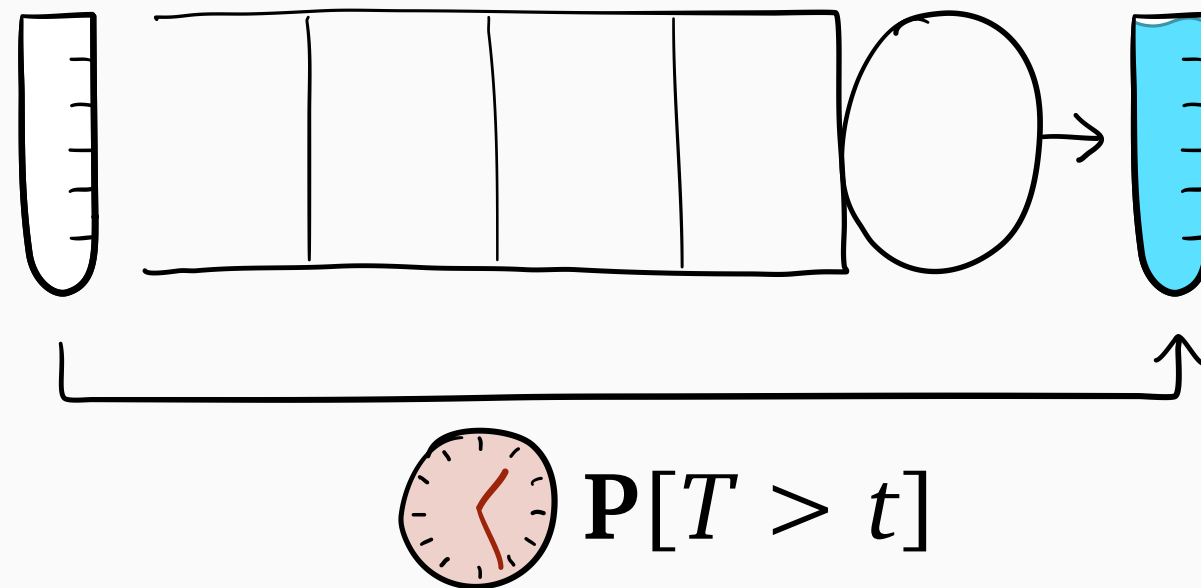
Two new Gittins applications

Tail scheduling



Two new Gittins applications

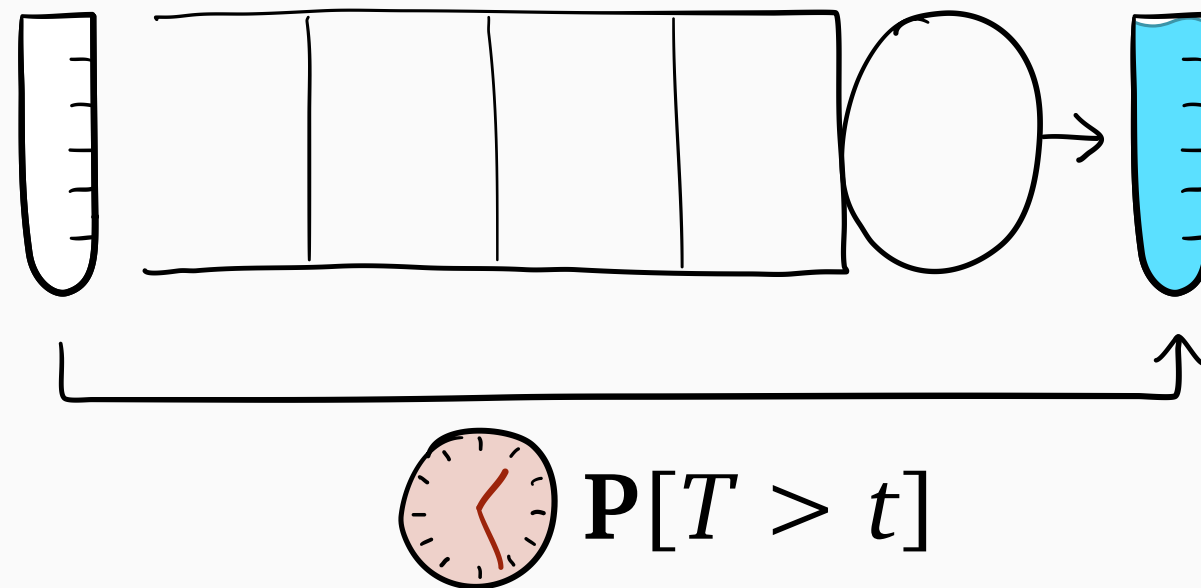
Tail scheduling



Goal: minimize probability of very long response time

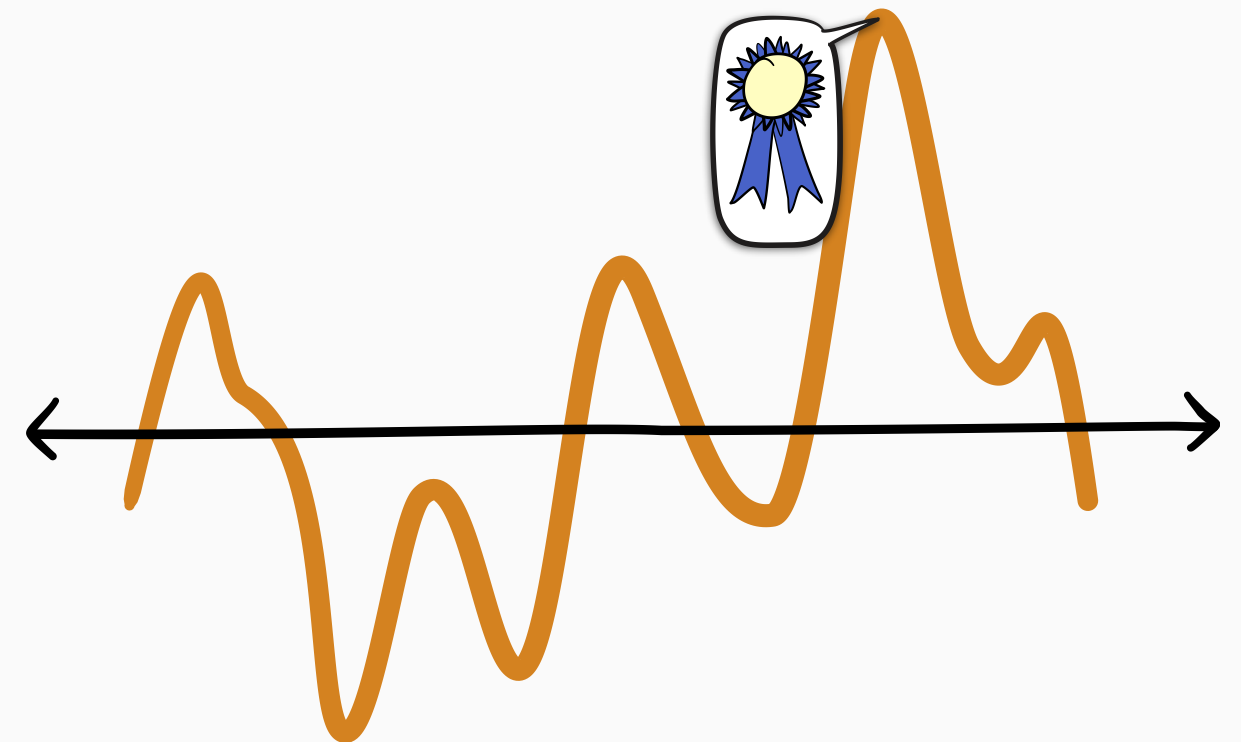
Two new Gittins applications

Tail scheduling



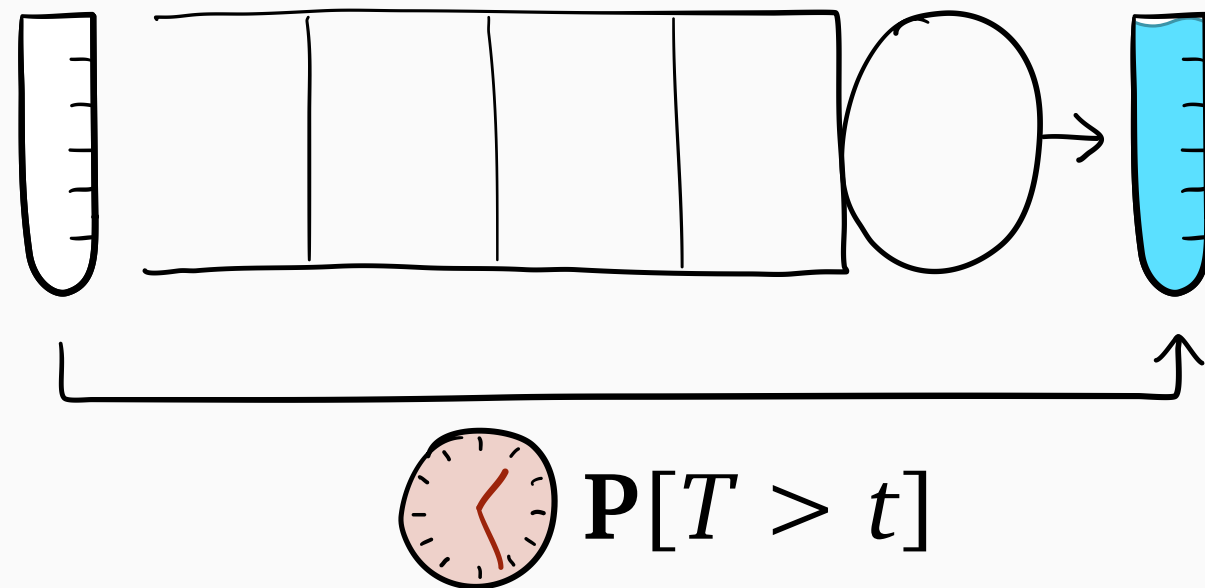
Goal: minimize probability of very long response time

Bayesian optimization



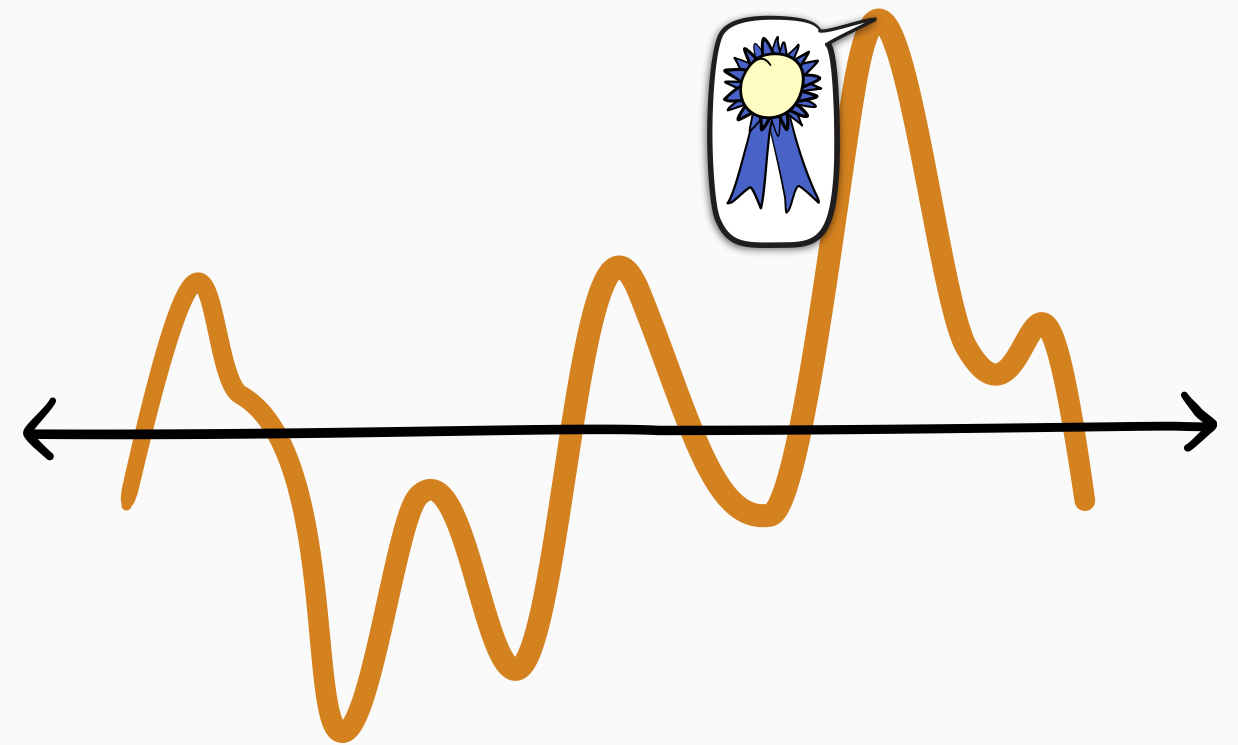
Two new Gittins applications

Tail scheduling



Goal: minimize probability of very long response time

Bayesian optimization

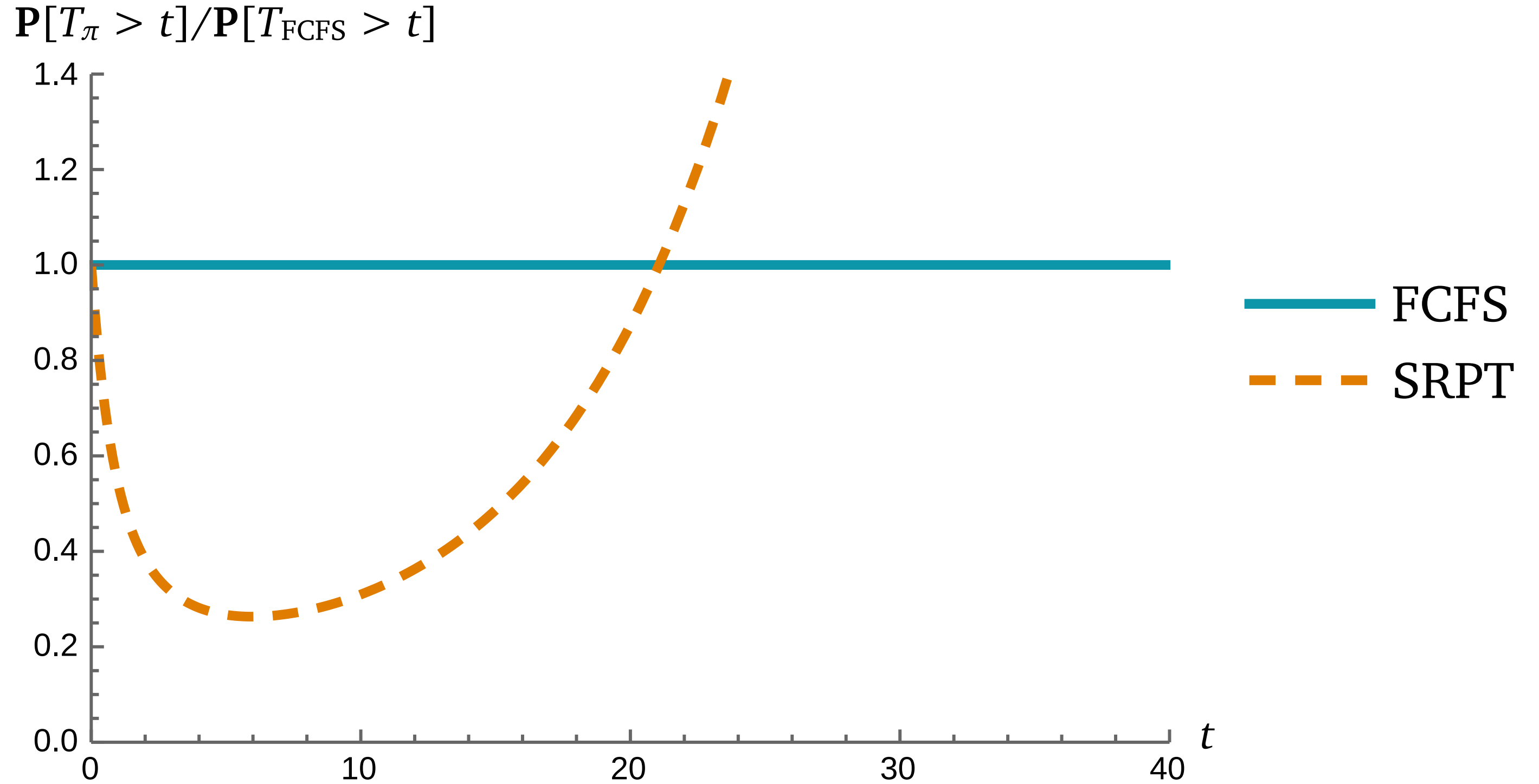


Goal: find large function value with few function evaluations

Empirical results: **tail scheduling**



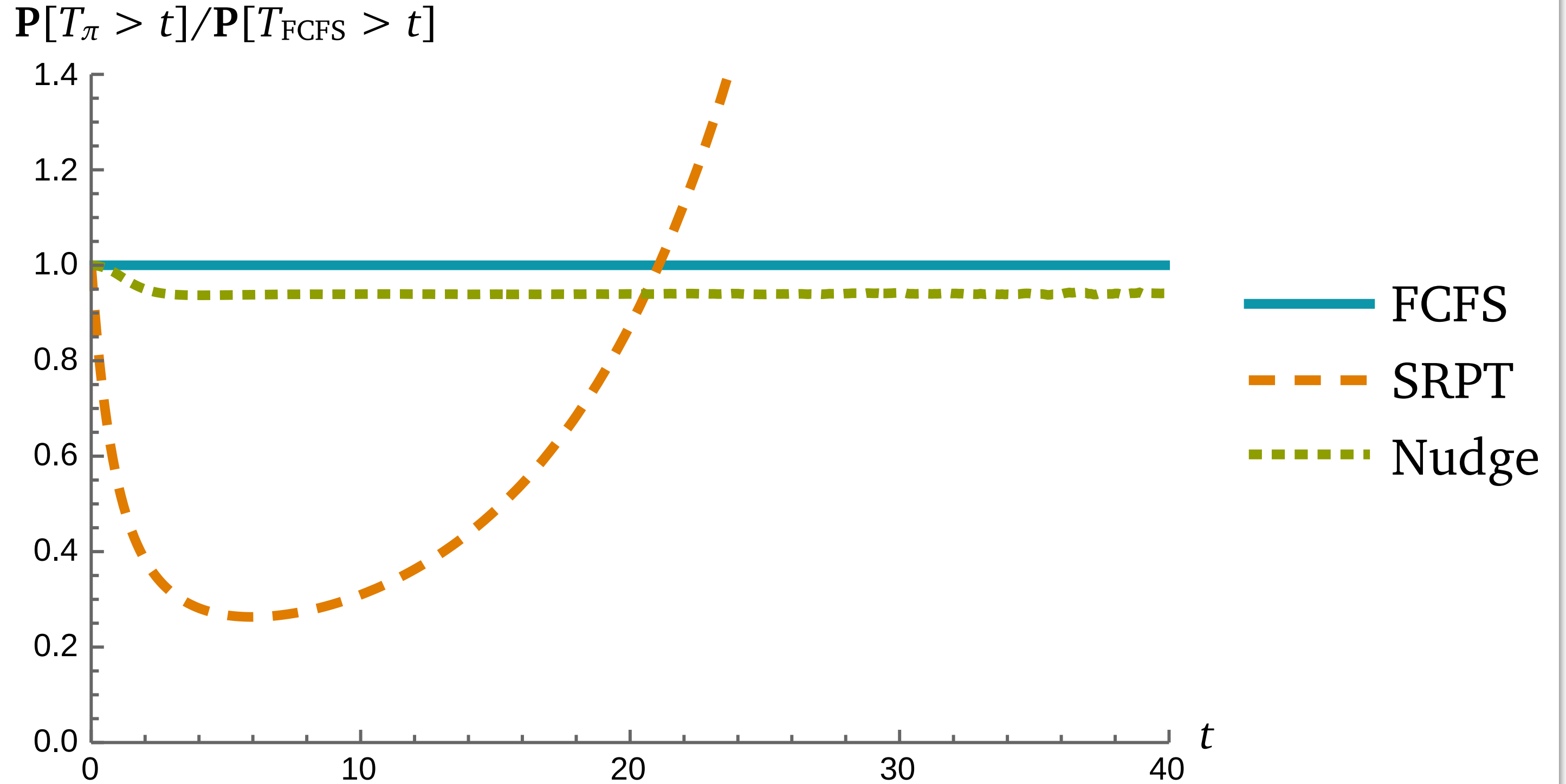
George Yu



Empirical results: **tail scheduling**



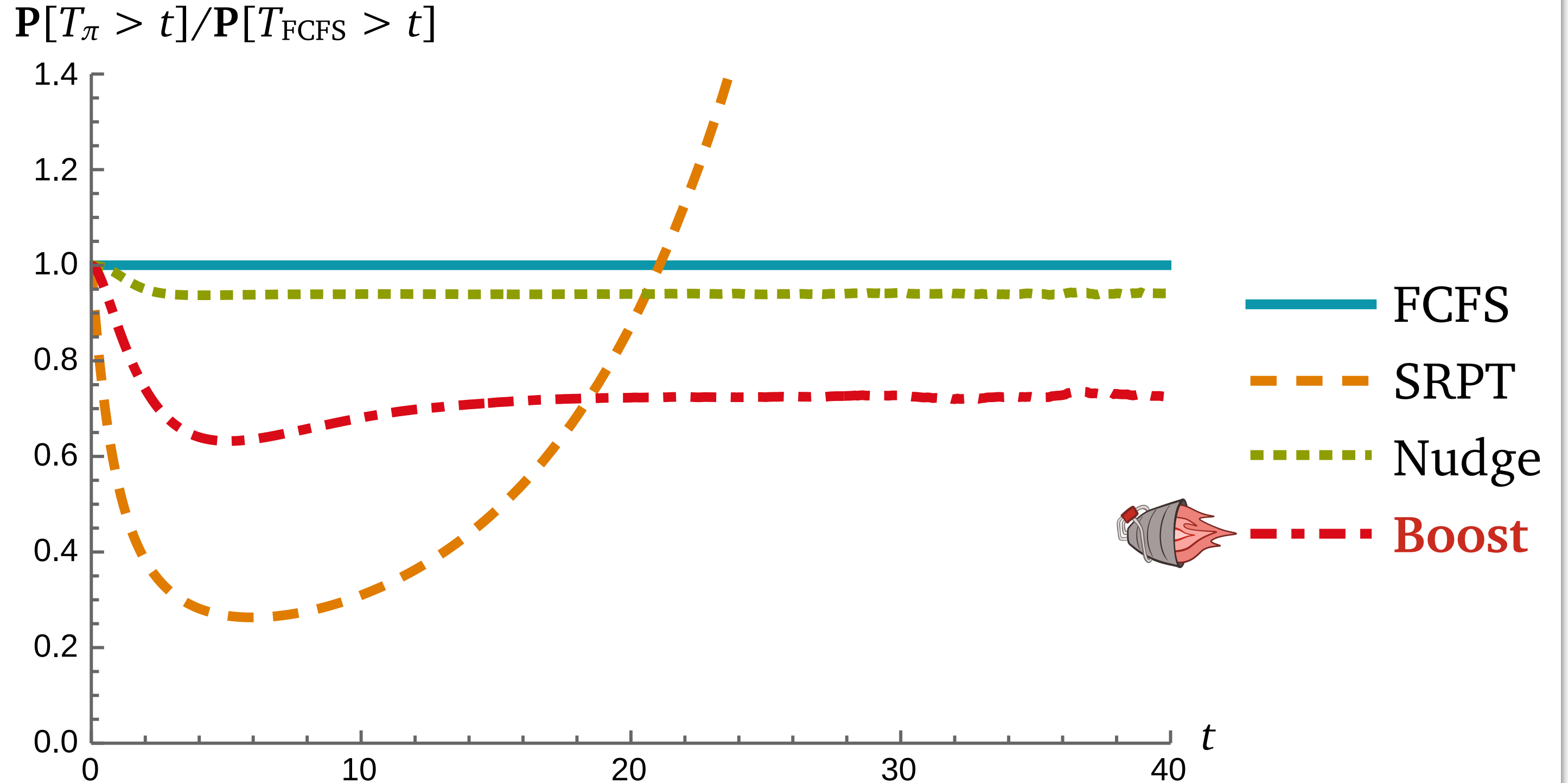
George Yu



Empirical results: **tail scheduling**



George Yu



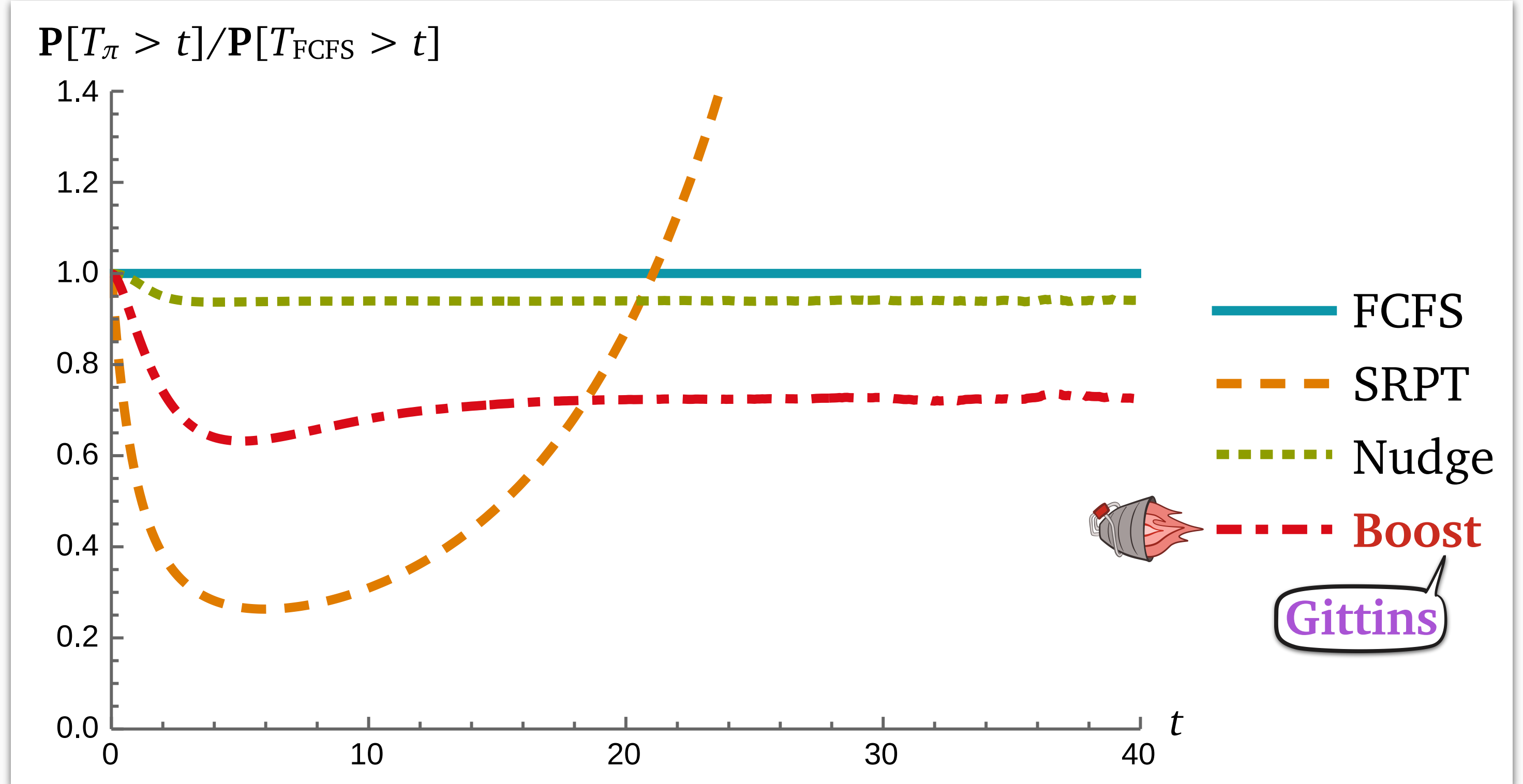
Empirical results: **tail scheduling**



George Yu

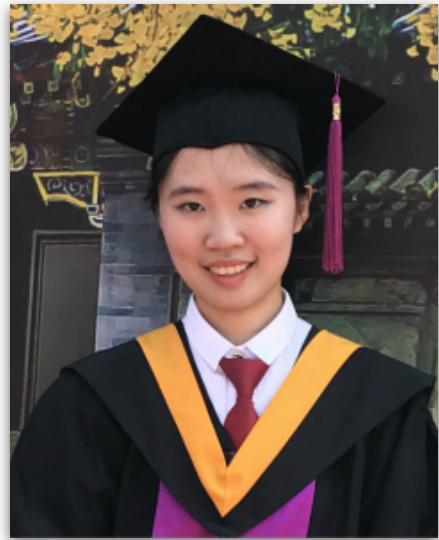


Amit Harlev



See our SIGMETRICS papers [Yu & Scully, 2024; Harlev et al., 2025; Yu et al, 2026]

Empirical results: Bayesian optimization



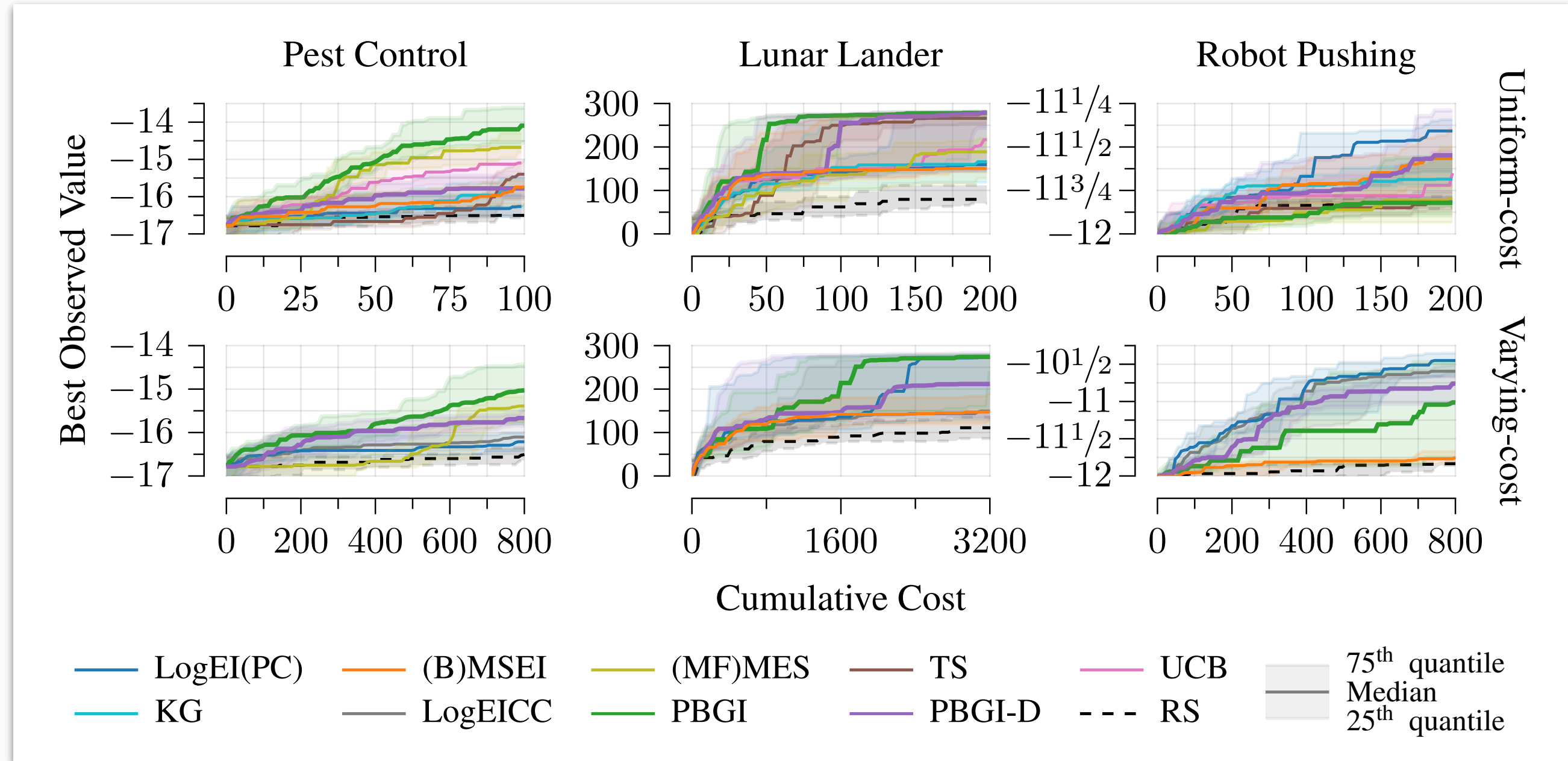
Qian Xie



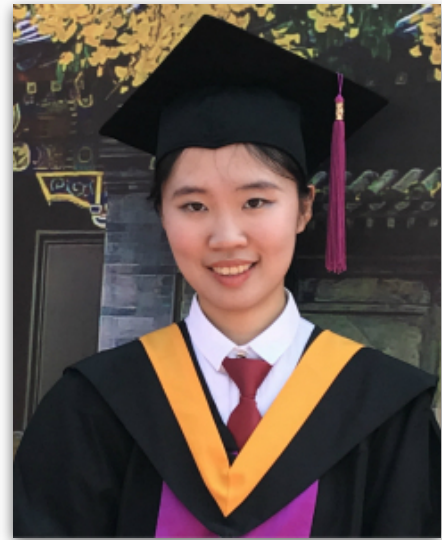
Raul Astudillo



Peter Frazier



Empirical results: Bayesian optimization



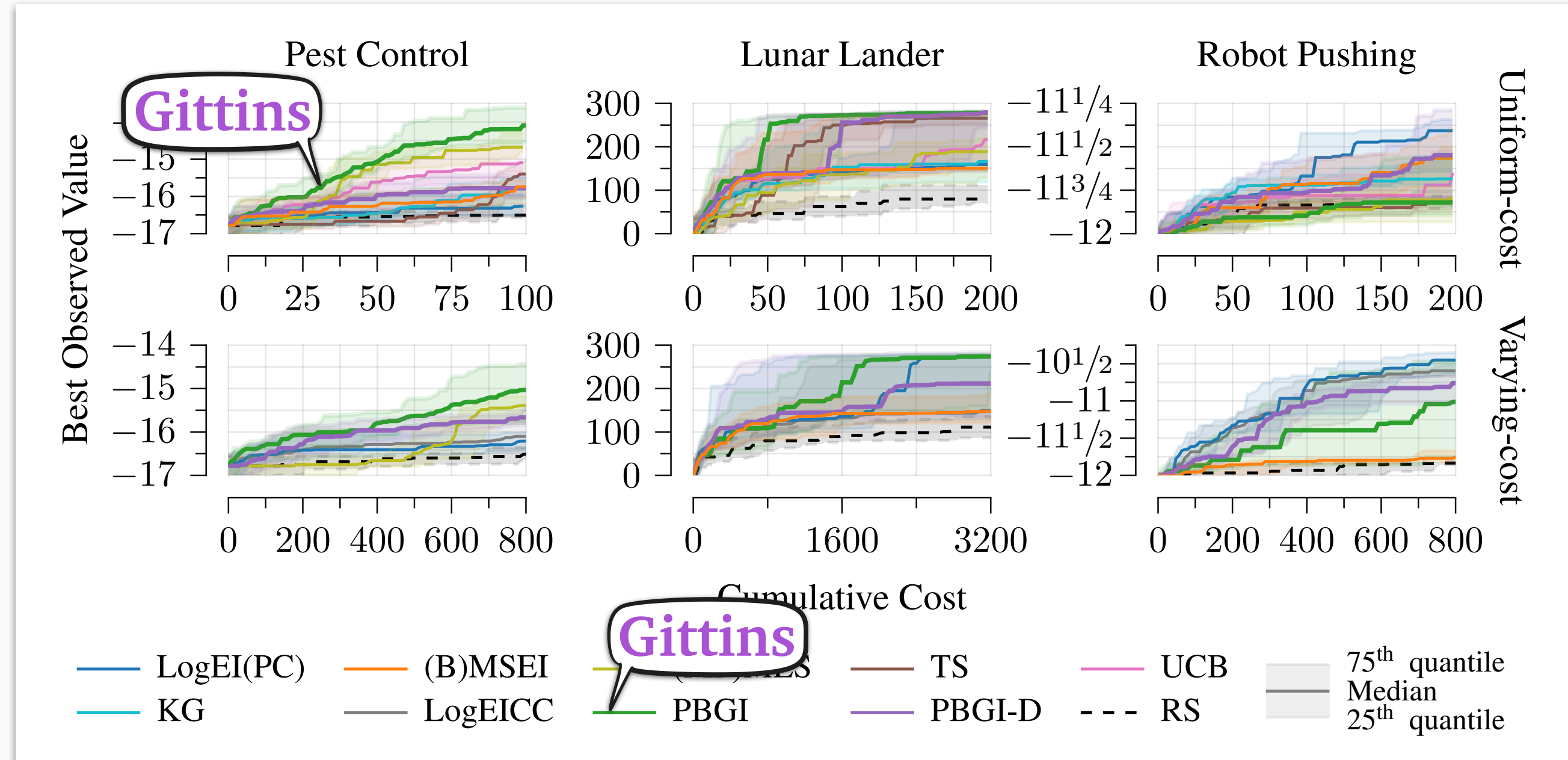
Qian Xie



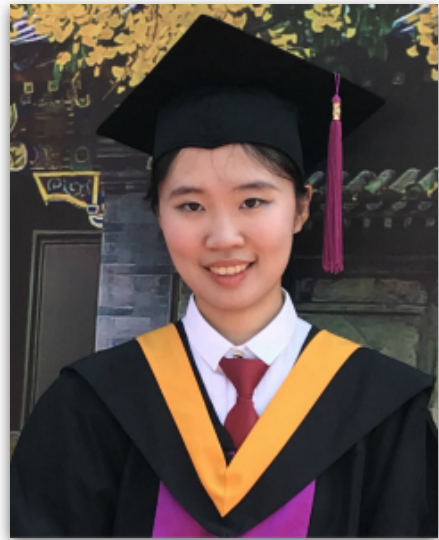
Raul Astudillo



Peter Frazier



Empirical results: Bayesian optimization



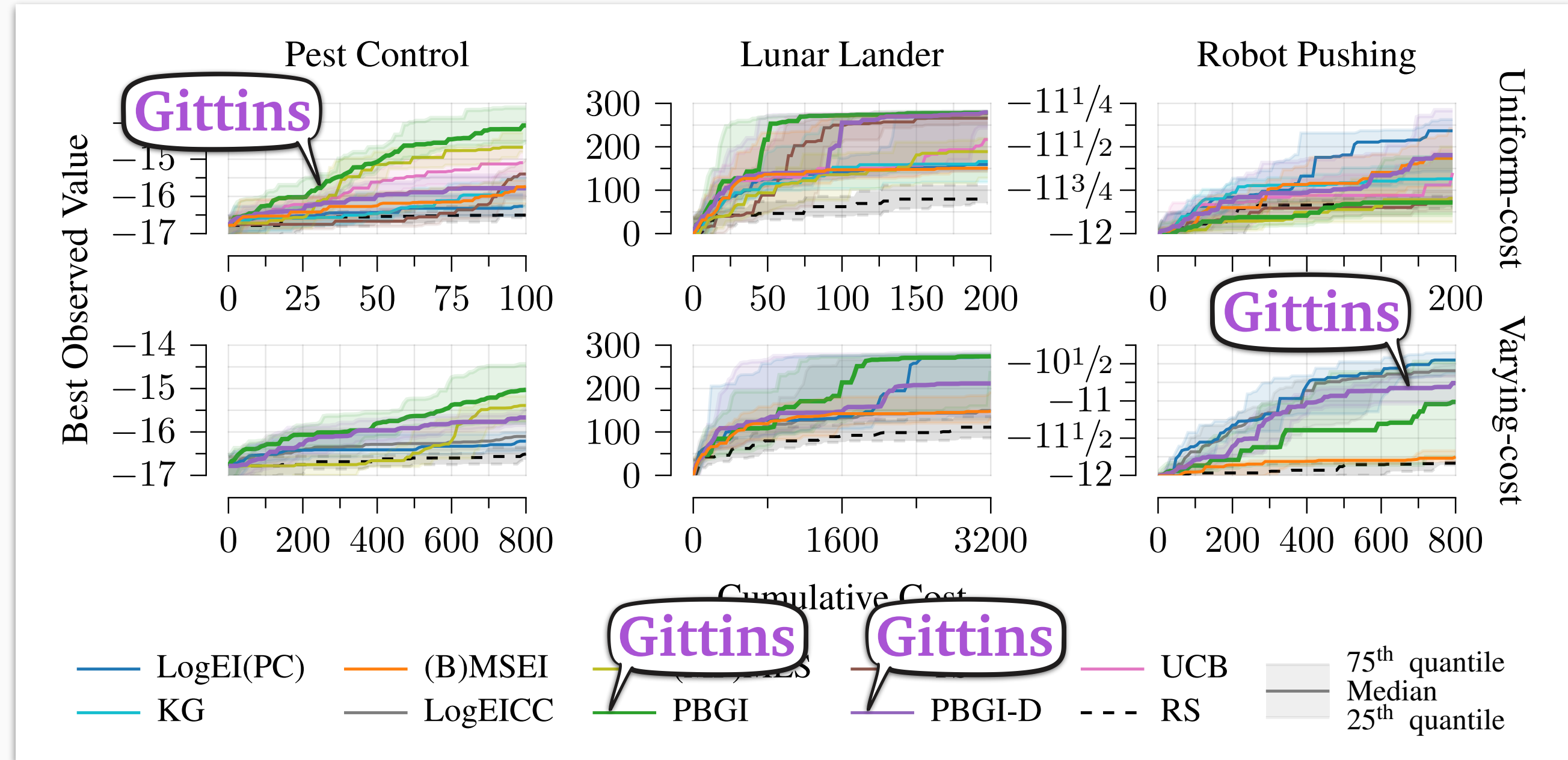
Qian Xie



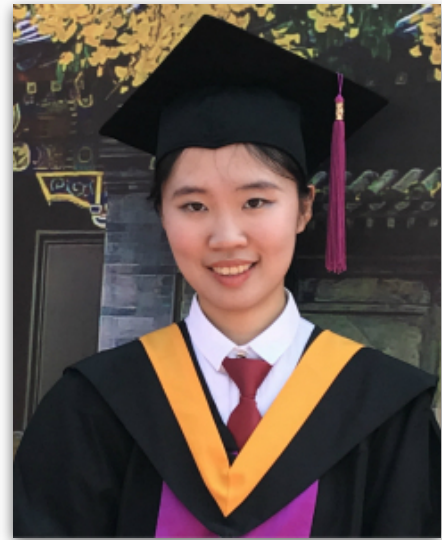
Raul Astudillo



Peter Frazier



Empirical results: Bayesian optimization



Qian Xie



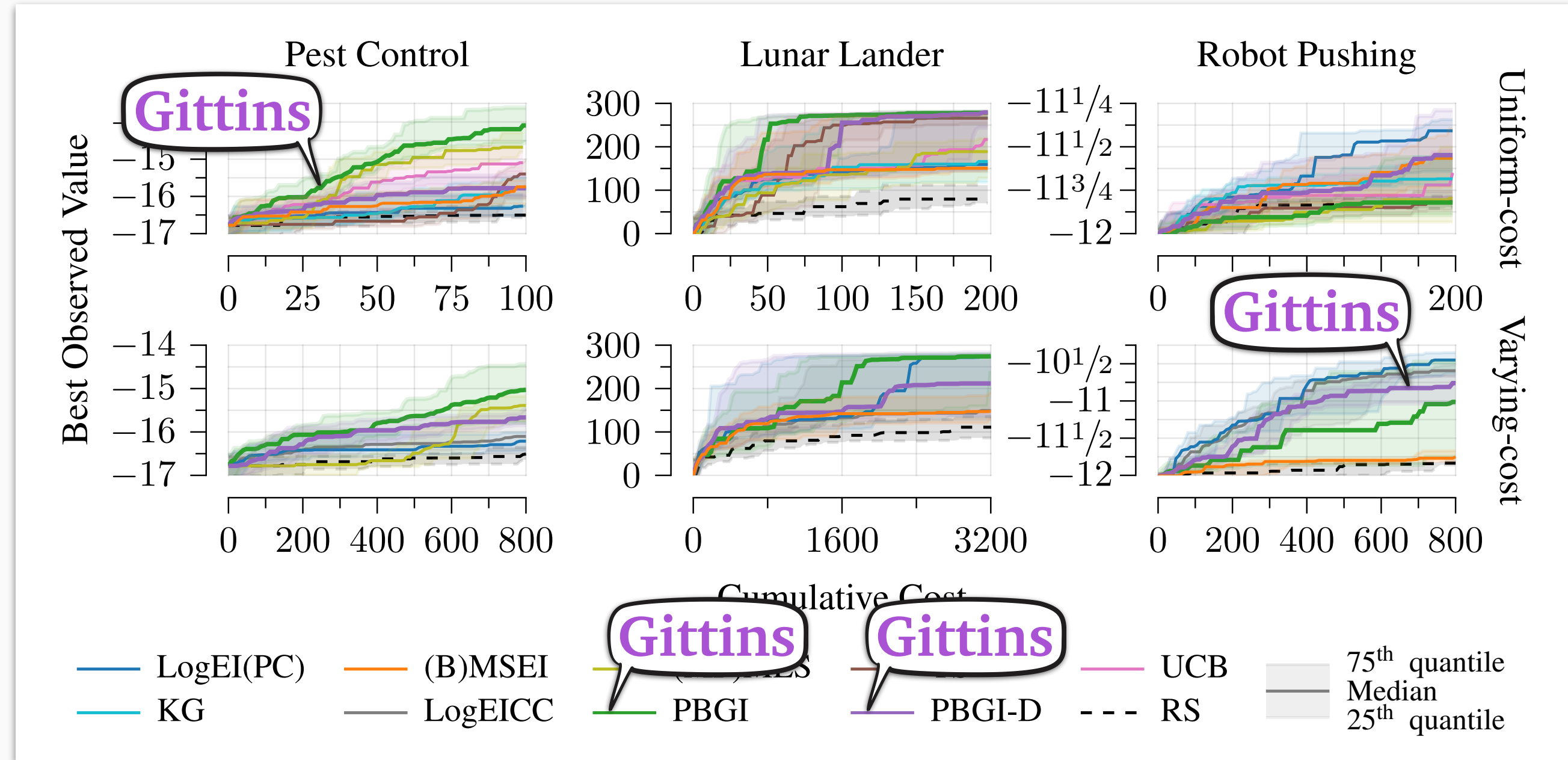
Raul Astudillo



Peter Frazier



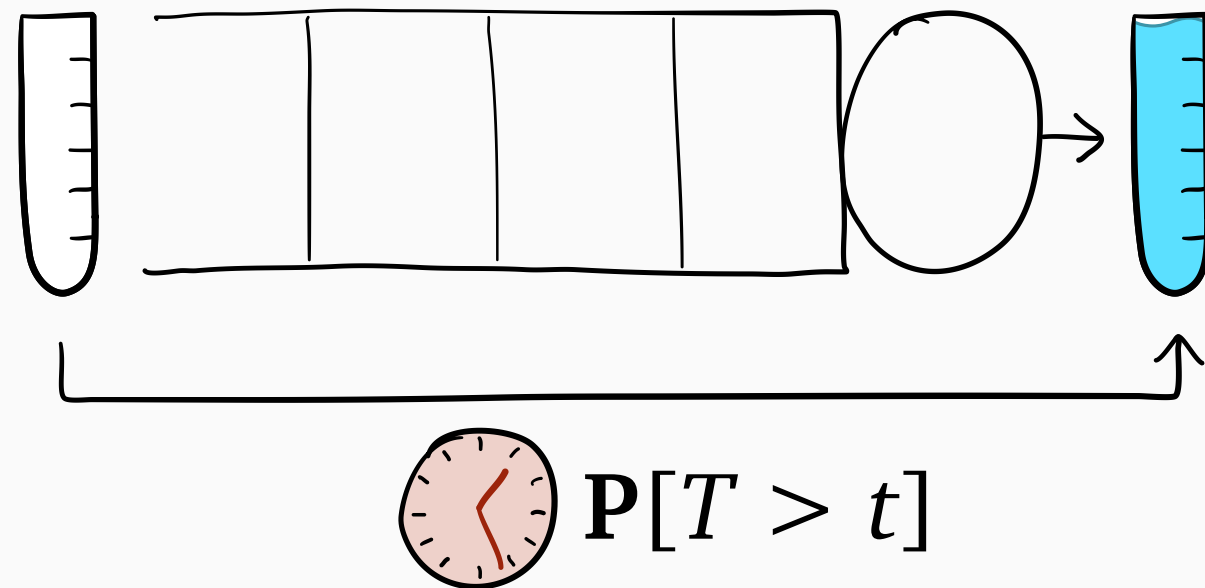
Linda Cai



See our NeurIPS 2024 paper and arXiv preprint [Xie et al., 2024; Xie et al., 2025]

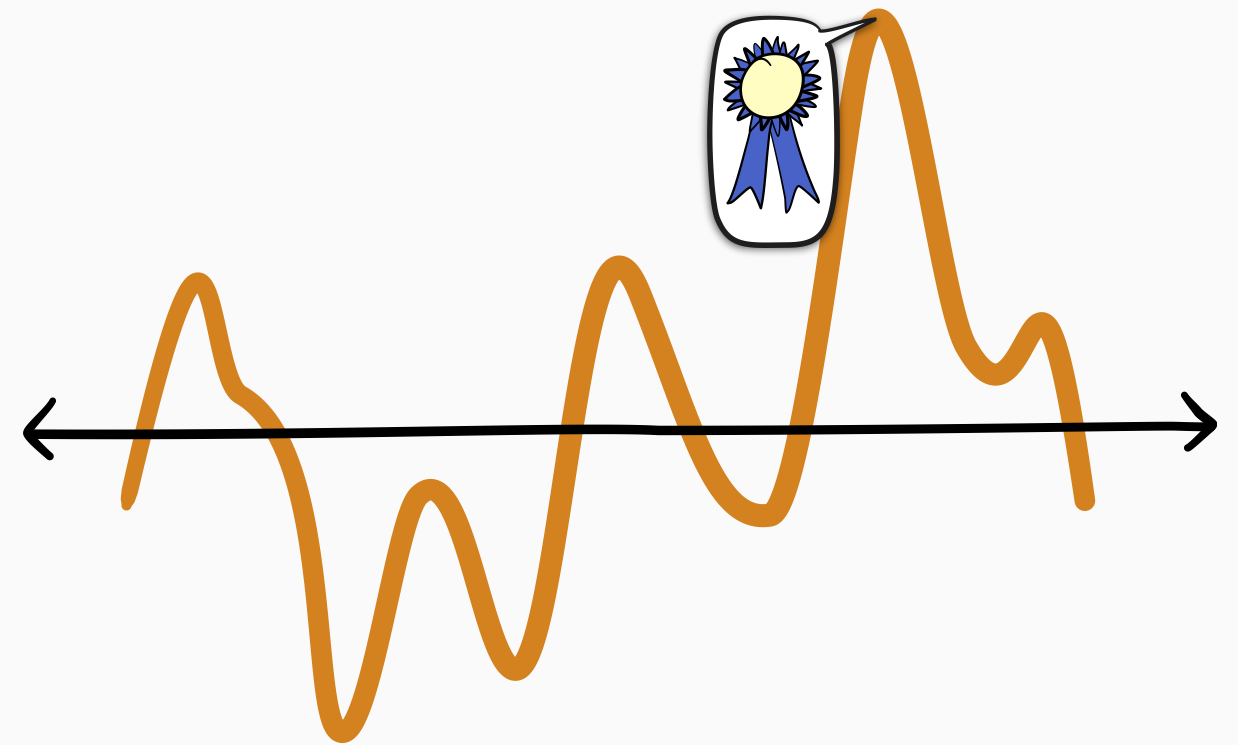
Two new Gittins applications

Tail scheduling



Goal: minimize probability of very long response time

Bayesian optimization

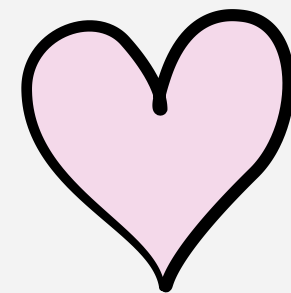


Goal: find large function value with few function evaluations

Two new **Gittins** applications

Tail scheduling

Bayesian optimization



Potential for *practical impact*

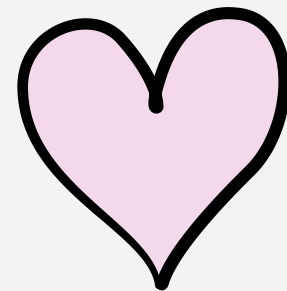
Goal: minimize
of very long

function value
in evaluations

Two new Gittins applications

Tail scheduling

Bayesian optimization



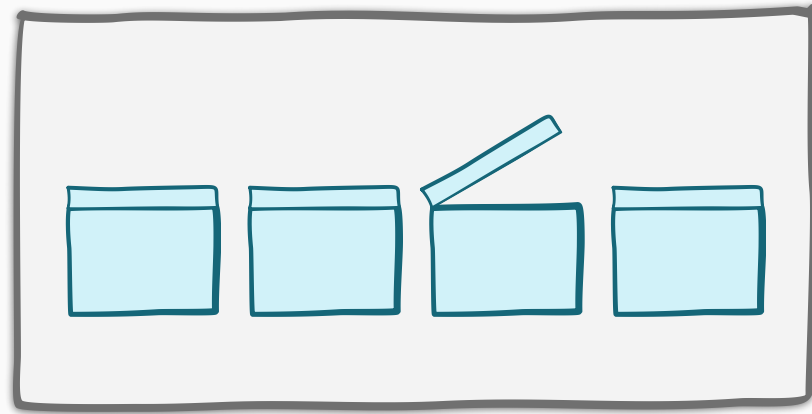
Potential for *practical impact*



Out of scope of classic theory

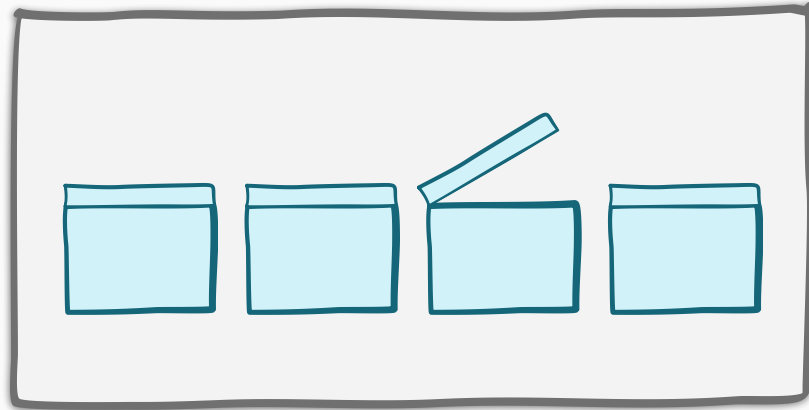
Goal: minim
of very long

function value
evaluations

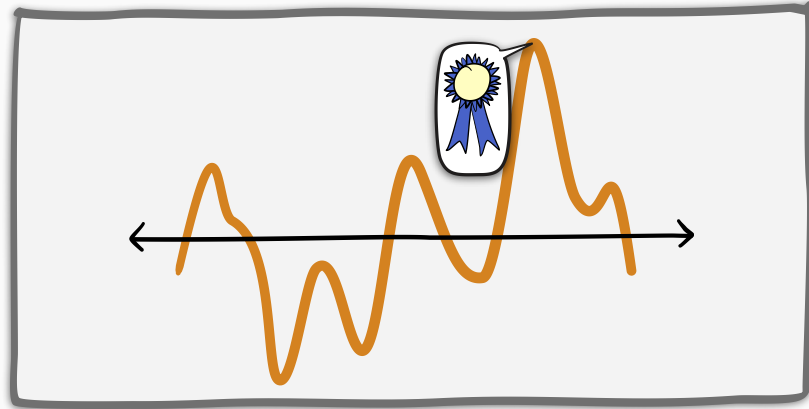


Part I:

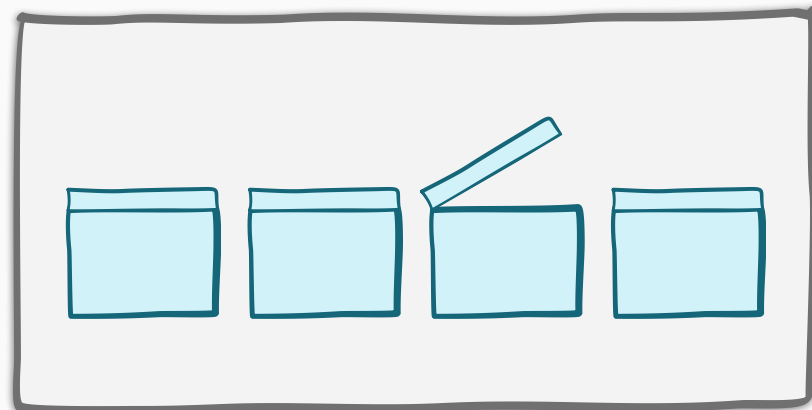
Introducing **Gittins** via **Pandora's box**



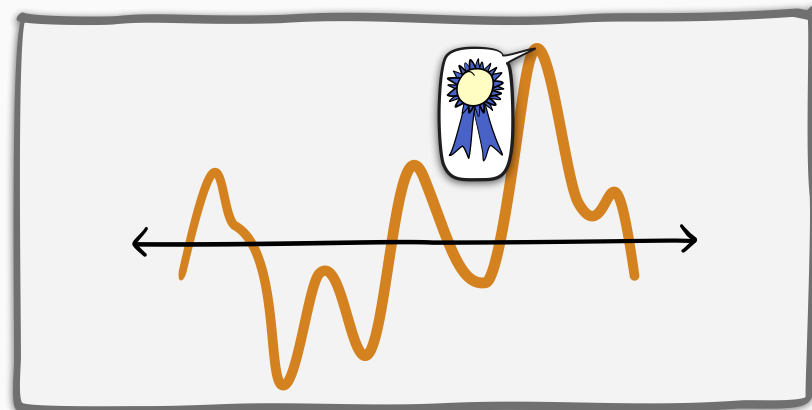
Part I:
Introducing **Gittins** via **Pandora's box**



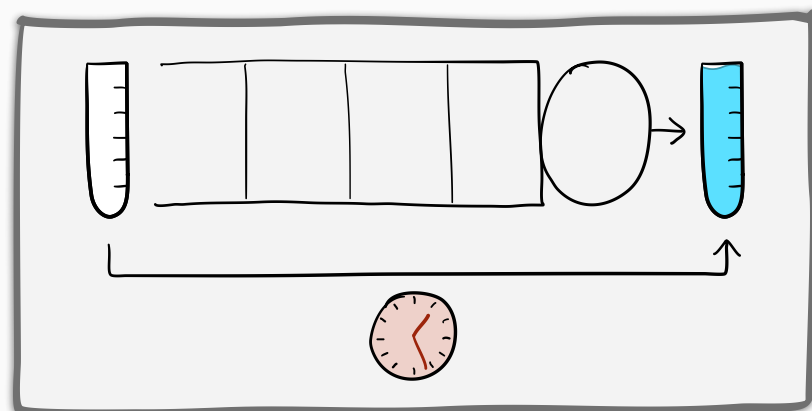
Part II: 
Gittins for **Bayesian optimization**



Part I:
Introducing **Gittins** via **Pandora's box**



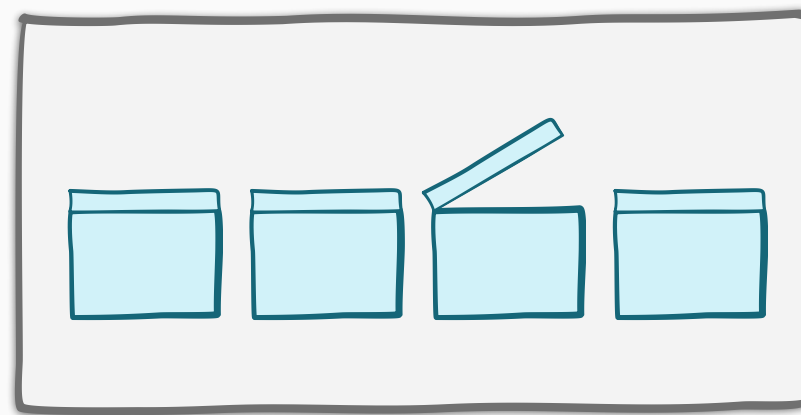
Part II: **NEW!**
Gittins for **Bayesian optimization**



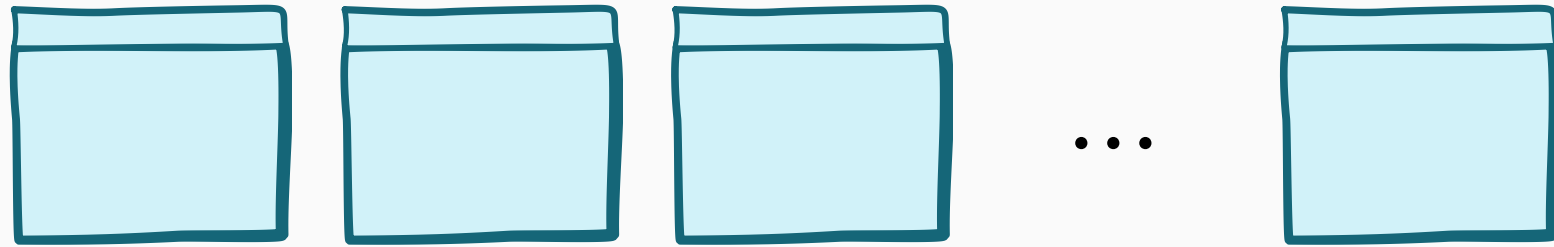
Part III: **NEW!**
Gittins for **tail scheduling**

Part I:

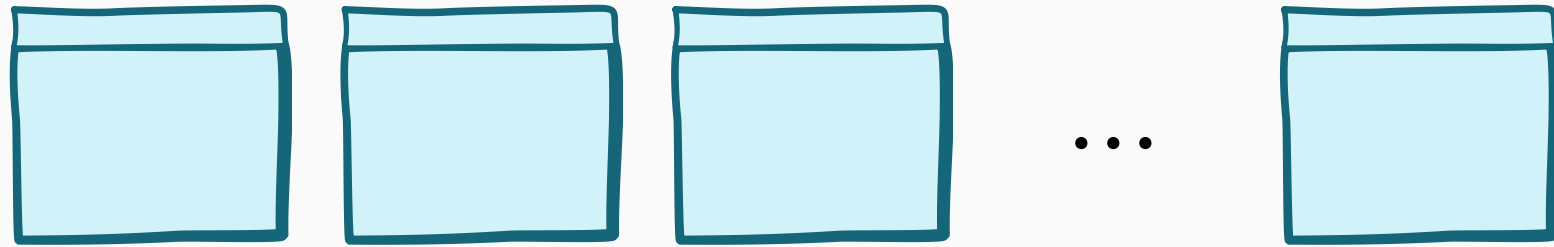
Introducing Gittins via Pandora's box



Pandora's box problem

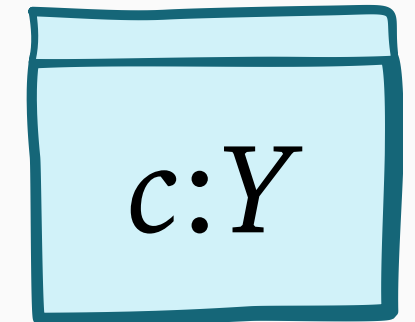


Pandora's box problem

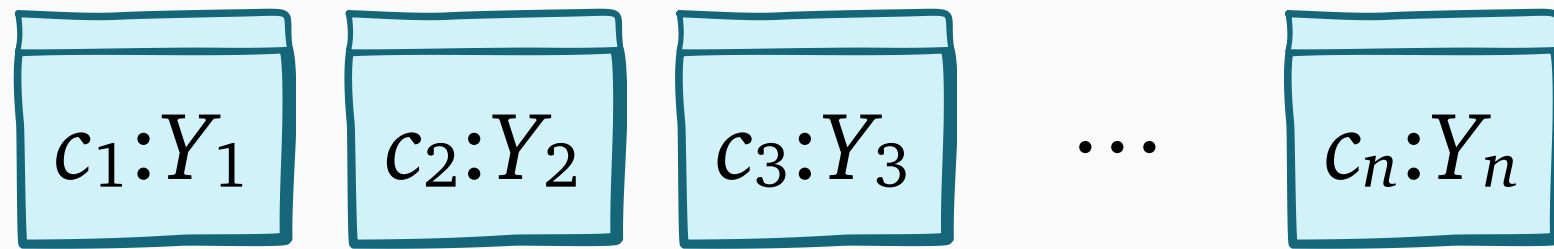


Each box:

- Opening cost c
- Hidden reward Y

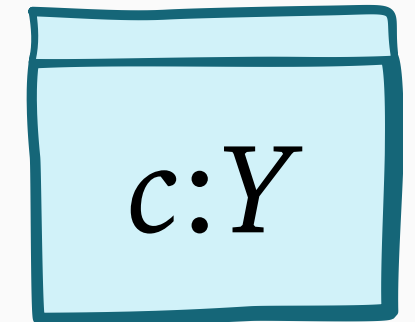


Pandora's box problem

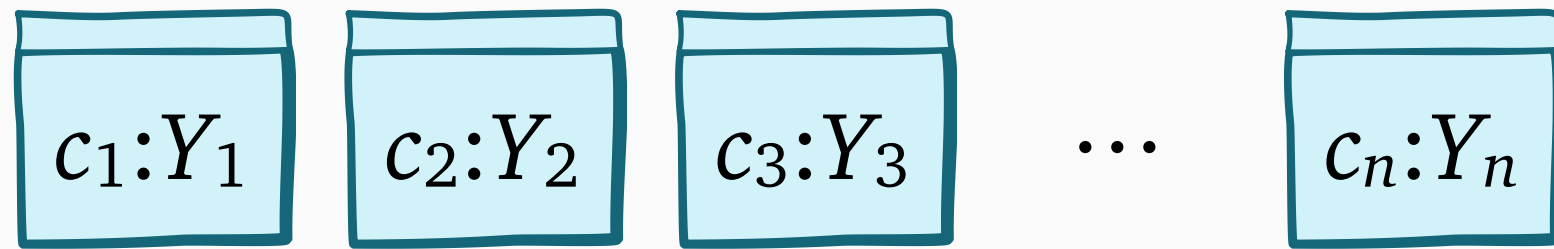


Each box:

- Opening cost c
- Hidden reward Y



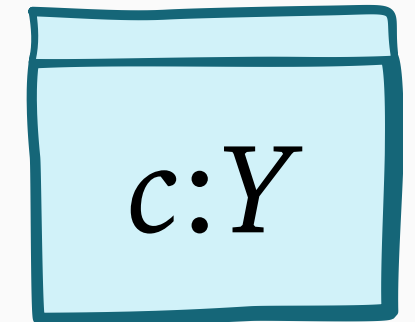
Pandora's box problem



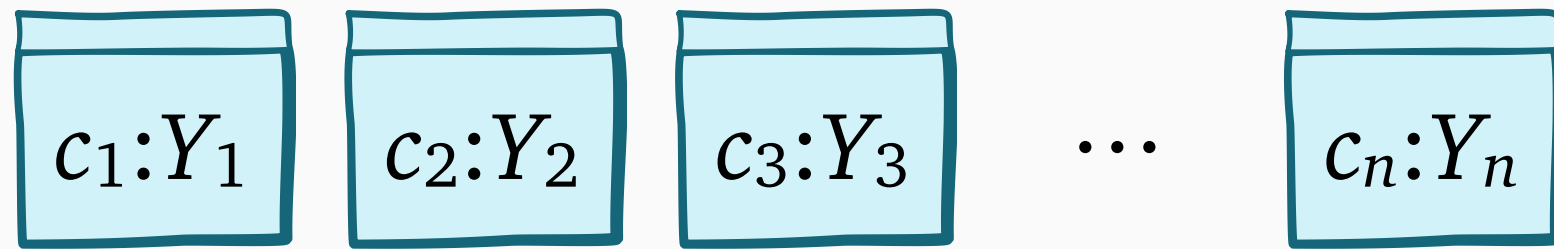
Each box:

- Opening cost c
- Hidden reward Y

independent



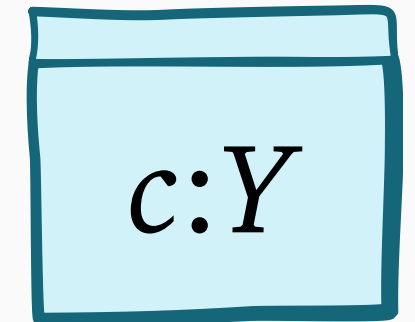
Pandora's box problem



Each box:

- Opening cost c
- Hidden reward Y

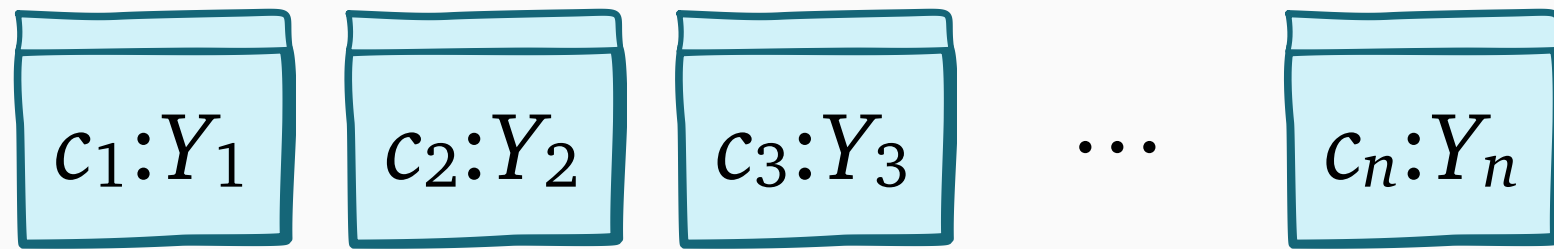
independent



Decision process:

- Open boxes one at a time
- Stop by selecting open box

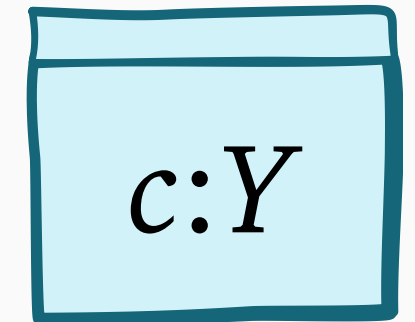
Pandora's box problem



Each box:

- Opening cost c
- Hidden reward Y

independent

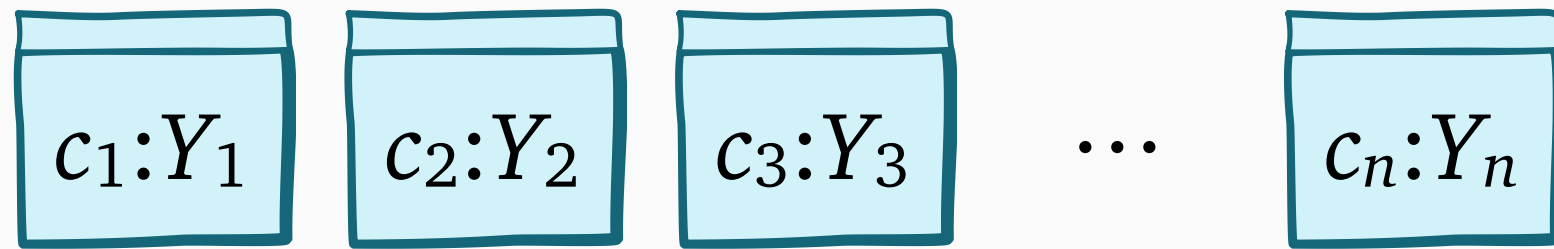


Decision process:

- Open boxes one at a time
- Stop by selecting open box

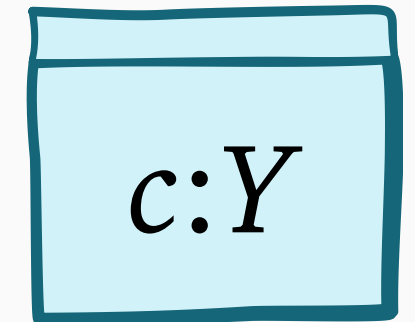
Goal: maximize $\mathbf{E}\left[Y_{\text{selected}} - \sum_{i \text{ opened}} c_i\right]$

Pandora's box problem



Each box:

- Opening cost c
- Hidden reward Y



independent

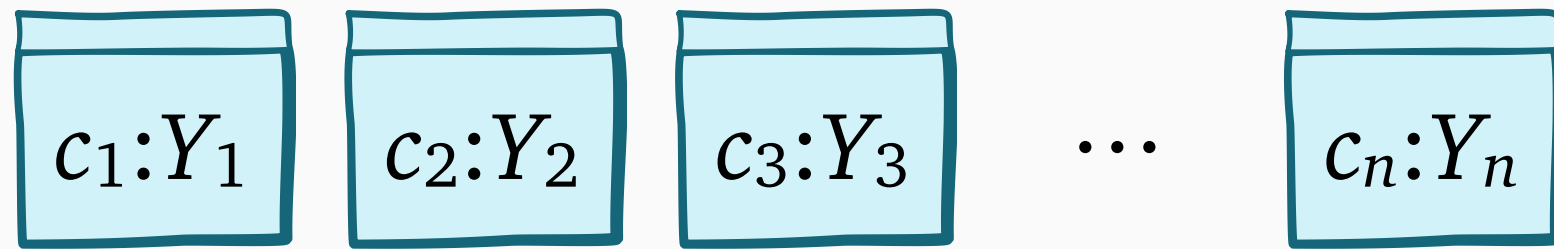
Decision process:

- Open boxes one at a time
- Stop by selecting open box

? Which box to open?

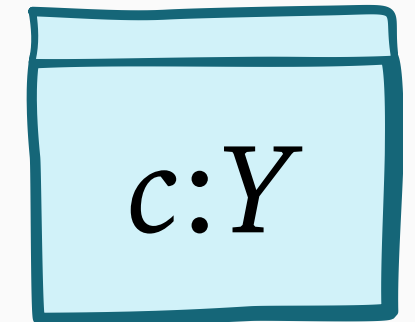
Goal: maximize $\mathbf{E}\left[Y_{\text{selected}} - \sum_{i \text{ opened}} c_i\right]$

Pandora's box problem



Each box:

- Opening cost c
- Hidden reward Y



independent

Decision process:

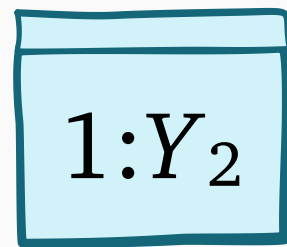
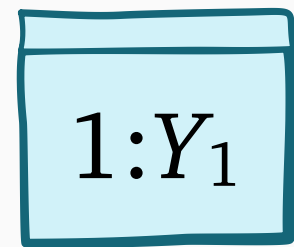
- Open boxes one at a time
- Stop by selecting open box

Goal: maximize $\mathbb{E}\left[Y_{\text{selected}} - \sum_{i \text{ opened}} c_i\right]$

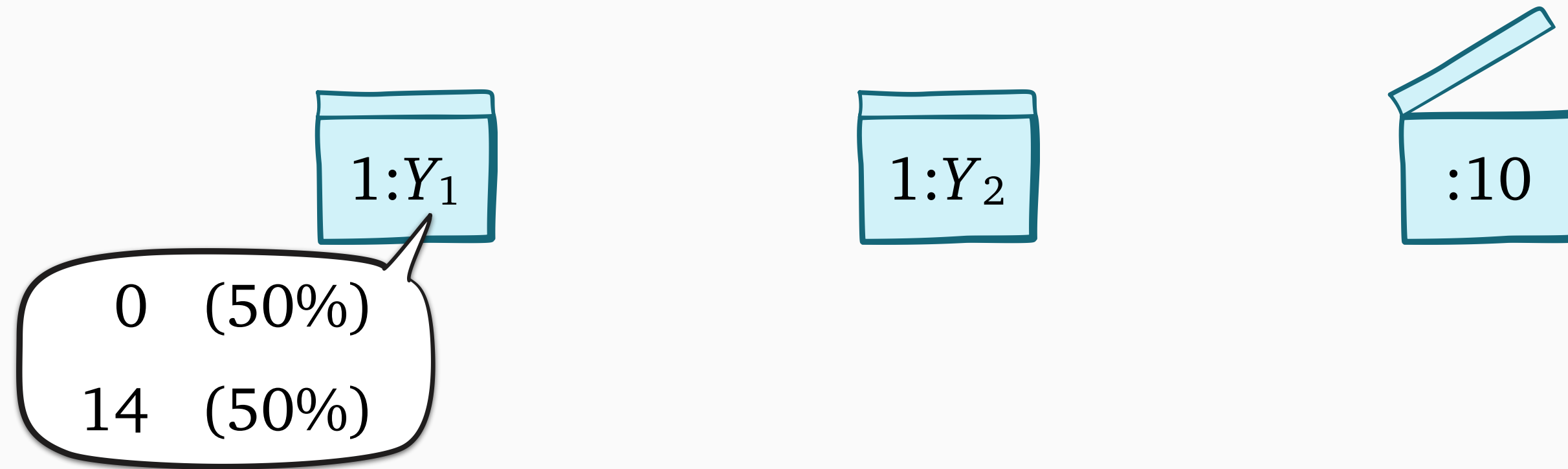
? Which box to open?

? Is it time to stop?

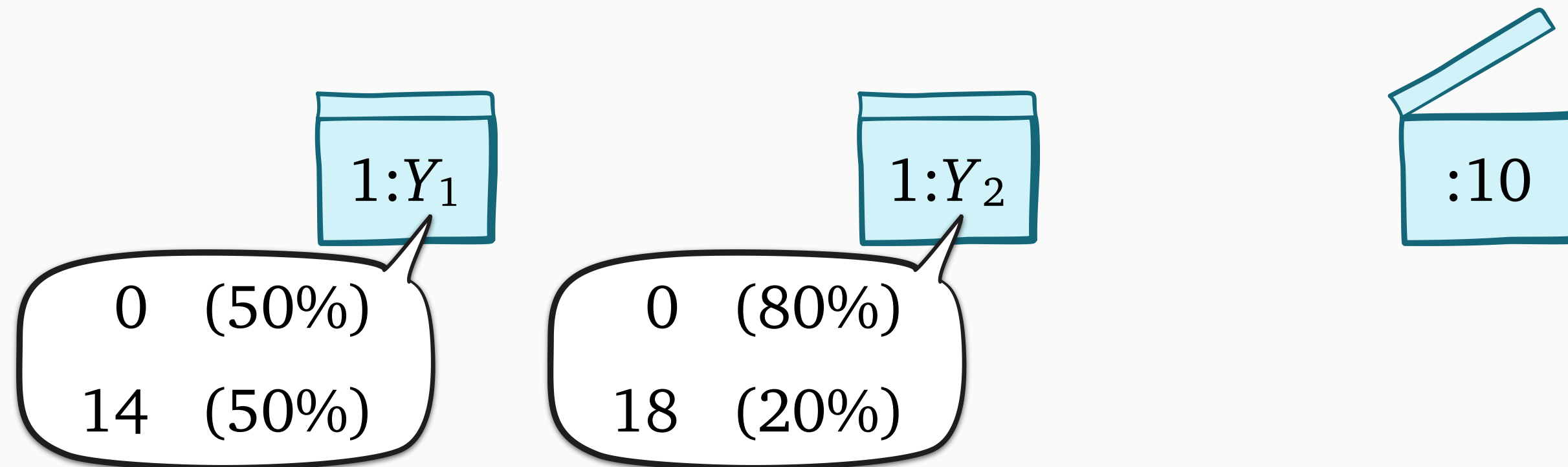
Why is Pandora's box hard?



Why is Pandora's box hard?



Why is Pandora's box hard?

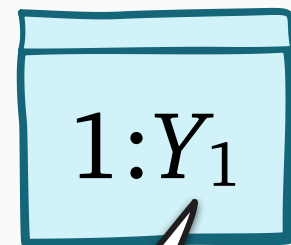


Why is Pandora's box hard?

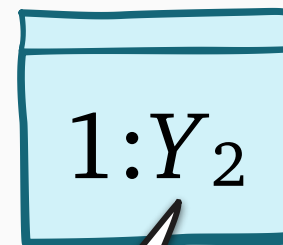


Use expected *net value*?

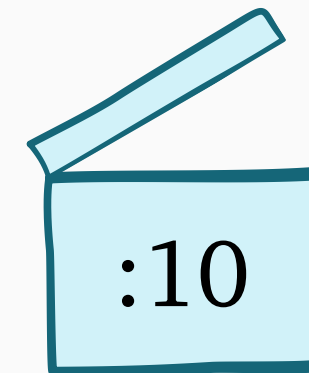
$$E[Y] - c$$



0	(50%)
14	(50%)



0	(80%)
18	(20%)

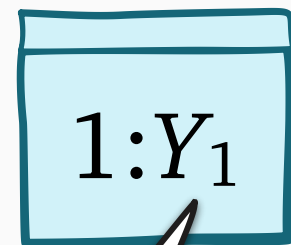


Why is Pandora's box hard?

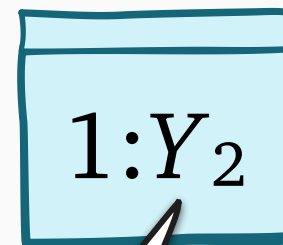


Use expected *net value*?

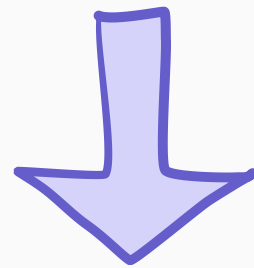
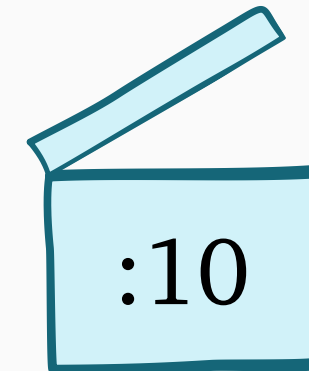
$$E[Y] - c$$



0	(50%)
14	(50%)



0	(80%)
18	(20%)



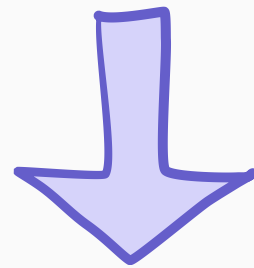
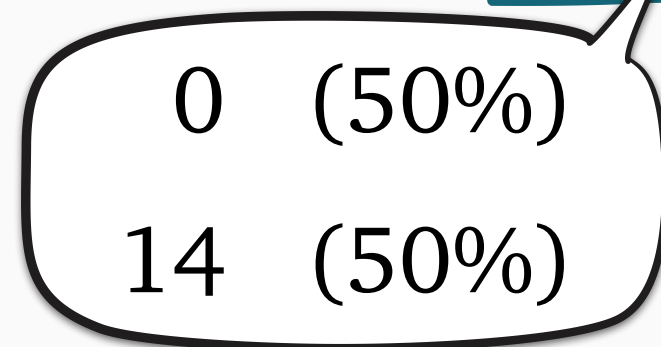
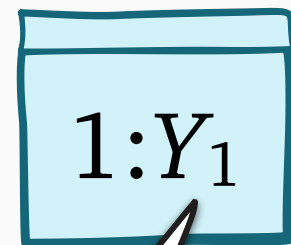
$$E[Y_1] - c_1 = 6$$

Why is Pandora's box hard?

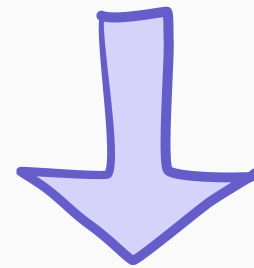
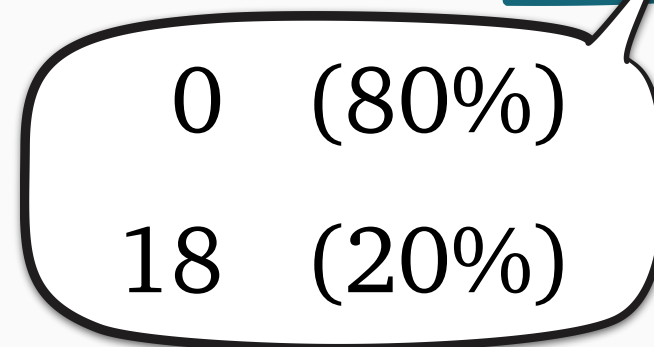
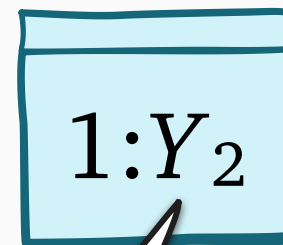


Use expected *net value*?

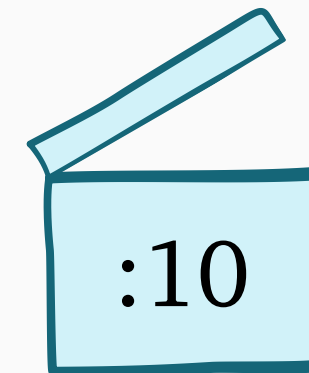
$$E[Y] - c$$



$$E[Y_1] - c_1 = 6$$



$$E[Y_2] - c_2 = 2.6$$

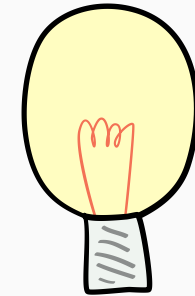


Why is Pandora's box hard?



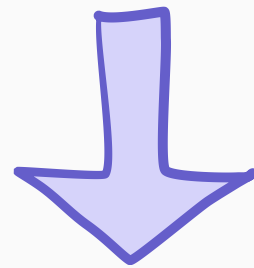
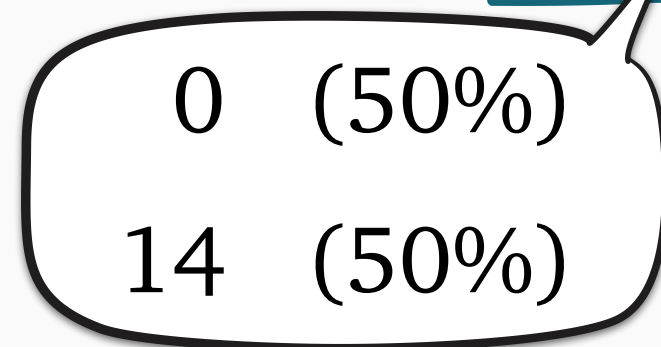
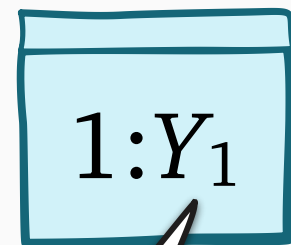
Use expected *net value*?

$$\mathbf{E}[Y] - c$$

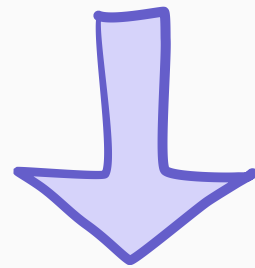
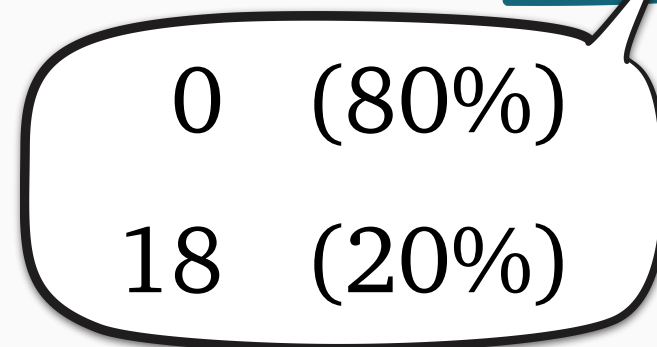
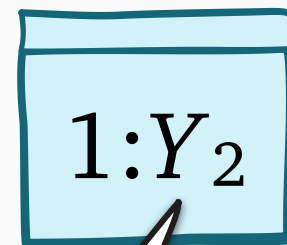


Expected improvement of Y over r :

$$\text{EI}(Y, r) = \mathbf{E}[\max(Y - r, 0)]$$



$$\mathbf{E}[Y_1] - c_1 = 6$$



$$\mathbf{E}[Y_2] - c_2 = 2.6$$

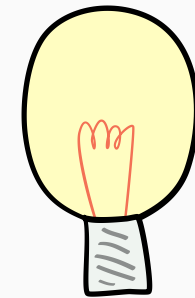


Why is Pandora's box hard?



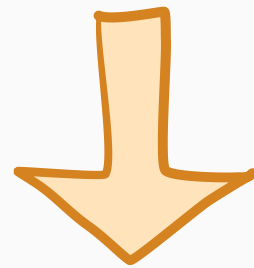
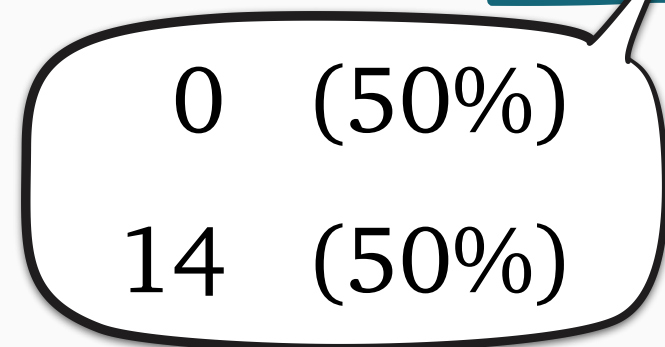
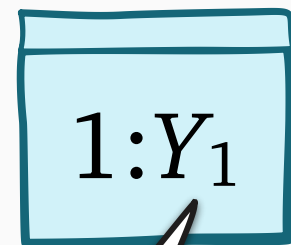
Use expected *net value*?

$$\mathbf{E}[Y] - c$$

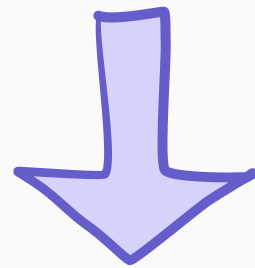
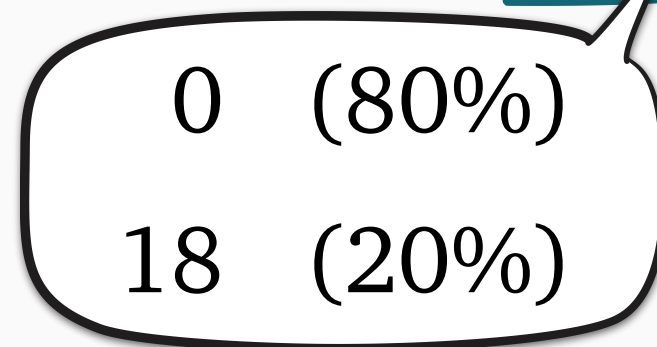
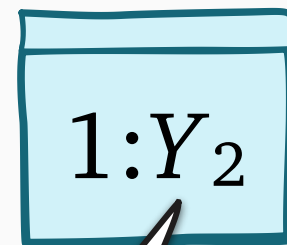


Expected improvement of Y over r :

$$\text{EI}(Y, r) = \mathbf{E}[\max(Y - r, 0)]$$



$$\text{EI}(Y_1, 10) = 2$$



$$\mathbf{E}[Y_2] - c_2 = 2.6$$

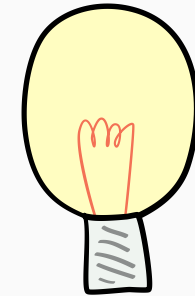


Why is Pandora's box hard?



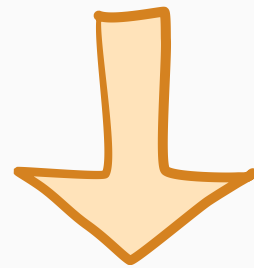
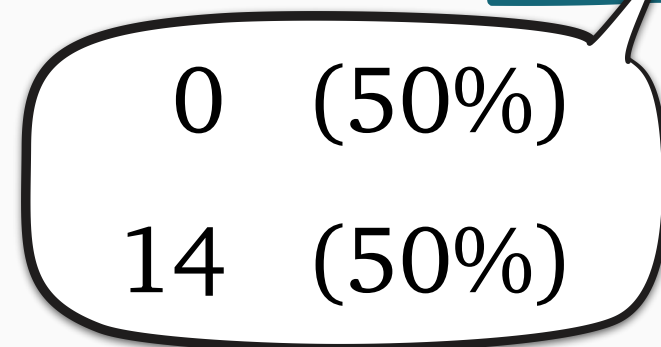
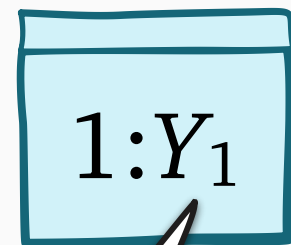
Use expected *net value*?

$$\mathbf{E}[Y] - c$$

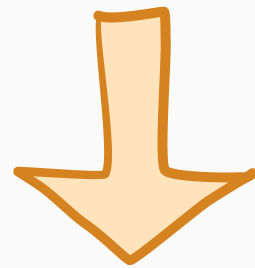
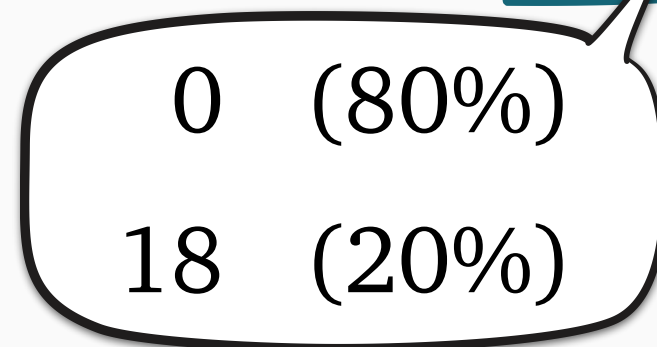
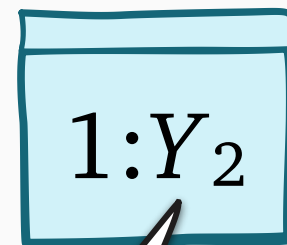


Expected improvement of Y over r :

$$\text{EI}(Y, r) = \mathbf{E}[\max(Y - r, 0)]$$



$$\text{EI}(Y_1, 10) = 2$$



$$\text{EI}(Y_2, 10) = 1.6$$

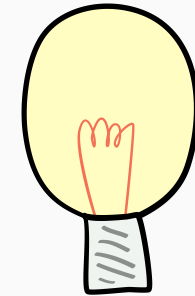


Why is Pandora's box hard?



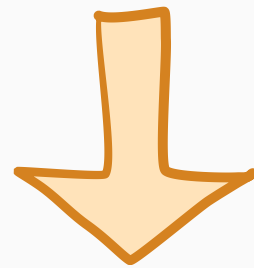
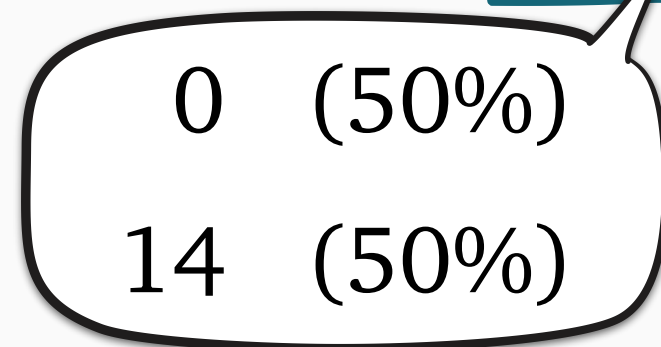
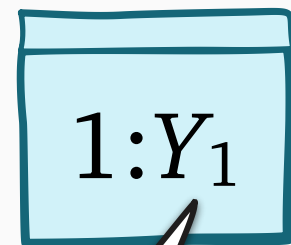
Use expected *net value*?

$$\mathbf{E}[Y] - c$$

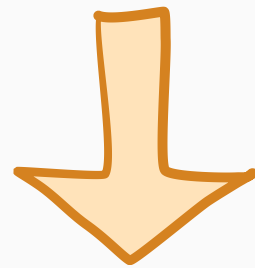
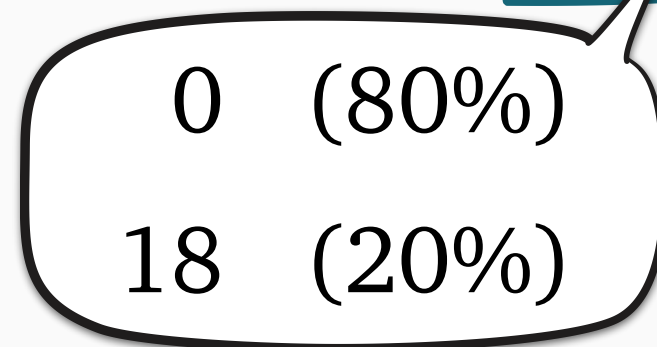
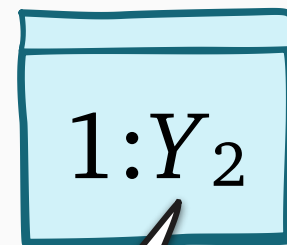


Expected improvement of Y over r :

$$\text{EI}(Y, r) = \mathbf{E}[\max(Y - r, 0)]$$



$$\text{EI}(Y_1, 10) = 2$$

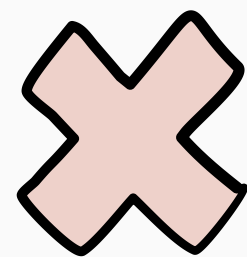


$$\text{EI}(Y_2, 10) = 1.6$$



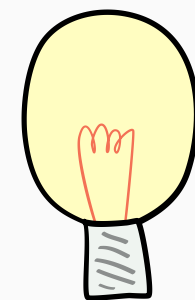
- Both boxes have $\text{EI}(Y_i, 10) > c_i$

Why is Pandora's box hard?



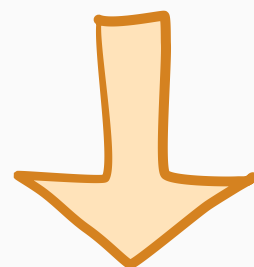
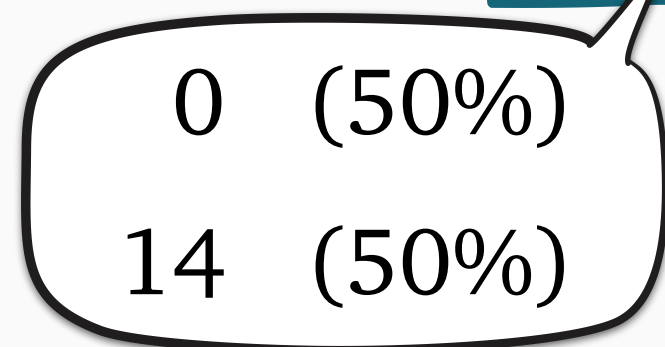
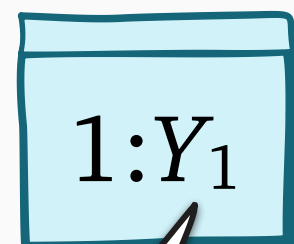
Use expected *net value*?

$$\mathbf{E}[Y] - c$$

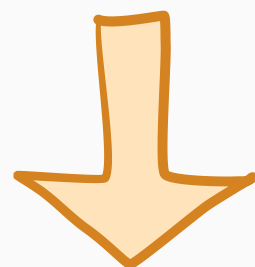
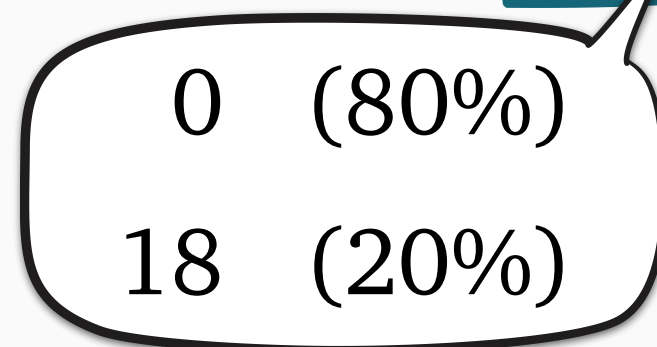
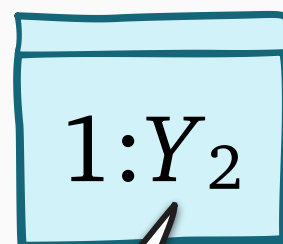


Expected improvement of Y over r :

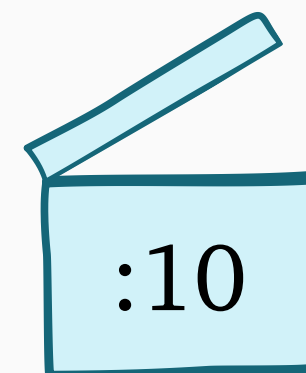
$$\text{EI}(Y, r) = \mathbf{E}[\max(Y - r, 0)]$$



$$\text{EI}(Y_1, 10) = 2$$

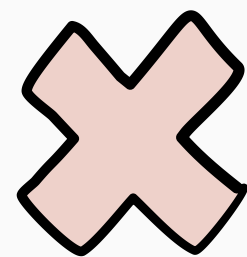


$$\text{EI}(Y_2, 10) = 1.6$$



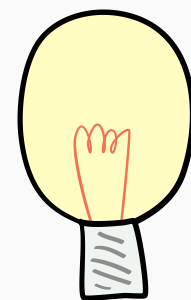
- Both boxes have $\text{EI}(Y_i, 10) > c_i$

Why is Pandora's box hard?



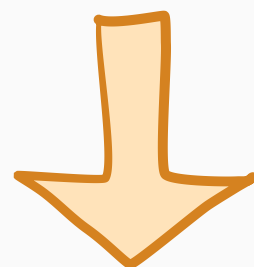
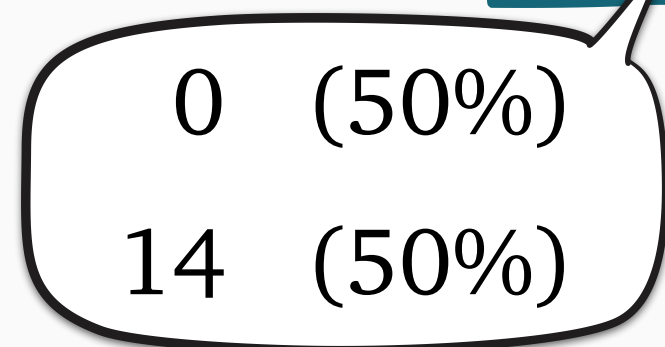
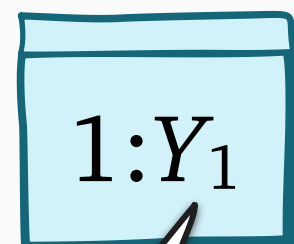
Use expected *net value*?

$$\mathbf{E}[Y] - c$$

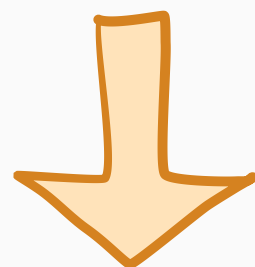
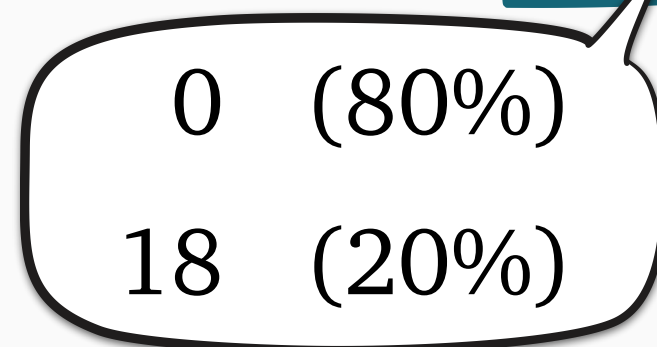
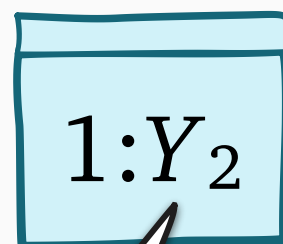


Expected improvement of Y over r :

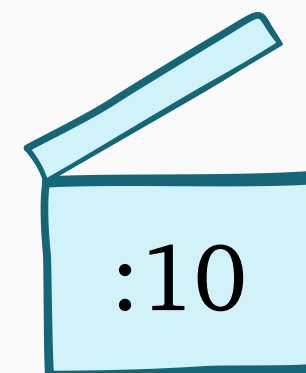
$$\text{EI}(Y, r) = \mathbf{E}[\max(Y - r, 0)]$$



$$\text{EI}(Y_1, 10) = 2$$

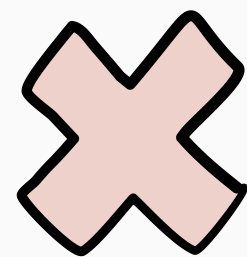


$$\text{EI}(Y_2, 10) = 1.6$$



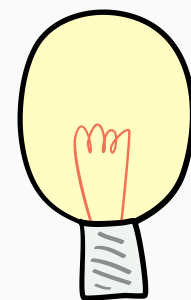
- Both boxes have $\text{EI}(Y_i, 10) > c_i$
- Box 1 has better EI

Why is Pandora's box hard?



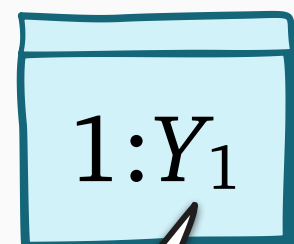
Use expected *net value*?

$$\mathbf{E}[Y] - c$$

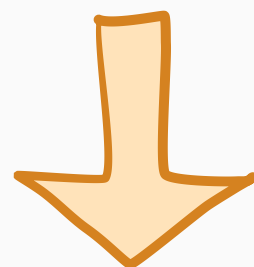


Expected improvement of Y over r :

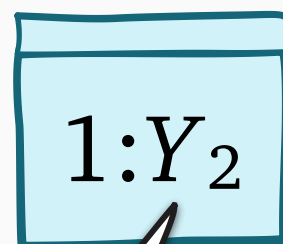
$$\text{EI}(Y, r) = \mathbf{E}[\max(Y - r, 0)]$$



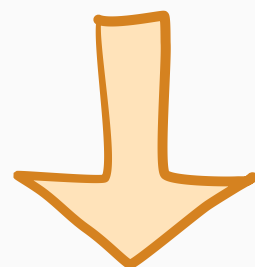
0	(50%)
14	(50%)



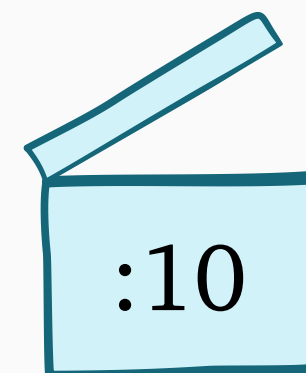
$$\text{EI}(Y_1, 10) = 2$$



0	(80%)
18	(20%)



$$\text{EI}(Y_2, 10) = 1.6$$

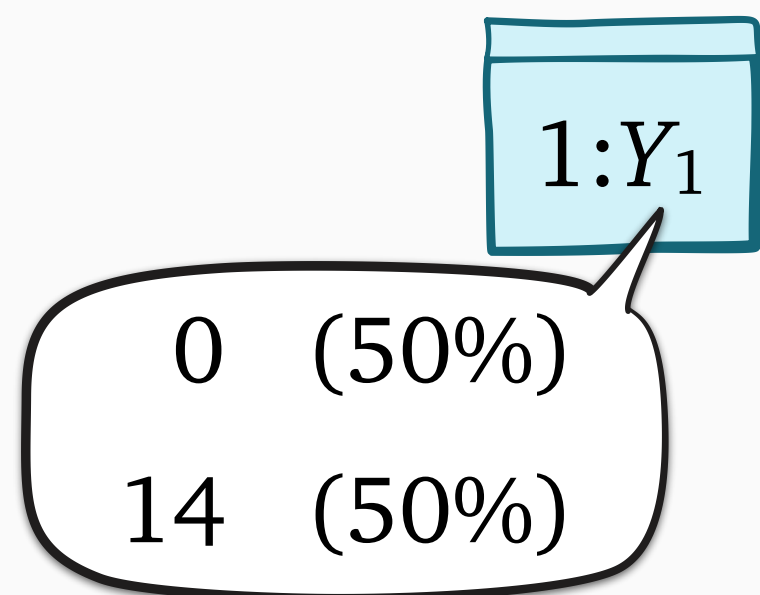


- Both boxes have $\text{EI}(Y_i, 10) > c_i$
- Box 1 has better EI
- **Optimal action:** *open box 2!*

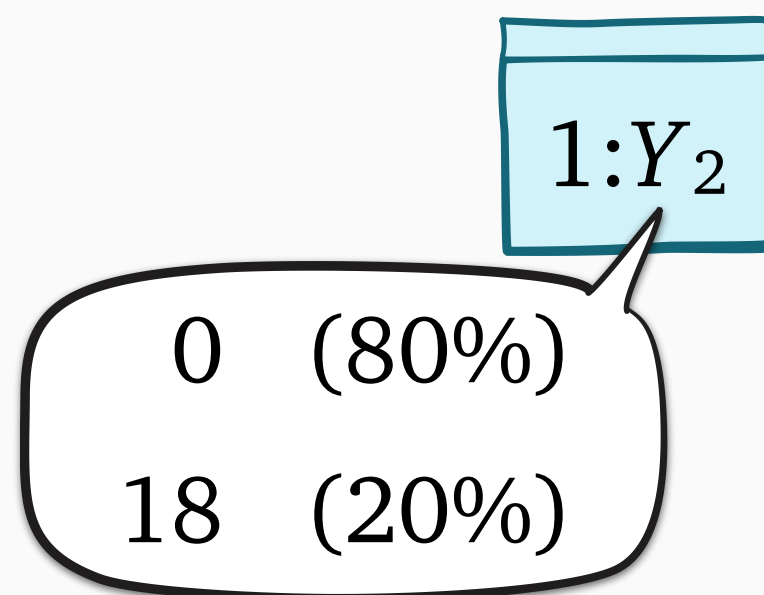
Why is Pandora's box hard?

✗ Use expected *net value*?
 $E[Y] - c$

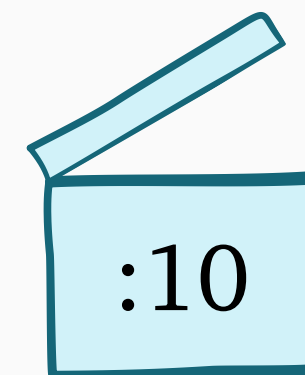
✗ Expected improvement of Y over r :
 $EI(Y, r) = E[\max(Y - r, 0)]$



$EI(Y_1, 10) = 2$



$EI(Y_2, 10) = 1.6$



- Both boxes have $EI(Y_i, 10) > c_i$
- Box 1 has better EI
- **Optimal action:** *open box 2!*

Optimal policy: Gittins

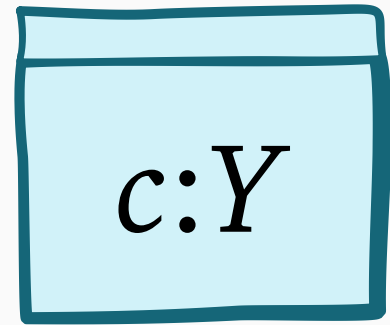
Optimal policy: **Gittins**

Step 1: *rate* each box separately

Step 2: *act* on box of best rating

Optimal policy: Gittins

Step 1: *rate* each box separately



Step 2: *act* on box of best rating

Optimal policy: **Gittins**

Step 1: *rate* each box separately



Step 2: *act* on box of best rating

Optimal policy: **Gittins**

Step 1: *rate* each box separately



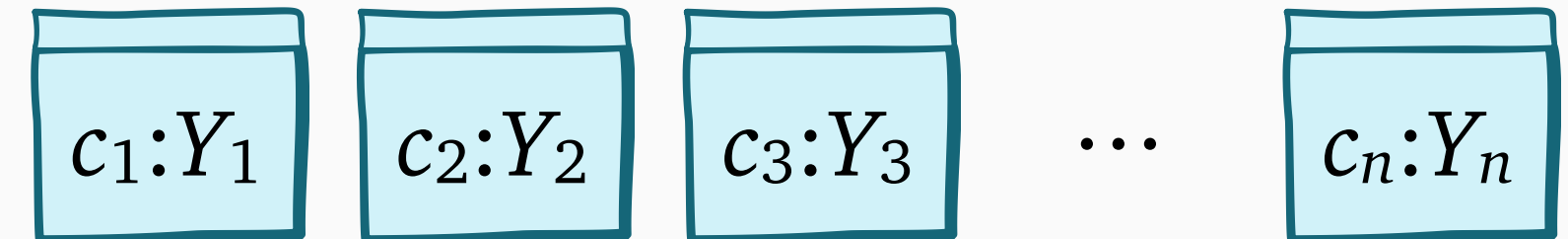
Step 2: *act* on box of best rating

Optimal policy: Gittins

Step 1: *rate* each box separately



Step 2: *act* on box of best rating

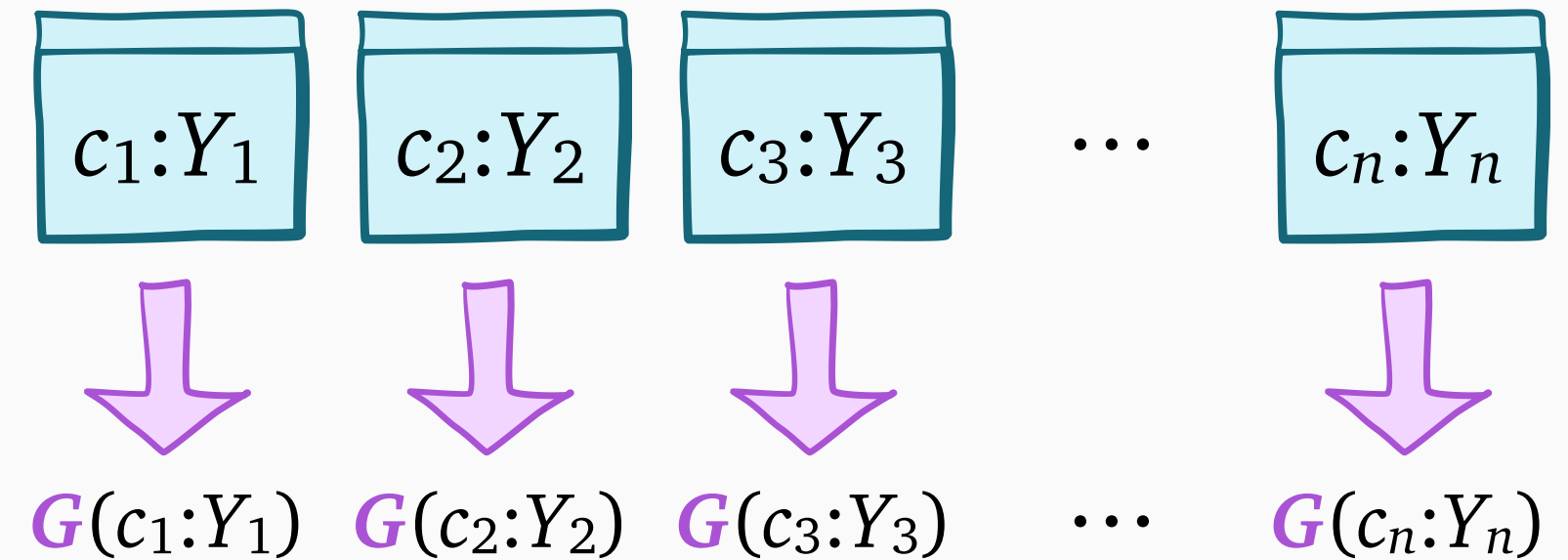


Optimal policy: **Gittins**

Step 1: *rate* each box separately



Step 2: *act* on box of best rating

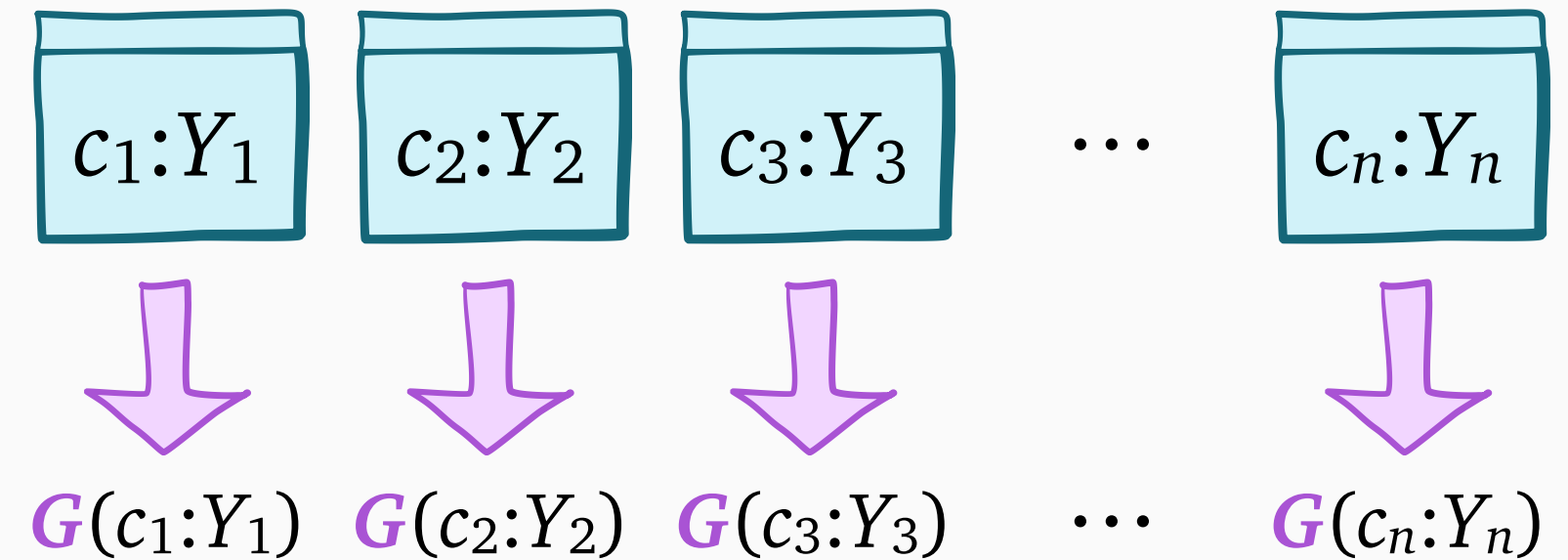


Optimal policy: **Gittins**

Step 1: *rate* each box separately



Step 2: *act* on box of best rating



Gittins policy: if box of max **Gittins** index is...

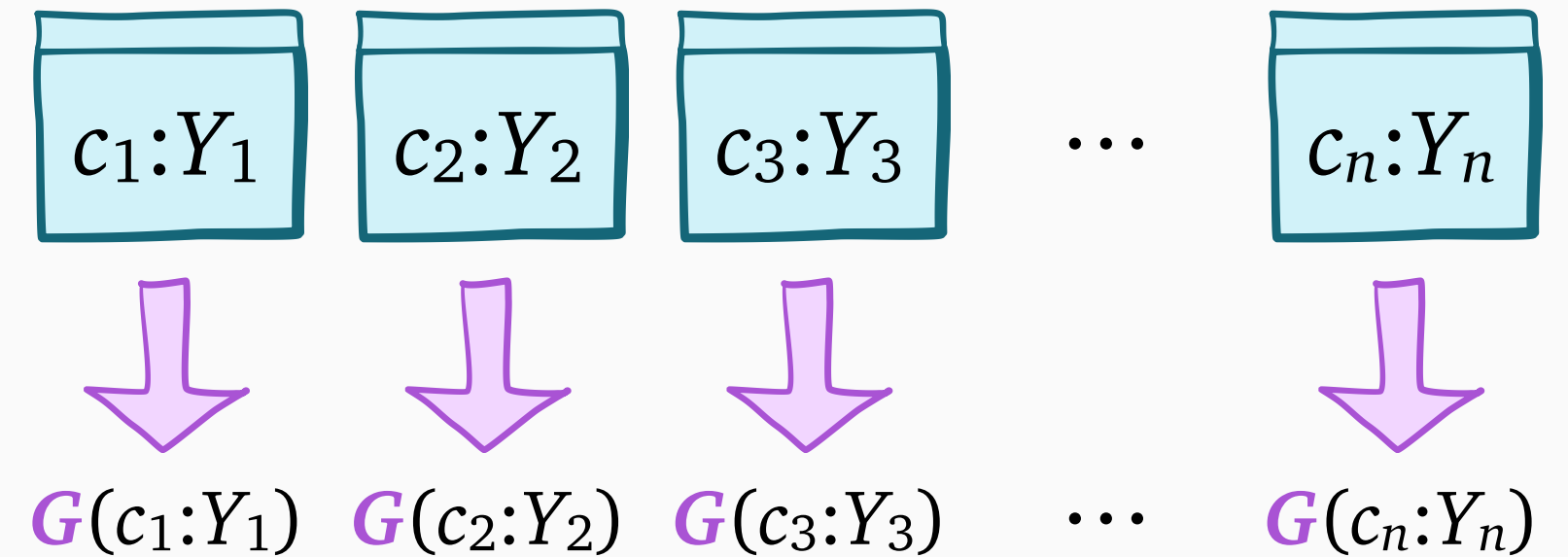
- *closed*: open it
- *open*: select it

Optimal policy: **Gittins**

Step 1: *rate* each box separately



Step 2: *act* on box of best rating

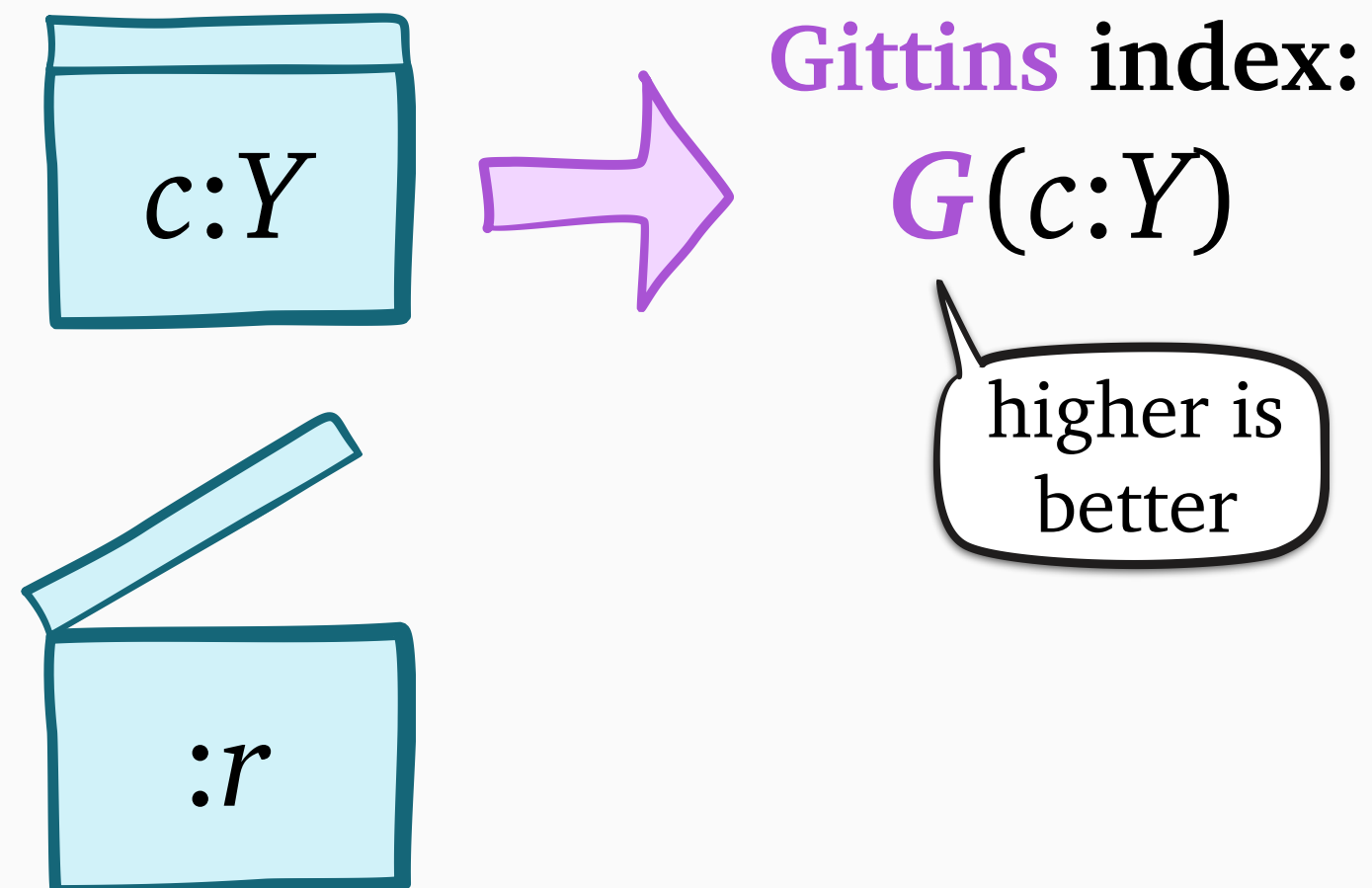


Gittins policy: if box of max **Gittins** index is...

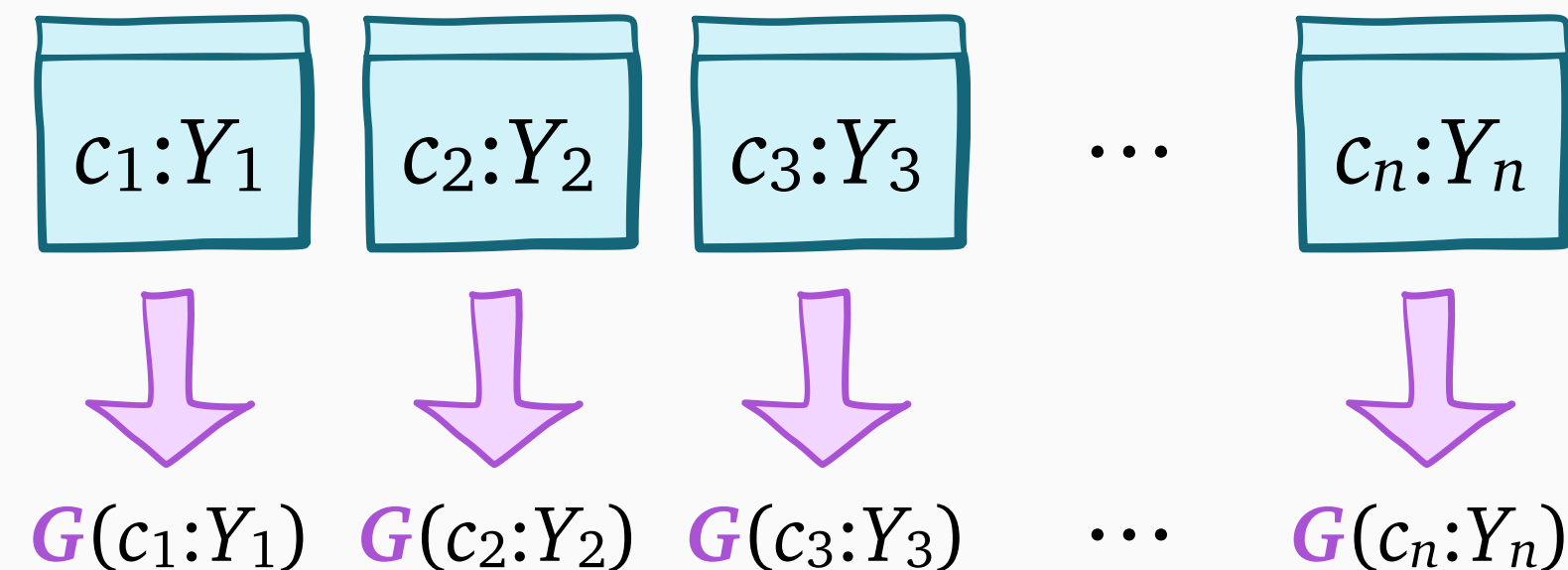
- *closed*: open it
 - *open*: select it
- } *act* on it

Optimal policy: **Gittins**

Step 1: *rate* each box separately



Step 2: *act* on box of best rating

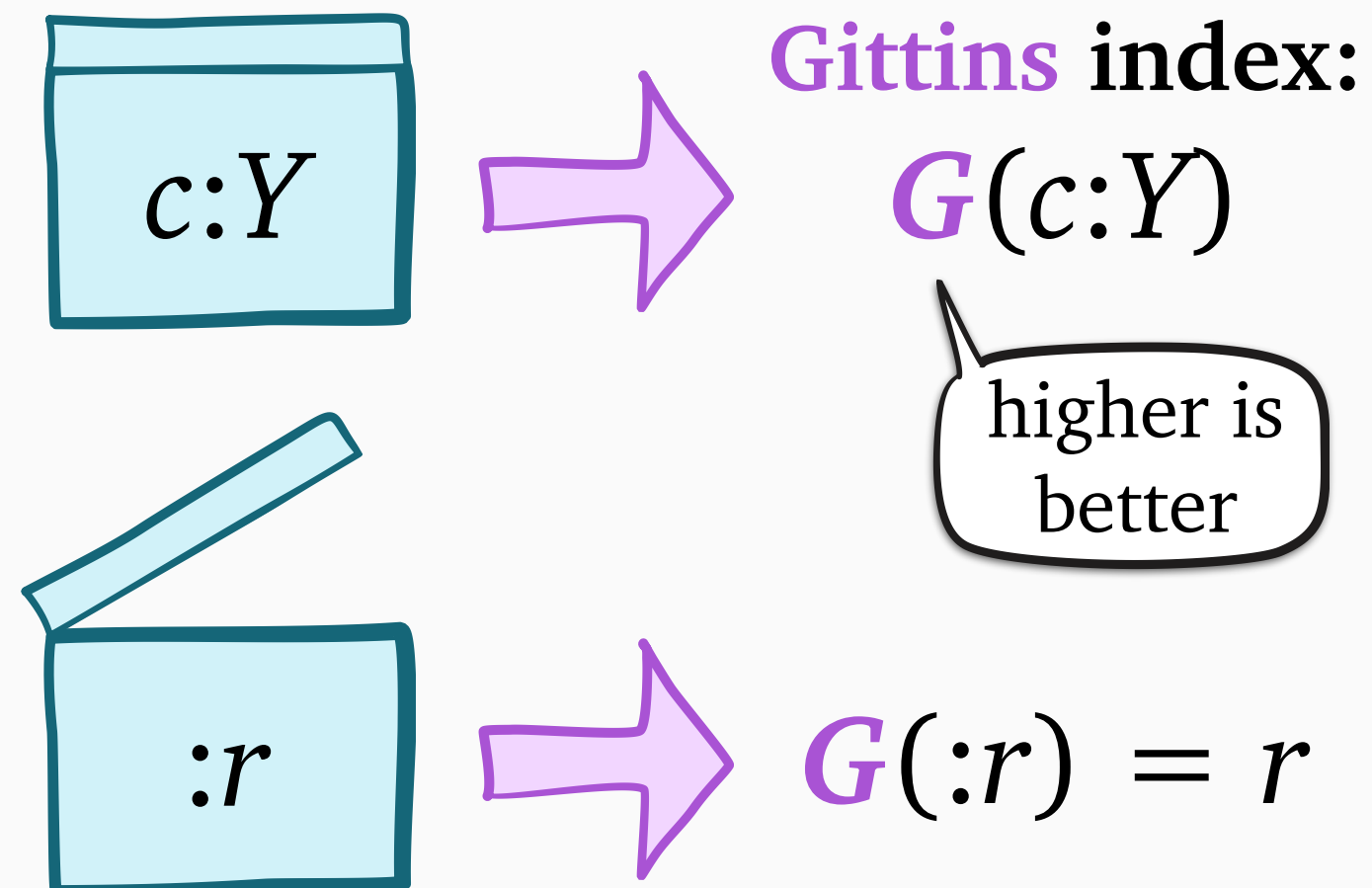


Gittins policy: if box of max **Gittins** index is...

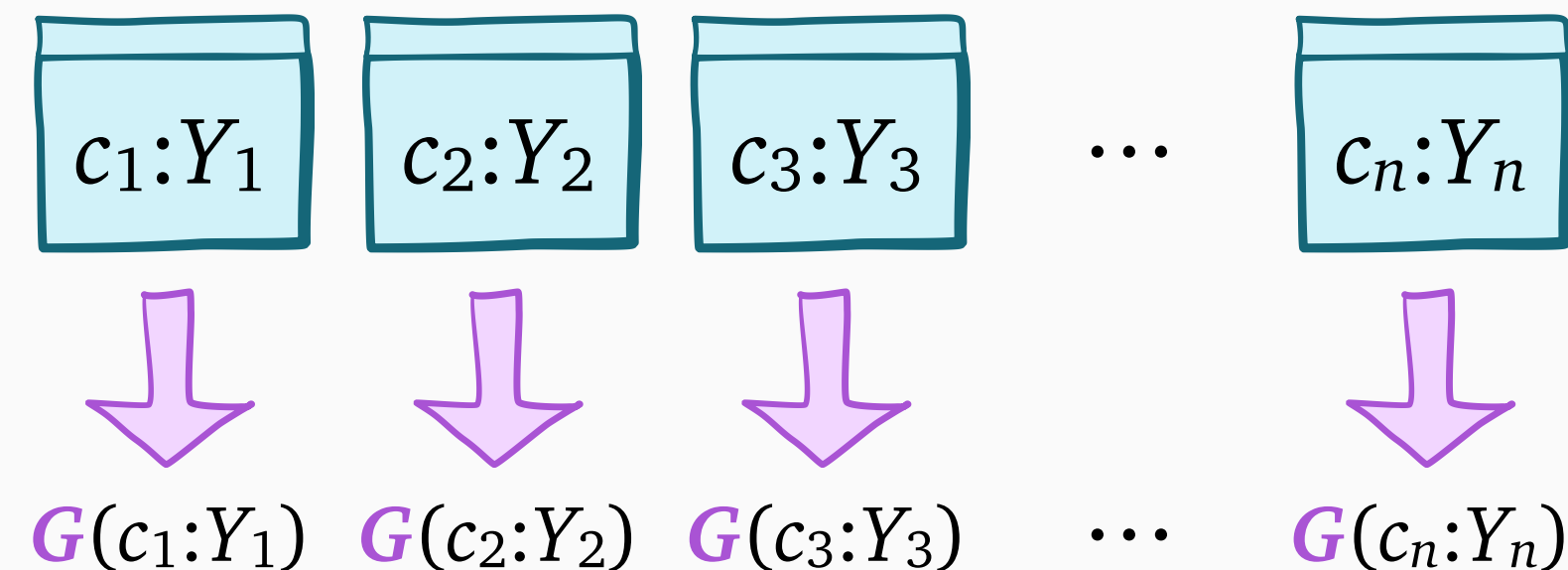
- *closed*: open it
 - *open*: select it
- } *act* on it

Optimal policy: **Gittins**

Step 1: *rate* each box separately



Step 2: *act* on box of best rating

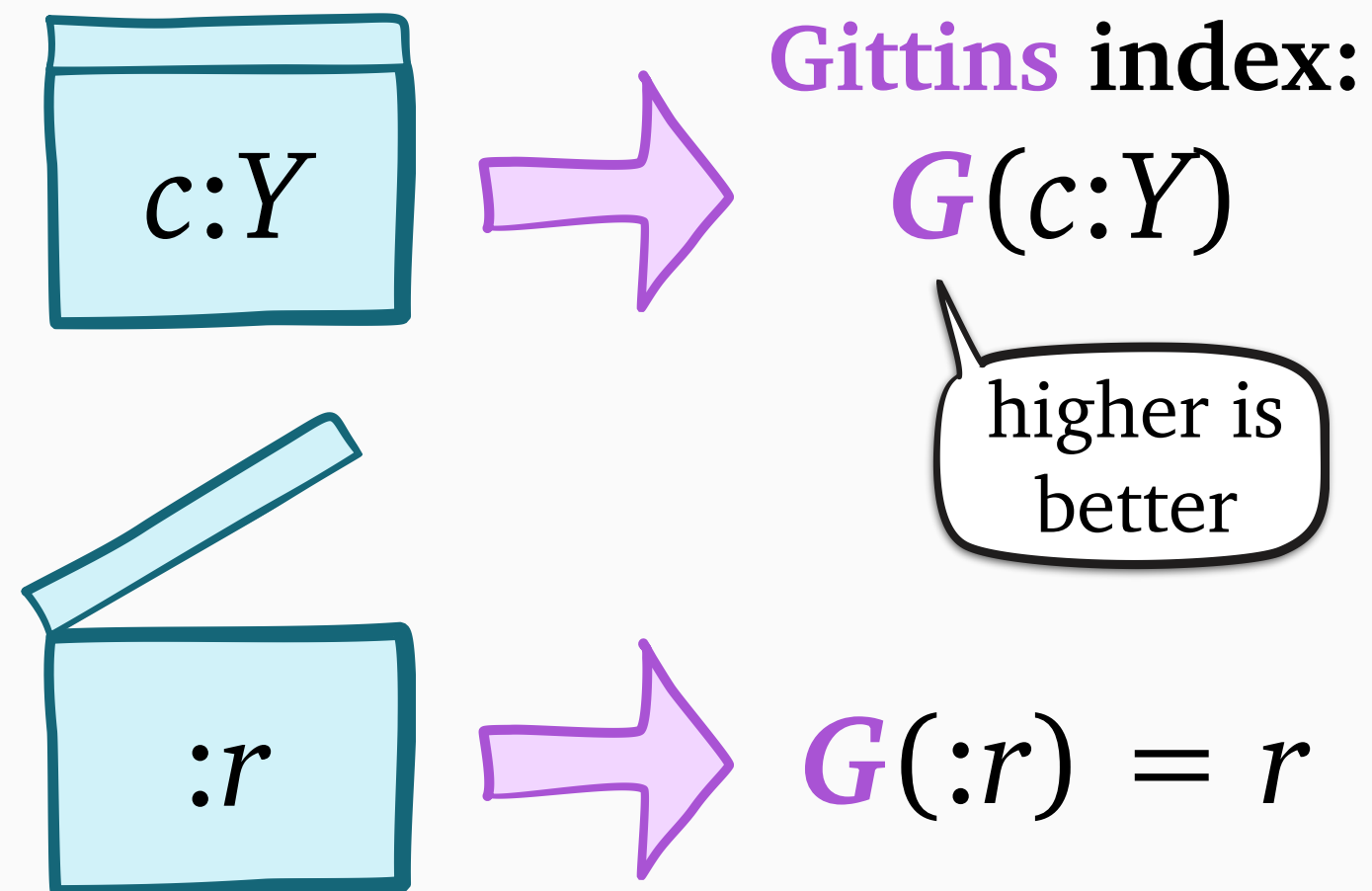


Gittins policy: if box of max **Gittins** index is...

- *closed*: open it
 - *open*: select it
- } *act* on it

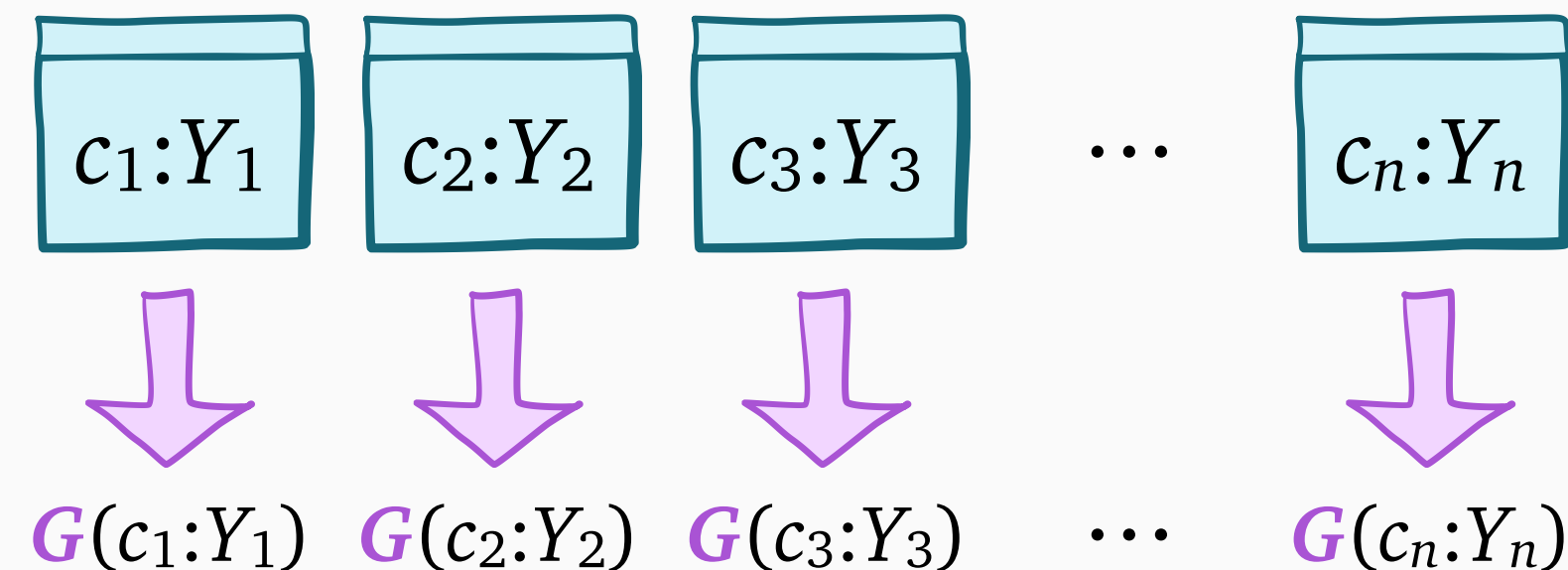
Optimal policy: **Gittins**

Step 1: *rate* each box separately



Theorem: [Weitzman, 1979]
the **Gittins** policy is optimal

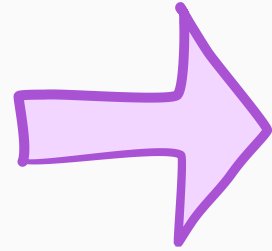
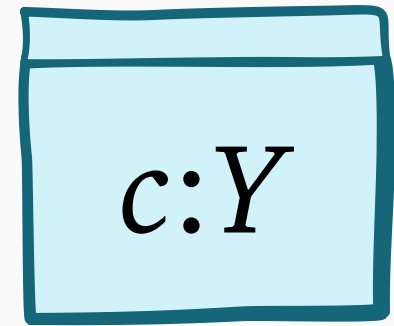
Step 2: *act* on box of best rating



Gittins policy: if box of max **Gittins** index is...

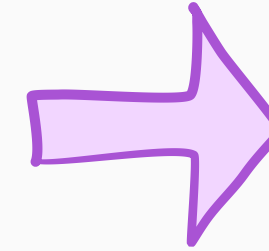
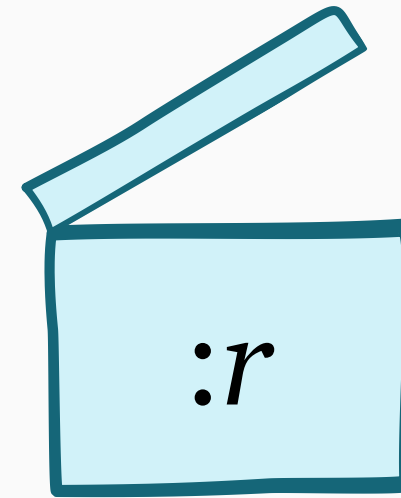
- *closed*: open it
 - *open*: select it
- } *act on it*

Gittins index of a box



Gittins index:

$$G(c:Y)$$



$$G(:r) = r$$

Gittins index of a box



? **Key question:** which is better *as a function of r* ?

Gittins index of a box



? **Key question:** which is better *as a function of r* ?

r small $\longleftrightarrow$ r large

Gittins index of a box



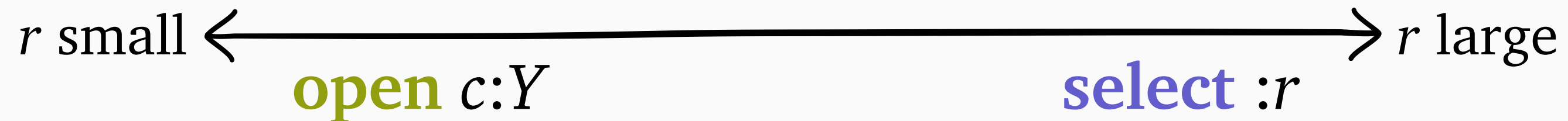
? **Key question:** which is better *as a function of r* ?

r small $\leftarrow$ $\xrightarrow{\text{select } :r}$ r large

Gittins index of a box



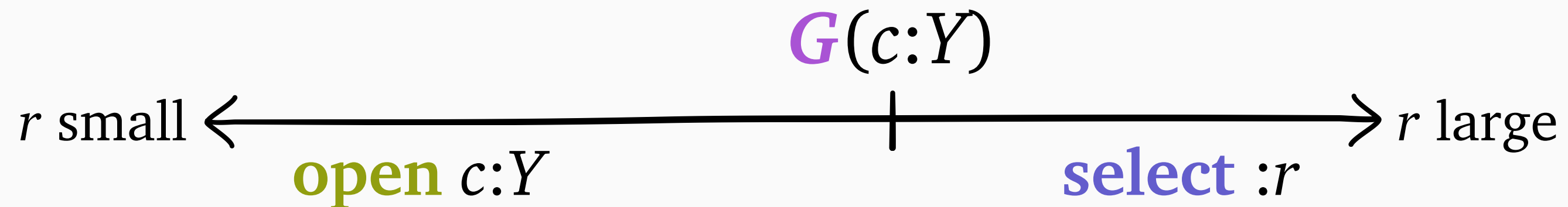
? **Key question:** which is better *as a function of r* ?



Gittins index of a box



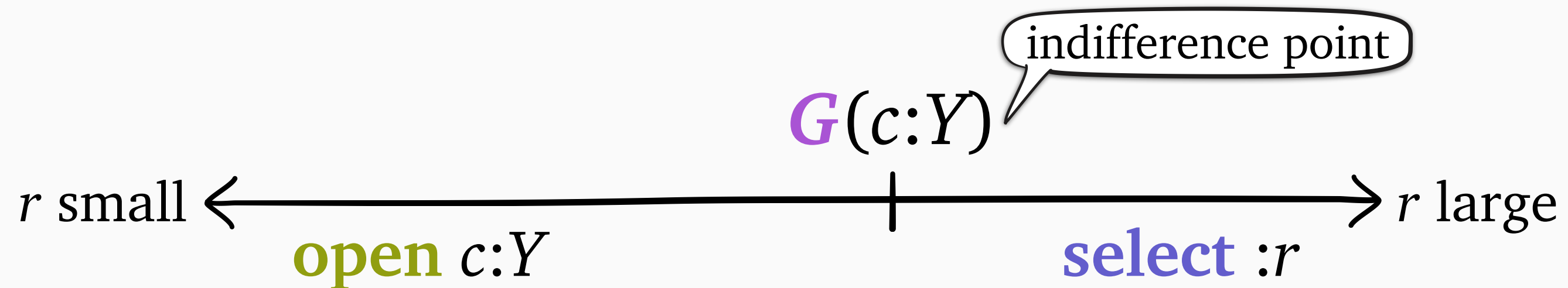
? **Key question:** which is better *as a function of r* ?



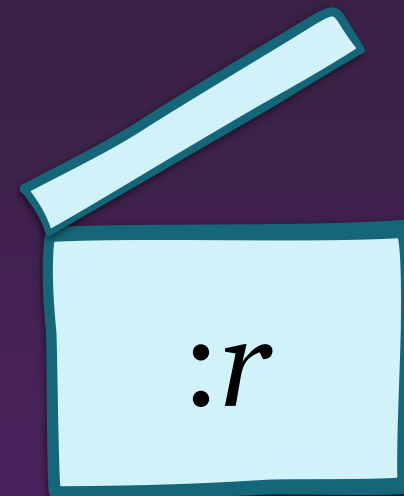
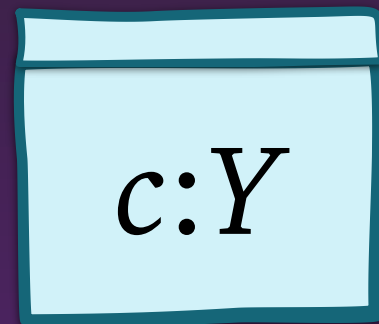
Gittins index of a box



? **Key question:** which is better *as a function of r* ?

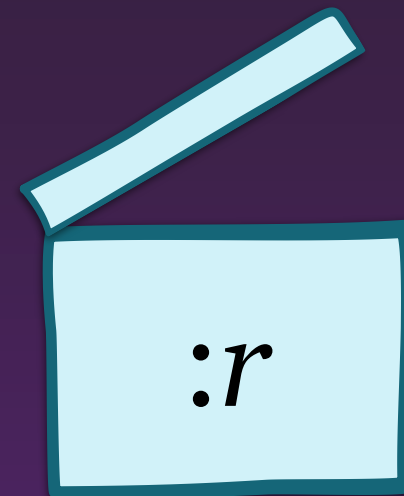
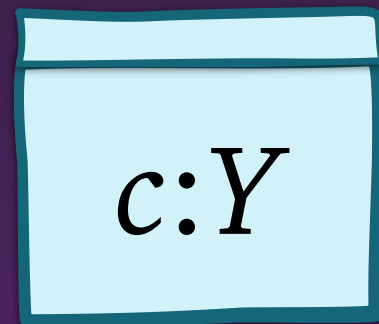


The *Gittins index* of $c:Y$ is the value of r such that we're indifferent between “open $c:Y$ ” and “select $:r$ ”



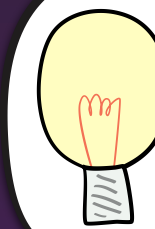
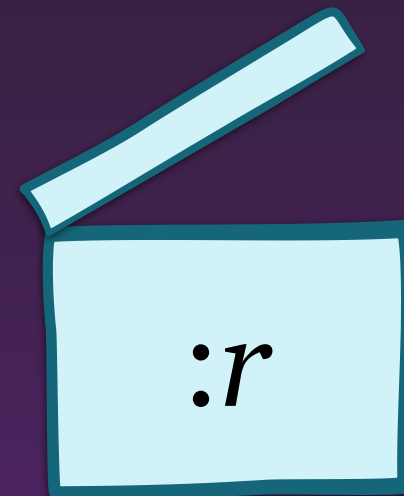
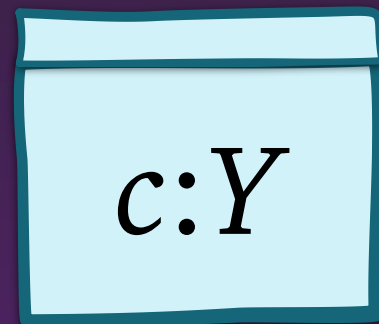
$G(c:Y)$

The *Gittins index* of $c:Y$ is the value of r such that we're indifferent between “open $c:Y$ ” and “select $:r$ ”



$G(c:Y)$

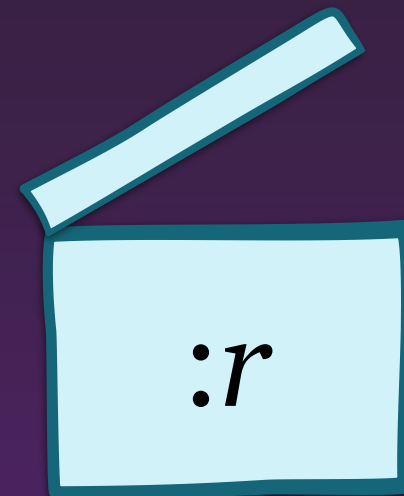
The *Gittins index* of $c:Y$ is the value of r such that we're indifferent between “open $c:Y$ ” and “select $:r$ ”



evaluate stochastic options by *calibrating to deterministic options*

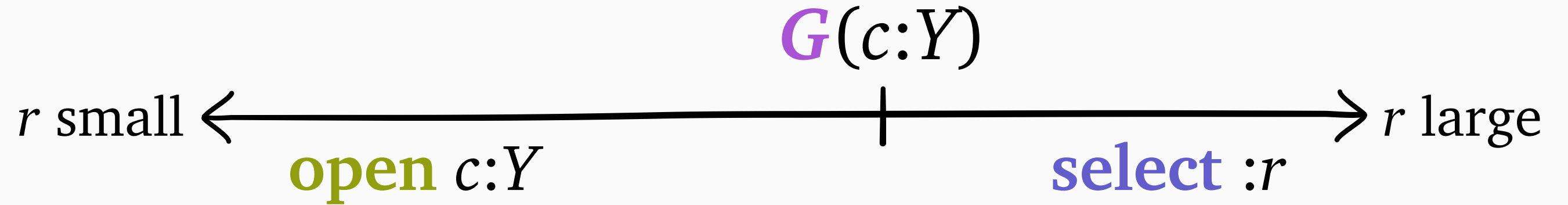
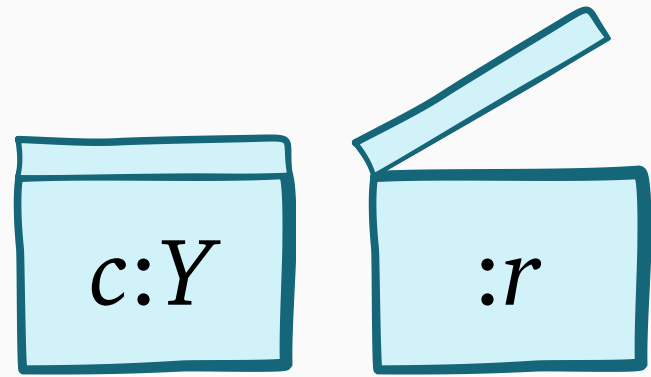
$G(???)$

The *Gittins index* of ???
is the value of r such that
we're indifferent between
“act on ???” and “select $:r$ ”

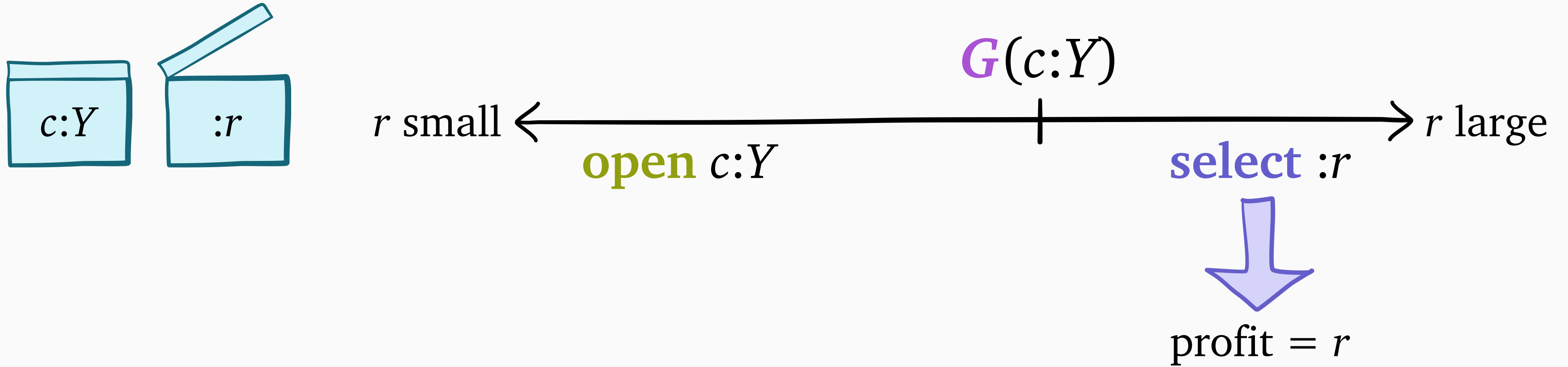


evaluate stochastic
options by *calibrating*
to *deterministic options*

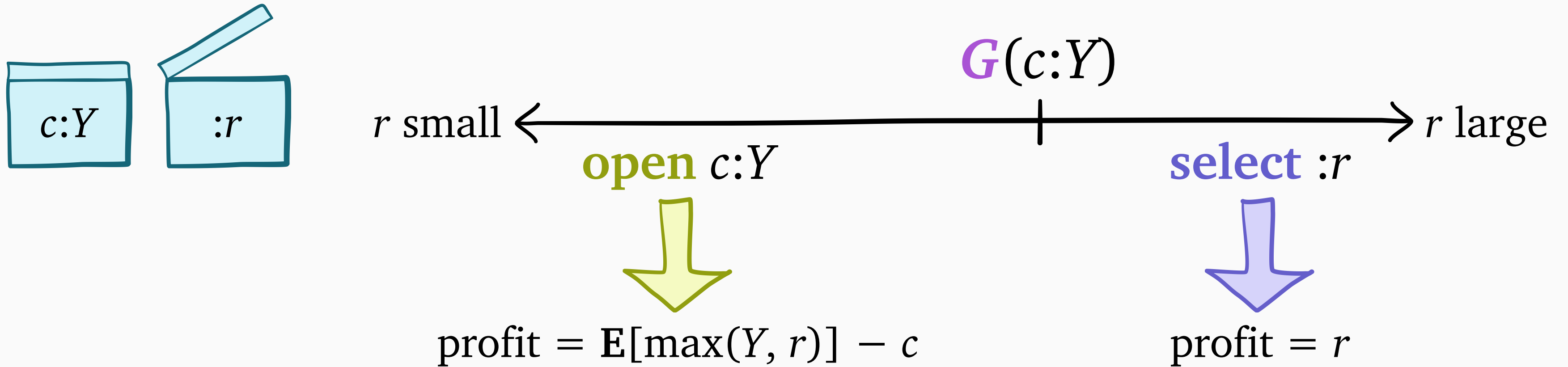
Formula for the Gittins index



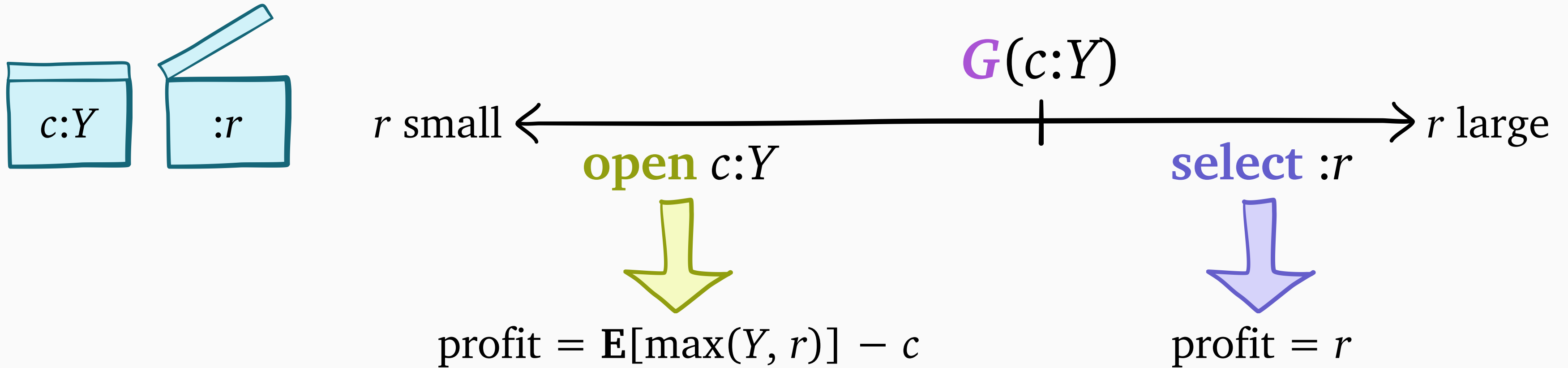
Formula for the Gittins index



Formula for the Gittins index



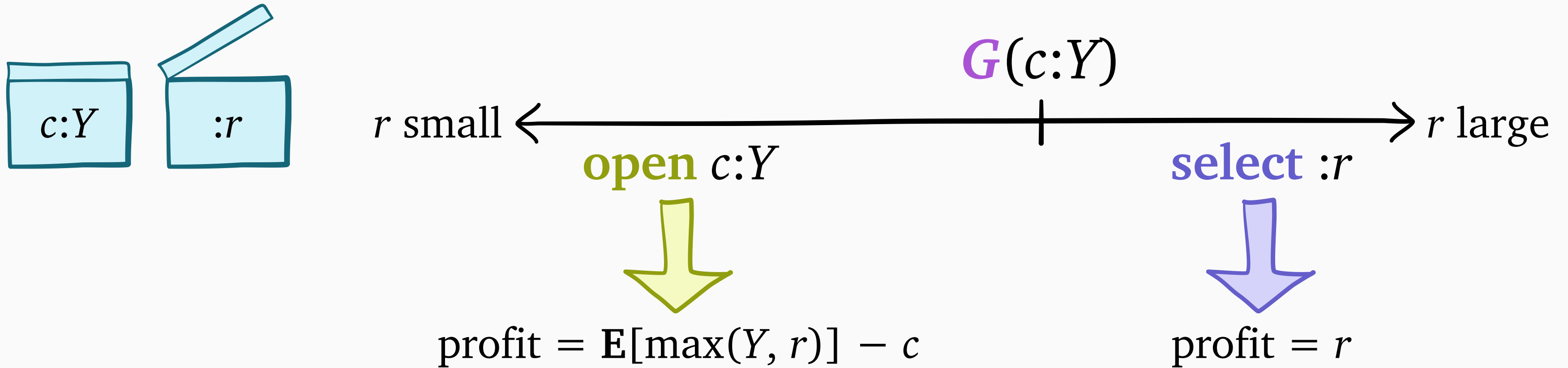
Formula for the Gittins index



Indifference point $r = G(c:Y)$

profit of **open** $c:Y$ = profit of select $:r$

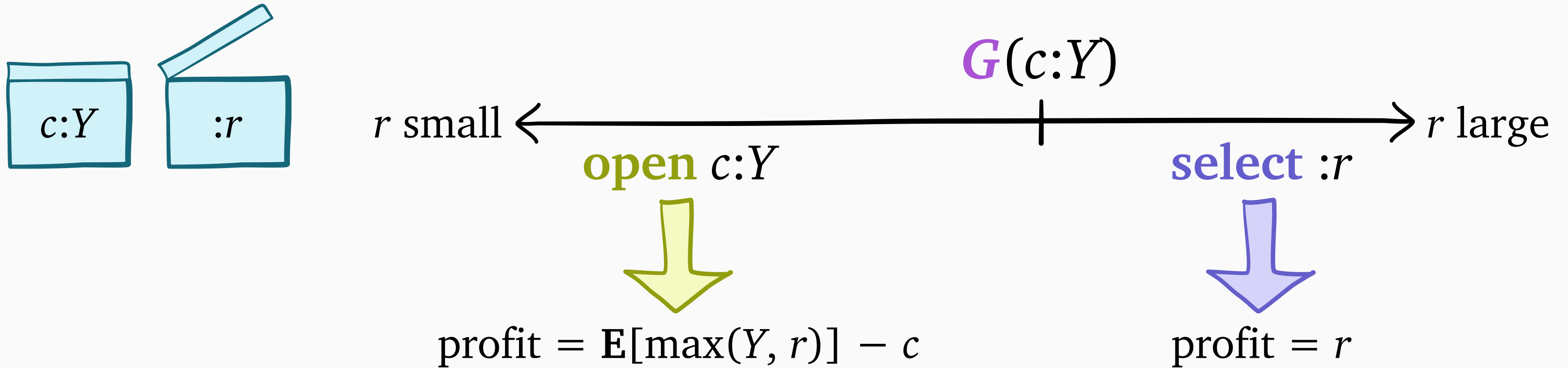
Formula for the Gittins index



Indifference point $r = G(c:Y)$

$$E[\max(Y, r)] - c = \text{profit of select } :r$$

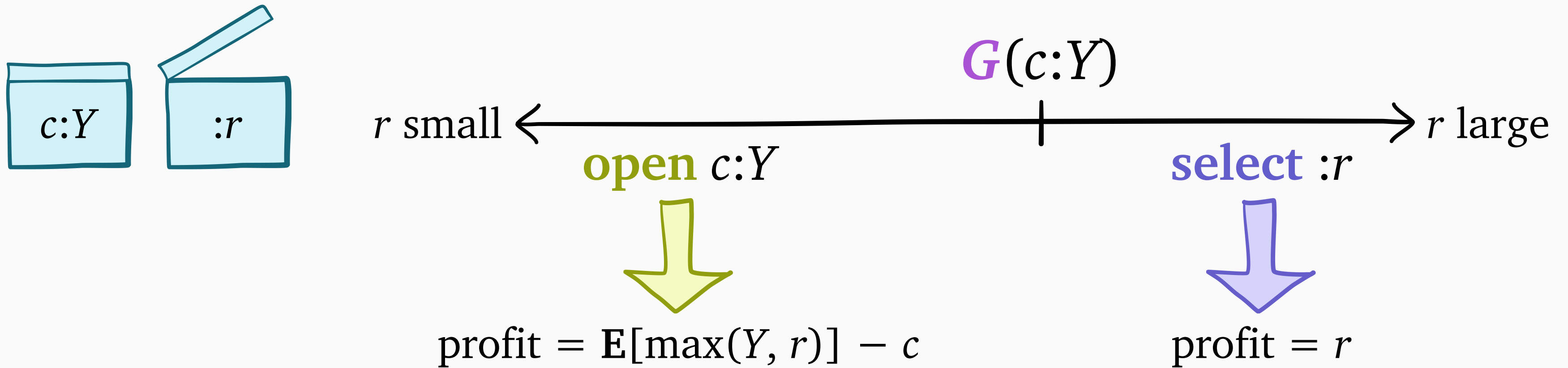
Formula for the **Gittins index**



Indifference point $r = G(c:Y)$

$$E[\max(Y, r)] - c = r$$

Formula for the Gittins index

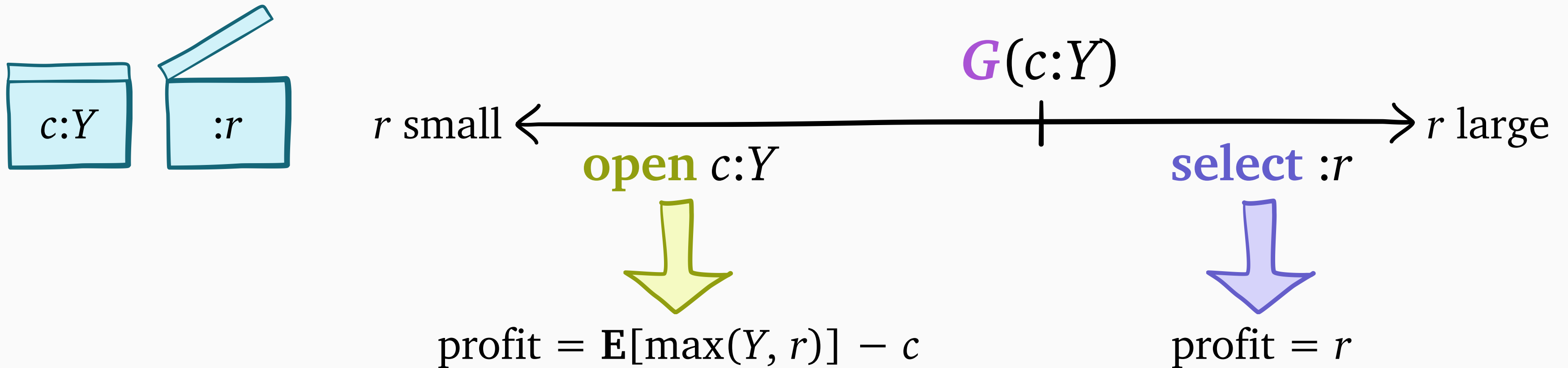


Indifference point $r = G(c:Y)$

$$E[\max(Y, r)] - c = r$$

$$\Rightarrow E[\max(Y - r, 0)] = c$$

Formula for the Gittins index



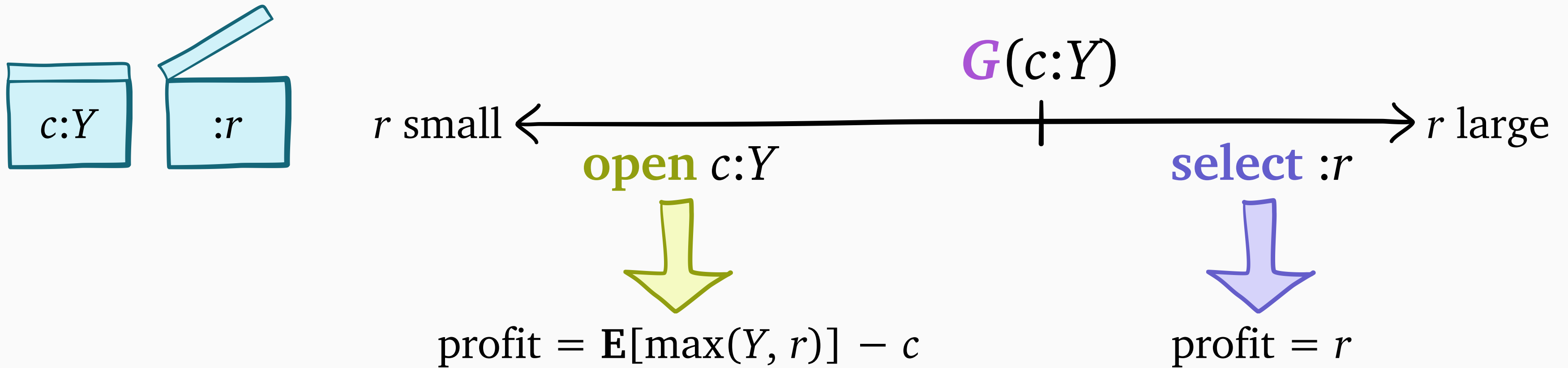
Indifference point $r = G(c:Y)$

$$E[\max(Y, r)] - c = r$$

$$\Rightarrow E[\max(Y - r, 0)] = c$$

EI(Y, r)

Formula for the **Gittins index**



Indifference point $r = G(c:Y)$

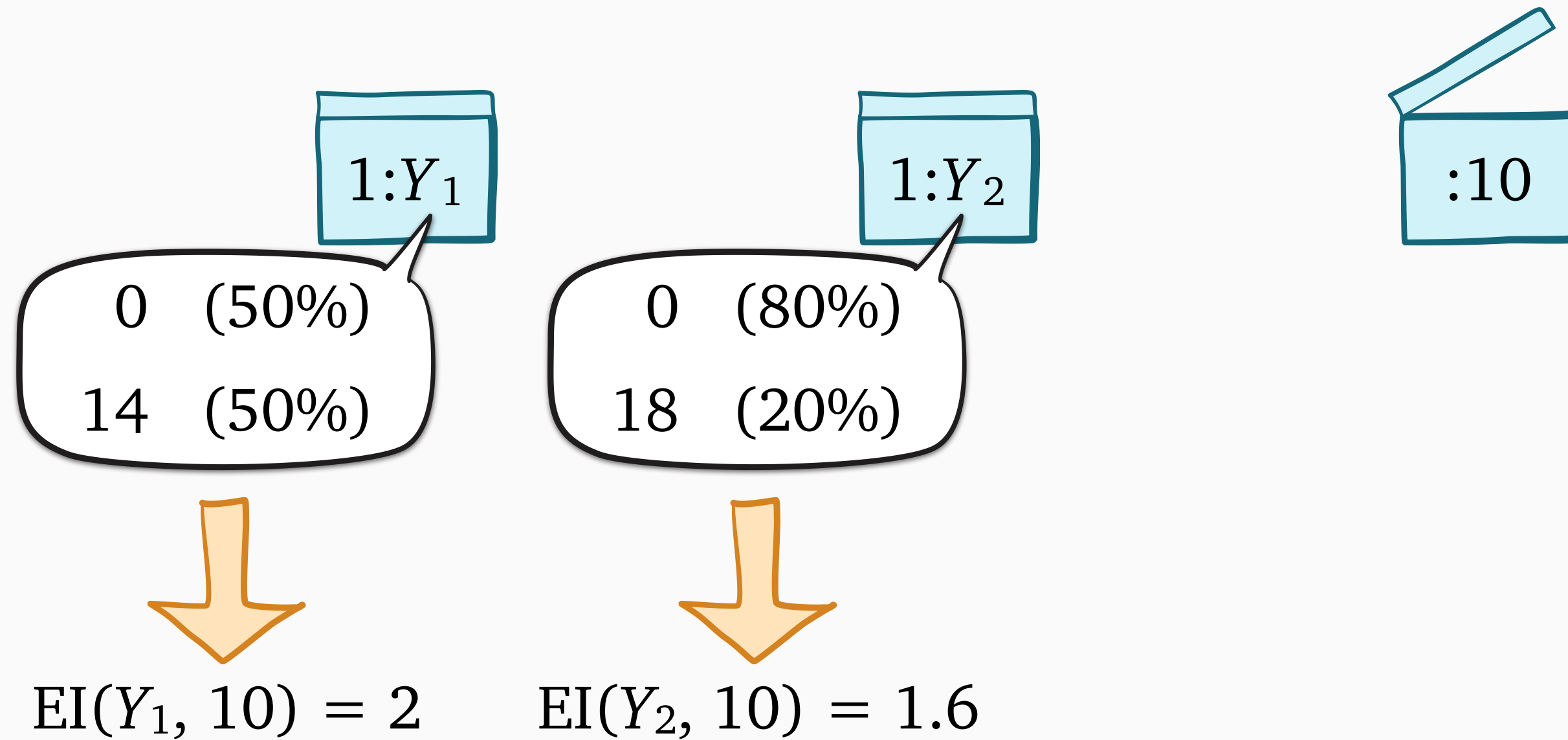
$$E[\max(Y, r)] - c = r$$
$$\Rightarrow E[\max(Y - r, 0)] = c$$

$EI(Y, r)$

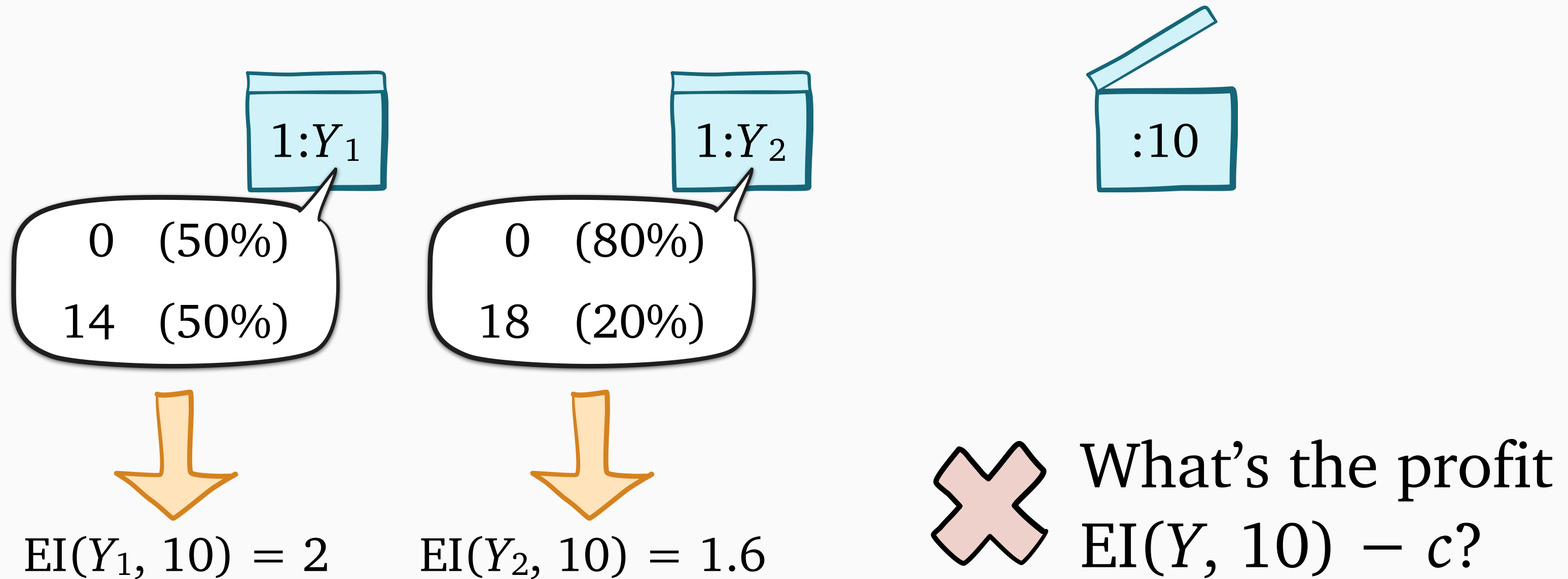
Definition: $G(c:Y)$ is solution r to

$$EI(Y, r) = c$$

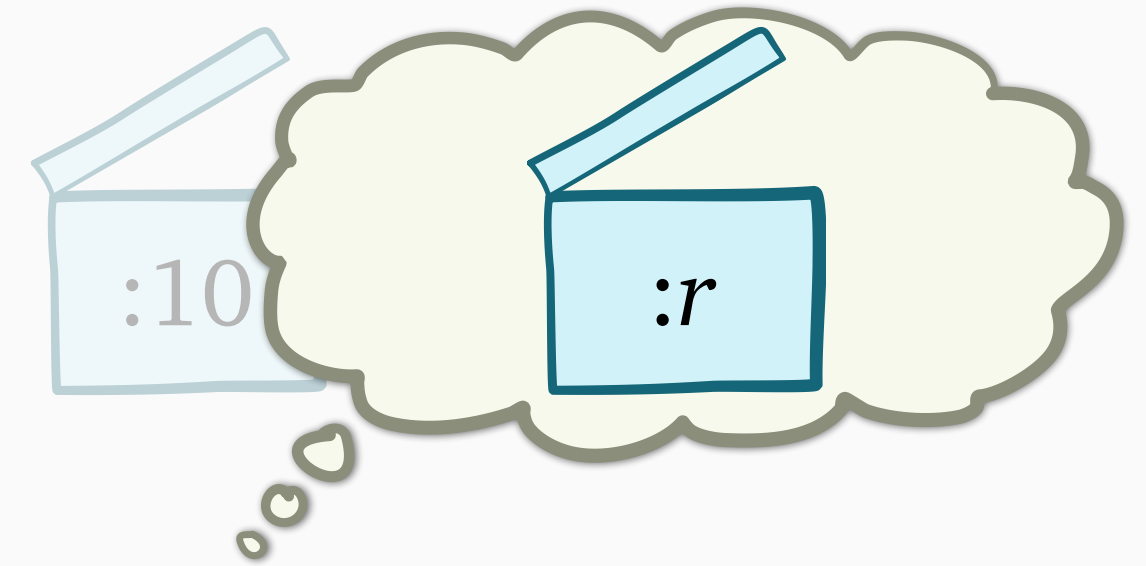
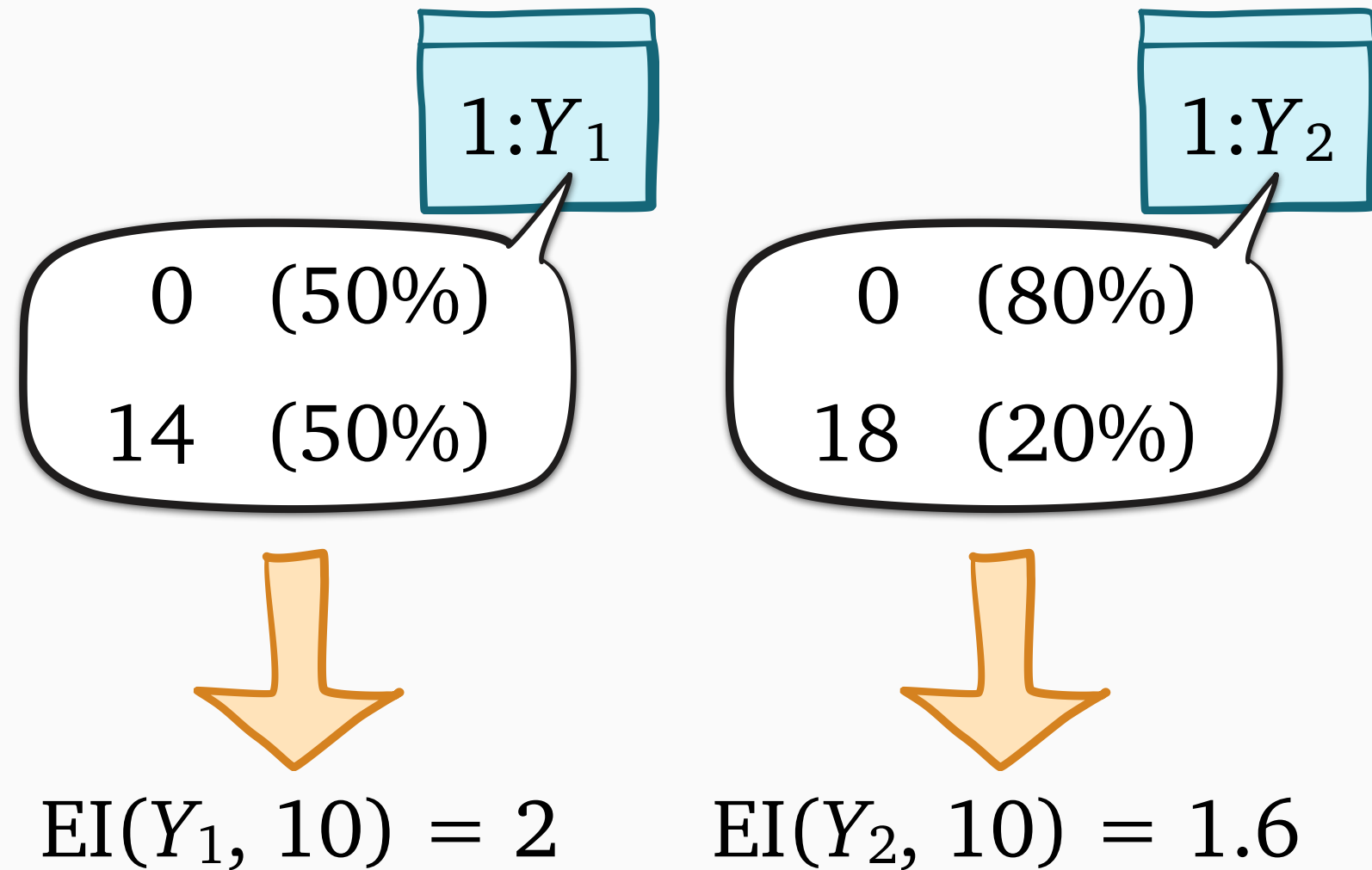
Expected improvement vs. **Gittins**



Expected improvement vs. **Gittins**



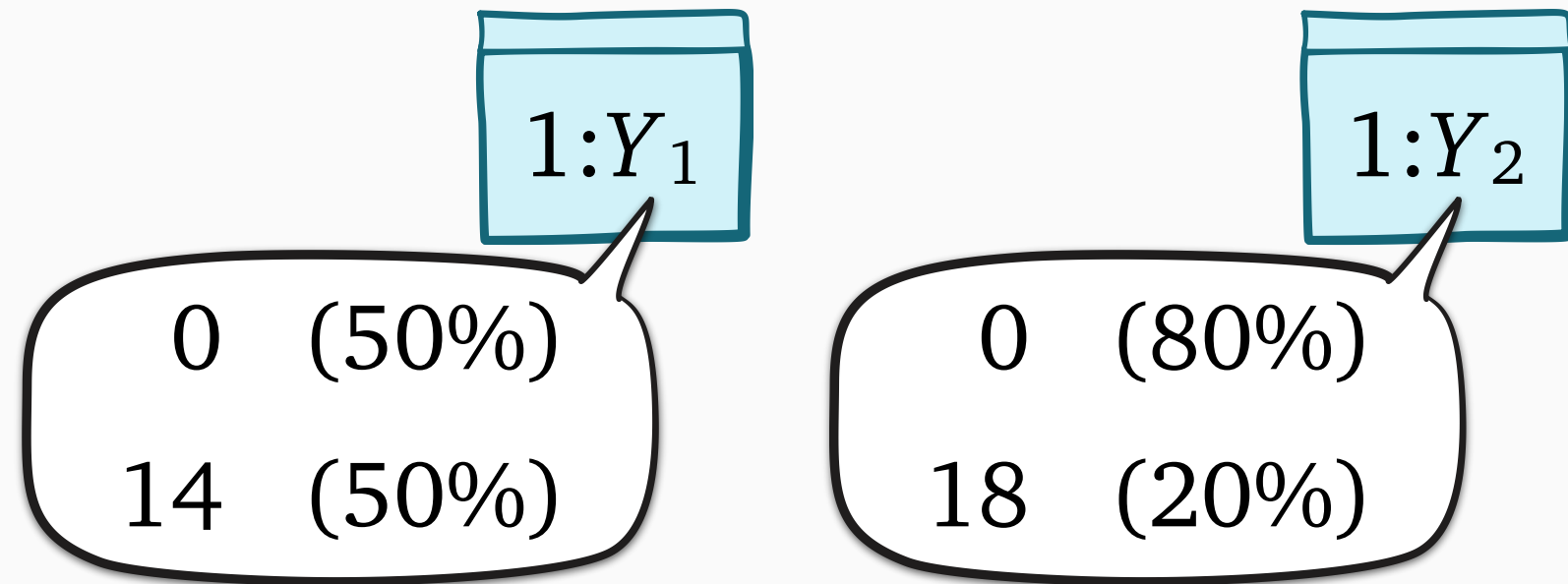
Expected improvement vs. **Gittins**



✗ What's the profit
 $EI(Y, 10) - c$?

✓ What r gives zero profit
 $EI(Y, r) - c = 0$?

Expected improvement vs. **Gittins**

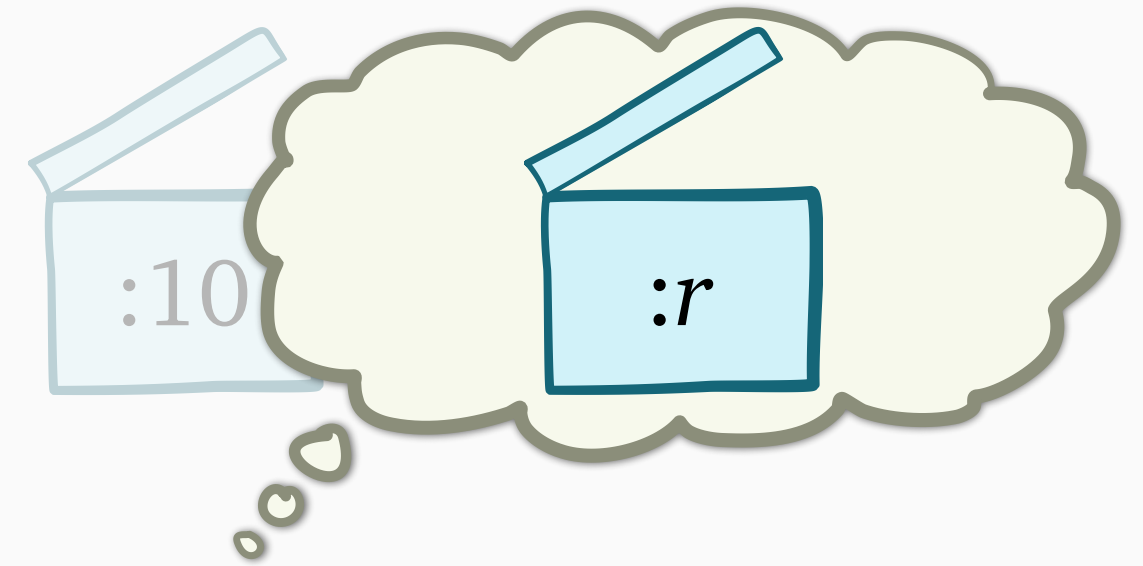


↓

$$EI(Y_1, 10) = 2$$
$$\textcolor{violet}{G}(1:Y_1) = 12$$

↓

$$EI(Y_2, 10) = 1.6$$



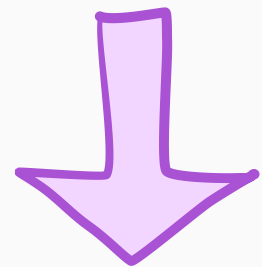
✗ What's the profit
 $EI(Y, 10) - c$?

✓ What r gives zero profit
 $EI(Y, r) - c = 0$?

Expected improvement vs. **Gittins**

1: Y_1

0	(50%)
14	(50%)



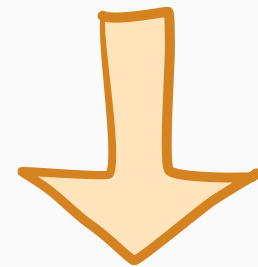
$$EI(Y_1, 10) = 2$$

$$\mathbf{G}(1:Y_1) = 12$$

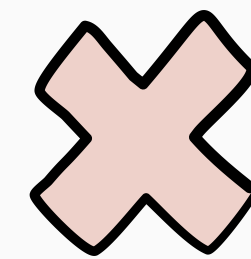
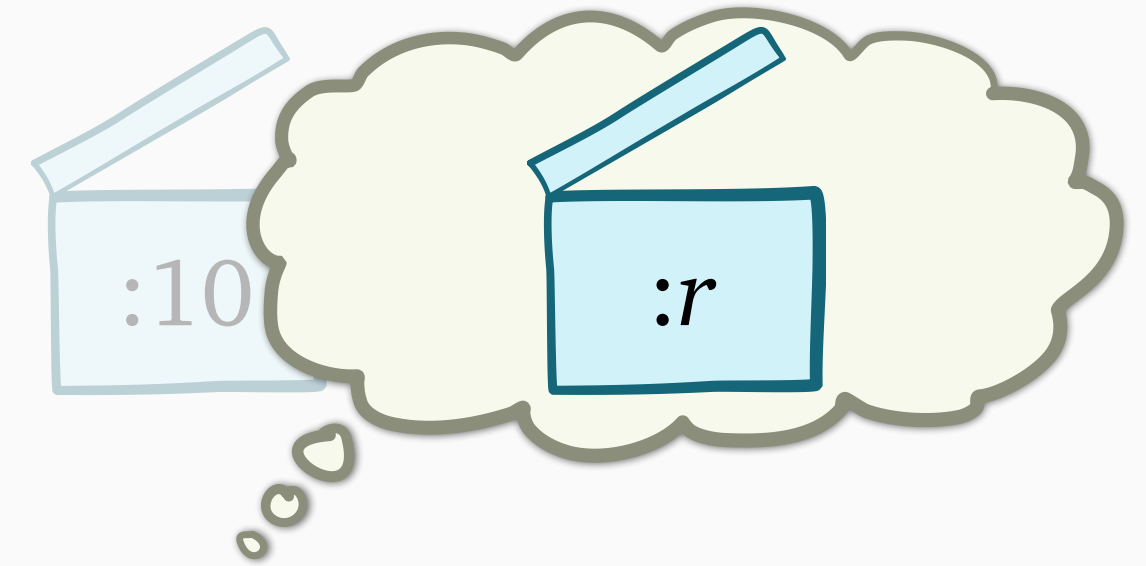
$$EI(Y_1, 12) = 1$$

1: Y_2

0	(80%)
18	(20%)



$$EI(Y_2, 10) = 1.6$$



What's the profit
 $EI(Y, 10) - c$?

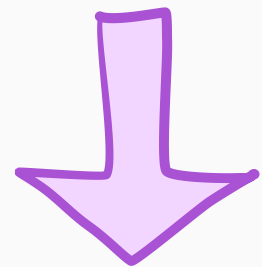


What r gives zero profit
 $EI(Y, r) - c = 0$?

Expected improvement vs. **Gittins**

1: Y_1

0	(50%)
14	(50%)



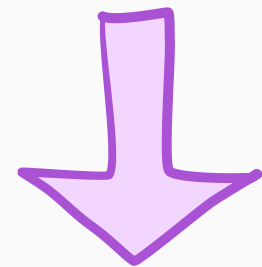
$$\text{EI}(Y_1, 10) = 2$$

$$\mathbf{G}(1:Y_1) = 12$$

$$\text{EI}(Y_1, 12) = 1$$

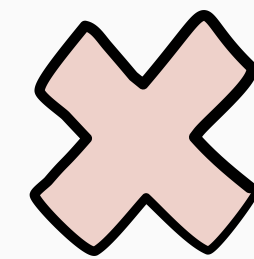
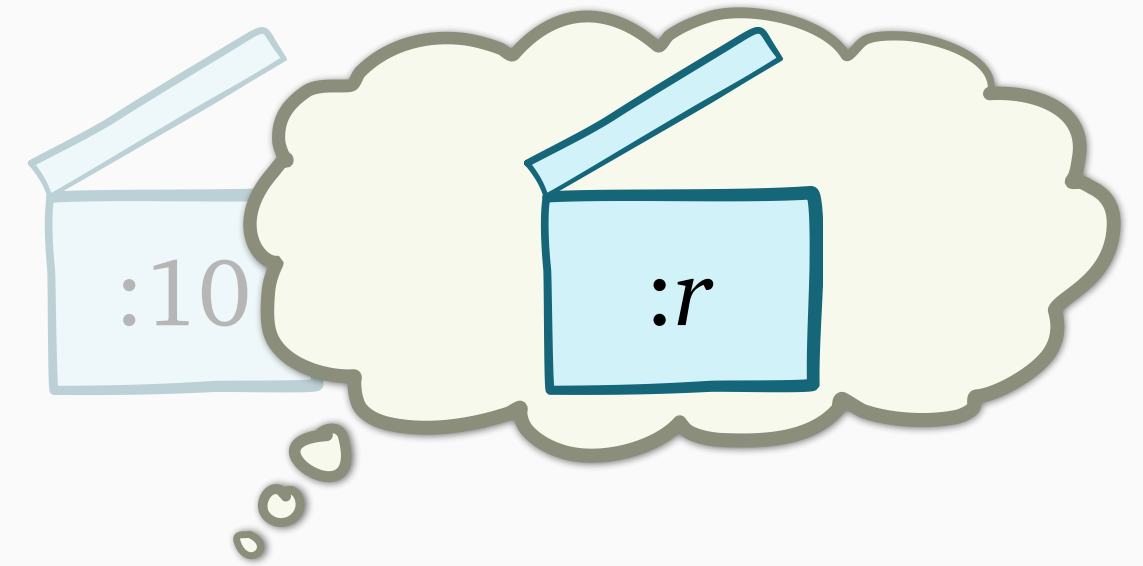
1: Y_2

0	(80%)
18	(20%)



$$\text{EI}(Y_2, 10) = 1.6$$

$$\mathbf{G}(1:Y_2) = 13$$



What's the profit
 $\text{EI}(Y, 10) - c$?

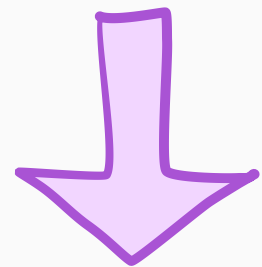


What r gives zero profit
 $\text{EI}(Y, r) - c = 0$?

Expected improvement vs. **Gittins**

1: Y_1

0	(50%)
14	(50%)



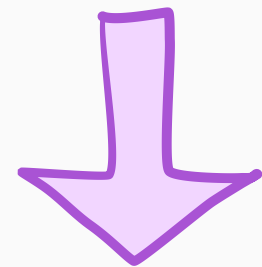
$$EI(Y_1, 10) = 2$$

$$G(1:Y_1) = 12$$

$$EI(Y_1, 12) = 1$$

1: Y_2

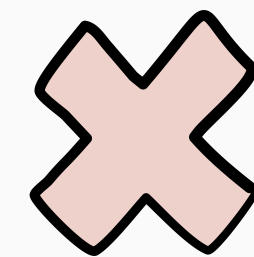
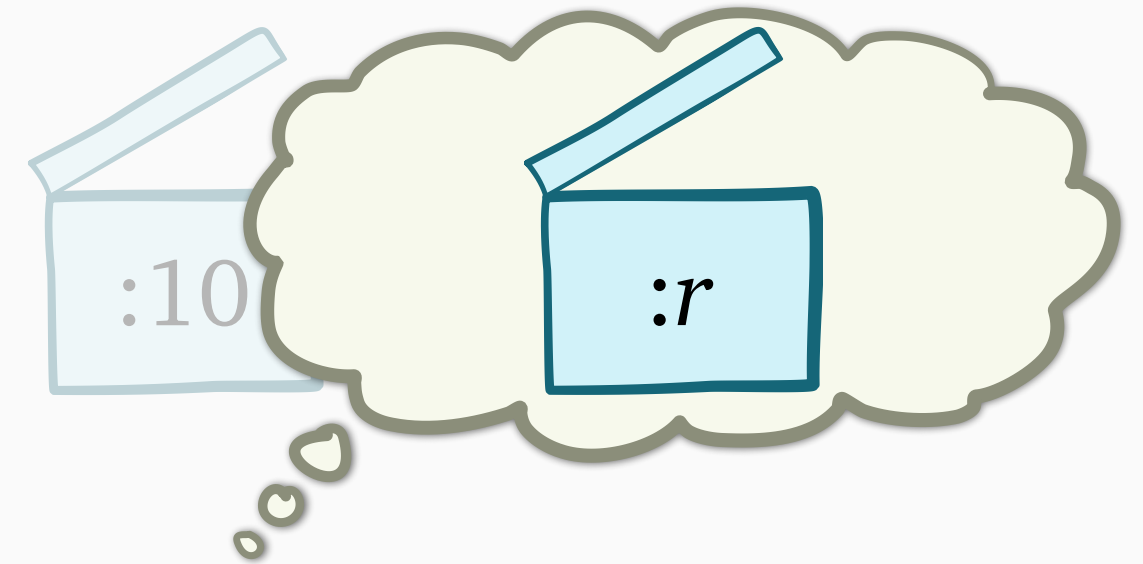
0	(80%)
18	(20%)



$$EI(Y_2, 10) = 1.6$$

$$G(1:Y_2) = 13$$

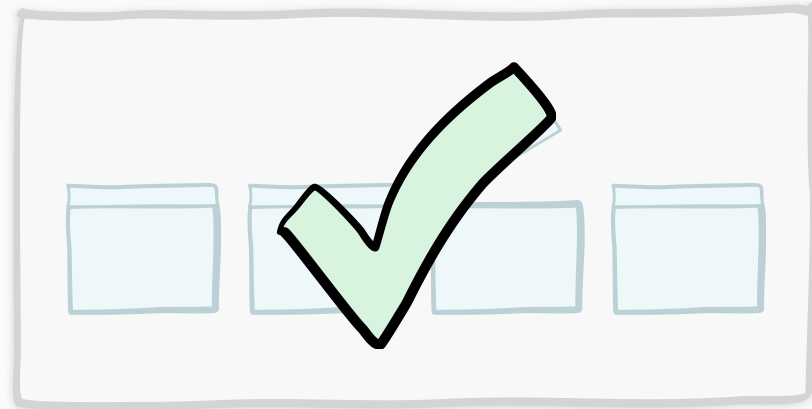
$$EI(Y_2, 13) = 1$$



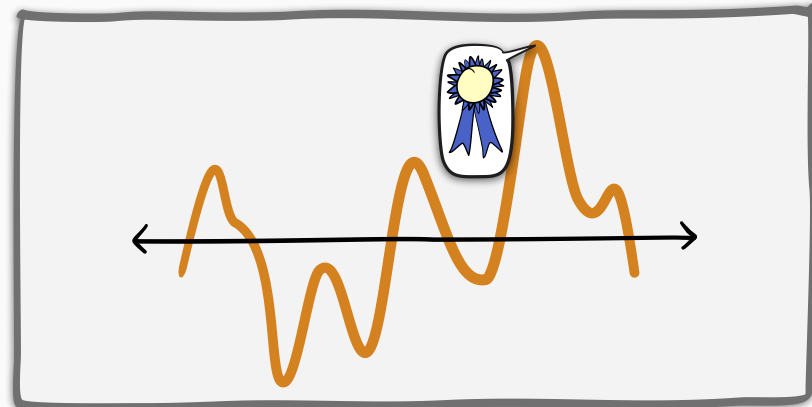
What's the profit
 $EI(Y, 10) - c$?



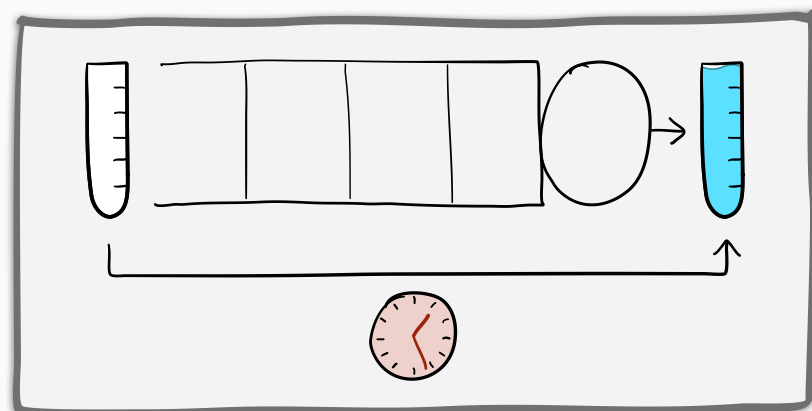
What r gives zero profit
 $EI(Y, r) - c = 0$?



Part I:
Introducing **Gittins** via **Pandora's box**

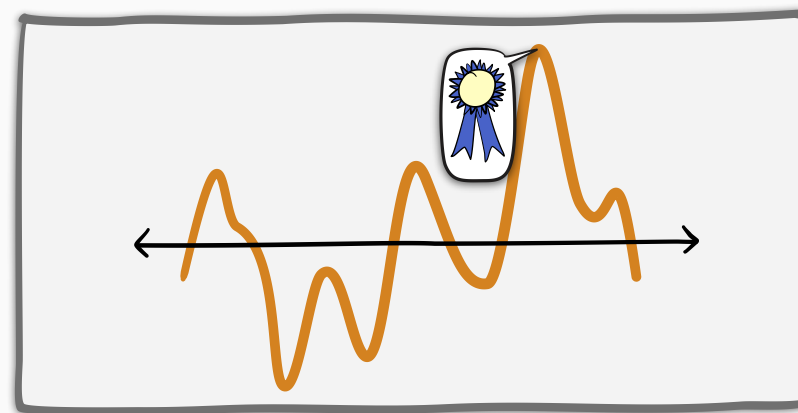


Part II: **NEW!**
Gittins for **Bayesian optimization**



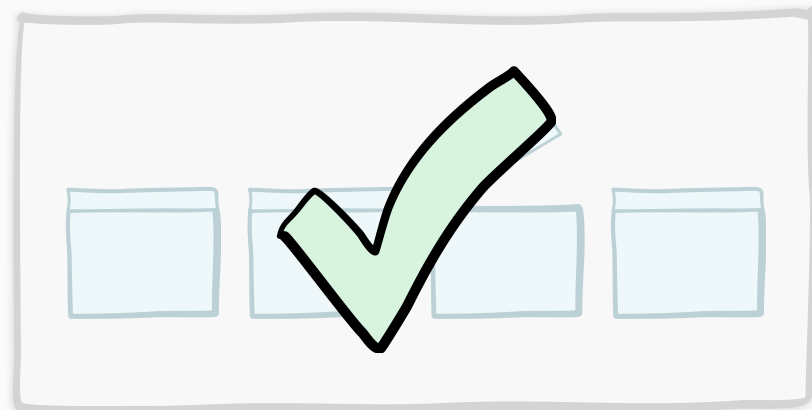
Part III: **NEW!**
Gittins for **tail scheduling**

Part II: Gittins for Bayesian optimization

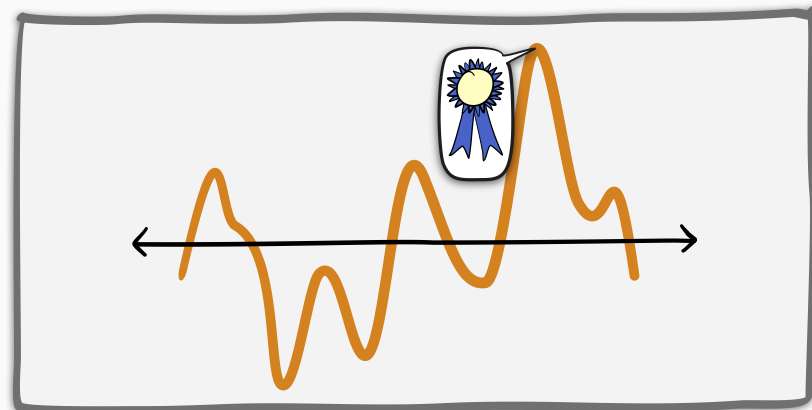


Click here for slides

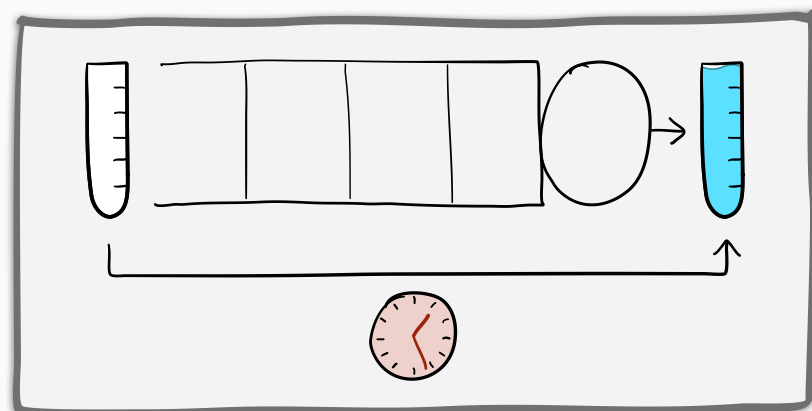
<https://presentations.avt.im/2025-10-27-Gittins-Index-Tutorial/>



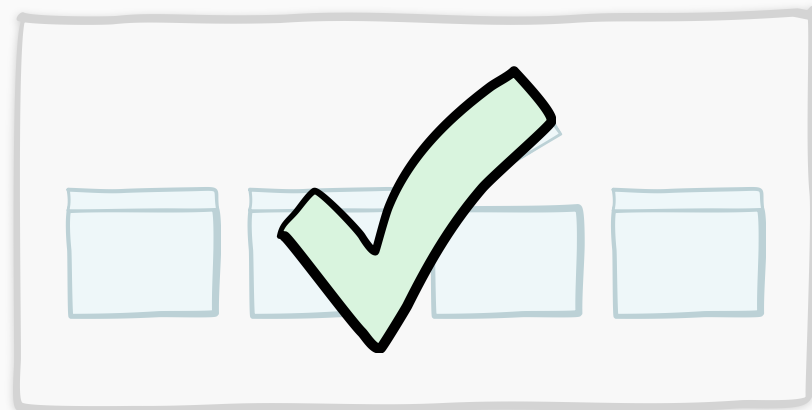
Part I:
Introducing **Gittins** via **Pandora's box**



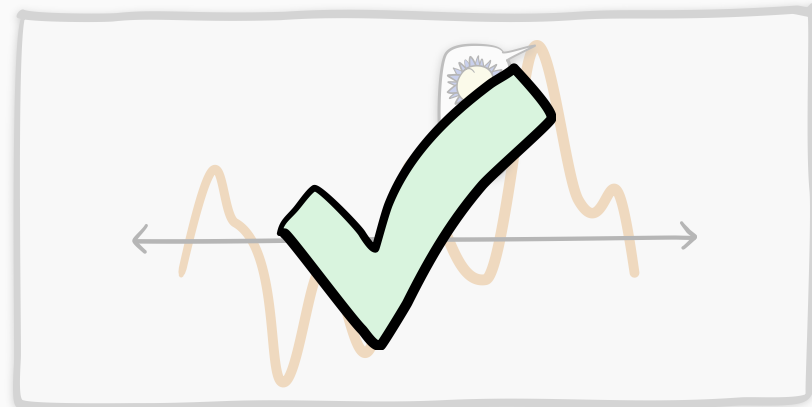
Part II: **NEW!**
Gittins for **Bayesian optimization**



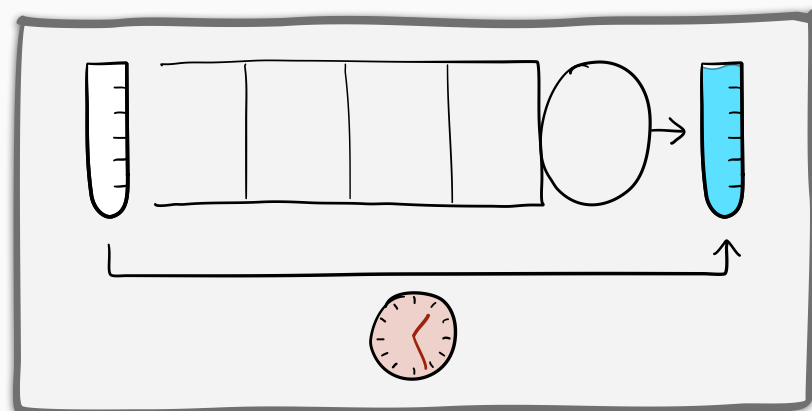
Part III: **NEW!**
Gittins for **tail scheduling** and more



Part I:
Introducing **Gittins** via **Pandora's box**

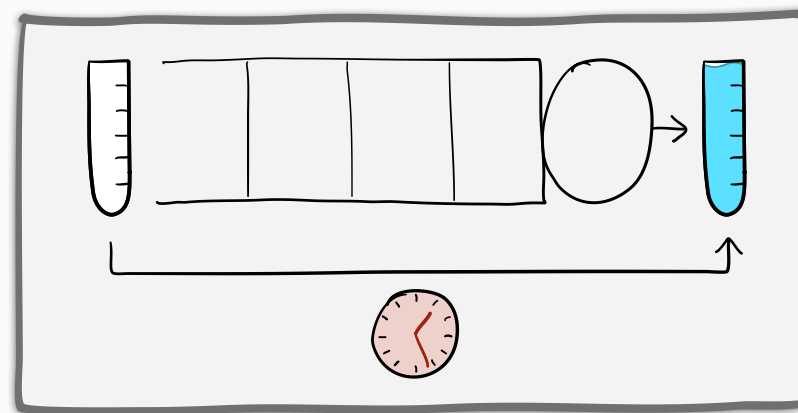


Part II: **NEW!**
Gittins for **Bayesian optimization**

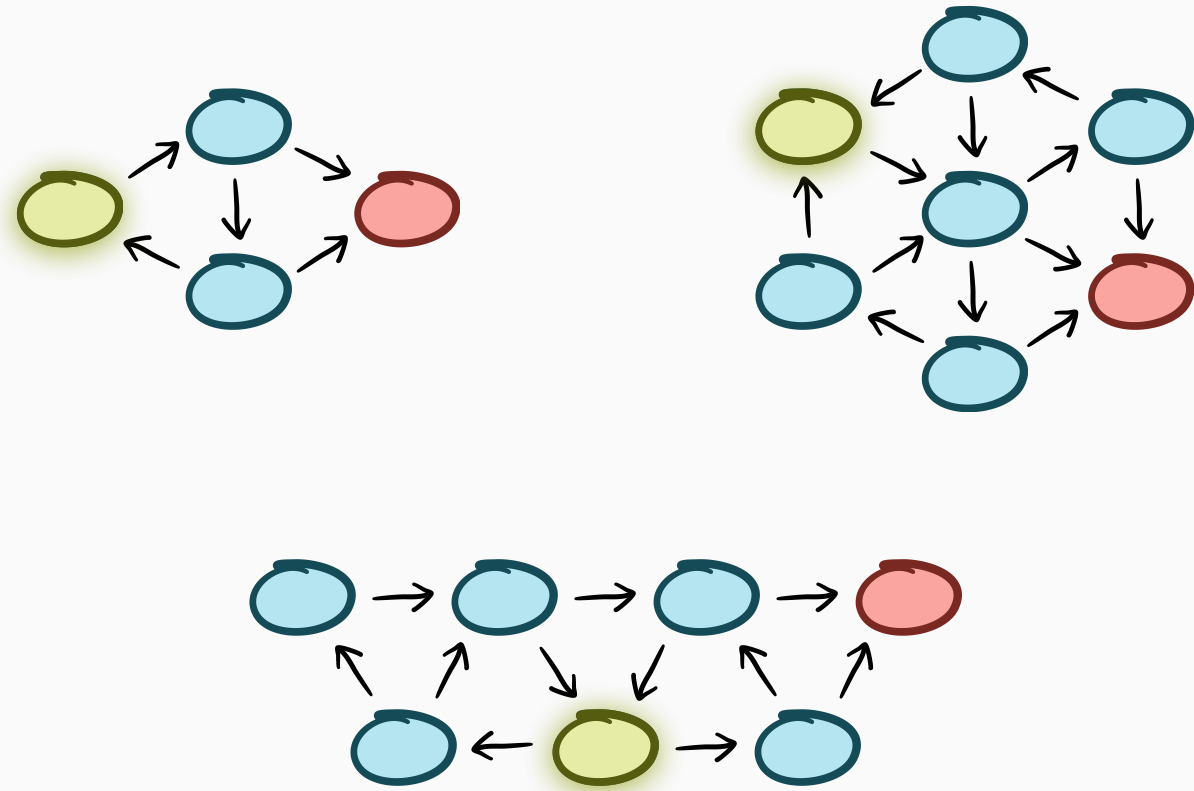


Part III: **NEW!**
Gittins for **tail scheduling** and more

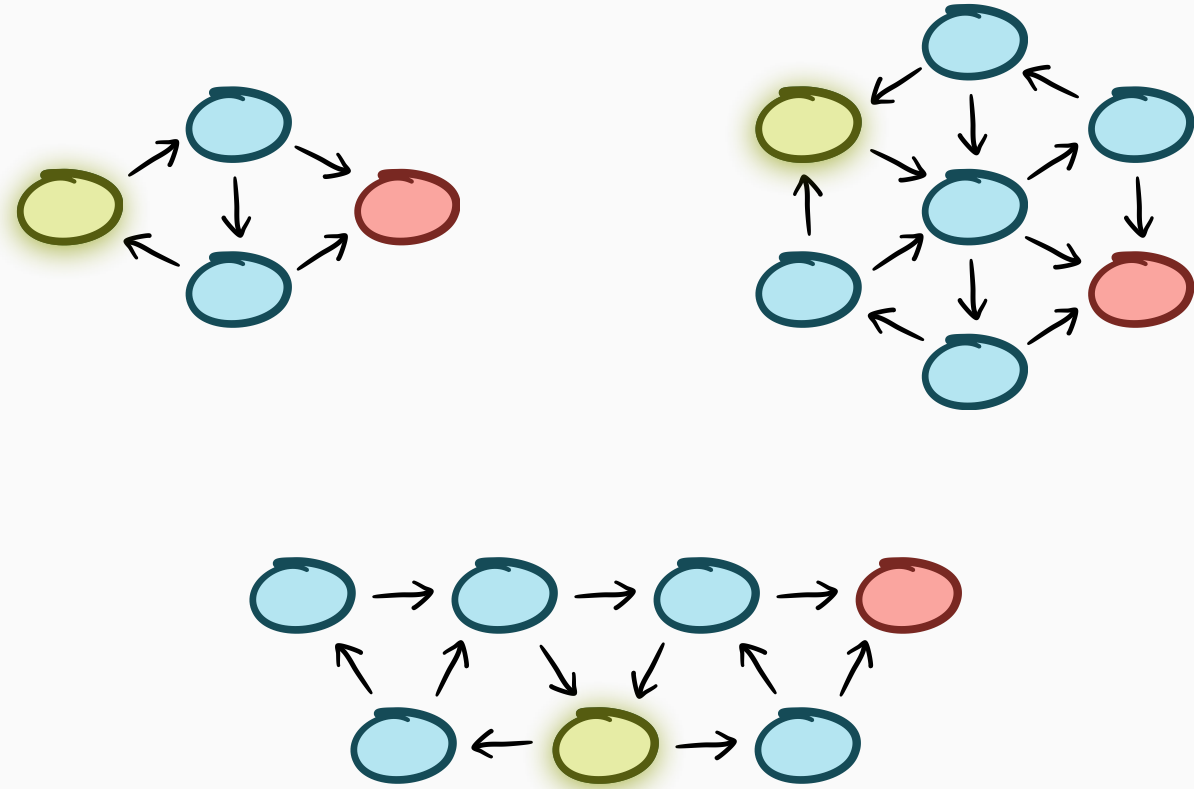
Part III: Gittins for tail scheduling



Markov chain selection (MCS)



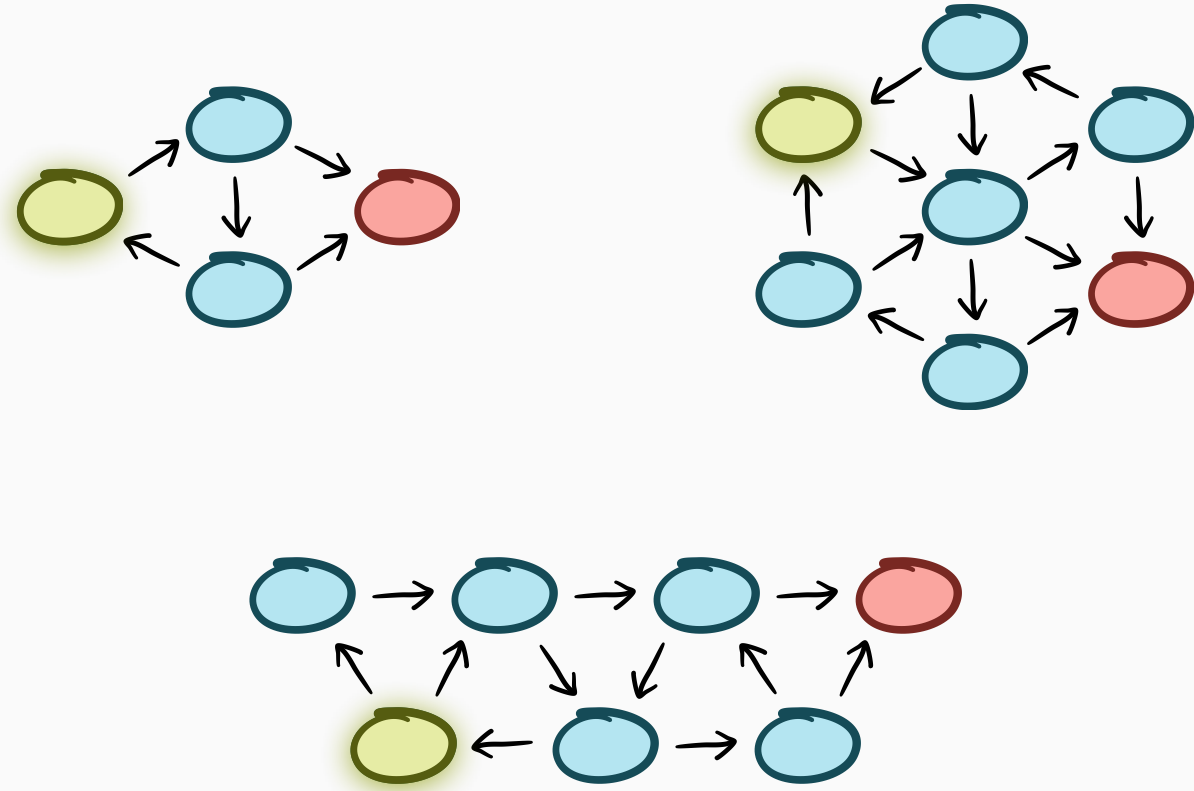
Markov chain selection (MCS)



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

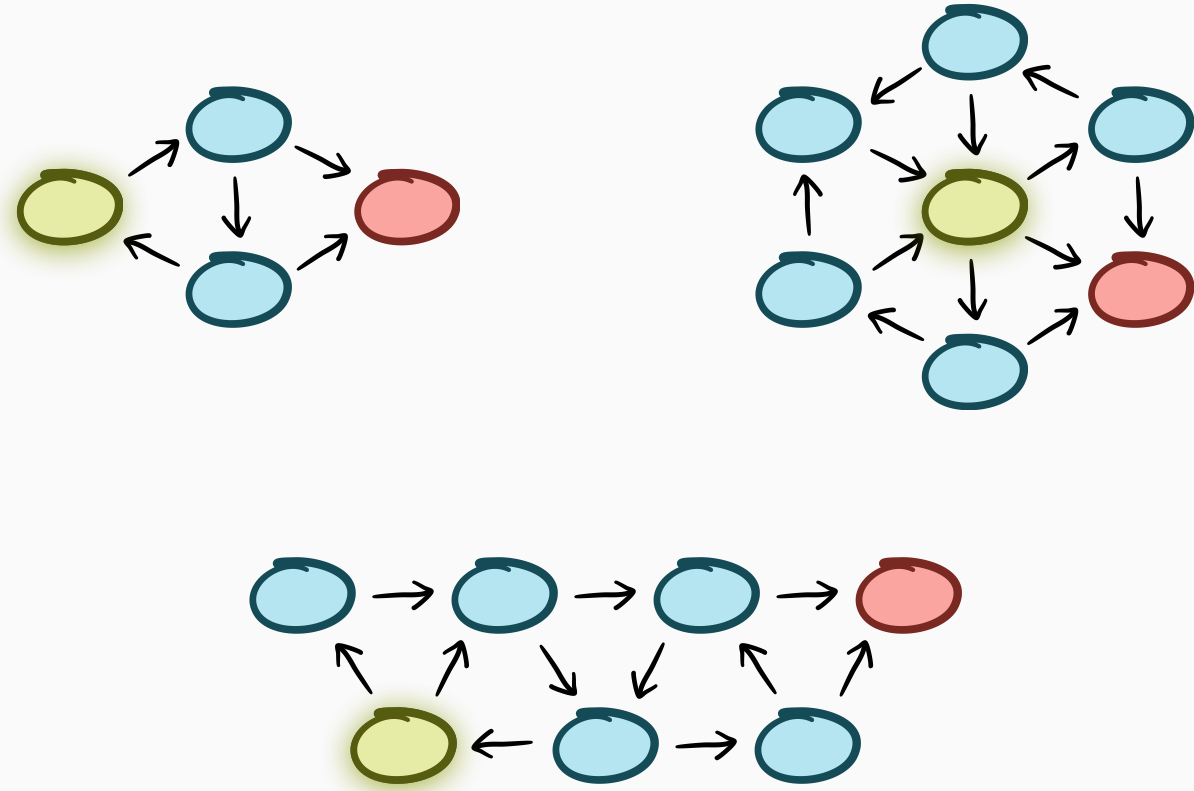
Markov chain selection (MCS)



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

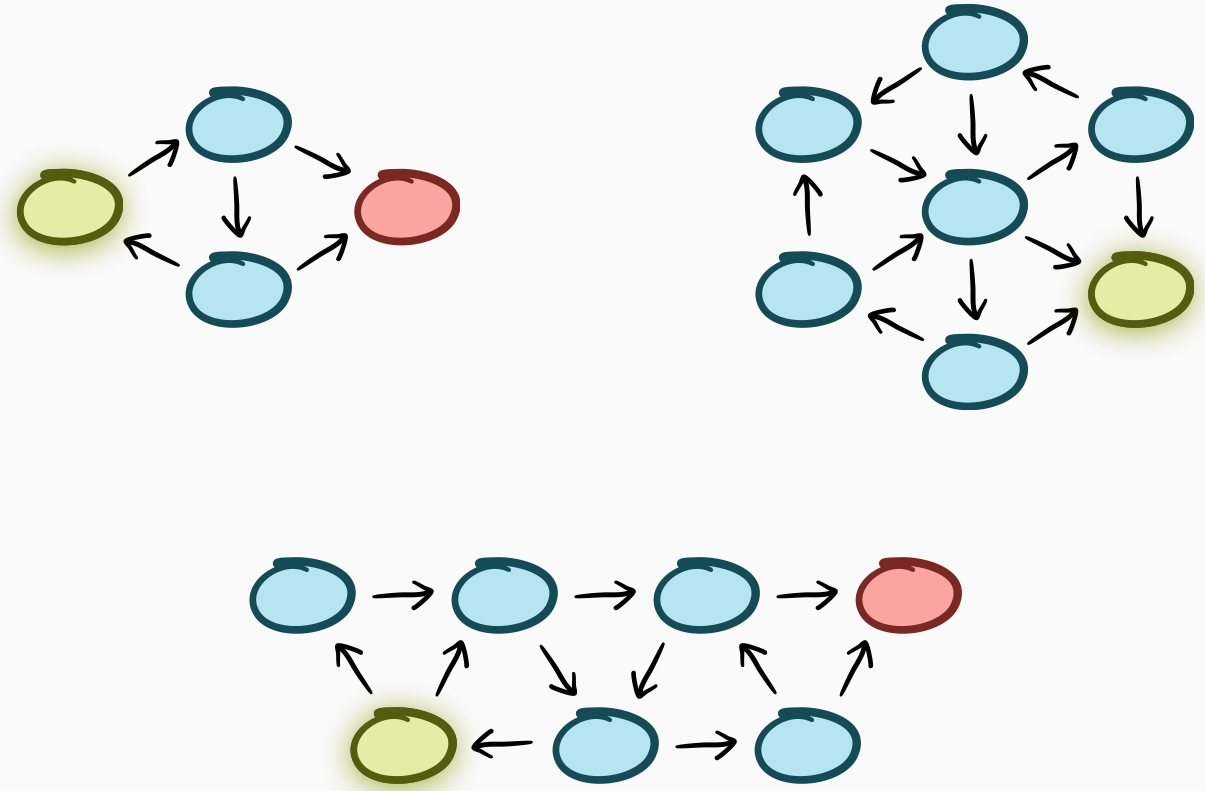
Markov chain selection (MCS)



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

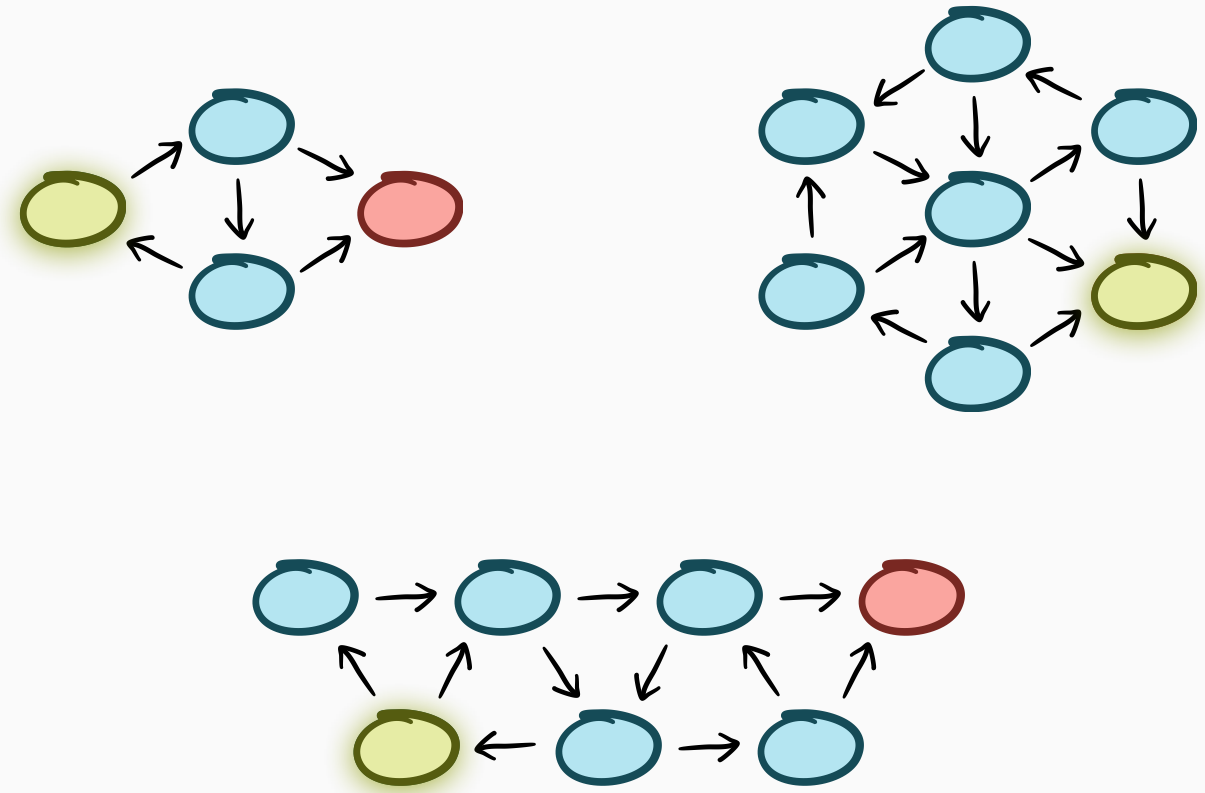
Markov chain selection (MCS)



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Markov chain selection (MCS)

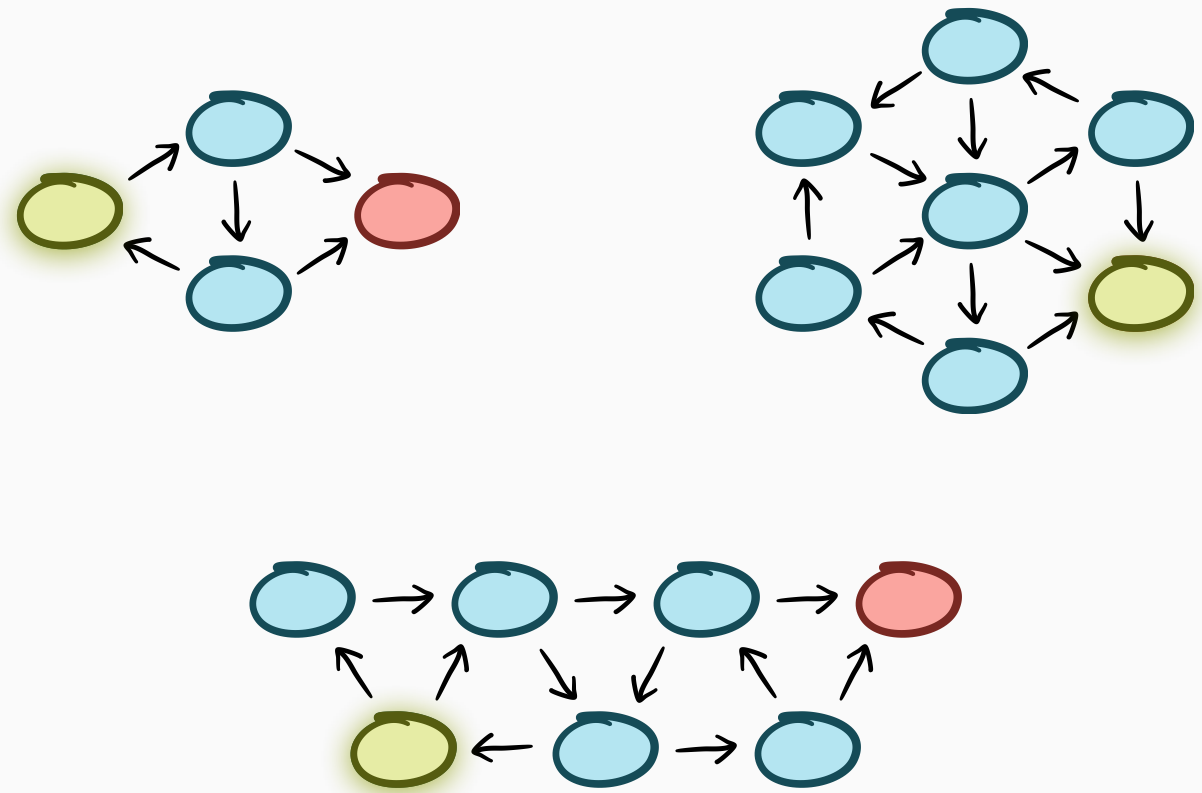


Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$

Markov chain selection (MCS)



Decision process:

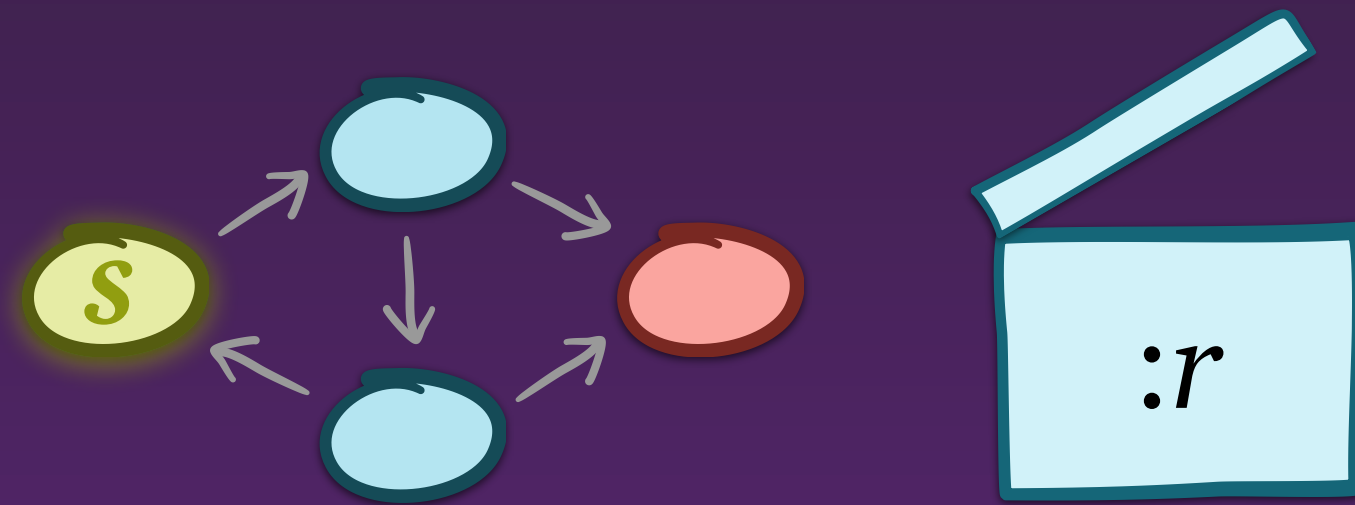
- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$



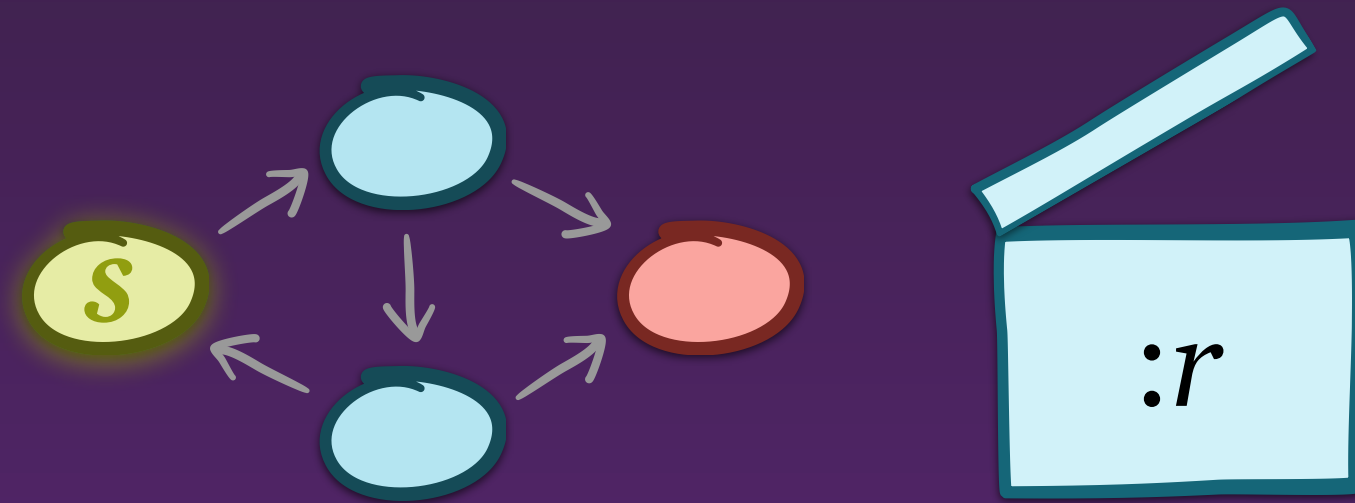
Optimal policy: advance Markov chain whose state has greatest *Gittins index*

The *Gittins index* of **state s** is the value of r such that we're indifferent between “act on **s** ” and “get r now”



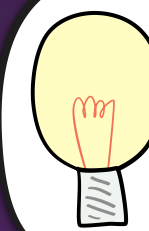
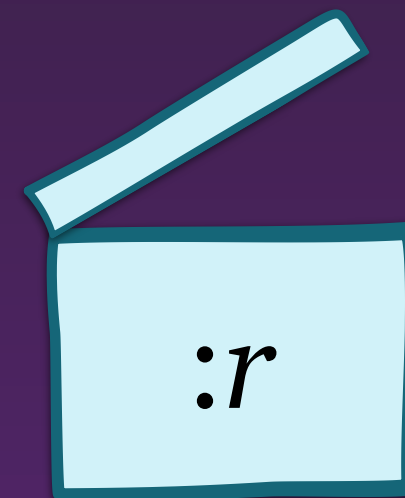
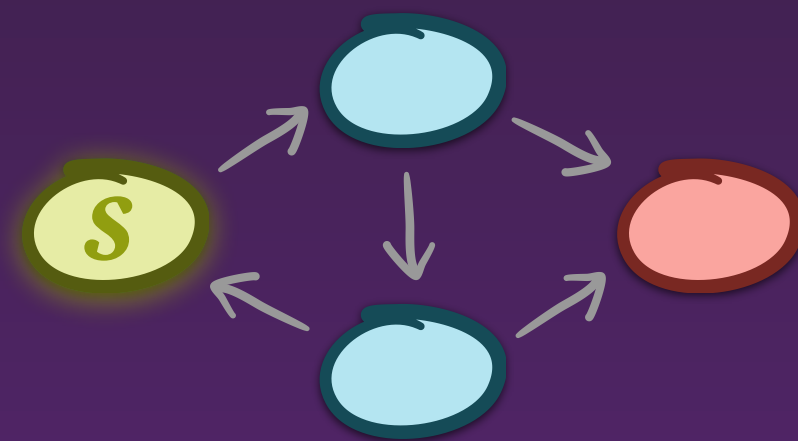
$$G(s)$$

The *Gittins index* of **state s** is the value of r such that we're indifferent between “act on **s** ” and “get r now”



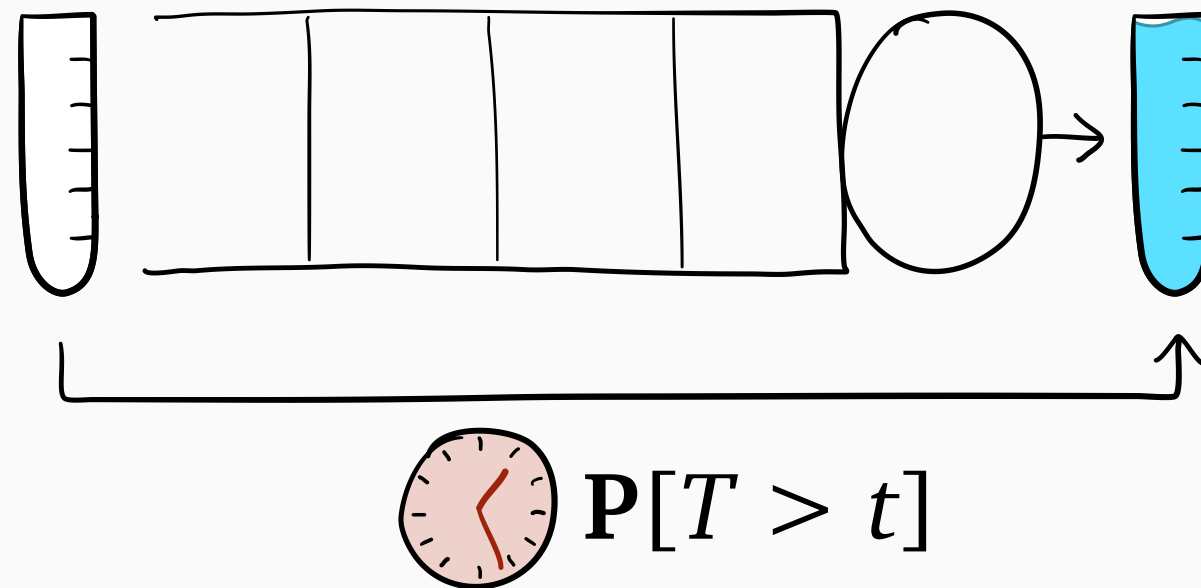
$$G(s)$$

The *Gittins index* of **state s** is the value of r such that we're indifferent between “act on **s** ” and “get r now”



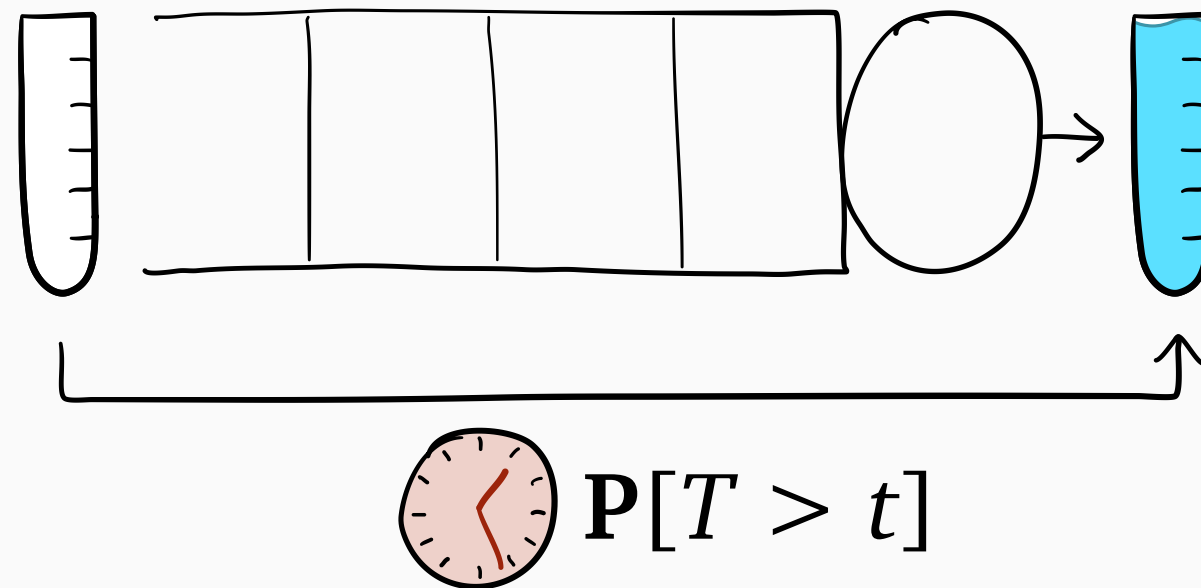
evaluate stochastic options by *calibrating to deterministic options*

Tail scheduling



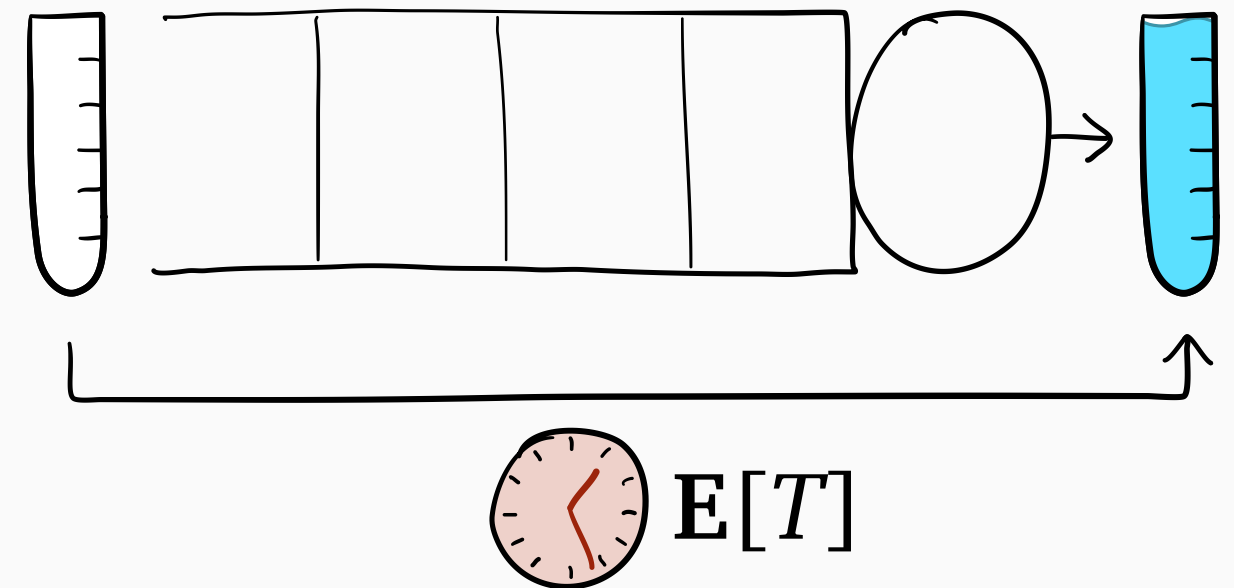
Goal: minimize probability of very long response time

Tail scheduling

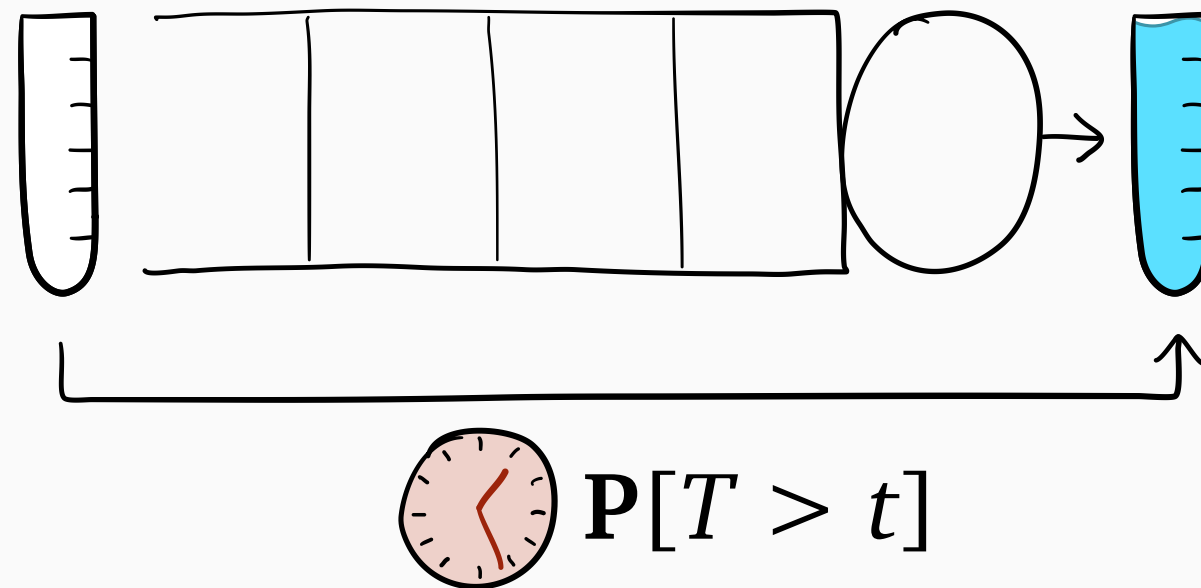


Goal: minimize probability of very long response time

Mean scheduling

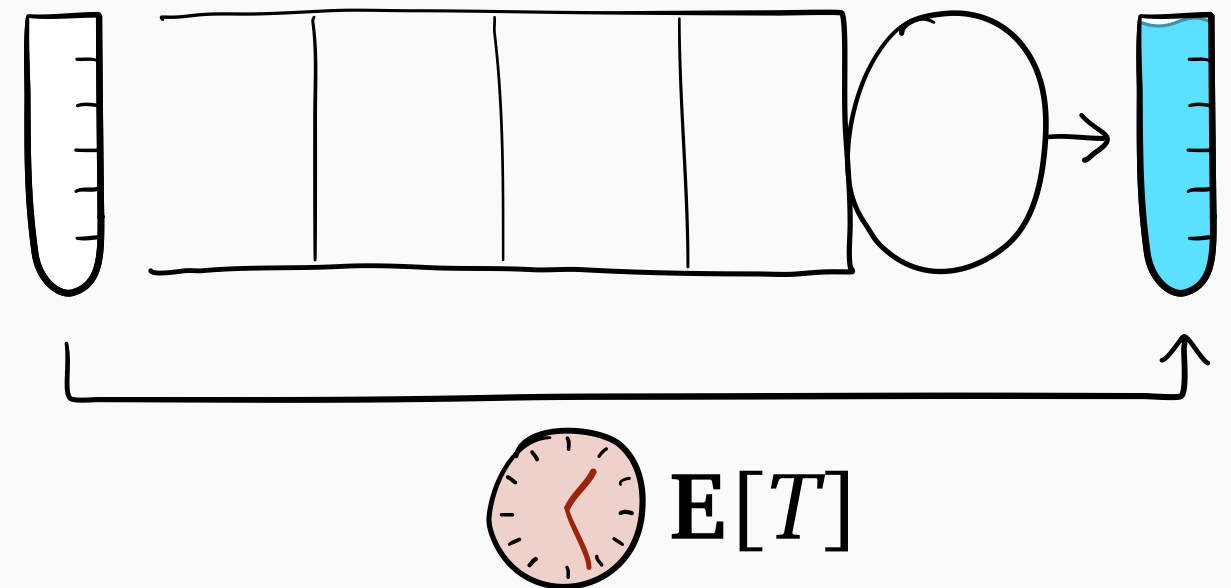


Tail scheduling



Goal: minimize probability of very long response time

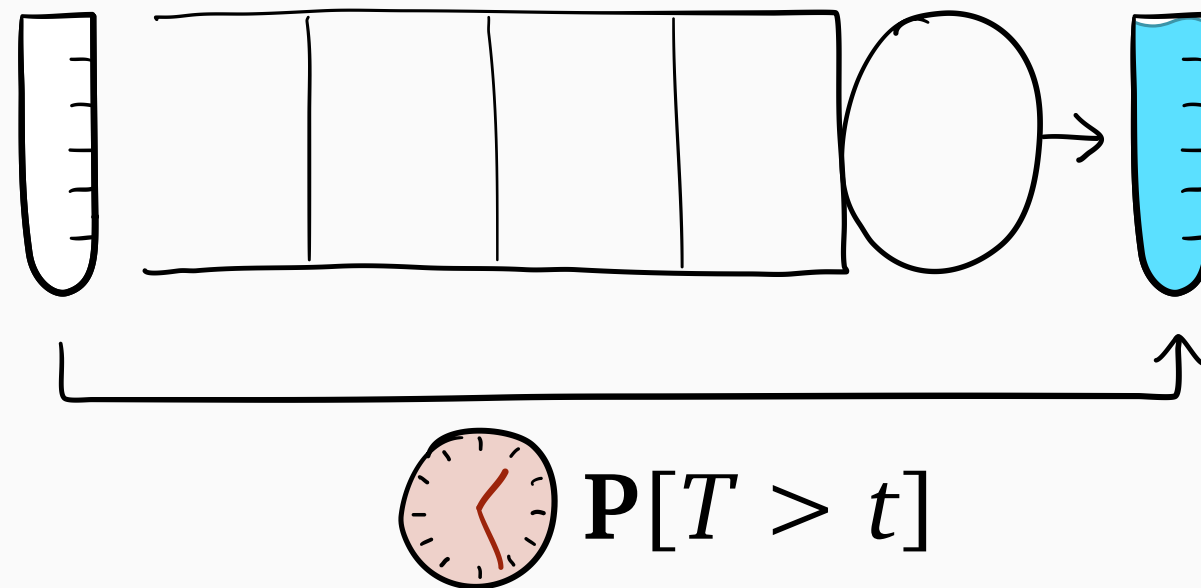
Mean scheduling



Goal: minimize average response time (or # jobs)

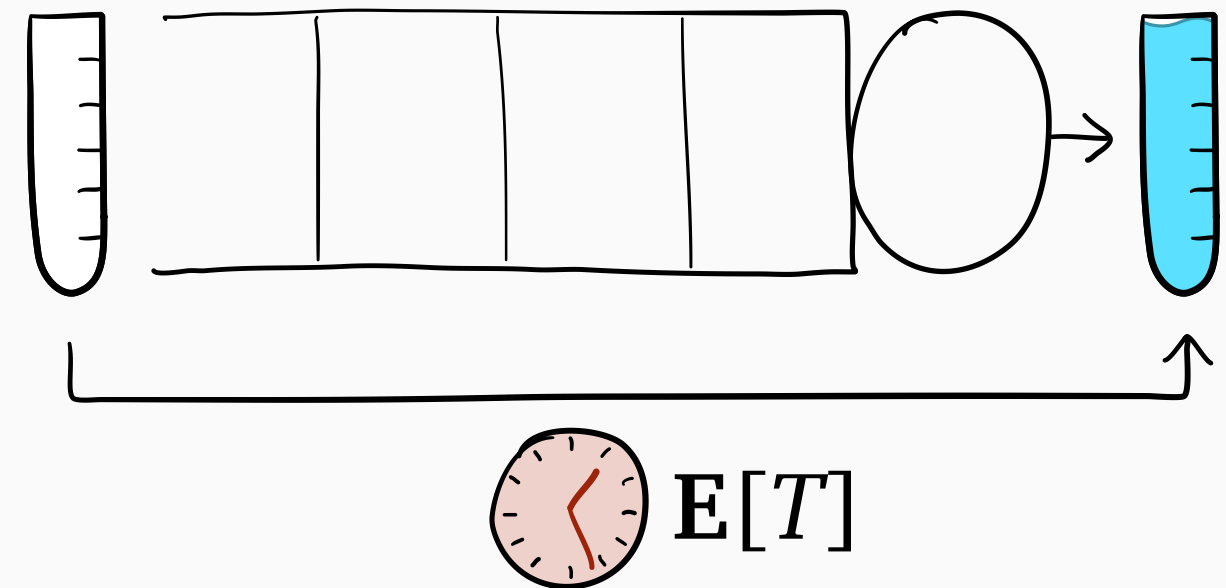
classic **Gittins** application
[Sevcik 1971; von Olivier, 1972;
Klimov, 1975; Gittins, 1979]

Tail scheduling



Goal: minimize probability of very long response time

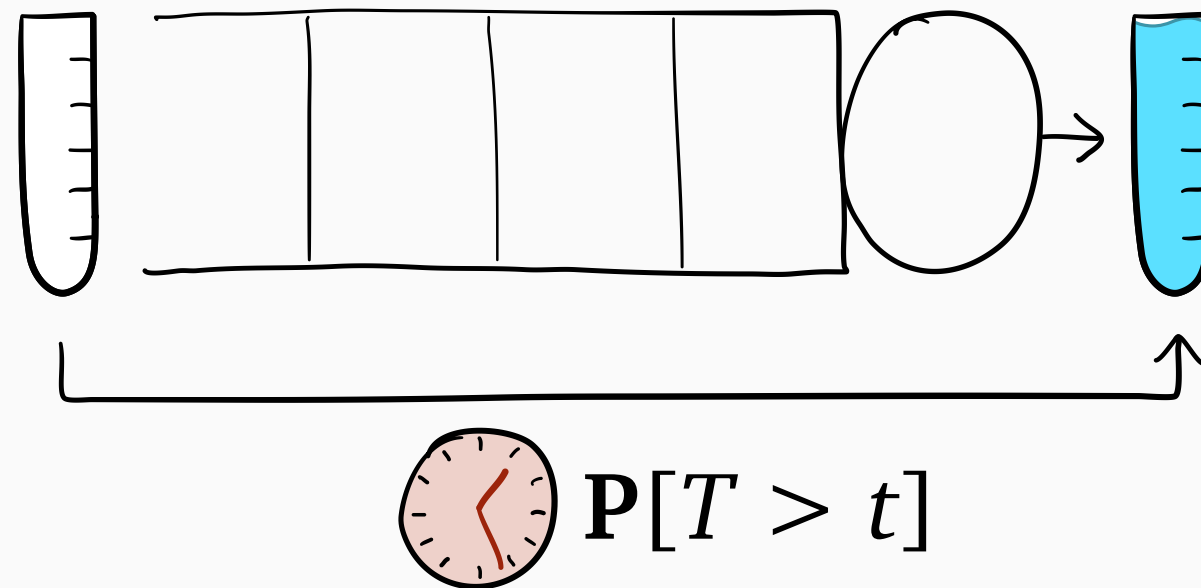
Mean scheduling



Goal: minimize average response time (or # jobs)

Gittins seems
inapplicable...

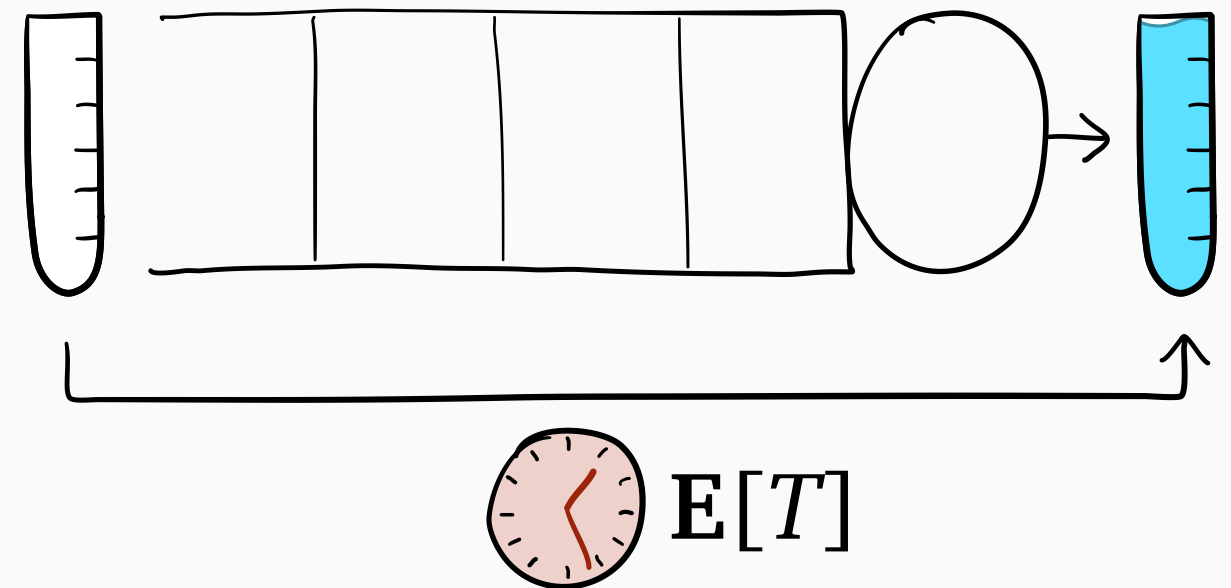
Tail scheduling



Goal: minimize probability
of very long response time

classic Gittins application
[Sevcik 1971; von Olivier, 1972;
Klimov, 1975; Gittins, 1979]

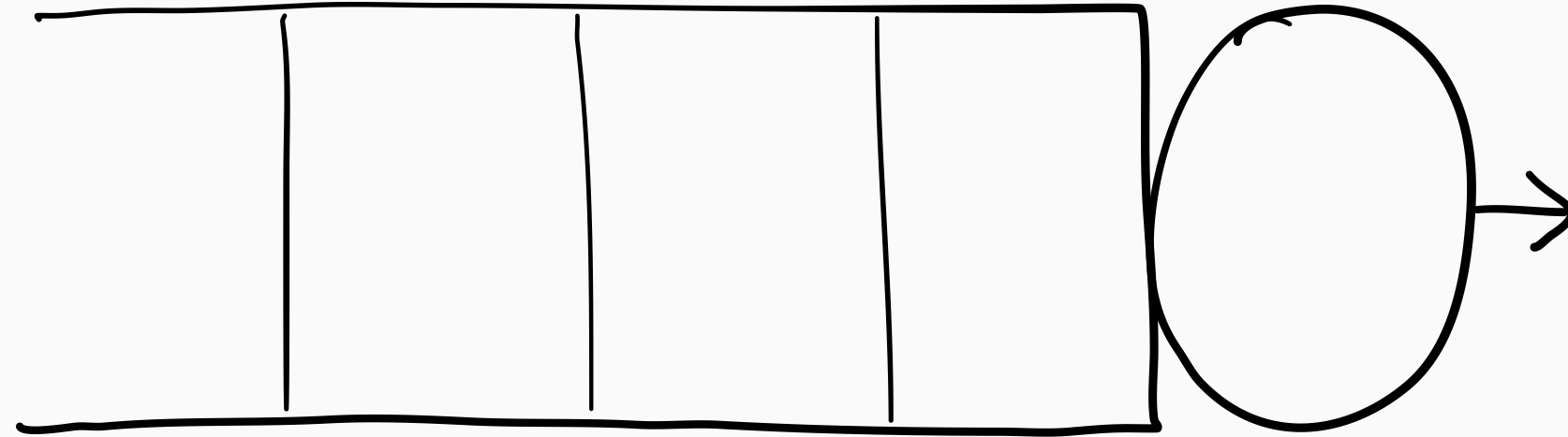
Mean scheduling



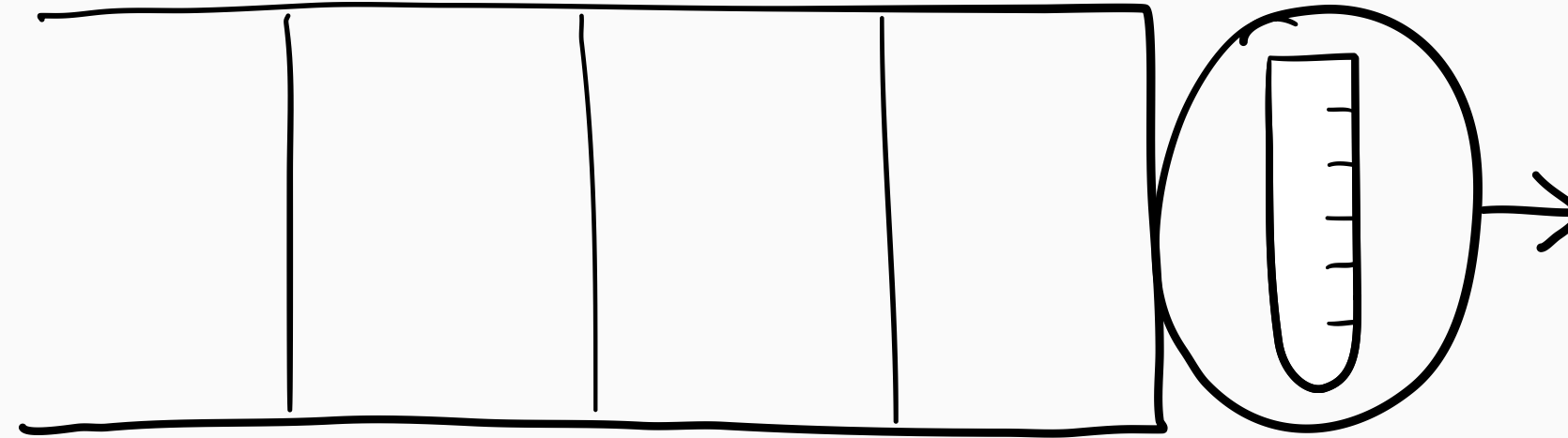
Goal: minimize average
response time (or # jobs)

Mean scheduling in the M/G/1

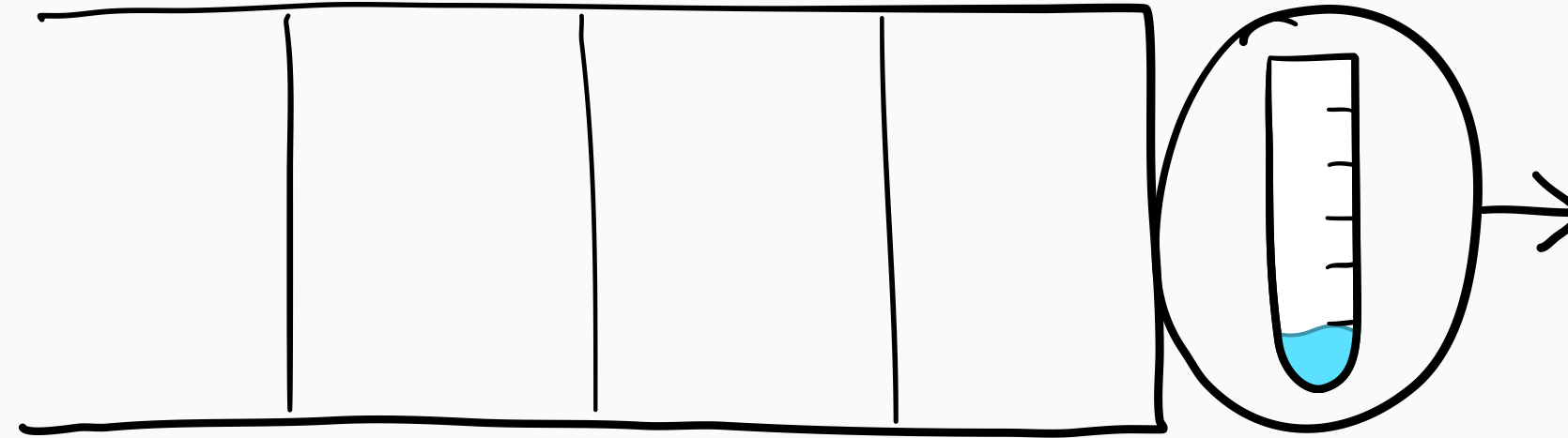
Mean scheduling in the M/G/1



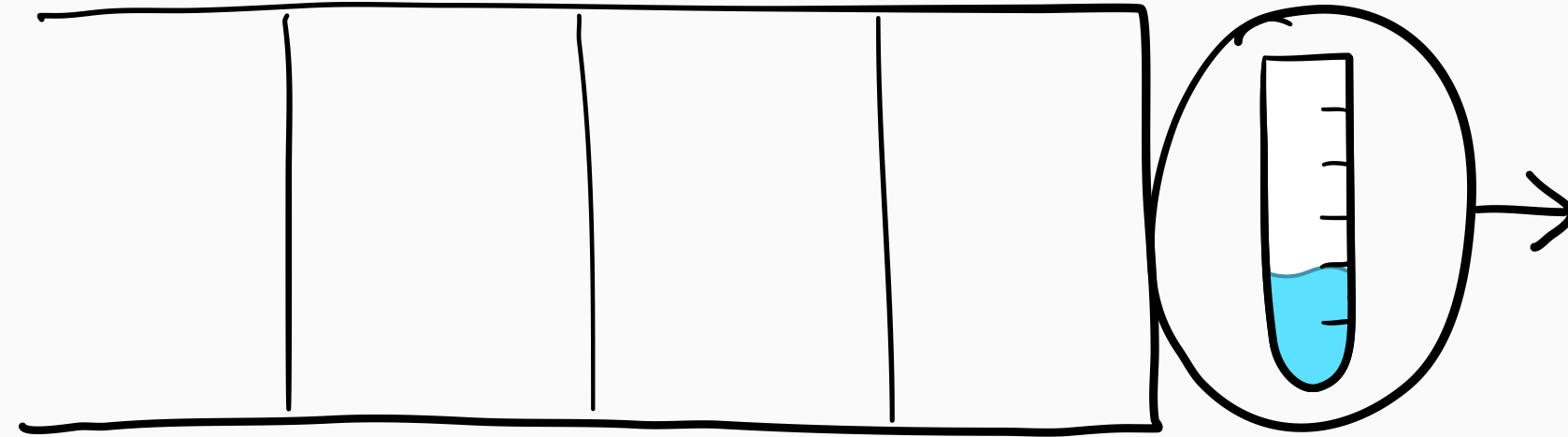
Mean scheduling in the M/G/1



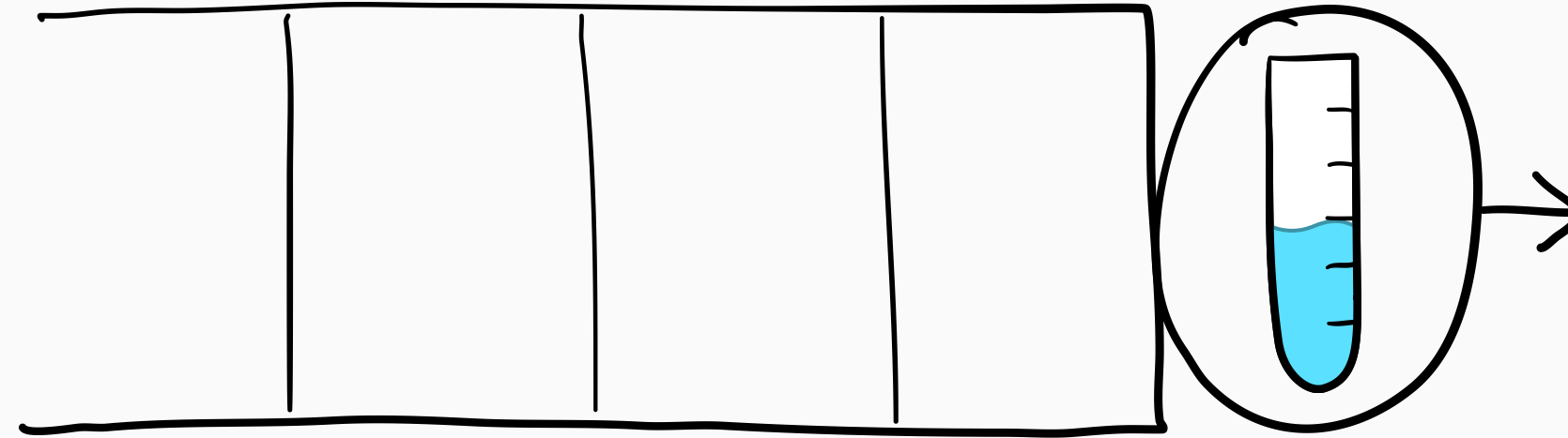
Mean scheduling in the M/G/1



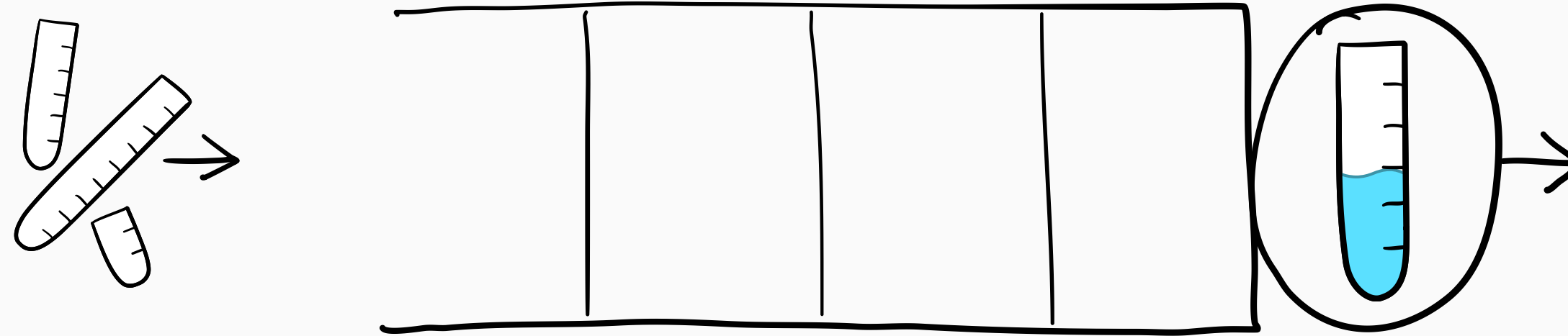
Mean scheduling in the M/G/1



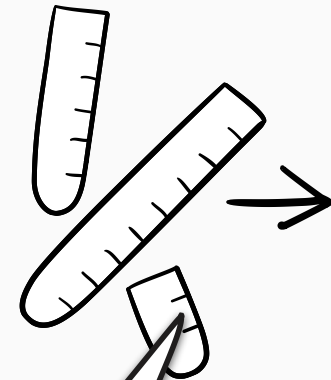
Mean scheduling in the M/G/1



Mean scheduling in the M/G/1

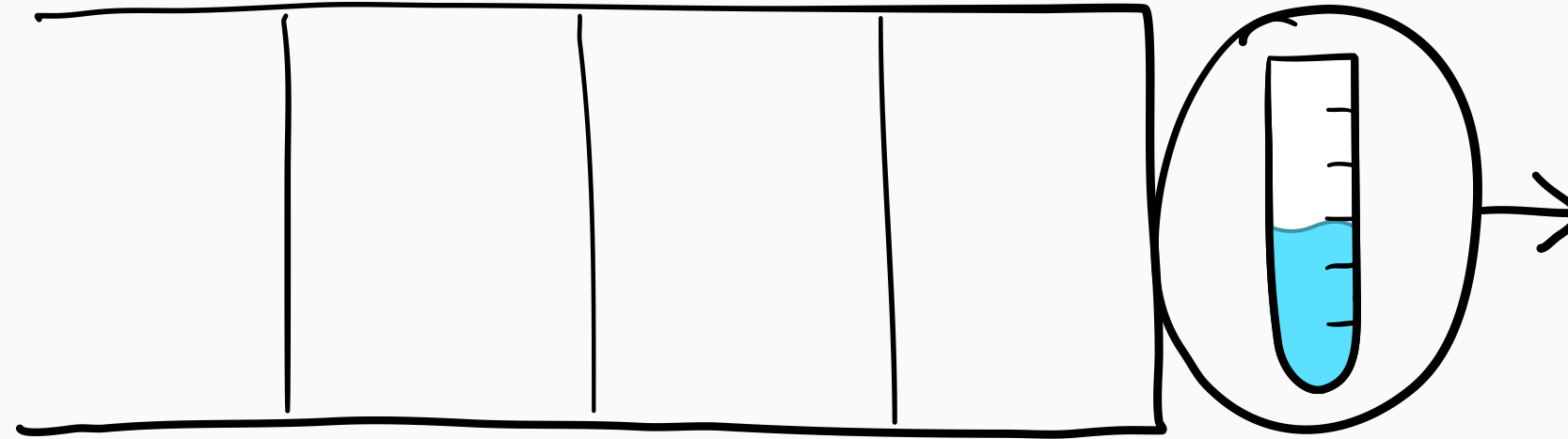


Mean scheduling in the M/G/1

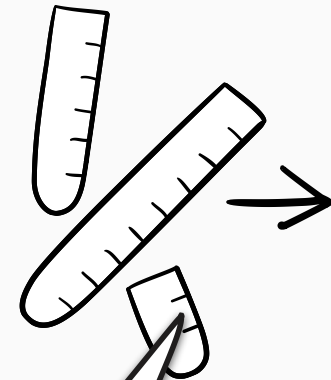


M/G arrivals:

- arrival rate λ
- size distribution Z

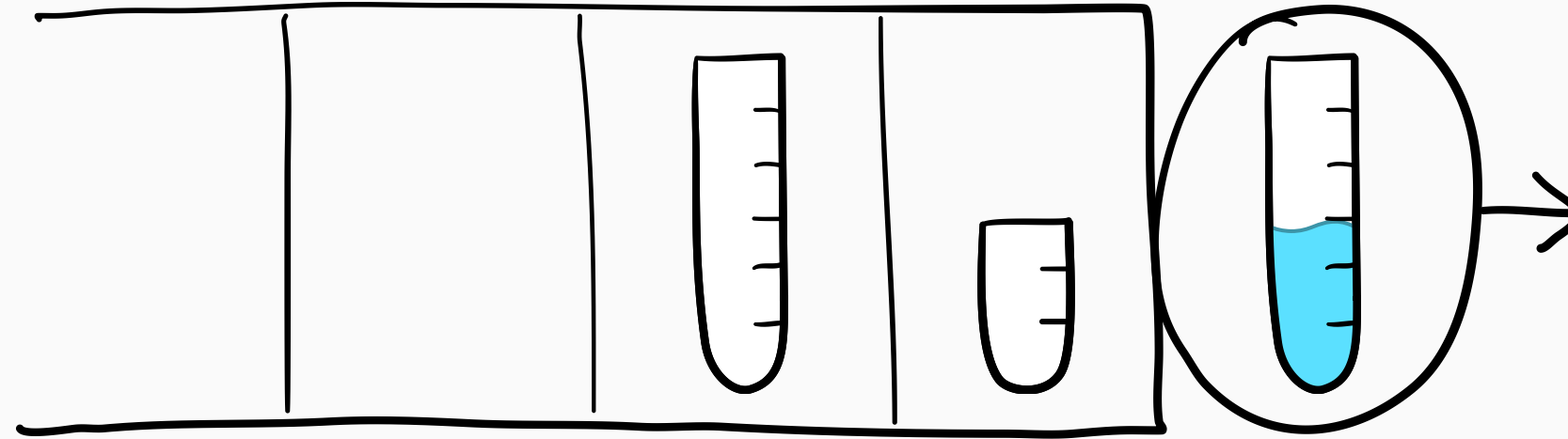


Mean scheduling in the M/G/1



M/G arrivals:

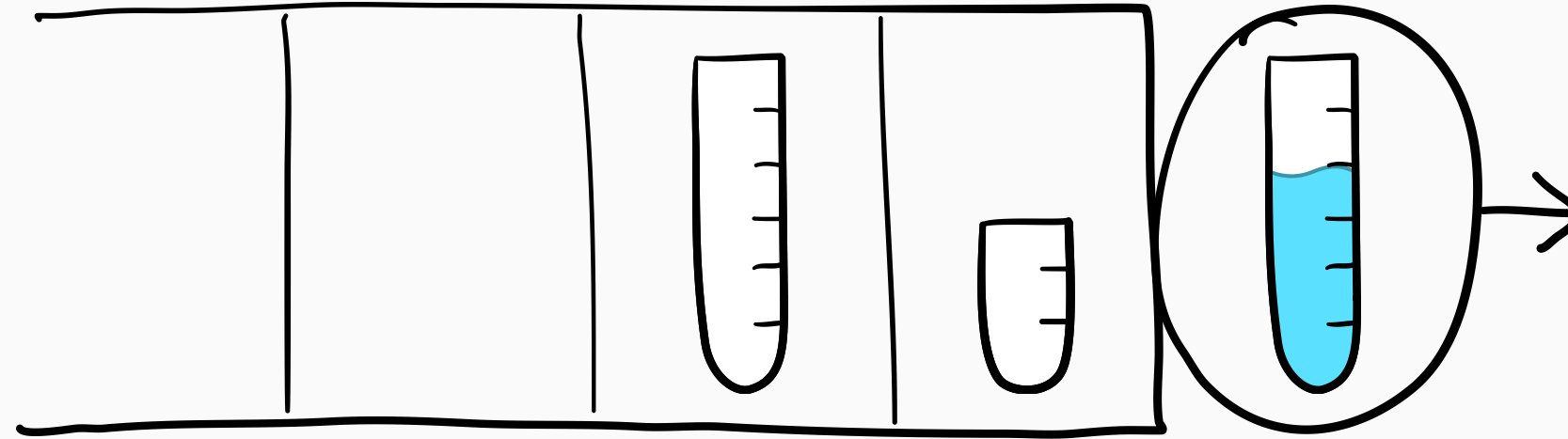
- arrival rate λ
- size distribution Z



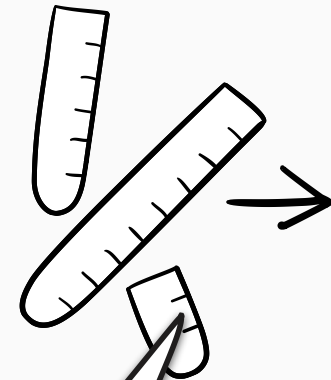
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z

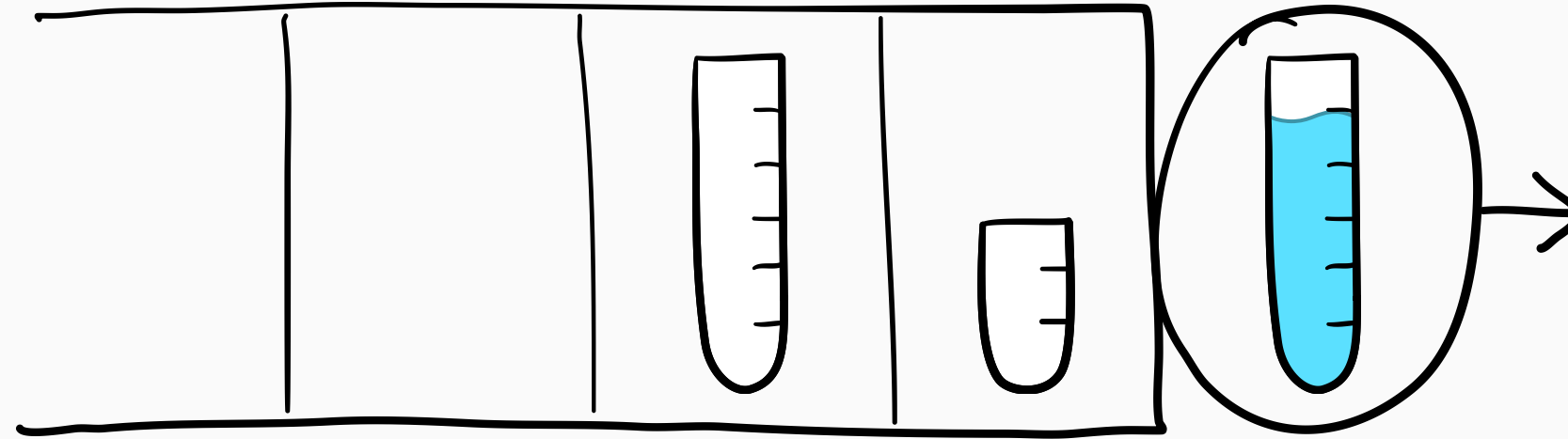


Mean scheduling in the M/G/1



M/G arrivals:

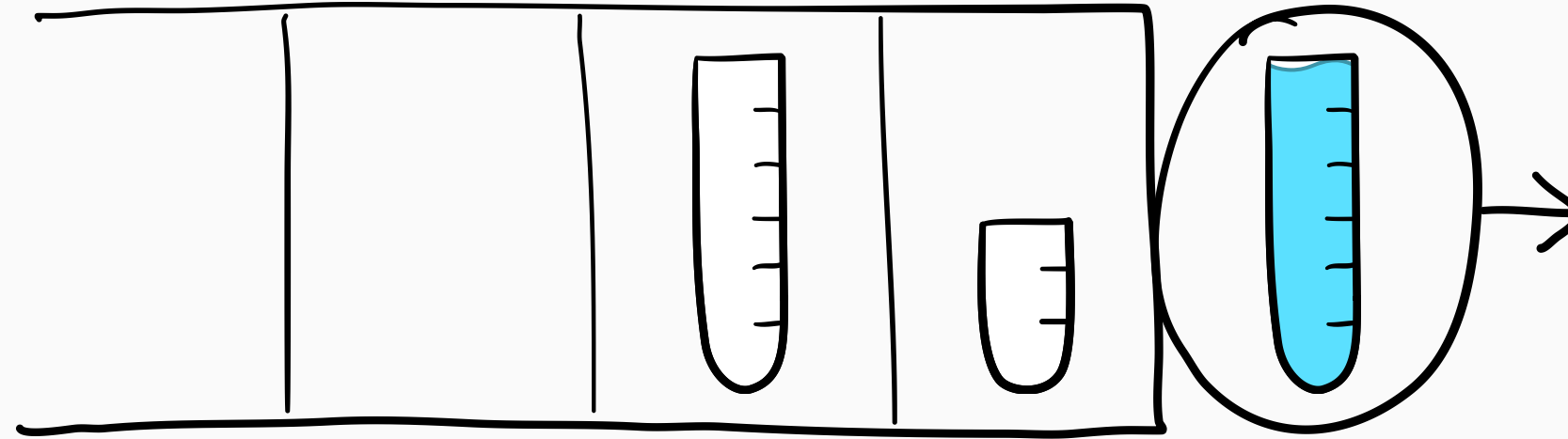
- arrival rate λ
- size distribution Z



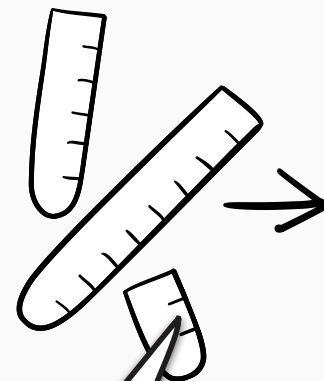
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z

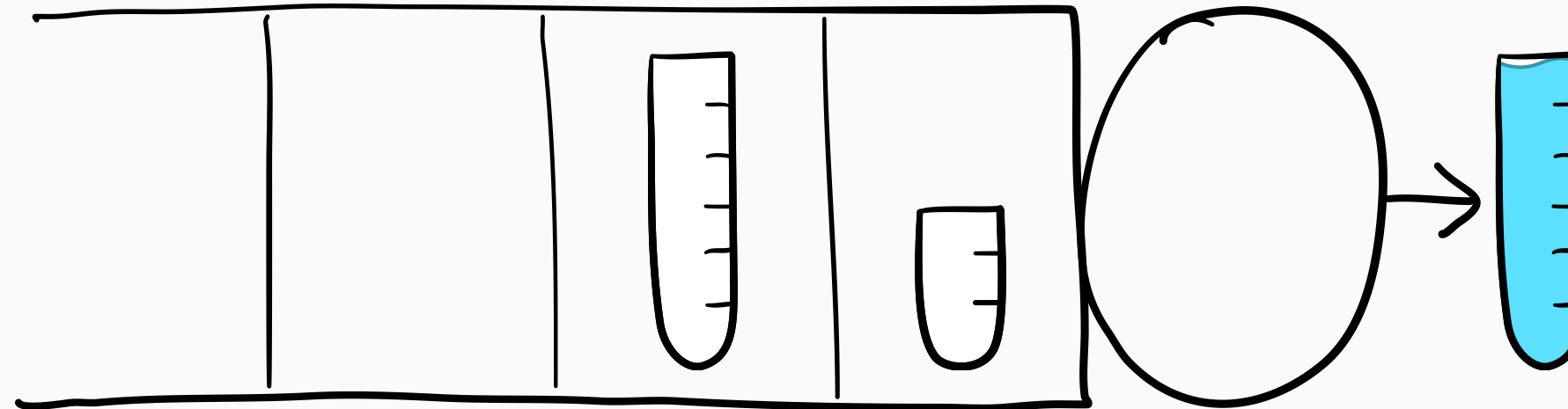


Mean scheduling in the M/G/1



M/G arrivals:

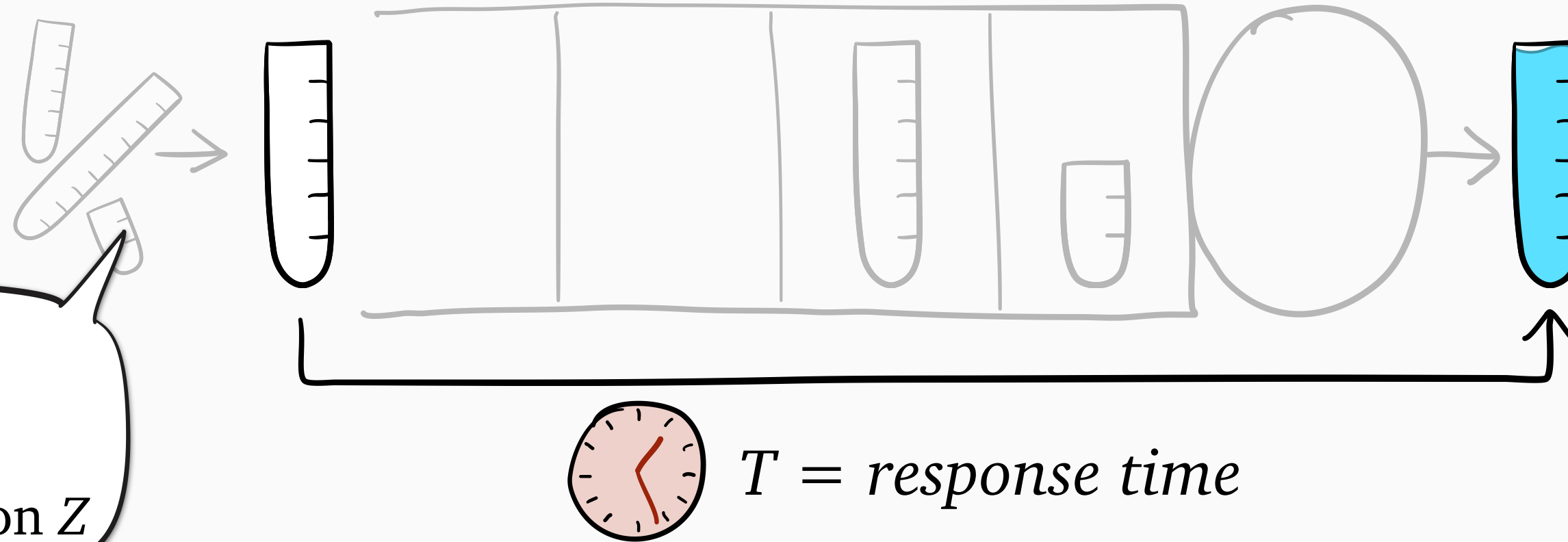
- arrival rate λ
- size distribution Z



Mean scheduling in the M/G/1

M/G arrivals:

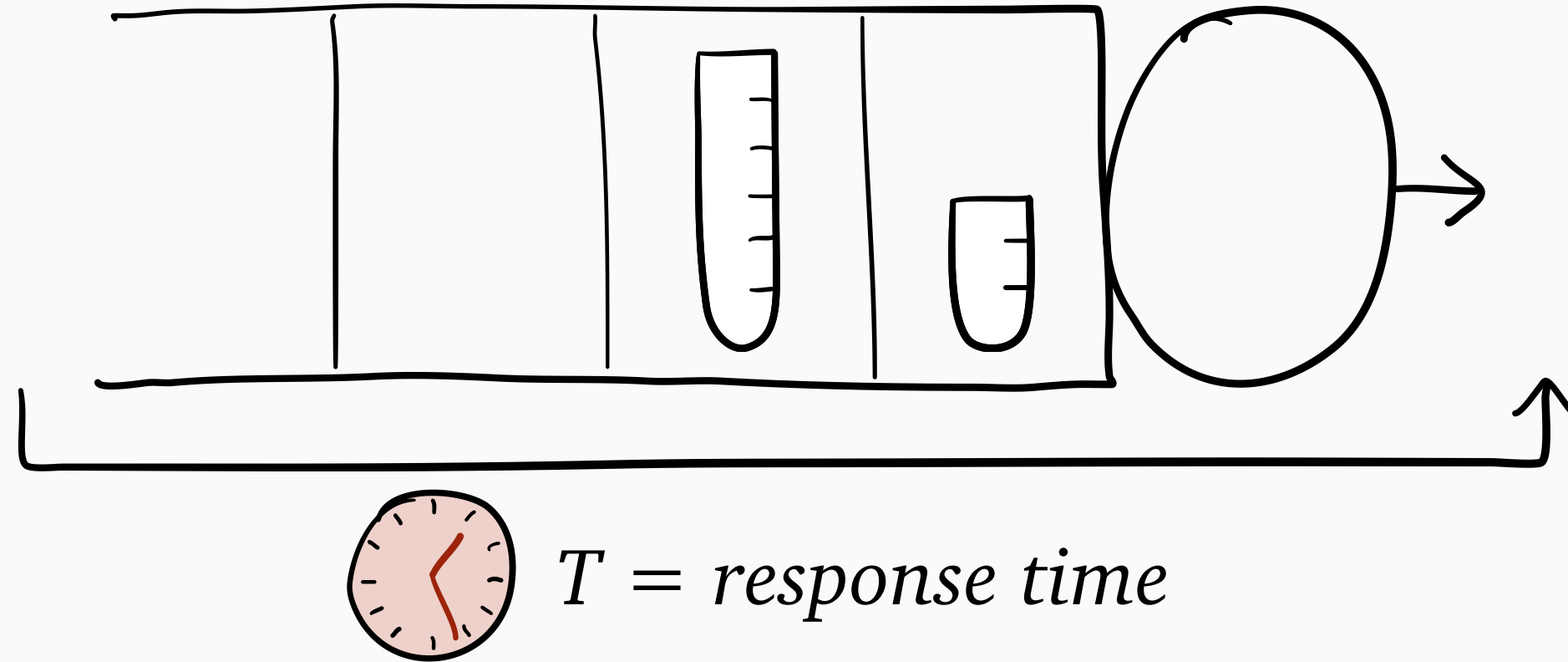
- arrival rate λ
- size distribution Z



Mean scheduling in the M/G/1

M/G arrivals:

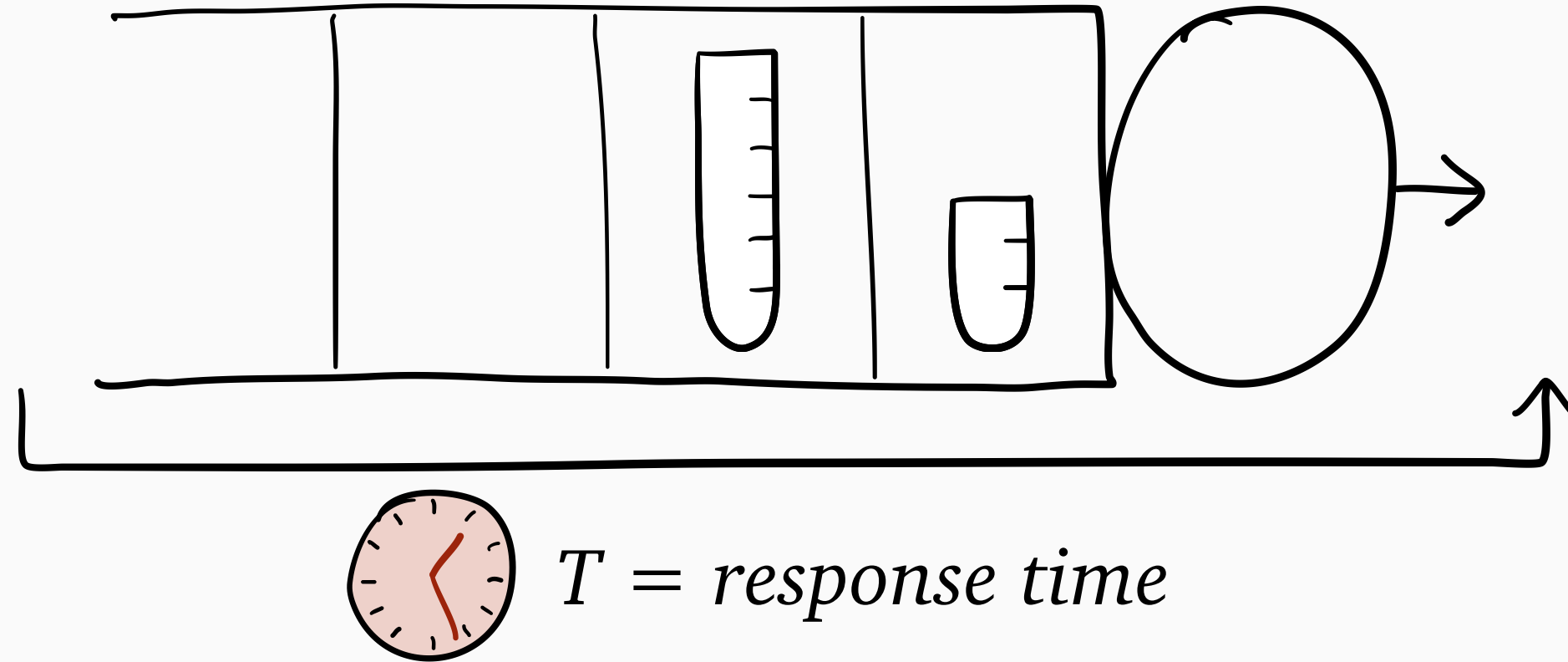
- arrival rate λ
- size distribution Z



Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z

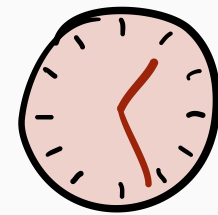
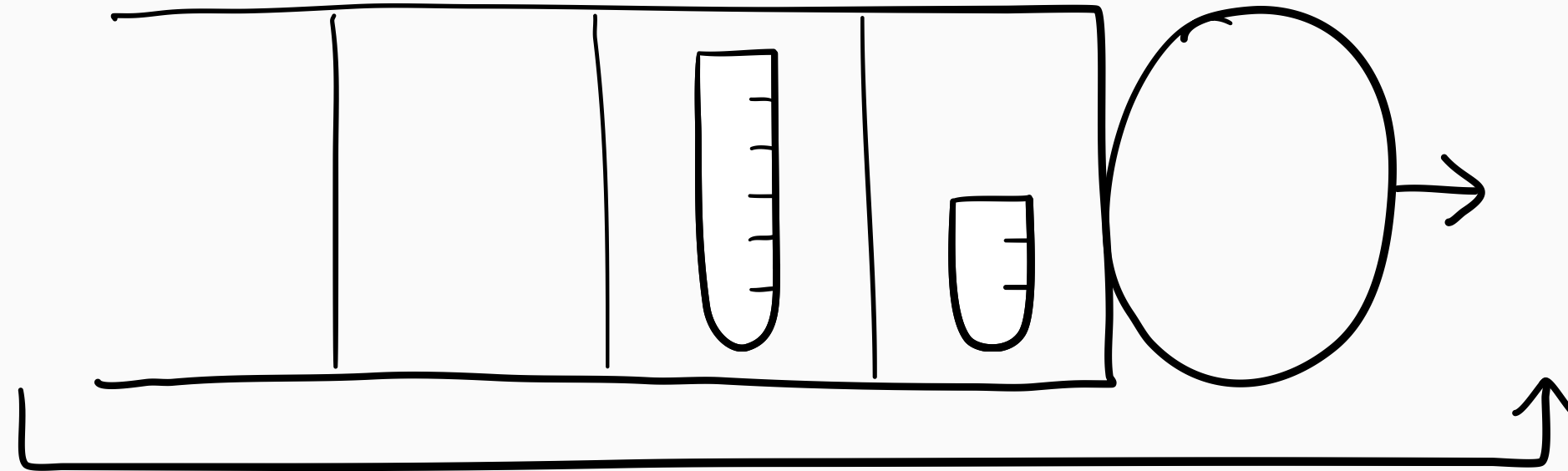


Minimize $E[T]$?

Mean scheduling in the M/G/1

M/G arrivals:

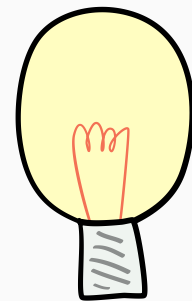
- arrival rate λ
- size distribution Z



$T = \text{response time}$



Minimize $E[T]$?

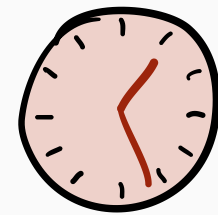
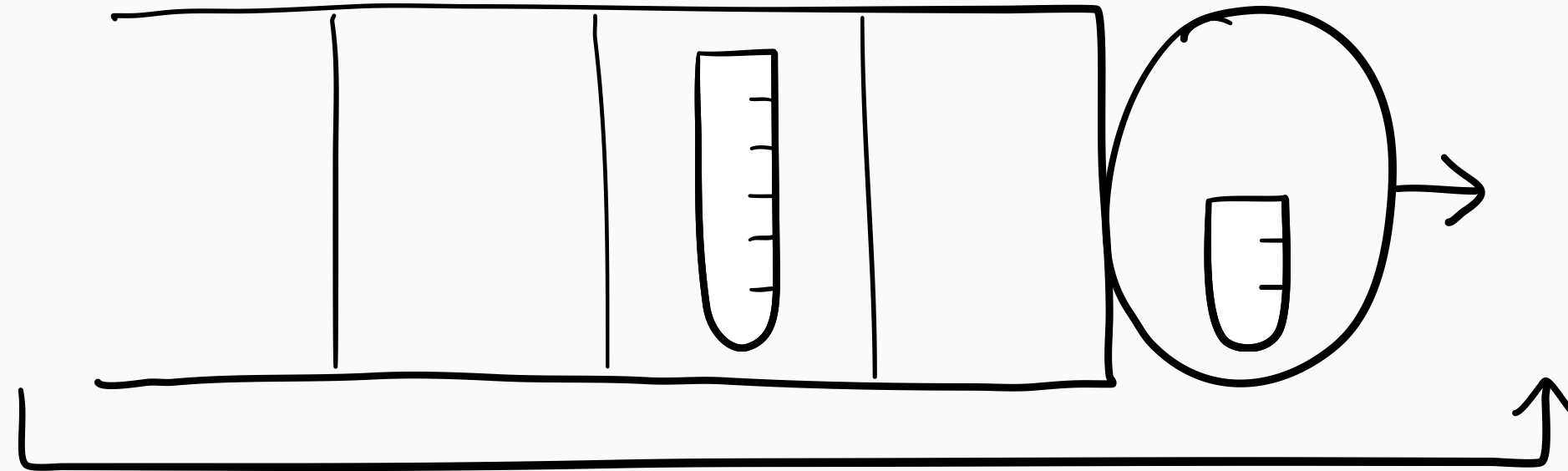


Serve short jobs
before long jobs

Mean scheduling in the M/G/1

M/G arrivals:

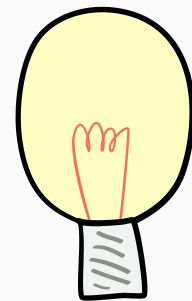
- arrival rate λ
- size distribution Z



$T = \text{response time}$



Minimize $E[T]$?

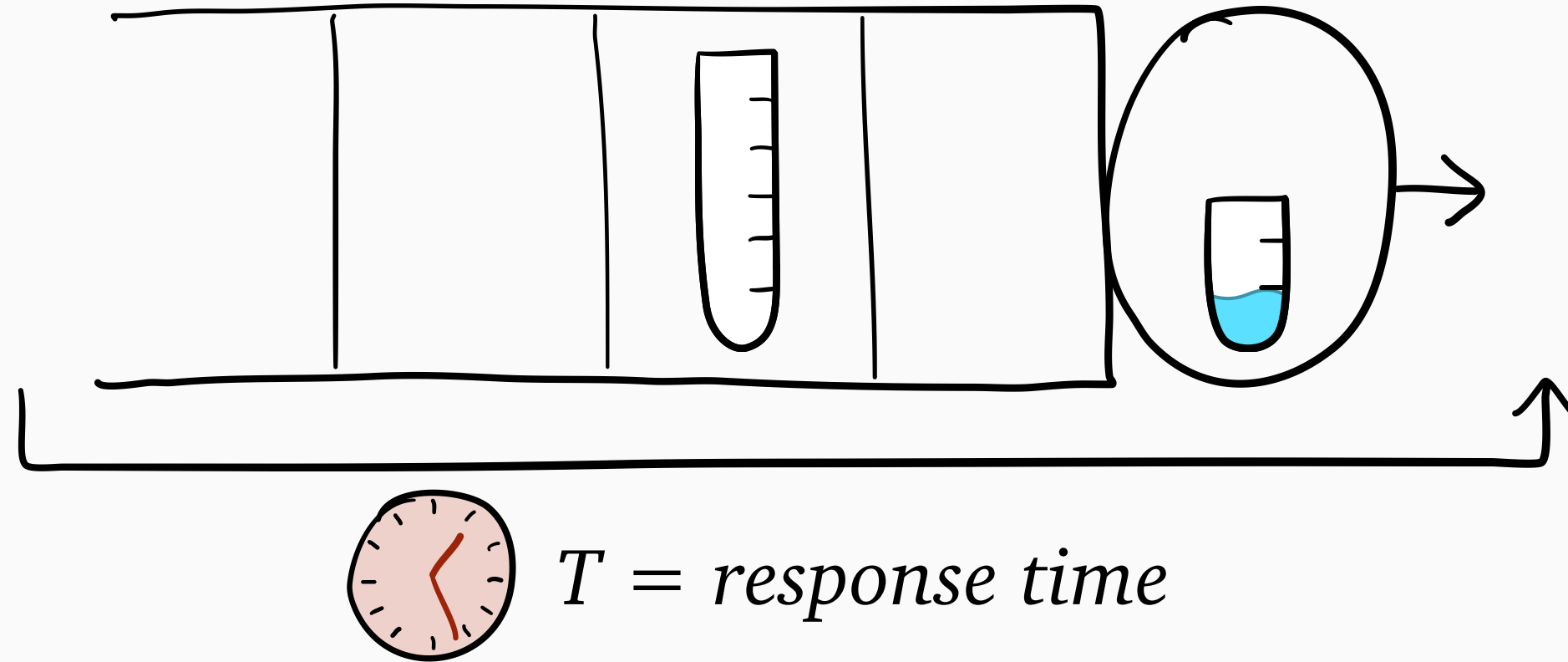


Serve short jobs
before long jobs

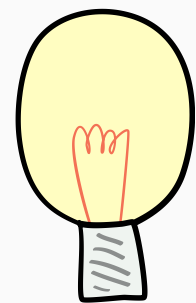
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

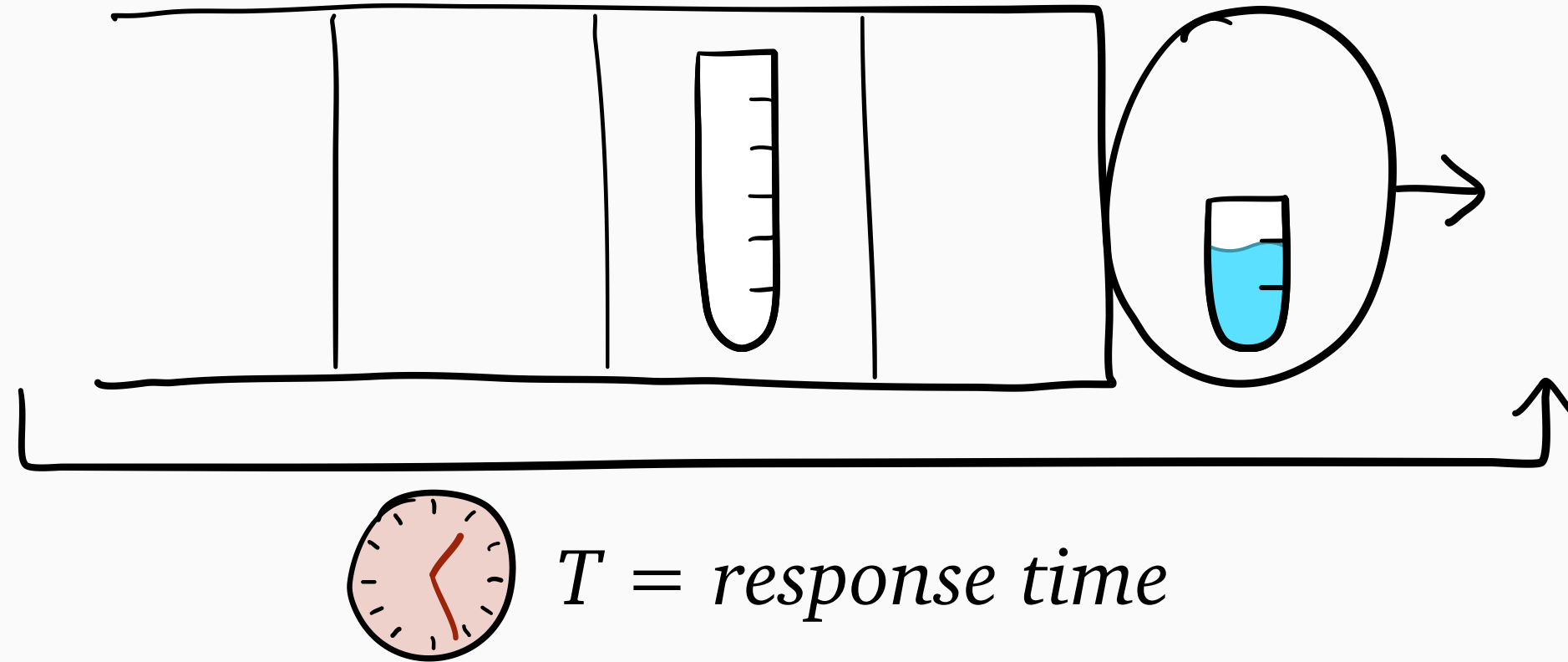


Serve short jobs
before long jobs

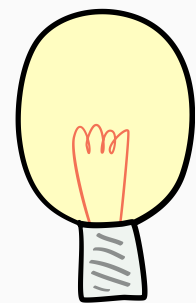
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

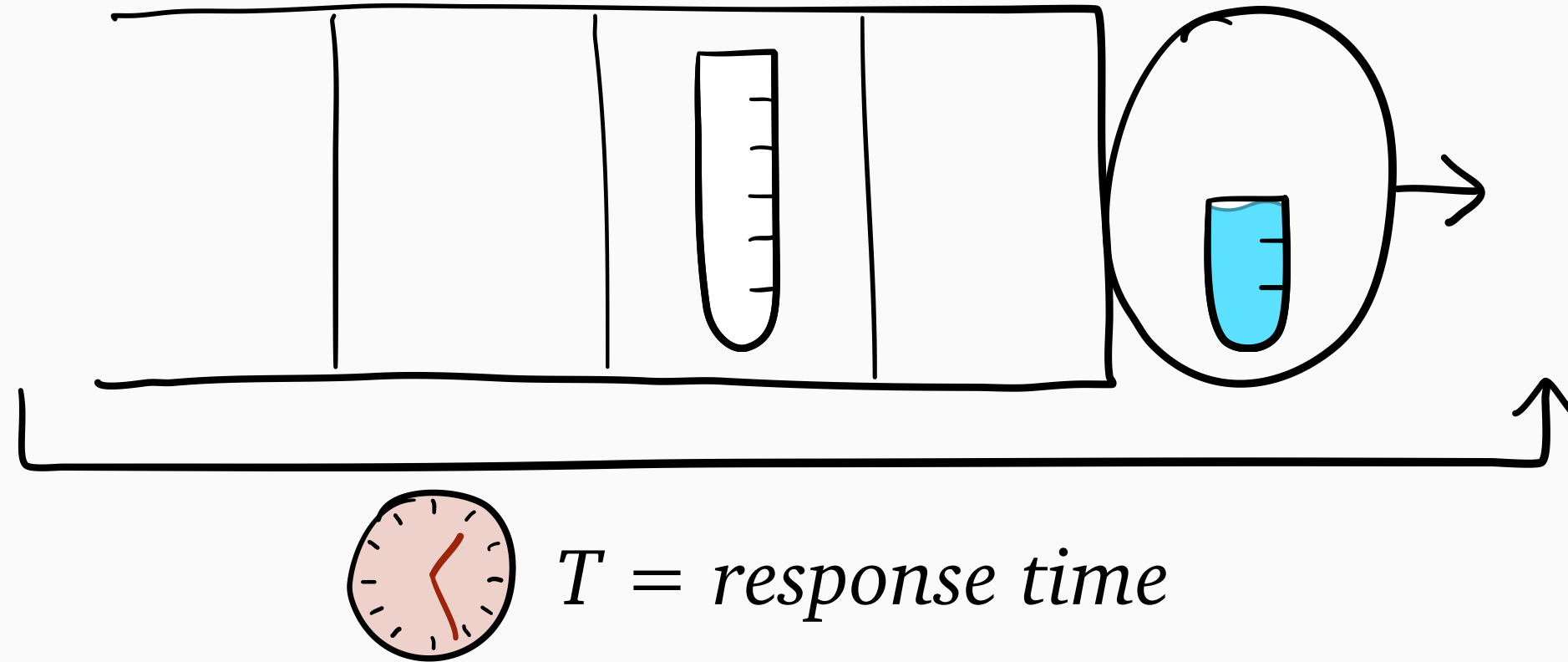


Serve short jobs
before long jobs

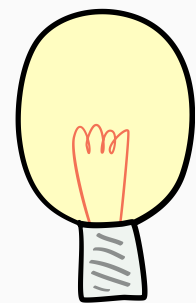
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

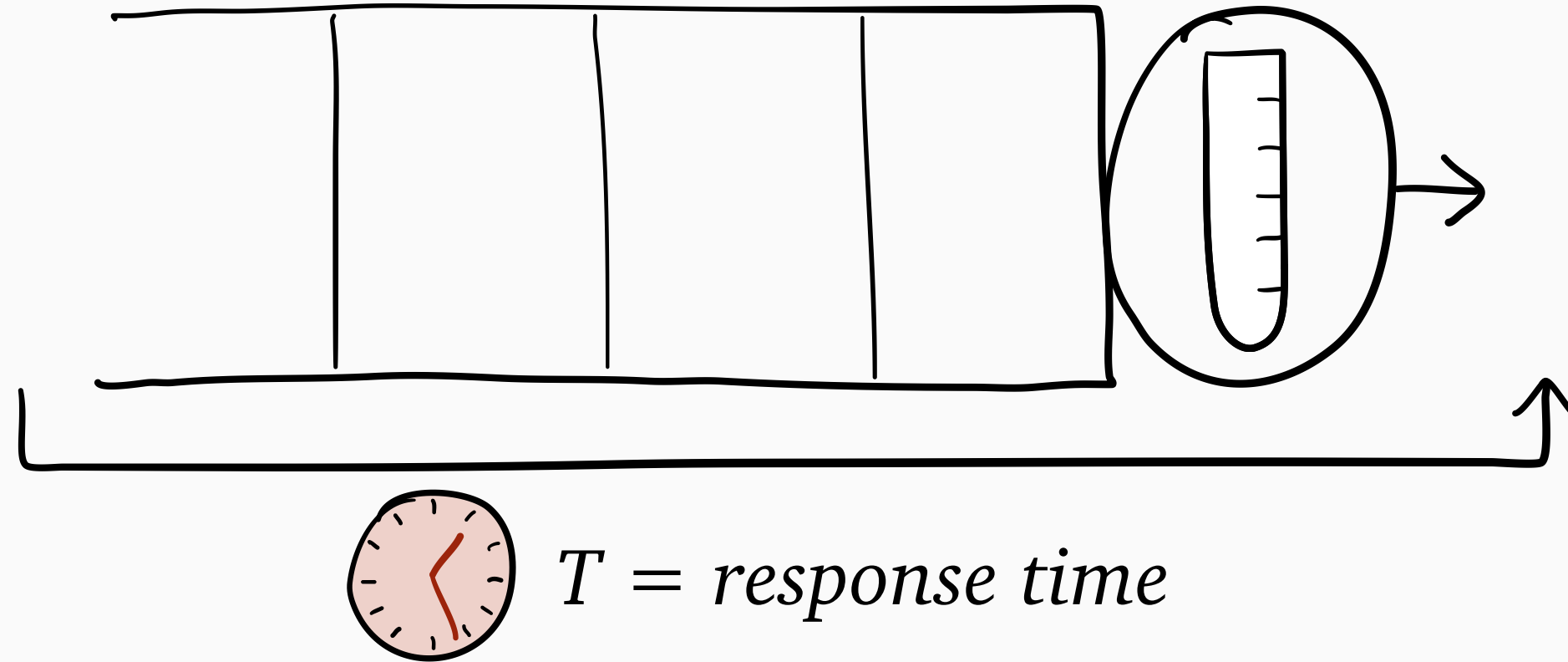


Serve short jobs
before long jobs

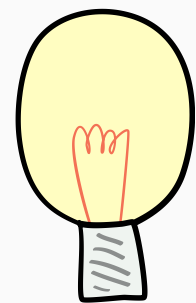
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

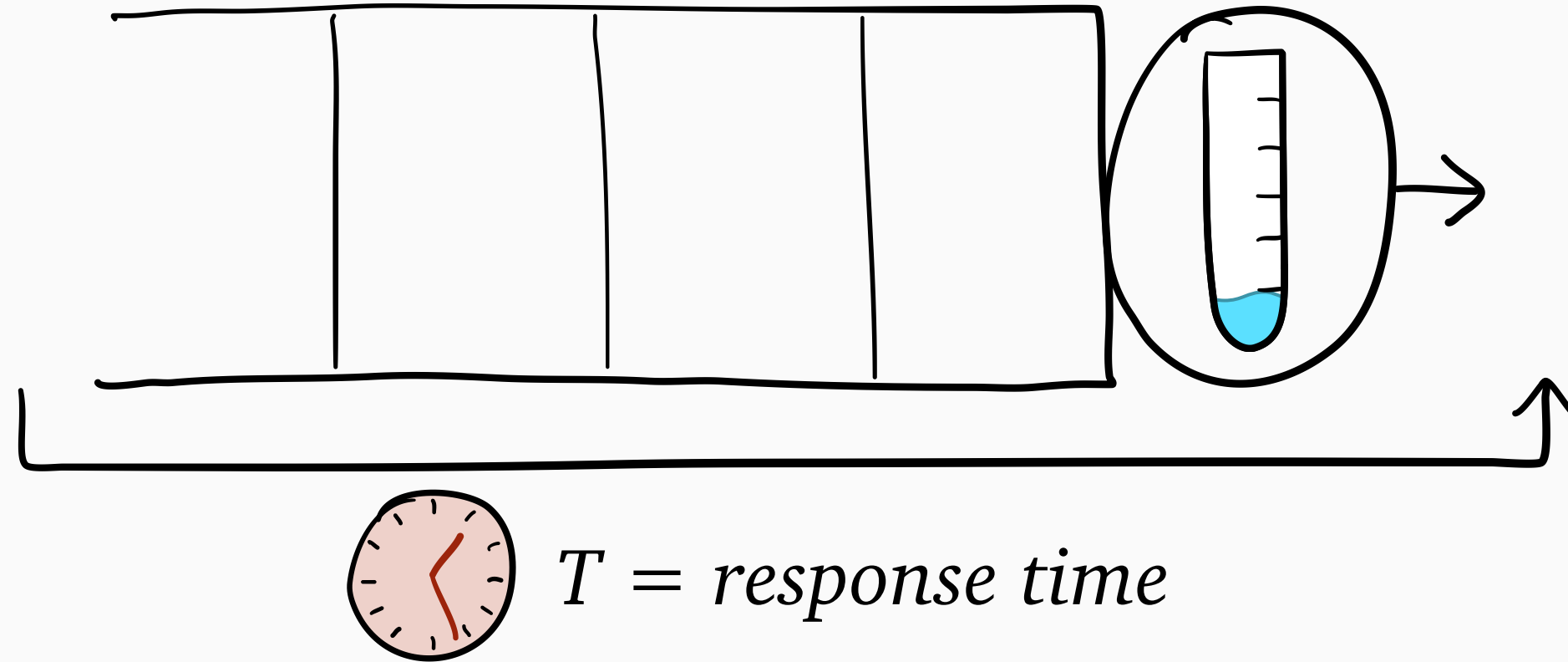


Serve short jobs
before long jobs

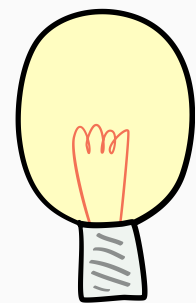
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

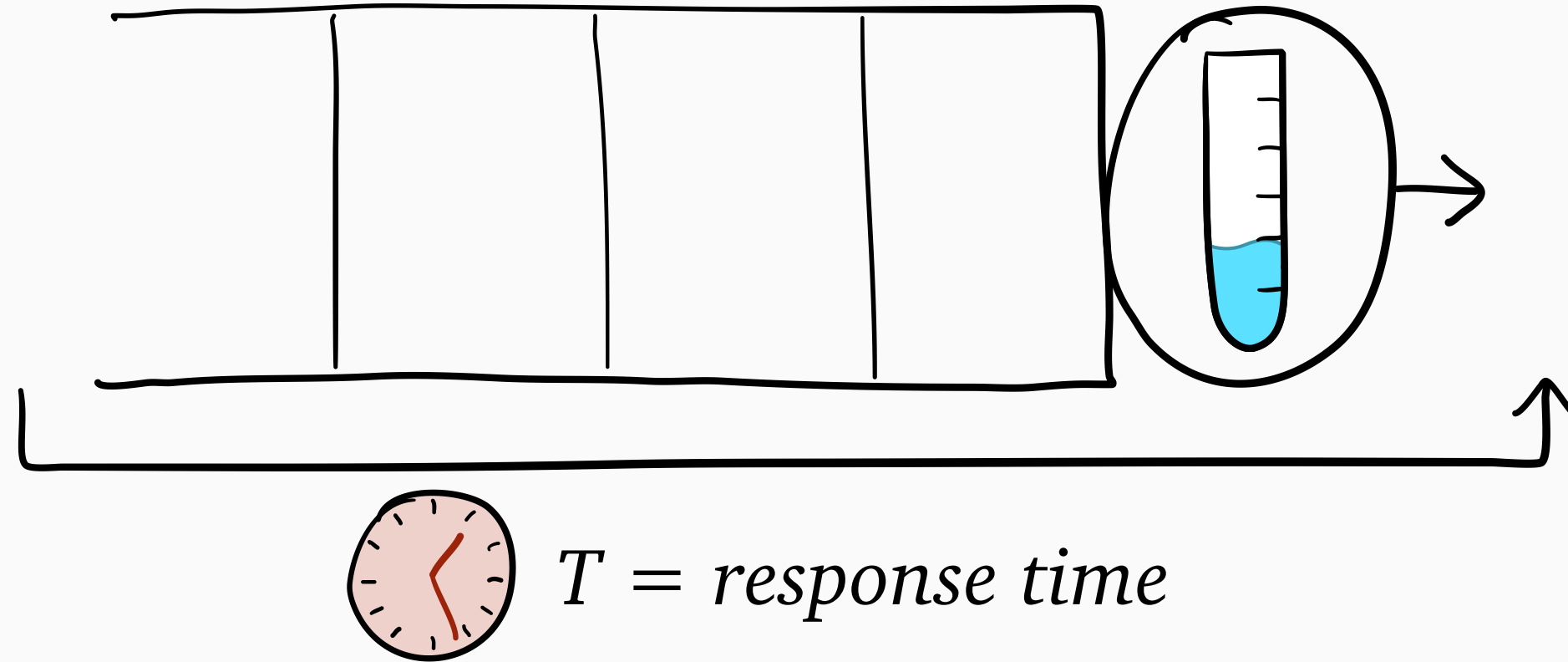


Serve short jobs
before long jobs

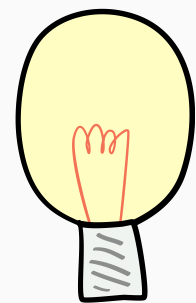
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

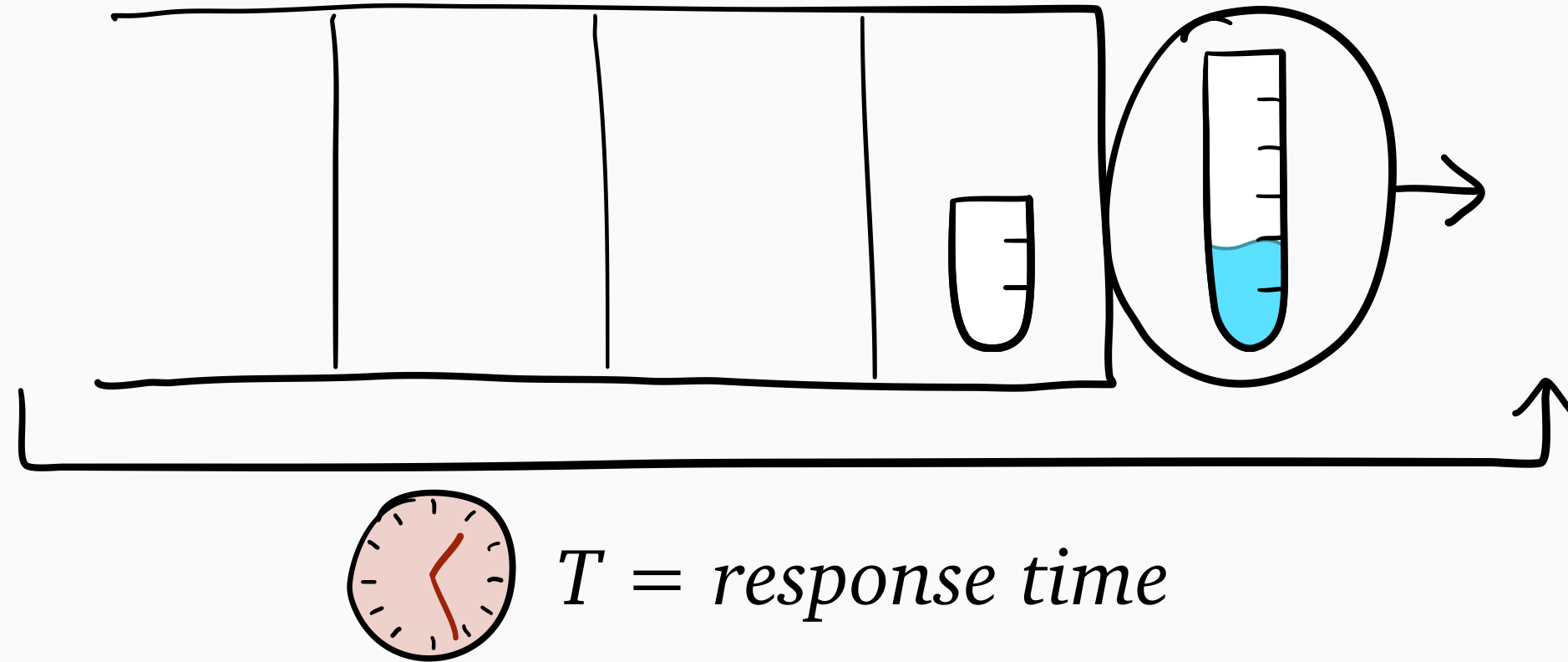


Serve short jobs
before long jobs

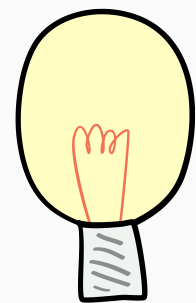
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

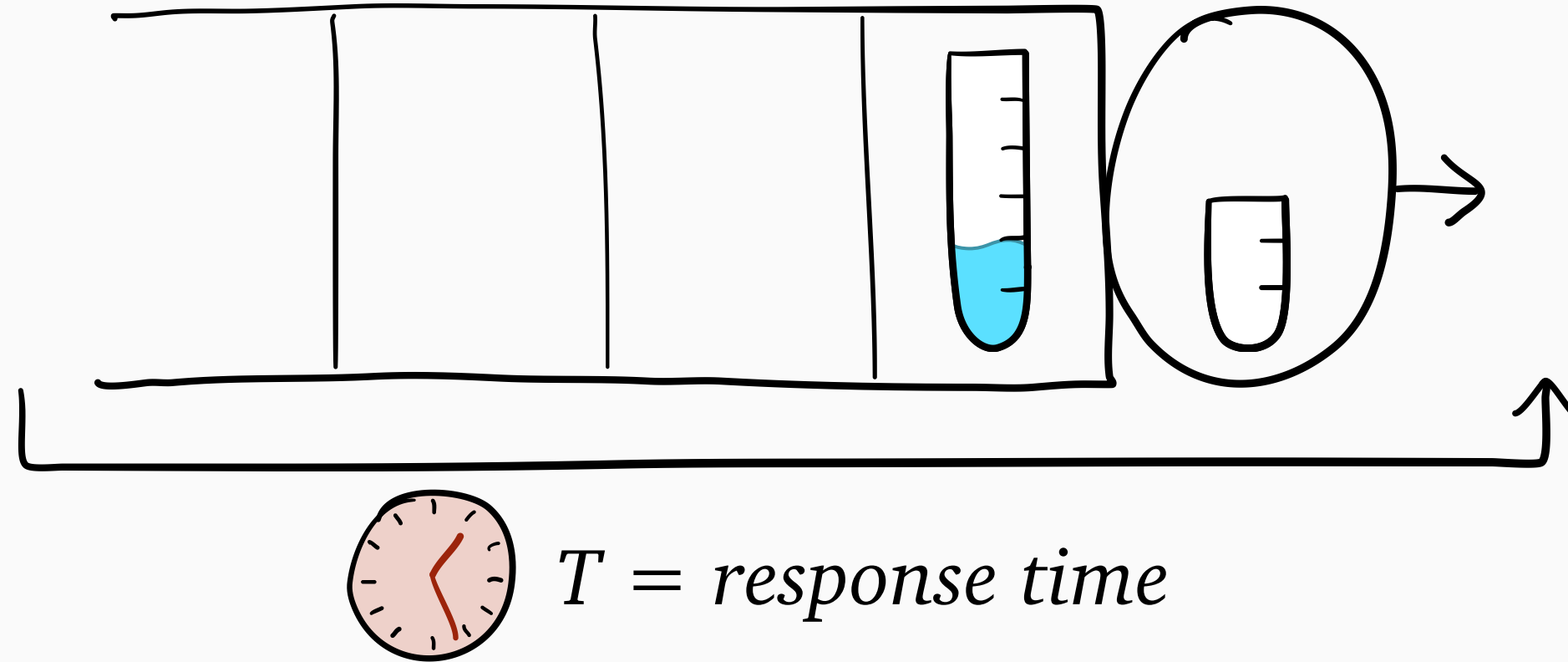


Serve short jobs
before long jobs

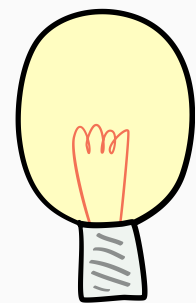
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

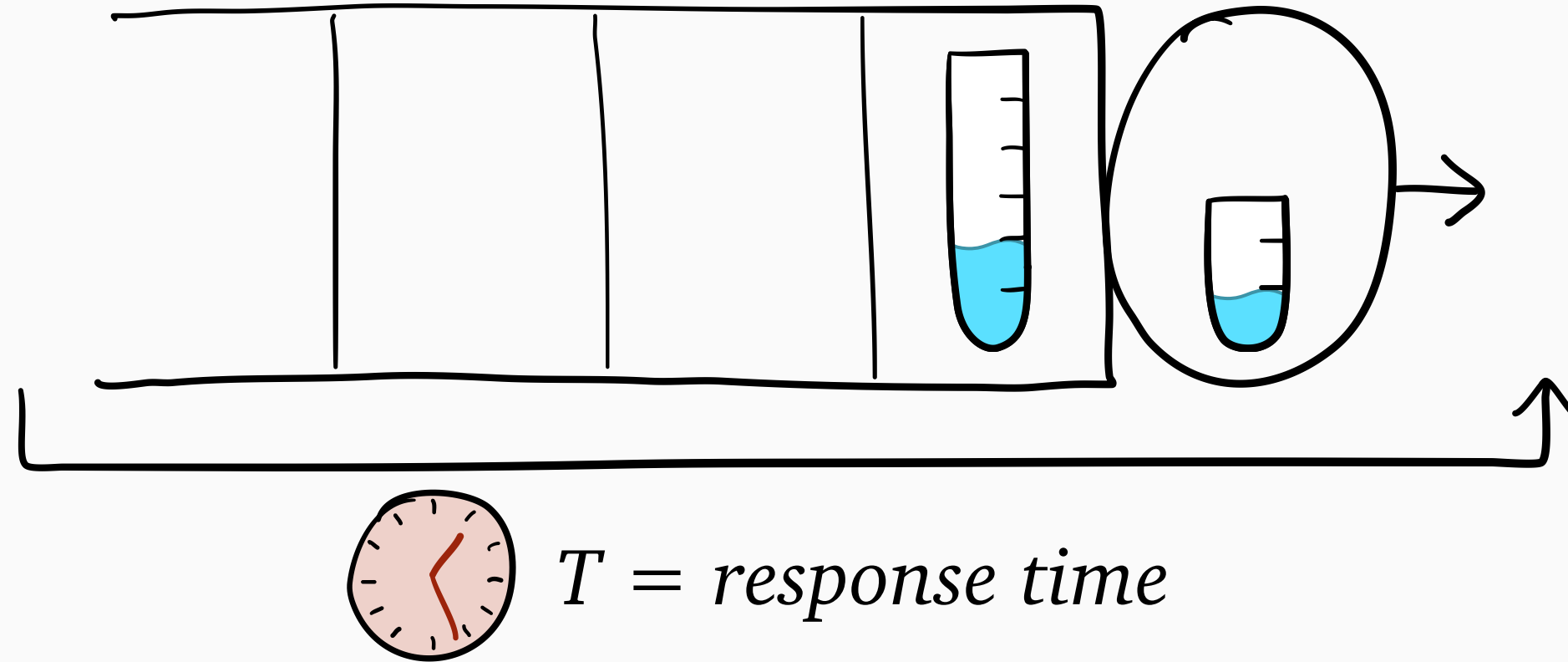


Serve short jobs
before long jobs

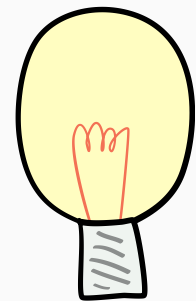
Mean scheduling in the M/G/1

M/G arrivals:

- arrival rate λ
- size distribution Z



Minimize $E[T]$?

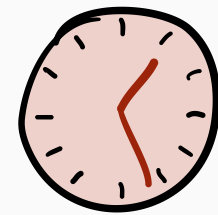
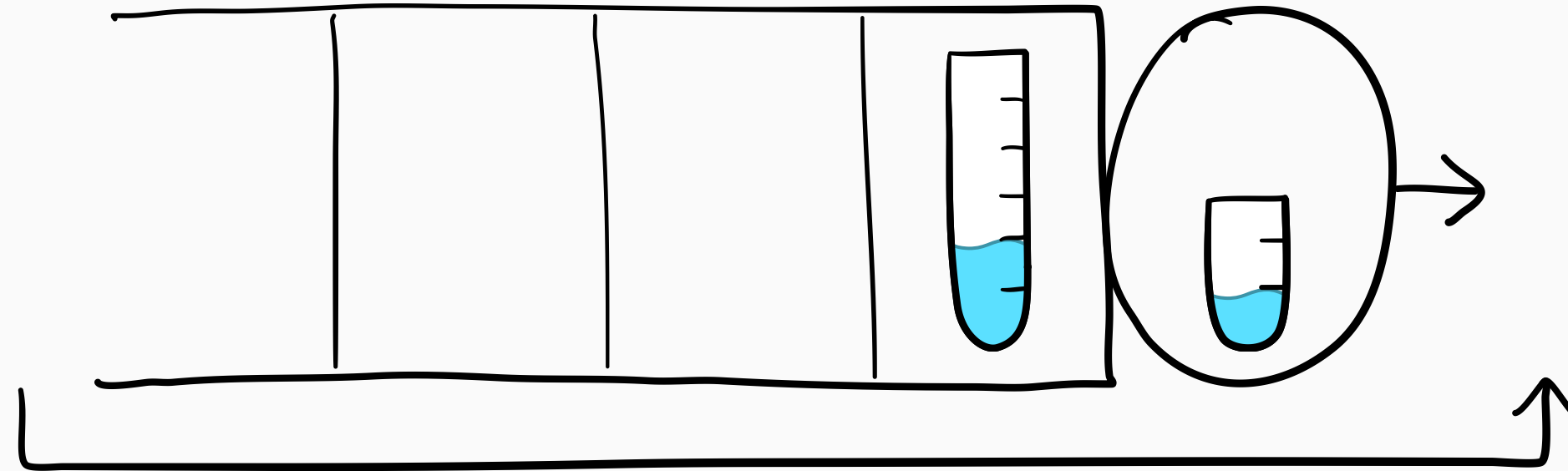


Serve short jobs
before long jobs

Mean scheduling in the M/G/1

M/G arrivals:

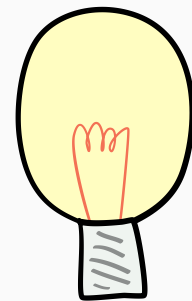
- arrival rate λ
- size distribution Z



$T = \text{response time}$



Minimize $E[T]$?



Serve short jobs
before long jobs



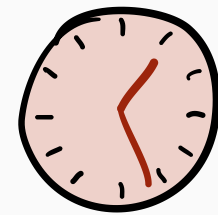
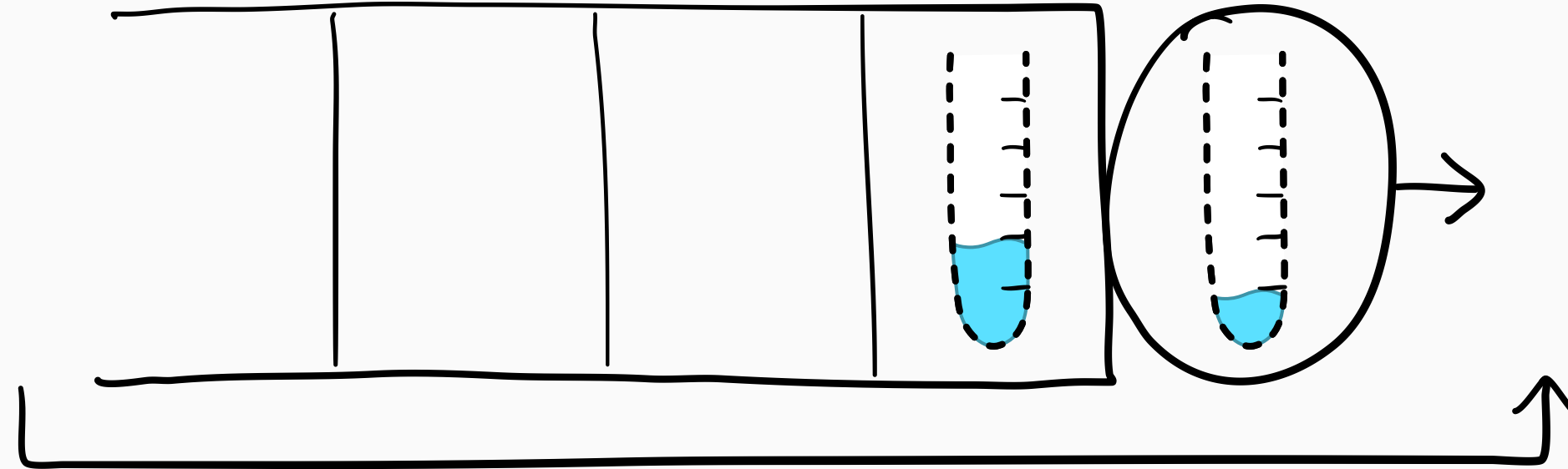
SRPT: minimizes $E[T]$

shortest remaining
processing time

Unknown job sizes

M/G arrivals:

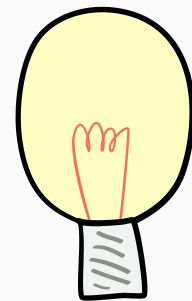
- arrival rate λ
- size distribution Z



$T = \text{response time}$



Minimize $E[T]$?



Serve short jobs
before long jobs



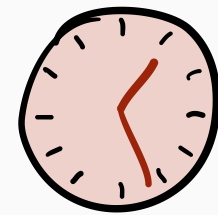
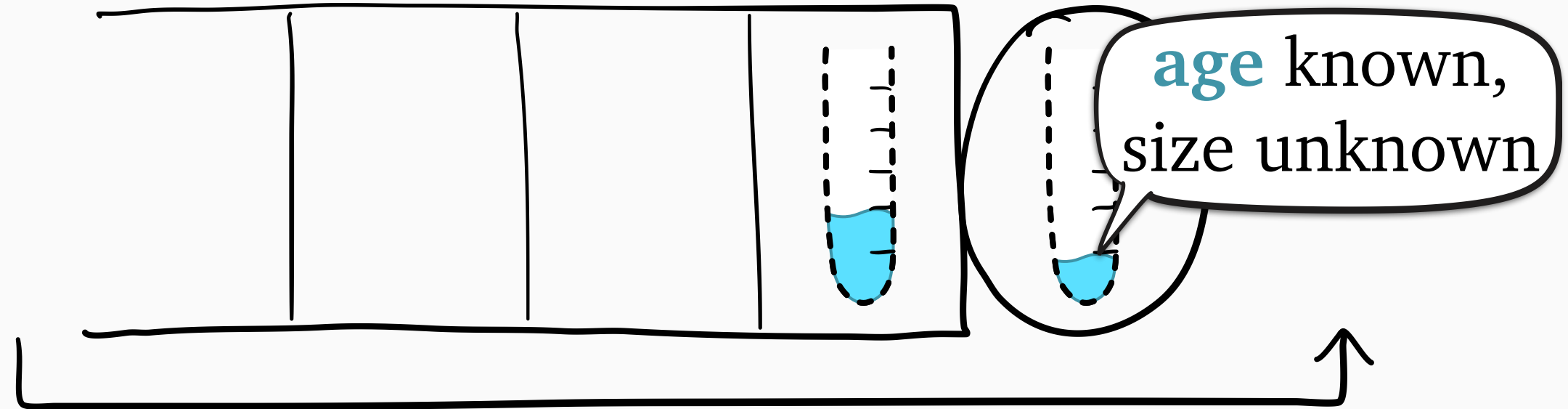
SRPT: minimizes $E[T]$

shortest remaining
processing time

Unknown job sizes

M/G arrivals:

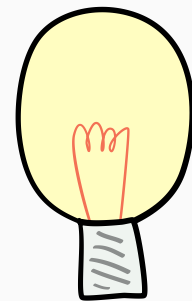
- arrival rate λ
- size distribution Z



$T = \text{response time}$



Minimize $E[T]$?



Serve short jobs
before long jobs



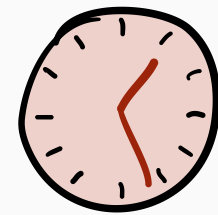
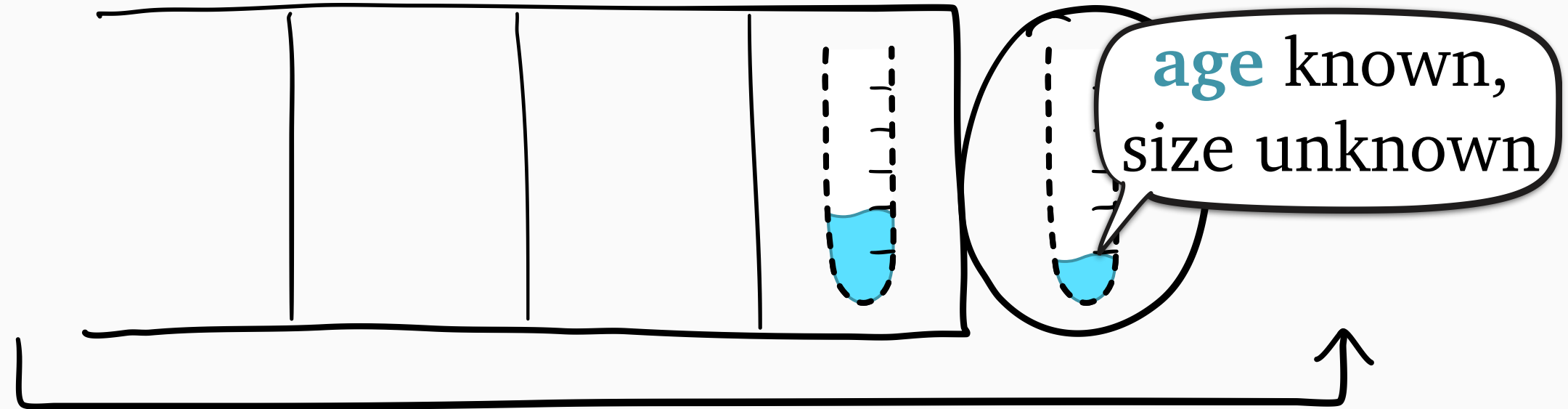
shortest remaining
processing time

SRPT: minimizes $E[T]$

Unknown job sizes

M/G arrivals:

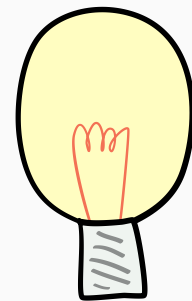
- arrival rate λ
- size distribution Z



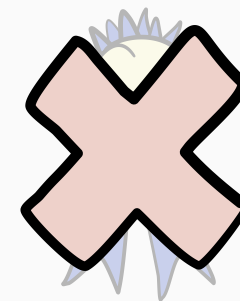
$T = \text{response time}$



Minimize $E[T]$?



Serve short jobs
before long jobs



shortest remaining
processing time

SRPT: minimizes $E[T]$

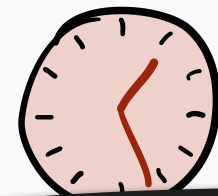
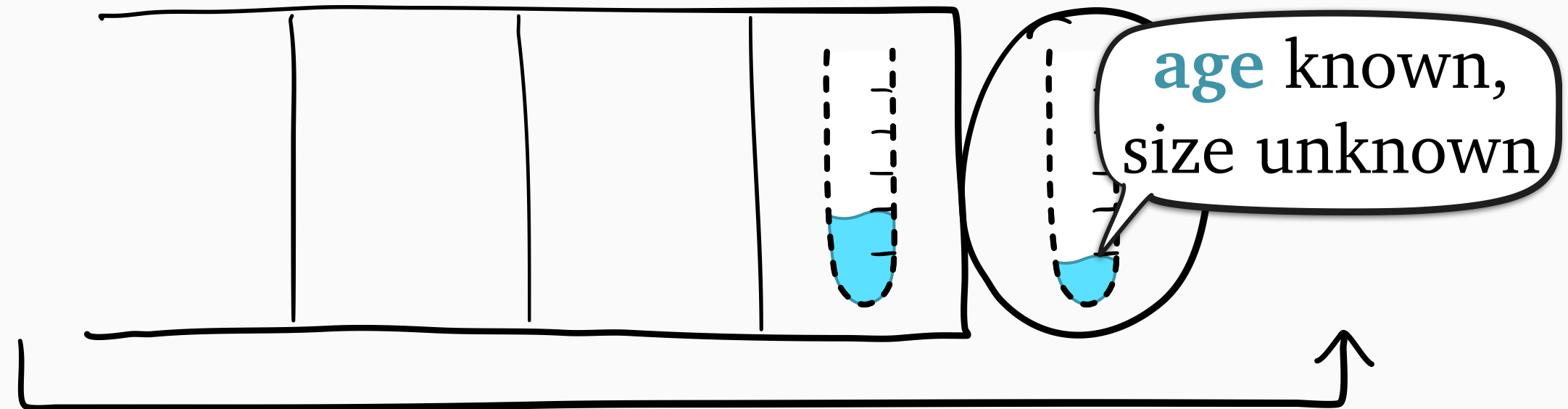
Unknown job sizes

M/G arrivals:

- arrival rate λ
- size distribution Z

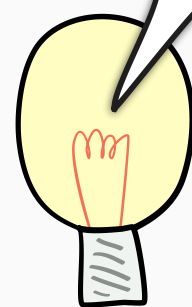


Minimize $E[T]$?

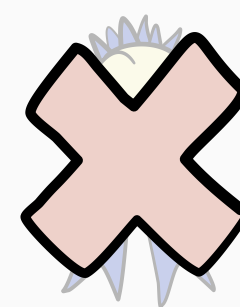


$T = \text{response time}$

... using **ages** and
size distribution Z



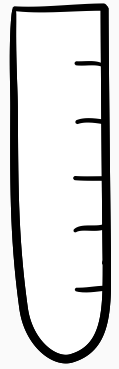
Serve short jobs
before long jobs



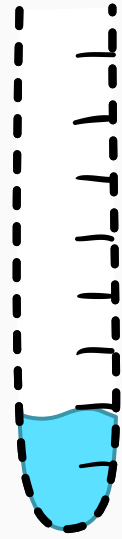
shortest remaining
processing time

SRPT: minimizes $E[T]$

Gittins index of a job

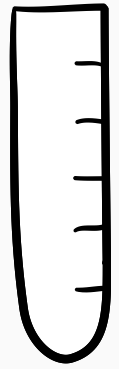


known
size r

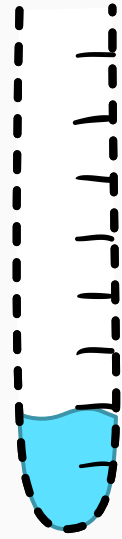


unknown size
 $(Z \mid Z > \text{age})$

Gittins index of a job



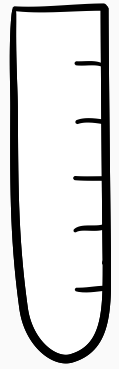
known
size r



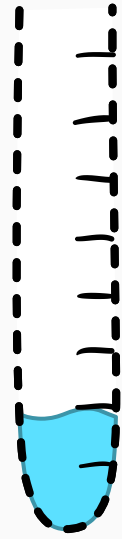
unknown size
($Z \mid Z > \text{age}$)



Gittins index of a job



known
size r

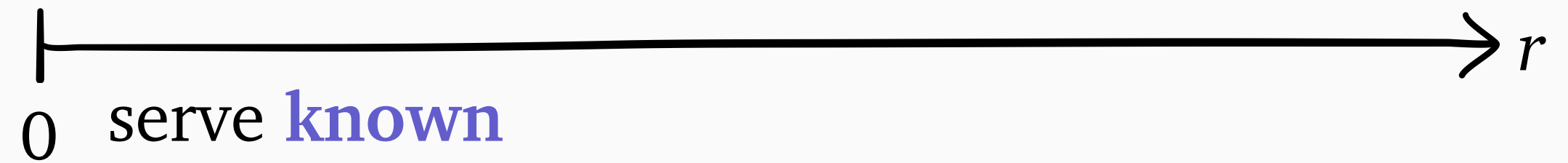
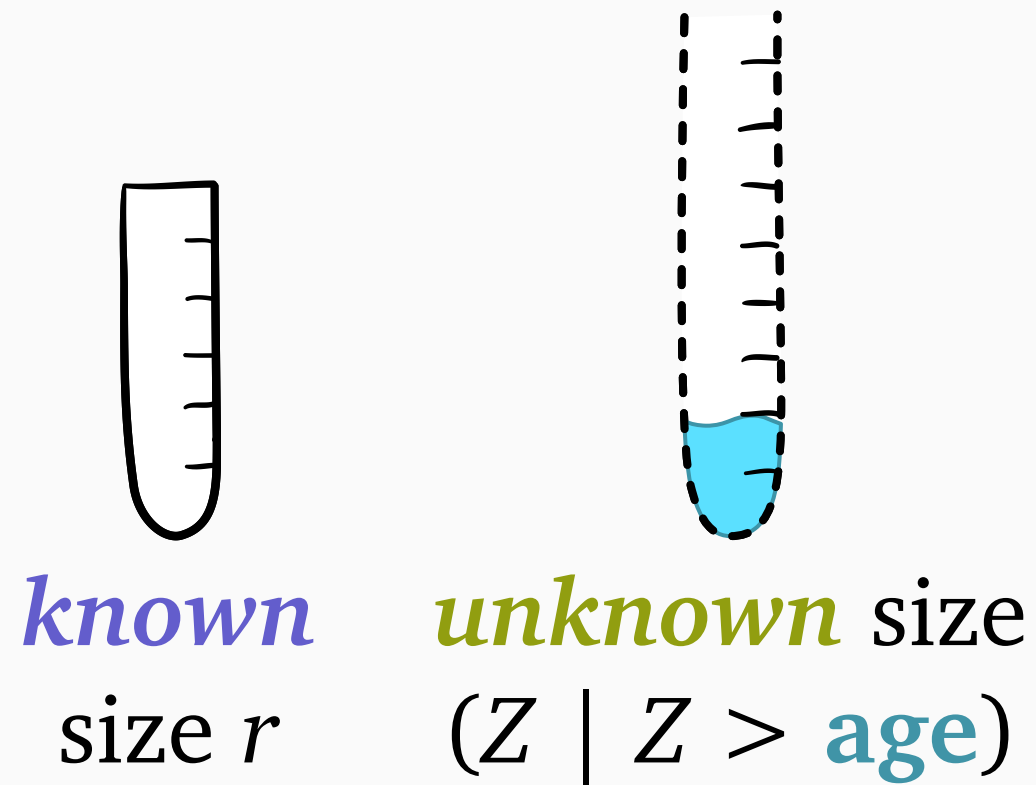


unknown size
($Z \mid Z > \text{age}$)



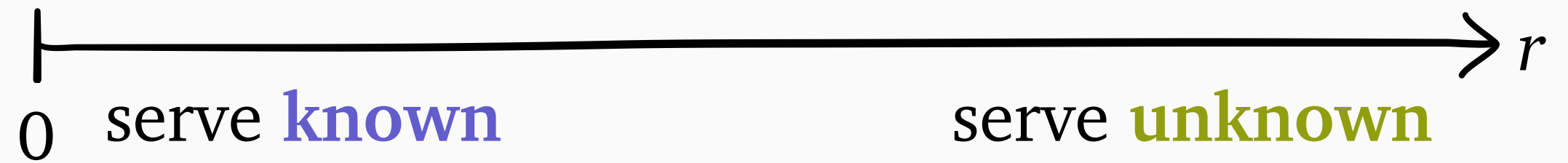
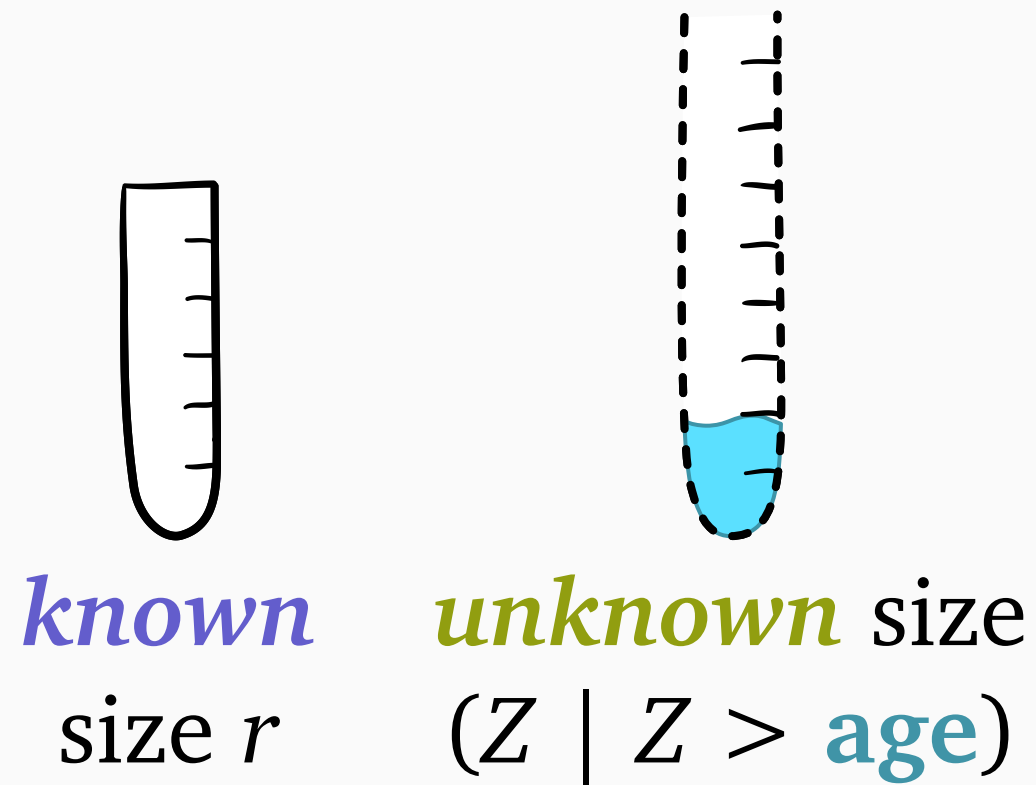
? Key question: for which r
should we serve *unknown*?

Gittins index of a job



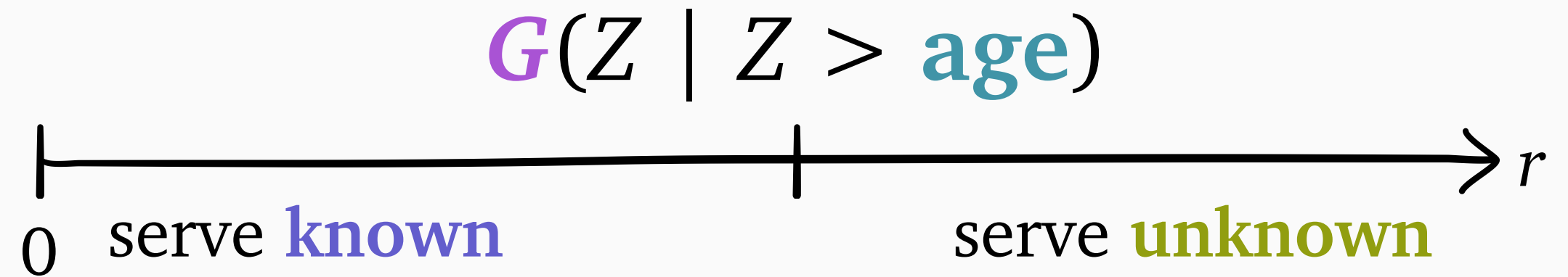
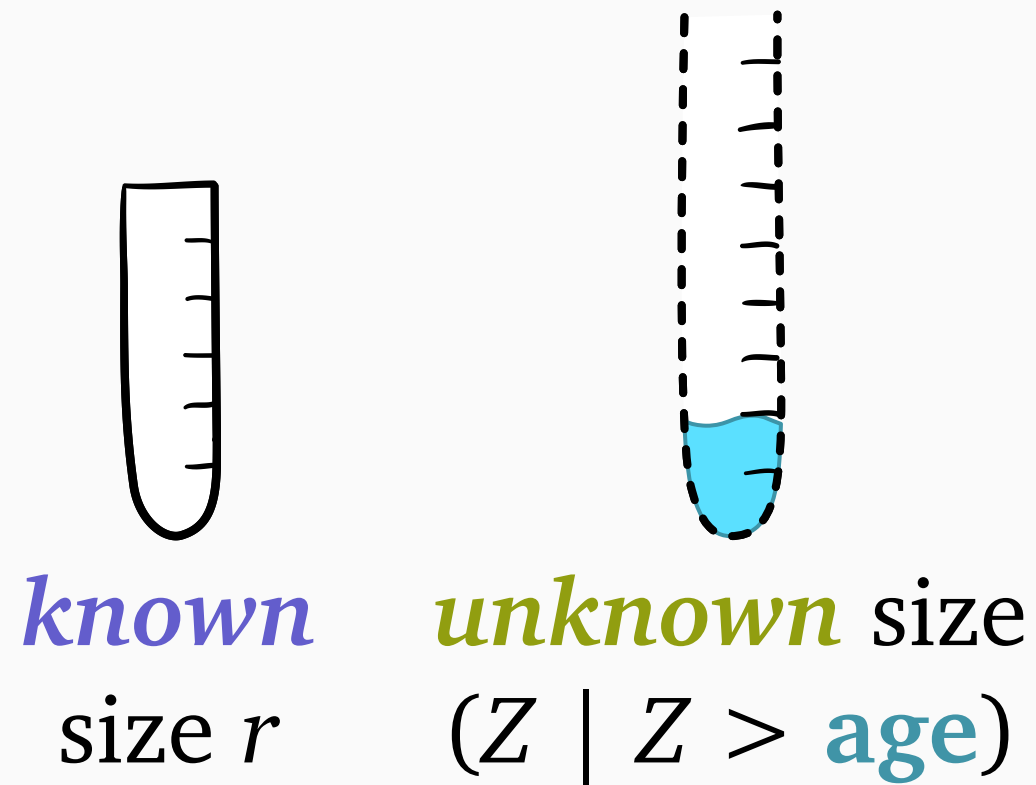
? Key question: for which r should we serve *unknown*?

Gittins index of a job



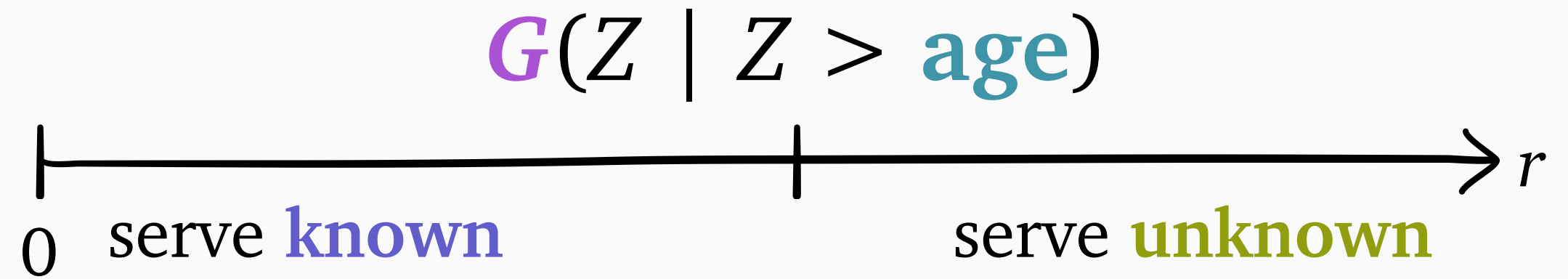
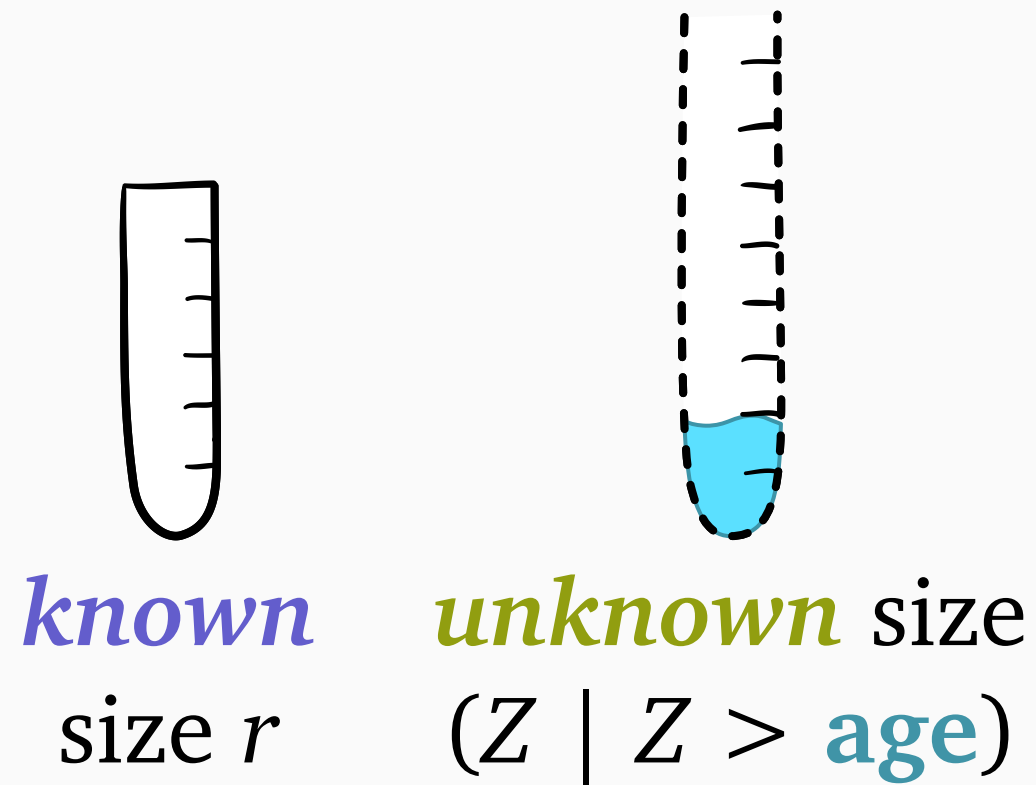
? Key question: for which r should we serve *unknown*?

Gittins index of a job



? Key question: for which r should we serve **unknown**?

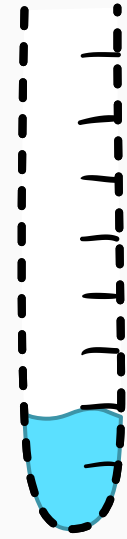
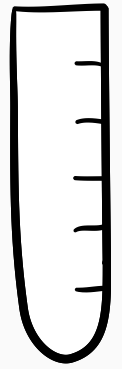
Gittins index of a job



1	(50%)
19	(50%)

? Key question: for which r should we serve **unknown**?

Gittins index of a job



known
size r

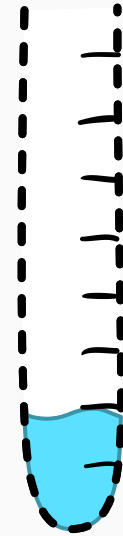
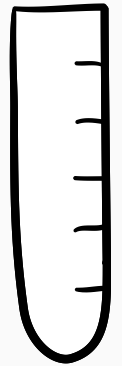
unknown size
($Z \mid Z > \text{age}$)



? Key question: for which r
should we serve *unknown*?

1	(50%)
19	(50%)

Gittins index of a job



known
size r

unknown size
($Z \mid Z > \text{age}$)

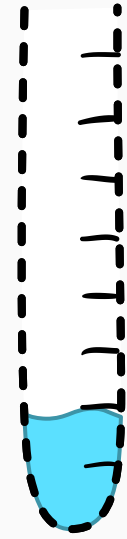
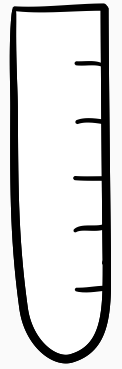


1 (50%)

19 (50%)

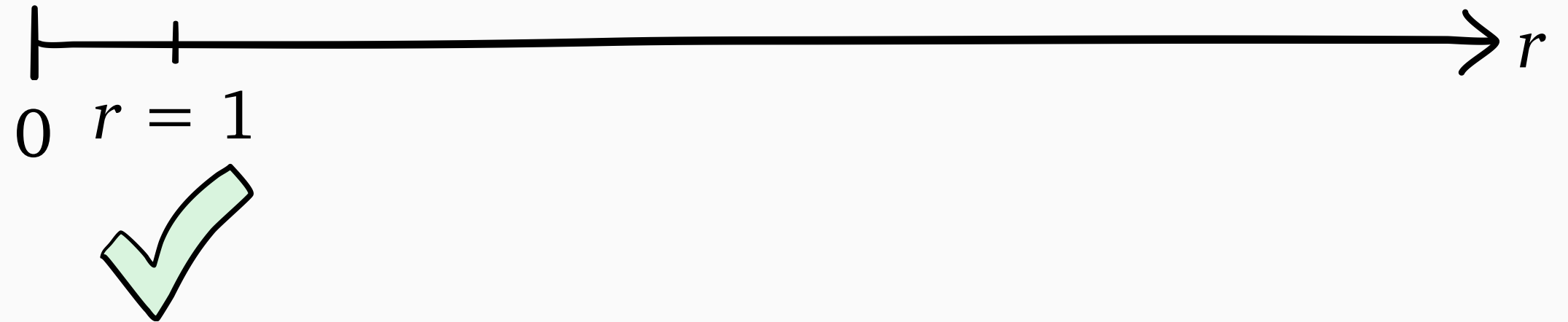
? Key question: for which r
should we serve *unknown*?

Gittins index of a job



known
size r

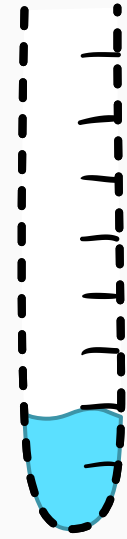
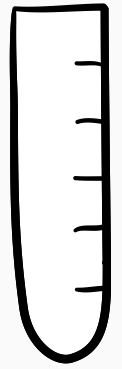
unknown size
($Z \mid Z > \text{age}$)



1	(50%)
19	(50%)

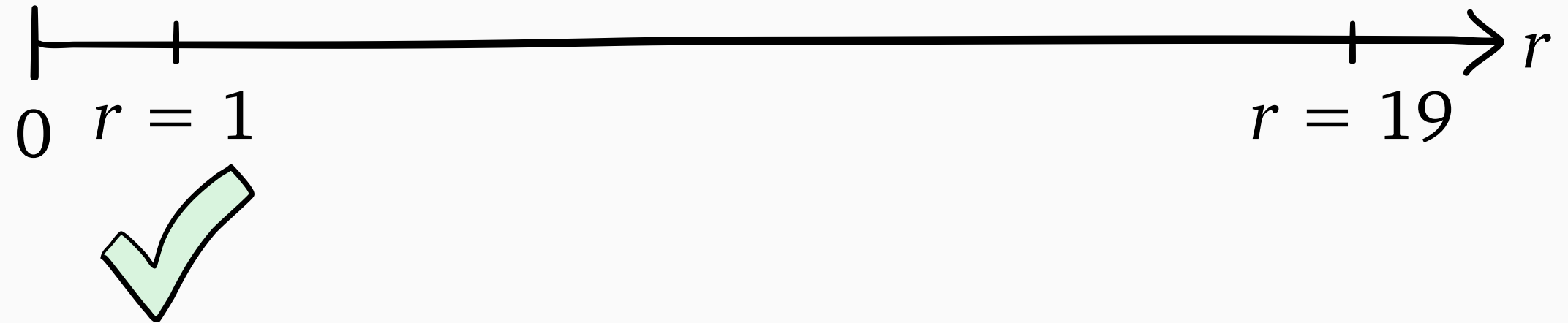
? Key question: for which r should we serve *unknown*?

Gittins index of a job



known
size r

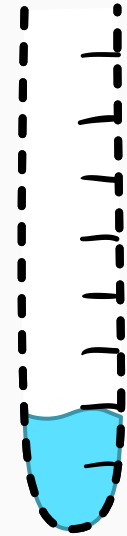
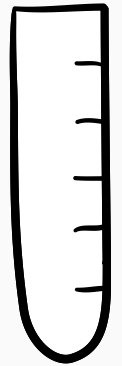
unknown size
($Z \mid Z > \text{age}$)



? Key question: for which r
should we serve *unknown*?

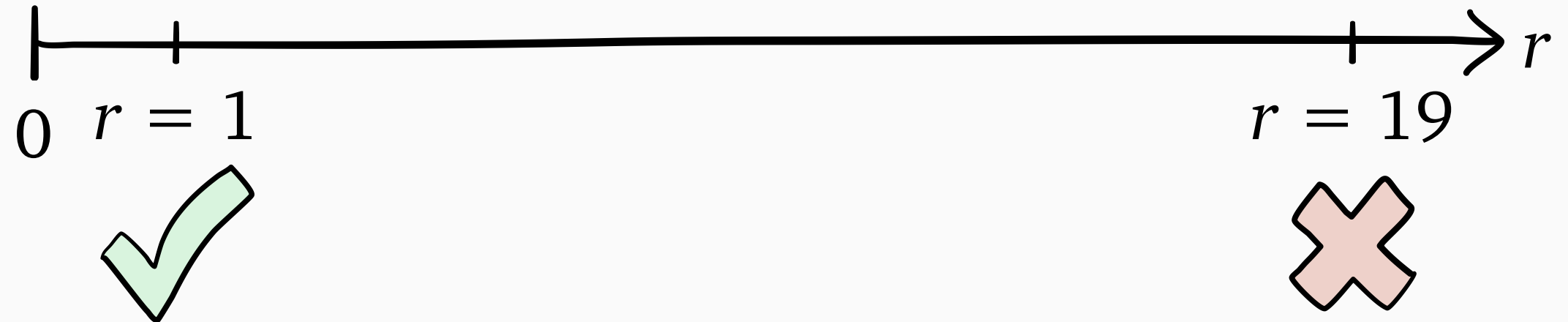
1	(50%)
19	(50%)

Gittins index of a job



known
size r

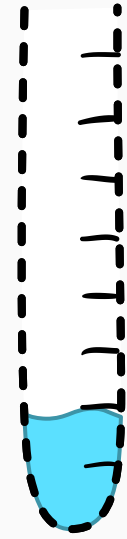
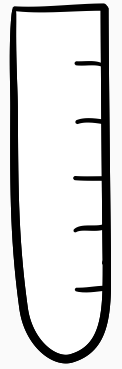
unknown size
($Z \mid Z > \text{age}$)



1 (50%)
19 (50%)

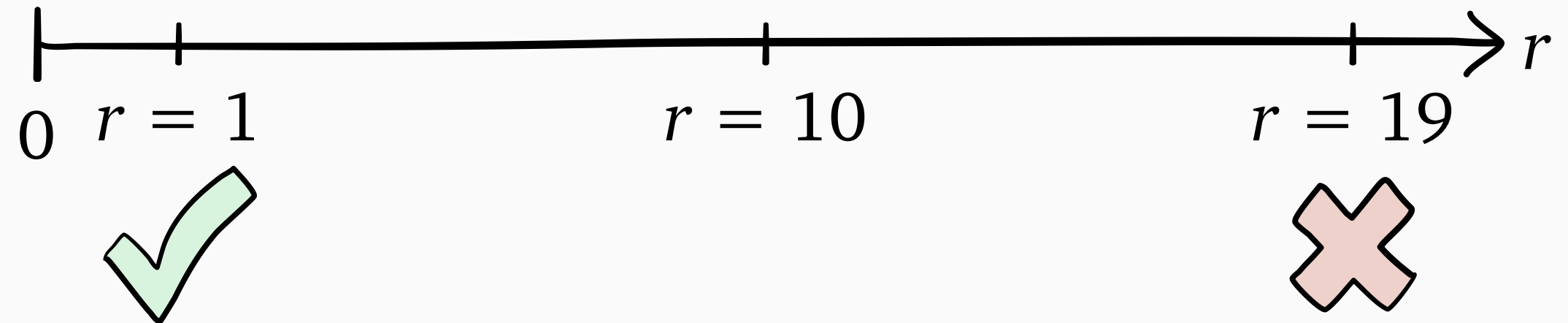
? Key question: for which r
should we serve *unknown*?

Gittins index of a job



known
size r

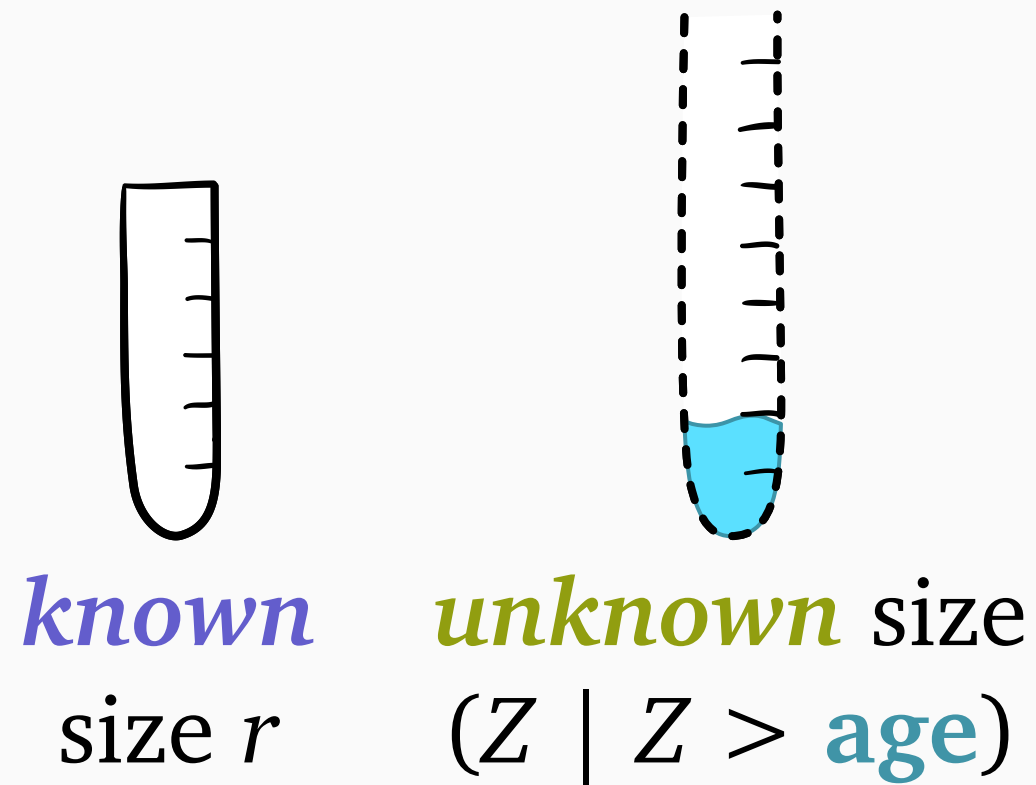
unknown size
($Z \mid Z > \text{age}$)



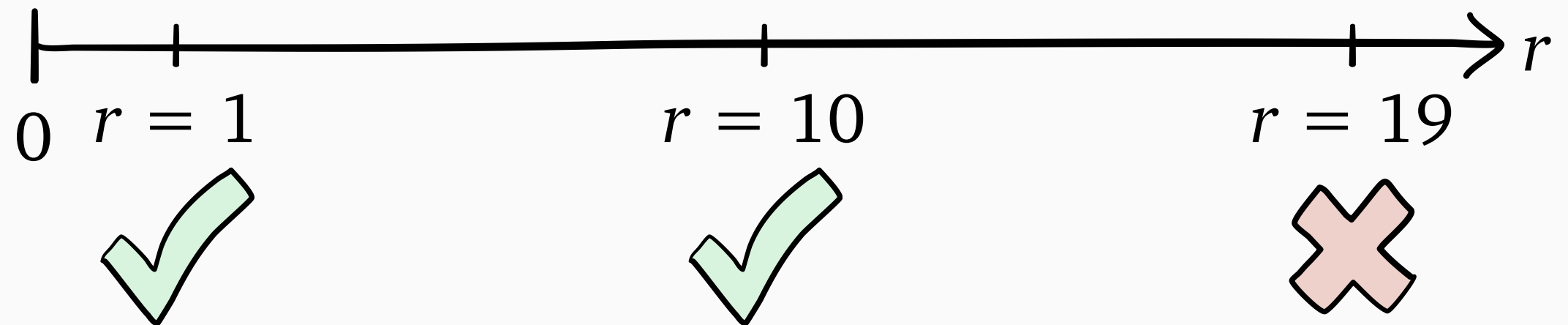
1 (50%)
19 (50%)

? Key question: for which r
should we serve *unknown*?

Gittins index of a job

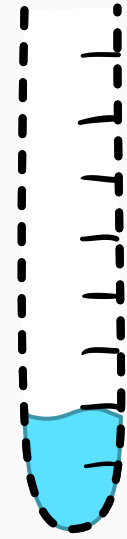
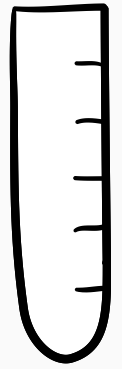


1	(50%)
19	(50%)



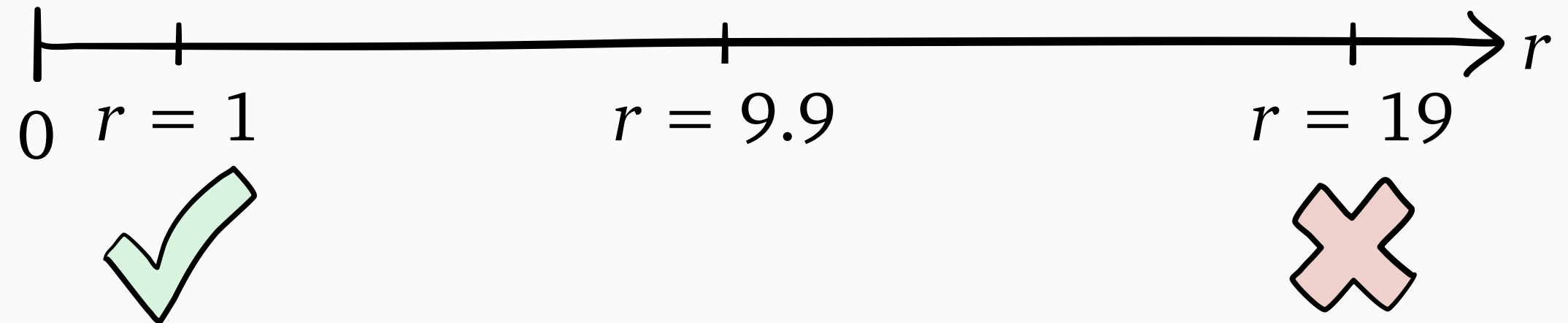
? Key question: for which r should we serve **unknown**?

Gittins index of a job



known
size r

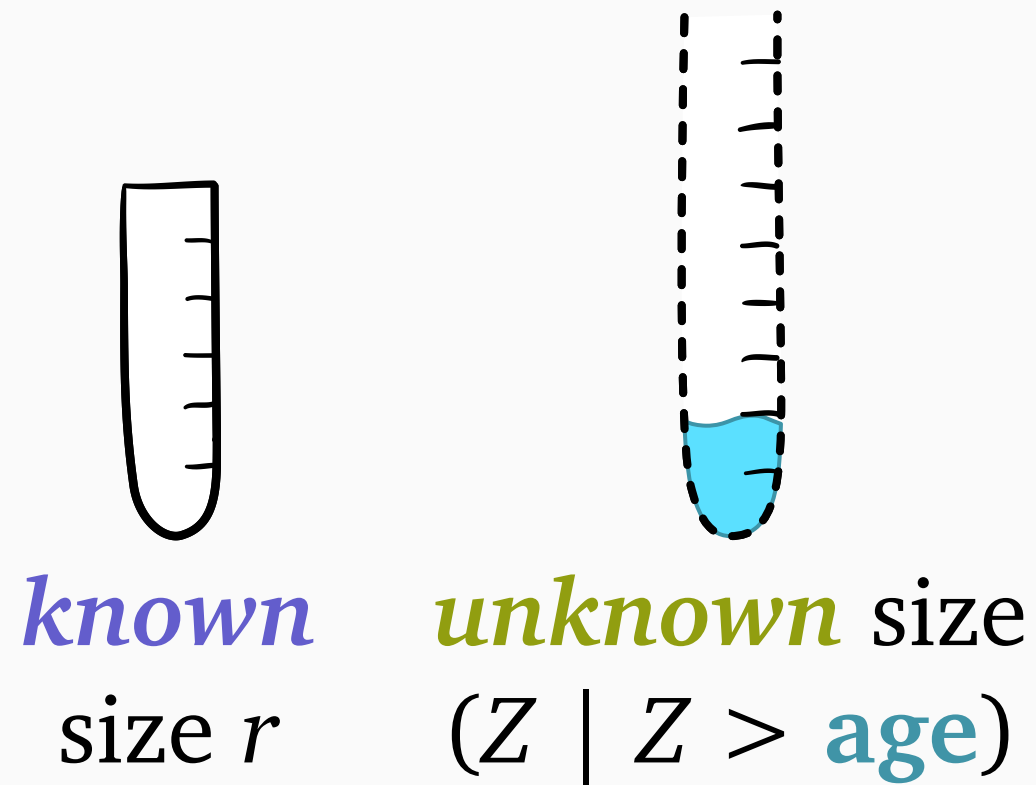
unknown size
($Z \mid Z > \text{age}$)



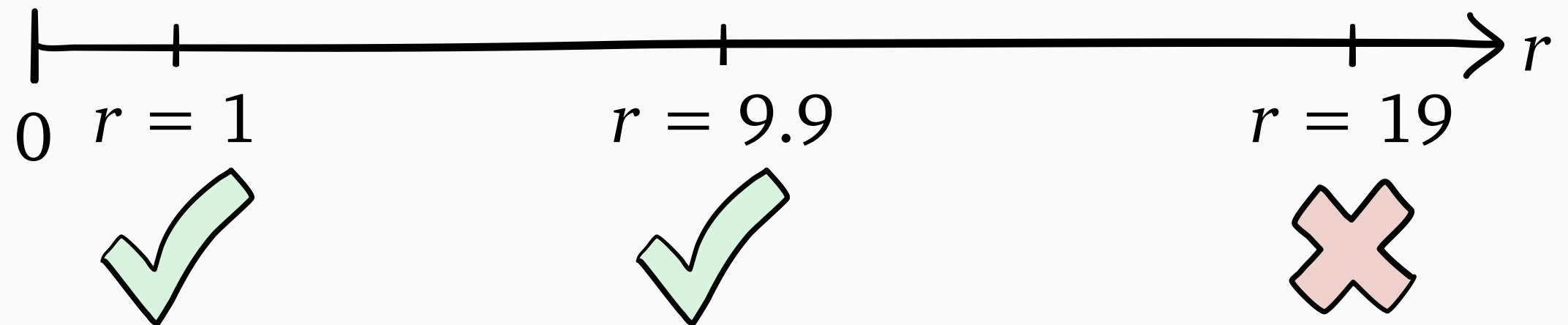
? Key question: for which r
should we serve *unknown*?

1 (50%)
19 (50%)

Gittins index of a job

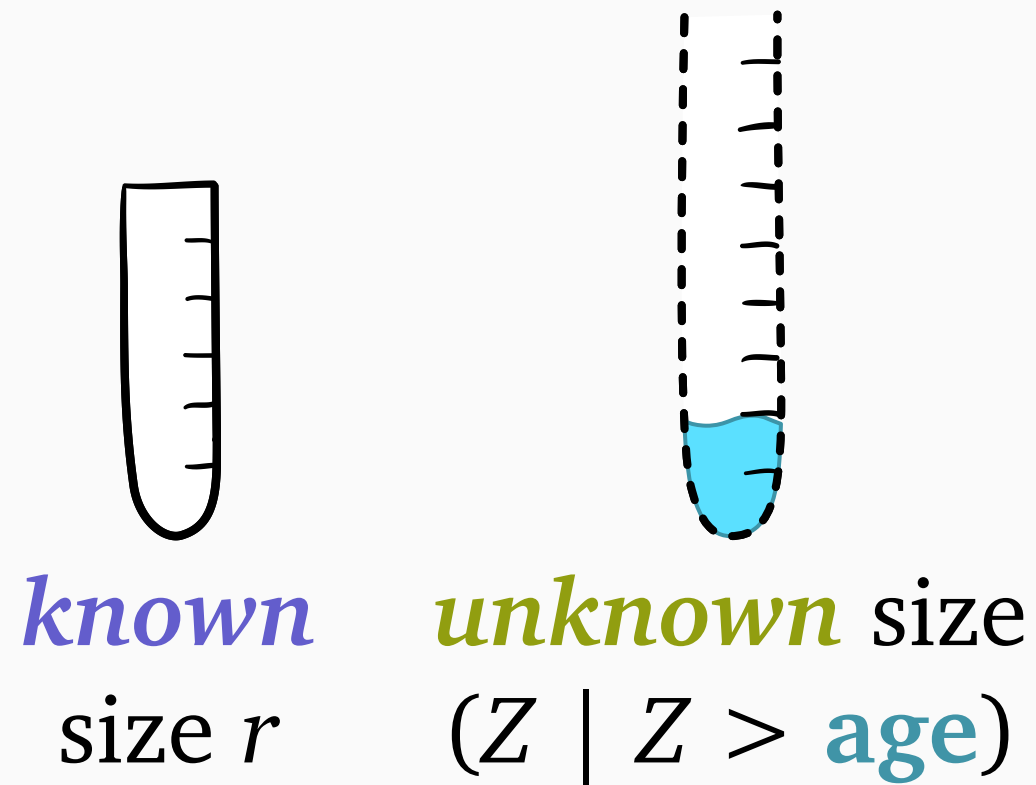


1	(50%)
19	(50%)

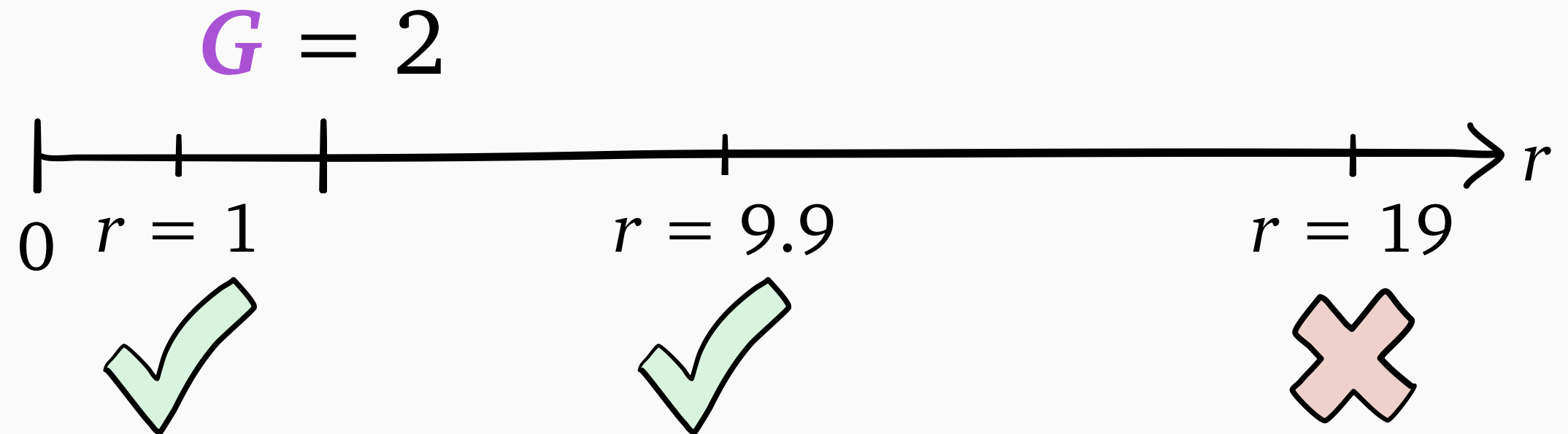


? Key question: for which r should we serve **unknown**?

Gittins index of a job

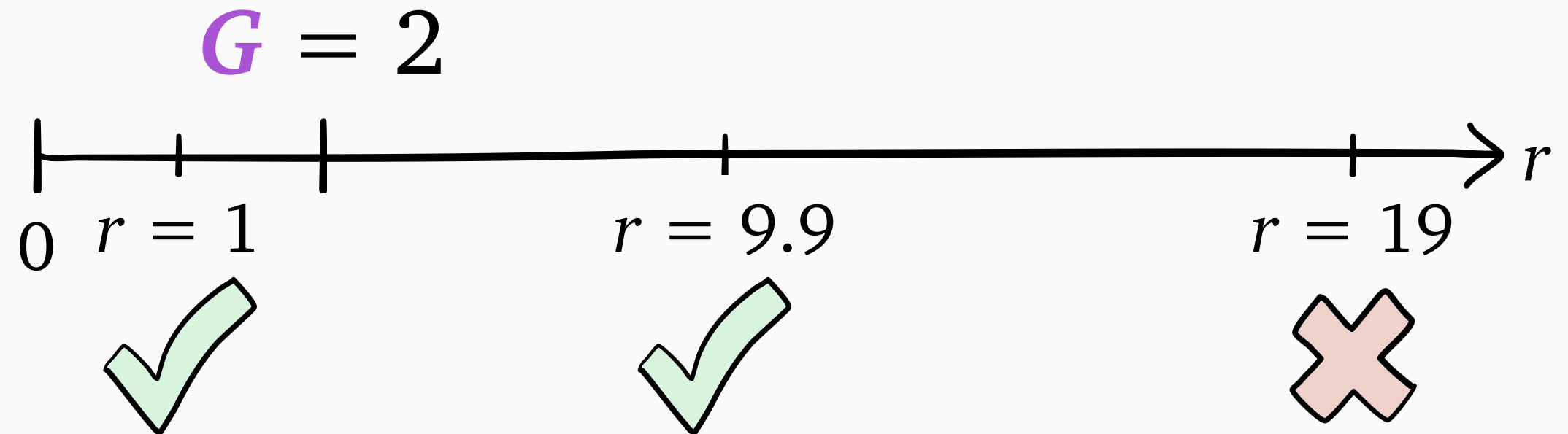
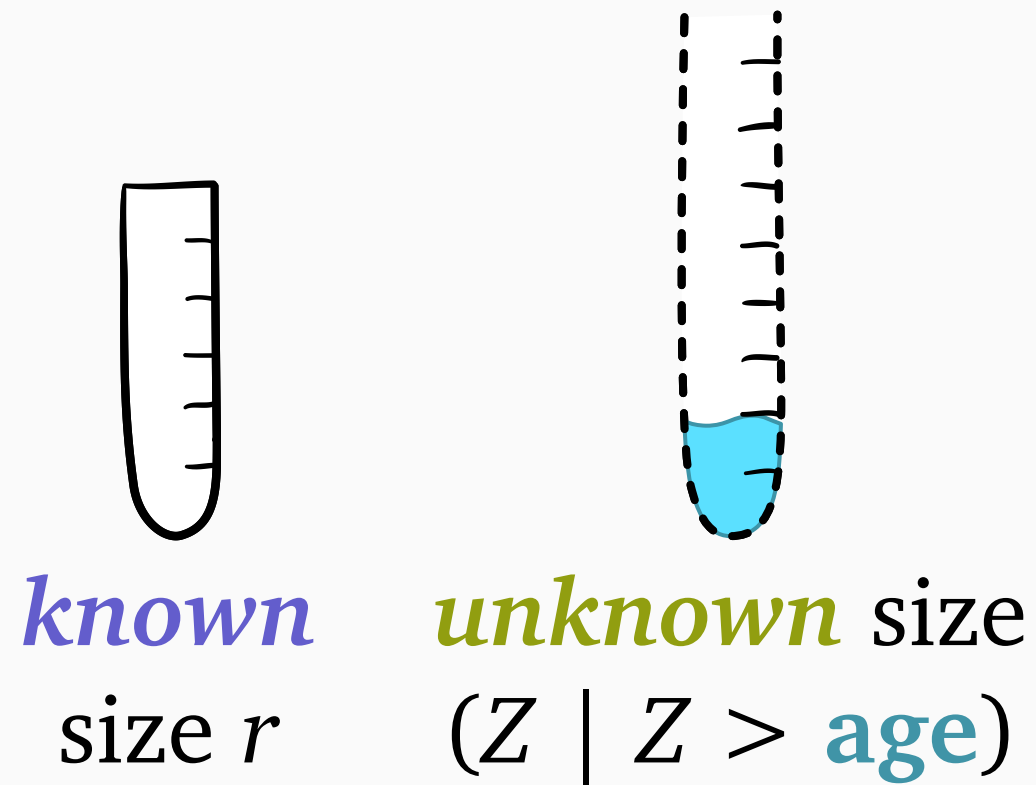


1	(50%)
19	(50%)



? Key question: for which r should we serve **unknown**?

Gittins index of a job



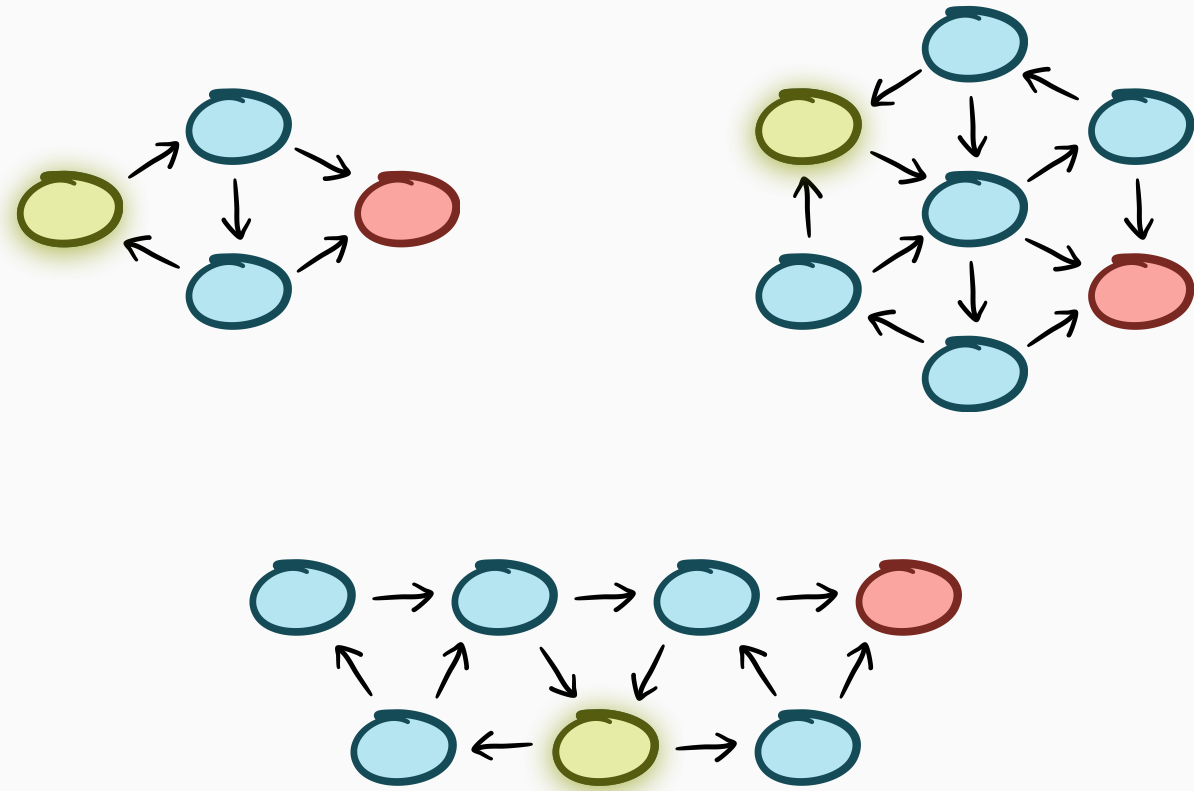
1 (50%)
19 (50%)

0 (50%)
18 (50%)

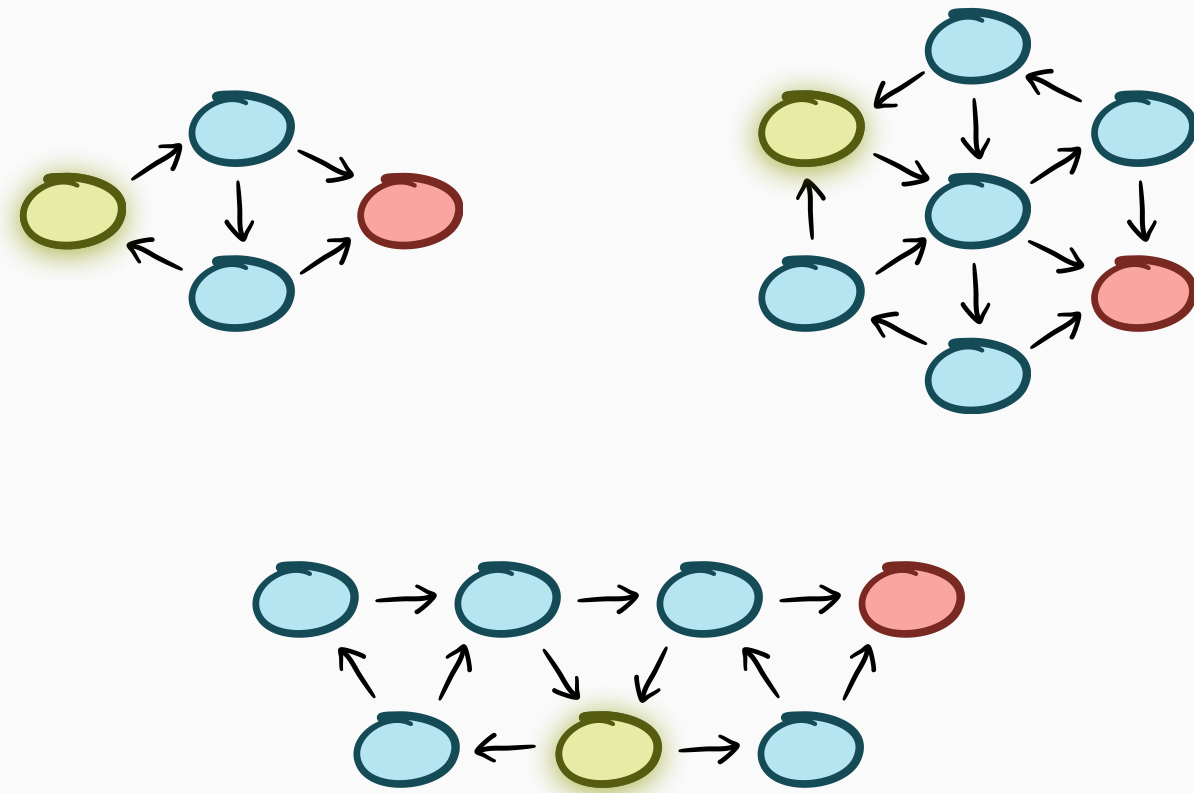
1:Y

? Key question: for which r should we serve **unknown**?

Job scheduling as MCS



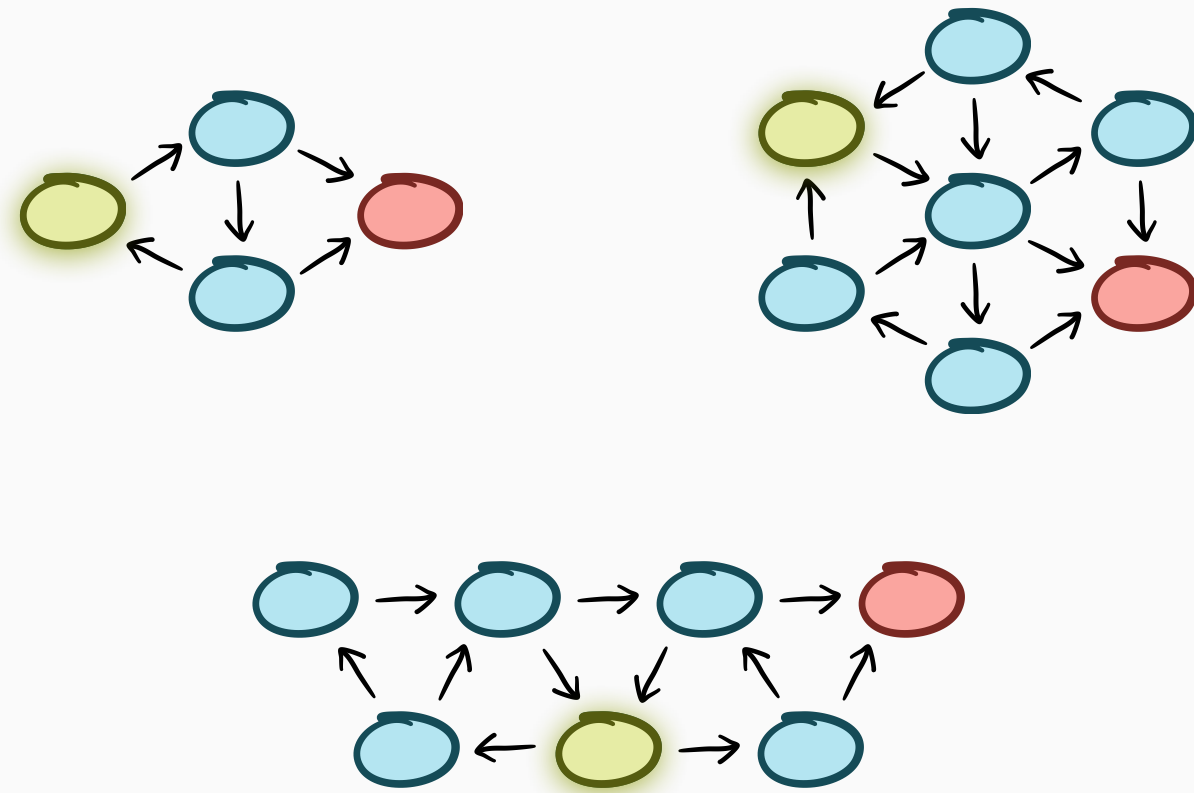
Job scheduling as MCS



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Job scheduling as MCS

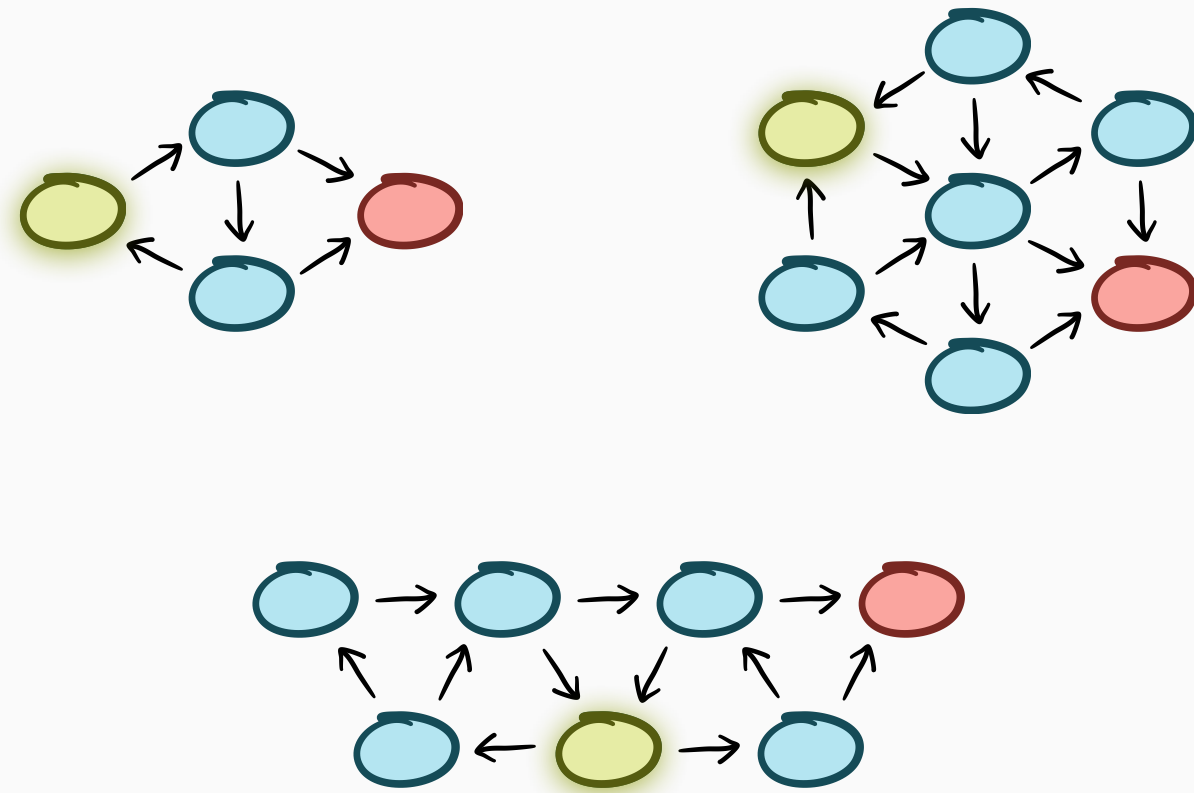


Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$

Job scheduling as MCS



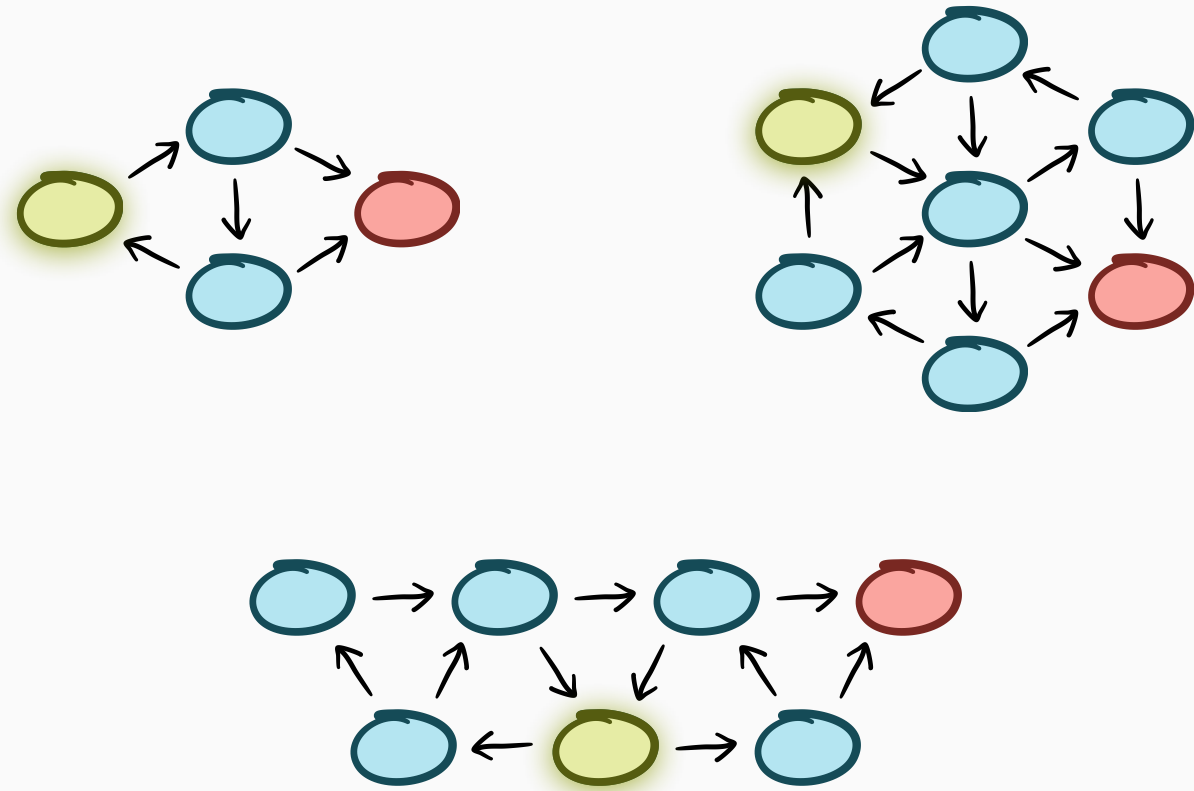
Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$

Job scheduling:

Job scheduling as MCS



Decision process:

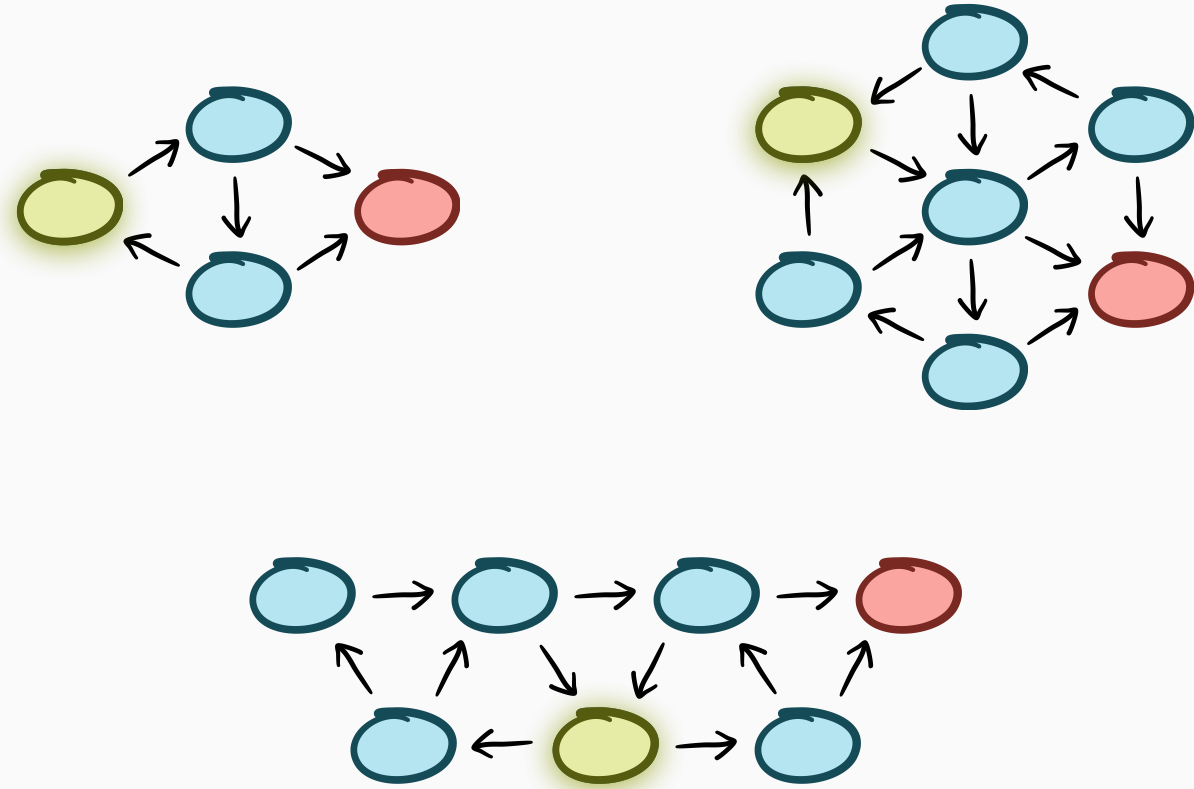
- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$

Job scheduling:

- All transitions incur reward -1

Job scheduling as MCS



Decision process:

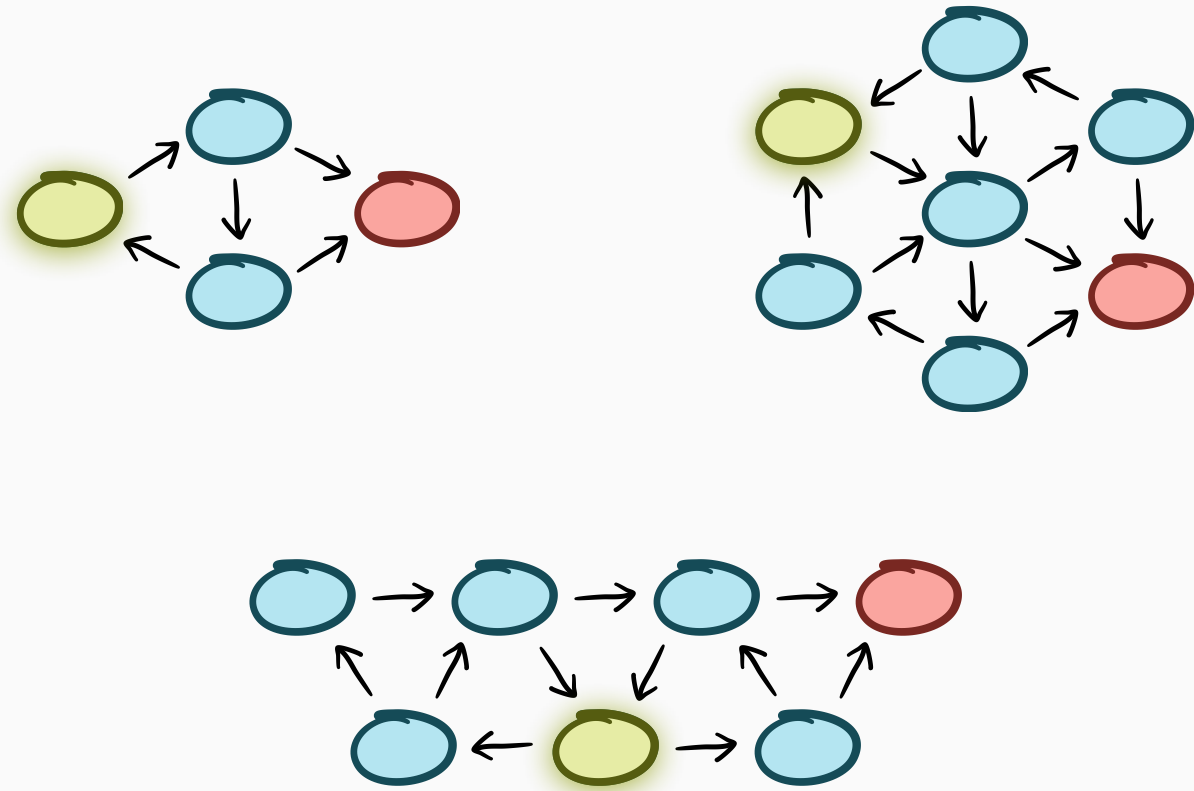
- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$

Job scheduling:

- All transitions incur reward -1
- **Terminal state** = “job complete”

Job scheduling as MCS



Decision process:

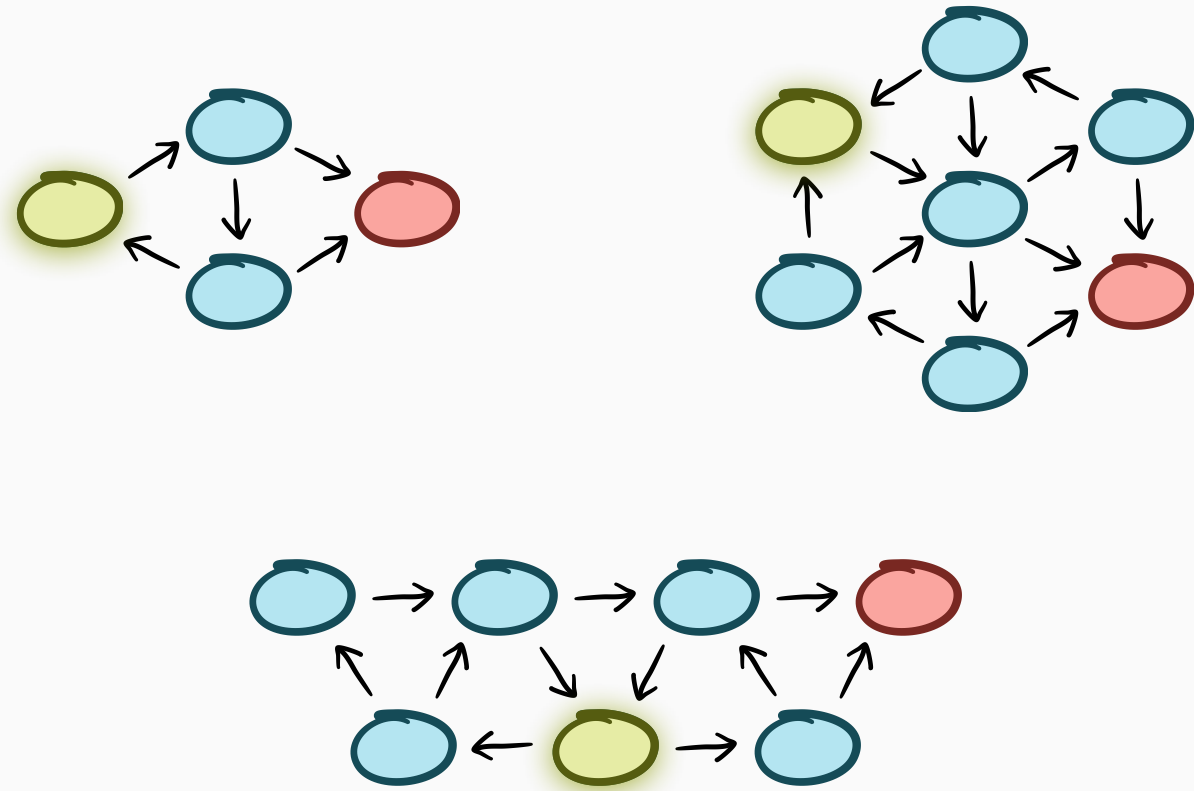
- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$

Job scheduling:

- All transitions incur reward -1
- **Terminal state** = “job complete”
- Minimizing $E[\text{time to first completion}]$

Job scheduling as MCS



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

Goal: maximize $E[\text{total reward}]$

Job scheduling:

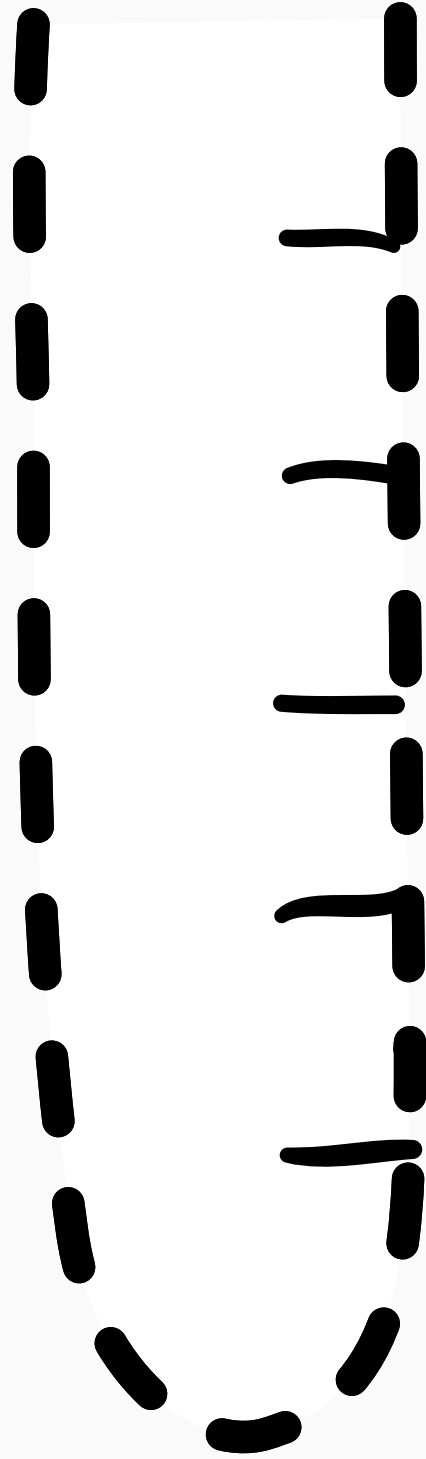
- All transitions incur reward -1
- **Terminal state** = “job complete”
- Minimizing $E[\text{time to first completion}]$

Want to minimize, for all k ,
 $E[\text{time to } k\text{th completion}]$

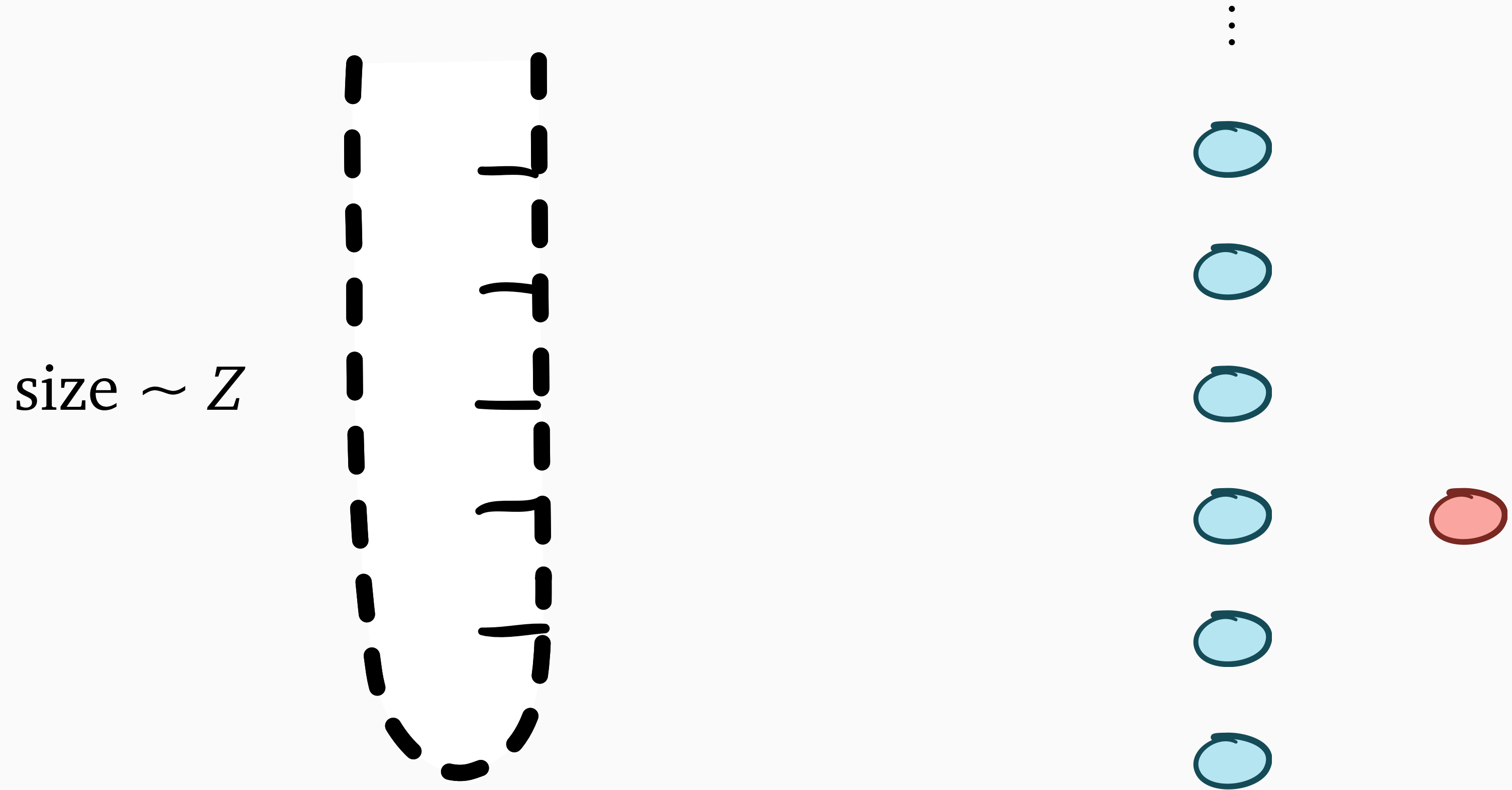
Jobs as Markov chains

Jobs as Markov chains

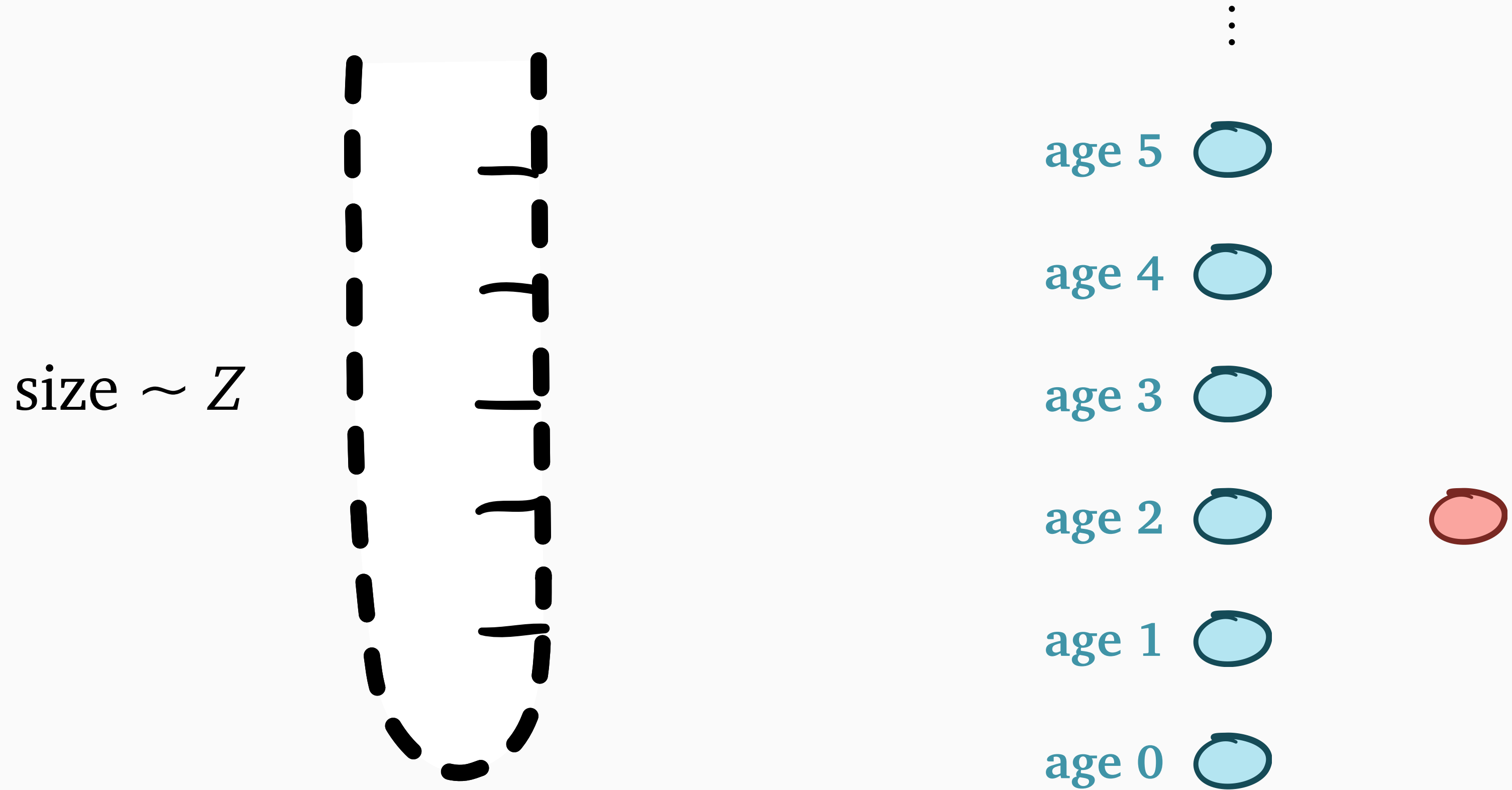
size $\sim Z$



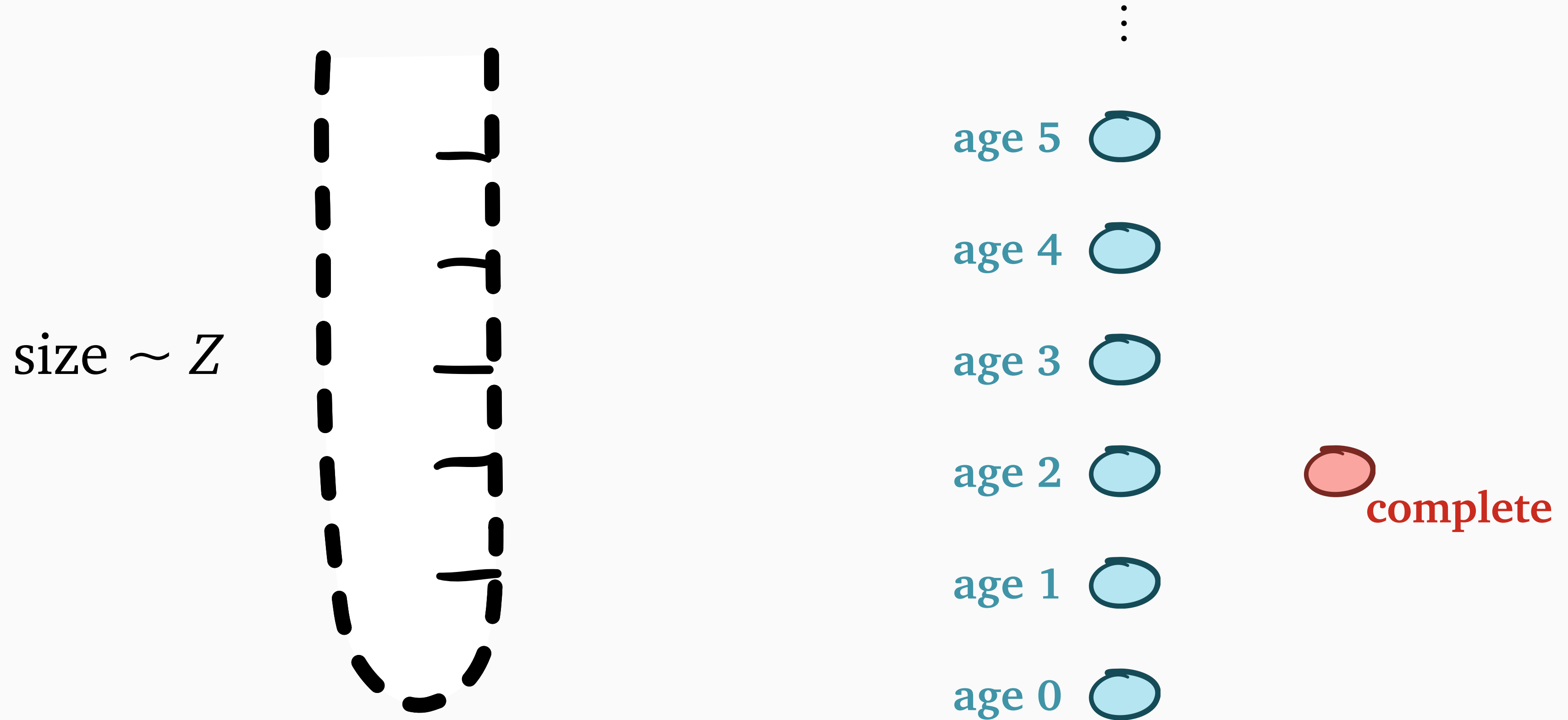
Jobs as Markov chains



Jobs as Markov chains

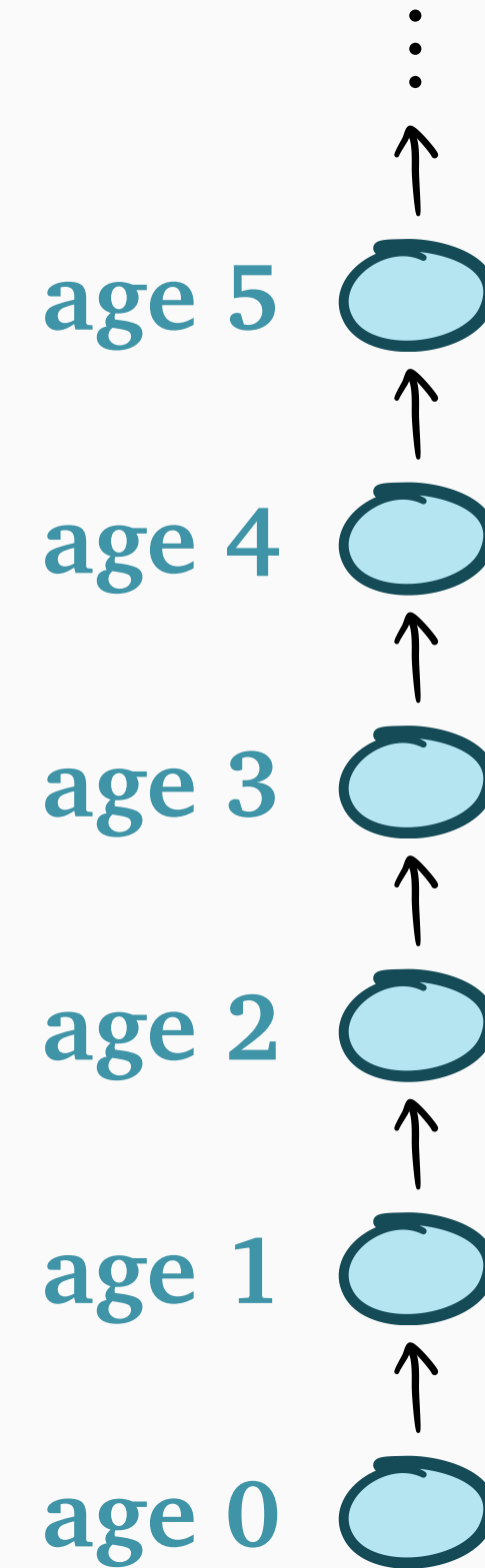
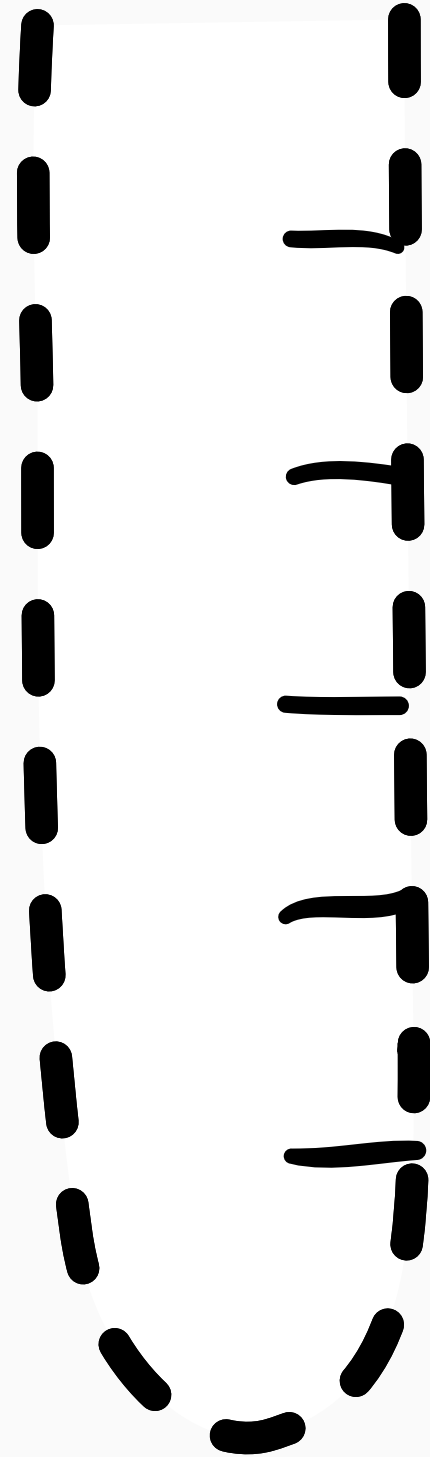


Jobs as Markov chains



Jobs as Markov chains

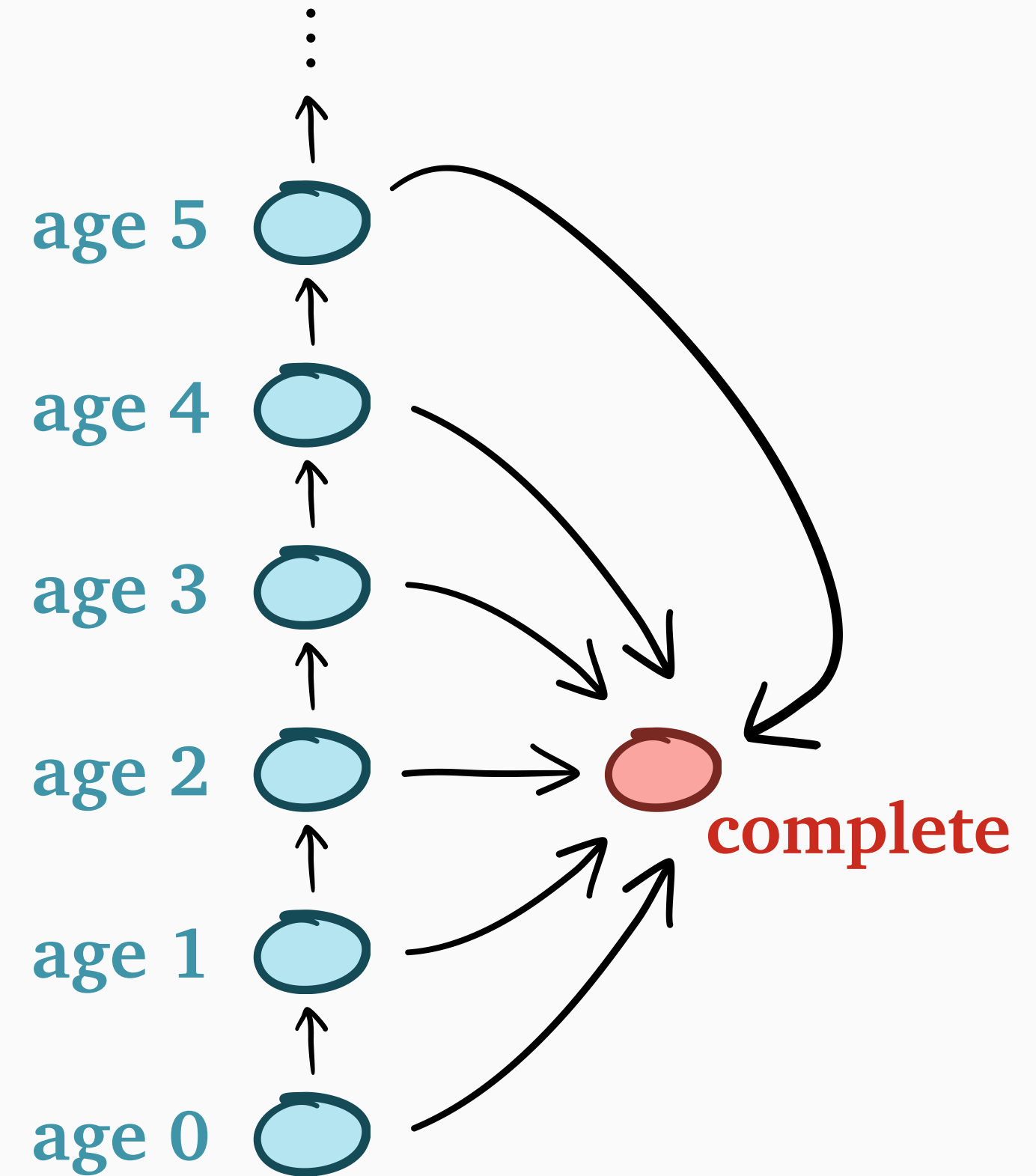
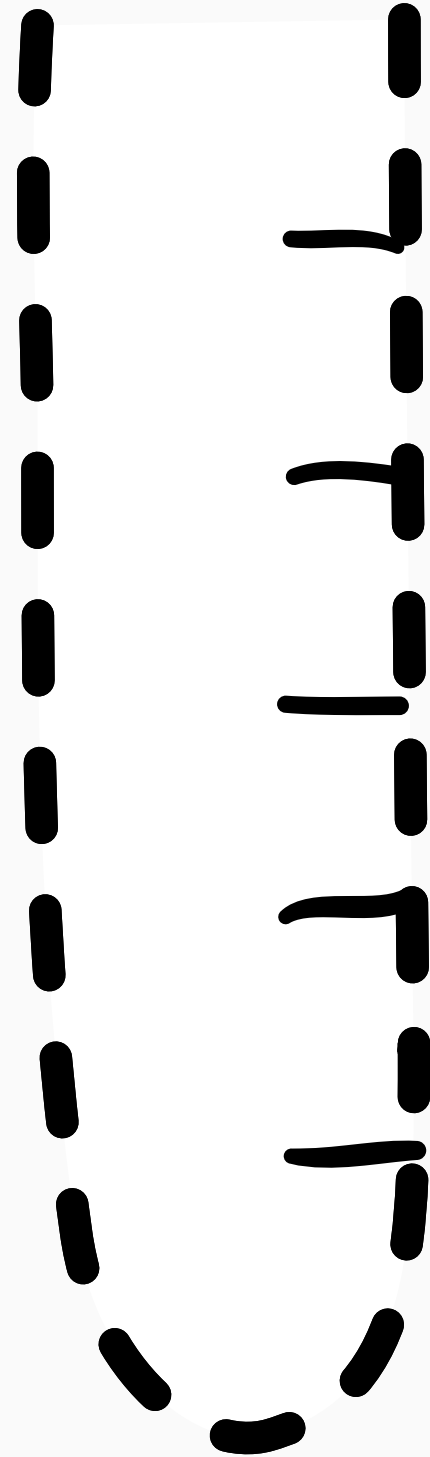
size $\sim Z$



 **complete**

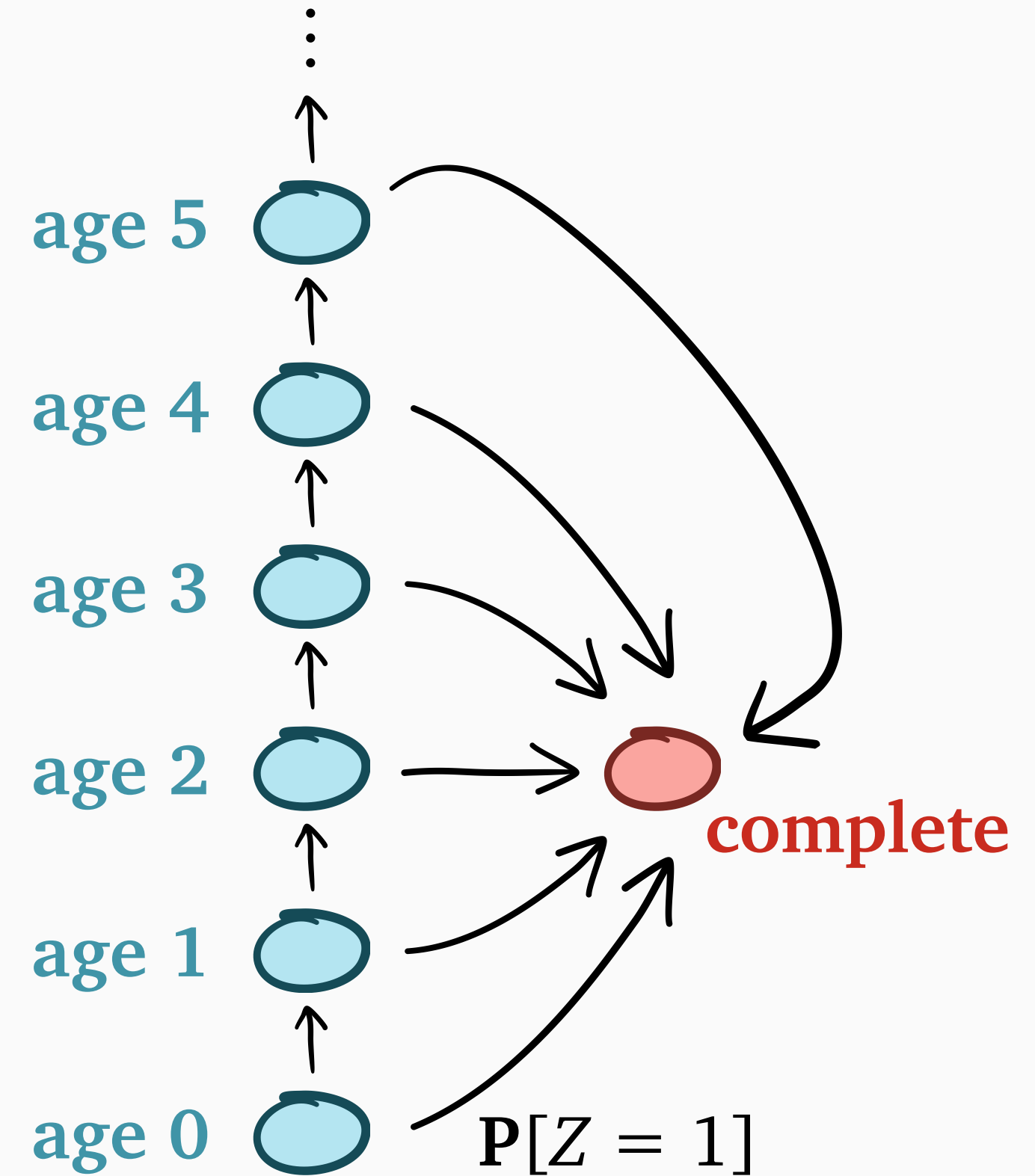
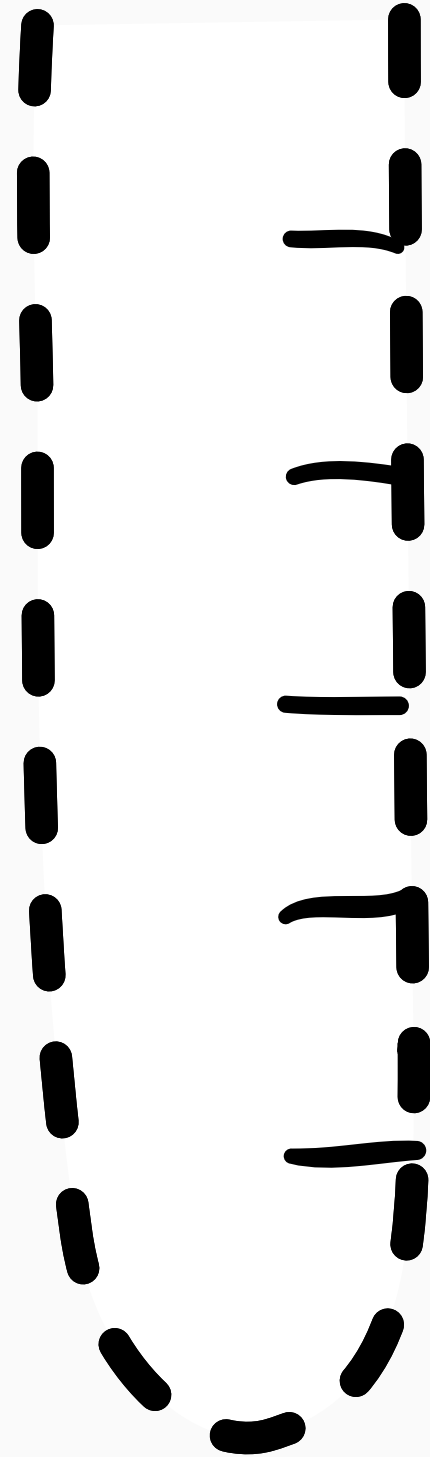
Jobs as Markov chains

size $\sim Z$

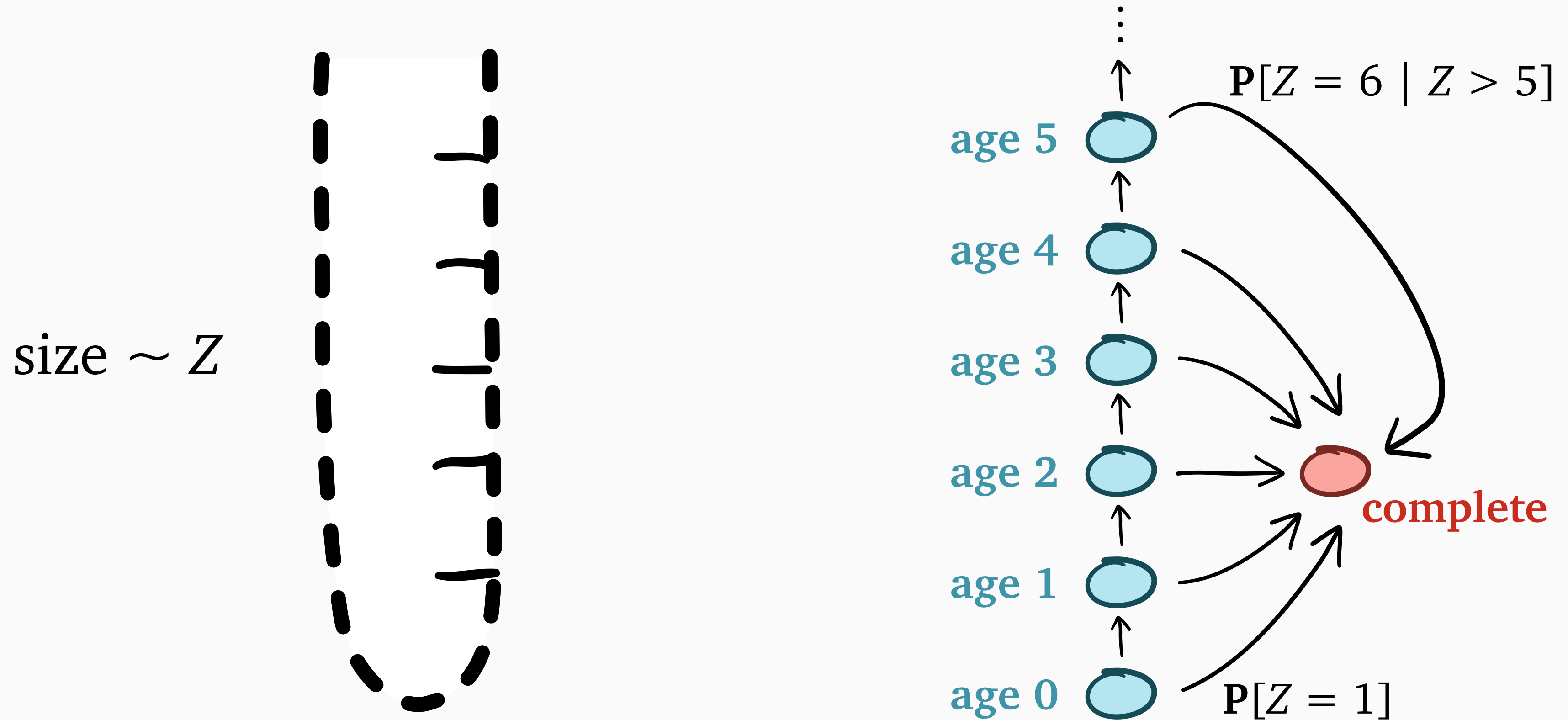


Jobs as Markov chains

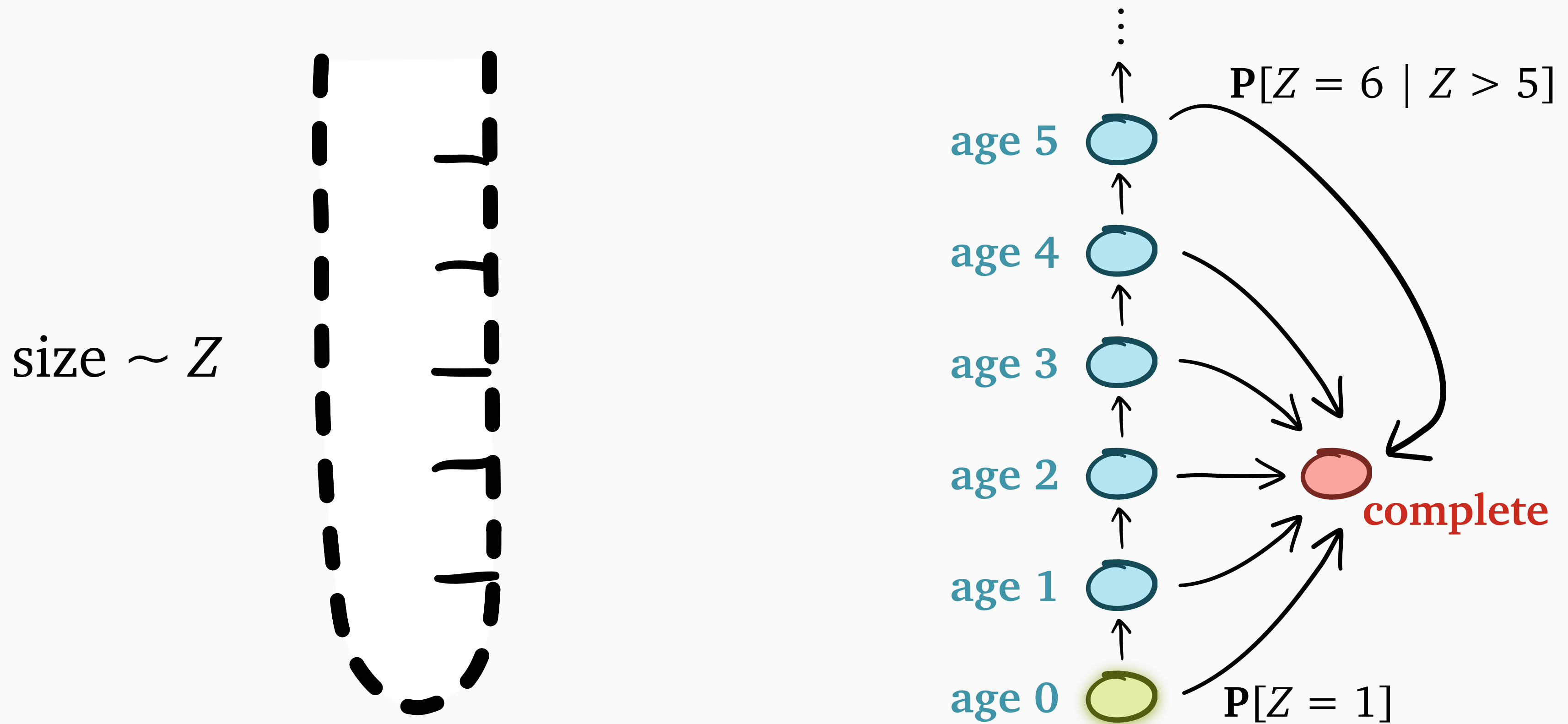
size $\sim Z$



Jobs as Markov chains

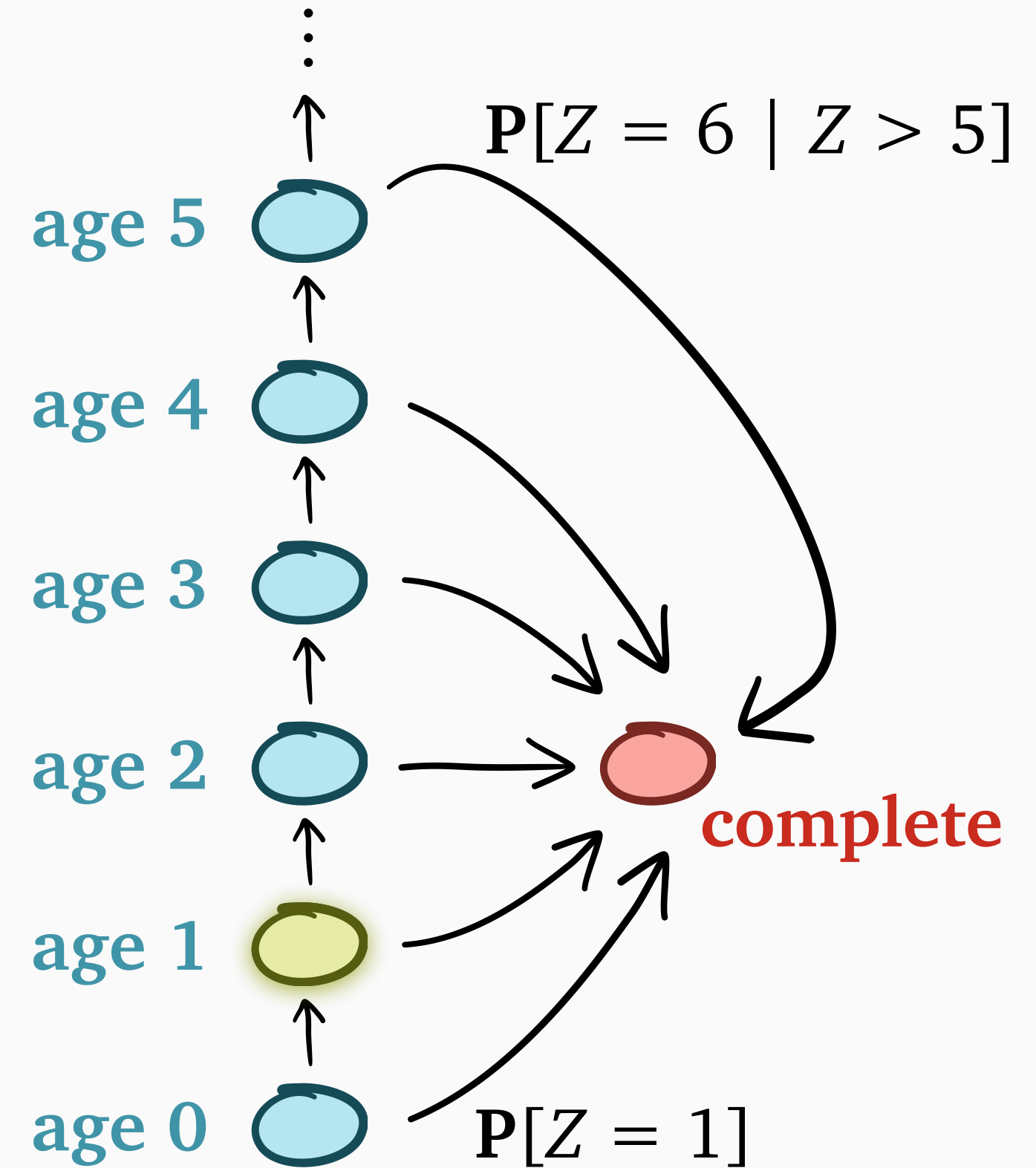
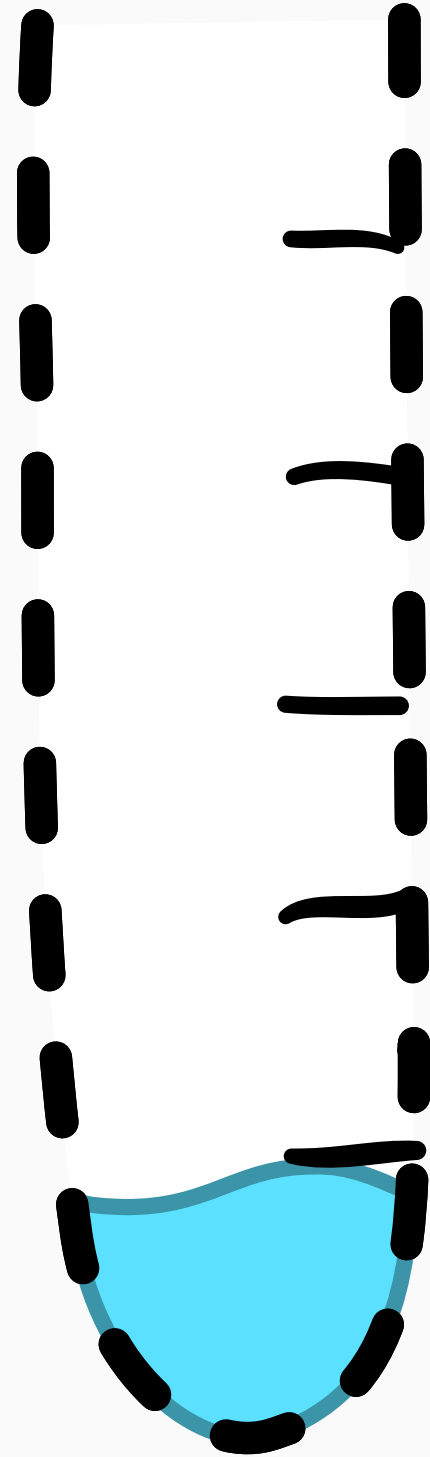


Jobs as Markov chains

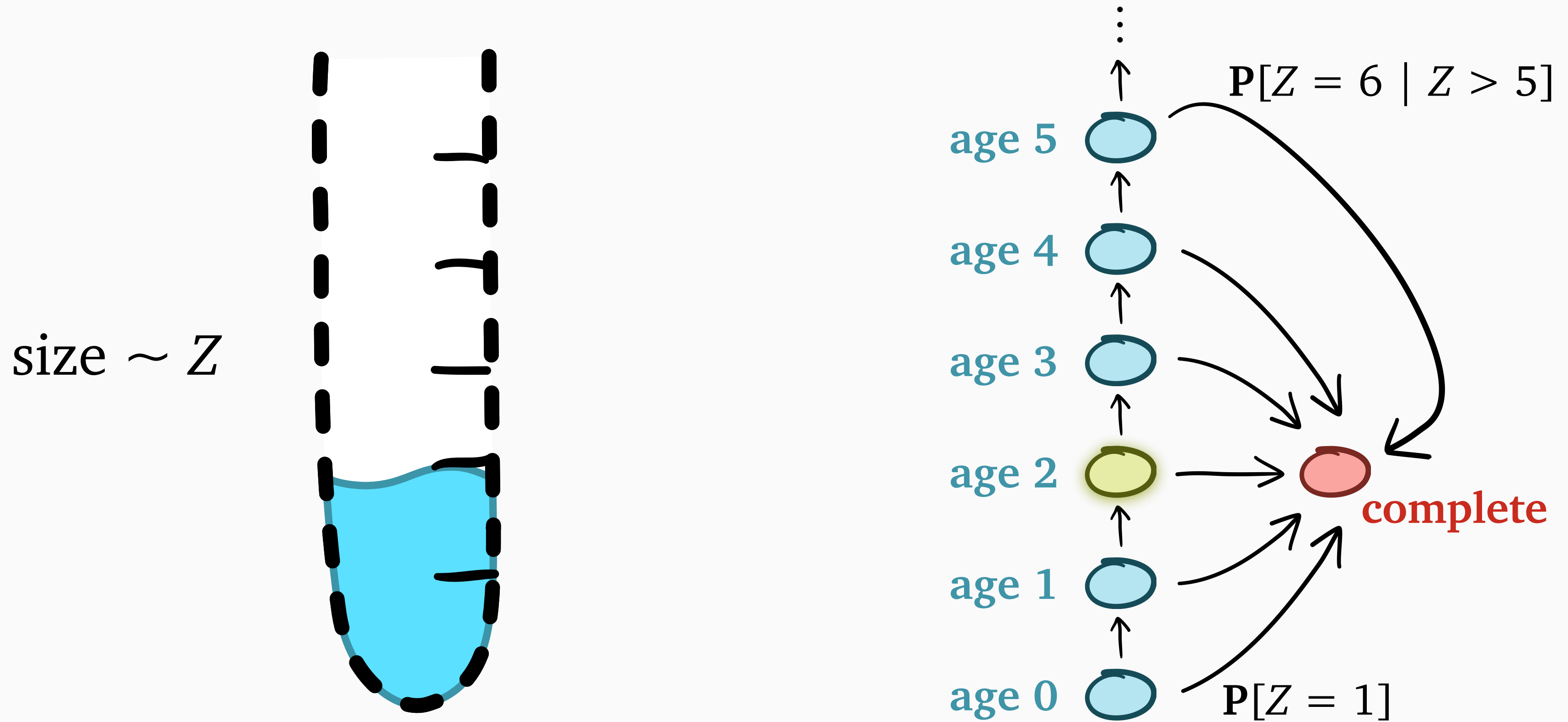


Jobs as Markov chains

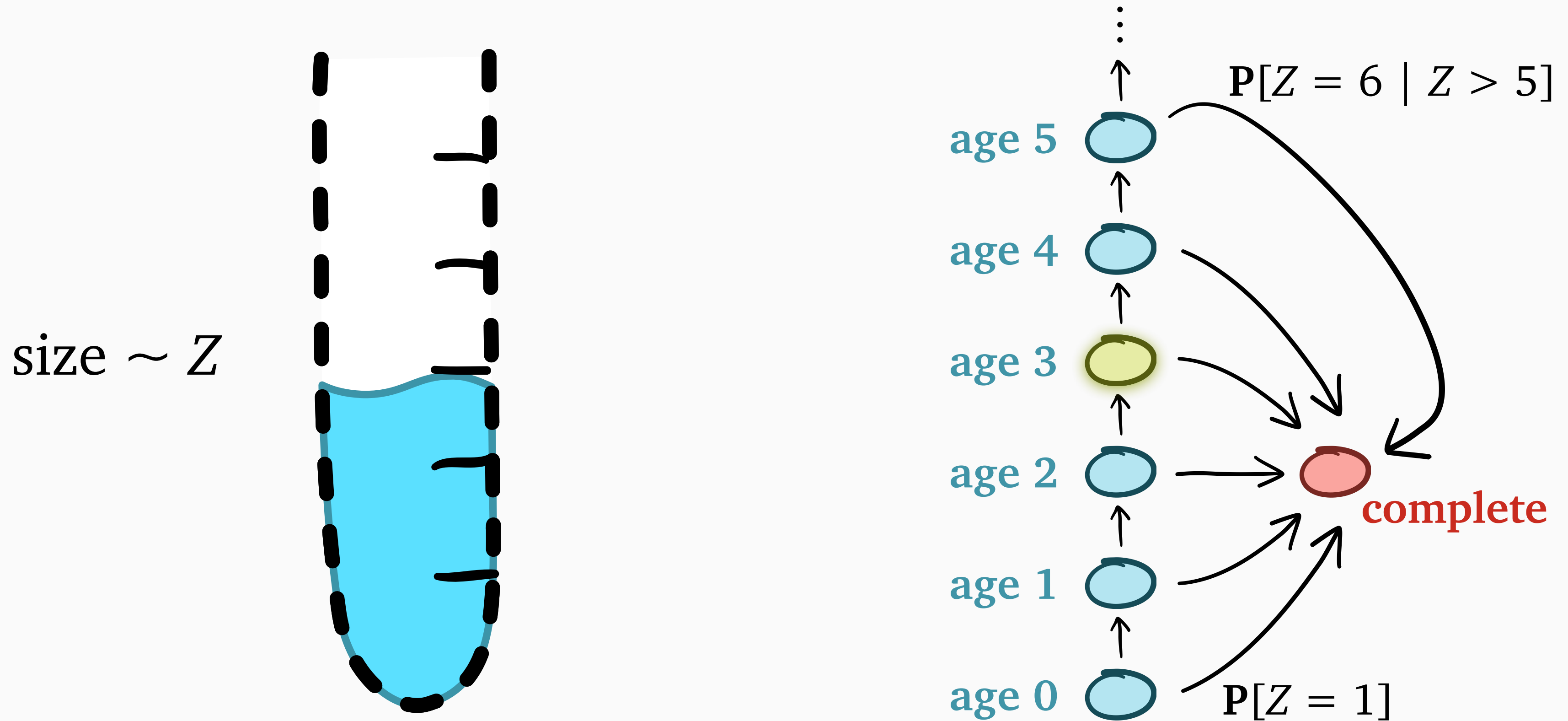
size $\sim Z$



Jobs as Markov chains

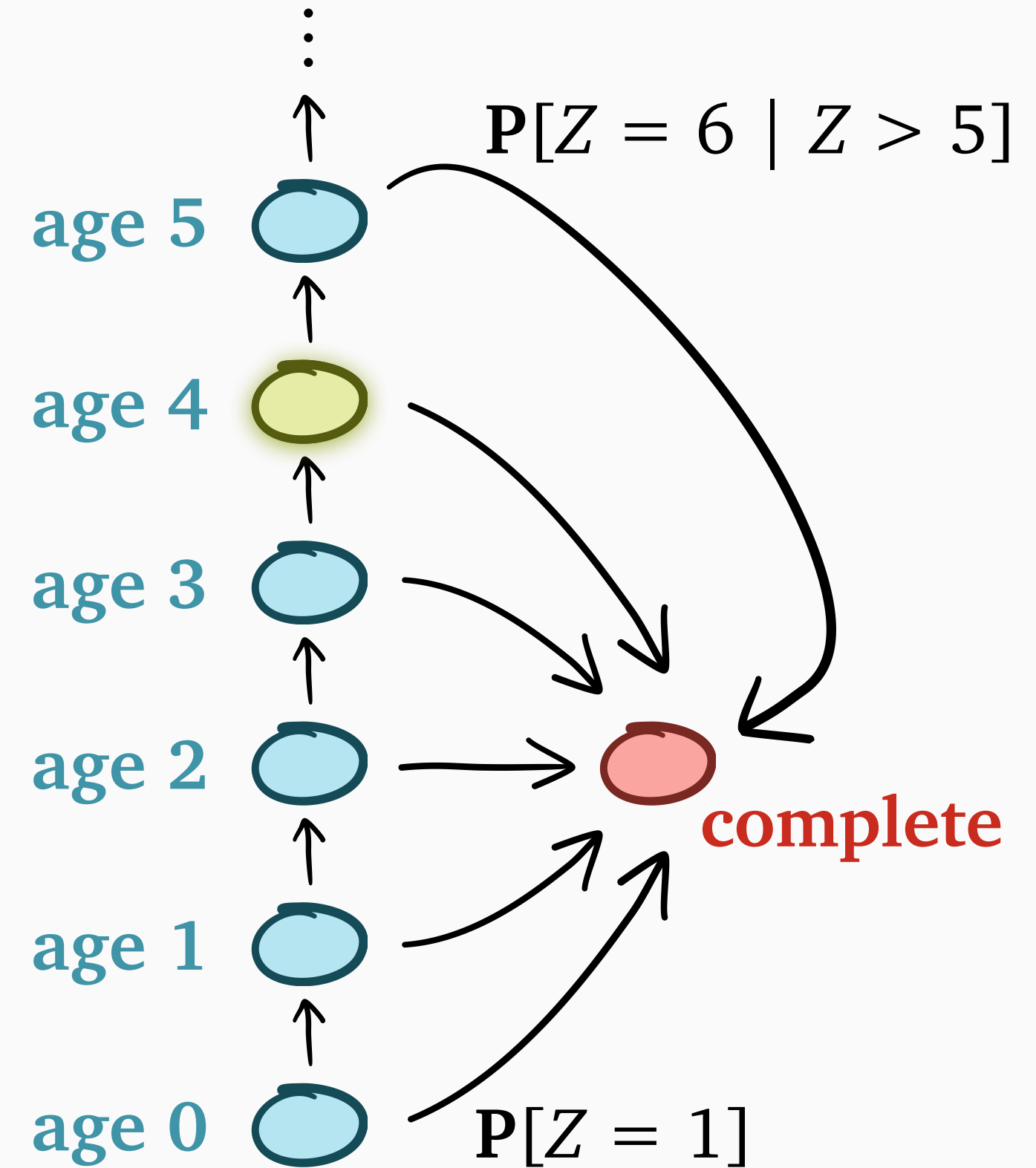
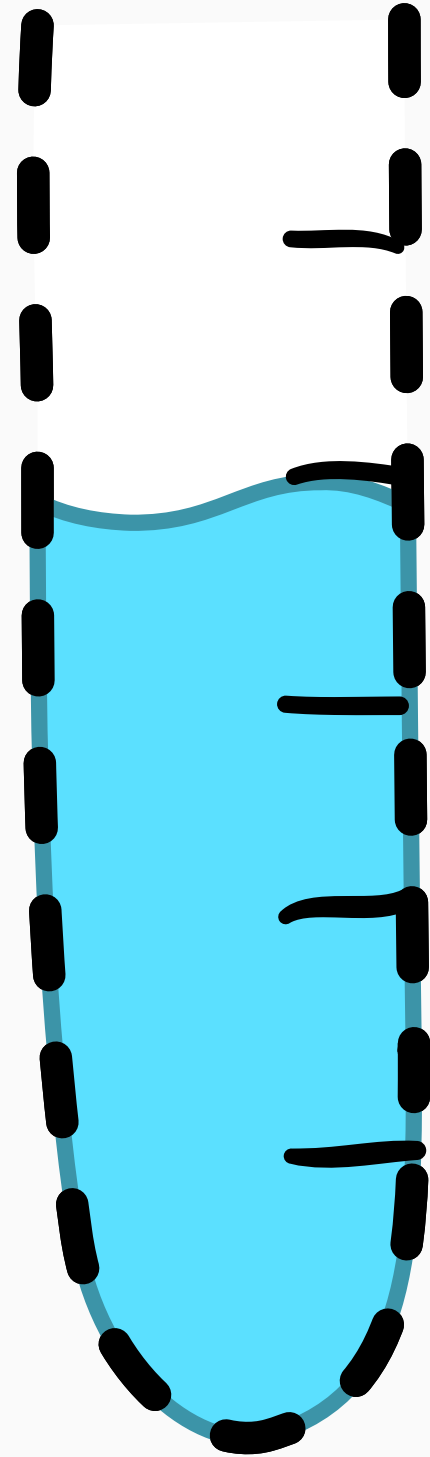


Jobs as Markov chains



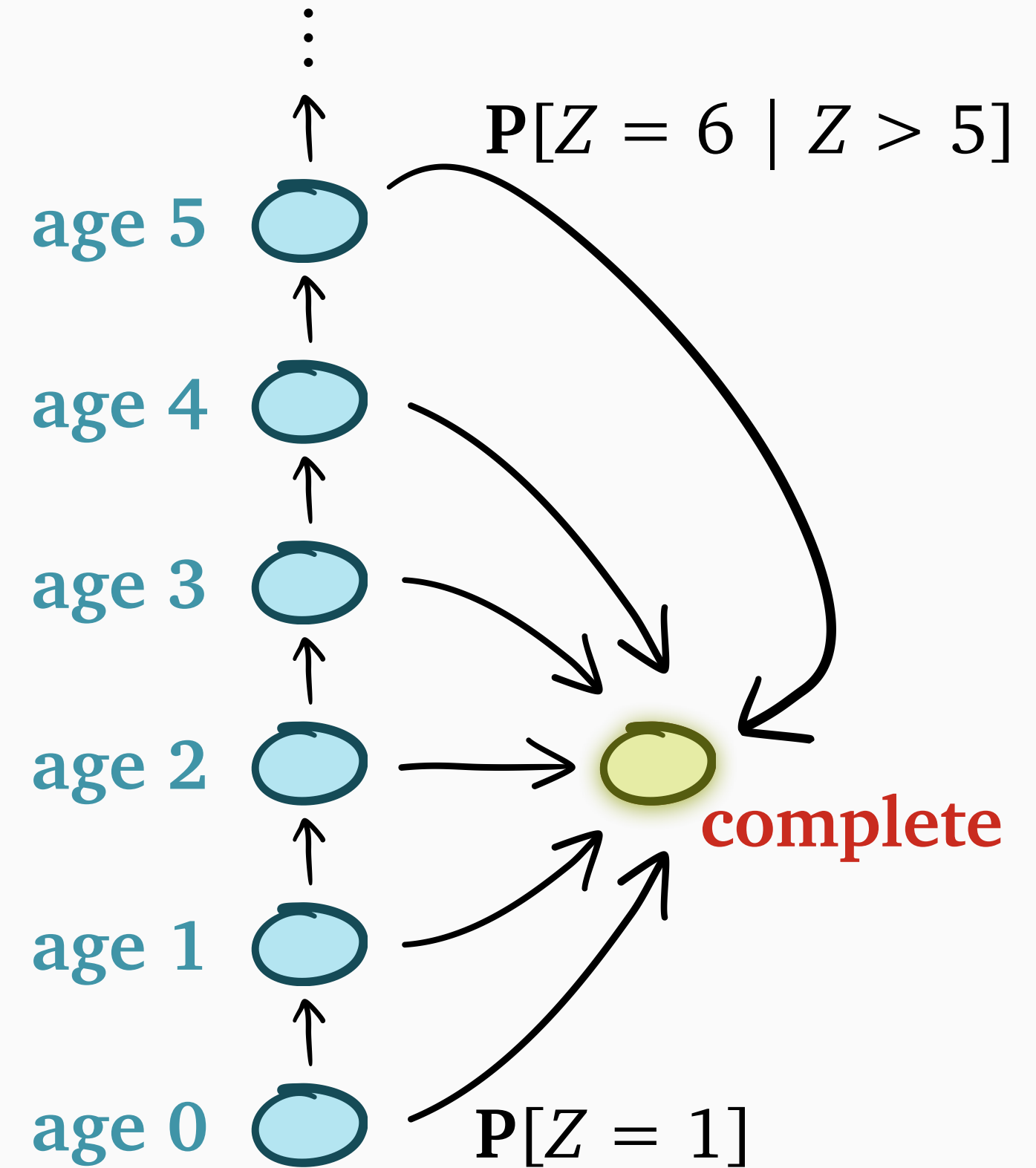
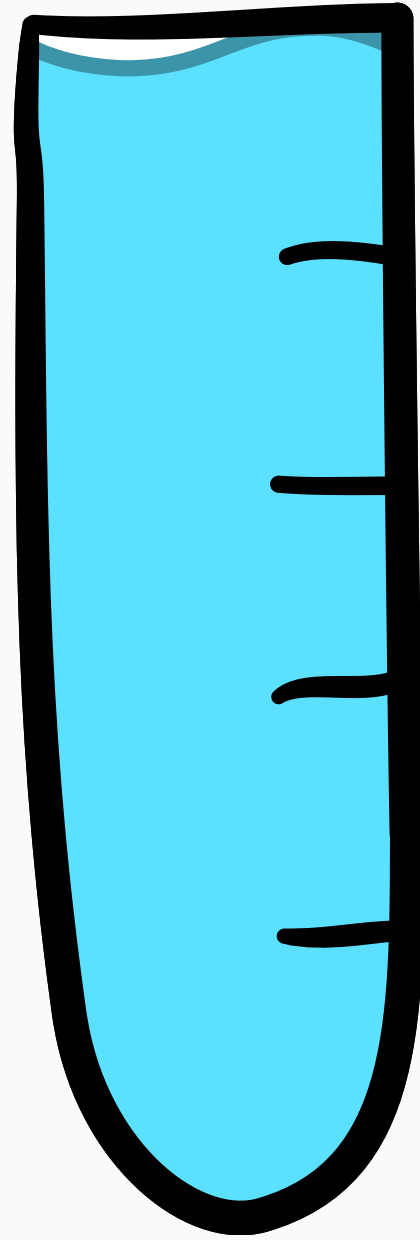
Jobs as Markov chains

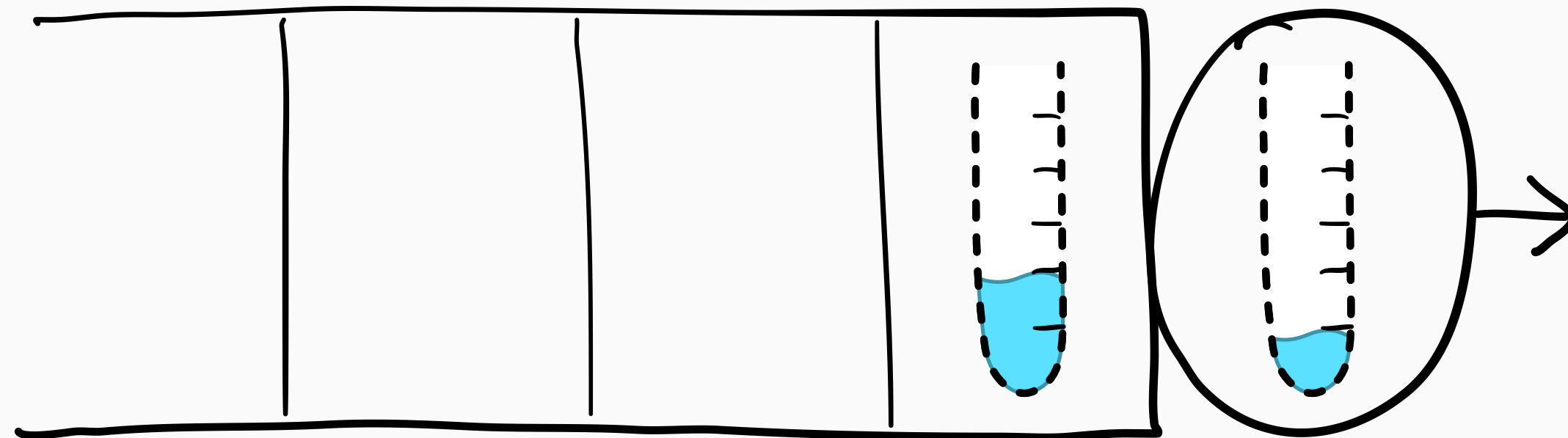
size $\sim Z$

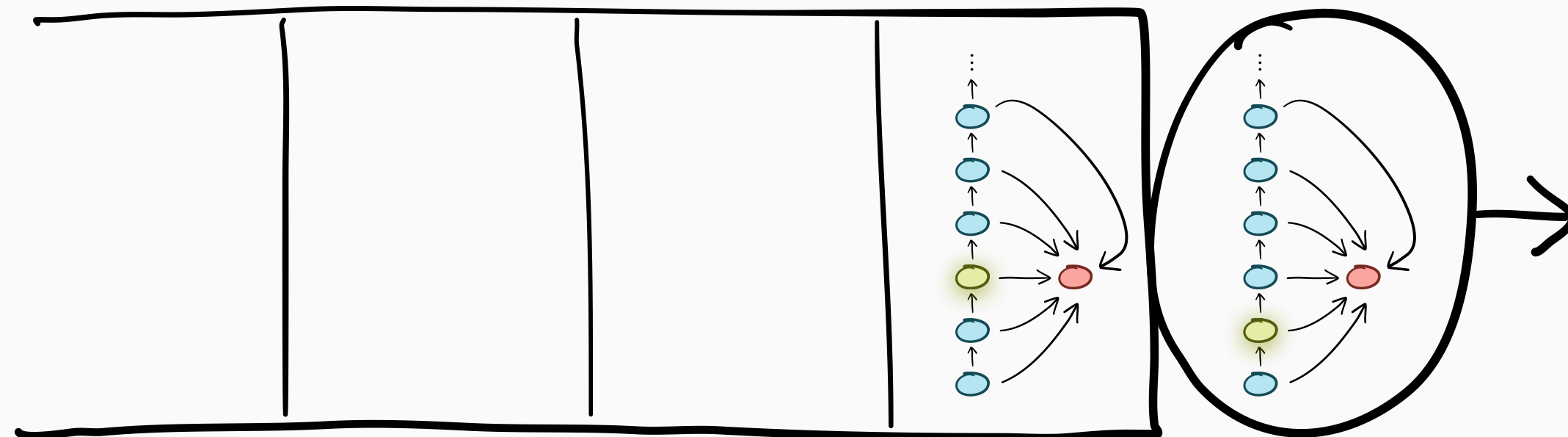


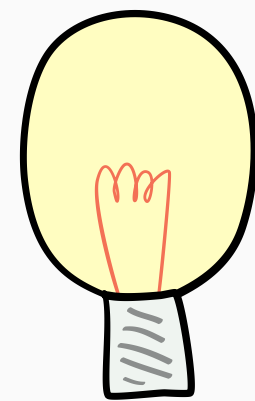
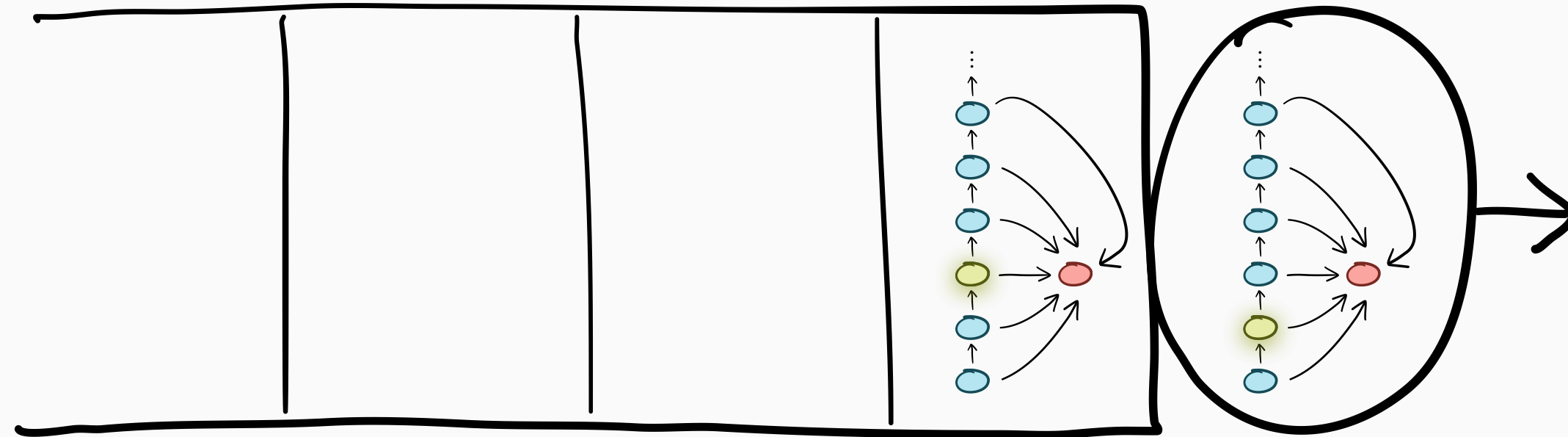
Jobs as Markov chains

size $\sim Z$

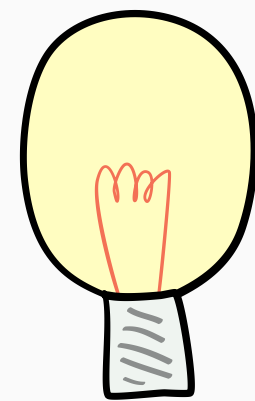
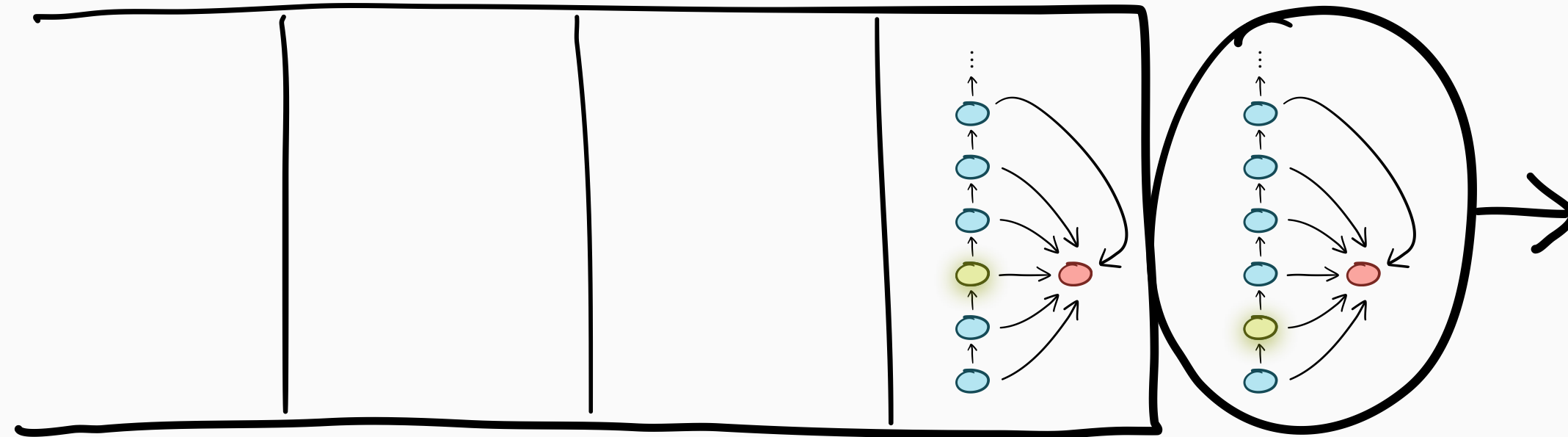






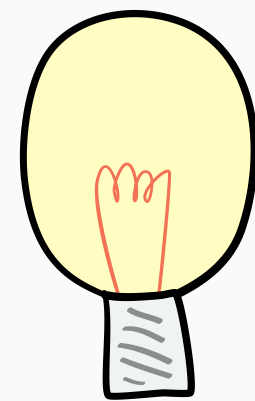
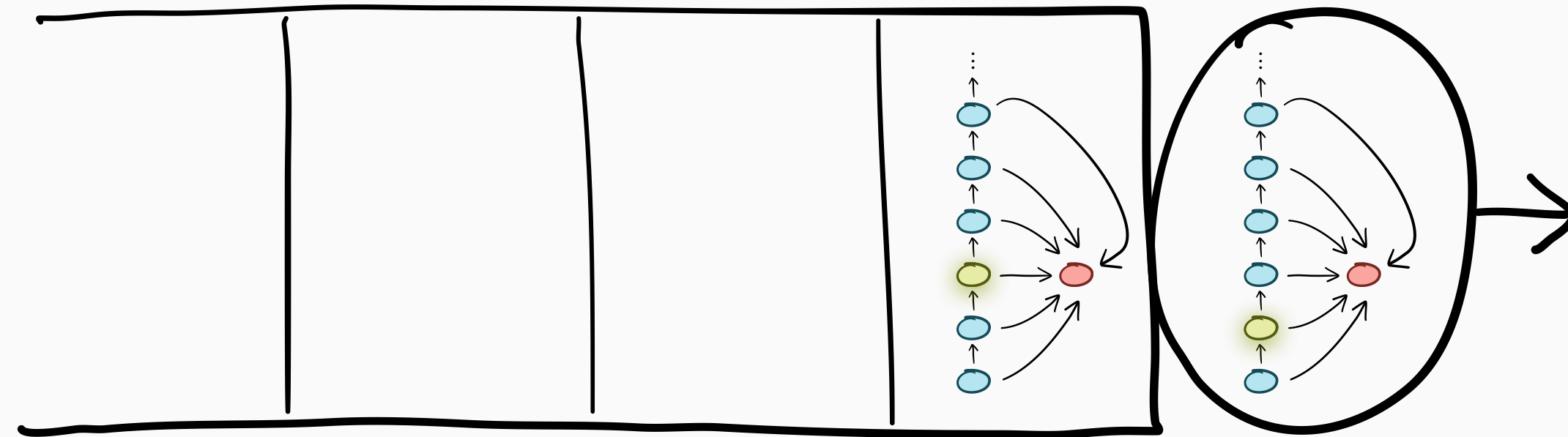


Markov chain represents
information model



Markov chain represents
information model

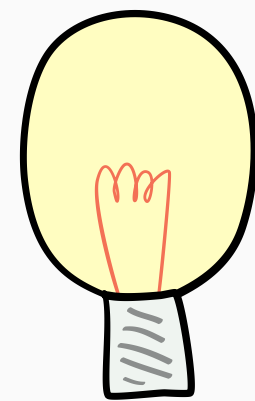
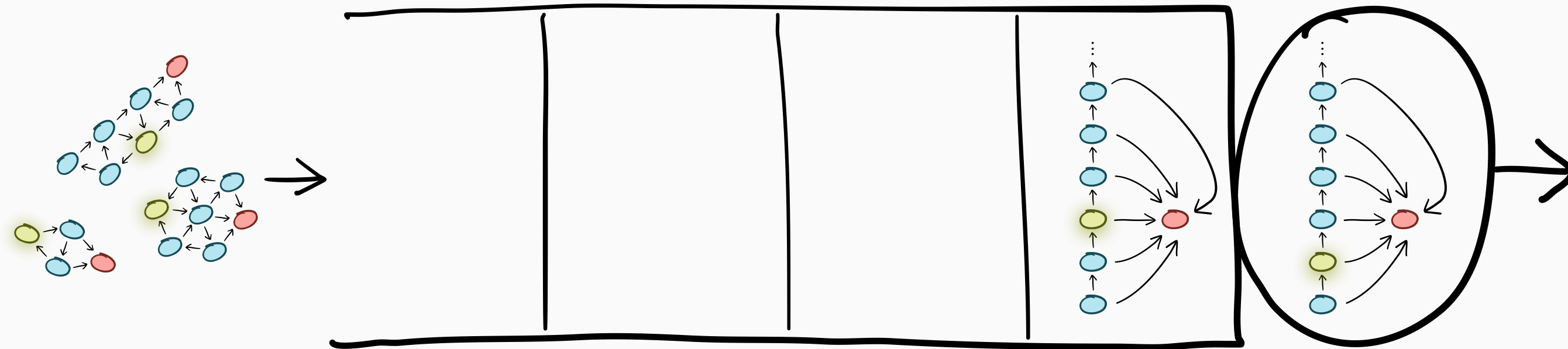
transition probabilities = info about *jobs in general*



Markov chain represents
information model

transition probabilities = info about *jobs in general*

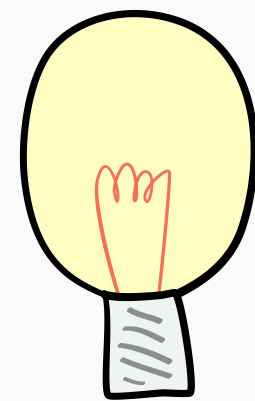
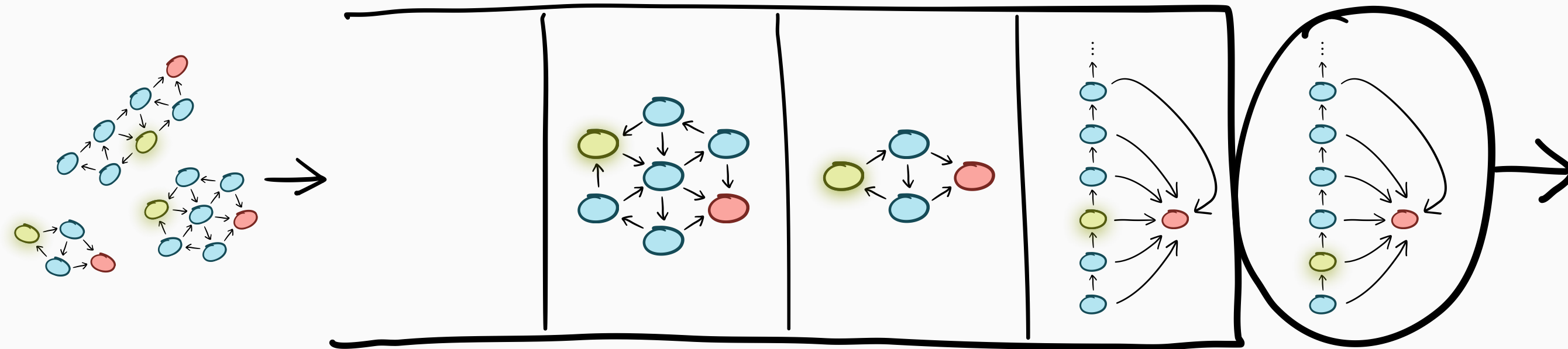
current state = info about *this specific job*



Markov chain represents
information model

transition probabilities = info about *jobs in general*

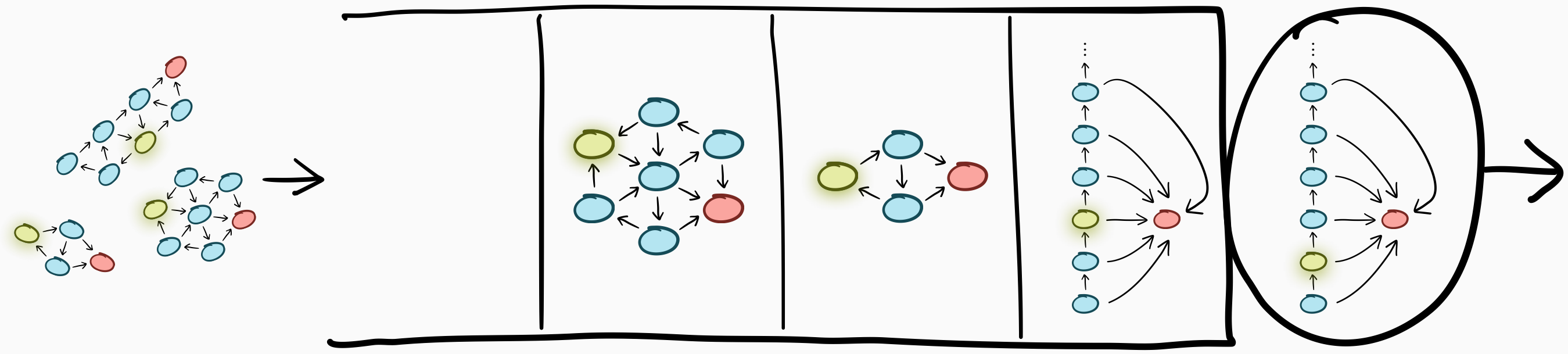
current state = info about *this specific job*

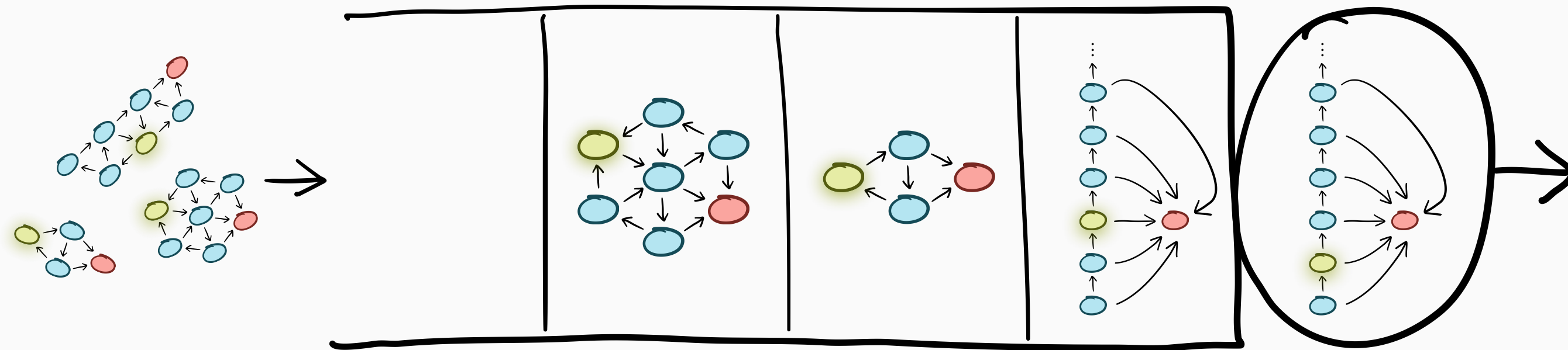


Markov chain represents
information model

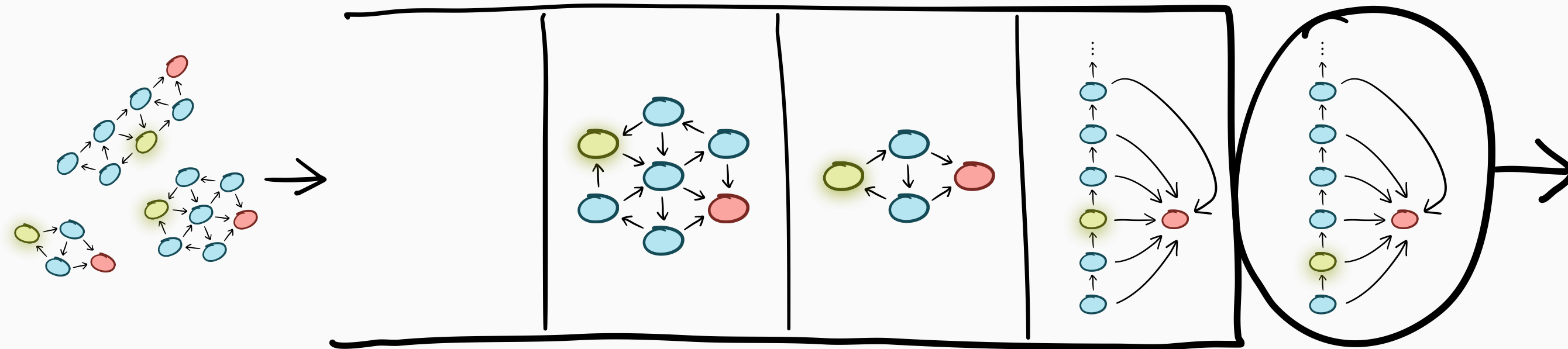
transition probabilities = info about *jobs in general*

current state = info about *this specific job*



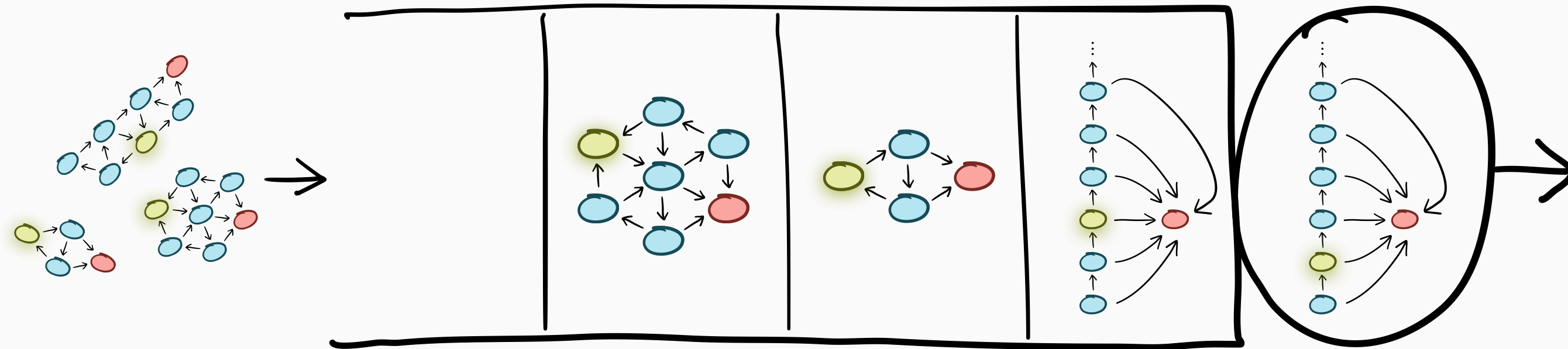


Gittins optimality needs
Poisson arrivals



⚠ **Gittins** optimality needs
Poisson arrivals

⚠ **Gittins** optimality needs
single server



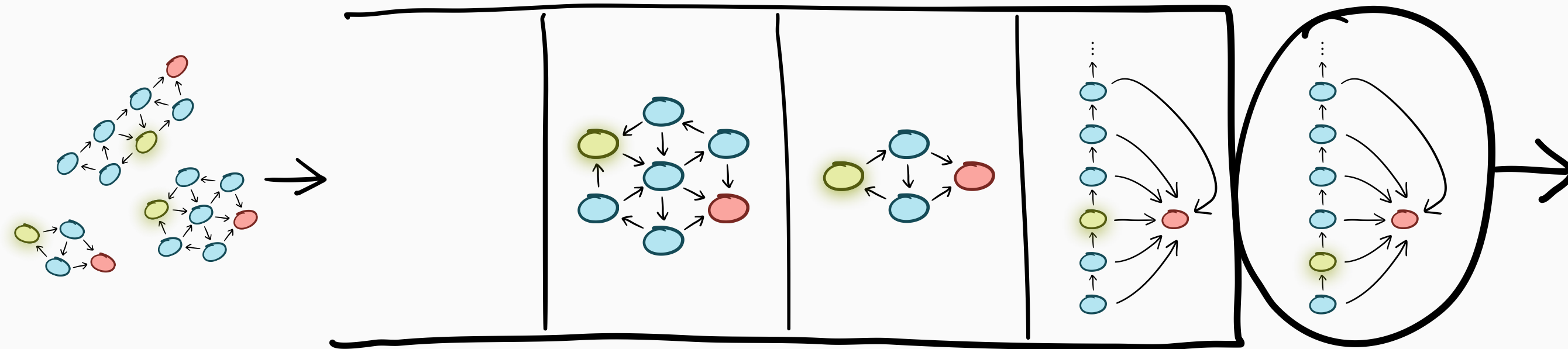
Gittins optimality needs
Poisson arrivals



Gittins optimality needs
single server

suboptimality bound
for G/G arrivals

[Hong & Scully, 2023]



Gittins optimality needs
Poisson arrivals

suboptimality bound
for *G/G arrivals*

[Hong & Scully, 2023]

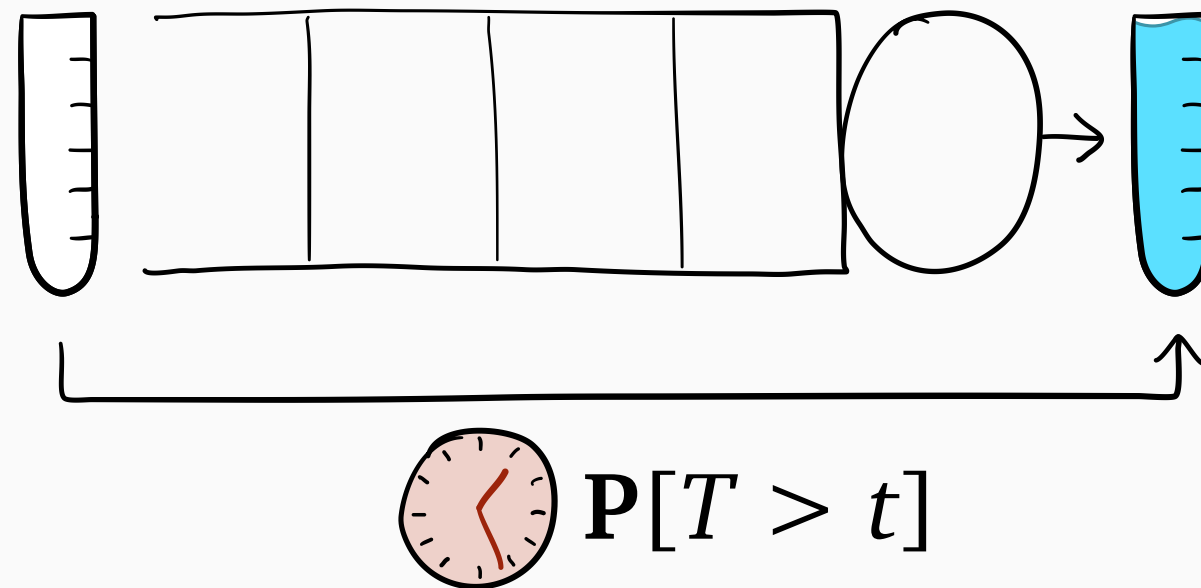


Gittins optimality needs
single server

suboptimality bounds
for *multiserver settings*

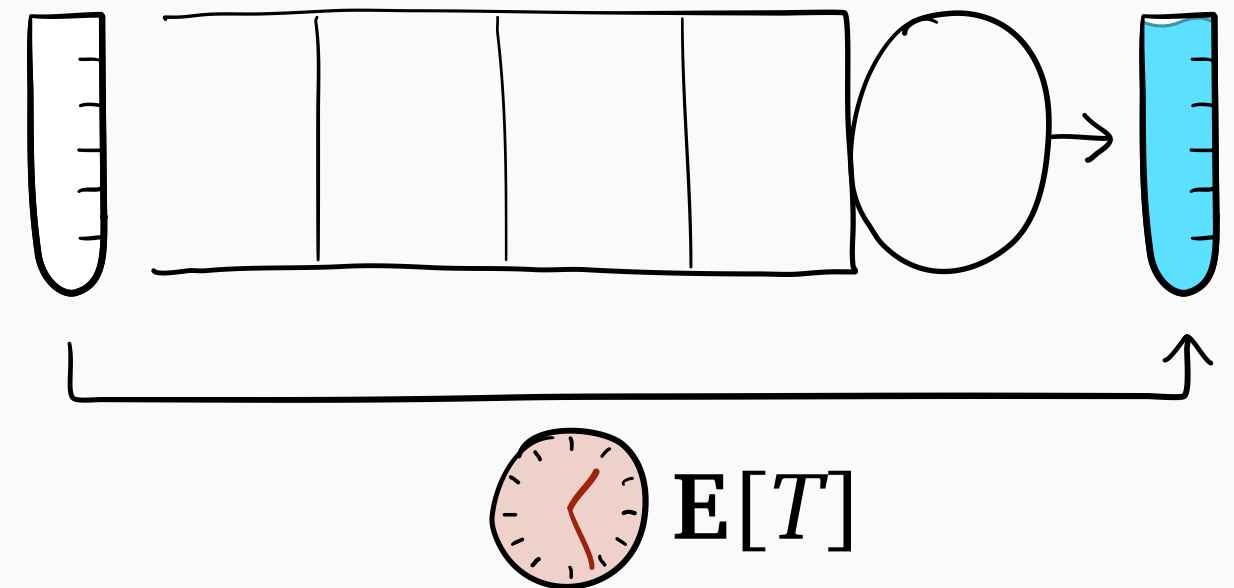
many—see [Grosz, 2023]

Tail scheduling



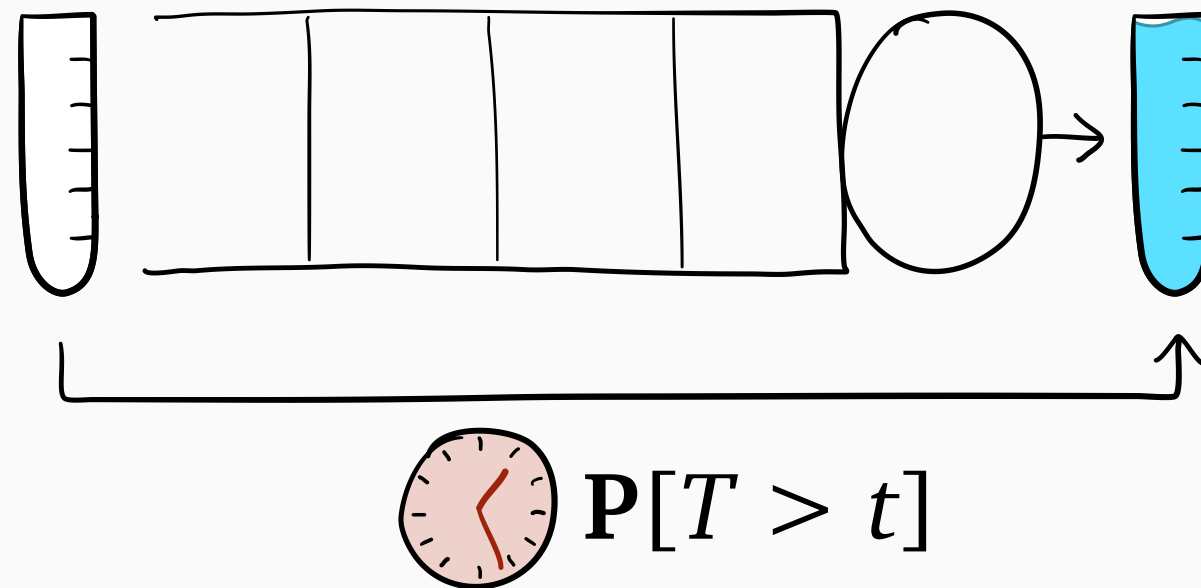
Goal: minimize probability of very long response time

Mean scheduling



Goal: minimize average response time (or # jobs)

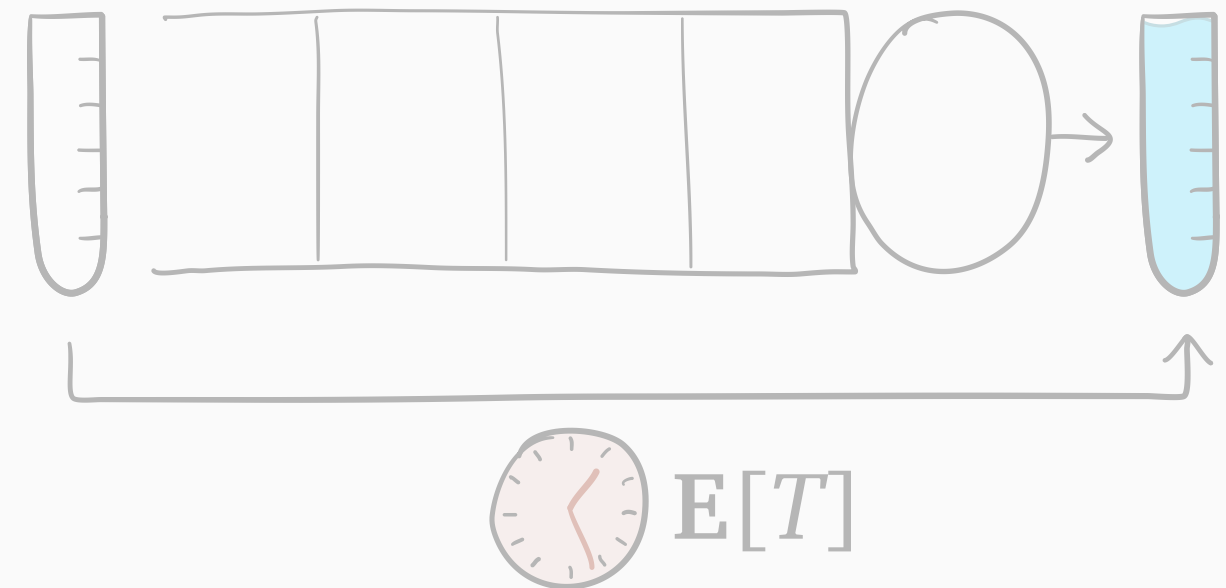
Tail scheduling



Goal: minimize probability of very long response time

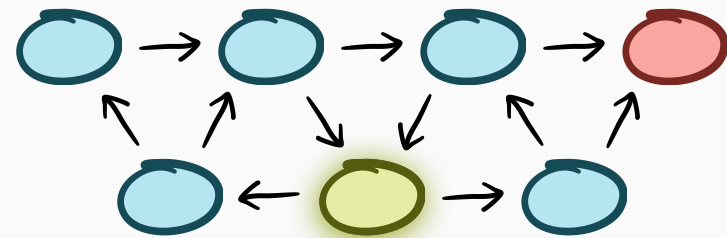
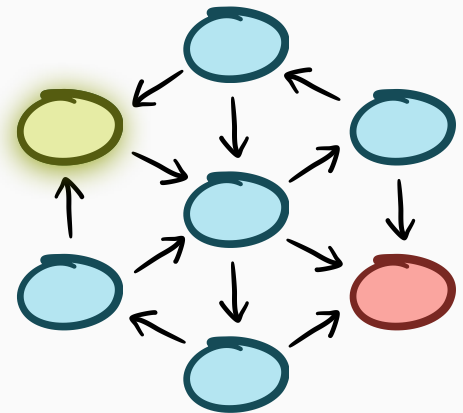
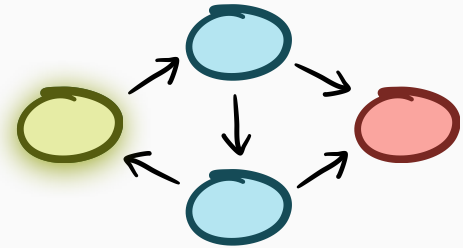


Mean scheduling



Goal: minimize average response time (or # jobs)

MCS with discounting



Decision process:

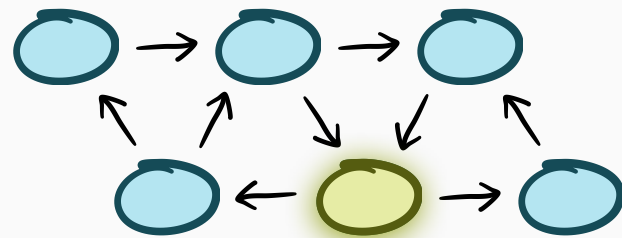
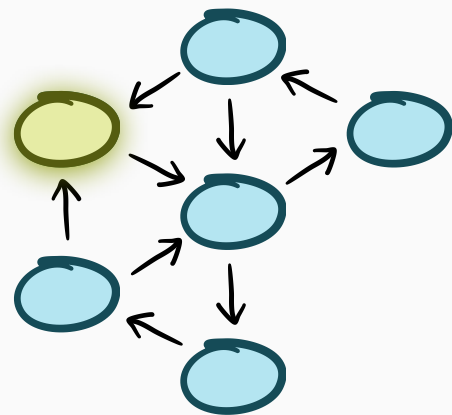
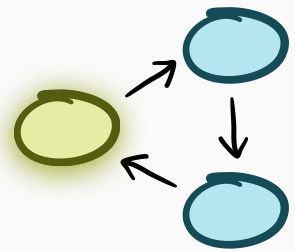
- Advance one Markov chain every step
- Get reward based on its **current state**
- Stop when one Markov chain **terminates**

MCS with **discounting**

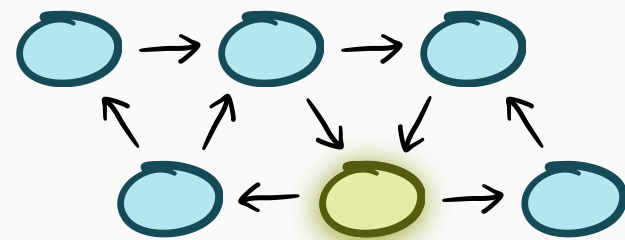
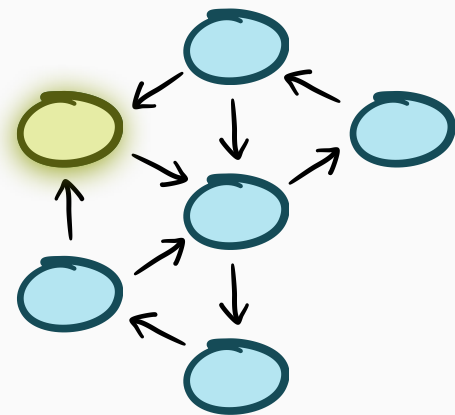
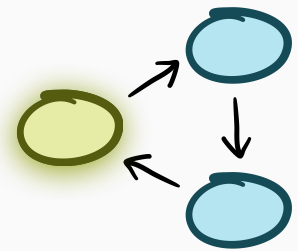
Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**

✗ Stop when one Markov chain **terminates**



MCS with **discounting**



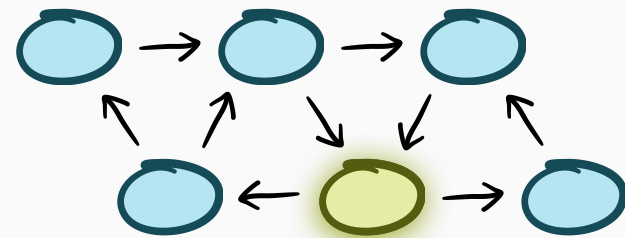
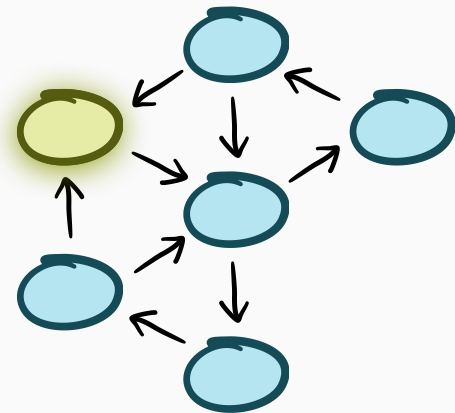
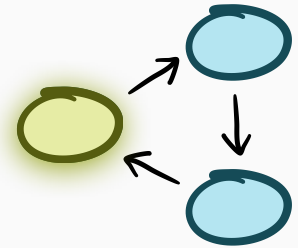
Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**

✗ Stop when one Markov chain **terminates**

Goal: maximize $E[\text{discounted reward}]$?

MCS with **discounting**



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**

✗ Stop when one Markov chain **terminates**

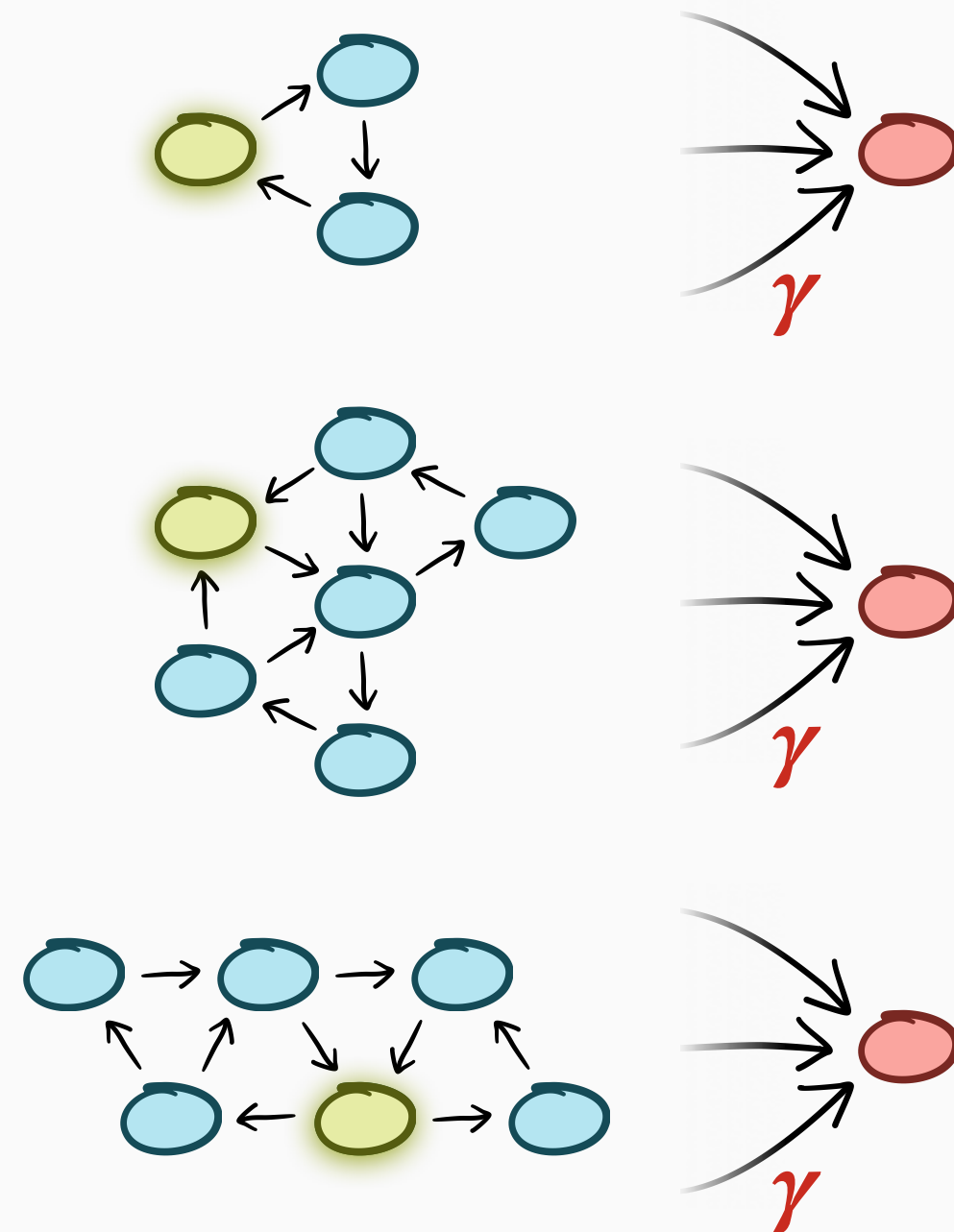
Goal: maximize $E[\text{discounted reward}]$?



Use constant **termination** probability

γ = **discount** factor

MCS with **discounting**



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**

✗ Stop when one Markov chain **terminates**

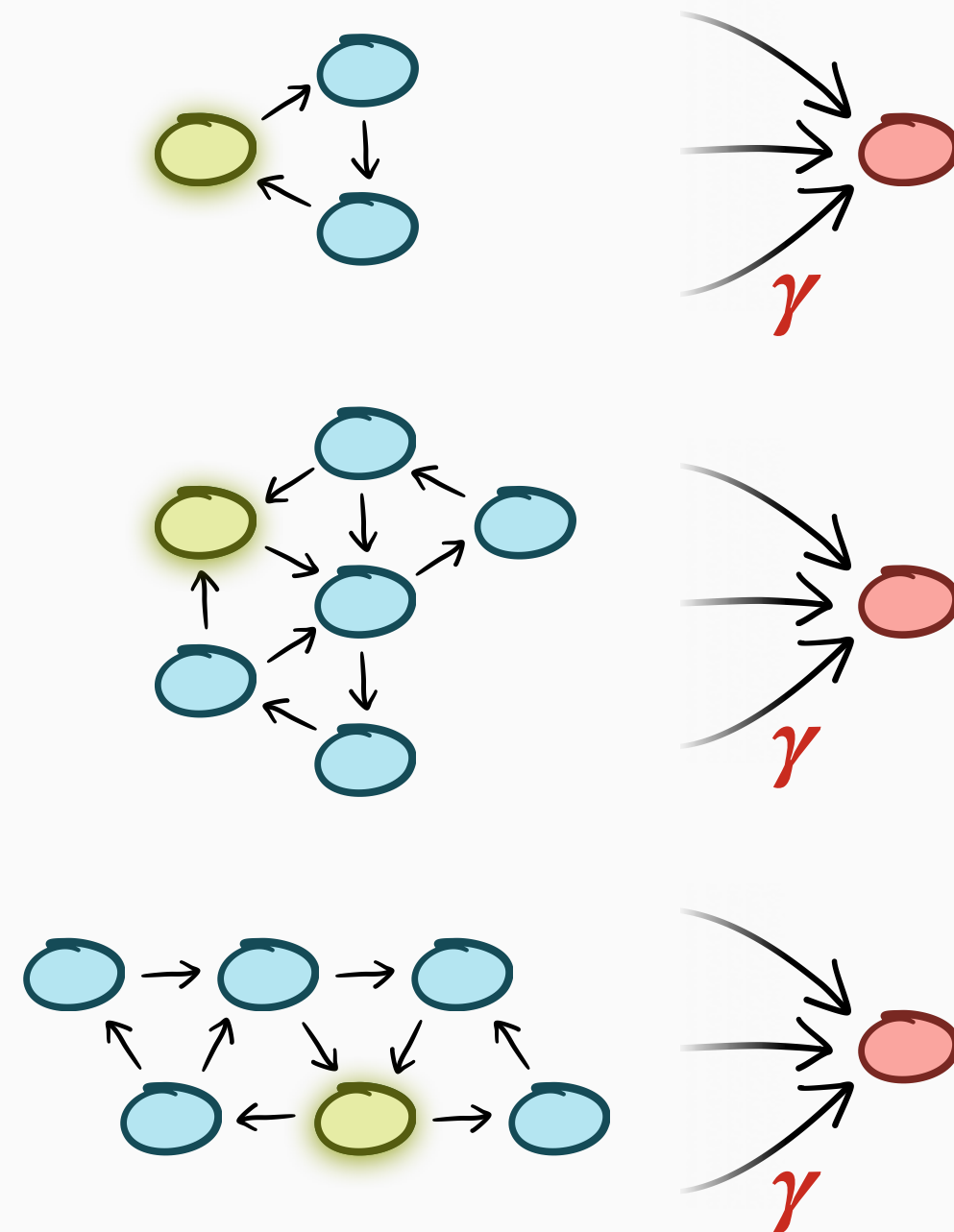
Goal: maximize $E[\text{discounted reward}]$?



Use constant **termination** probability

γ = **discount** factor

MCS with **discounting**



Decision process:

- Advance one Markov chain every step
 - Get reward based on its **current state**
- ✗ Stop when one Markov chain **terminates**

Goal: maximize $E[\text{discounted reward}]?$

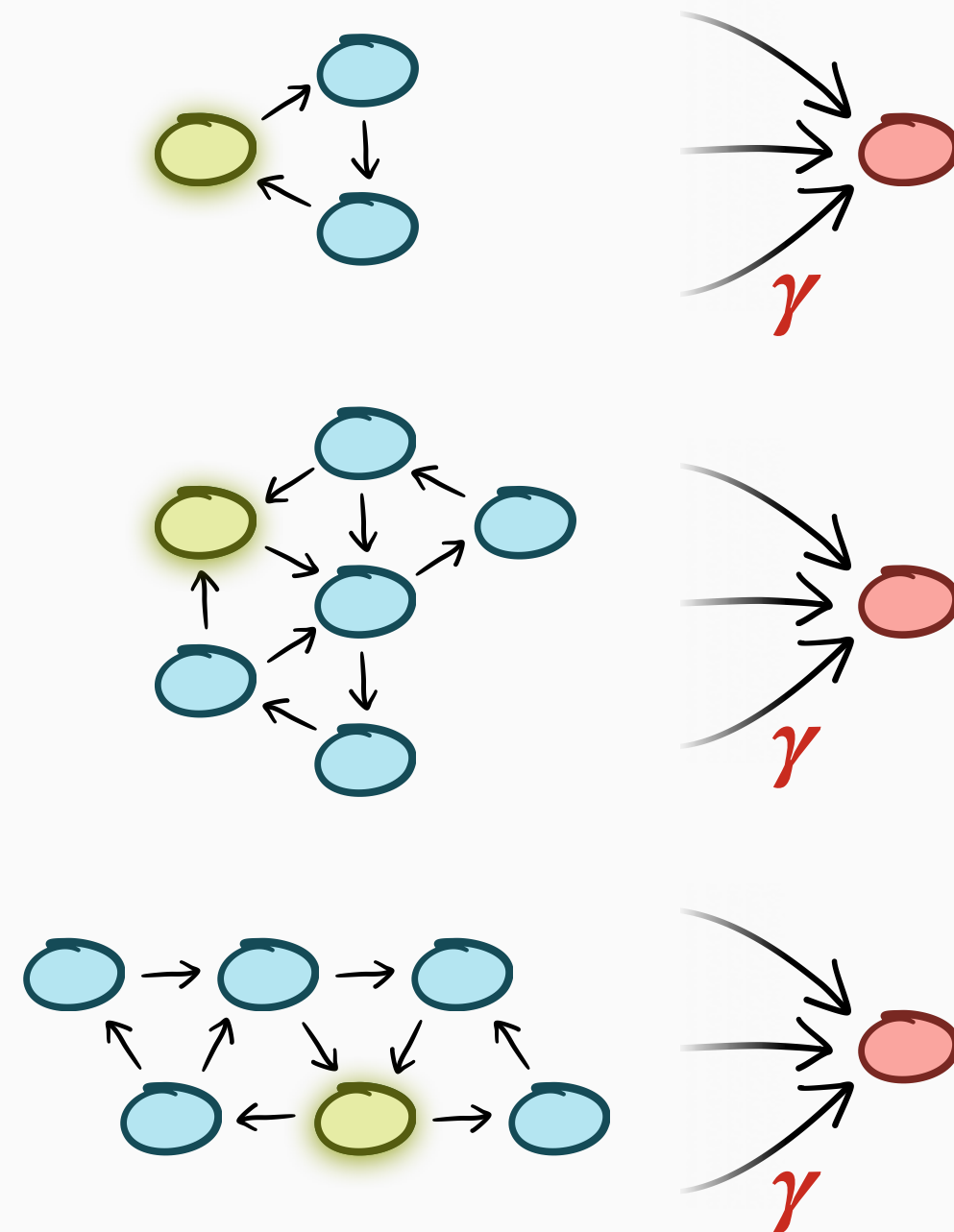
$E[\text{total reward}]$



Use constant **termination** probability

$\gamma = \text{discount factor}$

MCS with **discounting**



Decision process:

- Advance one Markov chain every step
- Get reward based on its **current state**
- ✓ Stop when one Markov chain **terminates**

Goal: maximize $E[\text{discounted reward}]$?

$E[\text{total reward}]$

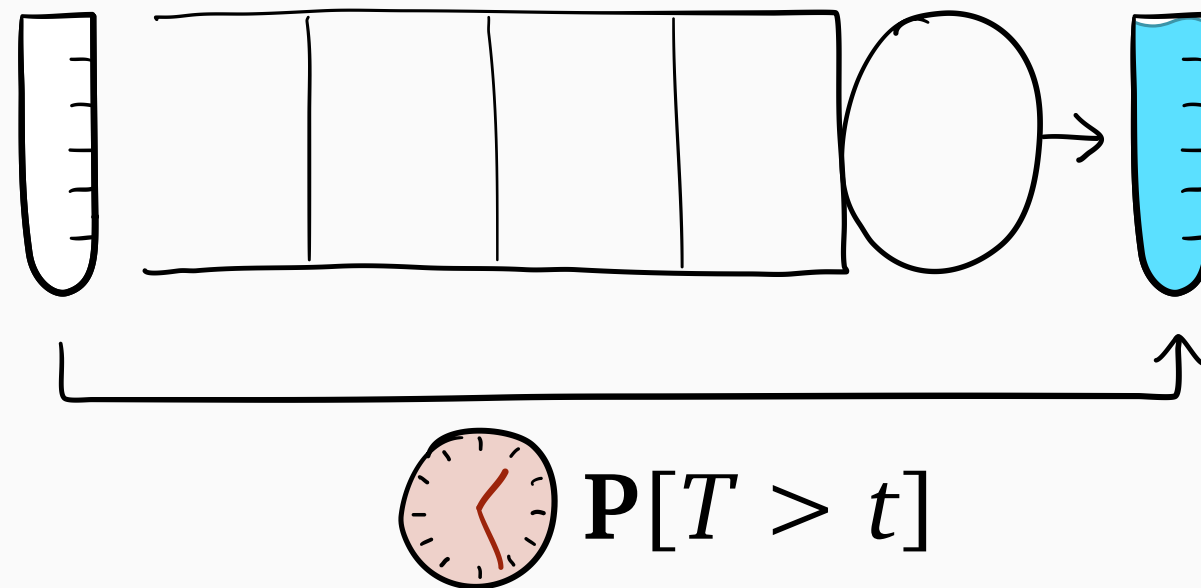


Use constant **termination** probability

$\gamma = \text{discount factor}$

How **discounting** helps **tail scheduling**

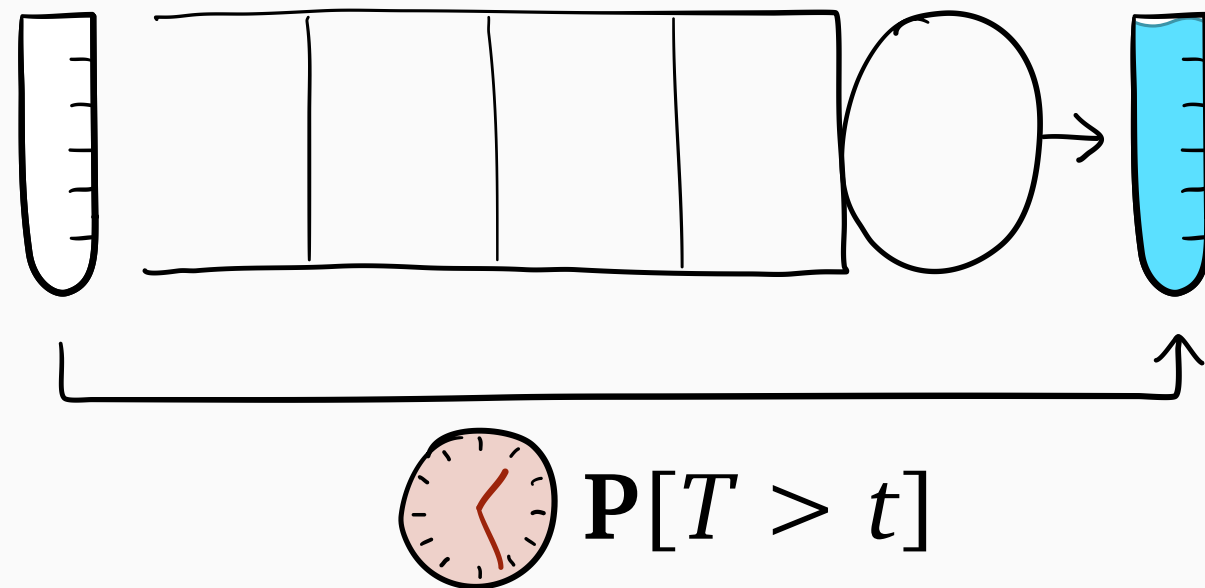
Tail scheduling



Goal: minimize probability
of very long response time

How **discounting** helps **tail scheduling**

Tail scheduling



Goal: minimize probability of very long response time

Moment generating function fact:

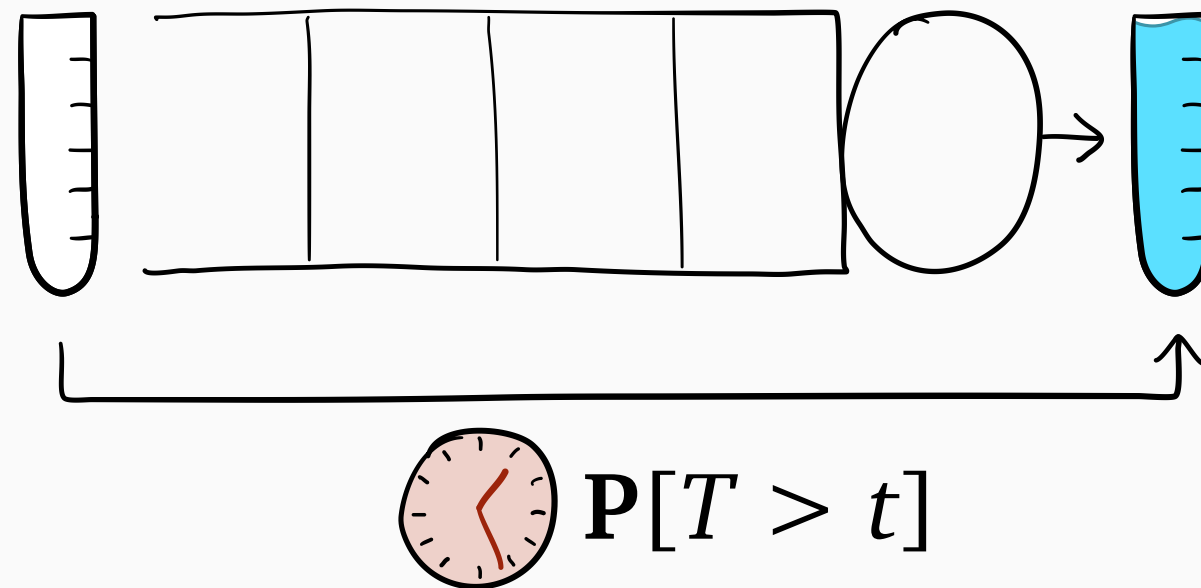
$$\mathbf{P}[T > t] \sim C e^{-\theta t} \quad (t \rightarrow \infty)$$

$\Leftrightarrow$

$$\mathbf{E}[e^{(1-\varepsilon)\theta T}] \sim \frac{C}{\varepsilon} \quad (\varepsilon \rightarrow 0)$$

How **discounting** helps **tail scheduling**

Tail scheduling



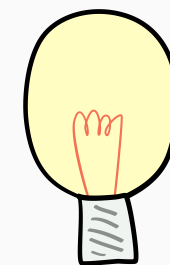
Goal: minimize probability of very long response time

Moment generating function fact:

$$P[T > t] \sim C e^{-\theta t} \quad (t \rightarrow \infty)$$



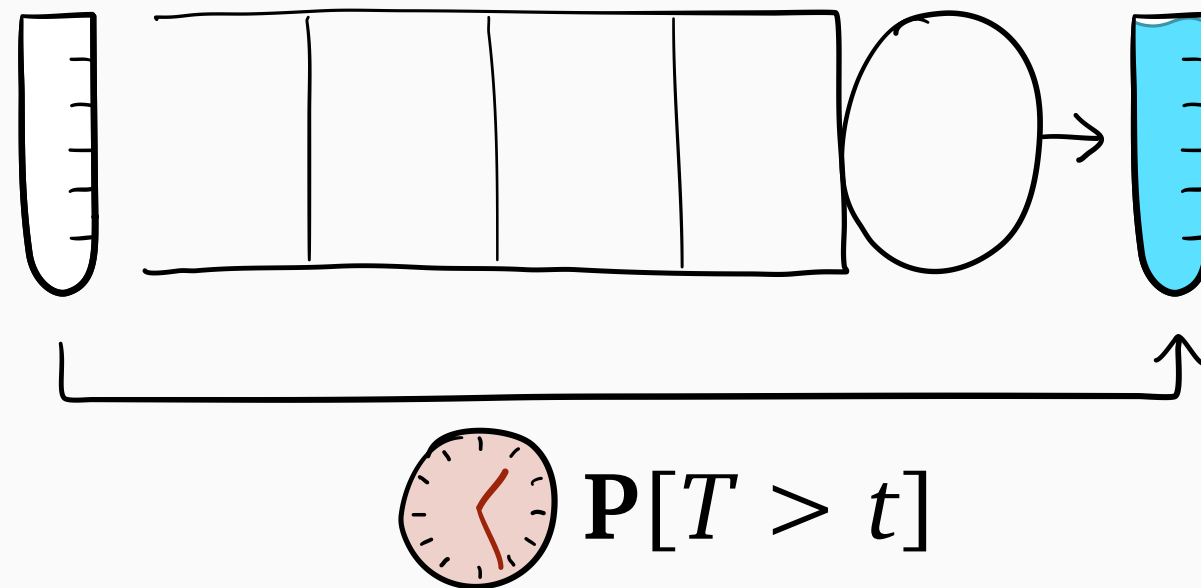
$$E[e^{(1-\varepsilon)\theta T}] \sim \frac{C}{\varepsilon} \quad (\varepsilon \rightarrow 0)$$



On completion, job pays cost 1, which **inflates** during time T

How **discounting** helps **tail scheduling**

Tail scheduling



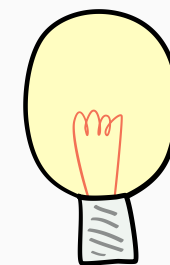
Goal: minimize probability of very long response time

Moment generating function fact:

$$P[T > t] \sim C e^{-\theta t} \quad (t \rightarrow \infty)$$



$$E[e^{(1-\varepsilon)\theta T}] \sim \frac{C}{\varepsilon} \quad (\varepsilon \rightarrow 0)$$



On completion, job pays cost 1, which **inflates** during time T

“**discounting**”
with $\gamma > 1$

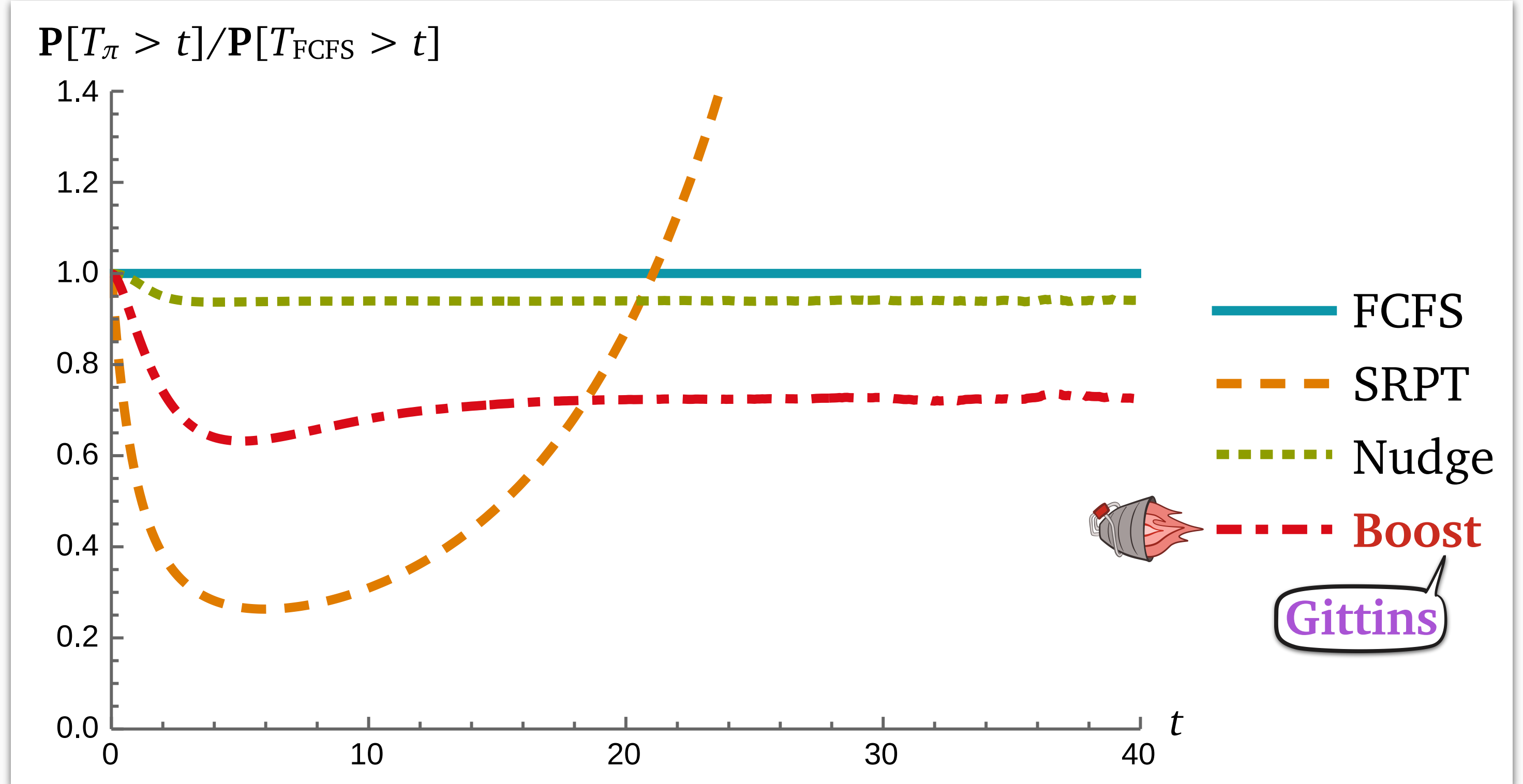
Empirical results: **tail scheduling**



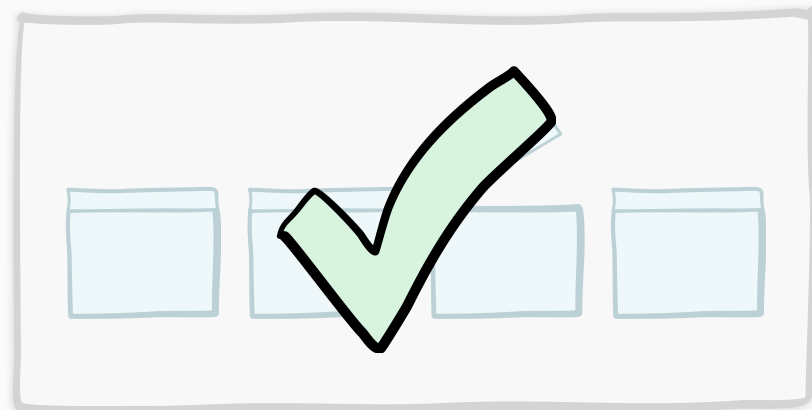
George Yu



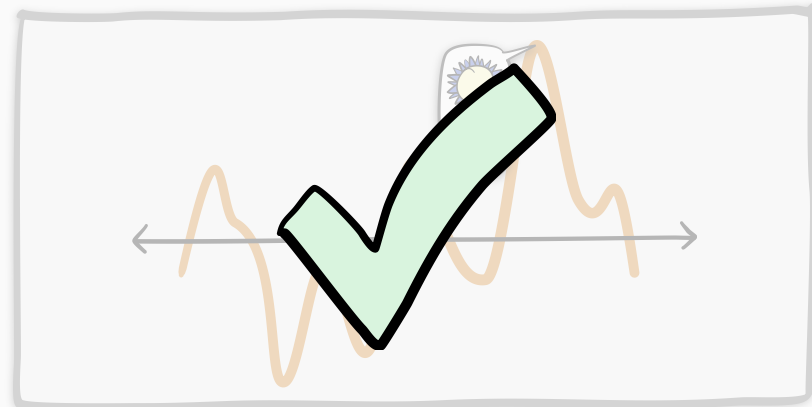
Amit Harlev



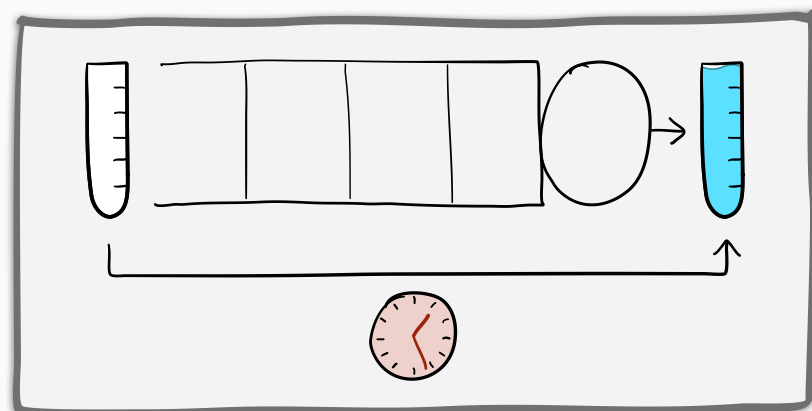
See our SIGMETRICS papers [Yu & Scully, 2024; Harlev et al., 2025; Yu et al, 2026]



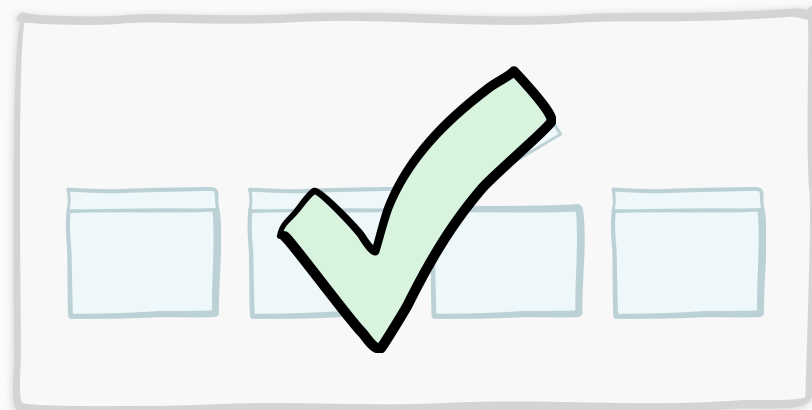
Part I:
Introducing **Gittins** via **Pandora's box**



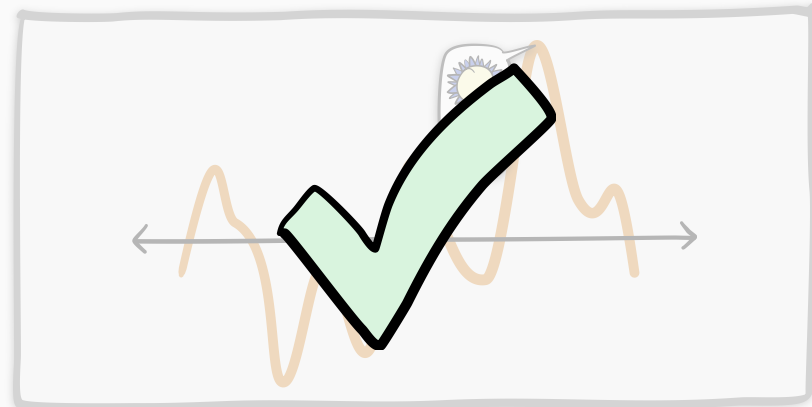
Part II: **NEW!**
Gittins for **Bayesian optimization**



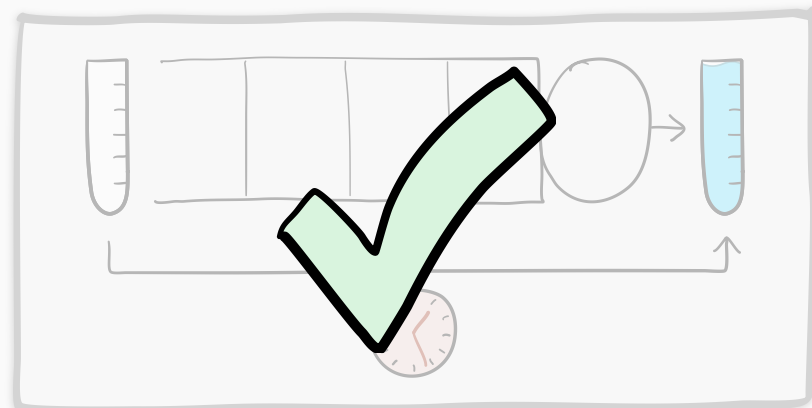
Part III: **NEW!**
Gittins for **tail scheduling** and more



Part I:
Introducing **Gittins** via **Pandora's box**



Part II: **NEW!**
Gittins for **Bayesian optimization**



Part III: **NEW!**
Gittins for **tail scheduling** and more



Open problems and
practical obstacles



Open problems and practical obstacles

- *Computation*, especially for continuous state spaces



Open problems and practical obstacles

- *Computation*, especially for continuous state spaces
- *Multiple actions* within each process, i.e. **MDP selection** (a.k.a. *bandit superprocesses*)



Open problems and practical obstacles

- *Computation*, especially for continuous state spaces
- *Multiple actions* within each process, i.e. **MDP selection** (a.k.a. *bandit superprocesses*)

e.g. **Pandora's box**
with *optional inspection*



Open problems and practical obstacles

- *Computation*, especially for continuous state spaces
- *Multiple actions* within each process, i.e. **MDP selection** (a.k.a. *bandit superprocesses*)
- “*Restless*” process state changes occurring even when not actively advanced

e.g. **Pandora's box**
with *optional inspection*



Open problems and practical obstacles

- *Computation*, especially for continuous state spaces
- *Multiple actions* within each process, i.e. **MDP selection** (a.k.a. *bandit superprocesses*)
- “*Restless*” process state changes occurring even when not actively advanced

e.g. **Pandora's box**
with *optional inspection*

more generally,
weakly coupled MDPs



Open problems and practical obstacles

- *Computation*, especially for continuous state spaces
- *Multiple actions* within each process, i.e. **MDP selection** (a.k.a. *bandit superprocesses*)
- “*Restless*” process state changes occurring even when not actively advanced
- *Model inference*, or another path to a data-driven version of **Gittins**

e.g. **Pandora's box**
with *optional inspection*

more generally,
weakly coupled MDPs

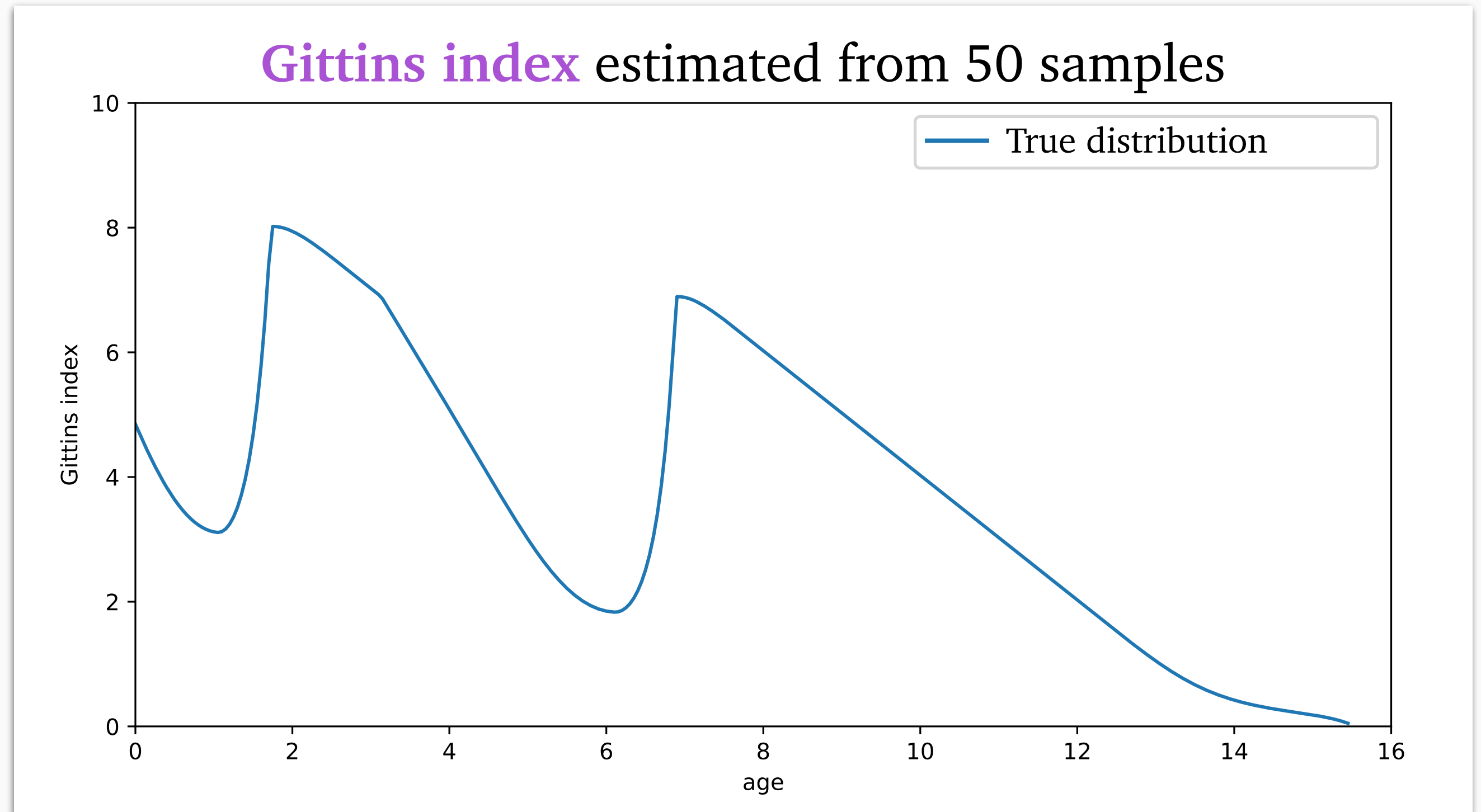
Gittins from empirical job size data



Shefali Ramakrishna



Amit Harlev



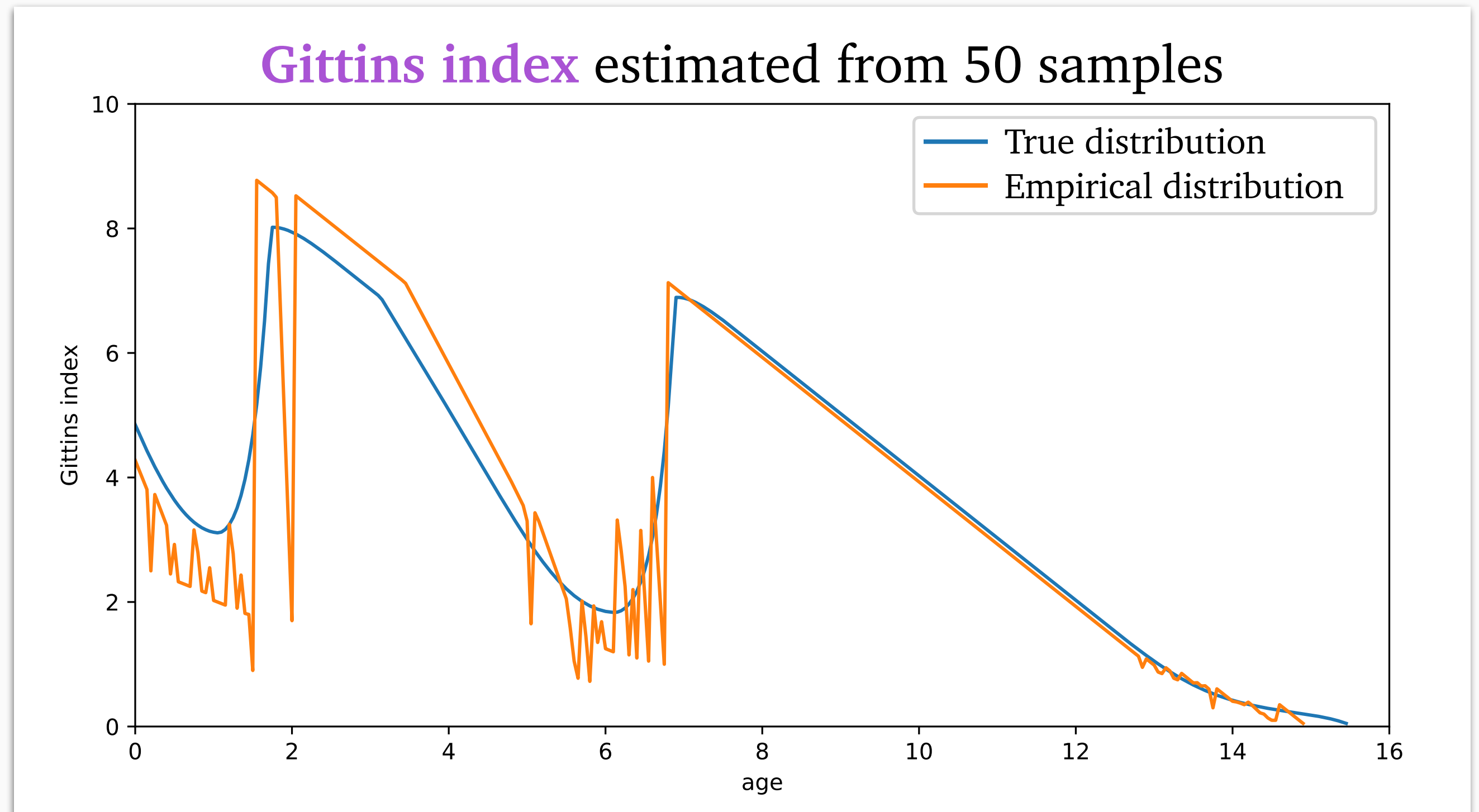
Gittins from empirical job size data



Shefali Ramakrishna



Amit Harlev



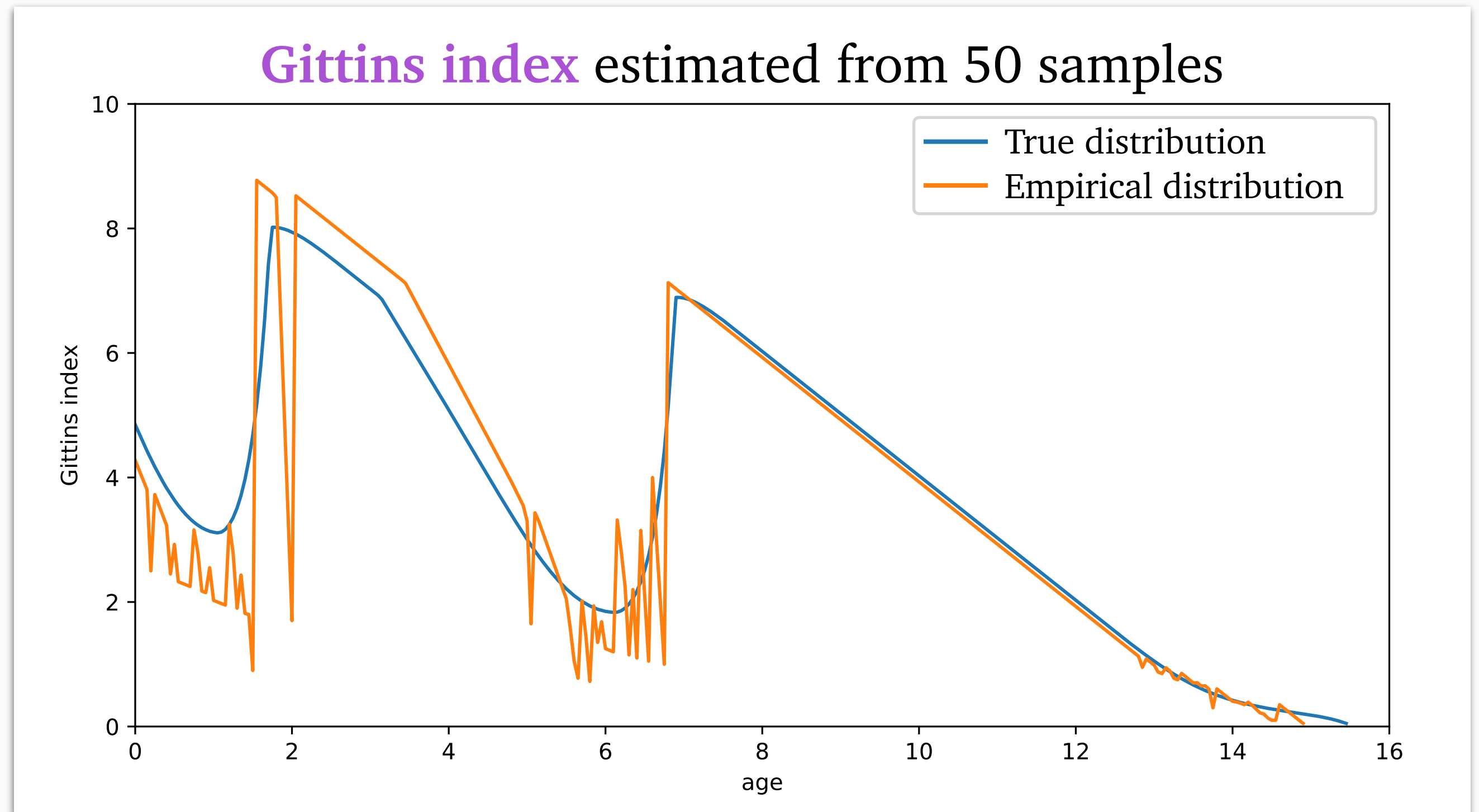
Gittins from empirical job size data



Shefali Ramakrishna



Amit Harlev



Discrete sizes: empirical works

[Ramakrishna et al., 2024]

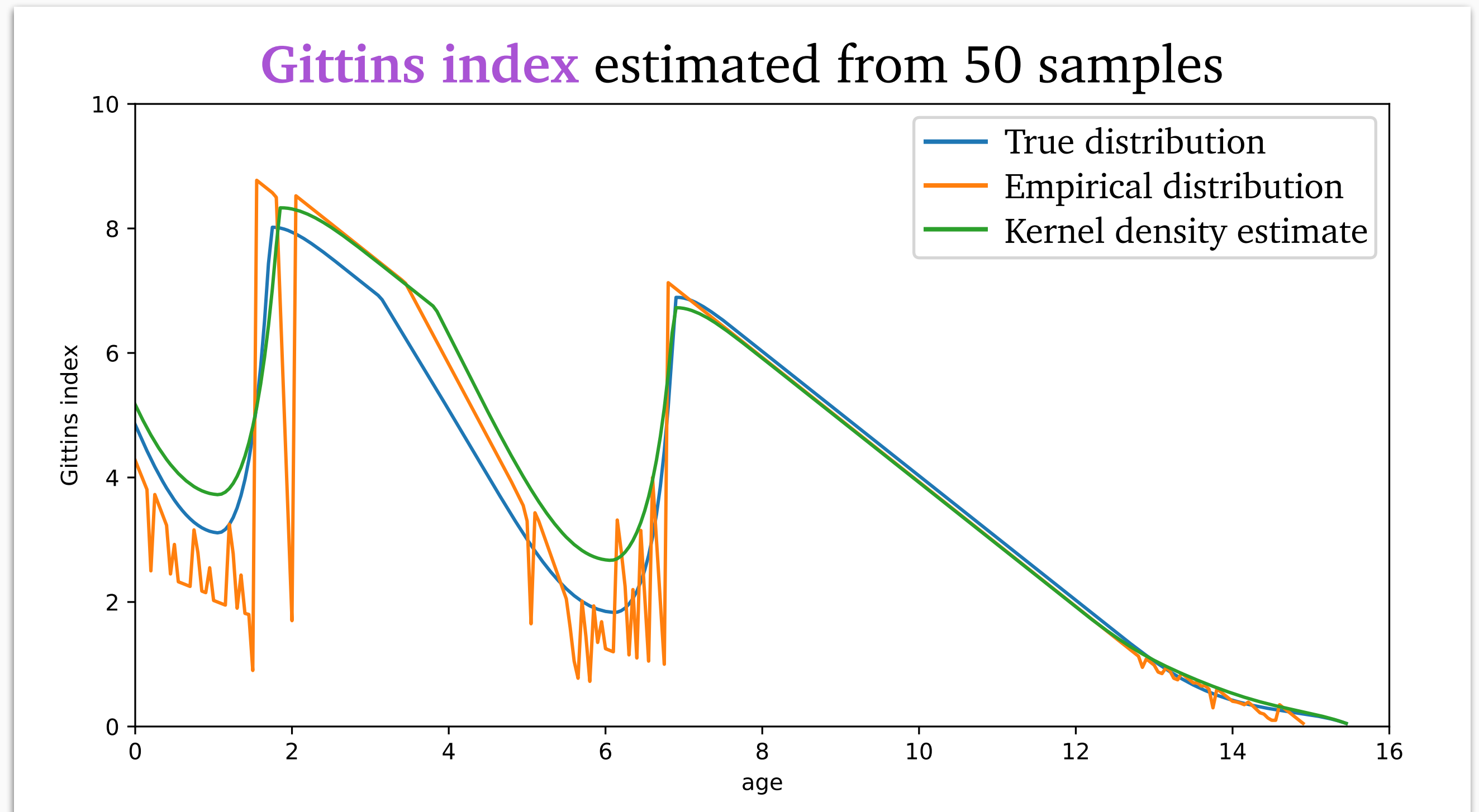
Gittins from empirical job size data



Shefali Ramakrishna



Amit Harlev



Discrete sizes: empirical works

[Ramakrishna et al., 2024]

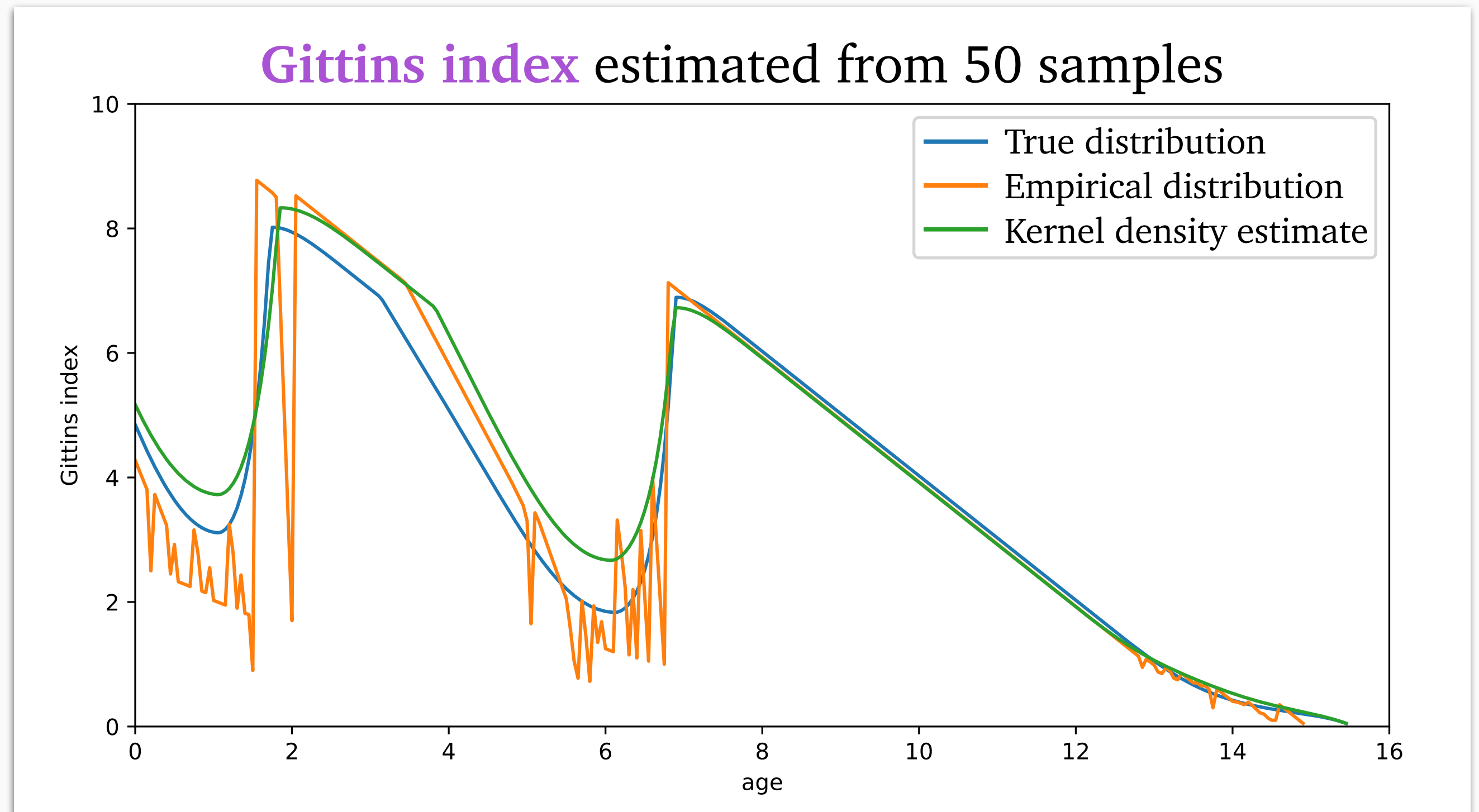
Gittins from empirical job size data



Shefali Ramakrishna



Amit Harlev

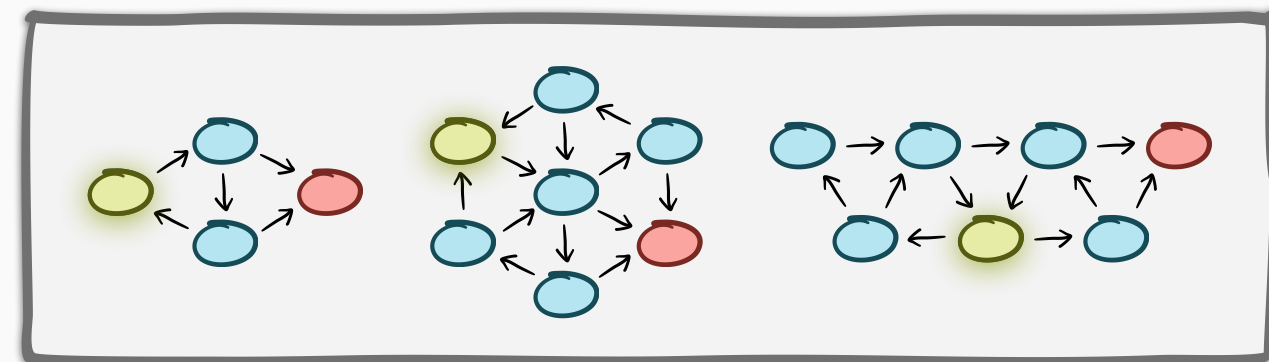


Discrete sizes: empirical works
[Ramakrishna et al., 2024]

Continuous sizes: *empirical still works*
[Ramakrishna et al., in submission]

The Gittins Index:

A Design Principle for Decision Making Under Uncertainty



Ziv Scully

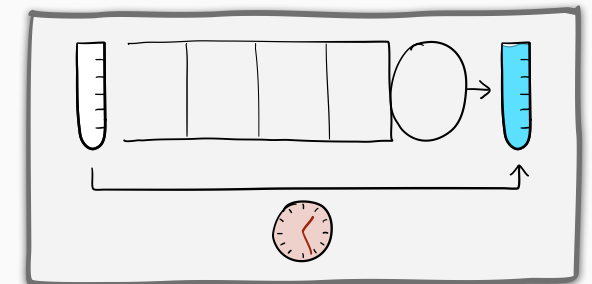
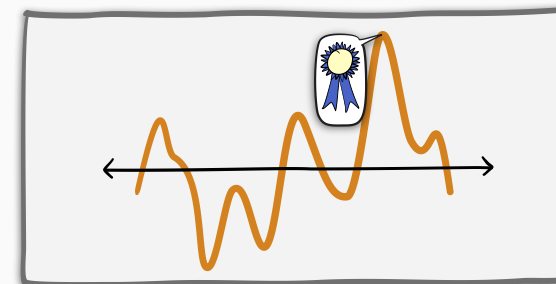
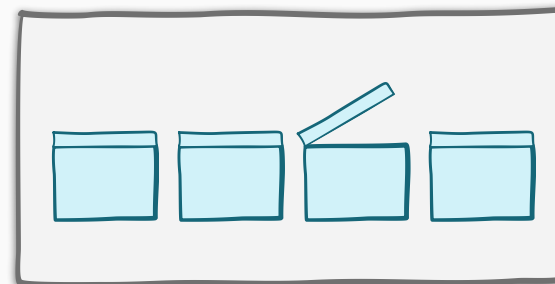
Cornell ORIE

<https://ziv.codes>

Alex Terenin

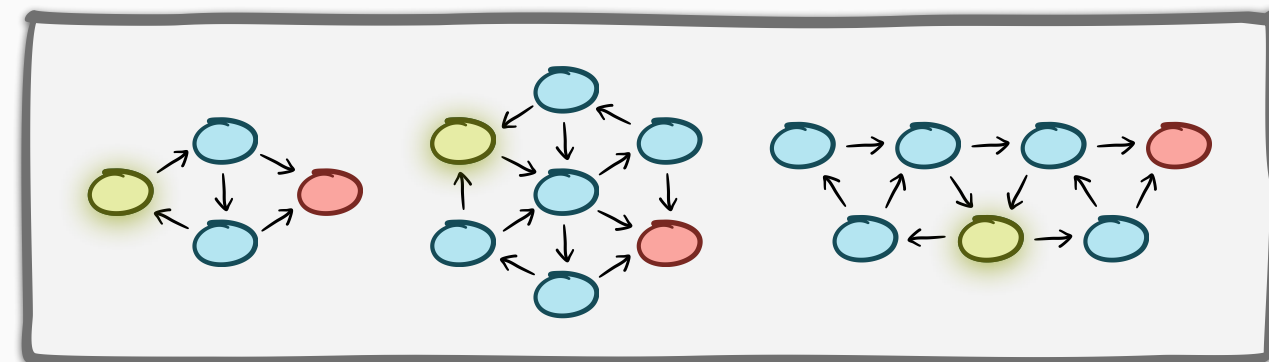
Cornell CDSES

<https://avt.im>



The Gittins Index:

A Design Principle for Decision Making Under Uncertainty



Ziv Scully

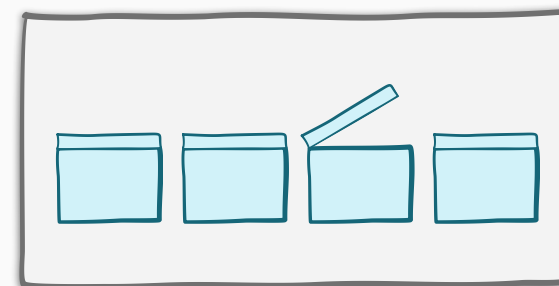
Cornell ORIE

<https://ziv.codes>

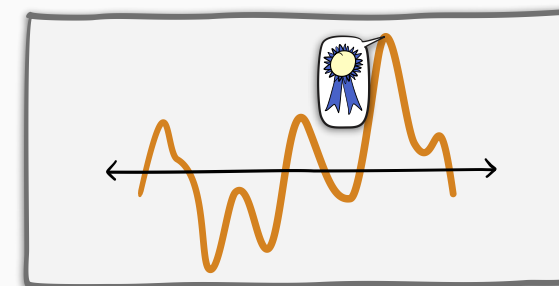
Alex Terenin

Cornell CDSES

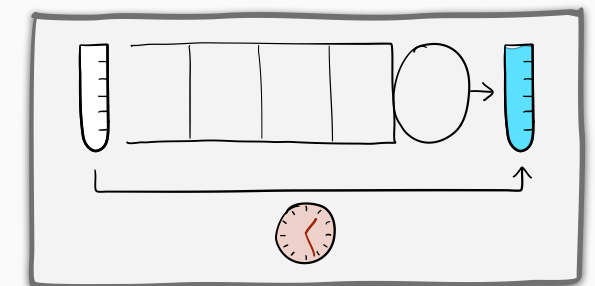
<https://avt.im>



Pandora's box



Bayesian optimization

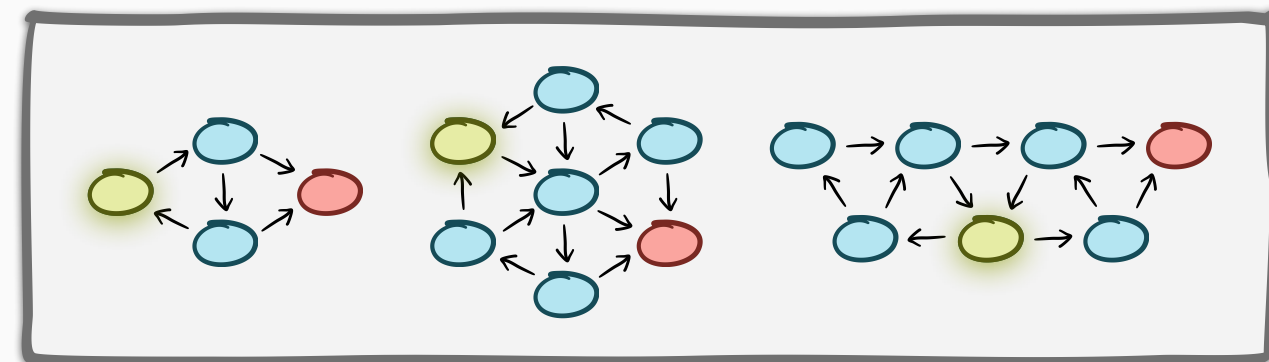


Tail scheduling

The Gittins Index:

A Design Principle for Decision Making Under Uncertainty

Markov chain selection



Ziv Scully

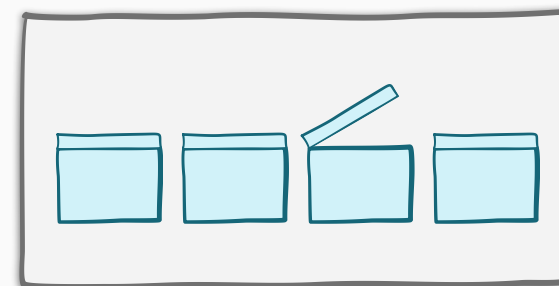
Cornell ORIE

<https://ziv.codes>

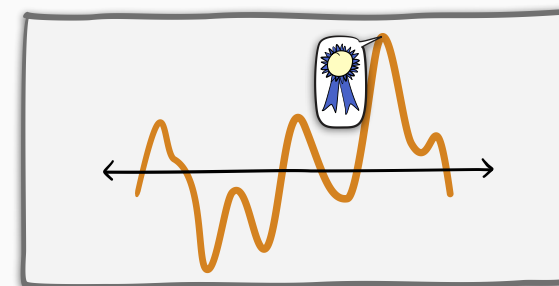
Alex Terenin

Cornell CDSES

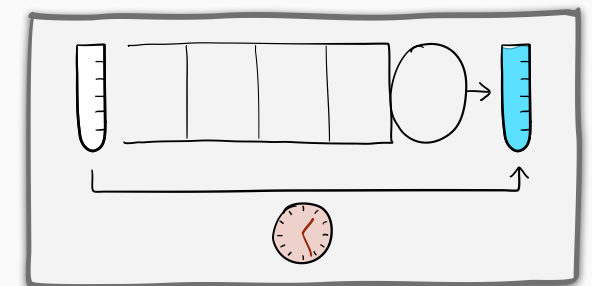
<https://avt.im>



Pandora's box



Bayesian optimization



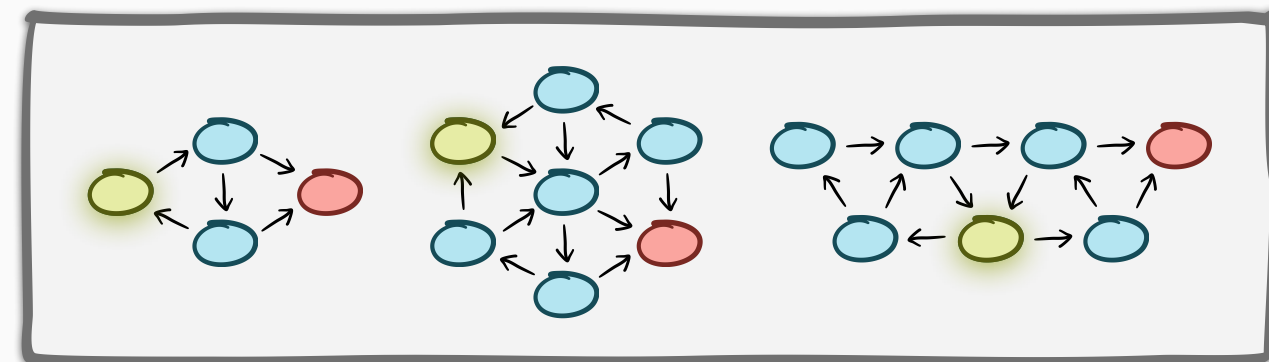
Tail scheduling

The Gittins Index

evaluate stochastic options by *calibrating to deterministic options*

A Design Principle for Decision Making Under Uncertainty

Markov chain selection



Ziv Scully

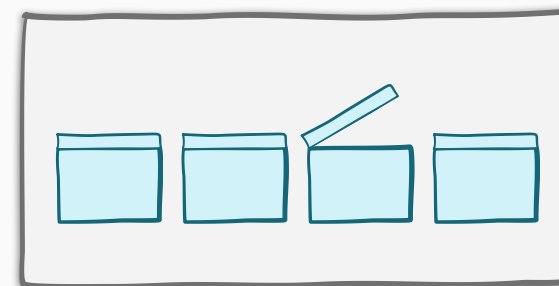
Cornell ORIE

<https://ziv.codes>

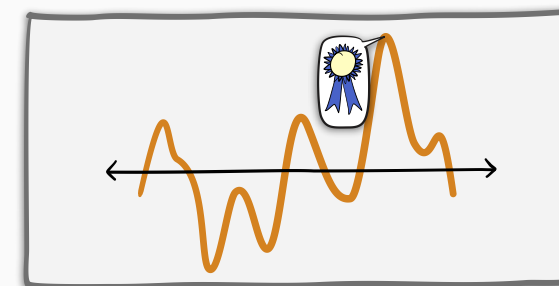
Alex Terenin

Cornell CDSES

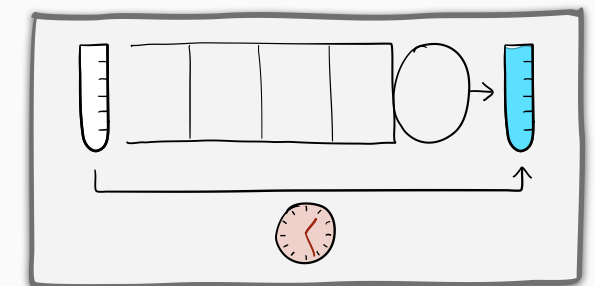
<https://avt.im>



Pandora's box

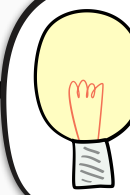


Bayesian optimization



Tail scheduling

The Gittins Index



evaluate stochastic options by *calibrating* to deterministic options

A Design Principle for Decision Making Under Uncertainty



arXiv:2506.10872

Ziv Scully

Cornell ORIE

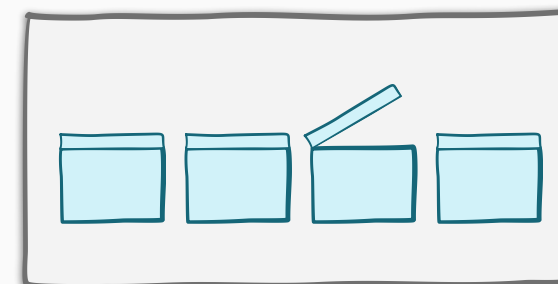
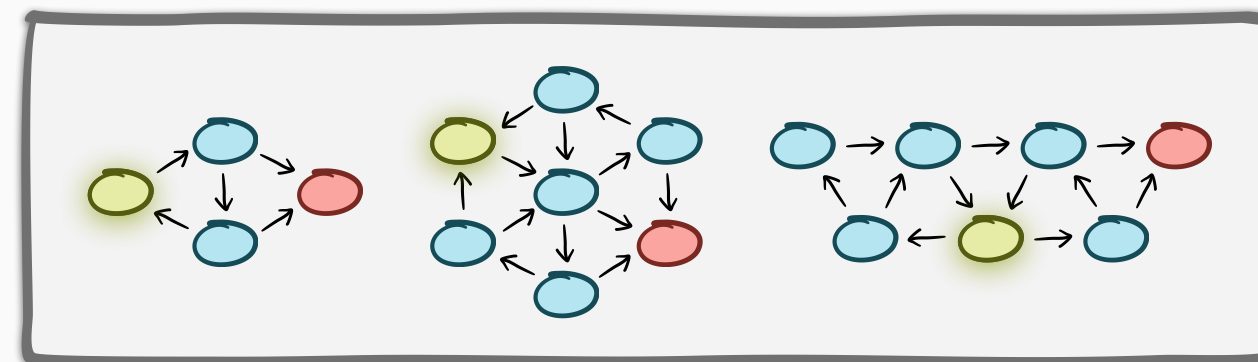
<https://ziv.codes>

Alex Terenin

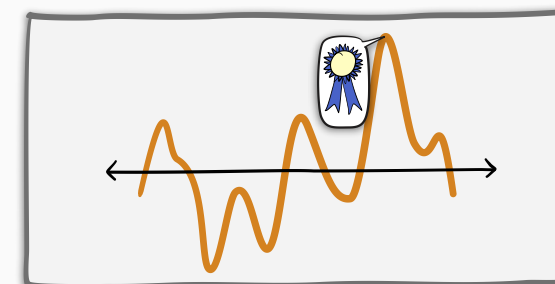
Cornell CDSES

<https://avt.im>

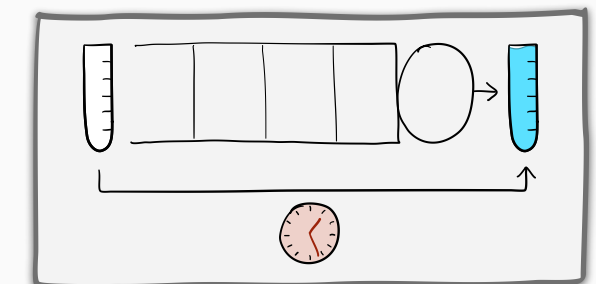
Markov chain selection



Pandora's box



Bayesian optimization

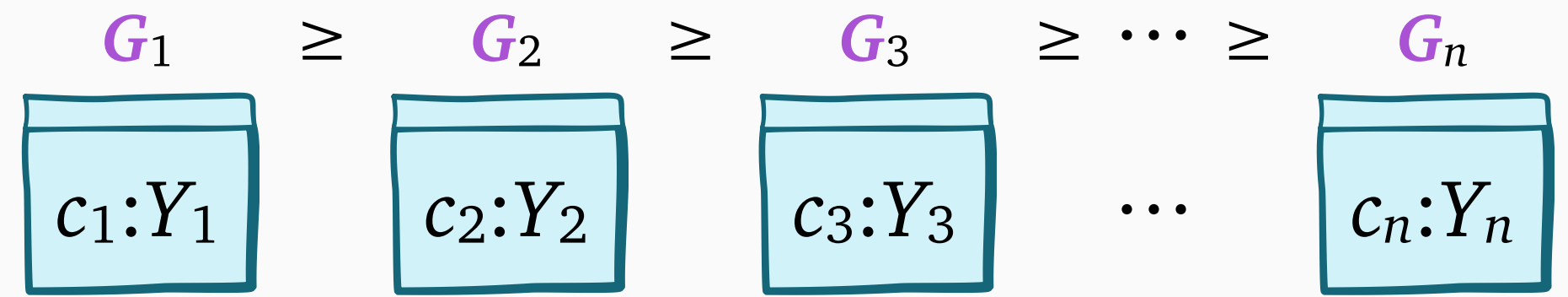


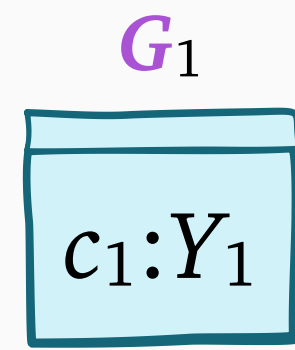
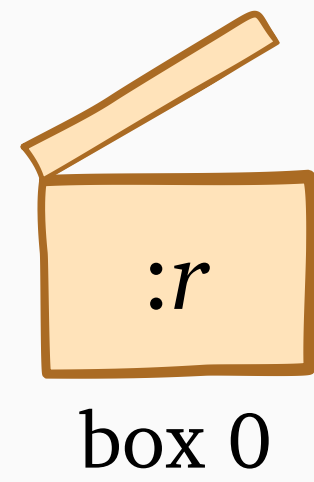
Tail scheduling

Bonus slides

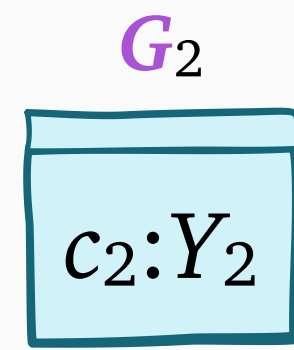
- Optimality proof sketch
- Quizzes
- **Pandora's box** with optional inspection

Optimality proof sketch

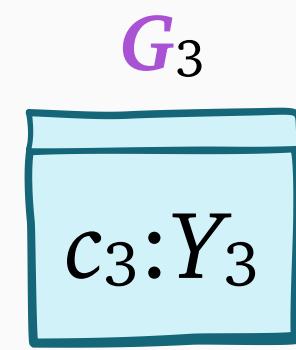




$\geq$



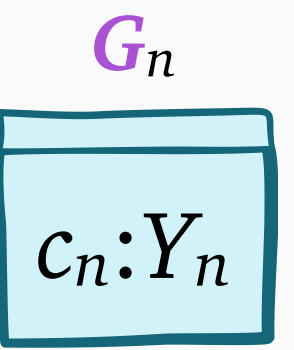
$\geq$



$\geq$

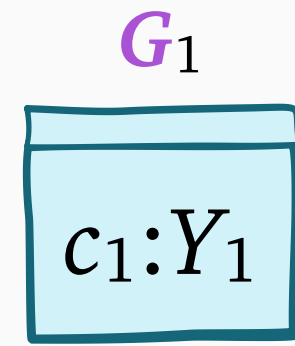
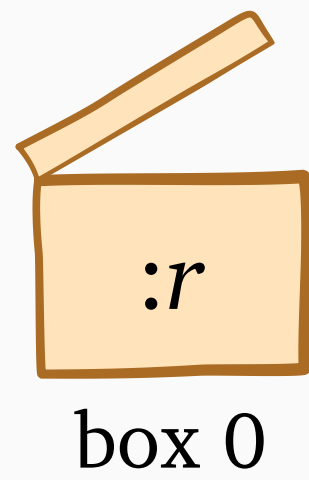
$\dots$

$\geq$

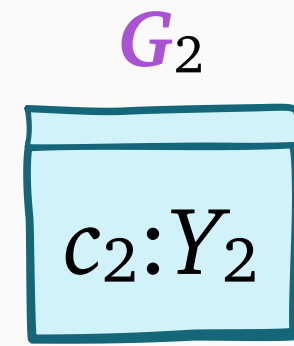


$\dots$

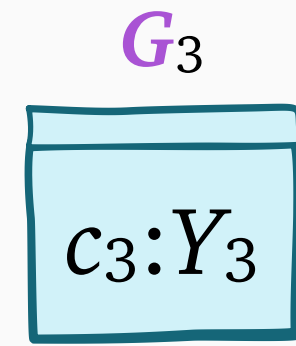
Approach: start at $r = \infty$,
then decrease to $r = 0$



$\geq$



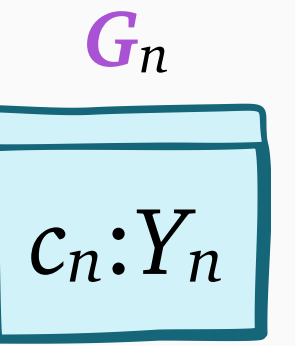
$\geq$



$\geq$

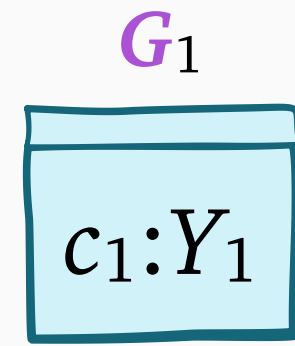
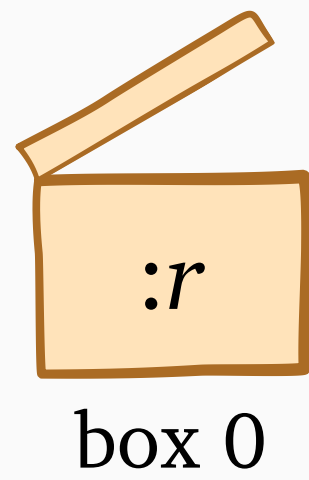
$\dots$

$\geq$

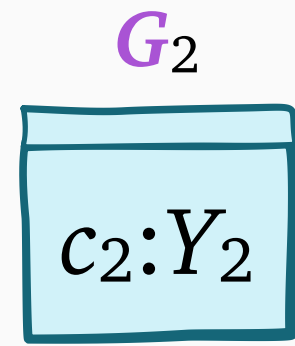


Approach: start at $r = \infty$,
then decrease to $r = 0$

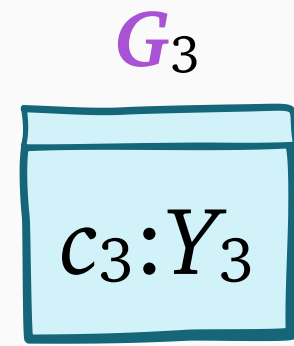
- $r \geq G_1$:



$\geq$



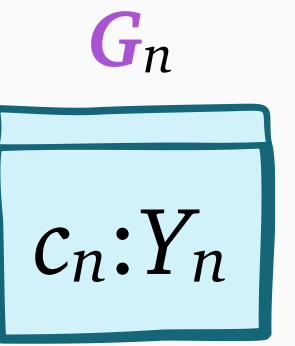
$\geq$



$\geq$

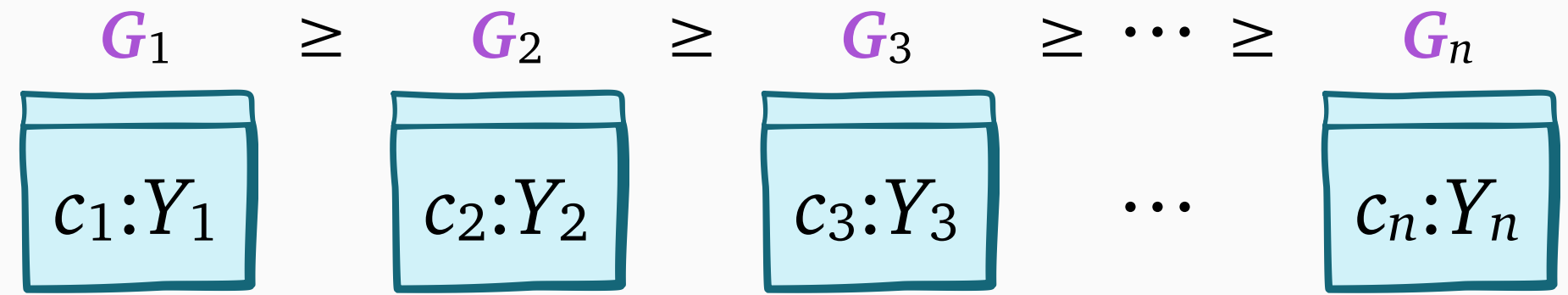
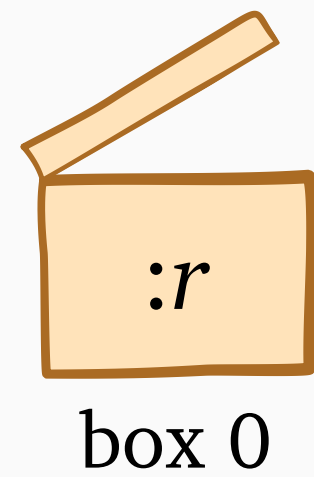
$\dots$

$\geq$

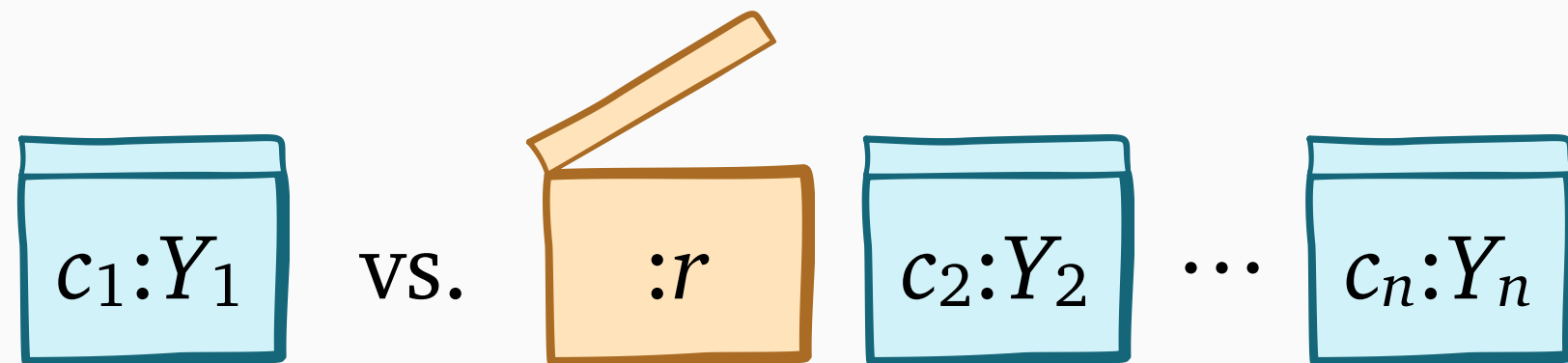


Approach: start at $r = \infty$,
then decrease to $r = 0$

- $r \geq G_1$:

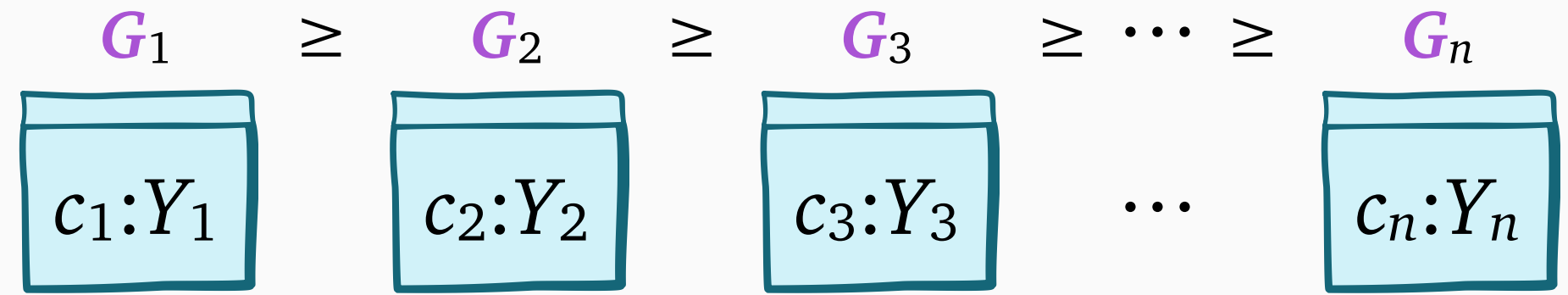
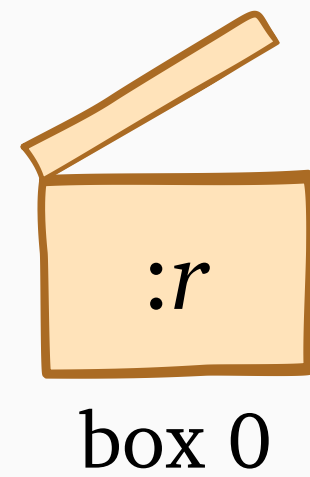


Box 1's perspective

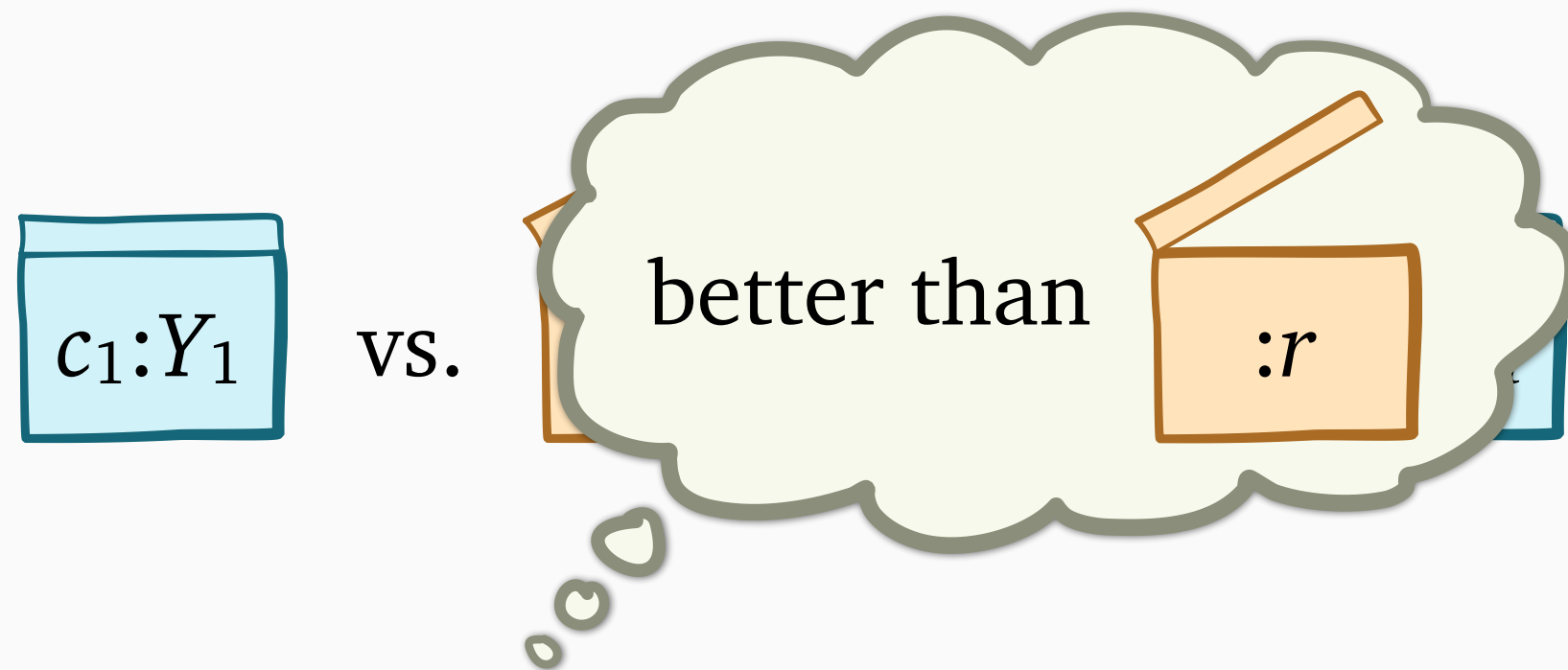


Approach: start at $r = \infty$,
then decrease to $r = 0$

- $r \geq G_1$:

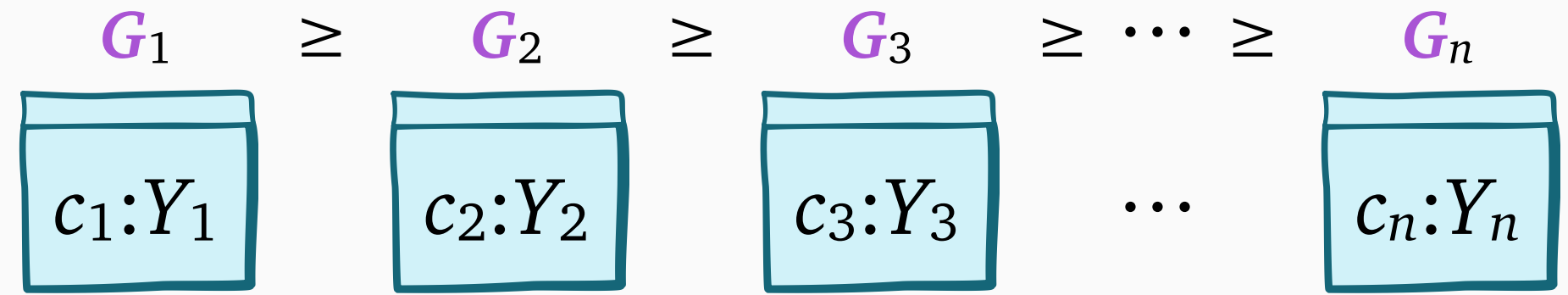
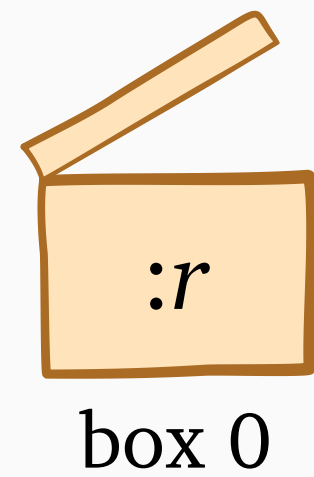


Box 1's perspective

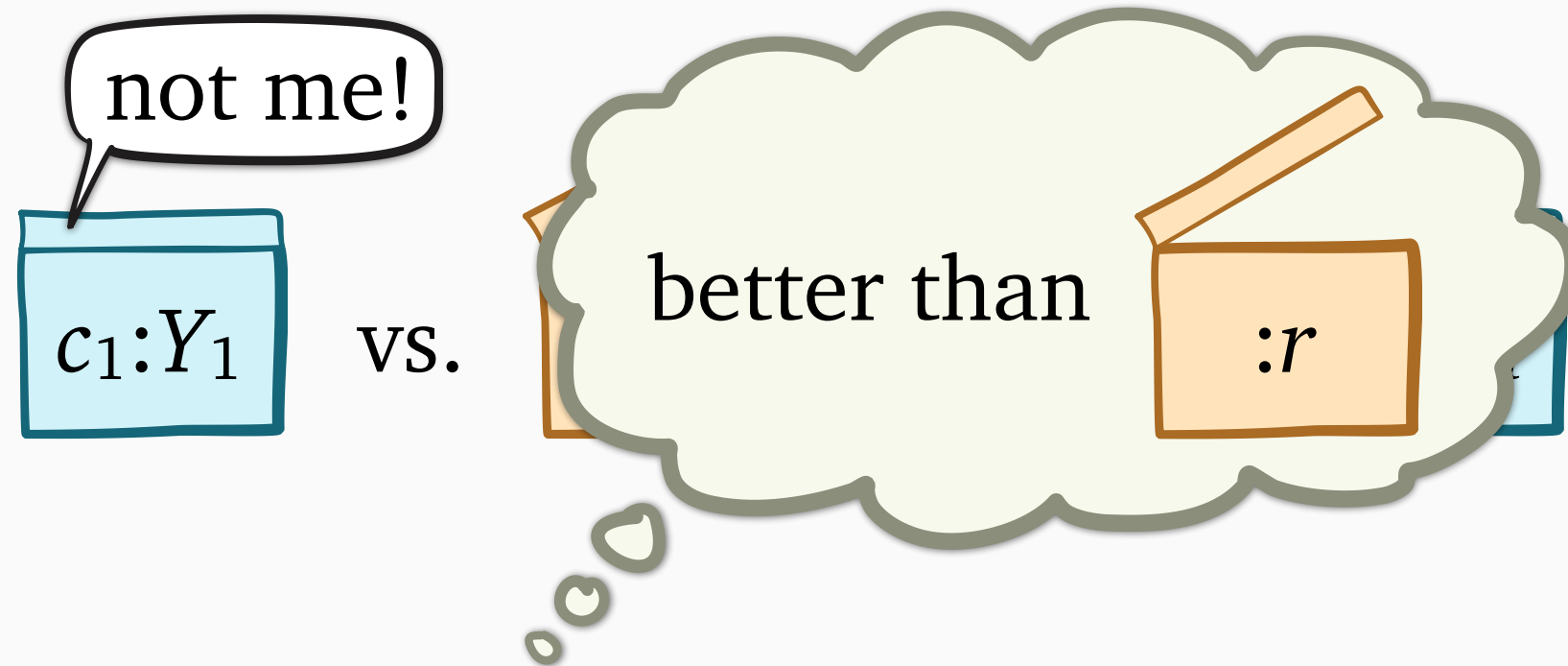


Approach: start at $r = \infty$,
then decrease to $r = 0$

- $r \geq G_1$:

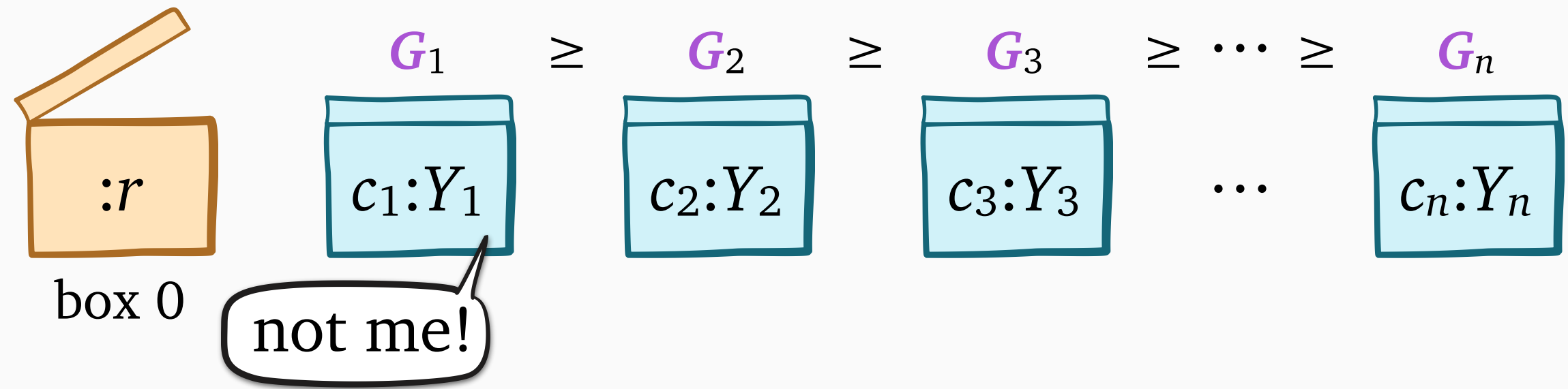


Box 1's perspective

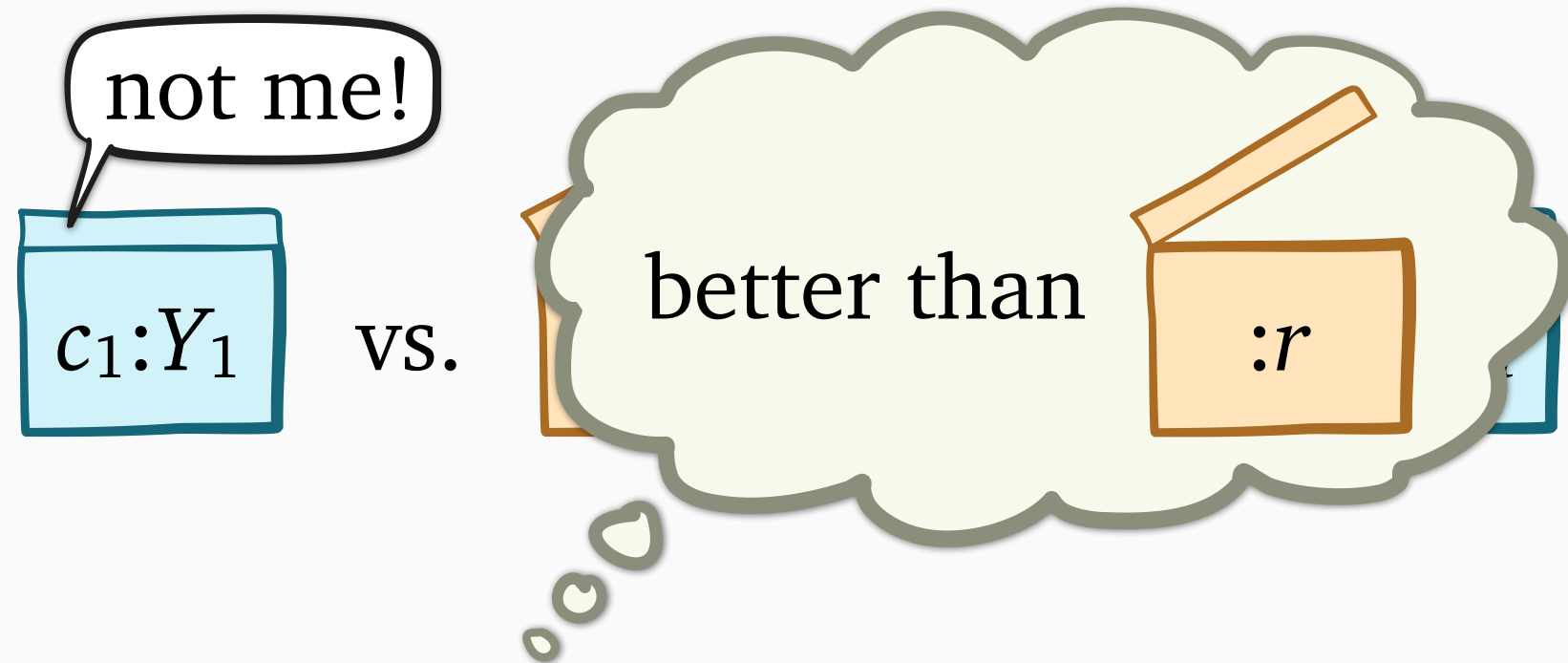


Approach: start at $r = \infty$,
then decrease to $r = 0$

- $r \geq G_1$:

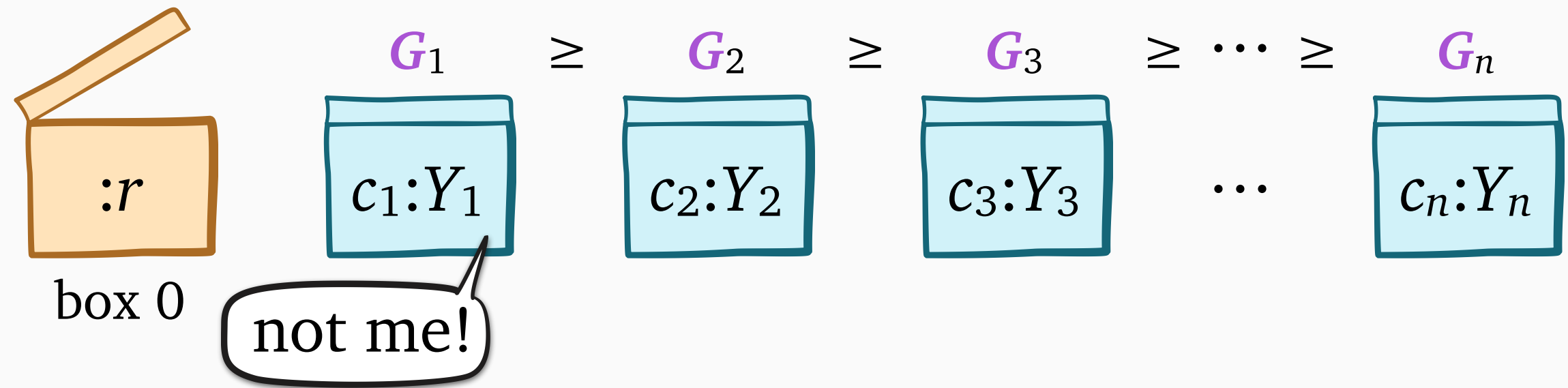


Box 1's perspective

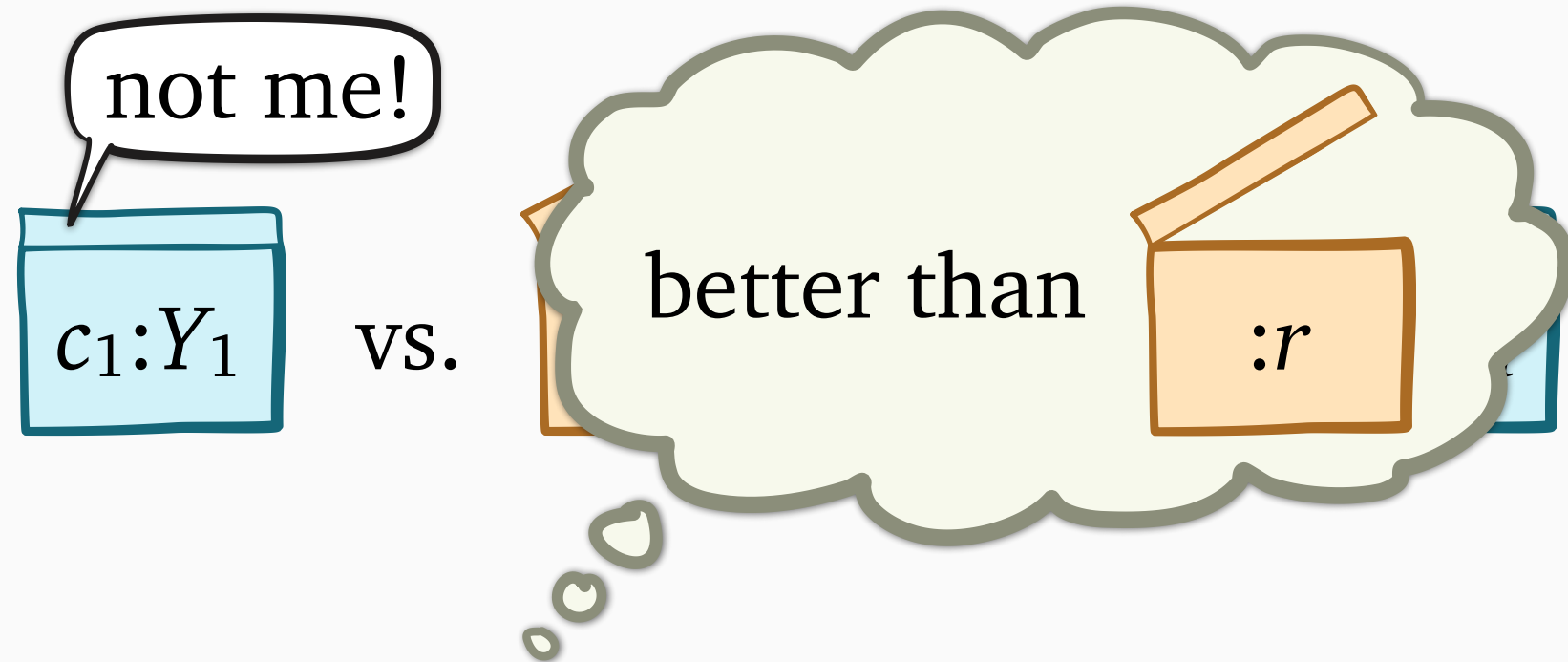


Approach: start at $r = \infty$,
then decrease to $r = 0$

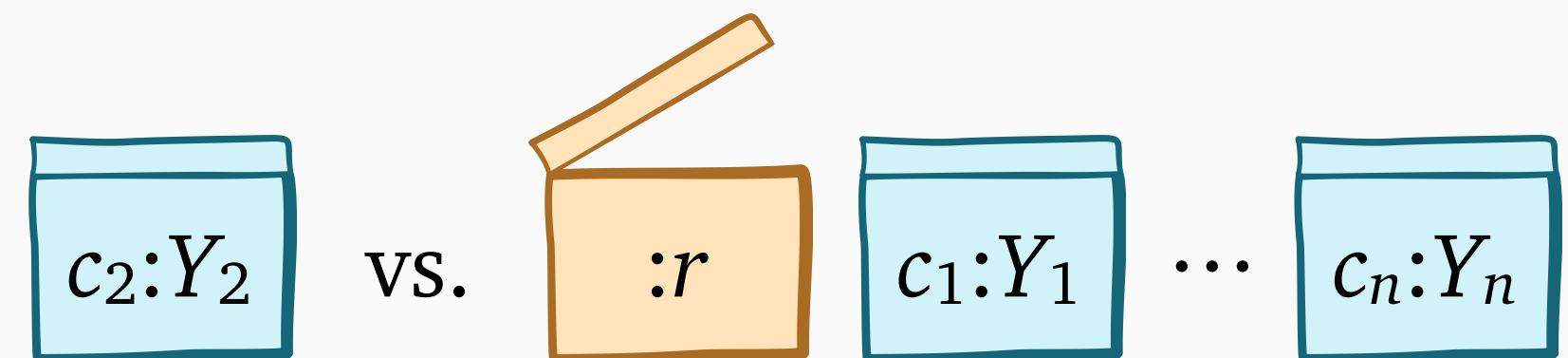
- $r \geq G_1$:



Box 1's perspective

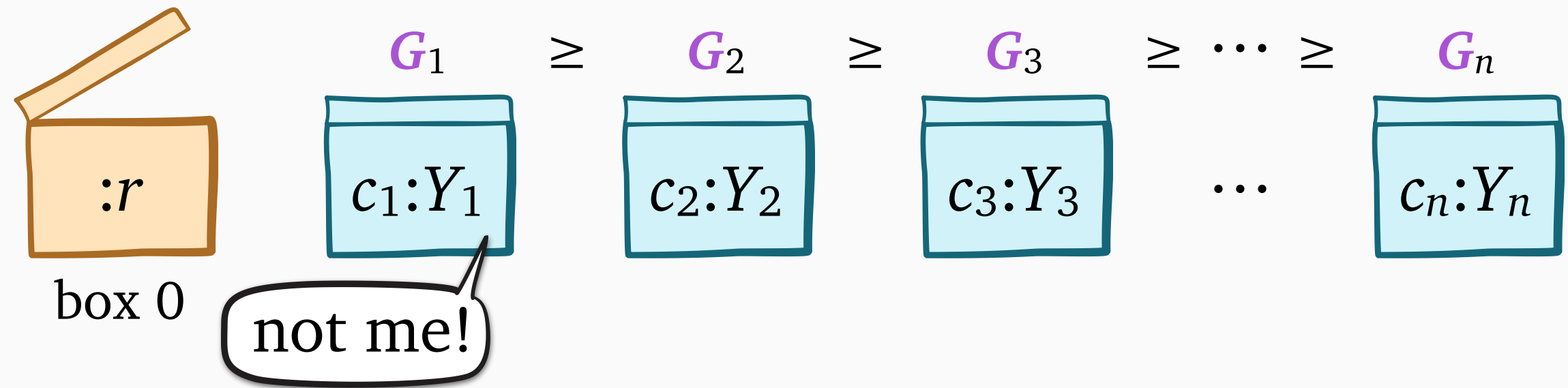


Box 2's perspective

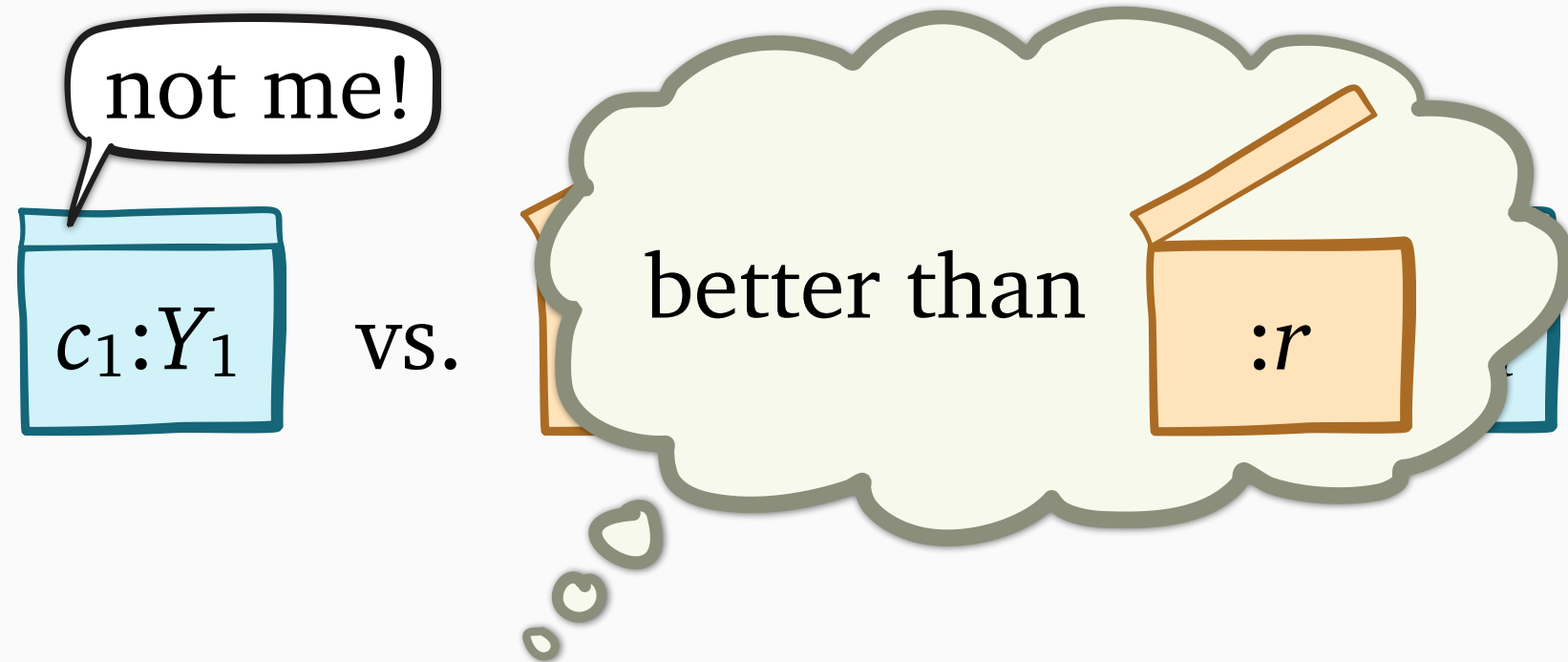


Approach: start at $r = \infty$,
then decrease to $r = 0$

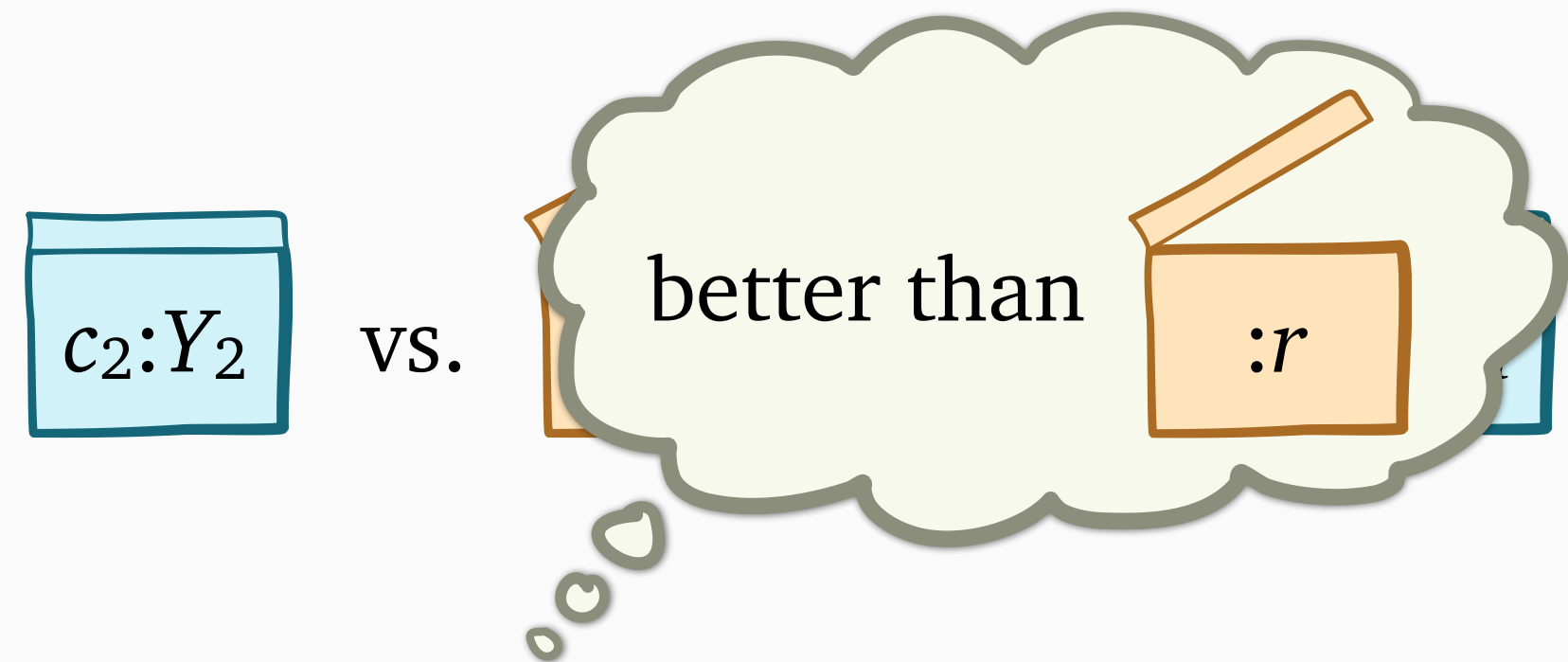
- $r \geq G_1$:



Box 1's perspective

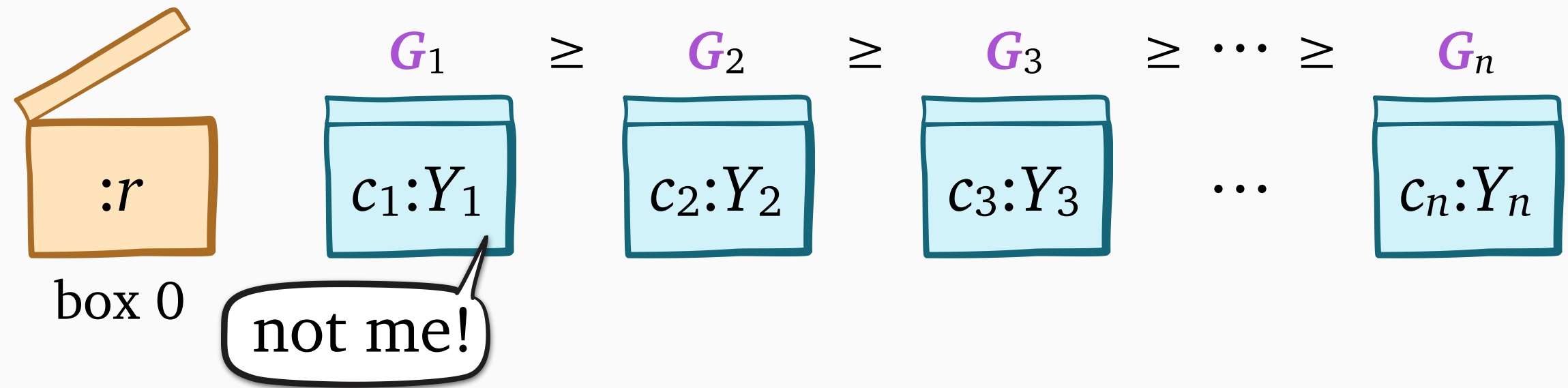


Box 2's perspective

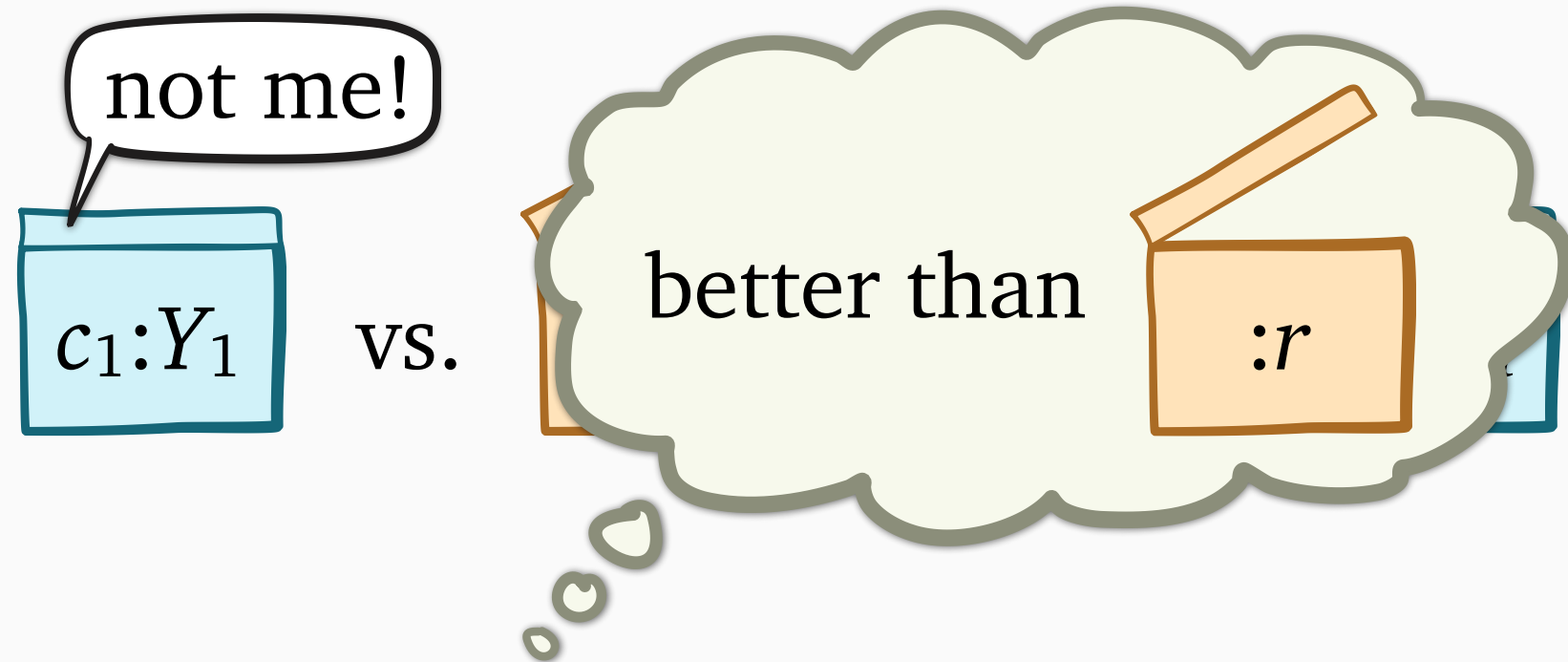


Approach: start at $r = \infty$,
then decrease to $r = 0$

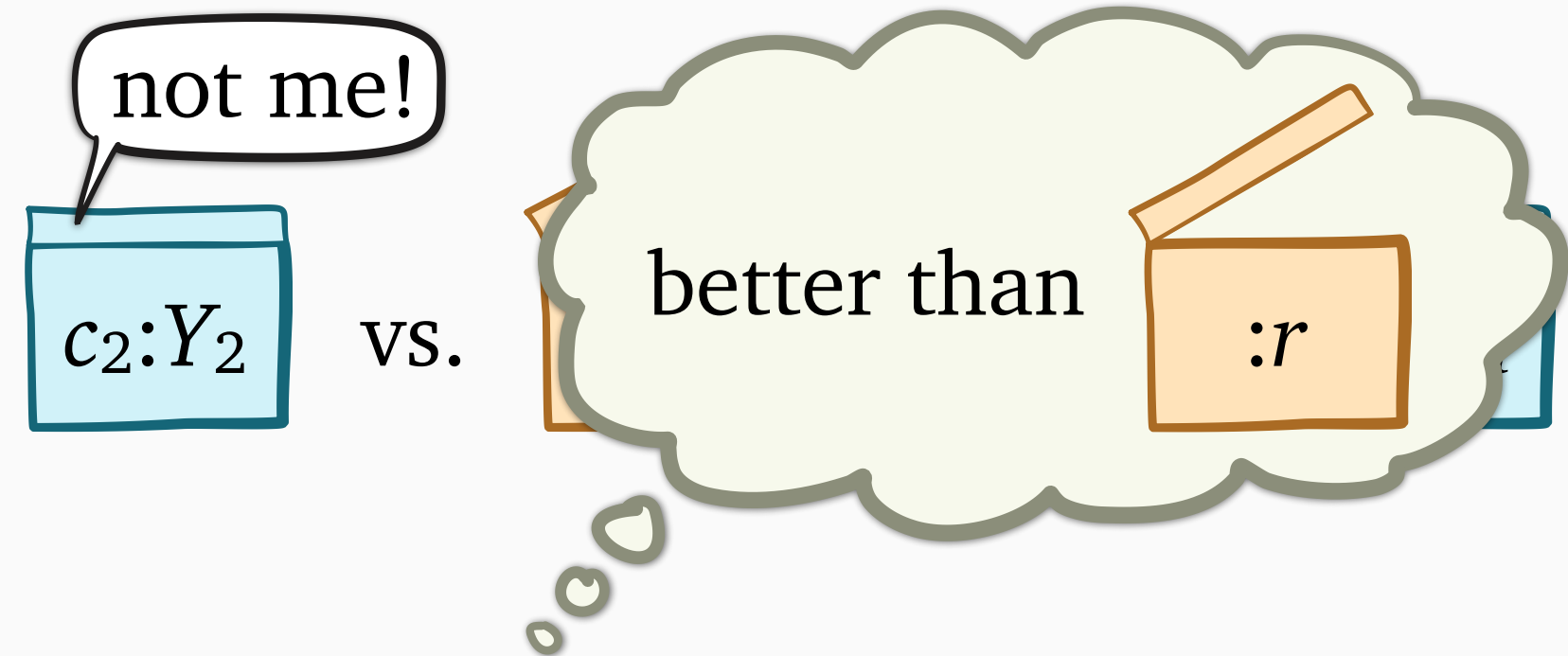
- $r \geq G_1$:



Box 1's perspective

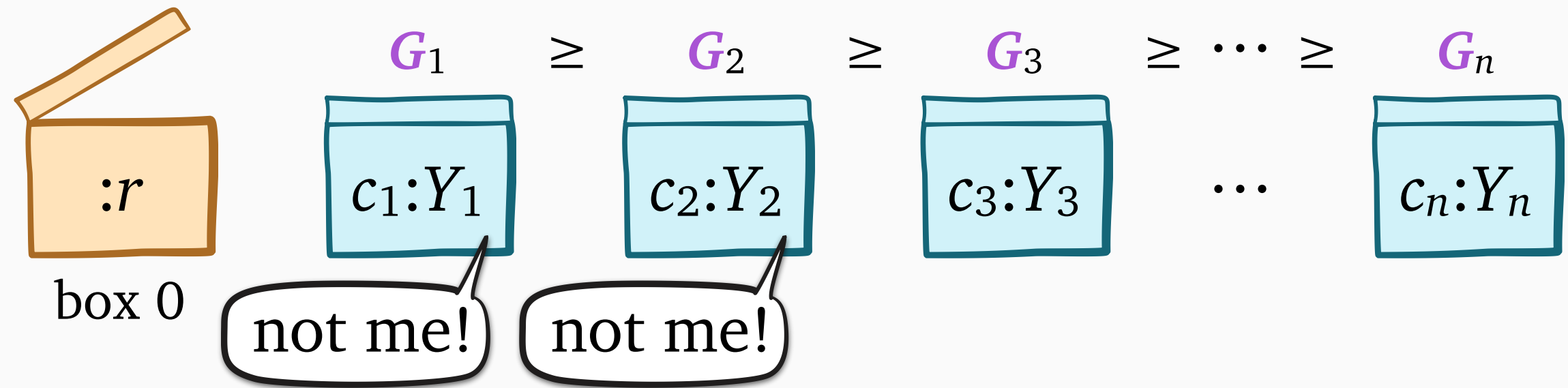


Box 2's perspective

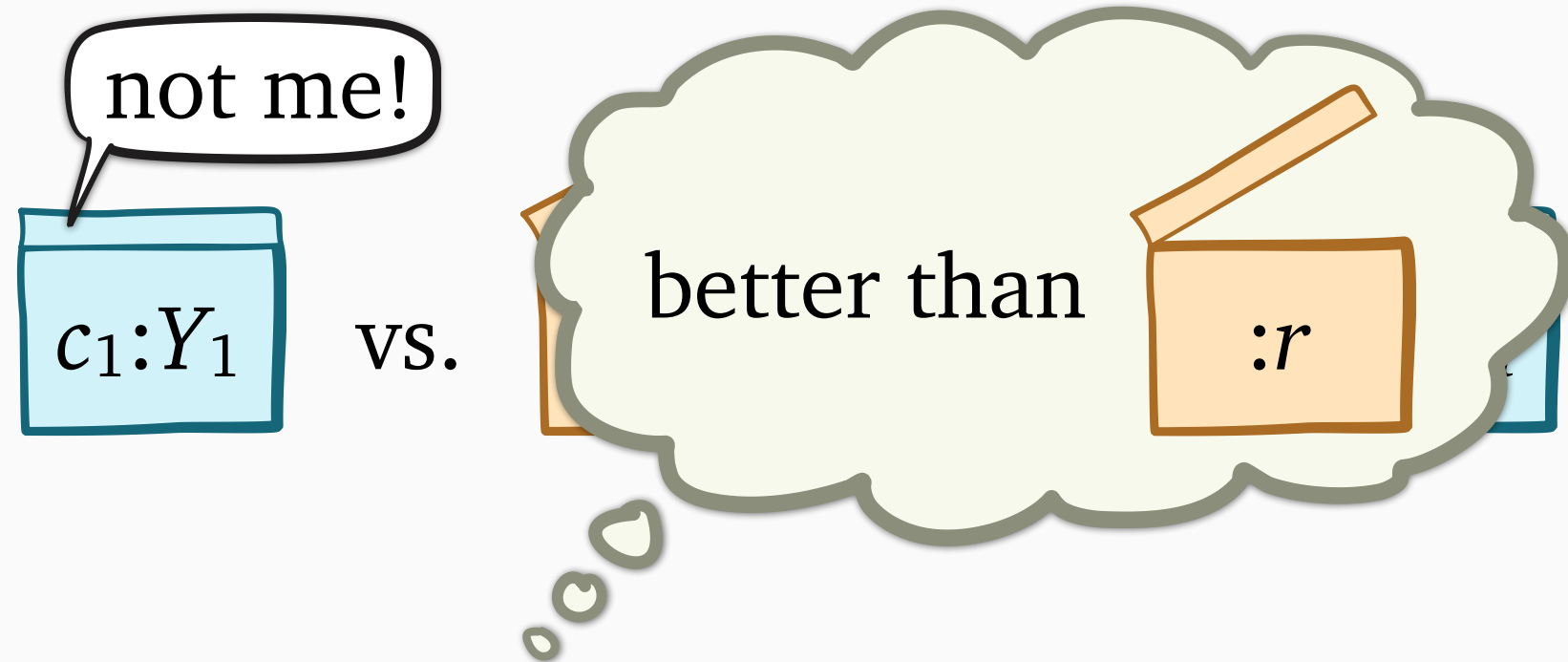


Approach: start at $r = \infty$,
then decrease to $r = 0$

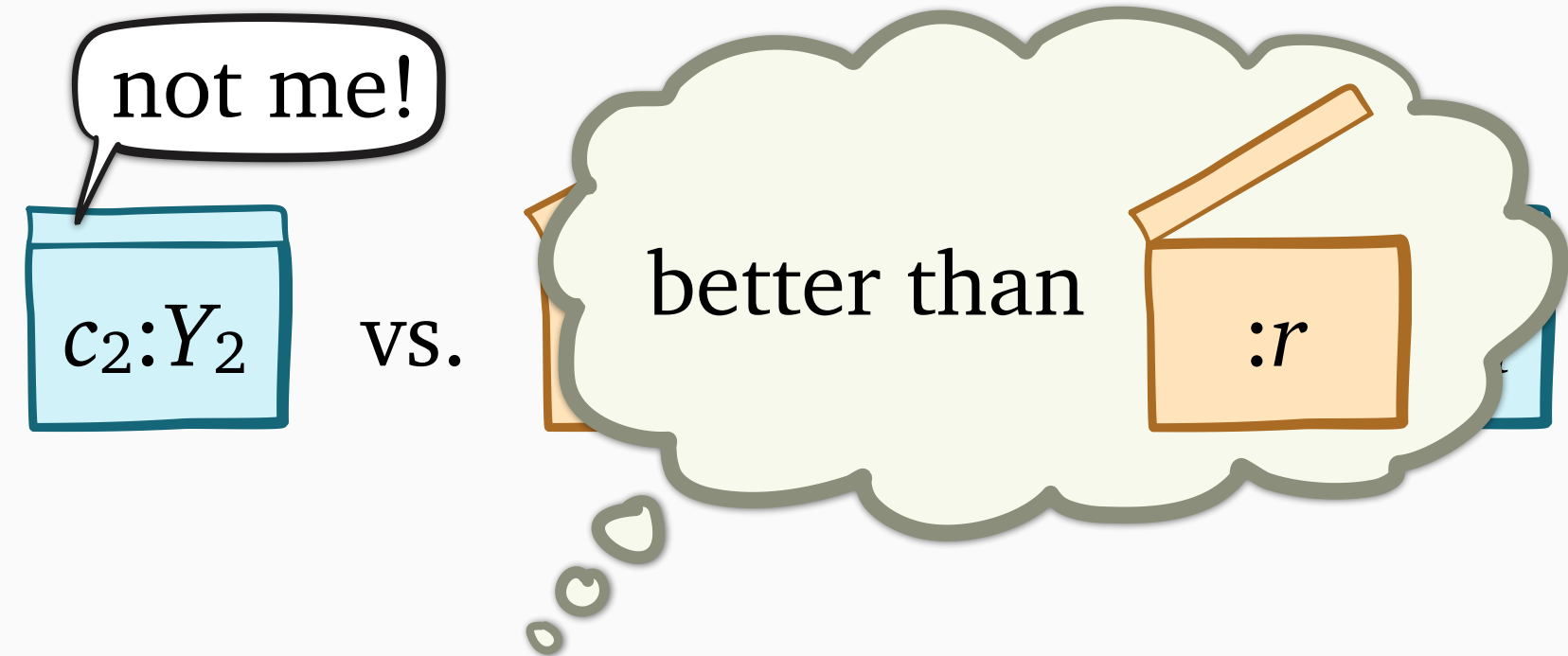
- $r \geq G_1$:



Box 1's perspective

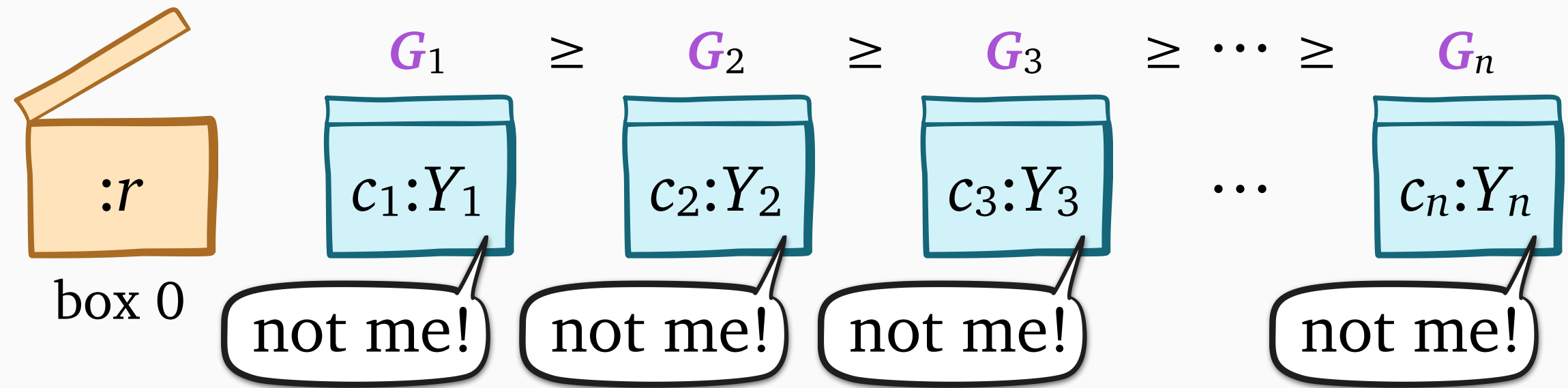


Box 2's perspective

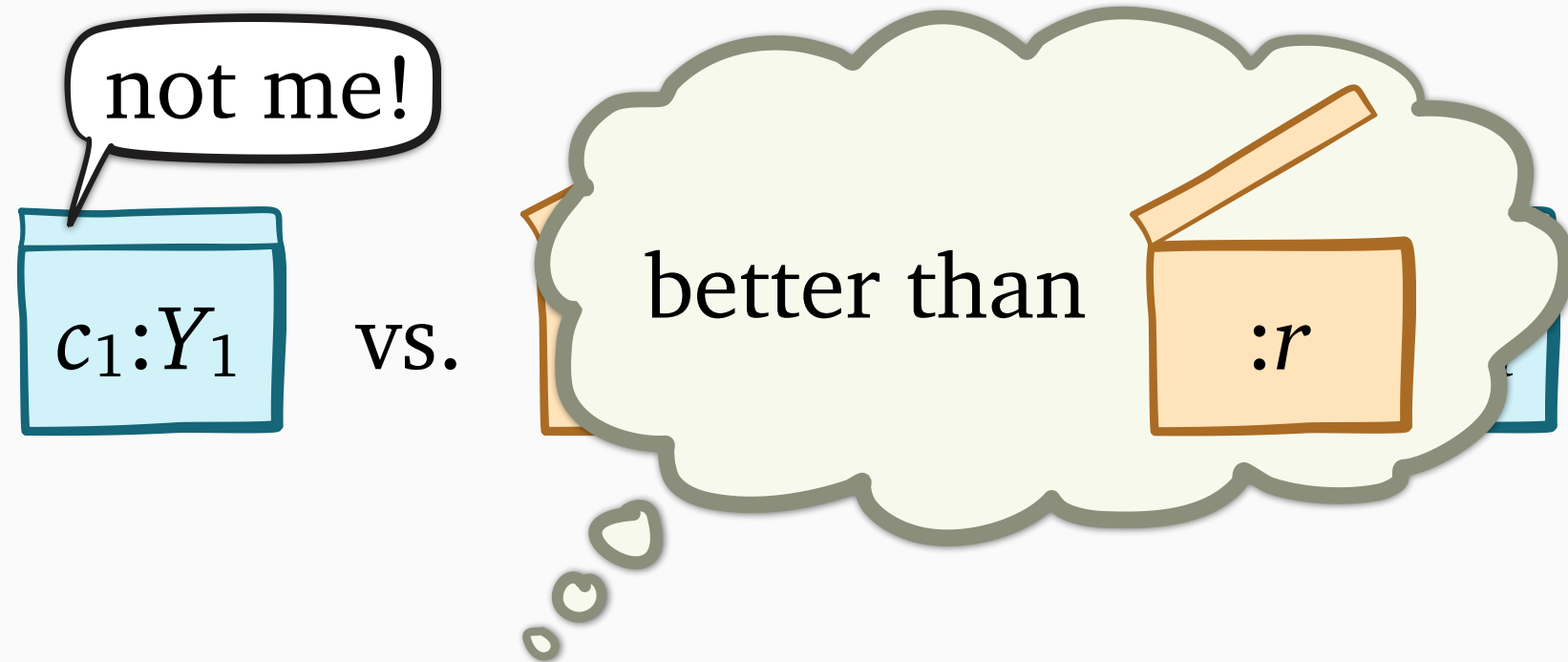


Approach: start at $r = \infty$,
then decrease to $r = 0$

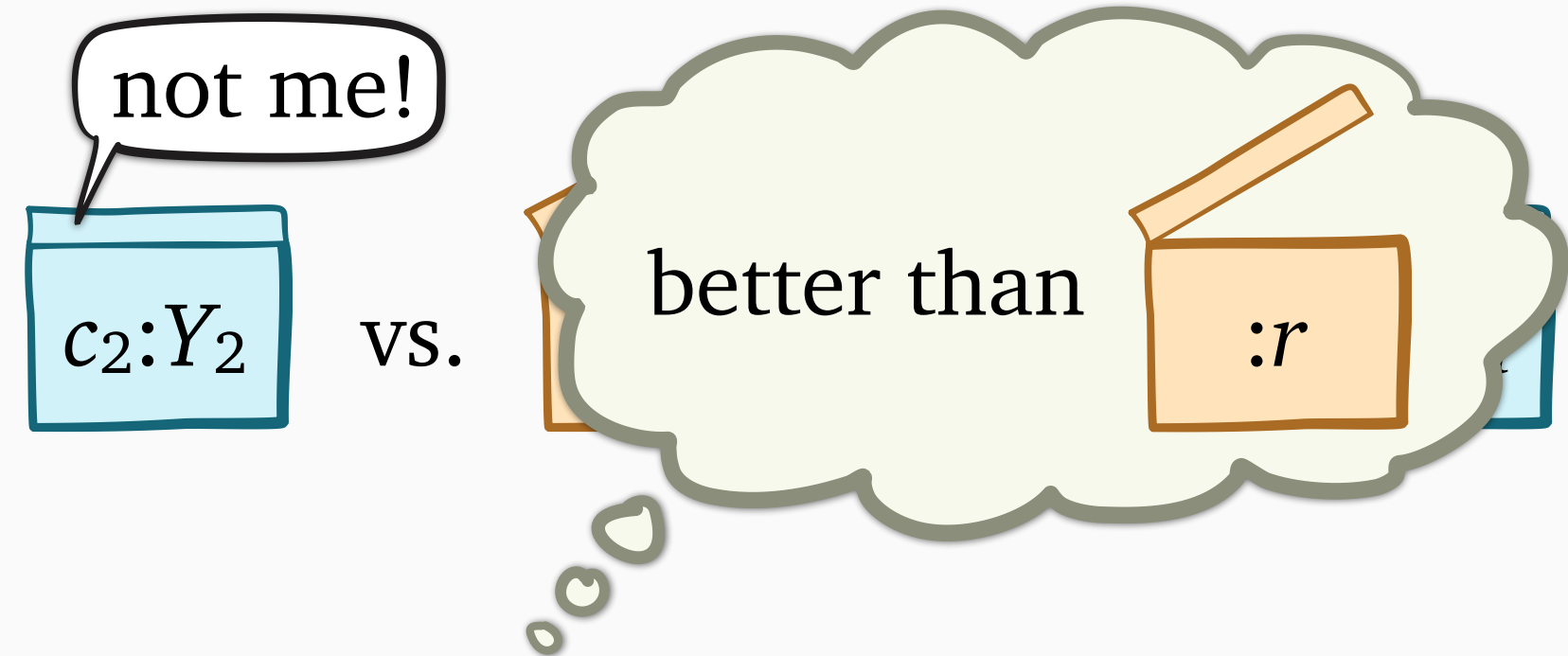
- $r \geq G_1$:



Box 1's perspective

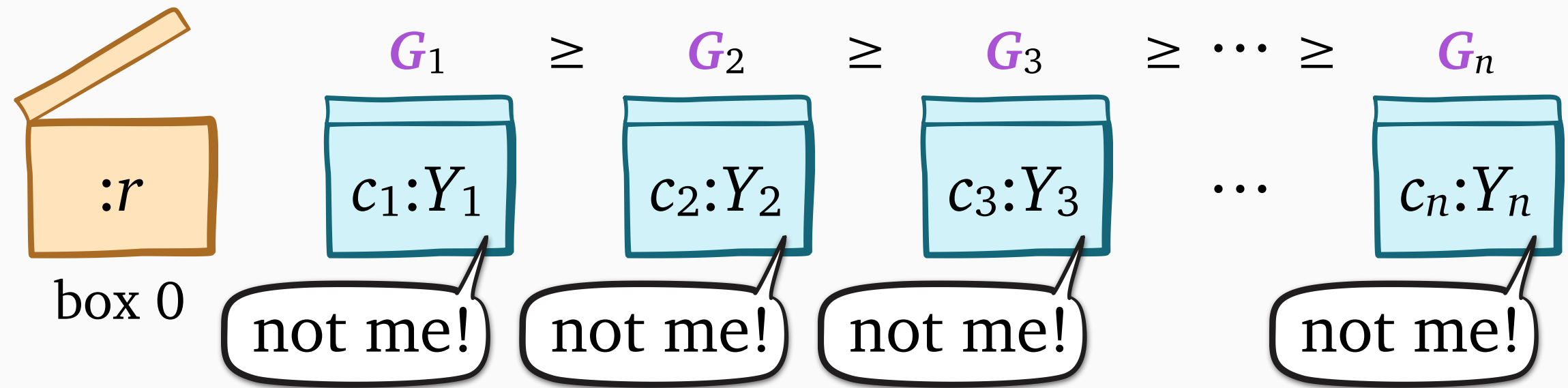


Box 2's perspective

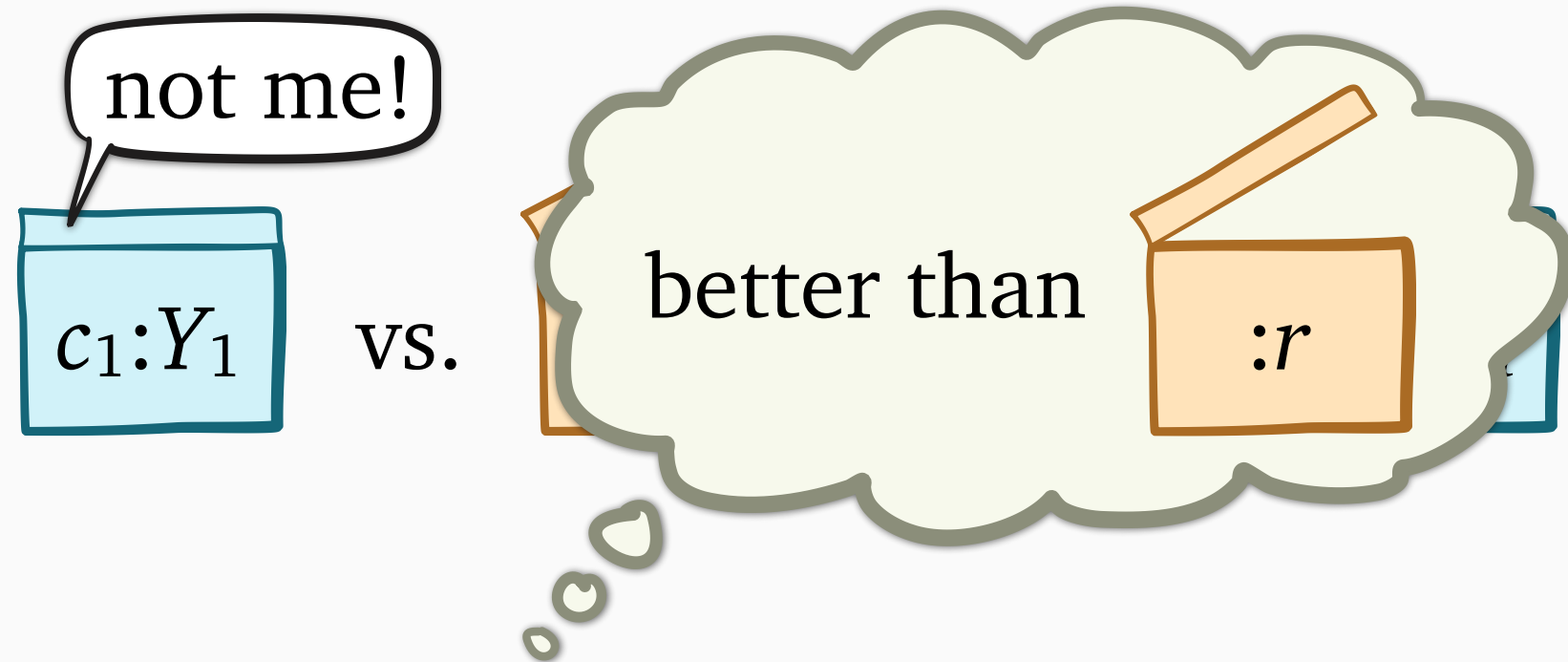


Approach: start at $r = \infty$,
then decrease to $r = 0$

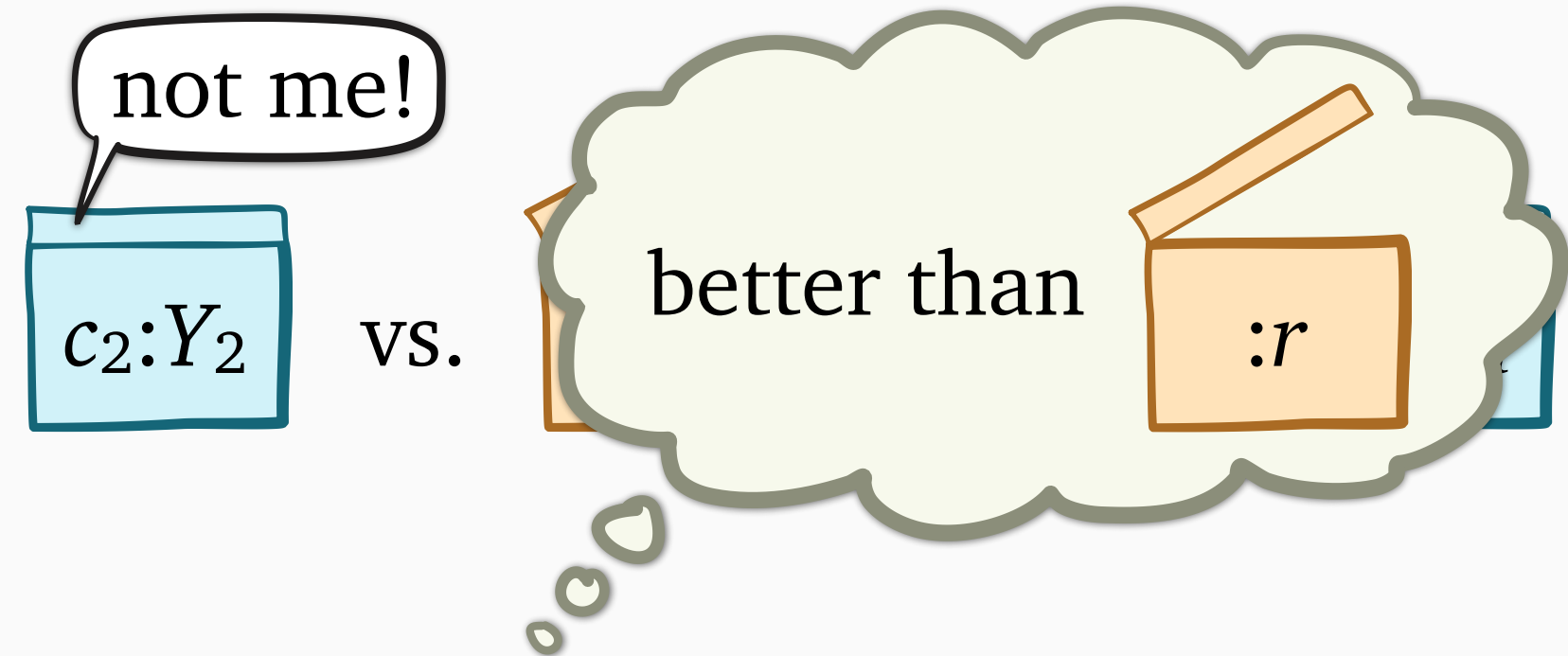
- $r \geq G_1$: select box 0



Box 1's perspective

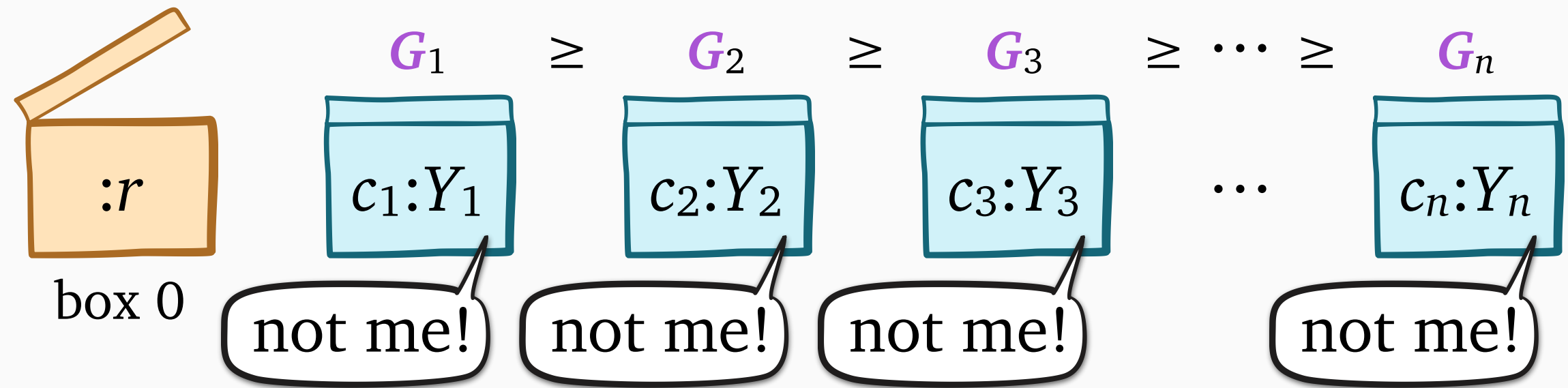


Box 2's perspective

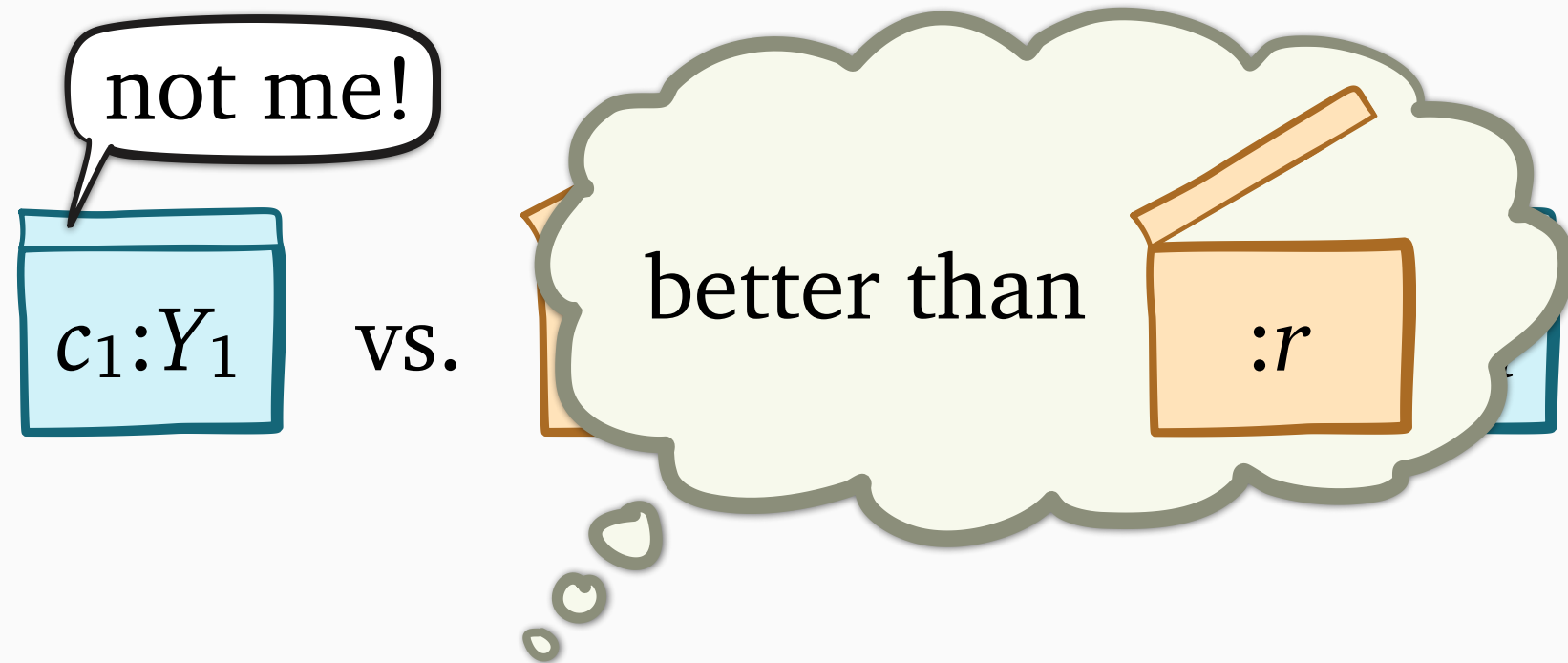


Approach: start at $r = \infty$,
then decrease to $r = 0$

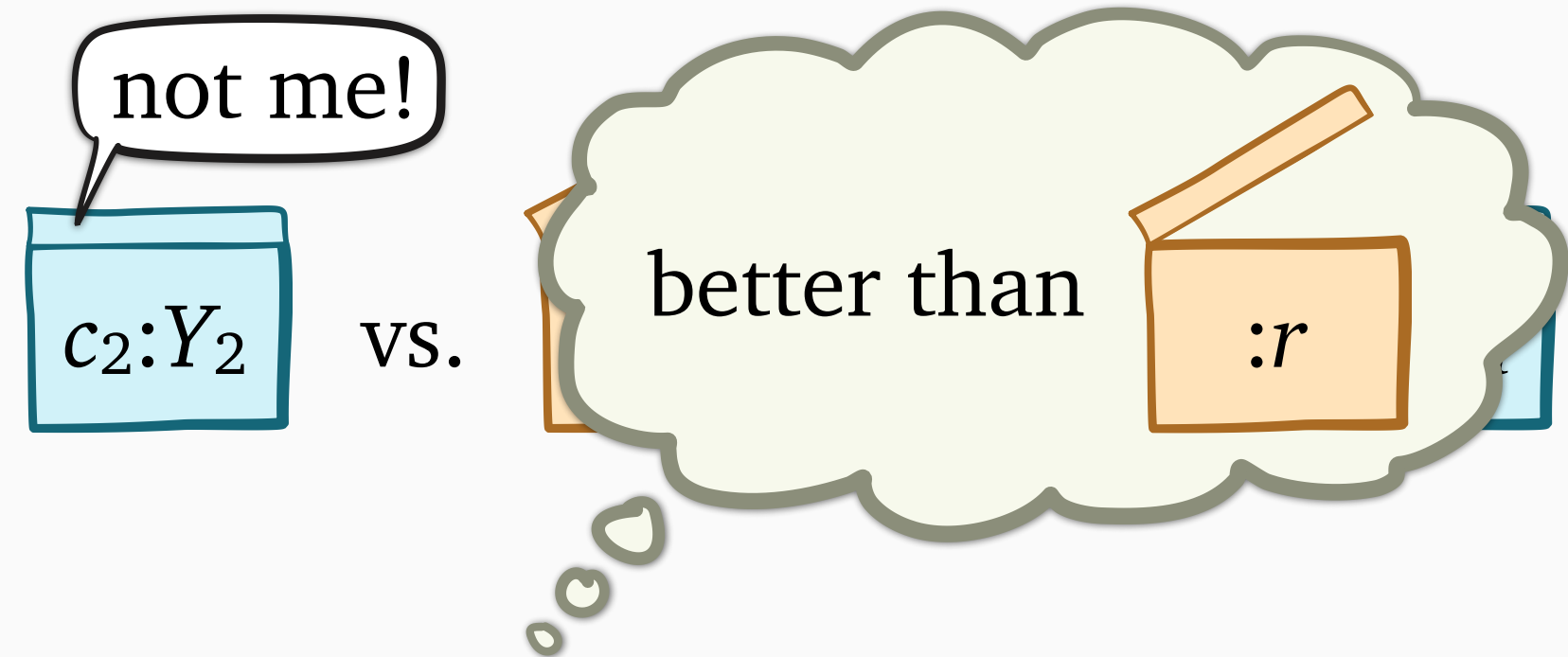
- $r \geq G_1$: select box 0
- $r = G_1$:



Box 1's perspective

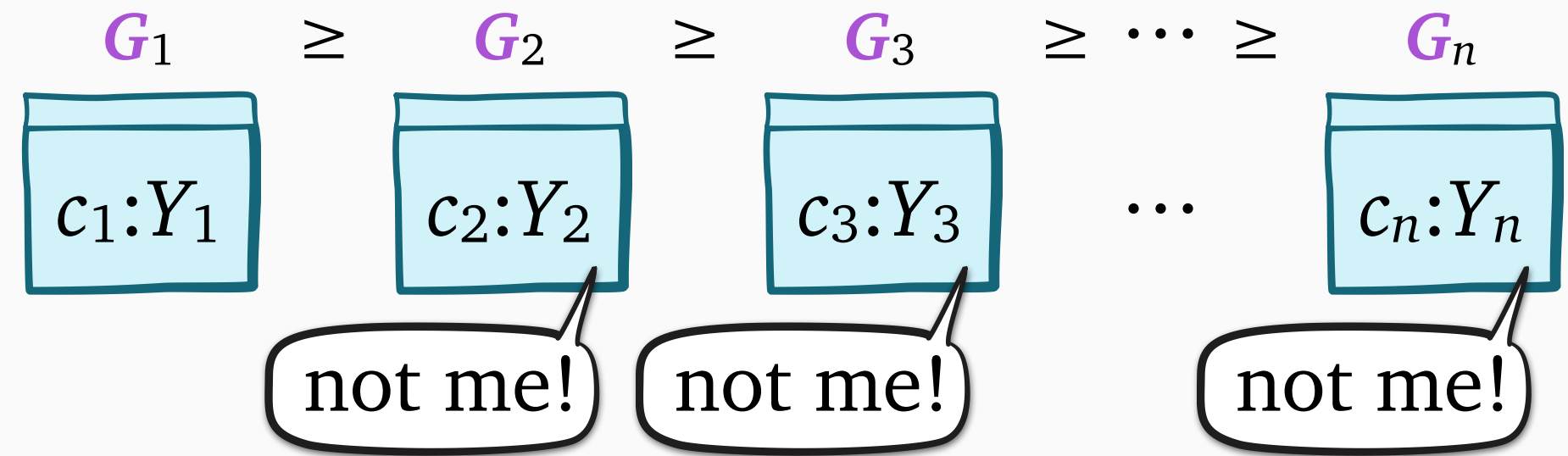
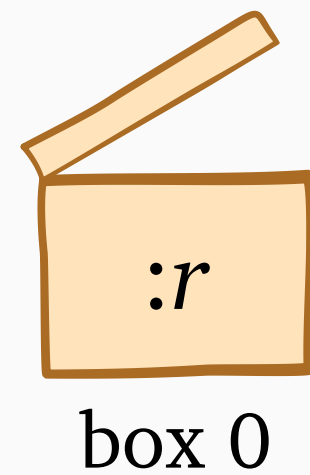


Box 2's perspective

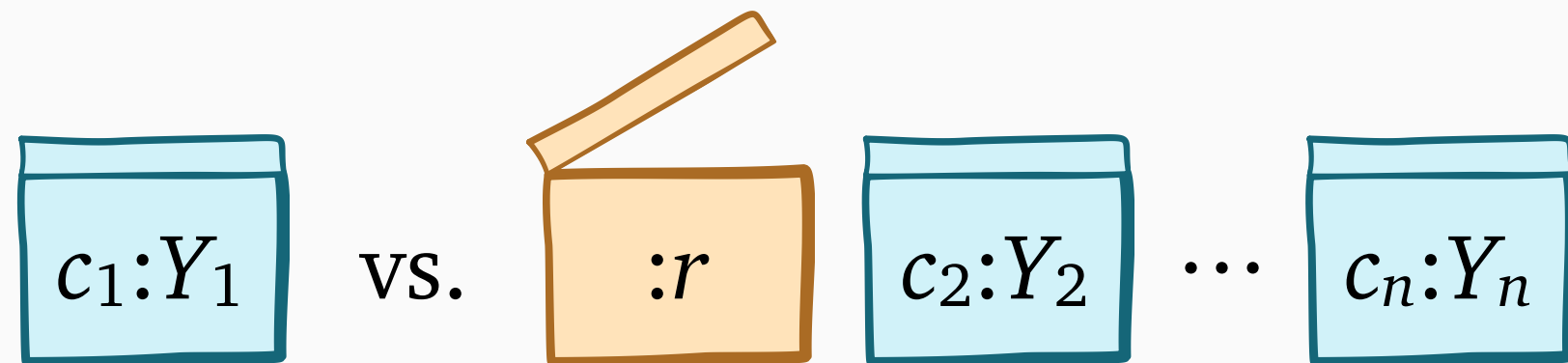


Approach: start at $r = \infty$,
then decrease to $r = 0$

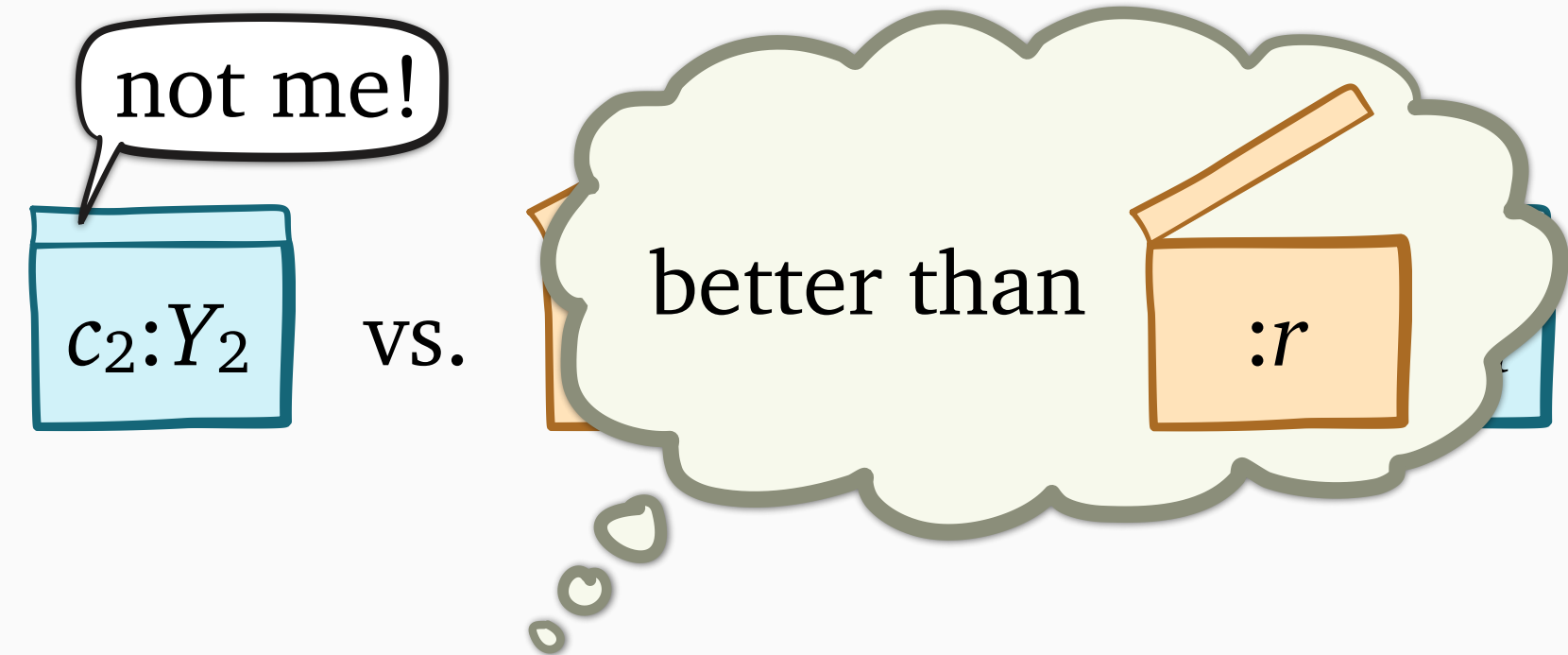
- $r \geq G_1$: select box 0
- $r = G_1$:



Box 1's perspective

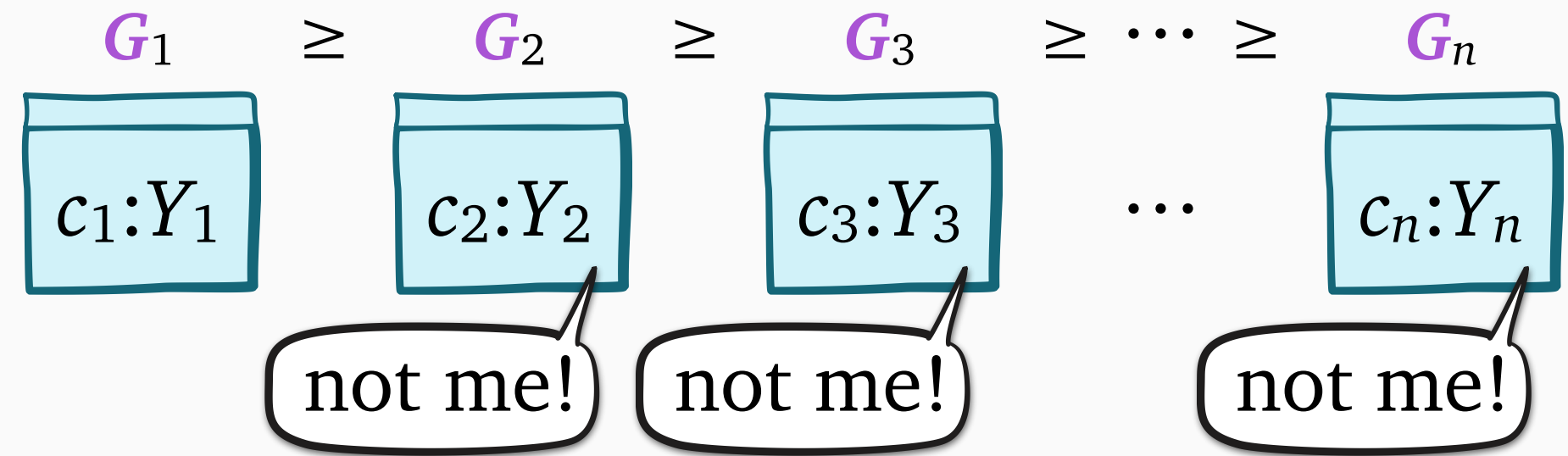
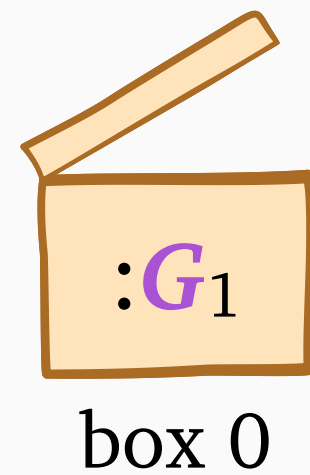


Box 2's perspective

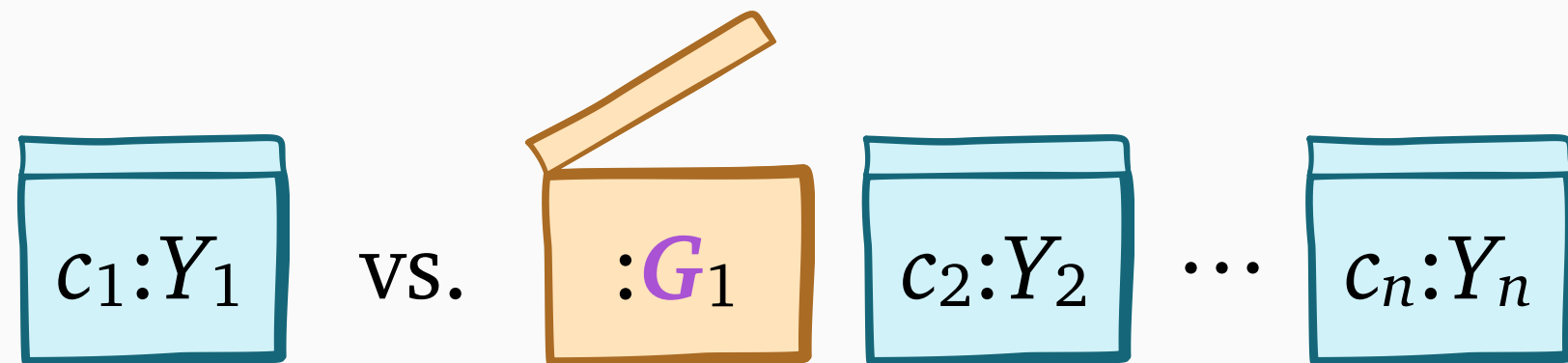


Approach: start at $r = \infty$,
then decrease to $r = 0$

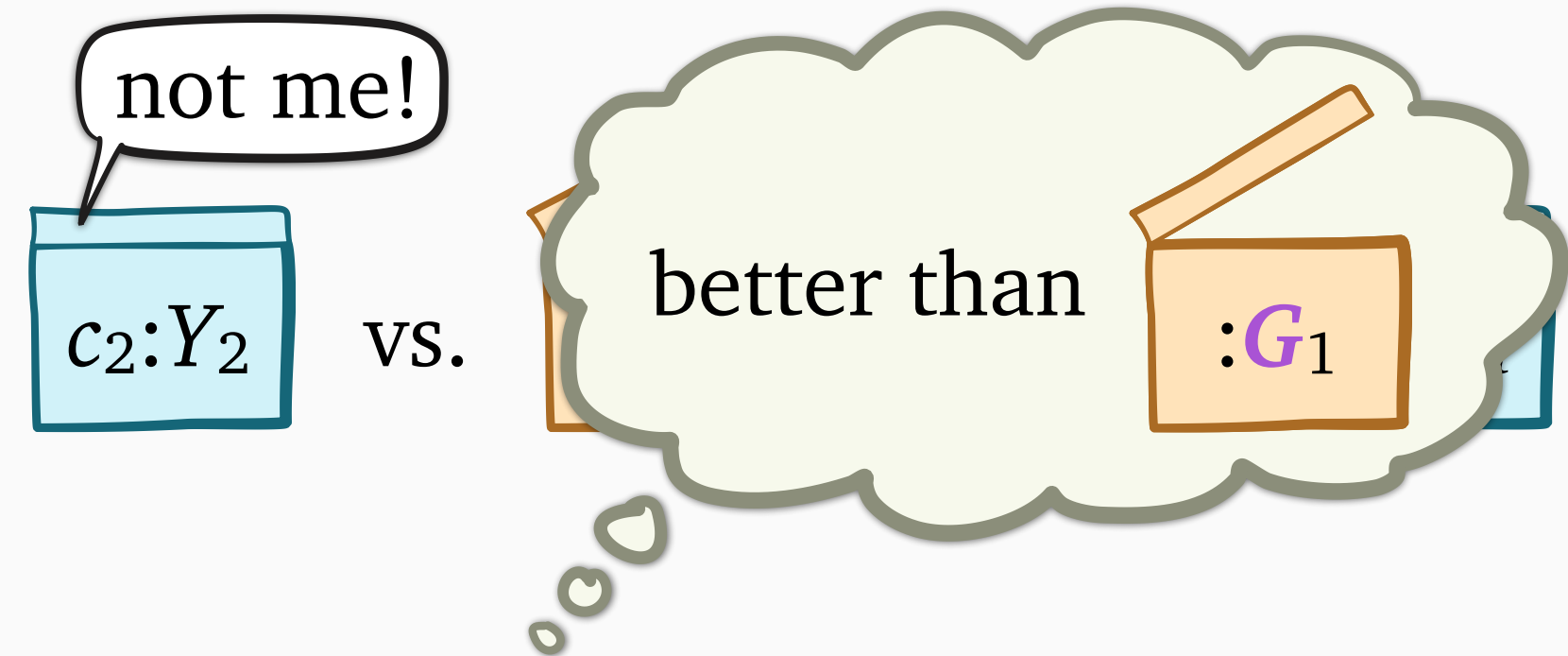
- $r \geq G_1$: select box 0
- $r = G_1$:



Box 1's perspective

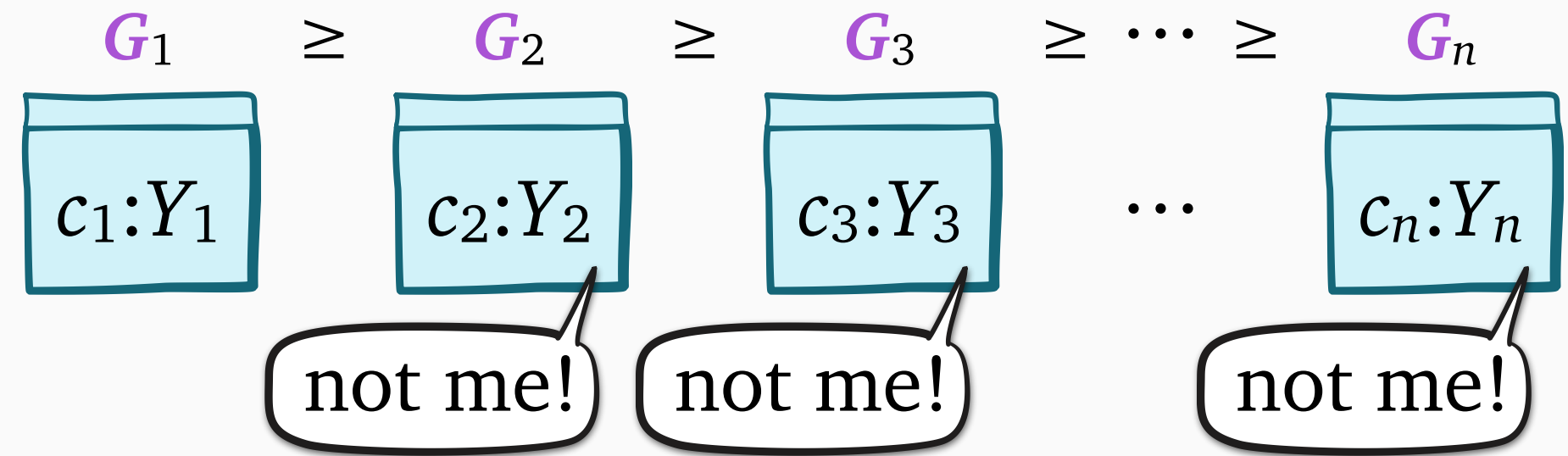
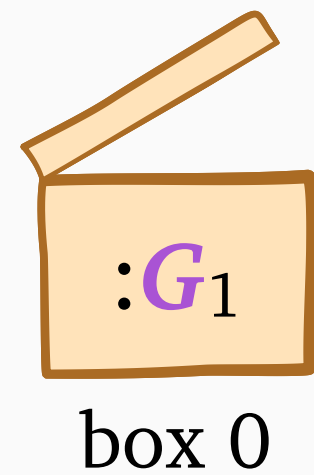


Box 2's perspective

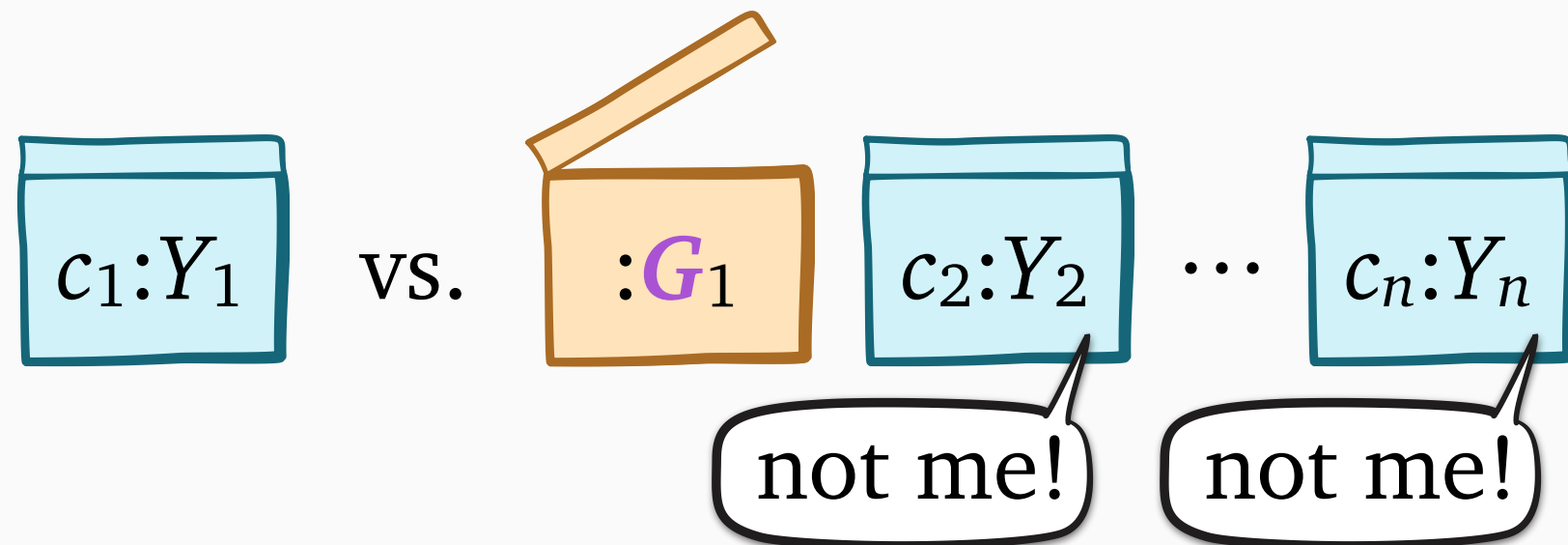


Approach: start at $r = \infty$,
then decrease to $r = 0$

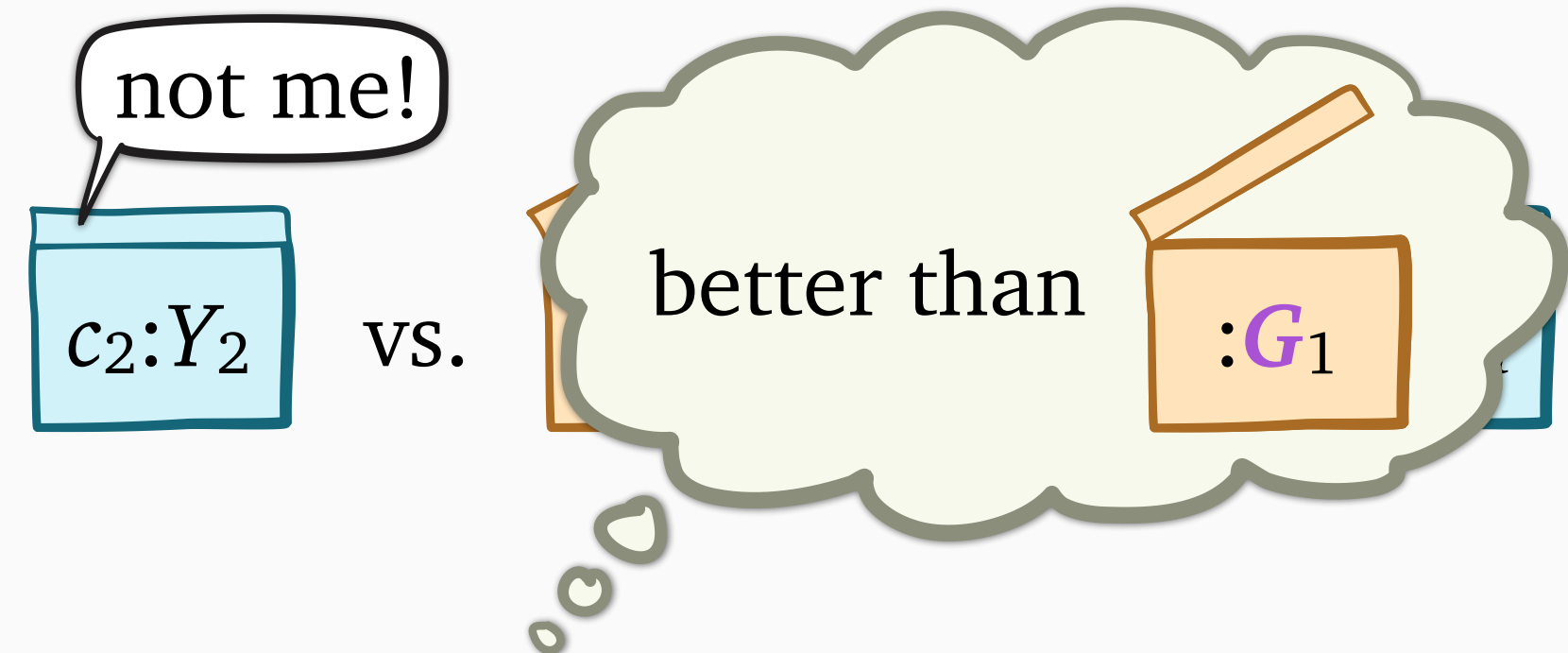
- $r \geq G_1$: select box 0
- $r = G_1$:



Box 1's perspective

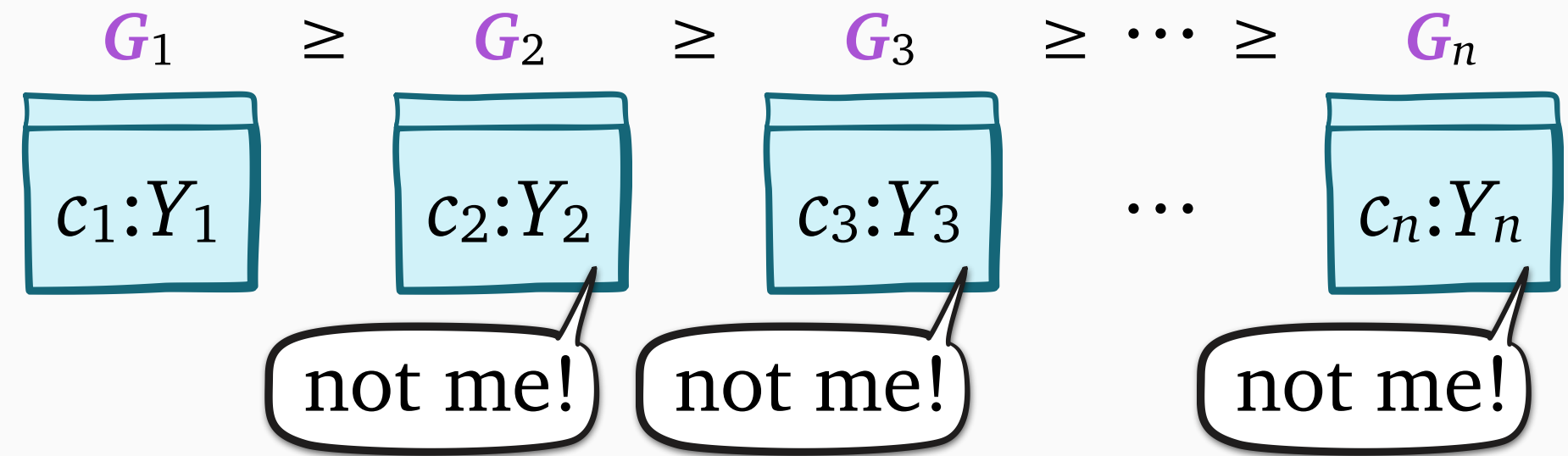
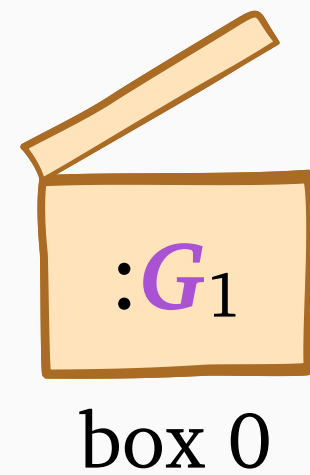


Box 2's perspective

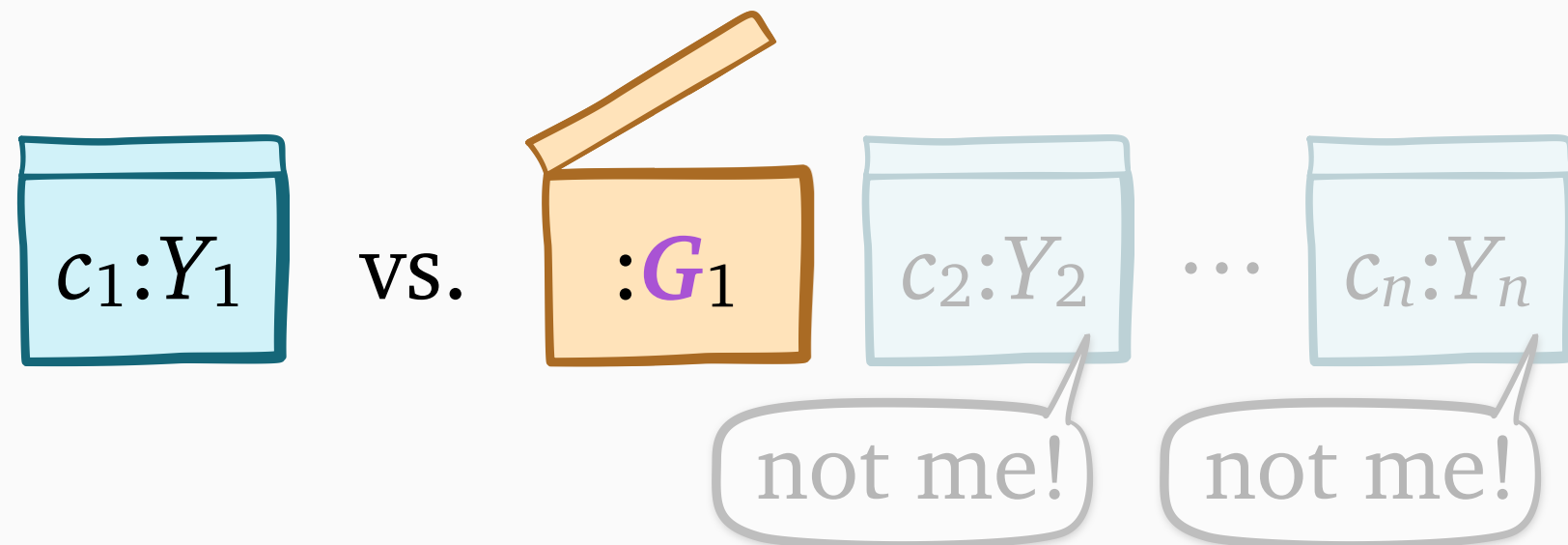


Approach: start at $r = \infty$,
then decrease to $r = 0$

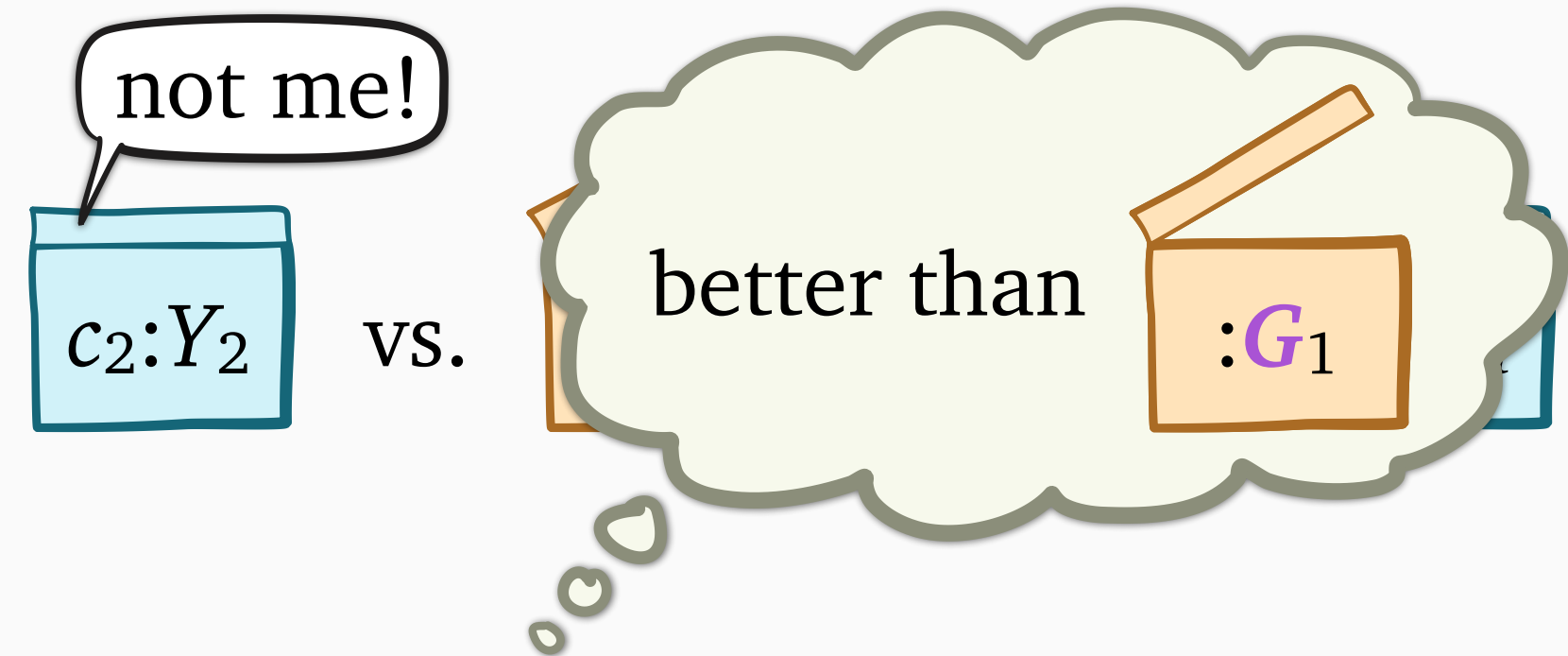
- $r \geq G_1$: select box 0
- $r = G_1$:



Box 1's perspective

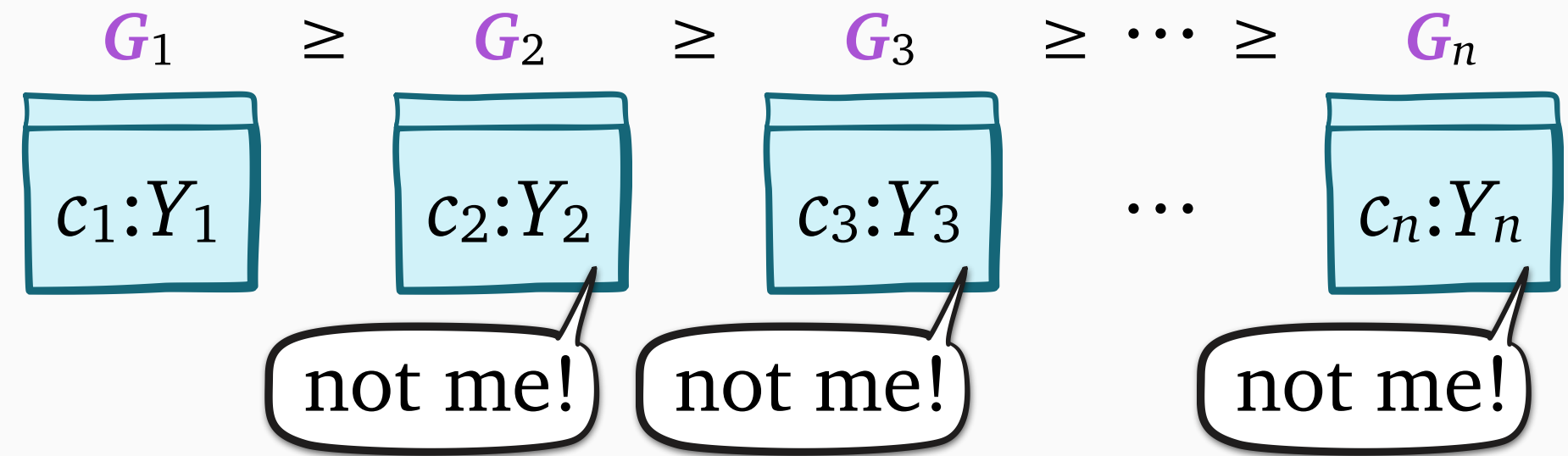
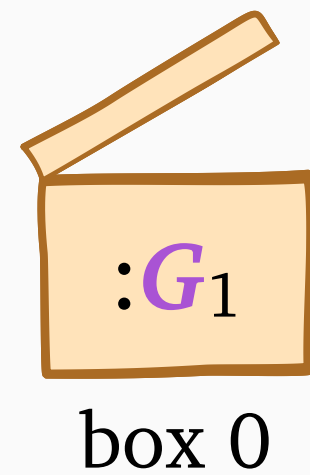


Box 2's perspective

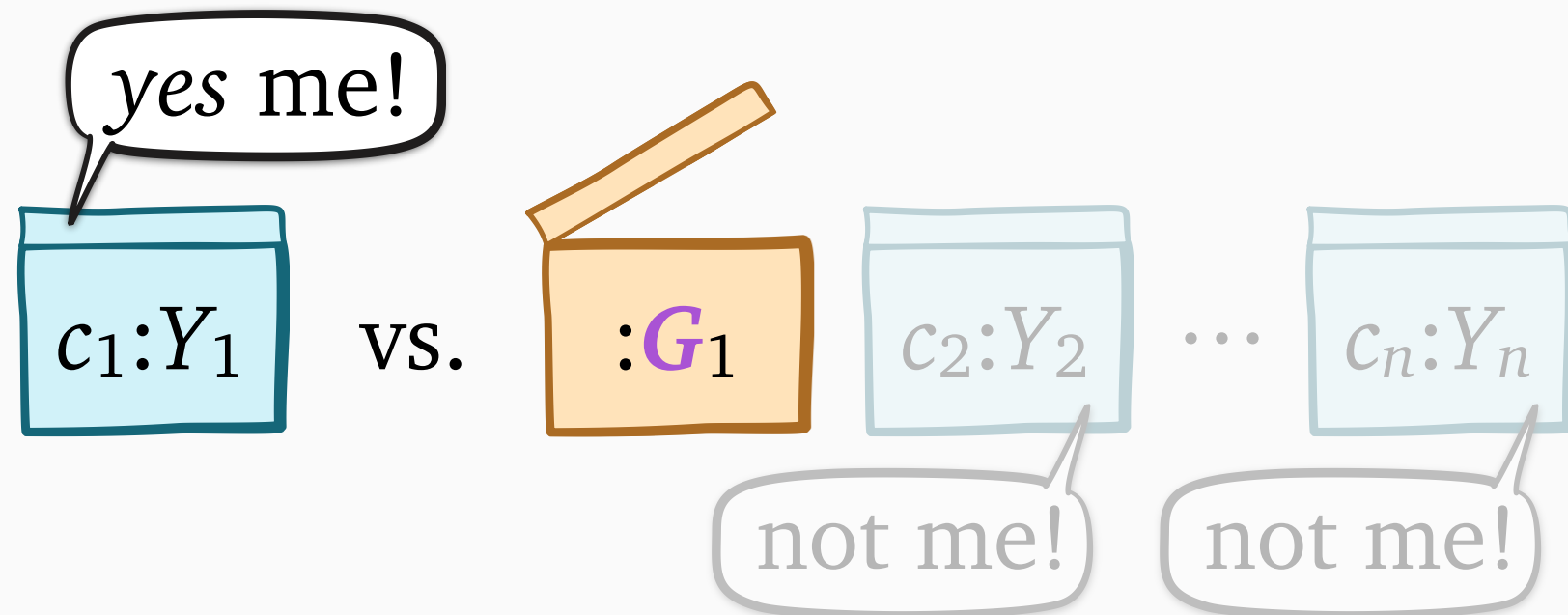


Approach: start at $r = \infty$,
then decrease to $r = 0$

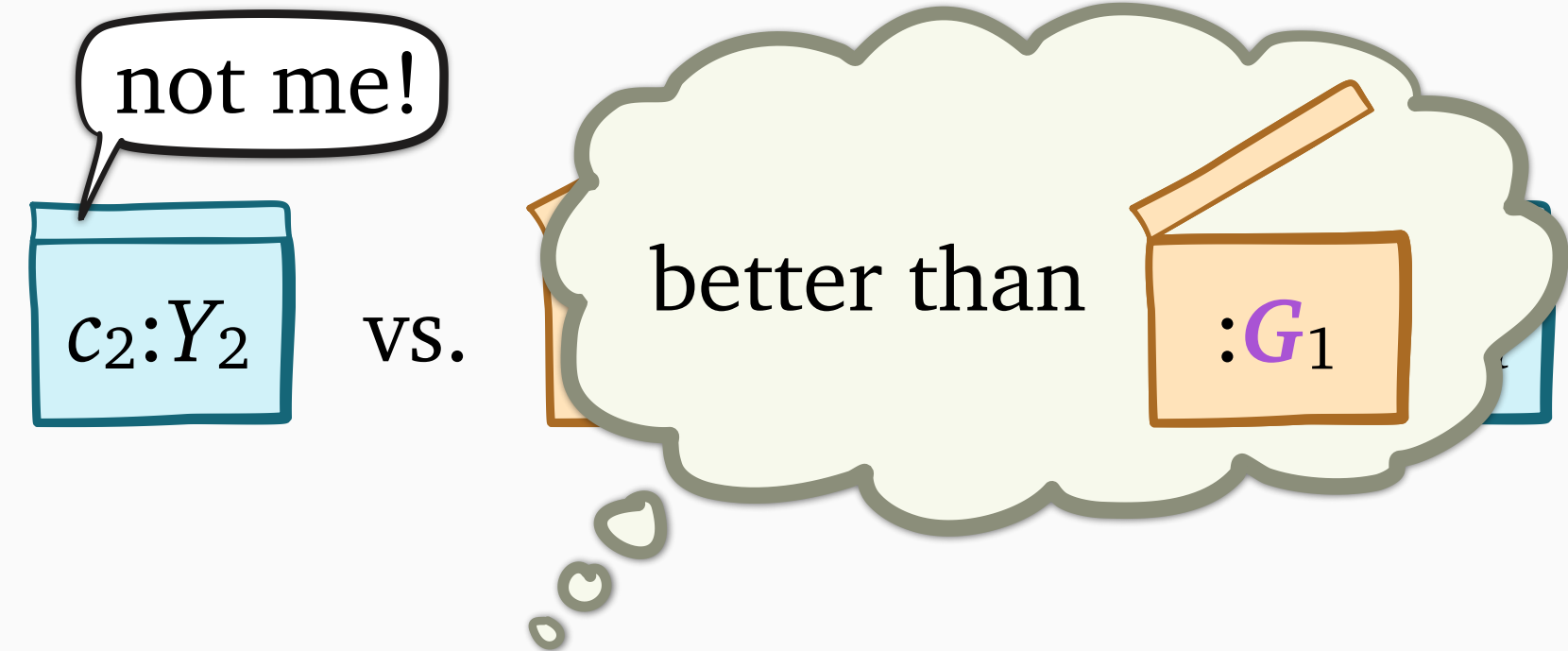
- $r \geq G_1$: select box 0
- $r = G_1$:



Box 1's perspective

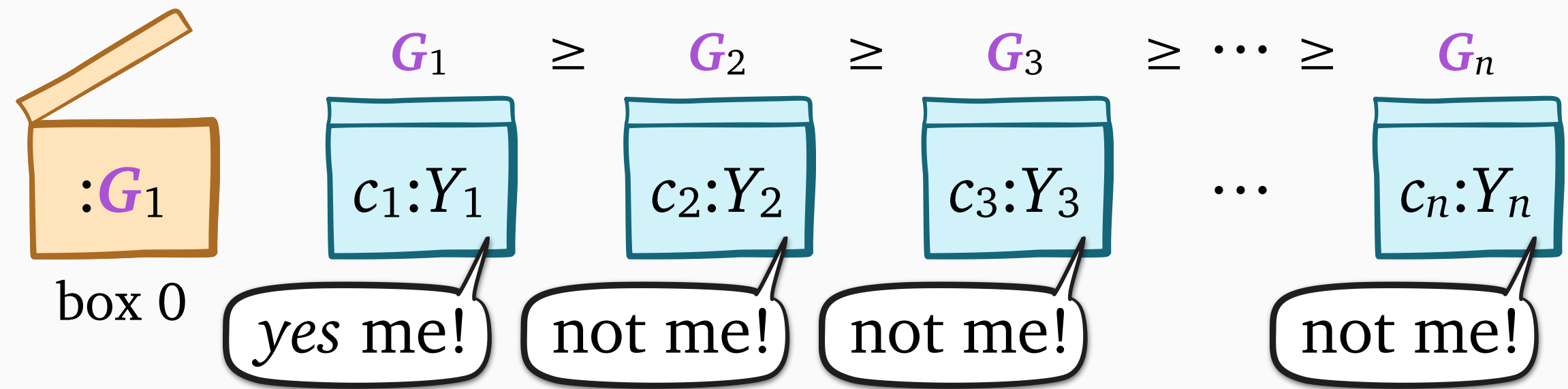


Box 2's perspective

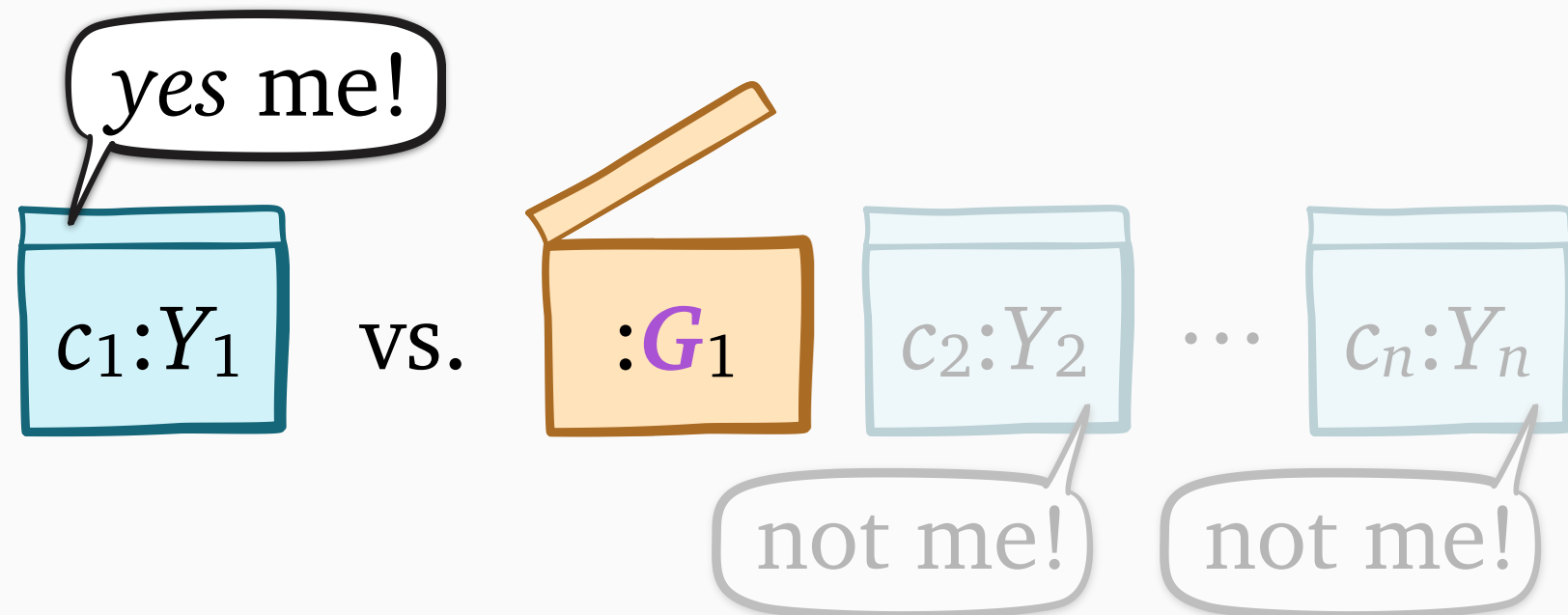


Approach: start at $r = \infty$,
then decrease to $r = 0$

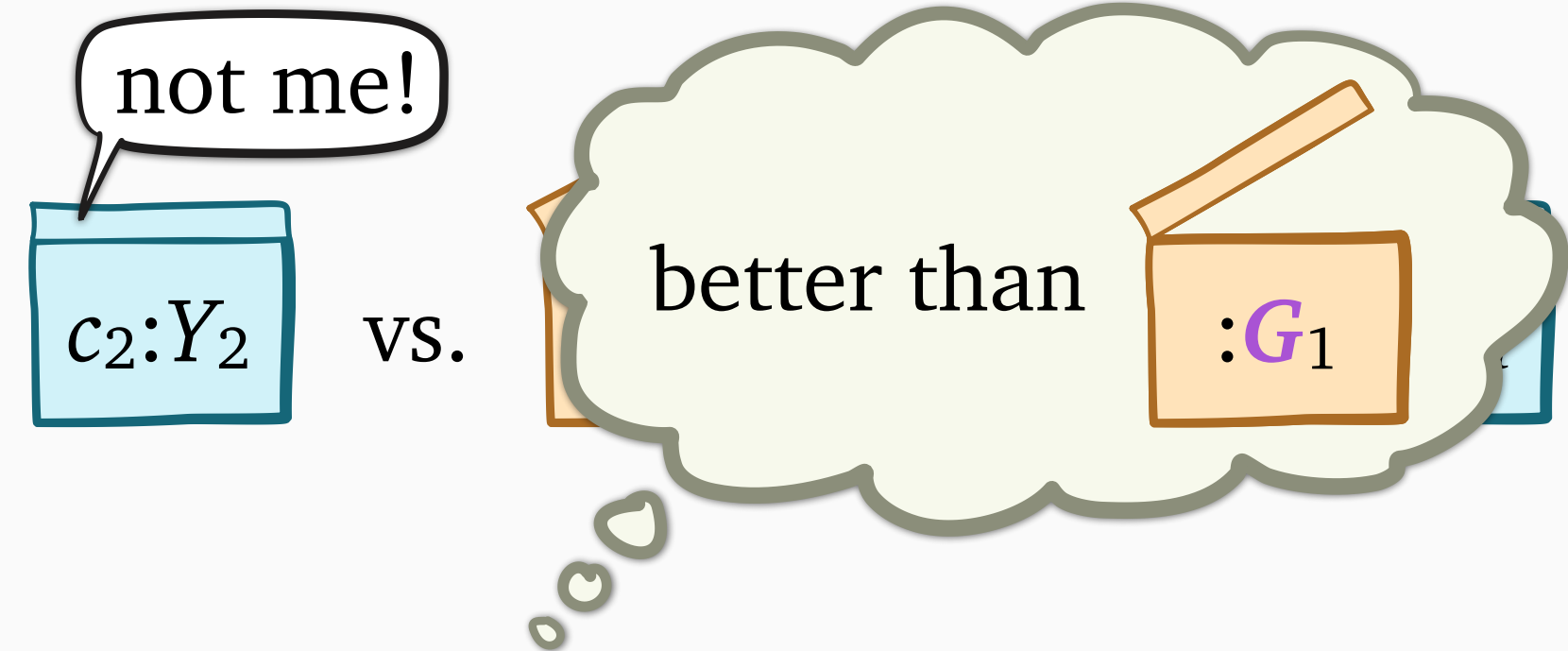
- $r \geq G_1$: select box 0
- $r = G_1$:



Box 1's perspective

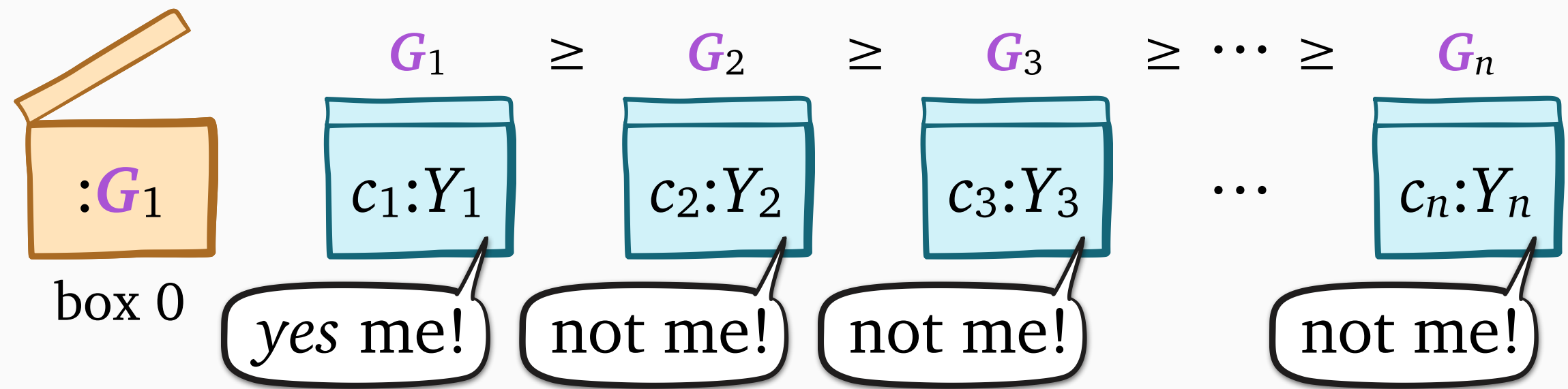


Box 2's perspective

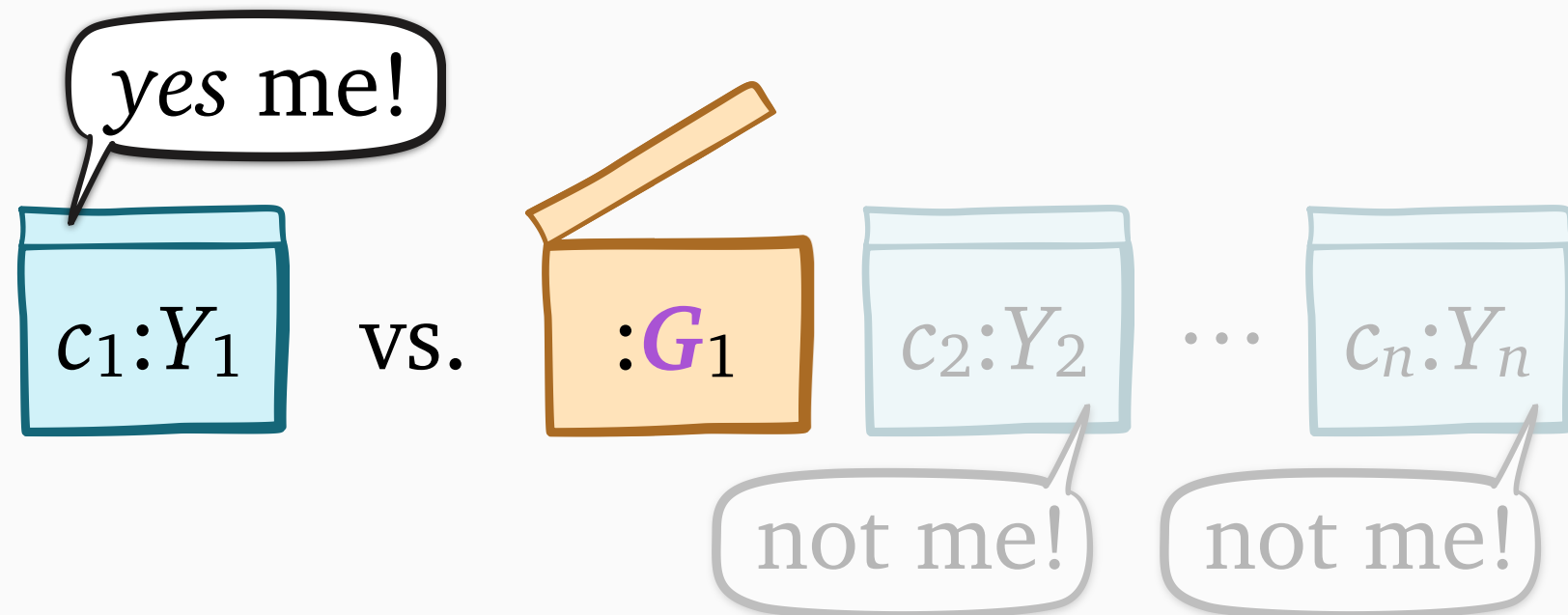


Approach: start at $r = \infty$,
then decrease to $r = 0$

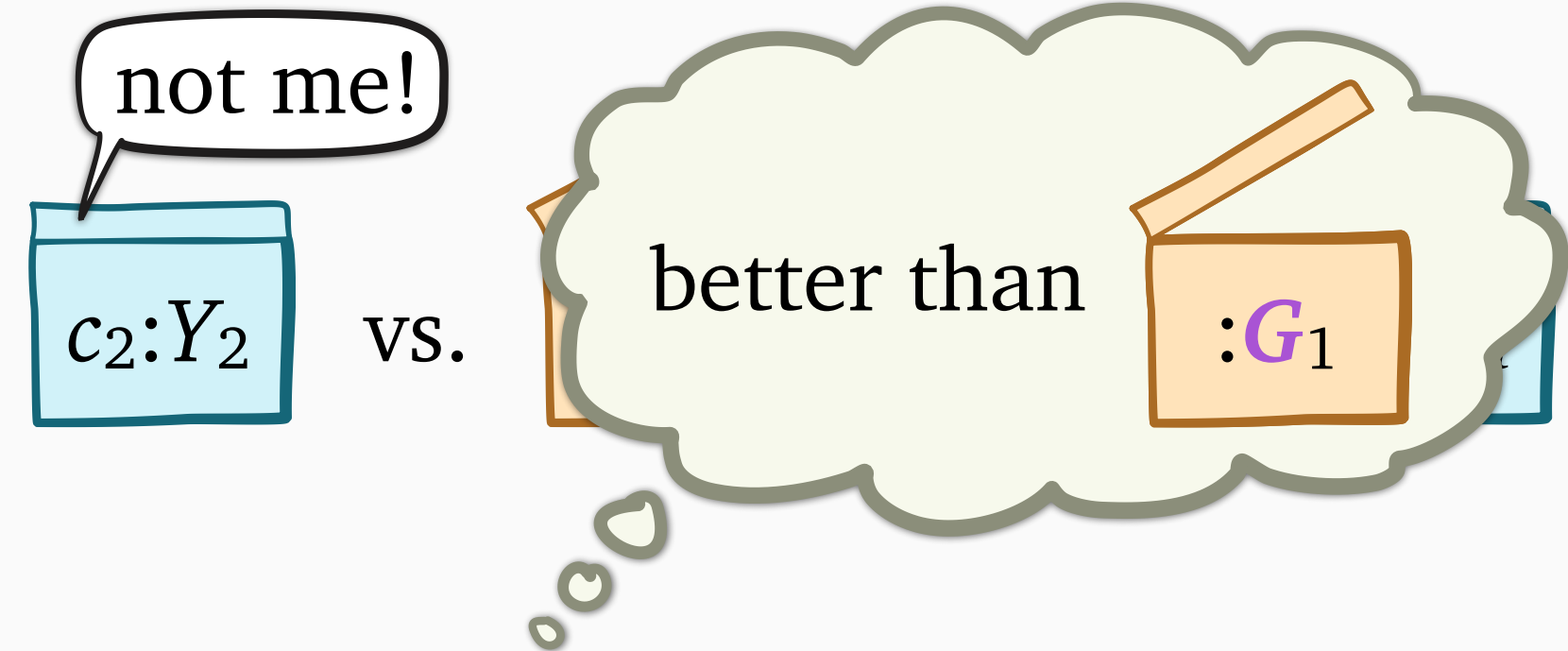
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1



Box 1's perspective

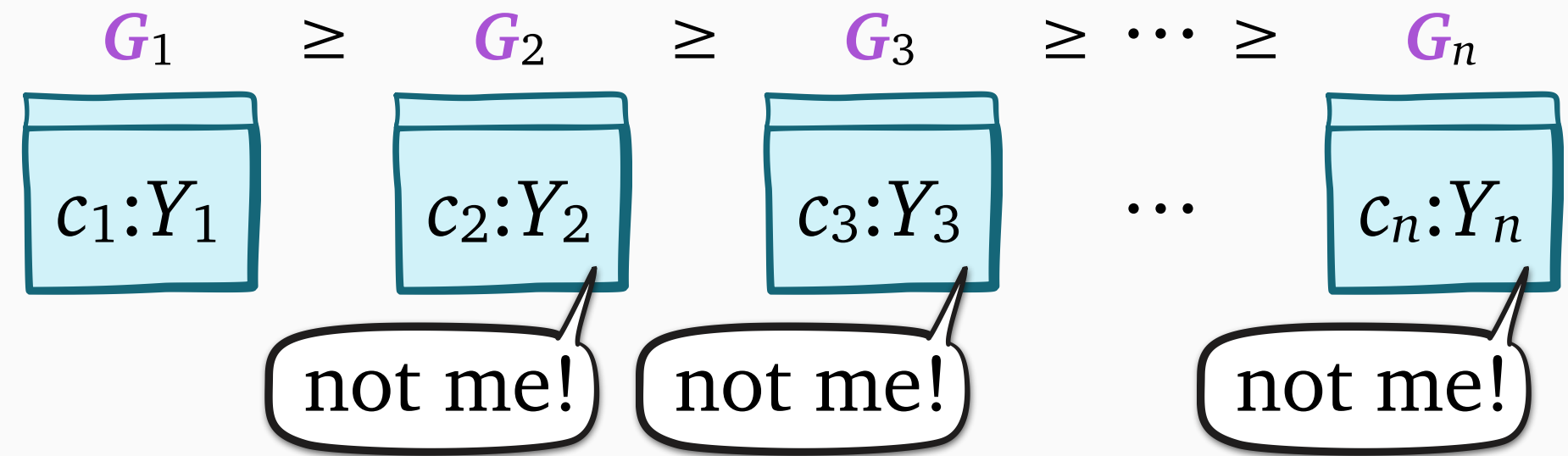
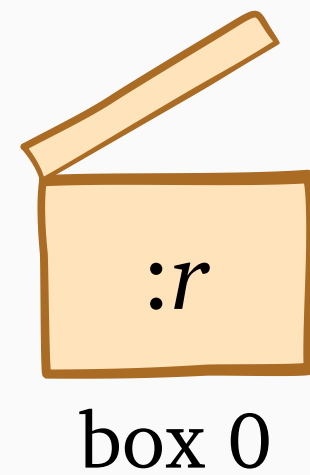


Box 2's perspective

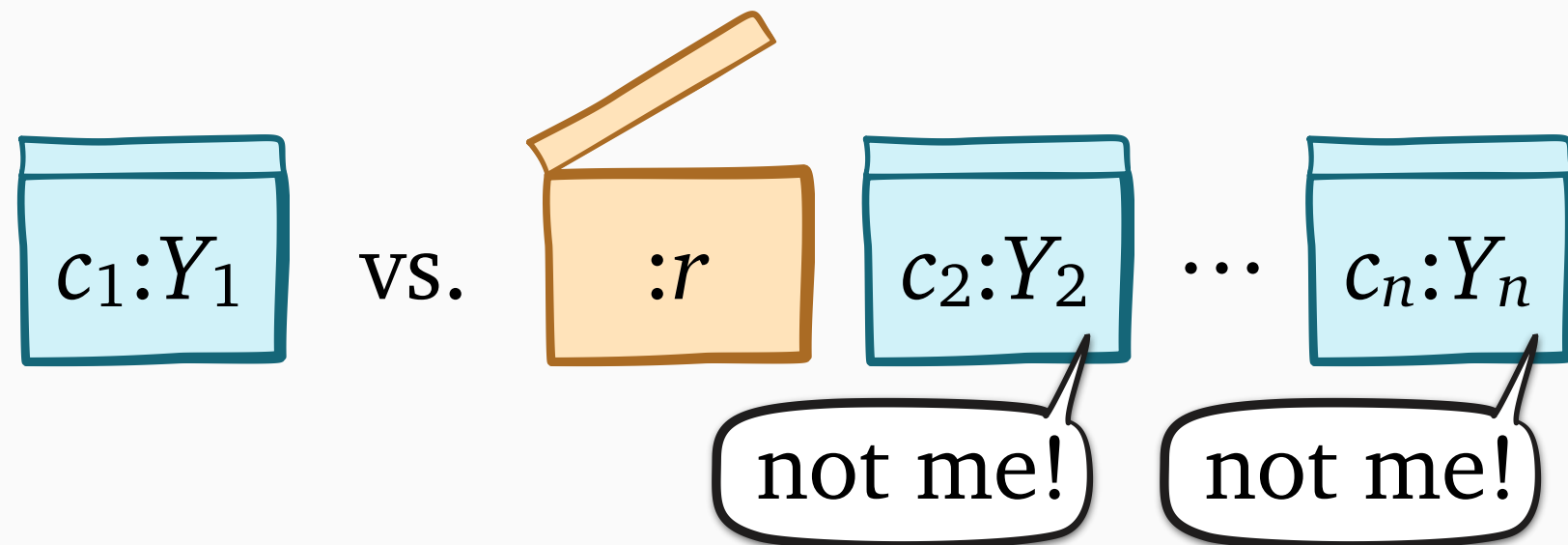


Approach: start at $r = \infty$,
then decrease to $r = 0$

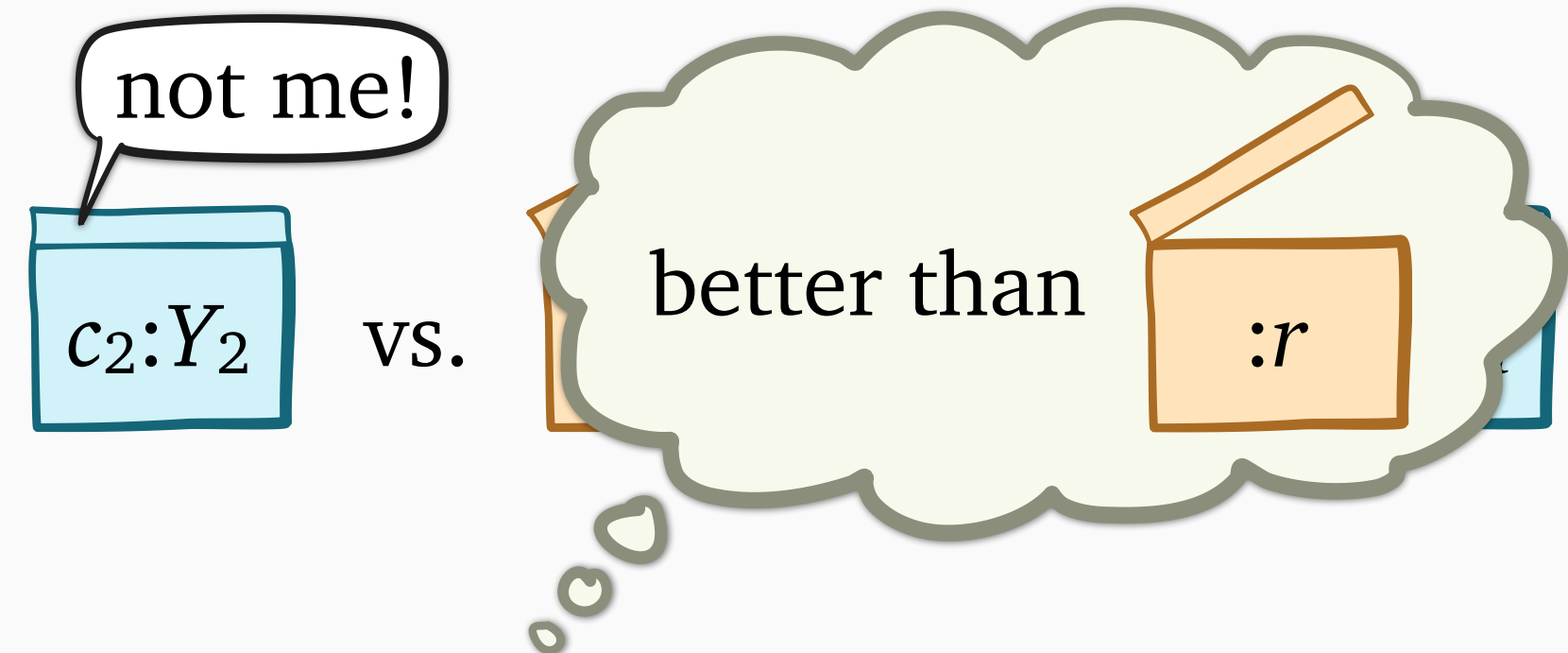
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$:



Box 1's perspective

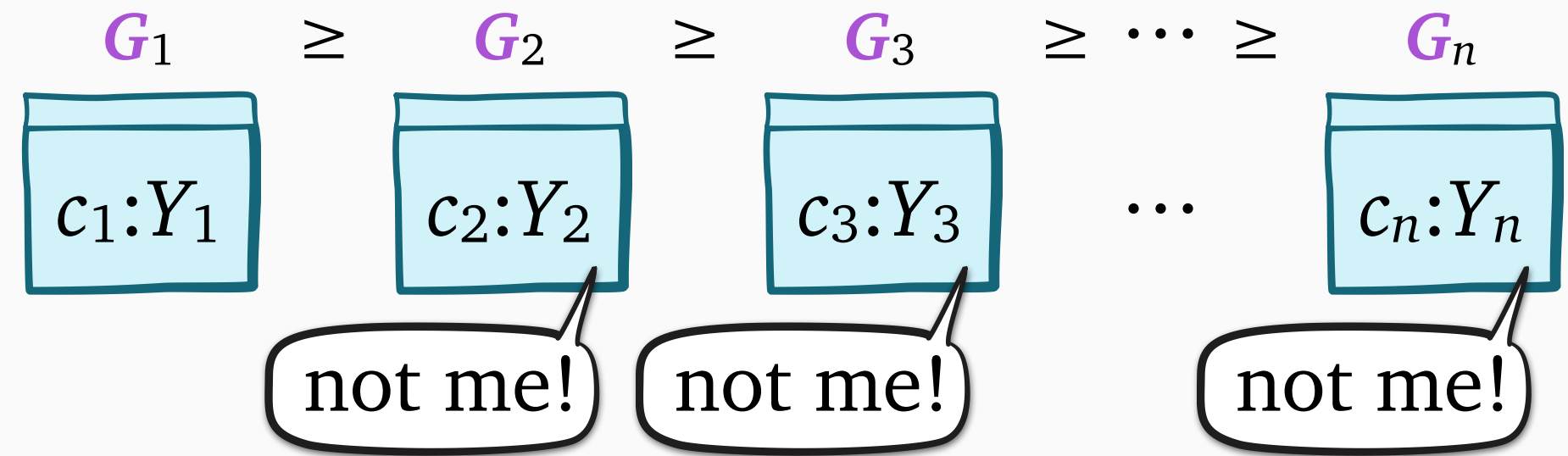
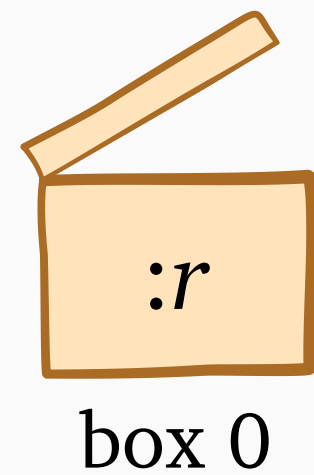


Box 2's perspective

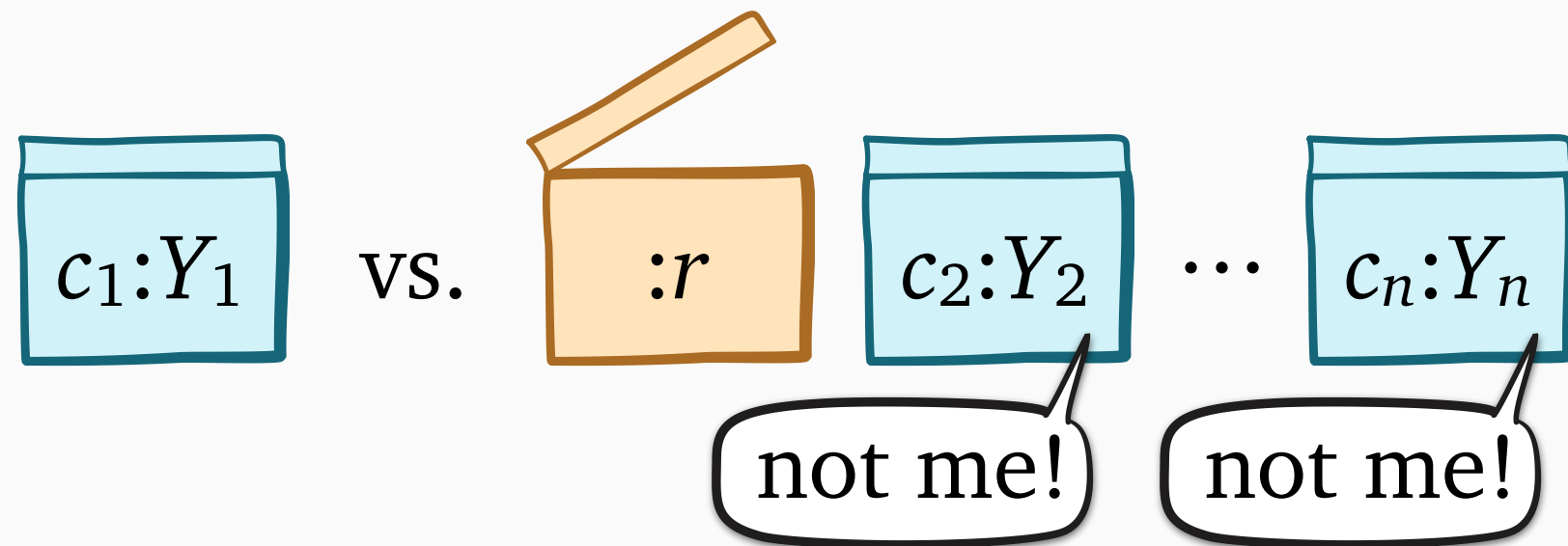


Approach: start at $r = \infty$, then decrease to $r = 0$

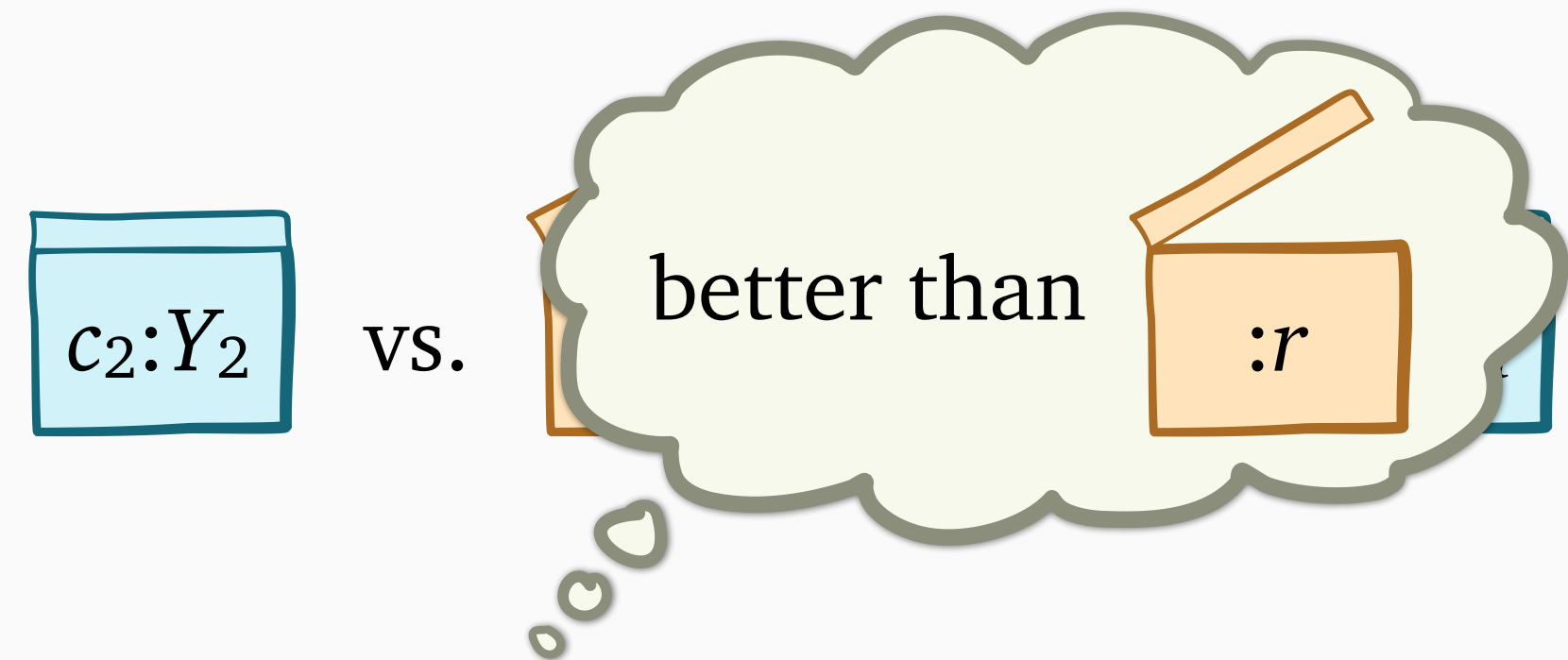
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$:



Box 1's perspective

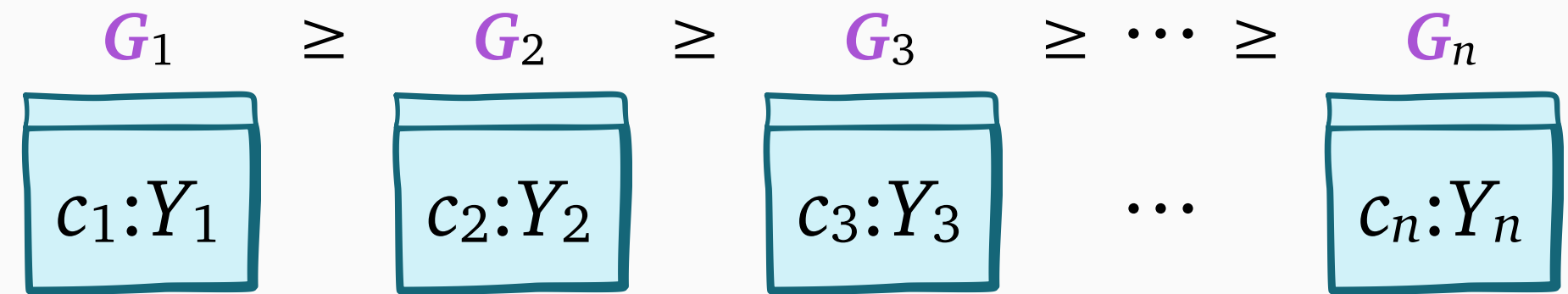
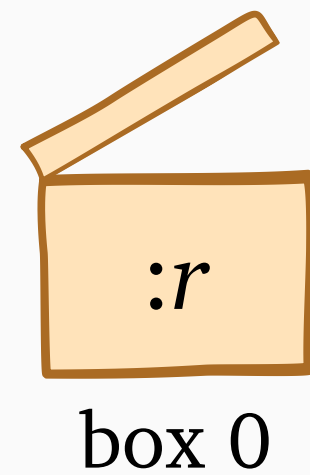


Box 2's perspective

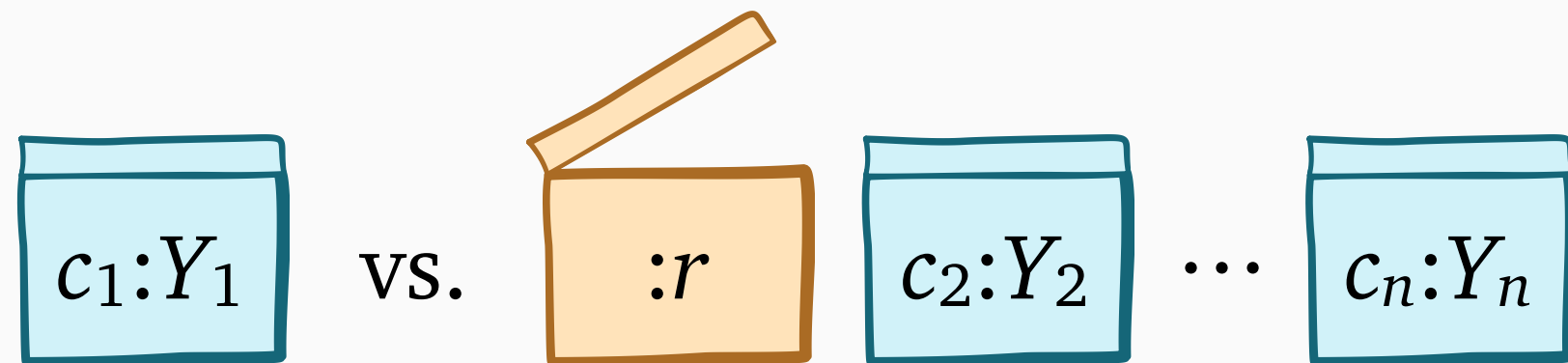


Approach: start at $r = \infty$,
then decrease to $r = 0$

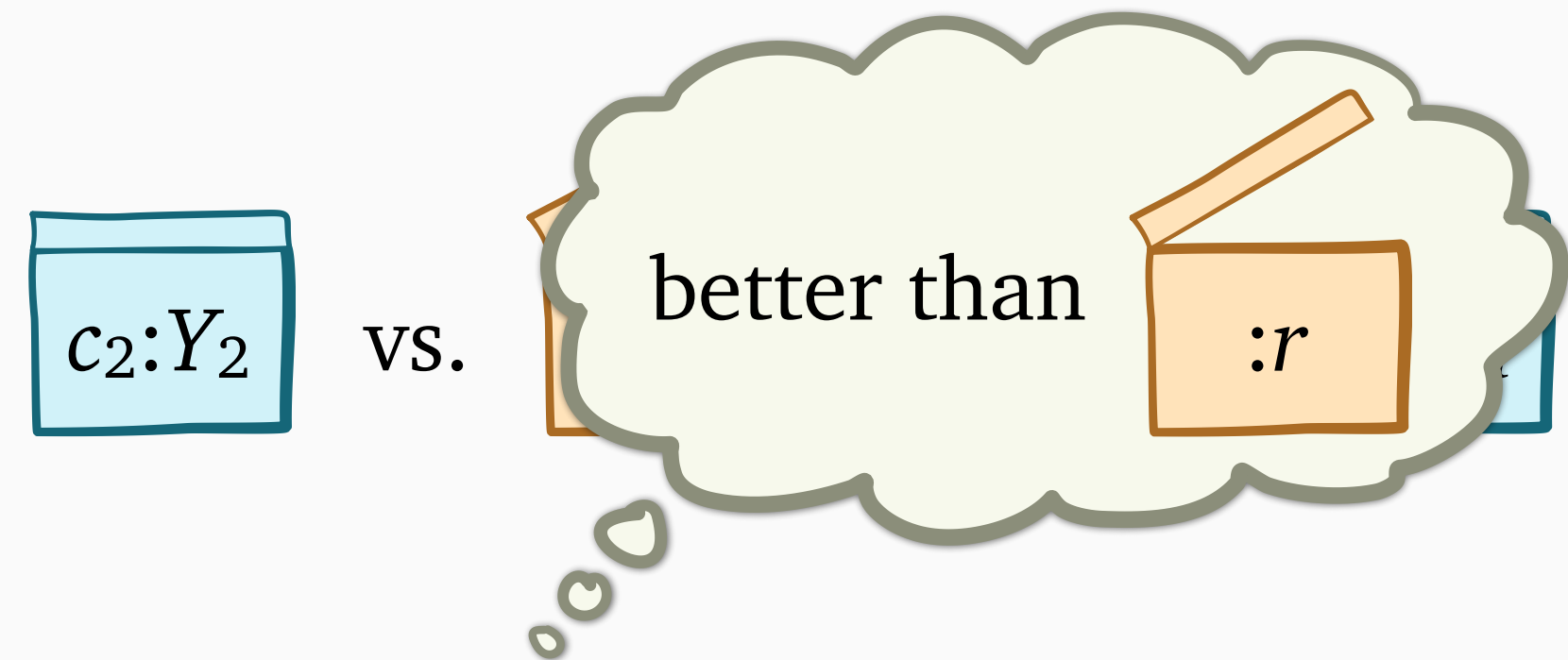
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$:



Box 1's perspective

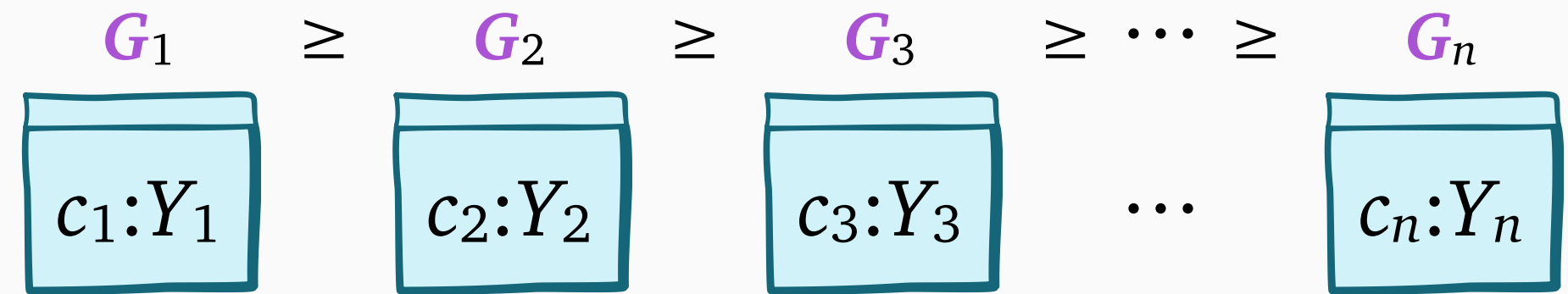
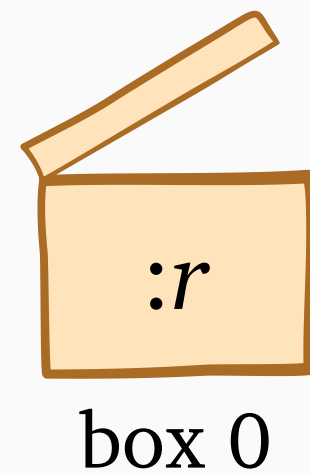


Box 2's perspective

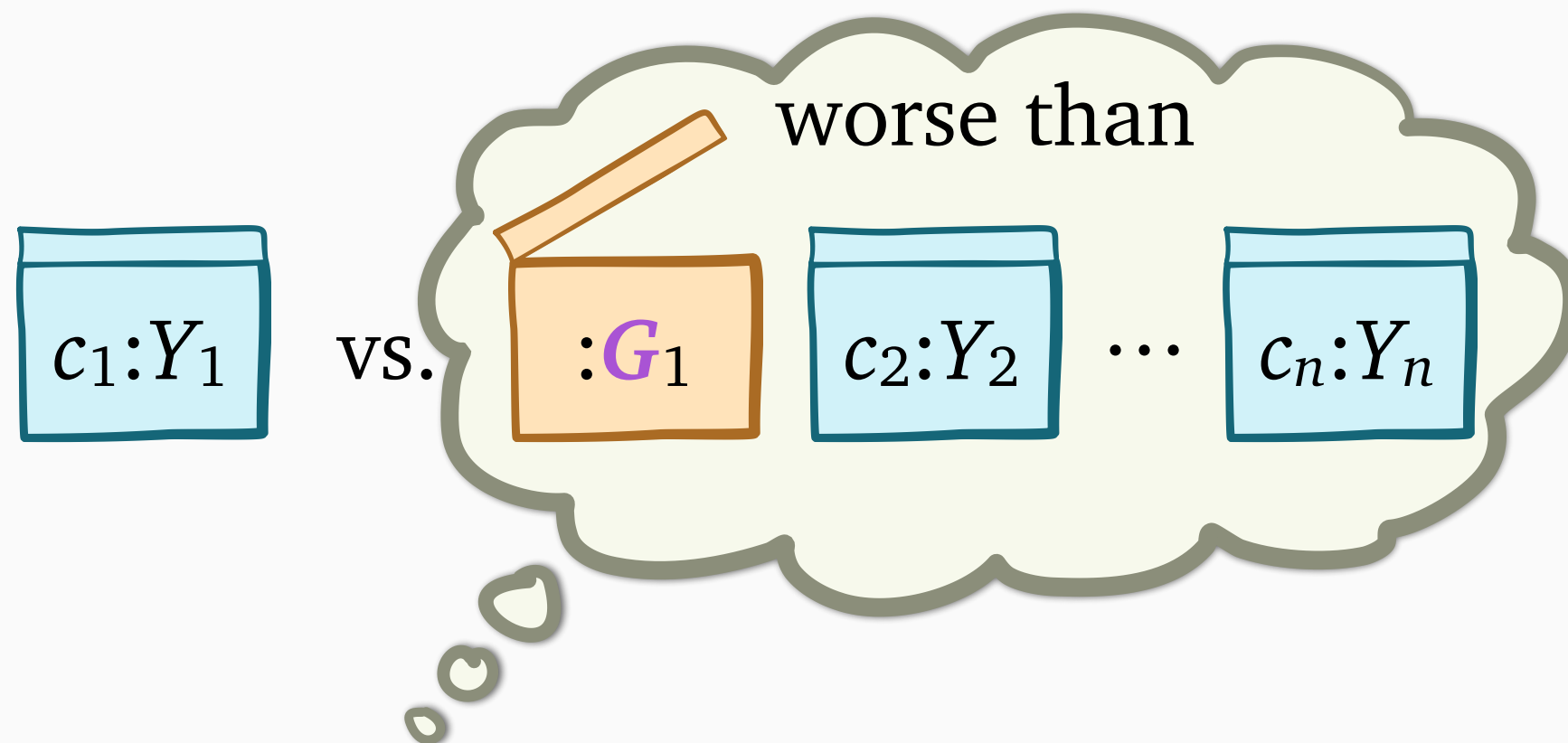


Approach: start at $r = \infty$, then decrease to $r = 0$

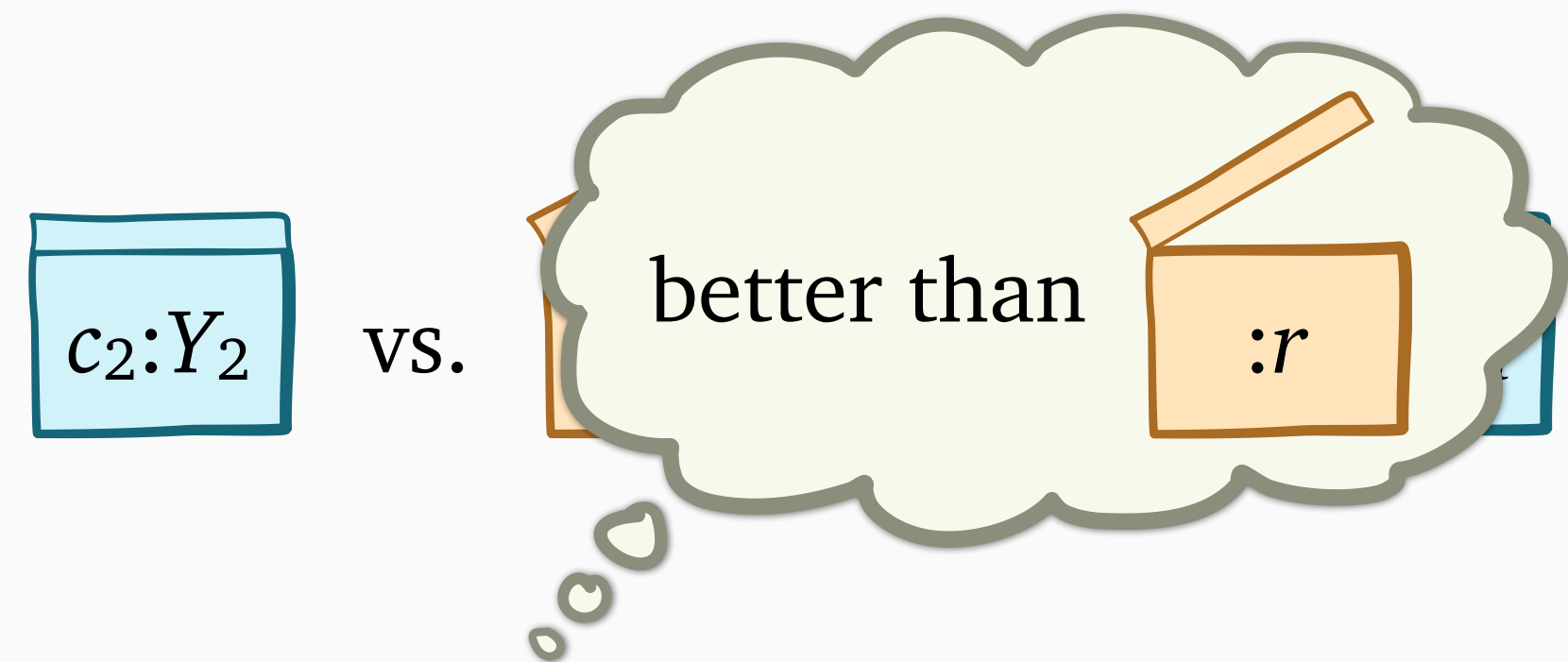
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$:



Box 1's perspective

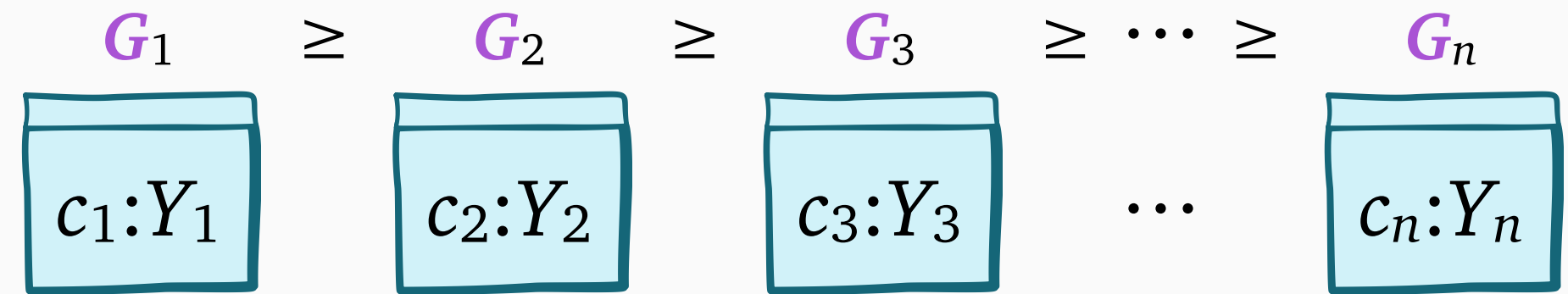
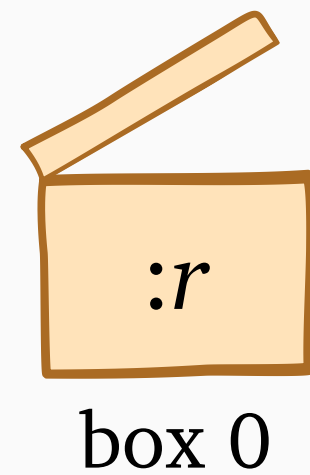


Box 2's perspective

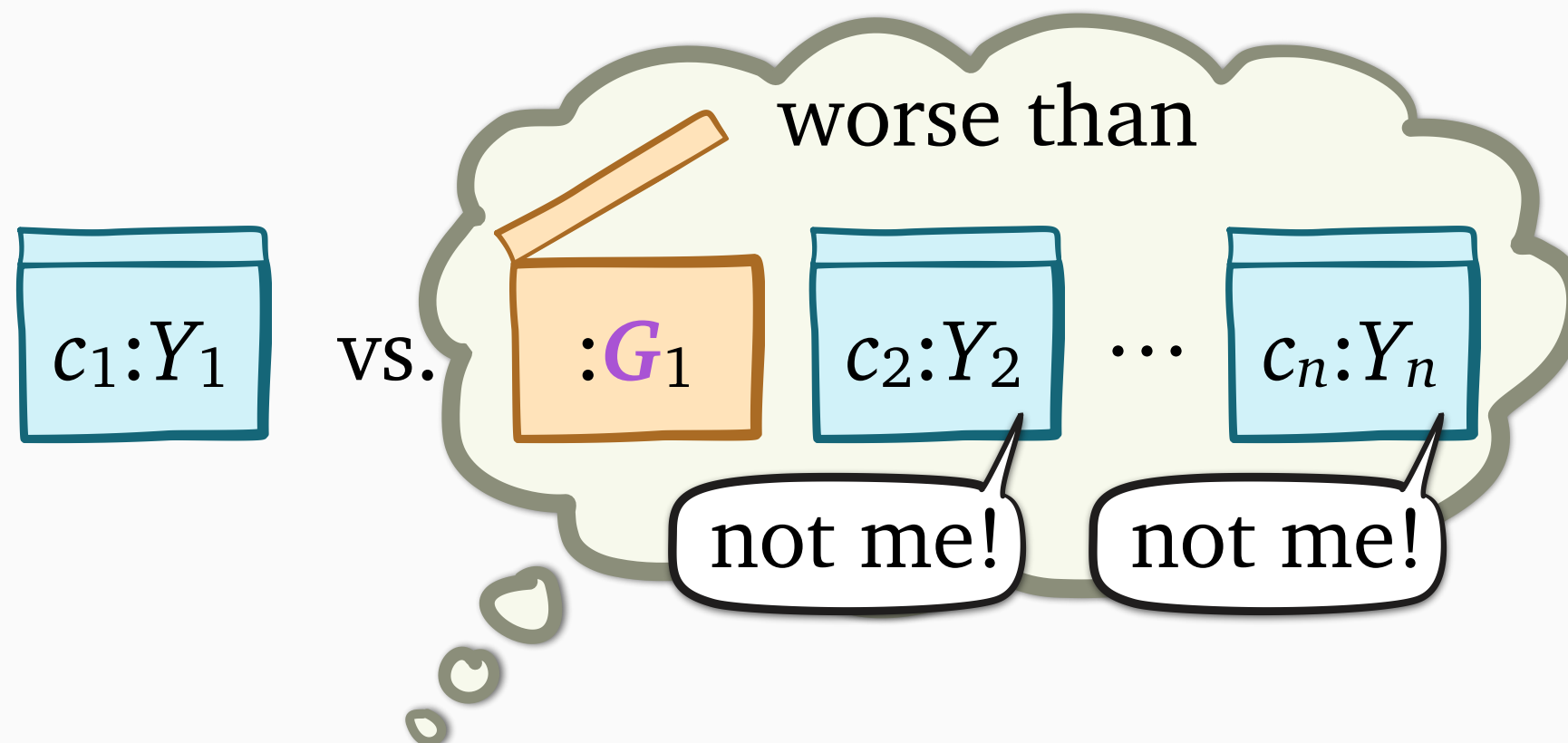


Approach: start at $r = \infty$, then decrease to $r = 0$

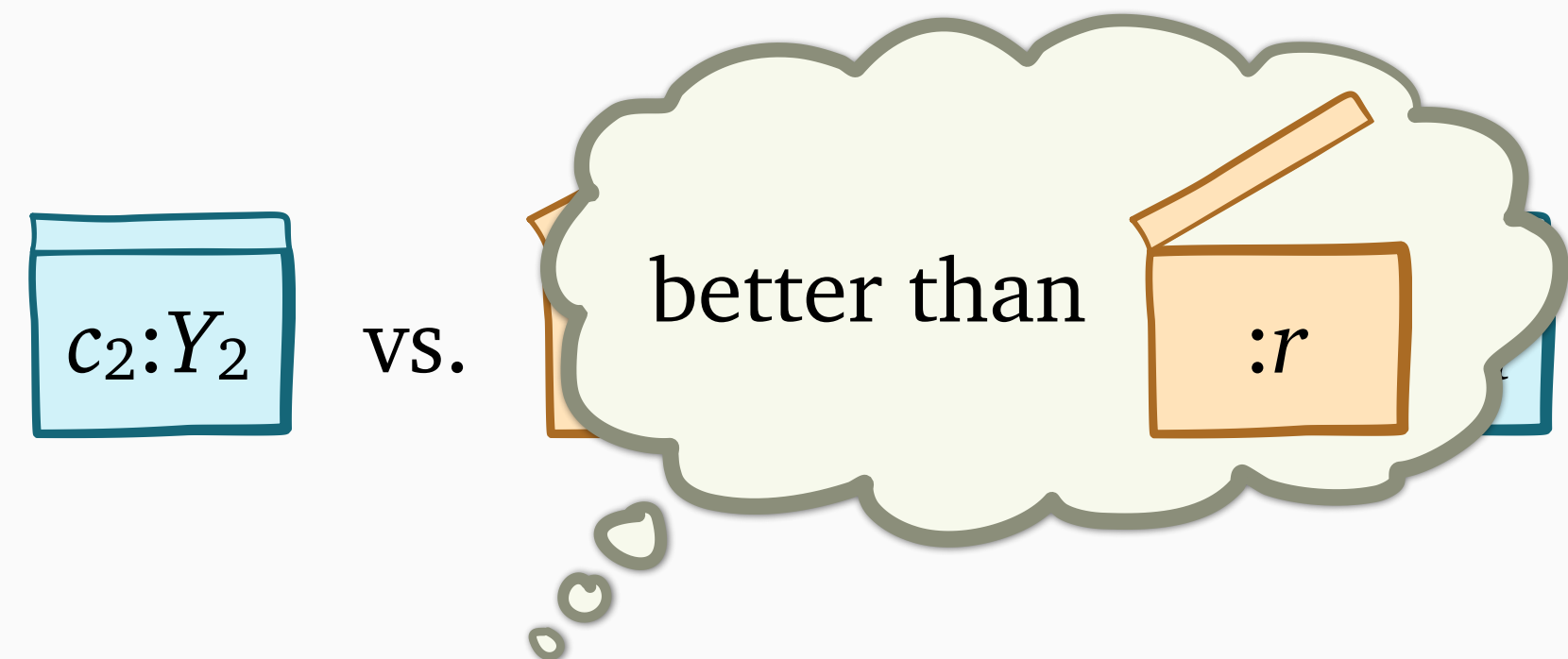
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$:



Box 1's perspective

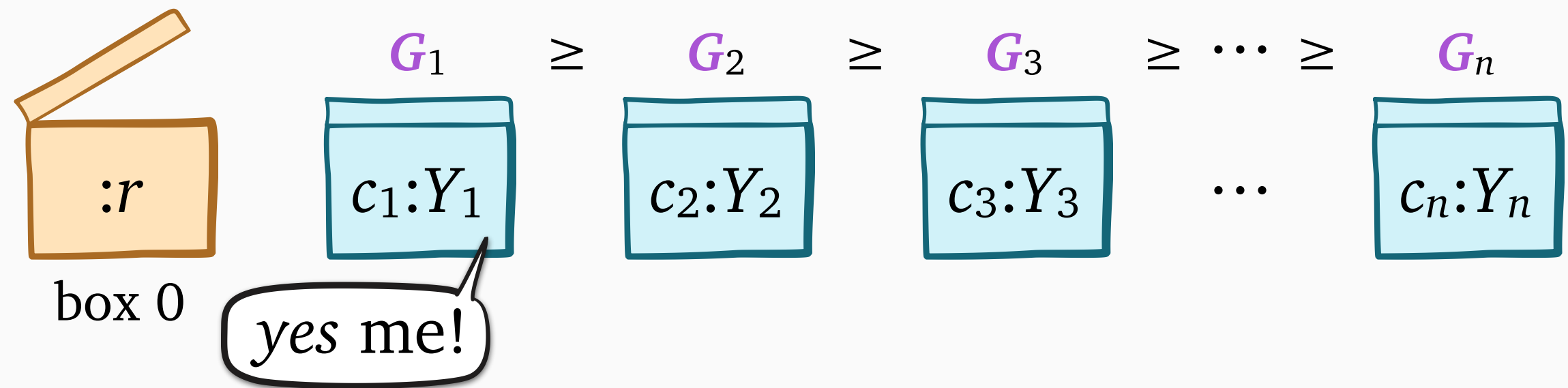


Box 2's perspective

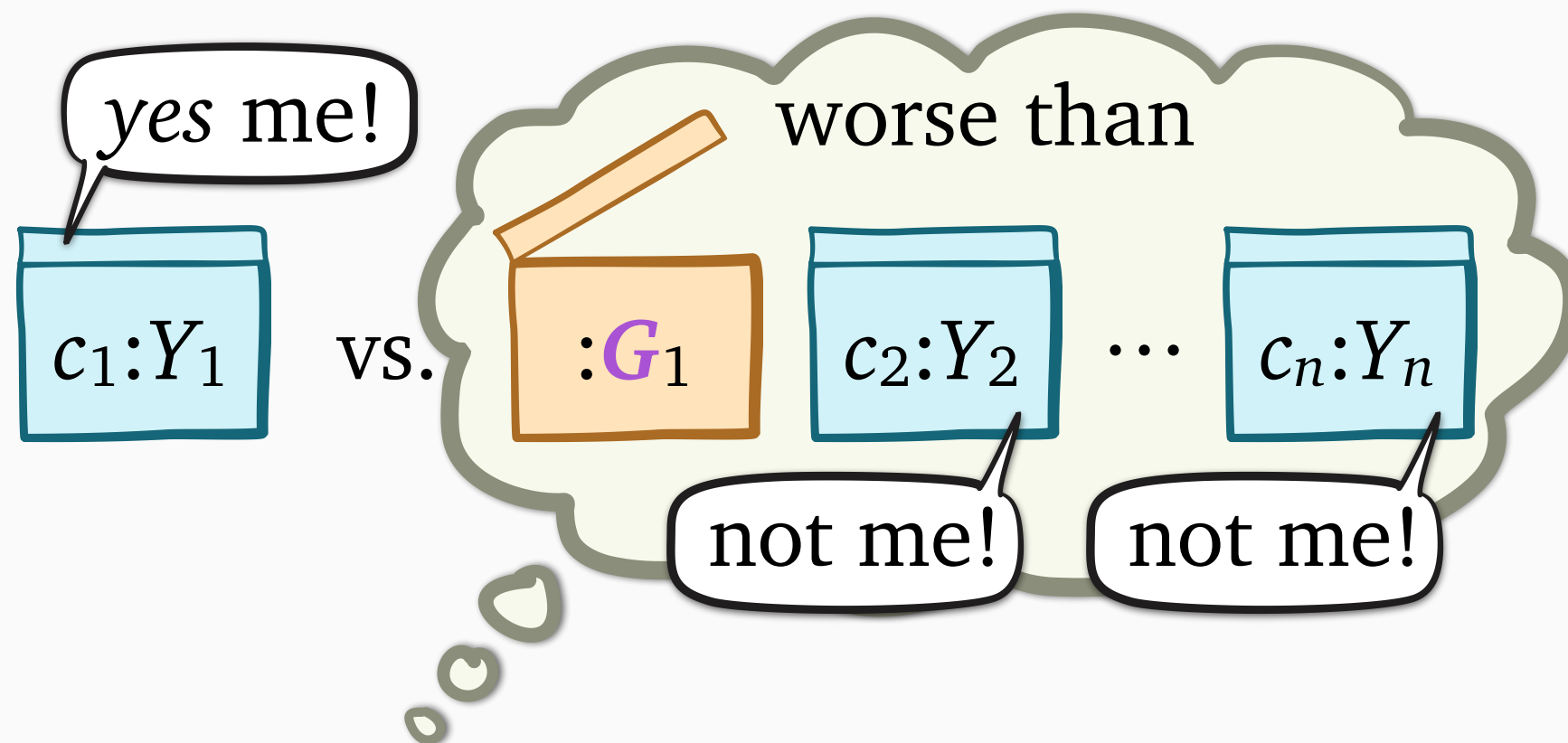


Approach: start at $r = \infty$, then decrease to $r = 0$

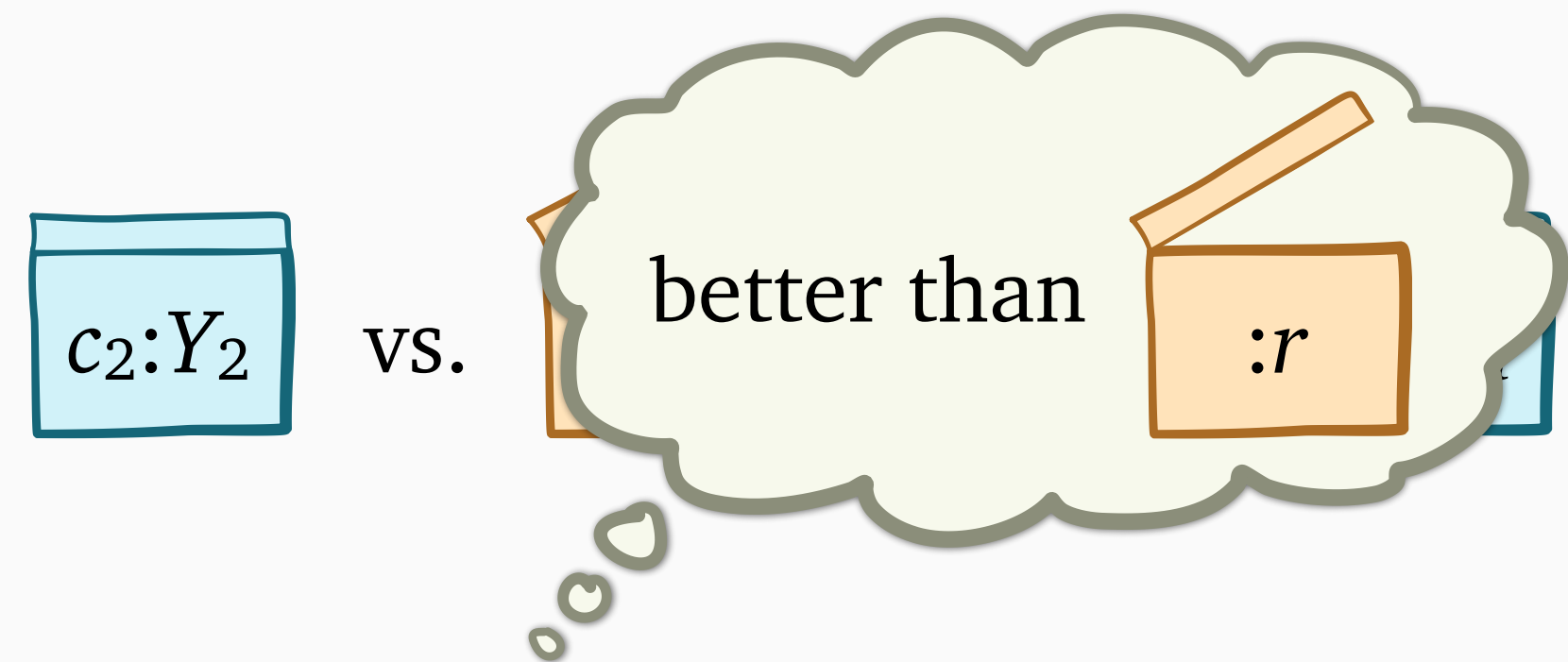
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$:



Box 1's perspective

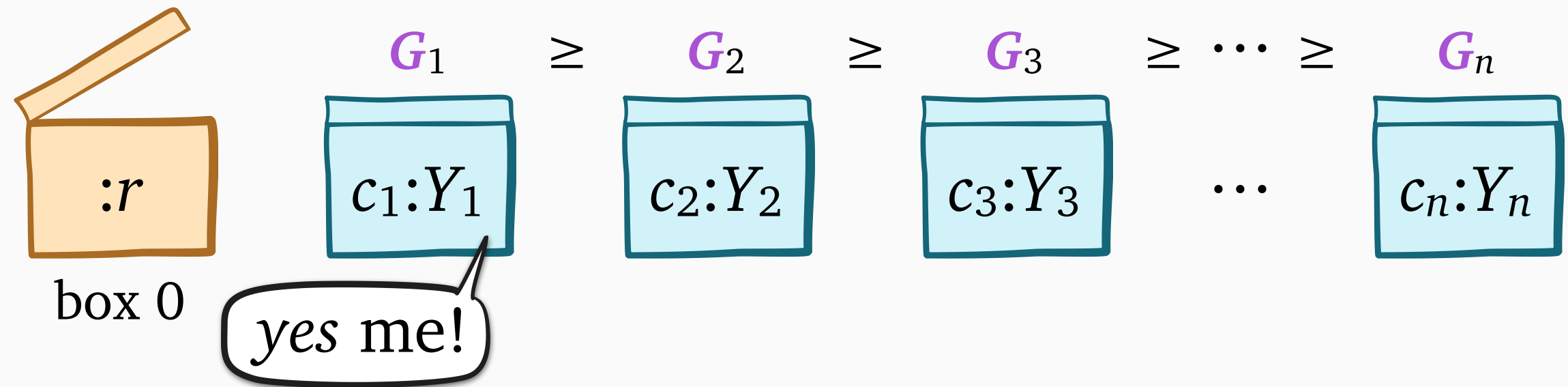


Box 2's perspective

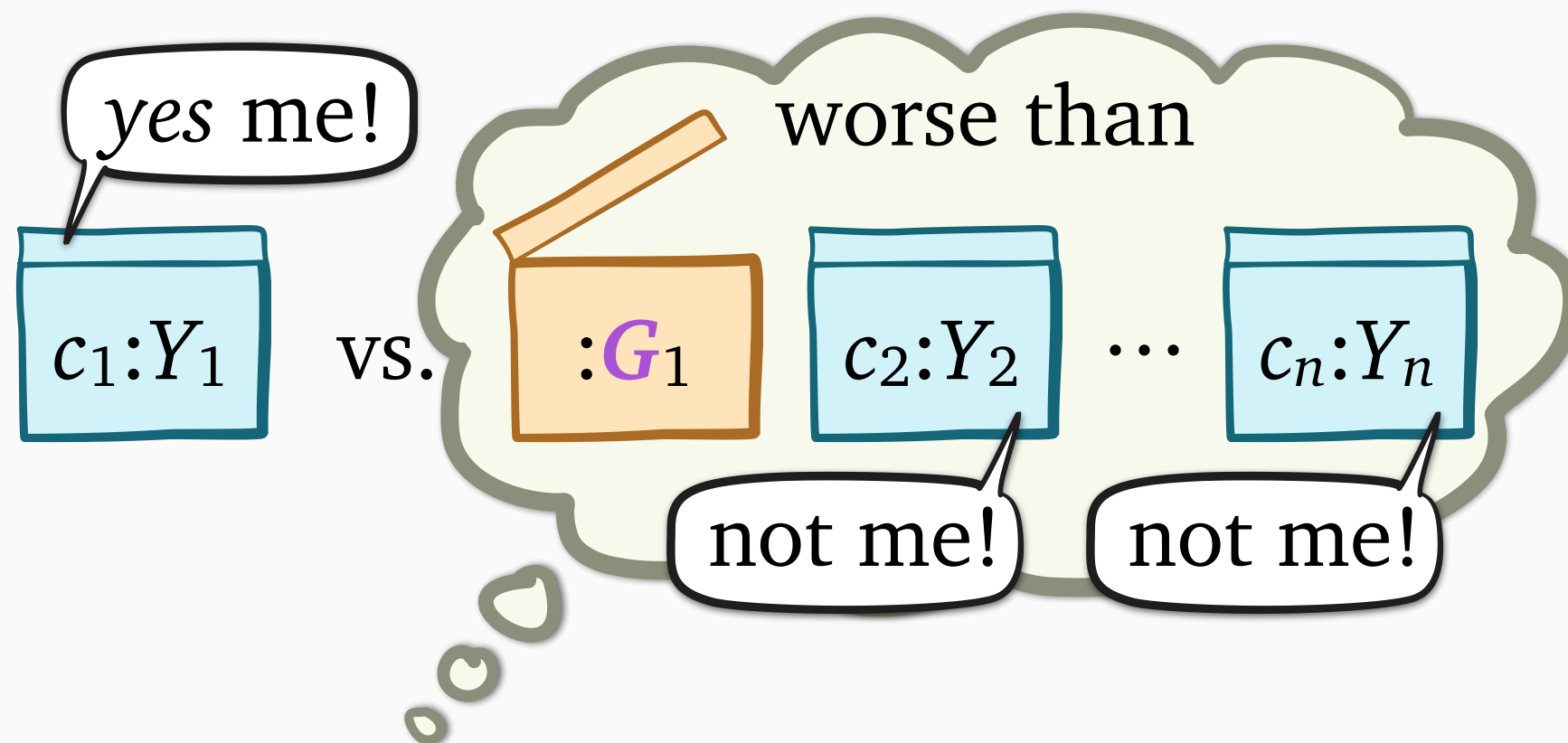


Approach: start at $r = \infty$, then decrease to $r = 0$

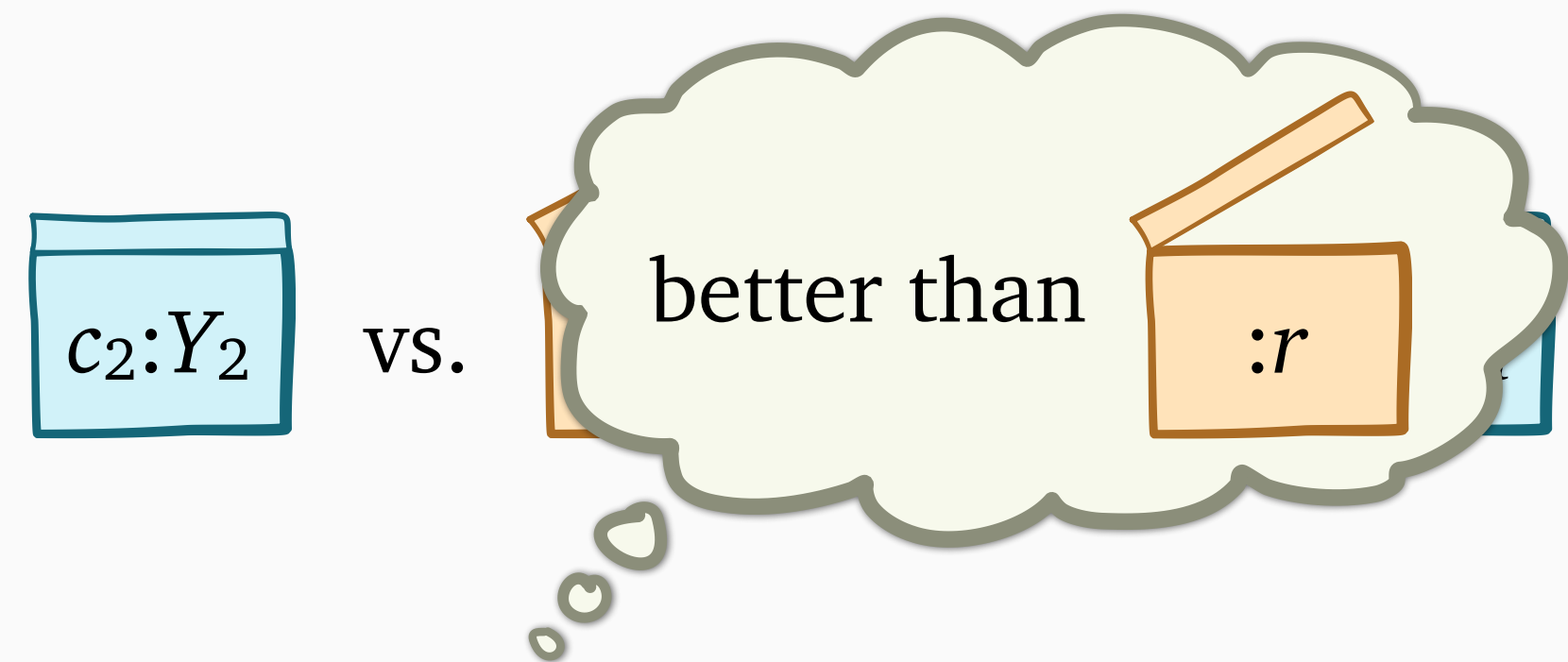
- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$: open box 1



Box 1's perspective

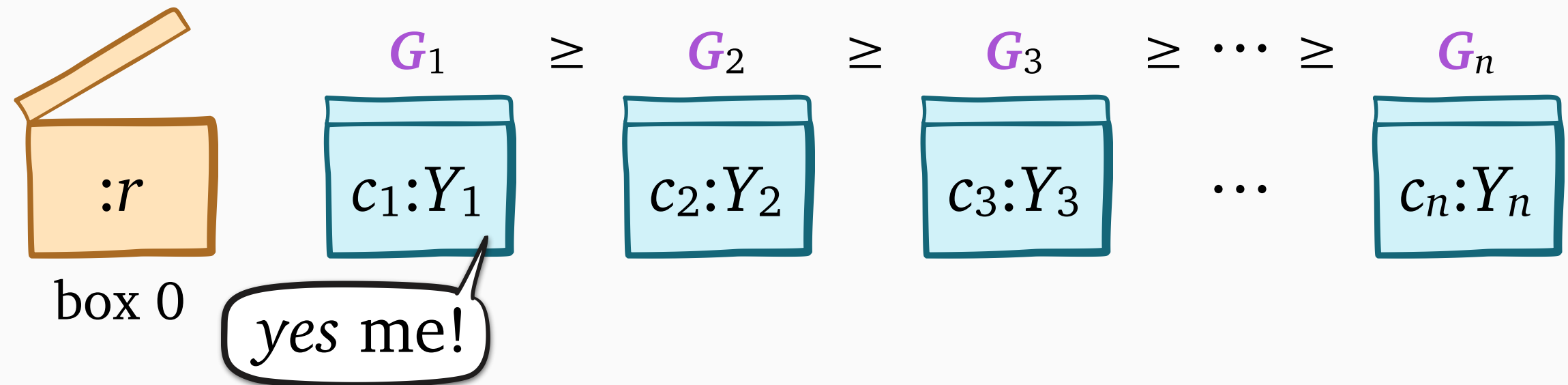


Box 2's perspective

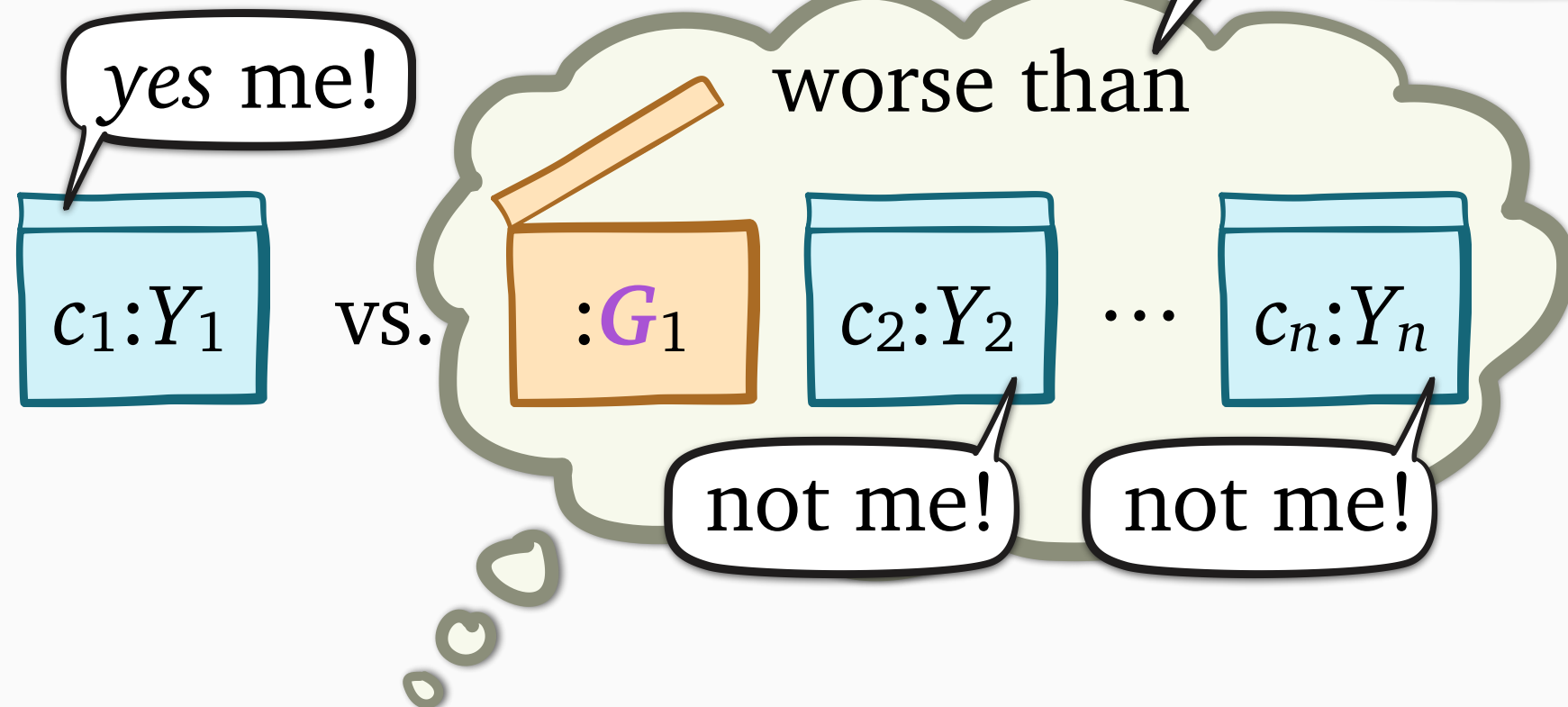


Approach: start at $r = \infty$, then decrease to $r = 0$

- $r \geq G_1$: select box 0
- $r = G_1$: open box 1
- $r < G_1$: open box 1



Box 1's perspective



! formalizing is tricky

Box 2's perspective



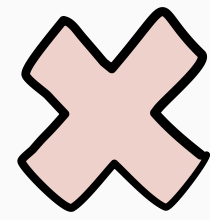
Quizzes

Pandora's box:
is **Gittins** still
optimal if...

Pandora's box:
is **Gittins** still
optimal if...

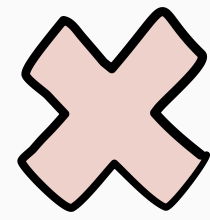
... we open k boxes at once?

Pandora's box:
is **Gittins** still
optimal if...



... we open k boxes at once?

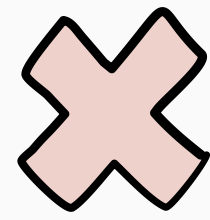
Pandora's box:
is **Gittins** still
optimal if...



... we open k boxes at once?

... we select k boxes at the end?

Pandora's box:
is **Gittins** still
optimal if...

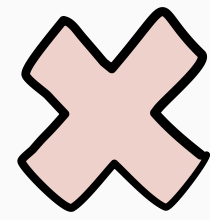


... we open k boxes at once?



... we select k boxes at the end?

Pandora's box:
is **Gittins** still
optimal if...



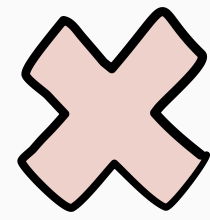
... we open k boxes at once?



... we select k boxes at the end?

... we select a *spanning tree* of boxes at the end?

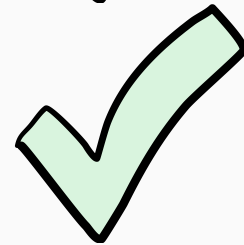
Pandora's box:
is **Gittins** still
optimal if...



... we open k boxes at once?



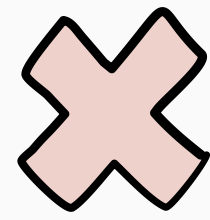
... we select k boxes at the end?



... we select a *spanning tree* of boxes at the end?

[Singla, 2018; Gupta et al., 2019]

Pandora's box:
is **Gittins** still
optimal if...



... we open k boxes at once?



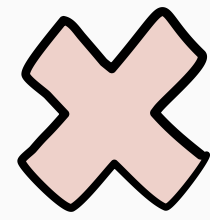
... we select k boxes at the end?



... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

... we can open at most n boxes?

Pandora's box:
is **Gittins** still
optimal if...



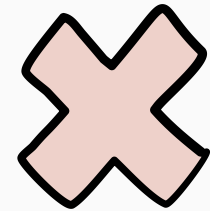
... we open k boxes at once?



... we select k boxes at the end?

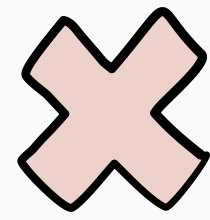


... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]



... we can open at most n boxes?

Pandora's box:
is **Gittins** still
optimal if...



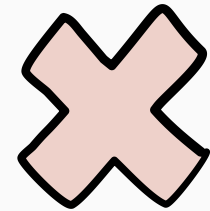
... we open k boxes at once?



... we select k boxes at the end?



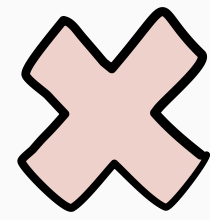
... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]



... we can open at most n boxes?

... we can open at most n boxes *in expectation*?

Pandora's box:
is **Gittins** still
optimal if...



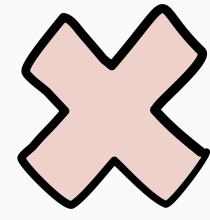
... we open k boxes at once?



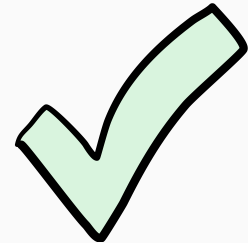
... we select k boxes at the end?



... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

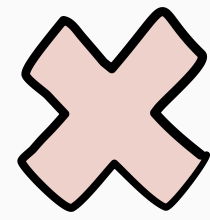


... we can open at most n boxes?



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

Pandora's box:
is **Gittins** still
optimal if...



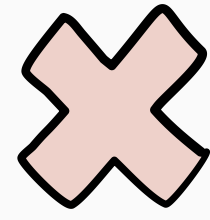
... we open k boxes at once?



... we select k boxes at the end?



... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]



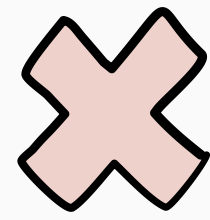
... we can open at most n boxes?



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

... we can select a closed box?

Pandora's box:
is **Gittins** still
optimal if...



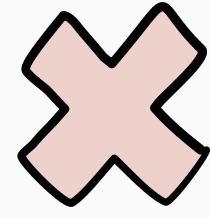
... we open k boxes at once?



... we select k boxes at the end?



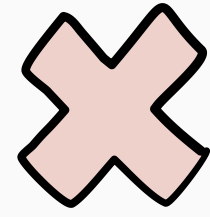
... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]



... we can open at most n boxes?

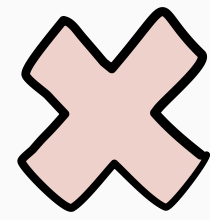


... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

Pandora's box:
is **Gittins** still
optimal if...



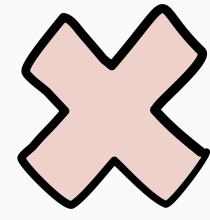
... we open k boxes at once?



... we select k boxes at the end?

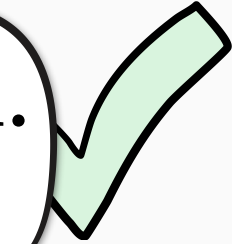


... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

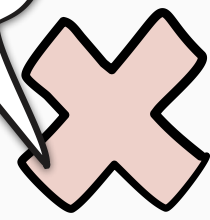


... we can open at most n boxes?

MDP selection, a.k.a.
bandit superprocess

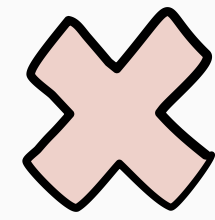


... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

Pandora's box:
is **Gittins** still
optimal if...



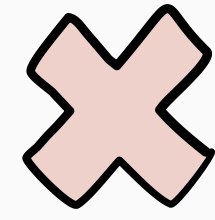
... we open k boxes at once?



... we select k boxes at the end?

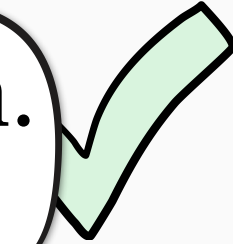


... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

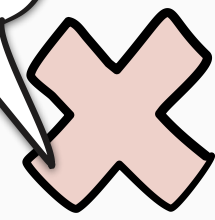


... we can open at most n boxes?

MDP selection, a.k.a.
bandit superprocess



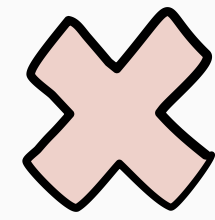
... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

... there are correlations between box values?

Pandora's box:
is **Gittins** still
optimal if...



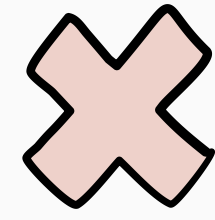
... we open k boxes at once?



... we select k boxes at the end?

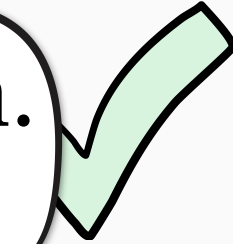


... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

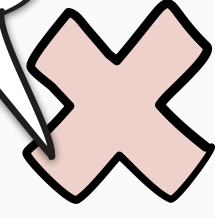


... we can open at most n boxes?

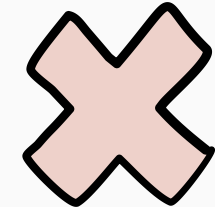
MDP selection, a.k.a.
bandit superprocess



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]

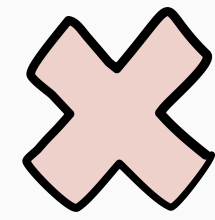


... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]



... there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

Pandora's box:
is **Gittins** still
optimal if...



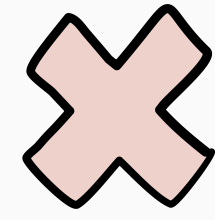
... we open k boxes at once?



... we select k boxes at the end?

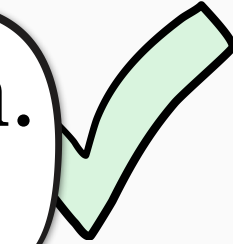


... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

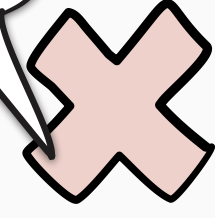


... we can open at most n boxes?

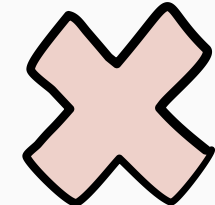
MDP selection, a.k.a.
bandit superprocess



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



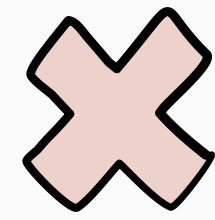
... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]



... there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

... there are multiple inspection steps?

Pandora's box: is **Gittins** still optimal if...



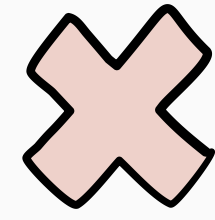
... we open k boxes at once?



... we select k boxes at the end?

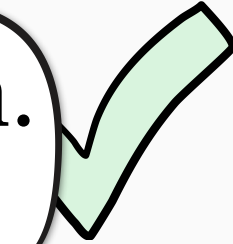


... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

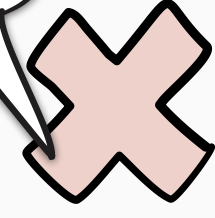


... we can open at most n boxes?

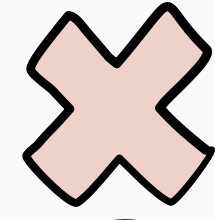
MDP selection, a.k.a.
bandit superprocess



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

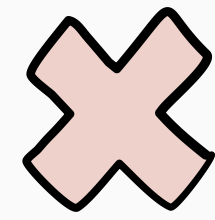


... there are correlations between box values?
[Gergatsouli & Tzamos, 2023]



... there are multiple inspection steps?

Pandora's box:
is **Gittins** still
optimal if...



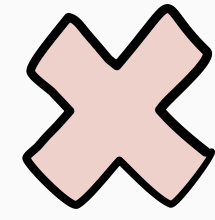
... we open k boxes at once?



... we select k boxes at the end?

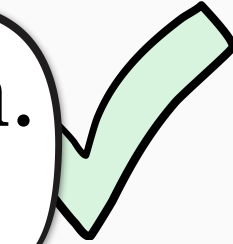


... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]

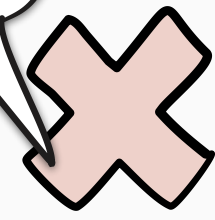


... we can open at most n boxes?

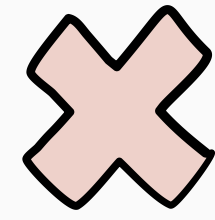
MDP selection, a.k.a.
bandit superprocess



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]



... there are correlations between box values?
[Gergatsouli & Tzamos, 2023]



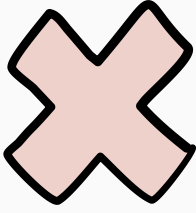
... there are multiple inspection steps?

... and with **discounted**
costs/rewards

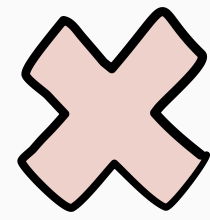
Mean scheduling:
is **Gittins** still
optimal if...

Mean scheduling:
is **Gittins** still
optimal if...

... we serve k jobs at once?

Mean scheduling:  ... we serve k jobs at once?
is **Gittins** still
optimal if...

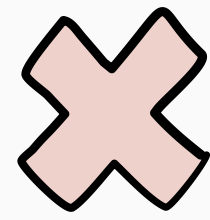
Mean scheduling:
is **Gittins** still
optimal if...



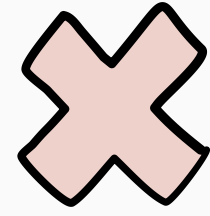
... we serve k jobs at once?

... jobs arrive over time (arbitrary)

Mean scheduling:
is **Gittins** still
optimal if...

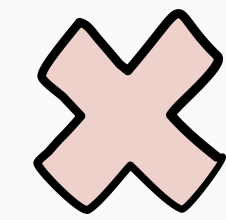
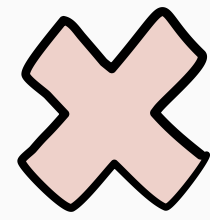


... we serve k jobs at once?



... jobs arrive over time (arbitrary)

Mean scheduling:
is **Gittins** still
optimal if...

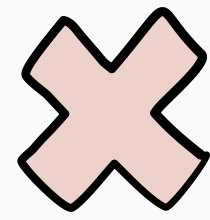


... we serve k jobs at once?

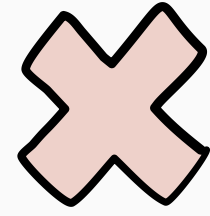
... jobs arrive over time (arbitrary)

... jobs arrive over time (Poisson)

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?

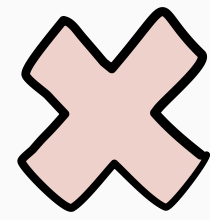


... jobs arrive over time (arbitrary)

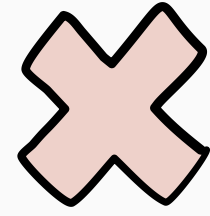


... jobs arrive over time (Poisson)

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



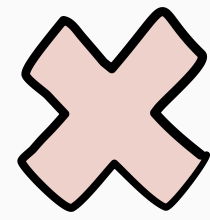
... jobs arrive over time (arbitrary)



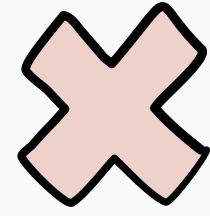
... jobs arrive over time (Poisson)

... Poisson arrivals affected by job in service?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



... jobs arrive over time (arbitrary)

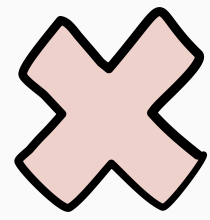


... jobs arrive over time (Poisson)

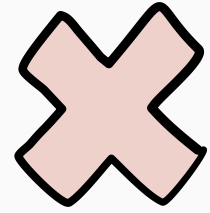


... Poisson arrivals affected by job in service?

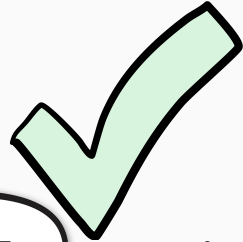
Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?

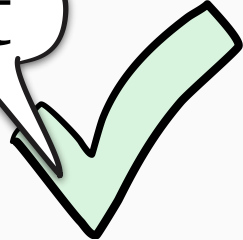


... jobs arrive over time (arbitrary)



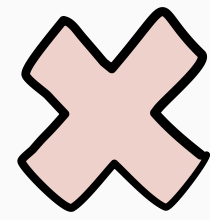
... jobs arrive over time (Poisson)

branching bandit

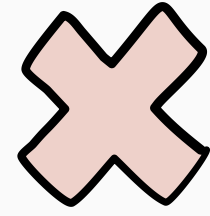


... Poisson arrivals affected by job in service?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

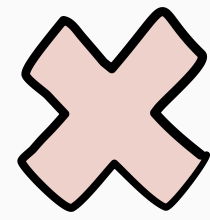
branching bandit



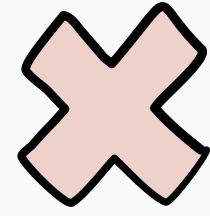
... Poisson arrivals affected by job in service?

... different jobs have different holding costs?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?

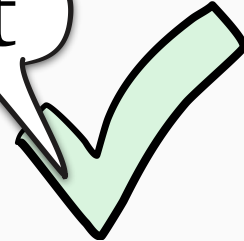


... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

branching bandit

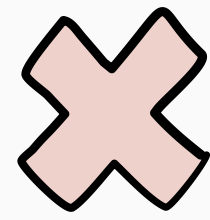


... Poisson arrivals affected by job in service?

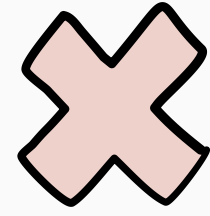


... different jobs have different holding costs?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?

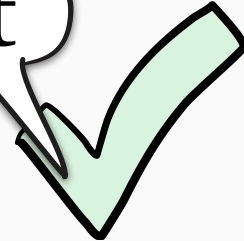


... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

branching bandit



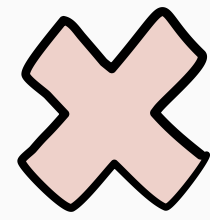
... Poisson arrivals affected by job in service?



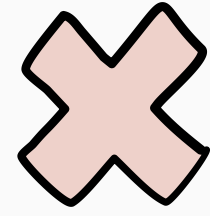
... different jobs have different holding costs?

... holding costs change over time (arbitrary)?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?

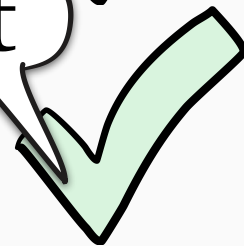


... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

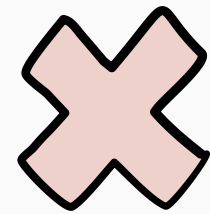
branching bandit



... Poisson arrivals affected by job in service?

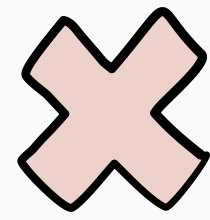


... different jobs have different holding costs?

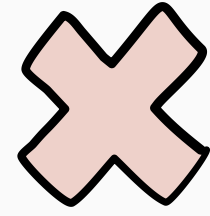


... holding costs change over time (arbitrary)?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

branching bandit

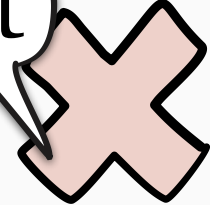


... Poisson arrivals affected by job in service?



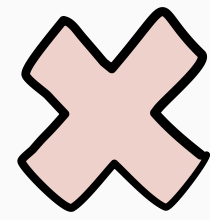
... different jobs have different holding costs?

restless bandit

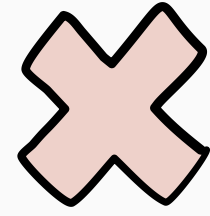


... holding costs change over time (arbitrary)?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

branching bandit

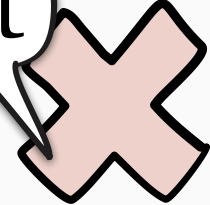


... Poisson arrivals affected by job in service?



... different jobs have different holding costs?

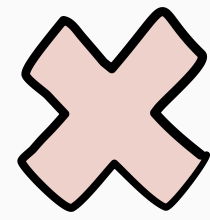
restless bandit



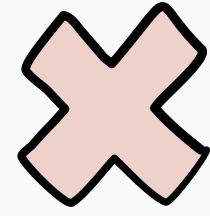
... holding costs change over time (arbitrary)?

... holding costs change over time (convex)?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

branching bandit

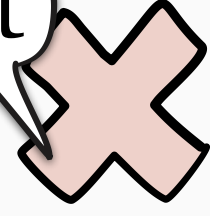


... Poisson arrivals affected by job in service?

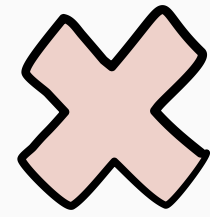


... different jobs have different holding costs?

restless bandit

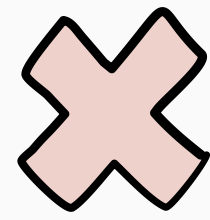


... holding costs change over time (arbitrary)?

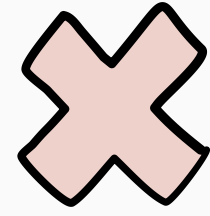


... holding costs change over time (convex)?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

branching bandit

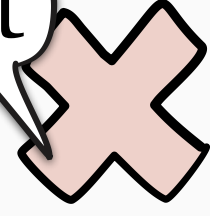


... Poisson arrivals affected by job in service?

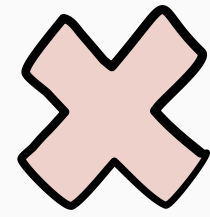


... different jobs have different holding costs?

restless bandit



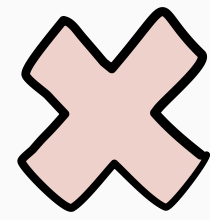
... holding costs change over time (arbitrary)?



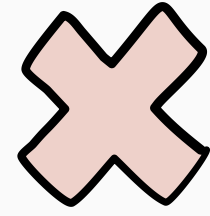
... holding costs change over time (convex)?

... holding costs change *during service*?

Mean scheduling:
is **Gittins** still
optimal if...



... we serve k jobs at once?



... jobs arrive over time (arbitrary)



... jobs arrive over time (Poisson)

branching bandit

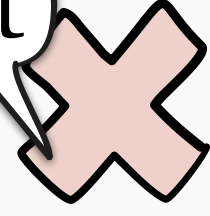


... Poisson arrivals affected by job in service?

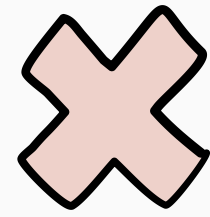


... different jobs have different holding costs?

restless bandit



... holding costs change over time (arbitrary)?



... holding costs change over time (convex)?



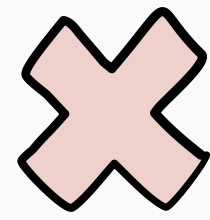
... holding costs change *during service*?

Mean scheduling:
is **Gittins** still
optimal if...

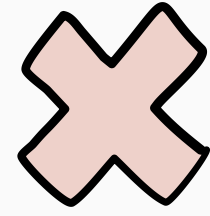
branching bandit

restless bandit

... and with **discounted**
costs/rewards



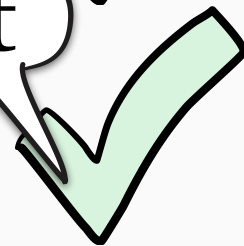
... we serve k jobs at once?



... jobs arrive over time (arbitrary)



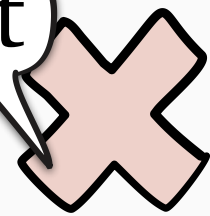
... jobs arrive over time (Poisson)



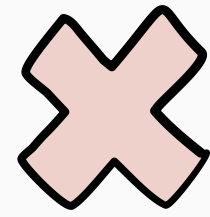
... Poisson arrivals affected by job in service?



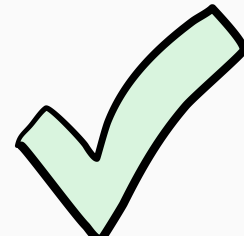
... different jobs have different holding costs?



... holding costs change over time (arbitrary)?



... holding costs change over time (convex)?

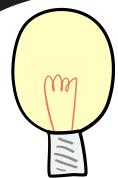


... holding costs change *during service*?

Mean scheduling: is **Gittins** still optimal if...

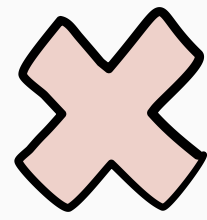
branching bandit

restless bandit

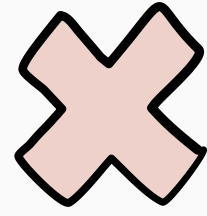


or **inflated!**

... and with **discounted**
costs/rewards



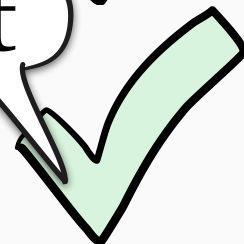
... we serve k jobs at once?



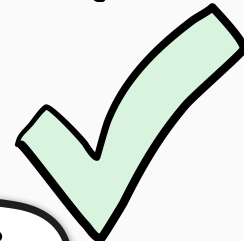
... jobs arrive over time (arbitrary)



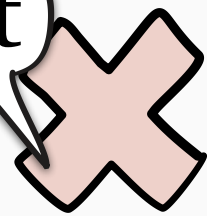
... jobs arrive over time (Poisson)



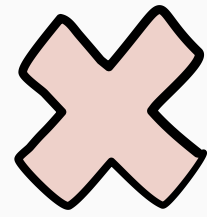
... Poisson arrivals affected by job in service?



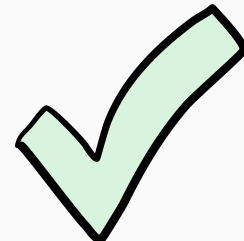
... different jobs have different holding costs?



... holding costs change over time (arbitrary)?



... holding costs change over time (convex)?

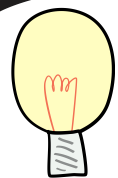


... holding costs change *during service*?

Mean scheduling: is **Gittins** still optimal if...

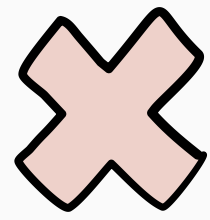
branching bandit

restless bandit

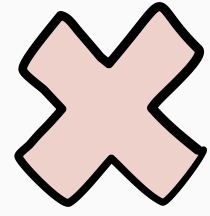


or **inflated!**

... and with **discounted**
costs/rewards



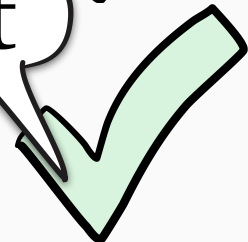
... we serve k jobs at once?



... jobs arrive over time (arbitrary)



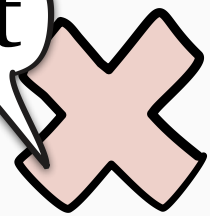
... jobs arrive over time (Poisson)



... Poisson arrivals affected by job in service?



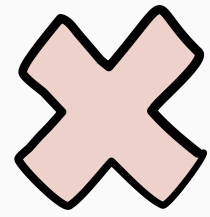
... different jobs have different holding costs?



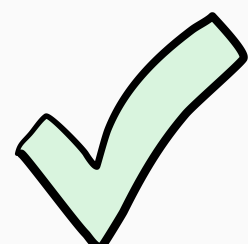
... holding costs change over time (arbitrary)?



exponential



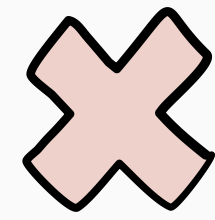
... holding costs change over time (convex)?



... holding costs change *during service*?

Pandora's box with optional inspection

Pandora's box:
is **Gittins** still
optimal if...



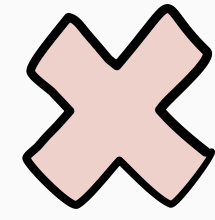
... we open k boxes at once?



... we select k boxes at the end?



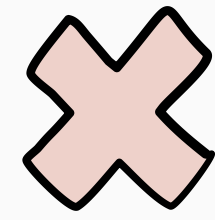
... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]



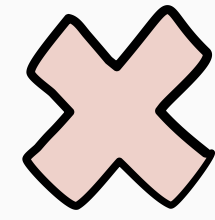
... we can open at most n boxes?



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

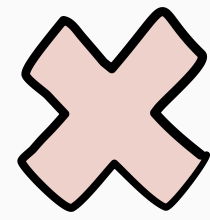


... there are correlations between box values?
[Gergatsouli & Tzamos, 2023]



... there are multiple inspection steps?

Pandora's box: is **Gittins** still optimal if...



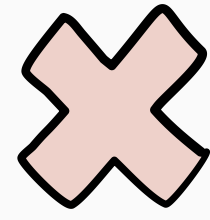
... we open k boxes at once?



... we select k boxes at the end?



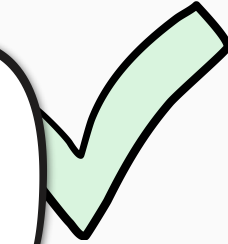
... we select a *spanning tree* of boxes at the end?
[Singla, 2018; Gupta et al., 2019]



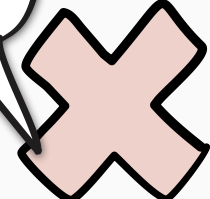
... we can open at most n boxes?



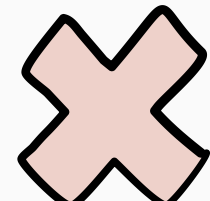
Choices are
difficult



... we can open at most n boxes *in expectation*?
[Aminian, Manshadi, & Niazadeh, 2025]



... we can select a closed box?
[Fu, Li, & Liu, 2023; Beyhaghi & Cai, 2023]

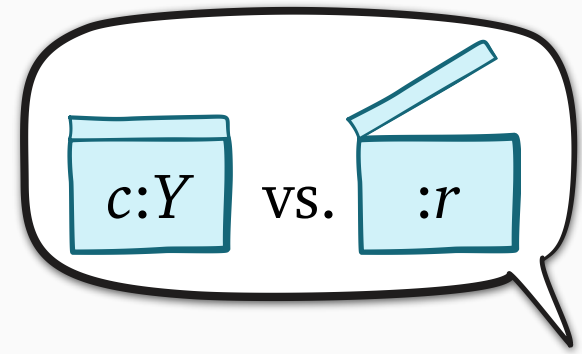


... there are correlations between box values?
[Gergatsouli & Tzamos, 2023]

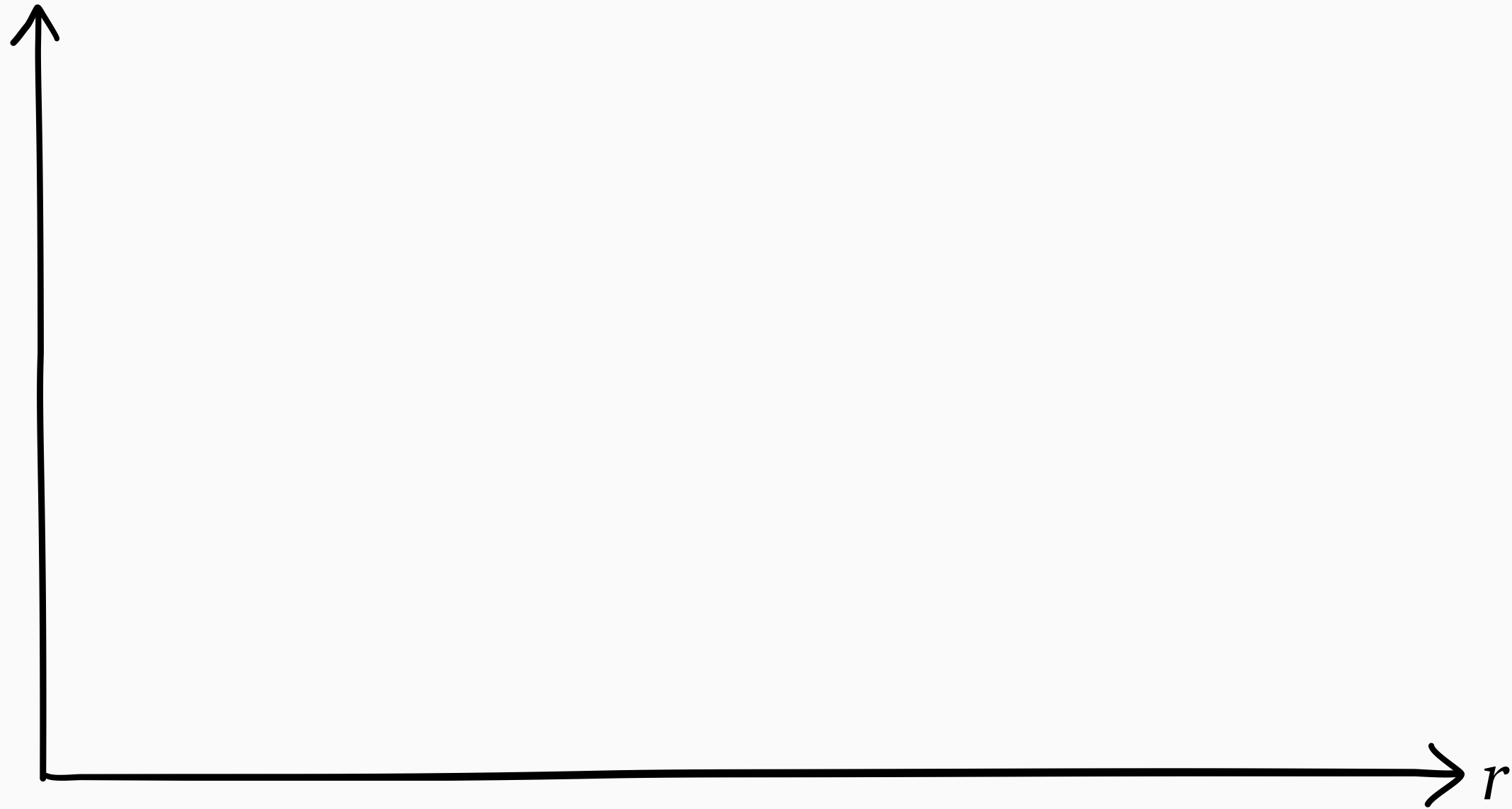


... there are multiple inspection steps?

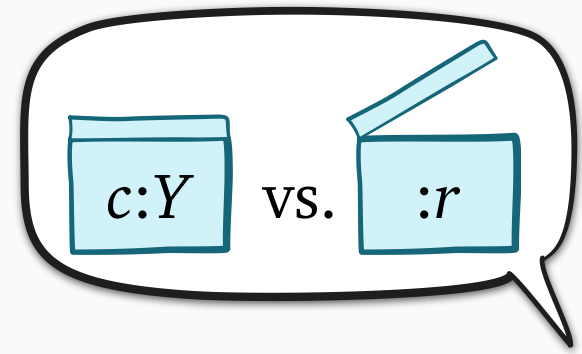
Choices are hard if they depend on context



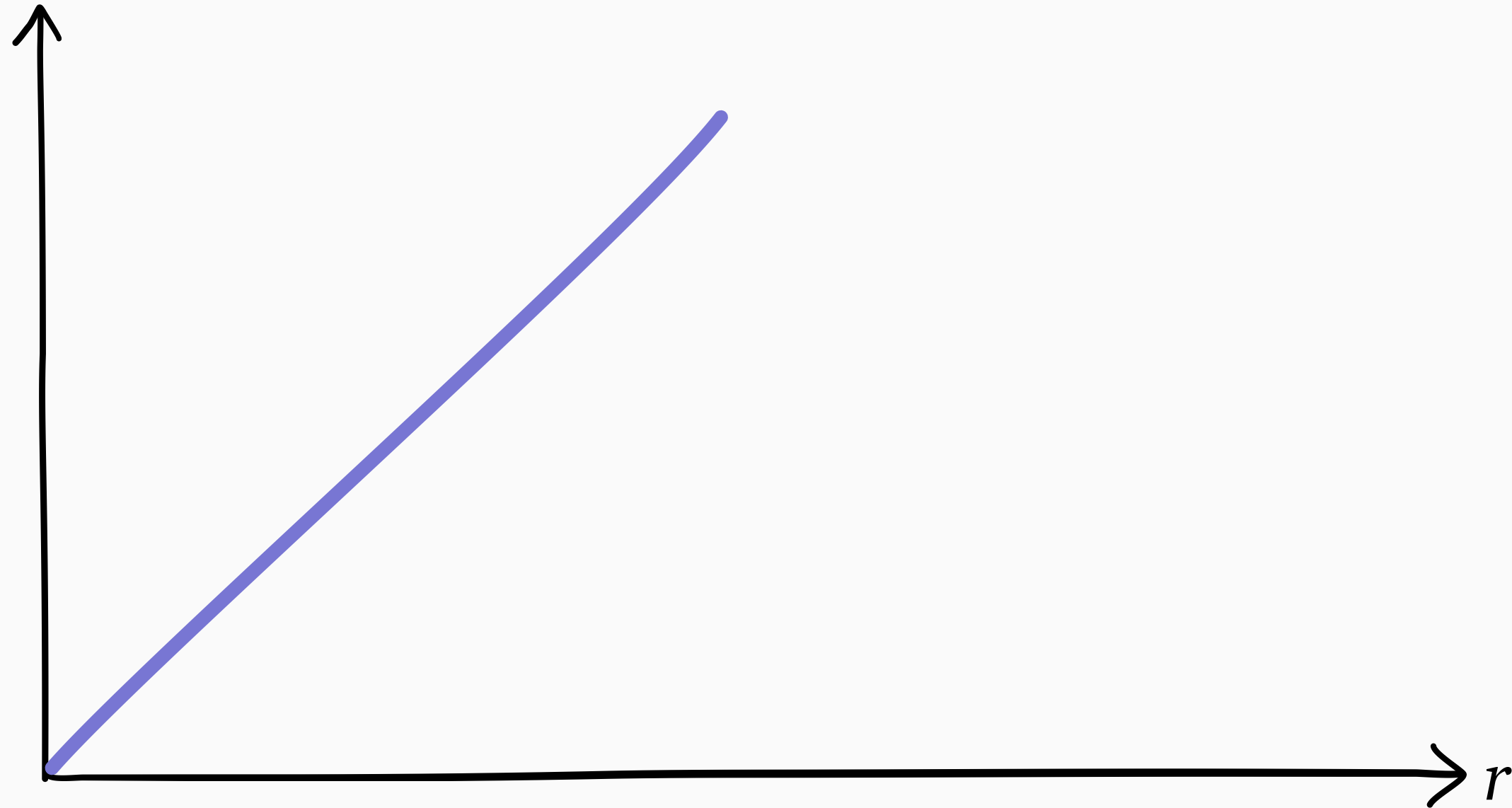
$E[\text{cost}(r)]$



Choices are hard if they depend on context

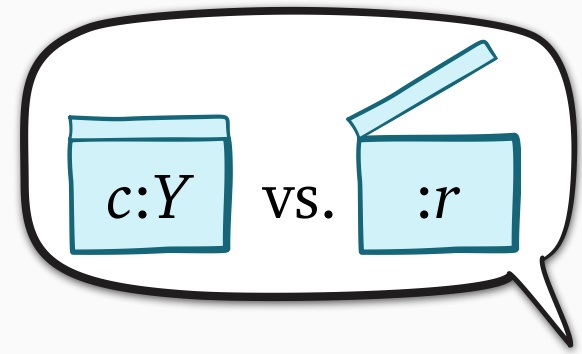


$E[\text{cost}(r)]$

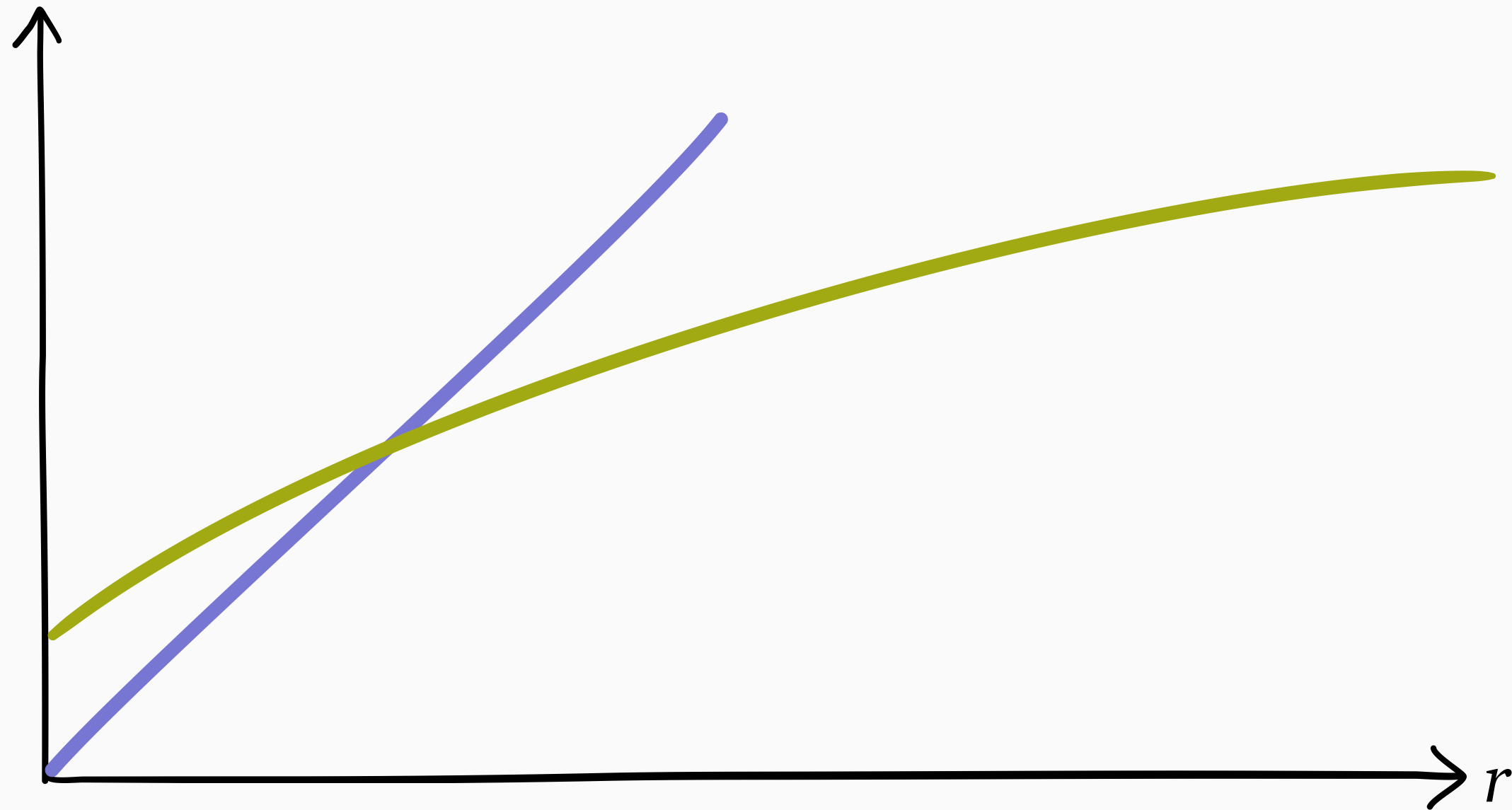


select $:r$

Choices are hard if they depend on context



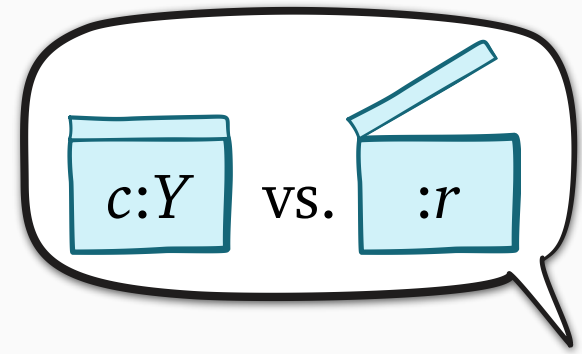
$E[\text{cost}(r)]$



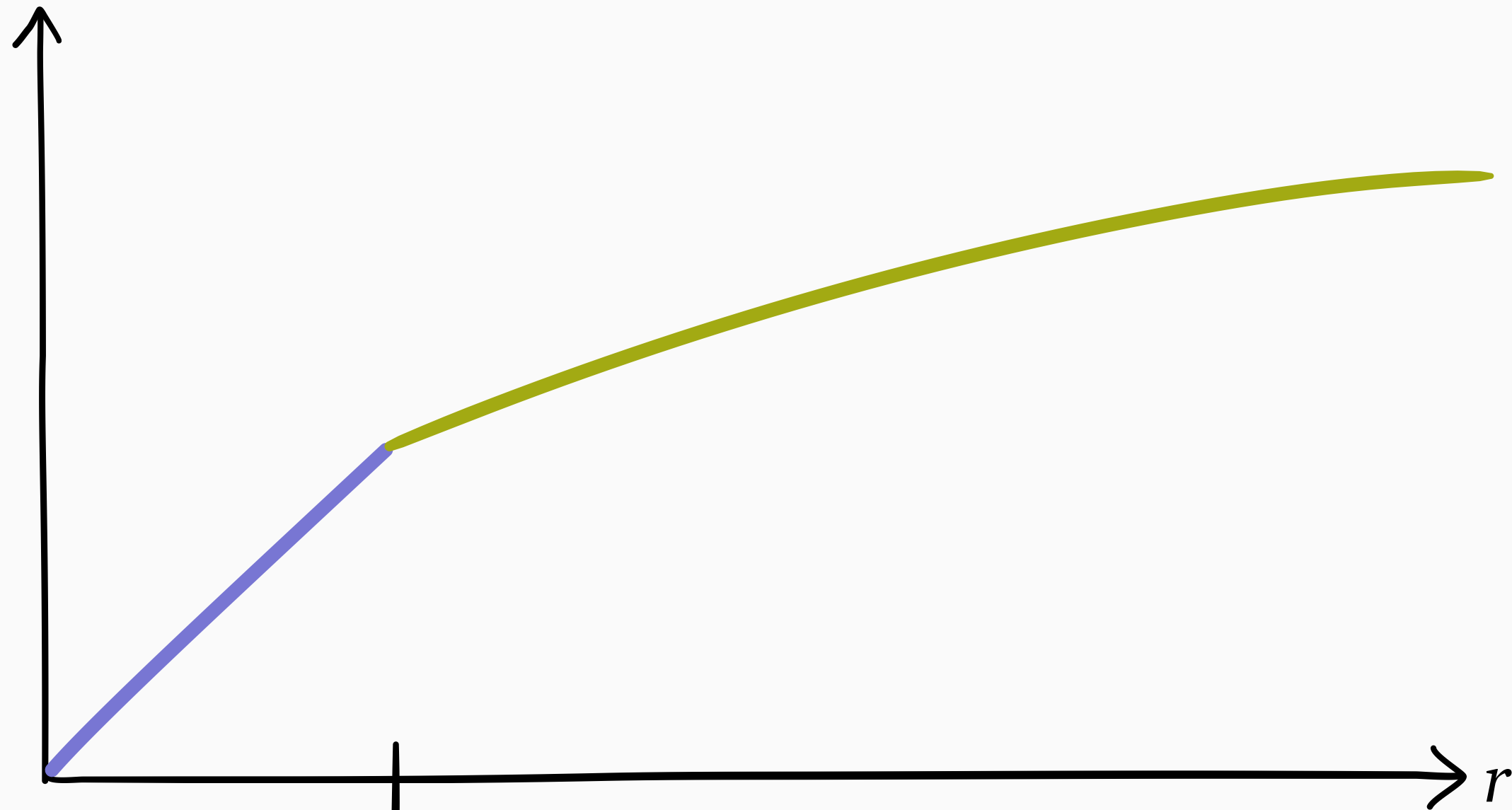
select $:r$

open $c:Y$

Choices are hard if they depend on context



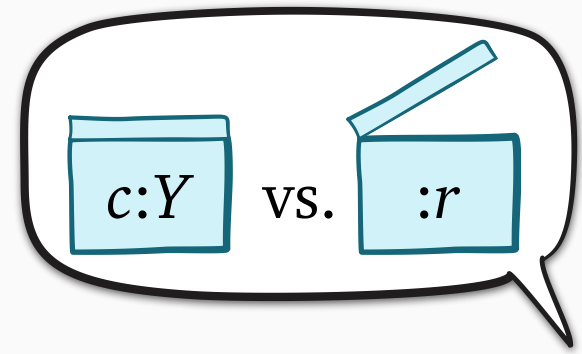
$E[\text{cost}(r)]$



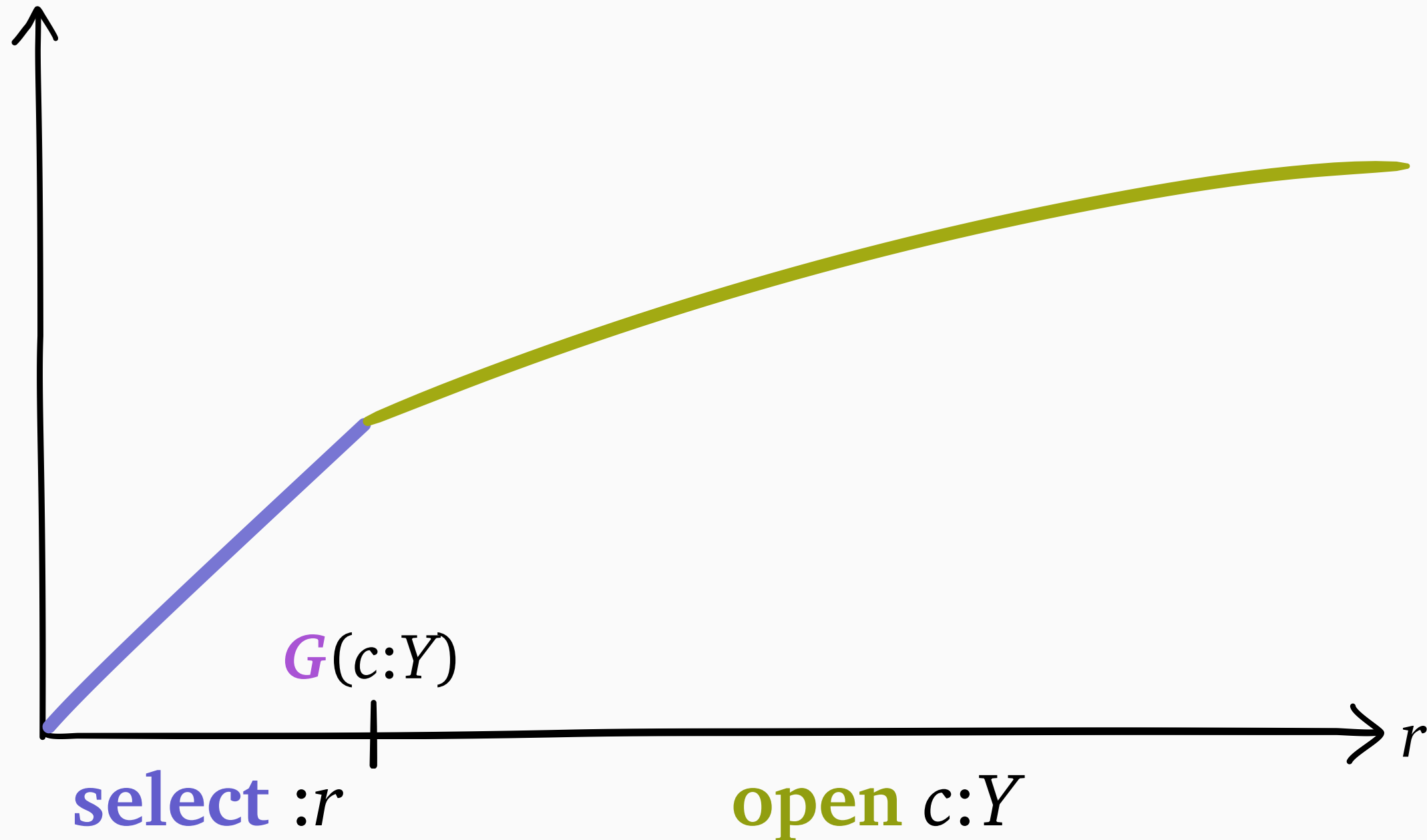
select $:r$

open $c:Y$

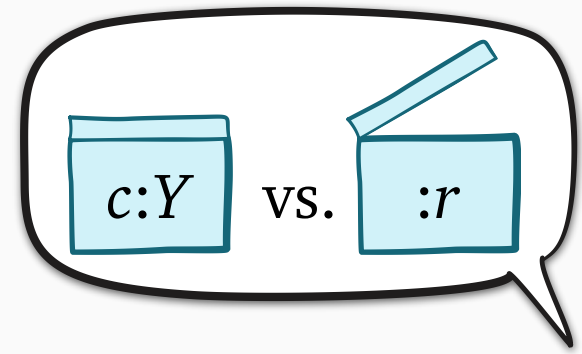
Choices are hard if they depend on context



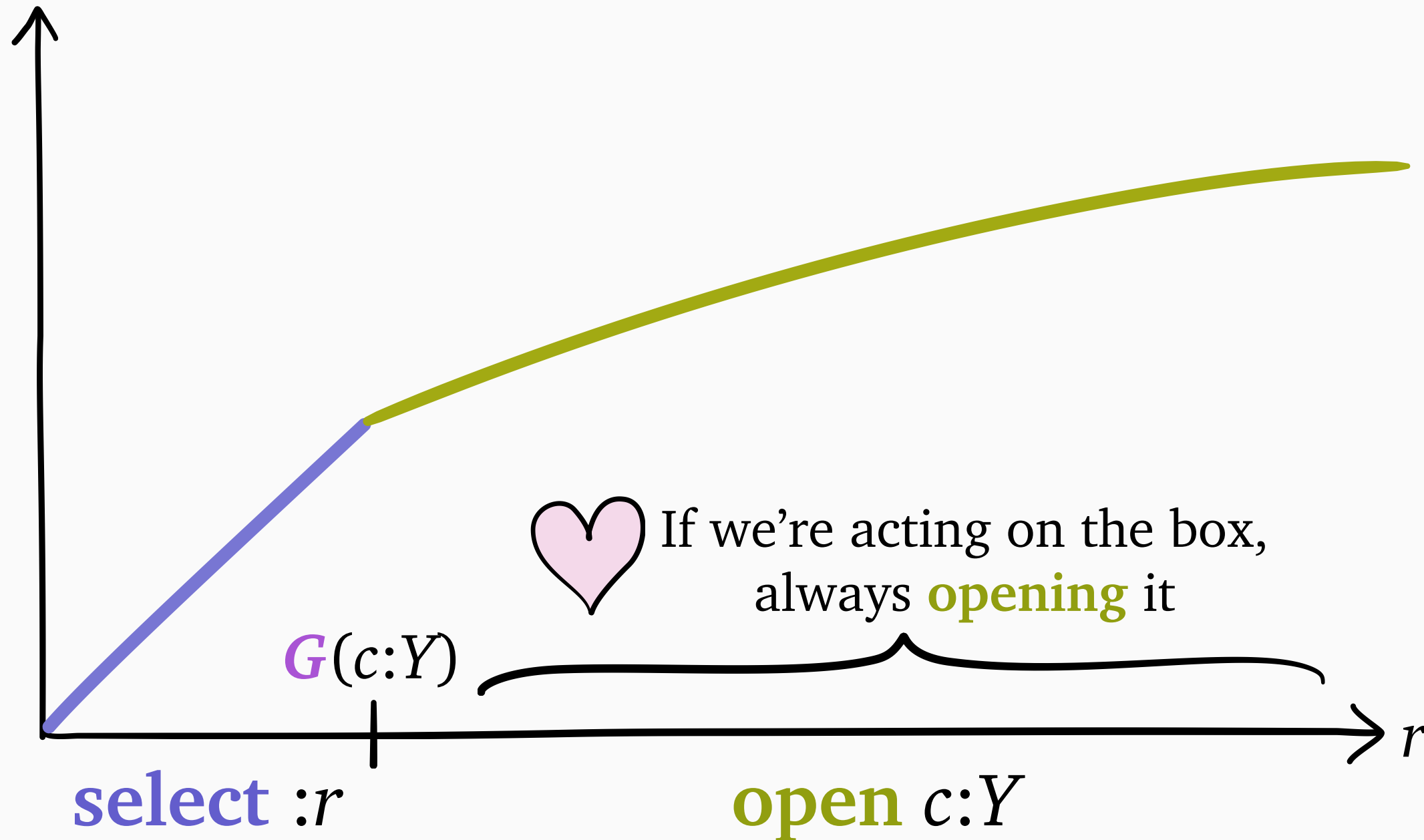
$E[\text{cost}(r)]$



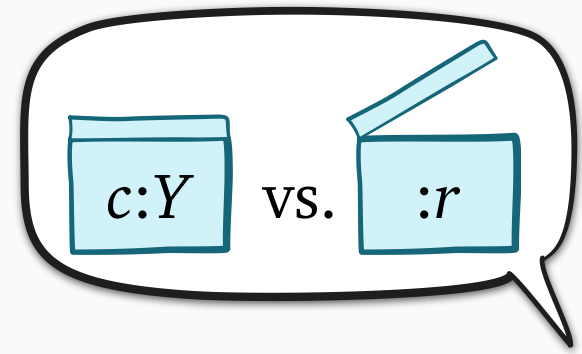
Choices are hard if they depend on context



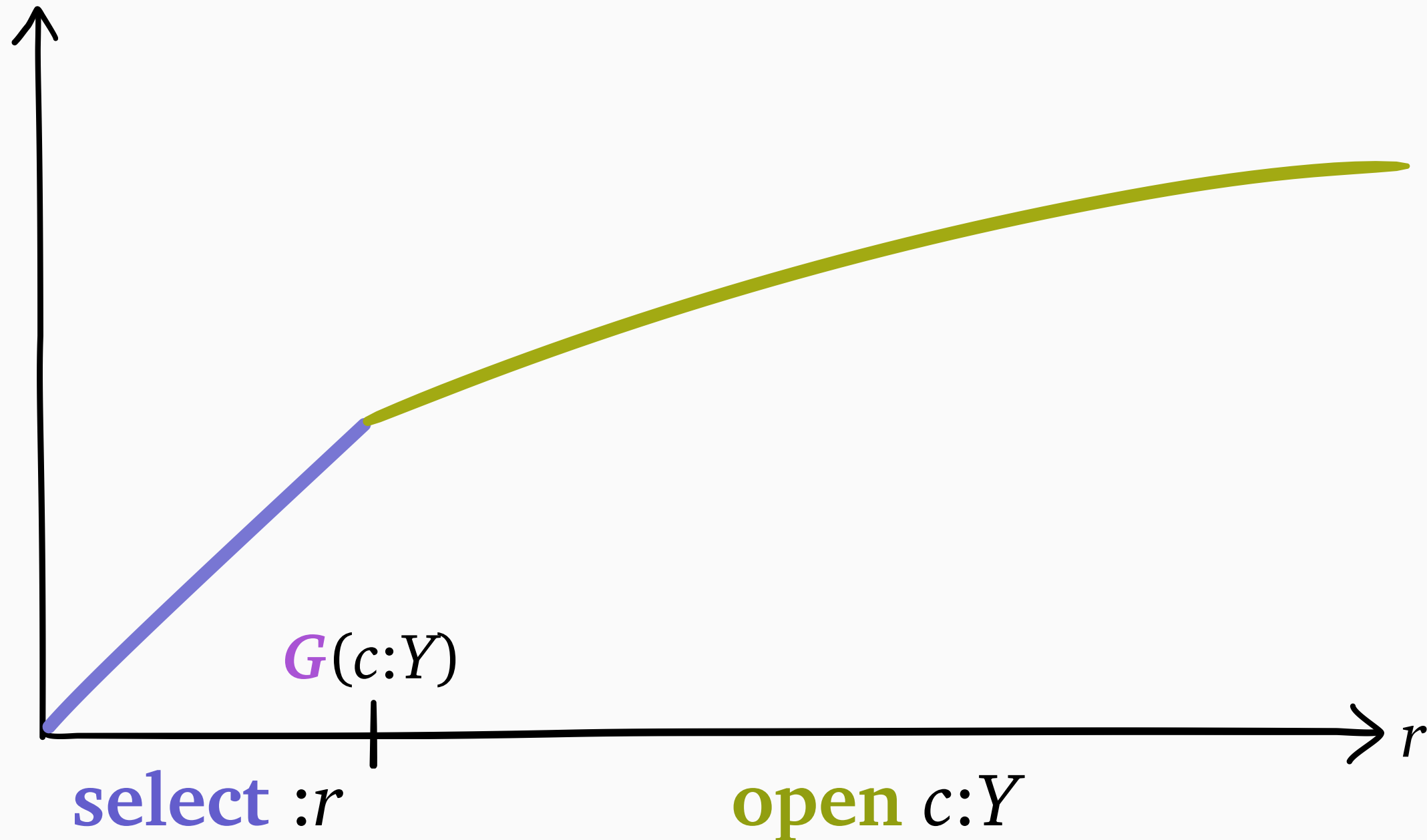
$E[\text{cost}(r)]$



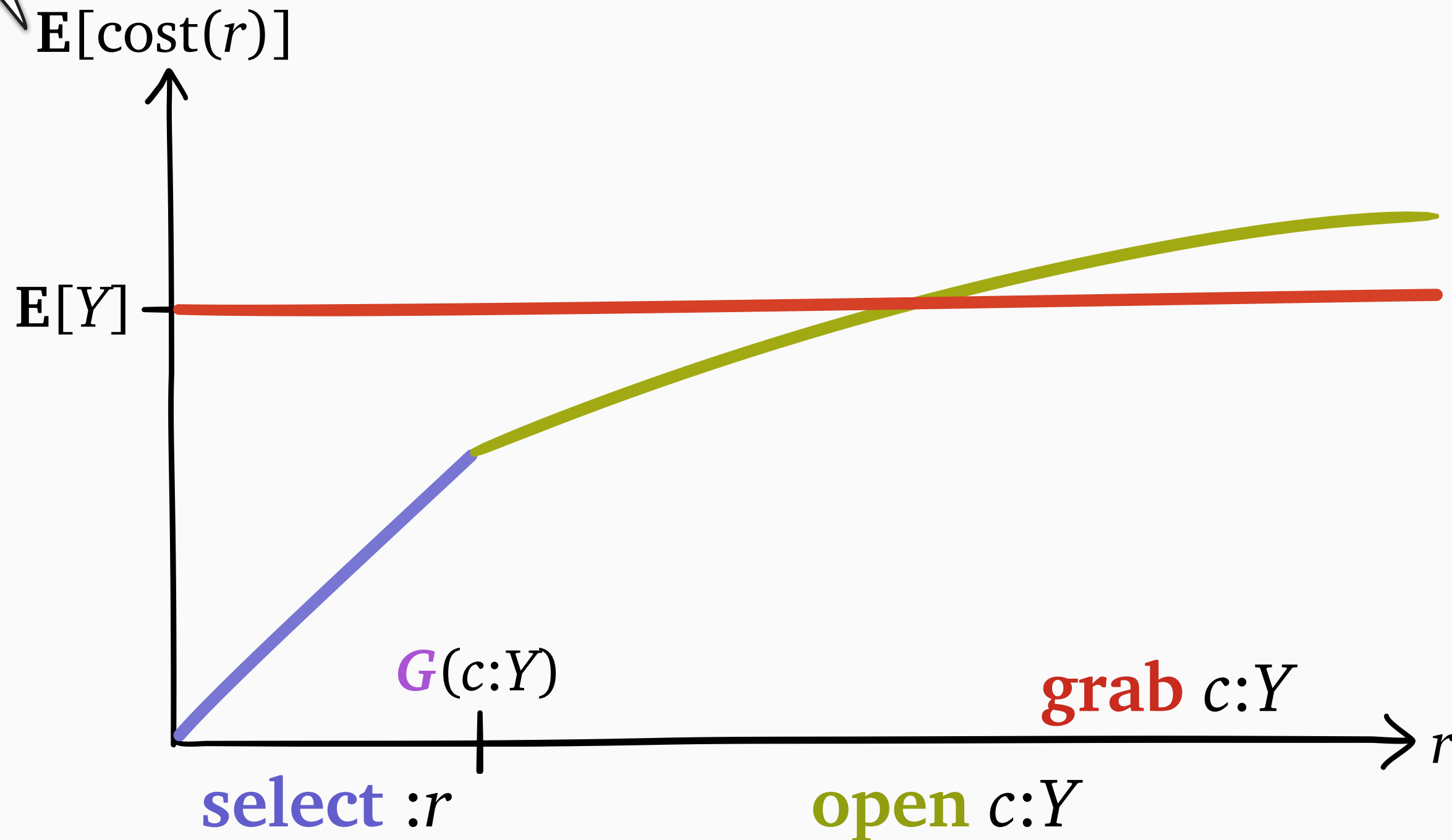
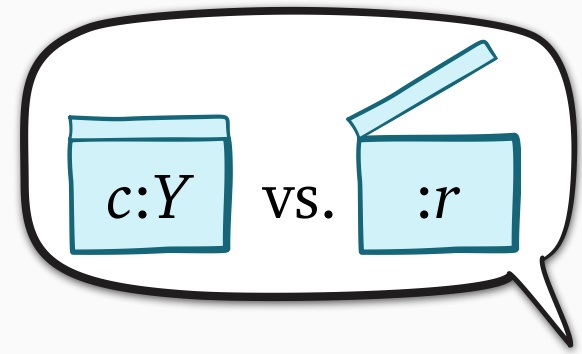
Choices are hard if they depend on context



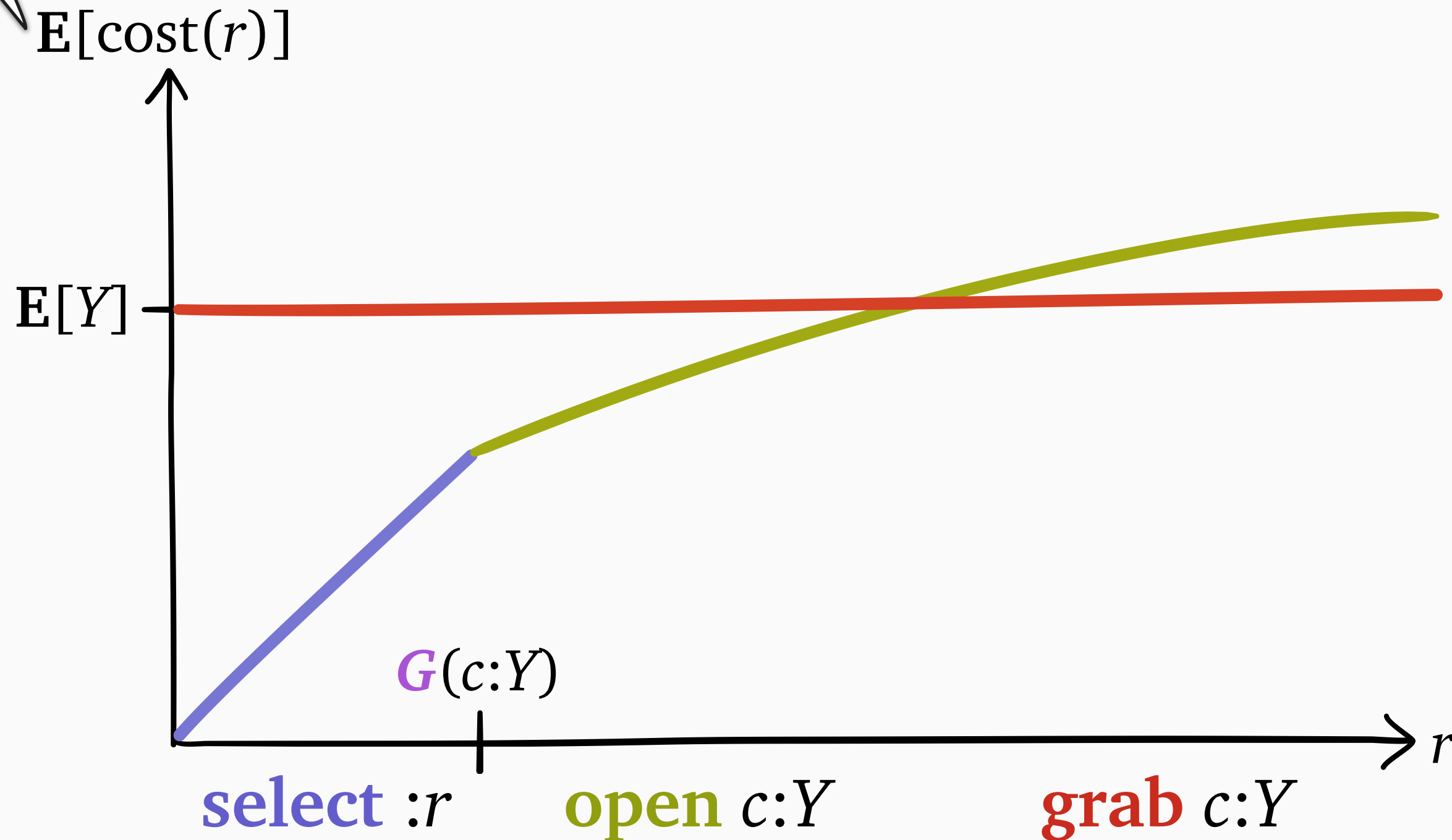
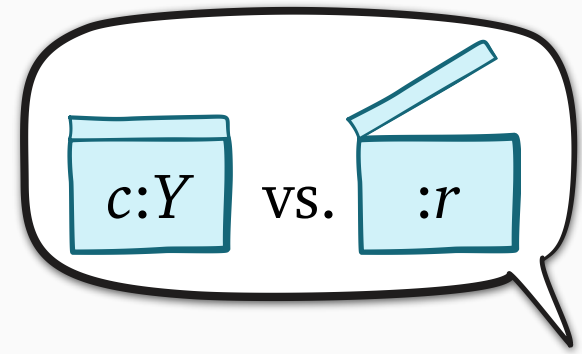
$E[\text{cost}(r)]$



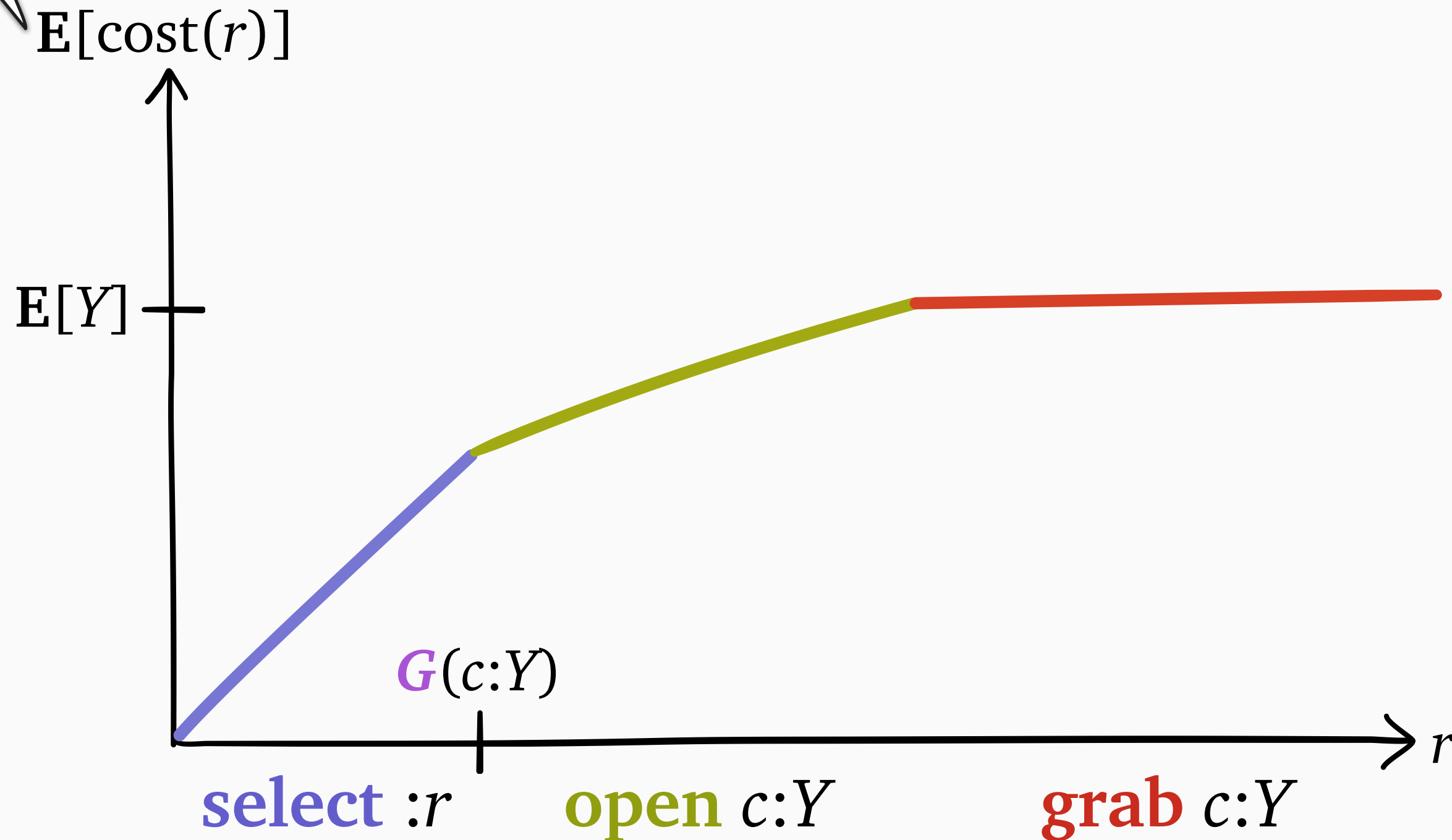
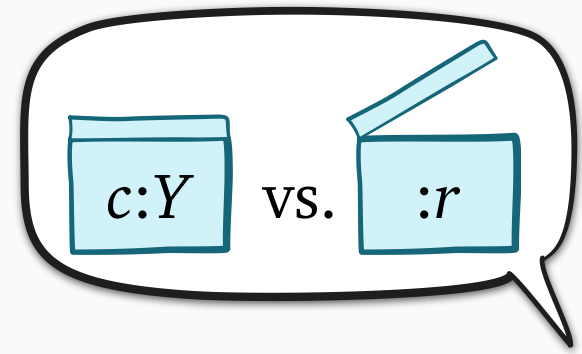
Choices are hard if they depend on context



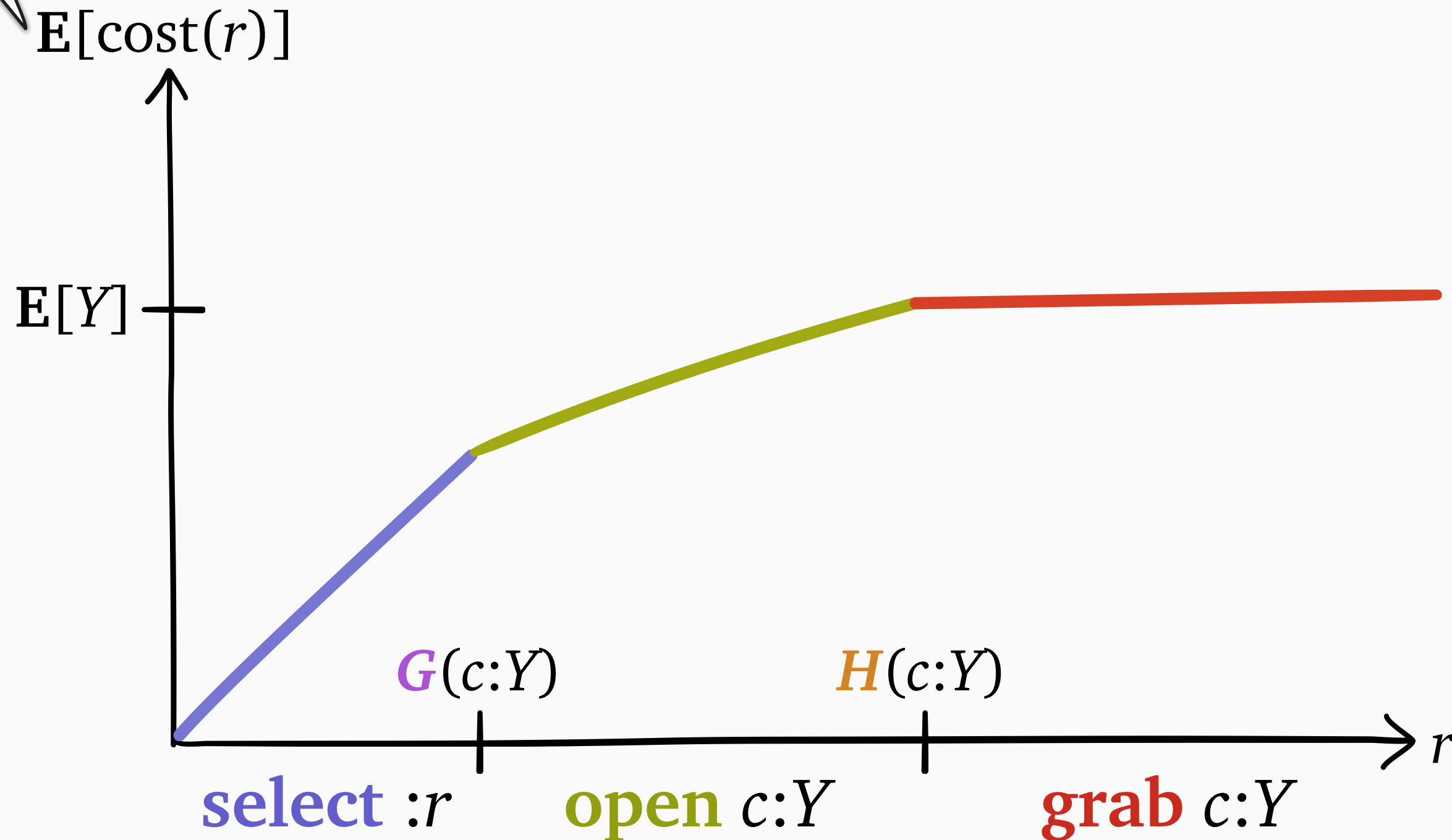
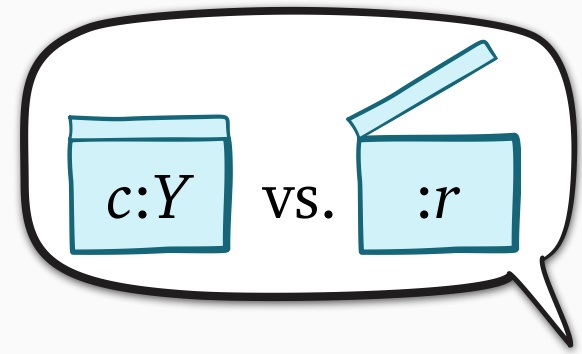
Choices are hard if they depend on context



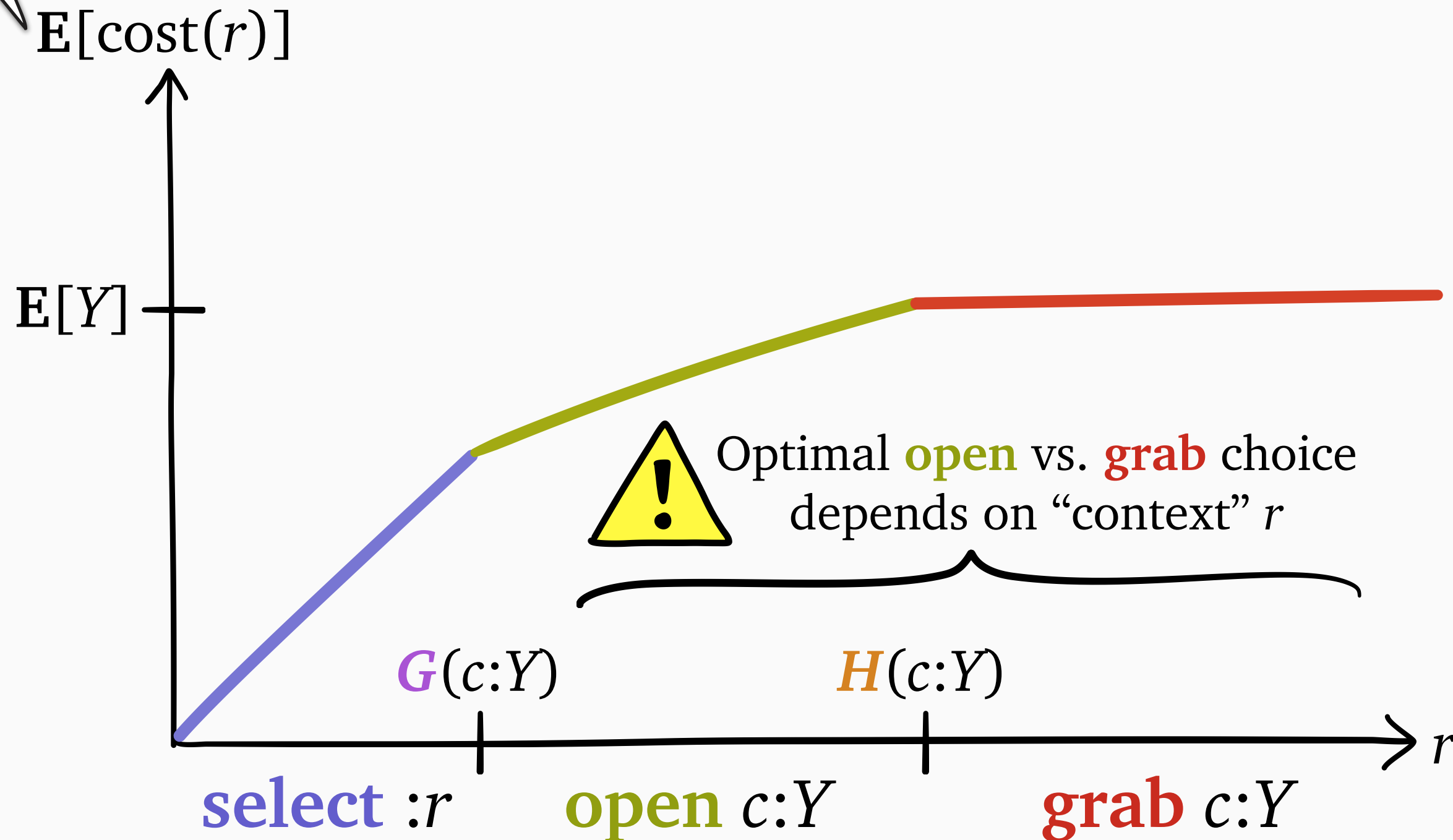
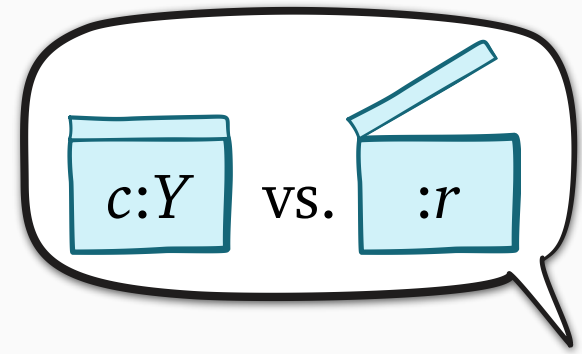
Choices are hard if they depend on context



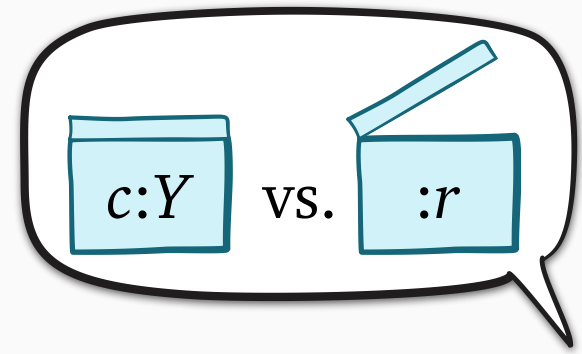
Choices are hard if they depend on context



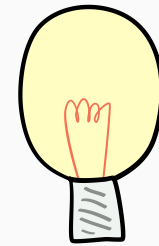
Choices are hard if they depend on context



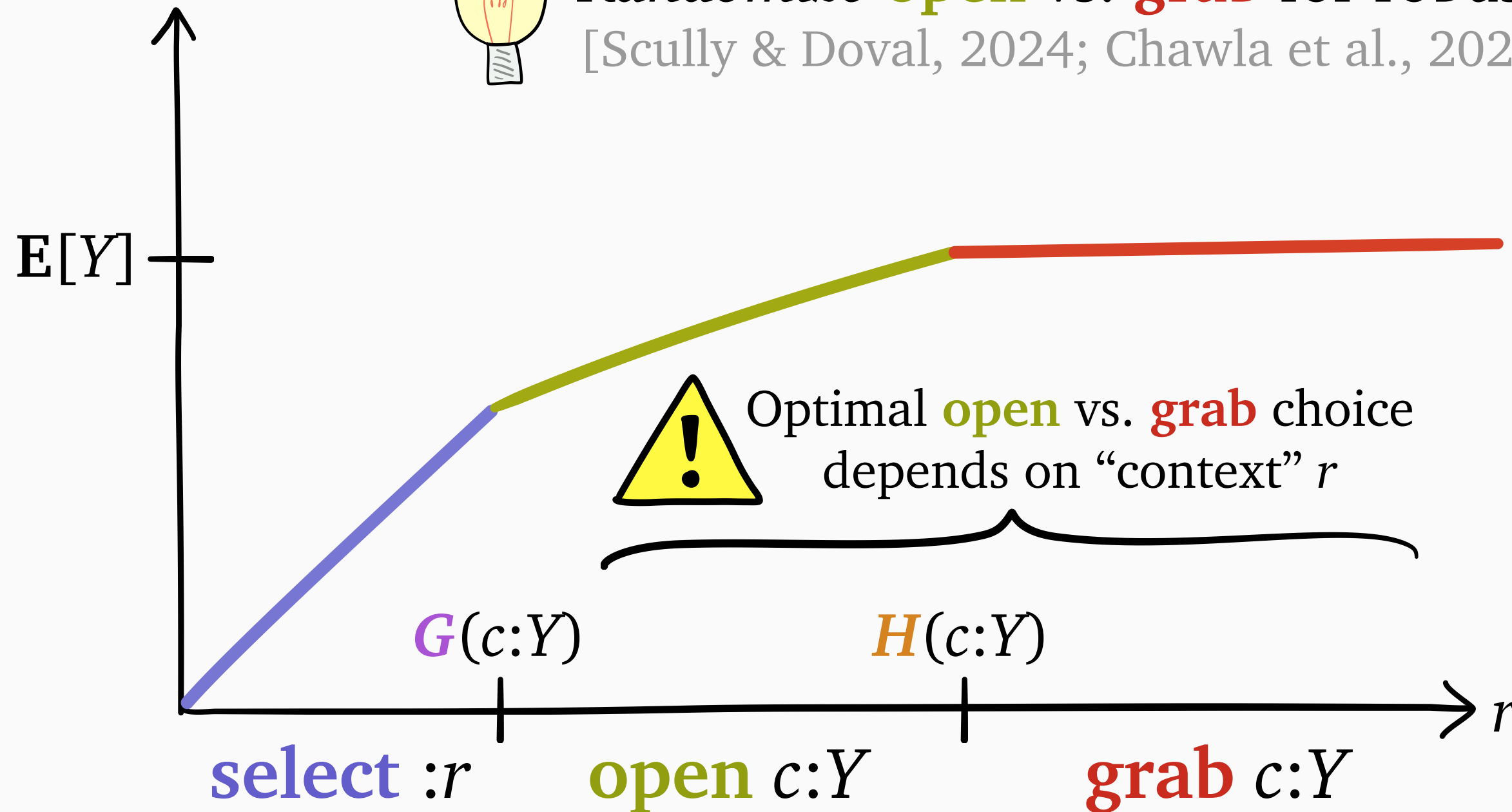
Choices are hard if they depend on context



$E[\text{cost}(r)]$



Randomize **open** vs. **grab** for robustness to all r
[Scully & Doval, 2024; Chawla et al., 2026]



Laura Doval