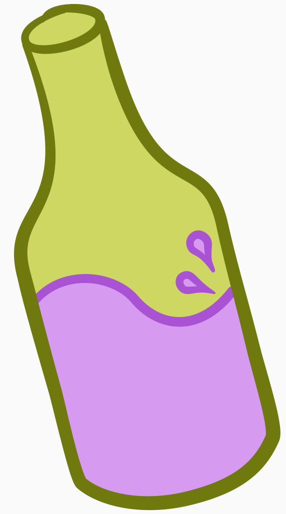
WINE



A New Queueing Identity *for* Analyzing Scheduling Policies *in* Multiserver Systems

Ziv Scully
CMU (now) → UC Berkeley → MIT/Harvard → Cornell (Fall 2023)

$$N = \int_0^\infty \frac{W(r)}{r^2}\, \mathrm{d}r$$

WINE

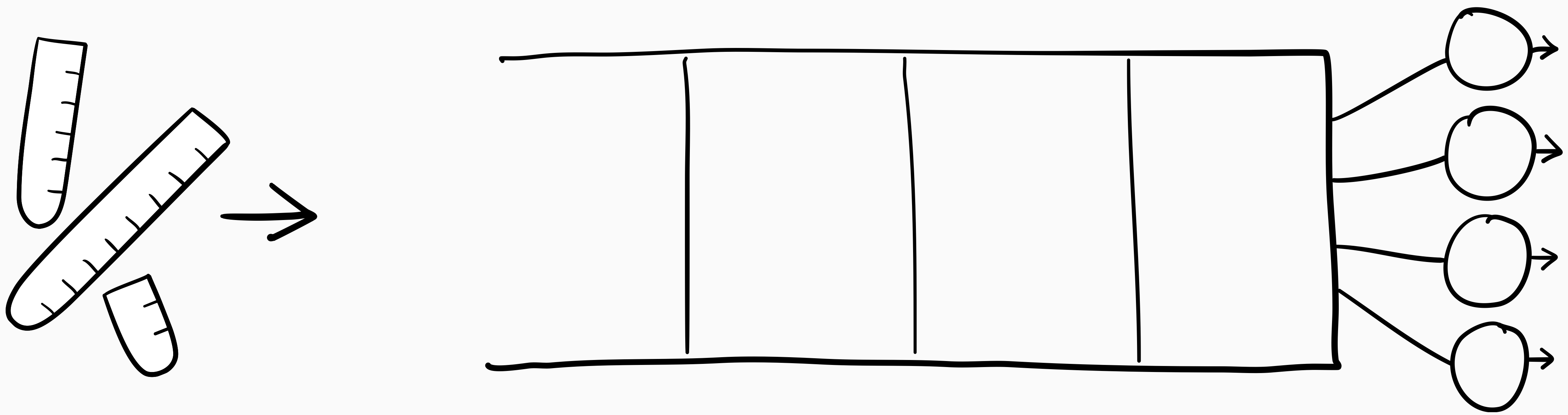
A New Queueing Identity *for* Analyzing Scheduling Policies *in* Multiserver Systems

Ziv Scully
CMU (now) → UC Berkeley → MIT/Harvard → Cornell (Fall 2023)

SRPT minimizes mean response time in single-server systems

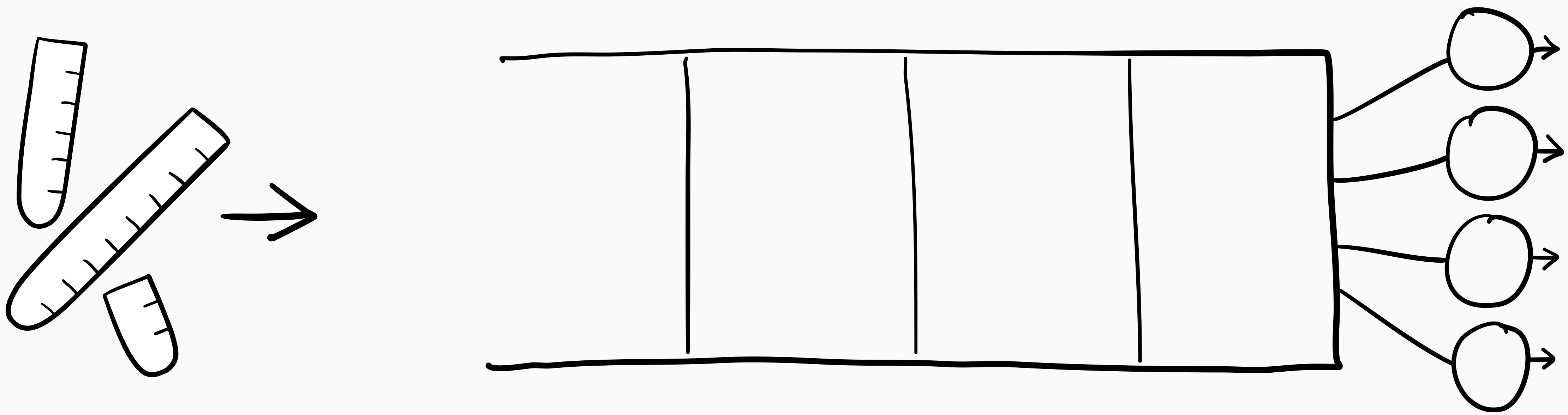
How good is SRPT in multiserver systems?

Multiserver queueing system



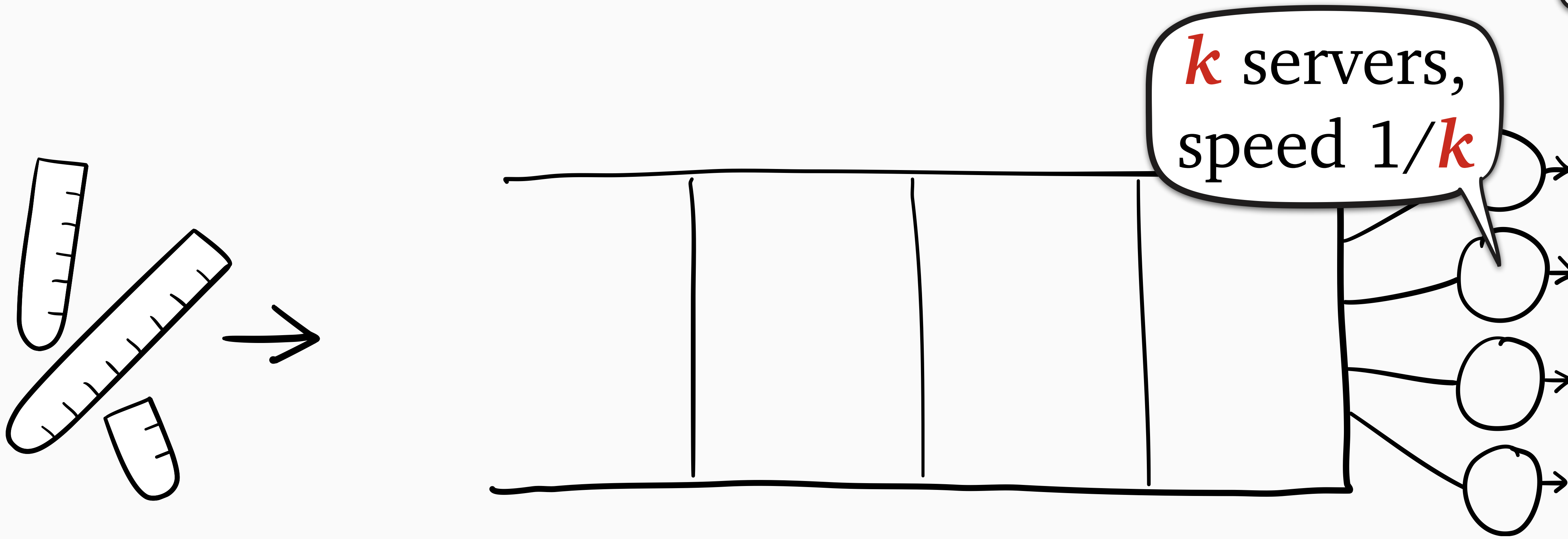
Multiserver queueing system

M/G/k

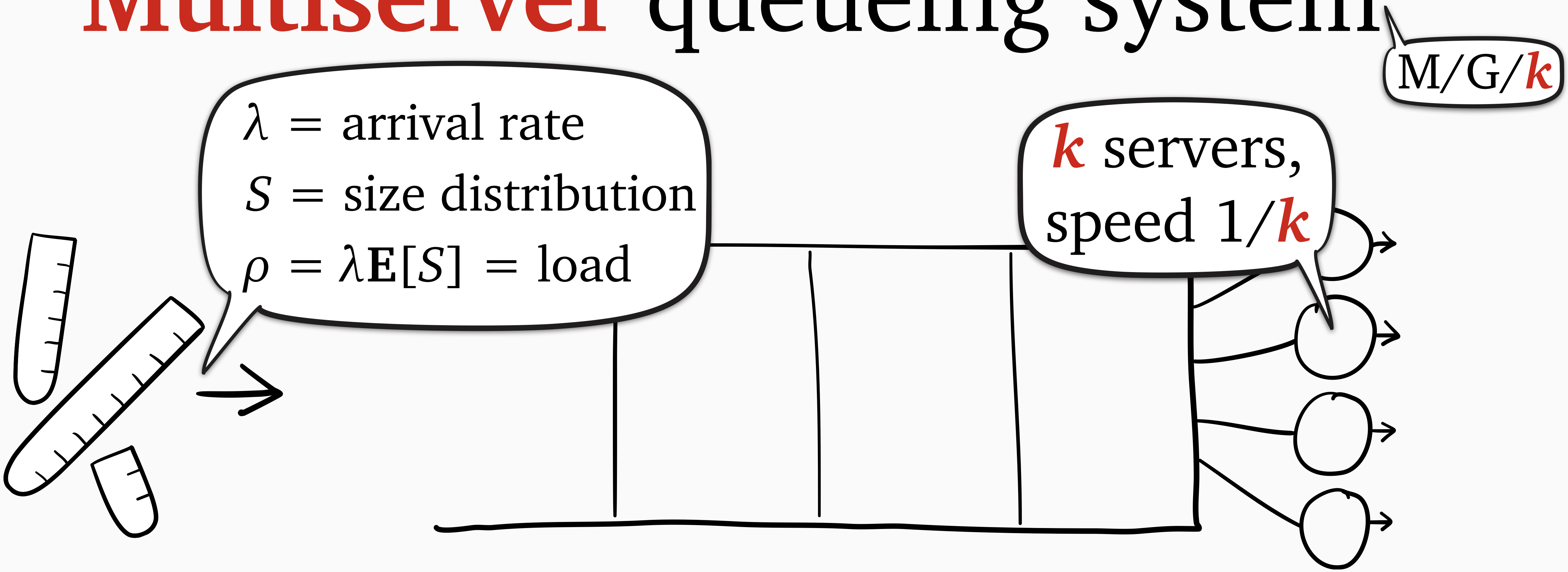


Multiserver queueing system

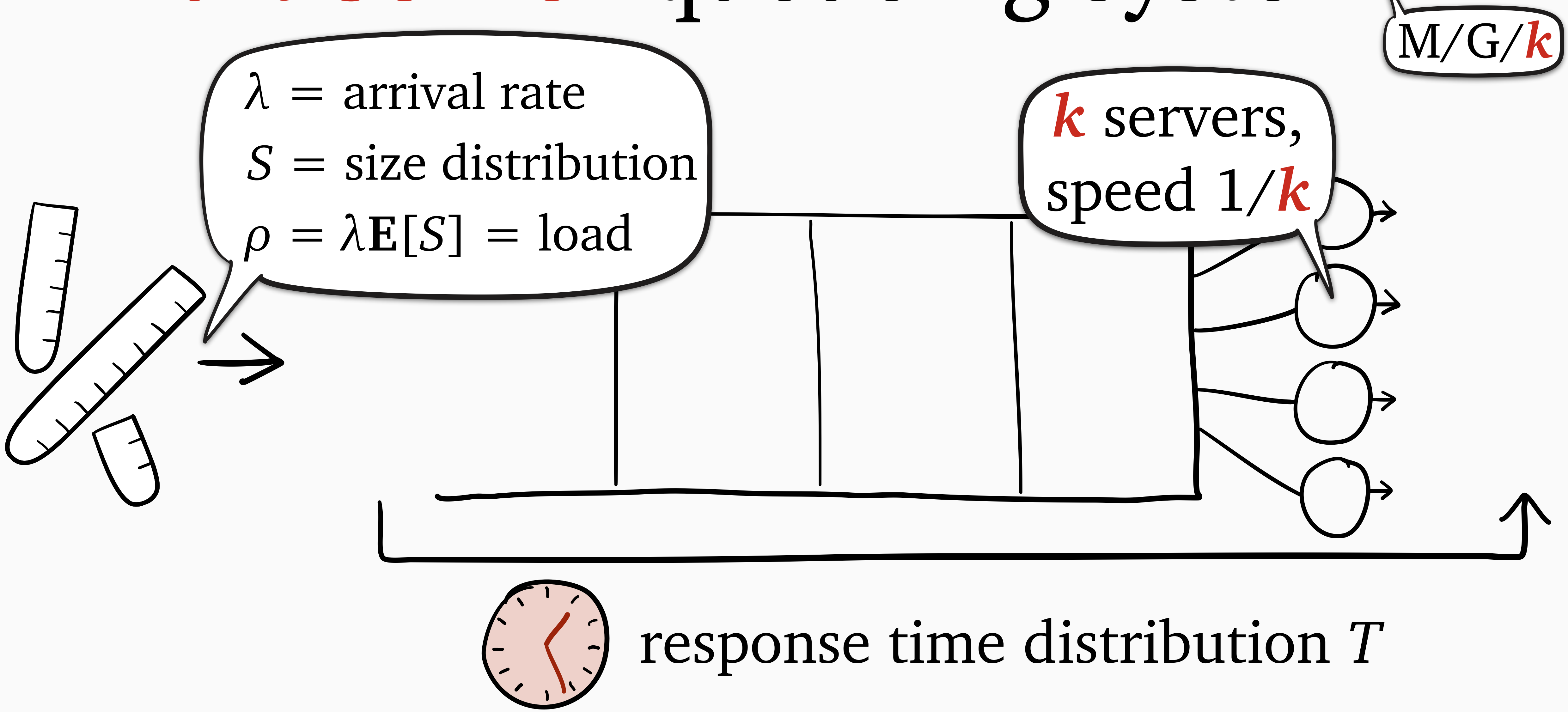
M/G/*k*



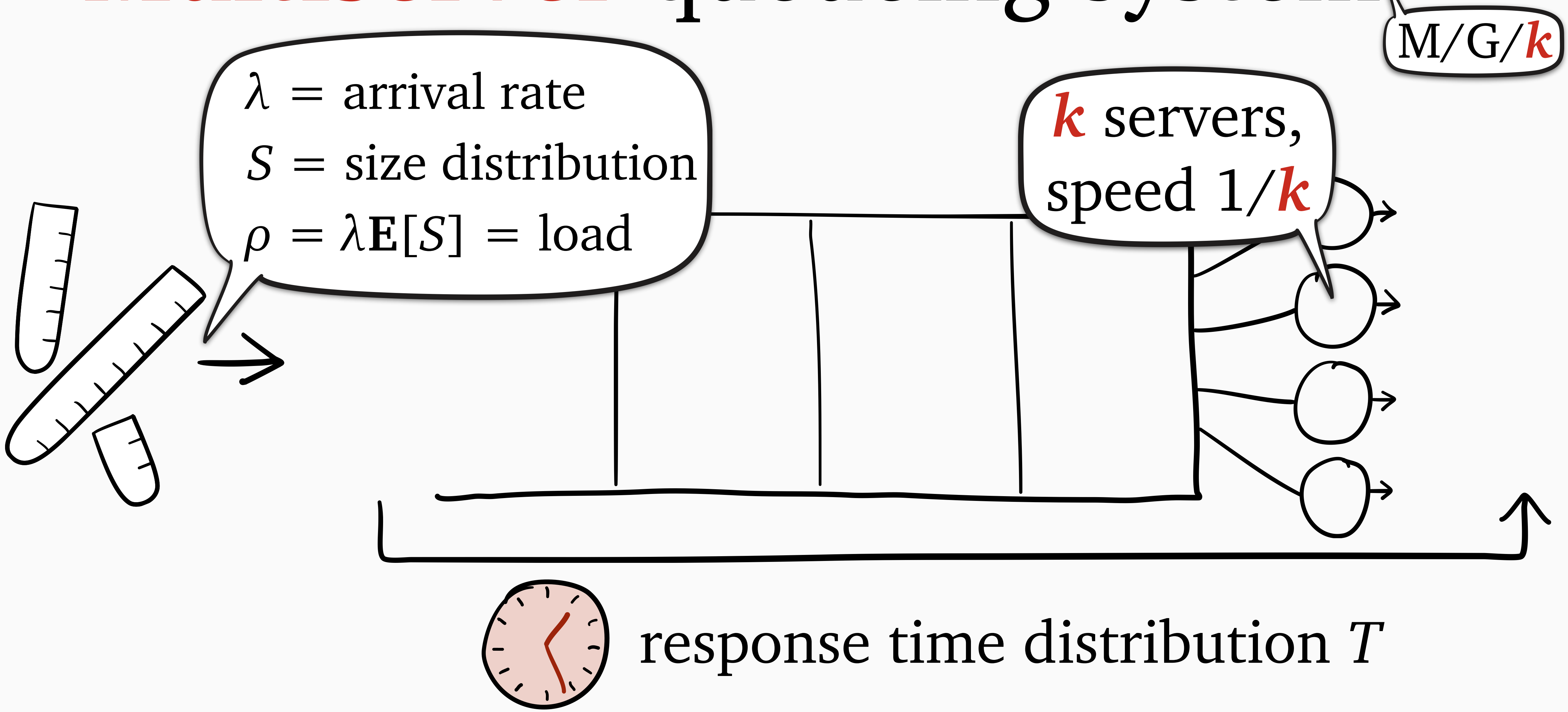
Multiserver queueing system



Multiserver queueing system

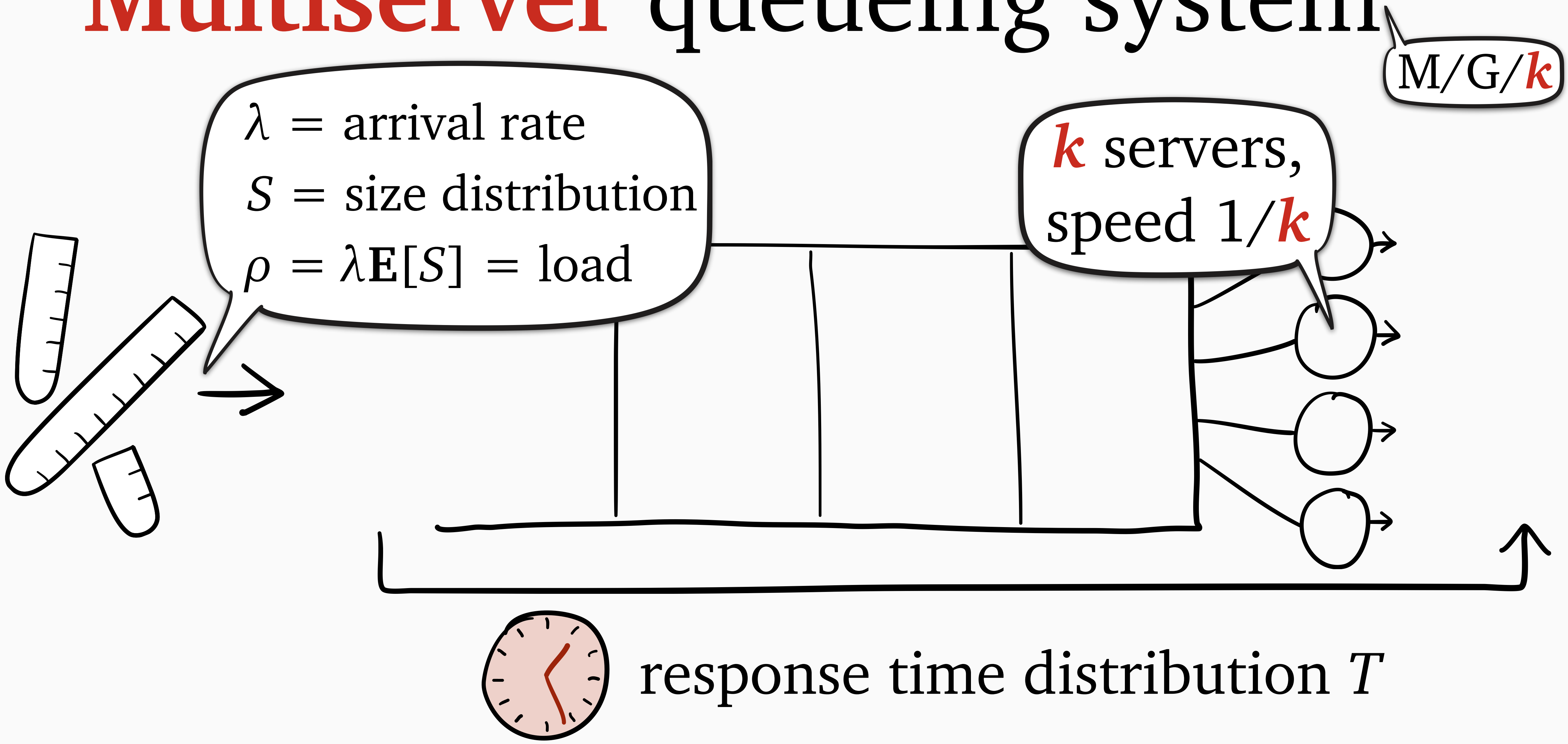


Multiserver queueing system



SRPT: rank = remaining size

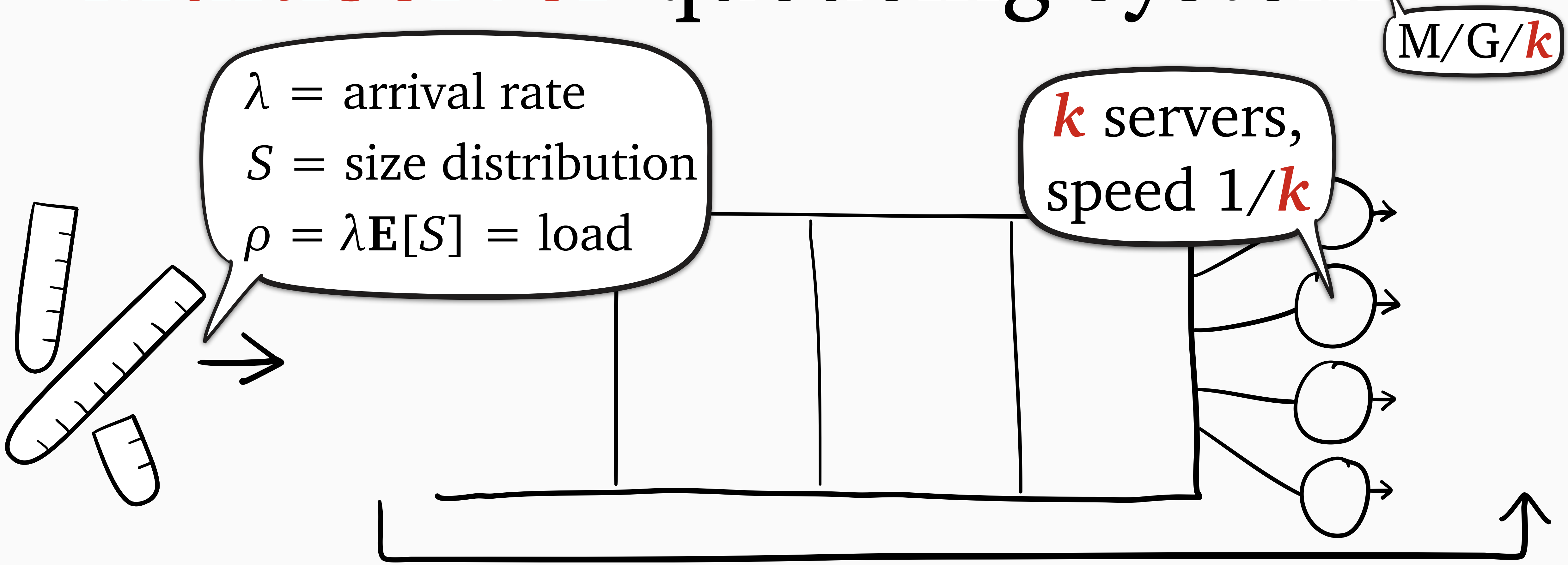
Multiserver queueing system

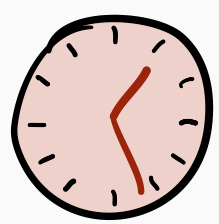


rank = priority,
lower is better

SRPT: rank = remaining size

Multiserver queueing system





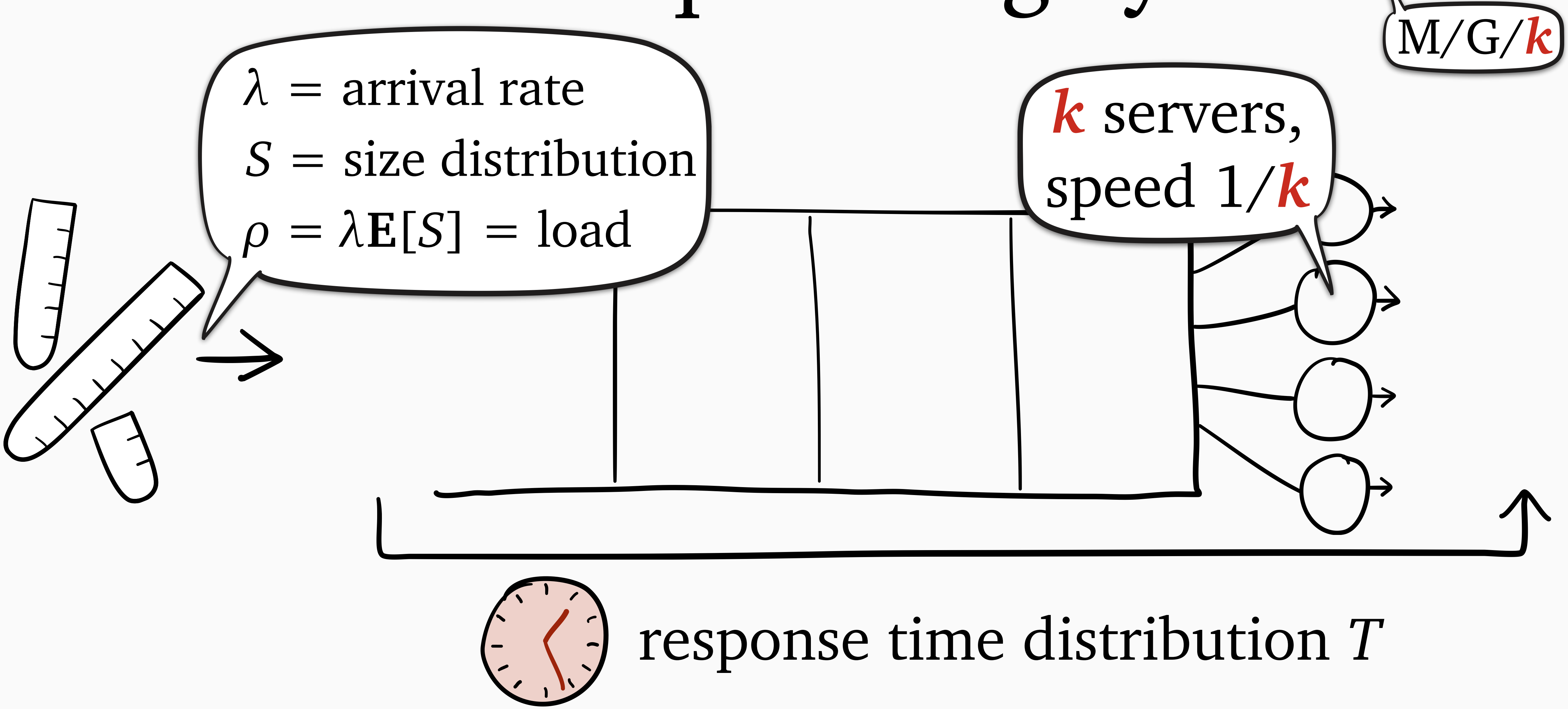
response time distribution T

rank = priority,
lower is better

SRPT: rank = remaining size

SRPT-1 (single-server): serves the job of least rank

Multiserver queueing system

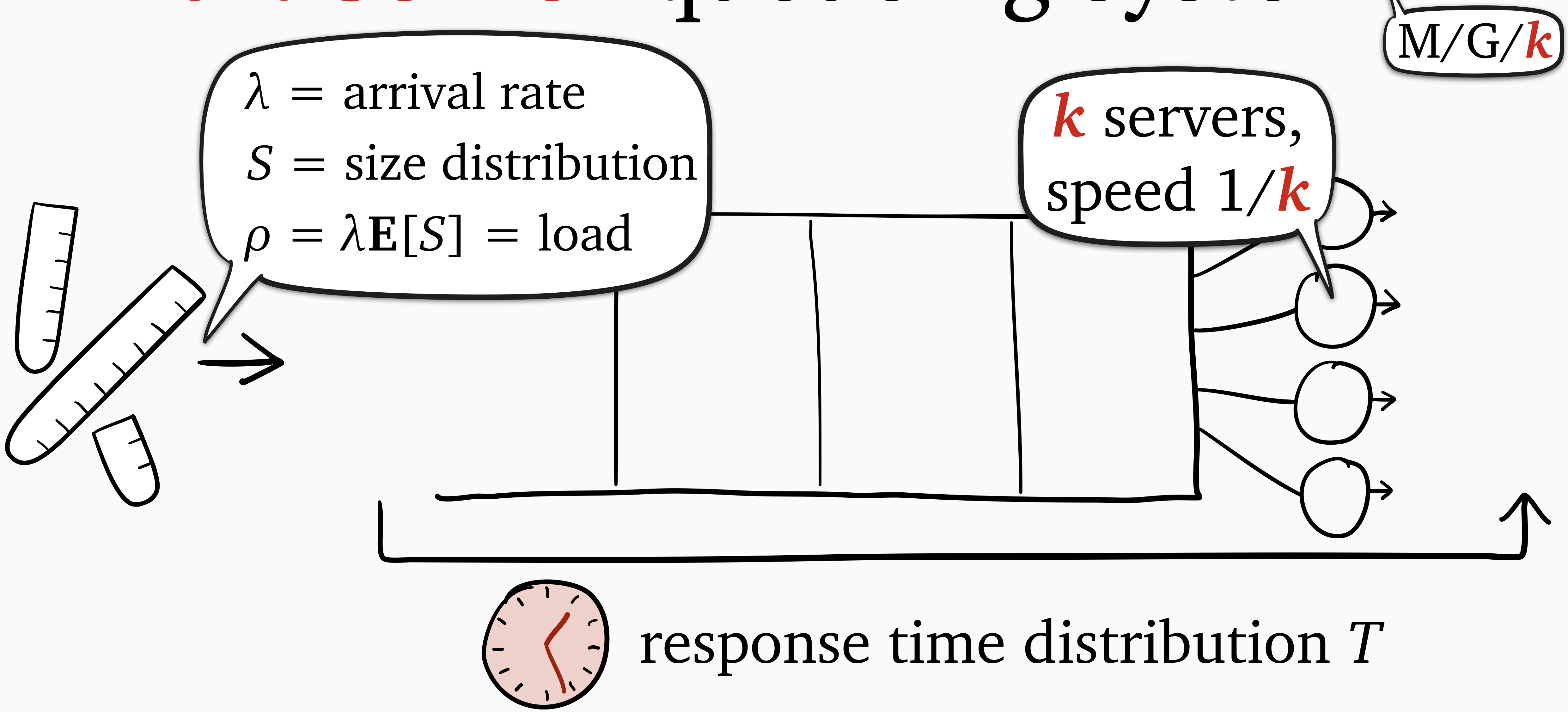


SRPT-k (multiserver): serves the k jobs of least **rank**

SRPT: **rank** = remaining size

SRPT-1 (single-server): serves the job of least **rank**

Multiserver queueing system



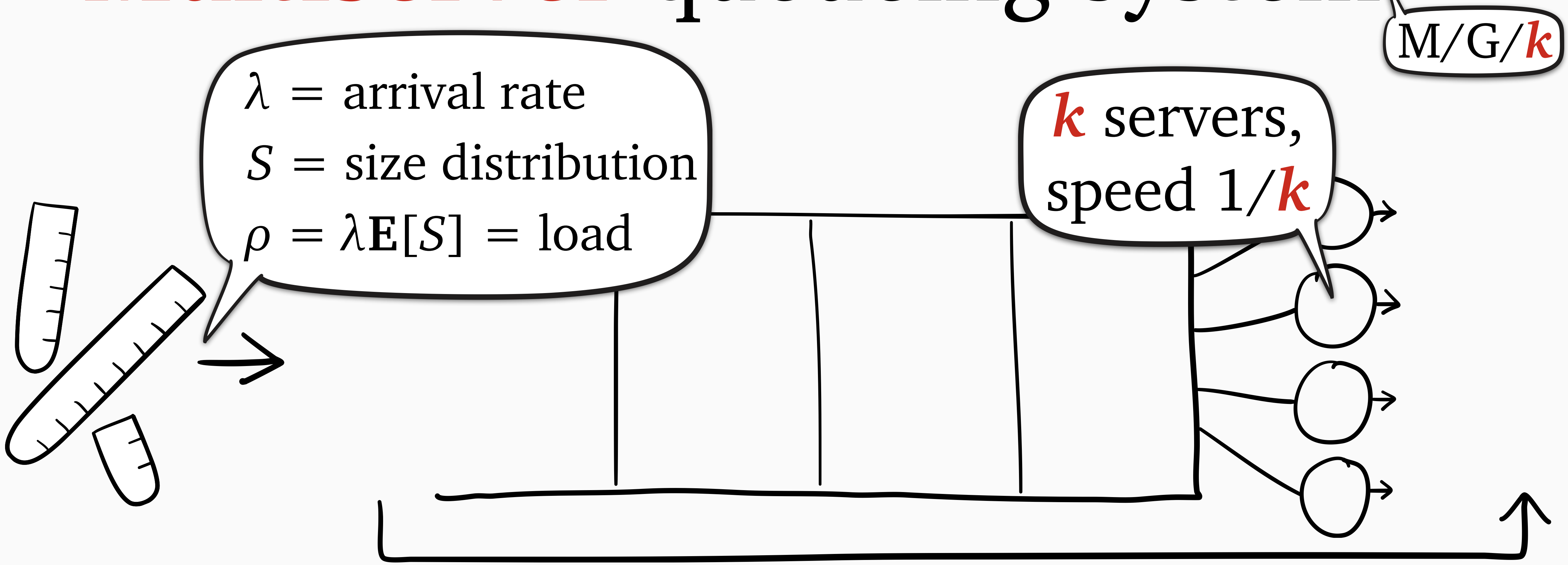
SRPT-k (multiserver): serves the k jobs of least **rank**

SRPT: **rank** = remaining size

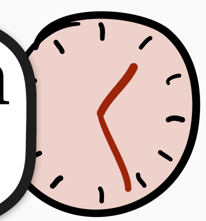
$\mathbf{E}[T]$ known

SRPT-1 (single-server): serves the job of least **rank**

Multiserver queueing system



$\mathbf{E}[T]$ unknown (pre-2018)



response time distribution T

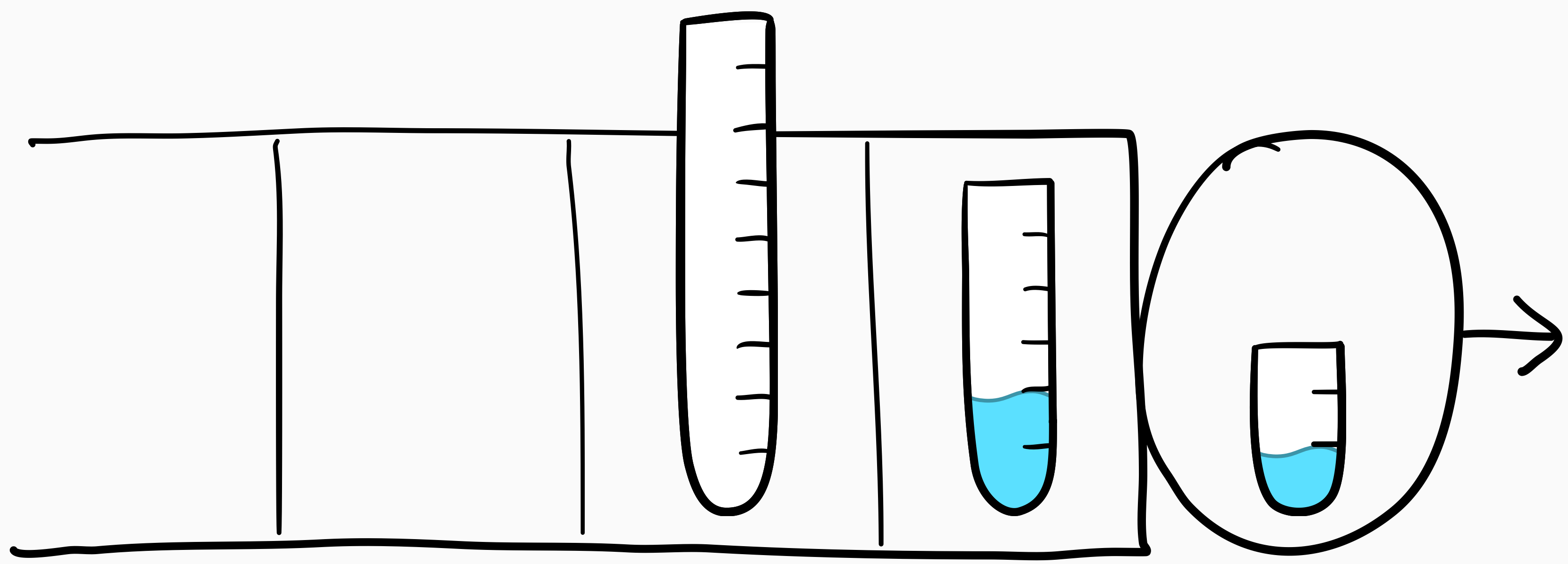
SRPT-*k* (**multiserver**): serves the k jobs of least **rank**

SRPT: **rank** = remaining size

$\mathbf{E}[T]$ known

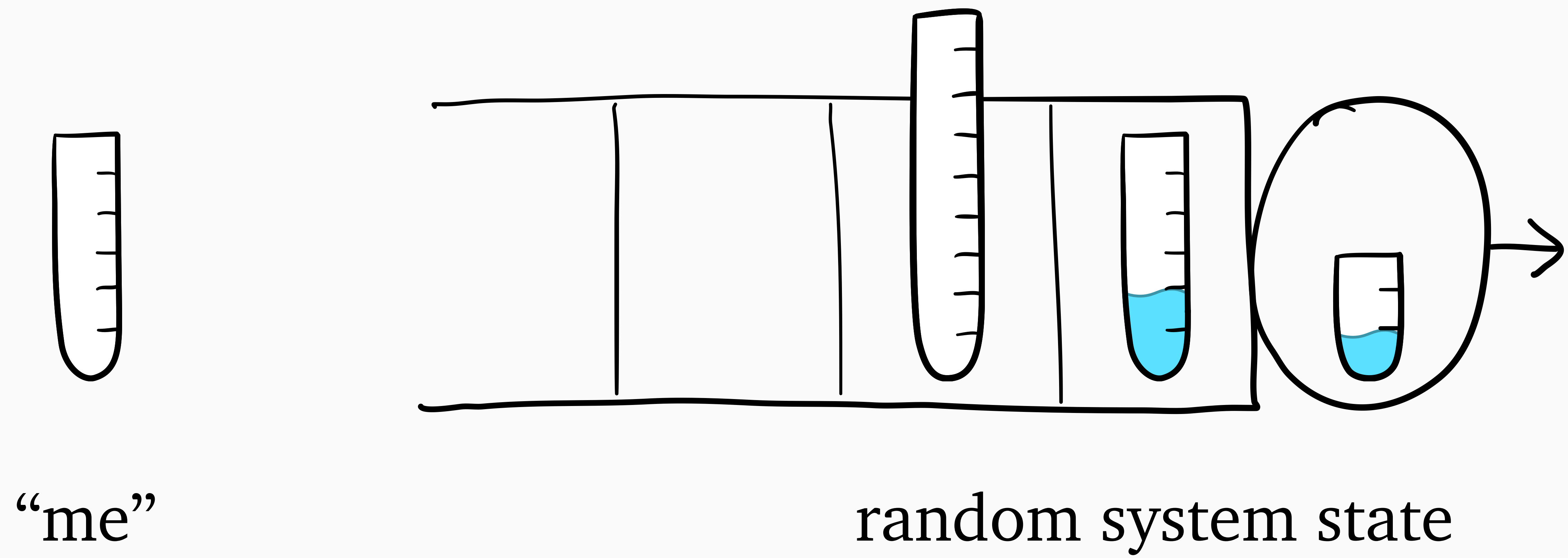
SRPT-1 (**single-server**): serves the job of least **rank**

SRPT-1: tagged job analysis

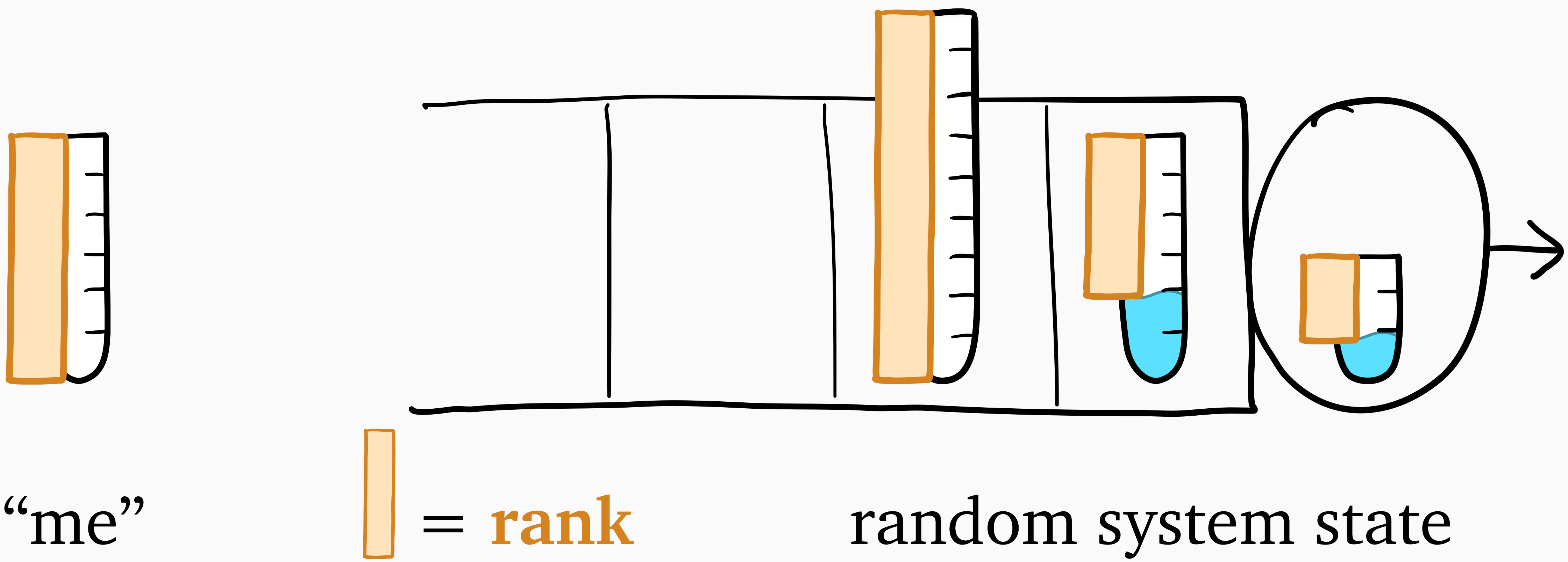


random system state

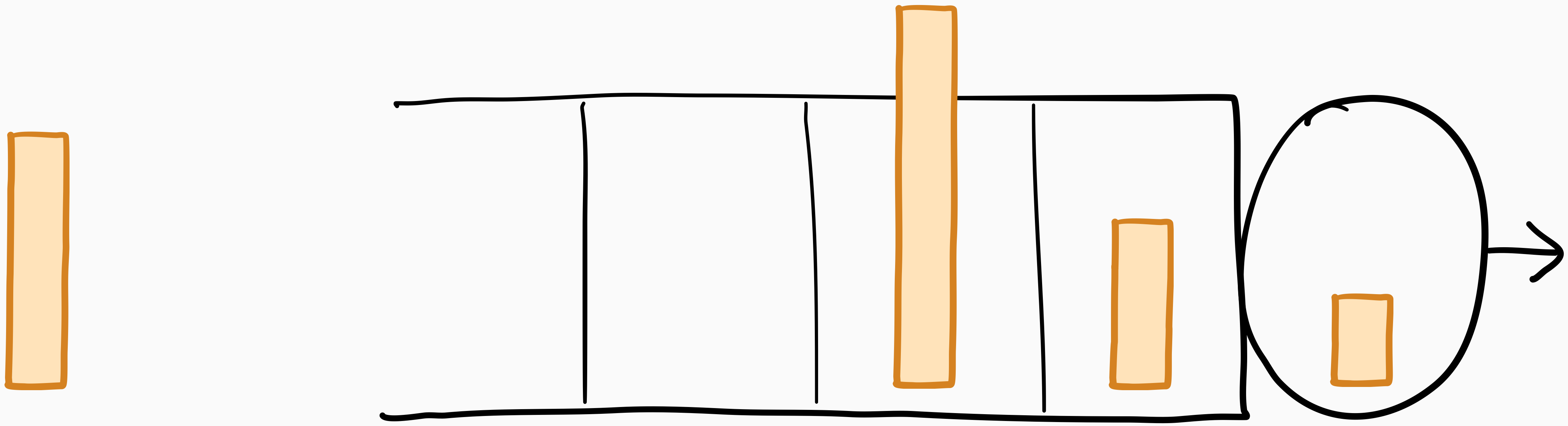
SRPT-1: tagged job analysis

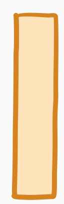


SRPT-1: tagged job analysis

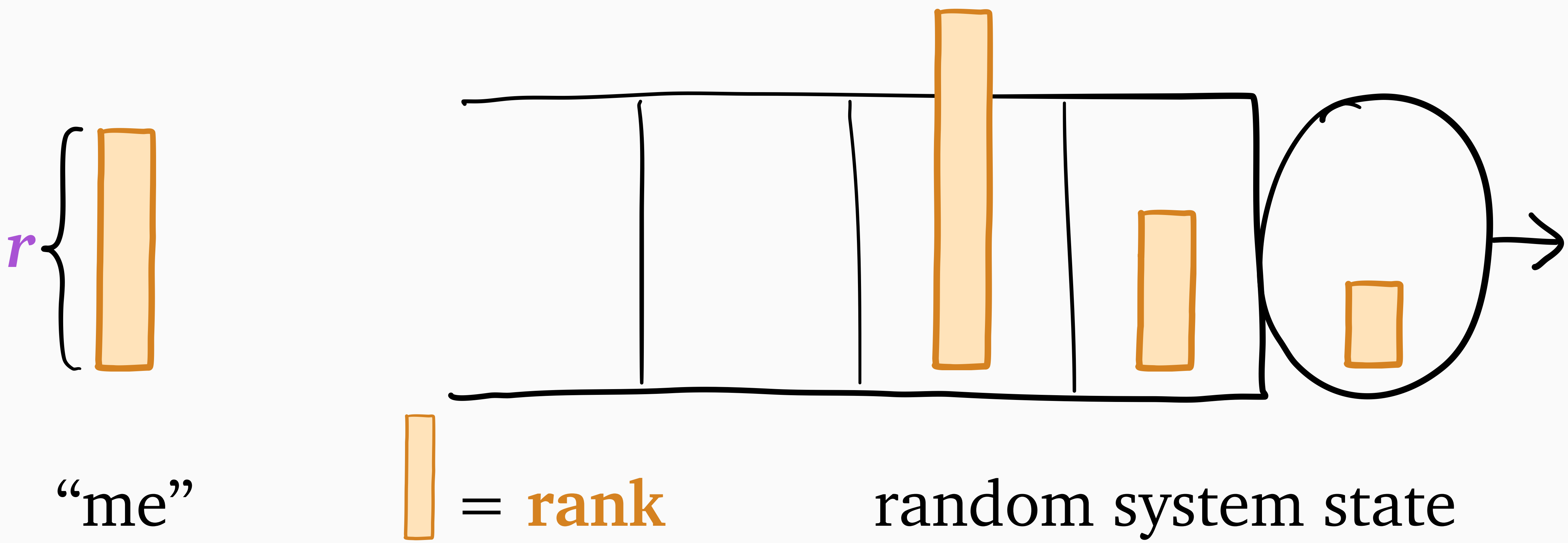


SRPT-1: tagged job analysis



"me"  = **rank** random system state

SRPT-1: tagged job analysis

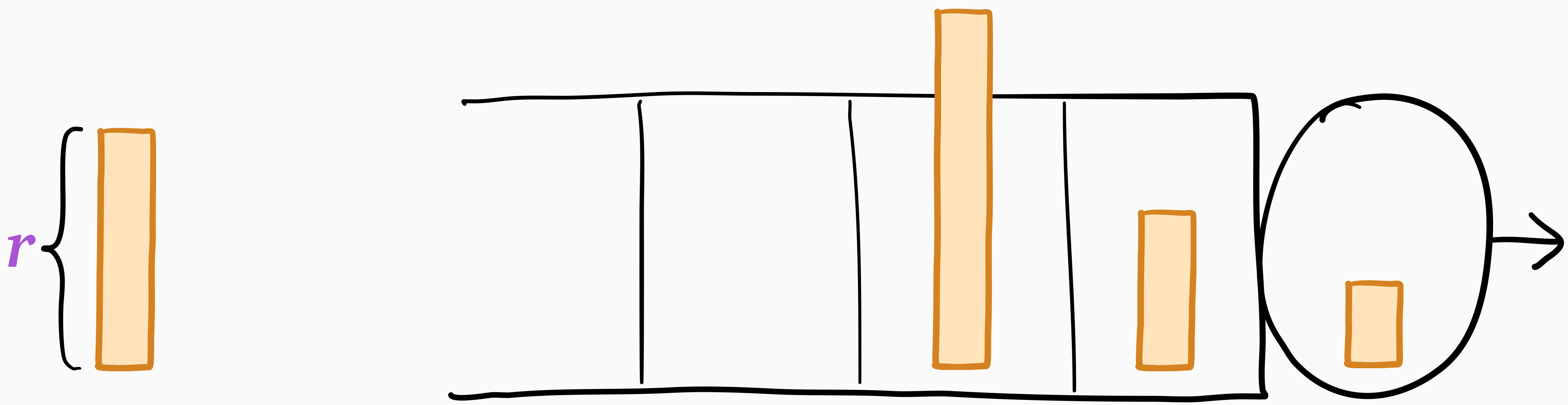


Key quantity:

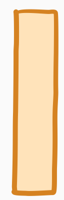
$W(r)$ = work relevant to job of **rank** r

r-work

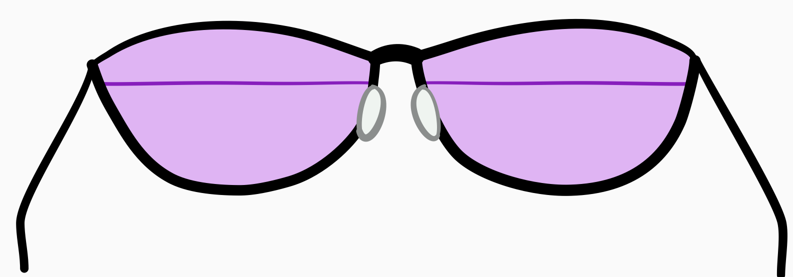
SRPT-1: tagged job analysis



"me"

 = **rank**

random system state

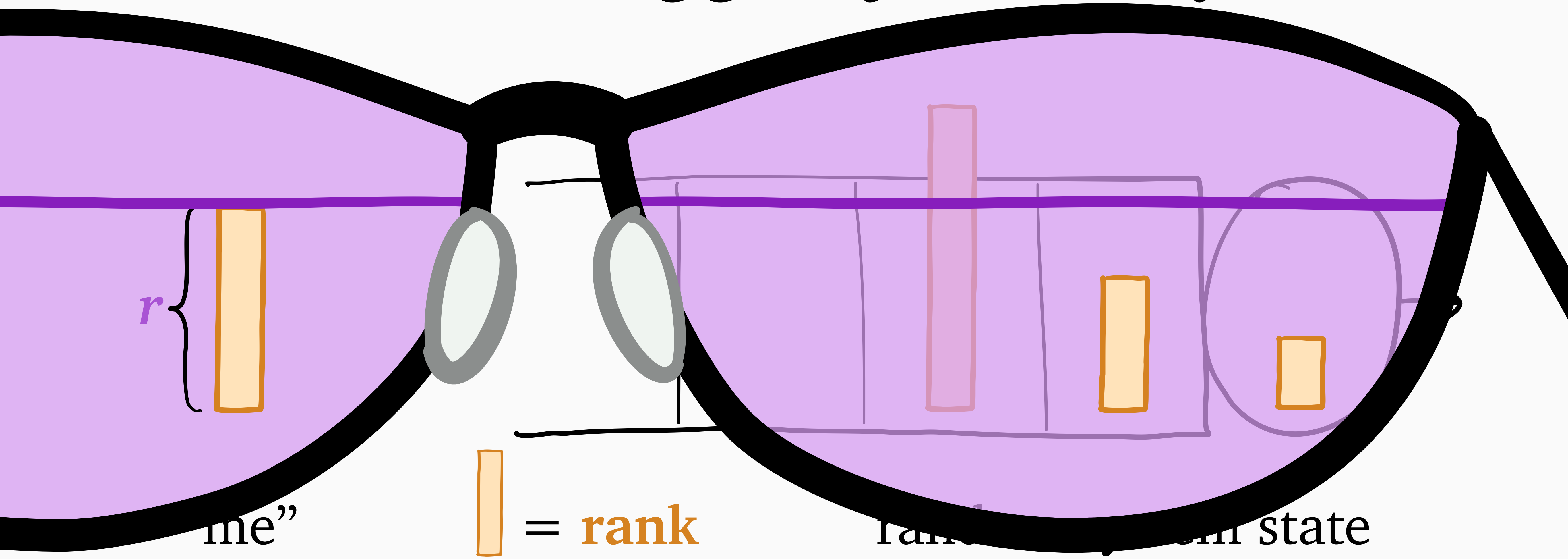


Key quantity:

$W(r)$ = work relevant to job of **rank** r

r-work

SRPT-1: tagged job analysis

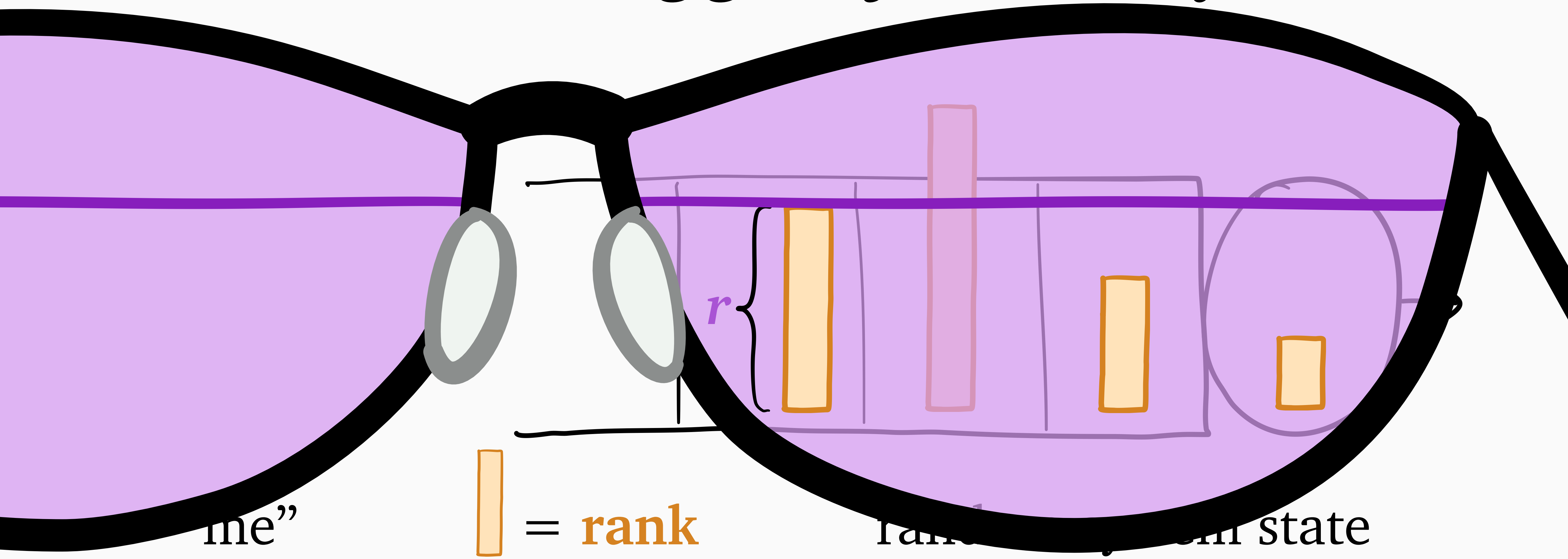


Key quantity:

$W(r)$ = work relevant to job of **rank** r

r-work

SRPT-1: tagged job analysis

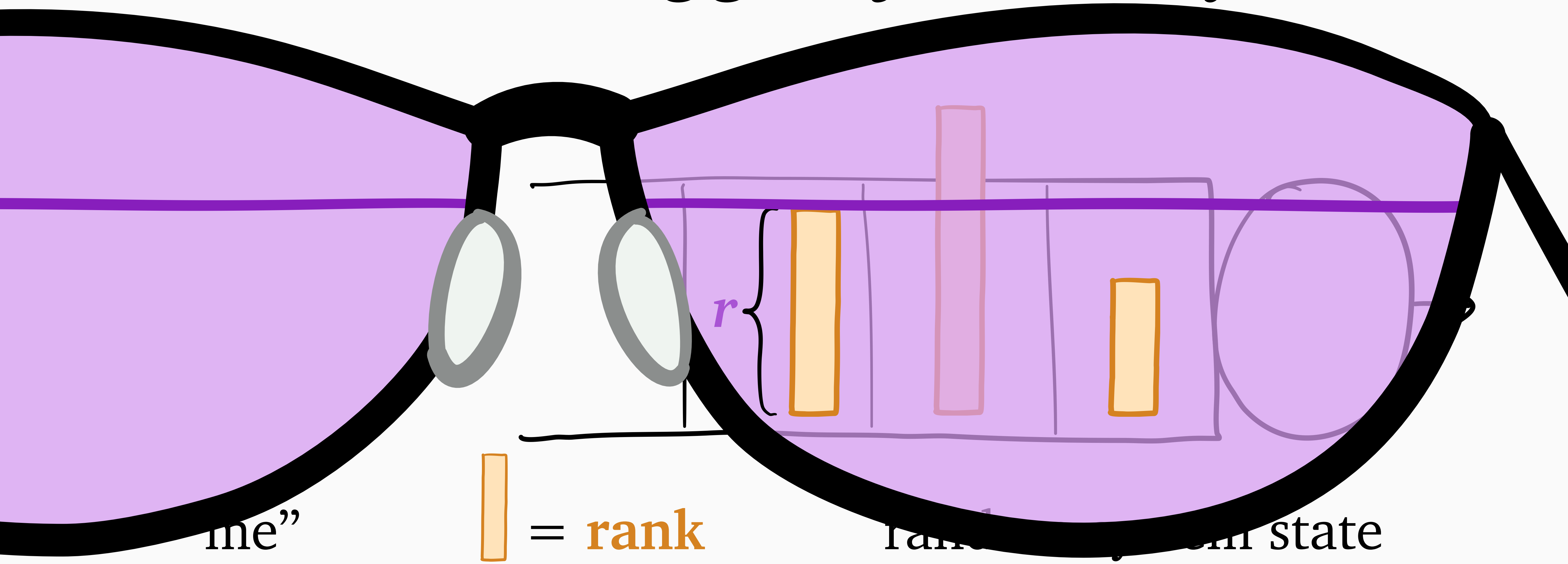


Key quantity:

$W(r)$ = work relevant to job of **rank** r

r-work

SRPT-1: tagged job analysis

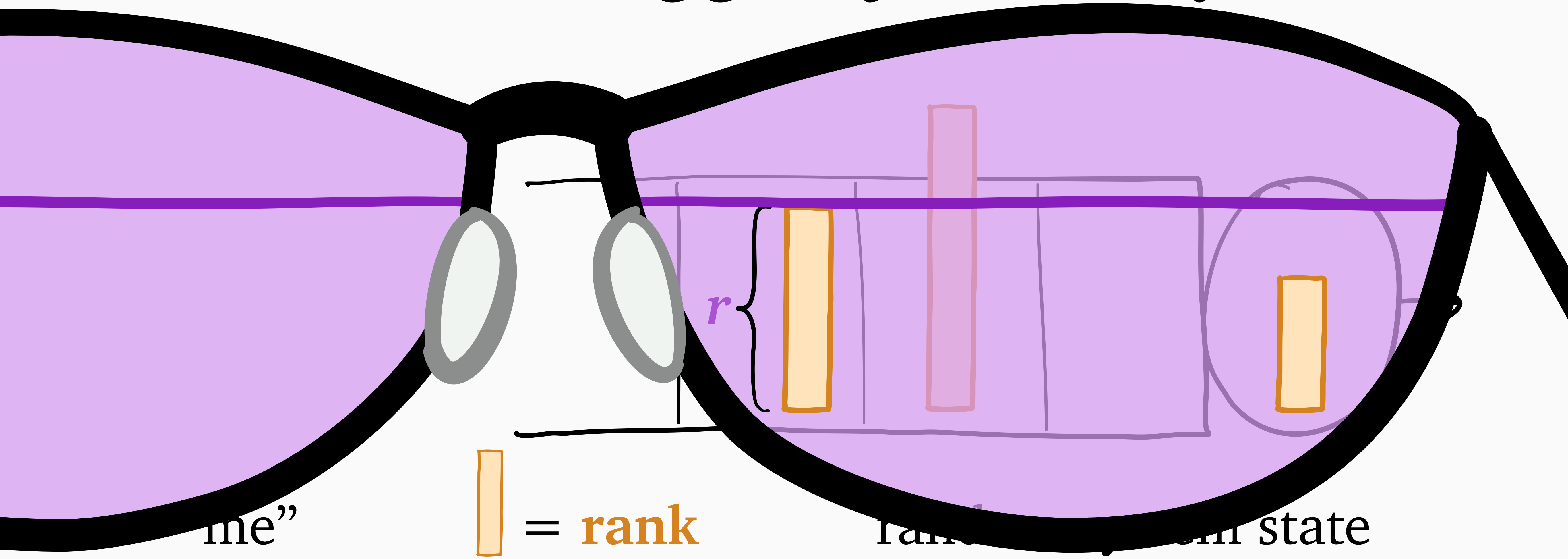


Key quantity:

$W(r)$ = work relevant to job of **rank** r

r-work

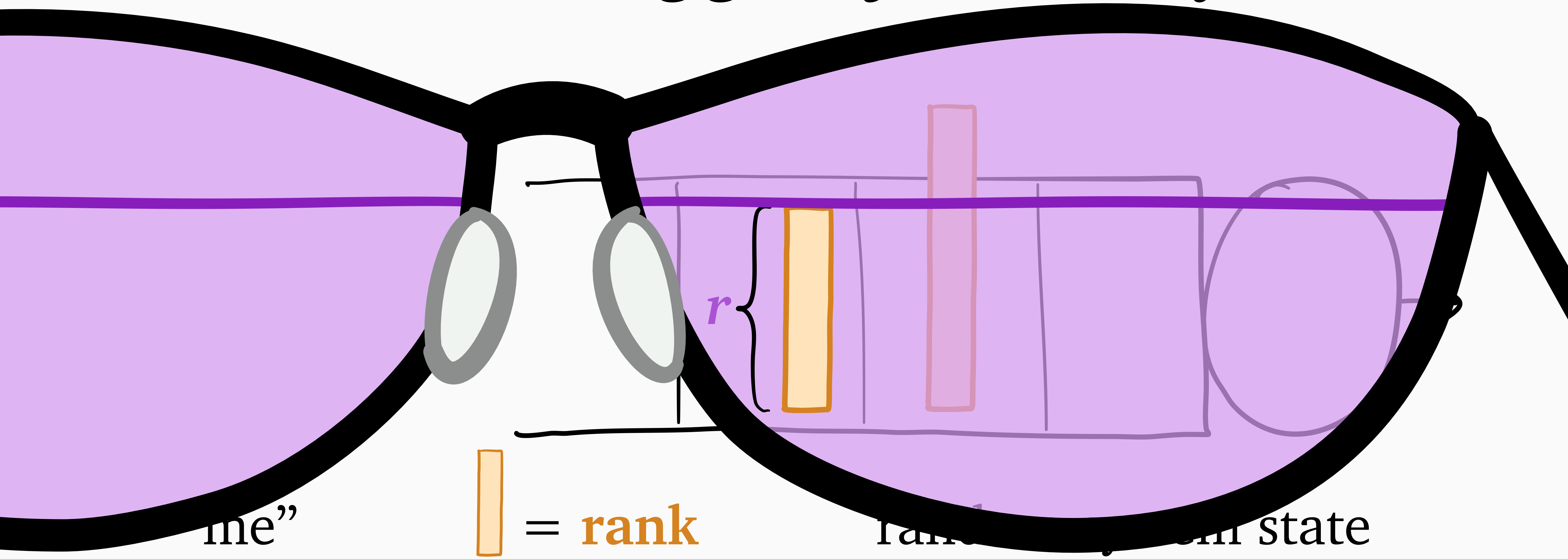
SRPT-1: tagged job analysis



Key quantity:

$W(r)$ = work relevant to job of **rank** r

SRPT-1: tagged job analysis

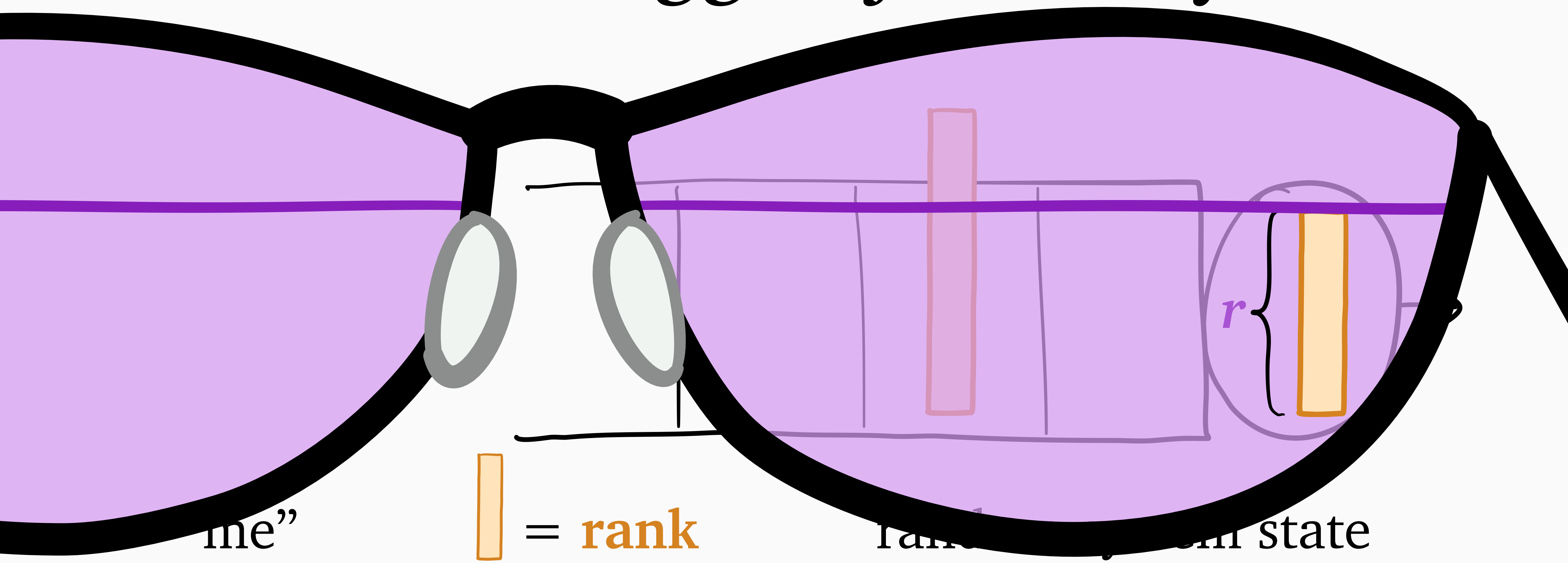


Key quantity:

$W(r)$ = work relevant to job of **rank** r

r-work

SRPT-1: tagged job analysis

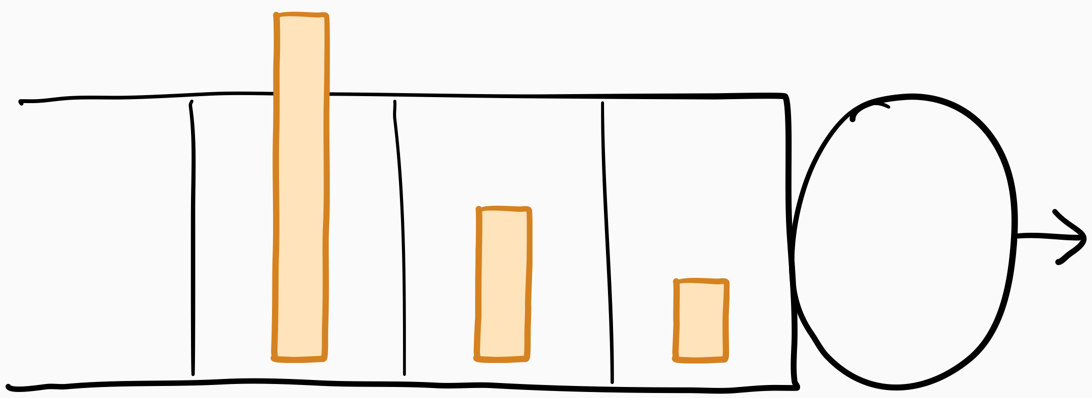


Key quantity:

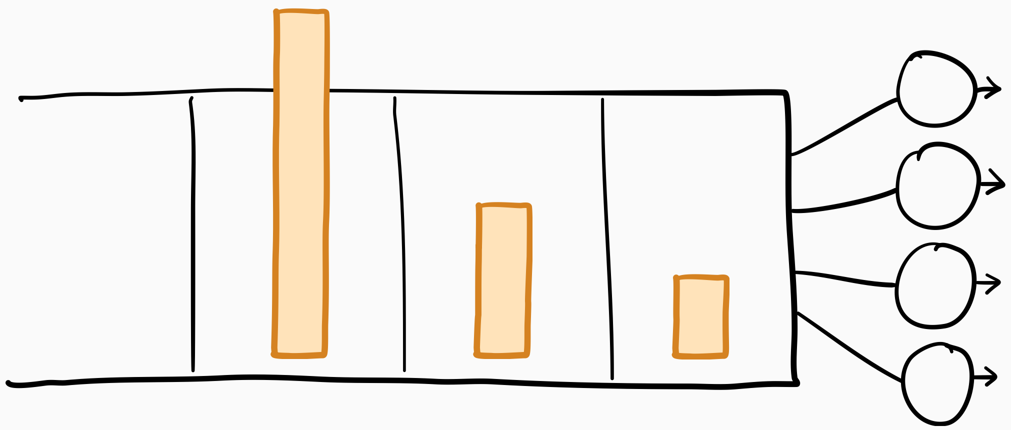
$W(r)$ = work relevant to job of **rank** r

r-work

Single-server system

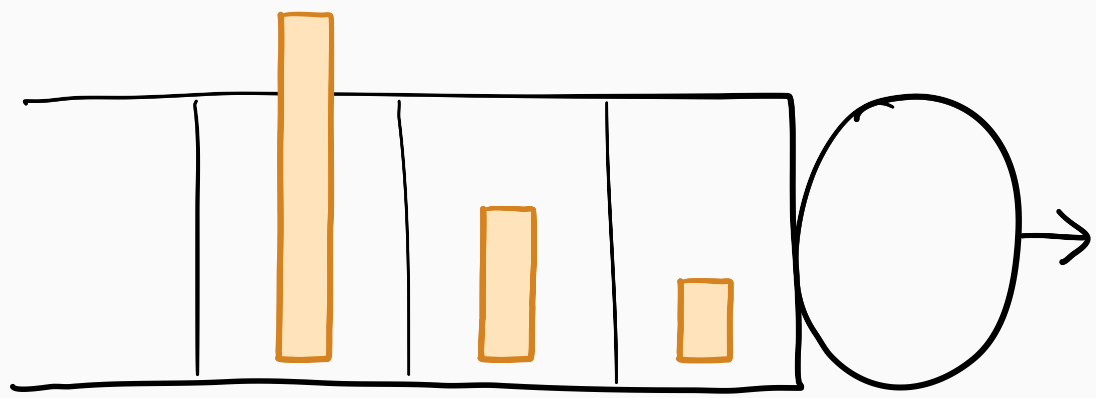


Multiserver system

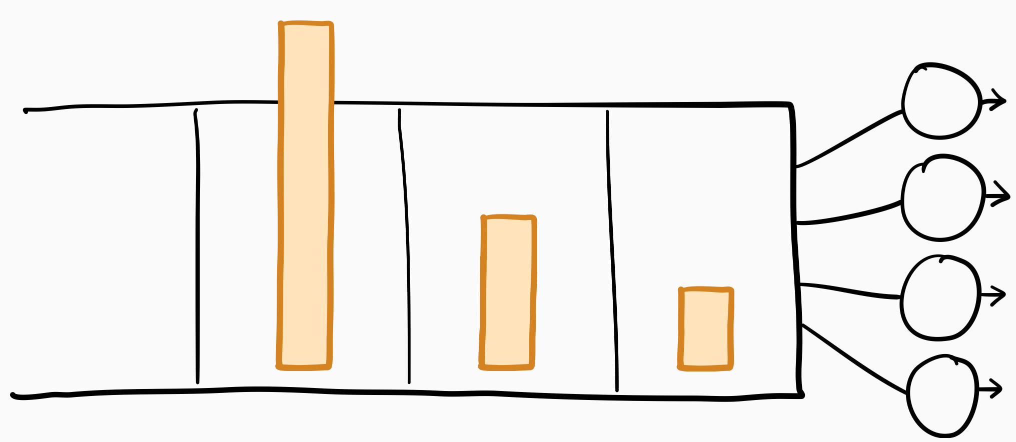


Single-server system

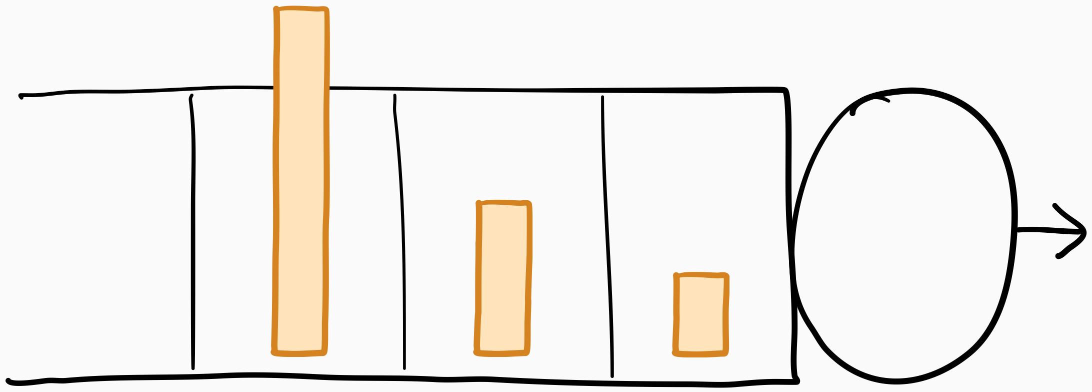
server is “choke point”



Multiserver system



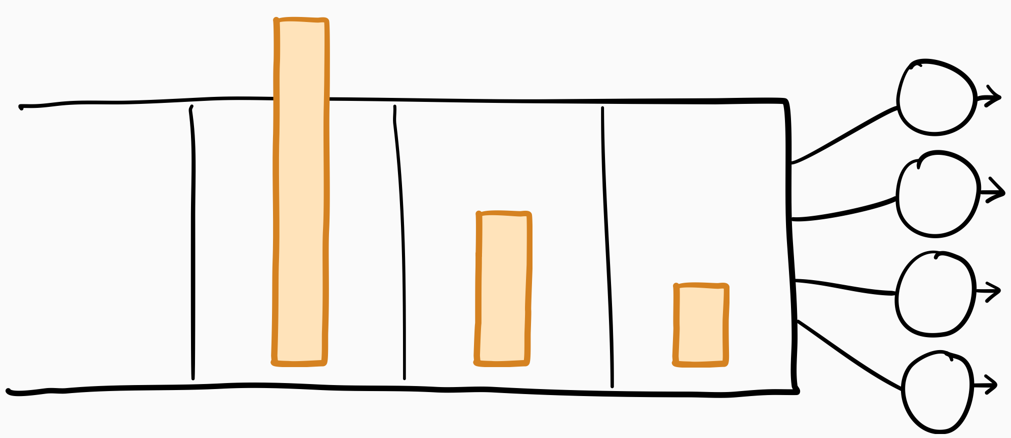
Single-server system



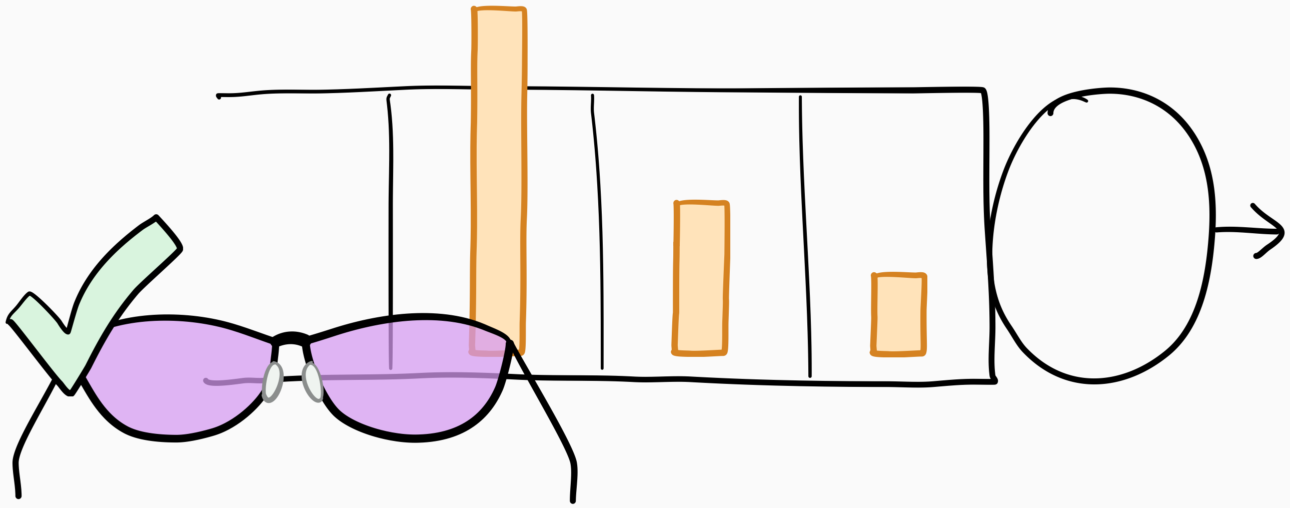
server is "choke point"

rank ordering absolute

Multiserver system



Single-server system



server is “choke point”

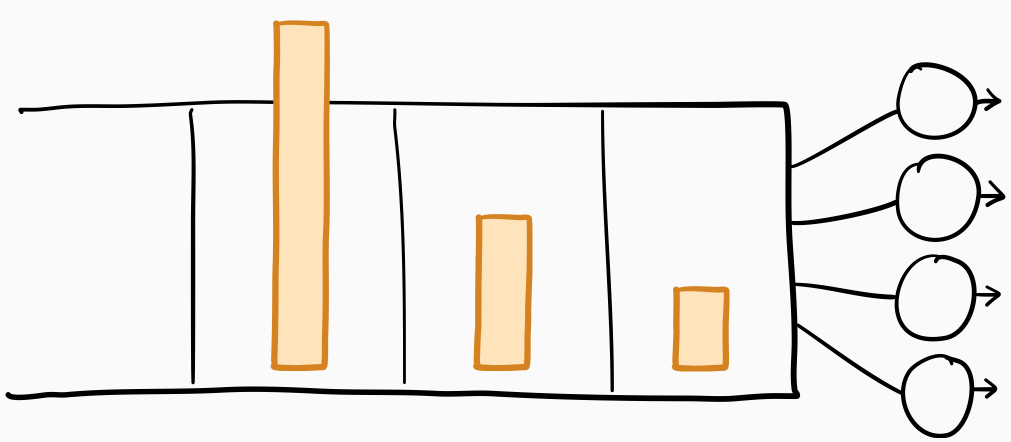
↓

rank ordering absolute

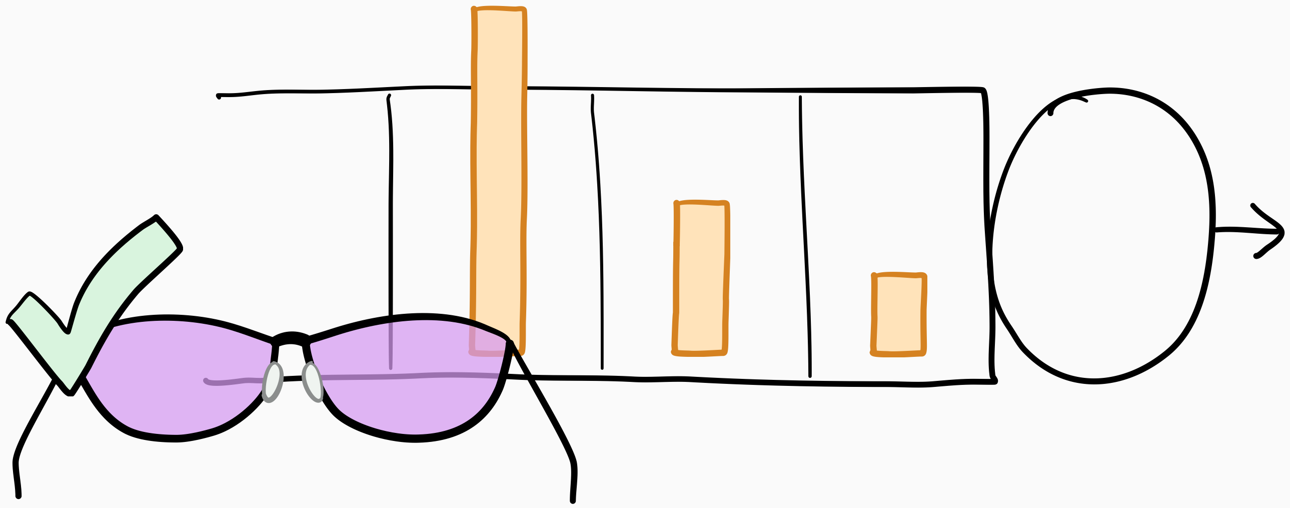
↓

observed r-work determines T

Multiserver system



Single-server system



server is "choke point"

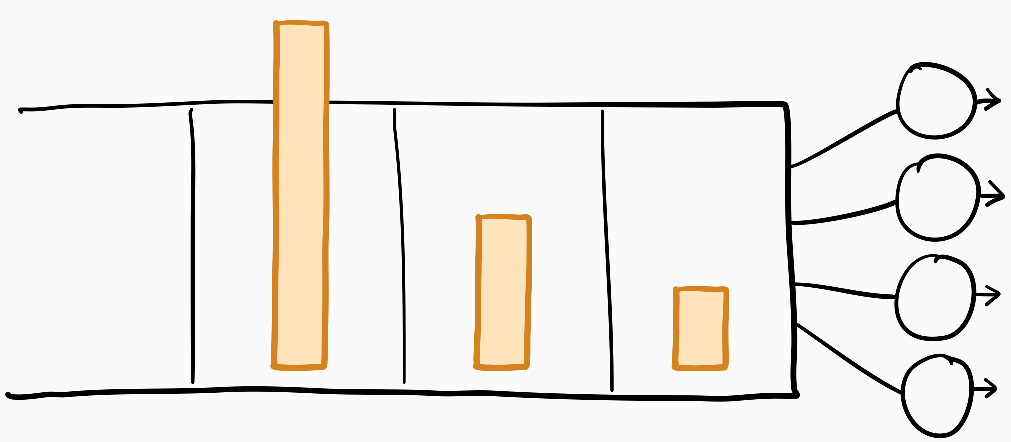
↓

rank ordering absolute

↓

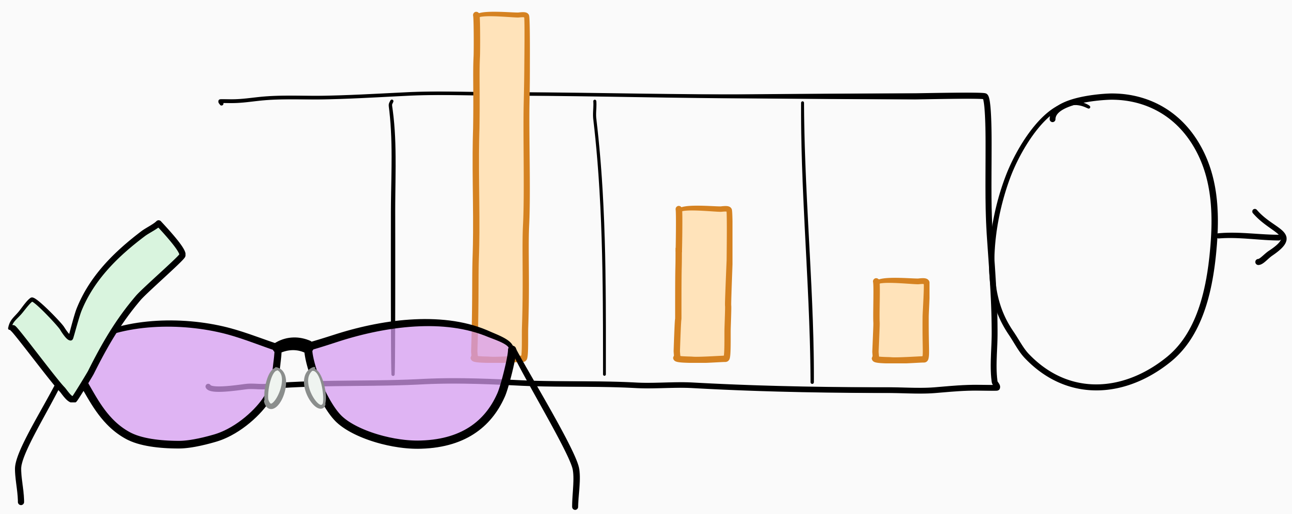
observed r-work determines T

Multiserver system



no single "choke point"

Single-server system



server is "choke point"

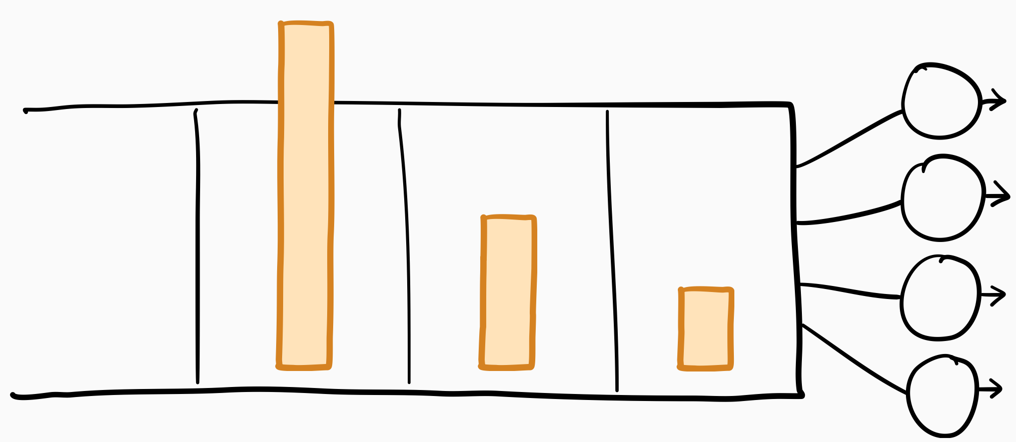
↓

rank ordering absolute

↓

observed *r*-work determines *T*

Multiserver system

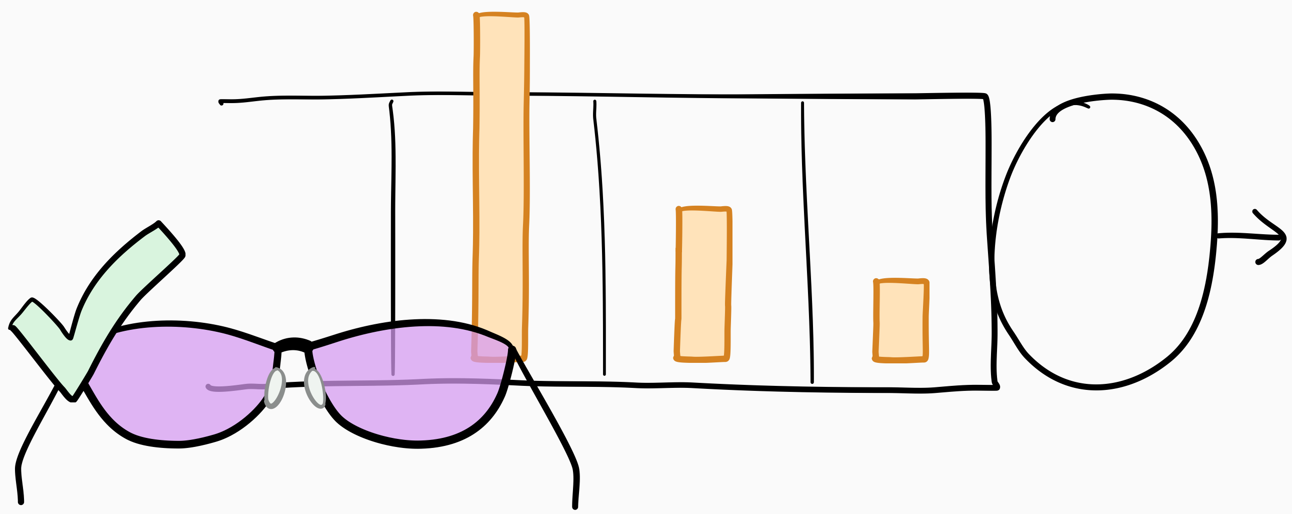


no single "choke point"

↓

rank ordering *not* absolute

Single-server system



server is "choke point"

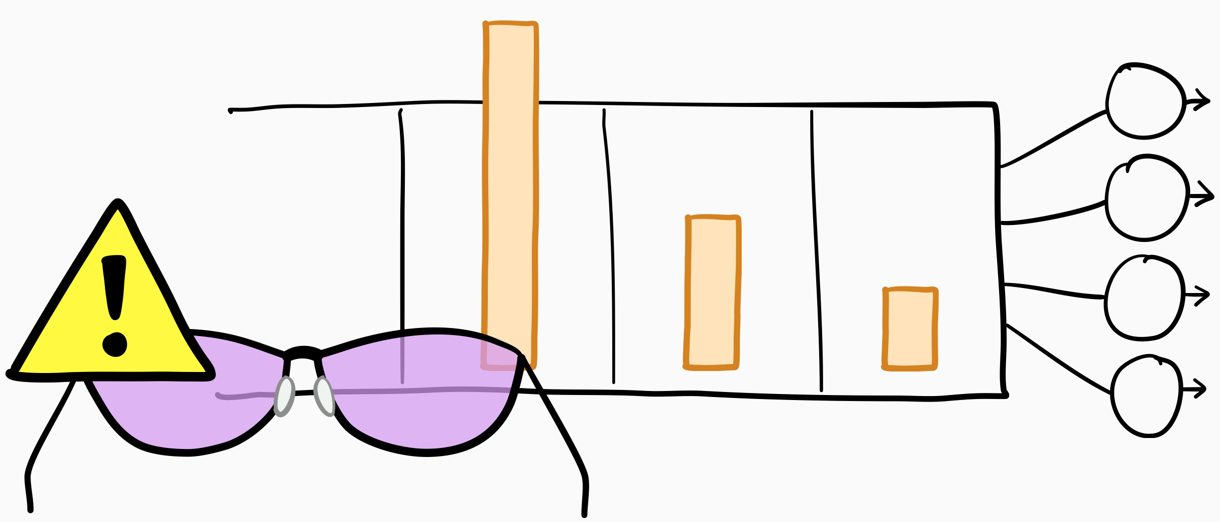
↓

rank ordering absolute

↓

observed r-work determines T

Multiserver system



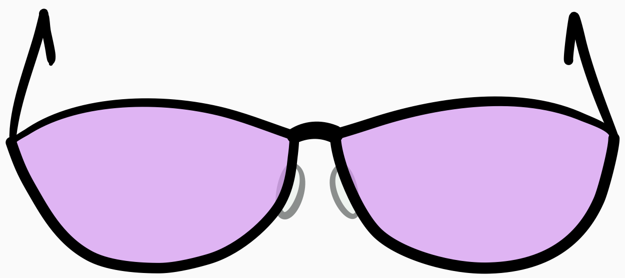
no single "choke point"

↓

rank ordering *not* absolute

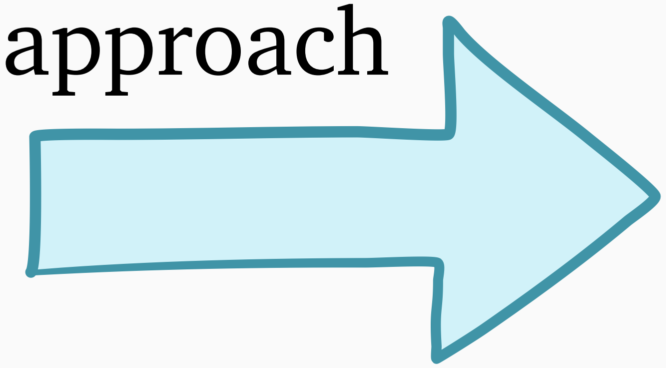
↓

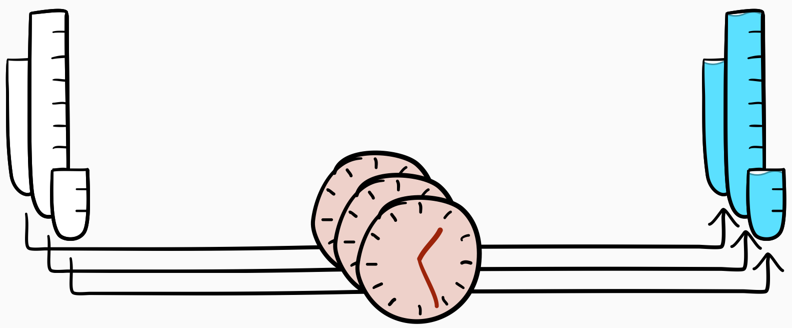
observed r-work *not enough!*



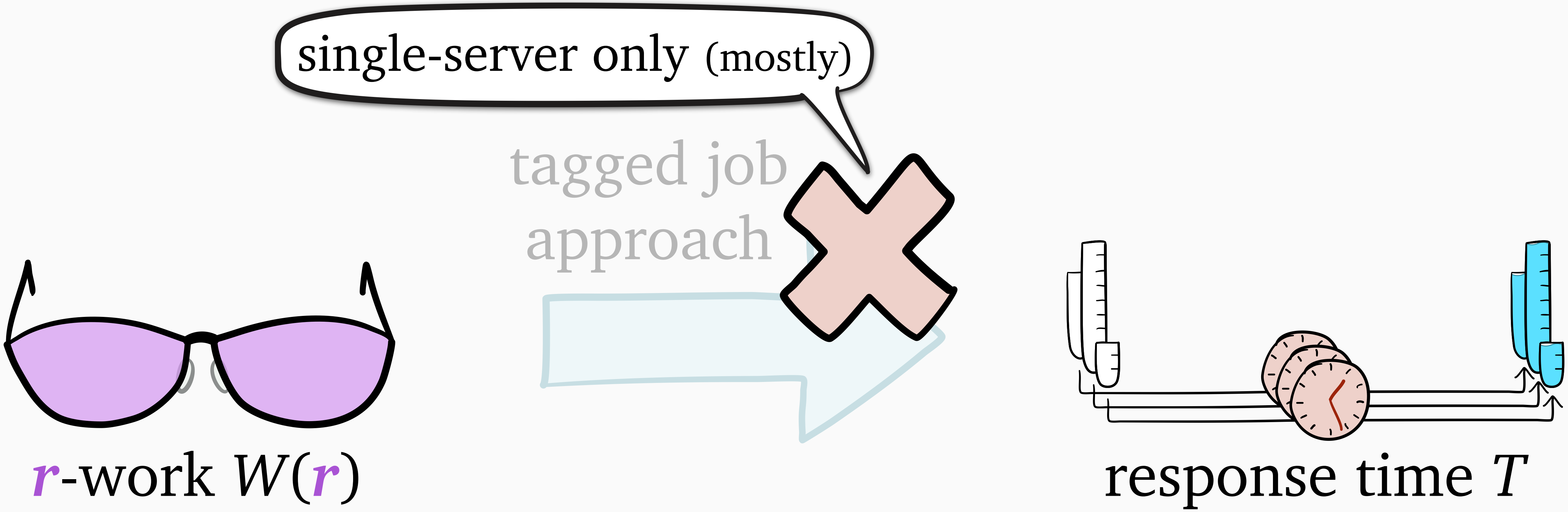
r-work $W(r)$

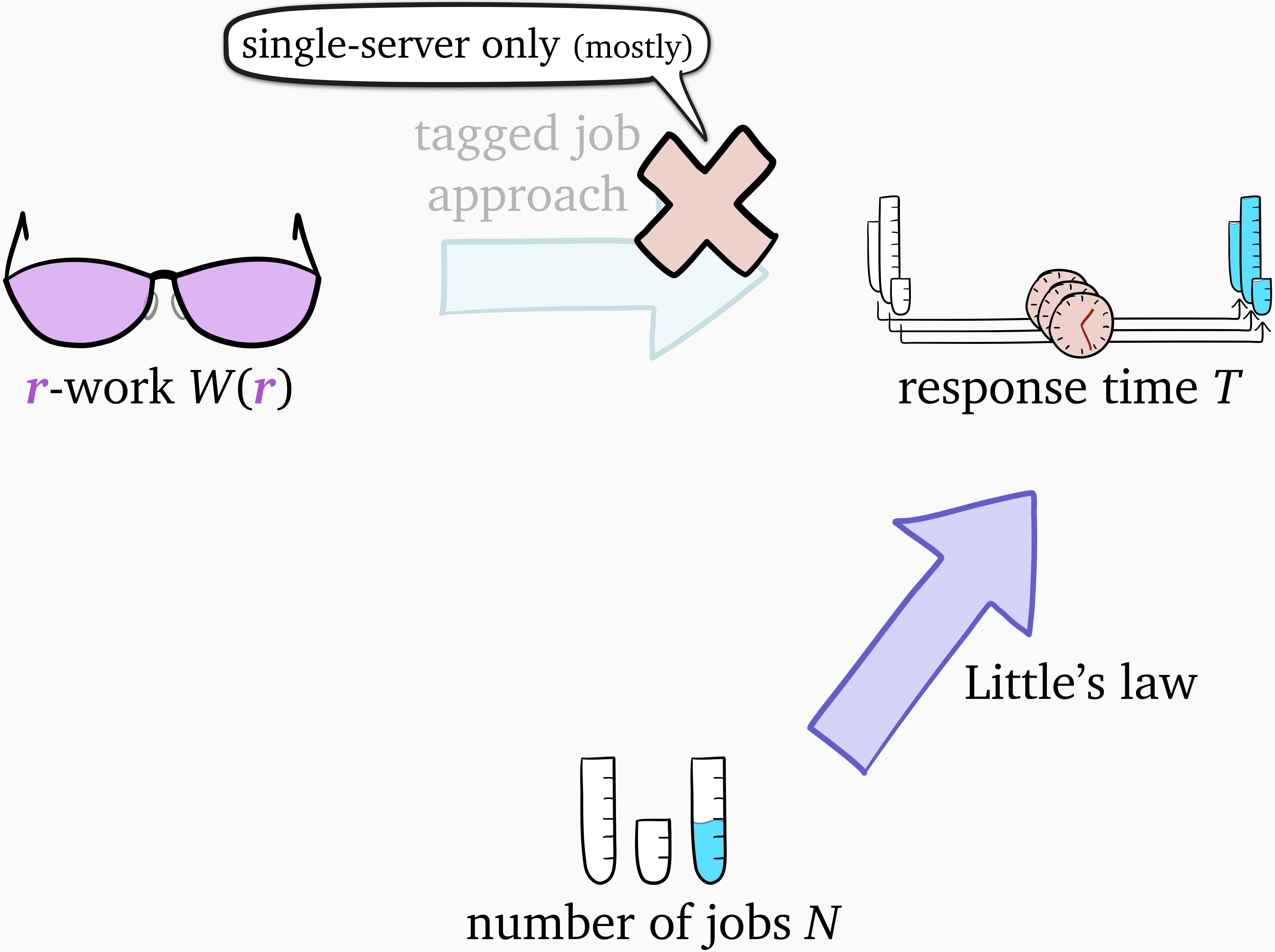
tagged job approach

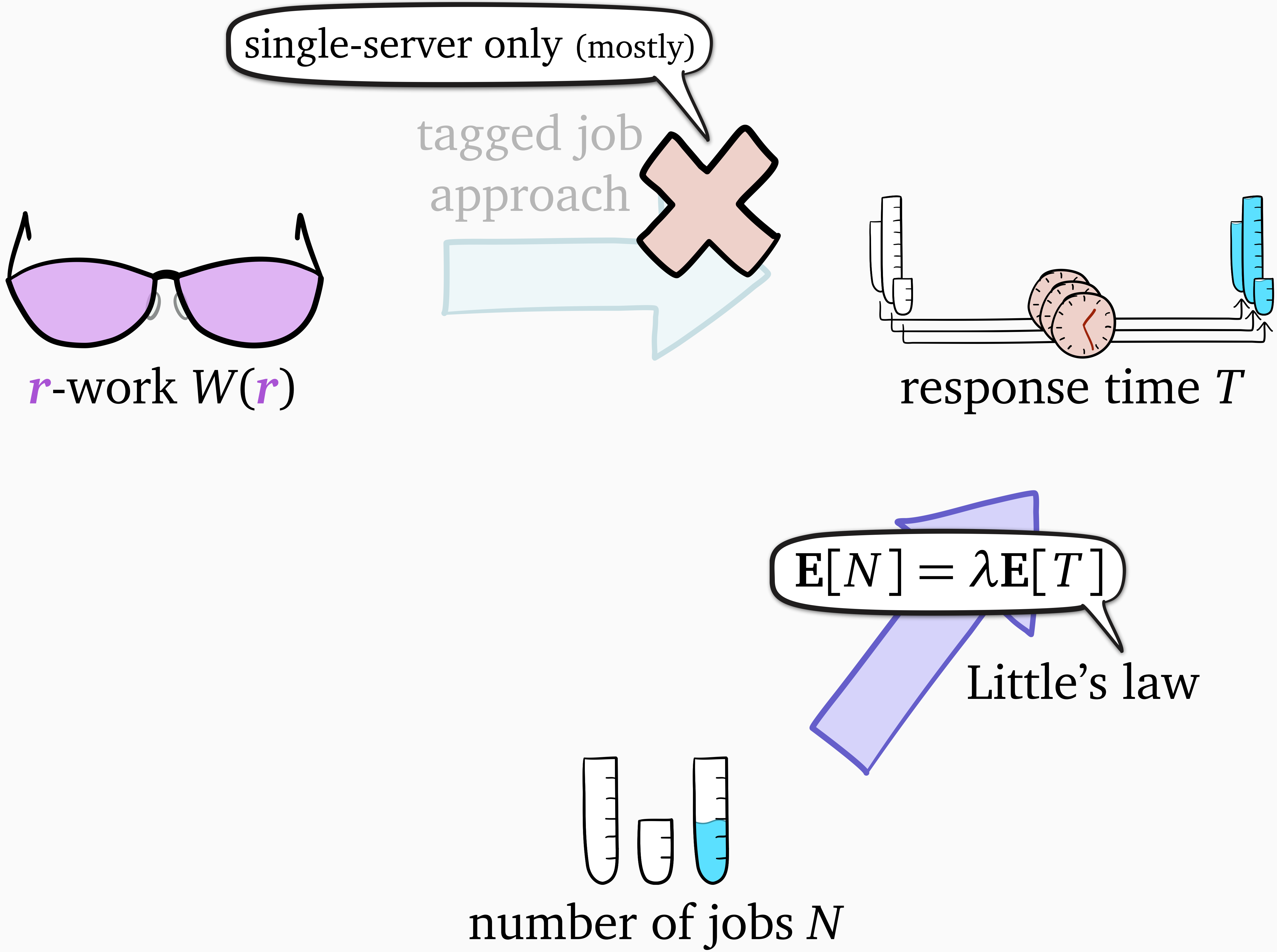


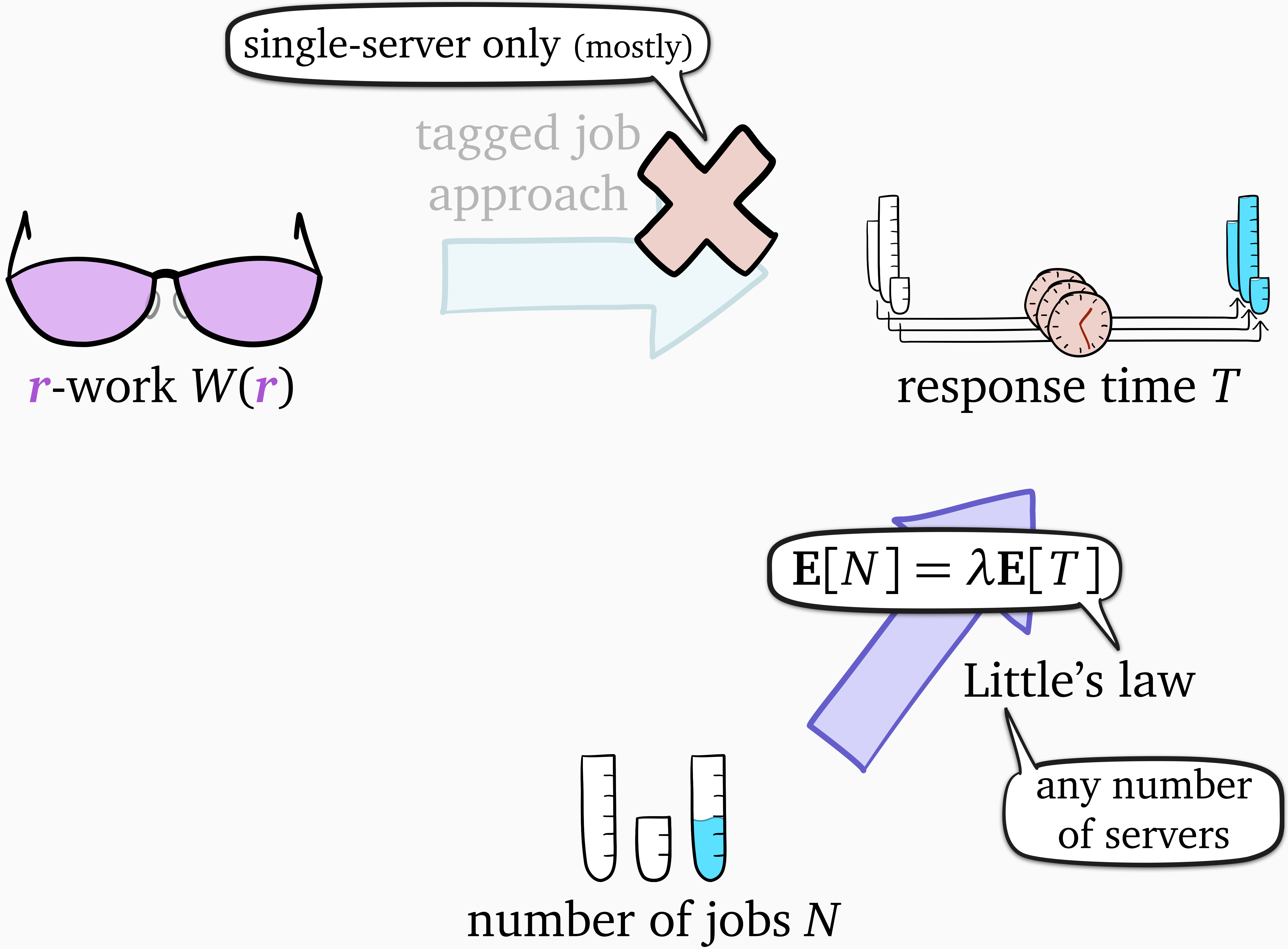


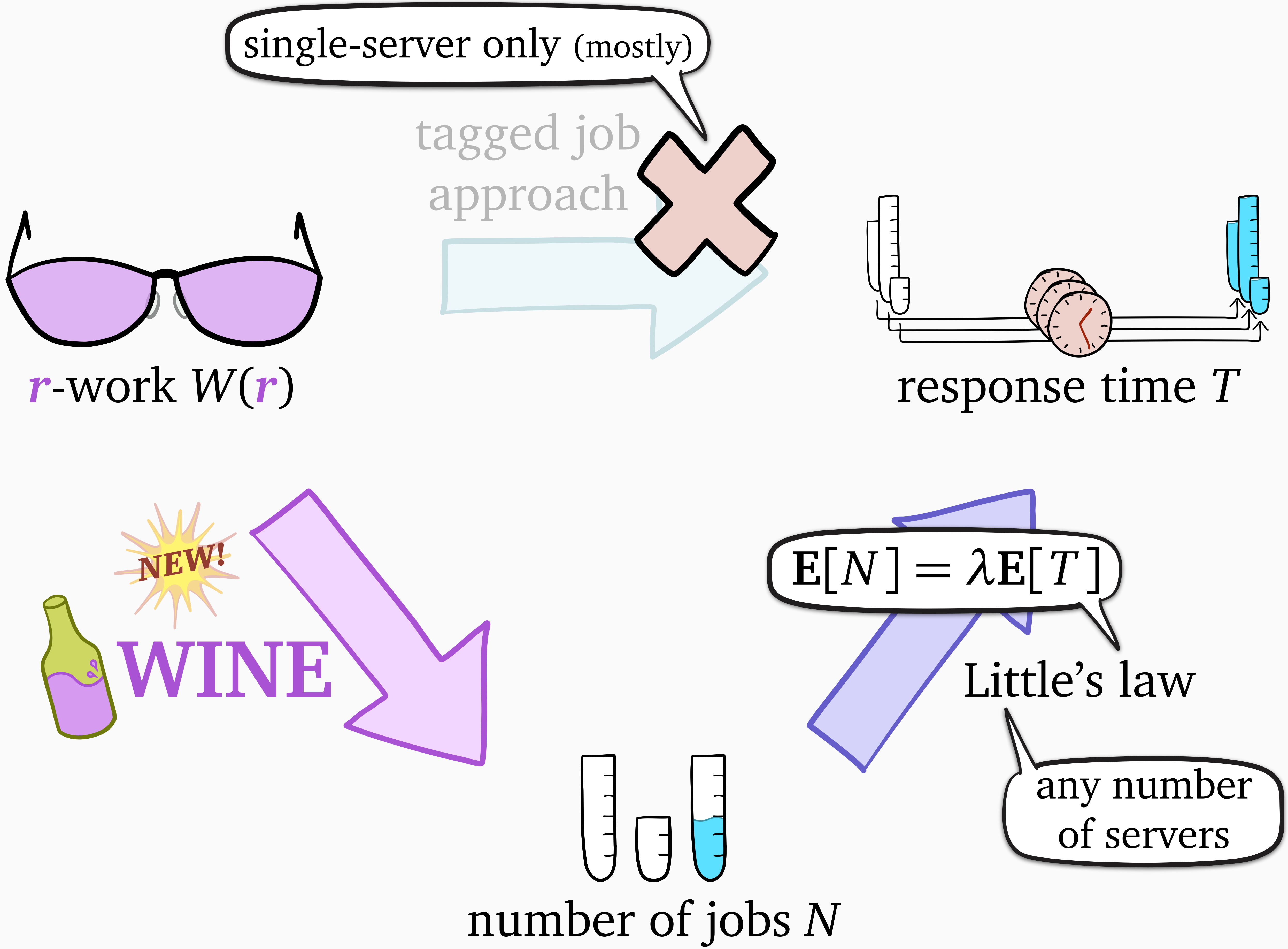
response time T

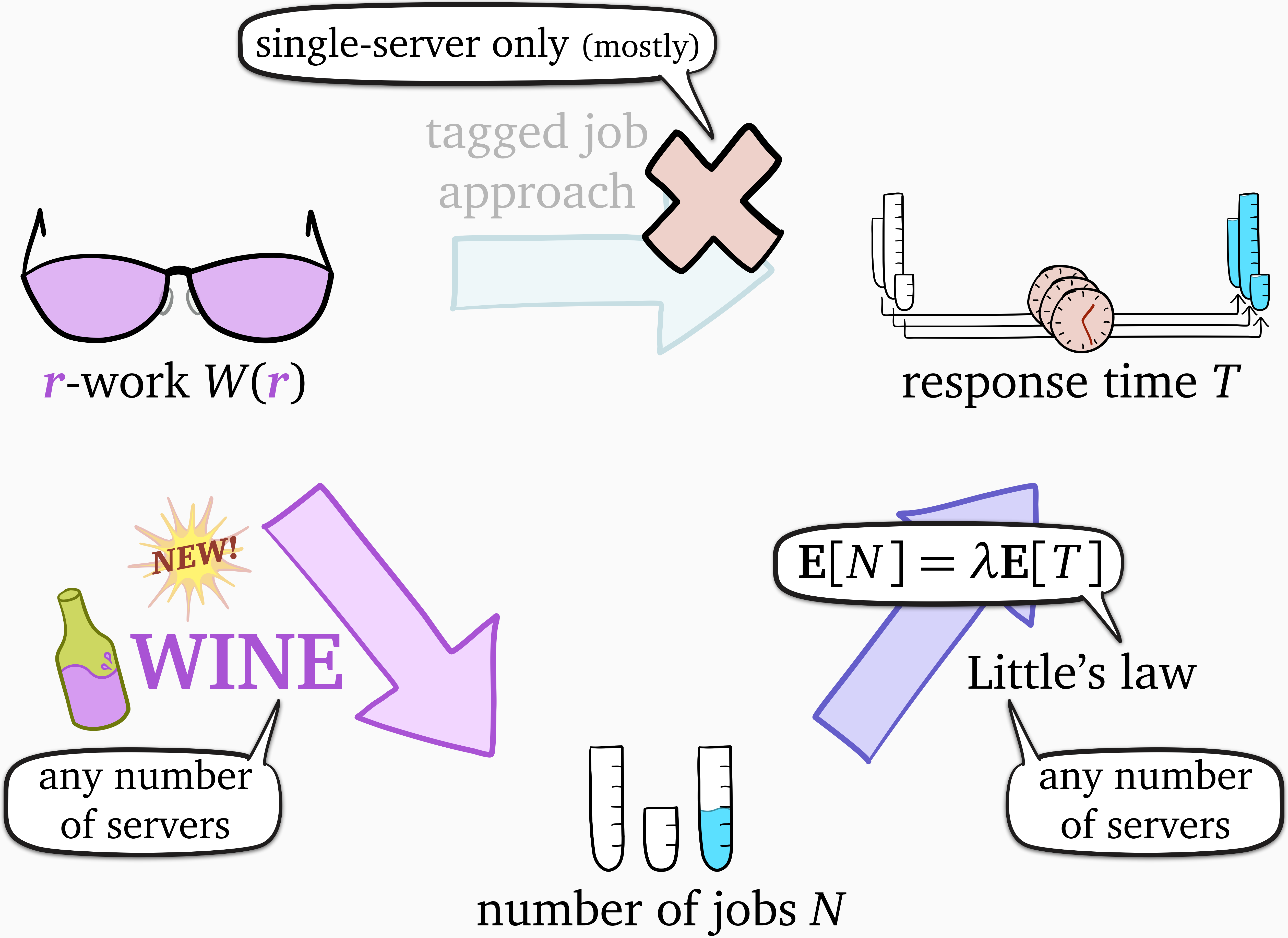

single-server only (mostly)
tagged job approach
r-work $W(r)$
response time T


single-server only (mostly)
tagged job approach
r-work W(r)
response time T
Little's law
number of jobs N


single-server only (mostly)
tagged job approach
r-work $W(r)$
response time T
$\mathbf{E}[N] = \lambda\mathbf{E}[T]$
Little's law
number of jobs N

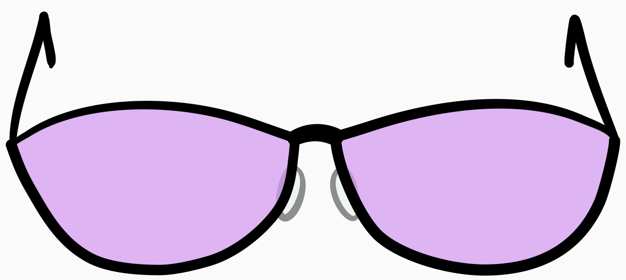

single-server only (mostly)
tagged job approach
r-work $W(r)$
response time T
$\mathbf{E}[N] = \lambda \mathbf{E}[T]$
Little's law
any number of servers
number of jobs N


single-server only (mostly)
tagged job approach
r-work W(r)
response time T
NEW!
WINE
E[N] = λE[T]
Little's law
any number of servers
number of jobs N

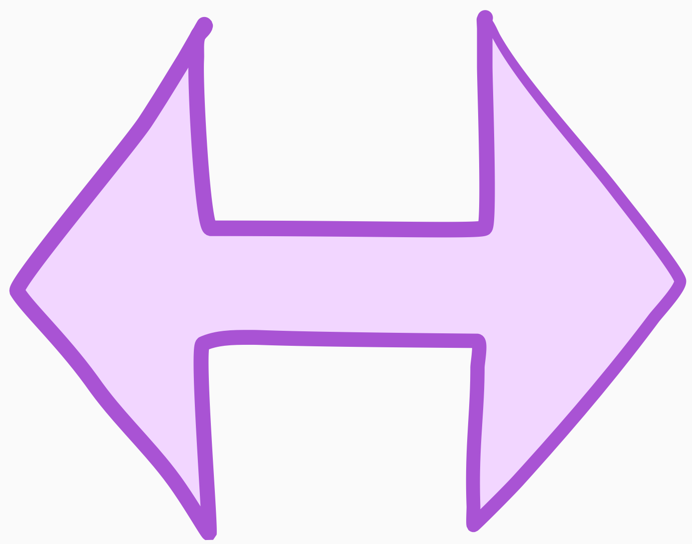

single-server only (mostly)
tagged job approach
r-work $W(r)$
response time T
NEW!
WINE
any number of servers
$\mathbf{E}[N] = \lambda \mathbf{E}[T]$
Little's law
any number of servers
number of jobs N

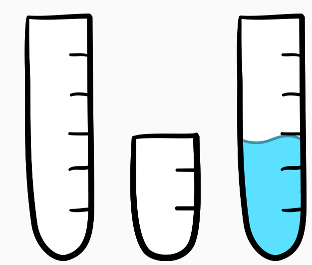
WINE

Work **I**ntegral **N**umber **E**quality



r-work $W(r)$

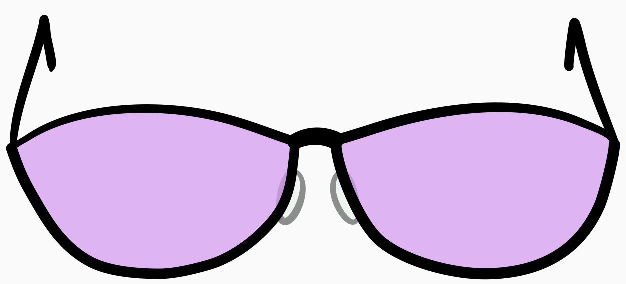




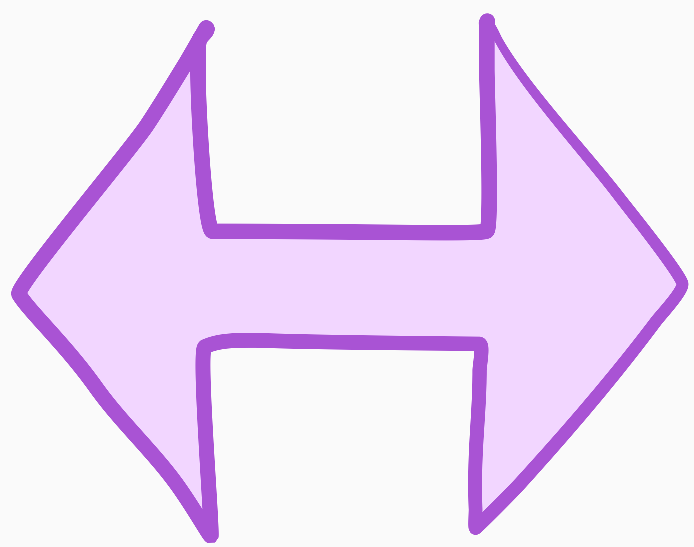
number of jobs N

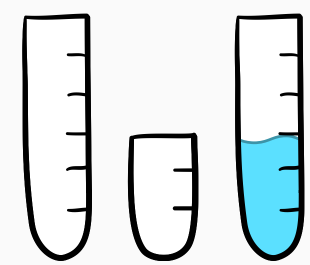
WINE

Work **I**ntegral **N**umber **E**quality

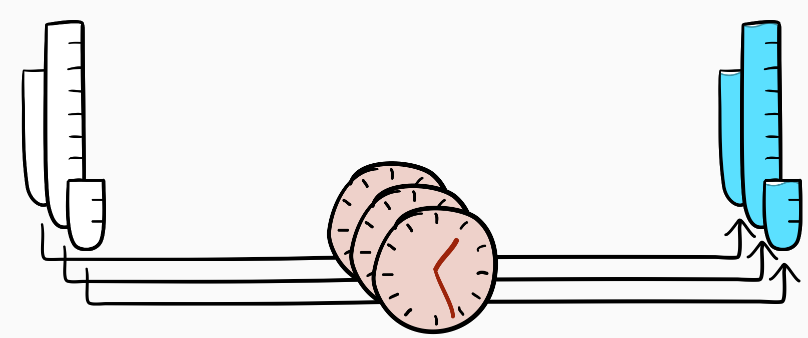


r-work $W(r)$





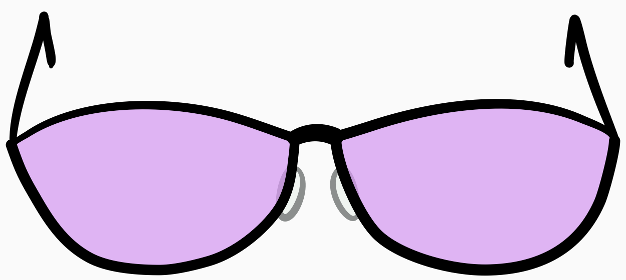
number of jobs N



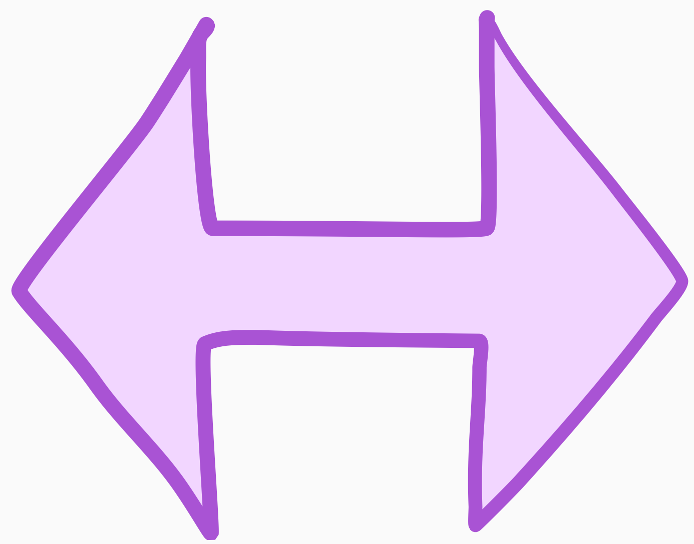
mean response time $\mathbf{E}[T]$

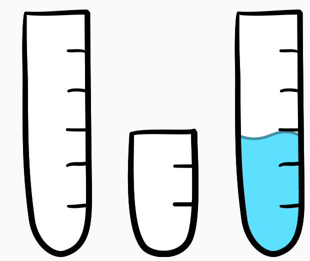
WINE

Work **I**ntegral **N**umber **E**quality

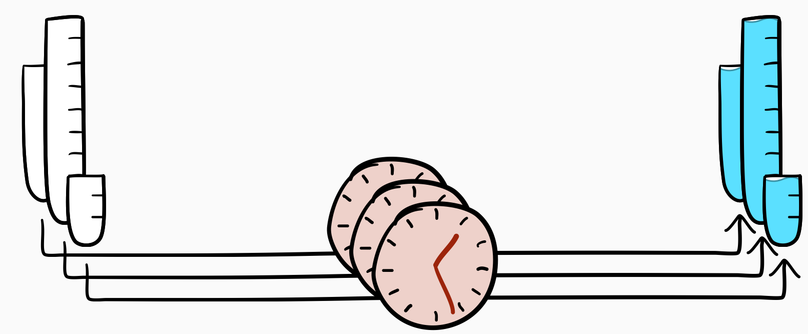


r-work $W(r)$





number of jobs N

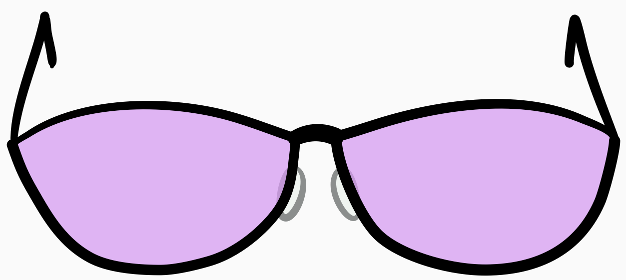


mean response time $\mathbf{E}[T]$

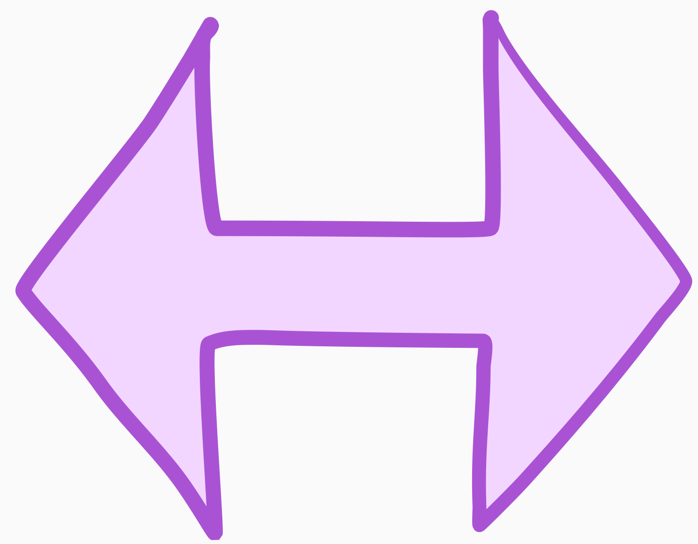
NEW! $\mathbf{E}[T]$ bounds for SRPT-k, **Gittins**-k, **noisy** size estimates

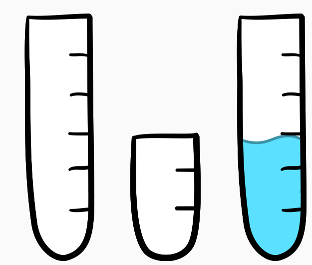
WINE

Work Integral Number Equality

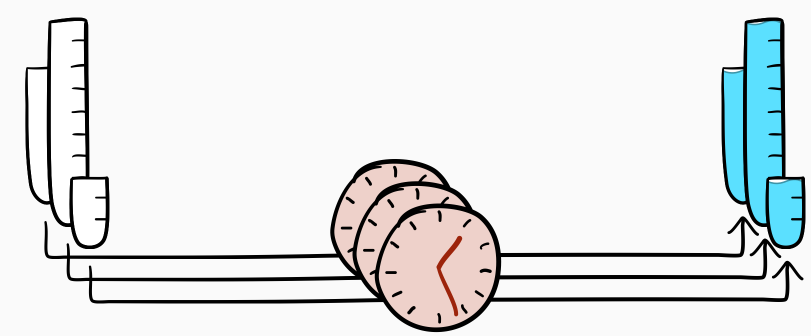


r-work $W(r)$





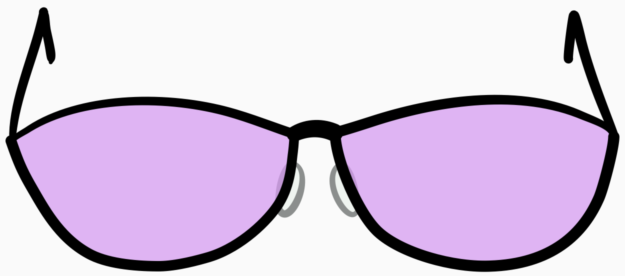
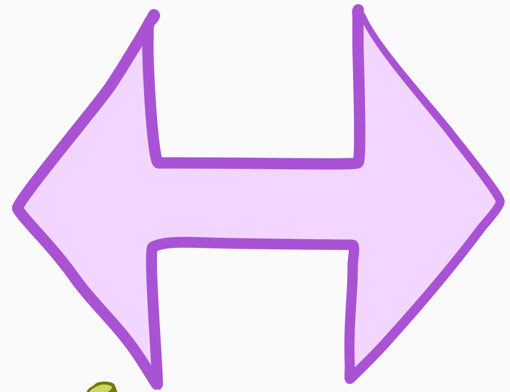
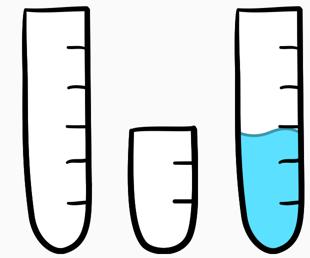
number of jobs N



mean response time $\mathbf{E}[T]$

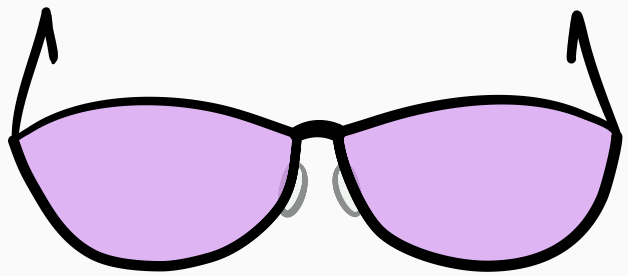
Gittins-1 minimizes $\mathbf{E}[T]$ when sizes unknown

$\mathbf{E}[T]$ bounds for SRPT-k, Gittins-k, noisy size estimates

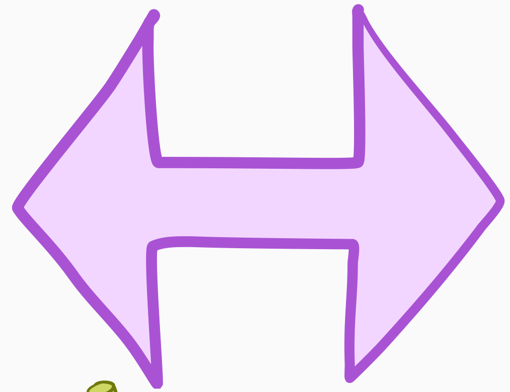
  

r-work $W(r)$ number of jobs N

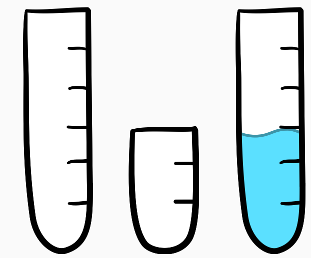
 **WINE**



r-work $W(r)$

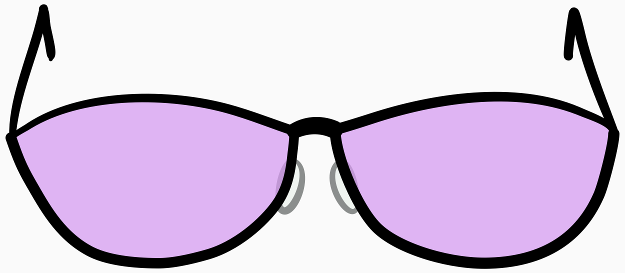


WINE

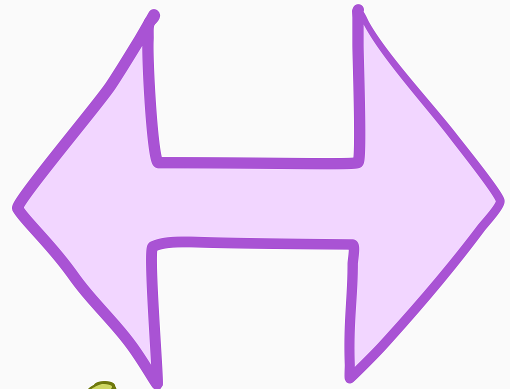


number of jobs N

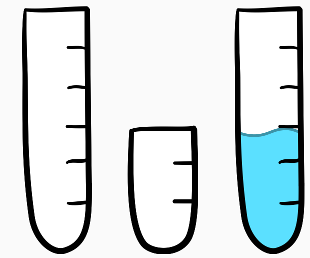
What is r-work?



r-work $W(r)$



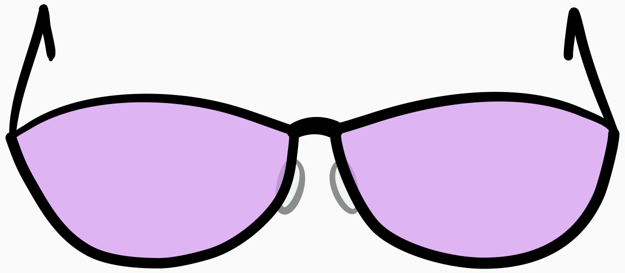
WINE



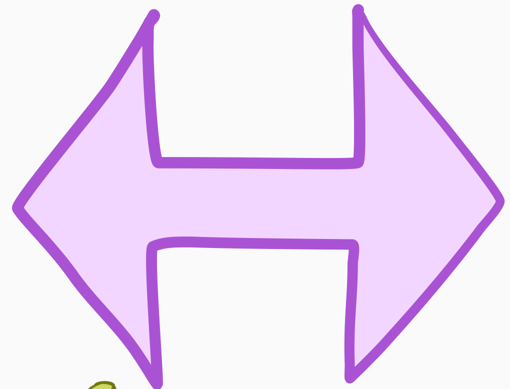
number of jobs N

What is r-work?

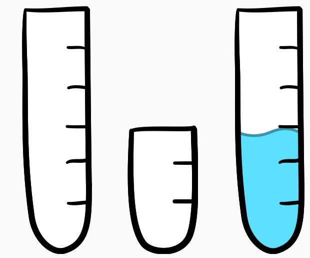
How do we get number of jobs from r-work?



r-work $W(r)$



WINE

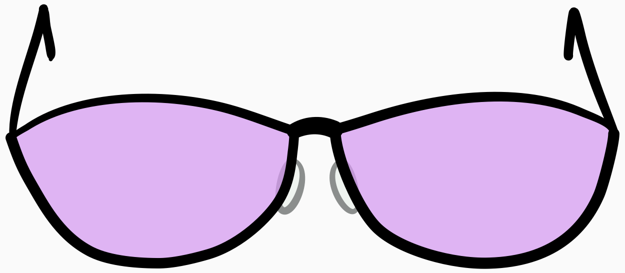


number of jobs N

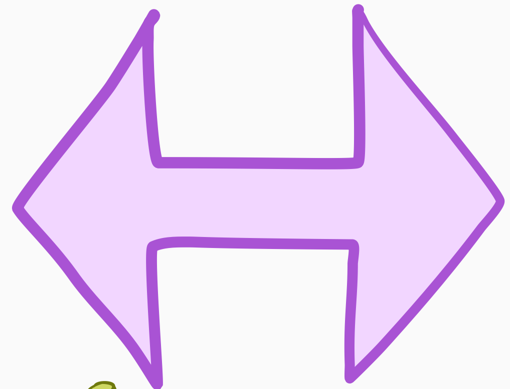
What is r-work?

How do we get number of jobs from r-work?

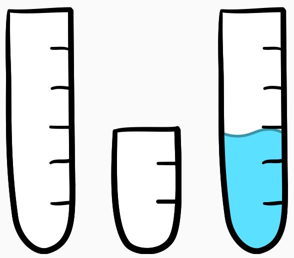
How do we analyze r-work?



r-work $W(r)$



WINE



number of jobs N

This talk:
SRPT-*k*

What is *r*-work?

How do we get number of jobs from *r*-work?

How do we analyze *r*-work?

Defining r-work

for SRPT

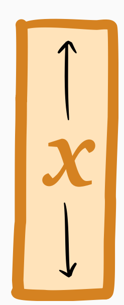
$W(r)$ = work relevant to **rank** r

Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$w_x(r)$ = r-work of *single job* of rem. size x = $\left\{ \right.$

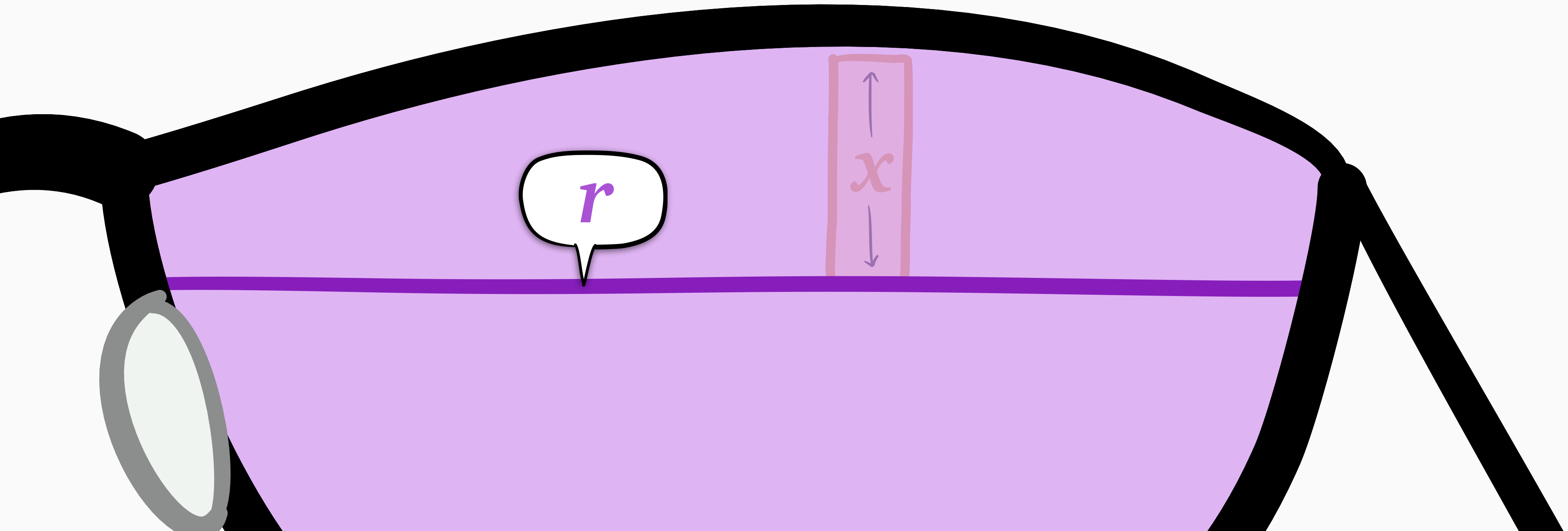


Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$w_x(r)$ = r-work of *single job* of rem. size x = $\Big\{$

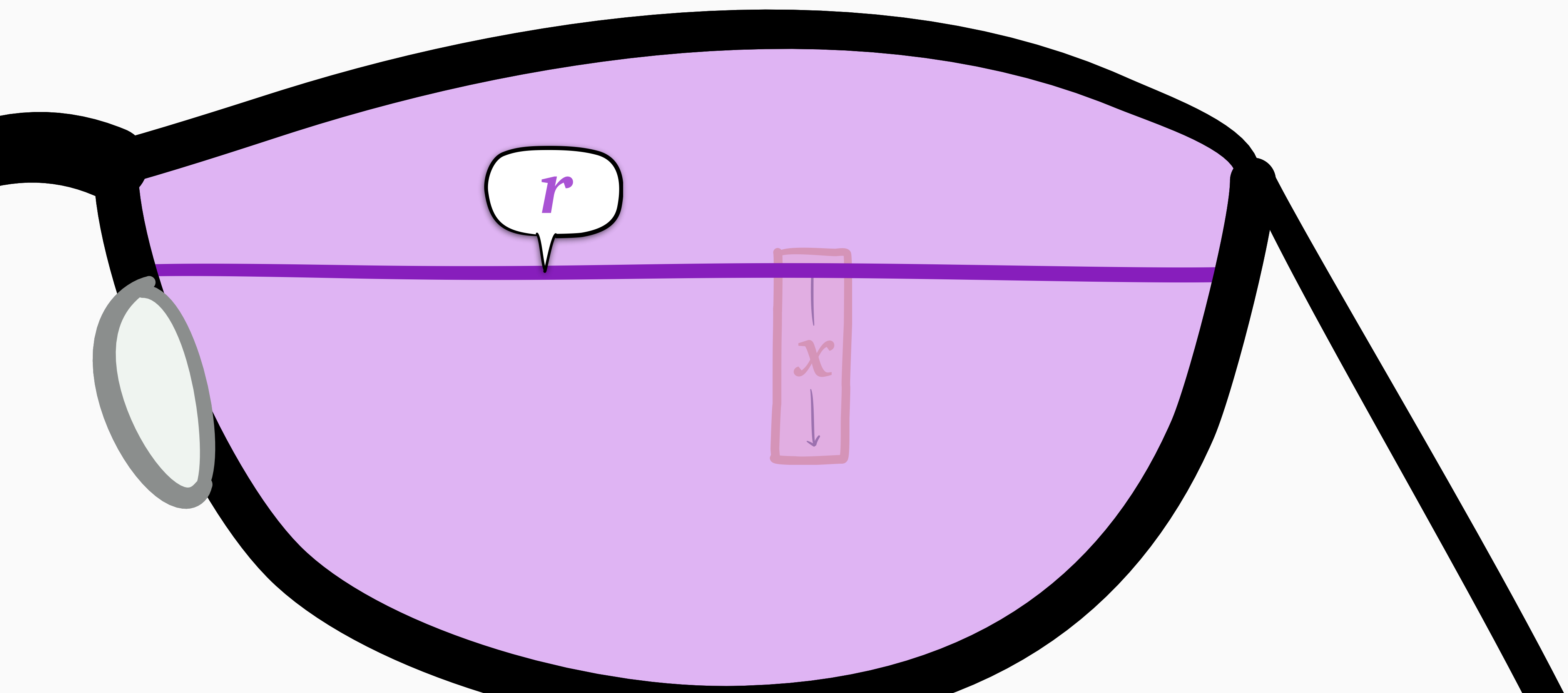


Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$w_x(r)$ = r-work of *single job* of rem. size x = $\{$

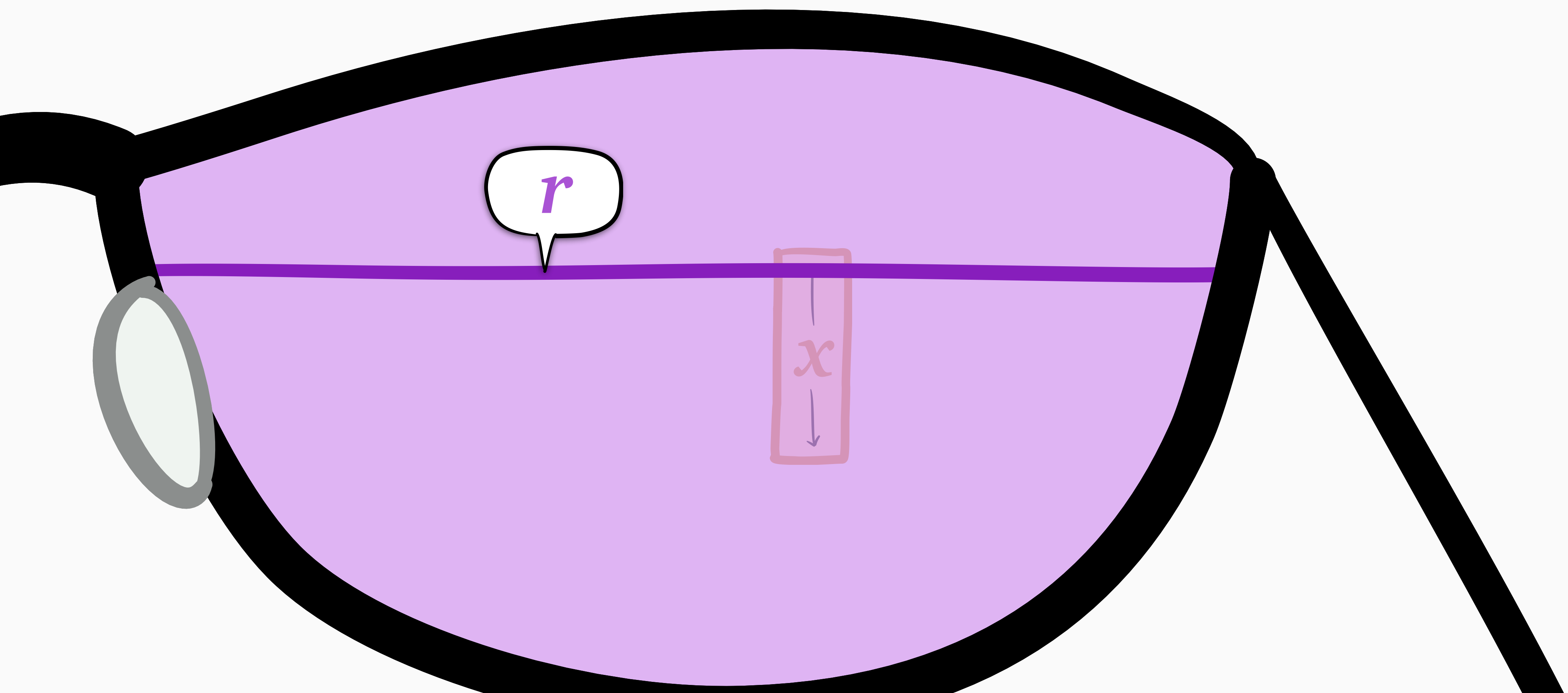


Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$$w_x(r) = r\text{-work of } \textit{single job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ & \end{cases}$$

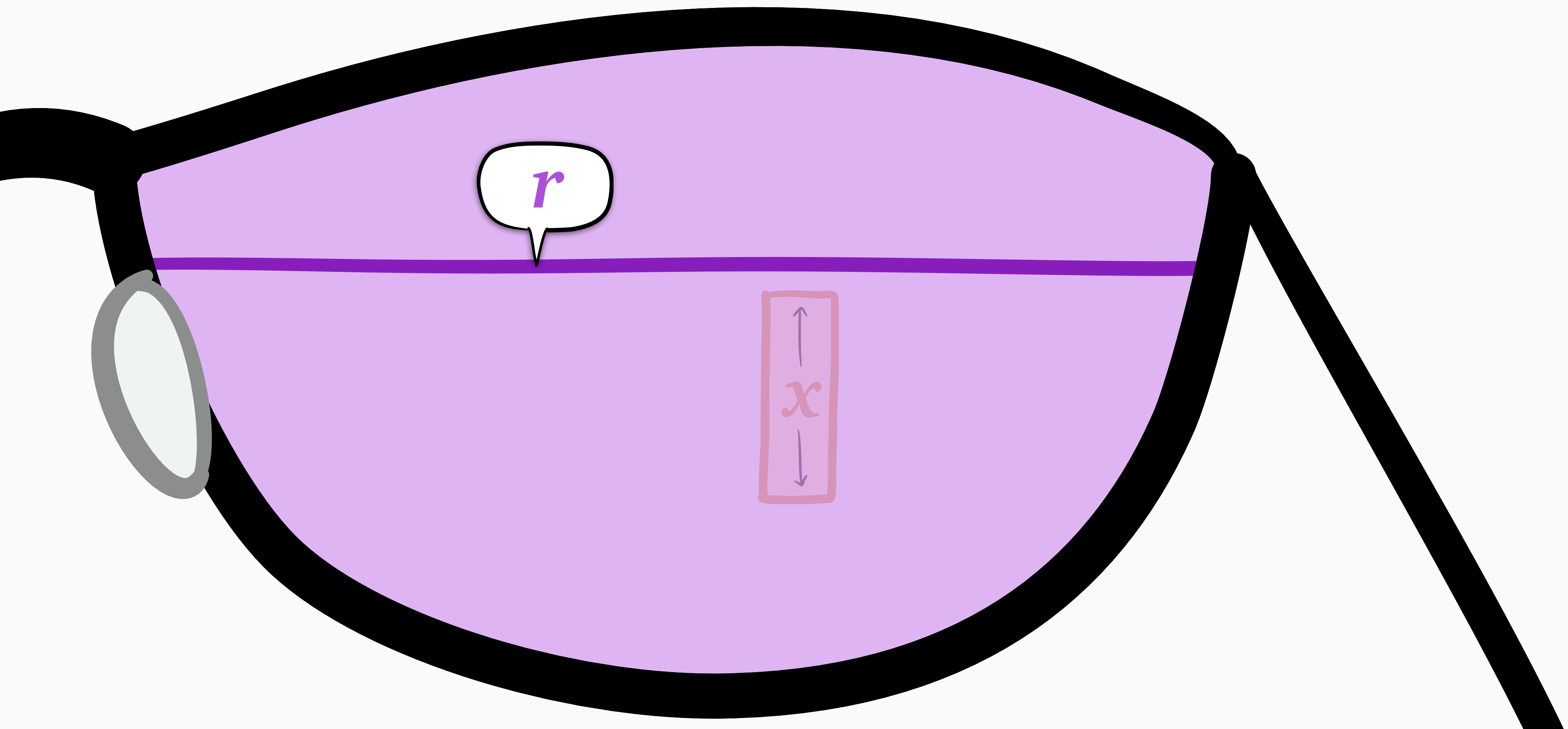


Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$$w_x(r) = r\text{-work of } \textit{single job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ & \end{cases}$$

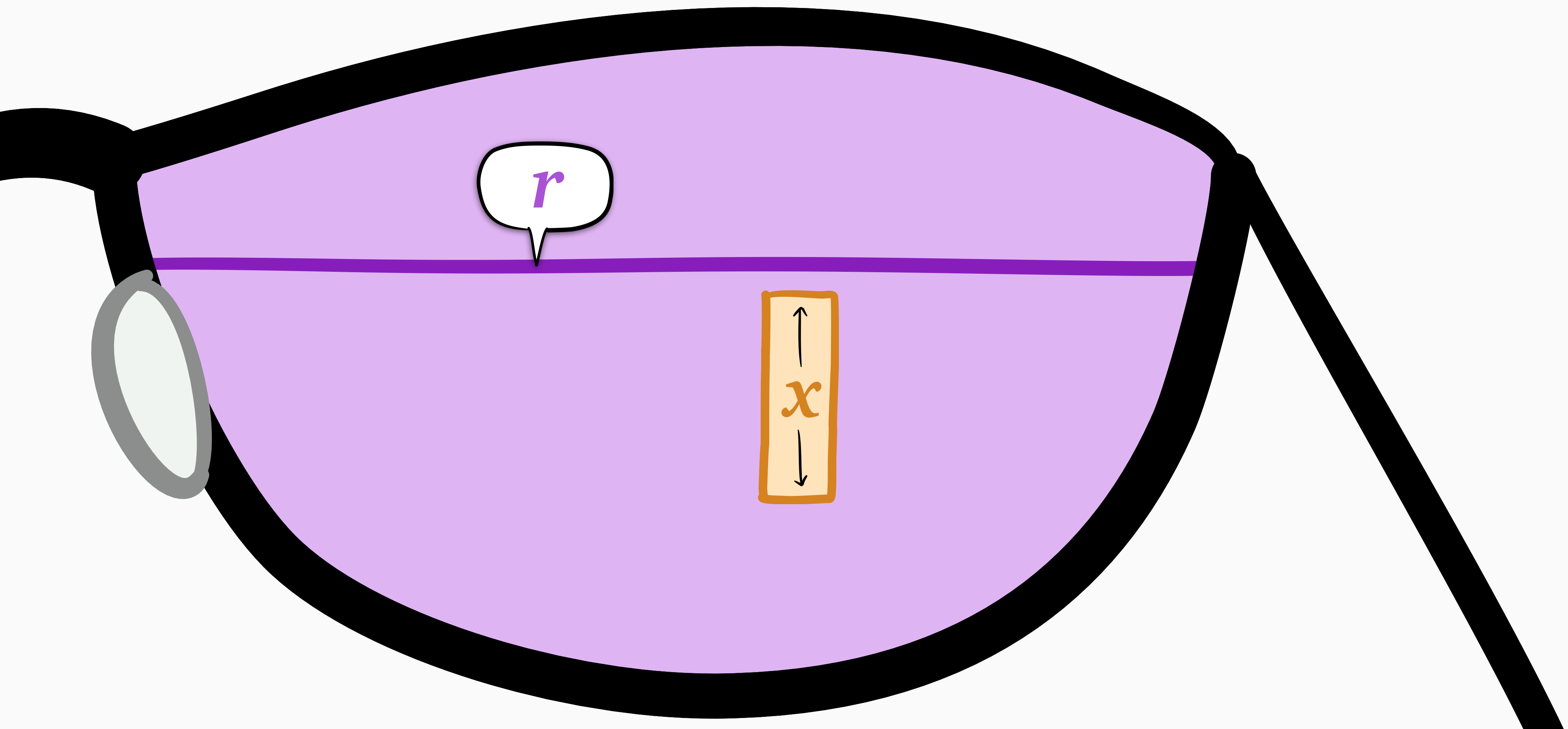


Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$w_x(r)$ = r-work of *single job* of rem. size x = $\begin{cases} 0 & \text{if } r < x \\ & \end{cases}$

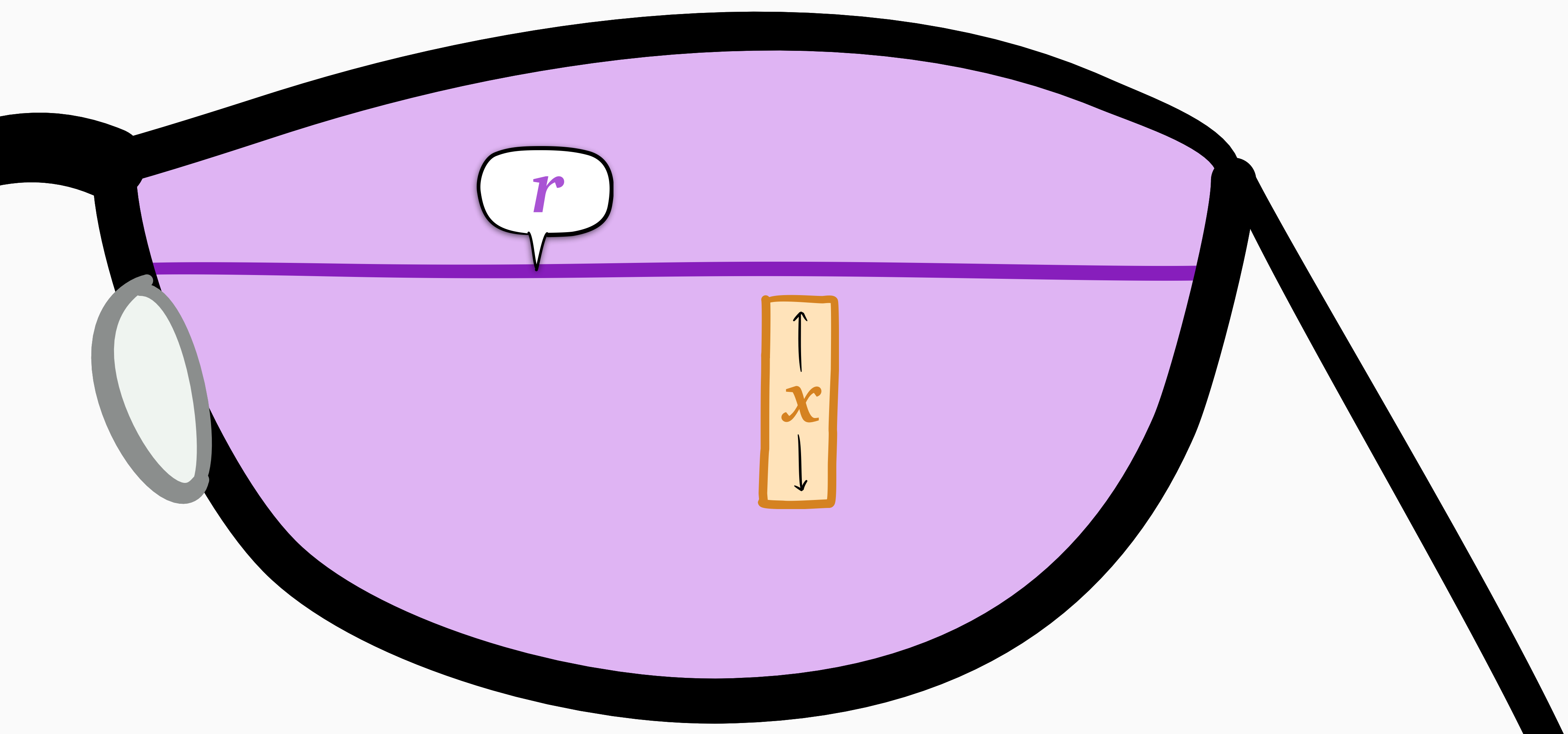


Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$w_x(r)$ = r-work of *single job* of rem. size x $= \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$

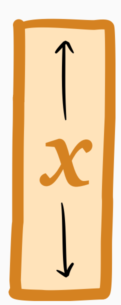


Defining r-work

for SRPT

$W(r)$ = work relevant to **rank** r

$$w_x(r) = r\text{-work of } \textit{single job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

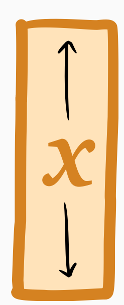


Defining r-work

for SRPT

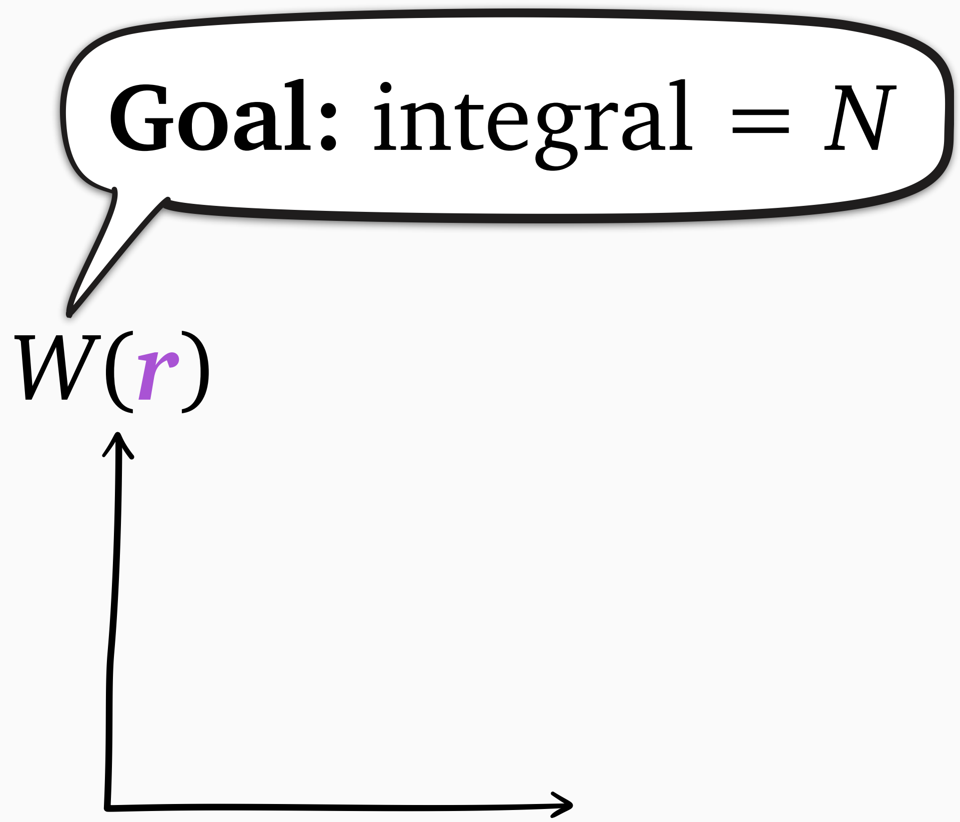
$W(r)$ = work relevant to **rank** r
= total r-work of all jobs

$$w_x(r) = r\text{-work of } \textit{single job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

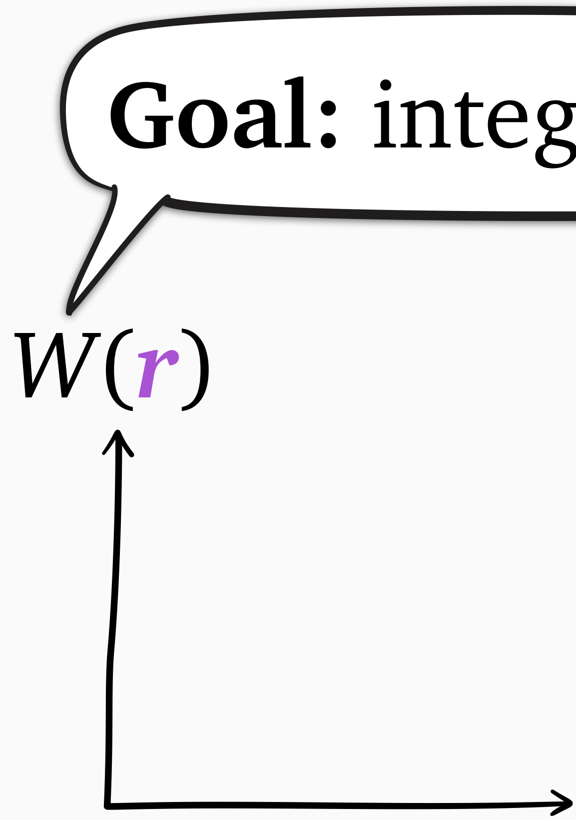


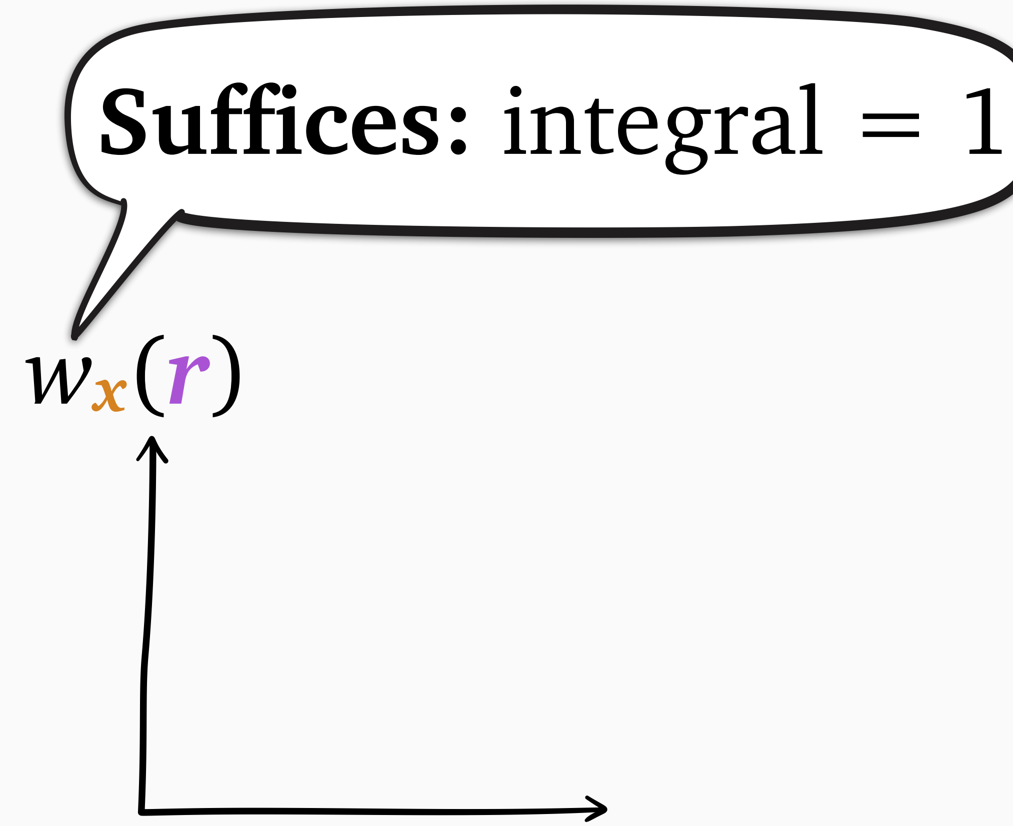
From r-work to number of jobs N

From r-work to number of jobs N



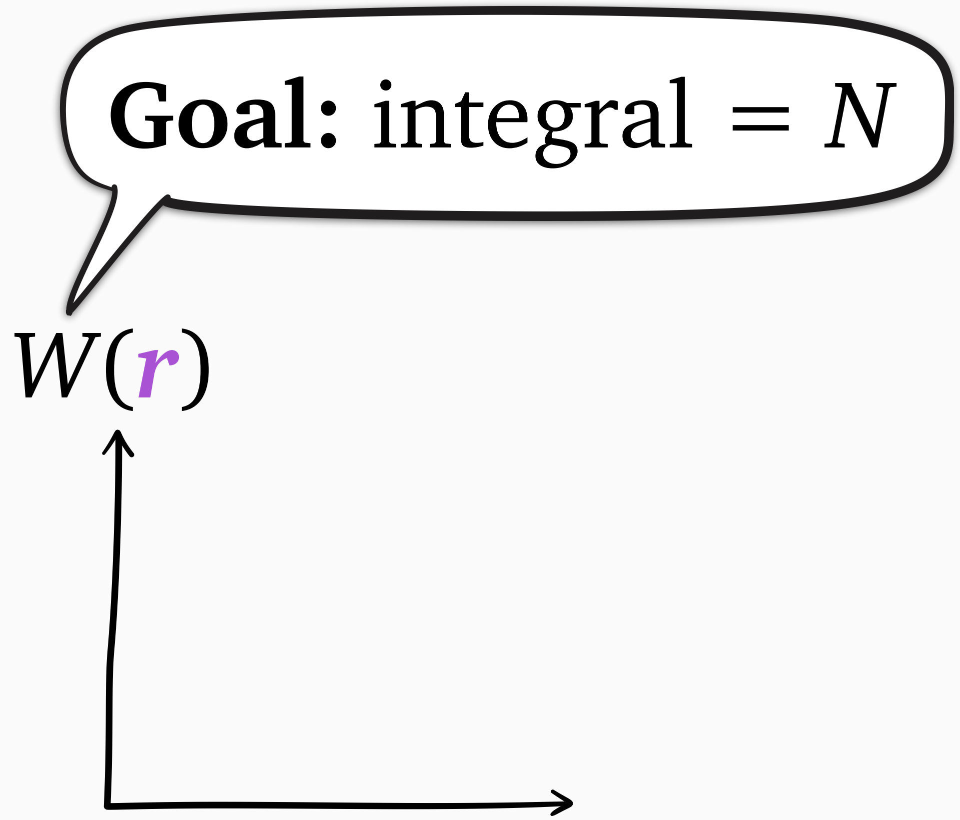
From r-work to number of jobs N

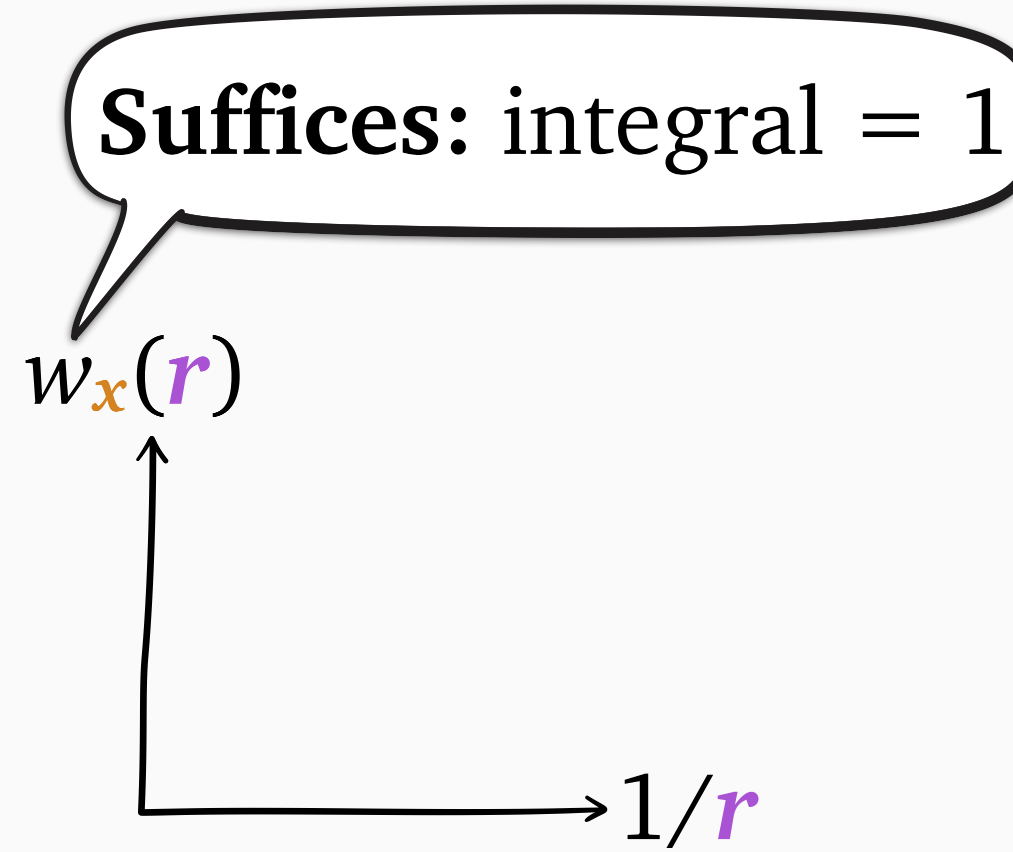




$$w_x(r) = r\text{-work of } \textit{job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

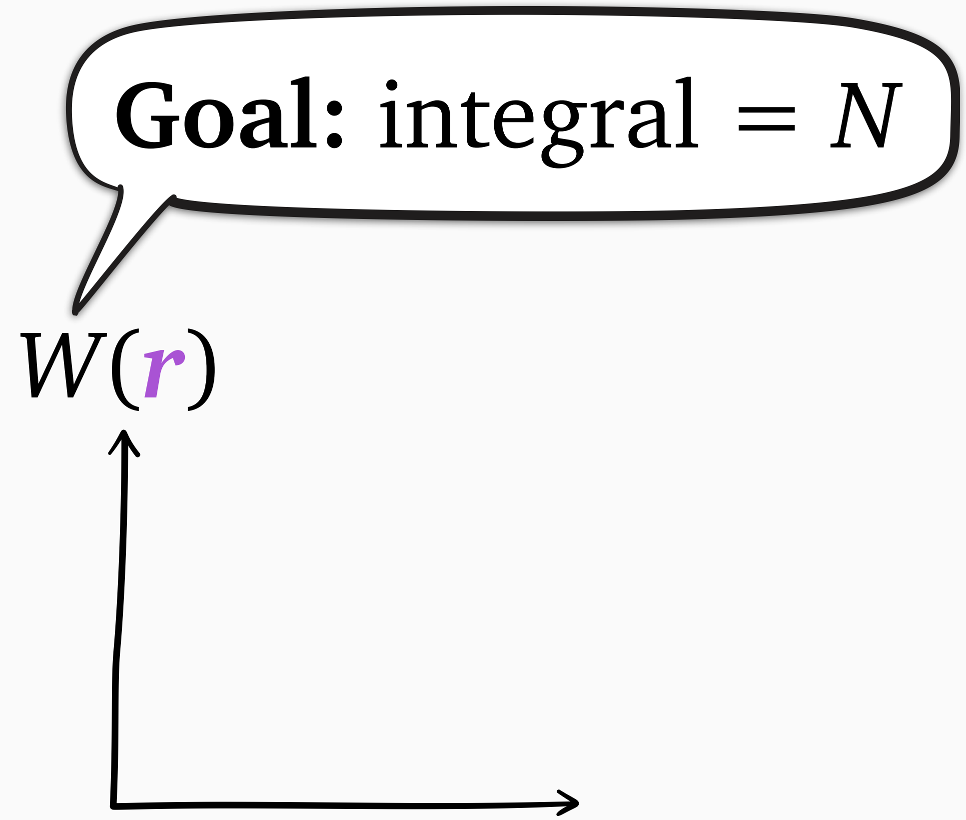
From r-work to number of jobs N

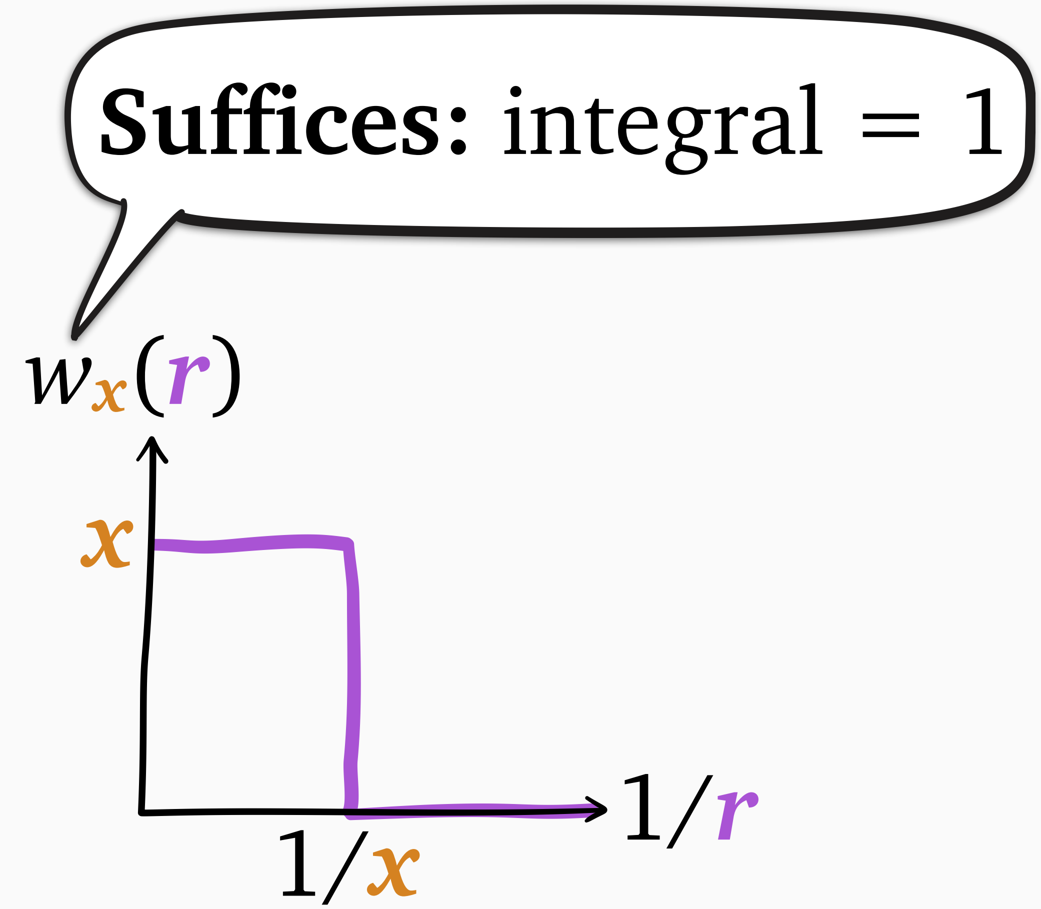




$$w_x(r) = r\text{-work of } \textit{job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

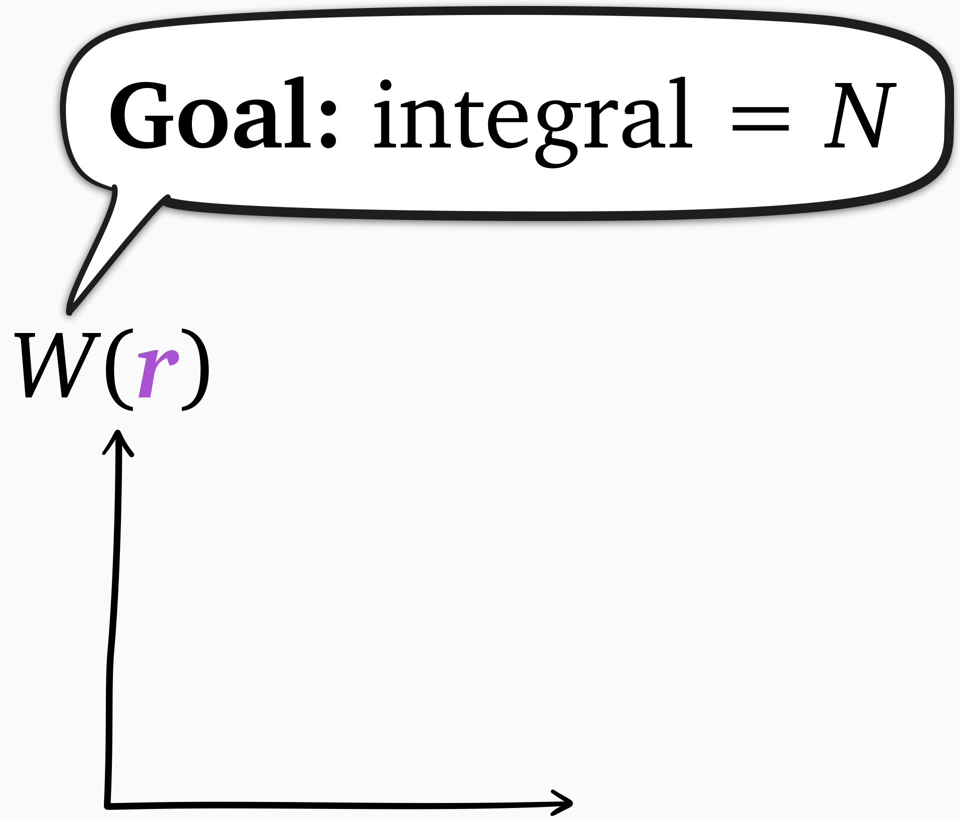
From r-work to number of jobs N

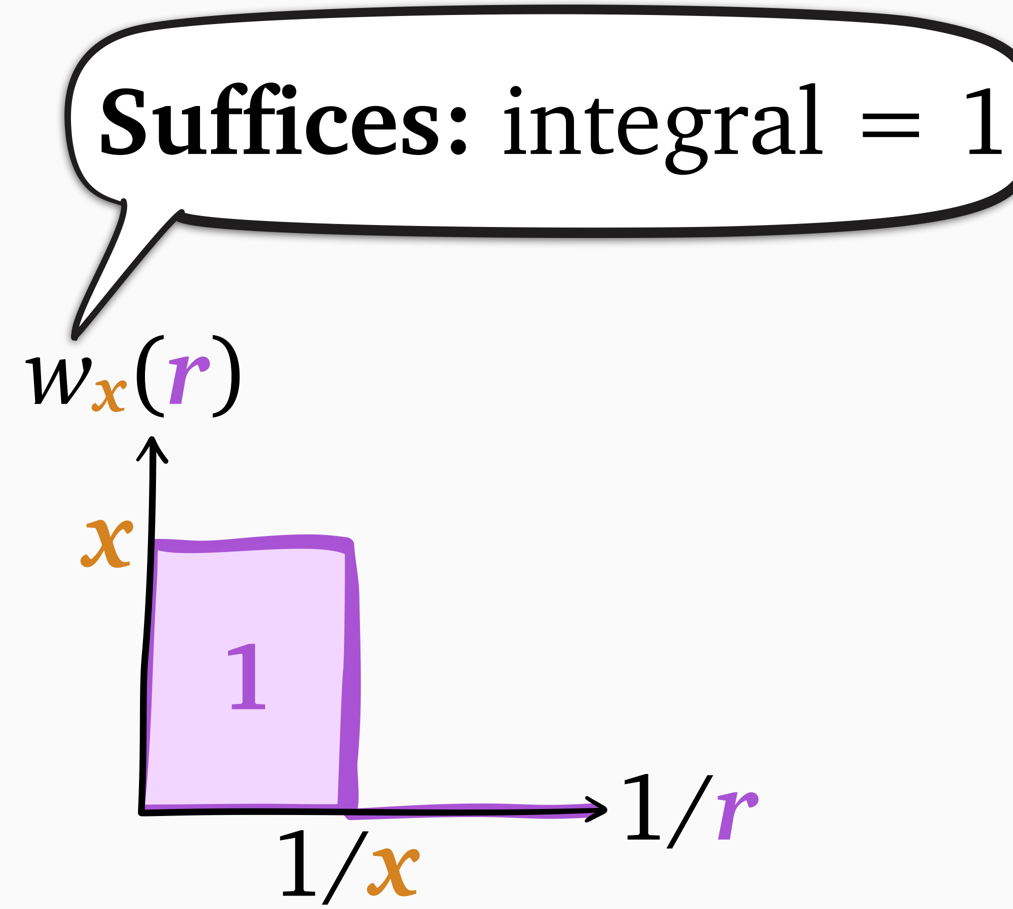




$$w_x(r) = r\text{-work of } \textit{job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

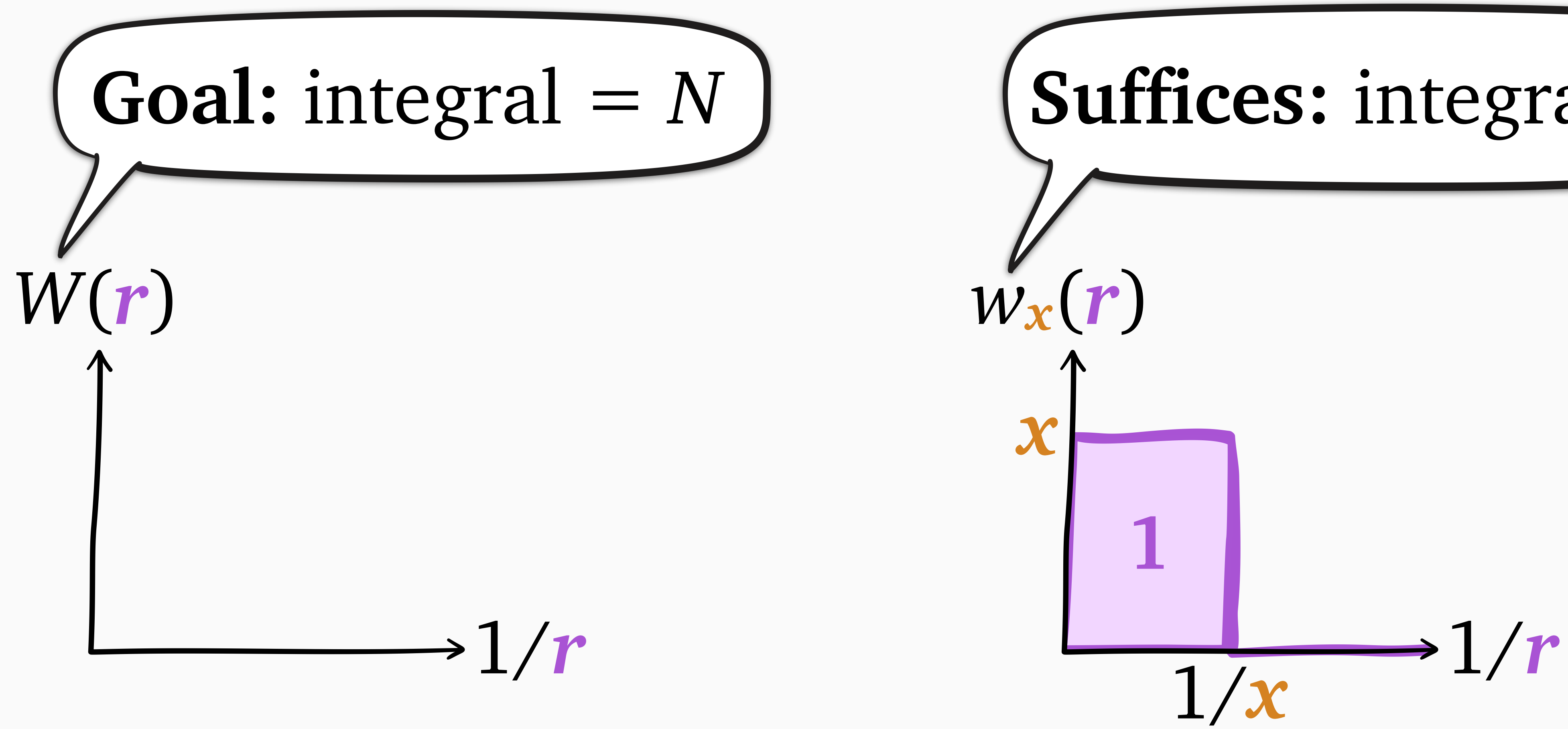
From r-work to number of jobs N





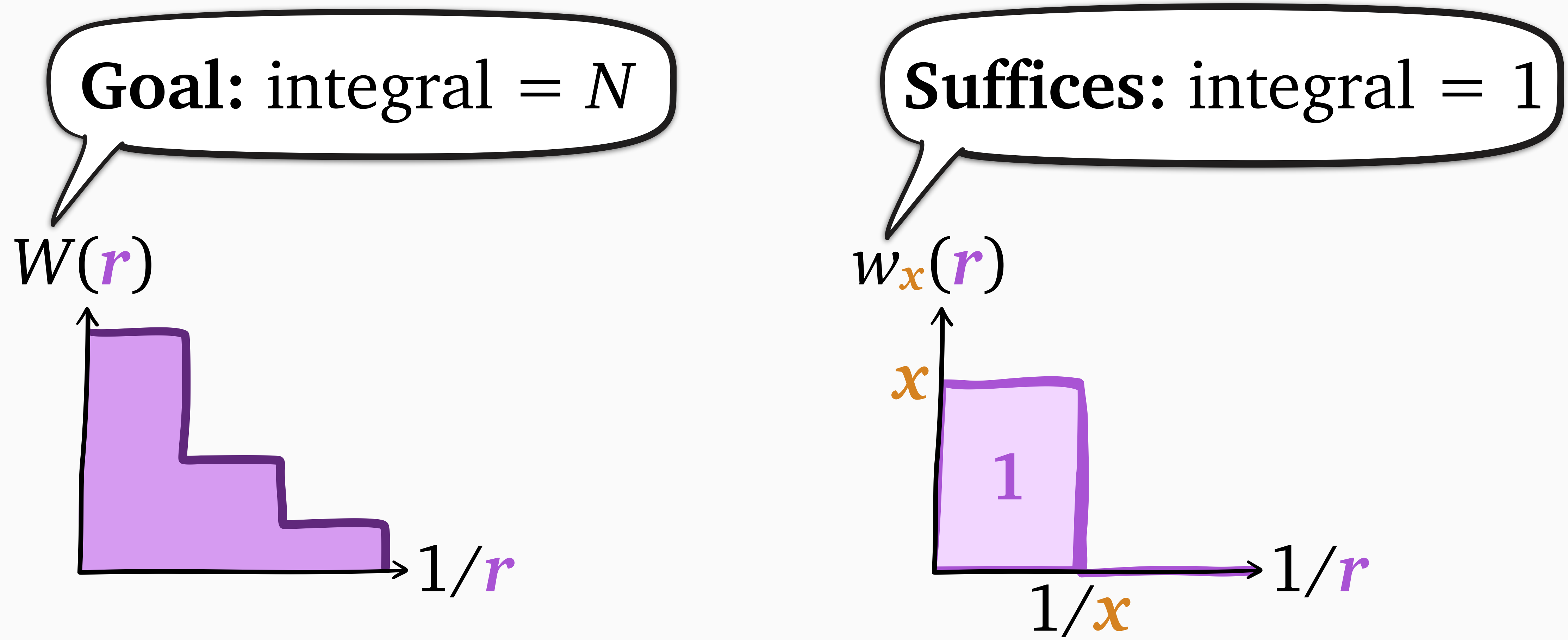
$$w_x(r) = r\text{-work of } \textit{job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

From r-work to number of jobs N



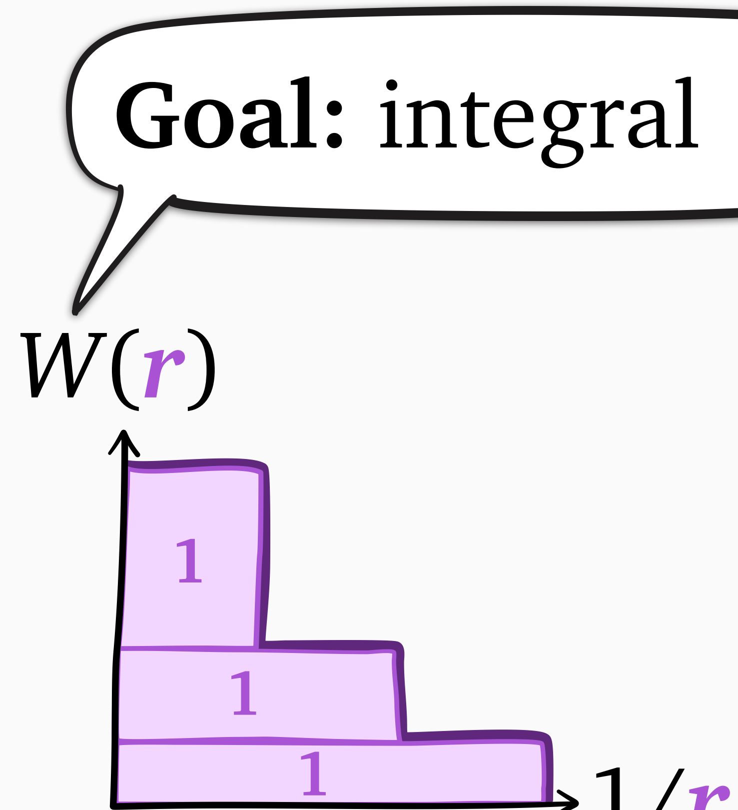
$$w_x(r) = r\text{-work of } \textit{job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

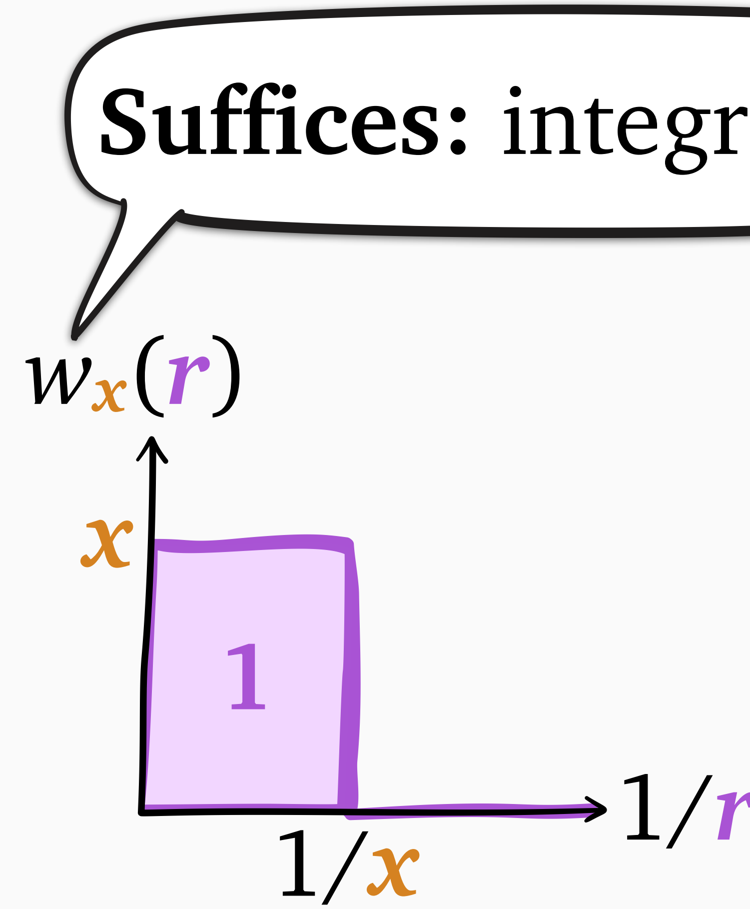
From r-work to number of jobs N



$$w_x(r) = r\text{-work of } \textit{job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

From r-work to number of jobs N

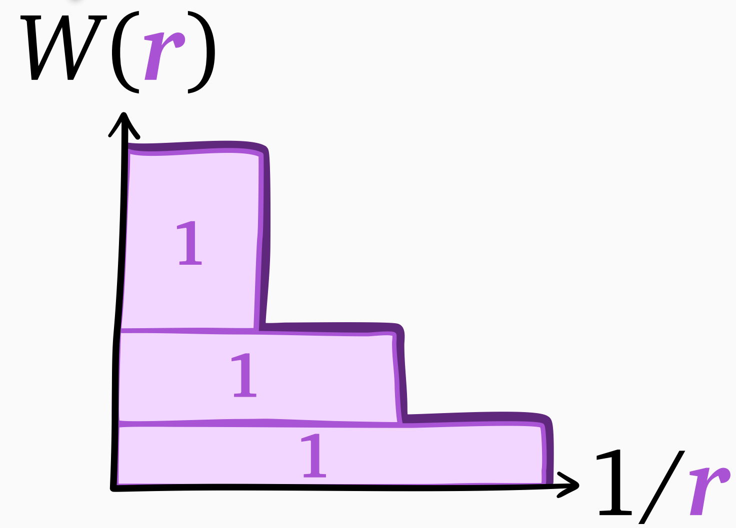




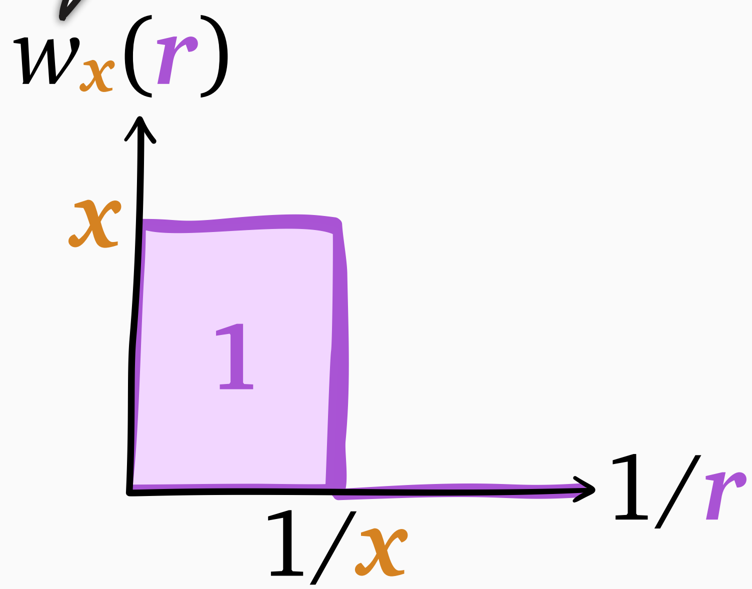
$$w_x(r) = r\text{-work of } \textit{job} \text{ of rem. size } x = \begin{cases} 0 & \text{if } r < x \\ x & \text{if } r \geq x \end{cases}$$

From r-work to number of jobs N

Goal: integral $= N$



Suffices: integral $= 1$



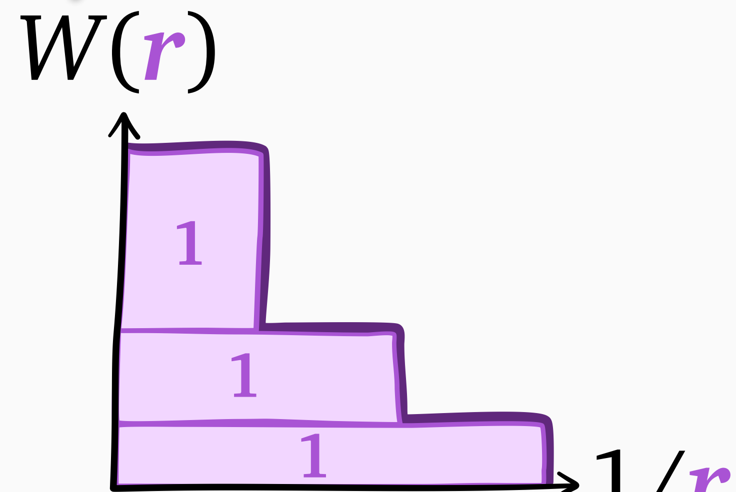
NEW!

Theorem:

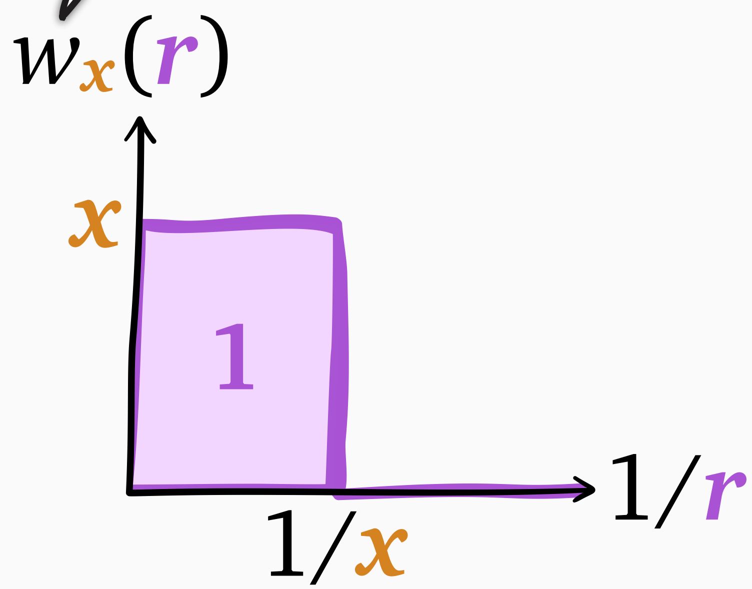
$$N = \int_0^\infty \frac{W(r)}{r^2}\,\mathrm{d}r$$

From r-work to number of jobs N

Goal: integral $= N$



Suffices: integral $= 1$



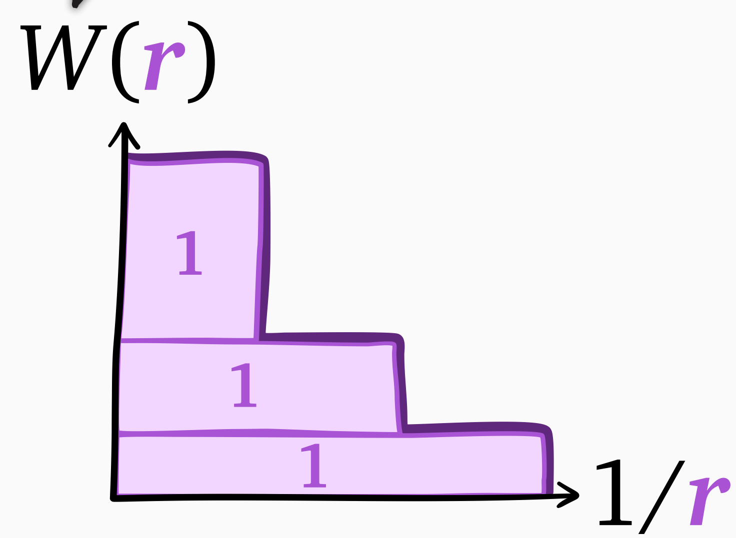
WINE

Theorem:

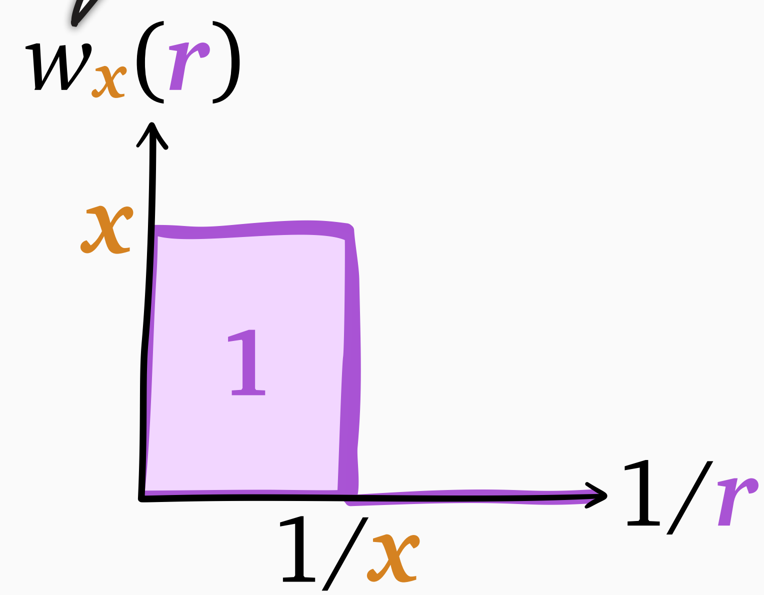
$$N = \int_0^\infty \frac{W(r)}{r^2}\, \mathrm{d}r$$

From r-work to number of jobs N

Goal: integral $= N$



Suffices: integral $= 1$



WINE

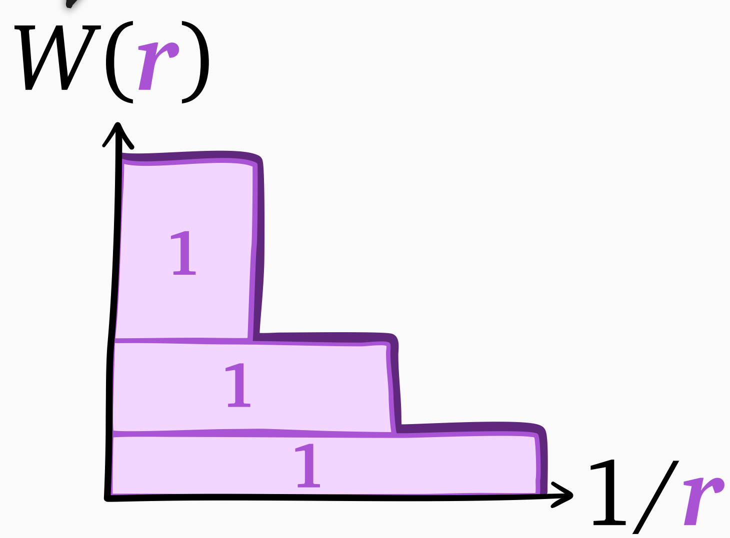
Theorem:

$$N = \int_0^\infty \frac{W(r)}{r^2}\,\mathrm{d}r$$

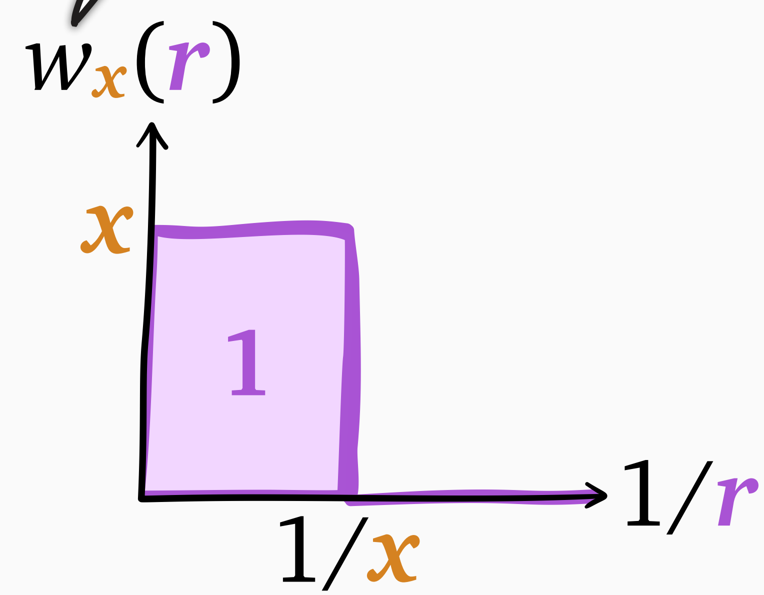
uses **rank** = rem. size

From r-work to number of jobs N

Goal: integral $= N$



Suffices: integral $= 1$



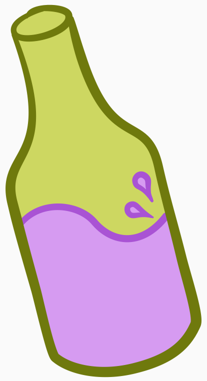
WINE

Theorem: under *any* policy,

$$N = \int_0^\infty \frac{W(r)}{r^2}\,\mathrm{d}r$$

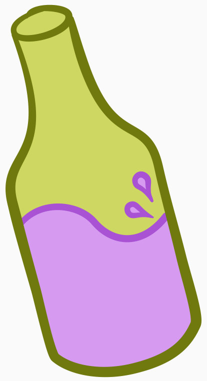
uses **rank** = rem. size

Impact of **WINE**



Impact of **WINE**

multiserver systems

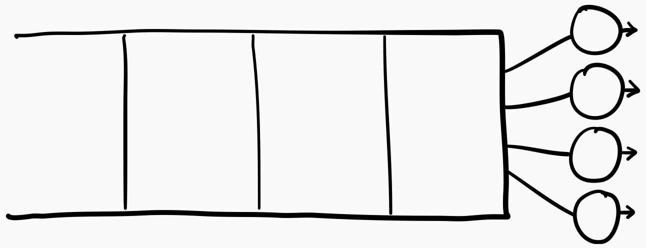


Impact of **WINE**

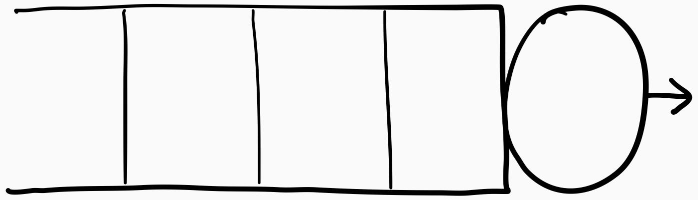
multiserver systems

noisy size estimates

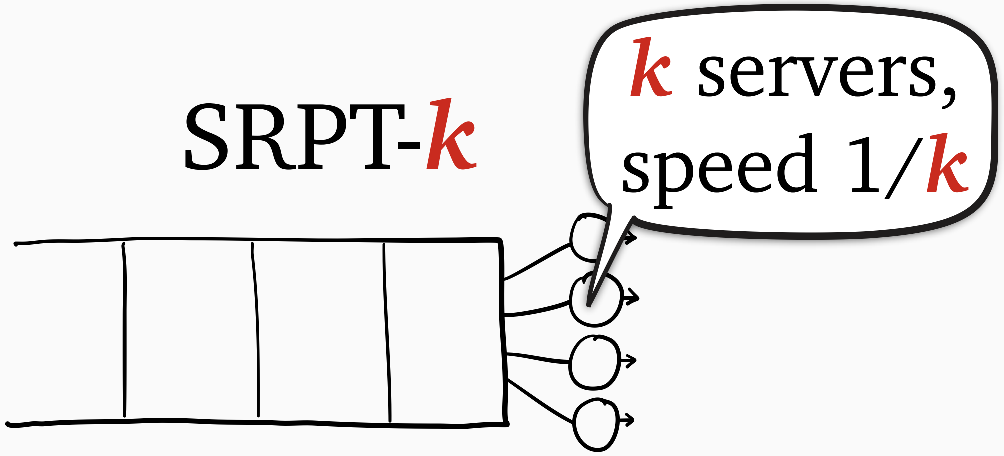
SRPT-*k*



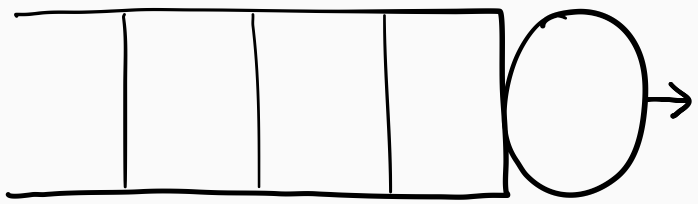
SRPT-1

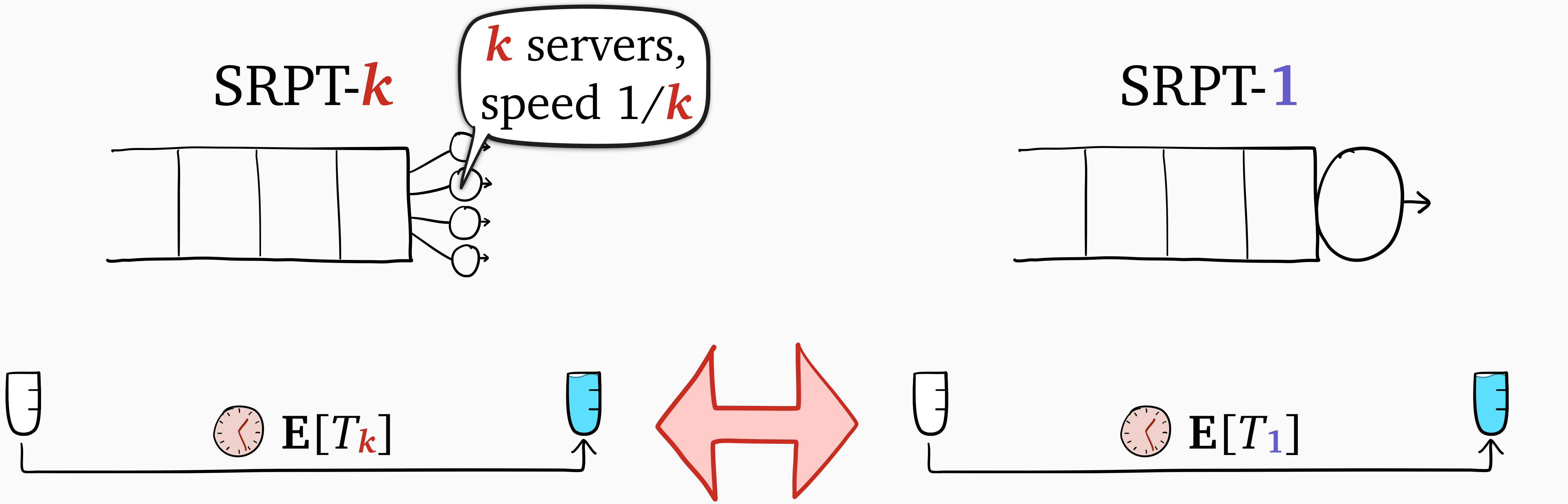


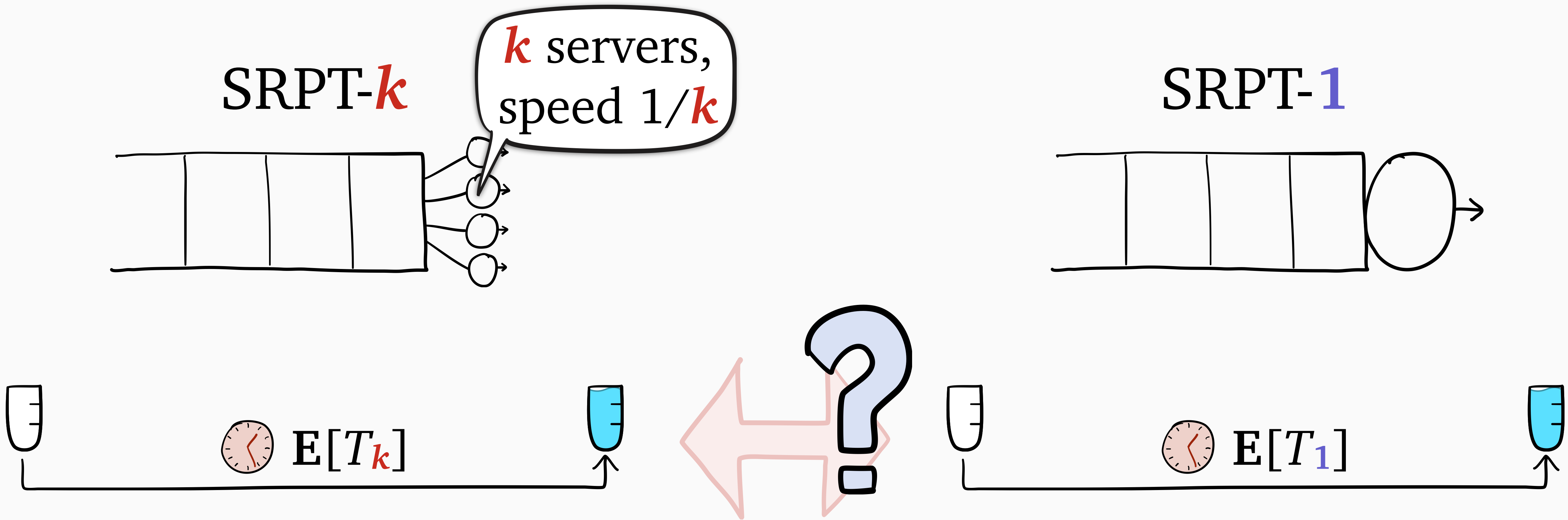
SRPT-k

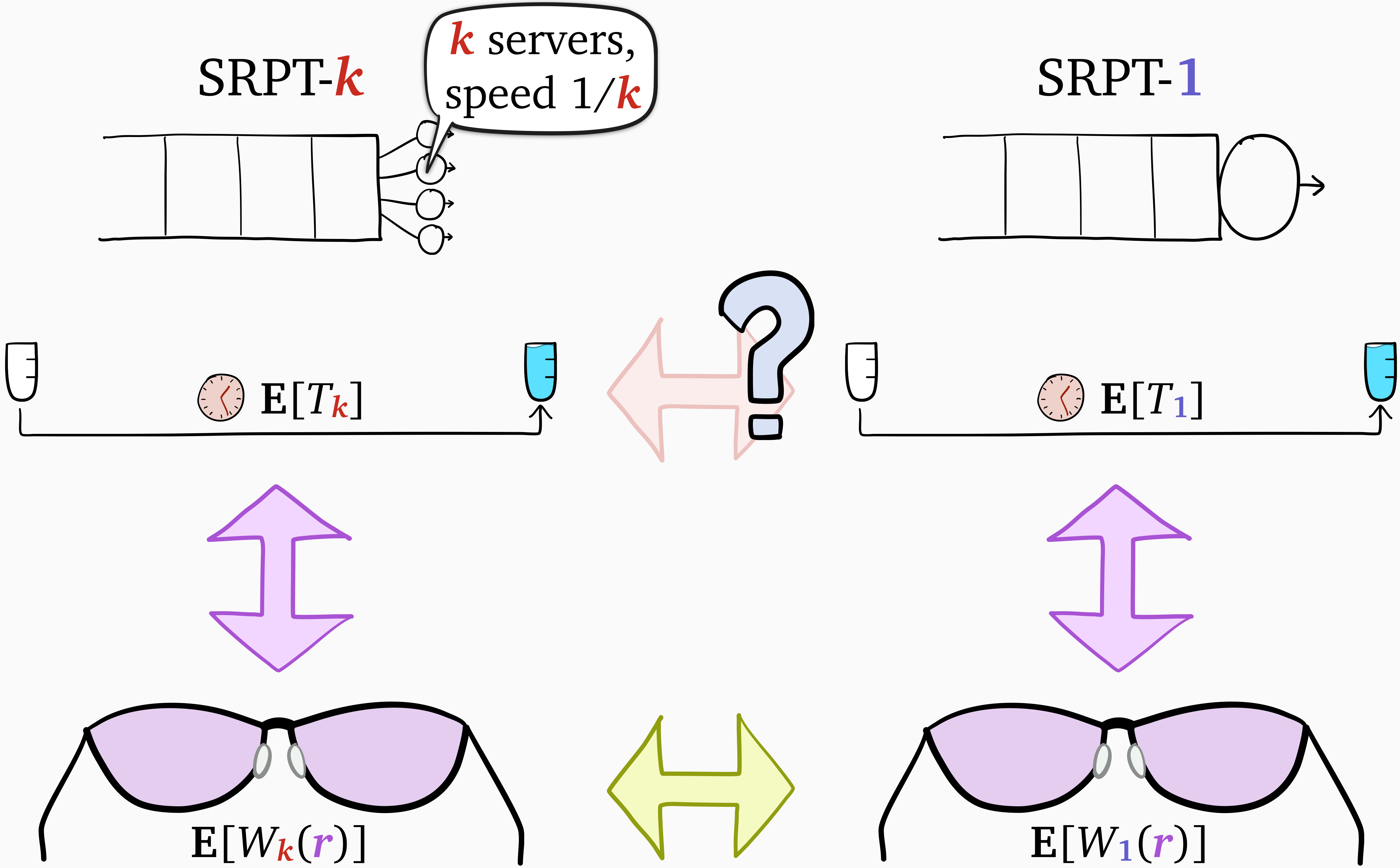


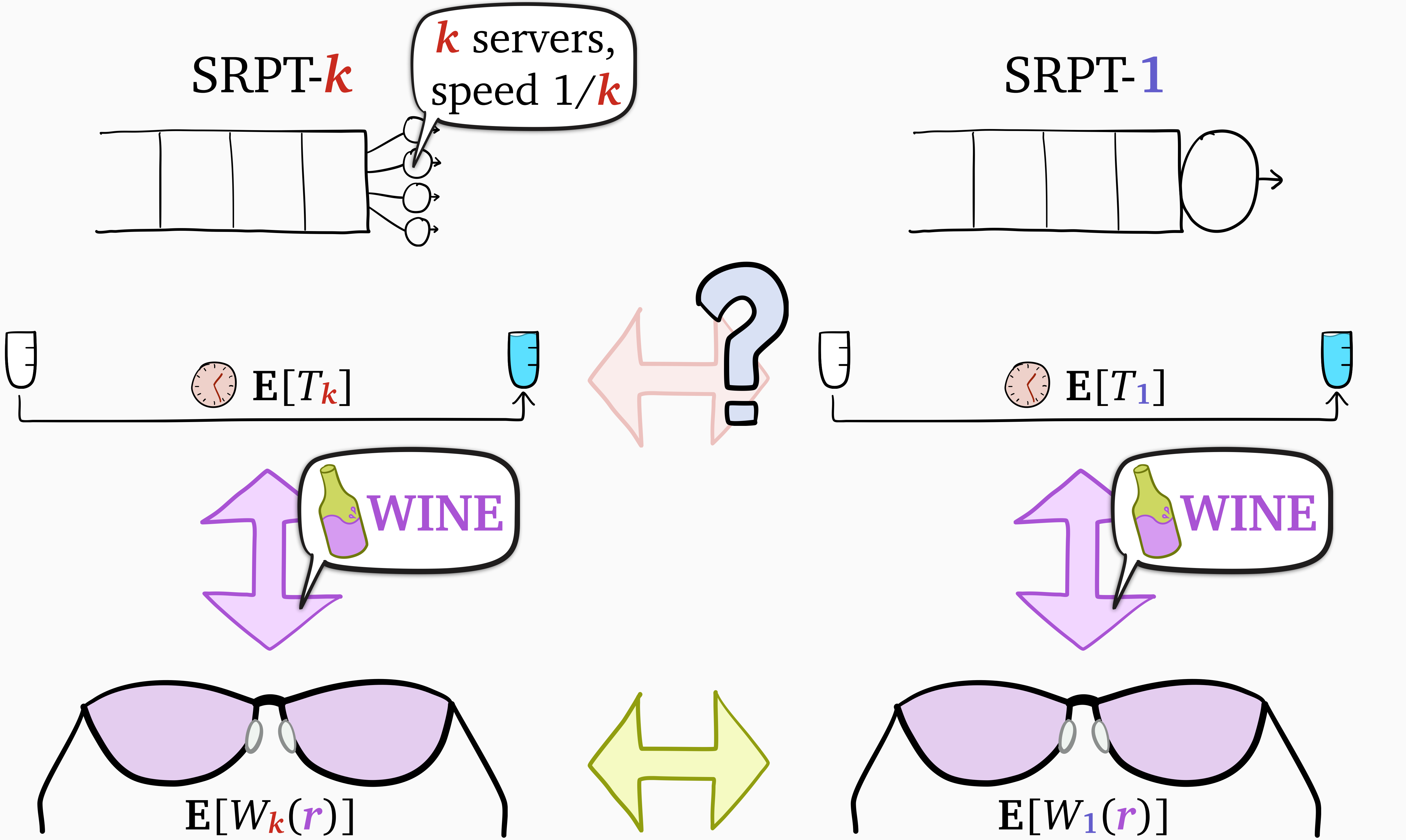
SRPT-1

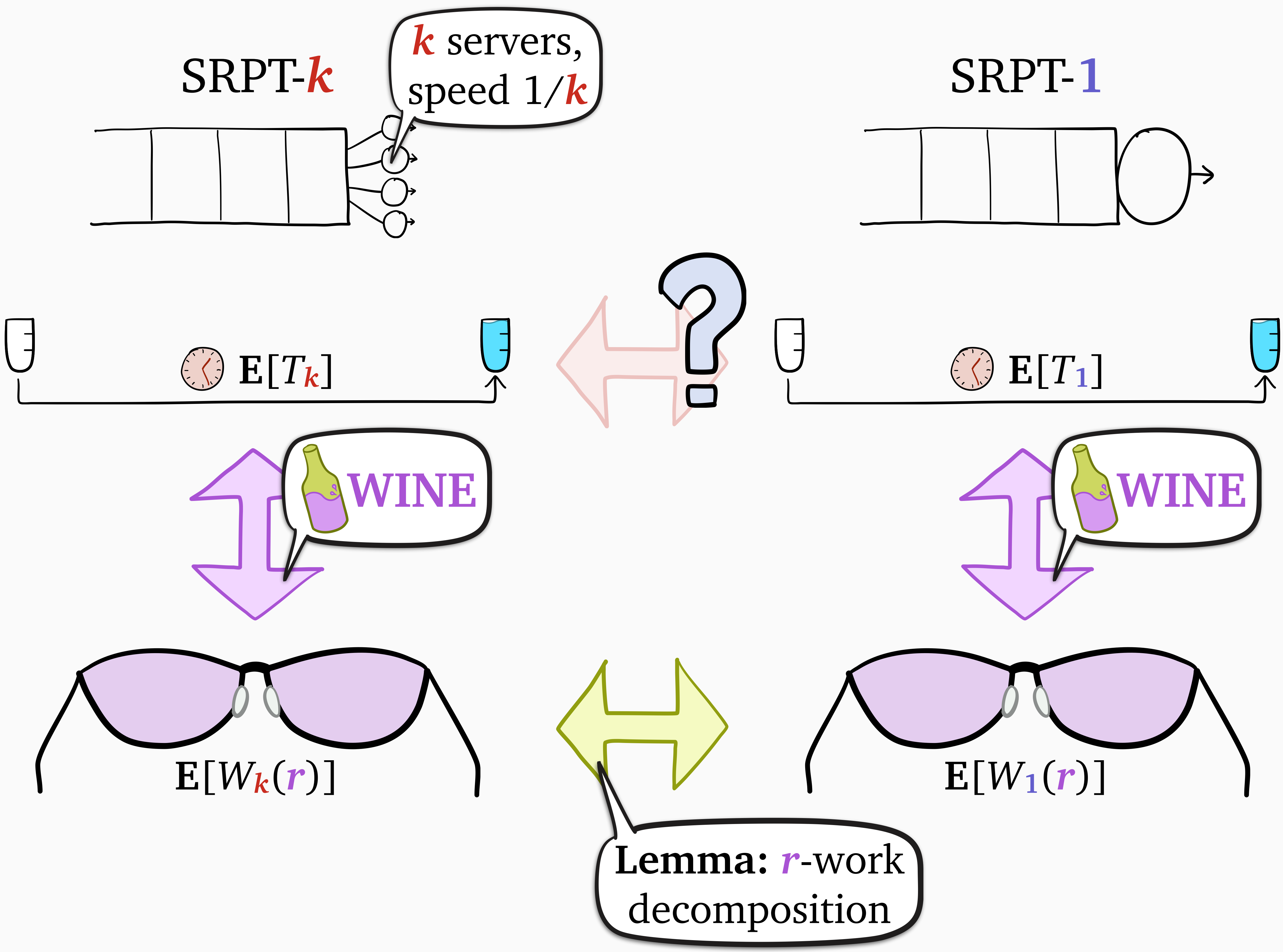


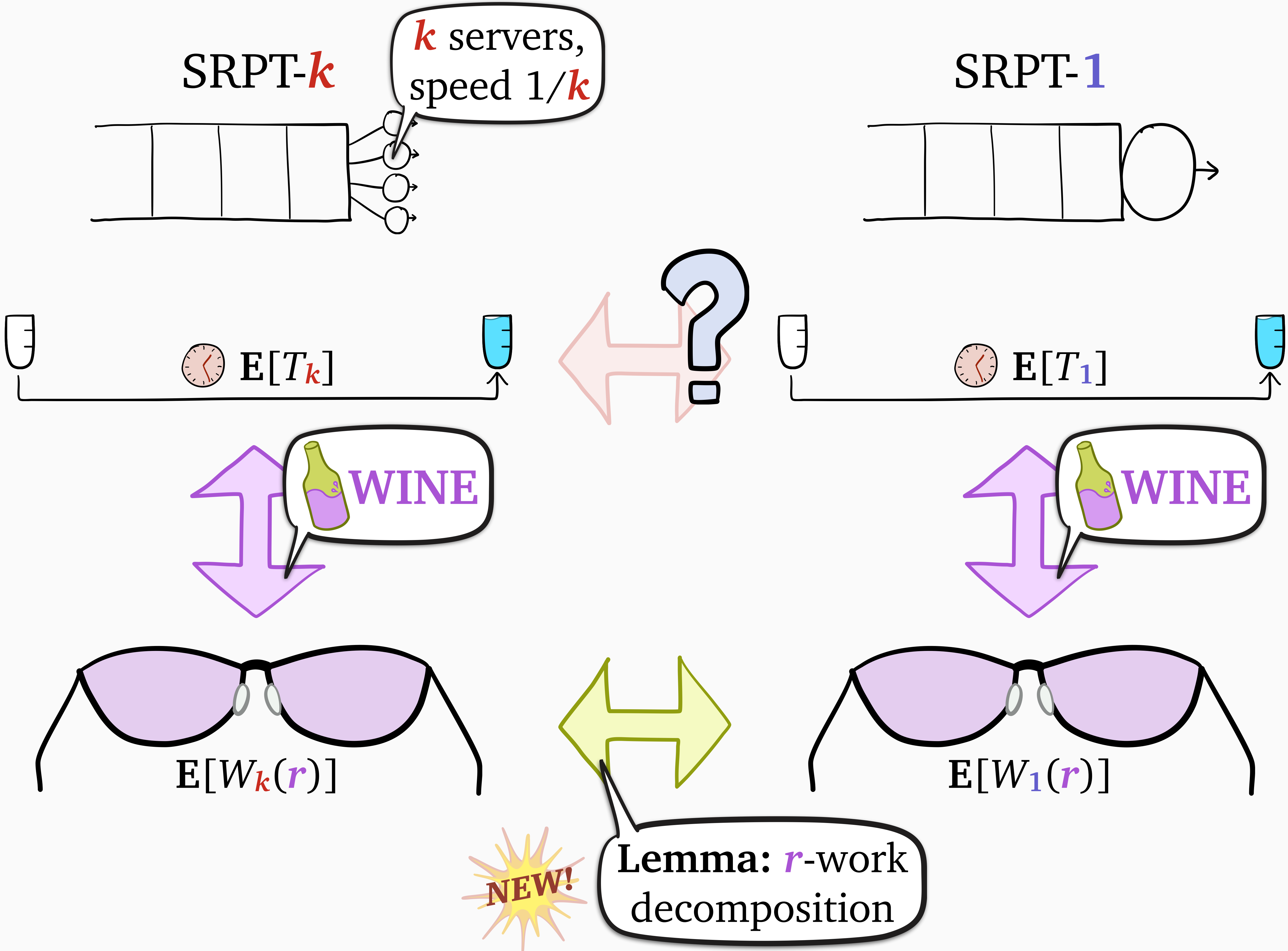

SRPT-k
k servers, speed $1/k$
SRPT-1
$\mathbf{E}[T_k]$
$\mathbf{E}[T_1]$

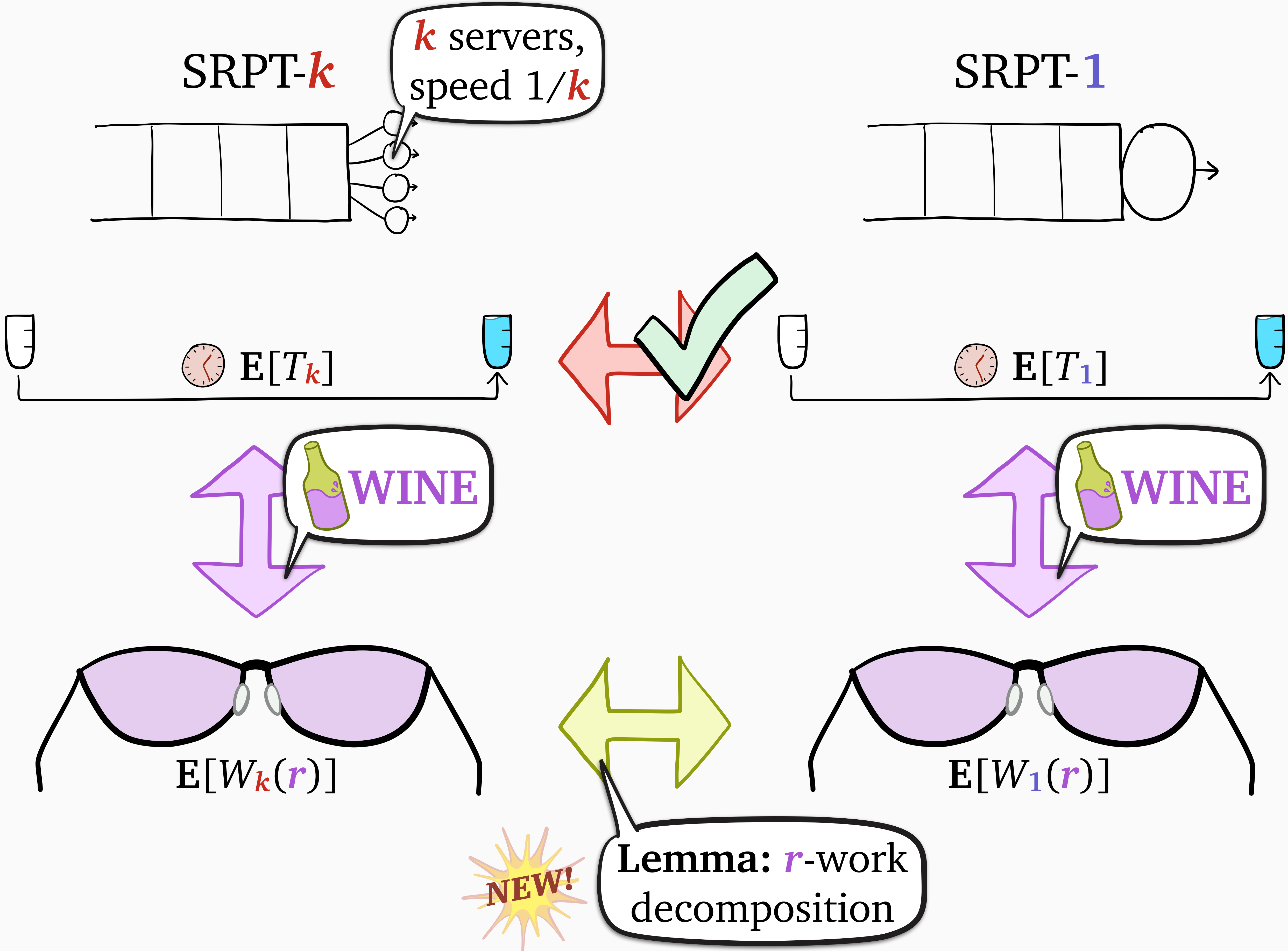

SRPT-k
k servers, speed $1/k$
SRPT-1
$\mathbf{E}[T_k]$
$\mathbf{E}[T_1]$

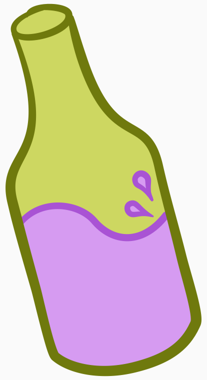

SRPT-k
k servers, speed $1/k$
SRPT-1
$\mathbf{E}[T_k]$
$\mathbf{E}[T_1]$
$\mathbf{E}[W_k(r)]$
$\mathbf{E}[W_1(r)]$


SRPT-k
k servers, speed $1/k$
SRPT-1
$\mathbf{E}[T_k]$
$\mathbf{E}[T_1]$
WINE
WINE
$\mathbf{E}[W_k(r)]$
$\mathbf{E}[W_1(r)]$


SRPT-k
k servers, speed $1/k$
SRPT-1
$\mathbf{E}[T_k]$
$\mathbf{E}[T_1]$
WINE
WINE
$\mathbf{E}[W_k(r)]$
$\mathbf{E}[W_1(r)]$
Lemma: r-work decomposition


SRPT-k
k servers, speed $1/k$
SRPT-1
$\mathbf{E}[T_k]$
$\mathbf{E}[T_1]$
WINE
WINE
$\mathbf{E}[W_k(r)]$
$\mathbf{E}[W_1(r)]$
NEW!
Lemma: r-work decomposition

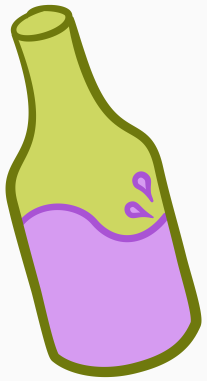

SRPT-k
k servers, speed $1/k$
SRPT-1
$\mathbf{E}[T_k]$
$\mathbf{E}[T_1]$
WINE
WINE
$\mathbf{E}[W_k(r)]$
$\mathbf{E}[W_1(r)]$
NEW!
Lemma: r-work decomposition



Impact of **WINE**

multiserver systems

noisy size estimates



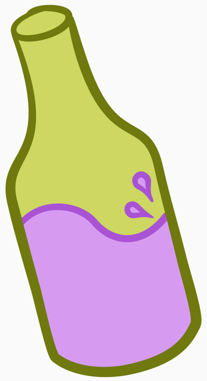
Impact of WINE

multiserver systems

$\mathbf{E}[T]$ bounds for

- SRPT-k
- Gittins-k

noisy size estimates



Impact of WINE

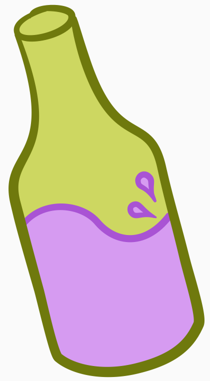
multiserver systems

$\mathbf{E}[T]$ bounds for

- SRPT-k
- Gittins-k

noisy size estimates

compare **noisy-info** r-work to **perfect-info** r-work



Impact of WINE

multiserver systems

noisy size estimates

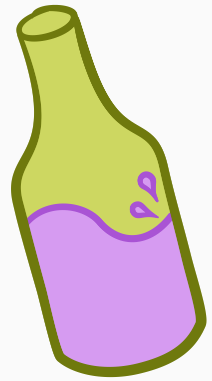
compare **noisy-info** r-work to **perfect-info** r-work

$\mathbf{E}[T]$ bounds for

- SRPT-k
- **Gittins**-k

Theorem:

$$N = \int_0^\infty \frac{W(r)}{r^2} \, \mathrm{d}r$$



Impact of WINE

multiserver systems

E[T] bounds for

- SRPT-k
- Gittins-k

noisy size estimates

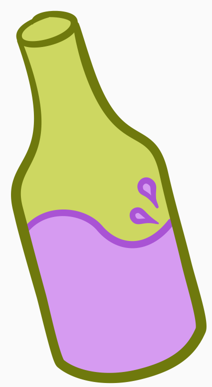
compare **noisy-info** r-work
to **perfect-info** r-work

Theorem:

$$N = \int_0^\infty \frac{W(r)}{r^2}\,\mathrm{d}r$$

References:

- Scully, Grosof, & Harchol-Balter (POMACS 2020 / SIGMETRICS 2021)
- Scully & Harchol-Balter (WiOpt 2021)
- Scully, Grosof, & Mitzenmacher, (ITCS 2022)

 Impact of **WINE**

multiserver systems

E[T] bounds for

- SRPT-k
- **Gittins**-k

noisy size estimates

compare **noisy-info** r-work
to **perfect-info** r-work

Theorem:

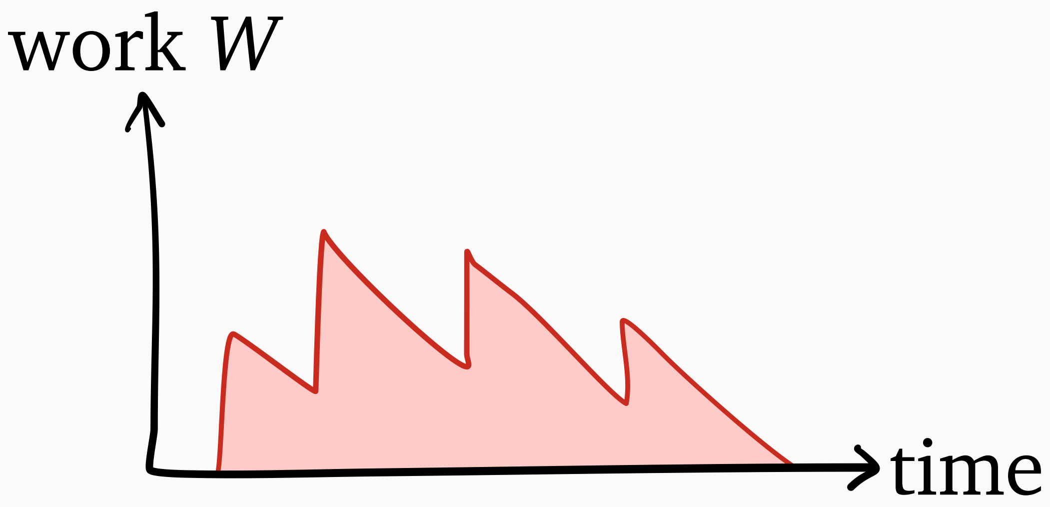
$$N = \int_0^\infty \frac{W(r)}{r^2} \, \mathrm{d}r$$

r-work decomposition

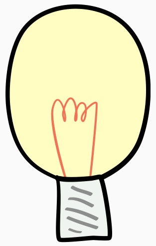
SRPT-k and **Gittins**-k
E[T] bounds

noisy size estimates

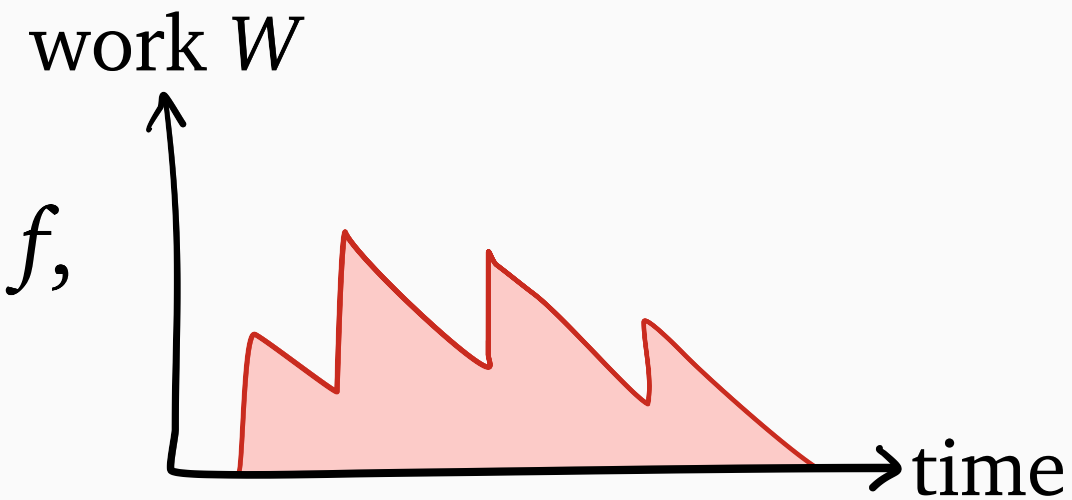
Work (and *r*-work) decomposition



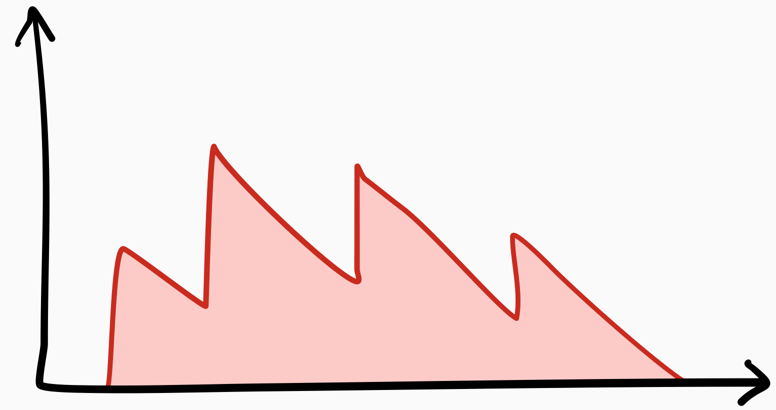
Work (and *r*-work) decomposition



In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time



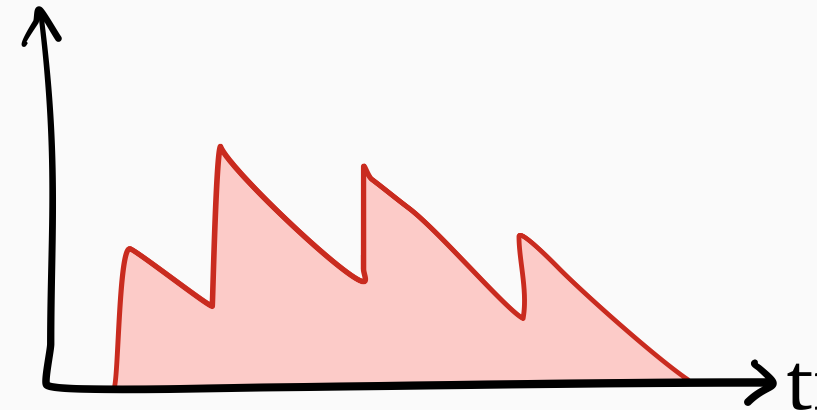
Work (and *r*-work) decomposition



In steady-state system, for any f,
$\mathbf{E}[f(W)]$ constant w.r.t. time

we use
$f(w) = w^2$

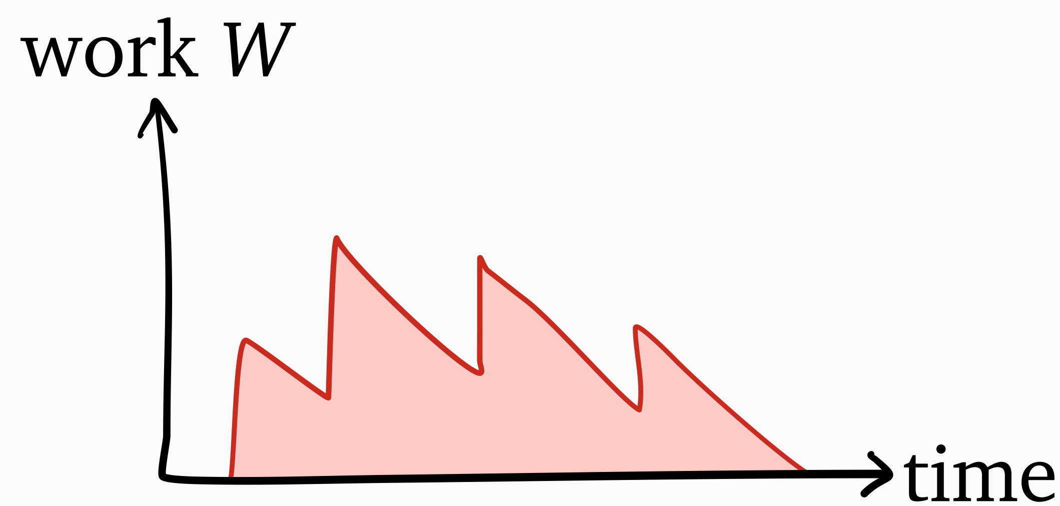
Work (and *r*-work) decomposition



$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

Work (and r-work) decomposition



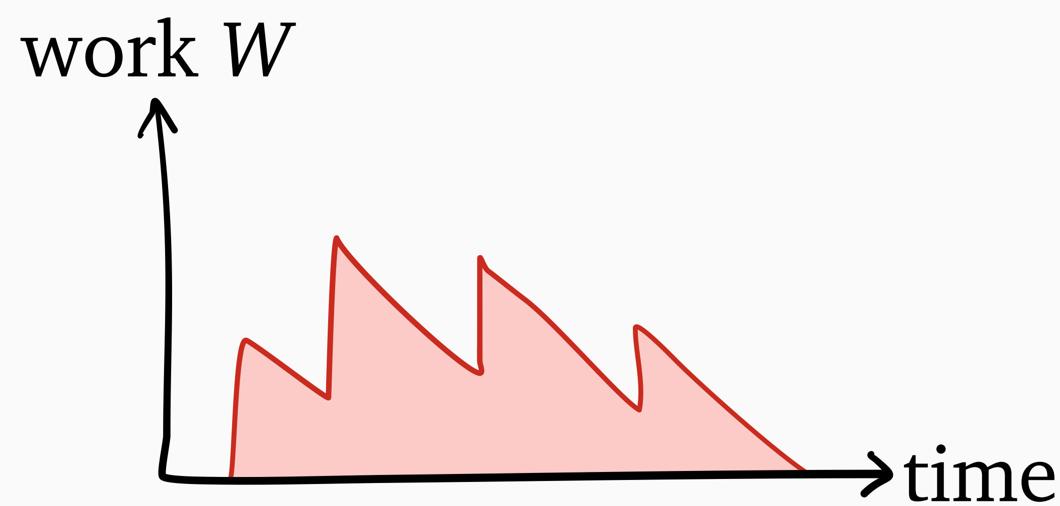
B = service rate, a.k.a. fraction of servers busy

$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

Work (and *r*-work) decomposition

B = service rate, a.k.a. fraction of servers busy



$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

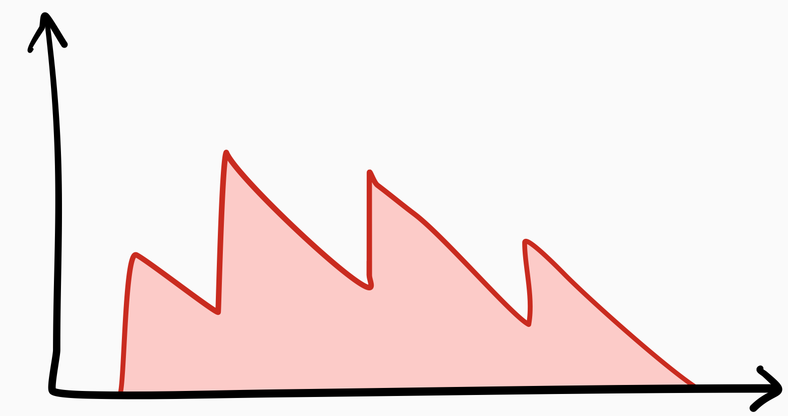
$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

Work (and *r*-work) decomposition

B = service rate, a.k.a. fraction of servers busy

$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

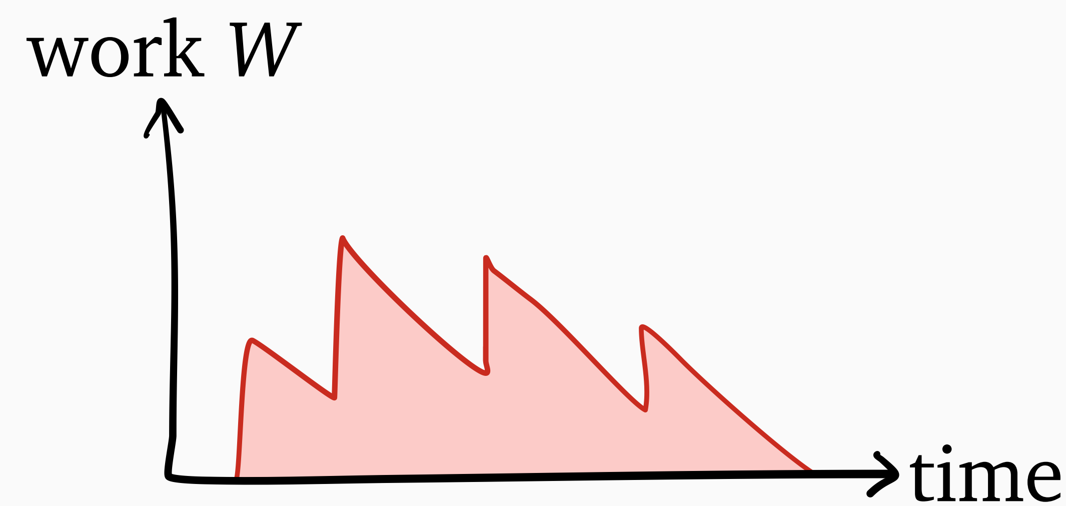


if **single server**: $(1-B)W = 0$

$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

Work (and r-work) decomposition

B = service rate, a.k.a. fraction of servers busy



$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

if **single server**: $(1-B)W = 0$

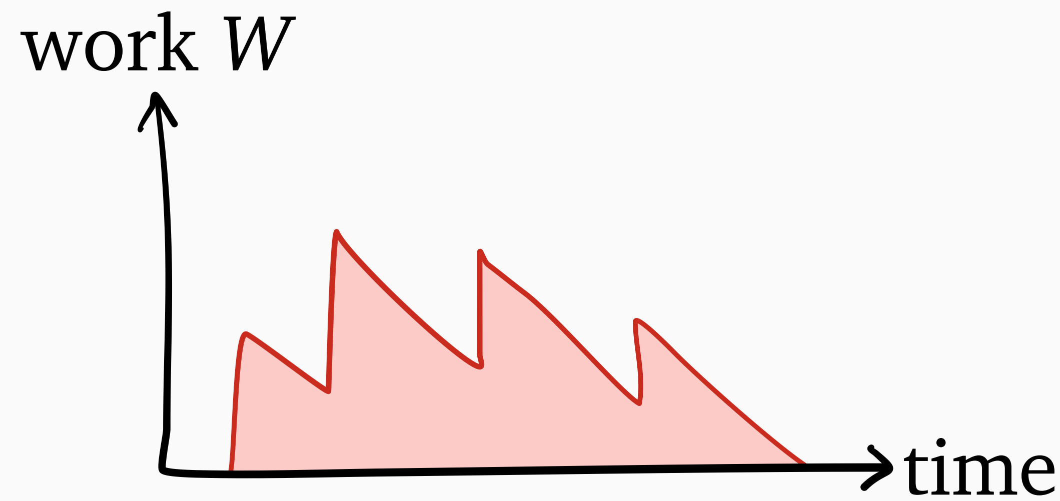
$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

Lemma:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

Work (and r-work) decomposition

B = service rate, a.k.a. fraction of servers busy



$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda \mathbf{E}[(W+S)^2 - W^2]$$

if **single server**: $(1-B)W = 0$

$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

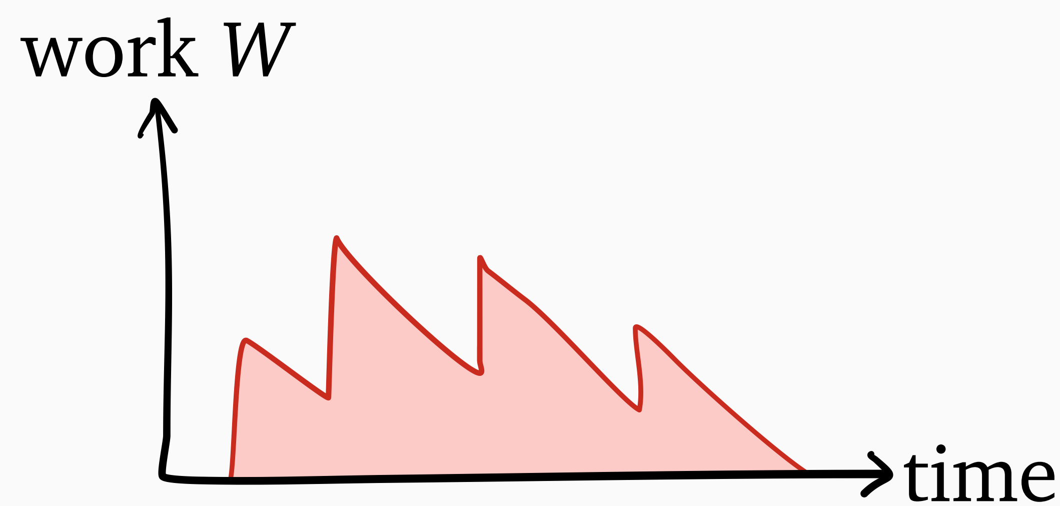
newish

Lemma:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

Work (and *r*-work) decomposition

B = service rate, a.k.a. fraction of servers busy



$$\mathbf{E}[W^2 \text{ decrease rate}] = 2\mathbf{E}[BW]$$

$$\mathbf{E}[W^2 \text{ increase rate}] = \lambda\mathbf{E}[(W+S)^2 - W^2]$$

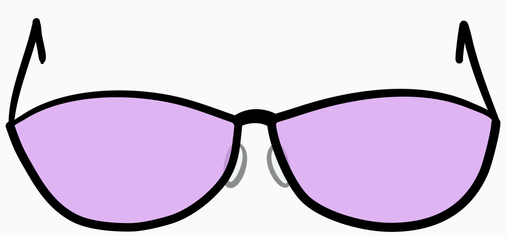
if **single server**: $(1-B)W = 0$

newish

$$\mathbf{E}[W] = \frac{\frac{\lambda}{2}\mathbf{E}[S^2]}{1-\rho} + \frac{\mathbf{E}[(1-B)W]}{1-\rho}$$

Lemma:

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$



Similar story with *r*-work NEW!

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

Suppose $S \leq s_{\max}$ with probability 1

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

Suppose $S \leq s_{\max}$ with probability 1

$\leq (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\leq (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\leq (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

$$\leq \mathbf{E}[W_1] + (k-1)s_{\max}$$

Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\leq (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

$$\leq \mathbf{E}[W_1] + (k-1)s_{\max}$$

"work of $\leq k-1$ jobs"

Suppose $S \le s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\le (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

$$\le \mathbf{E}[W_1] + (k-1)s_{\max}$$

"work of $\le k-1$ jobs"

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of } k-1 \text{ jobs”}$$

Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\leq (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

$$\leq \mathbf{E}[W_1] + (k-1)s_{\max}$$

"work of $\leq k-1$ jobs"

Single job's r-work is at most r

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of } k-1 \text{ jobs”}$$

Suppose $S \le s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\le (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1-B_k)W_k]}{1-\rho}$$

$$\le \mathbf{E}[W_1] + (k-1)s_{\max}$$

"work of $\le k-1$ jobs"

Single job's r-work is at most r

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of } k-1 \text{ jobs”}$$

$$\le \mathbf{E}[W_1] + (k-1)r$$

Suppose $S \leq s_{\max}$ with probability 1

$\mathbf{E}[B] = \rho$

$\leq (k-1)s_{\max}$

$$\mathbf{E}[W_k] = \mathbf{E}[W_1] + \frac{\mathbf{E}[(1 - B_k)W_k]}{1 - \rho}$$

$$\leq \mathbf{E}[W_1] + (k-1)s_{\max}$$

"work of $\leq k - 1$ jobs"

Single job's r-work is at most r

$$\mathbf{E}[W_k(r)] = \mathbf{E}[W_1(r)] + \text{“}r\text{-work of } k-1 \text{ jobs”}$$

$$\leq \mathbf{E}[W_1] + (k-1)r$$

can improve

SRPT-k and Gittins-k $\mathbf{E}[T]$ bounds

Theorem: for SRPT and Gittins,

$$\mathbf{E}[T_k] \le \mathbf{E}[T_1] + (k-1)\cdot O\left(\log \frac{1}{1-\rho}\right)$$

SRPT-k and Gittins-k $\mathbf{E}[T]$ bounds

Theorem: for SRPT and Gittins,

$$\mathbf{E}[T_k] \leq \mathbf{E}[T_1] + \underbrace{(k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)}_{o(\mathbf{E}[T_1])}$$

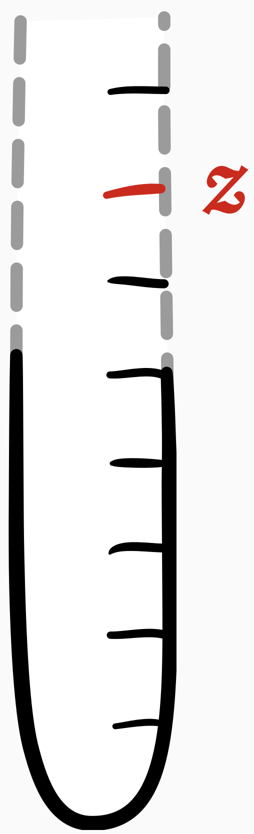
SRPT-k and Gittins-k $\mathbf{E}[T]$ bounds

Theorem: for SRPT and Gittins,

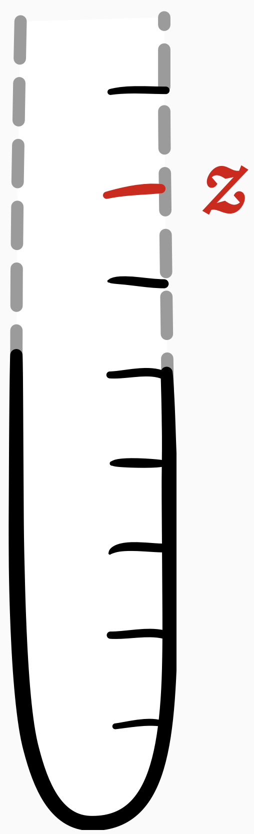
$$\mathbf{E}[T_k] \le \mathbf{E}[T_1] + \underbrace{(k-1) \cdot O\left(\log \frac{1}{1-\rho}\right)}_{o(\mathbf{E}[T_1])}$$

Corollary: SRPT and Gittins minimize $\mathbf{E}[T]$ in heavy traffic (in their respective settings)

Noisy size estimates



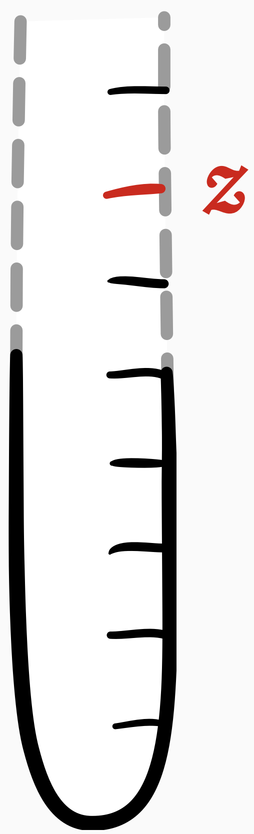
Noisy size estimates



Model: (β, α)-bounded noise

true size $s \quad \Rightarrow \quad$ **estimated** size $z \in [\beta s, \alpha s]$

Noisy size estimates

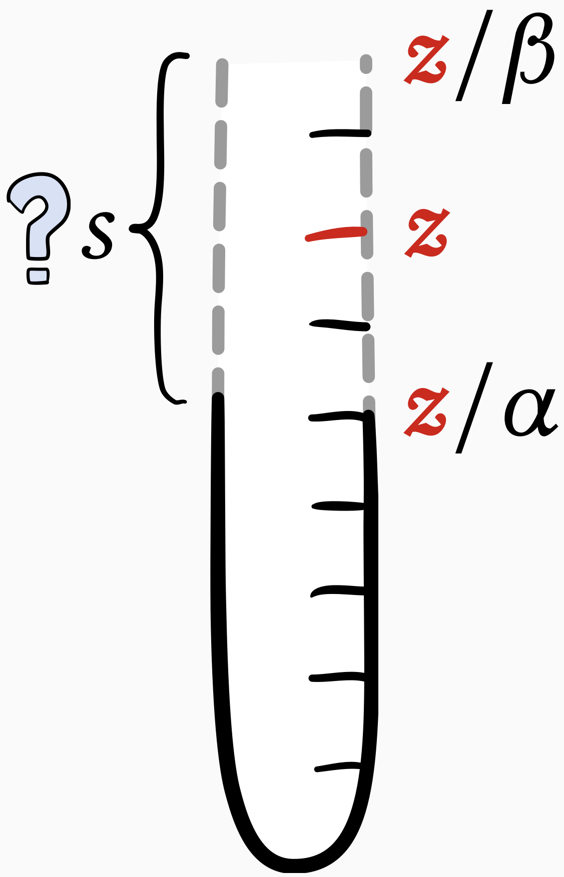


below above

Model: (β, α)-bounded noise

true size s $\Rightarrow$ **estimated** size $z \in [\beta s, \alpha s]$

Noisy size estimates

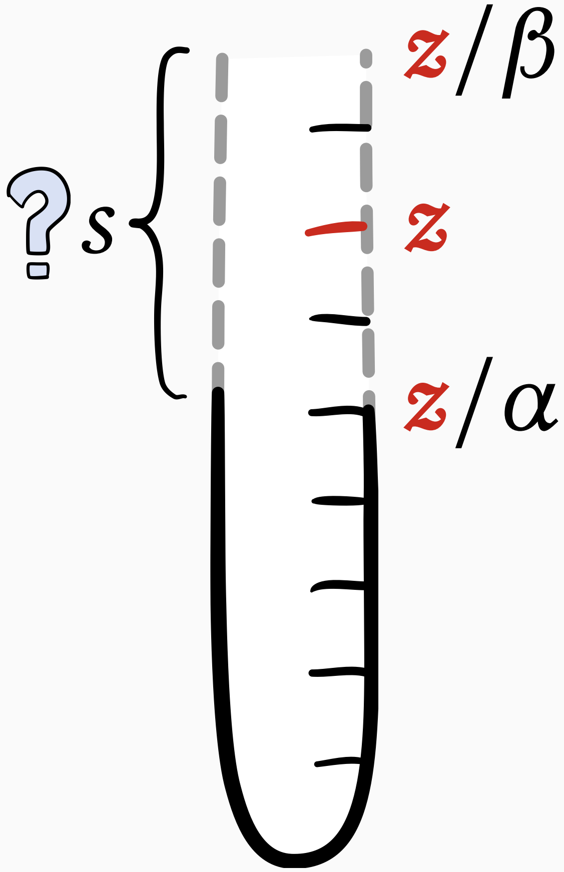


below above

Model: (β, α)-bounded noise

true size $s \quad \Rightarrow \quad$ **estimated** size $z \in [\beta s, \alpha s]$

Noisy size estimates



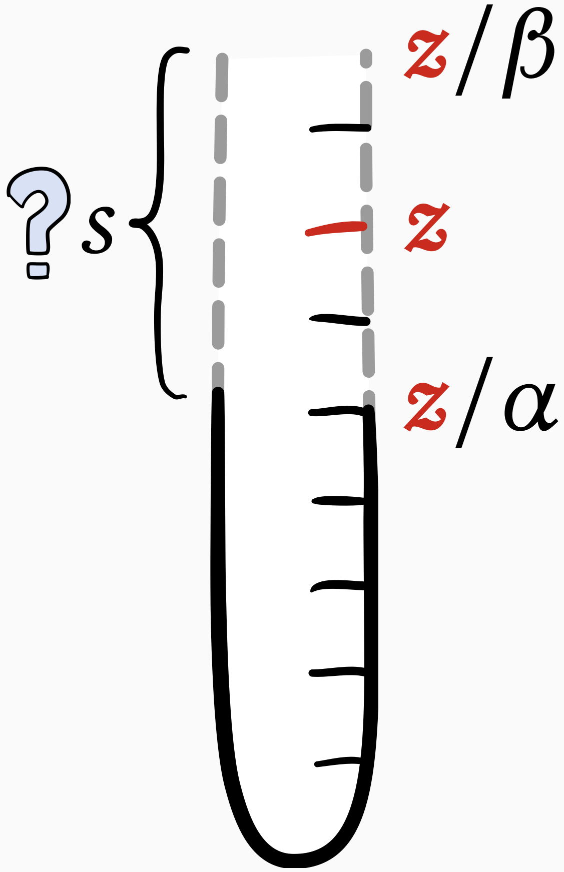
below above

Model: (β, α)-bounded noise

true size s $\Rightarrow$ **estimated** size $z \in [\beta s, \alpha s]$

Generated by *joint distribution* (S, Z)

Noisy size estimates



below above

Model: (β, α)-bounded noise

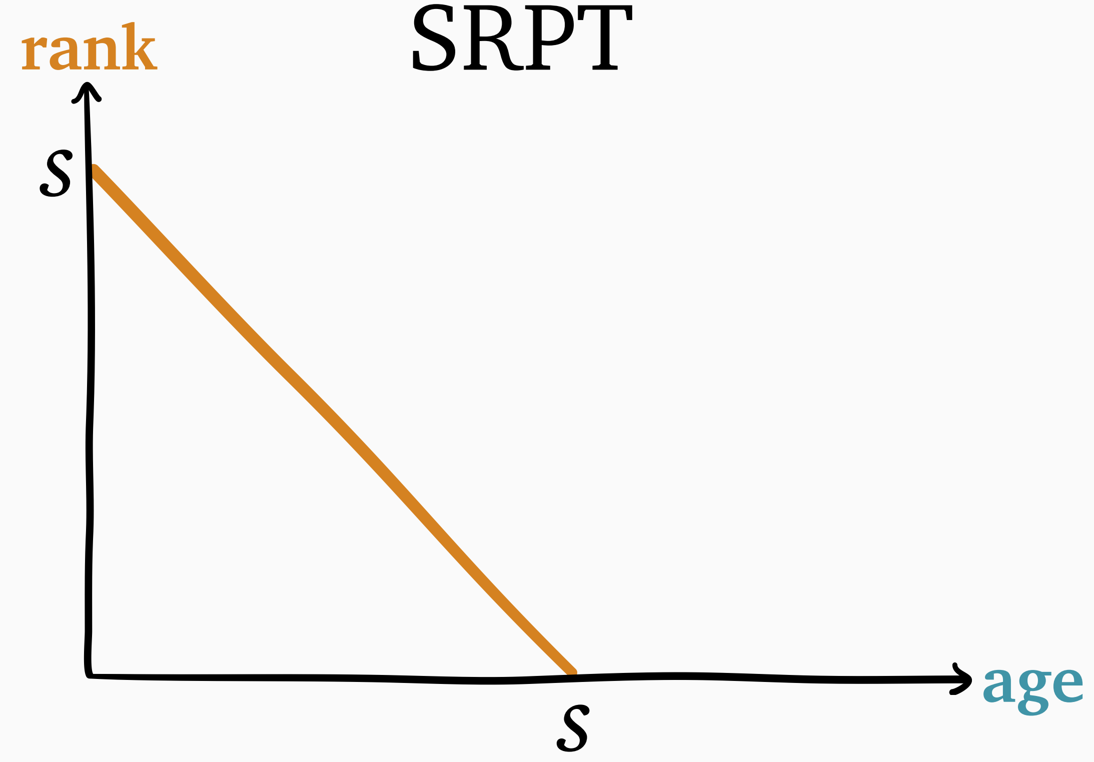
true size s $\Rightarrow$ **estimated** size $z \in [\beta s, \alpha s]$

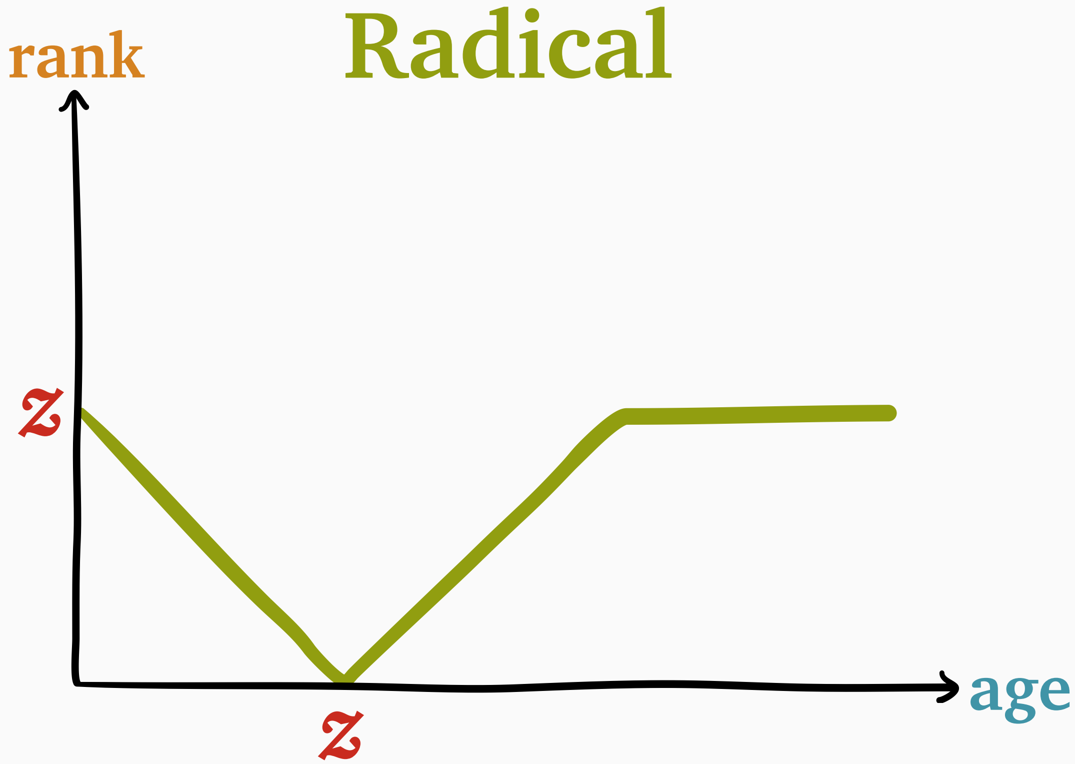
Generated by *joint distribution* (S, Z)

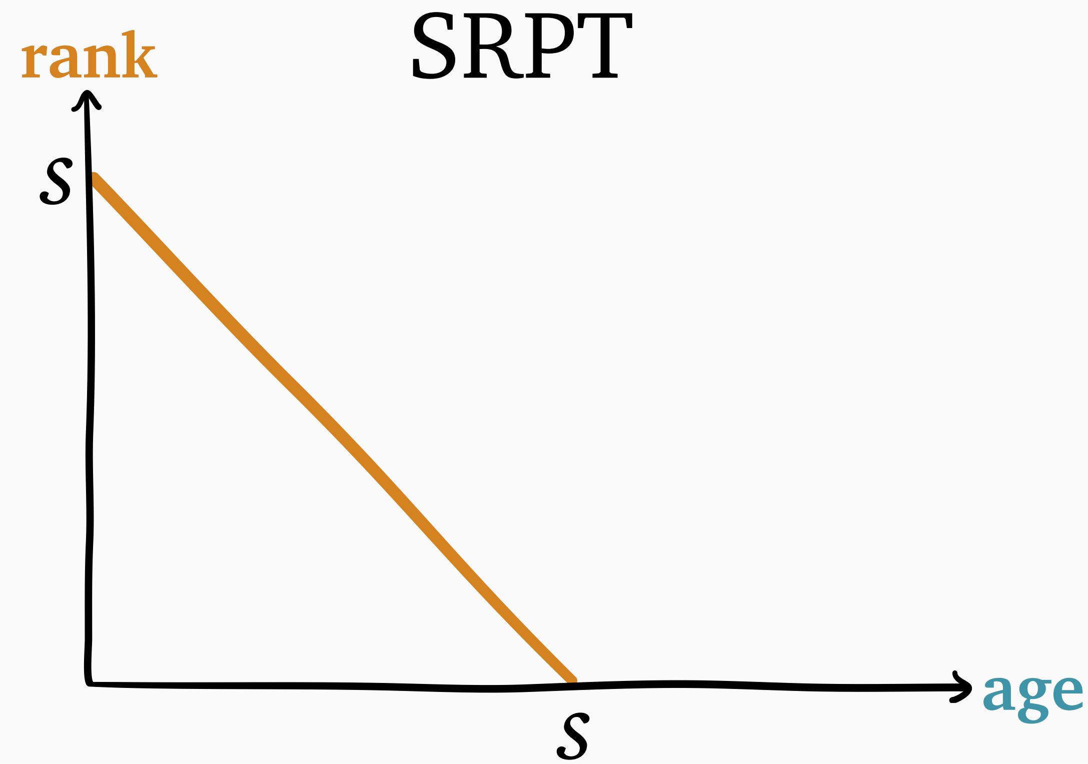
Goal: design a policy with "good" $\mathbf{E}[T]$ for

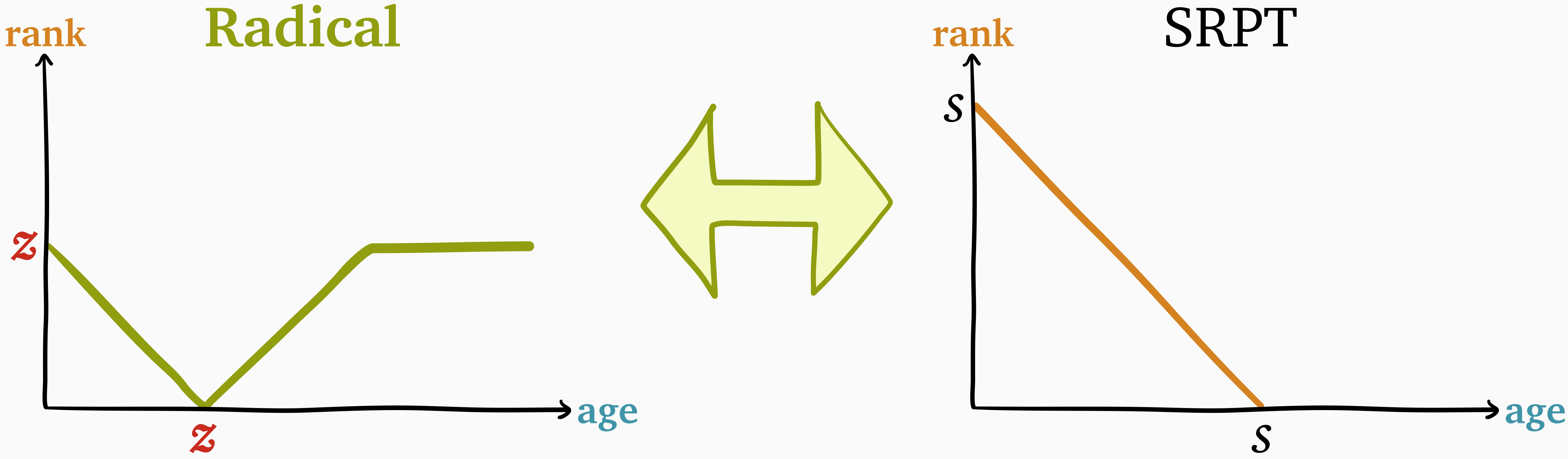
- *any* joint distribution (S, Z)
- *any* values of α, β

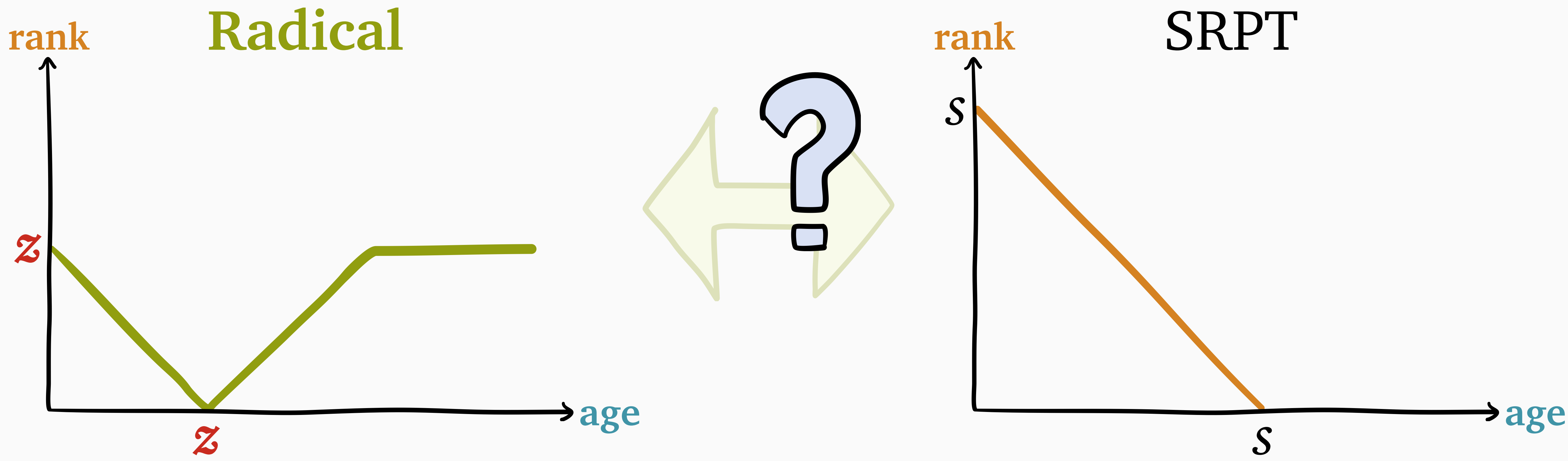
SRPT



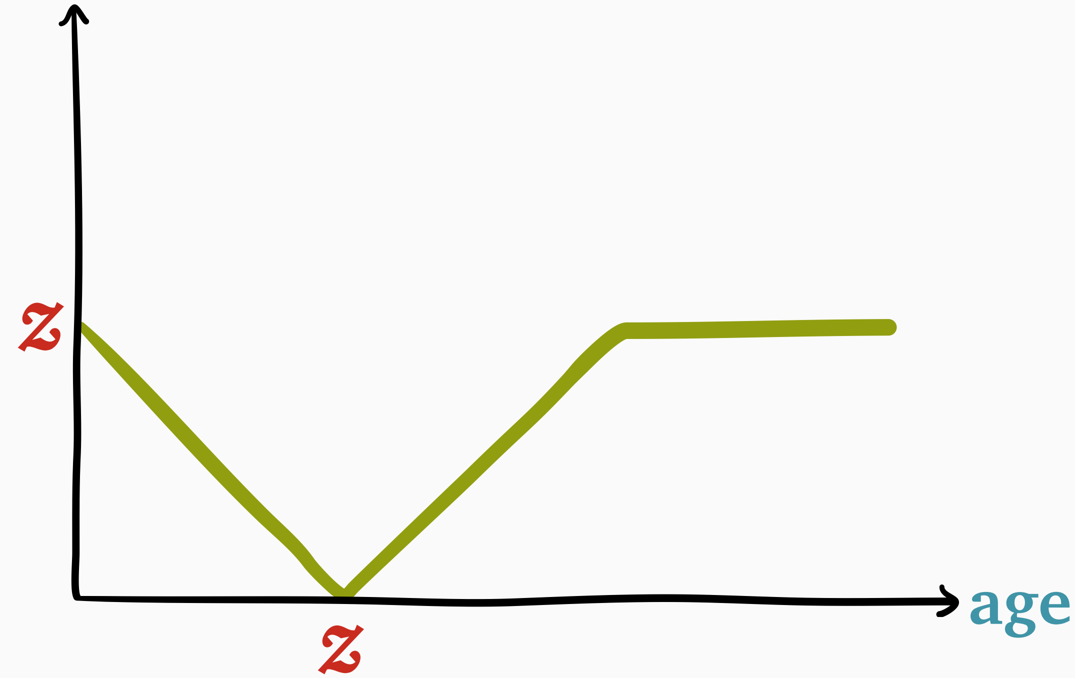

Radical
rank
z
z
age

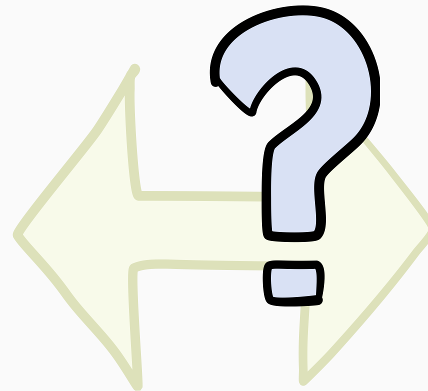

SRPT
rank
s
s
age


Radical
rank
z
z
age
SRPT
rank
s
s
age

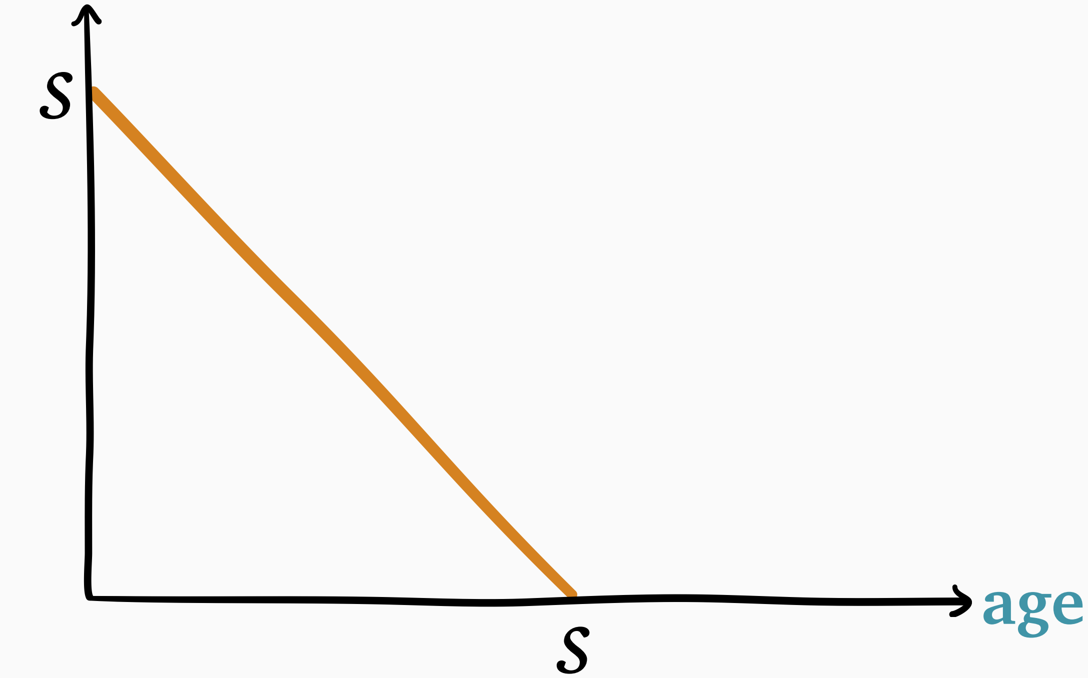

Radical
rank
z
z
age
SRPT
rank
s
s
age

Radical

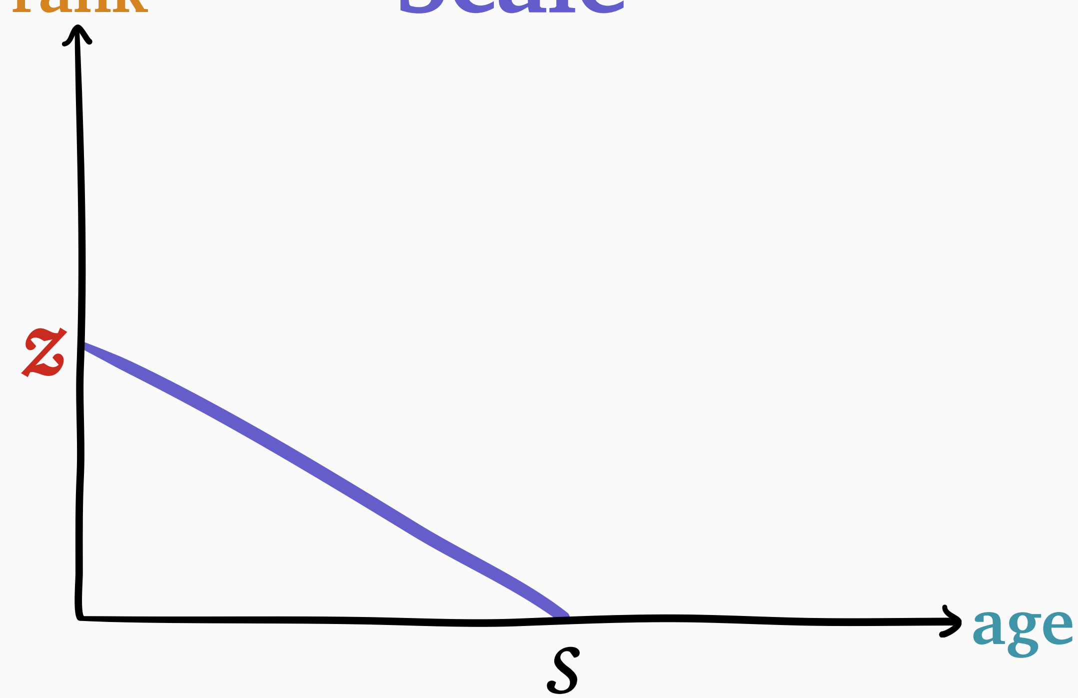


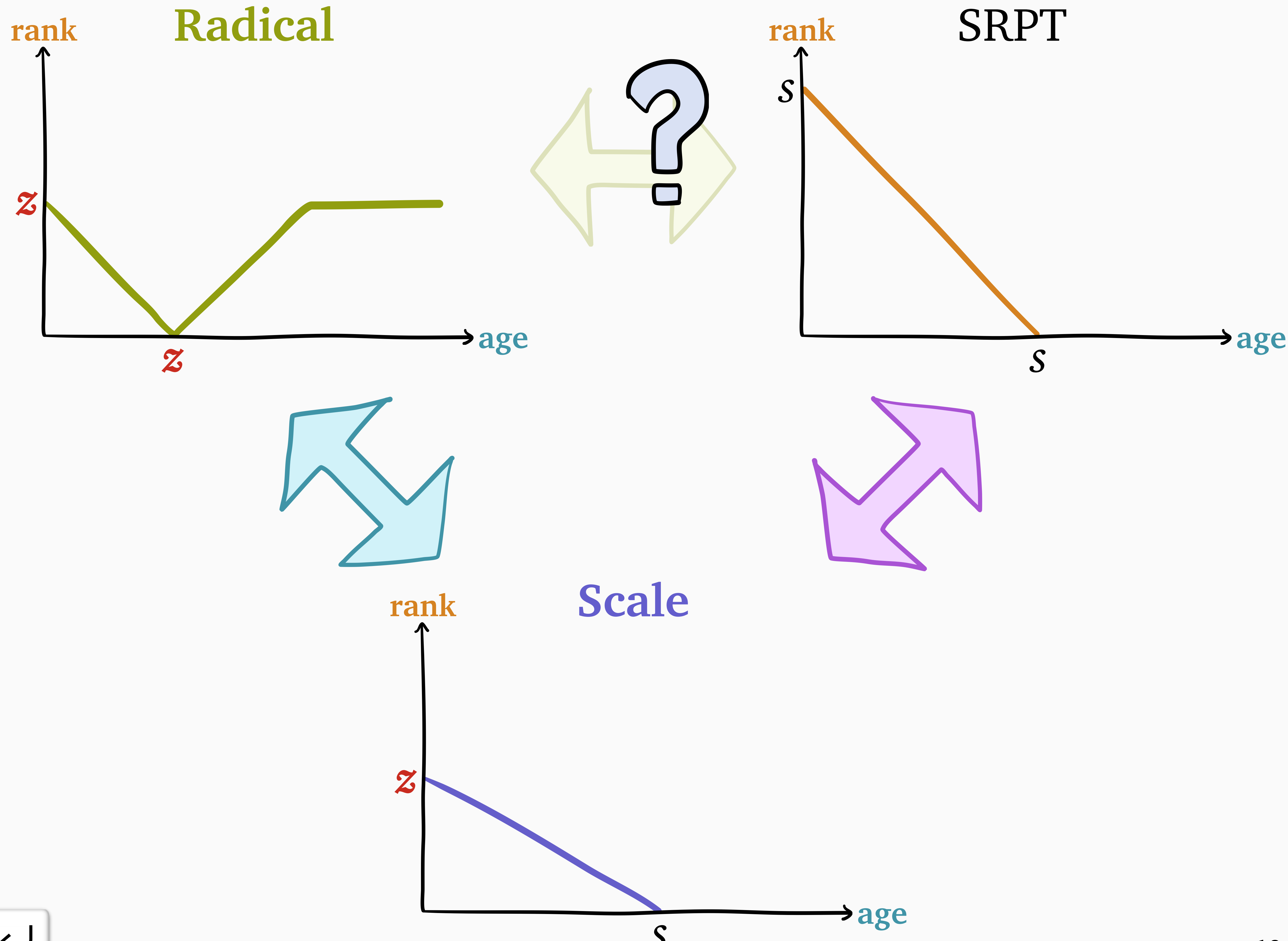


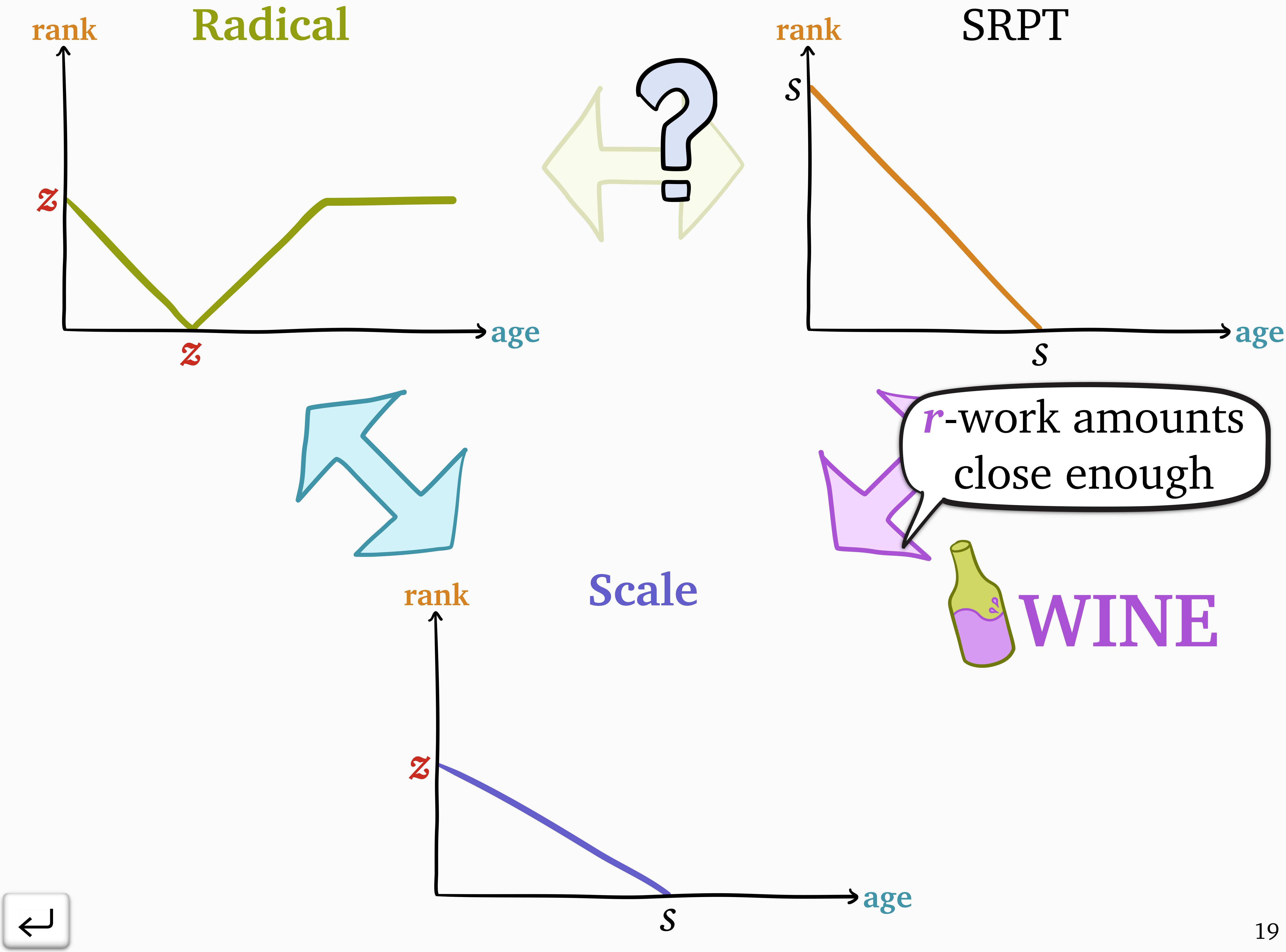
SRPT



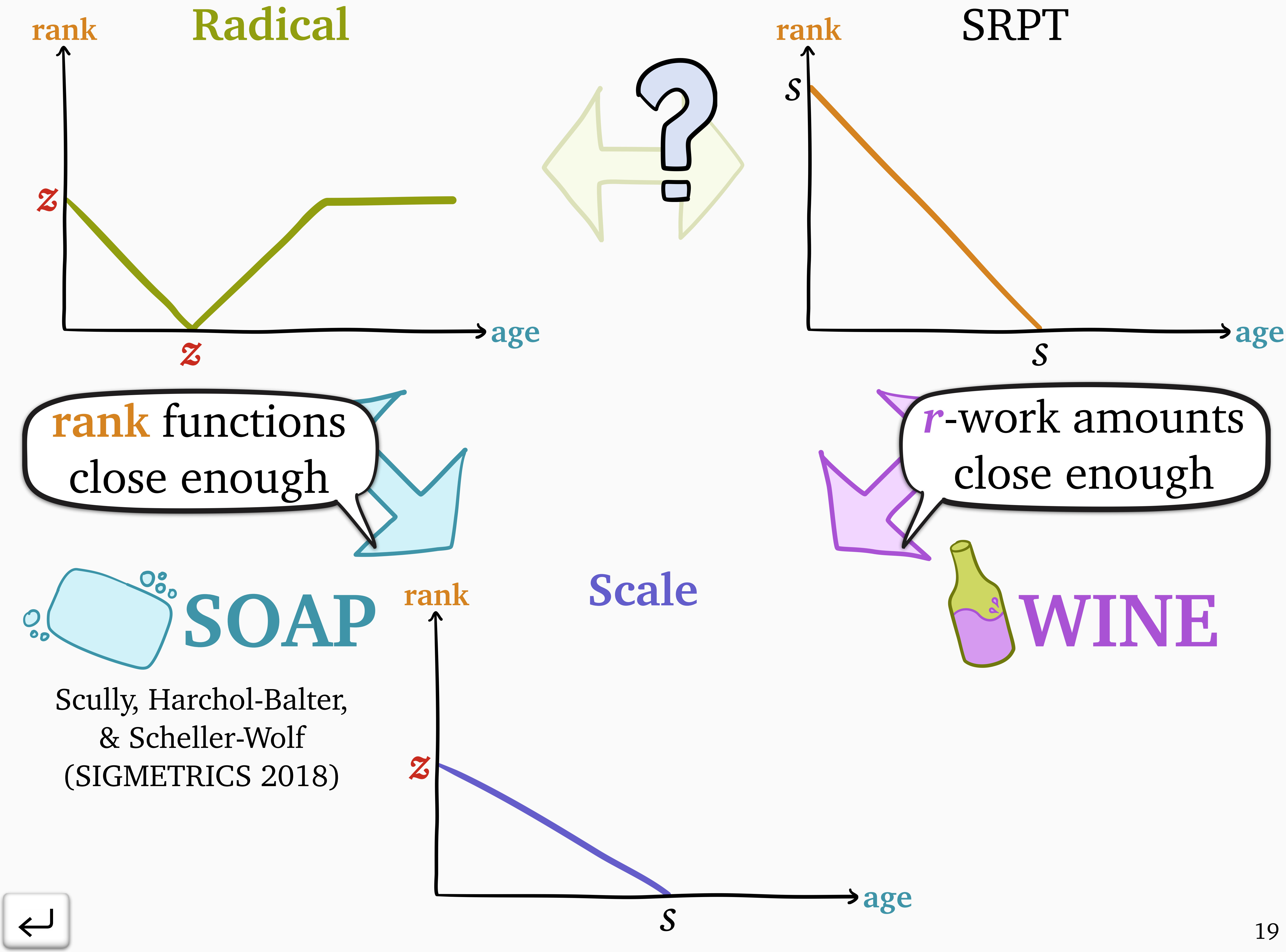
Scale



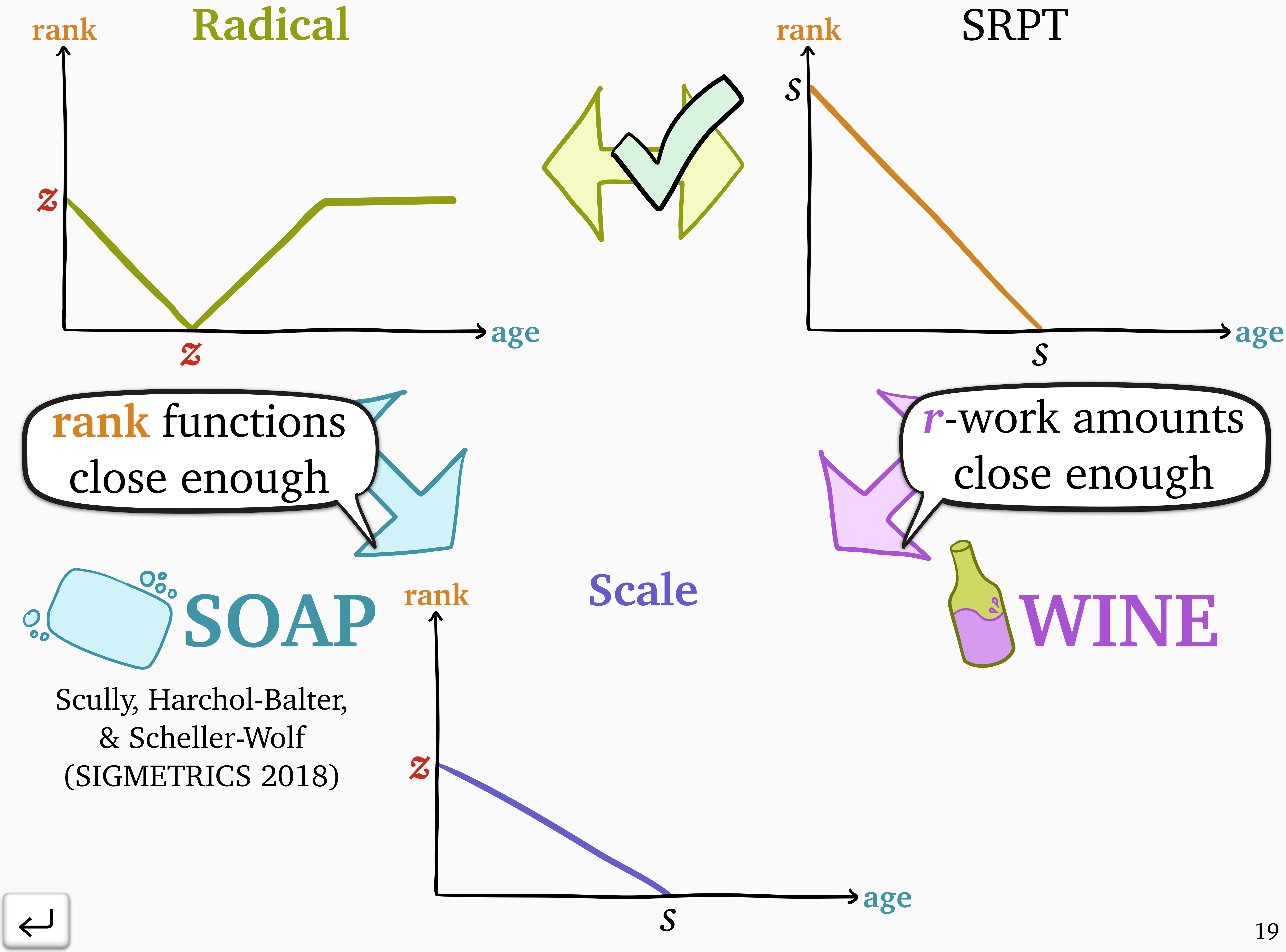

Radical
rank
z
z
age
SRPT
rank
s
s
age
Scale
rank
z
s
age

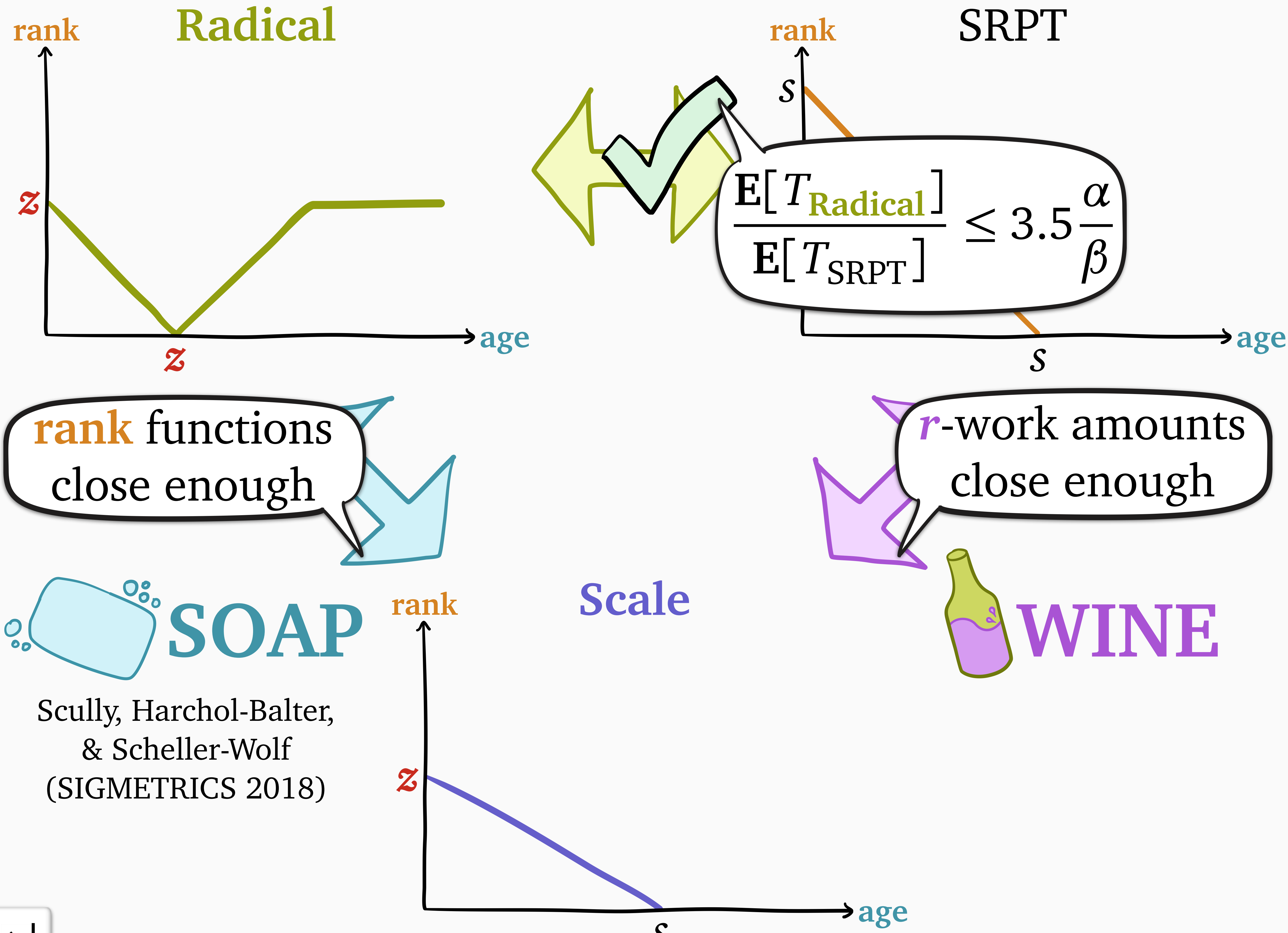


Radical
rank
age
z
z
SRPT
rank
age
s
s
?
r-work amounts close enough
WINE
Scale
rank
age
z
s



Radical
rank
age
z
z
SRPT
rank
age
s
s
?
rank functions close enough
r-work amounts close enough
SOAP
Scully, Harchol-Balter, & Scheller-Wolf (SIGMETRICS 2018)
Scale
rank
age
z
s
WINE


Radical
rank
z
z
age
SRPT
rank
s
s
age
rank functions close enough
r-work amounts close enough
SOAP
Scully, Harchol-Balter, & Scheller-Wolf (SIGMETRICS 2018)
Scale
rank
z
s
age
WINE



Radical
rank
age
z
z
SRPT
rank
s
s
age
$\frac{\mathbf{E}[T_{\text{Radical}}]}{\mathbf{E}[T_{\text{SRPT}}]} \le 3.5\frac{\alpha}{\beta}$
rank functions close enough
r-work amounts close enough
SOAP
Scully, Harchol-Balter, & Scheller-Wolf (SIGMETRICS 2018)
Scale
rank
z
s
age
WINE

SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\mathbf{Scale}}(\boldsymbol{r})] \leq \mathbf{E}\left[W_{\mathbf{SRPT}}\left(\frac{\alpha}{\beta}\boldsymbol{r}\right)\right]$$

Key steps:

SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\mathrm{SRPT}}(r)] \le \mathbf{E}[W_{\mathrm{Scale}}(r)] \le \mathbf{E}\left[W_{\mathrm{SRPT}}\left(\frac{\alpha}{\beta} r\right)\right]$$

Key steps:

SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\mathrm{SRPT}}(r)] \le \mathbf{E}[W_{\mathrm{Scale}}(r)] \le \mathbf{E}\left[W_{\mathrm{SRPT}}\left(\frac{\alpha}{\beta} r\right)\right]$$

Key steps:

1. SRPT minimizes mean r-work

SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\mathrm{SRPT}}(r)] \leq \mathbf{E}[W_{\mathrm{Scale}}(r)] \leq \mathbf{E}\left[W_{\mathrm{SRPT}}\left(\frac{\alpha}{\beta} r\right)\right]$$

Key steps:

1. SRPT minimizes mean r-work

SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\mathrm{SRPT}}(r)] \le \mathbf{E}[W_{\mathrm{Scale}}(r)] \le \mathbf{E}\left[W_{\mathrm{SRPT}}\left(\frac{\alpha}{\beta}r\right)\right]$$

Key steps:

1. SRPT minimizes mean r-work
2. Scale minimizes mean **noise-scaled**-r-work

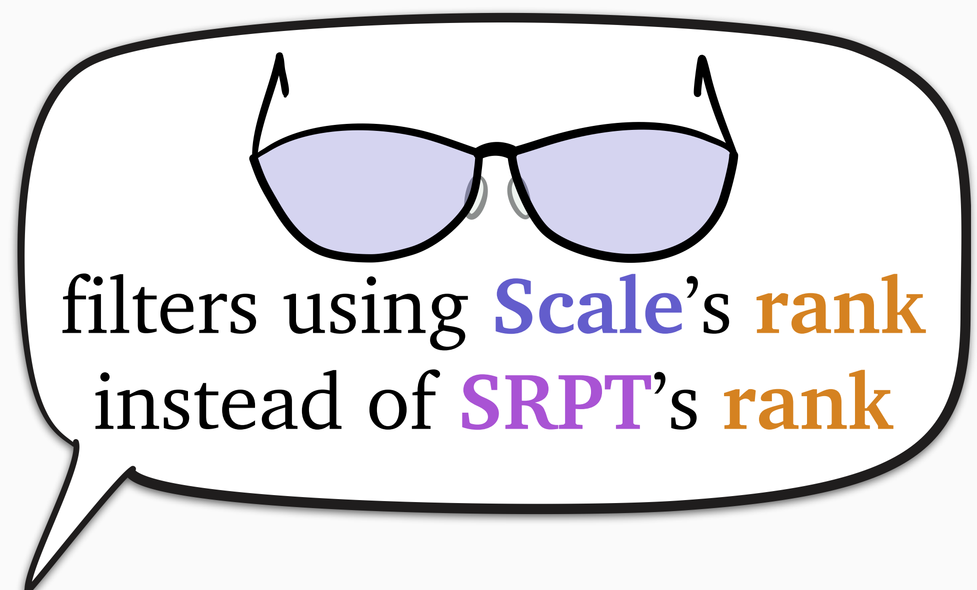
SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\text{SRPT}}(r)] \le \mathbf{E}[W_{\text{Scale}}(r)] \le \mathbf{E}\left[W_{\text{SRPT}}\left(\tfrac{\alpha}{\beta} r\right)\right]$$

Key steps:

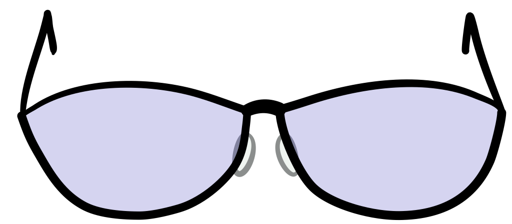
1. **SRPT** minimizes mean r-work
2. **Scale** minimizes mean **noise-scaled**-r-work



SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\mathrm{SRPT}}(r)] \le \mathbf{E}[W_{\mathrm{Scale}}(r)] \le \mathbf{E}\left[W_{\mathrm{SRPT}}\left(\tfrac{\alpha}{\beta} r\right)\right]$$



filters using **Scale**'s **rank** instead of **SRPT**'s **rank**

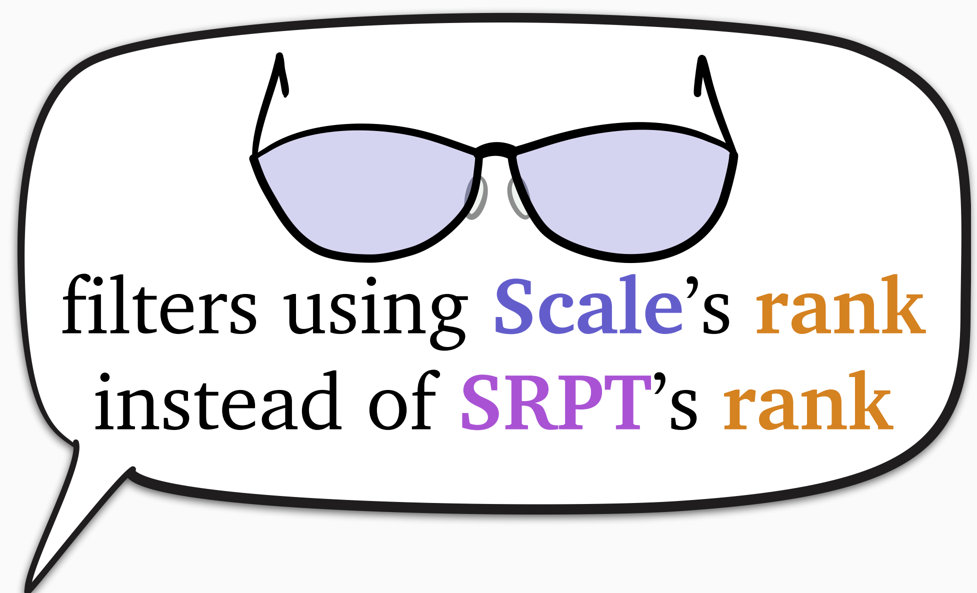
Key steps:

1. SRPT minimizes mean r-work
2. **Scale** minimizes mean **noise-scaled**-r-work
3. Under any policy,

SRPT vs. Scale proof sketch

Lemma:

$$\mathbf{E}[W_{\mathrm{SRPT}}(r)] \leq \mathbf{E}[W_{\mathrm{Scale}}(r)] \leq \mathbf{E}\left[W_{\mathrm{SRPT}}\left(\tfrac{\alpha}{\beta}r\right)\right]$$



Key steps:

1. SRPT minimizes mean r-work
2. Scale minimizes mean noise-scaled-r-work
3. Under any policy,

   r-work ≤ noise-scaled-αr-work ≤ $\frac{\alpha}{\beta} r$-work

SRPT vs. Scale proof sketch

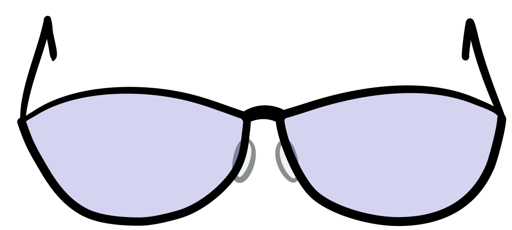
Lemma:

$$\mathbf{E}[W_{\mathrm{SRPT}}(r)] \le \mathbf{E}[W_{\mathrm{Scale}}(r)] \le \mathbf{E}\left[W_{\mathrm{SRPT}}\left(\tfrac{\alpha}{\beta} r\right)\right]$$

Key steps:

1. SRPT minimizes mean r-work
2. Scale minimizes mean **noise-scaled**-r-work
3. Under any policy,

$$r\text{-work} \le \textbf{noise-scaled-}\alpha r\text{-work} \le \tfrac{\alpha}{\beta} r\text{-work}$$



filters using Scale's **rank**
instead of SRPT's **rank**