Multi-objective Optimization of Long-run Average and Total Rewards

Tim Quatmann, Joost-Pieter Katoen

MOVES Seminar

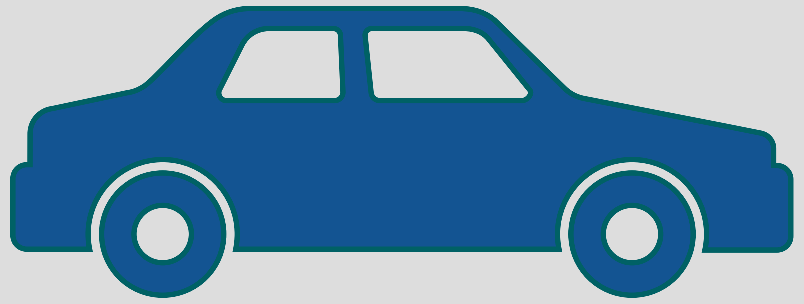
March 16, 2021

Software Modeling and Verification Chair | RWTH AACHEN UNIVERSITY

Multi-objective Model Checking

- Study tradeoffs between objectives

Example
Can the car drive fast, safe, **and** cost-efficient?



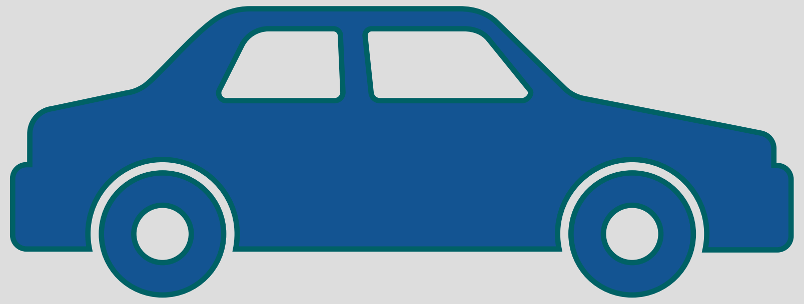
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Multi-objective Model Checking

- Study tradeoffs between objectives

Example

Can the car drive fast, safe, **and** cost-efficient?



Models

- Markov decision processes (MDP)

probabilistic branching
nondeterminism
rewards/costs

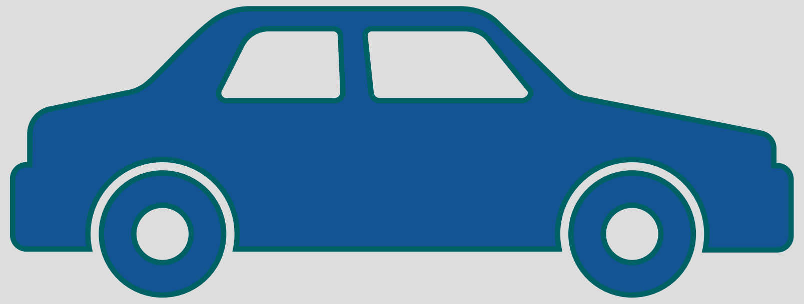
- Markov automata (MA)

MDP + continuous time

Multi-objective Model Checking

- Study tradeoffs between objectives

<u>Example</u>
Can the car drive fast, safe, **and** cost-efficient?



Models

- Markov decision processes (MDP)
- Markov automata (MA)

probabilistic branching
nondeterminism
rewards/costs

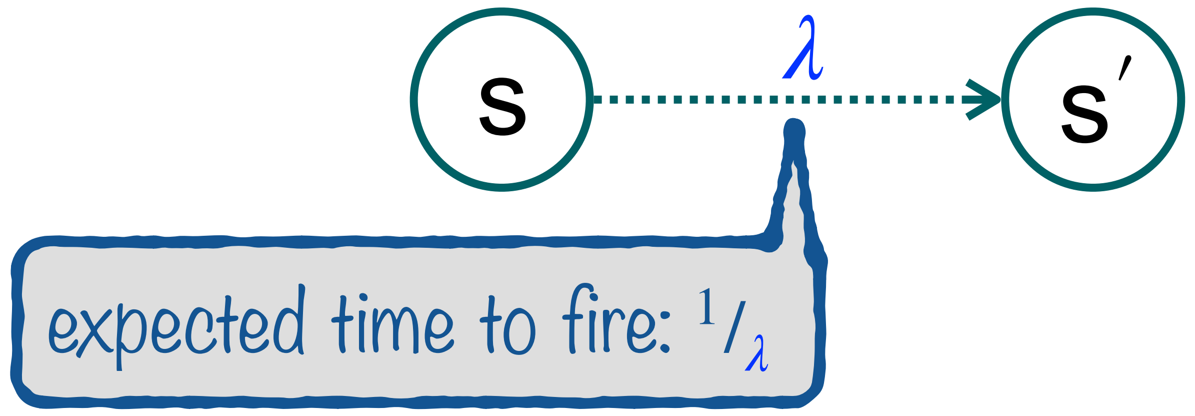
MDP + continuous time

Objectives

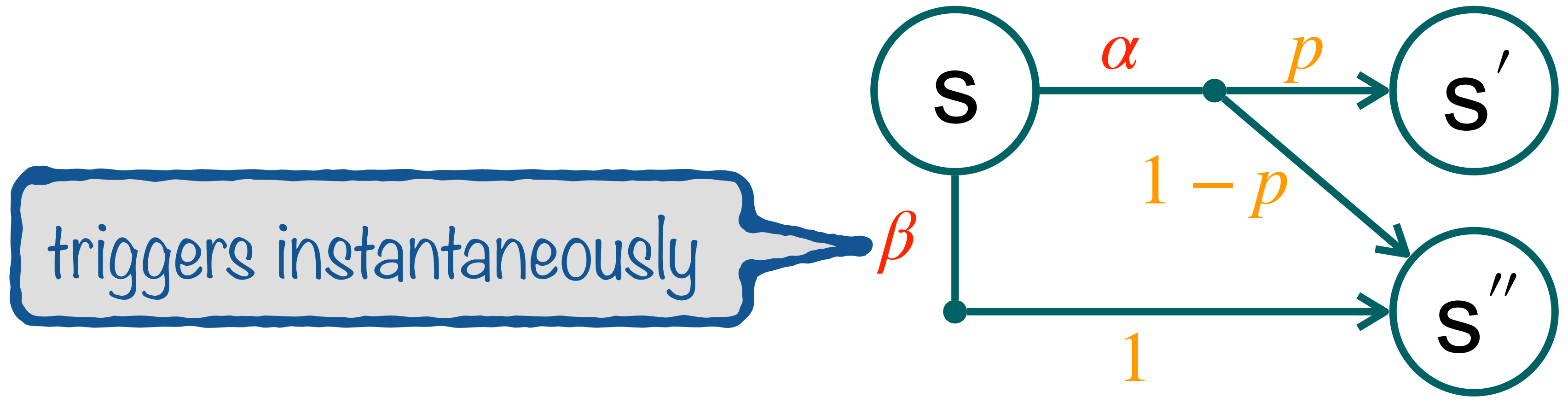
- Expected total rewards
- Expected long-run average rewards
- …

Two types of transitions

- Markovian: exponentially distributed time delay

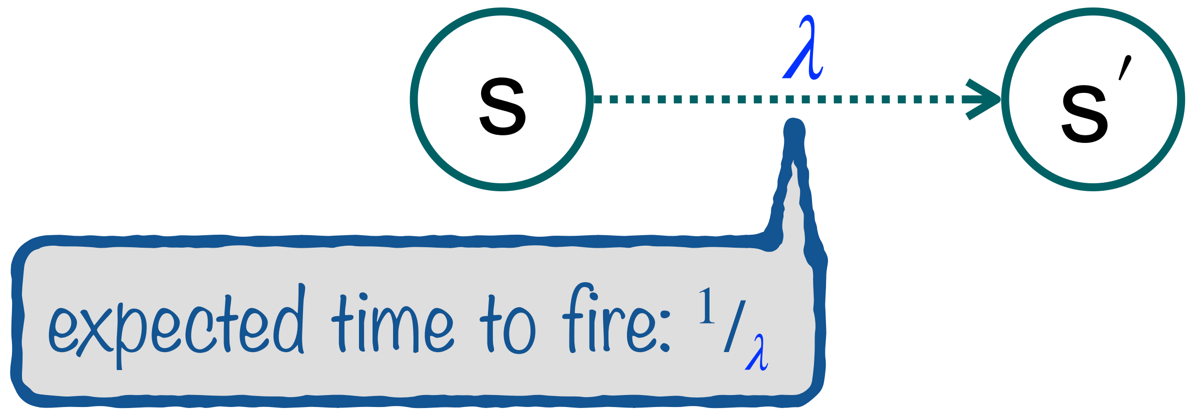


- Probabilistic: nondeterminism + branching

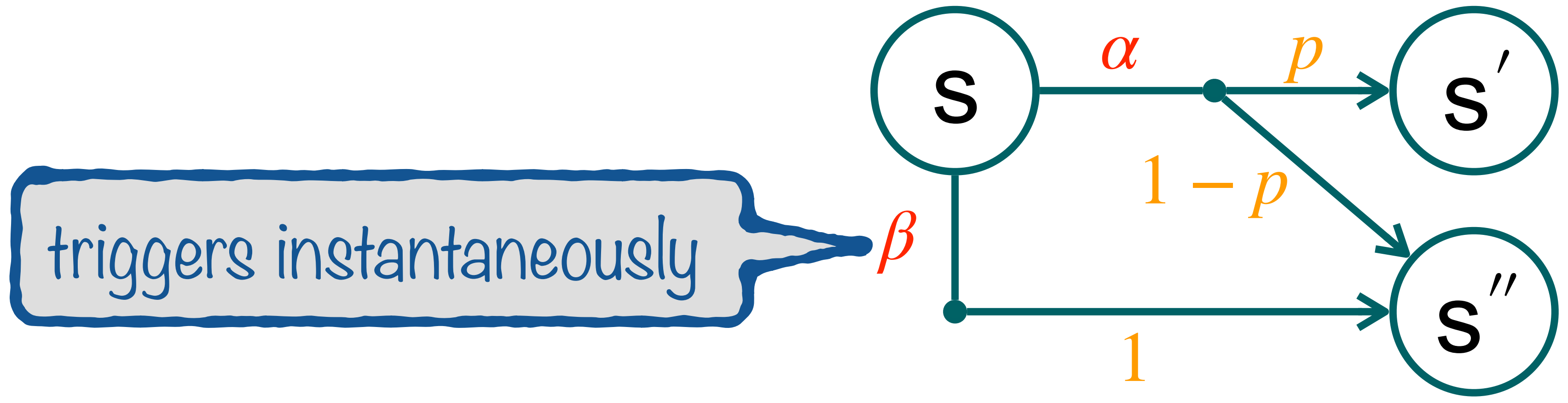


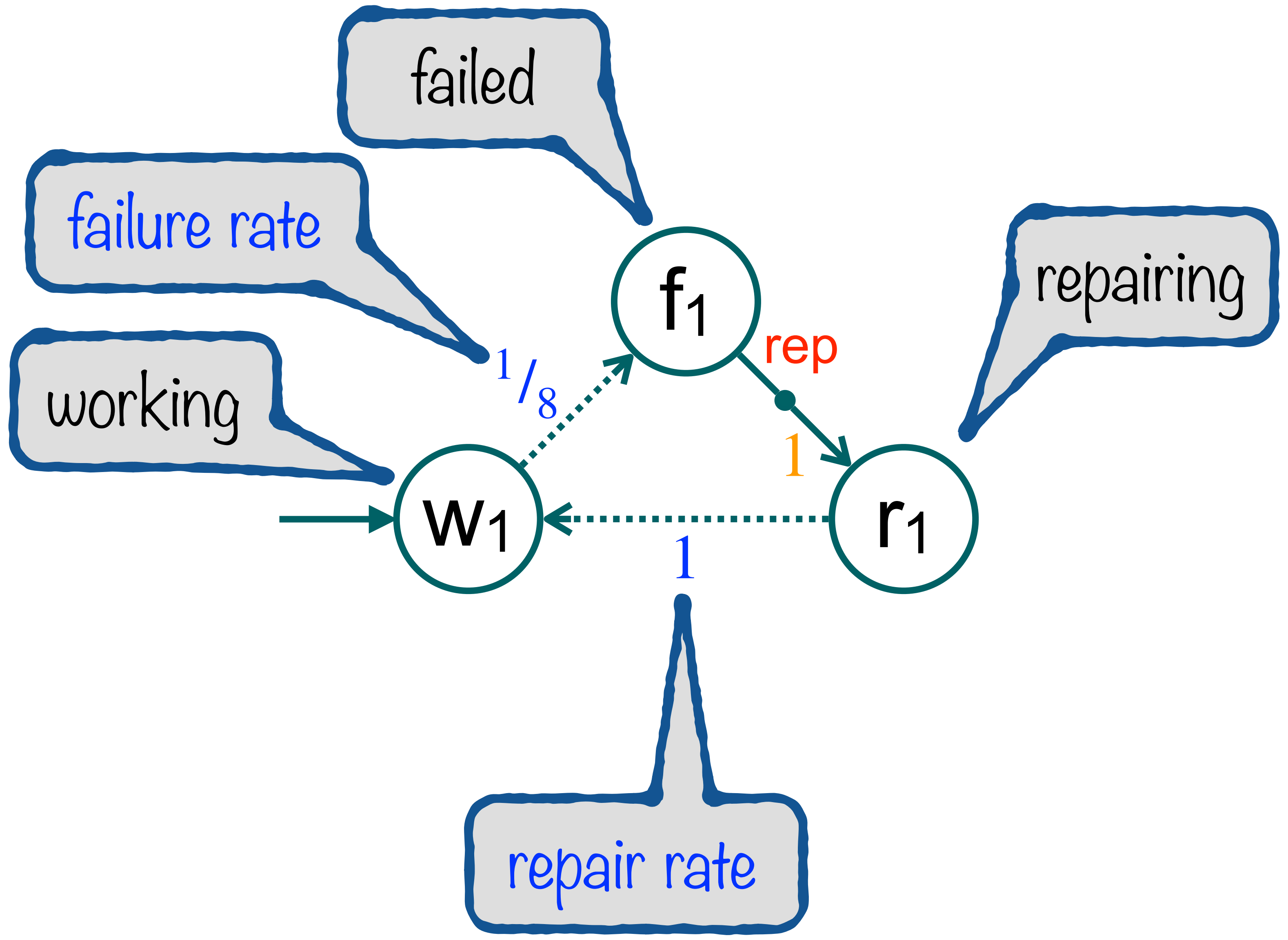
Two types of transitions

- Markovian: exponentially distributed time delay



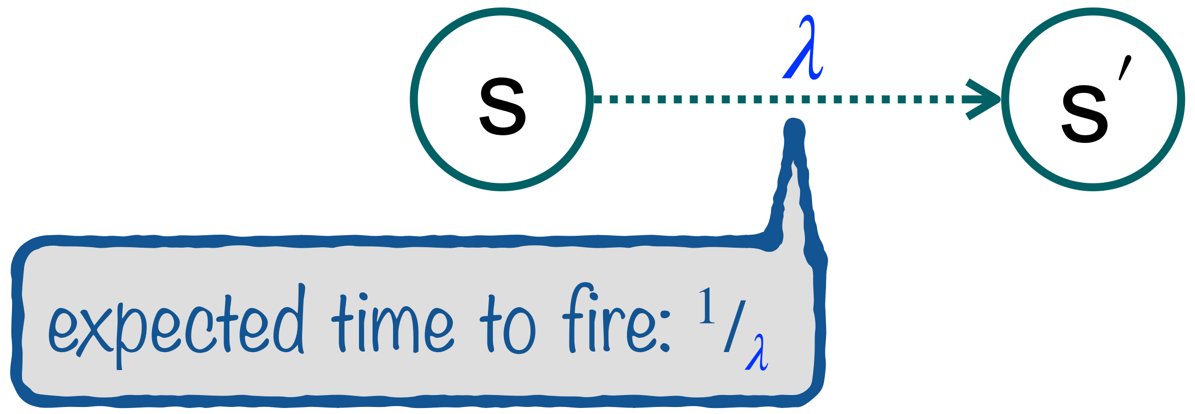
- Probabilistic: nondeterminism + branching



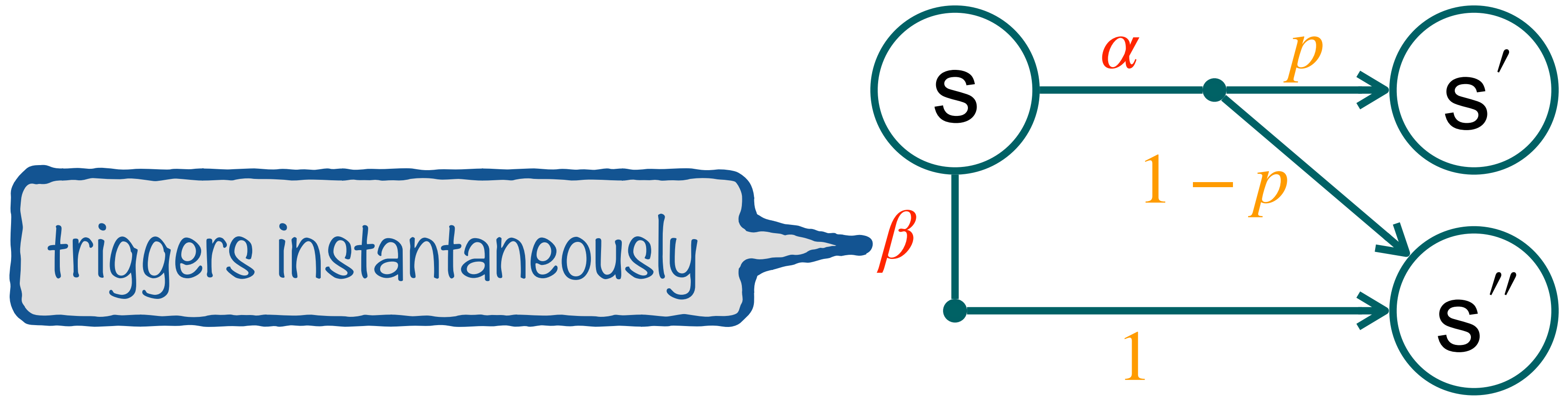


Two types of transitions

- Markovian: exponentially distributed time delay

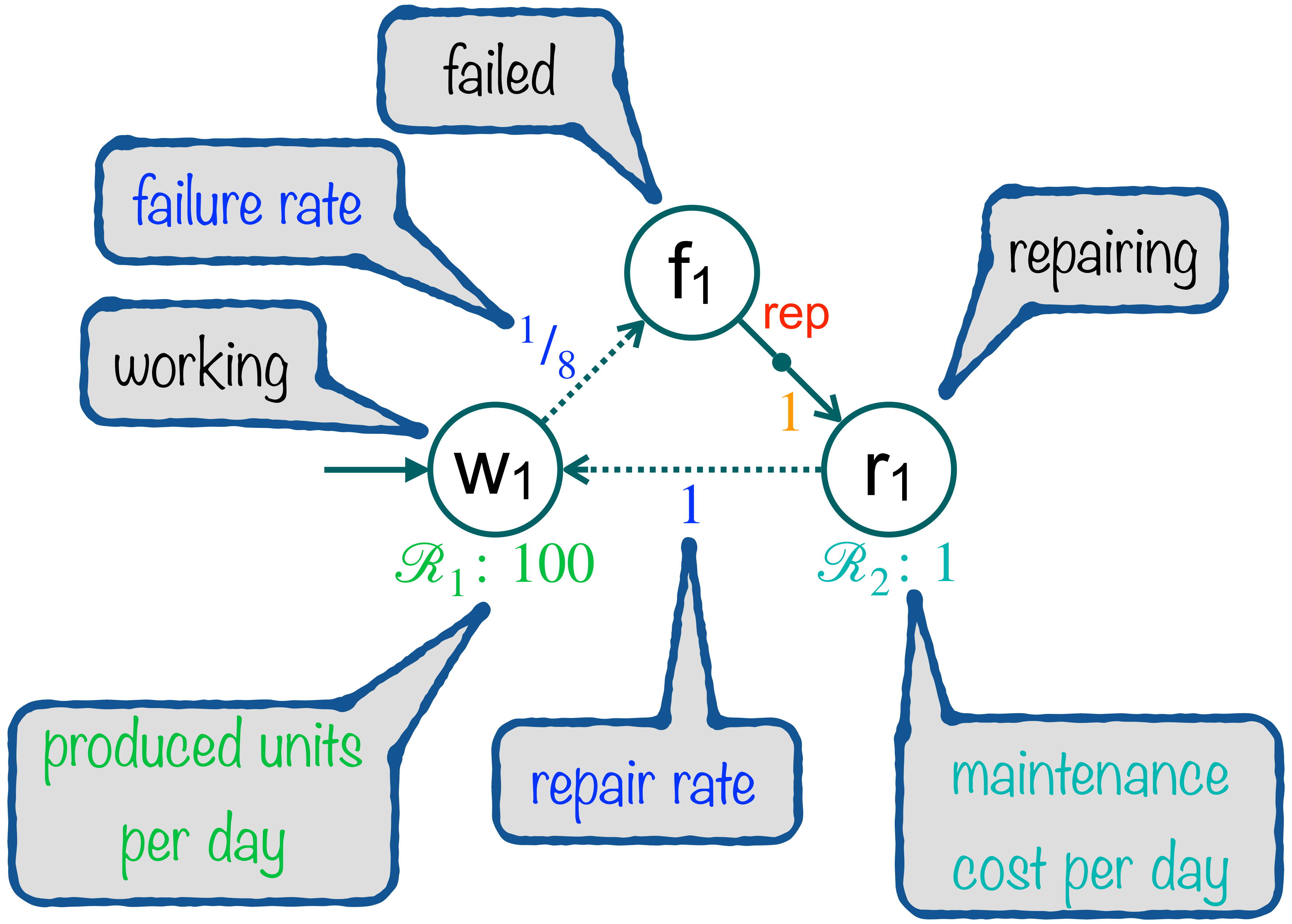


- Probabilistic: nondeterminism + branching



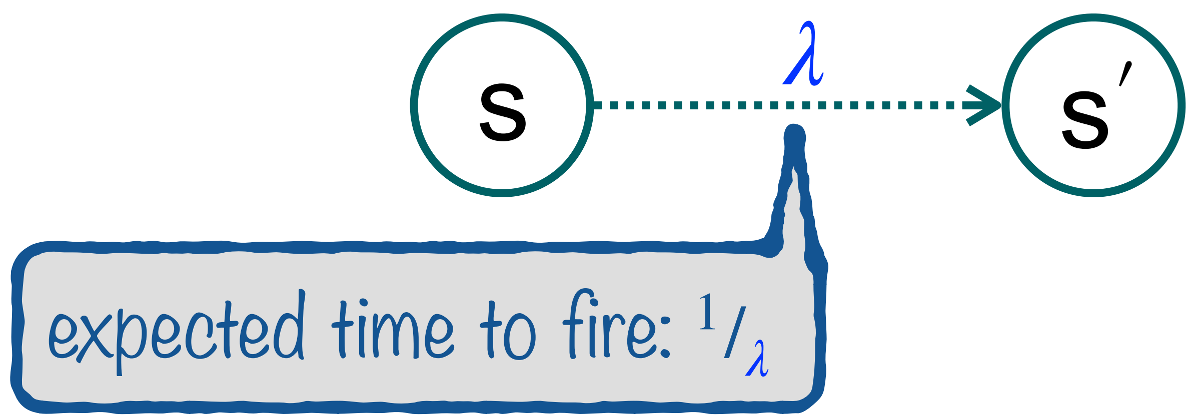
Multiple reward assignments $\mathscr{R}_1, \mathscr{R}_2, \mathscr{R}_3, \ldots$

- State rewards collected over time
- One-off transition rewards

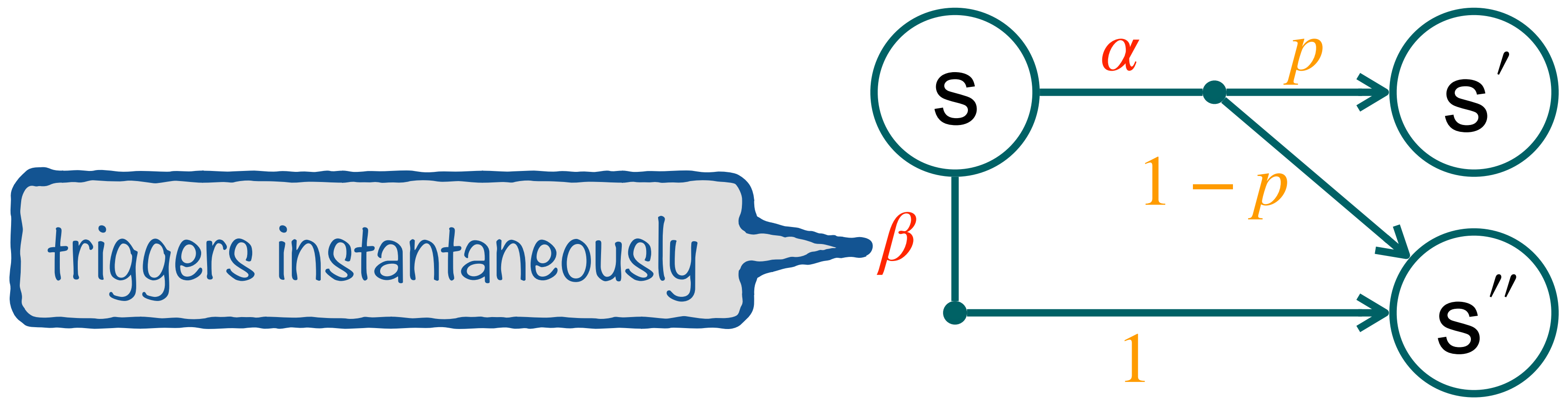


Two types of transitions

- Markovian: exponentially distributed time delay



- Probabilistic: nondeterminism + branching

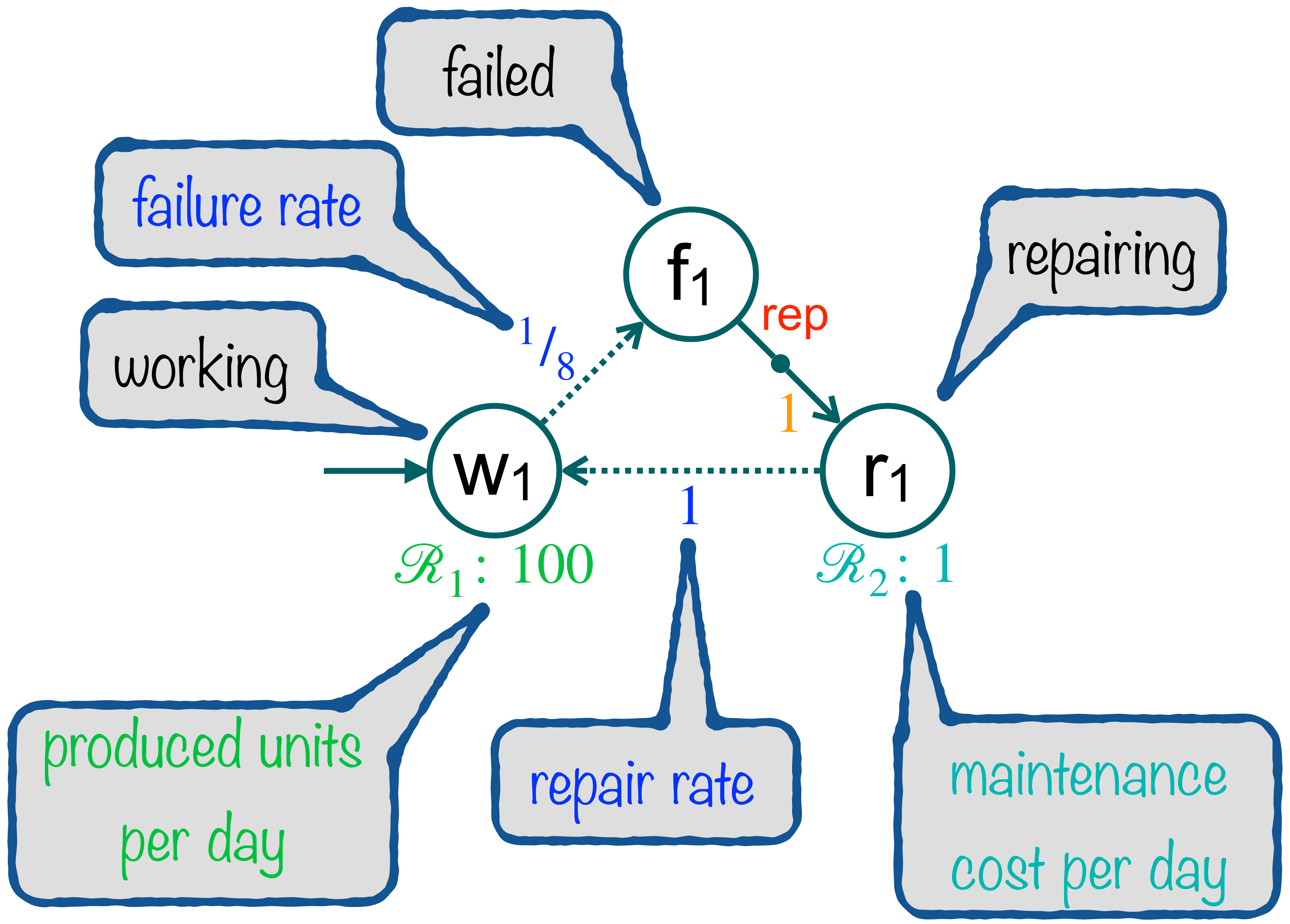


Multiple reward assignments $\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3, \ldots$

- State rewards collected over time
- One-off transition rewards

Expected maintenance cost per day:

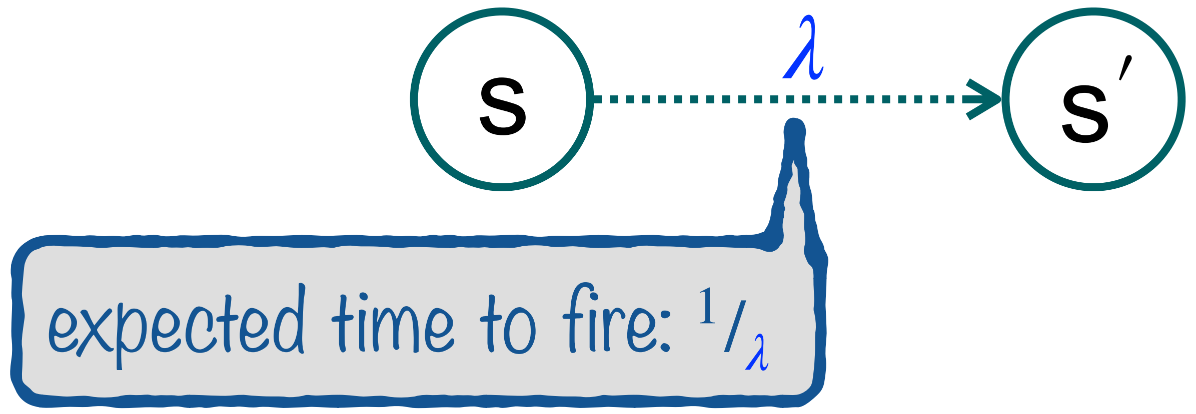
$$1/9 \cdot 1 \approx 0.11$$



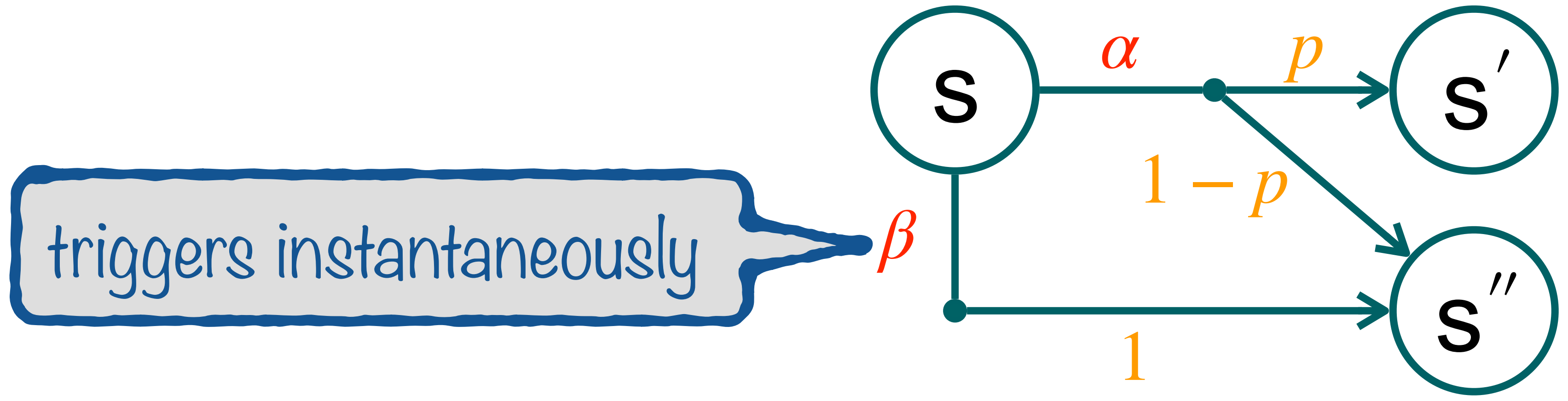
Markov Automata — Example

Two types of transitions

- Markovian: exponentially distributed time delay



- Probabilistic: nondeterminism + branching



Multiple reward assignments $\mathscr{R}_1, \mathscr{R}_2, \mathscr{R}_3, \ldots$

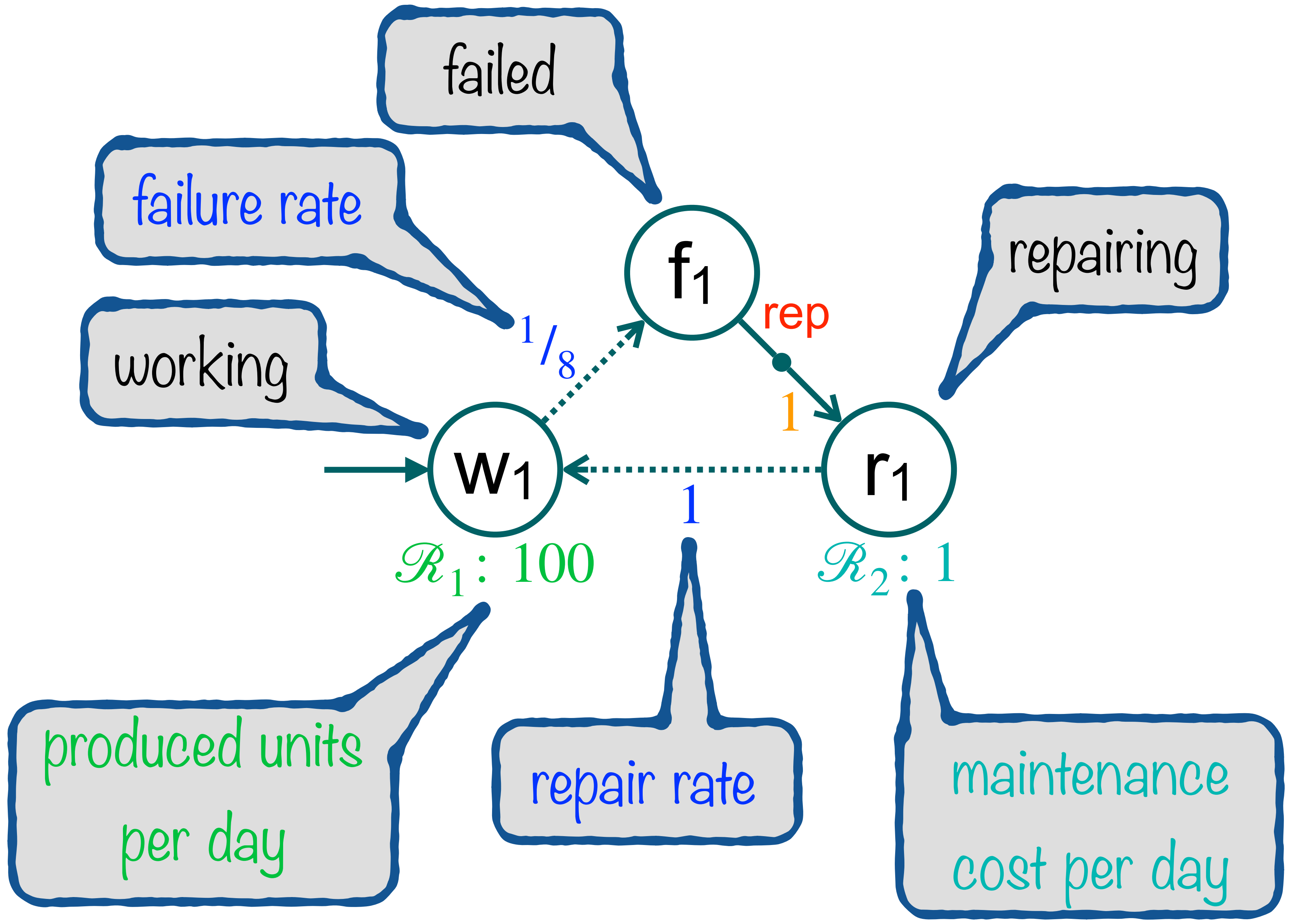
- State rewards collected over time
- One-off transition rewards

Expected maintenance cost per day:

$$^1/_9 \cdot 1 \approx 0.11$$

Expected number of produced units per day:

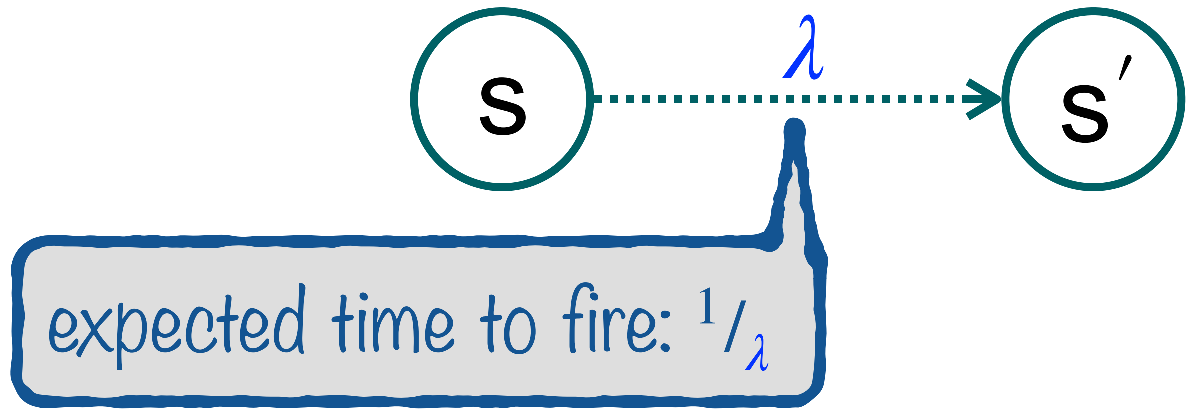
$$100 \cdot {}^8/_9 \approx 88.9$$



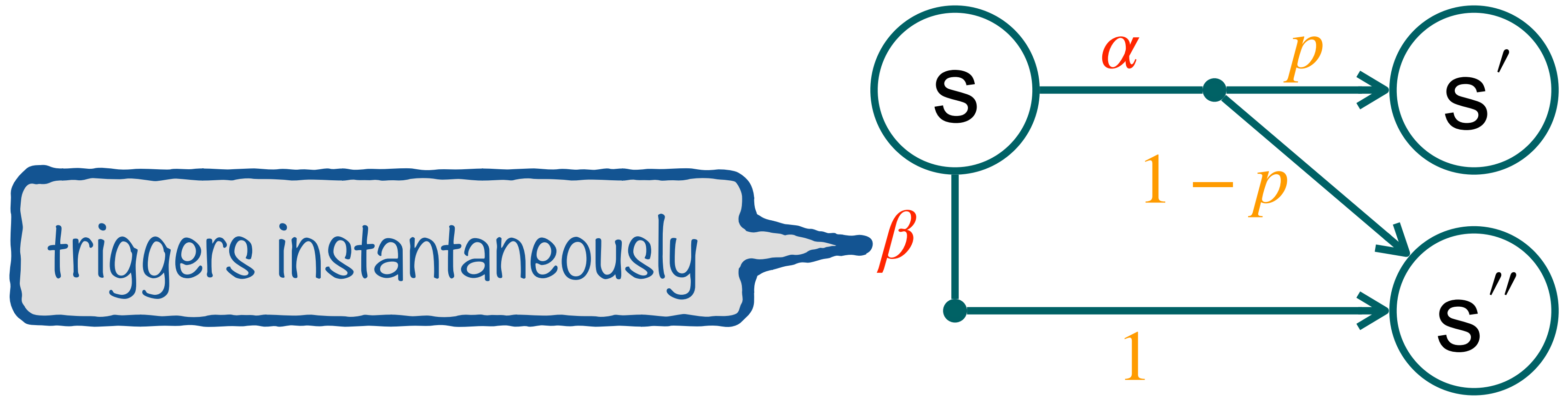
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Two types of transitions

- Markovian: exponentially distributed time delay

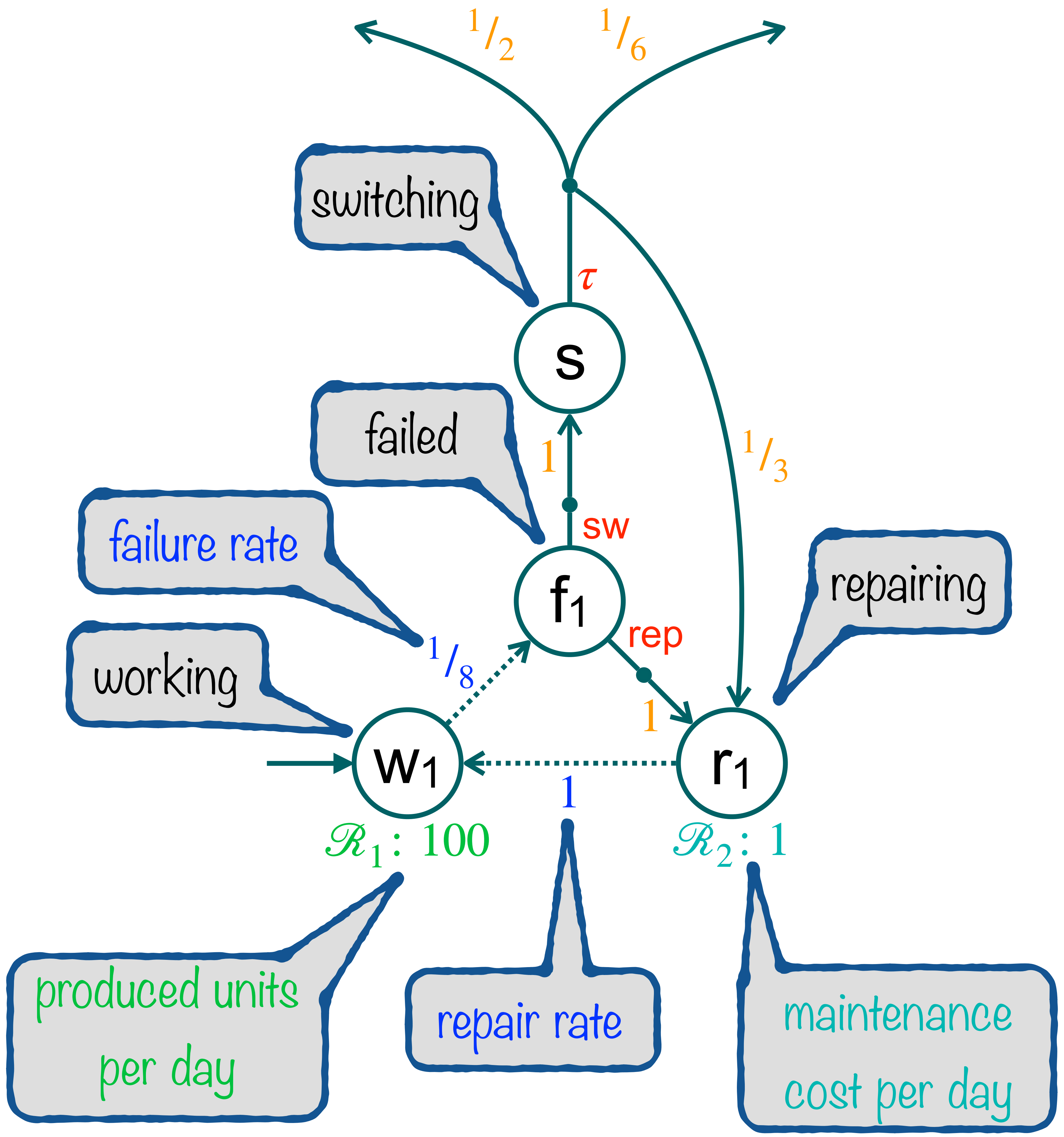


- Probabilistic: nondeterminism + branching



Multiple reward assignments $\mathscr{R}_1, \mathscr{R}_2, \mathscr{R}_3, \ldots$

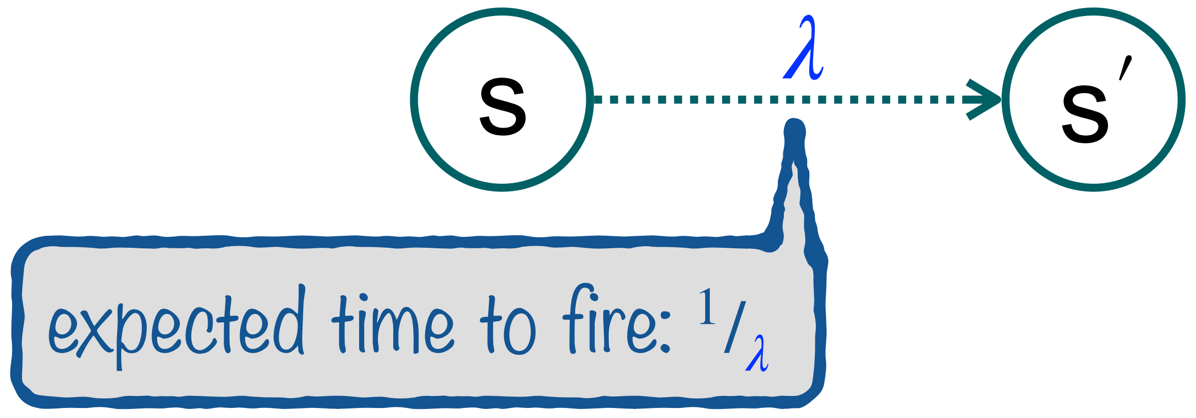
- State rewards collected over time
- One-off transition rewards



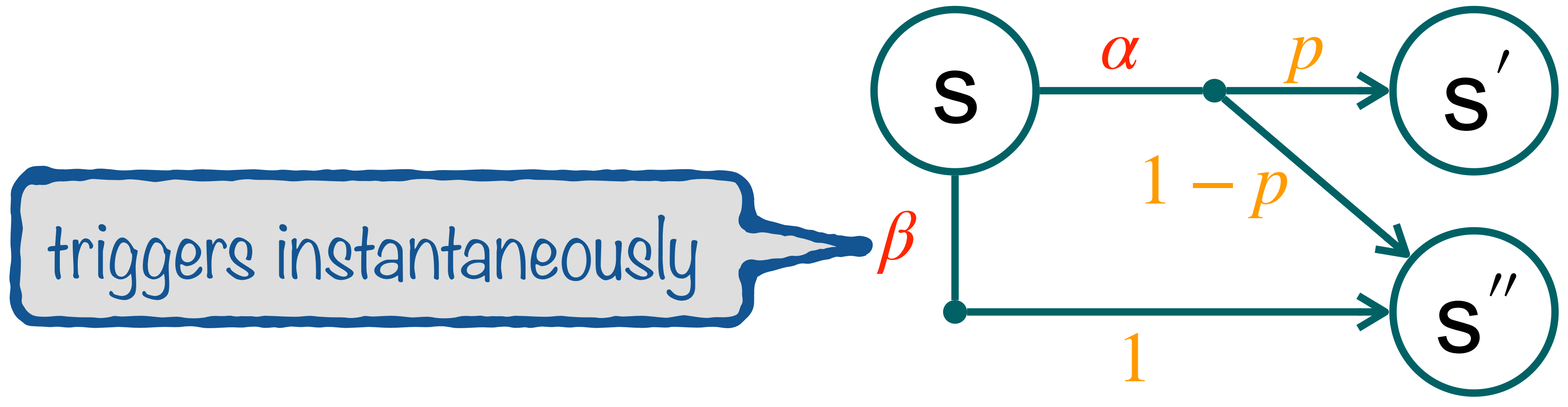
Markov Automata — Example

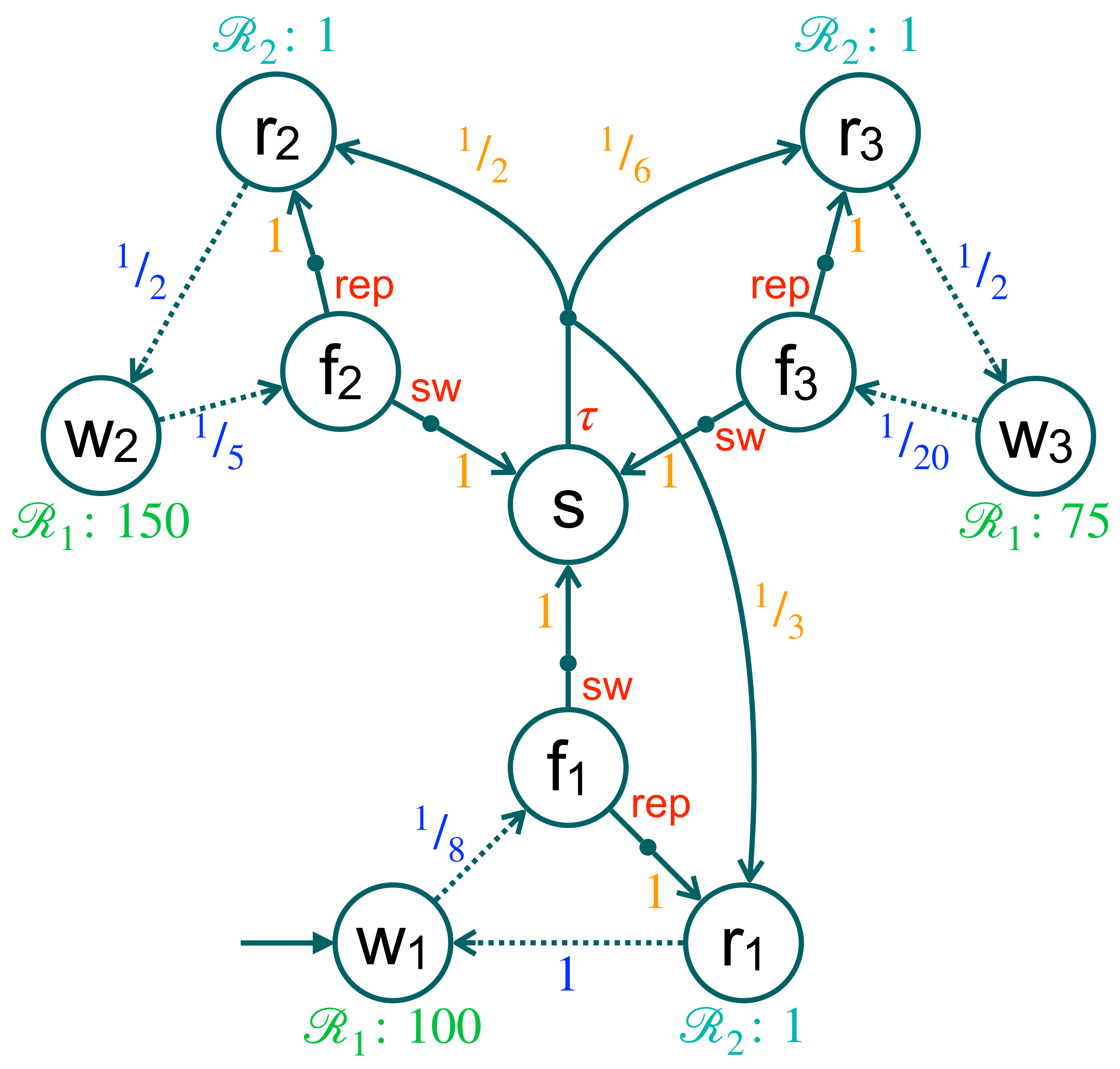
Two types of transitions

- Markovian: exponentially distributed time delay



- Probabilistic: nondeterminism + branching



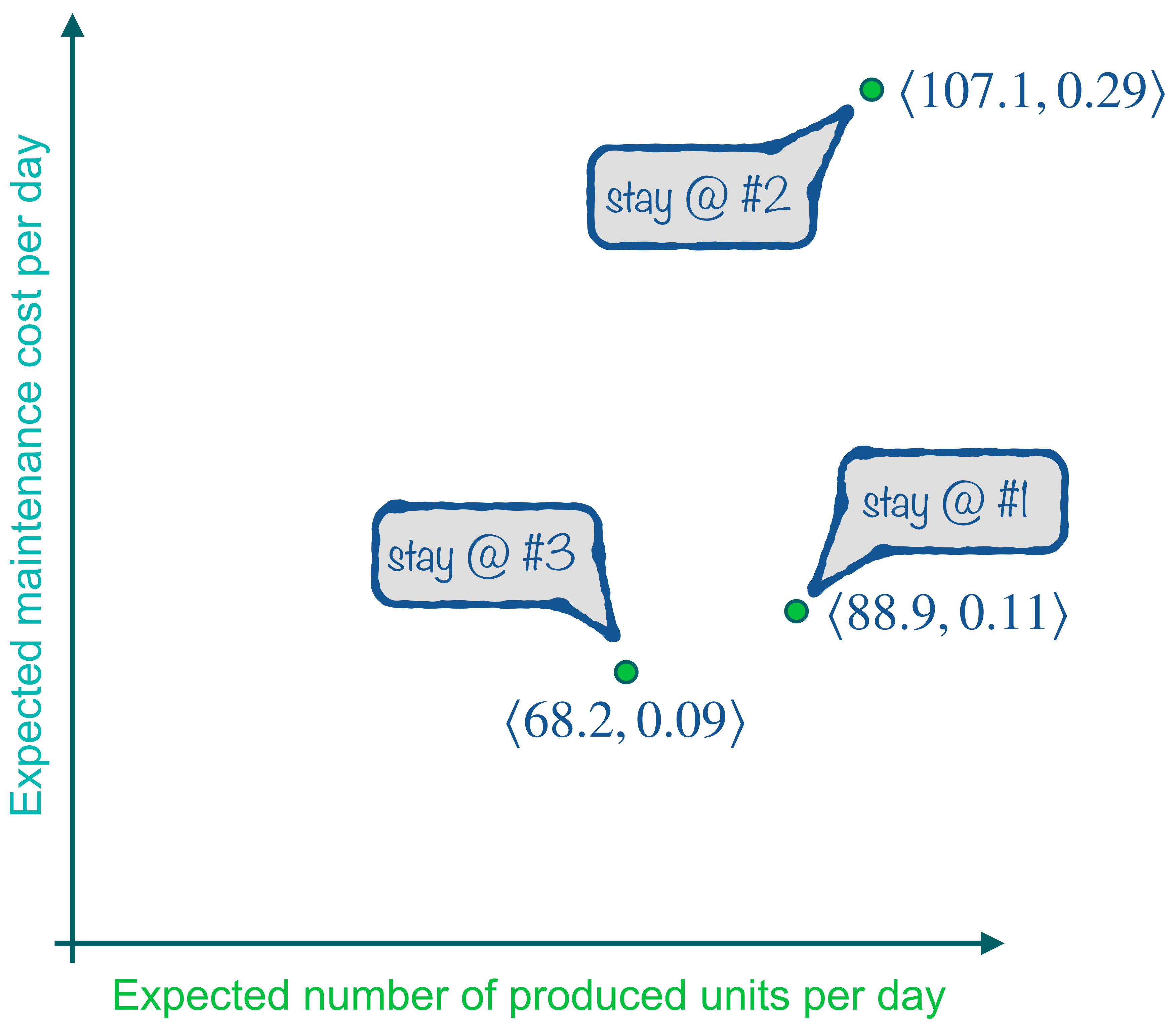


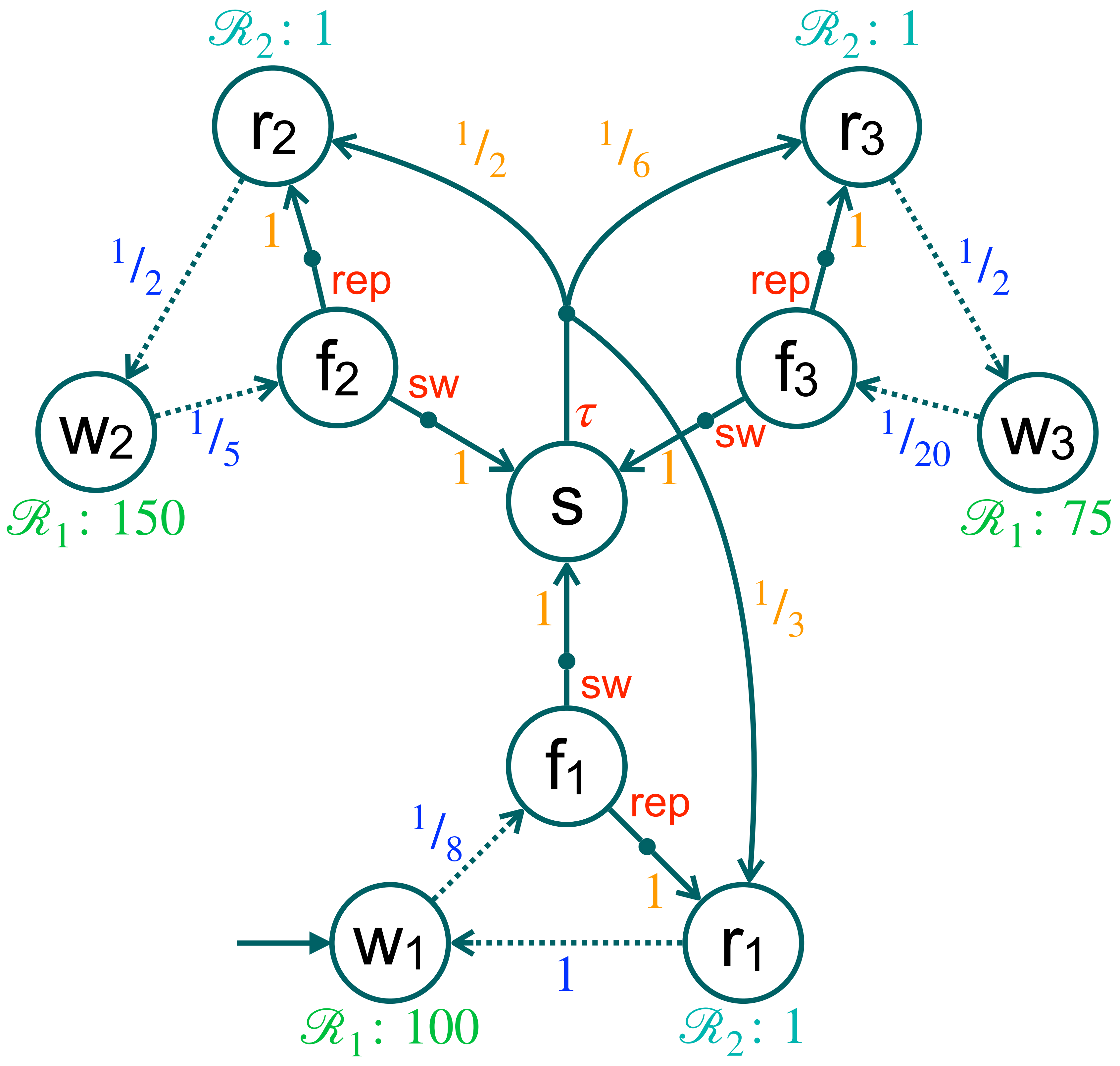
Multiple reward assignments $\mathscr{R}_1, \mathscr{R}_2, \mathscr{R}_3, \ldots$

- State rewards collected over time

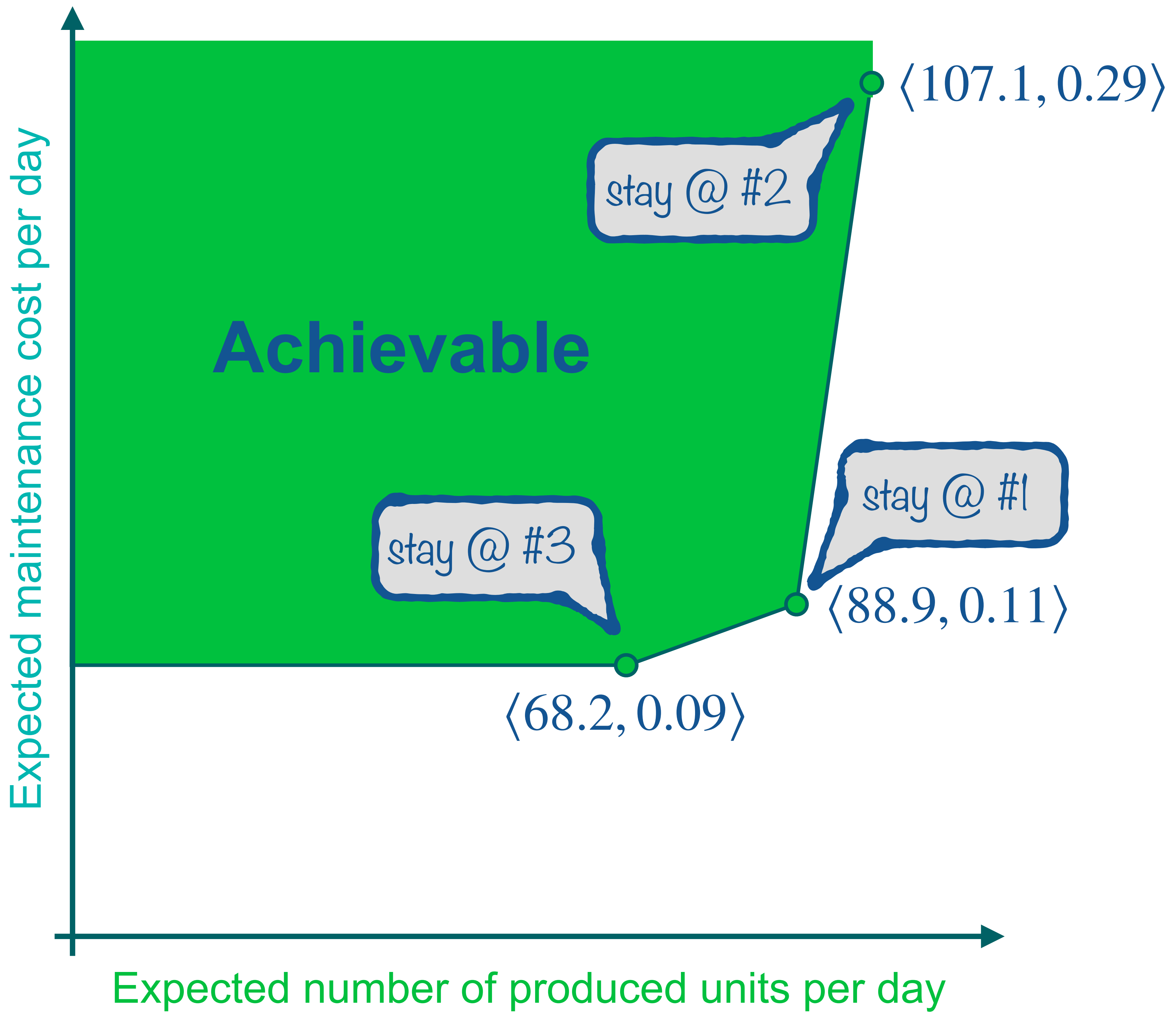
- One-off transition rewards

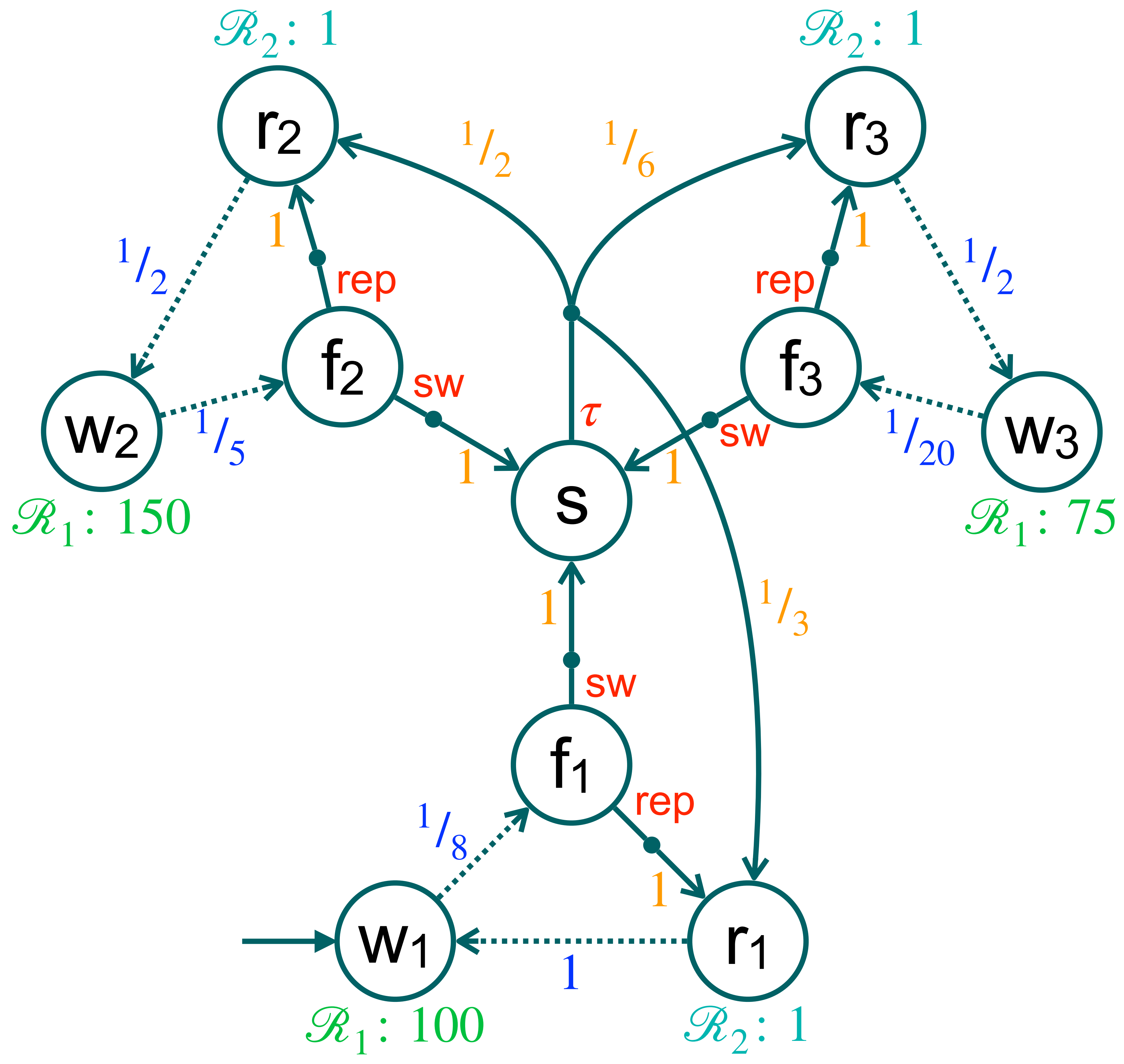
Markov Automata — Example





Markov Automata — Example





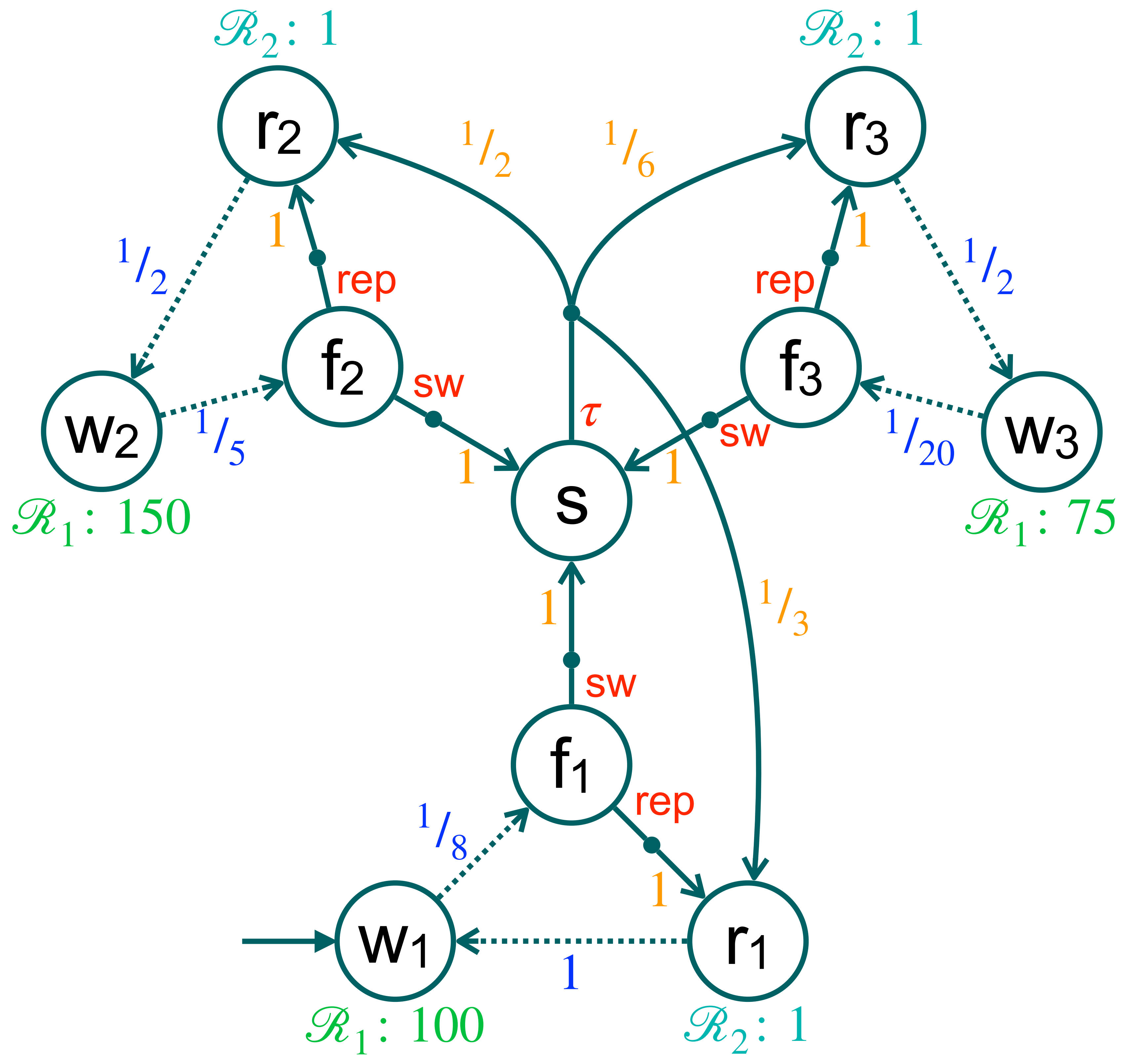
Tim Quatmann, Joost-Pieter Katoen
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Markov Automata — Notations

- **Path**: alternating sequence of states and durations/actions

infinite $\pi = \left(w_1 \xrightarrow{1} f_1 \xrightarrow{\text{rep}} r_1 \xrightarrow{1}\right)^\omega$

finite $\hat{\pi} = w_1 \xrightarrow{7.2} f_1 \xrightarrow{\text{sw}} s \xrightarrow{\tau} r_2 \xrightarrow{3.2} w_2 \xrightarrow{4.8} f_2$



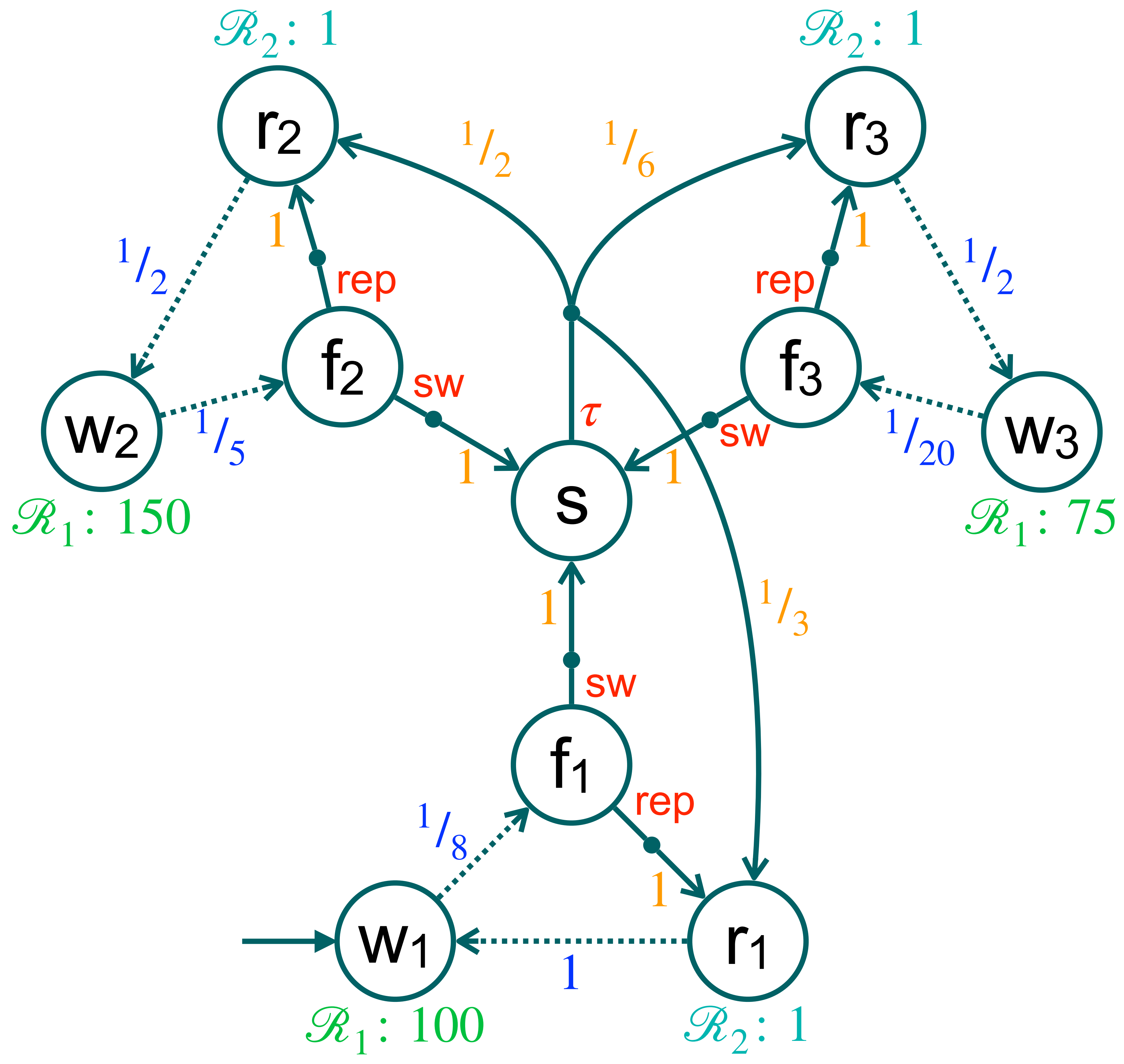
Markov Automata — Notations

- **Path**: alternating sequence of states and durations/actions
- **Accumulated reward** $\mathscr{R}(\hat{\pi})$ for finite path $\hat{\pi}$

infinite $\pi = \left(w_1 \xrightarrow{1} f_1 \xrightarrow{\text{rep}} r_1 \xrightarrow{1}\right)^{\omega}$

finite $\hat{\pi} = w_1 \xrightarrow{7.2} f_1 \xrightarrow{\text{sw}} s \xrightarrow{\tau} r_2 \xrightarrow{3.2} w_2 \xrightarrow{4.8} f_2$

$$\mathscr{R}_1(\hat{\pi}) = 100 \cdot 7.2 + 150 \cdot 4.8 = 1440$$



Markov Automata — Notations

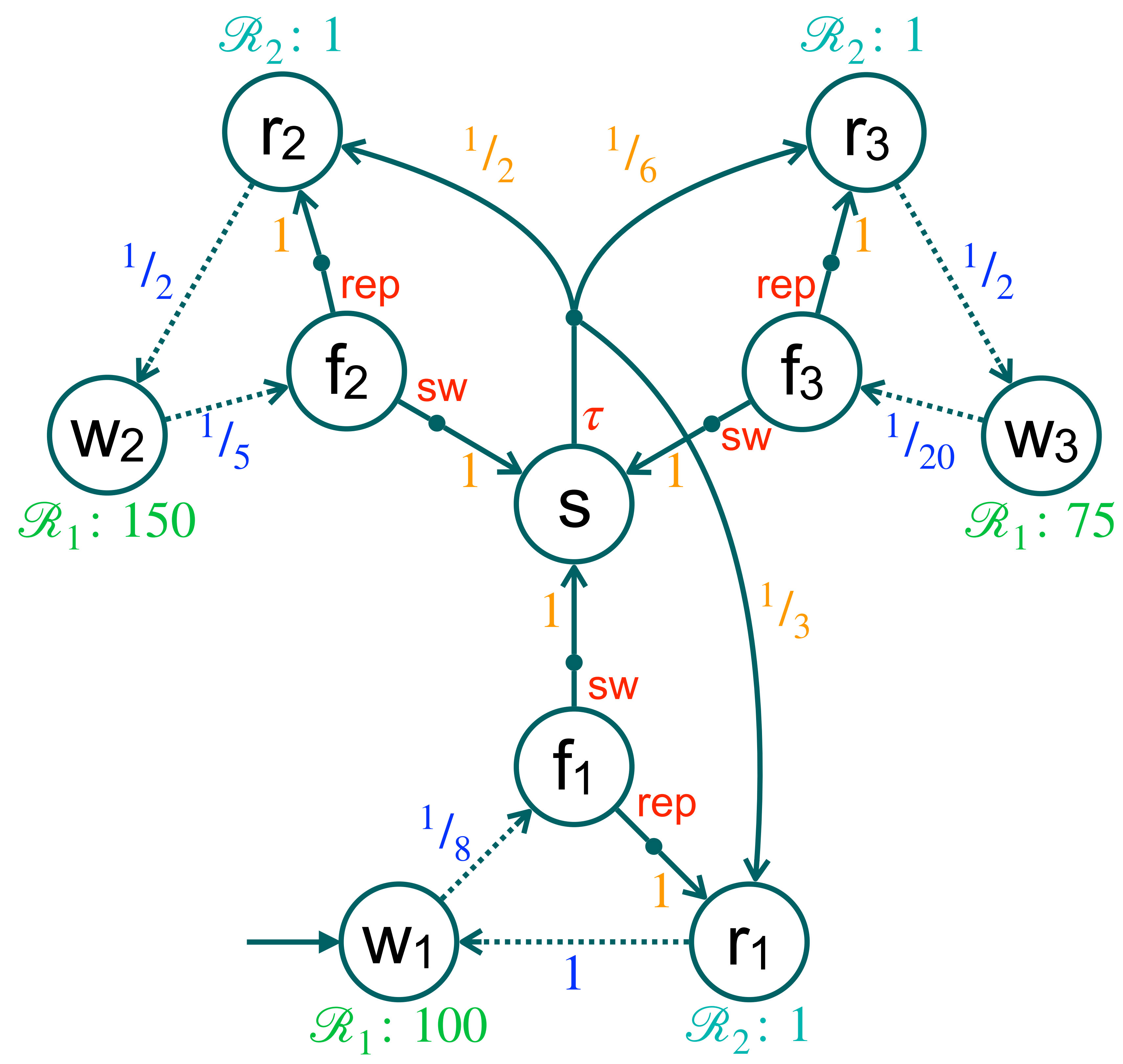
- **Path**: alternating sequence of states and durations/actions
- **Accumulated reward** $\mathscr{R}(\hat{\pi})$ for finite path $\hat{\pi}$
- **Strategy** σ: $\mathrm{Paths}_{\mathrm{fin}} \to \mathrm{Distr}(\mathrm{Act})$ — resolves nondeterminism

infinite: $\pi = \left(\mathrm{w}_1 \xrightarrow{1} \mathrm{f}_1 \xrightarrow{\mathrm{rep}} \mathrm{r}_1 \xrightarrow{1}\right)^{\omega}$

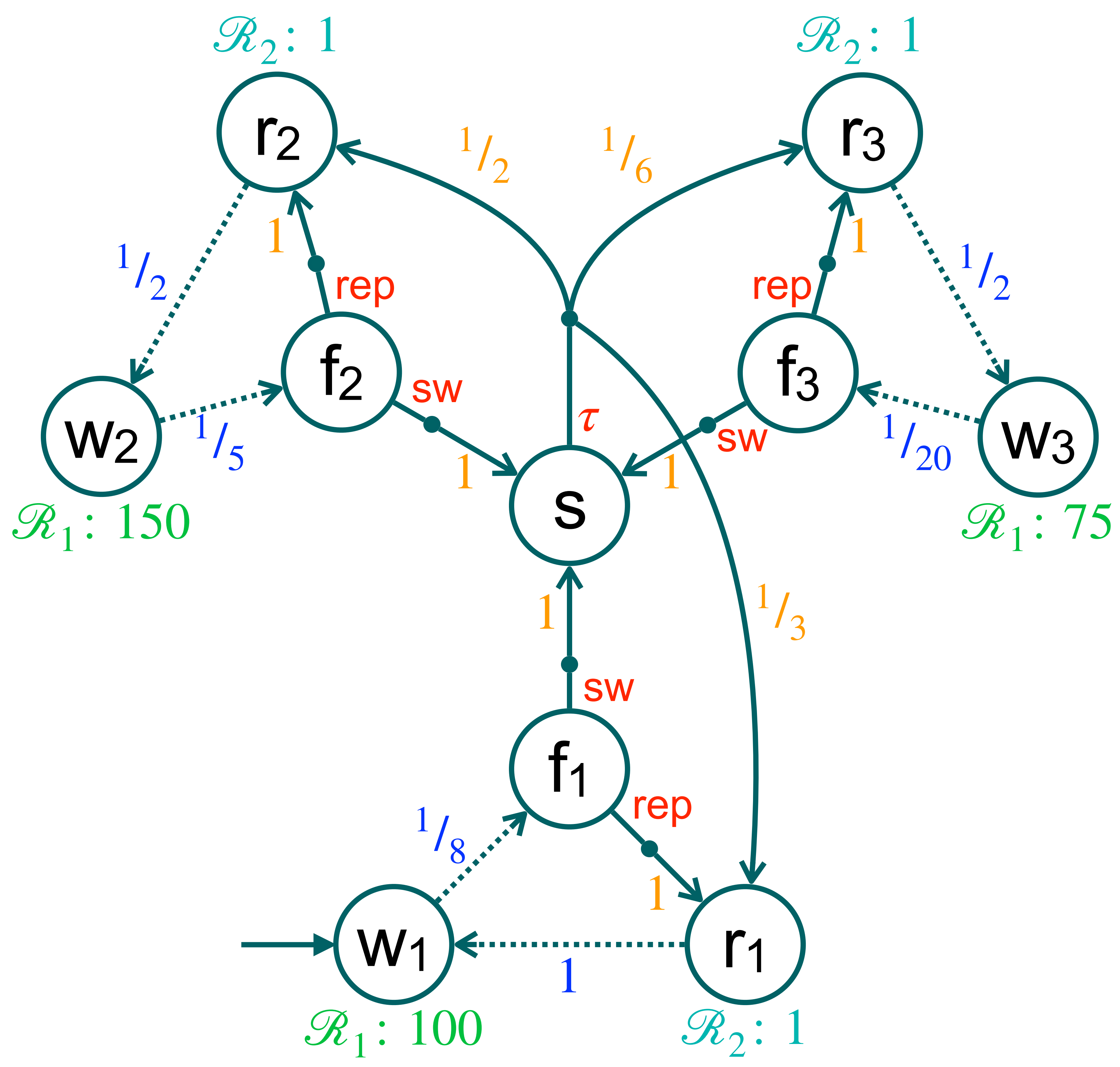
finite: $\hat{\pi} = \mathrm{w}_1 \xrightarrow{7.2} \mathrm{f}_1 \xrightarrow{\mathrm{sw}} \mathrm{s} \xrightarrow{\tau} \mathrm{r}_2 \xrightarrow{3.2} \mathrm{w}_2 \xrightarrow{4.8} \mathrm{f}_2$

$$\mathscr{R}_1(\hat{\pi}) = 100 \cdot 7.2 + 150 \cdot 4.8 = 1440$$

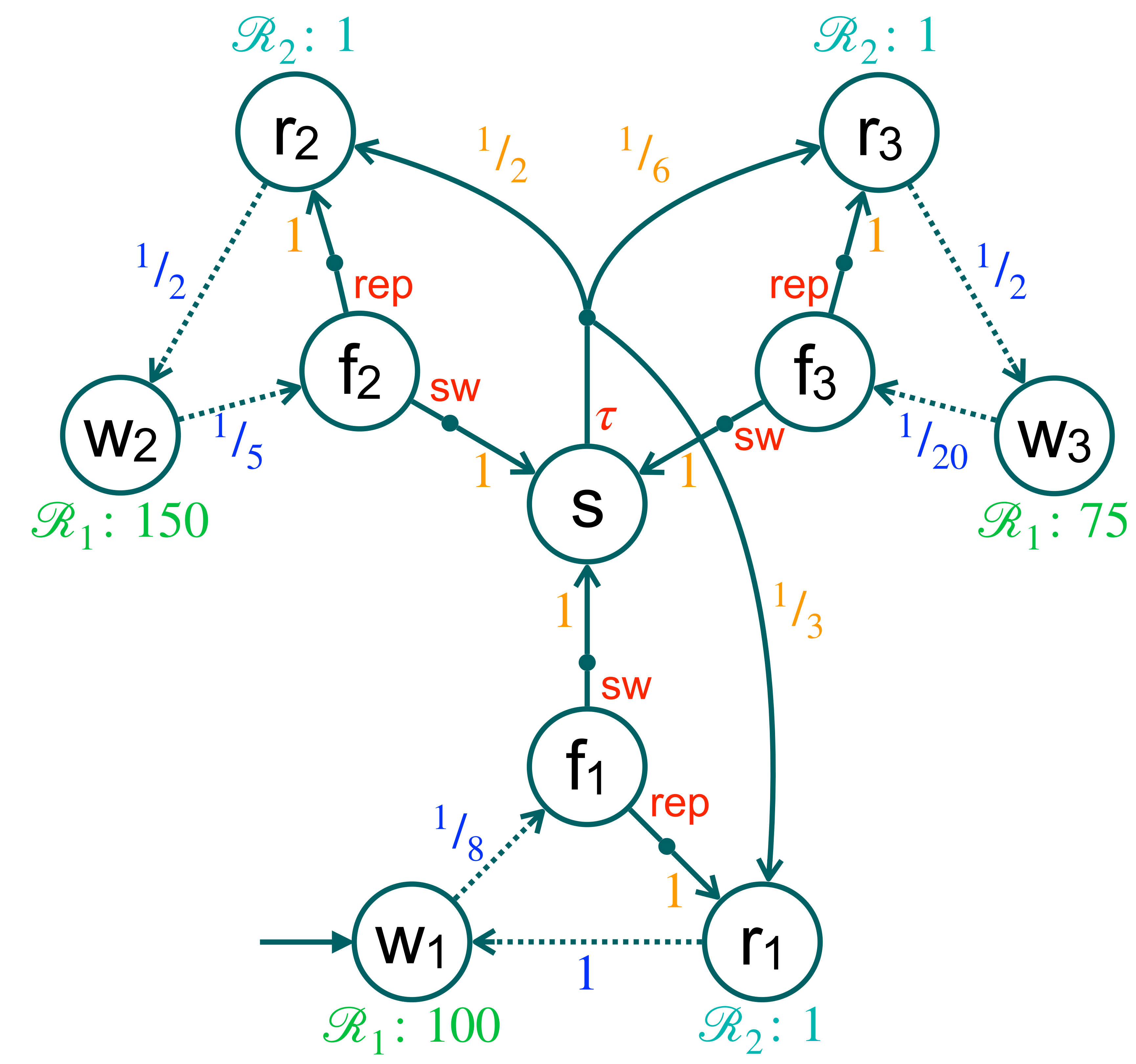
$$\sigma(\hat{\pi}) = \left\{\mathrm{rep} \mapsto {}^1/_3,\ \mathrm{sw} \mapsto {}^2/_3\right\}$$



- **Path**: alternating sequence of states and durations/actions
- **Accumulated reward** $\mathscr{R}(\hat{\pi})$ for finite path $\hat{\pi}$
- **Strategy** $\sigma\colon \mathrm{Paths}_{\mathrm{fin}} \to \mathrm{Distr}(\mathrm{Act})$ — resolves nondeterminism
- **Probability measure** Pr_σ on paths



- **Path**: alternating sequence of states and durations/actions
- **Accumulated reward** $\mathscr{R}(\hat{\pi})$ for finite path $\hat{\pi}$
- **Strategy** σ: $\text{Paths}_{\text{fin}} \to \text{Distr}(\text{Act})$ — resolves nondeterminism
- **Probability measure** Pr_σ on paths
- **Expected value** $\text{Ex}_\sigma(f) := \int f(\pi)\, d\text{Pr}_\sigma(\pi)$
  - … for **objective** f: $\text{Paths}_{\text{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$

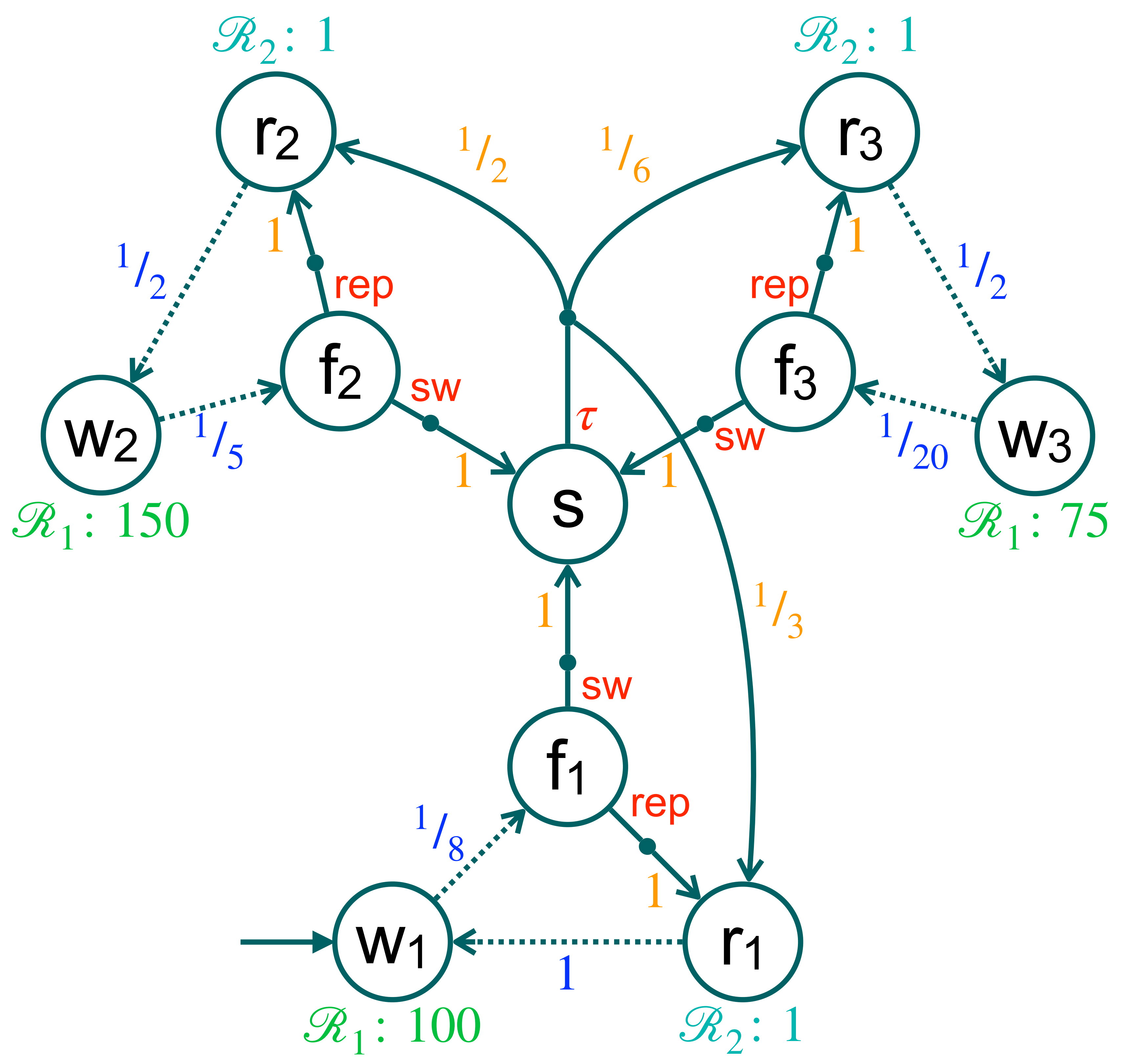


Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

- **Path**: alternating sequence of states and durations/actions
- **Accumulated reward** $\mathscr{R}(\hat{\pi})$ for finite path $\hat{\pi}$
- **Strategy** $\sigma\colon \mathrm{Paths}_{\mathrm{fin}} \to \mathrm{Distr}(\mathrm{Act})$ — resolves nondeterminism
- **Probability measure** Pr_σ on paths
- **Expected value** $\mathrm{Ex}_\sigma(f) := \int f(\pi)\, d\mathrm{Pr}_\sigma(\pi)$
  - … for **objective** $f\colon \mathrm{Paths}_{\mathrm{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$

Total reward objective:

$$tot(\mathscr{R})\colon \pi \mapsto \lim_{k\to\infty} \mathscr{R}\big(pref(\pi, k)\big)$$

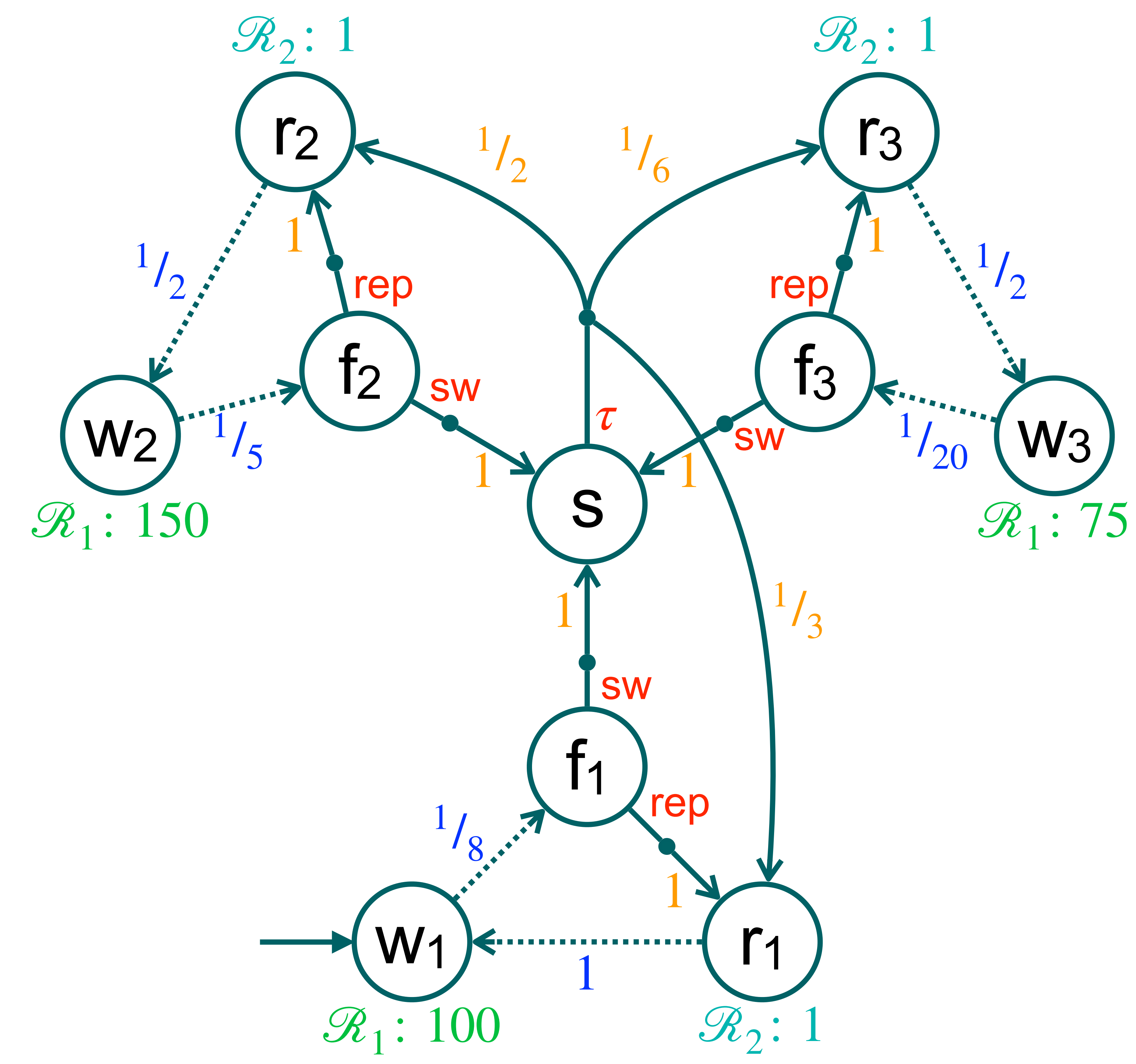


Markov Automata — Notations

- **Path**: alternating sequence of states and durations/actions
- **Accumulated reward** $\mathscr{R}(\hat{\pi})$ for finite path $\hat{\pi}$
- **Strategy** $\sigma\colon \mathrm{Paths}_{\mathrm{fin}} \to \mathrm{Distr}(\mathrm{Act})$

  resolves nondeterminism

- **Probability measure** Pr_σ on paths
- **Expected value** $\mathrm{Ex}_\sigma(f) := \int f(\pi)\, d\mathrm{Pr}_\sigma(\pi)$
  - … for **objective** $f\colon \mathrm{Paths}_{\mathrm{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$

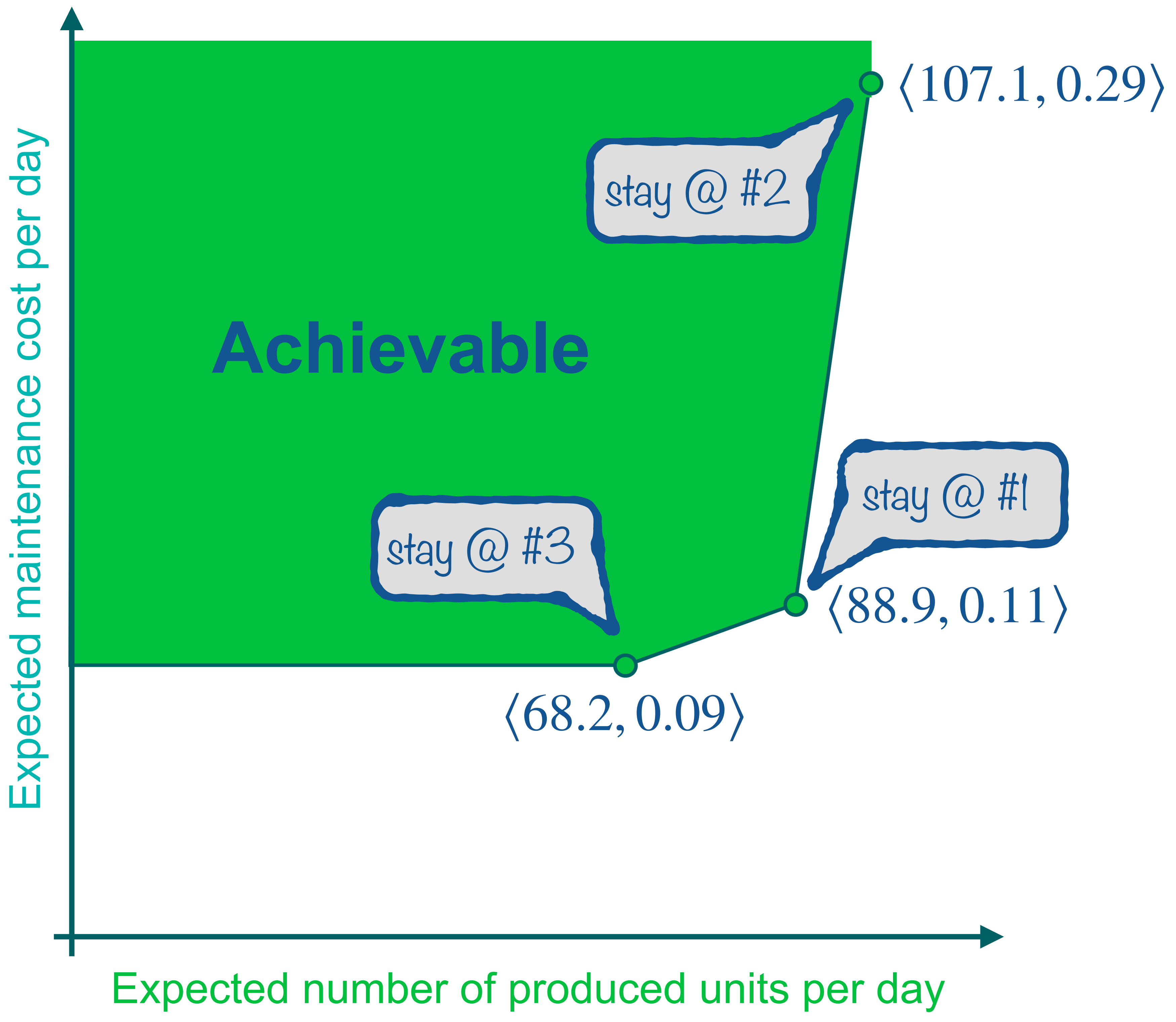


Total reward objective:

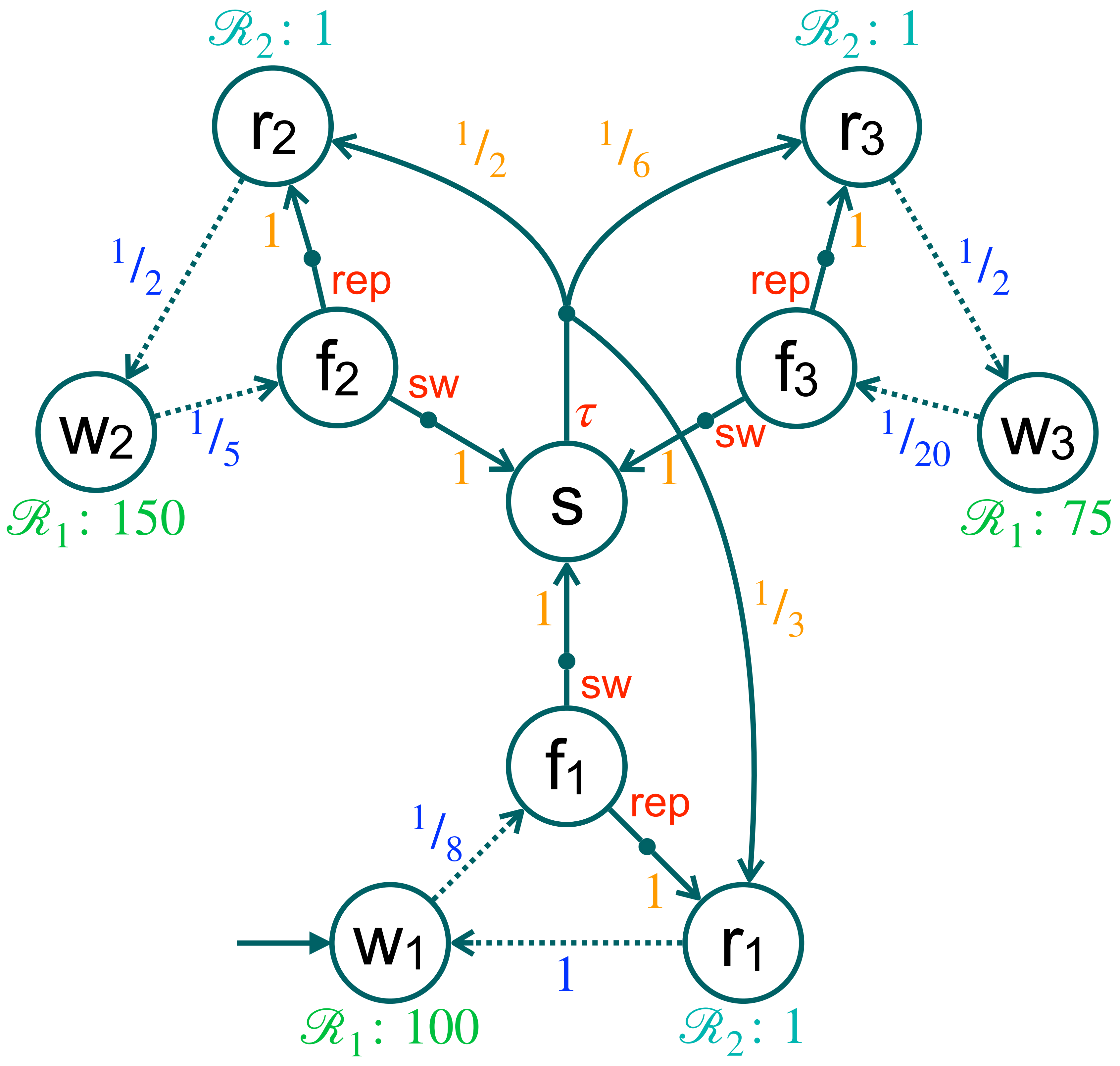
$$tot(\mathscr{R})\colon \pi \mapsto \lim_{k\to\infty} \mathscr{R}\big(pref(\pi,k)\big)$$

Long-run average reward objective:

$$lra(\mathscr{R})\colon \pi \mapsto \lim_{k\to\infty} \frac{\mathscr{R}\big(pref(\pi,k)\big)}{time\big(pref(\pi,k)\big)}$$

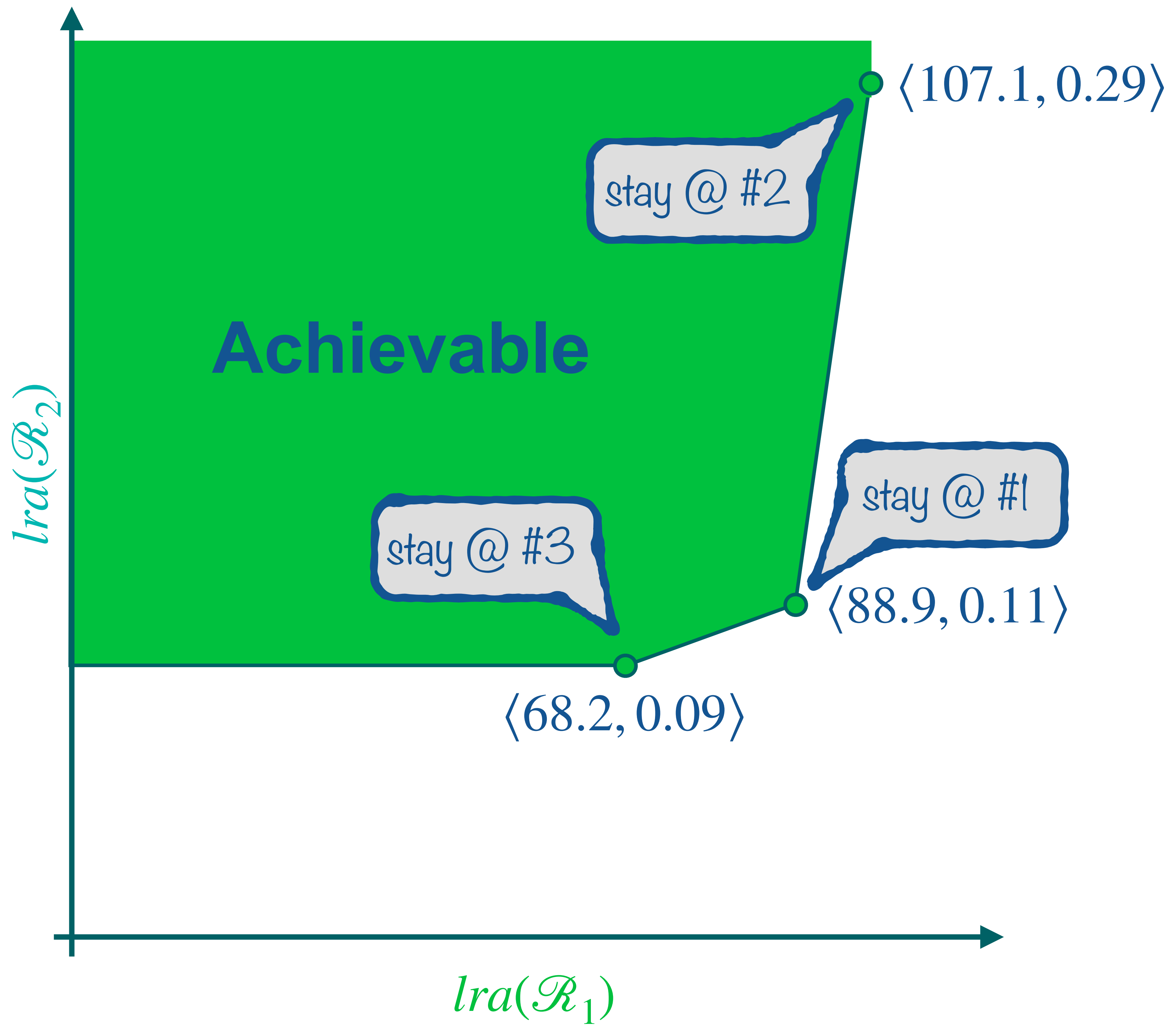


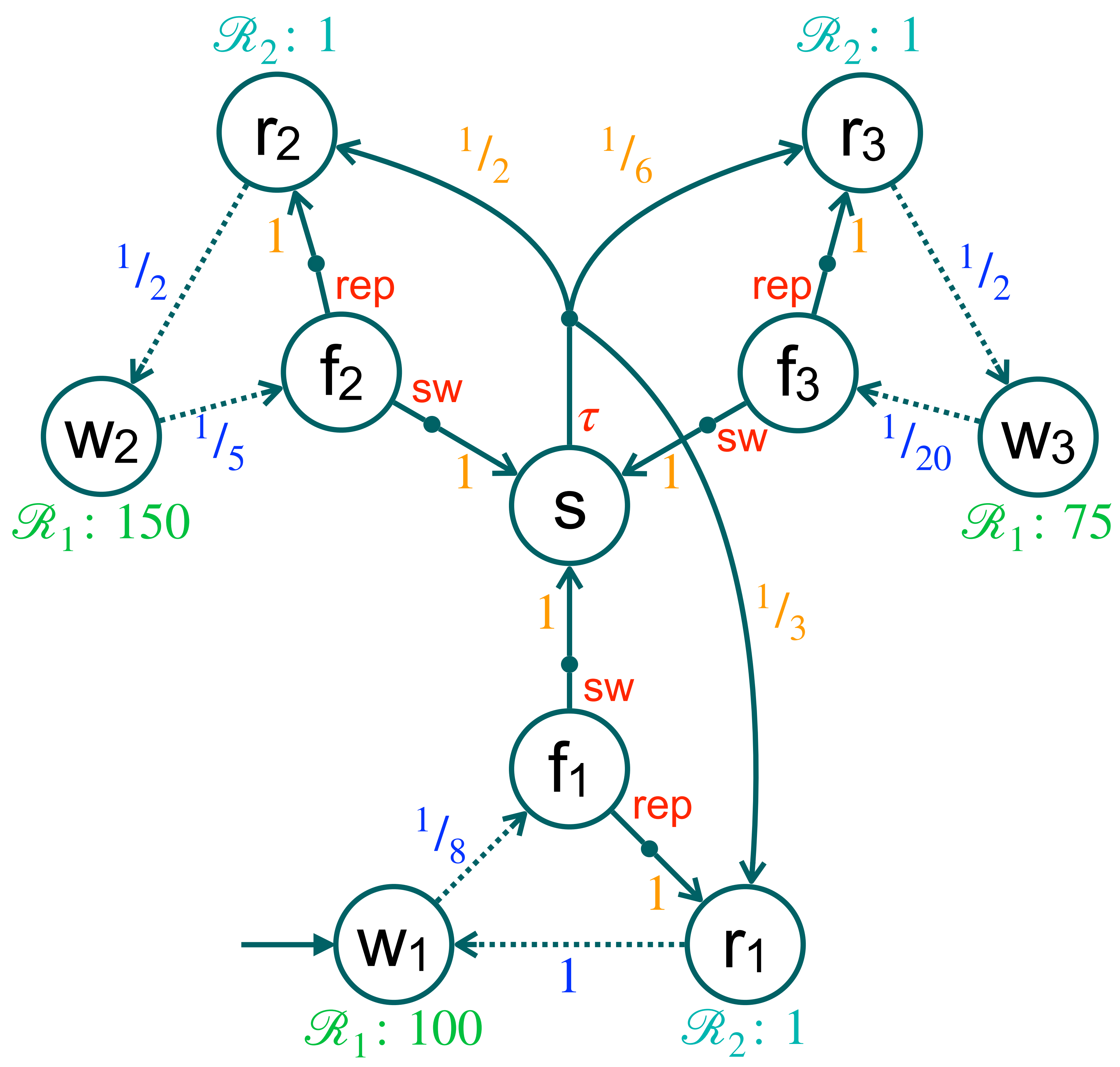
Achievable
stay @ #2
stay @ #1
stay @ #3
⟨107.1, 0.29⟩
⟨88.9, 0.11⟩
⟨68.2, 0.09⟩
Expected maintenance cost per day
Expected number of produced units per day



$\mathcal{R}_2$: 1
$\mathcal{R}_2$: 1
$\mathcal{R}_2$: 1
$\mathcal{R}_1$: 150
$\mathcal{R}_1$: 75
$\mathcal{R}_1$: 100
r2
r3
f2
f3
w2
w3
s
f1
w1
r1
rep
sw
τ
1/2
1/6
1/5
1/20
1/3
1/8
1

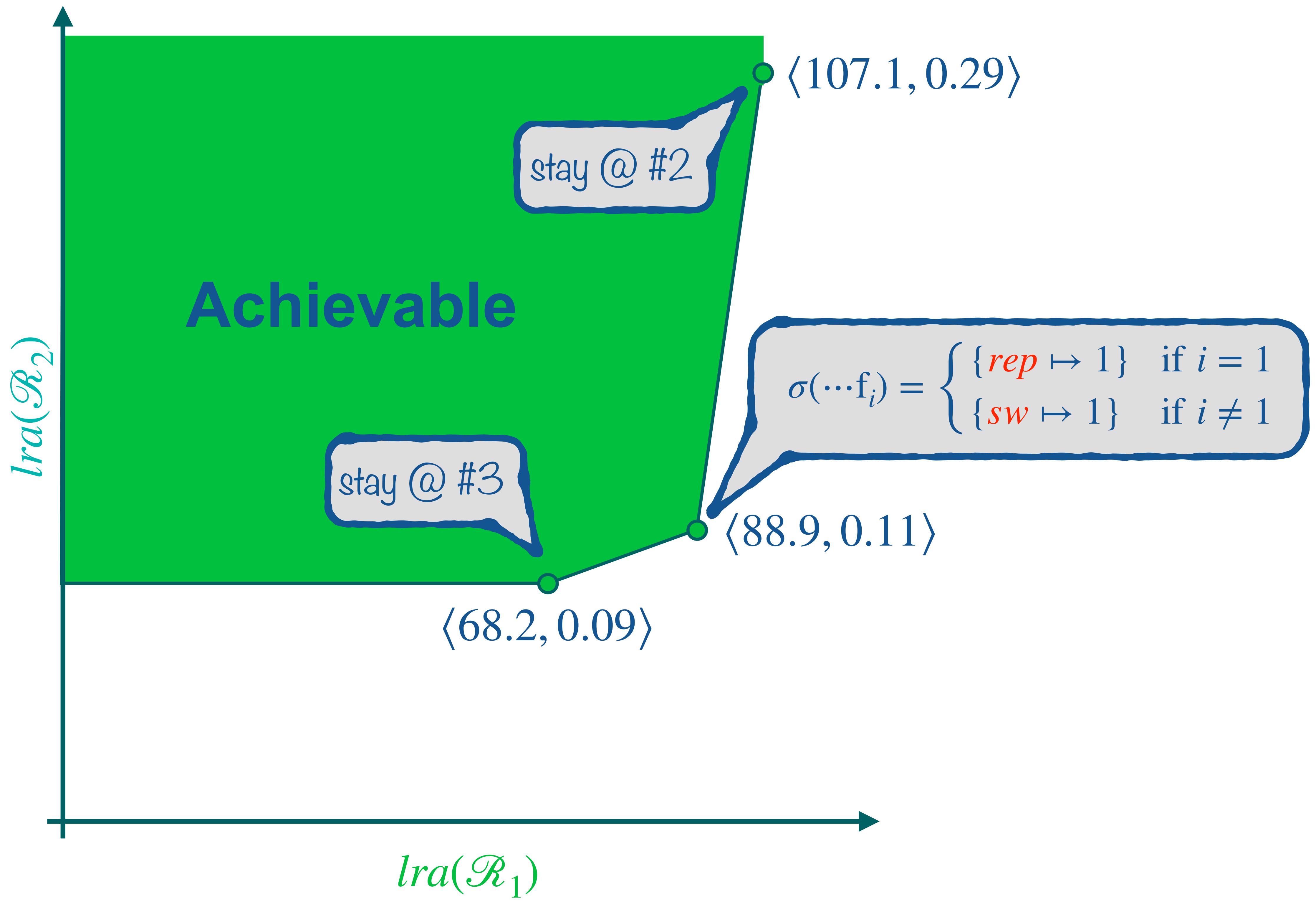
Markov Automata — Achievable Points

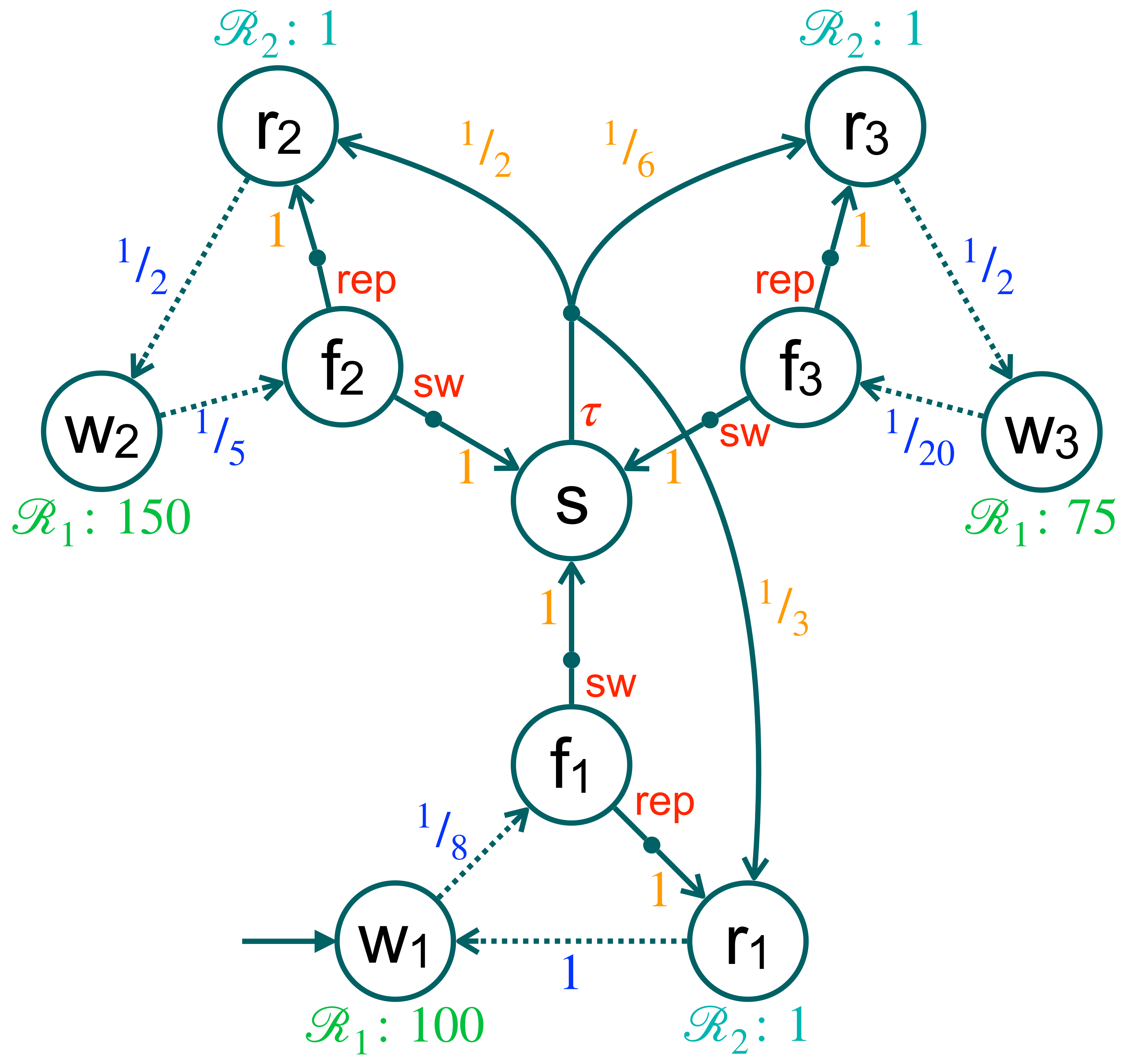




Tim Quatmann, Joost-Pieter Katoen

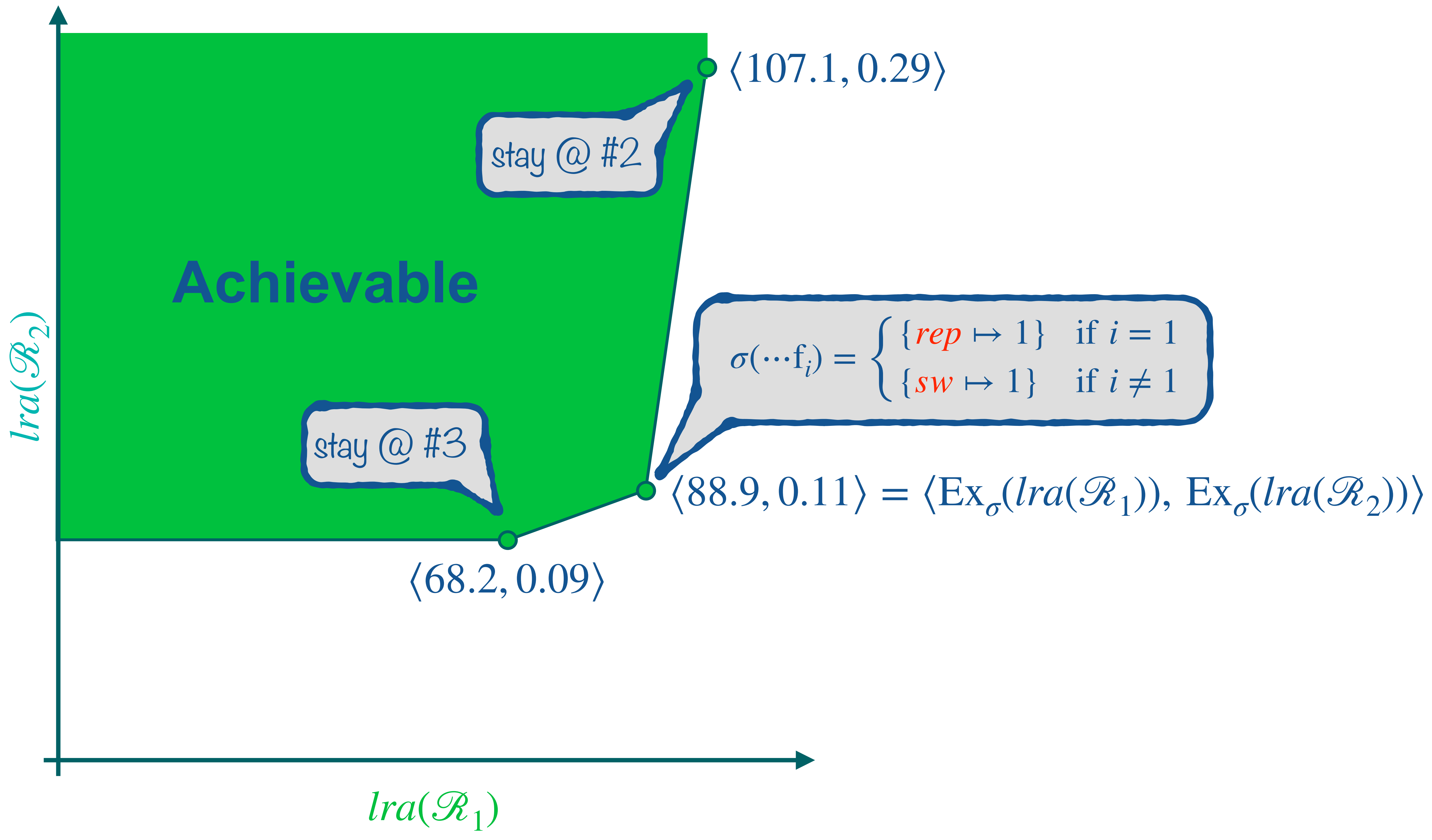
Markov Automata — Achievable Points

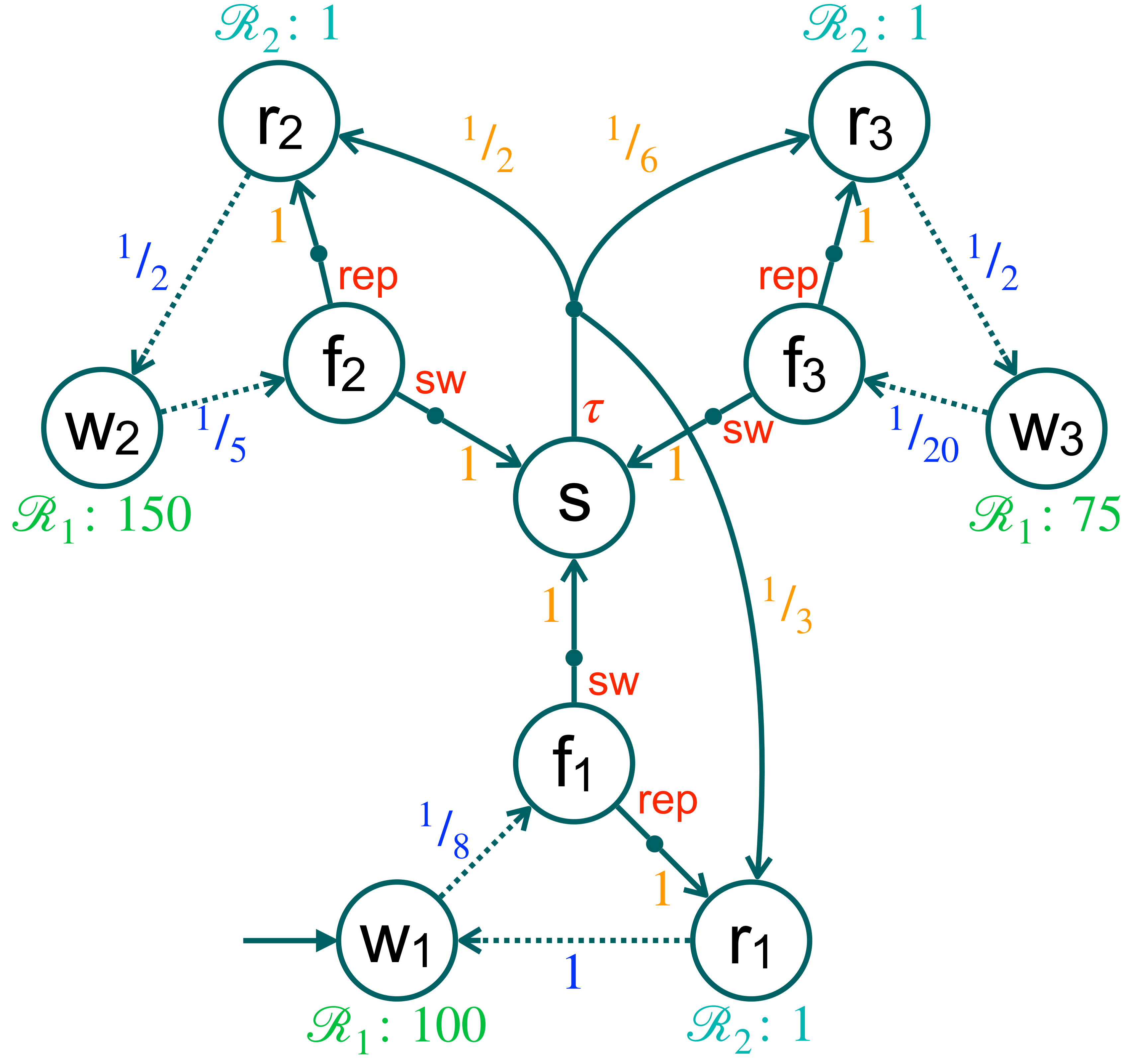




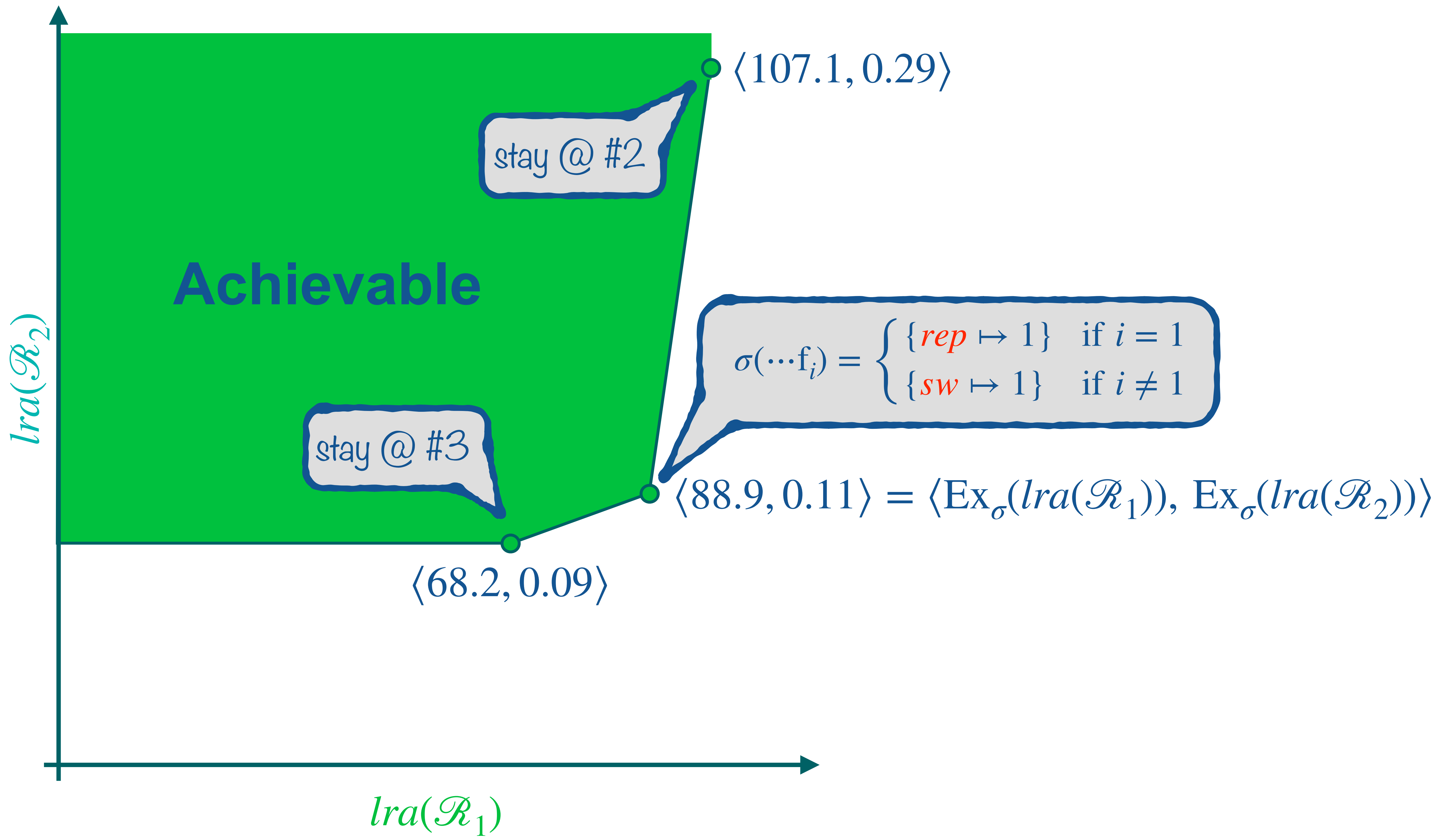
Tim Quatmann, Joost-Pieter Katoen
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

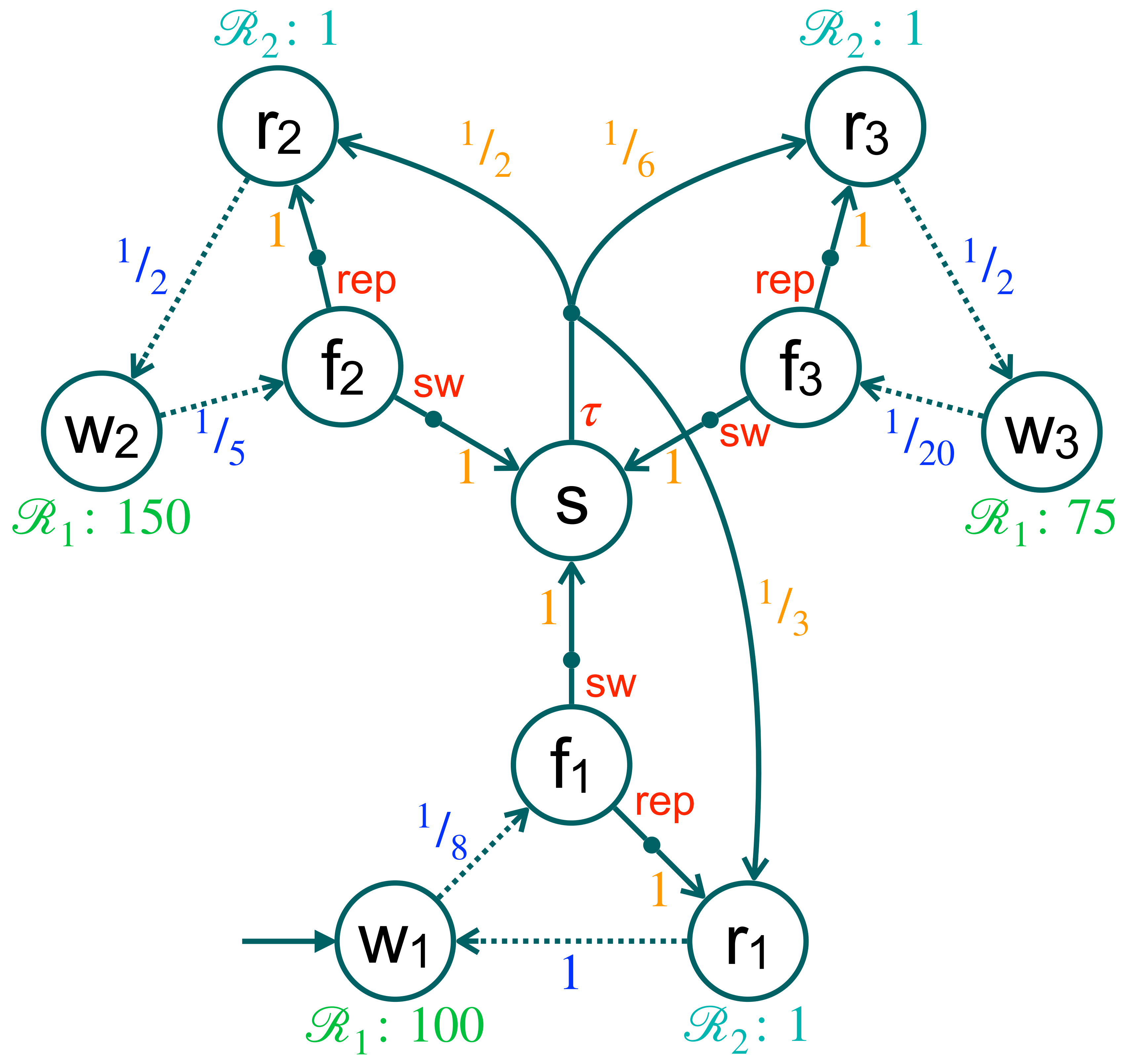
Markov Automata — Achievable Points





Markov Automata — Achievable Points





Set of **achievable points** for $\Phi = \langle f_1, \dots, f_\ell \rangle$:

$$Ach(\Phi) := \left\{ \mathbf{p} \in \mathbb{R}^\ell \;\middle|\; \exists \sigma \colon \mathbf{p} \leq \left\langle \mathrm{Ex}_\sigma(f_1), \;\dots,\; \mathrm{Ex}_\sigma(f_\ell) \right\rangle \right\}$$

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Multi-objective Model Checking

Set of **achievable points** for $\Phi = \langle f_1, \dots, f_\ell \rangle$:

$$Ach(\Phi) := \left\{ \mathbf{p} \in \mathbb{R}^\ell \;\middle|\; \exists \sigma \colon \mathbf{p} \leq \left\langle \mathrm{Ex}_\sigma(f_1), \; \dots, \; \mathrm{Ex}_\sigma(f_\ell) \right\rangle \right\}$$

Multi-objective Model Checking

Set of **achievable points** for $\Phi = \langle f_1, \ldots, f_\ell \rangle$:

$$Ach(\Phi) := \left\{ \mathbf{p} \in \mathbb{R}^\ell \;\middle|\; \exists \sigma \colon \mathbf{p} \leq \left\langle \mathrm{Ex}_\sigma(f_1),\ \ldots,\ \mathrm{Ex}_\sigma(f_\ell) \right\rangle \right\}$$

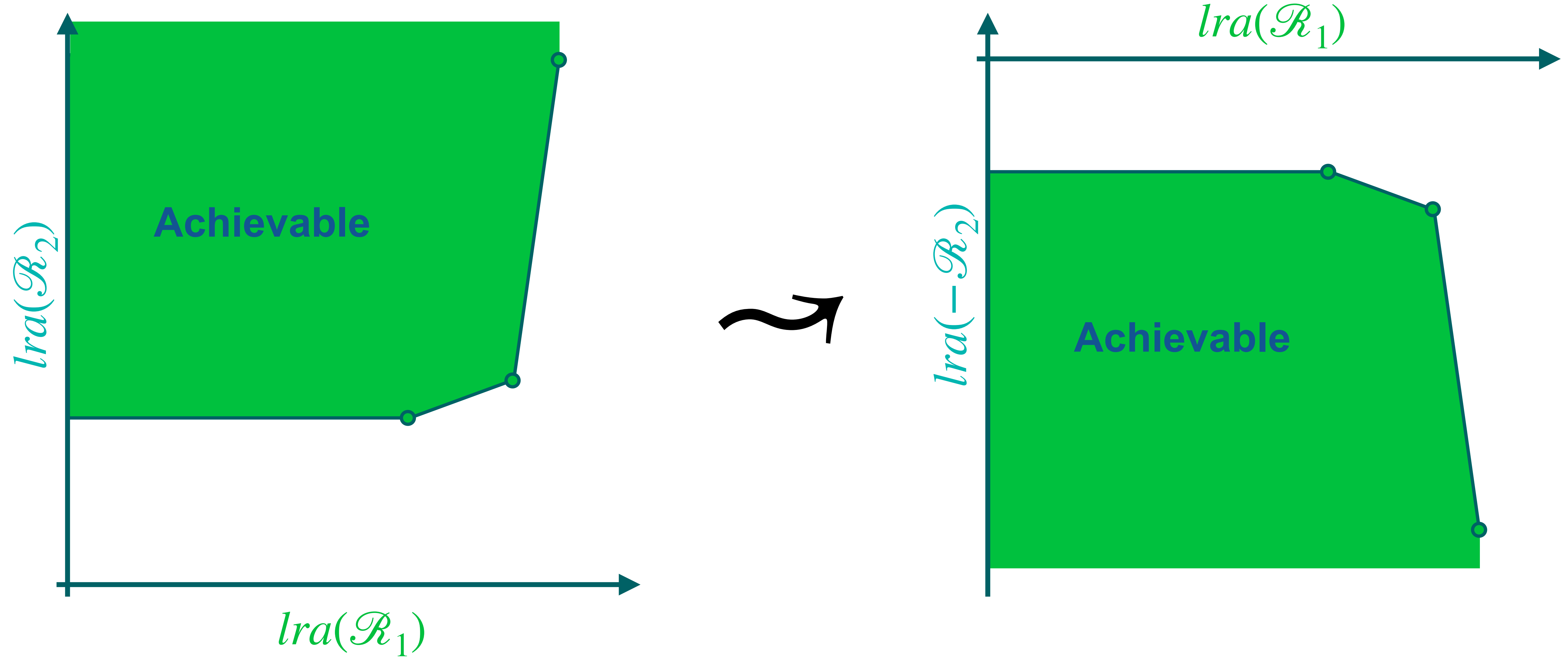
- Point $\mathbf{p}$ is **achievable** if there is a single strategy σ yielding expected values at least as large as $\mathbf{p}$
- Assumption: Large expected values $\mathrm{Ex}_\sigma(f_i)$ are "good"

Set of **achievable points** for $\Phi = \langle f_1, \ldots, f_\ell \rangle$:

$$Ach(\Phi) := \left\{ \mathbf{p} \in \mathbb{R}^\ell \;\middle|\; \exists \sigma\colon \mathbf{p} \leq \left\langle \mathrm{Ex}_\sigma(f_1),\; \ldots,\; \mathrm{Ex}_\sigma(f_\ell) \right\rangle \right\}$$

- Point $\mathbf{p}$ is **achievable** if there is a single strategy σ yielding expected values at least as large as $\mathbf{p}$
- Assumption: Large expected values $\mathrm{Ex}_\sigma(f_i)$ are "good"

negate minimizing objectives:
$tot(\mathcal{R}) \rightsquigarrow tot(-\mathcal{R})$
$lra(\mathcal{R}) \rightsquigarrow lra(-\mathcal{R})$



Set of **achievable points** for $\Phi = \langle f_1, \ldots, f_\ell \rangle$:

$$Ach(\Phi) := \left\{ \mathbf{p} \in \mathbb{R}^\ell \;\middle|\; \exists \sigma \colon \mathbf{p} \leq \left\langle \mathrm{Ex}_\sigma(f_1),\ \ldots,\ \mathrm{Ex}_\sigma(f_\ell) \right\rangle \right\}$$

- Point $\mathbf{p}$ is **achievable** if there is a single strategy σ yielding expected values at least as large as $\mathbf{p}$
- Assumption: Large expected values $\mathrm{Ex}_\sigma(f_i)$ are "good"

negate minimizing objectives:
$tot(\mathcal{R}) \rightsquigarrow tot(-\mathcal{R})$
$lra(\mathcal{R}) \rightsquigarrow lra(-\mathcal{R})$

Task: Compute an (approximation of) $Ach(\Phi)$.

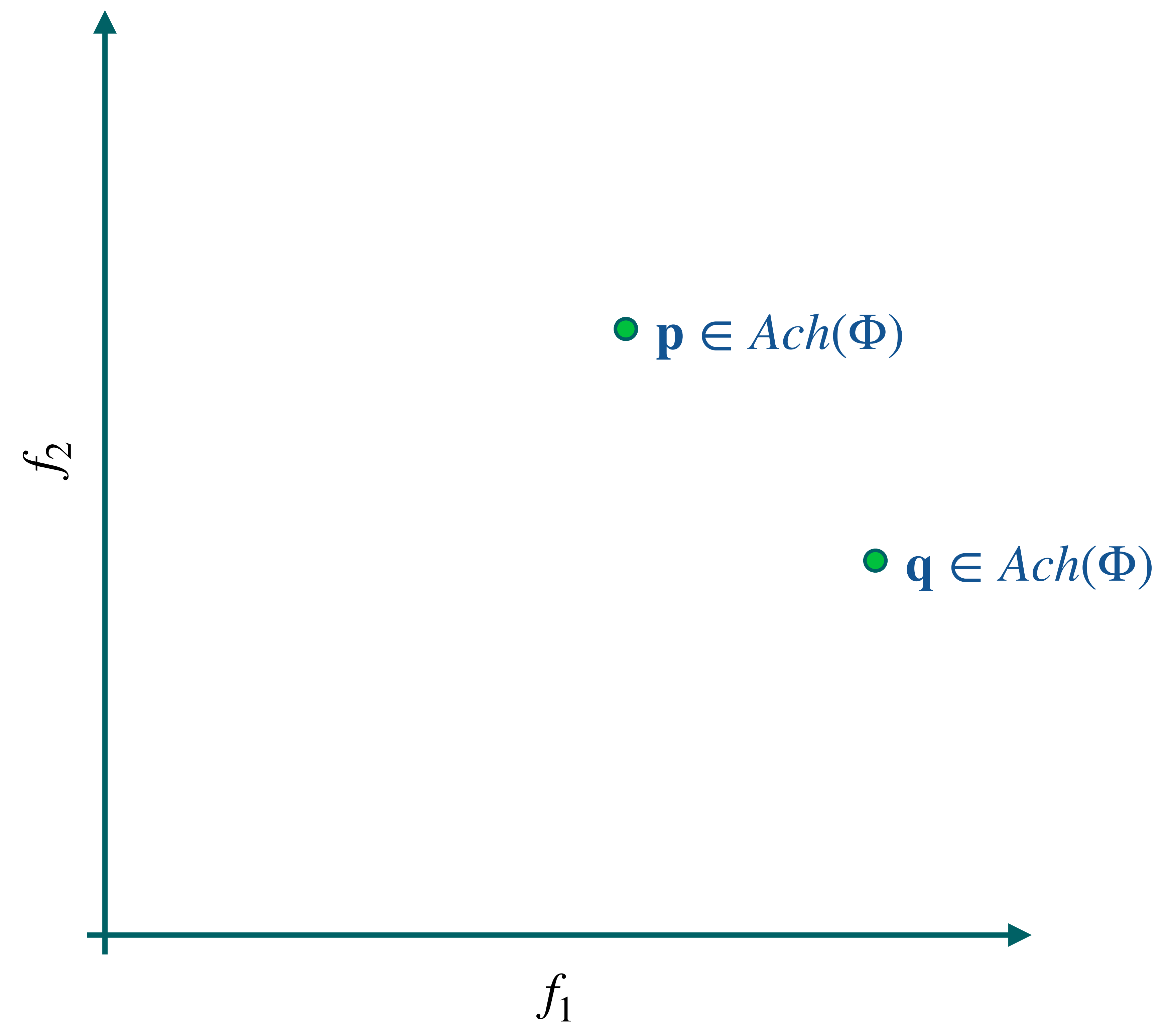
Properties of $Ach(\Phi)$

Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed**

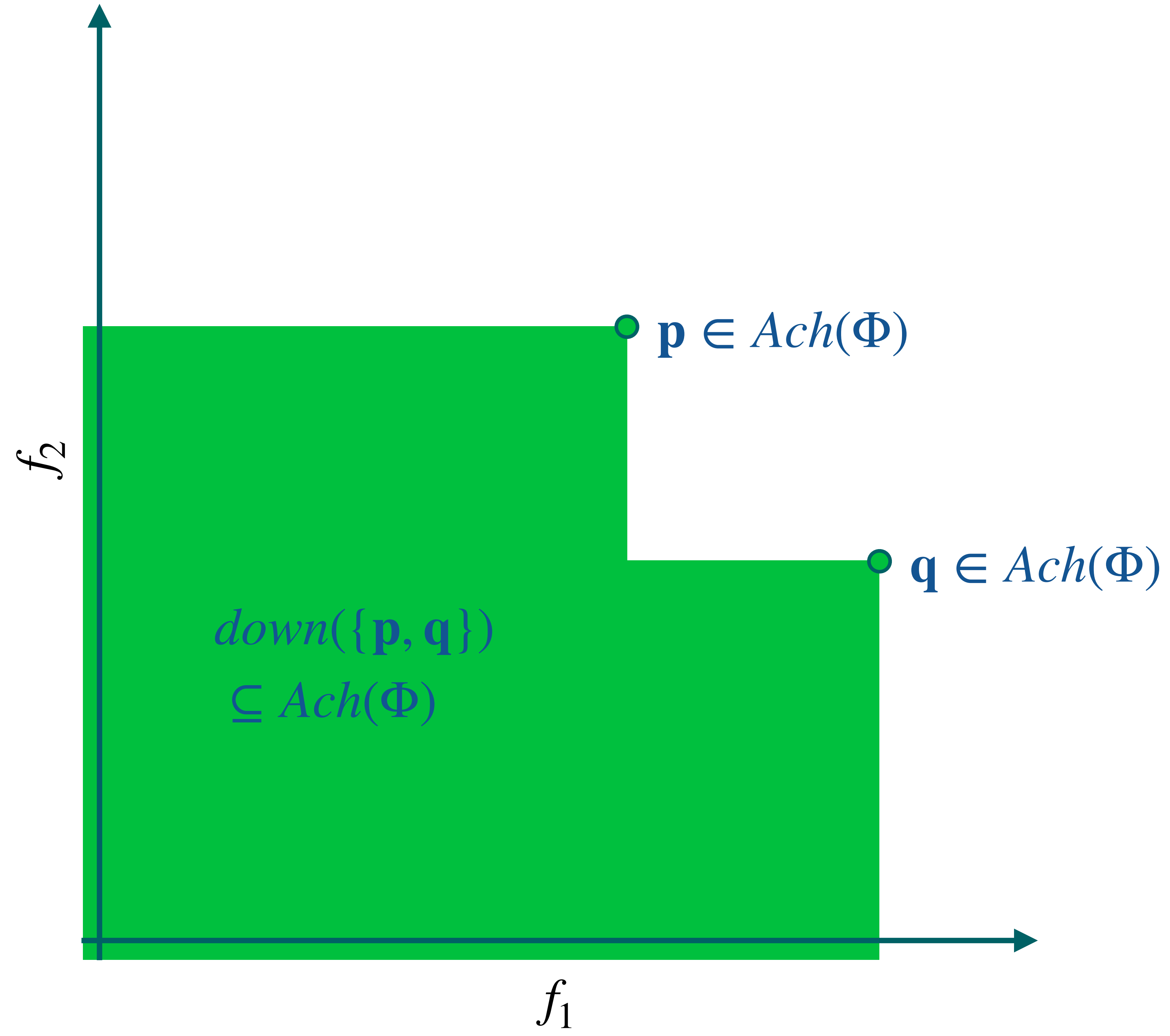
Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed**



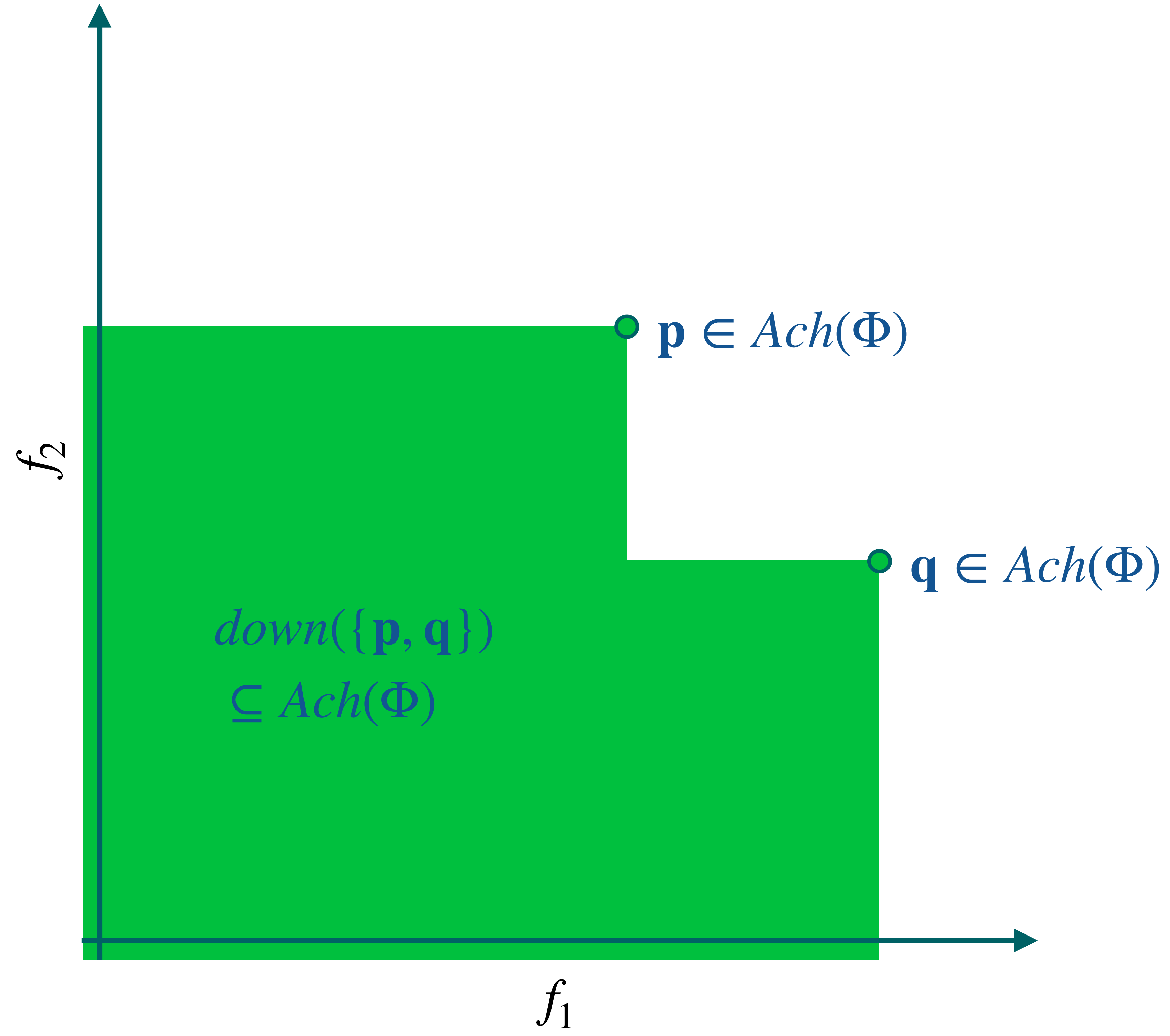
Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed**



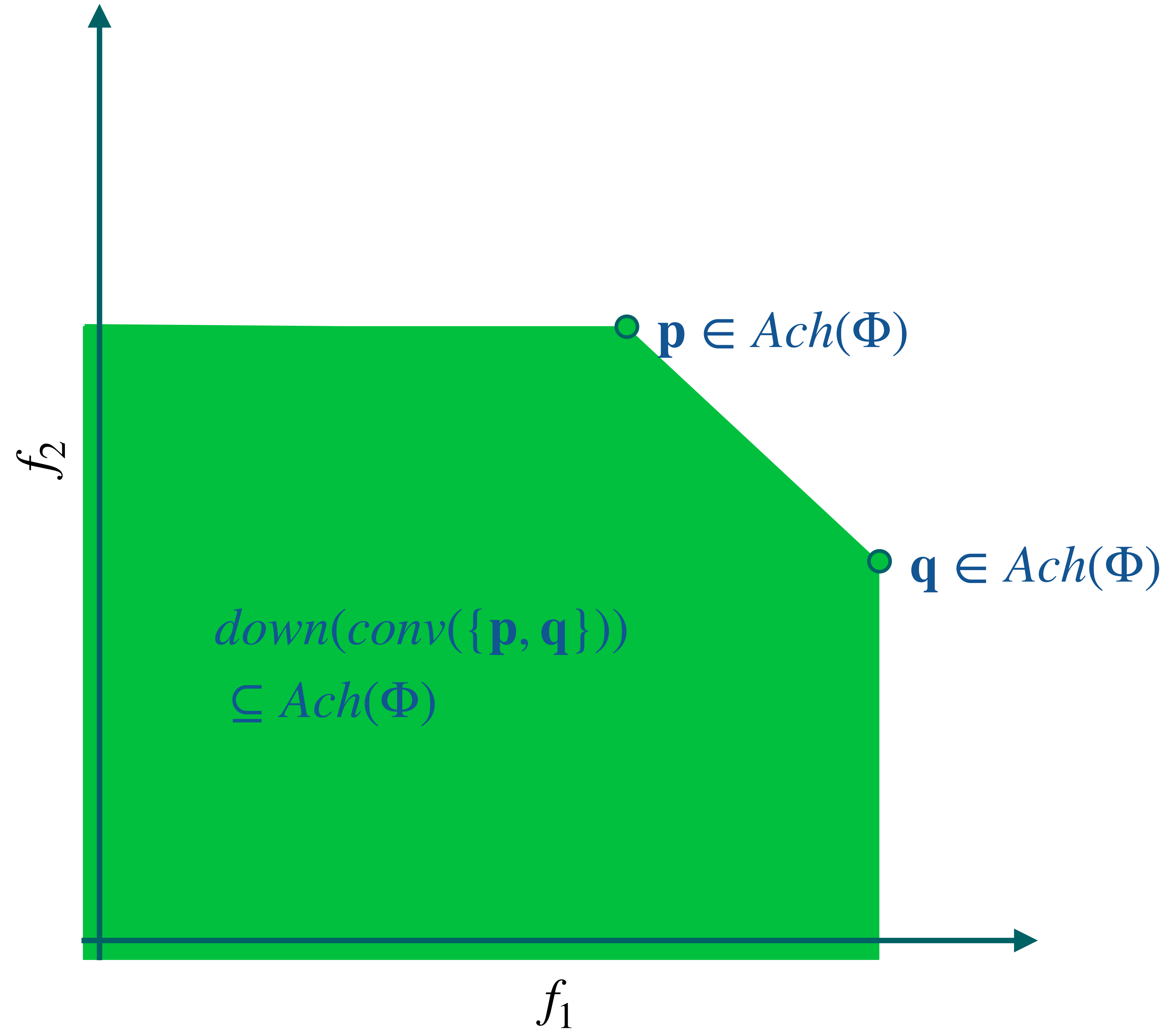
Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed** and **convex**



Properties of $Ach(\Phi)$

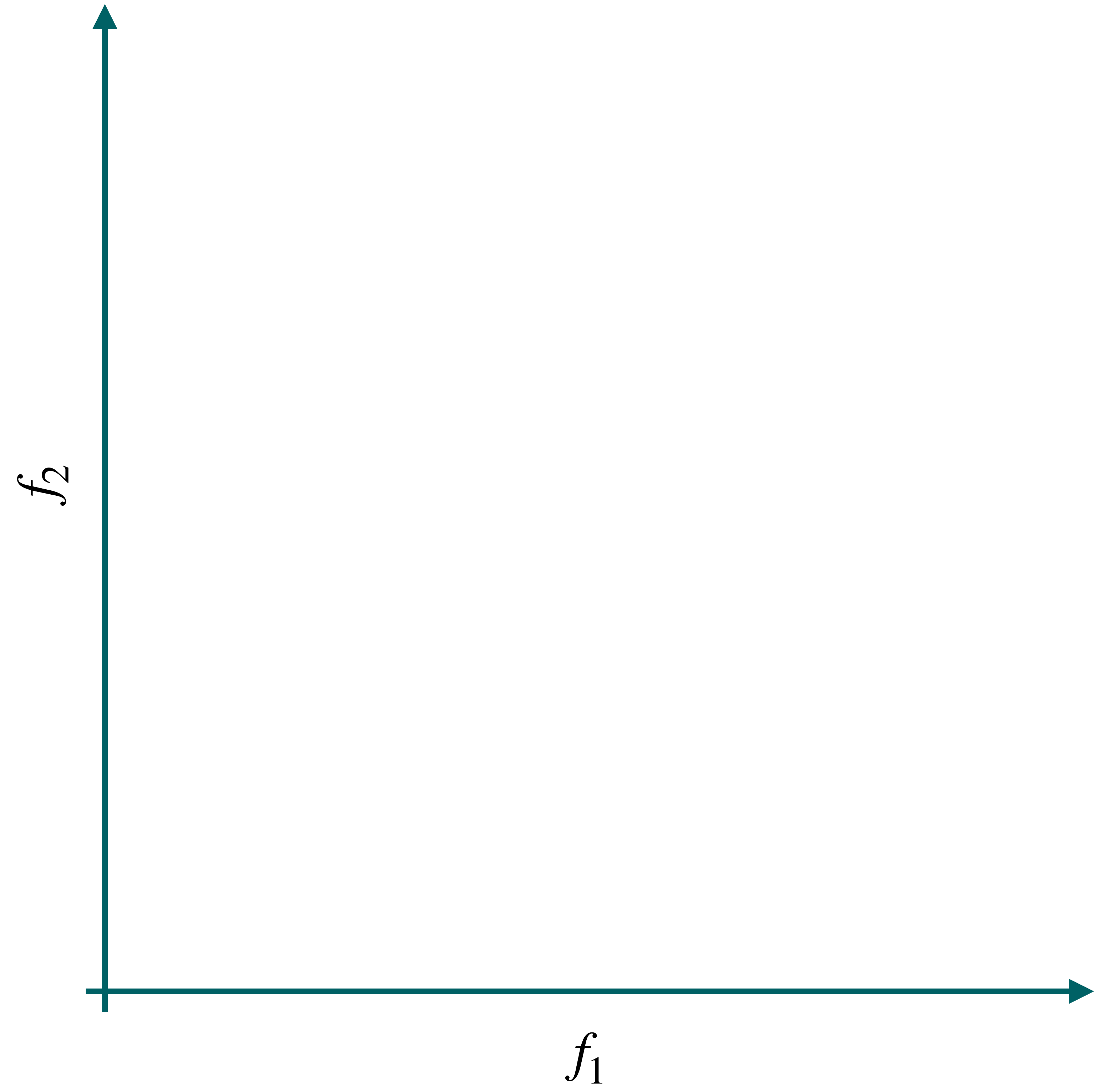
- $Ach(\Phi)$ is **downward closed** and **convex**



Properties of $Ach(\Phi)$

$\text{Ex}_\sigma(\Phi) := \langle \text{Ex}_\sigma(f_1), \ldots, \text{Ex}_\sigma(f_\ell) \rangle$

- $Ach(\Phi)$ is **downward closed** and **convex**
- For all $\mathbf{w} \in (\mathbb{R}_{\geq 0})^\ell$: $Ach(\Phi) \subseteq \left\{ \mathbf{p} \in \mathbb{R}^\ell \mid \mathbf{w} \cdot \mathbf{p} \leq \sup_\sigma \left(\mathbf{w} \cdot \text{Ex}_\sigma(\Phi) \right) \right\}$

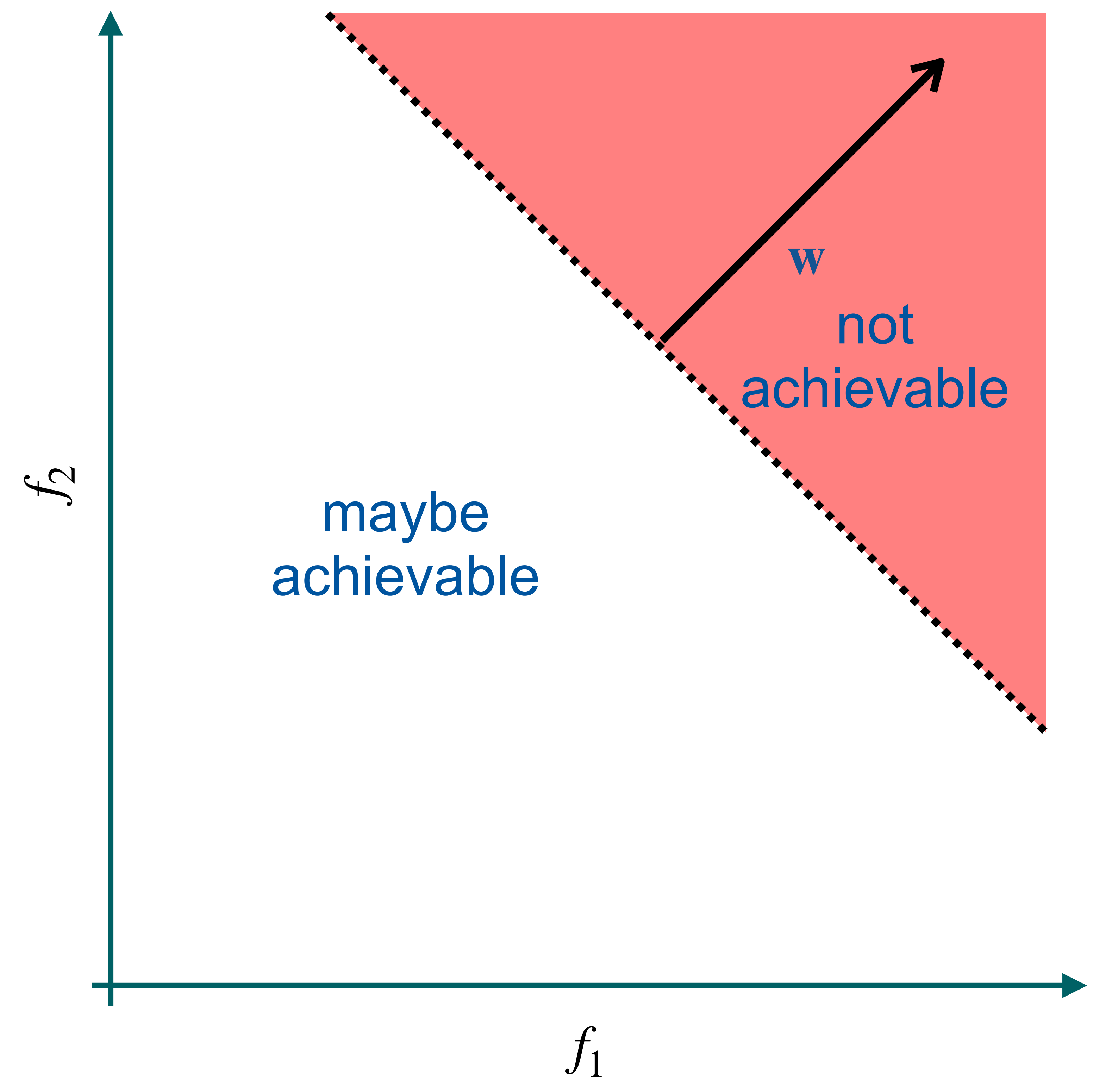


Tim Quatmann, Joost-Pieter Katoen

Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed** and **convex**
- For all $\mathbf{w} \in (\mathbb{R}_{\geq 0})^\ell$: $Ach(\Phi) \subseteq \left\{ \mathbf{p} \in \mathbb{R}^\ell \mid \mathbf{w} \cdot \mathbf{p} \leq \sup_\sigma \left(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi) \right) \right\}$

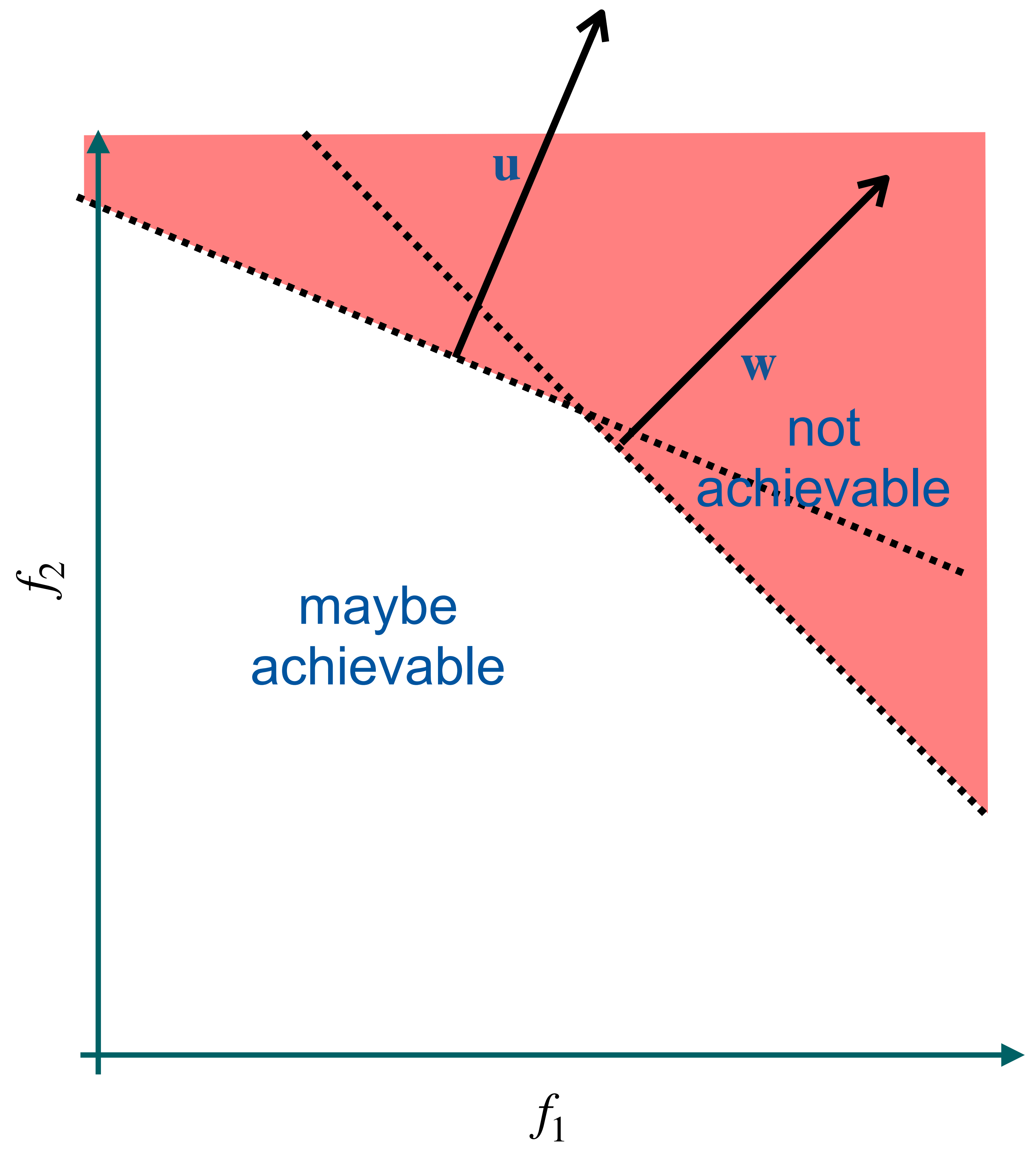
$\mathrm{Ex}_\sigma(\Phi) := \left\langle \mathrm{Ex}_\sigma(f_1), \ldots, \mathrm{Ex}_\sigma(f_\ell) \right\rangle$



Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed** and **convex**
- For all $\mathbf{w} \in (\mathbb{R}_{\geq 0})^\ell$: $Ach(\Phi) \subseteq \left\{\mathbf{p} \in \mathbb{R}^\ell \mid \mathbf{w} \cdot \mathbf{p} \leq \sup_\sigma\left(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\right)\right\}$

$\mathrm{Ex}_\sigma(\Phi) := \left\langle \mathrm{Ex}_\sigma(f_1), \ldots, \mathrm{Ex}_\sigma(f_\ell) \right\rangle$



Tim Quatmann, Joost-Pieter Katoen
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

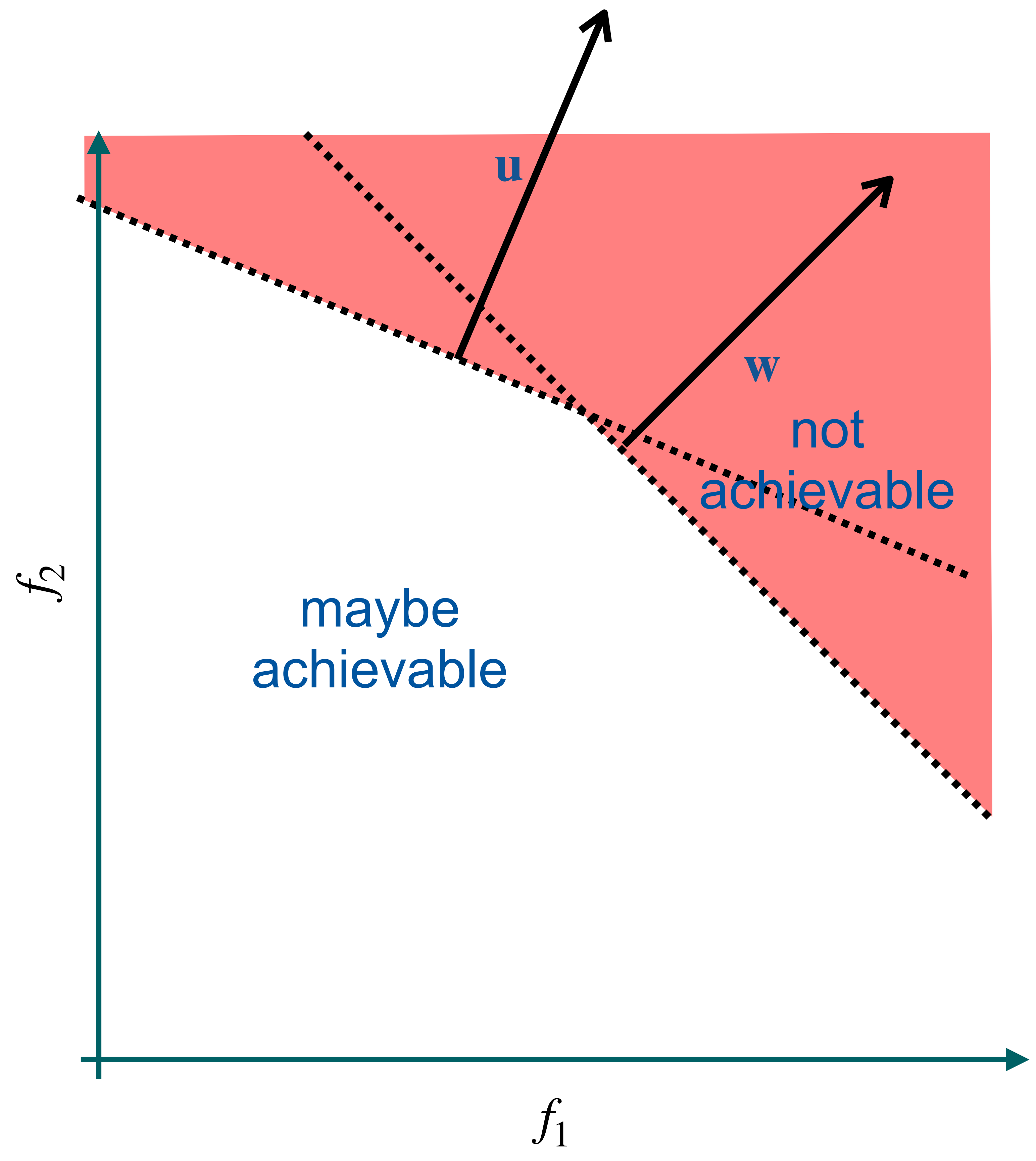
Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed** and **convex**

$\mathrm{Ex}_\sigma(\Phi) := \langle \mathrm{Ex}_\sigma(f_1), \ldots, \mathrm{Ex}_\sigma(f_\ell) \rangle$

- For all $\mathbf{w} \in (\mathbb{R}_{\geq 0})^\ell$: $Ach(\Phi) \subseteq \{\mathbf{p} \in \mathbb{R}^\ell \mid \mathbf{w} \cdot \mathbf{p} \leq \sup_\sigma(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi))\}$

- $Ach(\Phi)$ is **closed**—assuming that $\forall f_i$:
  - $f_i \in \{tot(\mathscr{R}_j), lra(\mathscr{R}_j)\}$ and …
  - $\forall \sigma$: $\mathrm{Ex}_\sigma(f_i) \leq +\infty$



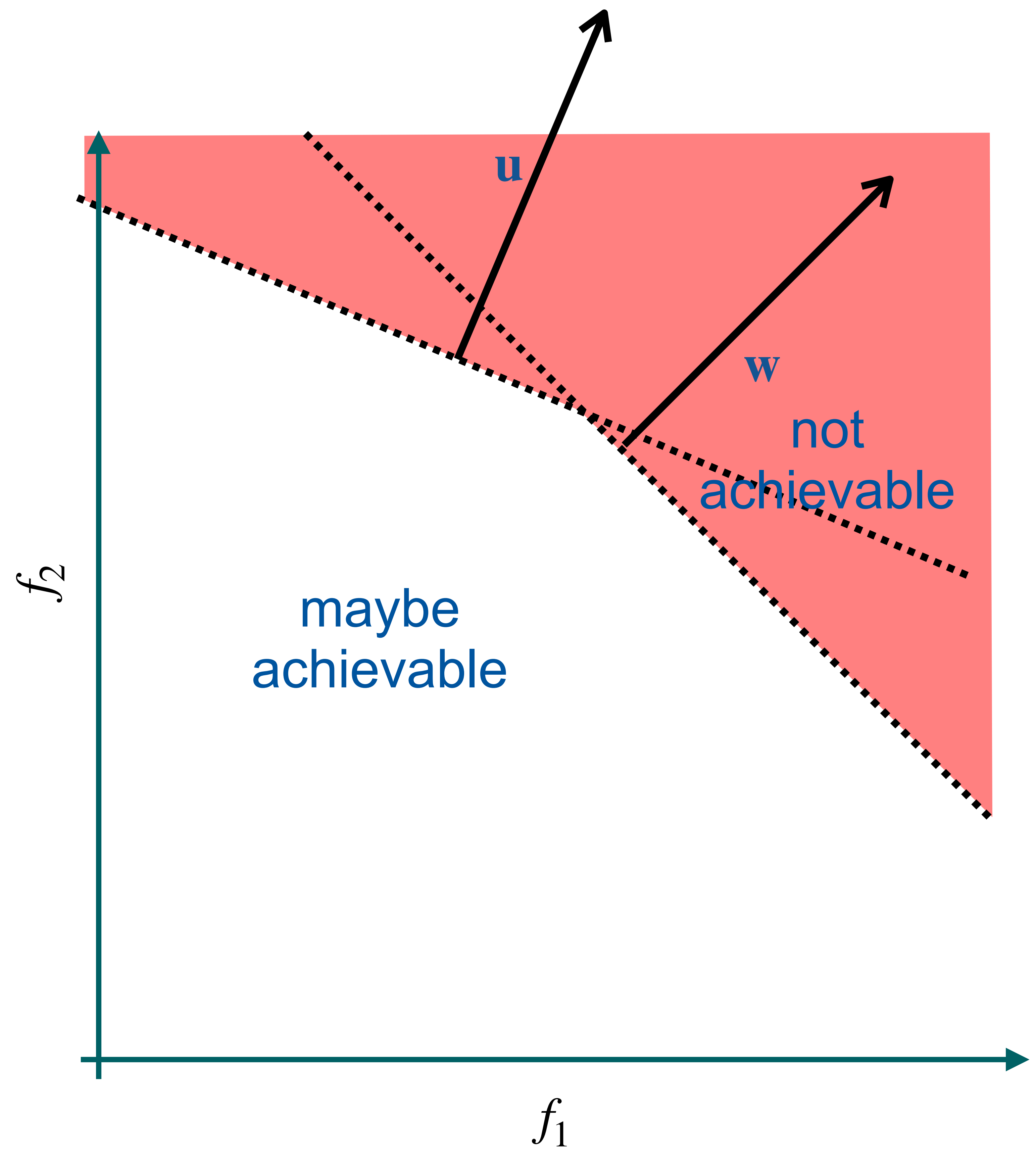
Properties of $Ach(\Phi)$

- $Ach(\Phi)$ is **downward closed** and **convex**
- For all $\mathbf{w} \in (\mathbb{R}_{\geq 0})^\ell$: $Ach(\Phi) \subseteq \{\mathbf{p} \in \mathbb{R}^\ell \mid \mathbf{w} \cdot \mathbf{p} \leq \sup_\sigma(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi))\}$

  max

  $\mathrm{Ex}_\sigma(\Phi) := \langle \mathrm{Ex}_\sigma(f_1), \ldots, \mathrm{Ex}_\sigma(f_\ell) \rangle$

- $Ach(\Phi)$ is **closed**—assuming that $\forall f_i$:
  - $f_i \in \{tot(\mathscr{R}_j),\ lra(\mathscr{R}_j)\}$ and …
  - $\forall \sigma$: $\mathrm{Ex}_\sigma(f_i) \leq +\infty$



Sandwich Algorithm

[Solanki, Appino, & Cohon'93; Forejt, Kwiatkowska, & Parker'12]

convex multi-objective optimization

MDP + total rewards

Sandwich Algorithm

[Solanki, Appino, & Cohon'93; Forejt, Kwiatkowska, & Parker'12]

- Compute $\sigma_{\mathbf{w}} \in \arg\max_{\sigma} \left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi) \right)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^{\ell}$

$$\rightsquigarrow \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi) \in Ach(\Phi) \subseteq \left\{ \mathbf{p} \in \mathbb{R}^{\ell} \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi) \right\}$$

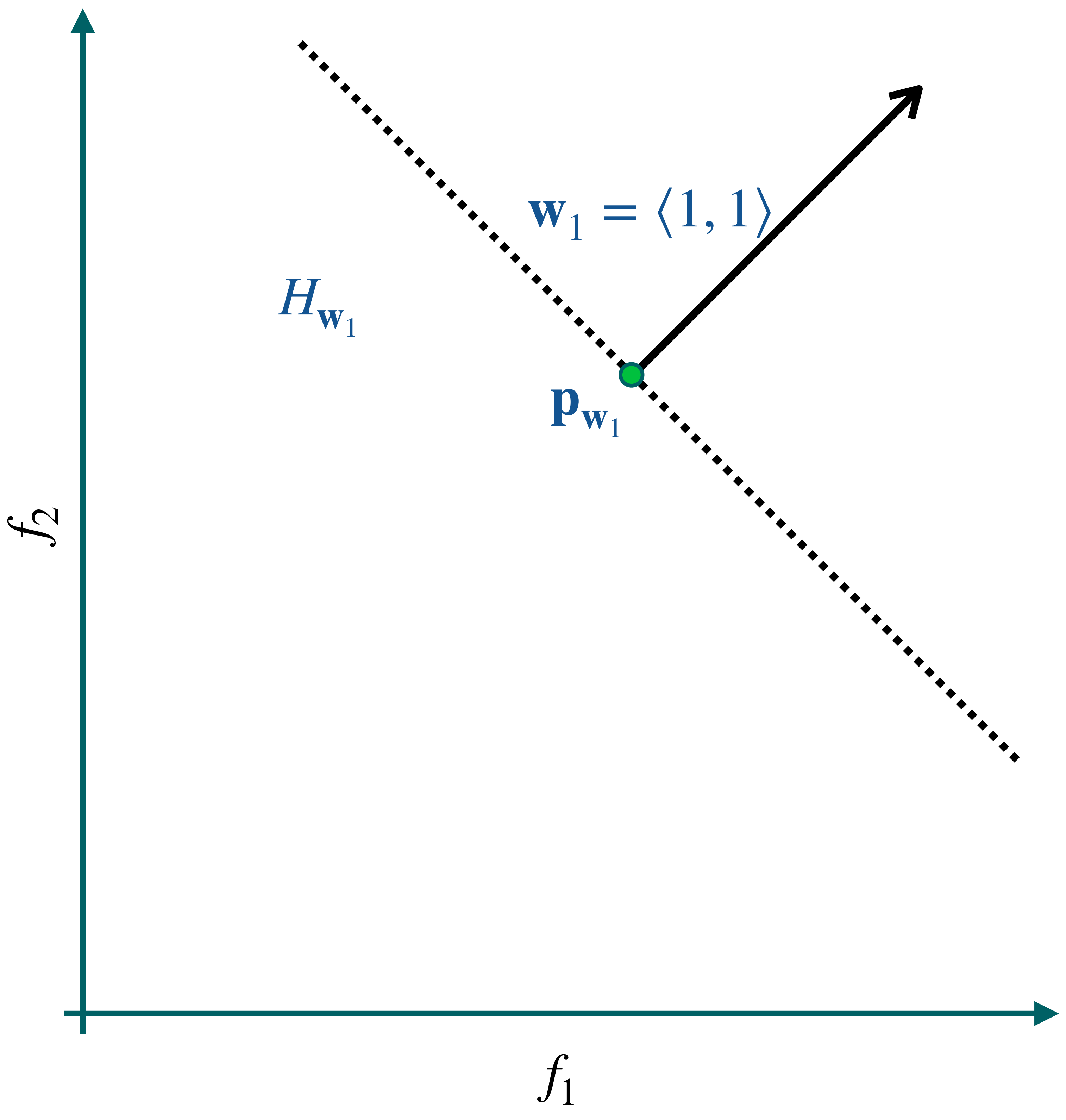
- Compute $\sigma_{\mathbf{w}} \in \arg\max_{\sigma} \left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi) \right)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^{\ell}$

$$\rightsquigarrow \underbrace{\mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)}_{=:\mathbf{p}_{\mathbf{w}}} \in Ach(\Phi) \subseteq \underbrace{\left\{ \mathbf{p} \in \mathbb{R}^{\ell} \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi) \right\}}_{=:H_{\mathbf{w}}}$$

[Solanki, Appino, & Cohon'93; Forejt, Kwiatkowska, & Parker'12]

- Compute $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^{\ell}$

$$\rightsquigarrow \underbrace{\mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)}_{=:\mathbf{p}_{\mathbf{w}}} \in Ach(\Phi) \subseteq \underbrace{\left\{\mathbf{p} \in \mathbb{R}^{\ell} \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)\right\}}_{=:H_{\mathbf{w}}}$$

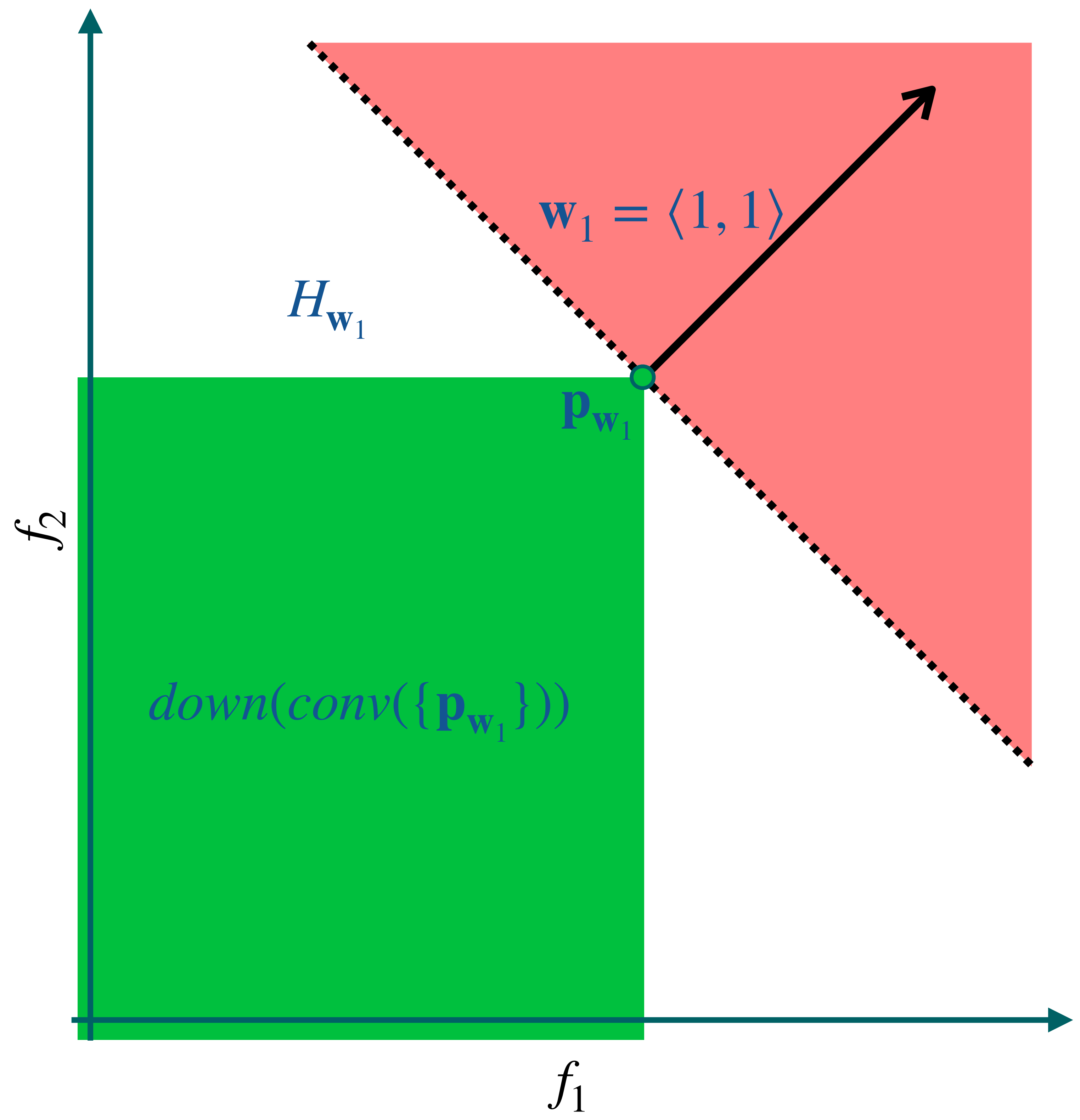


Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

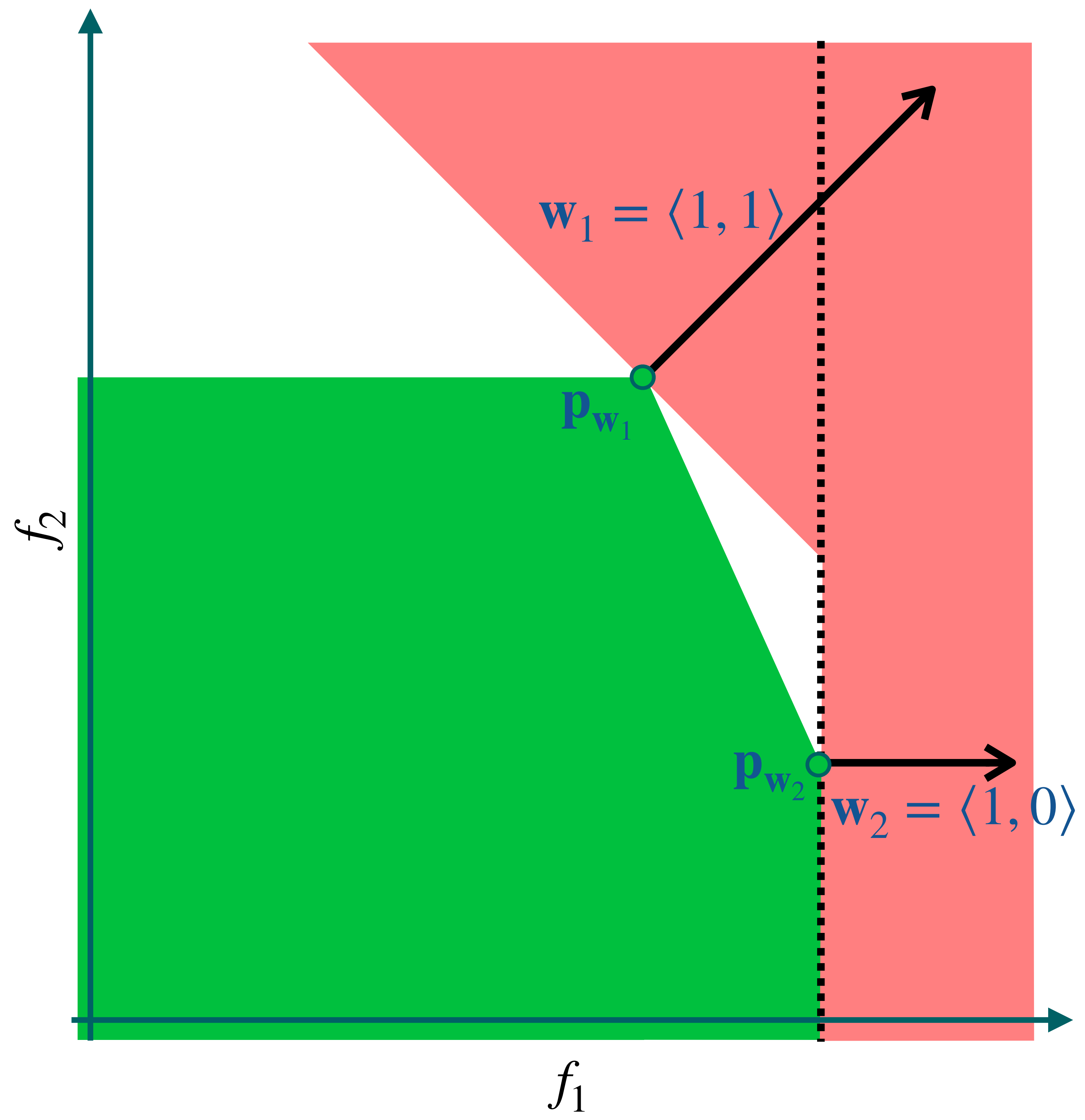
- Compute $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^{\ell}$

$$\rightsquigarrow \underbrace{\mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)}_{=:\mathbf{p}_{\mathbf{w}}} \in Ach(\Phi) \subseteq \underbrace{\left\{\mathbf{p} \in \mathbb{R}^{\ell} \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)\right\}}_{=:H_{\mathbf{w}}}$$



- Compute $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^{\ell}$

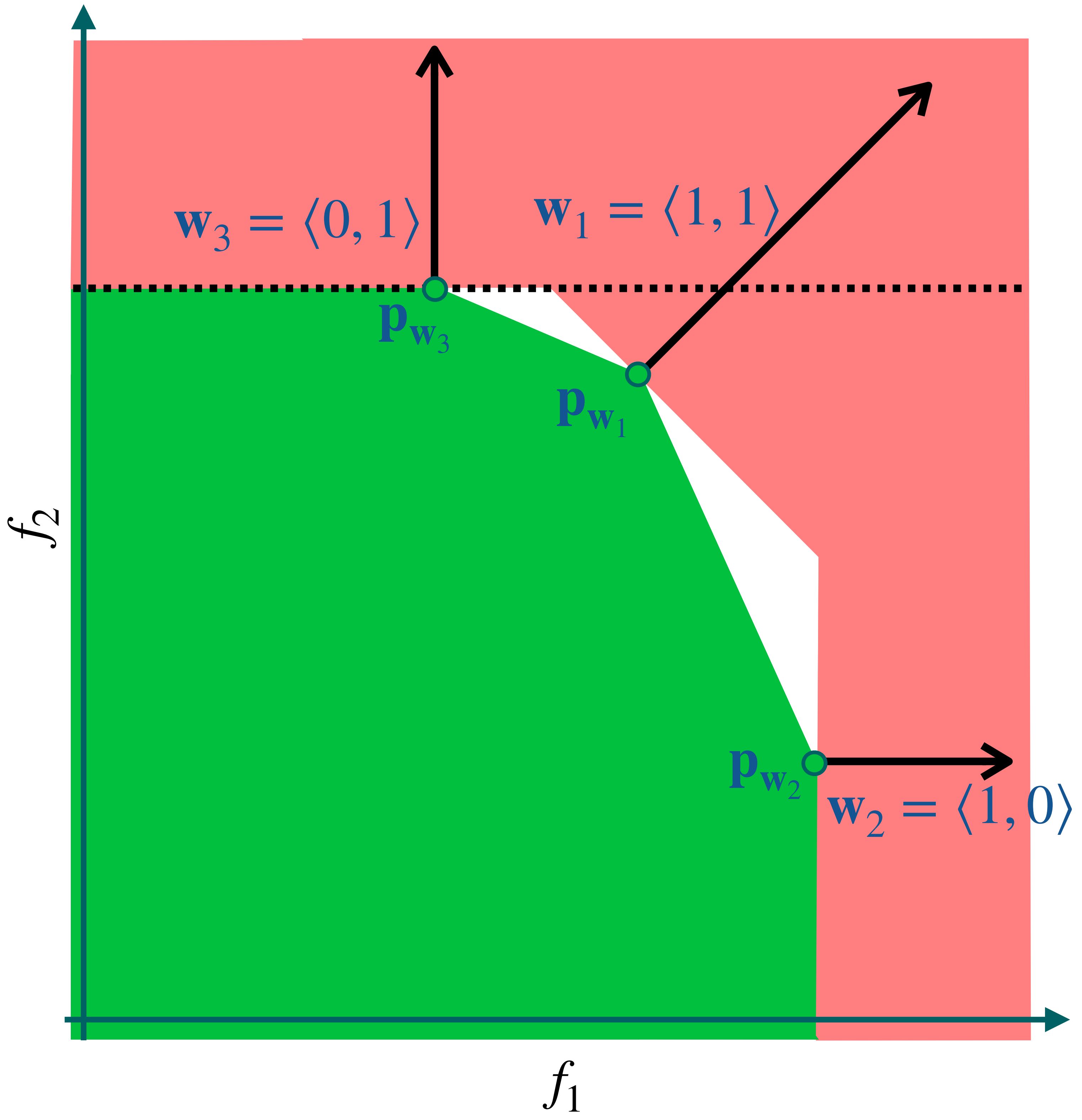
$$\rightsquigarrow \underbrace{\mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)}_{=:\mathbf{p}_{\mathbf{w}}} \in \mathit{Ach}(\Phi) \subseteq \underbrace{\left\{\mathbf{p} \in \mathbb{R}^{\ell} \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)\right\}}_{=:H_{\mathbf{w}}}$$



Sandwich Algorithm

- Compute $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\big(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\big)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^{\ell}$

$$\rightsquigarrow \underbrace{\mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)}_{=:\mathbf{p}_{\mathbf{w}}} \in Ach(\Phi) \subseteq \underbrace{\{\mathbf{p} \in \mathbb{R}^{\ell} \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)\}}_{=:H_{\mathbf{w}}}$$

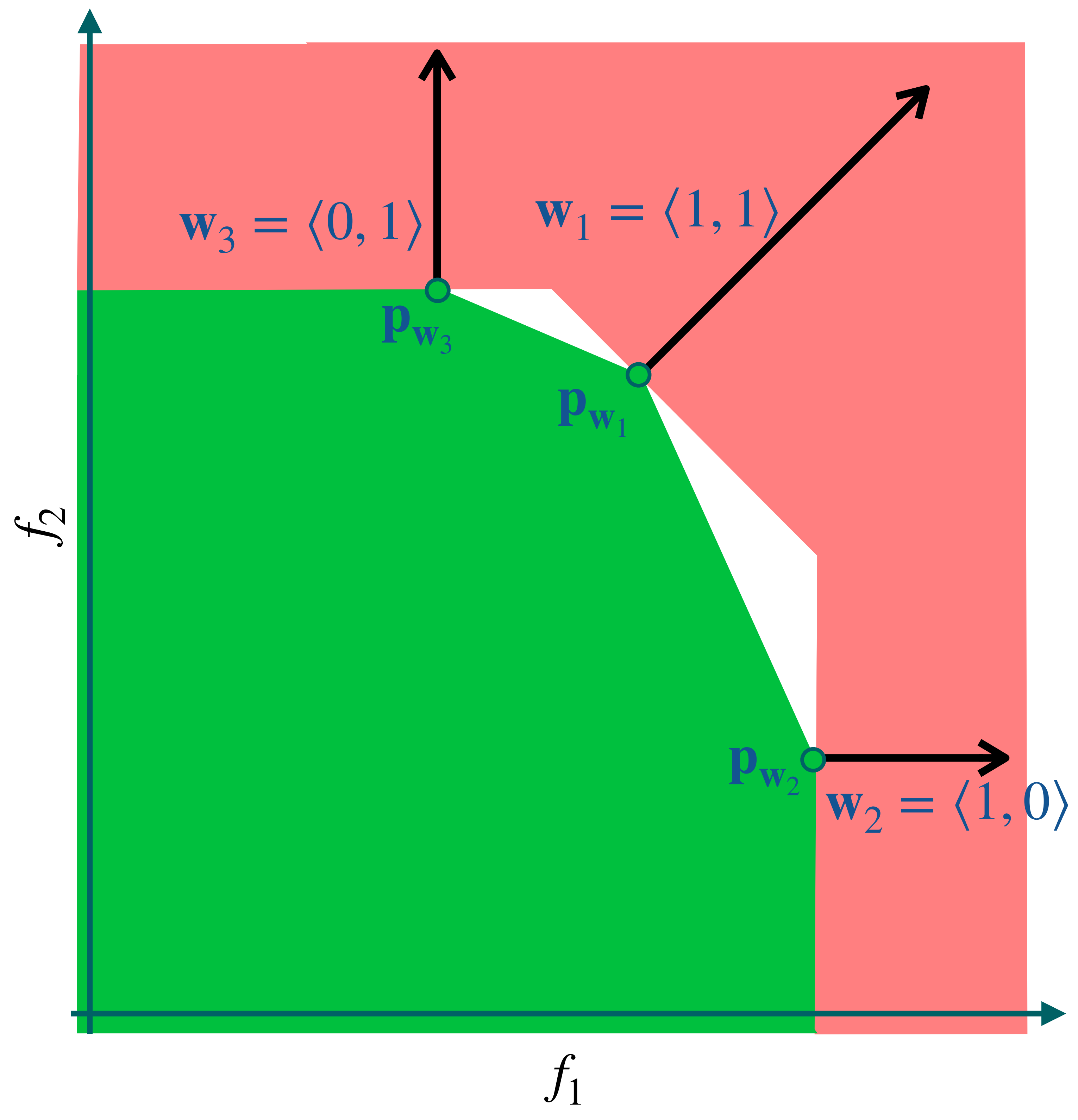


- Compute $\sigma_{\mathbf{w}} \in \arg\max_\sigma\left(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\right)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^\ell$

$$\rightsquigarrow \underbrace{\mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)}_{=:\mathbf{p}_{\mathbf{w}}} \in \mathit{Ach}(\Phi) \subseteq \underbrace{\left\{\mathbf{p} \in \mathbb{R}^\ell \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)\right\}}_{=:H_{\mathbf{w}}}$$

- Invariant:

$$\mathit{down}\left(\mathit{conv}\left(\bigcup_{\mathbf{w}} \{\mathbf{p}_{\mathbf{w}}\}\right)\right) \subseteq \mathit{Ach}(\Phi) \subseteq \bigcap_{\mathbf{w}} H_{\mathbf{w}}$$



[Solanki, Appino, & Cohon'93; Forejt, Kwiatkowska, & Parker'12]

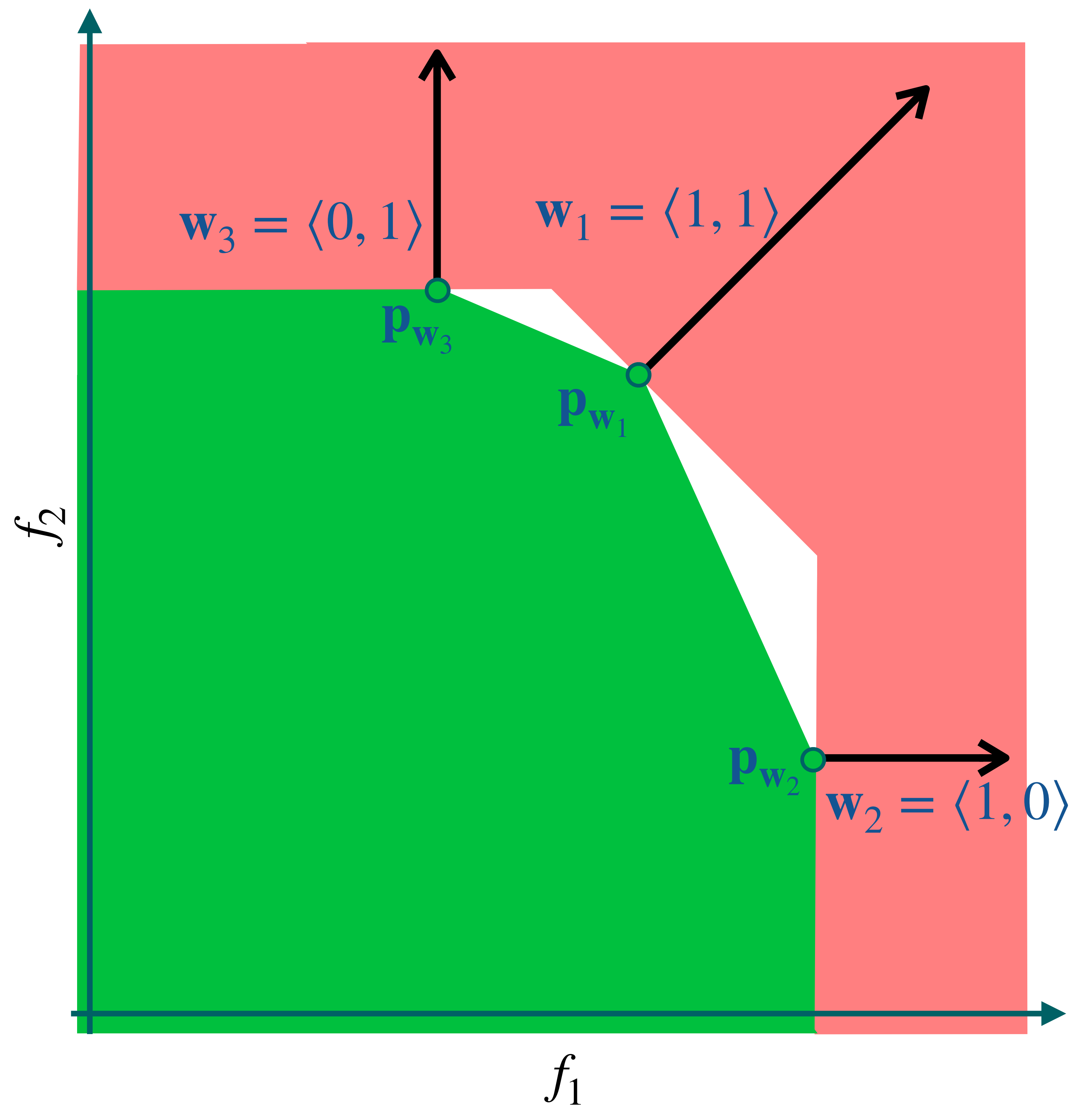
- Compute $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$ for different weight vectors $\mathbf{w} \in (\mathbb{R}_{\geq 0})^{\ell}$

$$\rightsquigarrow \underbrace{\mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)}_{=:\mathbf{p}_{\mathbf{w}}} \in Ach(\Phi) \subseteq \underbrace{\left\{\mathbf{p} \in \mathbb{R}^{\ell} \mid \mathbf{w} \cdot \mathbf{p} \leq \mathbf{w} \cdot \mathrm{Ex}_{\sigma_{\mathbf{w}}}(\Phi)\right\}}_{=:H_{\mathbf{w}}}$$

- Invariant:

$$down\left(conv\left(\bigcup_{\mathbf{w}} \{\mathbf{p}_{\mathbf{w}}\}\right)\right) \subseteq Ach(\Phi) \subseteq \bigcap_{\mathbf{w}} H_{\mathbf{w}}$$

- Stop when approximation of $Ach(\Phi)$ is sufficiently precise



Optimizing Weighted Sums

- Approach is applicable to **all** kinds of objectives $\Phi = \langle f_1, \ldots, f_\ell \rangle, \quad f_i \colon \mathrm{Paths}_{\mathrm{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$

Optimizing Weighted Sums

- Approach is applicable to **all** kinds of objectives $\Phi = \langle f_1, \ldots, f_\ell \rangle, \quad f_i \colon \mathrm{Paths_{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$
- Core: an algorithm to compute $\sigma_{\mathbf{w}} \in \arg\max_\sigma \big(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\big)$

Optimizing Weighted Sums

- Approach is applicable to **all** kinds of objectives $\Phi = \langle f_1, \ldots, f_\ell \rangle, \quad f_i \colon \text{Paths}_{\text{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$
- Core: an algorithm to compute $\sigma_{\mathbf{w}} \in \arg\max_\sigma \big(\mathbf{w} \cdot \text{Ex}_\sigma(\Phi)\big)$
- Problem: The product $\mathbf{w} \cdot \text{Ex}_\sigma(\Phi)$ might be $\pm\infty$ or undefined ("$\infty - \infty$") ... however …

Optimizing Weighted Sums

- Approach is applicable to **all** kinds of objectives $\Phi = \langle f_1, \ldots, f_\ell \rangle, \quad f_i \colon \mathrm{Paths}_{\mathrm{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$
- Core: an algorithm to compute $\sigma_{\mathbf{w}} \in \arg\max_\sigma \big(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\big)$
- Problem: The product $\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)$ might be $\pm\infty$ or undefined ("$\infty - \infty$") ... however …
  - If $\mathrm{Ex}_\sigma(f_i) = -\infty$, no point $\mathbf{p} \in \mathbb{R}^\ell$ will be achieved by σ
    $\rightsquigarrow$ Ignore such strategies σ — *needs to be enforced algorithmically*

Optimizing Weighted Sums

- Approach is applicable to **all** kinds of objectives $\Phi = \langle f_1, \ldots, f_\ell \rangle$, $\quad f_i\colon \mathrm{Paths}_{\mathrm{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$

- Core: an algorithm to compute $\sigma_{\mathbf{w}} \in \arg\max_\sigma \big(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\big)$

- Problem: The product $\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)$ might be $\pm\infty$ or undefined ("$\infty - \infty$") ... however …
  - If $\mathrm{Ex}_\sigma(f_i) = -\infty$, no point $\mathbf{p} \in \mathbb{R}^\ell$ will be achieved by σ
    $\rightsquigarrow$ Ignore such strategies σ (needs to be enforced algorithmically)
  - If $\mathrm{Ex}_\sigma(f_i) = +\infty$, mimic σ with a small prob. $\varepsilon > 0$; focus on remaining objectives with prob. $1 - \varepsilon$
    $\rightsquigarrow$ Ignore such objectives f_i

Tim Quatmann, Joost-Pieter Katoen
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Optimizing Weighted Sums

- Approach is applicable to **all** kinds of objectives $\Phi = \langle f_1, \ldots, f_\ell \rangle$, $\quad f_i \colon \mathrm{Paths}_{\mathrm{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$
- Core: an algorithm to compute $\sigma_{\mathbf{w}} \in \arg\max_\sigma \big(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\big)$
- Problem: The product $\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)$ might be $\pm\infty$ or undefined ("$\infty - \infty$") ... however …
  - If $\mathrm{Ex}_\sigma(f_i) = -\infty$, no point $\mathbf{p} \in \mathbb{R}^\ell$ will be achieved by σ
    $\rightsquigarrow$ Ignore such strategies σ — *needs to be enforced algorithmically*
  - If $\mathrm{Ex}_\sigma(f_i) = +\infty$, mimic σ with a small prob. $\varepsilon > 0$; focus on remaining objectives with prob. $1 - \varepsilon$
    $\rightsquigarrow$ Ignore such objectives f_i — *$Ach(\Phi)$ is not necessarily closed anymore*

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Optimizing Weighted Sums

- Approach is applicable to **all** kinds of objectives $\Phi = \langle f_1, \ldots, f_\ell \rangle, \quad f_i \colon \mathrm{Paths}_{\mathrm{inf}} \to \mathbb{R} \cup \{-\infty, +\infty\}$
- Core: an algorithm to compute $\sigma_{\mathbf{w}} \in \arg\max_\sigma \big(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\big)$
- Problem: The product $\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)$ might be $\pm\infty$ or undefined ("$\infty - \infty$") ... however …
  - If $\mathrm{Ex}_\sigma(f_i) = -\infty$, no point $\mathbf{p} \in \mathbb{R}^\ell$ will be achieved by σ
    $\rightsquigarrow$ Ignore such strategies σ — *needs to be enforced algorithmically*
  - If $\mathrm{Ex}_\sigma(f_i) = +\infty$, mimic σ with a small prob. $\varepsilon > 0$; focus on remaining objectives with prob. $1 - \varepsilon$
    $\rightsquigarrow$ Ignore such objectives f_i — *$Ach(\Phi)$ is not necessarily closed anymore*

From now assume that $\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi) \in \mathbb{R}$ is well-defined.

Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\big(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\big)$

[Forejt, Kwiatkowska, & Parker'12]

For $\Phi_{tot} = \langle tot(\mathscr{R}_1), \ldots, tot(\mathscr{R}_\ell)\rangle$:

Push the weighted sum to the rewards: $\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi_{tot}) = \mathrm{Ex}_{\sigma}\Big(tot\big(\overbrace{\textstyle\sum_{i=1}^{\ell} \mathbf{w}[i] \cdot \mathscr{R}_i}^{=:\mathscr{R}_{\mathbf{w}}}\big)\Big)$

$\rightsquigarrow$ Use single objective methods to get $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\big(\mathrm{Ex}_{\sigma}(tot(\mathscr{R}_{\mathbf{w}}))\big)$

Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$

[Forejt, Kwiatkowska, & Parker'12]

For $\Phi_{tot} = \langle tot(\mathscr{R}_1), \dots, tot(\mathscr{R}_\ell)\rangle$:

Push the weighted sum to the rewards: $\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi_{tot}) = \mathrm{Ex}_{\sigma}\Big(tot\big(\overbrace{\textstyle\sum_{i=1}^{\ell} \mathbf{w}[i] \cdot \mathscr{R}_i}^{=: \mathscr{R}_{\mathbf{w}}}\big)\Big)$

$\leadsto$ Use single objective methods to get $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathrm{Ex}_{\sigma}(tot(\mathscr{R}_{\mathbf{w}}))\right)$

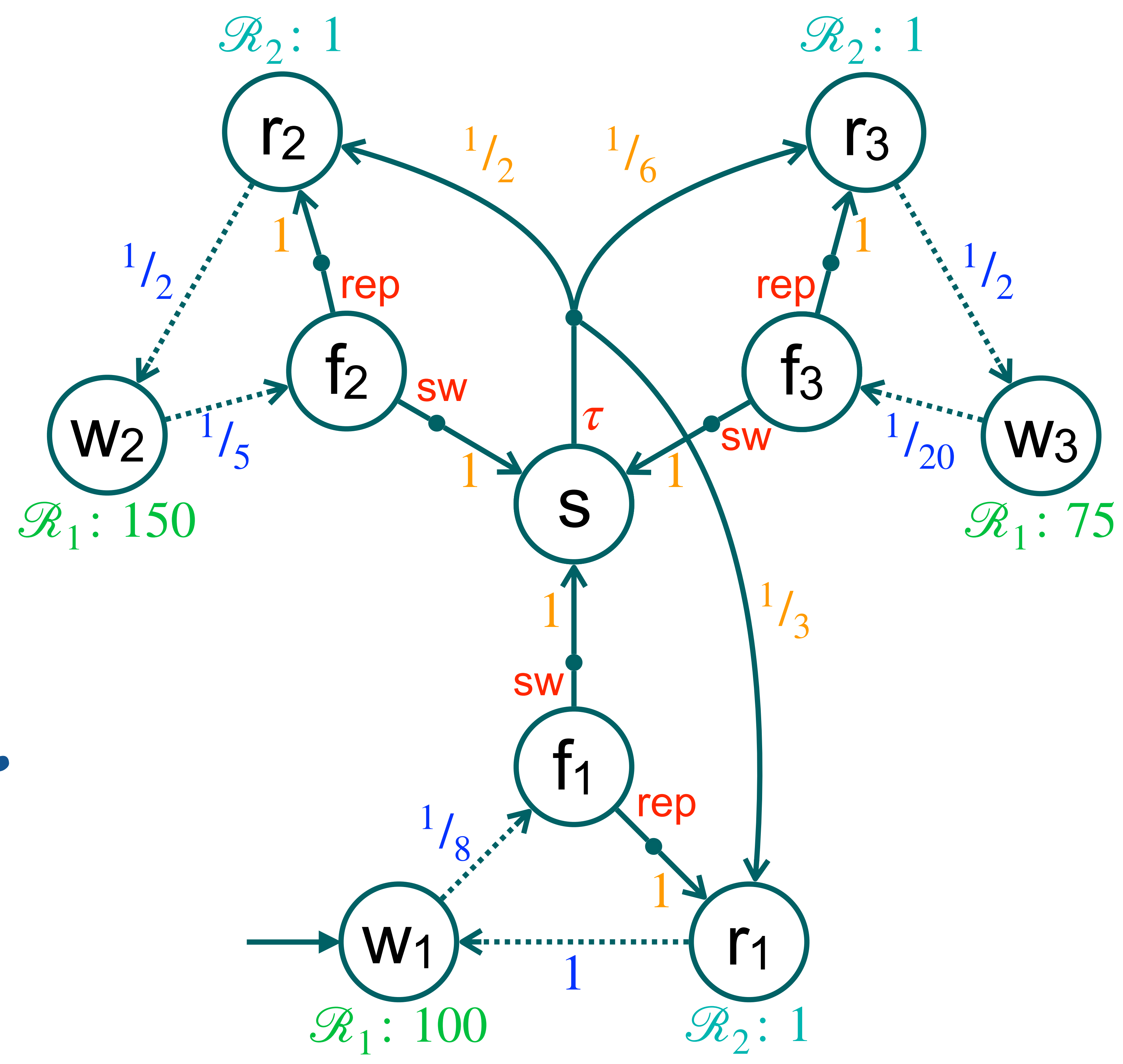
For $\Phi_{lra} = \langle lra(\mathscr{R}_1), \dots, lra(\mathscr{R}_\ell)\rangle$:

Ditto: $\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi_{lra}) = \mathrm{Ex}_{\sigma}(lra(\mathscr{R}_{\mathbf{w}}))$

$\leadsto$ Use single objective methods to get $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathrm{Ex}_{\sigma}(lra(\mathscr{R}_{\mathbf{w}}))\right)$

Intermezzo: Long-run Average Rewards via End Components

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

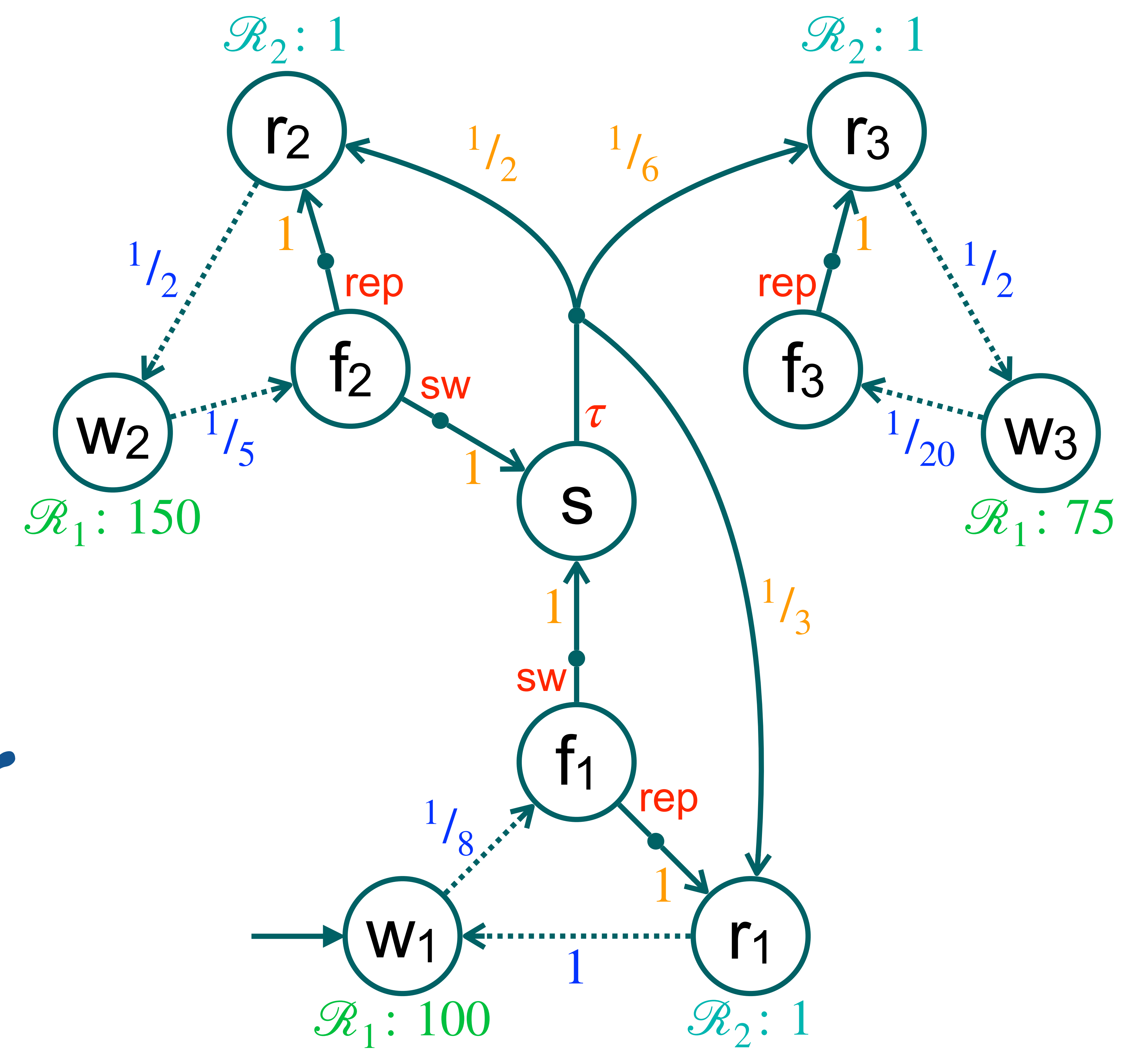


Four End Components:

- $\{w_i, f_i, r_i\}$ for $i = 1,2,3$
- $\{s\} \cup \bigcup_{i=1}^{3} \{w_i, f_i, r_i\}$

Intermezzo: Long-run Average Rewards via End Components

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

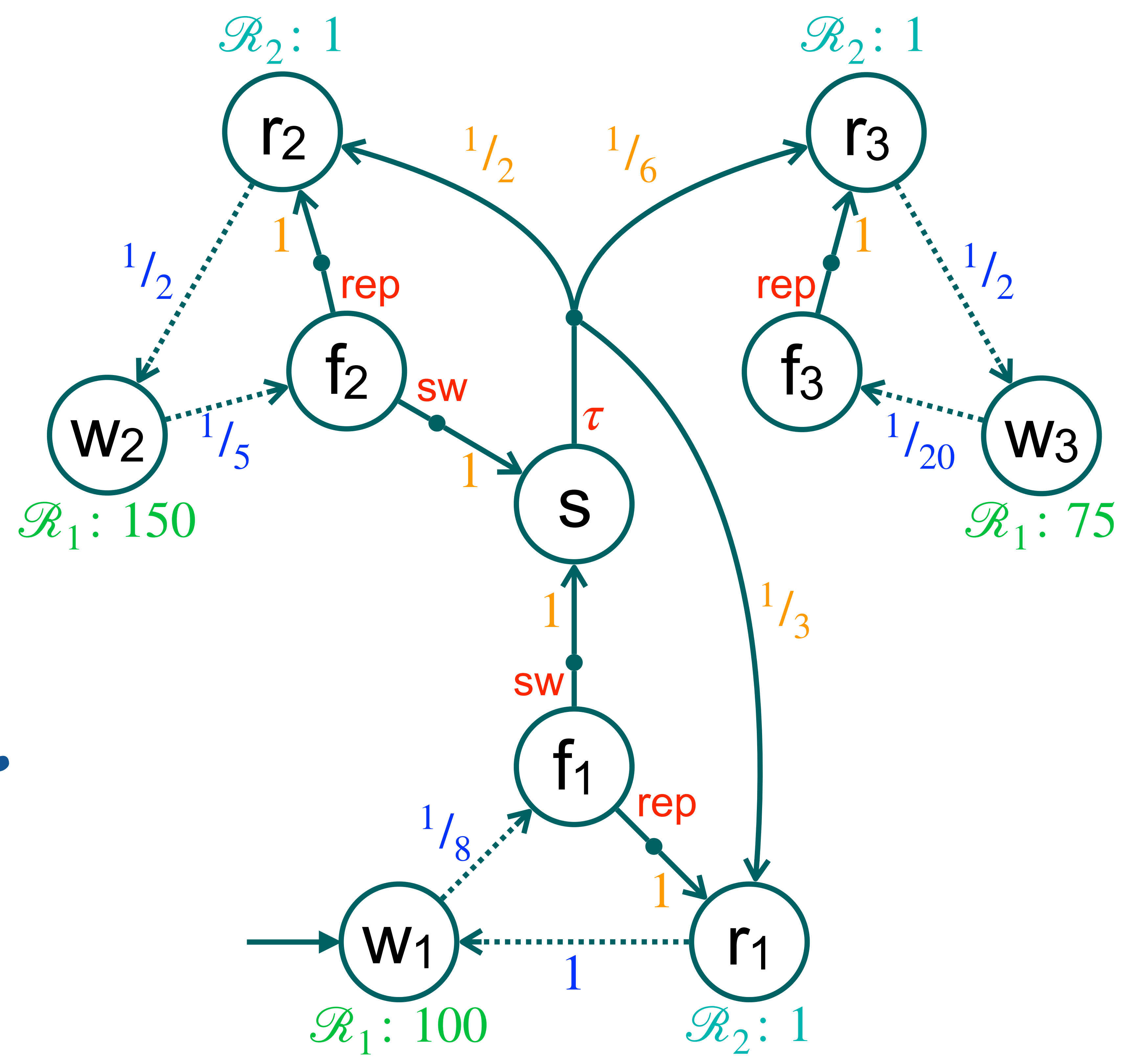


Three End Components:

- $\{w_i, f_i, r_i\}$ for $i = 1,2,3$
- ~~$\{s\} \cup \bigcup_{i=1}^{3} \{w_i, f_i, r_i\}$~~

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

- Only states within ECs can be visited infinitely often
- For $lra(\mathscr{R})$, only rewards within ECs are relevant



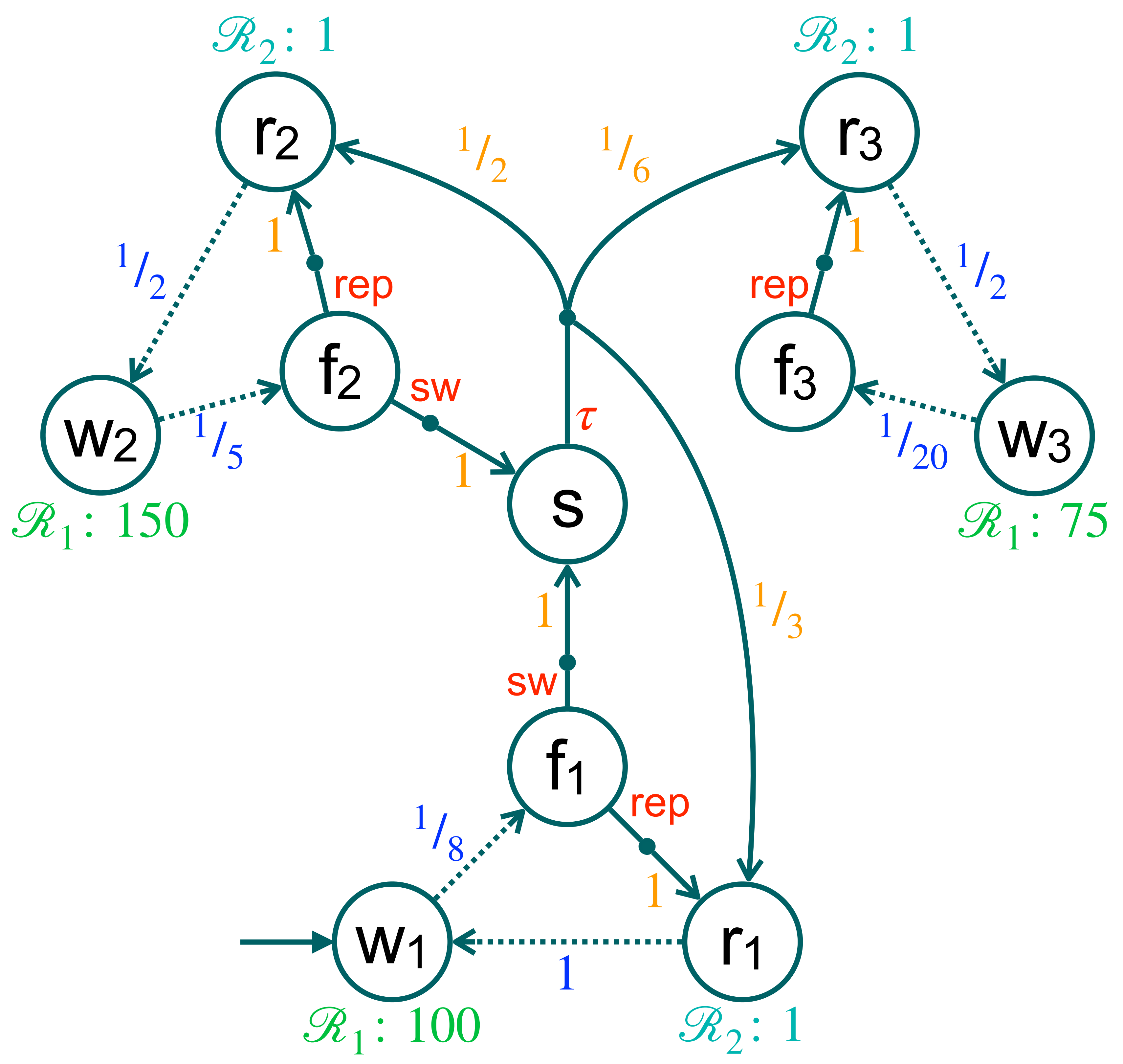
Three End Components:

- $\{w_i, f_i, r_i\}$ for $i = 1,2,3$
- ~~$\{s\} \cup \bigcup_{i=1}^{3} \{w_i, f_i, r_i\}$~~

Intermezzo: Long-run Average Rewards via End Components

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

- Only states within ECs can be visited infinitely often
- For $lra(\mathscr{R})$, only rewards within ECs are relevant

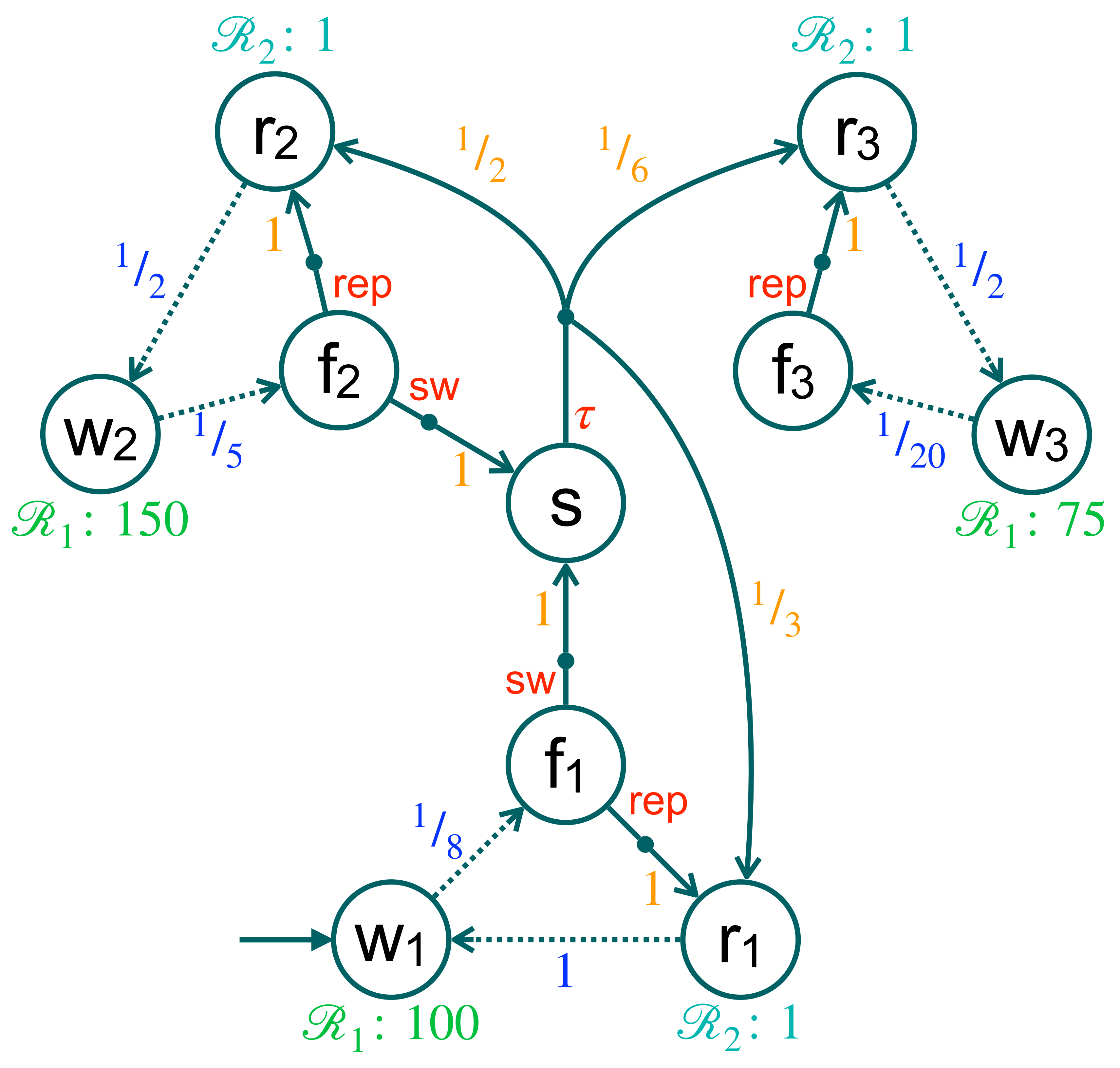


Computing single-objective long-run average rewards:

- Analyze $lra(\mathscr{R})$ within each (maximal) EC in isolation
- Fuse EC results together via a total reward analysis on a slightly modified model

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

- Only states within ECs can be visited infinitely often
- For $lra(\mathcal{R})$, only rewards within ECs are relevant



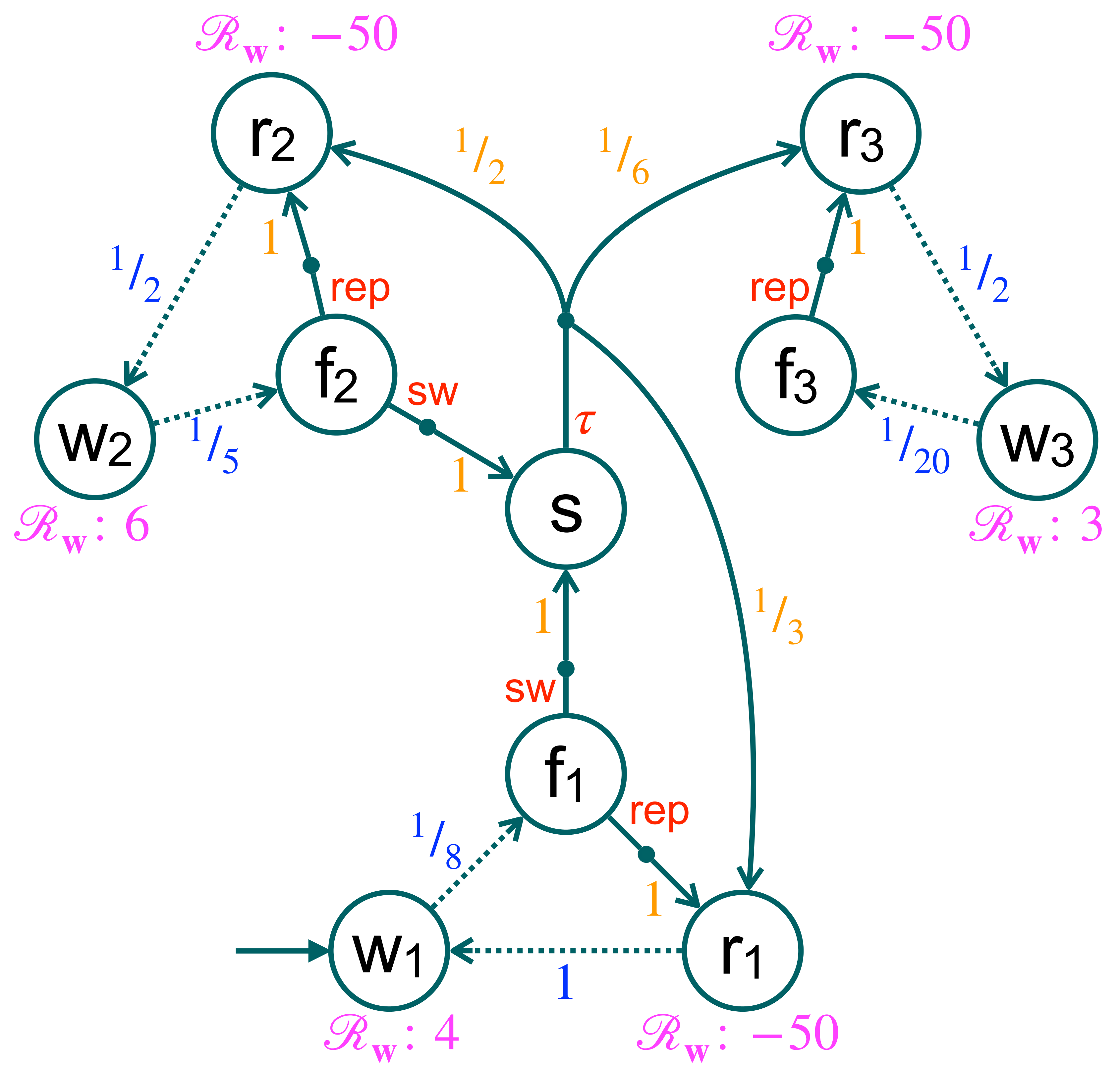
Computing single-objective long-run average rewards:

- Analyze $lra(\mathcal{R})$ within each (maximal) EC in isolation
- Fuse EC results together via a total reward analysis on a slightly modified model

$$\mathbf{w} = \langle {}^1/_{25}, 50\rangle \rightsquigarrow \mathcal{R}_{\mathbf{w}} = {}^1/_{25} \cdot \mathcal{R}_1 + 50 \cdot (-\mathcal{R}_2)$$

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

- Only states within ECs can be visited infinitely often
- For $lra(\mathscr{R})$, only rewards within ECs are relevant



Computing single-objective long-run average rewards:

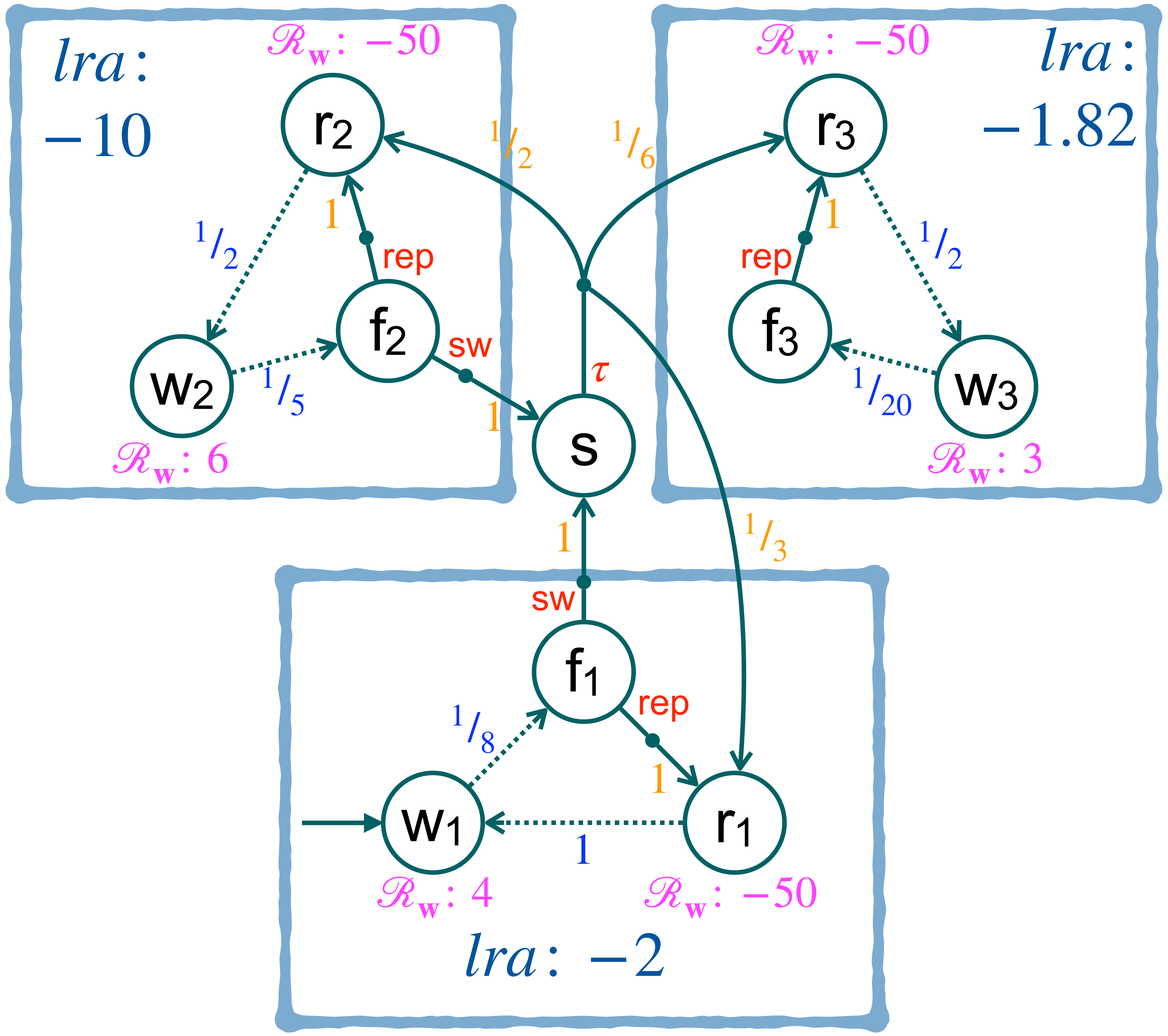
- Analyze $lra(\mathscr{R})$ within each (maximal) EC in isolation
- Fuse EC results together via a total reward analysis on a slightly modified model

$$\mathbf{w} = \langle {}^{1}/_{25}, 50 \rangle \rightsquigarrow \mathscr{R}_\mathbf{w} = {}^{1}/_{25} \cdot \mathscr{R}_1 + 50 \cdot (-\mathscr{R}_2)$$

Intermezzo: Long-run Average Rewards via End Components

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

- Only states within ECs can be visited infinitely often
- For $lra(\mathscr{R})$, only rewards within ECs are relevant



Computing single-objective long-run average rewards:

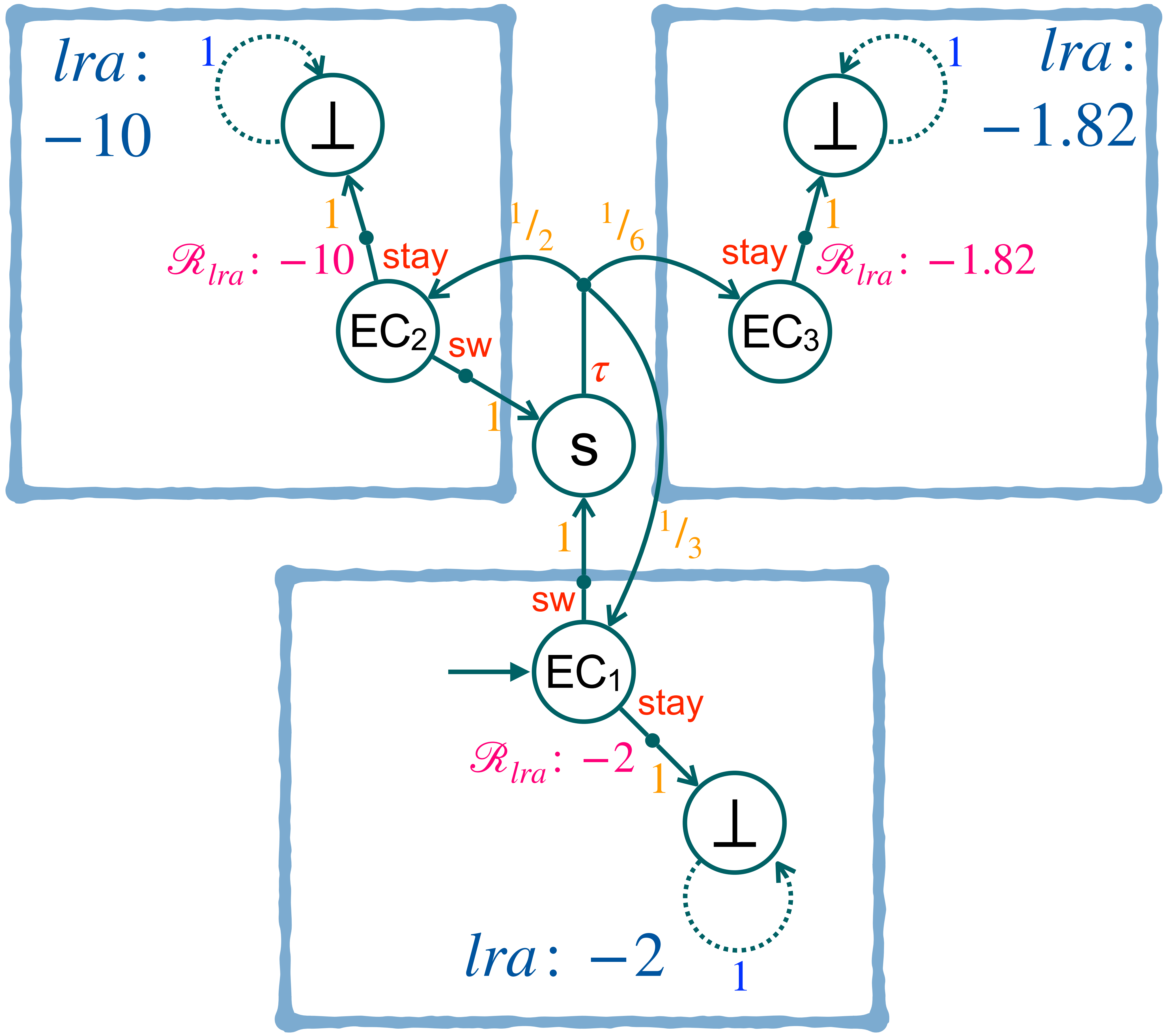
- Analyze $lra(\mathscr{R})$ within each (maximal) EC in isolation
- Fuse EC results together via a total reward analysis on a slightly modified model

$$\mathbf{w} = \langle {}^1/_{25}, 50\rangle \rightsquigarrow \mathscr{R}_{\mathbf{w}} = {}^1/_{25} \cdot \mathscr{R}_1 + 50 \cdot (-\mathscr{R}_2)$$

Intermezzo: Long-run Average Rewards via End Components

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

- Only states within ECs can be visited infinitely often
- For $lra(\mathscr{R})$, only rewards within ECs are relevant

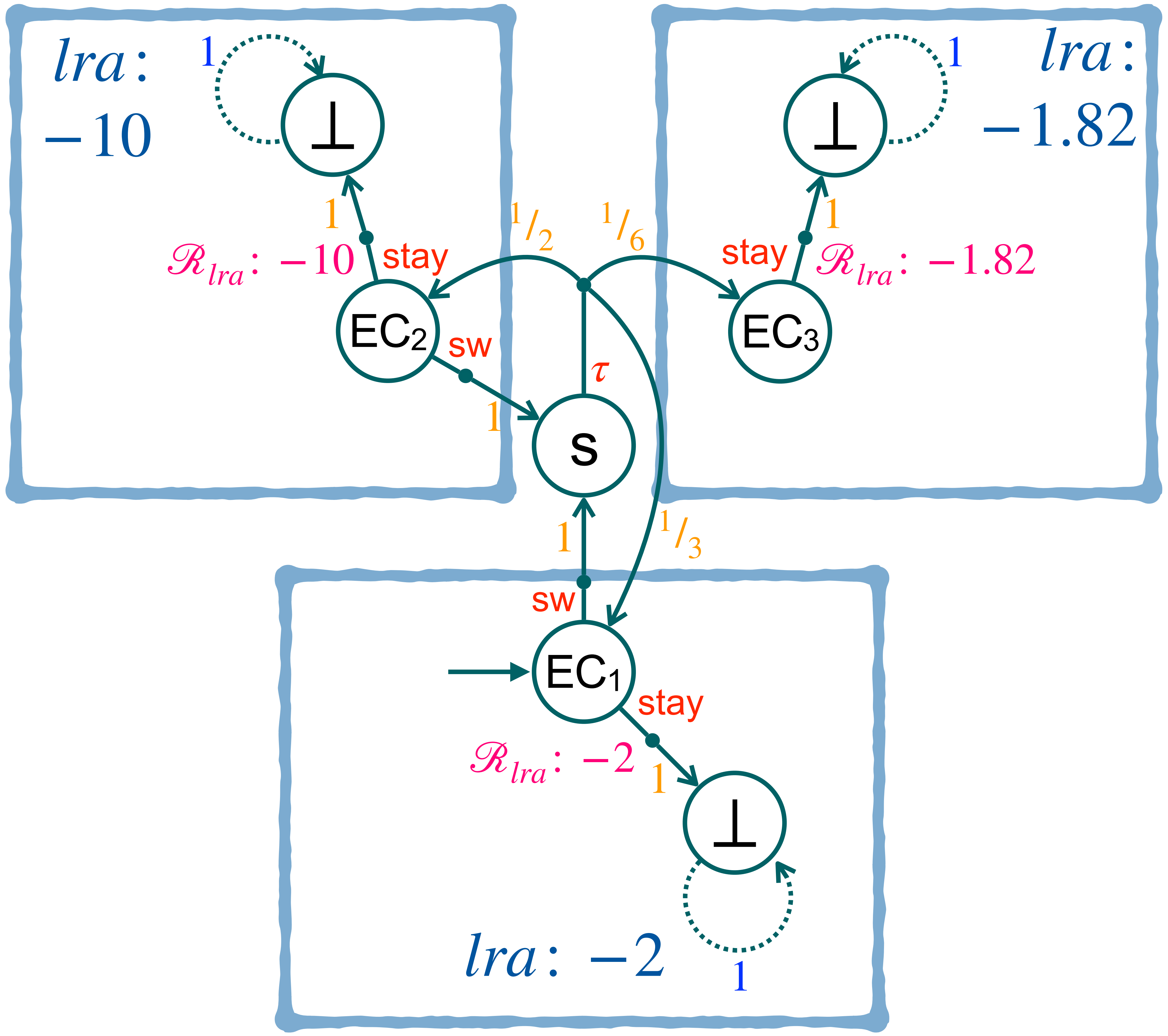


Computing single-objective long-run average rewards:

- Analyze $lra(\mathscr{R})$ within each (maximal) EC in isolation
- Fuse EC results together via a total reward analysis on a slightly modified model

End component (EC):
Strongly connected sub-model that–under some strategy–will never be left

- Only states within ECs can be visited infinitely often
- For $lra(\mathcal{R})$, only rewards within ECs are relevant



Computing single-objective long-run average rewards:

- Analyze $lra(\mathcal{R})$ within each (maximal) EC in isolation
- Fuse EC results together via a total reward analysis on a slightly modified model

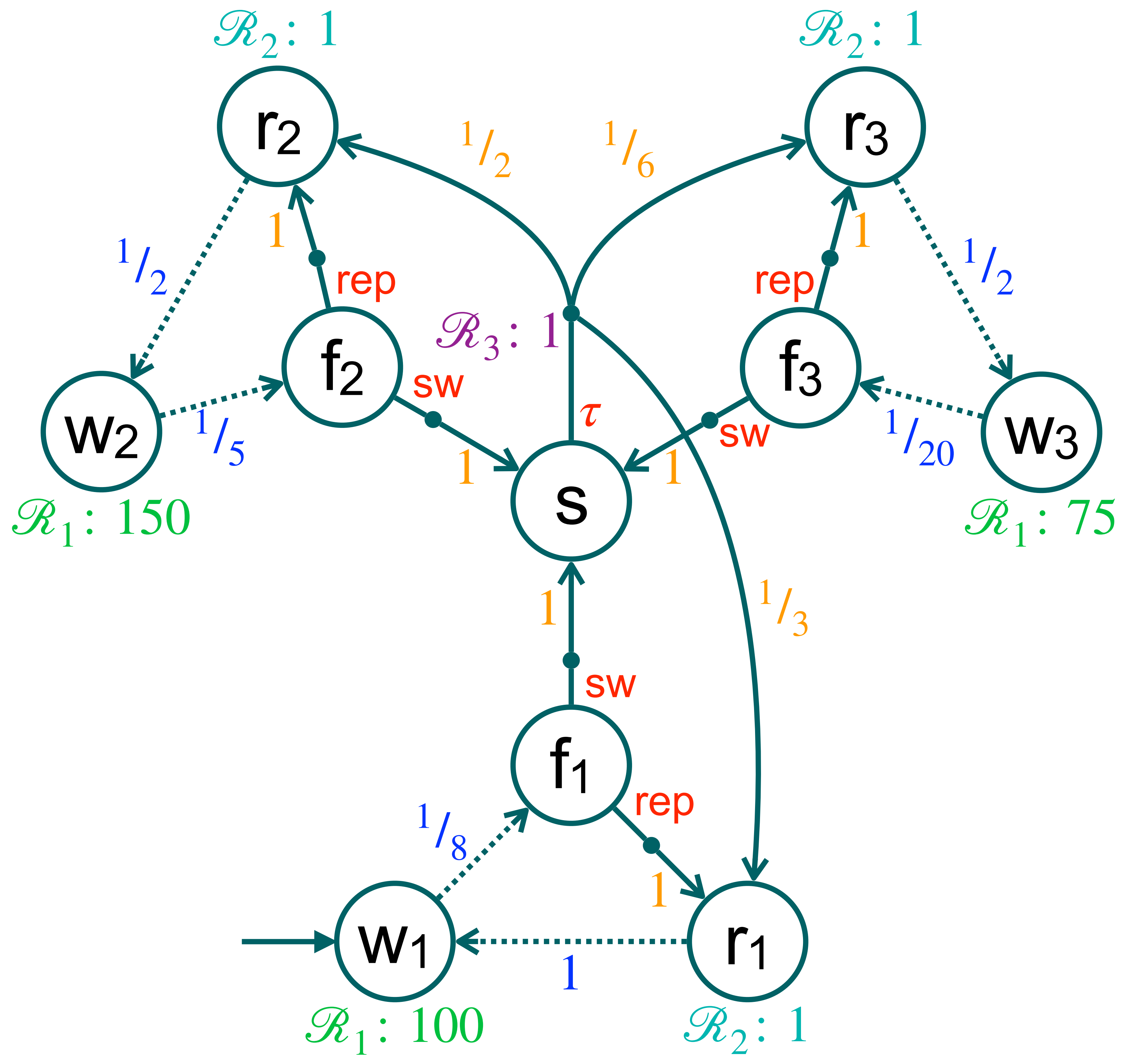
$$\arg\max_\sigma\left(\mathrm{Ex}_\sigma(tot(\mathcal{R}_{lra}))\right) = \left\{\mathsf{EC}_1, \mathsf{EC}_2 \mapsto \mathsf{sw},\ \mathsf{EC}_3 \mapsto \mathsf{stay}\right\}$$

Computing $\sigma_{\mathbf{w}} \in \arg\max_\sigma \left(\mathbf{w} \cdot \mathrm{Ex}_\sigma(\Phi)\right)$

For $\Phi_{lra+tot} = \langle lra(\mathcal{R}_1), \dots, lra(\mathcal{R}_k), tot(\mathcal{R}_{k+1}), \dots, tot(\mathcal{R}_\ell)\rangle$:

Idea:

- Analyze objective $lra\Big(\overbrace{\textstyle\sum_{i=1}^{k} \mathbf{w}[i] \cdot \mathcal{R}_i}^{=:\mathcal{R}_{\mathbf{w}}^{lra}}\Big)$ in maximal ECs

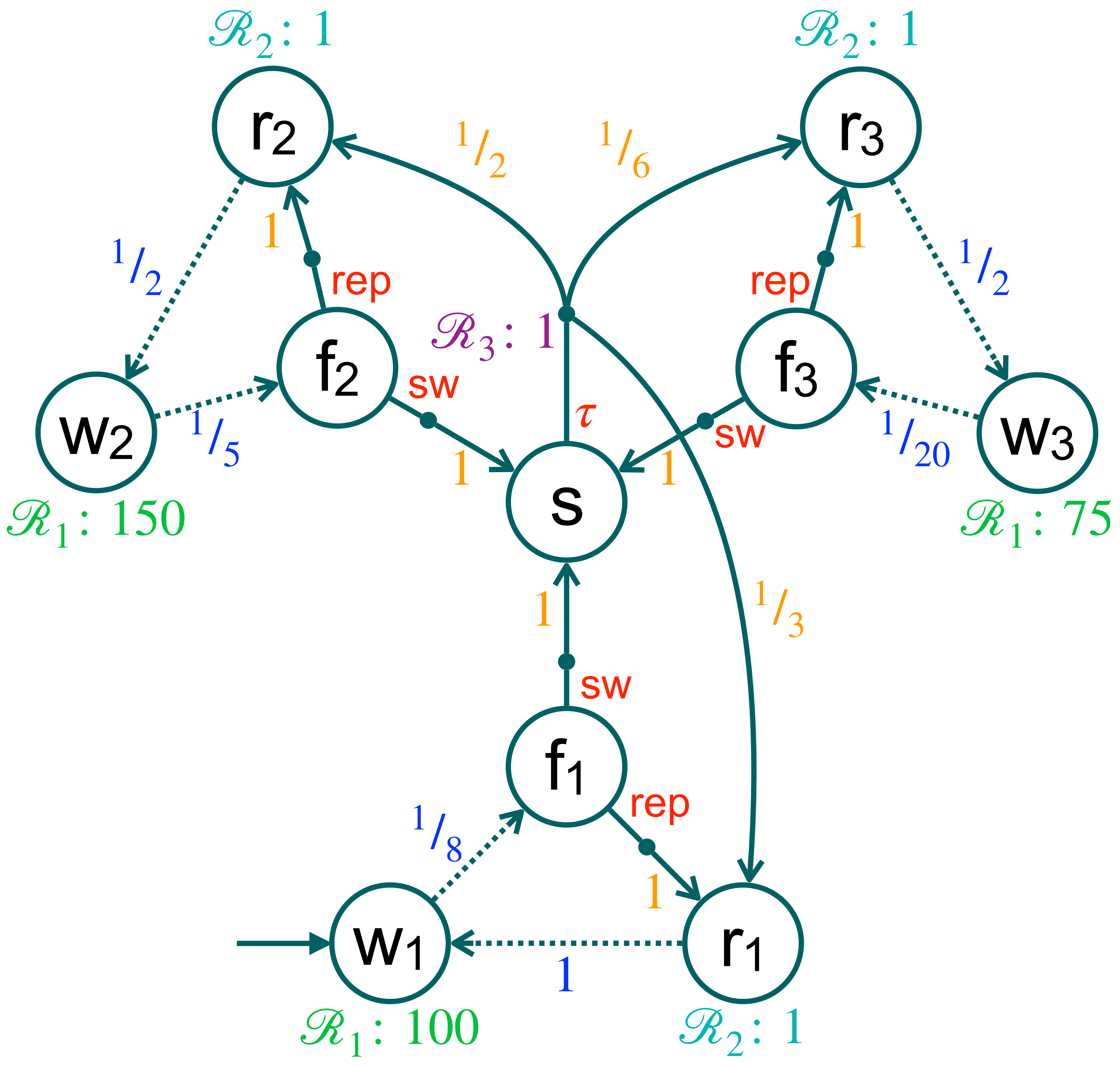


Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\big(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\big)$

For $\Phi_{lra+tot} = \langle lra(\mathcal{R}_1), \dots, lra(\mathcal{R}_k),$
$tot(\mathcal{R}_{k+1}), \dots, tot(\mathcal{R}_{\ell})\rangle$:

Idea:

- Analyze objective $lra\Big(\overbrace{\sum_{i=1}^{k} \mathbf{w}[i] \cdot \mathcal{R}_i}^{=:\mathcal{R}^{lra}_{\mathbf{w}}}\Big)$ in maximal ECs
  - Avoid $\mathrm{Ex}_{\sigma_{\mathbf{w}}}(tot(\mathcal{R}_j)) = \pm\infty$:
    Restrict to ECs without total rewards



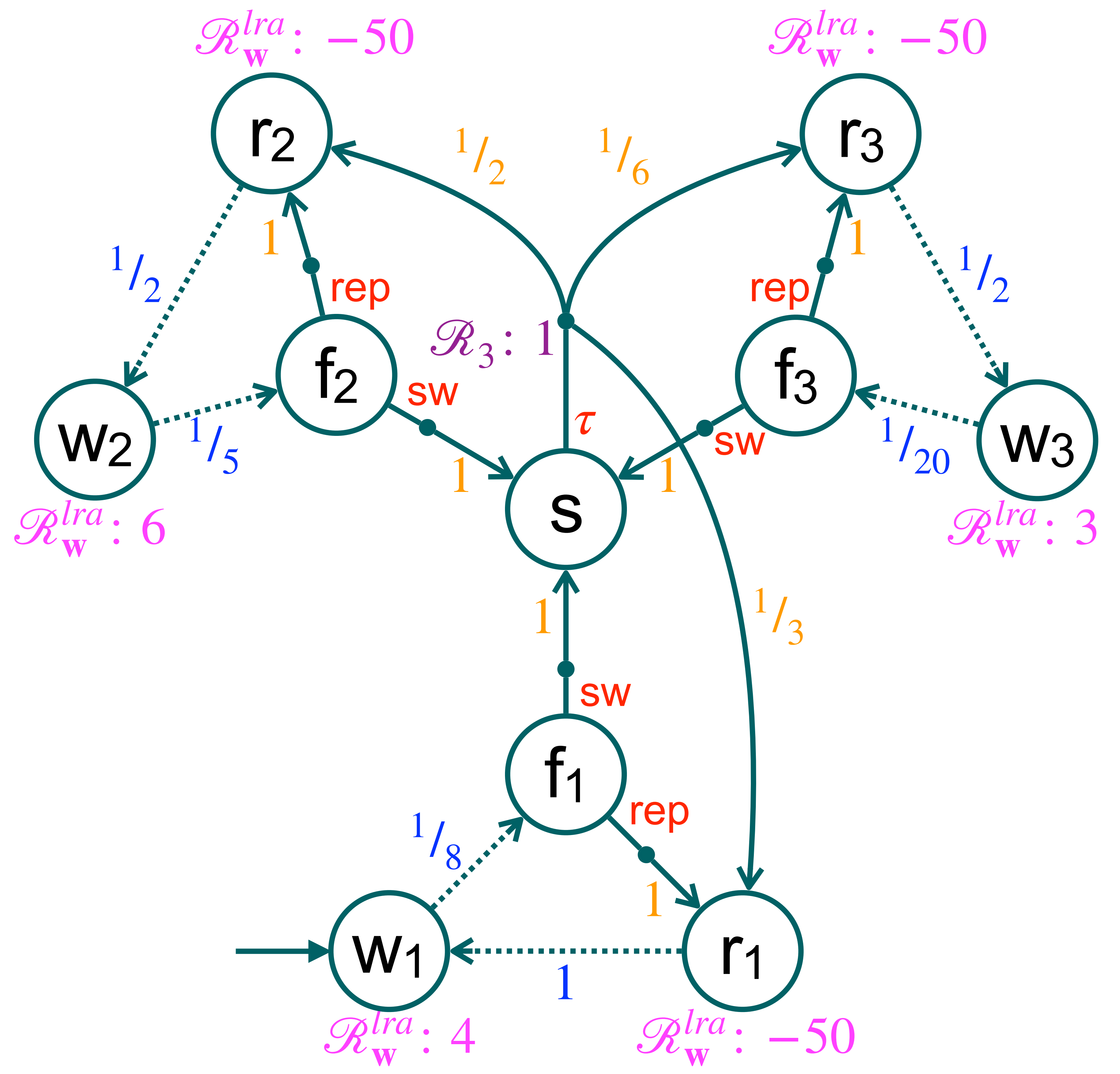
Tim Quatmann, Joost-Pieter Katoen

Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$

For $\Phi_{lra+tot} = \langle lra(\mathscr{R}_1), \ldots, lra(\mathscr{R}_k),$
$tot(\mathscr{R}_{k+1}), \ldots, tot(\mathscr{R}_\ell)\rangle$:

Idea:

- Analyze objective $lra\Big(\overbrace{\textstyle\sum_{i=1}^{k} \mathbf{w}[i] \cdot \mathscr{R}_i}^{=:\mathscr{R}^{lra}_{\mathbf{w}}}\Big)$ in maximal ECs
  - Avoid $\mathrm{Ex}_{\sigma_{\mathbf{w}}}(tot(\mathscr{R}_j)) = \pm\infty$:
    Restrict to ECs without total rewards



$$\Phi = \langle lra(\mathscr{R}_1), lra(-\mathscr{R}_2), tot(-\mathscr{R}_3)\rangle$$

$$\mathbf{w} = \langle {}^1/_{25}, 50, \mathbf{1}\rangle \;\rightsquigarrow\; \mathscr{R}^{lra}_{\mathbf{w}} = {}^1/_{25} \cdot \mathscr{R}_1 + 50 \cdot (-\mathscr{R}_2)$$

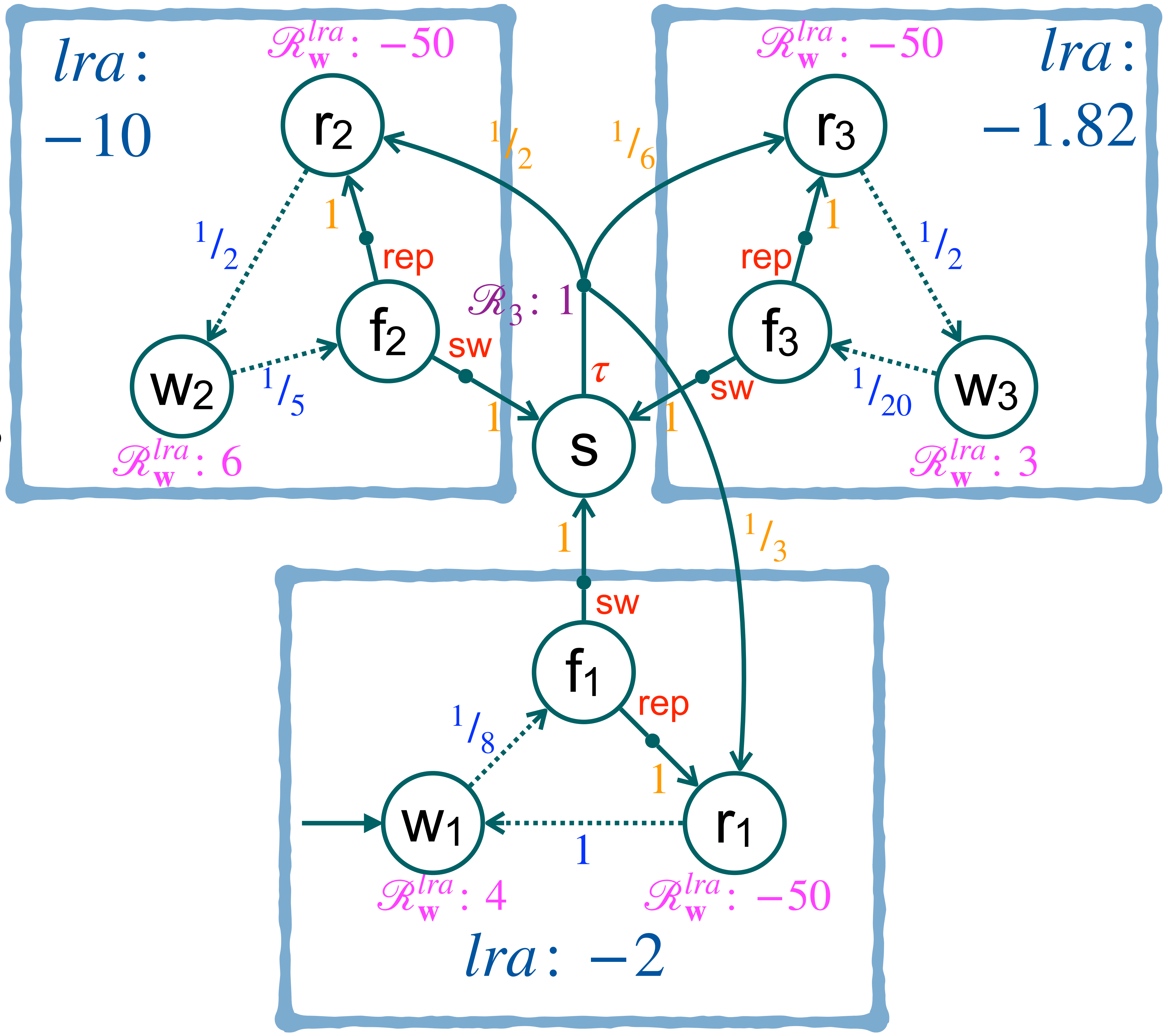
Tim Quatmann, Joost-Pieter Katoen
Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\big(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\big)$

For $\Phi_{lra+tot} = \langle lra(\mathcal{R}_1), \dots, lra(\mathcal{R}_k), tot(\mathcal{R}_{k+1}), \dots, tot(\mathcal{R}_\ell)\rangle$:

Idea:

- Analyze objective $lra\big(\overbrace{\sum_{i=1}^{k} \mathbf{w}[i] \cdot \mathcal{R}_i}^{=:\mathcal{R}^{lra}_{\mathbf{w}}}\big)$ in maximal ECs
  - Avoid $\mathrm{Ex}_{\sigma_{\mathbf{w}}}(tot(\mathcal{R}_j)) = \pm\infty$: Restrict to ECs without total rewards



$$\Phi = \langle lra(\mathcal{R}_1), lra(-\mathcal{R}_2), tot(-\mathcal{R}_3)\rangle$$

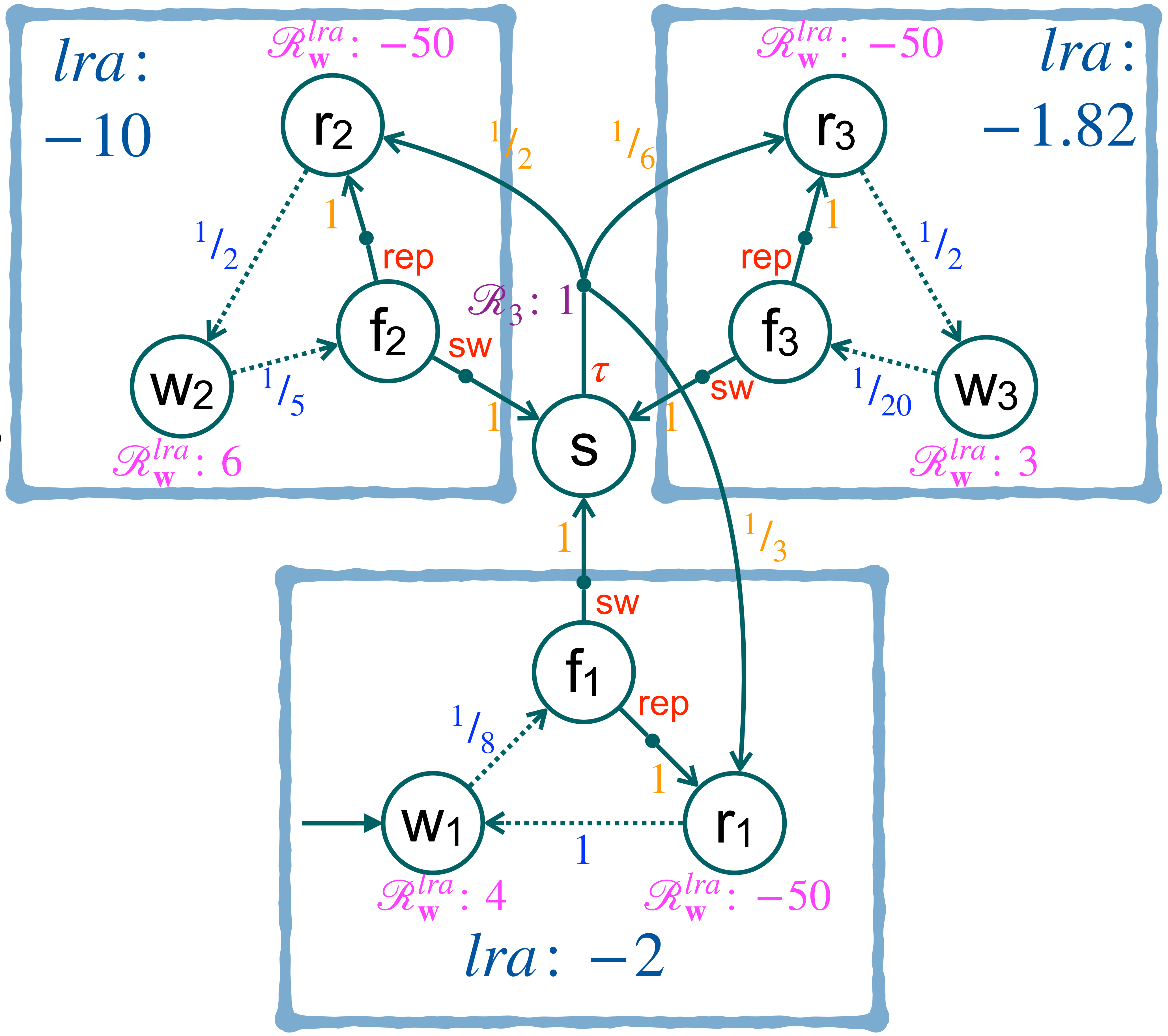
$$\mathbf{w} = \langle {}^{1}/_{25}, 50, \mathbf{1}\rangle \;\rightsquigarrow\; \mathcal{R}^{lra}_{\mathbf{w}} = {}^{1}/_{25} \cdot \mathcal{R}_1 + 50 \cdot (-\mathcal{R}_2)$$

Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$

For $\Phi_{lra+tot} = \langle lra(\mathcal{R}_1), \ldots, lra(\mathcal{R}_k), tot(\mathcal{R}_{k+1}), \ldots, tot(\mathcal{R}_\ell)\rangle$:

Idea:

- Analyze objective $lra\left(\overbrace{\sum_{i=1}^{k} \mathbf{w}[i] \cdot \mathcal{R}_i}^{=:\mathcal{R}_{\mathbf{w}}^{lra}}\right)$ in maximal ECs
  - Avoid $\mathrm{Ex}_{\sigma_{\mathbf{w}}}(tot(\mathcal{R}_j)) = \pm\infty$: Restrict to ECs without total rewards
- When fusing EC results together, also incorporate total rewards, i.e. consider $tot\left(\mathcal{R}_{\mathbf{w}}^{lra} + \sum_{i=k+1}^{\ell} \mathbf{w}[i] \cdot \mathcal{R}_i\right)$



$\Phi = \langle lra(\mathcal{R}_1), lra(-\mathcal{R}_2), tot(-\mathcal{R}_3)\rangle$

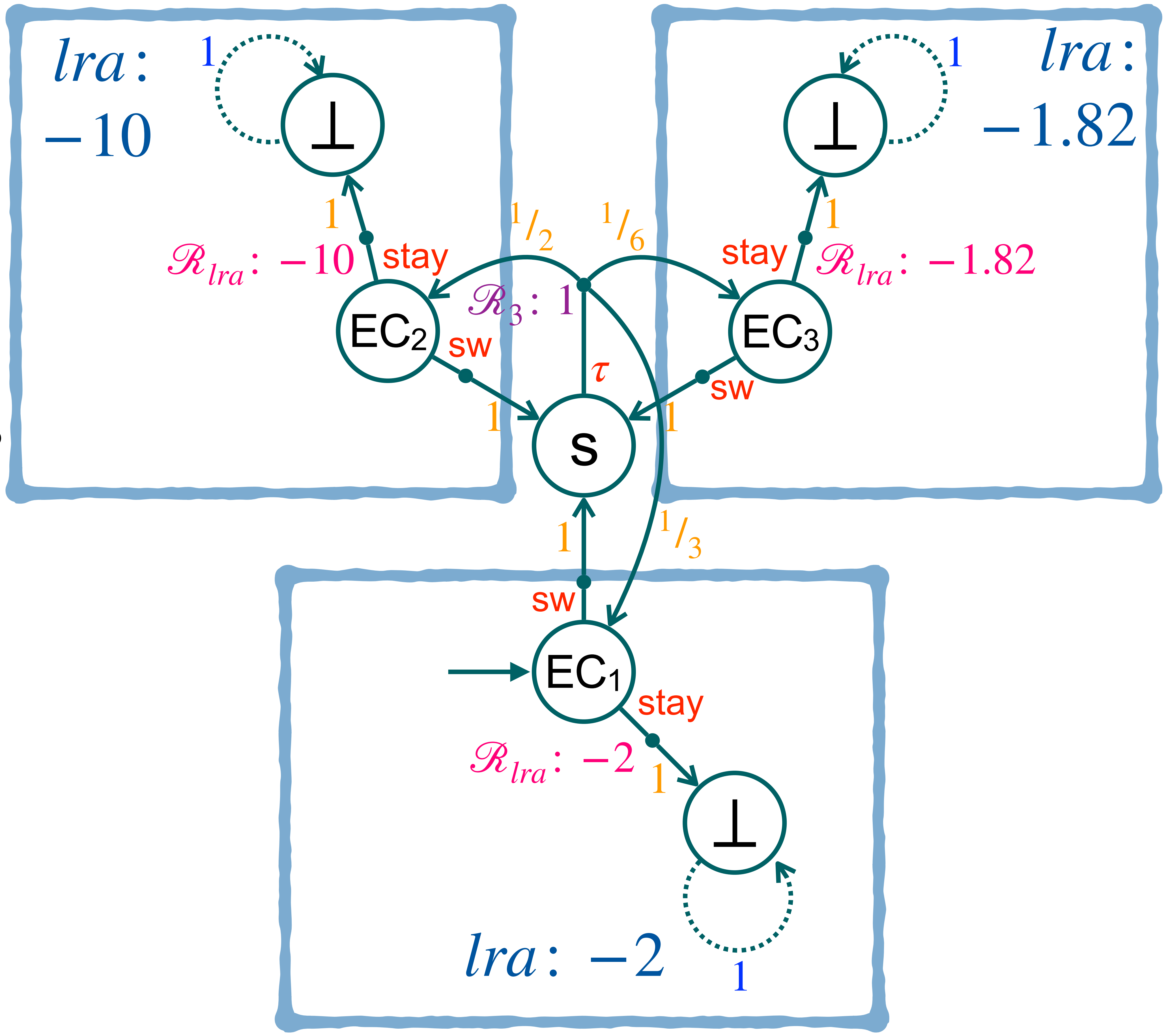
$\mathbf{w} = \langle {}^1/_{25}, 50, \mathbf{1}\rangle \rightsquigarrow \mathcal{R}_{\mathbf{w}}^{lra} = {}^1/_{25} \cdot \mathcal{R}_1 + 50 \cdot (-\mathcal{R}_2)$

Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\left(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\right)$

For $\Phi_{lra+tot} = \langle lra(\mathcal{R}_1), \ldots, lra(\mathcal{R}_k), tot(\mathcal{R}_{k+1}), \ldots, tot(\mathcal{R}_\ell)\rangle$:

Idea:

- Analyze objective $lra\Big(\overbrace{\sum_{i=1}^{k} \mathbf{w}[i] \cdot \mathcal{R}_i}^{=:\mathcal{R}^{lra}_{\mathbf{w}}}\Big)$ in maximal ECs
  - Avoid $\mathrm{Ex}_{\sigma_{\mathbf{w}}}(tot(\mathcal{R}_j)) = \pm\infty$: Restrict to ECs without total rewards
- When fusing EC results together, also incorporate total rewards, i.e. consider $tot\Big(\mathcal{R}_{lra} + \sum_{i=k+1}^{\ell} \mathbf{w}[i] \cdot \mathcal{R}_i\Big)$

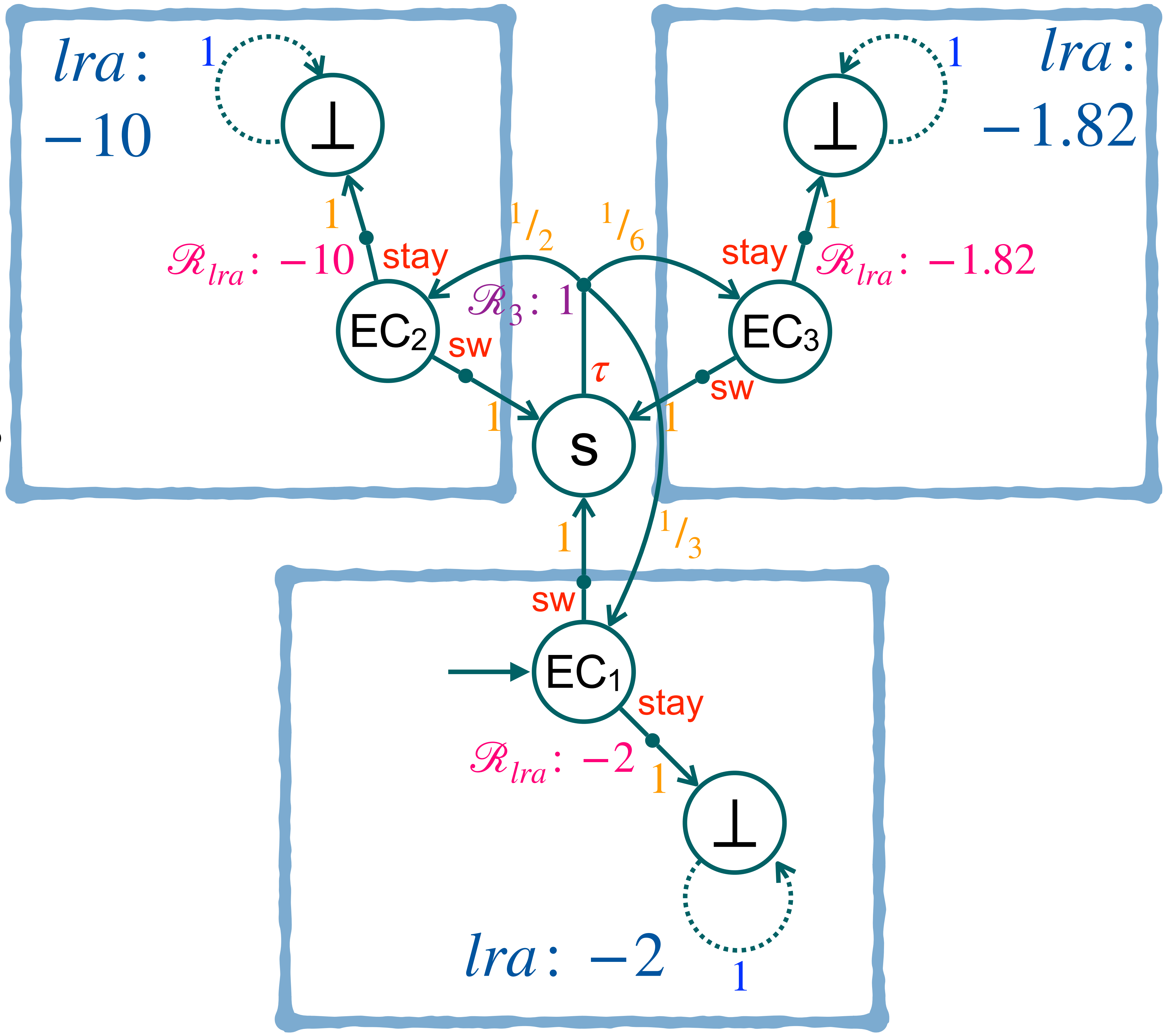


Computing $\sigma_{\mathbf{w}} \in \arg\max_{\sigma}\big(\mathbf{w} \cdot \mathrm{Ex}_{\sigma}(\Phi)\big)$

For $\Phi_{lra+tot} = \langle lra(\mathcal{R}_1), \dots, lra(\mathcal{R}_k), tot(\mathcal{R}_{k+1}), \dots, tot(\mathcal{R}_\ell)\rangle$:

Idea:

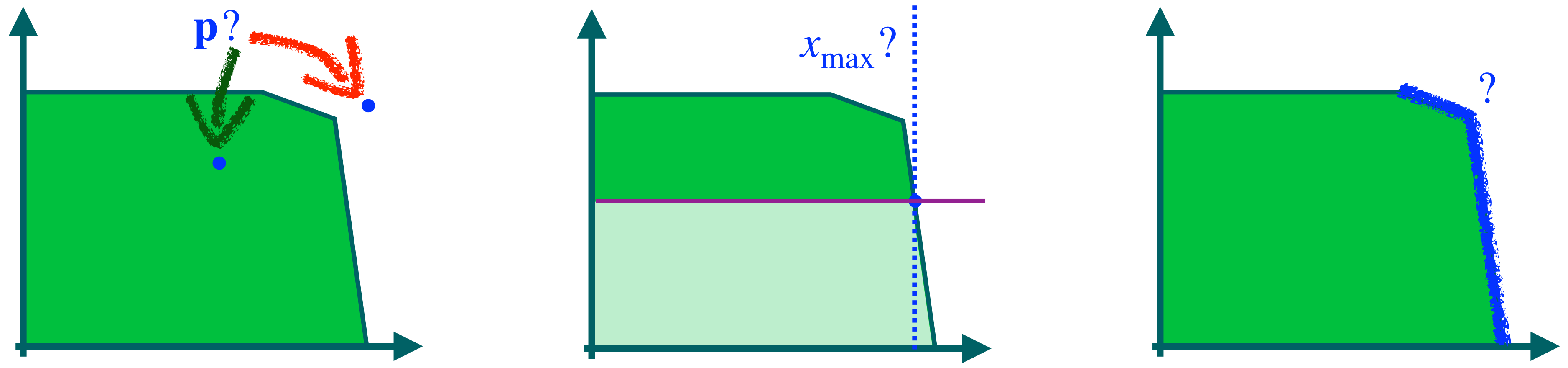
- Analyze objective $lra\Big(\overbrace{\sum_{i=1}^{k} \mathbf{w}[i] \cdot \mathcal{R}_i}^{=:\mathcal{R}^{lra}_{\mathbf{w}}}\Big)$ in maximal ECs
  - Avoid $\mathrm{Ex}_{\sigma_{\mathbf{w}}}(tot(\mathcal{R}_j)) = \pm\infty$: Restrict to ECs without total rewards
- When fusing EC results together, also incorporate total rewards, i.e. consider $tot\Big(\mathcal{R}_{lra} + \sum_{i=k+1}^{\ell} \mathbf{w}[i] \cdot \mathcal{R}_i\Big)$



$$\arg\max_{\sigma}\Big(\mathrm{Ex}_{\sigma}(tot(\mathcal{R}_{lra} + 1 \cdot (-\mathcal{R}_3)))\Big) = \Big\{\mathrm{EC}_1 \mapsto \text{stay}, \;\cdots\Big\}$$

Implementation

- Supports MDP and MA models specified in PRISM or JANI
- Qualitative / Quantitative / Pareto Queries



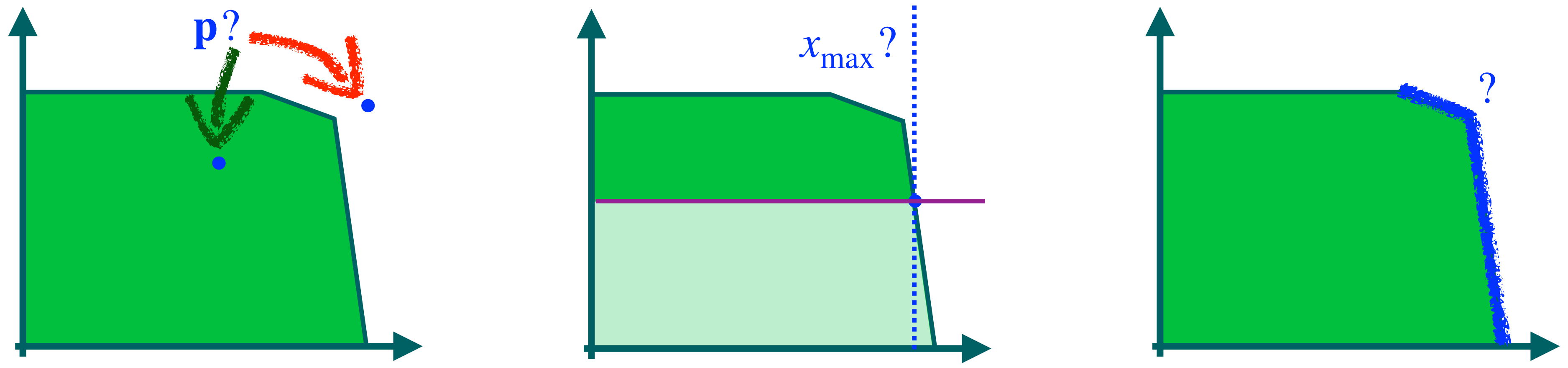
- $lra(\cdot)$ via **value iteration** [Butkova, Wimmer, & Hermanns'17; Ashok et al.'17]
- $tot(\cdot)$ via **sound value iteration** [Quatmann & Katoen'18]
- Also supports time- and step-bounded objectives

Storm

www.stormchecker.org

Implementation

- Supports MDP and MA models specified in PRISM or JANI
- Qualitative / Quantitative / Pareto Queries



- $lra(\cdot)$ via **value iteration** [Butkova, Wimmer, & Hermanns'17; Ashok et al.'17]
- $tot(\cdot)$ via **sound value iteration** [Quatmann & Katoen'18]
- Also supports time- and step-bounded objectives

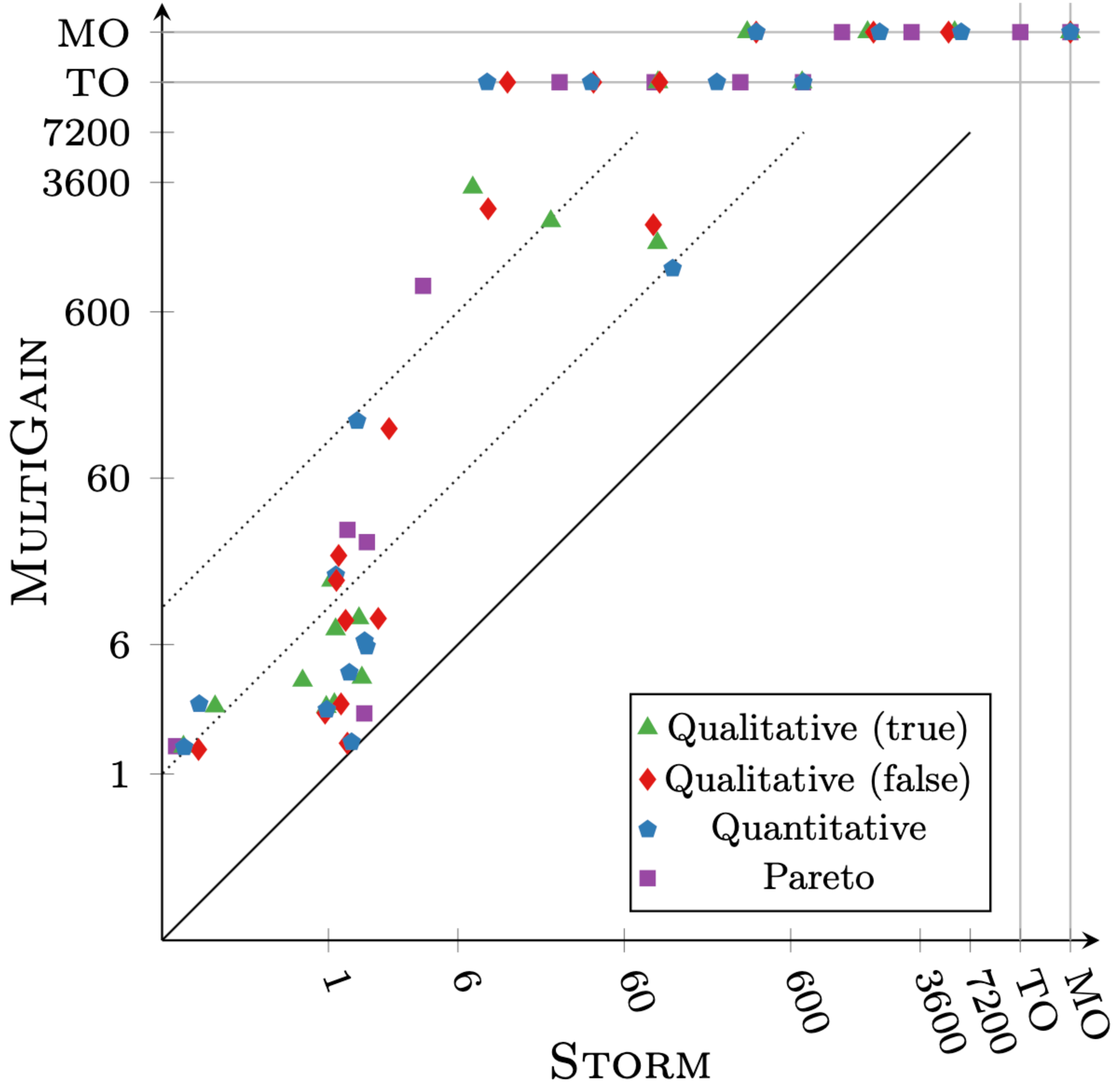
Storm

www.stormchecker.org

Experiments

- Comparison with **MultiGain** [Brázdil et al.'15]
  - Supports "only" long-run average reward objectives for MDP
  - Employs linear programming; using LP solver **Gurobi**
- 10 case studies × 3 instances ⇝ 12 MA and 18 MDP models
- Resource limits: 2 hours / 32 GB RAM

Storm vs. MultiGain



- Storm is often several orders of magnitude faster
- MultiGain is often stuck in expensive LP solving

Further results (excerpt)

Model	Par.	#lra-#tot	$\lvert S \rvert$	$\lvert MS \rvert$	$\lvert \Delta \rvert$	#EC	$\lvert S_{EC} \rvert$	#iter	STORM runtime
csn	3	3-0	177		427	38	158	9	1.23
csn	4	4-0	945		2753	176	880	30	109
csn	5	5-0	4833		$2{\cdot}10^4$	782	4622		TO
mut	3	2-0	$3{\cdot}10^4$		$5{\cdot}10^4$	1	$3{\cdot}10^4$	15	3.7
mut	4	2-0	$7{\cdot}10^5$		$1{\cdot}10^6$	1	$7{\cdot}10^5$	14	91.4
mut	5	2-0	$1{\cdot}10^7$		$3{\cdot}10^7$	1	$1{\cdot}10^7$	12	3197
clu	8-3	2-0	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	11	287
clu	16-4	2-0	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	10	4199
clu	32-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$		TO
clu	8-3	1-1	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	7	163
clu	16-4	1-1	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	9	3432
clu	32-3	1-1	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$	7	3328
rqs	2-2	2-0	1619	628	2296	1	1618	63	4.52
rqs	3-3	2-0	$9{\cdot}10^4$	$4{\cdot}10^4$	$1{\cdot}10^5$	1	$9{\cdot}10^4$	106	162
rqs	5-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$4{\cdot}10^6$	1	$2{\cdot}10^6$	97	4345
rqs	2-2	1-1	2805	1039	4159	1	1618	3	< 1
rqs	3-3	1-1	$1{\cdot}10^5$	$6{\cdot}10^4$	$3{\cdot}10^5$	1	$9{\cdot}10^4$	3	4.51
rqs	5-3	1-1	$3{\cdot}10^6$	$2{\cdot}10^6$	$7{\cdot}10^6$	1	$2{\cdot}10^6$	3	182

Further results (excerpt)

Model	Par.	#lra-#tot	$\lvert S\rvert$	$\lvert MS\rvert$	$\lvert\Delta\rvert$	#EC	$\lvert S_{EC}\rvert$	#iter	STORM runtime
csn	3	3-0	177		427	38	158	9	1.23
csn	4	4-0	945		2753	176	880	30	109
csn	5	5-0	4833		$2 \cdot 10^4$	782	4622		TO
mut	3	2-0	$3 \cdot 10^4$		$5 \cdot 10^4$	1	$3 \cdot 10^4$	15	3.7
mut	4	2-0	$7 \cdot 10^5$		$1 \cdot 10^6$	1	$7 \cdot 10^5$	14	91.4
mut	5	2-0	$1 \cdot 10^7$		$3 \cdot 10^7$	1	$1 \cdot 10^7$	12	3197
clu	8-3	2-0	$2 \cdot 10^5$	$1 \cdot 10^5$	$4 \cdot 10^5$	4	$2 \cdot 10^5$	11	287
clu	16-4	2-0	$2 \cdot 10^6$	$9 \cdot 10^5$	$4 \cdot 10^6$	5	$2 \cdot 10^6$	10	4199
clu	32-3	2-0	$2 \cdot 10^6$	$1 \cdot 10^6$	$5 \cdot 10^6$	4	$2 \cdot 10^6$		TO
clu	8-3	1-1	$2 \cdot 10^5$	$1 \cdot 10^5$	$4 \cdot 10^5$	4	$2 \cdot 10^5$	7	163
clu	16-4	1-1	$2 \cdot 10^6$	$9 \cdot 10^5$	$4 \cdot 10^6$	5	$2 \cdot 10^6$	9	3432
clu	32-3	1-1	$2 \cdot 10^6$	$1 \cdot 10^6$	$5 \cdot 10^6$	4	$2 \cdot 10^6$	7	3328
rqs	2-2	2-0	1619	628	2296	1	1618	63	4.52
rqs	3-3	2-0	$9 \cdot 10^4$	$4 \cdot 10^4$	$1 \cdot 10^5$	1	$9 \cdot 10^4$	106	162
rqs	5-3	2-0	$2 \cdot 10^6$	$1 \cdot 10^6$	$4 \cdot 10^6$	1	$2 \cdot 10^6$	97	4345
rqs	2-2	1-1	2805	1039	4159	1	1618	3	< 1
rqs	3-3	1-1	$1 \cdot 10^5$	$6 \cdot 10^4$	$3 \cdot 10^5$	1	$9 \cdot 10^4$	3	4.51
rqs	5-3	1-1	$3 \cdot 10^6$	$2 \cdot 10^6$	$7 \cdot 10^6$	1	$2 \cdot 10^6$	3	182

- Storm can handle
  - millions of states

Further results (excerpt)

Model	Par.	#lra-#tot	$\|S\|$	$\|MS\|$	$\|\Delta\|$	#EC	$\|S_{EC}\|$	#iter	Storm runtime
csn	3	3-0	177		427	38	158	9	1.23
csn	4	4-0	945		2753	176	880	30	109
csn	5	5-0	4833		$2{\cdot}10^4$	782	4622		TO
mut	3	2-0	$3{\cdot}10^4$		$5{\cdot}10^4$	1	$3{\cdot}10^4$	15	3.7
mut	4	2-0	$7{\cdot}10^5$		$1{\cdot}10^6$	1	$7{\cdot}10^5$	14	91.4
mut	5	2-0	$1{\cdot}10^7$		$3{\cdot}10^7$	1	$1{\cdot}10^7$	12	3197
clu	8-3	2-0	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	11	287
clu	16-4	2-0	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	10	4199
clu	32-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$		TO
clu	8-3	1-1	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	7	163
clu	16-4	1-1	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	9	3432
clu	32-3	1-1	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$	7	3328
rqs	2-2	2-0	1619	628	2296	1	1618	63	4.52
rqs	3-3	2-0	$9{\cdot}10^4$	$4{\cdot}10^4$	$1{\cdot}10^5$	1	$9{\cdot}10^4$	106	162
rqs	5-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$4{\cdot}10^6$	1	$2{\cdot}10^6$	97	4345
rqs	2-2	1-1	2805	1039	4159	1	1618	3	<1
rqs	3-3	1-1	$1{\cdot}10^5$	$6{\cdot}10^4$	$3{\cdot}10^5$	1	$9{\cdot}10^4$	3	4.51
rqs	5-3	1-1	$3{\cdot}10^6$	$2{\cdot}10^6$	$7{\cdot}10^6$	1	$2{\cdot}10^6$	3	182

- Storm can handle
  - millions of states
  - four objectives

Further results (excerpt)

Model	Par.	#lra-#tot	$\|S\|$	$\|MS\|$	$\|\Delta\|$	#EC	$\|S_{EC}\|$	#iter	Storm runtime
csn	3	3-0	177		427	38	158	9	1.23
csn	4	4-0	945		2753	176	880	30	109
csn	5	5-0	4833		$2{\cdot}10^4$	782	4622		TO
mut	3	2-0	$3{\cdot}10^4$		$5{\cdot}10^4$	1	$3{\cdot}10^4$	15	3.7
mut	4	2-0	$7{\cdot}10^5$		$1{\cdot}10^6$	1	$7{\cdot}10^5$	14	91.4
mut	5	2-0	$1{\cdot}10^7$		$3{\cdot}10^7$	1	$1{\cdot}10^7$	12	3197
clu	8-3	2-0	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	11	287
clu	16-4	2-0	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	10	4199
clu	32-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$		TO
clu	8-3	1-1	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	7	163
clu	16-4	1-1	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	9	3432
clu	32-3	1-1	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$	7	3328
rqs	2-2	2-0	1619	628	2296	1	1618	63	4.52
rqs	3-3	2-0	$9{\cdot}10^4$	$4{\cdot}10^4$	$1{\cdot}10^5$	1	$9{\cdot}10^4$	106	162
rqs	5-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$4{\cdot}10^6$	1	$2{\cdot}10^6$	97	4345
rqs	2-2	1-1	2805	1039	4159	1	1618	3	<1
rqs	3-3	1-1	$1{\cdot}10^5$	$6{\cdot}10^4$	$3{\cdot}10^5$	1	$9{\cdot}10^4$	3	4.51
rqs	5-3	1-1	$3{\cdot}10^6$	$2{\cdot}10^6$	$7{\cdot}10^6$	1	$2{\cdot}10^6$	3	182

- Storm can handle
  - millions of states
  - four objectives
- Similar runtimes for
  - MA vs. MDP

Further results (excerpt)

Model	Par.	#lra-#tot	$\|S\|$	$\|MS\|$	$\|\Delta\|$	#EC	$\|S_{EC}\|$	#iter	Storm runtime
csn	3	3-0	177		427	38	158	9	1.23
csn	4	4-0	945		2753	176	880	30	109
csn	5	5-0	4833		$2{\cdot}10^4$	782	4622		TO
mut	3	2-0	$3{\cdot}10^4$		$5{\cdot}10^4$	1	$3{\cdot}10^4$	15	3.7
mut	4	2-0	$7{\cdot}10^5$		$1{\cdot}10^6$	1	$7{\cdot}10^5$	14	91.4
mut	5	2-0	$1{\cdot}10^7$		$3{\cdot}10^7$	1	$1{\cdot}10^7$	12	3197
clu	8-3	2-0	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	11	287
clu	16-4	2-0	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	10	4199
clu	32-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$		TO
clu	8-3	1-1	$2{\cdot}10^5$	$1{\cdot}10^5$	$4{\cdot}10^5$	4	$2{\cdot}10^5$	7	163
clu	16-4	1-1	$2{\cdot}10^6$	$9{\cdot}10^5$	$4{\cdot}10^6$	5	$2{\cdot}10^6$	9	3432
clu	32-3	1-1	$2{\cdot}10^6$	$1{\cdot}10^6$	$5{\cdot}10^6$	4	$2{\cdot}10^6$	7	3328
rqs	2-2	2-0	1619	628	2296	1	1618	63	4.52
rqs	3-3	2-0	$9{\cdot}10^4$	$4{\cdot}10^4$	$1{\cdot}10^5$	1	$9{\cdot}10^4$	106	162
rqs	5-3	2-0	$2{\cdot}10^6$	$1{\cdot}10^6$	$4{\cdot}10^6$	1	$2{\cdot}10^6$	97	4345
rqs	2-2	1-1	2805	1039	4159	1	1618	3	< 1
rqs	3-3	1-1	$1{\cdot}10^5$	$6{\cdot}10^4$	$3{\cdot}10^5$	1	$9{\cdot}10^4$	3	4.51
rqs	5-3	1-1	$3{\cdot}10^6$	$2{\cdot}10^6$	$7{\cdot}10^6$	1	$2{\cdot}10^6$	3	182

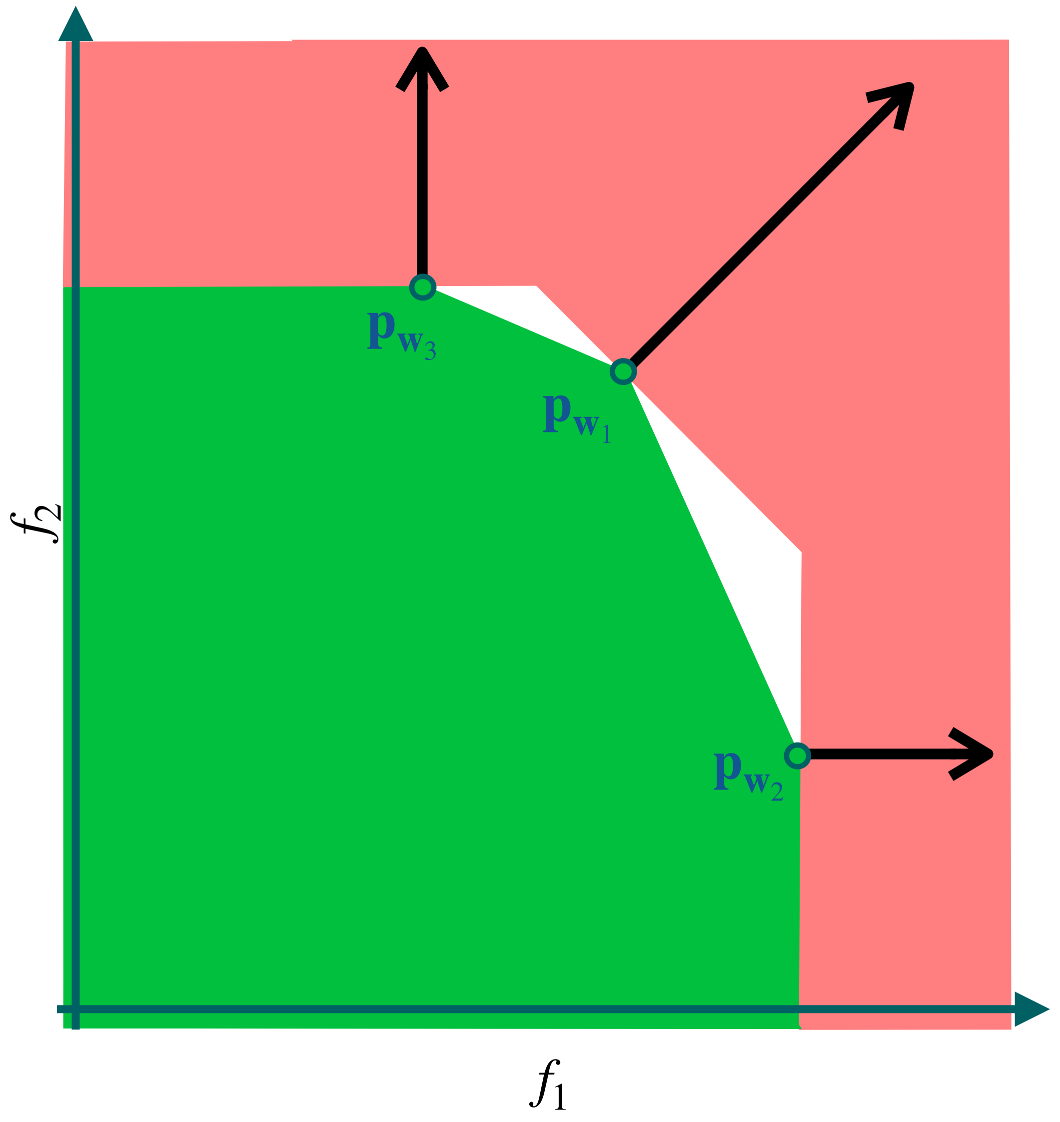
- Storm can handle
  - millions of states
  - four objectives
- Similar runtimes for
  - MA vs. MDP
  - pure LRA queries vs. mixtures

Software Modeling and Verification Chair
RWTH AACHEN UNIVERSITY

Conclusion

Anytime algorithm for approximating the set of **achievable points**

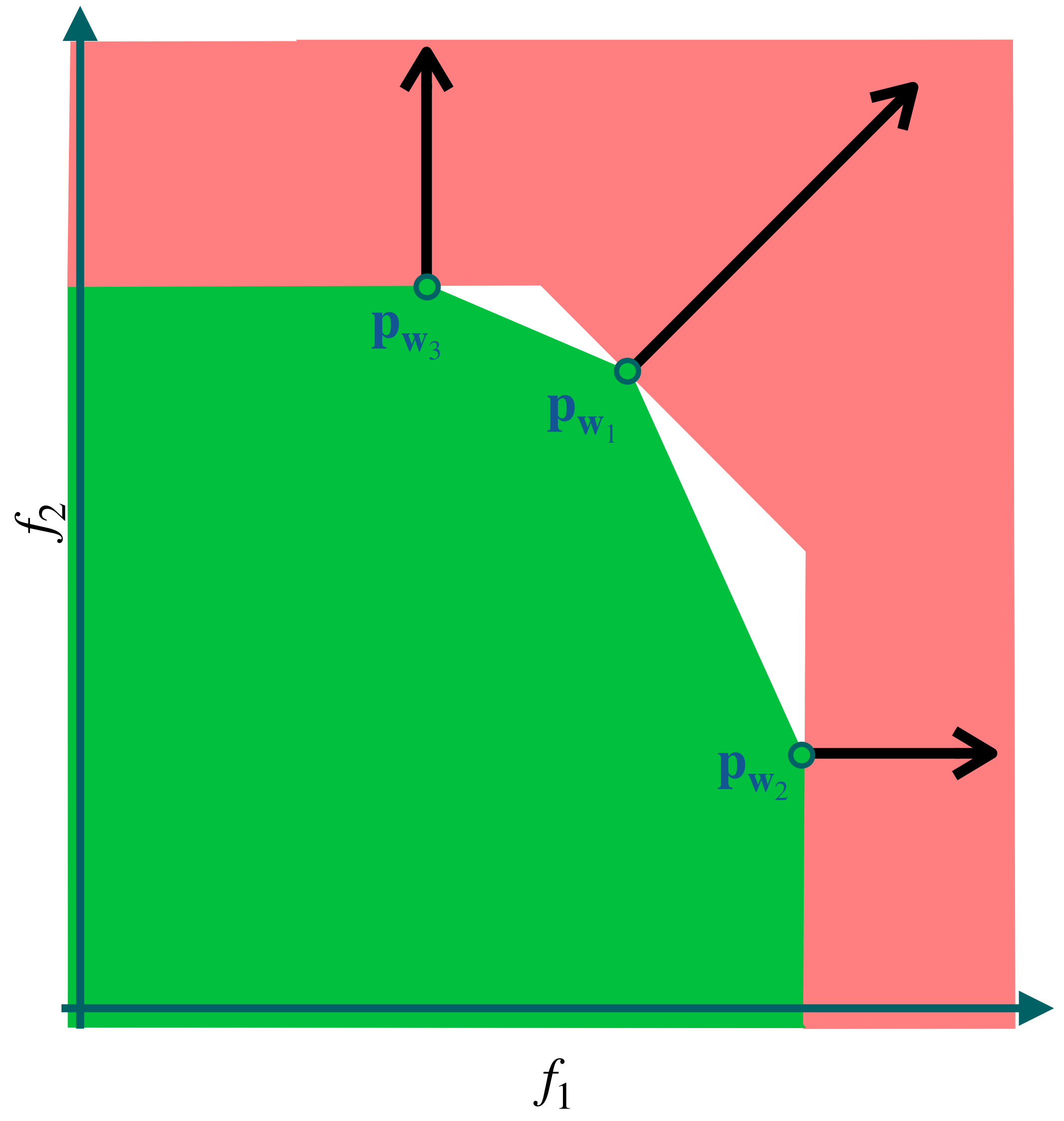
- Allows reusing single-objective techniques
- Applicable to all kinds of objectives, in particular mixtures of
  - **long-run average rewards** and
  - **total rewards**



Conclusion

Anytime algorithm for approximating the set of **achievable points**

- Allows reusing single-objective techniques
- Applicable to all kinds of objectives, in particular mixtures of
  - **long-run average rewards** and
  - **total rewards**



Implementation outperforms existing LP-based approach

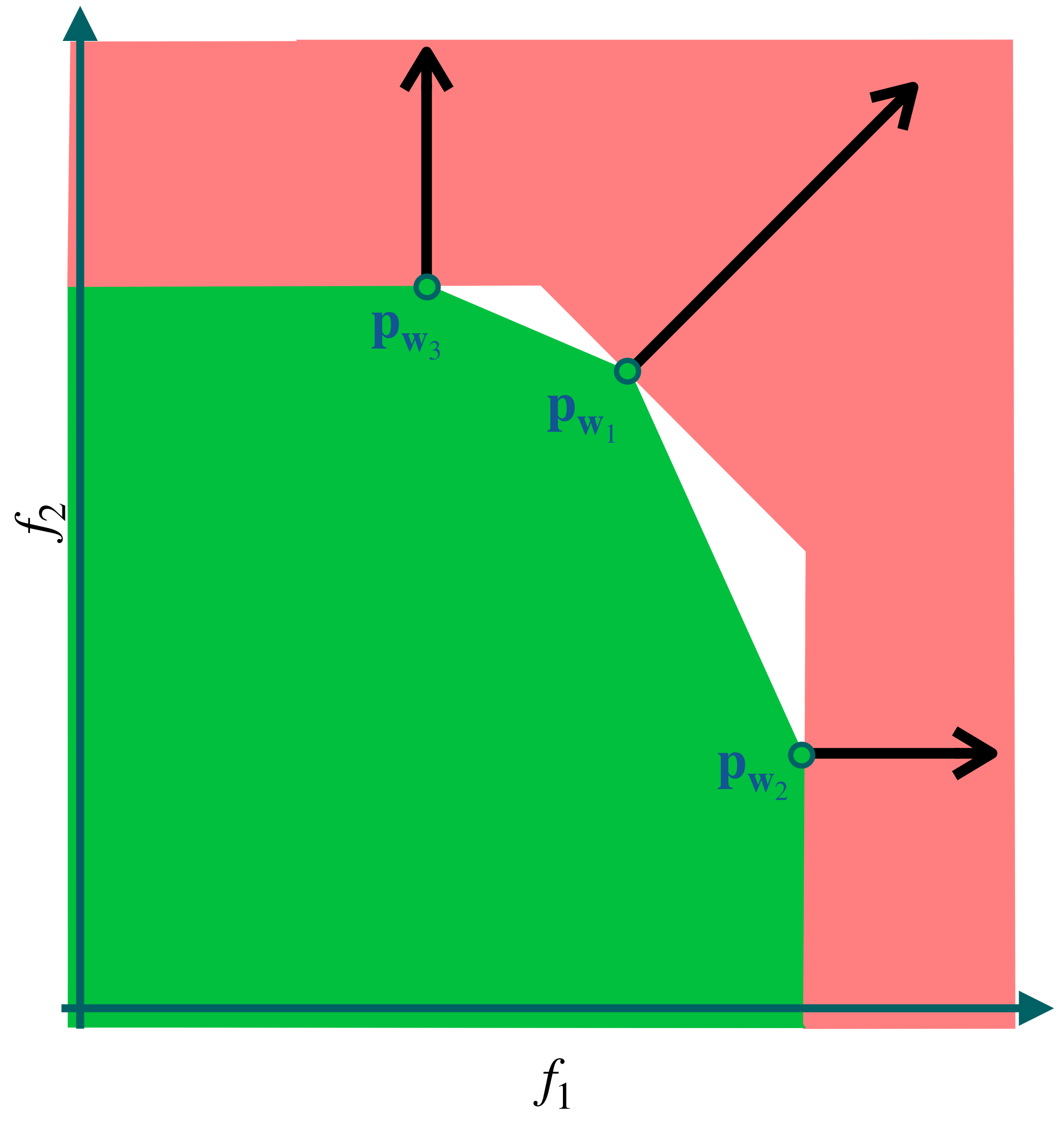
Storm

www.stormchecker.org

Conclusion

Anytime algorithm for approximating the set of **achievable points**

- Allows reusing single-objective techniques
- Applicable to all kinds of objectives, in particular mixtures of
  - **long-run average rewards** and
  - **total rewards**



Implementation outperforms existing LP-based approach

Storm

www.stormchecker.org

Future work:

- Partially observable models
- Stochastic games