CAV 2023

See also:
[Watanabe, van der Vegt, Hasuo, Rot & Junges, TACAS'24]
[Watanabe, van der Vegt, Junges & Hasuo, CAV'24]

bit.ly/4dMxY51

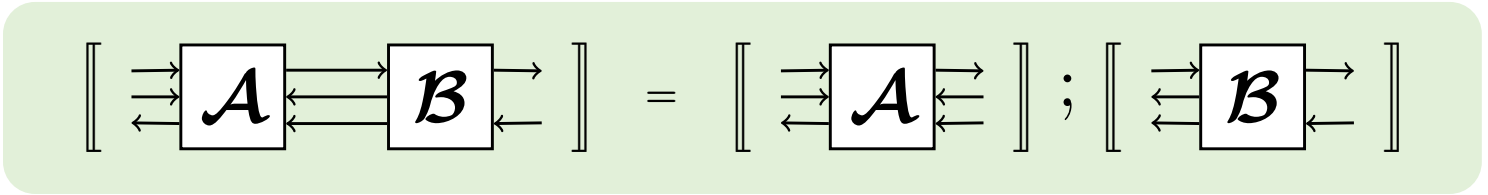
Compositional Probabilistic Model Checking with String Diagrams of MDPs

Kazuki Watanabe[1,2], Clovis Eberhart[1,3], Kazuyuki Asada[4], Ichiro Hasuo[1,2]

1: National Institute of Informatics, Tokyo, Japan
2: SOKENDAI (The Graduate University for Advanced Studies), Japan
3: Japanese-French Laboratory for Informatics (IRL 3527), Tokyo, Japan
4: Research Institute of Electrical Communication, Tohoku University, Sendai, Japan

Supported by ERATO HASUO Metamathematics for Systems Design Project No. JPMJER1603 and the JST Grant No. JPMJFS2136.

Outline



- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

A Paradigm with Conceptual Value, Performance Advantage, and Mathematical Blessing

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

A Paradigm with Conceptual Value, Performance Advantage, and Mathematical Blessing

The solution of a composition is obtained by ...

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

first solving the components $\mathcal{A}, \mathcal{B}$ and then composing their solutions

A Paradigm with Conceptual Value, Performance Advantage, and Mathematical Blessing

The solution of a composition is obtained by ...

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

Composition of ***systems*** (seqComp, parComp, sum, ...)

Composition of ***semantics***

first solving the components $\mathcal{A}$, $\mathcal{B}$ and then composing their solutions

Hasuo (NII, Tokyo)

A Paradigm with Conceptual Value, Performance Advantage, and Mathematical Blessing

The solution of a composition is obtained by ...

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

Composition of ***systems*** (seqComp, parComp, sum, ...)

Composition of ***semantics***

first solving the components $\mathcal{A}$, $\mathcal{B}$ and then composing their solutions

Conceptual Value

- "**Divide-and-Conquer**": simplifies a problem into smaller subproblems
- $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ are **summaries** of components $\mathcal{A}$, $\mathcal{B}$. Unnecessary details get abstracted away

Performance Advantage

- Clear adv. when there are **duplicates** (reuse $\mathcal{S}(\mathcal{A})$ **!**)

$$\mathcal{S}(\mathcal{A} \star \cdots \star \mathcal{A}) = \mathcal{S}(\mathcal{A}) \star \cdots \star \mathcal{S}(\mathcal{A})$$

- (In some cases you don't need duplicates, e.g. mergesort)

Mathematical Blessing

- Compositionality means that the solution

$$\mathcal{S} : \mathbb{M} \longrightarrow \mathbb{S}$$

is a **homomorphism**, preserving the operation ★

“Granularity of Semantics”: Minimal Enrichment to Account for Interaction

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

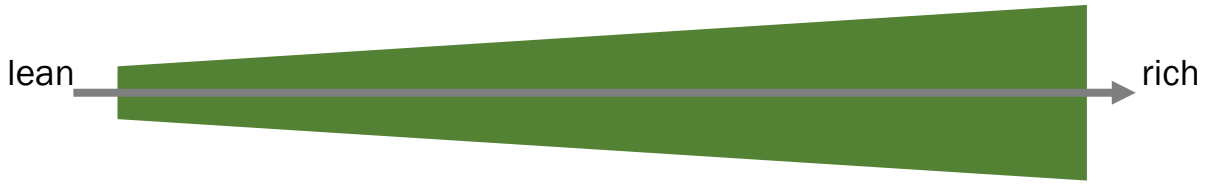
Q. What information should the summaries $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ carry?

"Granularity of Semantics": Minimal Enrichment to Account for Interaction

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

Imagine:
- estimating the output of a team of two
- $Av(X \cup Y) \neq \frac{Av(X) + Av(Y)}{2}$

Q. What information should the summaries $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ carry?

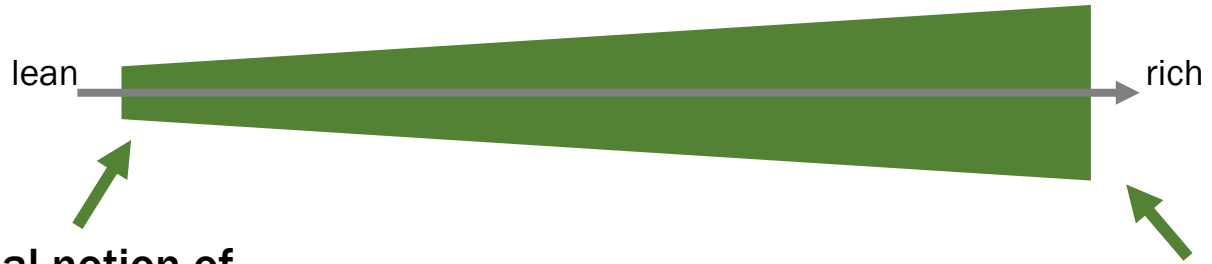


"Granularity of Semantics": Minimal Enrichment to Account for Interaction

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

Imagine:
- estimating the output of a team of two
- $Av(X \cup Y) \neq \frac{Av(X) + Av(y)}{2}$

Q. What information should the summaries $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ carry?



the **original notion of solution** (only)
→
- enough when $\mathcal{A}$, $\mathcal{B}$ operate separately;
- not enough otherwise

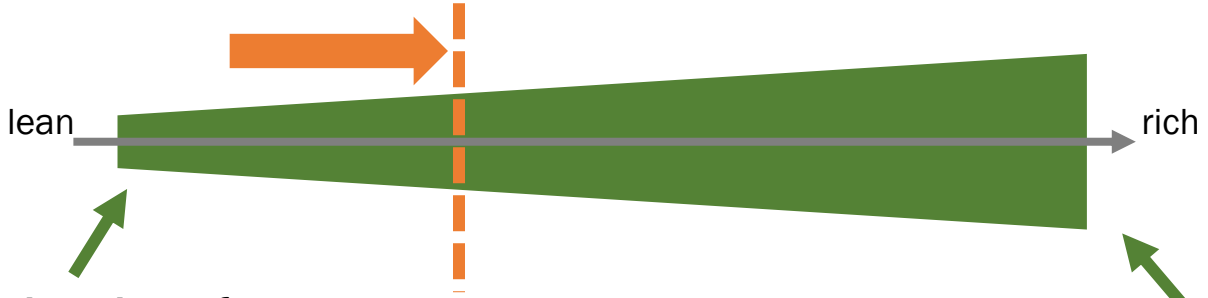
the **component itself**
→
no suppressed details,
no performance advantage

"Granularity of Semantics": Minimal Enrichment to Account for Interaction

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$

Imagine:
- estimating the output of a team of two
- $Av(X \cup Y) \neq \frac{Av(X) + Av(y)}{2}$

Q. What information should the summaries $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ carry?

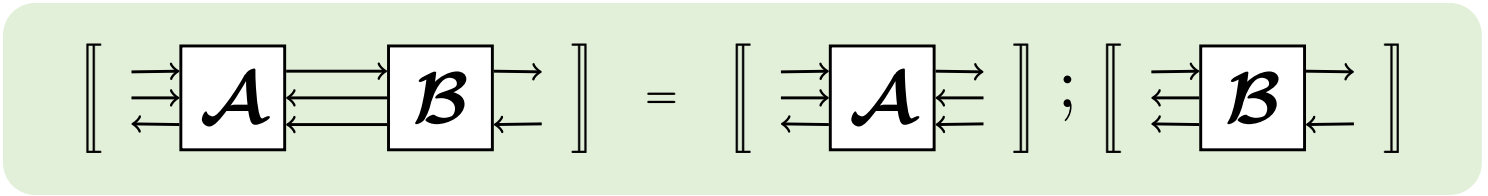


the **original notion of solution** (only)
→
- enough when $\mathcal{A}$, $\mathcal{B}$ operate separately;
- not enough otherwise

find **minimal enrichment** that addresses interaction

the **component itself**
→
no suppressed details,
no performance advantage

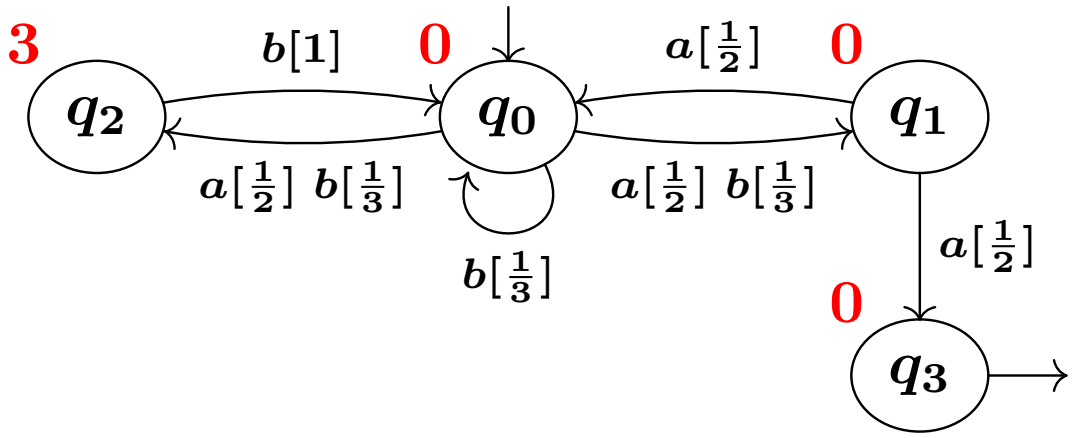
Outline



- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

Optimal Expected Reward of MDPs: Scheduler Synthesis + Its Performance Guarantee

Markov Decision Process (MDP)



- State-based model with **actions** (a, b, ...) and **probabilistic uncertainties**
- Basic framework in many research areas (e.g. reinforcement learning)
- General modeling formalism for decision making in an uncertain environment

Goal: Compute Optimal Expected Reward

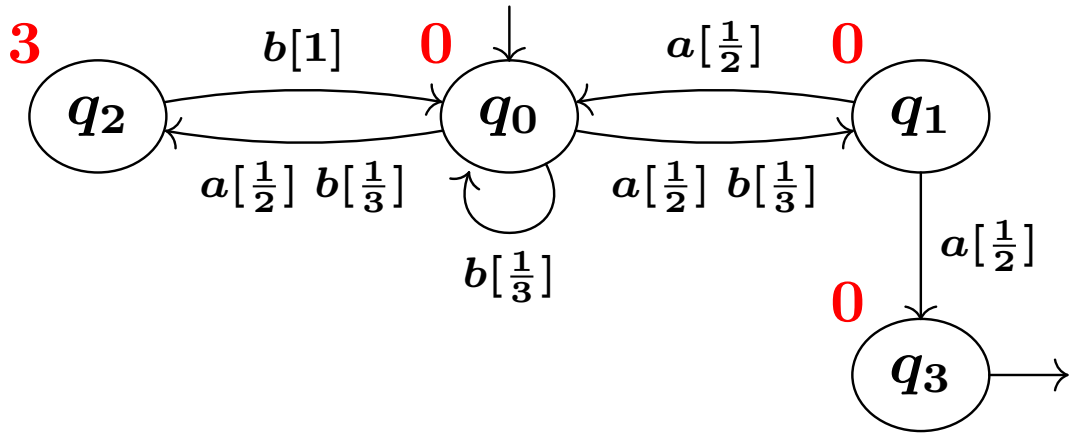
Problem:

- Given an MDP,
- Compute the optimal *scheduler* (~ controller, strategy; it chooses actions) and its expected cumulative reward

Applications:

- **Scheduler synthesis**
  "what is the best strategy?"
- **Formal verificaiton**
  "How much cumulative reward can I expect?"
  "Is the expectation correct?"

Optimal Expected Reward of MDPs: Details



Def. An ***MDP*** $\mathcal{A}$ consists of...

- a finite set Q of ***positions***,
- an ***initial position*** q_I and a ***final position*** q_F,
- a ***transition function***

  $P\colon Q \times A \times Q \longrightarrow [0,1]$

  s.t., for each q, a, $\sum_{q'} P(q,a,q') \in \{0,1\}$
- and a ***reward function***

  $R\colon Q \longrightarrow \mathbb{R}_{\geq 0}$

In this work, we accumulate rewards

- without discount ($\gamma = 1$), and
- only **along terminating paths** ($\mathbf{Path}^{\mathcal{A}} = \{\text{paths from } q_I \text{ to } q_F\}$)

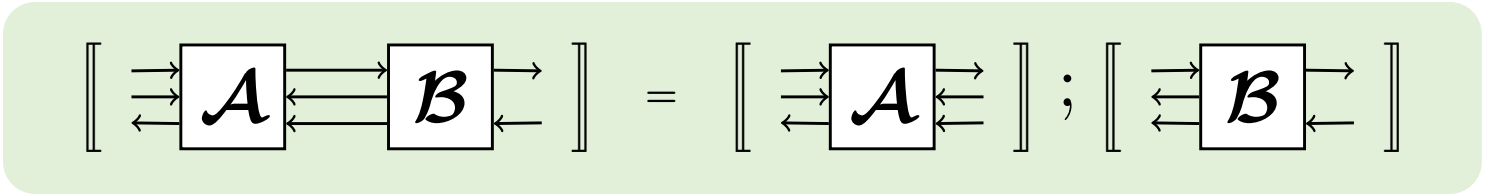
The latter, as is well-known, implies that memoryless schedulers suffice.

A scheduler $\tau\colon Q \to A$ determines

$$\mathbf{ERw}^{\mathcal{A},\tau} := \sum_{\pi \in \mathbf{Path}^{\mathcal{A}}} \mathbf{Pr}^{\mathcal{A},\tau}(\pi) \cdot \mathbf{Rw}^{\mathcal{A}}(\pi)$$

We want an optimal scheduler τ.

Outline

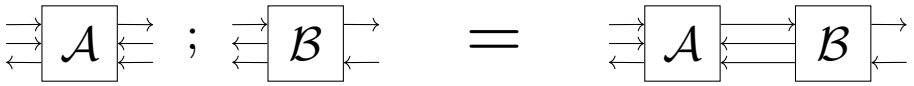


- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

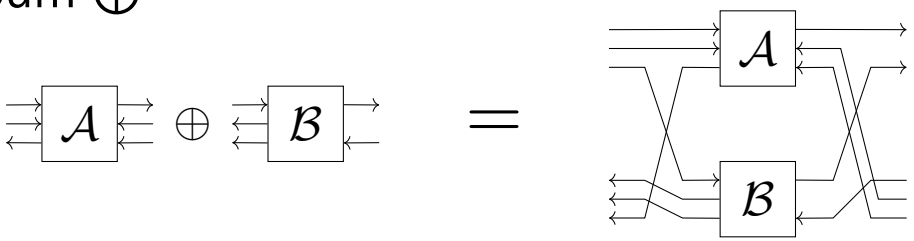
String Diagrams of MDPs: Planar Composition with SeqComp ; and Sum ⊕

String Diagram of MDPs

- Sequential composition ;



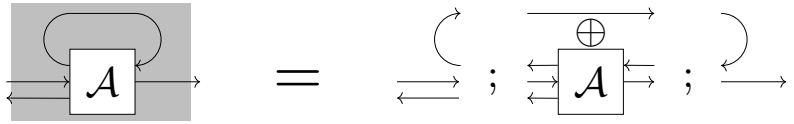
- Sum ⊕



- and some "constants" ↺ , ↻ , ⤨ , ...

→ **planar composition** of MDPs
(mostly *sequential* composition; not parallel)

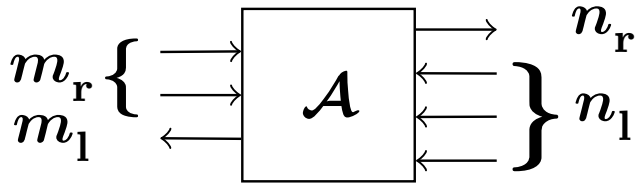
- Loop is a derived operation:



Background: Monoidal Categories

- Well-established topic of category theory
(Mac Lane, Kelly, Joyal, Street, ...)
- Used for many applications:
quantum field theory (Khavanov, ...),
quantum computation (Abramsky, Coecke, Vicary, Heunen, ...),
linguistics (Sadrzadeh, Coecke, ...),
signal flow diagrams (Bonchi, Sobocinski, Zanasi, ...), ...
- String diagrams as an *internal language* for monoidal categories [Joyal & Street, Adv. Math. 1991]
  - nicely expressive (planar composition, see left)
  - comes with a rich metatheory (see later)
- In particular, we use string diagrams for **compact closed categories** (compCCs).
  - Dual object A^* for each A
  - Example: fin.-dim. vector spaces, MDPs, ...

String Diagrams of MDPs: Details



- ***Open MDPs*** extend MDPs with ***open ends***:
  (left, right) × (entrance, exit)

- An open MDP thus comes with an ***arity***.
  E.g. $\mathcal{A}\colon (2,1) \longrightarrow (1,3)$

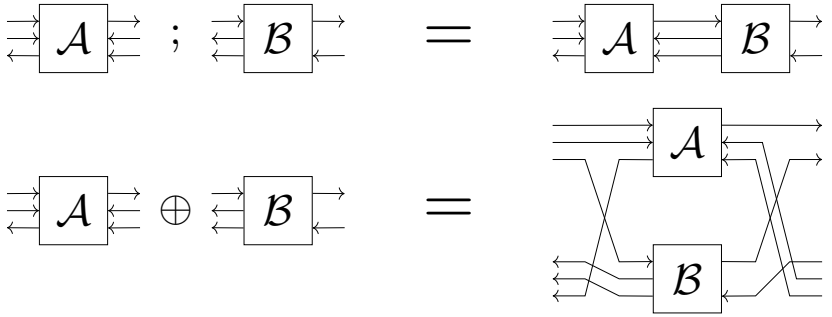
- Open MDPs form
  a compact closed category $\mathbf{oMDP}$, with

$$\frac{\mathcal{A} : (m_r, m_l) \longrightarrow (n_r, n_l) \qquad \mathcal{B} : (n_r, n_l) \longrightarrow (k_r, k_l)}{\mathcal{A}\,;\mathcal{B} \;\; : (m_r, m_l) \longrightarrow (k_r, k_l)}$$

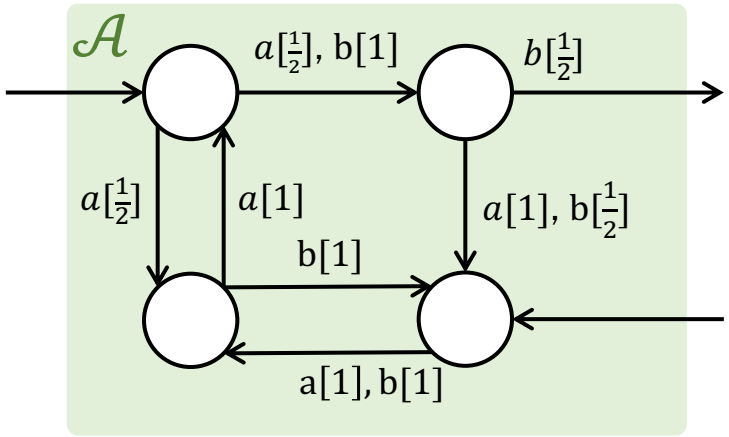
$$\frac{\mathcal{A} : (m_r, m_l) \longrightarrow (n_r, n_l) \qquad \mathcal{B} : (k_r, k_l) \longrightarrow (l_r, l_l)}{\mathcal{A} \oplus \mathcal{B} \;\; : (m_r + k_r, m_l + k_l) \longrightarrow (n_r + l_r, n_l + l_l)}$$

Def. (open MDP, oMDP) Let A be a non-empty finite set, whose elements are called *actions*. An *open MDP* $\mathcal{A}$ (*over* the action set A) is the tuple $(\overline{m}, \overline{n}, Q, A, E, P, R)$ of the following data. We say that it is *from* $\overline{m}$ *to* $\overline{n}$.

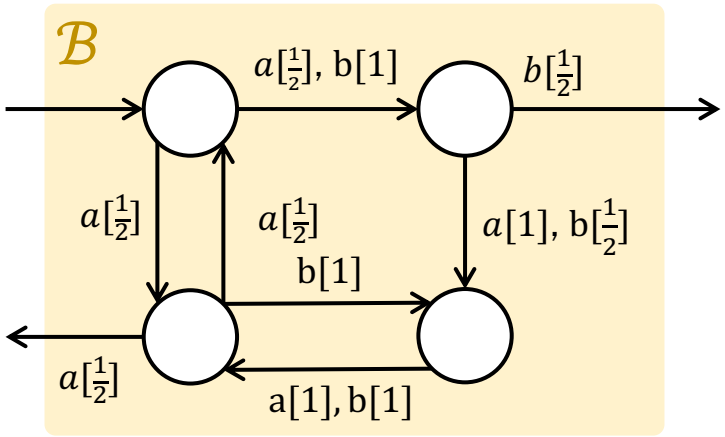
1. $\overline{m} = (m_r,\ m_l)$ and $\overline{n} = (n_r,\ n_l)$ are pairs of natural numbers; they are called the *left-arity* and the *right-arity*, respectively. Moreover, elements of $[m_r + n_l]$ are called *entrances*, and those of $[n_r + m_l]$ are called *exits*.
2. Q is a finite set of *positions*.
3. $E : [m_r + n_l] \to Q + [n_r + m_l]$ is an *entry function*, which maps each entrance to either a position (in Q) or an exit (in $[n_r + m_l]$).
4. $P : Q \times A \times (Q + [n_r + m_l]) \to \mathbb{R}_{\geq 0}$ determines *transition probabilities*, where we require $\sum_{s' \in Q + [n_r + m_l]} P(s, a, s') \in \{0, 1\}$ for each $s \in Q$ and $a \in A$.
5. R is a *reward function* $R : Q \to \mathbb{R}_{\geq 0}$.
6. We impose the following "unique access to each exit" condition. Let $\mathbf{exits} : ([m_r + n_l] + Q) \to \mathcal{P}([n_r + m_l])$ be the *exit function* that collects all immediately reachable exits, that is, 1) for each $s \in Q$, $\mathbf{exits}(s) = \{t \in [n_r + m_l] \mid \exists a \in A. P(s, a, t) > 0\}$, and 2) for each entrance $s \in [m_r + n_l]$, $\mathbf{exits}(s) = \{E(s)\}$ if $E(s)$ is an exit and $\mathbf{exits}(s) = \emptyset$ otherwise.
   - For all $s, s' \in [m_r + n_l] + Q$, if $\mathbf{exits}(s) \cap \mathbf{exits}(s') \neq \emptyset$, then $s = s'$.
   - We further require that each exit is reached from an identical position by at most one action. That is, for each exit $t \in [n_r + m_l]$, $s \in Q$, and $a, b \in A$, if both $P(s, a, t) > 0$ and $P(s, b, t) > 0$, then $a = b$.



String Diagrams of MDPs: Examples

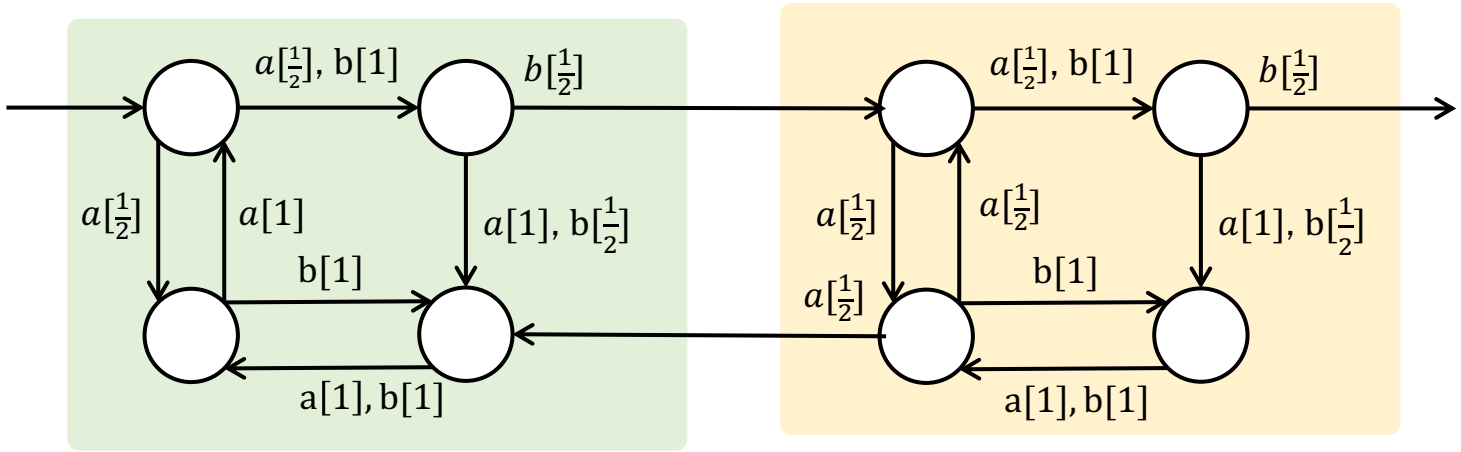


$: (1,0) \to (1,1)$



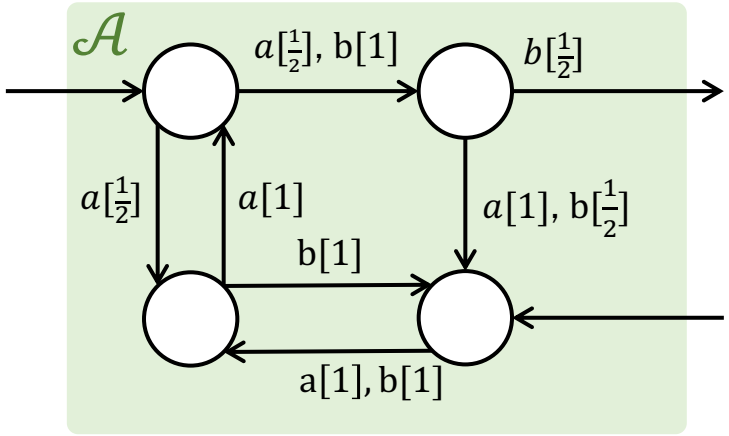
$: (1,1) \to (1,0)$

$\mathcal{A}\,;\,\mathcal{B} =$

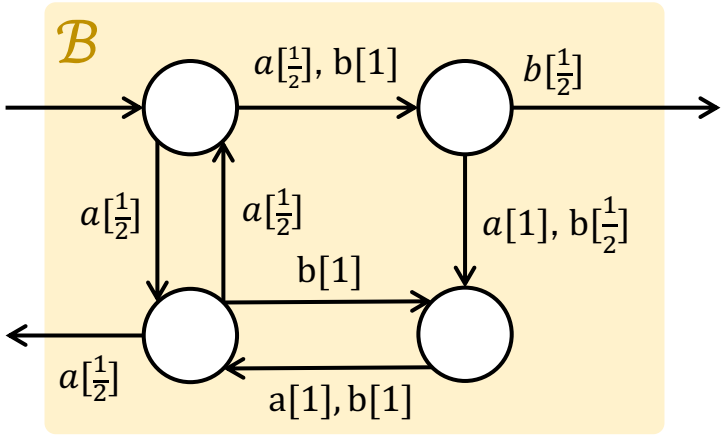


$: (1,0) \to (1,0)$

String Diagrams of MDPs: Examples

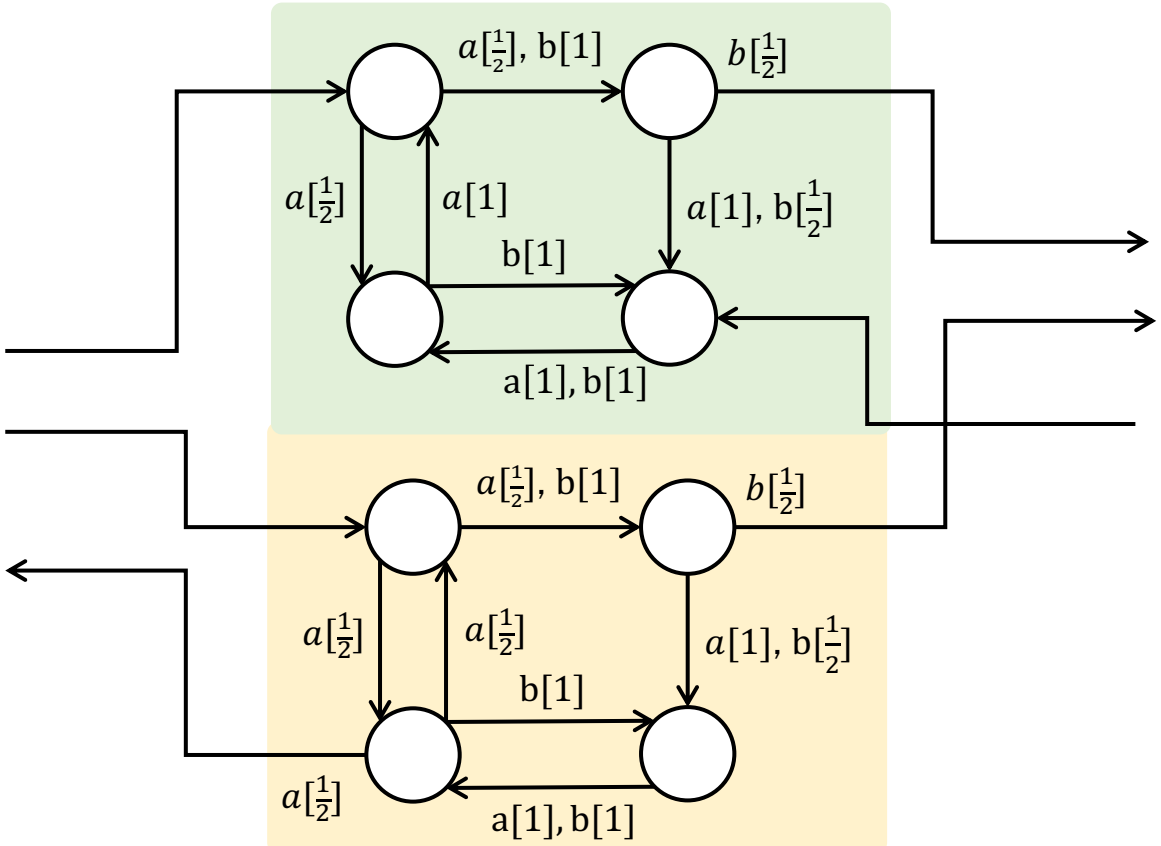


$: (1,0) \to (1,1)$



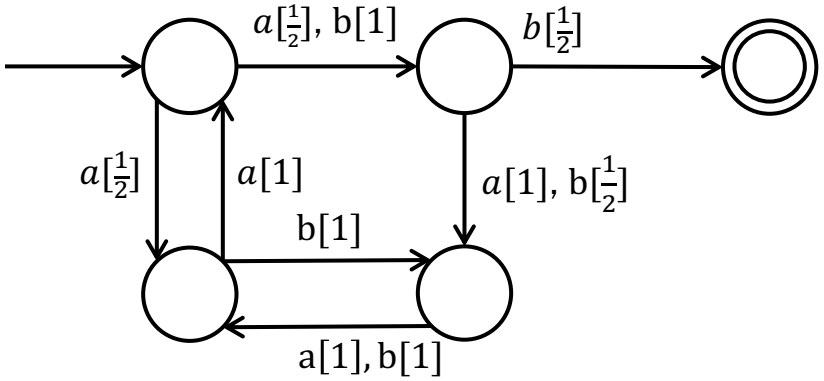
$: (1,1) \to (1,0)$

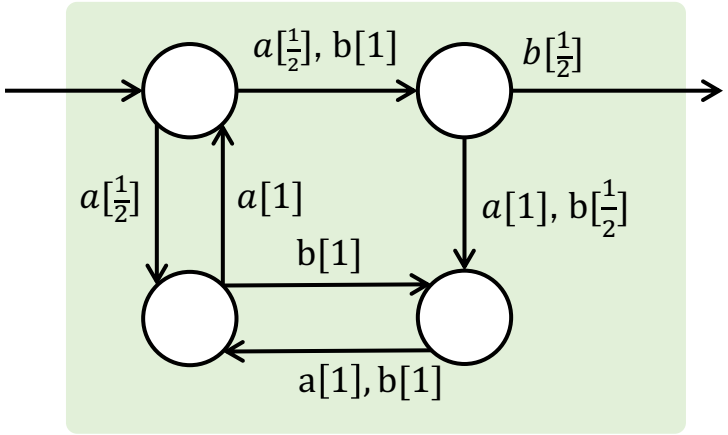
$\mathcal{A} \oplus \mathcal{B} =$



$: (2,1) \to (2,1)$

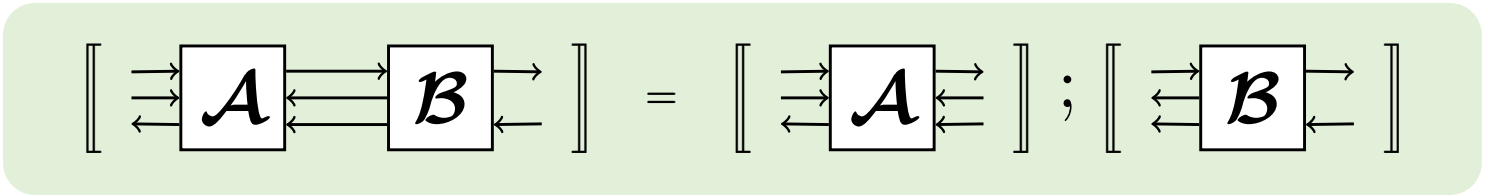
String Diagrams of MDPs: (Usual) MDPs as Open MDPs





$: (1,0) \to (1,0)$

Outline



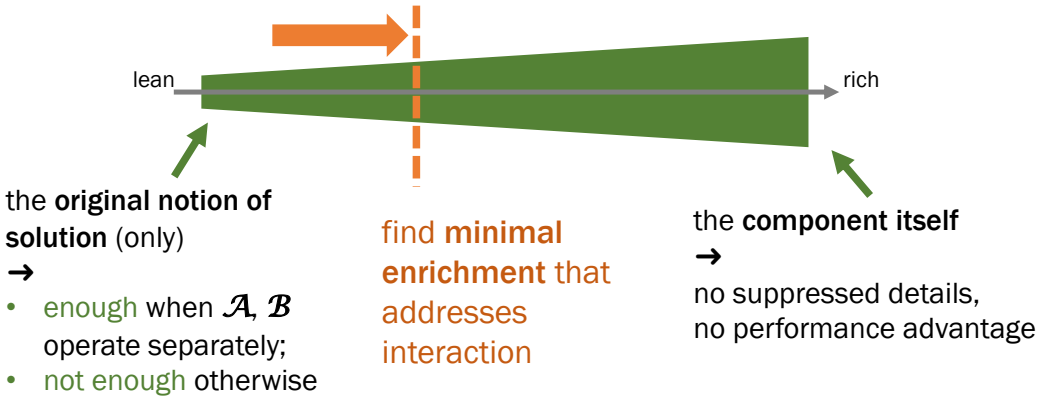
- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

Monoidal Categories for Compositional Algorithms: Two Benefits

Benefit 1: Settling "Granularity of Semantics"

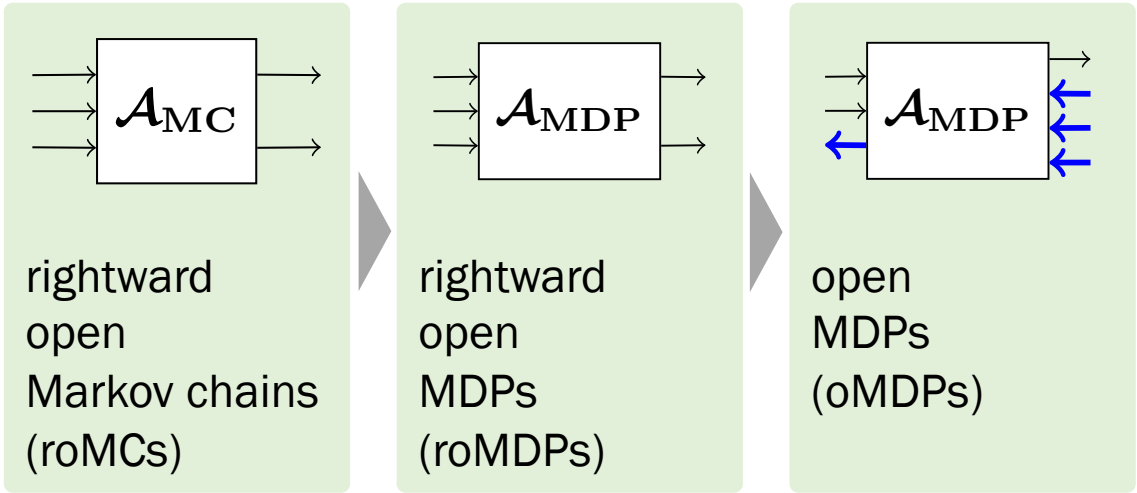
What information should the "summaries" $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ carry?

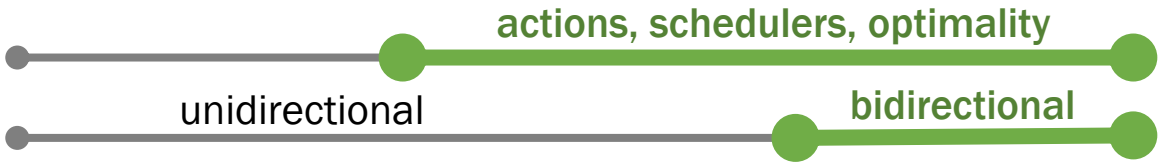
$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$



Benefit 2: System Upgrades for Free

We want to design compositional algorithms (= semantics) **incrementally**, by

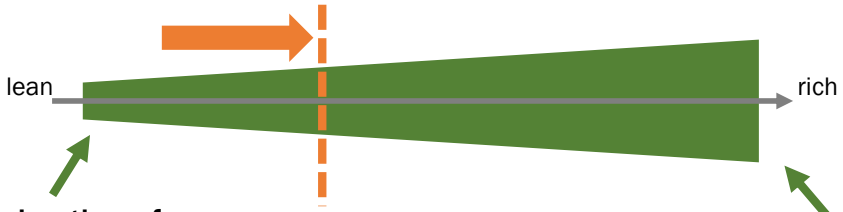




Monoidal Categories for Compositional Algorithms: Two Benefits

Q. What information should the "summaries" $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ carry?

$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$



the **original notion of solution** (only)
→
- enough when $\mathcal{A}$, $\mathcal{B}$ operate separately;
- not enough otherwise

find **minimal enrichment** that addresses interaction

the **component itself**
→
no suppressed details,
no performance advantage

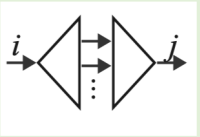
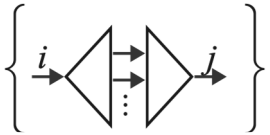
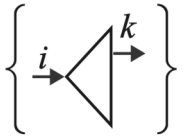
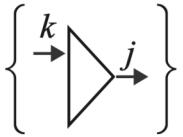
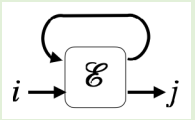
Benefit 1: Settling "Granularity of Semantics"

- Compositional solution is a homomorphism

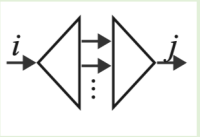
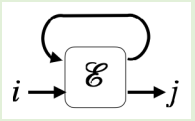
$$\mathcal{S} : \mathbb{M} \longrightarrow \mathbb{S}$$

- The domain $\mathbb{M}$ is a compact closed category (compCC) of MDPs
  → **We need $\mathbb{S}$ to be a compCC too!**
  - Minimally enrich the original solution domain (optimal expected reward) till it becomes a compCC
- Turns out: it suffices to additionally compute **reachability probabilities**
  - "decomposition equalities"

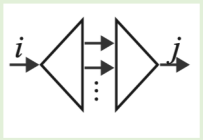
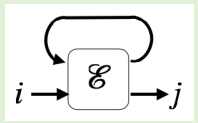
Decomposition Equalities for (rightward open) Markov Chains

	reachability probability	expected reward
seq. comp. 	$\mathbf{RPr}\left\{ \; i \; \right\} = \sum_{k} \mathbf{RPr}\left\{ \; i \; \right\} \times \mathbf{RPr}\left\{ \; k \; \right\}$    (Folklore)	
trace  (trace is primitive in a uni-dir. setting)		

Decomposition Equalities for (rightward open) Markov Chains

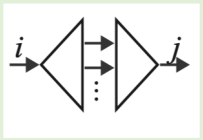
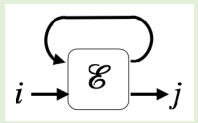
	reachability probability	expected reward
seq. comp. 	$\mathbf{RPr}\left\{ i \to \cdots \to j \right\} = \sum_{k} \mathbf{RPr}\left\{ i \to k \right\} \times \mathbf{RPr}\left\{ k \to j \right\}$ (Folklore)	
trace  (trace is primitive in a uni-dir. setting)	$\Pr\left[i \to \mathscr{C} \to j \right] = \Pr\left[i \to \mathscr{C} \to j \right] + \sum_{d \in \mathbb{N}} \Pr\left[i \to \mathscr{C} \to \cdots \to \mathscr{C} \to j \right]$ (d times) (Girard's *execution formula*)	

Decomposition Equalities for (rightward open) Markov Chains

	reachability probability	expected reward
seq. comp. 	$\mathbf{RPr}\left\{ \begin{array}{c} i \\ \rightarrow \\ j \end{array} \right\} = \sum_{k} \mathbf{RPr}\left\{ \begin{array}{c} i \\ \rightarrow \\ k \end{array} \right\} \times \mathbf{RPr}\left\{ \begin{array}{c} k \\ \rightarrow \\ j \end{array} \right\}$ (Folklore)	$\mathbf{ERw}\left\{ \begin{array}{c} i \\ \rightarrow \\ j \end{array} \right\} = \sum_{k} \mathbf{RPr}\left\{ \begin{array}{c} i \\ \rightarrow \\ k \end{array} \right\} \times \mathbf{ERw}\left\{ \begin{array}{c} k \\ \rightarrow \end{array} \right\} + \sum_{k} \mathbf{ERw}\left\{ \begin{array}{c} i \\ \rightarrow \\ k \end{array} \right\} \times \mathbf{RPr}\left\{ \begin{array}{c} k \\ \rightarrow \end{array} \right\}$ (Prop. 3.2)
trace  (trace is primitive in a uni-dir. setting)	$\mathrm{Pr}\left[\begin{array}{c} i \\ \rightarrow \\ \mathscr{C} \\ \rightarrow \\ j \end{array} \right] = \mathrm{Pr}\left[\begin{array}{c} i \\ \rightarrow \\ \mathscr{C} \\ \rightarrow \\ j \end{array} \right] + \sum_{d \in \mathbb{N}} \mathrm{Pr}\left[\begin{array}{c} i \\ \rightarrow \\ \mathscr{C} \\ \rightarrow \\ \mathscr{C} \\ \rightarrow \\ \cdots \\ \rightarrow \\ \mathscr{C} \\ \rightarrow \\ \mathscr{C} \\ \rightarrow \\ j \end{array} \right]$ (d times) (Girard's *execution formula*)	$\left[\mathbf{ERw}^{\mathbf{tr}_{l;m,n}(\mathcal{E})}(i,j) \right]_{i,j} = \left[\mathbf{ERw}^{\mathcal{E}}(l+i, l+j) \right]_{i,j} + \sum_{d \in \mathbb{N}} \begin{bmatrix} \begin{pmatrix} \left[\mathbf{RPr}^{\mathcal{E}}(l+i,k) \right]_{i,k} & \left[\mathbf{ERw}^{\mathcal{E}}(l+i,k) \right]_{i,k} \end{pmatrix} \\ \cdot \begin{pmatrix} \left[\mathbf{RPr}^{\mathcal{E}}(k,k') \right]_{k,k'} & \left[\mathbf{ERw}^{\mathcal{E}}(k,k') \right]_{k,k'} \\ \left[0 \right]_{k,k'} & \left[\mathbf{RPr}^{\mathcal{E}}(k,k') \right]_{k,k'} \end{pmatrix}^{d} \\ \cdot \begin{pmatrix} \left[\mathbf{ERw}^{\mathcal{E}}(k', l+j) \right]_{k',j} \\ \left[\mathbf{RPr}^{\mathcal{E}}(k', l+j) \right]_{k',j} \end{pmatrix} \end{bmatrix}$ (Prop. 3.2)

Decomposition Equalities for (Uncountable) Markov Chains

For ERw, record RPr as well!

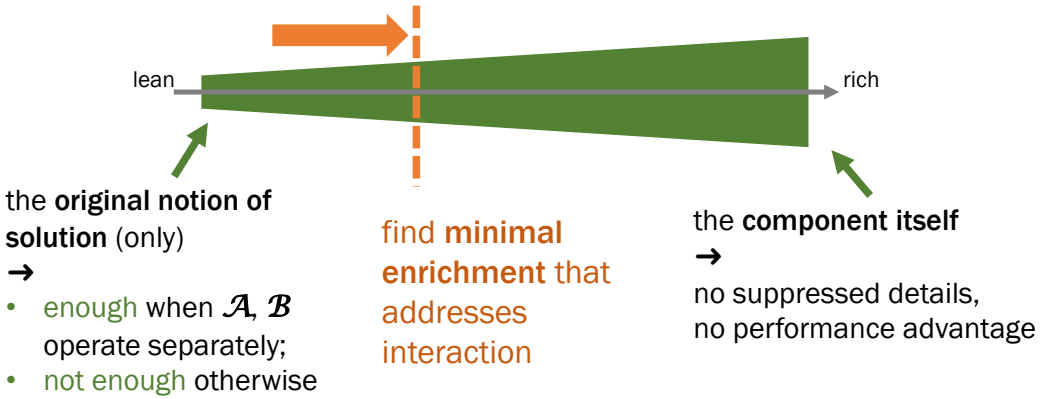
	reachability pro	expected reward
seq. comp. 	$\mathbf{RPr}\left\{ i \to \bullet \to j \right\} = \sum_{k} \mathbf{RPr}\left\{ i \to k \right\} \times \mathbf{RPr}\left\{ k \to j \right\}$ (Folklore)	$\mathbf{ERw}\left\{ i \to \bullet \to j \right\} = \sum_{k} \mathbf{RPr}\left\{ i \to k \right\} \times \mathbf{ERw}\left\{ k \to \bullet \right\} + \sum_{k} \mathbf{ERw}\left\{ i \to k \right\} \times \mathbf{RPr}\left\{ k \to \bullet \right\}$ (Prop. 3.2)
trace  (trace is primitive in a uni-dir. setting)	$\Pr\left[i \to \mathscr{C} \to j \right] = \Pr\left[i \to \mathscr{C} \to j \right] + \sum_{d \in \mathbb{N}} \Pr\left[i \to \mathscr{C} \xrightarrow{d \text{ times}} \mathscr{C} \to j \right]$ (Girard's *execution formula*)	$\left[\mathbf{ERw}^{\mathrm{tr}_{l;m,n}(\mathcal{E})}(i,j) \right]_{i,j} = \left[\mathbf{ERw}^{\mathcal{E}}(l+i, l+j) \right]_{i,j} + \sum_{d \in \mathbb{N}} \begin{bmatrix} \begin{pmatrix} \left[\mathbf{RPr}^{\mathcal{E}}(l+i,k) \right]_{i,k} & \left[\mathbf{ERw}^{\mathcal{E}}(l+i,k) \right]_{i,k} \end{pmatrix} \\ \cdot \begin{pmatrix} \left[\mathbf{RPr}^{\mathcal{E}}(k,k') \right]_{k,k'} & \left[\mathbf{ERw}^{\mathcal{E}}(k,k') \right]_{k,k'} \\ \left[0 \right]_{k,k'} & \left[\mathbf{RPr}^{\mathcal{E}}(k,k') \right]_{k,k'} \end{pmatrix}^{d} \\ \cdot \begin{pmatrix} \left[\mathbf{ERw}^{\mathcal{E}}(k', l+j) \right]_{k',j} \\ \left[\mathbf{RPr}^{\mathcal{E}}(k', l+j) \right]_{k',j} \end{pmatrix} \end{bmatrix}$ (Prop. 3.2)

Monoidal Categories for Compositional Algorithms: Two Benefits

Benefit 1: Settling "Granularity of Semantics"

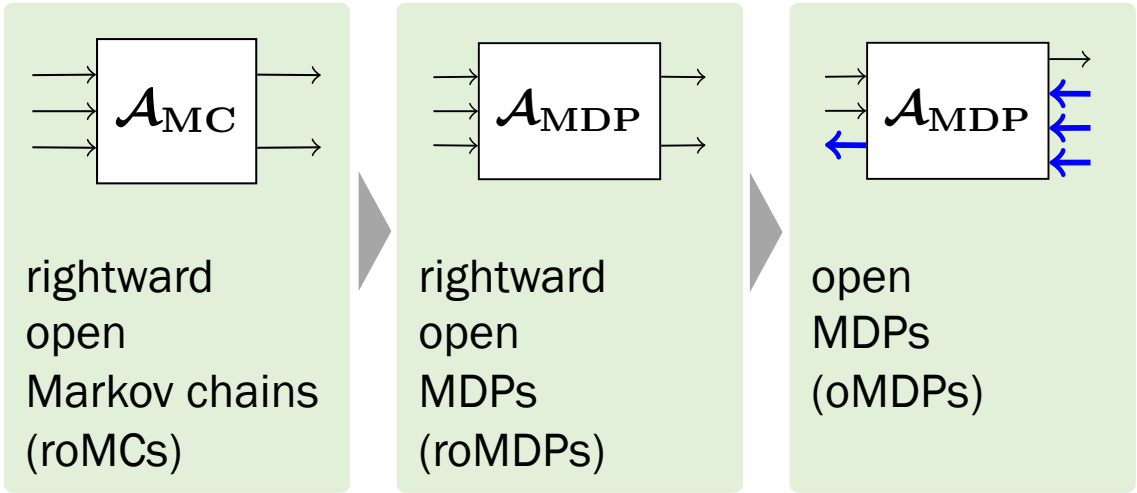
What information should the "summaries" $\mathcal{S}(\mathcal{A})$, $\mathcal{S}(\mathcal{B})$ carry?

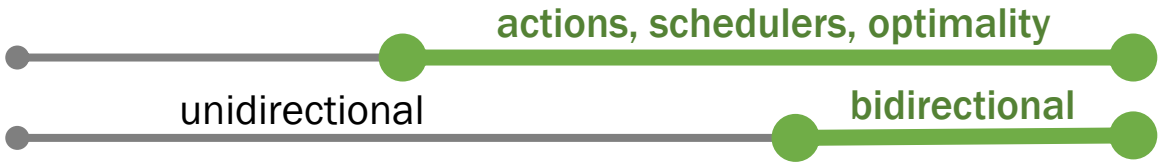
$$\mathcal{S}(\mathcal{A} \star \mathcal{B}) = \mathcal{S}(\mathcal{A}) \star \mathcal{S}(\mathcal{B})$$



Benefit 2: System Upgrades for Free

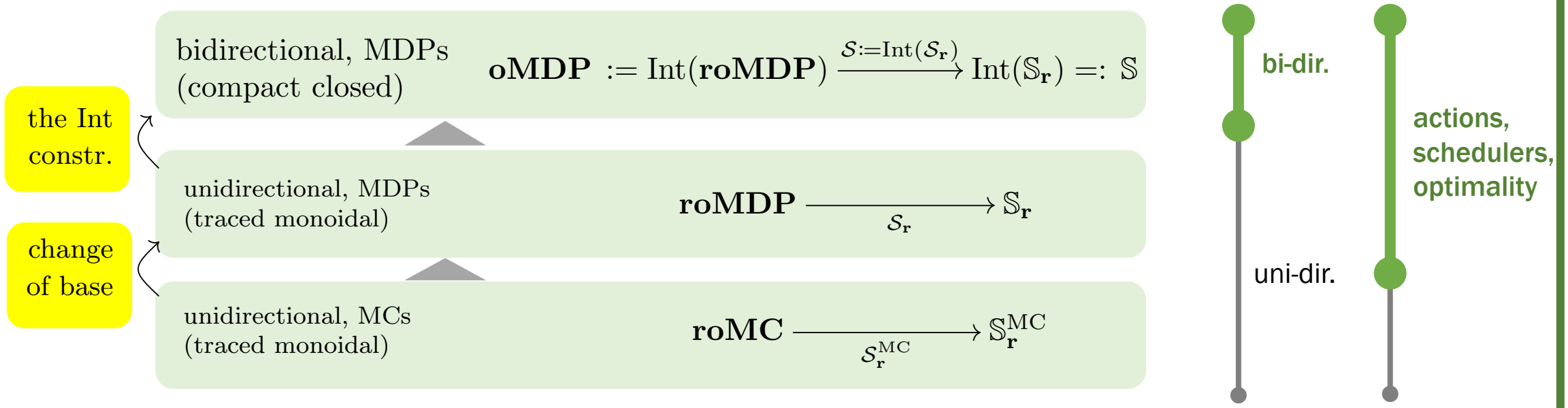
We want to design compositional algorithms (= semantics) **incrementally**, by





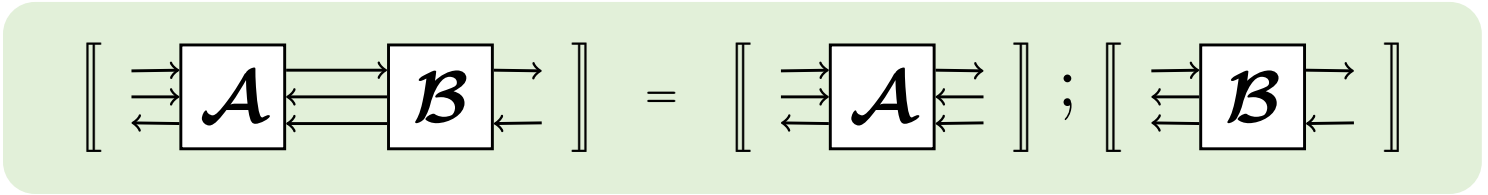
Monoidal Categories for Compositional Algorithms: Two Benefits

Benefit 2: Framework Upgrades for Free (adding actions, adding bidirectionality)



- Building semantical frameworks is easy in a simpler setting (*rightward-open MCs*; no action, end points are all rightward)
- Upgrades are **for free,** exploiting the categorical metatheory change of base (Eilenberg, Kelly, …), the Int construction (Joyal, Street, Verity)

Outline



- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

Compositional Semantics for Rightward Open MCs

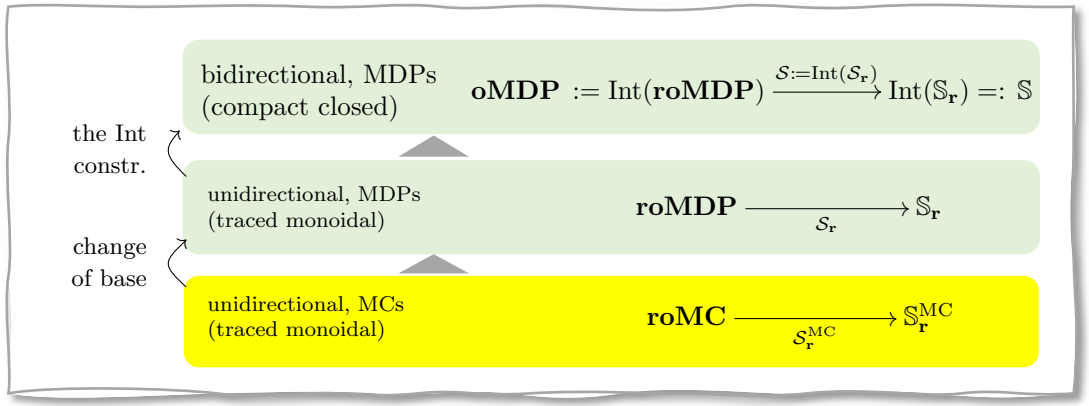
Def. ($\mathbb{S}_{\mathbf{r}}^{\mathbf{MC}}$)
Object: natural number m
Arrow:

$$\frac{f : m \longrightarrow n \quad \text{in } \mathbb{S}_{\mathbf{r}}^{\mathbf{MC}}}{f = (p_{i,j}, r_{i,j})_{i \in [1,m], j \in [1,n]}}$$

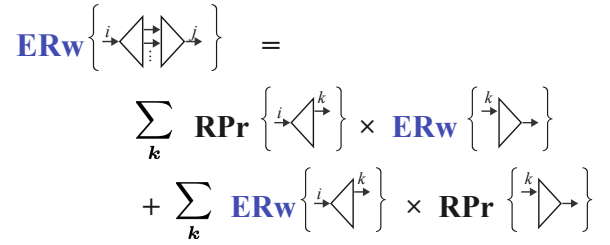
where

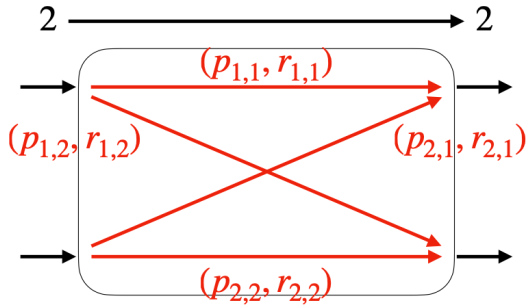
- (Subnormality) $\sum_{j \in [1,n]} p_{i,j} \leq 1$ for each $i \in [1, m]$.
- (Realizability) $p_{i,j} = 0$ implies $r_{i,j} = 0$.

Thm.
Both **roMC** and $\mathbb{S}_{\mathbf{r}}^{\mathbf{MC}}$ are traced symmetric monoidal categories (TSMCs). □



Decomposition equalities:

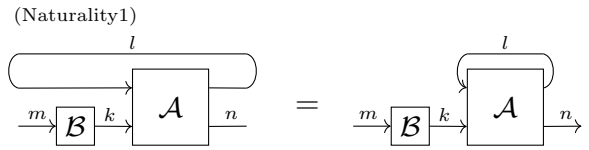




recording both RPr and ERw...

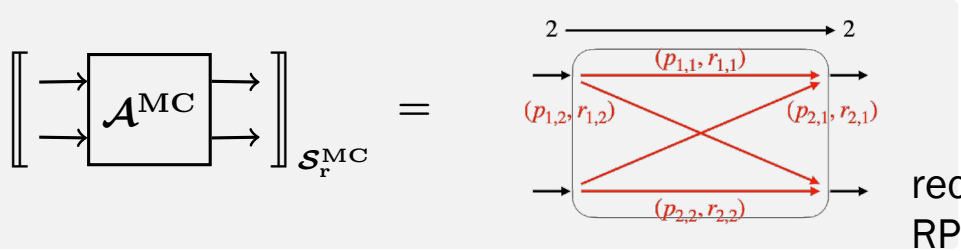
$\mathbb{S}_{\mathbf{r}}^{\mathbf{MC}}$ has TSMC (alg.) operations ; , ⊕, tr mirroring the decomposition equalties...

w/ equational axioms e.g.

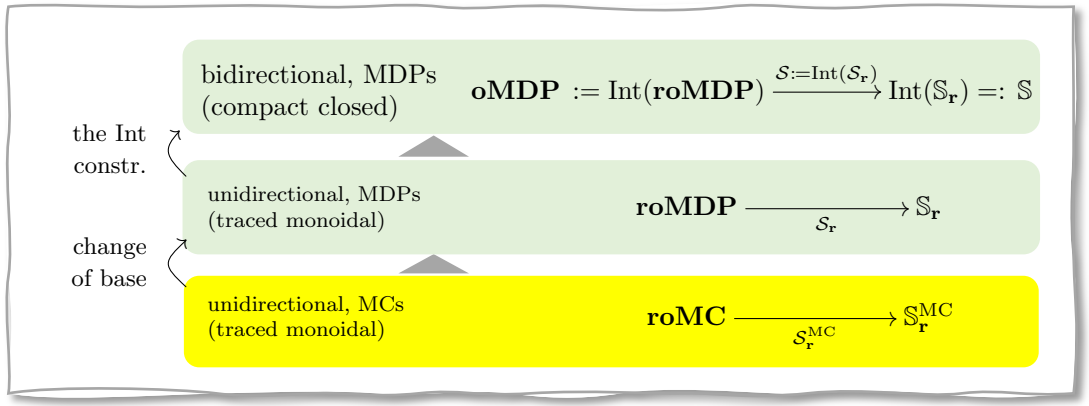


Compositional Semantics for Rightward Open MCs

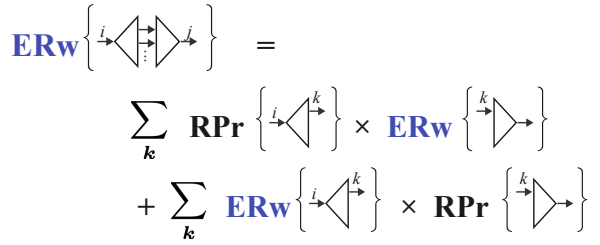
The solution functor $\mathcal{S}^{\mathbf{MC}}_{\mathbf{r}} : \mathbf{roMC} \longrightarrow \mathbb{S}^{\mathbf{MC}}_{\mathbf{r}}$ is defined in a natural manner:



recording both RPr and ERw...



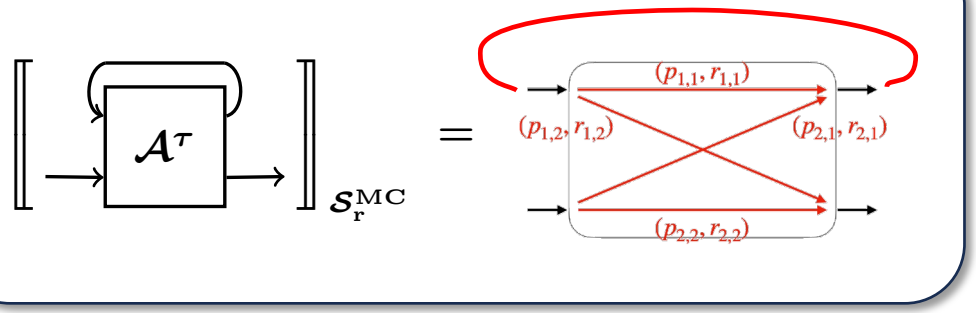
Decomposition equalities:



Thm.
$\mathcal{S}^{\mathbf{MC}}_{\mathbf{r}} : \mathbf{roMC} \longrightarrow \mathbb{S}^{\mathbf{MC}}_{\mathbf{r}}$ is a traced symmetric monoidal functor, i.e. a homomorphism of TSMCs. □

Thus, for rightward open MCs,

$$\begin{aligned} \mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathcal{A} \,;\, \mathcal{B}) &= \mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathcal{A}) \,;\, \mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathcal{B}) \\ \mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathcal{A} \oplus \mathcal{B}) &= \mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathcal{A}) \oplus \mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathcal{B}) \\ \mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathrm{tr}(\mathcal{A})) &= \mathrm{tr}(\mathcal{S}^{\mathbf{MC}}_{\mathbf{r}}(\mathcal{A})) \end{aligned}$$



➔ **compositional model checking** of MCs!

Change of Base for Accommodating Actions, Schedulers, Optimality

[Eilenberg & Kelly '66] [Cruttwell, PhD thesis, '08] ...

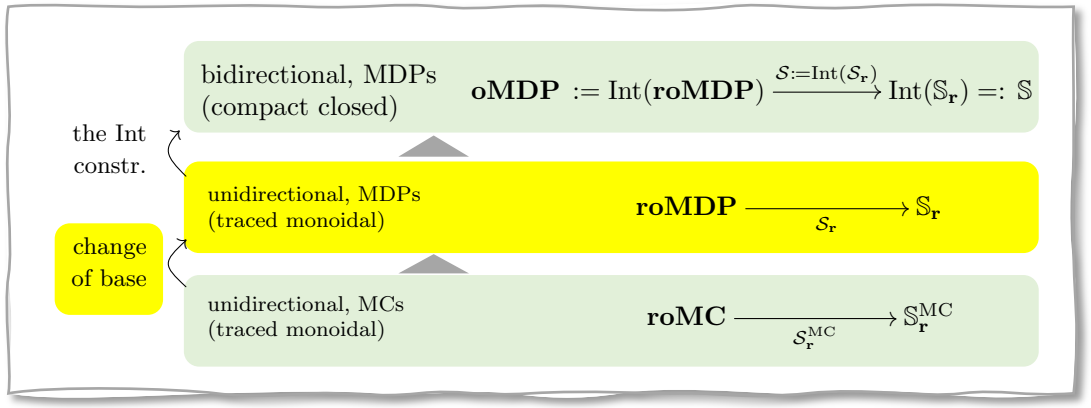
We apply change of base,
wrt. the powerset functor $\mathcal{P}\colon \mathbf{Set} \longrightarrow \mathbf{Set}$,
to *upgrade* $\mathbb{S}_{\mathbf{r}}^{\mathbf{MC}}$ (for MCs) to $\mathbb{S}_{\mathbf{r}}$ (for MDPs).

Concretely,

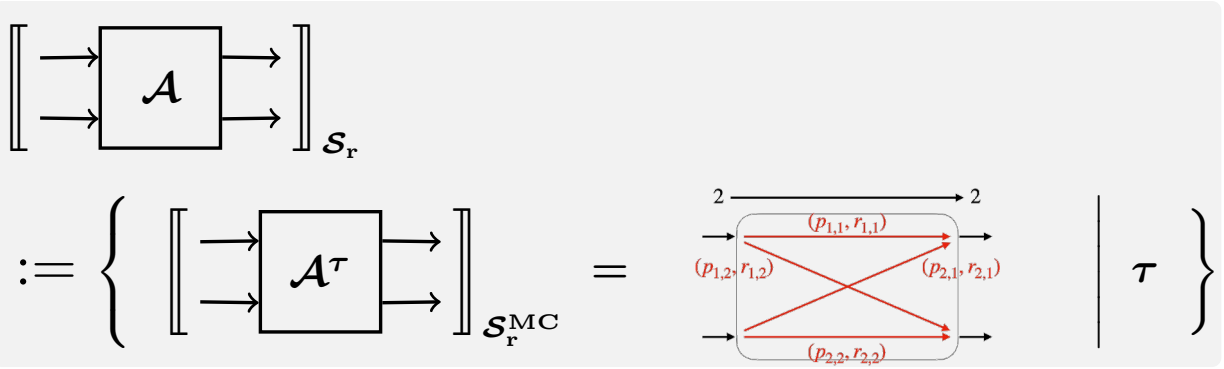
Def. $(\mathbb{S}_{\mathbf{r}})$
Object: natural number m
Arrow:

$$\frac{F\colon m \longrightarrow n \quad \text{in } \mathbb{S}_{\mathbf{r}}}{F \text{ is a set } \{f_i \mid f_i\colon m \to n \text{ in } \mathbb{S}_{\mathbf{r}}^{\mathbf{MC}}\}}$$

$\mathbb{S}_{\mathbf{r}}$ is a TSMC with pointwise extension of opr. of $\mathbb{S}_{\mathbf{r}}^{\mathbf{MC}}$.
E.g. $F \circ G = \{f_i \circ g_j\}_{i,j}$
(Being a category: by general theory of change of base.
Being traced monoidal: not covered, but easy.)



The solution functor $\mathcal{S}_{\mathbf{r}}\colon \mathbf{roMDP} \longrightarrow \mathbb{S}_{\mathbf{r}}$
is defined by bunding up different schedulers' behaviors

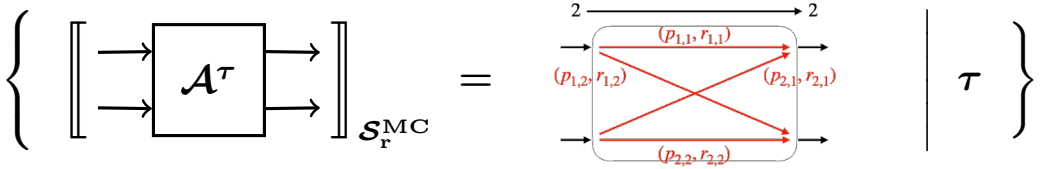
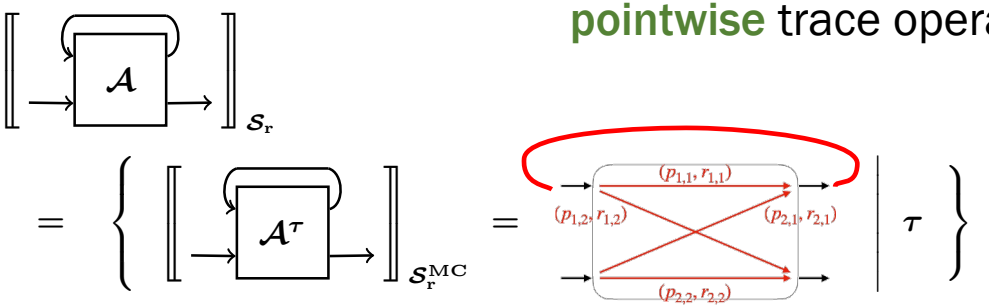
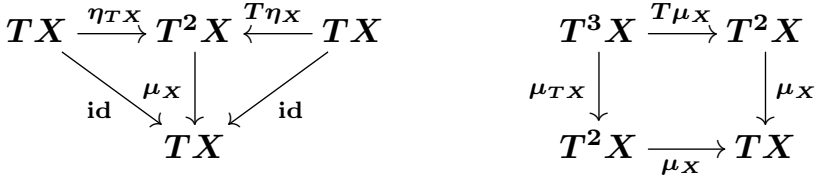
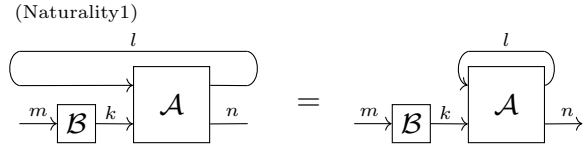


RPr and ERw,
under a scheduler τ

Thm.
$\mathcal{S}_{\mathbf{r}}\colon \mathbf{roMDP} \longrightarrow \mathbb{S}_{\mathbf{r}}$ is a traced symmetric monoidal functor, i.e. a homomorphism of TSMCs. □

➔ **compositional model checking** of MDPs!

Intermission: Probability and Nondeterminism?

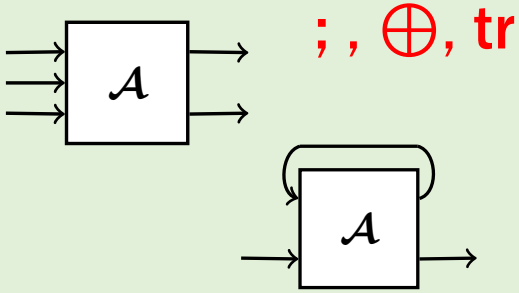
	coalgebraic/monadic modeling	monoidal/change-of-base modeling
semantics as...	(roughly) as a **coalgebra** $X \xrightarrow{c} \mathcal{P}\mathcal{D}X$	the behavior of **each scheduler** τ, bundled up 
repetition, multi-step behaviors	Kleisli composition...? $X \xrightarrow{c} \mathcal{P}\mathcal{D}X \xrightarrow{\mathcal{P}\mathcal{D}c} \mathcal{P}\mathcal{D}\mathcal{P}\mathcal{D}X \xrightarrow{\mu^{\mathcal{P}\mathcal{D}}} \mathcal{P}\mathcal{D}X$ ➜ **No multiplication μ!** [Plotkin] [Varacca & Winskel, MSCS'06] [Klin & Salamanca, MFPS'18] [Zwart & Marsden, LICS'19]	**pointwise** trace operator 
equational axioms	monad laws (**fail!**) 	TSMC axioms (**hold!**), e.g. 
schedulers	arbitrary	**memoryless** (= positional)

➜ Memorylessness has an obvious complexity advantage (#{schedulers} < ∞ even for tr($\mathcal{A}$)).
It may also resolve the famous semantical glitch (future work)

The Int Construction from Unidirectional to Bidirectional

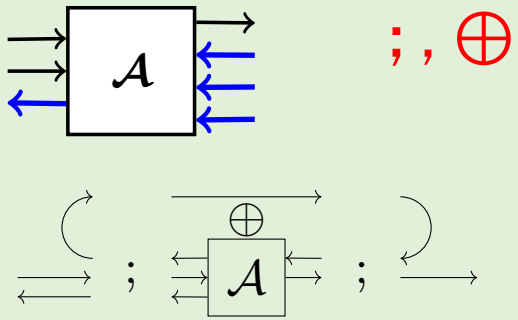
The *Int construction*: [Joyal, Street & Verity '96]
a general construction that turns

unidirectional string diagrams with loops



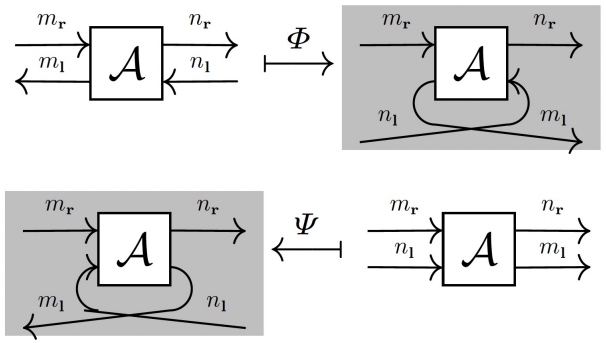
(traced sym. monoidal categories (TSMC))

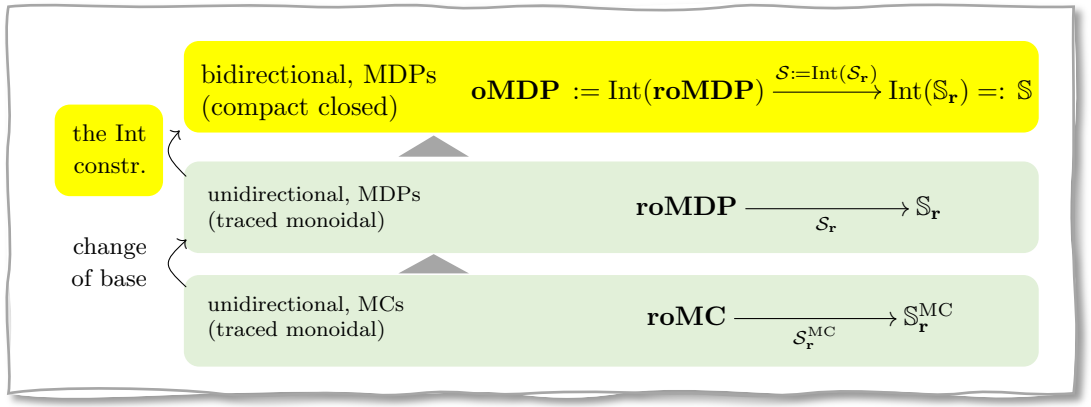
bidirectional string diagrams



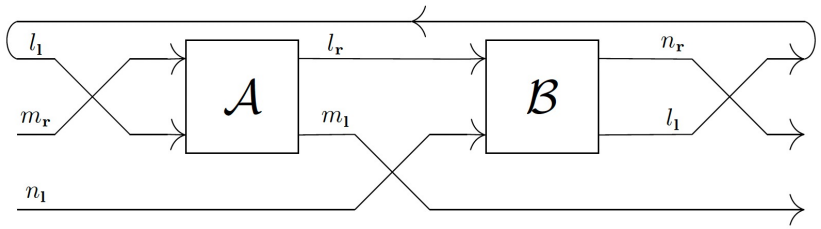
(compact closed categories (compCC))

by *twisting*:





In particular, the bidirectional seqComp $\mathcal{A}\,;\mathcal{B}$ is



Int extends to functors (and 2-cells):

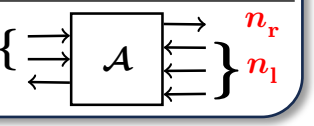
$$\mathbf{Int}\colon \mathbf{TSMC} \longrightarrow \mathbf{CompCC}$$

We thus apply $\mathbf{Int}$ to $\mathcal{S}_r\colon \mathbf{roMDP} \longrightarrow \mathbb{S}_r$ and get

$$\mathcal{S}\colon \mathbf{oMDP} \longrightarrow \mathbb{S}$$

Object: (m_r, m_l)
Arrow: $\mathcal{A} : (m_r, m_l) \longrightarrow (n_r, n_l)$ in $\mathbf{oMDP}$

an open MDP



This is a compact closed functor.

Hasuo (NII, Tokyo)

Framework Upgrades for Free: Compositional Solutions for MC, MDP, and bi-dir. MDP

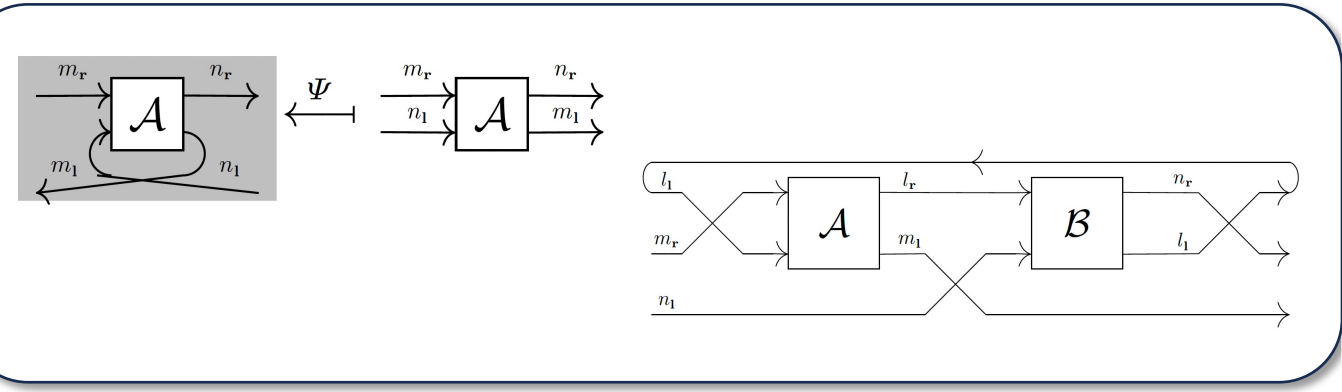
$$\mathbf{oMDP} \xrightarrow{\mathcal{S}} \mathbb{S}$$

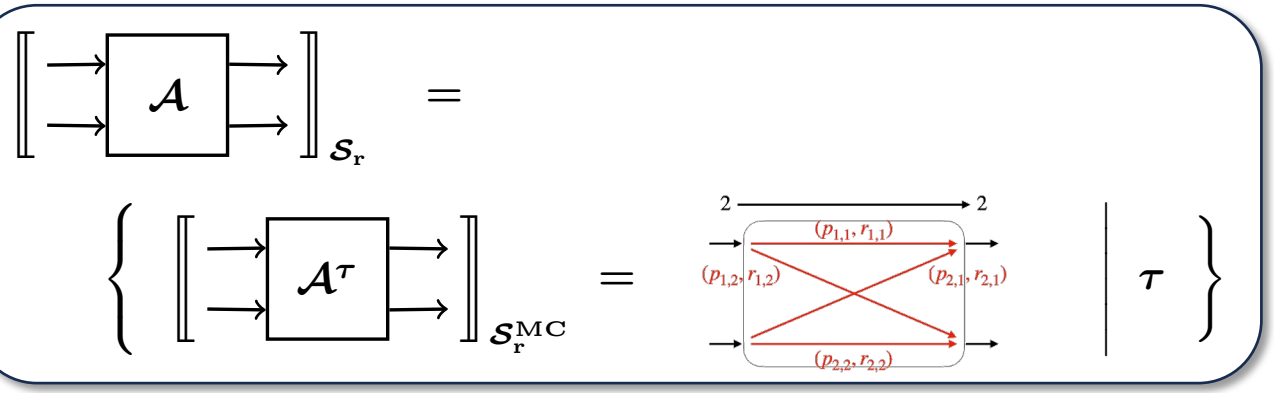
the Int construction

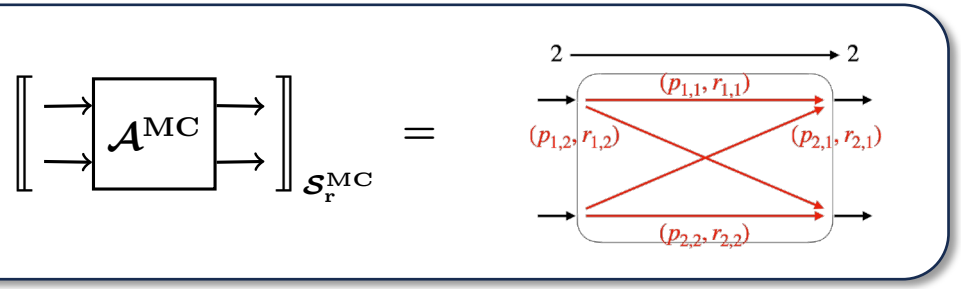
$$\mathbf{roMDP} \xrightarrow[\mathcal{S}_{\mathbf{r}}]{} \mathbb{S}_{\mathbf{r}}$$

change of base w/ $\mathcal{P}$

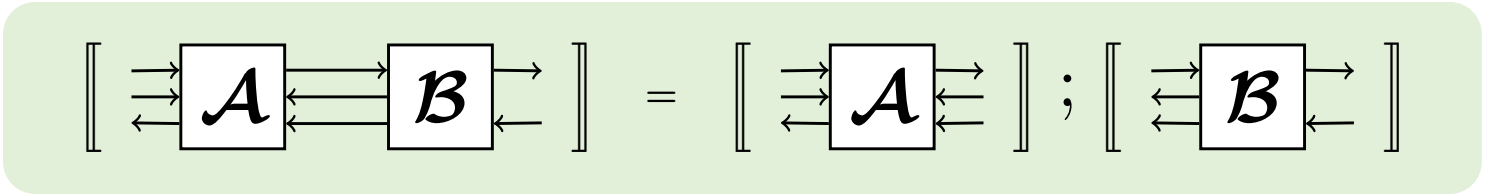
$$\mathbf{roMC} \xrightarrow[\mathcal{S}_{\mathbf{r}}^{\mathrm{MC}}]{} \mathbb{S}_{\mathbf{r}}^{\mathrm{MC}}$$







Outline



- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

CompMDP: a Compositional MDP Model Checking Algorithm

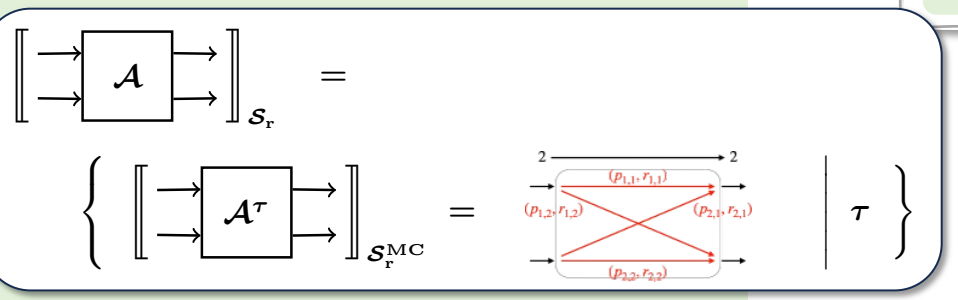
function CompMDP'($\mathcal{A}$)

Input:

a "string diagram" $\mathcal{A}$: $(m_\mathrm{r}, m_\mathrm{l}) \to (n_\mathrm{r}, n_\mathrm{l})$ of open MDPs, composed with ; (seqComp) and $\oplus$ (sum)

Output:

a set $\left\{ \left(p^{\tau}_{i,j}, r^{\tau}_{i,j}\right)_{i \in [m_\mathrm{r}+n_\mathrm{l}],\, j \in [n_\mathrm{r}+m_\mathrm{l}]} \right\}_{\tau}$



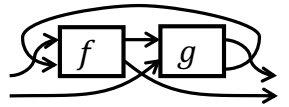
if $\mathcal{A}$ is atomic **then**

return $\left\{ \left(\mathbf{RPr}(\mathcal{A}^\tau)(i,j),\ \mathbf{ERw}(\mathcal{A}^\tau)(i,j) \right)_{i \in [m_\mathrm{r}+n_\mathrm{l}],\, j \in [n_\mathrm{r}+m_\mathrm{l}]} \,\middle|\, \begin{array}{l}\tau \text{ is a memoryless} \\ \text{scheduler of } \mathcal{A}\end{array} \right\}_{\tau}$

elsif $\mathcal{A} = \mathcal{B}\,;\,\mathcal{C}$ **then**

return CompMDP'($\mathcal{B}$) ; CompMDP'($\mathcal{C}$) (computed in $\mathbb{S}$)

concretely,

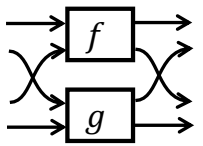


$f \in \mathrm{CompMDP}'(\mathcal{B}),\ g \in \mathrm{CompMDP}'(\mathcal{C})$

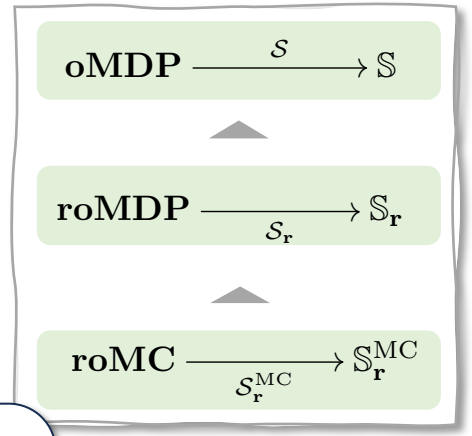
elsif $\mathcal{A} = \mathcal{B} \oplus \mathcal{C}$ **then**

return CompMDP'($\mathcal{B}$) $\oplus$ CompMDP'($\mathcal{C}$) (computed in $\mathbb{S}$)

concretely,



$f \in \mathrm{CompMDP}'(\mathcal{B}),\ g \in \mathrm{CompMDP}'(\mathcal{C})$



CompMDP: a Compositional MDP Model Checking Algorithm

function CompMDP'($\mathcal{A}$)

Input:

a "string diagram" $\mathcal{A}\colon (m_{\mathrm{r}}, m_{\mathrm{l}}) \to (n_{\mathrm{r}}, n_{\mathrm{l}})$ of open MDPs, composed with ; (seqComp) and $\oplus$ (sum)

Output:

a set $\left\{ \left(p^{\tau}_{i,j}, r^{\tau}_{i,j}\right)_{i \in [m_{\mathrm{r}}+n_{\mathrm{l}}],\ j \in [n_{\mathrm{r}}+m_{\mathrm{l}}]} \right\}_{\tau}$

if $\mathcal{A}$ is atomic **then**

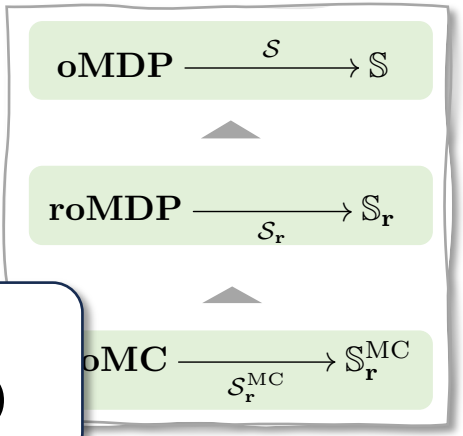
return $\left\{ \left(\mathbf{RPr}(\mathcal{A}^{\tau})(i, j) \right.$

elsif $\mathcal{A} = \mathcal{B}\,;\,\mathcal{C}$ **then**

return CompMDP'($\mathcal{B}$) ; C

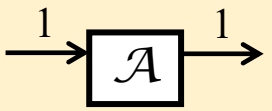
elsif $\mathcal{A} = \mathcal{B} \oplus \mathcal{C}$ **then**

return CompMDP'($\mathcal{B}$) $\oplus$



A (usual) MDP is identified with an open MDP $\mathcal{A}\colon (1, 0) \to (1, 0)$

function CompMDP($\mathcal{A}$)



Input:

an string diagram $\mathcal{A}\colon (1, 0) \to (1, 0)$ of open MDPs, composed with ; (seqComp) and $\oplus$ (sum)

Output:

the optimal expected reward of $\mathcal{A}$, identified with an (ordinary) MDP

$\left\{ \left(p^{\tau}, r^{\tau}\right) \right\}_{\tau} \leftarrow$ CompMDP'($\mathcal{A}$)

return $\max_{\tau} r^{\tau}$

CompMDP: a Compositional MDP Model Checking Algorithm

function CompMDP'($\mathcal{A}$)

Input:

a "string diagram" $\mathcal{A}$: $(m_\mathrm{r}, m_\mathrm{l}) \to (n_\mathrm{r}, n_\mathrm{l})$ of open MDPs, composed with ; (seqComp) and $\oplus$ (sum)

Output:

a set $\left\{ \left(p^\tau_{i,j}, r^\tau_{i,j}\right)_{i \in [m_\mathrm{r}+n_\mathrm{l}],\, j \in [n_\mathrm{r}+m_\mathrm{l}]} \right\}_\tau$

Meager semantics:
We can throw away τ that is totally subsumed:

$$p^\tau_{i,j} \le p^{\tau'}_{i,j} \quad \text{and} \quad r^\tau_{i,j} \le r^{\tau'}_{i,j}\,, \qquad \forall i, j$$

The actual implementation works bottom-up, to identify identical components

if $\mathcal{A}$ is atomic **then**

return $\left\{ \left(\mathbf{RPr}(\mathcal{A}^\tau)(i,j),\ \mathbf{ERw}(\mathcal{A}^\tau)(i,j) \right)_{i \in [m_\mathrm{r}+n_\mathrm{l}],\, j \in [n_\mathrm{r}+m_\mathrm{l}]} \,\middle|\, \begin{array}{l}\tau \text{ is a memoryless} \\ \text{scheduler of } \mathcal{A}\end{array} \right\}_\tau$

elsif $\mathcal{A} = \mathcal{B}\,;\,\mathcal{C}$ **then**

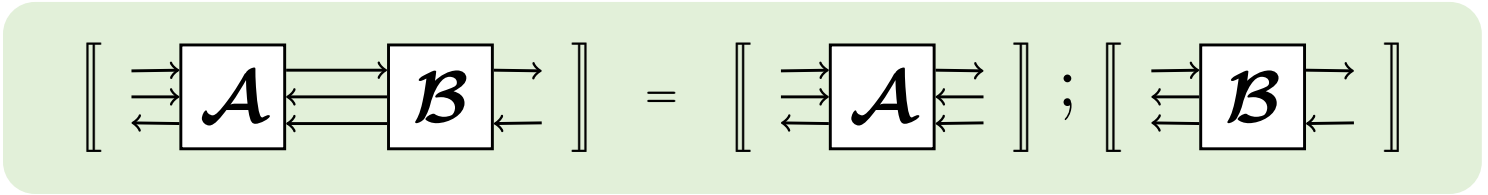
return CompMDP'($\mathcal{B}$) ; CompMDP'($\mathcal{C}$) (computed in $\mathbb{S}$)

$$\left\{ \left(p^\tau, r^\tau\right) \right\}_\tau$$

elsif $\mathcal{A} = \mathcal{B} \oplus \mathcal{C}$ **then**

return CompMDP'($\mathcal{B}$) $\oplus$ CompMDP'($\mathcal{C}$) (computed in $\mathbb{S}$)

Outline



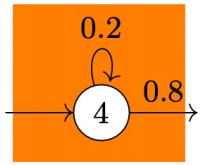
- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

Experiment Results

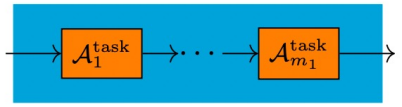
- Compositional algorithm can be **arbitrary faster** (reuse $\mathcal{S}(\mathcal{A})$!)

$$\mathcal{S}(\mathcal{A} \star \cdots \star \mathcal{A}) = \mathcal{S}(\mathcal{A}) \star \cdots \star \mathcal{S}(\mathcal{A})$$

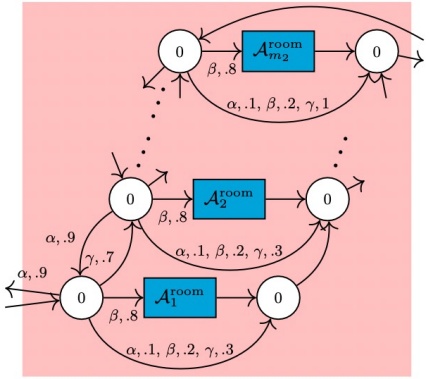
- Overall, we do indeed witness the performance advantage of compositionality
- We need MDPs given in a compositional formalism. This is realistic. Our *Patrol* benchmark:



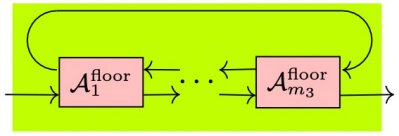
(a) A *task* $\mathcal{A}_i^{\text{task}}$.



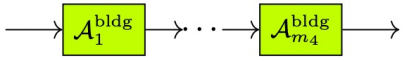
(b) A *room* $\mathcal{A}_i^{\text{room}}$ combines tasks.



(c) A *floor* $\mathcal{A}_i^{\text{floor}}$ combines rooms.



(d) A *building* $\mathcal{A}_i^{\text{bldg}}$ combines floors.



(e) A *neighborhood* $\mathcal{A}^{\text{nbd}}$ combines buildings.

benchmark	$\lvert Q\rvert$	$\lvert E\rvert$	exec. time [s] DI-high	DI-mid	DI-low
Patrol1	$\mathbf{10^8}$	$\mathbf{10^8}$	21	42	83
Patrol2	$\mathbf{10^8}$	$\mathbf{10^8}$	23	48	90
Patrol3	$\mathbf{10^9}$	$\mathbf{10^9}$	22	43	89
Patrol4	$\mathbf{10^9}$	$\mathbf{10^9}$	30	60	121
Wholesale1	$\mathbf{10^8}$	$\mathbf{2 \cdot 10^8}$	130	260	394
Wholesale2	$\mathbf{10^8}$	$\mathbf{2 \cdot 10^8}$	92	179	274
Wholesale3	$\mathbf{2 \cdot 10^8}$	$\mathbf{4 \cdot 10^8}$	6	12	23
Wholesale4	$\mathbf{2 \cdot 10^8}$	$\mathbf{4 \cdot 10^8}$	129	260	393

benchmark	$\lvert Q\rvert$	$\lvert E\rvert$	exec. time [s] FZ-none	FZ-int.	FZ-all (PRISM)
Packets1	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	65
Packets2	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	64
Packets3	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	56
Packets4	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	56
Patrol5	$\mathbf{10^8}$	$\mathbf{10^8}$	**22**	**22**	TO
Wholesale5	$\mathbf{5 \cdot 10^7}$	$\mathbf{10^8}$	TO	14	TO

$\lvert Q\rvert$ is the number of positions; $\lvert E\rvert$ is the number of transitions (only counting action branching, not probabilistic branching); execution time is the average of five runs, in sec.; timeout (TO) is 1200 sec.

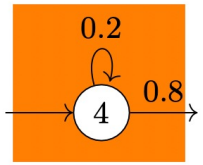
Apple MacBook Pro 2.3 GHz Dual-Core Intel Core i5 with 16GB of RAM

Experiment Results

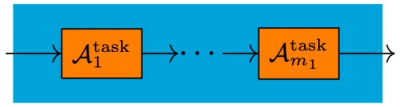
- Compositional algorithm can be **arbitrary faster** (reuse $\mathcal{S}(\mathcal{A})$!)

$$\mathcal{S}(\mathcal{A} \star \cdots \star \mathcal{A}) = \mathcal{S}(\mathcal{A}) \star \cdots \star \mathcal{S}(\mathcal{A})$$

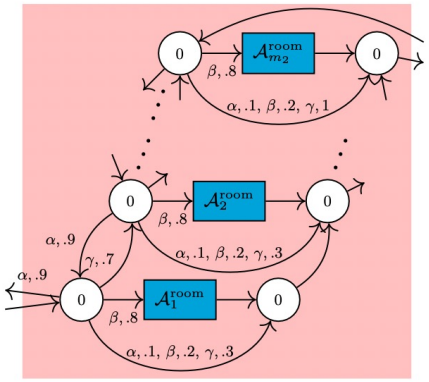
- Overall, we do indeed witness the performance advantage of compositionality
- We need MDPs given in a compositional formalism. This is realistic. Our *Patrol* benchmark:



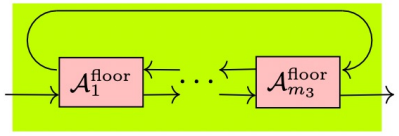
(a) A *task* $\mathcal{A}_i^{\text{task}}$.



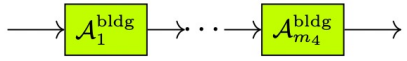
(b) A *room* $\mathcal{A}_i^{\text{room}}$ combines tasks.



(c) A *floor* $\mathcal{A}_i^{\text{floor}}$ combines rooms.



(d) A *building* $\mathcal{A}_i^{\text{bldg}}$ combines floors.



(e) A *neighborhood* $\mathcal{A}^{\text{nbd}}$ combines buildings.

DI (degree of identification): how much the same components are indeed recognized to be identical

			exec. time [s]		
benchmark	$\lvert Q\rvert$	$\lvert E\rvert$	DI-high	DI-mid	DI-low
Patrol1	$\mathbf{10^8}$	$\mathbf{10^8}$	21	42	83
Patrol2	$\mathbf{10^8}$	$\mathbf{10^8}$	23	48	90
Patrol3	$\mathbf{10^9}$	$\mathbf{10^9}$	22	43	89
Patrol4	$\mathbf{10^9}$	$\mathbf{10^9}$	30	60	121
Wholesale1	$\mathbf{10^8}$	$\mathbf{2 \cdot 10^8}$	130	260	394
Wholesale2	$\mathbf{10^8}$	$\mathbf{2 \cdot 10^8}$	92	179	274
Wholesale3	$\mathbf{2 \cdot 10^8}$	$\mathbf{4 \cdot 10^8}$	6	12	23
Wholesale4	$\mathbf{2 \cdot 10^8}$	$\mathbf{4 \cdot 10^8}$	129	260	393

performance improves

benchmark	$\lvert Q\rvert$	$\lvert E\rvert$	FZ-none	FZ-int.	FZ-all (PRISM)
Packets1	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	65
Packets2	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	64
Packets3	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	56
Packets4	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	56
Patrol5	$\mathbf{10^8}$	$\mathbf{10^8}$	**22**	**22**	TO
Wholesale5	$\mathbf{5 \cdot 10^7}$	$\mathbf{10^8}$	TO	14	TO

$\lvert Q\rvert$ is the number of positions; $\lvert E\rvert$ is the number of transitions (only counting action branching, not probabilistic branching); execution time is the average of five runs, in sec.; timeout (TO) is 1200 sec.

Apple MacBook Pro 2.3 GHz Dual-Core Intel Core i5 with 16GB of RAM

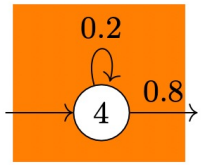
Experiment Results

Scalability:
big MDPs are model checked in realistic time

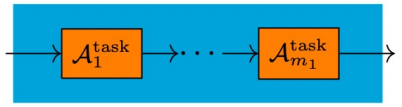
- Compositional algorithm can be **arbitrary faster** (reuse $\mathcal{S}(\mathcal{A})$!)

$$\mathcal{S}(\mathcal{A} \star \cdots \star \mathcal{A}) = \mathcal{S}(\mathcal{A}) \star \cdots \star \mathcal{S}(\mathcal{A})$$

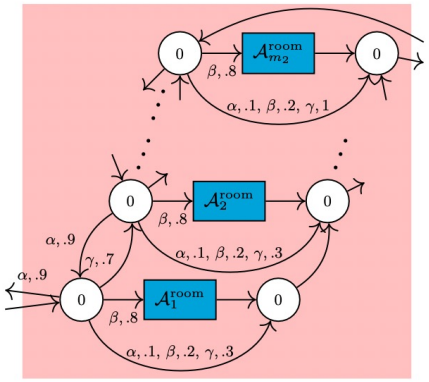
- Overall, we do indeed witness the performance advantage of compositionality
- We need MDPs given in a compositional formalism. This is realistic. Our *Patrol* benchmark:



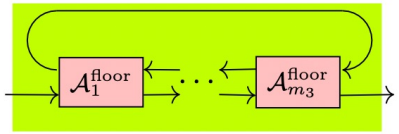
(a) A *task* $\mathcal{A}_i^{\text{task}}$.



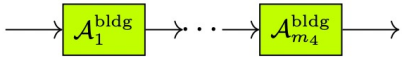
(b) A *room* $\mathcal{A}_i^{\text{room}}$ combines tasks.



(c) A *floor* $\mathcal{A}_i^{\text{floor}}$ combines rooms.



(d) A *building* $\mathcal{A}_i^{\text{bldg}}$ combines floors.



(e) A *neighborhood* $\mathcal{A}^{\text{nbd}}$ combines buildings.

benchmark	$\lvert Q\rvert$	$\lvert E\rvert$	exec. time [s] DI-high	DI-mid	DI-low
Patrol1	$\mathbf{10^8}$	$\mathbf{10^8}$	21	42	83
Patrol2	$\mathbf{10^8}$	$\mathbf{10^8}$	23	48	90
Patrol3	$\mathbf{10^9}$	$\mathbf{10^9}$	22	43	89
Patrol4	$\mathbf{10^9}$	$\mathbf{10^9}$	30	60	121
Wholesale1	$\mathbf{10^8}$	$\mathbf{2 \cdot 10^8}$	130	260	394
Wholesale2	$\mathbf{10^8}$	$\mathbf{2 \cdot 10^8}$	92	179	274
Wholesale3	$\mathbf{2 \cdot 10^8}$	$\mathbf{4 \cdot 10^8}$	6	12	23
Wholesale4	$\mathbf{2 \cdot 10^8}$	$\mathbf{4 \cdot 10^8}$	129	260	393

benchmark	$\lvert Q\rvert$	$\lvert E\rvert$	exec. time [s] FZ-none	FZ-int.	FZ-all (PRISM)
Packets1	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	65
Packets2	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	64
Packets3	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	56
Packets4	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	56
Patrol5	$\mathbf{10^8}$	$\mathbf{10^8}$	**22**	**22**	TO
Wholesale5	$\mathbf{5 \cdot 10^7}$	$\mathbf{10^8}$	TO	14	TO

$\lvert Q\rvert$ is the number of positions; $\lvert E\rvert$ is the number of transitions (only counting action branching, not probabilistic branching); execution time is the average of five runs, in sec.; timeout (TO) is 1200 sec.

Apple MacBook Pro 2.3 GHz Dual-Core Intel Core i5 with 16GB of RAM

Experiment Results

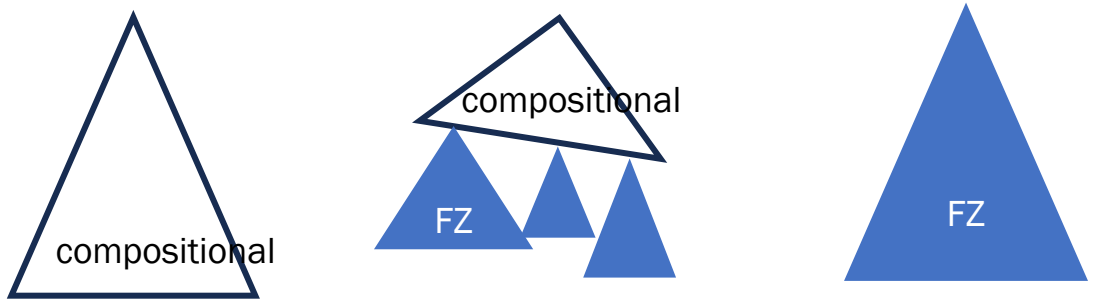
- Compositional algorithm can be **arbitrary faster** (reuse $\mathcal{S}(\mathcal{A})$!)

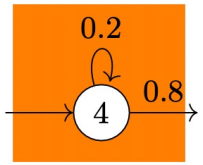
$$\mathcal{S}(\mathcal{A} \star \cdots \star \mathcal{A}) = \mathcal{S}(\mathcal{A}) \star \cdots \star \mathcal{S}(\mathcal{A})$$

- Overall, we do indeed witness the performance advantage of compositiona
- We need MDPs given in a compositional form
  This is realistic. Our *Patrol* benchmark:

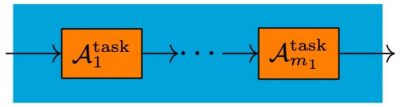
FZ (freezing):
We can stop doing compositionally at a certain depth

FZ-none **FZ-intermediate** **FZ-all** (we used PRISM)

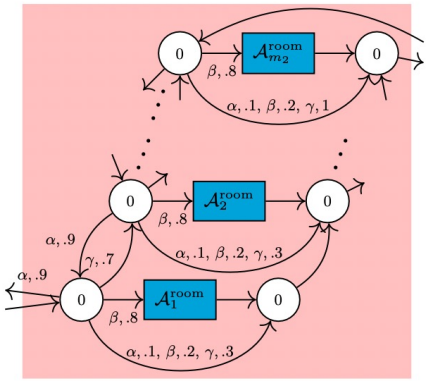




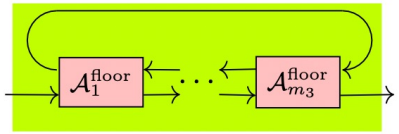
(a) A *task* $\mathcal{A}_i^{\text{task}}$.



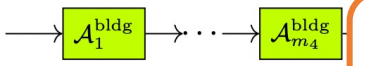
(b) A *room* $\mathcal{A}_i^{\text{room}}$ combines tasks.



(c) A *floor* $\mathcal{A}_i^{\text{floor}}$ combines rooms.



(d) A *building* $\mathcal{A}_i^{\text{bldg}}$ combines floors.



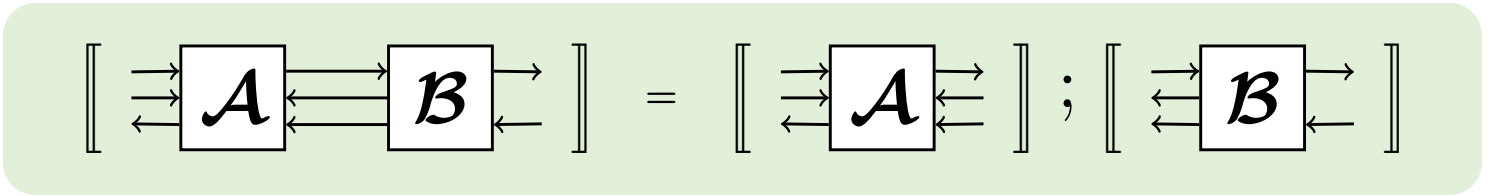
(e) A *neighborhood* $\mathcal{A}^{\text{nbd}}$ c
buildings.

benchmark	$\lvert Q\rvert$	$\lvert E\rvert$	exec. time [s] FZ-none	FZ-int.	FZ-all (PRISM)
Packets1	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	65
Packets2	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	64
Packets3	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	1	56
Packets4	$\mathbf{2.5 \cdot 10^5}$	$\mathbf{5 \cdot 10^5}$	TO	3	56
Patrol5	$\mathbf{10^8}$	$\mathbf{10^8}$	**22**	**22**	TO
Wholesale5	$\mathbf{5 \cdot 10^7}$	$\mathbf{10^8}$	TO	14	TO

$\lvert Q\rvert$ is the number of positions; $\lvert E\rvert$ is the number of transitions (only counting
average of

- Compositionality helps
- But going all the way down may not be a good idea

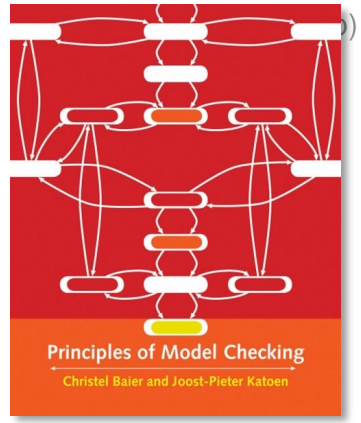
Outline



- Prelude: some general thoughts on compositionality
- Target problem: optimal expected reward of MDPs
- Composition formalism: string diagrams of MDPs
- Monoidal categories and their benefits
  - Settle granularity of semantics
  - Framework upgrades for free
- Semantical contstructions (= deriving algorithms)
- Algorithm
- Experimental evaluation
- Conclusions

Related Work (Compositional Probabilistic MC)

Baier & Katoen, MIT Press 2008



- **Probabilistic** model checking is an active field (Baier, Larsen, Katoen, Kwiatkowska, Parker, Raskin, …)
- **Compositionality** in model checking is an old problem [Clarke, Long & McMillan, LICS'89] [Tsukada & Ong, LICS'14] …
- Two closely related works on **compositional probabilistic** model checking:

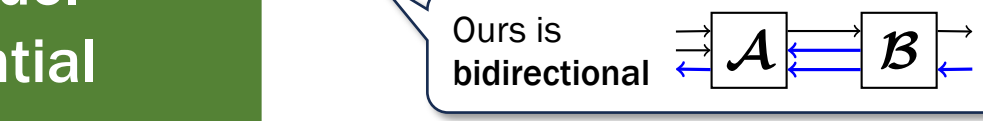
Probabilistic Model Checking wrt. Parallel Composition ||
[Kwiatkowska, Norman, Parker & Qu, Inf. Comp. '13]

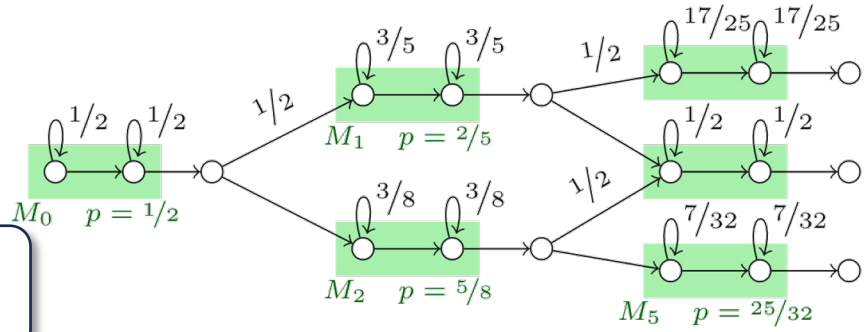
- Compositional model checking of **parallel composition** $\mathcal{A} \parallel \mathcal{B}$
- … but **assume-guarantee "contracts"** betw. $\mathcal{A}$ and $\mathcal{B}$ must be devised
- (more on this later)

Parametric MDP Model Checking for Sequential Composition
[Junges & Spaan, CAV'22]

- Sequential composition of **parametric MDPs**
- **Unidirectional** composition

Ours is **bidirectional**





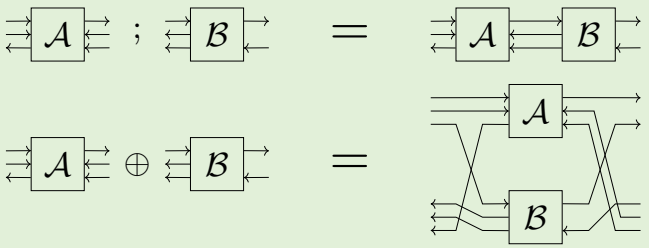
- Assumption: locally optimal schedulers are globally optimal, too
  (It holds if component exits are unique. We don't need this assumption)
- Compositional solution of **parametric** components $\mathcal{A}(p)$
  (We don't do this)

Related Work (String Diagrams and MC)

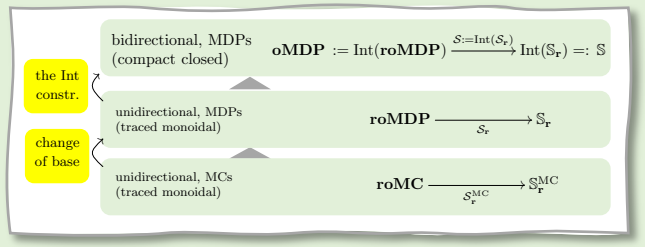
Our general methodology **goes beyond MDPs** and applies to **general graph-like formalisms**

"Our general methodology":

- Composition by **string diagrams**



- Semantic domains as **compact closed categories** (**"granularity of semantics"**)
- **Framework upgrades for free**



Successful applications to

- parity games [Watanabe, Eberhart, Asada & Hasuo, MFPS'21]
- mean-payoff games
  [Watanabe, Eberhart, Asada & Hasuo, submitted]

We have drawn inspirations from
higher-order model checking (HOMC)

- **Given** a higher-order program generating a tree T
- **Answer** if the tree T is accepted by a parity tree automaton $\mathcal{A}$

References

- Decidability [Ong, LICS'06]
- Type system and an efficient algorithm
  [Kobayashi & Ong, LICS'09]

$$\frac{}{\Gamma \cup \{x:\theta\} \vdash_{\mathcal{B}} x:\theta}\ (\text{T-Var})$$

$$\frac{\delta_{\mathcal{B}}(q,a) = q_1 \cdots q_n}{\Gamma \vdash_{\mathcal{B}} a : q_1 \to \cdots \to q_n \to q}\ (\text{T-Const})$$

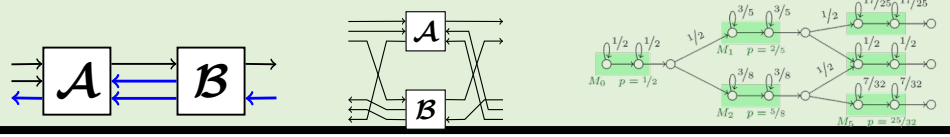
$$\frac{\Gamma \vdash_{\mathcal{B}} t_1 : \bigwedge_{i\in\{1,\ldots,n\}} \theta_i \to \theta \qquad \Gamma \vdash_{\mathcal{B}} t_2 : \theta_i \text{ (for each } i \in \{1,\ldots,n\})}{\Gamma \vdash_{\mathcal{B}} t_1 t_2 : \theta}\ (\text{T-App})$$

$$\frac{\Gamma \cup \{x:\theta_1,\ldots,x:\theta_n\} \vdash_{\mathcal{B}} t:\theta \qquad x \notin dom(\Gamma)}{\Gamma \vdash_{\mathcal{B}} \lambda x.t : \bigwedge_{i\in\{1,\ldots,n\}} \theta_i \to \theta}\ (\text{T-Abs})$$

- Type system = compositionality??
  [Tsukada & Ong, LICS'14]
- Categorical formalization
  [Grellois & Mellies, MFCS'15]
  → Main inspiration for our work
  [Watanabe, Eberhart, Asada & Hasuo, MFPS'21]

Sequential/Planar Composition vs Parallel Composition

Very different in nature, very different in difficulty!
Categorical treatment of parallel composition? (future work)

	sequential/planar composition (uni- and bi-directional) 	**parallel composition** $\mathcal{A} \parallel \mathcal{B}$ e.g. [Kwiatkowska, Norman, Parker & Qu, Inf. Comp. '13]
frequency of interaction	sparse	dense, in general
mode of interaction	by **resumptions** → **fixed**, once the type of systems is fixed (MCs, MDPs, parity games, mean-payoff games, ...)	by some synchronization mechanism → **variant** (common actions? handshaking? shared memory? channels?)
reasoning about interaction	by **pre- and post-conditions**, as in Hoare logic $\{P\}\,\alpha\,\{Q\}$	by **assume-guarantee contracts**, not easy to come up with
difficulty	easier	harder

My Coauthors

Kazuki Watanabe
PhD Student
NII & SOKENDAI, Tokyo

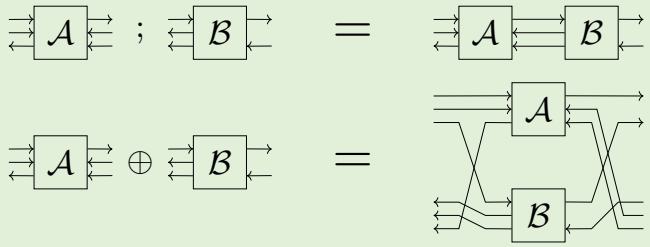
Clovis Eberhart
Project Assistant Professor
NII, Tokyo

Kazuyuki Asada
Assistant Professor
Tohoku University, Sendai

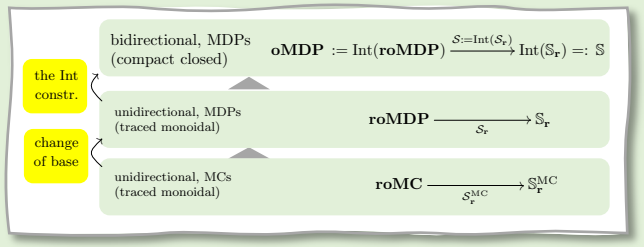
Monoidal Categories Guiding Planer-Compositional Model Checking

“Our general methodology”:

- Composition by **string diagrams**

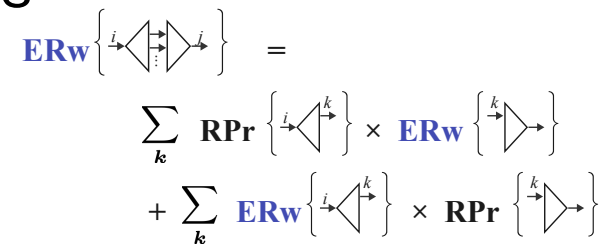


- Semantic domains as **compact closed categories** (“**granularity of semantics**”)
- **Framework upgrades for free**



We applied it to **MDP model checking**

- Semantic categories by decomposition equalities
- a **compositional algorithm** with clear performance advantage



Future work

- probability & nondeterminism, being memoryless
- parallel composition
- other “model checking” problems
- cyber-physical & robotics application