A Theory of Merge-and-Shrink for Stochastic Shortest Path Problems

Thorsten Klößner[1], **Álvaro Torralba**[2], **Marcel Steinmetz**[1], **Silvan Sievers**[3]

[1]Saarland University, Germany
[2]Aalborg University, Denmark
[3]Basel University, Switzerland
{kloessner, steinmetz}@cs.uni-saarland.de, alto@cs.aau.dk, silvan.sievers@unibas.ch

Abstract

The merge-and-shrink framework is a powerful tool to construct state space abstractions based on factored representations. One of its core applications in classical planning is the construction of admissible abstraction heuristics. In this paper, we develop a compositional theory of merge-and-shrink in the context of probabilistic planning, focusing on stochastic shortest path problems (SSPs). As the basis for this development, we contribute a novel factored state space model for SSPs. We show how general transformations, including abstractions, can be formulated on this model to derive admissible and/or perfect heuristics. To formalize the merge-and-shrink framework for SSPs, we transfer the fundamental merge-and-shrink transformations from the classical setting: shrinking, merging, and label reduction. We analyze the formal properties of these transformations in detail and show how the conditions under which shrinking and label reduction lead to perfect heuristics can be extended to the SSP setting.

Introduction

Probabilistic planning is a generalization of classical planning in which actions are stochastic, i.e., an action may result in several outcomes, each known to occur with a specific probability. Stochastic shortest path problems (SSPs, Bertsekas and Tsitsiklis 1991) are particular probabilistic planning problems in which the objective is the minimization of accumulated costs in expectation until a specific goal is reached. Heuristic search algorithms (Hansen and Zilberstein 2001; Bonet and Geffner 2003; Trevizan et al. 2017) can solve SSPs without constructing the full state space by utilizing an *admissible* heuristic, i. e., a function that underestimates the expected cost-to-goal of all states. Previous work on admissible heuristics include determinization-based approaches (e. g., Bonet and Geffner 2005; Kolobov, Mausam, and Weld 2010, 2012), which evaluate classical planning heuristics on the all-outcomes determinization of the problem; occupation measure heuristics (Trevizan, Thiébaux, and Haslum 2017), which can be seen as a generalization of operator-counting heuristics from classical planning (Pommerening et al. 2014); and pattern database heuristics for SSPs (Klößner and Hoffmann 2021).

Merge-and-shrink (Dräger, Finkbeiner, and Podelski 2009) is a powerful framework based on *factored* state space representations with a broad range of applications in classical planning, including the computation of abstraction heuristics (e. g., Helmert et al. 2014; Sievers and Helmert 2021), task reformulation (Torralba and Sievers 2019) or identifying irrelevant actions (Torralba and Kissmann 2015). The core concept of the merge-and-shrink framework is to represent a (large) labelled transition system implicitly as a factored transition system (FTS), a set of multiple smaller transition systems (called *factors*) with the same labels. Given such an FTS of an input transition system, the framework repeatedly applies *transformations*, while at the same time maintaining a factored state mapping (Helmert, Röger, and Sievers 2015) from the original transition system to the transformed FTS. All transformations can be categorized in terms of desirable properties that they have. For example, conservative transformations induce admissible heuristics and exact transformations induce perfect heuristics. Furthermore, transformations can be chained, and such *composed* transformations inherit the properties of the individual transformations. As all merge-and-shrink transformations are at least conservative, each factor of the transformed FTS induces an abstraction of the original transition system, which empowers computing admissible abstraction heuristics.

While factored state space representations also exist for MDPs (e. g., Boutilier, Dearden, and Goldszmidt 2000), the merge-and-shrink framework in particular has not been applied to this setting as of yet. Our main contribution is the extension of this framework as detailed by Sievers and Helmert (2021) to the SSP setting. We design a new factored state space representation suitable for our new framework and show how transformations and their properties must be adapted to this representation to operate as expected. We show that an exact reconstruction of the original state space is impossible if factors are represented as probabilistic transition systems. Instead, we introduce a new factored representation amenable for exact transformations by annotating each transition with the task-level probabilistic outcome it belongs to. Then, we define shrink, merge, and label reduction transformations within the new framework and analyze their properties, including conditions which guarantee exactness of such transformations. In particular, we show that shrink transformations based on probabilistic bisimula-

tion (Larsen and Skou 1991) are not exact, and subsequently adapt them to yield exactness. Our theory allows to fully exploit the power of merge-and-shrink for SSPs, including the computation of admissible heuristics, thus providing a method to compute arbitrary abstraction heuristics for SSPs.

Background

In this section, we introduce our probabilistic planning setting and recall the merge-and-shrink framework.

Mathematical Notation

We write $\mathcal{D}(X) := \{\delta : X \to [0,1] \mid \sum_{x \in X} \delta(x) = 1\}$ for the set of (discrete) probability distributions over X and $\mathrm{supp}(\delta) := \{x \mid \delta(x) > 0\}$ for the *support* of $\delta \in \mathcal{D}(X)$. We use $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, \infty\}$ for the *extended reals*, with the convention $0 \cdot \pm\infty = 0$. We write id_X for the *identity function* on X, using id if X it is clear from the context.

A partial function f from a set X to Y is a function from some domain $\mathrm{dom}(f) \subseteq X$ to Y and is denoted $f : X \rightharpoonup Y$. If $\mathrm{dom}(f) = X$ then f is total on X and we write $f : X \to Y$. The image of $X' \subseteq X$ under f is denoted $f(X') := \{f(x) \mid x \in X' \cap \mathrm{dom}(f)\}$ and the pre-image of $Y' \subseteq Y$ under f is denoted $f^{-1}(Y') := \{x \mid f(x) \in Y'\}$.

The restriction of f to $X \subseteq \mathrm{dom}(f)$ is denoted $f|_X$. The composition of $f : X \rightharpoonup Y$ and $g : Y \rightharpoonup Z$ is denoted $g \circ f : \mathrm{dom}(f) \cap f^{-1}(\mathrm{dom}(g)) \to Z, x \mapsto g(f(x))$. The disjoint union of $f : X \to Y$ and $g : X' \to Y'$ is denoted $f \cup g : X \cup X' \to Y \cup Y', x \mapsto f(x)$ if $x \in X$ else $g(x)$.

Probabilistic Planning

Stochastic Shortest Path Problems A *stochastic shortest path problem* (SSP, Bertsekas and Tsitsiklis 1991) is a tuple $\mathcal{S} = \langle S, L, c, P, G \rangle$, where S is a set of *states*, L is a set of *labels*, $c : L \to \mathbb{R}_{\geq 0}$ is a *label cost function*, $P : S \times L \times S \to [0,1]$ is a *transition probability function* with $\sum_{t \in S} P(s, \ell, t) \in \{0, 1\}$ for all $\langle s, \ell \rangle \in S \times L$, and $G \subseteq S$ is a set of *goal states*. The set of labels applicable in $s \in S$ is denoted $L(s) := \{\ell \in L \mid \sum_{t \in S} P(s, \ell, t) = 1\}$.

A *policy* for $\mathcal{S}$ is a function $\pi : S \rightharpoonup L$ either selecting a label $\pi(s) \in L(s)$ when s is encountered during execution resulting in the successor t with probability $P(s, \pi(s), t)$, or deciding to terminate if $s \notin \mathrm{dom}(\pi)$. An *$s$-proper* policy for $s \in S$ terminates in a goal state with probability 1 when executed from s. We write $\Pi(s)$ for the set of s-proper policies.

The *expected cost-to-goal function* $J^\pi_{\mathcal{S}}$ for π is defined as $J^\pi_{\mathcal{S}}(s) := \mathbb{E}_{\pi,s}[\sum_{i=0}^{T} c(\pi(S_i))]$, where the random variable S_i models the state after i decisions of π when starting in s and $T := \inf\{i \mid S_{i+1} \notin \mathrm{dom}(\pi)\}$ is the last execution step before termination. If $T = \infty$, the sum is a series.

The *optimal expected cost-to-goal function* $J^*_{\mathcal{S}}$ is defined by $J^*_{\mathcal{S}}(s) := \inf_{\pi \in \Pi(s)} J^\pi_{\mathcal{S}}(s)$, where ∞ signifies that dead ends are unavoidable from s. A policy π is *optimal* for $s \in S$ iff $J^\pi_{\mathcal{S}}(s) = J^*_{\mathcal{S}}(s)$. For $s \in S$, $J^*_{\mathcal{S}}(s)$ is characterizable by a linear program. To this end, let the Bellman (1957) optimality operator $\mathcal{B}_{\mathcal{S}} : (S \to \overline{\mathbb{R}}_{\geq 0}) \to (S \to \overline{\mathbb{R}}_{\geq 0})$ be defined by $(\mathcal{B}_{\mathcal{S}} J)(s) := \min_{\ell \in L(s)} [c(\ell) + \sum_{t \in S} P(s, \ell, t) J(t)]$ if $L(s) \neq \emptyset$, else $(\mathcal{B}_{\mathcal{S}} J)(s) := J(s)$. It holds that $J^*_{\mathcal{S}}(s) = \sup_{J : S \to \mathbb{R}_{\geq 0}} \{J(s) \mid J \leq \mathcal{B}_{\mathcal{S}} J \wedge \forall s \in G.\, J(s) = 0\}$.

SSP Heuristic Search Given an SSP $\mathcal{S} = \langle S, L, c, P, G \rangle$, a *heuristic* $h : S \to \overline{\mathbb{R}}_{\geq 0}$ provides estimates for $J^*_{\mathcal{S}}(s)$ for all states $s \in S$. h is *admissible* iff $h \leq J^*_{\mathcal{S}}$, *pessimistic* iff $h \geq J^*_{\mathcal{S}}$ and *perfect* iff $h = J^*_{\mathcal{S}}$. Furthermore, h is *goal-aware* iff $h(s) = 0$ for all $s \in G$, *consistent* iff $h \leq \mathcal{B}_{\mathcal{S}} h$ and *safe* iff $J^*_{\mathcal{S}}(s) \in \mathbb{R}_{\geq 0}$ implies $h(s) \in \mathbb{R}_{\geq 0}$. A goal-aware and consistent heuristic h is not necessarily admissible if ∞ is allowed as a heuristic value, unlike in classical planning. However, if h is additionally safe, then admissibility follows from the linear programming formulation of J^*.

A wide range of SSP heuristic search algorithms exists (e. g., Bonet and Geffner 2003; Trevizan et al. 2017). For all intents and purposes of this work, it suffices to know that these algorithms require an admissible heuristic and compute an optimal policy for a given initial state, if one exists.

Variable Spaces A *variable space* is a finite set of state variables $\mathcal{V}$ associated with non-empty sets $(\mathcal{D}(v))_{v \in \mathcal{V}}$, called the *variable domains*. An assignment is an element of $\mathcal{A}(\mathcal{V}) := \bigtimes_{v \in \mathcal{V}} \mathcal{D}(v)$. A partial assignment is an element of $\mathcal{A}^p(\mathcal{V}) := \bigcup_{W \subseteq \mathcal{V}} \bigtimes_{v \in W} \mathcal{D}(v)$. For $s, t \in \mathcal{A}^p(\mathcal{V})$, we say that s is *consistent* with t, written $s \models t$, iff $\mathrm{dom}(s) \supseteq \mathrm{dom}(t)$ and $s(v) = t(v)$ for all $v \in \mathrm{dom}(t)$. We denote with $s[\![t]\!] := s|_{\mathrm{dom}(s) \setminus \mathrm{dom}(t)} \cup t$ the *update* of s with t.

Probabilistic Planning Tasks As the top-level model of the planning problem, we consider *probabilistic planning tasks* (PPTs) with stochastic effects in finite-domain representation (Trevizan, Thiébaux, and Haslum 2017). A PPT is a tuple $\Pi = \langle \mathcal{V}, O, c, s_0, s_\star \rangle$. $\mathcal{V}$ is a variable space. O is the finite set of *operators*. Each operator $o \in O$ is associated with a *precondition* $\mathrm{pre}(o) \in \mathcal{A}^p(\mathcal{V})$, and an *effect probability distribution* $\Pr_o \in \mathcal{D}(\mathcal{A}^p(\mathcal{V}))$. $c : O \to \mathbb{R}_{\geq 0}$ is the *operator cost function*. Finally, $s_0 \in \mathcal{A}(\mathcal{V})$ is the *initial state* and $s_\star \in \mathcal{A}^p(\mathcal{V})$ is the *goal*. Π induces the SSP $\mathcal{S}(\Pi) := \langle \mathcal{A}(\mathcal{V}), O, c, P, \{s \mid s \models s_\star\} \rangle$ where $P(s, o, t) := \sum_{e \in \mathcal{A}^p(\mathcal{V}).\, s[\![e]\!] = t} \Pr_o(e)$ if $s \models \mathrm{pre}(o)$, else $P(s, o, t) := 0$. The objective is to find an optimal policy for s_0 in $\mathcal{S}(\Pi)$.

Merge-and-Shrink

We describe *merge-and-shrink* in terms of the compositional theory by Sievers and Helmert (2021), from which we adopt notation. A (labelled) transition system (TS) is a tuple $\Theta = \langle S, L, c, T, G \rangle$ where $T \subseteq S \times L \times S$ is a *transition relation* and all other components are the same as for SSPs. A *factored transition system* (FTS) $F = (\Theta_i)_{i \in I}$ is a family of transition systems (called *factors*) $\Theta_i = \langle S_i, L, c, T_i, G_i \rangle$ with common labels and label cost function. F implicitly represents the *synchronized product* $\bigotimes F := \langle \bigtimes_{i \in I} S_i, L, c, T_\otimes, \bigtimes_{i \in I} G_i \rangle$ over all of its factors, where $T_\otimes := \{\langle s, \ell, t \rangle \mid \forall i \in I.\, \langle s_i, \ell, t_i \rangle \in T_i\}$.

The merge-and-shrink algorithm is based on *transformations* of the represented TS. A transformation of Θ into Θ' is a tuple $\langle \Theta, \Theta', \sigma, \lambda \rangle$ where $\Theta = \langle S, L, c, T, G \rangle$ is the *original TS*, $\Theta' = \langle S', L', c', T', G' \rangle$ is the *transformed TS* and $\sigma : S \rightharpoonup S'$ and $\lambda : L \rightharpoonup L'$ are state and label mappings that relate the states and labels of both TS.

Transformations on the implicitly represented TS of the FTS are themselves carried out implicitly via *factored transformations*, which are based on *factored mappings* (FMs)

which serve as state mappings between FTS. An FM from a variable space $\mathcal{V}$ to a set of values D is an inductively defined tree-like data structure. For $v \in \mathcal{V}$ and a partial function $\alpha : \mathcal{D}(v) \rightharpoonup D$, $\mathrm{Atom}(v, \alpha)$ is an *atomic FM*. For FMs σ_L from $\mathcal{V}_L$ to D_L and σ_R from $\mathcal{V}_R$ to D_R and a mapping $\alpha : D_L \times D_R \to D$, $\mathrm{Merge}(\sigma_L, \sigma_R, \alpha)$ is a *merge FM*. Each FM σ from $\mathcal{V}$ to D represents a partial function $[\![\sigma]\!] : \mathcal{A}(\mathcal{V}) \rightharpoonup D$. For atomic FMs, $[\![\mathrm{Atom}(v, \alpha)]\!](s) := \alpha(s(v))$ and for merge FMs, $[\![\mathrm{Merge}(\sigma_L, \sigma_R, \alpha)]\!](s) := \alpha([\![\sigma_L]\!](s), [\![\sigma_R]\!](s))$. For a variable $v \in \mathcal{V}$, a *projection FM* is defined as $\pi_v := \mathrm{Atom}(v, \mathrm{id}_{\mathcal{D}(v)})$ with $[\![\pi_v]\!](s) = s(v)$.

A *factored-to-factored mapping* (F2FM) from a variable space $\mathcal{V}$ to another variable space $\mathcal{W}$ is a family $\Sigma = (\sigma_w)_{w \in \mathcal{W}}$ where each $\sigma_w, w \in \mathcal{W}$ is an FM from $\mathcal{V}$ to $\mathcal{D}(w)$ which represents the function $[\![\Sigma]\!] : \mathcal{A}(\mathcal{V}) \rightharpoonup \mathcal{A}(\mathcal{W}), s \mapsto ([\![\sigma_w]\!](s))_{w \in \mathcal{W}}$. For every two F2FMs Σ from $\mathcal{V}$ to $\mathcal{W}$ and Γ from $\mathcal{W}$ to $\mathcal{U}$, there is a composed F2FM $\Gamma \circ \Sigma$ from $\mathcal{V}$ to $\mathcal{U}$ with $[\![\Gamma \circ \Sigma]\!] = [\![\Gamma]\!] \circ [\![\Sigma]\!]$.

A factored transformation is a tuple $\langle F, F', \Sigma, \lambda \rangle$ where F is the original FTS, F' is the transformed FTS, Σ is an F2FM from F to F', where the variable domain of a factor is its state set, and λ is the label mapping. It naturally induces the transformation $\langle \bigotimes F, \bigotimes F', [\![\Sigma]\!], \lambda \rangle$.

For the purpose of computing admissible heuristics for a given TS Θ, the merge-and-shrink algorithm initially computes a factored representation F of Θ, the identity F2FM Σ from Θ to F as well as the identity label mapping λ on the labels of Θ. It then repeatedly applies factored transformations on F until reaching a specified limit or until F consists of a single factor. Every factor of F together with Σ induces an abstraction heuristic and thus the algorithm can be terminated at any point to return these abstraction heuristics.

There are four standard merge-and-shrink transformations. *Shrinking* applies a state abstraction on some factor $\Theta \in F$. *Merging* replaces two factors $\Theta, \Theta' \in F$ with the factor $\Theta \otimes \Theta'$. *Label reduction* applies an abstraction on the common label set of F. *Pruning* removes some states and their transitions in some factor $\Theta \in F$.

Transformations

Our first objective is the formalization of transformations in the probabilistic setting akin to transformations over transition systems. All proof details for proofs sketched or omitted in the rest of this paper can be found in our technical report (Klößner et al. 2023).

While a TS has a transition *relation* $T \subseteq S \times L \times S$ and an agent chooses a transition $\langle s, \ell, t \rangle \in T$ when in state s, the SSP model assumes a transition *function* $P : S \times L \times S \to [0, 1]$ with $\sum_{t \in S} P(s, \ell, t) \in \{0, 1\}$ for all $\langle s, \ell \rangle \in S \times L$, and the agent chooses a label $\ell \in L(s)$. To obtain a proper extension of the classical transformation theory, we first extend the SSP model to adhere to standard TS semantics by extending P to a relation and choosing transitions instead.

Definition 1 (Probabilistic Transition Systems) *A* probabilistic transition system *(PTS) is a tuple* $\Theta = \langle S, L, c, T, G \rangle$ *where* $T \subseteq S \times L \times \mathcal{D}(S)$ *is a finite probabilistic transition relation, and all other components are the same as for SSPs. We define* $T(s) := \{\langle s, \ell, \delta \rangle \in T\}$.

A policy for Θ is now a function $\pi : S \rightharpoonup T$ which terminates or executes a transition $\pi(s) = \langle s, \ell, \delta \rangle \in T(s)$ when visiting s, leading to $t \in S$ with probability $\delta(t)$. The set of s-proper policies $\Pi(s)$ and the cost-to-goal functions $J^\pi_\Theta(s)$ and $J^*_\Theta(s)$ are defined analogously. The Bellman optimality operator $\mathcal{B}_\Theta$ now becomes $(\mathcal{B}_\Theta J)(s) := J(s)$ if $T(s) = \emptyset$, else $(\mathcal{B}_\Theta J)(s) := \min_{\langle s, \ell, \delta \rangle \in T(s)} [c(\ell) + \sum_{t \in S} \delta(t) J(t)]$. As before, a heuristic for Θ is a function $h : S \to \mathbb{R}_{\geq 0}$. All heuristic properties are defined analogously.

The induced PTS of a PPT $\Pi = \langle \mathcal{V}, O, c, s_0, s_\star \rangle$ is the PTS $\Theta(\Pi) = \langle \mathcal{A}(\mathcal{V}), O, c, T, G \rangle$ where $G := \{s \mid s \models s_\star\}$ and $T := \{\langle s, o, \delta_{s,o} \rangle \mid s \models \mathrm{pre}(o)\}$ with $\delta_{s,o}(t) := \sum_{e \in \mathcal{A}^p(\mathcal{V}). s[\![e]\!] = t} \Pr_o(e)$. Clearly, $J^*_{S(\Pi)} = J^*_{\Theta(\Pi)}$.

We now formally define transformations for PTS. A transformation of Θ into Θ' is a tuple $\langle \Theta, \Theta', \sigma, \lambda \rangle$ where $\Theta = \langle S, L, c, T, G \rangle$ is the original PTS, $\Theta' = \langle S', L', c', T', G' \rangle$ is the transformed PTS, $\sigma : S \rightharpoonup S'$ is a state mapping and $\lambda : L \rightharpoonup L'$ is a label mapping. The composition of two transformations $\tau = \langle \Theta, \Theta', \sigma, \lambda \rangle$ and $\tau' = \langle \Theta', \Theta'', \sigma', \lambda' \rangle$ is the transformation $\tau' \circ \tau := \langle \Theta, \Theta'', \sigma' \circ \sigma, \lambda' \circ \lambda \rangle$.

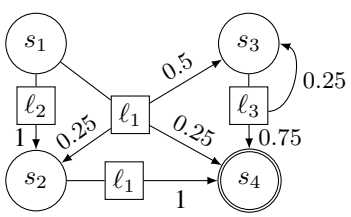
To introduce syntactic categories of such transformations, we introduce a natural way to lift a state mapping $\sigma : S \rightharpoonup S'$ to a state distribution mapping $\sigma_\mathcal{D} : \mathcal{D}(S) \rightharpoonup \mathcal{D}(S')$ by defining $\mathrm{dom}(\sigma_\mathcal{D}) := \{\delta \mid \mathrm{supp}(\delta) \subseteq \mathrm{dom}(\sigma)\}$ and $\sigma_\mathcal{D}(\delta)(s') := \sum_{s \in \sigma^{-1}(s')} \delta(s)$. The probability of s' w.r.t. $\sigma_\mathcal{D}(\delta)$ is the total probability of states in the preimage of s' w.r.t. δ. This lets us associate any transformation $\tau = \langle \Theta, \Theta', \sigma, \lambda \rangle$ with the mapping ind_τ from transitions of Θ to transitions of Θ' *induced* by τ, which is defined as $\mathrm{ind}_\tau(\langle s, \ell, \delta \rangle) := \langle \sigma(s), \lambda(\ell), \sigma_\mathcal{D}(\delta) \rangle$.

With this, we define the following syntactic categories of transformations, which are generalizations of the properties for transformations on transition systems as defined by Sievers and Helmert (2021). We do not consider an explicit initial state, so the properties $\mathbf{IND_I}$ and $\mathbf{CONS_I}$ are omitted.

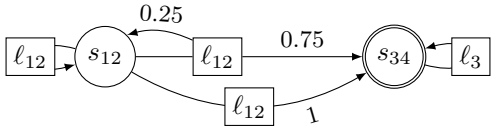
Definition 2 *Let* $\tau = \langle \Theta, \Theta', \sigma, \lambda \rangle$ *be a transformation of* $\Theta = \langle S, L, c, T, G \rangle$ *into* $\Theta' = \langle S', L', c', T', G' \rangle$. *We define the following list of syntactic properties for* τ.

$\mathbf{CONS_S}$	σ *is total on* S
$\mathbf{CONS_L}$	λ *is total on* L
$\mathbf{CONS_C}$	$\forall \ell \in \mathrm{dom}(\lambda). c'(\lambda(\ell)) \leq c(\ell)$
$\mathbf{CONS_T}$	$\mathrm{ind}_\tau(T) \subseteq T'$
$\mathbf{CONS_G}$	$\sigma(G) \subseteq G'$
$\mathbf{IND_S}$	σ *is surjective on* S'
$\mathbf{IND_L}$	λ *is surjective on* L'
$\mathbf{IND_C}$	$\forall \ell' \in L'. \exists \ell \in \lambda^{-1}(\ell'). c(\ell) = c'(\ell')$
$\mathbf{IND_T}$	$\mathrm{ind}_\tau(T) \supseteq T'$
$\mathbf{IND_G}$	$\sigma(G) \supseteq G'$
$\mathbf{REF_C}$	$\forall \ell' \in L'. \forall \ell \in \lambda^{-1}(\ell'). c(\ell) = c'(\ell')$
$\mathbf{REF_T}$	$\forall s' \in S'. T'(s') \subseteq \bigcap_{s \in \sigma^{-1}(s')} \mathrm{ind}_\tau(T(s))$
$\mathbf{REF_G}$	$\sigma^{-1}(G') \subseteq G$

We say that τ *is* conservative *(an* abstraction*) iff* τ *satisfies* $\mathbf{CONS_{S+L+C+T+G}}$, induced *iff* τ *satisfies* $\mathbf{IND_{S+L+C+T+G}}$ *and* refinable *iff* τ *satisfies* $\mathbf{REF_{C+T+G}}$. *Furthermore, we say that* τ *is* exact *iff it is conservative and refinable.*



(a) An example probabilistic transition system Θ.



(b) The transformed PTS $\Theta^{\sigma,\lambda}$ for the state mapping $\sigma : \{s_1, s_2, s_3, s_4\} \to \{s_{12}, s_{34}\}$ with $\sigma(s_1) = \sigma(s_2) = s_{12}$ and $\sigma(s_3) = \sigma(s_4) = s_{34}$ and the label mapping $\lambda : \{\ell_1, \ell_2, \ell_3\} \to \{\ell_{12}, \ell_3\}$ with $\lambda(\ell_1) = \lambda(\ell_2) = \ell_{12}$ and $\lambda(\ell_3) = \ell_3$.

Figure 1: Example of an induced abstraction transformation.

All properties listed in Definition 2 are semantically equivalent to their respective counterpart in the classical theory, except for **IND$_\mathbf{T}$**, **CONS$_\mathbf{T}$** and **REF$_\mathbf{T}$**, which are generalizations for stochastic transitions. If every transition is deterministic, i. e., has a unique successor with probability one, then these properties can also be considered equivalent.

As in the classical case, for every PTS Θ and total and surjective state and label mappings σ and λ, there is a unique PTS $\Theta^{\sigma,\lambda}$ induced by σ and λ such that $\langle \Theta, \Theta^{\sigma,\lambda}, \sigma, \lambda \rangle$ is induced conservative. An example is depicted in Figure 1.

Like their classical counterparts, all of these syntactic properties are compositional. If any of the above properties is shared individually by two composed transformations, the property is inherited by the composition.

Theorem 1 *Let τ and τ' satisfy one of the properties of Definition 2. Then $\tau' \circ \tau$ also satisfies this property.*

Proof. All properties except **CONS$_\mathbf{T}$**, **IND$_\mathbf{T}$** and **REF$_\mathbf{T}$** are semantically equivalent in the classical theory, where they compose. Let $\tau = \langle \Theta, \Theta', \sigma, \lambda \rangle$ and $\tau' = \langle \Theta', \Theta'', \sigma', \lambda' \rangle$. Let T, T' and T'' be the transitions of Θ, Θ' and Θ''.

Regarding **CONS$_\mathbf{T}$**, we have $\mathrm{ind}_\tau(T) \subseteq T'$ due to **CONS$_\mathbf{T}$** for τ, hence $\mathrm{ind}_{\tau' \circ \tau}(T) = \mathrm{ind}_{\tau'}(\mathrm{ind}_\tau(T)) \subseteq \mathrm{ind}_{\tau'}(T') \subseteq T''$ where the first subsumption involves the image of a superset under $\mathrm{ind}_{\tau'}$ and the last subsumption follows from **CONS$_\mathbf{T}$** for τ'. For **IND$_\mathbf{T}$**, the proof is identical, except that we replace $\subseteq$ and **CONS$_\mathbf{T}$** with $\supseteq$ and **IND$_\mathbf{T}$**.

For **REF$_\mathbf{T}$**, let $s'' \in \Theta''$. By **REF$_\mathbf{T}$** for τ' and τ, we have

$$T''(s'') \subseteq \bigcap_{s' \in \sigma'^{-1}(s'')} \mathrm{ind}_\tau(T'(s')) \subseteq \bigcap_{s' \in \sigma'^{-1}(s'')} \mathrm{ind}_\tau(\bigcap_{s \in \sigma^{-1}(s')} T(s)) \subseteq \bigcap_{\substack{s' \in \sigma'^{-1}(s'') \\ s \in \sigma^{-1}(s')}} \mathrm{ind}_\tau(T(s)) = \bigcap_{s \in (\sigma' \circ \sigma)^{-1}(s'')} \mathrm{ind}_\tau(T(s)).$$

where subsumption two involves the image of a superset and subsumption three is basic set algebra. □

For any transformation $\tau = \langle \Theta, \Theta', \sigma, \lambda \rangle$, we define a corresponding heuristic $h^\tau_\Theta(s) := J^*_{\Theta'}(\sigma(s))$ if $s \in \mathrm{dom}(\sigma)$ and $h^\tau_\Theta(s) := \infty$ otherwise for the TS Θ, as in the classical theory. Additionally, we define a corresponding heuristic for Θ' by $h^\tau_{\Theta'}(s') := \max_{s \in \sigma^{-1}(s')} J^*_\Theta(s)$ if $\sigma^{-1}(s') \neq \emptyset$, else $h^\tau_{\Theta'}(s') := 0$. Depending on the syntactic category of the transformation, we can derive guarantees for these heuristics which are similar to those observed in the classical setting.

Theorem 2 *Let $\tau = \langle \Theta, \Theta', \sigma, \lambda \rangle$ be a transformation. (i) If τ is conservative, then h^τ_Θ is goal-aware, consistent, safe and admissible. (ii) If τ is refinable, then $h^\tau_{\Theta'}$ is goal-aware, consistent, safe and admissible. (iii) If τ is refinable, then h^τ_Θ is pessimistic. (iv) If τ is exact, then h^τ_Θ is perfect.*

Proof (sketch). (iii) follows from (ii) since $h^\tau_\Theta(s) = J^*_{\Theta'}(\sigma(s)) \geq h^\tau_{\Theta'}(\sigma(s)) \geq J^*_\Theta(s)$ by admissibility if $s \in \mathrm{dom}(\sigma)$ and $h^\tau_\Theta(s) = \infty \geq J^*_\Theta(s)$ otherwise. Clearly, (iv) follows directly from (i) and (iii). Admissibility of the heuristics follows from goal-awareness, consistency and safety.

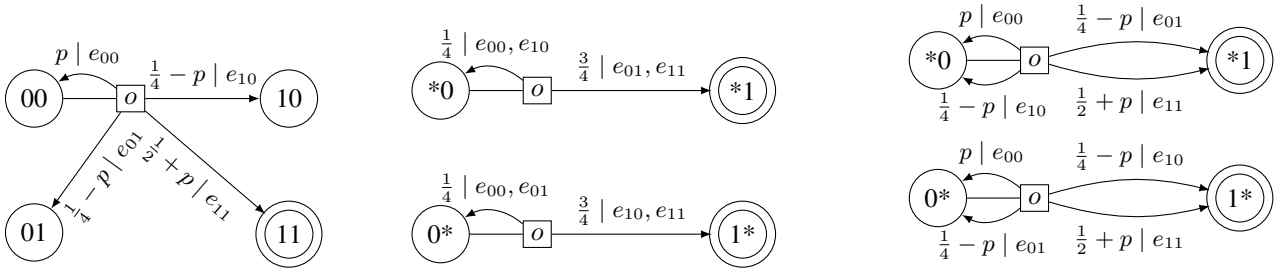
Goal-awareness of h^τ_Θ (respectively $h^\tau_{\Theta'}$) follows directly from **CONS$_\mathbf{S}$** and **CONS$_\mathbf{G}$** (respectively **REF$_\mathbf{G}$**).

Consistency of h^τ_Θ is shown by proving the inequality $h^\tau_\Theta(s) = J^*_{\Theta'}(\sigma(s)) \leq (\mathcal{B}_{\Theta'} J^*_{\Theta'})(\sigma(s)) \leq (\mathcal{B}_\Theta h^\tau_\Theta)(s)$ for all $s \in \Theta$, by making use of **CONS$_\mathbf{S+L+C+T}$** for the last inequality. Consistency of $h^\tau_{\Theta'}$ is shown by proving $J^*_\Theta(s) \leq (\mathcal{B}_\Theta J^*_\Theta)(s) \leq (\mathcal{B}_{\Theta'} h^\tau_{\Theta'})(s')$ for all $s' \in \Theta', s \in \sigma^{-1}(s')$, where the last inequality is derived using **REF$_\mathbf{C+T}$**.

Regarding safety of h^τ_Θ, we argue that the maximal goal probability of a state $s \in \Theta$ is lower or equal to the maximal goal probability of the state $\sigma(s) \in \Theta'$. Therefore, $J^*_\Theta(s) \in \mathbb{R}_{\geq 0}$ implies $h^\tau_\Theta(s) = J^*_{\Theta'}(\sigma(s)) \in \mathbb{R}_{\geq 0}$. For safety of $h^\tau_{\Theta'}$, we show that the maximal goal probability of a state $s' \in \Theta'$ is a lower bound for the maximal goal probability of all states $s \in \sigma^{-1}(s')$ of Θ. This means that $J^*_{\Theta'}(s') \in \mathbb{R}_{\geq 0}$ implies $J^*_\Theta(s) \in \mathbb{R}_{\geq 0}$ for all states $s \in \sigma^{-1}(s')$. Ultimately, we have $h^\tau_{\Theta'}(s') = \max_{s \in \sigma^{-1}(s')} J^*_\Theta(s) \in \mathbb{R}_{\geq 0}$. □

Beyond these basic properties, we can also pinpoint exactly when an induced and conservative transformation is also refinable. In classical planning, *bisimulations* can be used to characterize this property for transformations which only alter the states. We now revisit this property in our setting, this time formulated in terms of a variant of *probabilistic bisimulation* (Larsen and Skou 1991). For probabilistic transition systems, we define bisimulation as follows.

Definition 3 (PTS Bisimulation) *Let $\Theta = \langle S, L, c, T, G \rangle$ be a PTS. A PTS bisimulation on Θ is an equivalence relation $\sim \subseteq S \times S$ such that for all states $s_1, s_2 \in S$ with $s_1 \sim s_2$:* (**BISIM$_\mathbf{1}$**) *if $s_1 \in G$ then $s_2 \in G$ and* (**BISIM$_\mathbf{2}$**) *if $\langle s_1, \ell, \delta_1 \rangle \in T$, there is $\langle s_2, \ell, \delta_2 \rangle \in T$ with $\delta_1 \sim \delta_2$, where $\delta_1 \sim \delta_2$ if and only if $\sum_{s \in C} \delta(s) = \sum_{s \in C} \delta'(s)$ for all equivalence classes $C \in S/\sim$.*



(a) The induced PTS $\Theta(\Pi_p)$. (b) The atomic projections $\Theta^v(\Pi_p)$ and $\Theta^w(\Pi_p)$. (c) The atomic projections $A^v(\Pi_p)$ and $A^w(\Pi_p)$.

Figure 2: The induced PTS and atomic (A)PTS projections of the example planning task Π_p, where $p \in [0, \frac{1}{4}]$.

Naturally, a state mapping σ induces an equivalence relation on states $\sim_\sigma := \{(s,t) \mid \sigma(s) = \sigma(t)\}$. In direct correspondence to the classical situation, we will now prove the following, which will be useful when proving certain properties about shrink transformations later on.

Theorem 3 *Let Θ be a PTS and let $\tau = \langle \Theta, \Theta^{\sigma,\mathrm{id}}, \sigma, \mathrm{id} \rangle$. Then τ is exact if and only if $\sim_\sigma$ is a PTS bisimulation on Θ.*

Proof (sketch). We can follow the same argumentation as in the classical theory, showing the following properties:

(i) If $\sim_\sigma$ satisfies **BISIM$_1$** then τ satisfies **REF$_\mathbf{G}$**.

(ii) If $\sim_\sigma$ satisfies **BISIM$_2$** then τ satisfies **REF$_\mathbf{T}$**.

(iii) If τ satisfies **REF$_\mathbf{G}$**, then $\sim_\sigma$ satisfies **BISIM$_1$**.

(iv) If τ satisfies **REF$_\mathbf{T}$**, then $\sim_\sigma$ satisfies **BISIM$_2$**.

While (i) and (ii) show the reverse implication (note here that **REF$_\mathbf{C}$** is trivial because $\lambda = \mathrm{id}$), (iii) and (iv) show the forward implication. □

The Synchronized Product

We now devote our attention to state spaces generated from a probabilistic planning task. We want to give a definition of *atomic projections* in correspondence to their definition in classical merge-and-shrink, and investigate their role as a natural starting point for the merge-and-shrink algorithm. To this end, we define atomic projections as follows.

Definition 4 (Atomic PTS Projections) *Let $\Pi = \langle \mathcal{V}, O, c, s_0, s_\star \rangle$ be a PPT. The* atomic projection *to $v \in \mathcal{V}$ is defined as $\Theta^v(\Pi) := \langle \mathcal{D}(v), O, c, T^v, G^v \rangle$ where $G^v := \mathcal{D}(v)$ if $v \notin \mathrm{dom}(s_\star)$ and $G^v := \{s_\star(v)\}$ otherwise, and $T^v := \{\langle d, o, \delta_{d,o} \rangle \mid v \notin \mathrm{dom}(\mathrm{pre}(o))$ or $d = \mathrm{pre}(o)_v\}$ where $\delta_{d,o}(d') := \sum_{e \in \mathcal{A}^p(\mathcal{V}).(v \notin \mathrm{dom}(e) \wedge d' = d) \vee e(v) = d'} \Pr_o(e)$.*

This definition generalizes the definition of the transition relation in the most natural way while keeping all other components the same as in the traditional framework. In classical planning, $\Theta(\Pi)$ can be reconstructed from the atomic projections by computing the synchronized product. Therefore, the atomic projections yield a suitable initialization of the factored TS maintained during the algorithm. Regrettably, this guarantee is lost in presence of stochastic operators.

As an example, consider a parameterized PPT Π_p with $p \in [0, \frac{1}{4}]$. Π_p has two boolean variables $v, w \in \mathcal{V}$ and a single operator $o \in O$. Denoting the effect $\{v \mapsto x, w \mapsto y\}$ with e_{xy}, o has the precondition $\mathrm{pre}(o) = e_{00}$ and effect probabilities $\Pr_o(e_{00}) = p$, $\Pr_o(e_{01}) = \Pr_o(e_{10}) = \frac{1}{4} - p$ and $\Pr_o(e_{11}) = \frac{1}{2} + p$. The induced PTS $\Theta(\Pi_p)$ is depicted in Figure 2a and the atomic projections $\Theta^v(\Pi_p)$ and $\Theta^w(\Pi_p)$ are depicted in Figure 2b, where transitions are also annotated with the effects that contribute a positive probability. These annotations are not part of the state space model and are only depicted for demonstrative purposes. As we can see, $\Theta^v(\Pi_p)$ and $\Theta^w(\Pi_p)$ are completely independent from the parameter p, whereas $\Theta(\Pi_p)$ is not. Therefore, atomic projections do not contain enough information to uniquely determine and thus reconstruct the induced PTS of a planning task, which makes the definition of a reconstructing synchronized product impossible.

The problem is that we lose information about the individual outcomes of the operators when we build an atomic projection. If we want to synchronize the atomic projections correctly, we must not only make sure that the same operator is applied in all atomic projections, but also that the outcomes are synchronized and happen with the same probability. Regrettably, a transition in an atomic projection is still associated with its original operator that induced it, but can correspond to multiple distinct outcomes with an identical effect on the projection variable. The individual probabilities of these outcomes are lost, so synchronization is impossible.

In order to synchronize on the individual operator outcomes, we henceforth keep track of them in our underlying state space model. To this end, we integrate a set of *label effects* E, which are opaque identifiers for the outcomes of an operator and annotate the transitions with them accordingly. We additionally track the probability of each label effect. This leads to the following refinement of our model.

Definition 5 (Annotated PTS) *An* annotated *PTS (APTS) is a tuple $A = \langle S, L, c, E, D, T, G \rangle$ where E is a set of* label effects*, $D = (D_\ell)_{\ell \in L}$ is a family of probability distributions over E, $T \subseteq S \times L \times (E \to S)$ is a set of transitions and all other components are the same as for PTS.*

The probabilities are now encoded in the label effects instead of the transitions. Thus, we only store a *successor mapping*

$\alpha : E \to S$ from label effects to corresponding successors. Given a transition $\langle s, \ell, \alpha \rangle$, the intuition is that effect e occurs with probability $D_\ell(e)$ and leads to state $\alpha(e)$. If $D_\ell(e) = 0$, the associated successor is impossible and could be formally omitted by using partial mappings instead. We choose not to do so to keep the notation simple.

Of course, in the context of a transition with label ℓ, we can always cast a successor mapping $\alpha : E \to S$ to a successor distribution $\delta_\ell(\alpha) \in \mathcal{D}(S)$ defined as $\delta_\ell(\alpha)(t) := \sum_{e \in \alpha^{-1}(t)} D_\ell(e)$, dropping the additional information. Doing this for every transition, we obtain the natural interpretation as a PTS. Formally, given $A = \langle S, L, c, E, D, T, G \rangle$ we define this PTS as $\Theta(A) := \langle S, L, c, T', G \rangle$ where $T' := \{\langle s, \ell, \delta_\ell(\alpha) \rangle \mid \langle s, \ell, \alpha \rangle \in T\}$.

We now revisit the state space reconstruction problem and solve it by employing the new state space model. First, we define the induced APTS $A(\Pi) := \langle \mathcal{A}(\mathcal{V}), O, c, E, D, T, G \rangle$ of a PPT $\Pi = \langle \mathcal{V}, O, c, s_0, s_\star \rangle$ in the natural way. The label-effects $E := \bigcup_{o \in O} \operatorname{supp}(\Pr_o)$ correspond to the possible effects of the operators on the state variables, the label effect probabilities $D_o(e) := \Pr_o(e)$ match with the effect probabilities for the respective operator and the transition relation is defined by $T := \{\langle s, o, \alpha_s \rangle \mid s \models \operatorname{pre}(o)\}$ where $\alpha_s(e) := s[\![e]\!]$. The state space in Figure 2a can now be seen as a depiction of the induced APTS $A(\Pi_p)$ of the aforementioned planning task Π_p. The atomic APTS projection with respect to $v \in \mathcal{V}$ is defined analogously as follows.

Definition 6 (Atomic APTS Projection) *Let $\Pi = \langle \mathcal{V}, O, c, s_0, s_\star \rangle$ be a PPT. The* atomic APTS projection *to $v \in \mathcal{V}$ is defined as $A^v(\Pi) := \langle \mathcal{D}(v), O, c, E, D, T^v, G^v \rangle$ where $E := \bigcup_{o \in O} \operatorname{supp}(\Pr_o)$, $D := (\Pr_o)_{o \in O}$, $G^v := \mathcal{D}(v)$ if $v \notin \operatorname{dom}(s_\star)$ and $G^v := \{s_\star(v)\}$ otherwise, and $T^v := \{\langle d, o, \alpha_d \rangle \mid v \in \operatorname{dom}(\operatorname{pre}(o))$ implies $d = \operatorname{pre}(o)_v\}$ with $\alpha_d(e) := d$ if $v \notin \operatorname{dom}(e)$ and $\alpha_d(e) := e_v$ otherwise.*

As an example, reconsider the previously defined parameterized task Π_p. Figure 2c depicts the atomic APTS projections $A^v(\Pi_p)$ and $A^w(\Pi_p)$ for Π_p. Compared to the atomic PTS projections $\Theta^v(\Pi_p)$ and $\Theta^w(\Pi_p)$ in Figure 2b, each outcome of the operator o is now associated with its own transition, and the probabilities are maintained.

Lastly, we define the synchronized product. The intuition should be clear at this point: We synchronize transitions annotated with both the same label and label effect. To formalize this idea, for a family of successor mappings $\alpha = (\alpha_i)_{i \in I}$ where $\alpha_i : E \to S_i$, we define the synchronization $\bigotimes \alpha : E \to \times_{i \in I} S_i$ by $(\bigotimes \alpha)(e) := (\alpha_i(e))_{i \in I}$. This leads to the following definition.

Definition 7 (Synchronized Product of APTS) *The* synchronized product *of a family of APTS $F = (A_i)_{i \in I}$ where $A_i = \langle S_i, L, c, E, D, T_i, G_i \rangle$ for $i \in I$ is the APTS $\bigotimes F := \langle \times_{i \in I} S_i, L, c, E, D, T_\otimes, \times_{i \in I} G_i \rangle$ with $T_\otimes := \{\langle s, \ell, \bigotimes_{i \in I} \alpha_i \rangle \mid \forall i \in I. \langle s_i, \ell, \alpha_i \rangle \in T_v\}$.*

If we apply the synchronized product to $A^v(\Pi_p)$ and $A^w(\Pi_p)$ depicted in Figure 2c, we can see that this results exactly in $A(\Pi_p)$ as depicted in Figure 2a, as desired. We conclude this section by verifying that this holds in general.

Theorem 4 *Let Π be a probabilistic planning task with variables $\mathcal{V}$. Then $A(\Pi) = \bigotimes_{v \in \mathcal{V}} A^v(\Pi)$.*

Proof. The labels, label effects, label effect probabilities and the cost function coincide because they are unaffected by the product. The argument for the (goal) states matches the classical proof. Before we focus on the transitions, we note that $\alpha_s = \bigotimes_{v \in \mathcal{V}} \alpha_{s_v}$ for all states $s \in \mathcal{A}(\mathcal{V})$.

"$\subseteq$" Let $\langle s, o, \alpha \rangle \in A(\Pi)$. By definition of $A(\Pi)$, $\alpha = \alpha_s$ and $s \models \operatorname{pre}(o)$. Hence, we have $s_v = \operatorname{pre}(o)_v$ for all $v \in \operatorname{dom}(\operatorname{pre}(o))$. By definition of the $A^v(\Pi)$, $\langle s_v, o, \alpha_{s_v} \rangle \in A^v(\Pi)$ for all $v \in \mathcal{V}$. By definition of the product, we have $\langle s, o, \bigotimes_{v \in \mathcal{V}} \alpha_{s_v} \rangle = \langle s, o, \alpha_s \rangle \in \bigotimes_{v \in \mathcal{V}} A^v(\Pi)$.

"$\supseteq$" Let $\langle s, o, \alpha' \rangle \in \bigotimes_{v \in \mathcal{V}} A^v(\Pi)$. By definition of the product, there is α with $\alpha' = \bigotimes \alpha$ such that $\langle s_v, o, \alpha_v \rangle \in A^v(\Pi)$ for $v \in \mathcal{V}$. By definition of $A^v(\Pi)$, $\alpha_v = \alpha_{s_v}$ and $s_v = \operatorname{pre}(o)_v$ for all $v \in \operatorname{dom}(\operatorname{pre}(o))$. Ultimately, $s \models \operatorname{pre}(o)$ and $\langle s, o, \alpha_s \rangle = \langle s, o, \bigotimes_{v \in \mathcal{V}} \alpha_{s_v} \rangle \in A(\Pi)$. □

Merge-and-Shrink Transformations

As we have just seen, APTS are expressive enough to formulate an appropriate synchronous product. We will now use this state space model as a foundation to formally describe our merge-and shrink framework for probabilistic planning.

As the underlying factored representation of the state space, we maintain a family of APTS $F = (A_i)_{i \in I}$ where I is a finite index set. F implicitly represents the APTS $\bigotimes F$. We call F a *factored APTS* (FAPTS) and its elements the *factors* of F. All factors share the same labels, label effects, label effect probabilities and cost function. Given a task Π with variables $\mathcal{V}$, we start with the initialization $F = (A^v(\Pi))_{v \in \mathcal{V}}$, as justified by Theorem 4. Clearly, each $A^v(\Pi)$ is constructible in time polynomial in the size of Π.

To define the basic merge-and-shrink transformations, we first introduce APTS transformations in the same way as for probabilistic transition systems. A transformation of APTS A with states S and labels L into A' with states S' and labels L' is a tuple $\tau = \langle A, A', \sigma, \lambda \rangle$ where A is the original APTS, A' is the transformed APTS and σ and λ are again state and label mappings. The composition $\tau' \circ \tau$ of two APTS transformations τ and τ' is defined analogously. The FAPTS F is subject to *factored* transformations into some other FAPTS F', analogously defined as tuples $\langle F, F', \Sigma, \lambda \rangle$ where Σ is an F2FM. Such a factored transformation induces the non-factored transformation $\langle \bigotimes F, \bigotimes F', [\![\Sigma]\!], \lambda \rangle$.

The main purpose of the APTS model is to enable synchronization of the factors and hence to enable a definition of merge transformations with appropriate guarantees, as we will see. When we want to compute a heuristic from a factor A, we do however not need the additional information and compute the heuristic values from the induced PTS $\Theta(A)$. Therefore, for an APTS transformation $\tau = \langle A, A', \sigma, \lambda \rangle$, we define the induced heuristic $h^\tau_A(s) := h^{\tau_{\text{PTS}}}_{\Theta(A)}(s)$, where $\tau_{\text{PTS}} := \langle \Theta(A), \Theta(A'), \sigma, \lambda \rangle$ is the induced PTS transformation of τ. Since we define h^τ_A directly in terms of τ_{PTS}, we simply say that τ satisfies a syntactic property of Definition 2 iff τ_{PTS} satisfies it and obtain the equivalent of Theorem 2 for APTS transformations.

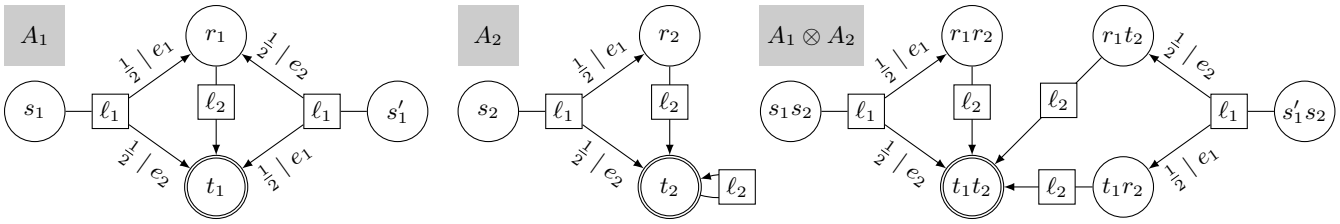


Figure 3: A counter-example illustrating inexactness of PTS bisimulation-based shrinking. Here, both labels have unit cost. The single effect of ℓ_2 is not explicitly drawn. On the left, two factors A_1 and A_2 and on the right, their implicitly represented APTS $A_1 \otimes A_2$ are depicted. While s_1 and s'_1 are clearly bisimilar in $\Theta(A_1)$, shrinking them would cause the states s_1s_2 and s'_1s_2 to be merged in $A_1 \otimes A_2$. This would decrease the optimal cost-to-goal of s'_1s_2 in $A_1 \otimes A_2$ from 2 to 1.5.

We now carry over three of the four factored transformations used in merge-and-shrink — Shrinking, Merging, Label Reduction — to our setting and show that fundamental properties still apply. We also investigate more advanced instantiations of these transformations which yield stronger guarantees for the constructed merge-and-shrink heuristic.

Shrink Transformations

Shrink transformation are conceptually unchanged. A shrink transformation applies some local abstraction on one of the factors. To this end, let $A = \langle S, L, c, E, D, T, G\rangle$ be an APTS and let $\sigma : S \to S'$ be a total state mapping. We will denote with A^σ the APTS $\langle S', L, c, E, D, T', \sigma(G)\rangle$ where $T' := \{\langle \sigma(s), \ell, \sigma \circ \alpha\rangle \mid \langle s, \ell, \alpha\rangle \in T\}$ in the following.

Definition 8 (Shrink Transformation) *Let $F = (A_i)_{i\in I}$ be a FAPTS and let σ be a total state mapping on some $A_k \in F$. The* shrink transformation *for σ and A_k is the factored transformation $\langle F, F', \Sigma, \mathrm{id}\rangle$ where $F' = (A_i)_{i\in I\setminus\{k\}} \cup \{k \mapsto A_k^\sigma\}$ and $\Sigma = (\pi_i)_{i\in I\setminus\{k\}} \cup \{k \mapsto \mathrm{Atom}(k, \sigma)\}$.*

In the classical merge-and-shrink framework, shrink transformations are generally induced and conservative. Fortunately, this property naturally extends to our framework.

Theorem 5 *Shrink transformations are ind. abstractions.*

Proof (sketch). The proofs for the properties $\mathbf{CONS_{S+L+C+G}}$ and $\mathbf{IND_{S+L+C+G}}$ are identical in the classical theory. For $\mathbf{CONS_T}$ and $\mathbf{IND_T}$, we extend the corresponding classical proof for transition systems in a straightforward manner by substituting every occurrence of a successor state with a successor mapping. □

An interesting classical class of shrinking strategies is based on *bisimulation* (Milner 1990). Bisimulation-based shrinking strategies are prominent in classical planning (Nissim, Hoffmann, and Helmert 2011) because they yield exact transformations. In Theorem 3, we have shown a similar relationship between PTS bisimulations and exact transformations. The natural extension of this strategy is then to shrink states in the selected factor $A_k \in F$ according to a PTS bisimulation on $\Theta(A_k)$. Unfortunately, unlike in classical planning, this transformation is not exact, as such a shrink transformation may not correspond to a PTS bisimulation on $\bigotimes F$. A counter-example is depicted in Figure 3.

The problem is that in the APTS A_1 in Figure 3, the transitions with label ℓ_1 have bisimilar successor distributions in $\Theta(A_1)$, but do not lead to bisimilar states for a fixed label effect. By adding this requirement, we could recover exactness. However, we will consider a slightly more general approach. We say that two label effects e_1 and e_2 are *locally equivalent* for label ℓ in A if $\langle s, \ell, \alpha\rangle \in A$ implies $\alpha(e_1) = \alpha(e_2)$. In the context of a FAPTS F, we say that two label effects e_1, e_2 are *A-combinable* for ℓ iff they are locally equivalent for ℓ in every other factor $A' \in F \setminus \{A\}$. We write $e_1 \simeq_{A,\ell} e_2$ when e_1 and e_2 are A-combinable for ℓ. Note that $\simeq_{A,\ell}$ is an equivalence relation. Now, instead of enforcing bisimilar successors for each label effect, we only require that the successor distributions are bisimilar under the condition that a fixed but arbitrary class of *A-combinable* effects is triggered. This leads to the following definition.

Definition 9 (APTS Bisimulation) *Let F be a FAPTS. An* APTS bisimulation *on a factor $A = \langle S, L, c, E, D, T, G\rangle$ of F is an equivalence relation $\sim \subseteq S \times S$ such that for all $s_1, s_2 \in S$ with $s_1 \sim s_2$, (i) if $s_1 \in G$ then $s_2 \in G$ and (ii) if $\langle s_1, \ell, \alpha_1\rangle \in T$ then there is $\langle s_2, \ell, \alpha_2\rangle \in T$ with $\alpha_1 \sim \alpha_2$, where $\alpha_1 \sim \alpha_2$ iff $\sum_{e\in D\cap\alpha_1^{-1}(C)} D_\ell(e) = \sum_{e\in D\cap\alpha_2^{-1}(C)} D_\ell(e)$ for all equivalence classes $C \in S/\!\sim$ and $D \in E/\!\simeq_{A,\ell}$.*

Theorem 6 *APTS bisimulation-based shrinking is exact.*

Proof (sketch). Let $\tau = \langle F, F', \Sigma, \mathrm{id}\rangle$ be a shrink transformation for σ and $A_k \in F$. The idea is to show that $\sim_{[\![\Sigma]\!]}$ is a PTS bisimulation on $\Theta(\bigotimes F)$ if $\sim_\sigma$ is an APTS bisimulation on A_k. Exactness then follows by Theorem 3. □

Merge Transformations

Merge transformations are also conceptually identical to their classical variant. They replace two factors by their synchronous product.

Definition 10 (Merge Transformation) *Let $F = (A_i)_{i\in I}$ be a FAPTS and let $j, k \in I, j \neq k$. The* merge transformation *for j and k is the factored transformation $\langle F, F', \Sigma, \mathrm{id}\rangle$ where $F' := (A_i)_{i\in I\setminus\{j,k\}} \cup \{\langle j, k\rangle \mapsto A_j \otimes A_k\}$ and $\Sigma := (\pi_i)_{i\in I\setminus\{j,k\}} \cup \{\langle j, k\rangle \mapsto \mathrm{Merge}(\pi_j, \pi_k, \mathrm{id})\}$.*

In classical planning, a merge transformation is not only exact, but $\bigotimes F$ and $\bigotimes F'$ are even identical up to renaming of states. Transformations with this property are also called *isomorphisms*. To show the analogous property for our setting, we define isomorphisms on APTS as transformations $\tau = \langle A, A^\sigma, \sigma, \mathrm{id}\rangle$ such that σ is bijective. It is straightforward to see that isomorphisms are composable, exact transformations as in classical planning.

Theorem 7 *Merge transformations are isomorphisms.*

Label Reduction

Lastly, we consider label reduction. In contrast to classical label reduction, it is harder to give a general definition for this transformation in our setting, since we also need to explain how to unify the label effects of the labels, and also unify their probabilities. Remember that a label effect is just a token that identifies a possible outcome for a specific label. The alphabet of tokens is shared across labels, but the semantics of a label effect in only defined in the context of a label, and is independent of other labels.

For label reductions, we assume a label mapping $\lambda : L \to L'$ that maps each label $\ell \in L$ of the FAPTS F to a reduced label $\lambda(\ell) \in L'$. To be able to interpret an outcome of ℓ as an outcome of the reduced label $\lambda(\ell)$, we additionally parameterize the reduction by a family of *label effect mappings* $\epsilon = (\epsilon_\ell)_{\ell\in L}$. Each $\epsilon_\ell : E \to E'$ bijectively maps from the old label effects E of F to a set of new label effects E'. We say that ϵ *unifies* label ℓ_1 and ℓ_2 iff for every $e_1, e_2 \in E$, $\epsilon_{\ell_1}(e_1) = \epsilon_{\ell_2}(e_2)$ implies $D_{\ell_1}(e_1) = D_{\ell_2}(e_2)$, i. e., label effects that map to the same new label effect have the same probability. Given a label mapping λ, we say that ϵ is a *unifier* for λ iff $\lambda(\ell_1) = \lambda(\ell_2)$ implies that ϵ unifies ℓ_1 and ℓ_2.

Now, given the total and surjective label mapping $\lambda : L \to L'$ and a unifier $\epsilon = (\epsilon_\ell)_{\ell\in L}$ for λ with $\epsilon_\ell : E \to E'$, we define the label reduced APTS of A by $A^{\lambda,\epsilon} := \{S, L', c', E', D', T', G\}$ where $D'_{\ell'} := D_\ell \circ \epsilon_\ell^{-1}$ for any $\ell \in \lambda^{-1}(\ell')$. Moreover, $c'(\ell') := \min_{\ell\in\lambda^{-1}(\ell')} c(\ell)$ and $T' := \{\langle s, \lambda(\ell), \alpha \circ \epsilon_\ell^{-1}\rangle \mid \langle s, \ell, \alpha\rangle \in T\}$. With this, label reduction transformations are defined as follows.

Definition 11 (Label Reduction) *Let $F = (A_i)_{i\in I}$ be a FAPTS with labels L. Let λ be a total, surjective label mapping defined on L and let ϵ be a label effect unifier for λ. The* label reduction transformation *for λ and ϵ is the factored transformation $\langle F, (A_i^{\lambda,\epsilon})_{i\in I}, (\pi_i)_{i\in I}, \lambda\rangle$.*

We can show that our formalization of label reduction satisfies the same syntactic properties as in the classical case.

Theorem 8 *Label reductions are abstractions and satisfy* $\mathbf{IND_{S+L+C+G}}$ *and* $\mathbf{REF_G}$. *Furthermore, they satisfy* $\mathbf{REF_C}$ *if and only if only labels with the same costs are reduced and satisfy* $\mathbf{IND_T}$ *if and only if they satisfy* $\mathbf{REF_T}$.

Proof (sketch). The proofs for properties $\mathbf{CONS_{S+L+C+G}}$, $\mathbf{IND_{S+L+C+G}}$ and $\mathbf{REF_G}$, and the necessary and sufficient condition of $\mathbf{REF_C}$ are given in the classical theory. Since $[\![\Sigma]\!] = \mathrm{id}$ we have $\bigcap_{s\in[\![\Sigma]\!]^{-1}(s')} \mathrm{ind}_\tau(T(s)) = \mathrm{ind}_\tau(s')$ and so $\mathbf{IND_T}$ and $\mathbf{REF_T}$ collapse to a common statement.

For $\mathbf{CONS_T}$, let $\langle s, \ell, \delta\rangle \in \Theta(\bigotimes F)$. By definition of $\Theta(A)$ and the product, there is some α such that $\delta = \delta_\ell(\bigotimes_{i\in I}\alpha_i)$ and $\langle s_i, \ell, \alpha_i\rangle \in A_i$ for all $i \in I$. By definition of $A_i^{\lambda,\epsilon}$, we get $\langle s_i, \lambda(\ell), \alpha_i \circ \epsilon_\ell^{-1}\rangle \in A_i^{\lambda,\epsilon}$ for all $i \in I$. By definition of the product and $\Theta(A)$, $\langle s, \lambda(\ell), \delta_{\lambda(\ell)}(\bigotimes_{i\in I}\alpha_i \circ \epsilon_\ell^{-1})\rangle \in \Theta(\bigotimes F')$. It is left to show that $\delta_\ell(\bigotimes_{i\in I}\alpha_i) = \delta_{\lambda(\ell)}(\bigotimes_{i\in I}\alpha_i \circ \epsilon_\ell^{-1})$. Due to space reasons, we give a proof of this claim in the techincal report. □

Last but not least, we propose an exact label reduction strategy. To this end, we only consider *atomic* label reductions, which only reduce exactly two labels. Recall that in classical planning, it is NP-hard to decide whether a non-atomic label reduction is exact, however, exactness of atomic label reductions can be characterized in terms of the notions of *label subsumption* and Θ-*combinability*. We will extend these notions to our setting by integrating the label effect mapping ϵ of a label reduction into these concepts.

For an APTS A, a label ℓ of A and a unifier ϵ for λ, we define $T_\epsilon(A, \ell) := \{\langle s, \alpha \circ \epsilon_\ell^{-1}\rangle \mid \langle s, \ell, \alpha\rangle \in A\}$. If there is no transition $\langle s, \ell, \alpha\rangle \in A$, then we say ℓ is *dead* in A. We say that ℓ_1 ϵ-*subsumes* ℓ_2 in A iff ϵ unifies ℓ_1 and ℓ_2 and if $T_\epsilon(A, \ell_1) \subseteq T_\epsilon(A, \ell_2)$. If ϵ-subsumption holds in both directions, we say l_1 and l_2 are ϵ-*equivalent* in A. Lastly, given a FAPTS F we say l_1 and l_2 are (A, ϵ)-*combinable* iff l_1 and l_2 are ϵ-equivalent in every other factor $A' \in F \setminus \{A\}$. With this, we state the following sufficient condition for exactness of atomic label reduction. In contrast to its classical counterpart, we do not claim that this condition is necessary.

Theorem 9 *A label reduction of a FAPTS F that combines exactly two labels ℓ_1, ℓ_2 of F is exact if $c(\ell_1) = c(\ell_2)$ and either (A) ℓ_1 ϵ-subsumes ℓ_2 in all $A \in F$ or vice versa, (B) ℓ_1 and ℓ_2 are (A, ϵ)-combinable for some $A \in F$ or (C) ℓ_1 and ℓ_2 are dead for some $A \in F$.*

Proof (sketch). The proof is an extension of the corresponding exactness proof in the classical theory. First, we show that each of the conditions (A), (B) and (C) imply the subsumption

$$\bigtimes_{i\in I} \bigcup_{\ell\in\lambda^{-1}(\ell')} T_\epsilon(A_i, \ell) \subseteq \bigcup_{\ell\in\lambda^{-1}(\ell')} \bigtimes_{i\in I} T_\epsilon(A_i, \ell).$$

In a second step, we can then show that this inclusion is sufficient to prove $\mathbf{IND_T}$ by contraposition. Exactness then follows directly from Theorem 8. □

Conclusion

We presented an extension of Merge-and-Shrink as a compositional theory of transformations tailored for probablistic planning. We have shown that our extension properly generalizes important properties of the classical theory and provides a new way of constructing admissible SSP abstraction heuristics. Possible future work in this direction includes a formalization of pruning transformations, as well as an experimental evaluation of Merge-and-Shrink as a construction technique for admissible SSP heuristics.

Acknowledgements

This work was funded by DFG grant 389792660 as part of TRR 248 (see https://perspicuous-computing.science). Silvan Sievers was partially supported by TAILOR, a project funded by the EU Horizon 2020 research and innovation programme under grant agreement no. 952215. He further has received funding for this work from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 817639).

References

Bellman, R. 1957. *Dynamic Programming*. Princeton University Press.

Bertsekas, D. P.; and Tsitsiklis, J. N. 1991. An Analysis of Stochastic Shortest Path Problems. *Mathematics of Operations Research*, 16: 580–595.

Bonet, B.; and Geffner, H. 2003. Labeled RTDP: Improving the Convergence of Real-Time Dynamic Programming. In Giunchiglia, E.; Muscettola, N.; and Nau, D., eds., *Proceedings of the 13th International Conference on Automated Planning and Scheduling (ICAPS'03)*, 12–21. Trento, Italy: AAAI Press.

Bonet, B.; and Geffner, H. 2005. mGPT: A Probabilistic Planner Based on Heuristic Search. *Journal of Artificial Intelligence Research*, 24: 933–944.

Boutilier, C.; Dearden, R.; and Goldszmidt, M. 2000. Stochastic dynamic programming with factored representations. *Artificial intelligence*, 121(1-2): 49–107.

Dräger, K.; Finkbeiner, B.; and Podelski, A. 2009. Directed model checking with distance-preserving abstractions. *International Journal on Software Tools for Technology Transfer*, 11(1): 27–37.

Hansen, E. A.; and Zilberstein, S. 2001. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129(1-2): 35–62.

Helmert, M.; Haslum, P.; Hoffmann, J.; and Nissim, R. 2014. Merge & Shrink Abstraction: A Method for Generating Lower Bounds in Factored State Spaces. *Journal of the Association for Computing Machinery*, 61(3): 16:1–16:63.

Helmert, M.; Röger, G.; and Sievers, S. 2015. On the Expressive Power of Non-Linear Merge-and-Shrink Representations. In Brafman, R.; Domshlak, C.; Haslum, P.; and Zilberstein, S., eds., *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS'15)*, 106–114. AAAI Press.

Klößner, T.; and Hoffmann, J. 2021. Pattern Databases for Stochastic Shortest Path Problems. In *Proceedings of the 14th Annual Symposium on Combinatorial Search (SOCS'21)*, 131–135. AAAI Press.

Klößner, T.; Torralba, Á.; Steinmetz, M.; and Sievers, S. 2023. A Theory of Merge-and-Shrink for Stochastic Shortest Path Problems - Technical Report. https://doi.org/10.5281/zenodo.7737867.

Kolobov, A.; Mausam; and Weld, D. S. 2010. Classical Planning in MDP Heuristics: with a Little Help from Generalization. In Brafman, R. I.; Geffner, H.; Hoffmann, J.; and Kautz, H. A., eds., *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS'10)*, 97–104. AAAI Press.

Kolobov, A.; Mausam; and Weld, D. S. 2012. Discovering Hidden Structure in Factored MDPs. *Artificial Intelligence*, 189: 19–47.

Larsen, K. G.; and Skou, A. 1991. Bisimulation through Probabilistic Testing. *Information and Computation*, 94(1): 1–28.

Milner, R. 1990. Operational and Algebraic Semantics of Concurrent Processes. In van Leeuwen, J., ed., *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics*, 1201–1242. Elsevier and MIT Press.

Nissim, R.; Hoffmann, J.; and Helmert, M. 2011. Computing Perfect Heuristics in Polynomial Time: On Bisimulation and Merge-and-Shrink Abstraction in Optimal Planning. In Walsh, T., ed., *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, 1983–1990. AAAI Press/IJCAI.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-Based Heuristics for Cost-Optimal Planning. In Chien, S.; Do, M.; Fern, A.; and Ruml, W., eds., *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS'14)*, 226–234. AAAI Press.

Sievers, S.; and Helmert, M. 2021. Merge-and-Shrink: A Compositional Theory of Transformations of Factored Transition Systems. *Journal of Artificial Intelligence Research*, 71: 781–883.

Torralba, Á.; and Kissmann, P. 2015. Focusing on What Really Matters: Irrelevance Pruning in Merge-and-Shrink. In Lelis, L.; and Stern, R., eds., *Proceedings of the 8th Annual Symposium on Combinatorial Search (SOCS'15)*, 122–130. AAAI Press.

Torralba, Á.; and Sievers, S. 2019. Merge-and-Shrink Task Reformulation for Classical Planning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI 2019*. IJCAI/AAAI Press.

Trevizan, F. W.; Thiébaux, S.; and Haslum, P. 2017. Occupation Measure Heuristics for Probabilistic Planning. In *Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS'17)*, 306–315. AAAI Press.

Trevizan, F. W.; Thiébaux, S.; Santana, P. H.; and Williams, B. 2017. I-dual: Solving Constrained SSPs via Heuristic Search in the Dual Space. In Sierra, C., ed., *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI'17)*, 4954–4958. AAAI Press/IJCAI.