Technische Universität Dresden • Fakultät Informatik

Data Preperation for PMC-Visualization

Bachelorarbeit zur Erlangung des ersten Hochschulgrades

Bachelor of Science (B.Sc.)

vorgelegt von

Franz Martin SCHMIDT

(geboren am 7. April 1999 in Halle (Saale))

Tag der Einreichung: June 11, 2023

Dipl. Inf Max Korn (Theoretische Informatik)

Contents

Abstract

Lorem ipsum

1 Introduction

2 Preliminaries

2.1 Mathematical Fundamentals

e.g.strongly connected components, equivalence relation, more?

we denote $[a]_R$ for the equivalence class with the representative a under the equivalence relation R

how much should be included? ...probably no set theory

2.2 Transition Systems

Motivation of transition systems

The following definition is directly taken form Principles of Modelchecking, Baier p. 20

Definition 2.1. A *transition system* TS is a tuple $(S, Act, \longrightarrow, I, AP, L)$ where

- S is a set of states,
- Act is a set of actions,
- $\longrightarrow \subseteq S \times Act \times S$ is transition relation,
- $I \subseteq S$ is a set of initial states,
- AP is a set of atomic propositions, and
- $L : S \to \mathcal{P}(AP)$

A transition system is called *finite* if S, AP and L are finite.

explanation of components

2.3 Markov Chain

NOTES BEGIN

- Markov Chain (MC)
- transition systems to markov chains: nondeterministic choices replaced by probablistic
- successor chosen according to probability distribution
- distribution only dependent on current state s (not path)
- system evolution not dependent on history but only current state $\to$ *memory-less property*

NOTES END

Definition 2.2. A *(discrete-time) Markov chain* is a tuple $\mathcal{M} = (S, \mathbf{P}, l_{init}, AP, L)$ where

- S is a countable, nonempty set of states,
- $\mathbf{P} : S \times S \to [0, 1]$ is the *transition probability function*, such that for all states s:

$$\sum_{s' \in S} \mathbf{P}(s, S') = 1.$$

- $l_{init} : S \to [0, 1]$ is the *initial distribution*, such that $\sum_{s \in S} l_{init}(s) = 1$, and
- AP is a set of atomic propositions and,
- $L : S \to \mathcal{P}(AP)$ a labeling function.

$\mathcal{M}$ is called *finite* if S and AP are finite. For finite $\mathcal{M}$, the *size* of $\mathcal{M}$, denoted *size*($\mathcal{M}$), is the number of states plus the number of pairs $(s, s') \in S \times S$ with $\mathbf{P}(s, s') > 0$.

NOTES BEGIN

- Probability Function $\mathbf{P}$ specifies for each state s the probability $\mathbf{P}$ (s,s') of moving from s to s' in one step.
- constraint on $\mathbf{P}$ ensures that $\mathbf{P}$ is distribution
- $l_{init}(s)$ specifies system evolution starts in s
- states s with $l_{init}(s) > 0$ are considered *initial states*
- states s' with $\mathbf{P}(s, s') > 0$ are view as possible successors of s
- has no actions
  "As compositional approaches for Markov models are outside the scope of this monograph, actions are irrelevant in this chapter and are therefore omitted."

NOTES END

2.4 Markov Decision Process

NOTES BEGIN

- Markov decision process (MDP)
- idea: Adding nondeterminism to markov chains. MDPs permit both probabilistic and nondeterministic choices
- probabilistic choices: possible outcomes for of randomized actions -¿ requires statistical experiments to obtain adequate distributions that model average behavior of the environment

- information not available or guarantee about system properties is required -¿ nondeterminism
- Another example: randomized distributed algorithms. Non-determinism: interleaving behavior: nondeterministic choice which process, probabilistic: have rather restricted set of actions that have a random nature
- used for abstraction in markov chains: states grouped by AP and have a wide range of transition probabilities -¿ essentially nondeterminism -¿ transition probabilities are replaced by nondeterminism

NOTES END

Definition 2.3. A *Markov decision process* is a tuple $\mathcal{M} = (S, Act, \mathbf{P}, l_{init}, AP, L)$ where

- S is a countable set of states,
- Act is a set of actions,
- $\mathbf{P} : S \times Act \times S \to [0, 1]$ is the transition probability function such that for all states $s \in S$ and actions $\alpha \in Act$:

$$\sum_{s' \in S} \mathbf{P}(s, \alpha, s') \in \{0, 1\},$$

- $l_{init} : S \to [0, 1]$ is the initial distribution such that $\sum_{s \in S} l_{init}(s) = 1$,
- AP is a set of atomic propositions and
- $L : S \to \mathcal{P}(AP)$ a labeling function.

An action α is *enabled* in state s if and only if $\sum_{s' \in S} \mathbf{P}(s, \alpha, s') = 1$. Let $Act(s)$ denote the set of enabled actions in s. For any state $s \in S$, it is required that $Act(s) \neq \emptyset$. Each state s' for which $\mathbf{P}(s, \alpha, s') > 0$ is called an *α-successor* of s.

An MDP is called *finite* if S, Act and AP are finite.

NOTES BEGIN

- $\mathbf{P}(s, \alpha, t)$ can be arbitrary real numbers in $[0, 1]$ (sum up to 1 or 0 for fixed s and α), for algorithmic purposes rational
- unique initial distribution l_{init}. Could be generalized to set of $l_{init}$with nondeterministic choice at the beginning. For sake of simplicity: one single distribution
- operational behavior:
  - starting state s_0 yielded by $l_{init}$with $l_{init}(s_0) > 0$

  - nondeterministic choice of enabled action (i.e. Probability sums up to one)
  - probabilistic choice of state (action fixed by nondeterministic selection)

- $MC = MDP \iff \forall s \in S : |Act(s)| = 1$
- $\implies$ MCs are a proper subset of MDPs

NOTES END

3 View

Views are the central objective of this thesis. The purpose of a view is to obtain a simplification of a given transition system (TS). It is an independent TS derived from a given TS and represents a (simplified) view on the given one - hence the name. Thereby the original TS is retained.

3.1 Grouping Function

The conceptional idea of a view is to group states by some criteria and structure the rest of the system accordingly. To formalize the grouping we define a dedicated function.

Definition 3.1. Let $\mathrm{TS} = (S, Act, \longrightarrow, I, AP, L)$ be a transition system an M be and arbitrary set. We call any function $F : S \to M$ a *grouping function*.

Two states are grouped (should be Definition?) to a new state if and only if the grouping function maps them to the same value. The definition offers an easy way of defining groups of states. It is also very close to the actual implementation later on. The exact mapping depends on the desired grouping. In order to define a new set of states for the view, we define an equivalence relation R based on a given grouping function F.

Definition 3.2. Let F be a grouping function. We define the equivalence relation $R := \{(s_1, s_2) \in S \times S \mid F(s_1) = F(s_2)\}$

R is an equivalence relation because the equality relation is one. The property directly conveys to R. We observe that two states s_1, s_2 are grouped to a new state if and only if $(s_1, s_2) \in R$. This is the case if and only if $s_1, s_2 \in [s_1]_R = [s_2]_R$ where $[s_i]_R$ for $i \in \{1, 2\}$ denotes the equivalence class of R.

3.2 Formal Definition

The definition of a view is dependent on a given transition system and a grouping function F. We derive the equivalence relation R as in Definition 3.2 and use its equivalence classes $[s]_R$ $(s \in S)$ as states for the view. The rest of the transition system is structured accordingly.

Definition 3.3. Let $\mathrm{TS} = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and F a grouping function. A *view* TS_F is a transition system $(S', Act', \longrightarrow', I', L')$ that is derived from TS with the grouping function F where

- $S' = \{[s]_R \mid s \in S\}$
- $Act' = Act$
- $\longrightarrow' = \{([s_1]_R, \alpha, [s_2]_R) \mid \exists s_1 \in [s_1]_R \exists s_2 \in [s_2]_R : ([s_1]_R, \alpha, [s_2]_R) \in \longrightarrow\}$
- $I' = \{[s']_R \in S' \mid \exists s \in [s']_R : s \in I\}$

- $L' : S' \to \mathcal{P}(AP), [s]_R \mapsto \bigcup_{s \in [s]_R} \{L(s)\}$

and R is the equivalence relation according to Definition 3.2.

Note that the definition is in a most general form in the sense that if in a view a property accounts to one piece of some entity the whole entity receives the property i.e.

- $(s_1, \alpha, s_2) \in \longrightarrow \;\Rightarrow\; ([s_1]_R, \alpha, [s_2]_R) \in \longrightarrow'$
- $s \in I \;\Rightarrow\; [s]_R \in I'$
- $\forall s \in S : L(s) \in L'([s]_R)$

3.3 Composition of Views

In essence views are simplification generated from a transition system. It seems rather obvious that the composition of views is a very practical feature. Therefore in this chapter we will introduce, formalize and discuss different notions of compositions. All variants will ensure that the effect caused by each partaking views also takes effect in the composed view. Moreover it is to be generated in a way that the effect of each individual view can be reverted from the composed one. There may be restrictions in the order of removal.

3.3.1 Parallel Composition

One of the most uncomplicated ideas is to group states that match in the function value of all given grouping function s. This idea is parallel in the sense that a set of grouping function is combined to a new grouping function in one single step.

Definition 3.4. Let $F_1, F_2, \ldots, F_n$ be grouping function s. A *parallel composed grouping function* is a grouping function $F_{F_1||F_2||\ldots||F_n} : S \to M$, $s \mapsto (F_1(s), F_2(s), \ldots, F_n(s))$.

A view of such a grouping function is as usual created in in accordance with definition 3.3. Furthermore it is obvious that the effect of a view is considered in the new grouping function and hence the view. Since we define a new grouping function, the given ones and its views are retained.
The operator $||$ used in the definition is derived from the operator used in electric circuits, when the respective elements are connected in parallel.
If we want to speak about this grouping function in a general way, where it is only of importance that we refer to this type of composition and the given grouping function s are of no importance, we will denote a *parallel composition grouping function* with $F_{||}$.

4 View Examples

In this chapter we will introduce and discuss some view examples created by the author. Their purpose is to understand the idea and concept of a view and get to know some views that might be useful in real world applications.
When considering views we only want into account those that utilize properties of MDPs or that do computations that are also feasible on normal graphs but are of explicit relevance MDPs.

4.1 Utilizing MDP components

Views based on MDP components

4.1.1 Atomic Propositions

The *Atomic Propositions View* groups all states to a new state that have the same set of atomic propositions.

Definition 4.1. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system. The view $TS_{F_{AP}}$ is defined by its grouping function F_{AP} grouping function with $F_{AP} : S \to M, s \mapsto L(s)$.

The grouping function is exactly the labeling function i.e. for all $s \in S$ it is $F_{AP}(s) = L(s)$. So it is $F_{AP}(s_1) = F_{AP}(s_2) \iff L(s_1) = L(s_2)$. According to definition 3.2 for $\tilde{s} \in S$ it is $[\tilde{s}]_R = \{s \in S \mid L(s) = L(\tilde{s})\}$.

By this we obtain the view $TS_{F_{AP}}$ for a given transition system TS where: $S' = \bigcup_{s \in S}\{[s]_R\} = \bigcup_{a \in AP}\{\{s \in S \mid L(s) = a\}\}$. All other components are constructed as in definition 3.3.

tikz example

example from the database of max

4.1.2 Initial States

The *Initial State View* groups all initial states into one single state. All other states are left untouched.

Definition 4.2. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system. The view TS_{F_I} is defined by its grouping function F_I grouping function with $F_I : S \to M$ with

$$s \mapsto \begin{cases} \emptyset, & \text{if } s \in I \\ \{s\}, & \text{otherwise} \end{cases}$$

and $M := \{\emptyset\} \cup \{\{s\} \mid s \in S\}$.

For $s_1, s_2 \in S$ it is $F_I(s_1) = F_I(s_2)$ if and only if $s_1, s_2 \in I$ or $s_1 = s_2$. According to definition 3.2 it is

$$\begin{aligned} [s]_R &= \{s \in S \mid F(s) = \emptyset\} && \text{for } s \in I \text{ and} \\ [s]_R &= \{s \in S \mid F(s) = \{s\}\} = \{s\} && \text{for } s \notin I. \end{aligned}$$

By this we obtain the view TS_{F_I} for a given transition system TS where: $S' = \bigcup_{s\in S}\{[s]_R\} = \{s \in S \mid s \in I\} \cup \bigcup_{s\in S\setminus I}\{\{s\}\}$.

All other components are constructed as in definition 3.3.

4.1.3 Outgoing Actions

define outgoing transition and outgoing action The *OutAction View* groups states that share some property regarding their actions of the outgoing transitions. Several variants are feasible. In the following we will use the expression "outgoing action α " equivalent with "transition with outgoing action α ".

The most obvious variant to group states is to group states that *have* a given outgoing action.

Definition 4.3. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\exists\overrightarrow{\alpha}}}$ is defined by its grouping function $F_{\exists\overrightarrow{\alpha}} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \exists s' \in S : (s, \alpha, s') \in \longrightarrow \\ s, & \text{otherwise} \end{cases}$$

and $M := Act \cup S$.

For $s_1, s_2 \in S$ it is $F_{\exists\overrightarrow{\alpha}}(s_1) = F_{\exists\overrightarrow{\alpha}}(s_2)$ if and only if there exist $s_a, s_b \in S$ with $(s_1, \alpha, s_a), (s_2, \alpha, s_b) \in \longrightarrow$ (i.e. they have the same outgoing action α) or $s_1 = s_2$. In accordance with definition 3.2 it is

$$\begin{aligned} [s]_R &= \{s \in S \mid F_{\exists\overrightarrow{\alpha}}(s) = \alpha\} && \exists s' \in S : (s, \alpha, s') \in \longrightarrow \\ [s]_R &= \{s \in S \mid F_{\exists\overrightarrow{\alpha}}(s) = s\} = \{s\} && \text{otherwise} \end{aligned}$$

Thereby we obtain the view $TS_{F_{\exists\overrightarrow{\alpha}}}$ for a given transition system TS where $S' = \bigcup_{s\in S}\{[s]_R\} =: S_1 \cup S_2$ where

$$\begin{aligned} S_1 &:= \{s \in S \mid \exists s' \in S : (s, \alpha, s') \in \longrightarrow\} = \{s \in S \mid s \text{ has outgoing action } \alpha\} \\ S_2 &:= \bigcup_{s\in S\setminus S_1} \{s\}. \end{aligned}$$

Since actions are a very important part of transition systems as well as of its more powerful siblings MDPs and MCs it seems useful to further enhance this view and look at variants of it. Instead of only grouping states that only *have* outgoing actions we could also quantify the amount of times that action should be outgoing.

For example we could require that a given action has to be outgoing a minimum amount of times.

Definition 4.4. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{n\leq\overrightarrow{\alpha}}}$ is defined by its grouping function $F_{n\leq\overrightarrow{\alpha}} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \exists s_1, \ldots, s_n \in S : Q_{n\leq\overrightarrow{\alpha}}(s, s_1, \ldots, s_n) \\ s, & \text{otherwise} \end{cases}$$

where $M := Act \cup S$, $n \in \mathbb{N}$ is the minimum amount of times a transition with action α has to be outgoing in order to be grouped with the other states and

$$Q_{n \leq \overrightarrow{\alpha}}(s, s_1, \ldots, s_n) := ((s, \alpha, s_1), \ldots, (s, \alpha, s_n) \in \longrightarrow) \wedge |\{s_1, \ldots, s_n\}| = n$$

is a first order logic predicate.

The number and The predicate $Q_{n \leq \overrightarrow{\alpha}}(s, s_1, \ldots, s_n)$ requires that there are transitions with action α to n distinct states.

For $s_1, s_2 \in S$ it is $F_{n \leq \overrightarrow{\alpha}}(s_1) = F_{n \leq \overrightarrow{\alpha}}(s_2)$ if and only if there exist distinct $s_{a_1}, \ldots, s_{a_n} \in S$ and distinct $s_{b_1}, \ldots, s_{b_n} \in S$ so that $(s_1, \alpha, s_{a_1}), \ldots, (s_1, \alpha, s_{a_n}) \in \longrightarrow$ and $(s_2, \alpha, s_{b_1}), \ldots, (s_2, \alpha, s_{b_n}) \in \longrightarrow$ or $s_1 = s_2$. According to definition 3.2 it is

$$\begin{aligned} &[s]_R = \{s \in S \mid F_{n \leq \overrightarrow{\alpha}}(s) = \alpha\} && \text{if } Q_{n \leq \overrightarrow{\alpha}}(s, s_1, \ldots, s_n) \\ &[s]_R = \{s \in S \mid F_{n \leq \overrightarrow{\alpha}}(s) = s\} = \{s\} && \text{otherwise} \end{aligned}$$

By this we obtain the view $TS_{F_{n \leq \overrightarrow{\alpha}}}$ for a given transition system TS where $S' = \bigcup_{s \in S}\{[s]_R\} =: S_1 \cup S_2$ where

$$\begin{aligned} S_1 &:= \{s \in S \mid \exists s_1, \ldots, s_n \in S : Q_{n \leq \overrightarrow{\alpha}}(s, s_1, \ldots, s_n)\} \\ &= \{s \in S \mid \text{the action } \alpha \text{ is outgoing at least } n \text{ times}\} \text{ and} \\ S_2 &:= \bigcup_{s \in S \setminus S_1} \{s\}. \end{aligned}$$

In a similar fashion we define view that groups states where at most a certain number of times a given action is outgoing.

Definition 4.5. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\overrightarrow{\alpha} \leq n}}$ is defined by its grouping function $F_{\overrightarrow{\alpha} \leq n} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \forall s_1, \ldots, s_{n+1} \in S : Q_{\overrightarrow{\alpha} \leq n}(s, s_1, \ldots, s_{n+1}) \\ s, & \text{otherwise} \end{cases}$$

where $M := Act \cup S$, $n \in \mathbb{N}$ is the maximal number of times a transition with action α may be outgoing and

$$Q_{\overrightarrow{\alpha} \leq n}(s, s_1, \ldots, s_{n+1}) := ((s, \alpha, s_1), \ldots, (s, \alpha, s_{n+1}) \in \longrightarrow) \implies \bigvee_{\substack{i,j \in \{1,\ldots,n+1\} \\ i<j}} s_i = s_j$$

is a first order logic predicate.

It ensures that if there are one more than n outgoing transitions with an action α at least two of the states where the transitions end are in fact the same. Since this is required for all possible combinations of $n+1$ states by the grouping function, only states that have at most n outgoing actions will be assigned with α by the grouping function. The reasoning about the equality of the grouping function values, the obtained equivalence classes and the resulting set of states S' of the view is analogous to $TS_{F_{n\leq\overrightarrow{\alpha}}}$.

Since we already defined grouping functions and hence views for a required minimal and maximal amount of times an action has to be outgoing it is now easily possible to define a view that groups states where the amount of outgoing actions is at least n and at most m.

Definition 4.6. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{m\leq\overrightarrow{\alpha}\leq n}}$ is defined by its grouping function $F_{m\leq\overrightarrow{\alpha}\leq n} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \exists s_1, \ldots, s_m \in S : Q_{m\leq\overrightarrow{\alpha}}(s, s_1, \ldots, s_m) \\ & \text{and } \forall s_1, \ldots, s_{n+1} \in S : Q_{\overrightarrow{\alpha}\leq n}(s, s_1, \ldots, s_{n+1}) \\ s, & \text{otherwise} \end{cases}$$

where $M := Act \cup S$ and $m, n \in \mathbb{N}$ are the minimal and maximal number of transitions with action α in order for state to be grouped. The predicates $Q_{n\leq\overrightarrow{\alpha}}(s, s_1, \ldots, s_n)$ and $Q_{\overrightarrow{\alpha}\leq n}(s, s_1, \ldots, s_{n+1})$ are the predicates from the definitions 4.4 and 4.5 respectively.

We already know that for a given $s \in S$ the expressions $\exists s_1, \ldots, s_n \in S : Q_{n\leq\overrightarrow{\alpha}}(s, s_1, \ldots, s_n)$ and $\forall s_1, \ldots, s_{m+1} \in S : Q_{\overrightarrow{\alpha}\leq m}(s, s_1, \ldots, s_{m+1})$ from the definitions 4.4 and 4.5 require that s has minimal and maximal amount of outgoing transitions with an action α respectively. Hence the conjunction will be true for states where the amount of outgoing transitions with action α is element of the set $\{m, n+1, \ldots, n-1, n\}$. We will write for this that the number of outgoing actions is *in the span*.

For a given state s and action α we set

$$C_{s,\overrightarrow{\alpha}} := \exists s_1, \ldots, s_m \in S : Q_{m\leq\overrightarrow{\alpha}}(s, s_1, \ldots, s_m) \wedge \forall s_1, \ldots, s_{n+1} \in S : Q_{\overrightarrow{\alpha}\leq n}(s, s_1, \ldots, s_{n+1})$$

for convenience. $C_{s,\overrightarrow{\alpha}}$ is true if and only if the number of outgoing actions is in the span. For $s_1, s_2 \in S$ it is $F_{m\leq\overrightarrow{\alpha}\leq n}(s_1) = F_{m\leq\overrightarrow{\alpha}\leq n}(s_2)$ if and only if $C_{s_1,\overrightarrow{\alpha}} \wedge C_{s_2,\overrightarrow{\alpha}}$ or $s_1 = s_2$. Then its equivalence classes are

$$\begin{aligned} [s]_R &= \{s \in S \mid F_{m\leq\overrightarrow{\alpha}\leq n}(s) = \alpha\} && \text{for } s \in S \text{ and } C_{s,\overrightarrow{\alpha}} \text{ true} \\ [s]_R &= \{s \in S \mid F_{m\leq\overrightarrow{\alpha}\leq n}(s) = s\} = \{s\} && \text{otherwise} \end{aligned}$$

The new set of states S' of the view $TS_{F_{m\leq\overrightarrow{\alpha}\leq n}}$ is the union of the equivalence classes of equivalence relation R on the set of states S of the original transition system. Hence it is $S' = \bigcup_{s\in S} [s]_R =: S_1 \cup S_2$ where

$$\begin{aligned} S_1 &:= \{s \in S \mid F_{m\leq\overrightarrow{\alpha}\leq n}(s) = \alpha\} \\ &= \{s \in S \mid C_{s,\overrightarrow{\alpha}} \text{ true}\} \\ &= \{s \in S \mid \text{the action } \alpha \text{ is outgoing } m \text{ to } n \text{ times}\} \text{ and} \\ S_2 &:= \bigcup_{s\in S\setminus S_1} \{s\}. \end{aligned}$$

The views above could be expanded to several actions with requirements

Instead of making requirements about states and group them based on whether they meet these requirements it also possible to group states that are very similar or even identical in regard to their outaction. We consider this idea with the *OutActionsIdent View* in two variants: strong and weak (idententiy). Firstly we will consider the variant of strong identity.

Definition 4.7. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\overrightarrow{Act}(s)_=}}$ is defined by its grouping function $F_{\overrightarrow{Act}(s)_=} : S \to M$ with

$$s \mapsto \{(\alpha, n) \mid \alpha \in Act, n \text{ is the number of times that } \alpha \text{ is outgoing from } s\}$$

and $M := Act \times \mathbb{N}_0$.

The grouping function asserts to each state a set of pairs. Note that a pair is contained int the set for each action $\alpha \in Act$. In case there is no outgoing transition from state s with an action α it is $(\alpha, 0) \in F_{\overrightarrow{Act}(s)_=}$. For $s_1, s_2 \in S$ it is $F_{\overrightarrow{Act}(s)_=}(s_1) = F_{\overrightarrow{Act}(s)_=}(s_2)$ if and only if s_1 and s_2 are mapped to the same set of pairs. By definition 3.2 the obtained equivalence classes of R are

$$[s]_R := \{s \in S \mid F_{\overrightarrow{Act}(s)_=}(s) = \{(\alpha_1, n_1), \ldots, (\alpha_l, n_l)\}, l = |Act|\}$$

According to definition 3.3 the set $S' := \bigcup_{s \in S} [s]_R$ is the set of states of $TS_{F_{\overrightarrow{Act}(s)_=}}$. All other components of $TS_{F_{\overrightarrow{Act}(s)_=}}$ are as usual structured in accordance with the definition 3.3. As mentioned earlier a weak variant of the OutActionsIdent view is also conceivable.

Definition 4.8. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\overrightarrow{Act}(s)_\approx}}$ is defined by its grouping function $F_{\overrightarrow{Act}(s)_\approx} : S \to M$ with

$$s \mapsto \{\alpha \in Act \mid \exists s' \in S : (s, \alpha, s') \in \longrightarrow\}$$

and $M := Act$.

condition of image-set written in inconsistent style to strong identity. Swap strong and weak (order)?

The grouping function maps to the set of outgoing actions of a state and thereby discards information about the number of times an actions is outgoing. If an action is not outgoing from a state it is not contained in the set.

For $s_1, s_2 \in S$ it is $F_{\overrightarrow{Act}(s)_\approx}(s_1) = F_{\overrightarrow{Act}(s)_\approx}(s_2)$ if and only if they are mapped to the same set of actions. Hence the equivalence classes of R are

$$[\tilde{s}]_R = \{s \in S \mid F_{\overrightarrow{Act}(s)_\approx}(s) = F_{\overrightarrow{Act}(s)_\approx}(\tilde{s}) =: \{\alpha_1, \ldots, \alpha_l\}, l \in \mathbb{N}\}.$$

According to definition 3.3 the set $S' := \bigcup_{s \in S} [s]_R$ is the set of states of $TS_{F_{\overrightarrow{Act}(s)_\approx}}$.

4.1.4 Ingoing Actions

Analogously to Outgoing Actions views of utilizing ingoing actions are feasable. Since there is no difference apart from the definitions itself, we only provide the definitions.

Definition 4.9. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\exists\overleftarrow{\alpha}}}$ is defined by its grouping function $F_{\exists\overleftarrow{\alpha}} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \exists s' \in S : (s', \alpha, s) \in \longrightarrow \\ s, & \text{otherwise} \end{cases}$$

and $M := Act \cup S$.

Definition 4.10. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{n\leq\overleftarrow{\alpha}}}$ is defined by its grouping function $F_{n\leq\overleftarrow{\alpha}} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \exists s_1, \ldots, s_n \in S : Q_{n\leq\overleftarrow{\alpha}}(s, s_1, \ldots, s_n) \\ s, & \text{otherwise} \end{cases}$$

where $M := Act \cup S$, $n \in \mathbb{N}$ is the minimum amount of times a transition with action α has to be ingoing in order to be grouped with the other states and

$$Q_{n\leq\overleftarrow{\alpha}}(s, s_1, \ldots, s_n) := ((s_1, \alpha, s), \ldots, (s_n, \alpha, s) \in \longrightarrow) \wedge |\{s_1, \ldots, s_n\}| = n$$

is a first order logic predicate.

Definition 4.11. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\overleftarrow{\alpha}\leq n}}$ is defined by its grouping function $F_{\overleftarrow{\alpha}\leq n} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \forall s_1, \ldots, s_{n+1} \in S : Q_{\overleftarrow{\alpha}\leq n}(s, s_1, \ldots, s_{n+1}) \\ s, & \text{otherwise} \end{cases}$$

where $M := Act \cup S$, $n \in \mathbb{N}$ is the maximal number of times a transition with action α may be ingoing and

$$Q_{\overleftarrow{\alpha}\leq n}(s, s_1, \ldots, s_{n+1}) := ((s_1, \alpha, s), \ldots, (s_{n+1}, \alpha, s) \in \longrightarrow) \implies \bigvee_{\substack{i,j \in \{1,\ldots,n+1\} \\ i<j}} s_i = s_j$$

is a first order logic predicate.

Definition 4.12. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{m\leq\overleftarrow{\alpha}\leq n}}$ is defined by its grouping function $F_{m\leq\overleftarrow{\alpha}\leq n} : S \to M$ with

$$s \mapsto \begin{cases} \alpha, & \text{if } \exists s_1, \ldots, s_m \in S : Q_{m\leq\overleftarrow{\alpha}}(s, s_1, \ldots, s_m) \\ & \text{and } \forall s_1, \ldots, s_{n+1} \in S : Q_{\overleftarrow{\alpha}\leq n}(s, s_1, \ldots, s_{n+1}) \\ s, & \text{otherwise} \end{cases}$$

where $M := Act \cup S$ and $m, n \in \mathbb{N}$ are the minimal and maximal number of transitions with action α in order for state to be grouped. The predicates $Q_{n\leq\overleftarrow{\alpha}}(s, s_1, \ldots, s_n)$ and $Q_{\overleftarrow{\alpha}\leq n}(s, s_1, \ldots, s_{n+1})$ are the predicates from the definitions 4.9 and 4.10 respectively.

Definition 4.13. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\overleftarrow{Act(s)}(s)_=}}$ is defined by its grouping function $F_{\overleftarrow{Act(s)}(s)_=} : S \to M$ with

$$s \mapsto \{(\alpha, n) \mid \alpha \in Act, n \text{ is the number of times that } \alpha \text{ is ingoing from } s\}$$

and $M := Act \times \mathbb{N}_0$.

Definition 4.14. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system and $\alpha \in Act$. The view $TS_{F_{\overrightarrow{Act}(s)_\approx}}$ is defined by its grouping function $F_{\overrightarrow{Act}(s)_\approx} : S \to M$ with

$$s \mapsto \{\alpha \in Act \mid \exists s' \in S : (s, \alpha, s') \in \longrightarrow\}$$

and $M := Act$.

4.1.5 Parameters

The concept of parameters is not part of the definitions of neither transition systems, MCs or MDPs. Even though, it is of great importance in practical applications. Parameters are used to represent states (in more detail).

For example a MDP could be used to model a computer program with human interaction. Every state of the MDP refers an overall state of the program during execution time. In this state of the program, its variables will have specific values. Instead of only assigning a state $s \in S$ that refers to the state of the program, we want to retain the information about the values of the variables of the program. There is no component in MDPs that fullfills this purpose. Variables and its current values are not only relevant to computer programs but also other systems. Many of those other systems rely on some kind of global state during execution, which can be expressed with variables and values assigned to them. Parameters are used to store this information.

OLD: Since transitions systems MCs and MDPs in practice are used to model, analyze and check real world systems it is very practical to not only name states but also describe the properties of the state in more detail. For a basic notion parameters are to be imagined as a set of variables that may have different values in different states thereby describing the characteristics of the state more thoroughly. ADDED: In practice most often they arise very naturally for example as variables of a computer program that is to be modeled with a MDP.

Because of the vast importance in practical applications we will consider some views that utilize them. To do so and being able to describe them formally we define we will formalize the notion of parameters by considering them as a subset of the atomic propositions AP that is assigned a value by a function.

Definition 4.15. The set $Par \subseteq AP$ is called *parameters*.

Definition 4.16. Let M be an arbitrary set. The function $ParEval : S \times Par \to M$ is called parameter evaluation function.

Most of the time we will use $ParEval$ to refer to the parameter evaluation function. When we speak about the value of a parameter in a state we refer to the image of $ParEval$ for that state and value. The set M is arbitrary so that arbitrary values can be assigned to a parameter. Speaking in terms of computer science and programming this loosens as an example the restriction of only being able to assign numbers and no booleans.

The most apparent idea for a view utilizing parameters is to group states that meet some requirement regarding the values of the parameters.

state has parameter not here uptil now because probably not used

Definition 4.17. Let

- $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system,
- $x \in Par \subseteq AP$ and
- $ParEval(s, x) = a$ where $s \in S$.

The view $TS_{F_{x=a}}$ is defined by its grouping function $F_{x=a} : S \to M$ with

$$s \mapsto \begin{cases} a, & \text{if } ParEval(s, x) = a \\ s, & \text{otherwise} \end{cases}$$

where $M := S \cup \{a\}$.

The view $TS_{F_{x=a}}$ groups states that share the same value for a given parameter. For s_1, s_2 it is $F_{x=a}(s_1) = F_{x=a}(s_2)$ if and only if $ParEval(s_1, x) = ParEval(s_2, x)$ or $s_1 = s_2$. The obtained equivalence classes are

$$[s]_R = \{s \in S \mid ParEval(s, x) = a\}$$
$$[s]_R = \{s \in S \mid F_{x=a}(s) = s\} = \{s\}$$

The set of states S' of $TS_{F_{x=a}}$ is the union of the equivalence classes of R. It is $S' = \bigcup_{s \in S} [s]_R =: S_1 \cup S_2$ where

$$\begin{aligned} S_1 &:= \{s \in S \mid F_{x=a}(s) = a\} \\ &= \{s \in S \mid ParEval(s, x) = a\} \text{ and} \\ S_2 &:= \bigcup_{s \in S \setminus S_1} \{s\}. \end{aligned}$$

If states are to be grouped with the requirement of several parameters equaling specified values this can be achieved by using parallel composition. disjunctive??

The view above reduces the transition system in a very precise but also manual manner, because it not only dictates the parameter but also its value. A more general approach is to stipulate only the parameter but not its value. This way states will be grouped that have the same value for that parameter with no regard to the actual value of that parameter. This idea could be achieved with a view based on the grouping function $F_{F_{x=a_1} || \ldots || F_{x=a_n}}$ with $ParEval(S, x) = \{a_1, \ldots, a_n\}$ and $|ParEval(S, x)| = n$. This grouping function just groups on every possible value for the parameter x. Since this is not very practical we define a view that achieves this result in a more direct and more efficient way.

Definition 4.18. Let $TS = (S, Act, \longrightarrow, I, AP, L)$ be a transition system. The view $TS_{F_{ParEval(s,x)}}$ is defined by its grouping function $F_{ParEval(s,x)} : S \to M$ with

$$s \mapsto ParEval(s, x)$$

and $M := ParEval(S, x)$.

With this grouping function we directly map to the value of the parameter. Hence for $s_1, s_2 \in S$ it is $F_{ParEval(s,x)}(s_1) = F_{ParEval(s,x)}(s_2)$ if and only if they are mapped to the value $a \in M$. Hence the equivalence classes of R are

$$[\tilde{s}]_R = \{s \in S \mid ParEval(s) = ParEval(\tilde{s})\}.$$

According to definition 3.3 the set $S' := \bigcup_{s \in S} [s]_R$ is the set of states of $TS_{F_{ParEval(s,x)}}$.

4.1.6 Distance

4.2 Computation on MDP-Graph

4.3 Comparison of the Examples

5 Outlook

References