The M in CoPaR

From Partition Refinement to Minimization

Hans-Peter Deifel

Oberseminar

09.06.2020

The M in CoPaR

From Partition Refinement to Minimization

Hans-Peter Deifel

Oberseminar

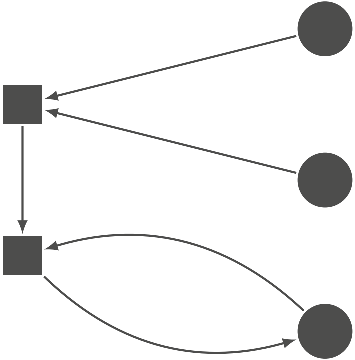
09.06.2020

work in progress

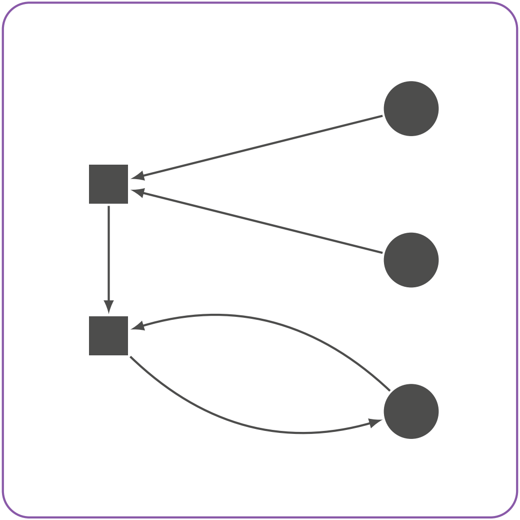
CoPaR

Coalgebraic Partition Refiner

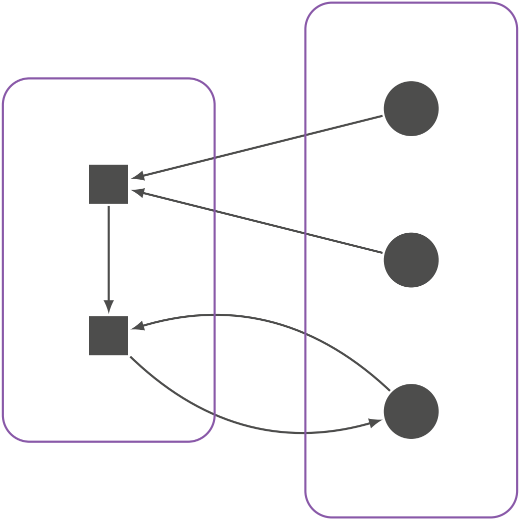
Partition Refinement



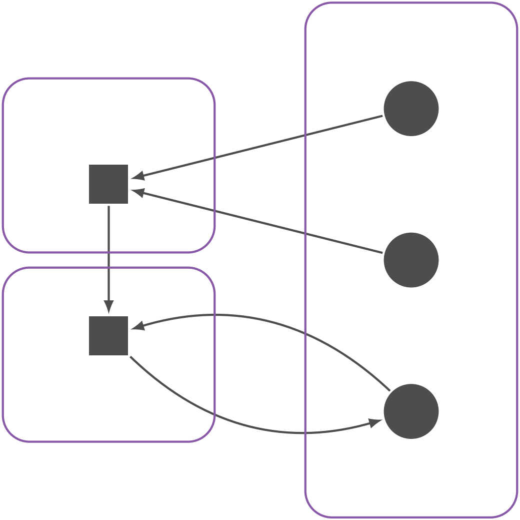
Partition Refinement



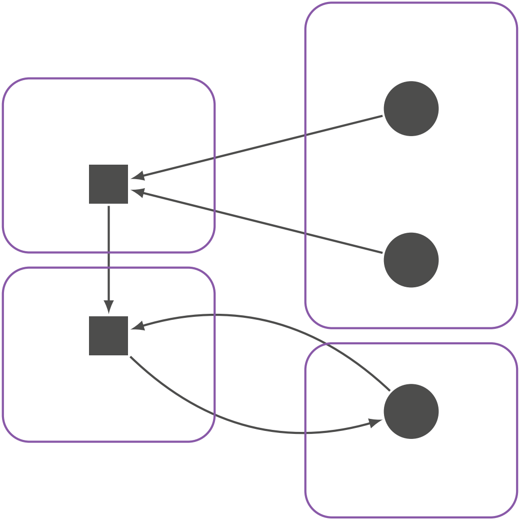
Partition Refinement



Partition Refinement



Partition Refinement



Partition Refinement

Different partition refinement algorithms:

- Pajge-Tarjan
- Hopcroft
- Markov-Chain lumping
- Color refinement

$\Rightarrow$ Used in countless minimization tools

Partition Refinement

Different partition refinement algorithms:

- Pajge-Tarjan
- Hopcroft
- Markov-Chain lumping
- Color refinement

$\Rightarrow$ Used in countless minimization tools

CoPaR

Coalgebraic Partition Refiner

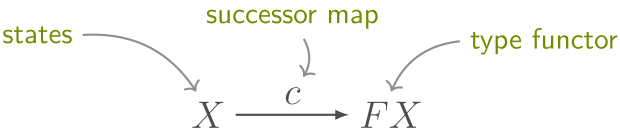
CoM

Coalgebraic Minimizer

Coalgebras

states — successor map — type functor

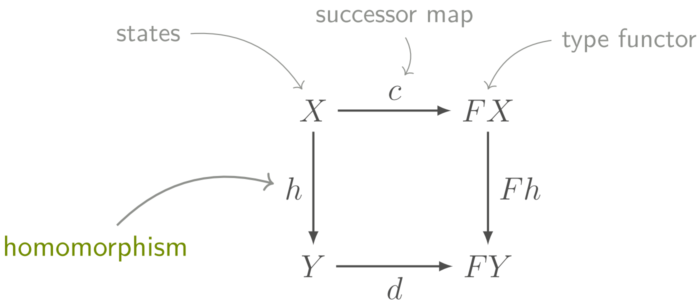
$$X \xrightarrow{c} FX$$



Type Functor $F\colon \mathsf{Set} \to \mathsf{Set}$

$FX =$	$\mathcal{P}(A \times X)$	$2 \times X^A$	$\mathbb{R}^{(X)}$	
	Labelled Transition Systems	Deterministic Automata	Markov Chains	$\dots$

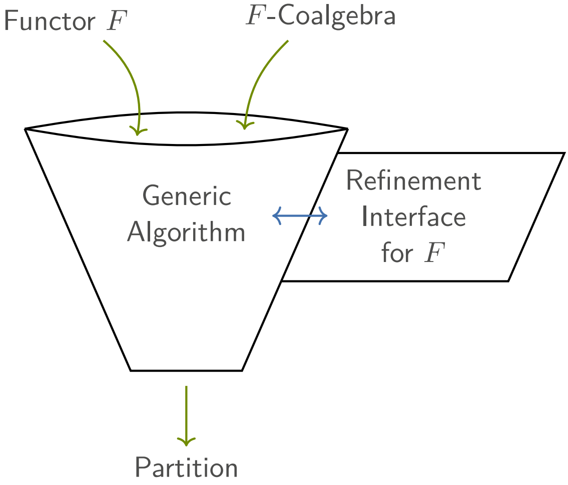
Behavioural Equivalence



Identified by Coalgebra Homomorphism

$FX =$	$\mathcal{P}(A \times X)$	$2 \times X^A$	$\mathbb{R}^{(X)}$	
	Bisimilarity	Language Equivalence	Weighted Bisimilarity	$\dots$

Coalgebraic Partition Refinement



Functor Encoding

Encoding for Functor F

- Abstract Type A of labels
- Encoding map

Bags

$$\flat_X : FX \to \mathcal{B}(A \times X)$$

Example: $\mathcal{P}(-)$

- $A = 1$
- $\flat_X(\{x_1, x_2, x_3\}) = \{(*, x_1), (*, x_2), (*, x_3)\}$

Functor Encoding

Encoding for Functor F

- Abstract Type A of labels
- Encoding map

Bags

$$\flat_X : FX \to \mathcal{B}(A \times X)$$

Example: $\mathbb{R}^{(-)}$

- $A = \mathbb{R}$
- $\flat_X(f) = \{(r, x) \mid x \in X, f(x) = r \neq 0\}$

Functor Encoding

Encoding for Functor F

- Abstract Type A of labels
- Encoding map

Bags

$$\flat_X : FX \to \mathcal{B}(A \times X)$$

Not Natural

Let $f \in \mathbb{R}^2$, $\qquad f(0) = 0.5, \quad f(1) = 0.5$

$$\begin{array}{ccc}
f & \overset{\flat_2}{\longmapsto} & \{(0.5, 0), (0.5, 1)\} \\
\mathbb{R}^{(!)} \big\downarrow & & \big\downarrow \mathcal{B}(\mathrm{id}, !) \\
(* \mapsto 1.0) & \overset{\flat_1}{\not\longmapsto} & \{(0.5, 1), (0.5, 1)\}
\end{array}$$

Functor Encoding

Encoding for Functor F

- Abstract Type A of labels
- Encoding map

Bags

$$\flat_X : FX \to \mathcal{B}(A \times X)$$

Refinement Interface

Type W (abstract, could be ints, reals, trees, ...)

$\texttt{init}\colon F1 \times \mathcal{B}A \to W$
$\texttt{update}\colon \mathcal{B}A \times W \to W \times F3 \times W$

Syntax

Input

```
DX

q1: {q2: 0.5, q3: 0.5}
q2: {q1: 0.5, q3: 0.5}
q3: {q3: 1}
```

Syntax

Input

```
DX

q1: {q2: 0.5, q3: 0.5}
q2: {q1: 0.5, q3: 0.5}
q3: {q3: 1}
```

Output

Block 0: q1, q2, q3

Syntax

Input

```
DX

q1: {q2: 0.5, q3: 0.5}
q2: {q1: 0.5, q3: 0.5}
q3: {q3: 1}
```

Cooler Output

```
DX

q1: {q1: 1}
```

Minimization

Two tasks:

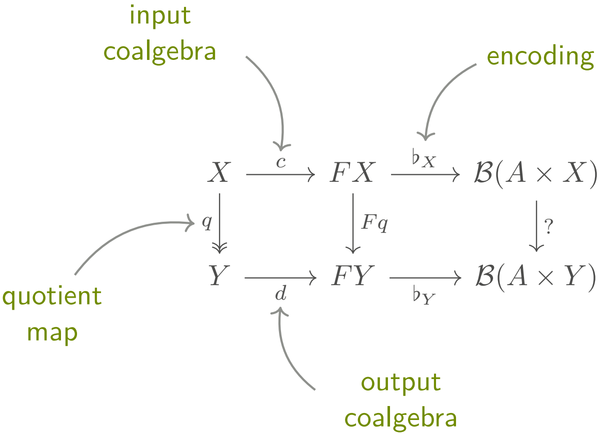
- Computing the (encoding of) the quotient coalgebra
- Removing unreachable states

Minimization

Two tasks:

- Computing the (encoding of) the quotient coalgebra
- Removing unreachable states

Quotient Encoding: Goal



Solution: Minimization Interface

$$\texttt{merge}\colon \mathcal{B}(A) \to \mathcal{B}(A)$$

$$\begin{array}{ccc} FX & \xrightarrow{\mathrm{fil}_S \cdot \flat} & \mathcal{B}(A) \\ {\scriptstyle F\chi_S}\downarrow & & \downarrow{\scriptstyle \texttt{merge}} \\ F2 & \xrightarrow{\mathrm{fil}_{\{1\}} \cdot \flat} & \mathcal{B}(A) \end{array}$$

and

$$\begin{array}{ccc} & & \mathcal{B}(A) \\ & \overset{\{\}}{\nearrow} & \\ 1 & & \downarrow{\scriptstyle \texttt{merge}} \\ & \underset{\{\}}{\searrow} & \\ & & \mathcal{B}(A) \end{array}$$

$$\mathrm{fil}_S(t)(a) = \sum_{x \in S} t(a, x)$$

Minimization Interface: Examples

Powerset

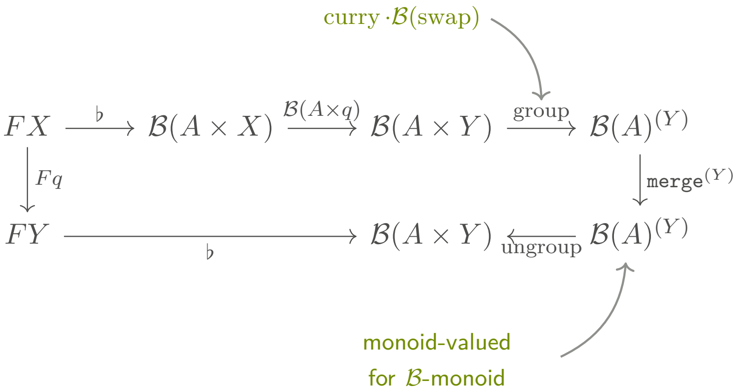
$$\texttt{merge}(\ell)(*) = \min(1, \ell(*))$$

Monoid-Valued

$$\texttt{merge}(\ell) = \begin{cases} \{\Sigma\ell\} & \Sigma\ell \neq 0 \\ \{\} & \text{otherwise} \end{cases}$$

Polynomial

$$\texttt{merge} = \mathrm{id}$$


curry ·ℬ(swap)
FX
♭
ℬ(A × X)
ℬ(A×q)
ℬ(A × Y)
group
ℬ(A)^(Y)
Fq
merge^(Y)
FY
♭
ℬ(A × Y)
ungroup
ℬ(A)^(Y)
monoid-valued
for ℬ-monoid

Nope! We additionally need the following assumption on $\flat$:

$$\begin{array}{ccccc} FX & \xrightarrow{\flat_X} & \mathcal{B}(A \times X) & & \\ \Big\downarrow {\scriptstyle F\chi_{\{x\}}} & & & \searrow^{\mathrm{fil}_{\{x\}}} & \\ & & & & \mathcal{B}(A) \\ & & & \nearrow_{\mathrm{fil}_{\{1\}}} & \\ F2 & \xrightarrow{\flat_2} & \mathcal{B}(A \times 2) & & \end{array}$$

Encoding Assumption: Counterexample

An encoding for Powerset

$A = 1 + 1$

$$\flat_2(\{x_1, \ldots, x_n\}) = (\{(\mathrm{inj}_1 *, x_1) \ldots, (\mathrm{inj}_1 *, x_n)\})$$
$$\flat_{X \neq 2}(\{x_1, \ldots, x_n\}) = (\{(\mathrm{inj}_2 *, x_1) \ldots, (\mathrm{inj}_2 *, x_n)\})$$

$\Rightarrow$ Lawful `merge` possible!

Encoding Assumption: Counterexample

An encoding for Powerset

$A = 1 + 1$

$$\flat_2(\{x_1, \ldots, x_n\}) = (\{(\mathrm{inj}_1 *, x_1) \ldots, (\mathrm{inj}_1 *, x_n)\})$$
$$\flat_{X \neq 2}(\{x_1, \ldots, x_n\}) = (\{(\mathrm{inj}_2 *, x_1) \ldots, (\mathrm{inj}_2 *, x_n)\})$$

$\Rightarrow$ Lawful `merge` possible!

But: Ruled out by assumption

Victory!

$$
\begin{array}{ccccccc}
FX & \xrightarrow{\ \flat\ } & \mathcal{B}(A \times X) & \xrightarrow{\mathcal{B}(A\times q)} & \mathcal{B}(A \times Y) & \xrightarrow{\text{group}} & \mathcal{B}(A)^{(Y)} \\
{\scriptstyle Fq}\big\downarrow & & & & & & \big\downarrow{\scriptstyle \texttt{merge}^{(Y)}} \\
FY & & \xrightarrow[\flat]{\hspace{6em}} & & \mathcal{B}(A \times Y) & \xleftarrow[\text{ungroup}]{} & \mathcal{B}(A)^{(Y)}
\end{array}
$$

Victory!

$$\begin{array}{ccccccc} FX & \xrightarrow{\flat} & \mathcal{B}(A\times X) & \xrightarrow{\mathcal{B}(A\times q)} & \mathcal{B}(A\times Y) & \xrightarrow{\text{group}} & \mathcal{B}(A)^{(Y)} \\ {\scriptstyle Fq}\downarrow & & & & & & \downarrow{\scriptstyle \texttt{merge}^{(Y)}} \\ FY & & \xrightarrow[\flat]{\qquad\qquad\qquad} & & \mathcal{B}(A\times Y) & \xleftarrow[\text{ungroup}]{} & \mathcal{B}(A)^{(Y)} \end{array}$$

Minimization

Two tasks:

- Computing the (encoding of) the quotient coalgebra
- Removing unreachable states

Minimization

Two tasks:

- Computing the (encoding of) the quotient coalgebra
- Removing unreachable states

Reachability: Problems

"Two tasks" all the way down:

- Pointed Coalgebras
- Computing the Reachable Subcoalgebra

Reachability: Problems

"Two tasks" all the way down:

- Pointed Coalgebras
- Computing the Reachable Subcoalgebra

Reachability: Problems

"Two tasks" all the way down:

- Pointed Coalgebras $\Rightarrow$ Simple matter of programming™
- Computing the Reachable Subcoalgebra

Reachability: Problems

"Two tasks" all the way down:

- Pointed Coalgebras $\Rightarrow$ Simple matter of programming™
- Computing the Reachable Subcoalgebra

Computing the Reachable Subcoalgebra

In Set: Standard graph search on canonical graph

In Set: Standard graph search on canonical graph

Canonical Graph

$$\tau_X \colon FX \to PX$$

$$\tau_X(t) = \{x \in X \mid 1 \xrightarrow{t} FX \text{ does not factorize through } F(X \setminus \{x\}) \xrightarrow{Fi} FX\}$$

For coalgebra: $C \xrightarrow{c} FC \xrightarrow{\tau_C} \mathcal{P}C$

Goal

We want:

- Canonical graph of original coalgebra and
- canonical graph of encoded coalgebra

to be the same!

$$\begin{array}{ccccc} C & \xrightarrow{c} & FC & \xrightarrow{\tau} & \mathcal{P}C \\ & & \downarrow{\scriptstyle \flat} & \nearrow{\scriptstyle \tau} & \\ & & \mathcal{B}(A \times C) & & \end{array}$$

Direction: $\tau(t) \subseteq \tau(\flat(t))$

Questment: Encoding contains canonical graph?

Direction: $\tau(t) \subseteq \tau(\flat(t))$

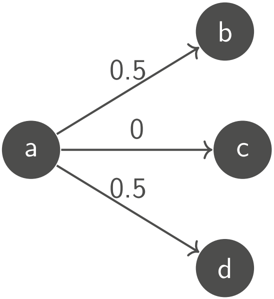
Questment: Encoding contains canonical graph?

$\Rightarrow$ Should be true™

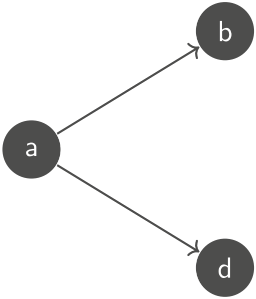
Counterexample for $\tau(t) \supseteq \tau(\flat(t))$

Functor: $\mathbb{R}^{(-)}$

Encoded Coalgebra



Canonical Graph



Requirement: Sub-Naturality

Encoding must be sub-natural:

$$
\begin{array}{ccccc}
X & \qquad & FX & \xrightarrow{\flat_X} & \mathcal{B}(A \times X) \\
{\scriptstyle f}\big\downarrow & & {\scriptstyle Ff}\big\downarrow & & \big\downarrow{\scriptstyle B(A \times f)} \\
Y & \qquad & FY & \xrightarrow{\flat_Y} & \mathcal{B}(A \times Y)
\end{array}
$$

Requirement: Sub-Naturality

Encoding must be sub-natural:

$$
\begin{array}{ccccc}
X & \qquad & FX & \xrightarrow{\flat_X} & \mathcal{B}(A \times X) \\
f \big\downarrow & & Ff \big\downarrow & & \big\downarrow B(A \times f) \\
Y & & FY & \xrightarrow{\flat_Y} & \mathcal{B}(A \times Y)
\end{array}
$$

$\Rightarrow$ now works!

Question: Connection?

Assumption 1

$$\begin{array}{ccccc} FX & \xrightarrow{\flat_X} & \mathcal{B}(A\times X) & & \\ & & & \searrow^{\mathrm{fil}_{\{x\}}} & \\ {\scriptstyle F\chi_{\{x\}}}\big\downarrow & & & & \mathcal{B}(A) \\ & & & \nearrow_{\mathrm{fil}_{\{1\}}} & \\ F2 & \xrightarrow{\flat_2} & \mathcal{B}(A\times 2) & & \end{array}$$

Assumption 2

$$\begin{array}{ccccccc} X & \qquad & FX & \xrightarrow{\flat_X} & \mathcal{B}(A\times X) \\ {\scriptstyle f}\big\downarrow & & {\scriptstyle Ff}\big\downarrow & & \big\downarrow{\scriptstyle B(A\times f)} \\ Y & & FY & \xrightarrow{\flat_Y} & \mathcal{B}(A\times Y) \end{array}$$