An Inequality for the Complexity of Bisimilarity Computations and a Fibrational Algorithm

Takahiro Sanada

joint work with Ryota Kojima, Yuichi Komorida, Koko Muroya, and Ichiro Hasuo

RIMS, Kyoto University

Introduction

A state-based system is a system with the set of states and the state transition rules. It is beneficial to reduce the number of states while keeping its properties. The bisimilarity on the state set identifies states that have essentially the same behaviour. Thus, to minimise state size of a system, we would like to compute the bisimilarity relation efficiently.

Our contibutions are devided two parts.

- We formalise and generalise Hopcroft's technique to bound the time complexity of partition refinement algorithms.
- We introduce a fibrational partition refinement algorithm that computes bisimilarity for various coalgebras.

The intriguing point is that a categorical notion and the inequality allow us to evaluate complexity of abstract algorithms.

Coalgebras as a State-Based System

Let $F\colon \mathrm{Set} \to \mathrm{Set}$ be a functor. An F-coalgebra

transition — *F specifies branching*

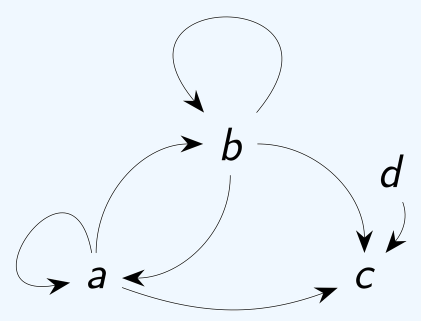
$$\boxed{c}\colon \boxed{C} \to \boxed{F}\,C \qquad \text{in Set}$$

the set of states

can be seen as a state-besed system.

If $F = \mathcal{P}$ (the powerset functor), then a coalgebra

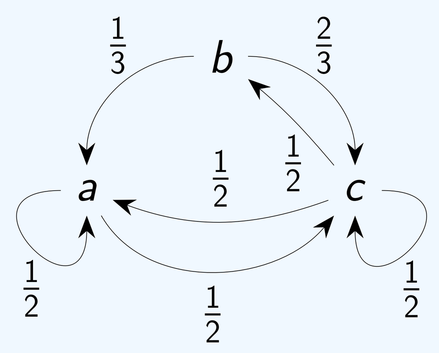
$$\begin{aligned} c\colon \{a,b,c,d\} &\to \mathcal{P}(\{a,b,c,d\}) \\ a &\mapsto \{a,b,c\} \\ b &\mapsto \{a,b,c\} \\ c &\mapsto \varnothing \\ d &\mapsto \{c\} \end{aligned} \tag{1}$$



is a transition system.

If $F = \mathcal{D}$ (the probabilisitic distribution functor), then a coalgebra

$$\begin{aligned} c\colon \{a,b,c\} &\to \mathcal{D}(\{a,b,c\}) \\ a &\mapsto \left(a \mapsto \tfrac{1}{2},\ b \mapsto 0,\ c \mapsto \tfrac{1}{2}\right) \\ b &\mapsto \left(a \mapsto \tfrac{1}{3},\ b \mapsto 0,\ c \mapsto \tfrac{2}{3}\right) \\ c &\mapsto \left(a \mapsto \tfrac{1}{4},\ b \mapsto \tfrac{1}{4},\ c \mapsto \tfrac{1}{2}\right) \end{aligned} \tag{2}$$



is a Markov chain.

The Bisimilarity Relations as a Coinductive Predicate

Recall the coalgbra (1). The behaviour of the two states a and b is the same. The state b can simulate the transition of the state a, and vice versa. We identify a with b, and obtain the smaller coalgebra: $a' \longrightarrow c' \longleftarrow d'$.

Formally, for a transition system $c\colon C \to \mathcal{P}C$, a relation $R \subseteq C \times C$ is a bisimulation if, for every $(x, y) \in R$, the following hold,

- If $x \to x'$, there exists y' such that $y \to y'$ and $(x', y') \in R$.
- If $y \to y'$, there exists x' such that $x \to x'$ and $(x', y') \in R$.

$$\begin{tikzcd} x \arrow[r] \arrow[d, dash, "R"'] & x' \arrow[d, dash, "R"] \\ y \arrow[r, dotted] & \exists y' \end{tikzcd} \qquad \begin{tikzcd} x \arrow[r, dotted] \arrow[d, dash, "R"'] & \exists x' \arrow[d, dash, "R"] \\ y \arrow[r] & y' \end{tikzcd}$$

The bisimilarity relation B is the largeset bisimulation, $B = \bigcup_{B:\ \text{bisimulation of } c} R$. We can minimise a transition system $c\colon C \to \mathcal{P}C$ by taking the quotient C/B by B.

The bisimilarity relation can be seen as a coinductive predicate on C. Let EqRel be the category whose objects are (S, R) where S is a set and R is an equivalence relation and a morphism $(S, R) \to (S', R')$ is a map $f\colon S \to S'$ such that $(x, y) \in R \longrightarrow (fx, fy) \in R'$. Consider the fibration $p\colon \mathrm{EqRel} \to \mathrm{Set}$. Each fibre category EqRel_S is a complete lattice. We can construct a lifting $\overline{\mathcal{P}}\colon \mathrm{EqRel}_C \to \mathrm{EqRel}_{\mathcal{P}C}$ of $\mathcal{P}$ such that the greatest fixed point $\nu(c^* \circ \overline{\mathcal{P}})$ is B.

$$\begin{tikzcd} \mathrm{EqRel} \arrow[d, "p"] \\ \mathrm{Set} \end{tikzcd} \qquad \begin{tikzcd} \mathrm{EqRel}_C \arrow[r, bend left, "\overline{\mathcal{P}}"] & \mathrm{EqRel}_{\mathcal{P}C} \arrow[l, bend left, "c^*"] \\ C \arrow[r, "c"'] & \mathcal{P}C \end{tikzcd}$$

The bisimilarity relation for markov chain $c\colon C \to \mathcal{D}C$ or other systems $c\colon C \to FC$ is also defined using appropriate lifting $\overline{\mathcal{D}}$ or $\overline{F}$.

Hopcroft's Inequality

We prove an inequality about a rooted finite tree with a weight function. For a rooted finite tree T, a weight function of T is a function

$$w\colon \boxed{V(T)} \to \mathbb{N} \quad \text{s.t.} \quad \forall v.\ \boxed{\sum_{u \in \mathrm{ch}(v)} w(u)} \le w(v).$$

the set of vertecies of T — *the sum of weights of children of v*

A weight function w is tight if the above inequality is equality.

Lemma (horizontal/vertical sum of weights)

For a subset S of the set $E(T)$ of edges of T, we have

the horizontal sum of weights — *"$=$" iff w is tight* — *the vertical sum of weights*

$$\boxed{\sum_{v \in V(T)} \boxed{\sum_{\substack{u \in \mathrm{ch}(v) \\ (v,u) \notin S}} w(u)}} \ \boxed{\ge}\ \boxed{\sum_{l \in L(T)} \boxed{\left|\mathrm{path}(r, l) \setminus S\right|} \cdot w(l)}. \tag{3}$$

the sum of weights of chldren of v that is not connected by a edge in S — *the number of edges from the root to the leaf l that is not in S*

A heavy child choice for a weight w is a function

$$h_{(-)}\colon V(T) \setminus L(T) \to V(T) \quad \text{such that}$$

$$\boxed{h_v \in \mathrm{ch}(v)} \quad \text{and} \quad \boxed{w(h_v) = \max_{u \in \mathrm{ch}(v)} w(u)}.$$

h_v is child of v — *h_v is the heaviest among the children of v*

For a heavy child choice h, we define

$$S_h := \{(v, h_v) \mid v \in V(T) \setminus L(T)\} \quad \subseteq E(T).$$

The following lemma is well known as Hopcroft's trick.

Lemma (Hopcroft's trick)

For a heavy child choice h for a weight w, we have

$$\boxed{\left|\mathrm{path}(r, v) \setminus S_h\right|} \le \log_2 w(r) - \log_2 w(v). \tag{4}$$

the number of edges from the root to v that is not in S_h

Combining above two lemmas, we obtain the following result.

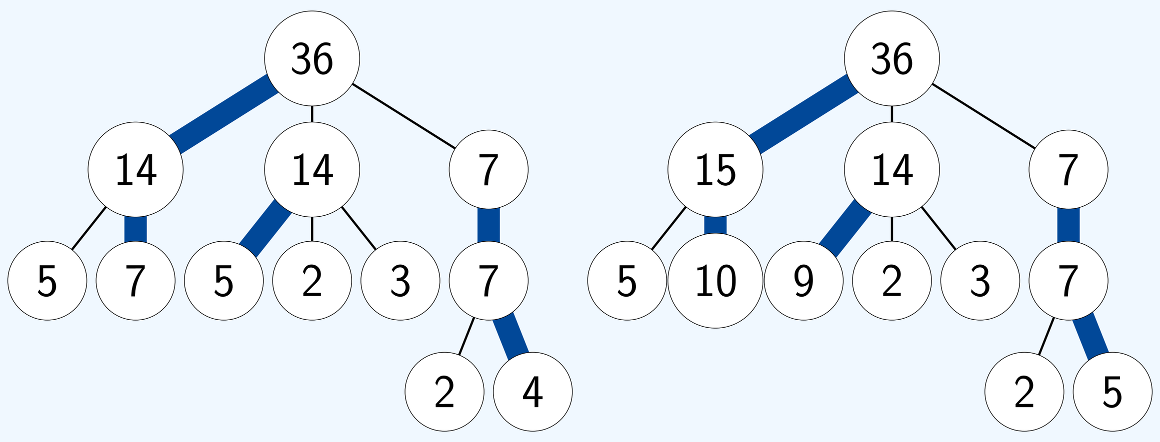
Theorem (Hopcroft's inequality)

For a heavy child choice h for a weight w, we have

$$\boxed{\sum_{v \in V(T)} \sum_{\substack{u \in \mathrm{ch}(v) \\ (v,u) \notin S_h}} w(u)} \le \boxed{w(r) \log_2 w(r)} - \boxed{\sum_{\substack{l \in L(T) \\ w(l) \neq 0}} w(l) \log_2 w(l)}. \tag{5}$$

the horizontal sum of weights — *determined by the root* — *determined by the leaves*

Note that, to prove the above theorem, we cannot directly apply the horizontal/vertical sum lemma because the direction of inequality is opposite. To resolve this problem, we observe that we can get a tight weight function from a non-tight one.



We call the conversion of weight function tightening.

Corollary

Let $t\colon V(T) \to \mathbb{N}$ be a map and K be a number. If $t(v) \le K \sum_{\substack{u \in \mathrm{ch}(v) \\ (v,u) \notin S_h}} w(u)$ for every $v \in V(V)$, then we have $\sum_{v \in V(V)} t(v) \le Kw(r) \log_2 w(r)$.

The above corollary says that if a step-by-step tree generation algorithm takes $\mathcal{O}(K \sum_{\substack{u \in \mathrm{ch}(v) \\ (v,u) \notin S_h}} w(u))$ time to generate children of a current leaf, the total time to generate a whole tree is bounded by $\mathcal{O}(Kw(r) \log w(r))$.

The fibrational coalgebraic partition refinement algorithm

Let $p\colon \mathbb{E} \to \mathbb{C}$ be a fibration, a functor $F\colon \mathbb{C} \to \mathbb{C}$, a lifting $\overline{F}\colon \mathbb{E} \to \mathbb{E}$ of F, and a weight function $w\colon \{\text{a subobject of } C\} \to C$. If they satisfy appropriate conditions, then the following algorithm computes the greatest fixed point $\nu(c^* \circ \overline{F})$-partitioning, where R-partitioning for $R \in \mathbb{E}$ is a generalised notion of a family of equivalence classes of an equivalence relation R.

Input: A coalgebra $c\colon C \to FC$ in $\mathbb{C}$.

Output: A mono-sink $\{\kappa_i\colon C_i \rightarrowtail C\}_{i \in I}$ for some I.

$T := \{\epsilon\} \subset \mathbb{N}^*$
$C_\epsilon := C$
$C_\epsilon^{\mathrm{cl}} := 0$
while there is $\rho \in L(T)$ such that $C_\rho^{\mathrm{cl}} \neq C_\rho$ **do**
    $Q := \bigsqcup_{\sigma \in L(T)} (\kappa_\sigma)_* (\top_{C_\sigma})$
    Choose a leaf $\rho \in L(T)$ such that $C_\rho^{\mathrm{cl}} \neq C_\rho$
    $R_\rho := (c \circ \kappa_\rho)^* (\overline{F}(Q))$
    if $R_\rho = \top_{C_\rho}$ **then**
        $C_\rho^{\mathrm{cl}} := C_\rho$
        continue
    Take an R_ρ-partitioning $\{\kappa_{\rho,k}\colon C_{\rho k} \rightarrowtail C_\rho\}_{k \in \{0,\dots,n_\rho\}}$
    Choose $k_0 \in \{0, \dots, n_\rho\}$ s.t. $w(C_{\rho k_0}) = \max_{k \in \{0,\dots,n_\rho\}} w(C_{\rho k})$
    $\textsc{MarkDirty}$
    $T := T \cup \{\rho 0, \dots, \rho n_\rho\}$
return $\{\kappa_\sigma\colon C_\sigma \rightarrowtail C\}_{\sigma \in L(T)}$

procedure $\textsc{MarkDirty}$
    for $k \in \{0, \dots, n_\rho\}$ **do**
        $C_{\rho k}^{\mathrm{cl}} := C_{\rho k}$
    Let B be the pullback of the following diagram:

$$\begin{tikzcd} B \arrow[r, tail] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & C \arrow[d, "c"] \\ F\left(C_{\rho k_0} \cup \left(\bigcup_{\sigma \in L(T) \setminus \{\rho\}} C_\sigma\right)\right) \arrow[r, tail] & FC \end{tikzcd}$$

    for $\tau \in L(T \cup \{\rho 0, \dots, \rho n_\rho\})$ **do**
        $C_\tau^{\mathrm{cl}} := C_\tau^{\mathrm{cl}} \cap B$

Complexity Analysis

From the corollary in the left column, we obtain the following proposition.

Proposition

If each call of the procedure $\textsc{MarkDirty}$ in the algorihtm takes $\mathcal{O}(K \sum_{k \in \{0,\dots,n_\rho\}} w(C_{\rho k}))$ time for some K, then the total time taken by the repeated calls of $\textsc{MarkDirty}$ is $\mathcal{O}(Kw(C) \log w(C))$.

We can instantiate the fibration $p\colon \mathbb{E} \to \mathbb{C}$ with $\mathrm{EqRel} \to \mathrm{Set}$, and optimise the algorithm. There are two options for weight functions. One is the size function $|-|$, and the other is the function pred that gives the number of predecessors in a set. If we choose former as a weight function, the obtained algorithm is essentially the same as Jacobs and Wißmann's algorihtm. By some argument, we can implement the algorithm so that it satisfies the premise of the above proposition.

Proposition

If $p\colon \mathbb{E} \to \mathbb{C}$ is $\mathrm{EqRel} \to \mathrm{Set}$, then we can implement the algorithm so that it takes $\mathcal{O}(Kw(C) \log w(C))$ time to compute $\textsc{MarkDirty}$ throughout the algorithm.

For example, when $F = \mathcal{P}$ and the weight function is pred, the time complexity of the algorithm is $\mathcal{O}(dm \log |C|)$, where d is the maximum out degree of the coalgbra $c\colon C \to \mathcal{P}C$ and m is the number of edges.

References:

1. John E. Hopcroft. An $n \log n$ algorithm for minimizing states in a finite automaton. In *Theory of Machines and Computations*, pages 189-196. Academic Press, 1971.
2. Jules Jacobs and Thorsten Wißmann. Fast coalgebraic bisimilarity minimization. In *Principles of Programming Languages*, POPL '23. ACM, 01 2023. to appear.