SIMULATION RELATION OF DYNAMICAL SYSTEMS

by

Youichi Kobuchi†

and

Karel Culik II§

Research Report CS-80-06

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
January 1980

† On leave from Department of Biophysics
Kyoto University
Kyoto, Japan

§ Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada

* This research was supported by the Natural Sciences and Engineering Council of Canada, Grant No. A7403.
Simulation Relation of Dynamical Systems

Youichi Kobuchi* and Karel Culik II

Department of Computer Science, University of Waterloo
Waterloo, Ontario, Canada

§1 Introduction

Let \( \langle X, h \rangle \) be a dynamical system where \( X \) is the state space and \( h \) is the mapping from \( X \) into itself. Given two such systems \( \langle X_1, h_1 \rangle \) and \( \langle X_2, h_2 \rangle \), we say that \( \langle X_2, h_2 \rangle \) simulates \( \langle X_1, h_1 \rangle \) if the dynamics of \( \langle X_1, h_1 \rangle \) may be completely represented by a part of the dynamics of \( \langle X_2, h_2 \rangle \). That is, we assume the existence of an injective mapping \( f \) from \( X_1 \) into \( X_2 \) such that the following diagram commutes:

\[
\begin{array}{c}
X_2 \xrightarrow{f} X_1 \\
\downarrow{h_2} \quad \downarrow{h_1} \\
X_2 \xleftarrow{f} X_1
\end{array}
\]

In this framework we can investigate the fundamental properties of the simulation relation, for example, through its decomposition. Here the decomposition means to express a given simulation relation as a certain composition of simpler relations.

To illustrate our general idea on a concrete example, we will consider the systems where the state space is a finitely generated free monoid and the mapping is defined by a deterministic generalized sequential machine (dgsm). [1] We also assume that the injective mapping \( f \) which

* On leave from Department of Biophysics, Kyoto University, Kyoto, Japan.
relates the two systems under consideration is realized by a dgsm. Especially we investigate in some detail the cases where there exists a dgsm mapping \( \tilde{f} \) such that \( \tilde{f} \) is an inverse of \( f \).

Analysis of simulation relation as reported here started in [2] where the mappings are restricted to be defined by monoid homomorphisms. The results of this report generalize those in [2] and add more, establishing the fundamental relationships between the decompositions of the systems and the simulation relation or its decomposition.
§2 Definitions and Preliminary Results

In what follows, capital Greek letters designate finite sets.

Let $\Sigma$ be a finite set and $\Sigma^*$ be the monoid generated by $\Sigma$ under the operation of concatenation. Let $\Sigma^+ = \Sigma^* - \{e\}$ where $e$ is the identity element of $\Sigma^*$. When $\Sigma$ is partitioned into $n$ mutually disjoint subsets $\Sigma_1, \Sigma_2, \ldots, \Sigma_n$, we write $\Sigma = \Sigma_1 \oplus \Sigma_2 \oplus \ldots \oplus \Sigma_n$.

**Def. 1** Let $G(\Sigma, \Delta)$ denote the set of all dgsm mappings $h: \Sigma^* \rightarrow \Delta^*$ such that for each $a \in \Delta$ there exists $u \in \Sigma^+$ for which $a$ appears in $h(u)$. $H(\Sigma, \Delta)$ is the subset of $G(\Sigma, \Delta)$ composed of one state dgsm mappings (i.e. monoid homomorphisms).

**Def. 2** We say that $h \in G(\Sigma, \Delta)(H(\Sigma, \Delta))$ has a decomposition $fg$ if $h(u) = f(g(u))$ for any $u \in \Sigma^*$ such that $f \in G(\Theta, \Delta)(H(\Theta, \Delta))$ and $g \in G(\Sigma, \Theta)(H(\Sigma, \Theta))$ for some alphabet $\Theta$.

The following lemmas are easy to prove.

**Lemma 1** Let $\Theta$ be a binary alphabet. Any $h \in G(\Sigma, \Delta)$ has a decomposition $fg$ such that $f \in G(\Theta, \Delta)$ is onto $\Delta^*$ and $g \in G(\Sigma, \Theta)$.

**Lemma 2** Given $h \in G(\Sigma, \Delta)$, we can have a decomposition $fg$ such that $f \in H(\Theta, \Delta)$ and $g \in G(\Sigma, \Theta)$, where $g$ is length preserving.

It is useful to note that a decomposition $fg$ of $h \in H(\Sigma, \Delta)$ signifies nothing but a way of expressing $h(a)'s$ as elements in $\{f(A) / A \in \Delta^*\}$ where $a \in \Sigma$. [2,3]

In this report, we restrict our attention to dynamical systems $<\Sigma^*, h>$ where $\Sigma$ is an arbitrary finite alphabet and $h \in G(\Sigma, \Sigma)$. (Note that if $h \in H(\Sigma, \Sigma)$, then the system $<\Sigma^*, h>$ is but a DOL scheme [4]).

**Def. 3** Given two systems $<\Sigma^*, h>$ and $<\Delta^*, g>$, we say that $<\Sigma^*, h>$ simulates $<\Delta^*, g>$ (or simply, $h$ simulates $g$) if there exists
f ∈ G(Δ,Σ) which is injective such that hf = fg. We write this relation symbolically as \( h \overset{f}{\leftarrow} g \) or \( g \overset{f}{\rightarrow} h \).

Note that it is decidable whether a given gsm mapping is injective by Theorem 5 in [5].

Some of the basic properties of the simulation relation are listed. The domain and the range alphabets of each mapping are assumed to be given appropriately. In the list below, lower case Greek letters denote permutations over some finite sets.

(P. 1) a) \( h \overset{\mu^{-1}}{\leftarrow} \overset{\mu}{\rightarrow} h \)

b) \( h \overset{f}{\leftarrow} g \overset{\mu^{-1}}{\leftarrow} \overset{\mu}{\rightarrow} \overset{f\pi}{\leftarrow} \overset{\pi^{-1}}{\rightarrow} g \).

By virtue of this, we may disregard the renaming of the alphabets in considering the simulation relation.

(P. 2) a) \( h \overset{f_1}{\leftarrow} g \overset{f_2}{\leftarrow} k = h \overset{f_1f_2}{\leftarrow} k \)

b) Let \( \lambda \) be an identity mapping.

\( h \overset{\lambda}{\leftarrow} h \overset{f}{\leftarrow} g = h \overset{f}{\leftarrow} g \overset{\lambda}{\leftarrow} g \).

These properties show that the set of systems under consideration makes a 'category' where the injective dgs m mappings are the morphisms.

(P. 3) Let \( n \) be an arbitrary positive integer.

a) \( h \overset{f}{\leftarrow} g = h^n \overset{f}{\leftarrow} g^n \)

b) When \( h \) is injective,

\( h \overset{f}{\leftarrow} g \overset{h^n}{\leftarrow} g \)

c) When \( g \) is injective,

\( h \overset{f}{\leftarrow} g = h \overset{f \pi}{\leftarrow} g \)

(P. 4) Let \( h \in G(\Sigma,\Sigma) \) and \( \bar{h} \in G(\Delta,\Delta) \) be injective mappings.

\( h = fg, \bar{h} = gf \Rightarrow h \overset{f}{\leftarrow} \bar{h} \overset{g}{\rightarrow} \bar{h} \)

This is the fundamental relationship between the decomposition of injective systems and their simulation relation.
(P. 5) Given an injective \( g \in G(\Sigma, \Sigma) \), we have a binary alphabet \( \Theta \), 
\( h \in G(\Theta, \Theta) \) and \( f \in H(\Sigma, \Theta) \) such that \( h \overset{f}{\sim} g \).

By Lemma 1, we have \( g = pf \) for some \( f \in G(\Sigma, \Theta) \) and \( p \in G(\Theta, \Sigma) \). Then \( fp \overset{f}{\sim} pf \). Put \( h = fp \in G(\Theta, \Theta) \).

(P. 4) and (P. 5) make the following definition meaningful.

**Def. 4** Consider a simulation relation \( h \overset{f}{\sim} g \). If \( h = fp \) for some dgsms
mapping \( p \), we call the simulation primitive.

Note that if \( h \overset{f}{\sim} g \) and \( h = fp \), then \( g = pf \).
§3 Decomposition of Simulation Relation

In this section we show two ways of decomposition of simulation relation: parallel and series. Before going into the details, we need some definitions. Let $\Sigma = \Sigma_1 \oplus \ldots \oplus \Sigma_n$. For any $x \in \Sigma^*$, $x|_{\Sigma_i}$ is the longest sparse subword of $x$ which is entirely composed of the elements in $\Sigma_i$. That is, if $x = a_1 \ldots a_i b_1 \ldots b_k$ where $a_j \in \Sigma_i^*$ and $b_j \in (\Sigma - \Sigma_i)^*$ for $j = 1, 2, \ldots, k$, then $x|_{\Sigma_i} = a_1 a_2 \ldots a_k$.

**Def. 5** Let $\Sigma = \Sigma_1 \oplus \ldots \oplus \Sigma_n$, $\Delta = \Delta_1 \oplus \ldots \oplus \Delta_n$, and let $\pi$ be a permutation over $\{1, 2, \ldots, n\}$. Given $n$ dsgm mappings $h_i \in G(\Sigma_i, \Delta_{\pi(i)})$ ($i = 1, 2, \ldots, n$), we define a dsgm mapping $h \in G(\Sigma, \Delta)$ out of $h_i$'s as follows:

(i) $h(a) = h_i(a)$ if $a \in \Sigma_i$, and

(ii) for any $x \in \Sigma^+$ and any $a \in \Sigma_i$, $h(xa) = h(x)(h_i(x|_{\Sigma_i}))^{-1}h_i(xa|_{\Sigma_i})$.

We denote $h = h_1 \oplus \ldots \oplus h_n$. In case $\pi$ is the identity mapping, we also write $h = h_1 \oplus \ldots \oplus h_n$.

**Lemma 3** Let $f = f_1 \oplus \ldots \oplus f_n(\pi)$ as defined in Def. 5. Then $f$ is injective if and only if all the $f_i$'s ($i = 1, 2, \ldots, n$) are injective.

**Proof** Let $f$ be not injective. There exist distinct $x, x' \in \Sigma^*$ such that $f(x) = f(x') = y$. When $y = \varepsilon$, there exist some $f_i$ and $\tilde{x} \in \Sigma_i^+$ such that $f_i(\tilde{x}) = \varepsilon$, which implies $f_i$ is not injective. When $y \neq \varepsilon$, let $y = y_1y_2 \ldots y_n$ and assume $y_i \in \Delta_1$ for some $i$. Then we can write $x = x_1w$ and $x' = x_1'w'$ where $x_1$ and $x_1'$ are the longest prefix in $\Sigma^*$ of $x$ and $x'$, respectively. As $f_i(x) = f_i(x_i)$, either $x_1 = x_1'$ or $f_i$ is not injective. When $x_1 = x_1'$, repeat the process using $w$ and $w'$ instead of $x$ and $x'$. It finally terminates and therefore some $f_i$ is not injective. Conversely, if some $f_i$ is not injective, then clearly $f$ is not injective.
3.1. Parallel Decomposition

Theorem 1 Let $\Sigma = \Sigma_1 \oplus \ldots \oplus \Sigma_n$ and $\Delta = \Delta_1 \oplus \ldots \oplus \Delta_n$. Assume

$h = h_1 \otimes \ldots \otimes h_n(\pi)$ and $f = f_1 \oplus \ldots \oplus f_n$ where $h \in G(\Sigma, \Sigma)$ and $f \in G(\Delta, \Sigma)$. Then we have for $g \in G(\Lambda, \Delta)$

$$hf = fg \Rightarrow h_{i_i} f_i = f_{\pi(i)} g_i \text{ for } i = 1, 2, \ldots, n$$

Proof Let $hf = fg$, that is, $(h_1 \otimes \ldots \otimes h_n)(f_1 \oplus \ldots \oplus f_n) = (f_1 \oplus \ldots \oplus f_n)g$. Define $g_i = g_i |_{\Delta_i}$, then $g_i$ must be in $G(\Delta_i, \Delta_i(\pi))$ by the assumption. Also for $i = 1, \ldots, n$, $h_{i_i} f_i = f_{\pi(i)} g_i$ must hold.

Because $$(h_1 \otimes \ldots \otimes h_n)(f_1 \oplus \ldots \oplus f_n) = h_1 f_1 \otimes \ldots \otimes h_n f_n = f_{\pi(1)} g_1 \otimes \ldots \otimes f_{\pi(n)} g_n = (f_1 \oplus \ldots \oplus f_n)g,$$ $g$ must be $g_1 \otimes \ldots \otimes g_n(\pi)$.

The rest of the proof is easy.

Cor. 1.1 Under the same condition as in the Theorem 1,

$$h^f g = h^f g_1 = g_1 = g_1$$

where $\pi_i(i) = i (i = 1, 2, \ldots, n)$.

Cor. 1.2 Under the same condition as in the Theorem 1, and if $\pi$ is identity,

$$h^f g = h_1 \otimes \ldots \otimes h_n(\pi)$$

In Def. 5, we defined a dgsm mapping $h = h_1 \otimes \ldots \otimes h_n(\pi)$ where each $h_i (i = 1, \ldots, n)$ has distinct domain and range. We here define another composite mapping where the ranges of all the $h_i$'s (i = 1, \ldots, n) are the same.

Def. 6 Let $\Sigma = \Sigma_1 \oplus \ldots \oplus \Sigma_n$ and assume that $n$ dgsm mappings $h_i \in G(\Sigma_i, \Delta_i) (i = 1, 2, \ldots, n)$ are given. We define a dgsm mapping $h \in G(\Sigma, \Delta)$ as $h = \mu(\gamma_1 h_1 \oplus \ldots \oplus \gamma_n h_n)$ where $\gamma_i : \Delta \rightarrow \Delta_i$, $\Delta_i = \{a^{(i)} | a \in \Delta\}$, $\gamma_i(a) = a^{(i)}$ for any $a \in \Delta (i = 1, \ldots, n)$, and $\mu : (\cup \Delta_i) \rightarrow \Delta$ such that $\mu(a^{(i)}) = a$. We denote this composition by $h = h_1 \circ \ldots \circ h_n$. 
Lemma 4. Let $\Delta = \Delta_1 \oplus \ldots \oplus \Delta_n$. Assume

$h \in G(\Sigma, \Sigma)$, $f = f_1 \oplus \ldots \oplus f_n \in G(\Delta, \Sigma)$, $f_i \in G(\Delta_i, \Sigma)$,

$g = g_1 \oplus \ldots \oplus g_n \in G(\Delta, \Delta)$ and $g_i \in G(\Delta_i, \Delta_i)$ (i=1,...,n). If f is injective, then

$$h \xrightarrow{f} g \Rightarrow h \xleftarrow{f} g_i \ (i=1,\ldots,n)$$

3.2. Series Decomposition

Given a simulation relation $h \xrightarrow{f} g$ and a decomposition

$f = f_1 f_2$ ($f_1$ injective), it is not necessarily possible to decompose the relation into $h \xleftarrow{f_1} k \xrightarrow{f_2} g$ for some system $k$. We have a sufficient condition for a simulation relation to be decomposed as follows.

Theorem 2. If $h \xrightarrow{f_2} g$ and $h \xleftarrow{f_1} k$, then we have $k \xleftarrow{f_2} g$.

Proof. We have $f_1 f_2 g = h f_1 f_2 = f_1 k f_2$. Because of the injectivity of $f_1$, we get $f_2 g = k f_2$. Note that $f_2$ is injective when $f_1 f_2$ is injective.

Cor. 2.1 \[
\begin{align*}
&f_1 h_1 \xleftarrow{f_1 f_2} g \\
&f_1 \text{: injective}
\end{align*}
\]

$\Rightarrow f_1 h_1 \xleftarrow{f_1} h_1 f_1 \xleftarrow{f_2} g \\
\text{(Note that the left hand side simulation relation is primitive.)}$

Cor. 2.2 \[
\begin{align*}
&f_1 h \xleftarrow{f_1 f_2} g \\
&h \xleftarrow{f_1} g
\end{align*}
\]

$\Rightarrow g \xleftarrow{f_2} g$

Cor. 2.3 \[
\begin{align*}
&h \xleftarrow{f_1 f_2} g \\
&h \xleftarrow{f_1} h
\end{align*}
\]

$\Rightarrow h \xleftarrow{f_2} g$

In view of the above Corollaries, a special type of simulation i.e., simulation by itself $(h \xrightarrow{f} h)$ seems to deserve attention. This is the case where two gsm mappings become commutative $(hf = fh)$ in the usual sense.
§4 Commutative Mappings

In this section we investigate several properties of commutative dgsm mappings. The class of commutative dgsm mappings for a given mapping is related not only to the simplest type of simulation but also to more general types as shown below.

Def. 7 Two dgsm mappings \( h, g \in G(\Sigma, \Sigma) \) are said to be commutative if \( hg = gh \). Let \( C(h) \) denote the class of dgsm mappings which commute with fixed dgsm \( h \), i.e., \( C(h) = \{ g \in G(\Sigma, \Sigma) | hg = gh \} \) for \( h \in G(\Sigma, \Sigma) \).

Some of the properties of \( C(h) \) for \( h \in G(\Sigma, \Sigma) \) are listed below.

(Pc. 1) a) \( f, g \in C(h) \Rightarrow fg \in C(h) \)

b) \( f \in C(h_1), f \in C(h_2) \Rightarrow f \in C(h_1 \circ h_2) \)

(Pc. 2) a) \( C(h) \subseteq C(h^n) \) for any integer \( n \geq 1 \)

b) \( C(h^m) \cap C(h^n) \subseteq C(h^{m-n}) \) when \( h \) is injective and \( m, n \) are integers such that \( m > n \geq 1 \).

(Pc. 3) If \( h = fg = gf \), then \( f, g \in C(h) \).

(Pc. 4) Let \( \Sigma = \Sigma_1 \circ \ldots \circ \Sigma_n \) and let \( h = h_1 \circ \ldots \circ h_n (\pi) \) and \( g = g_1 \circ \ldots \circ g_n (\mu) \).

Then \( g \in C(h) \Rightarrow \pi \mu = \mu \pi \) and

\[
h_{\mu(i)} g_{\pi(i)} = g_{\pi(i)} h_{\mu(i)} \quad (i=1, \ldots, n)
\]

This relates the commutative mappings with some types of simulation relation. For example, we have for the case \( n = 2, \pi \) is the identity, and \( \mu = (1,2) \),

Cor. Let \( h_1 \in G(\Sigma_1, \Sigma_1) (i=1,2), f_1 \in G(\Sigma_1, \Sigma_2) \), and \( f_2 \in G(\Sigma_2, \Sigma_1) \).

Assume that \( \Sigma_1 \cap \Sigma_2 = \phi \) and \( f_1, f_2 \) are both injective. Then

\[
h_1 \circ h_2 \in C(f_1 \circ f_2) \Rightarrow h_1 \xrightarrow{f_2} h_2
\]
§5 Simulations by Invertible Mappings

In this section, we restrict ourselves to the cases where the simulations are realized through an invertible dgsm mappings. We first examine properties of invertible dgsm mappings and then analyze the simulation relations utilizing the obtained properties.

Def. 8 Let \( f \in G(\Sigma, \Delta) \) be a dgsm mapping. We say that \( f \) is invertible if there exists a dgsm mapping \( \tilde{f} \in G(\Delta, \Sigma) \) such that \( \tilde{f}f \) is the identity mapping on \( \Sigma^* \). \( \tilde{f} \) is called an inverse of \( f \), and we denote by \( f^{-1} \) one of the inverses of \( f \).

Note that if \( f \in G(\Sigma, \Delta) \) is invertible, it must be injective by definition. The following are elementary properties of invertible mappings with respect to their decompositions.

Lemma 5 Let \( f = hg \) where \( f \in G(\Sigma, \Delta) \), \( h \in G(\Gamma, \Delta) \), and \( g \in G(\Sigma, \Gamma) \). Then we have

(i) If \( h \) and \( g \) are invertible, then so is \( f \).

(ii) If \( f \) is invertible, then so is \( g \).

Proof. (i) Put \( \tilde{f} = g^{-1}h^{-1} \). Then \( \tilde{f}f \tilde{f}g^{-1}h^{-1} = g^{-1}g = \lambda_{\Sigma^*} \).

(ii) Put \( g = f^{-1}h \). Then \( gg = f^{-1}hfg^{-1}f = \lambda_{\Sigma^*} \).

Lemma 6 Let \( f = f_1 \otimes f_2 \otimes \ldots \otimes f_n \) (\( \pi \)) as defined in Def. 5. Then \( f \) is invertible if and only if all the \( f_i \)'s (i=1, 2, \ldots, n) are invertible.

Proof. We show the case where \( n = 2 \). (It is easy to generalize the proof for arbitrary \( n \).) Let \( \Sigma = \Sigma_1 \oplus \Sigma_2 \) and \( \Delta = \Delta_1 \oplus \Delta_2 \) be alphabets such that \( f \in G(\Sigma, \Delta) \) and \( f_i \in G(\Sigma_i, \Delta_{\pi}(i)) \) (i=1, 2). (1) Assume \( f_1 \) and \( f_2 \) are invertible and let \( \tilde{g}_{\pi(i)} = f_i^{-1} \in G(\Delta_{\pi(i)}, \Sigma_i) \) (i=1, 2). If we put \( f = \tilde{g}_{\pi(1)} \otimes \tilde{g}_{\pi(2)} \) (\( \pi^{-1} \)), then we have \( \tilde{f}f = (\tilde{g}_{\pi(1)} \otimes \tilde{g}_{\pi(2)})(f_1 \otimes f_2) = \tilde{g}_{\pi(1)} f_1 \oplus \tilde{g}_{\pi(2)} f_2 = f_1^{-1}f_1 \oplus f_2^{-1}f_2 = \lambda_{\Sigma^*} \).

(2) Assume \( f \) is invertible,
then we have \( f^{-1}(f_1 \otimes f_2) = \lambda_{\Sigma_1^*} \otimes \lambda_{\Sigma_2^*} \). It is easy to see that \( f^{-1}|_{\Delta^*} \) is an inverse of \( f_i \) \((i=1,2)\).

Next, we shall give a characterization of the invertible dgsm mapping \( f \) through the properties of a dgsm \( M_f \) realizing it.

**Def. 9** Let \( M = (\Sigma, Q, \Delta, \lambda, \mu, q_o) \) be a (completely specified reduced) dgsm where \( \lambda: \Sigma \times Q \rightarrow Q \), \( \mu: \Sigma \times Q \rightarrow \Delta^* \), and \( q_o \in Q \). \( M \) is said to be a prefix if for every \( q \in Q \), the list \( \{\mu(a,q) | a \in \Sigma\} \) is a prefix. (A list of strings is a prefix if there exist no two elements in the list such that one is a (not necessarily proper) prefix of the other.)

**Theorem 3** Let \( f \in G(\Sigma, \Delta) \) be a mapping realized by a completely specified reduced dgsm \( M_f \). Then \( f \) is invertible if and only if \( M_f \) is a prefix.

**Proof.** (1) Assume \( M_f = (\Sigma, Q, \Delta, \lambda, \mu, q_o) \) is not a prefix. Assume also that \( \tilde{f} \) is a mapping such that \( \tilde{f}f = \lambda_{\Sigma^*} \). Then for some state \( q \in Q \), the list \( \{\mu(a,q) | a \in \Sigma\} \) is not a prefix. That is, there exist distinct \( a, b \in \Sigma \) such that \( \mu(a,q) = \mu(b,q)y \) where \( y \in \Delta^* \). Let \( u \in \Sigma^* \) be such that \( \lambda(u,q_o) = q \) and \( \mu(u,q_o) = x \). Then \( \tilde{f}(\chi\mu(b,q)) = ub \) and \( \tilde{f}(\chi\mu(b,q)y) = \tilde{f}(\chi\mu(a,q)) = ua \), which shows that \( \tilde{f} \) does not preserve initial subwords. Thus \( \tilde{f} \) can not be a dgsm mapping. (2) Assume that \( M_f \) is a prefix. Then it is easy to see that \( f \) is injective. We construct a dgsm that realizes a mapping \( \tilde{f} \) such that \( \tilde{f}f = \lambda_{\Sigma^*} \). a) First, define a sequential transducer \( M_f^{-1} = (\Delta, Q, \Sigma, H, q_o) \) where \( \Delta \) is the input alphabet, \( \Sigma \) is the output alphabet, \( Q \) is the state set, \( q_o \in Q \), and \( H \) is a finite subset of \( Q \times \Delta^* \times \Sigma^* \times Q \) such that \( (q,j,x,a,q_1) \in H \) if and only if \( \lambda(a,q_1) = q_j \) and \( \mu(a,q_1) = x \) where \( x \in \Delta^+ \) and \( a \in \Sigma \). b) Next, from \( M_f^{-1} \) defined above, we are able to obtain an (incompletely specified) dgsm \( \tilde{M}_f^{-1} \) as follows: i) Corresponding to each element \( (q_j,b_1b_2...b_k,a,q_1) \in H \) \((b_1,b_2,...,b_k \in \Delta)\)
define transitions

where \( q^r \)'s (\( r=1,2,\ldots,k-1 \)) are new states added to \( Q \).

ii) After the process i), merge a pair of states \( q' \) and \( q'' \) if there exist another states \( q \) such that \( q \) goes to both \( q' \) and \( q'' \) by the same input with \( \varepsilon \) output. (In the diagram, for some \( b \in \Delta \).) Repeat this process until no such merging could be possible. c) Finally, convert \( M^{-1}_f \) to obtain a completely specified dgsm \( M^{-1}_f \) by adding appropriate transitions. (One of the simplest ways would be to add a self-loop with \( \varepsilon \) output label whenever necessary.)

We shall give an example to illustrate the processes described in the proof (2) of Theorem 3.

Example

\[ \text{Example} \]

\[ M_f \]

\[ M^{-1}_f \]
b) - ii) \((q_{11}^i \text{ and } q_{10}^i \text{ are merged})\)

\[
\begin{align*}
q_0 &\rightarrow q_{01}^i \rightarrow q_{11}^i \\
q_0 &\rightarrow q_{10}^i \rightarrow q_{1} \\
q_0 &\rightarrow q_{11} \rightarrow q_{10} \\
q_1 &\rightarrow q_{01} \rightarrow q_{11} \\
q_1 &\rightarrow q_{10} \rightarrow q_{1} \\

a/\epsilon &\rightarrow b/0 \\
b/1 &\rightarrow a/1 \\
a/\epsilon &\rightarrow b/0
\end{align*}
\]

\[M_{f-1}\]
By this theorem, we can easily determine whether a given dgsm mapping is invertible or not, although the decidability result itself is already implicit under more general context in [6].

As a corollary to the theorem, we also have a machine independent characterization of invertible dgsm mappings.

**Cor.** Let \( f: \Sigma^* \to \Delta^* \) be a dgsm mapping. Then \( f \) is invertible if and only if for any \( u \in \Sigma^* \) and for any distinct pair \( a, b \in \Sigma \), \( f(u a) \) is not a prefix of \( f(u b) \).

Now we introduce a simple relation which may hold between two (dgsm) mappings having the same range as follows.

**Def. 10** For \( f \in G(\Gamma, \Delta) \) and \( h \in G(\Sigma, \Delta) \), we say that \( h \) covers \( f \) if \( h(\Sigma^*) \supseteq f(\Gamma^*) \).

This notion of covering plays a fundamental role when we consider the decompositions of dgsm mappings, as illustrated by the following lemmas.

**Lemma 7** Let \( f \in G(\Gamma, \Delta) \) and \( h \in G(\Sigma, \Delta) \). Assume that \( h \) is invertible. Then the following three conditions are equivalent.

1. \( h \) covers \( f \);
2. \( f = h h^{-1} f \);
3. \( f = h g \) for some dgsm mapping \( g \).

**Proof** (1) \( \to \) (2): \( h(\Sigma^*) \supseteq f(\Gamma^*) \) and \( h h^{-1} f = h(\Sigma^*) = \lambda_{h(\Sigma^*)} f(\Gamma^*) \) imply \( h^{-1} f = f \). (2) \( \to \) (3): trivial. (3) \( \to \) (1): Because \( f = h g \), we have \( h(\Sigma^*) \supseteq f(\Gamma^*) \).

**Lemma 8** Let \( f \in G(\Gamma, \Delta) \) and \( h \in G(\Sigma, \Gamma) \). Assume that \( fh \) is invertible. Then the following three conditions are equivalent.

1. \( fh \) covers \( f \);
(2) \( f = f_{hh^{-1}} \) for some \( h^{-1} \);

(3) \( f = gh^{-1} \) for some (invertible) mapping \( g \).

Proof. (1) \( \rightarrow \) (2): Put \( \tilde{h} = (fh)^{-1}f \). Then \( \tilde{h}h = (fh)^{-1}fh = \lambda_{\Sigma^*} \). That is, \( \tilde{h} \) is an inverse of \( h \). Let \( h^{-1} = \tilde{h} \). Then we have \( fhh^{-1} = fh(fh)^{-1}f = f \) because \( fh \) covers \( f \). (2) \( \rightarrow \) (3): trivial. (3) \( \rightarrow \) (1): We have \( fhh^{-1} = gh^{-1}hh^{-1} = gh^{-1} = f \), which implies that \( fh \) covers \( f \).

Now we turn to the simulation relation where the simulation \( gsm \) mapping is invertible. When \( h \in G(\Delta, \Delta) \) simulates \( g \in G(\Sigma, \Sigma) \) through an invertible \( f \in G(\Sigma, \Delta) \), we write \( h \overset{f}{\leftrightarrow} g \) instead of writing \( h \overset{f}{\rightarrow} g \) to emphasize the invertibility of \( f \). First, we note the following fundamental properties.

**Lemma 9** Let \( f \in G(\Sigma, \Delta) \) be invertible.

(i) For any \( h \in G(\Delta, \Delta) \), there exists a unique \( g \in G(\Sigma, \Sigma) \) such that \( h \overset{f}{\leftrightarrow} g \), if and only if \( f \) covers \( hf \). (In fact, \( g = f^{-1}hf \).)

(ii) For any \( g \in G(\Sigma, \Sigma) \), there exists \( h \in G(\Delta, \Delta) \) of the form \( fgf^{-1} \) such that \( h \overset{f}{\leftrightarrow} g \). (Note that this is a primitive simulation.)

Proof. (i) Let \( hf = fg \). Then \( g = f^{-1}hf \) and we have \( hf = ff^{-1}hf \), which means \( f \) covers \( hf \) by Lemma 7. If \( f \) covers \( hf \), then there exists \( g \) such that \( hf = fg \) by Lemma 7. The uniqueness of \( g \) is also guaranteed by the fact that \( f \) covers \( hf \). (ii) Let \( h = fgf^{-1} \) for an inverse \( f^{-1} \). Then \( hf = fgf^{-1}f = fg \).

As to the decomposition of simulation relation when the coding \( gsm \) is invertible, we have the following two basic results.

**Theorem 4** Let \( h \overset{f_1}{\leftrightarrow} g \) for \( h \in G(\Sigma, \Sigma) \), \( g \in G(\Delta, \Delta) \), \( f_1 \in G(\Theta, \Sigma) \), and \( f_2 \in G(\Delta, \Theta) \). Assume that \( f_1 \) is invertible. Then the simulation relation can be decomposed as \( h \overset{f_1}{\leftrightarrow} f_2 \) for some \( k \in G(\Theta, \Theta) \) if and only if \( f_1 \) covers \( hf_1 \).
Proof. By Theorem 2 and Lemma 7.

**Theorem 5** Let \( h_1 h_2 f = g \) for \( h_1 \in G(\Gamma, \Sigma) \), \( h_2 \in G(\Sigma, \Gamma) \), \( f \in G(\Delta, \Gamma) \), and \( g \in G(\Delta, \Delta) \). Assume that \( h_1 \) is invertible and \( h_2 \) is injective.

Then the simulation can be decomposed as \( h_1 h_2 \leftarrow h_2 h_1 \leftarrow f \) if and only if \( h_1 \) covers \( f \).

**Proof.** When the decomposition is occurring, we have \( h_2 h_1 h_1^{-1} f = h_1^{-1} f g = h_1^{-1} h_1 h_2 f = h_2 f \). Thus \( h_1 h_1^{-1} f = f \) which is equivalent to \( h_1 \) covers \( f \) by Lemma 7. If \( h_1 \) covers \( f \), then \( h_1 h_1^{-1} f = f \) by the same lemma. Then we have \( h_1 h_2 h_1 h_1^{-1} f = h_1 h_2 f = fg = h_1 h_1^{-1} fg \), which implies \( h_2 h_1 h_1^{-1} f = h_1^{-1} fg \).
§6 Concluding Remarks

In general systems theory, the principle of simulation plays an essential role [7]. We presented some basic properties of simulation relations of dynamical systems in the case where the dynamics and the simulation mapping are given by dgsms. We paid a special attention to the cases where the simulation mapping is invertible.

Although the treatments reported here are by no means complete, we hope the concepts and the methods introduced are useful in revealing the fundamental dynamical structure of systems.

Some of the relevant problems we have not touched upon in this report include:

1) For given two systems decide if there holds the simulation relation through some simulation mapping.

2) Establish suitable complexity measure of gsm mappings and relate it with their decomposition structures.

3) Investigate the properties (e.g., invertibility, decomposition structure, etc.) of gsm mappings with endmarker [6] and generalize the results of this report for that case.
References


