Synchronous Elastic Networks

Sava Krstić
Strategic CAD Labs, Intel Corporation
Hillsboro, Oregon, USA

Jordi Cortadella
Universitat Politècnica de Catalunya
Barcelona, Spain

Mike Kishinevsky, John O'Leary
Strategic CAD Labs, Intel Corporation
Hillsboro, Oregon, USA

***Abstract*— We formally define—at the stream transformer level—a class of synchronous circuits that tolerate any variability in the latency of their environment. We study behavioral properties of networks of such circuits and prove fundamental compositionality results. The paper contributes to bridging the gap between the theory of latency-insensitive systems and the correct implementation of efficient control structures for them.**

I. Introduction

The conventional abstract model for a synchronous circuit is a machine that reads inputs and writes outputs at every cycle. The outputs at cycle i are produced according to a calculation that depends on the inputs at cycles $0, \ldots, i$. Computations and data transfers are assumed to take zero delay.

Latency-insensitive design by Carloni et al. [2] aims to relax this model by elasticizing the time dimension and so decoupling the cycles from the calculations of the circuit. It enables the design of circuits tolerant to any discrete variation (in the number of cycles) of the computation and communication delays. With this modular approach, the functionality of the system only depends on the functionality of its components and not on their timing characteristics.

The motivation for latency-insensitive design comes from the difficulties with timing and communication in nanoscale technologies. The number of cycles required to transmit data from a sender to a receiver is governed by the distance between them, and often cannot be accurately known until the chip layout is generated late in the design process. Traditional design approaches require fixing the communication latencies up front, and these are difficult to amend when layout information finally becomes available. Elastic circuits offer a solution to this problem. In addition, their modularity promises novel methods for microarchitectural design that can use variable-latency components and tolerate static and dynamic changes in communication latencies, while—unlike asynchronous circuits—still employing standard synchronous design tools and methods.

Cortadella et al. [4] present a simple elastic protocol, called SELF (Synchronous Elastic Flow) and describe methods for efficient implementation of elastic systems and for conversion of regular synchronous designs into elastic form. Inspired by the original work on latency-insensitive design [2], SELF also differs from it in ways that render the theory developed in [2] hardly applicable.

In this paper we give theoretical foundations of SELF: a novel and arguably more practicable definition of elasticity, and the basic compositionality results. For space reasons, the proofs are omitted, but are available in the technical report [7].

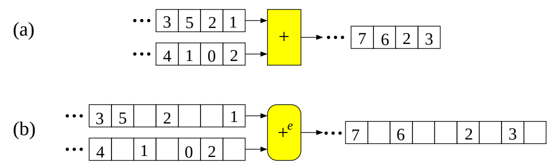

Fig. 1. (a) Conventional synchronous adder, (b) Synchronous elastic adder.

A. Overview

Figure 1(a) depicts the timing behavior of a conventional synchronous adder that reads input and produces output data at every cycle (boxes represent cycles). In this adder, the i-th output value is produced at the i-th cycle. Figure 1(b) depicts a related behavior of an elastic adder—a synchronous circuit too—in which data transfer occurs in some cycles and not in others. We refer to the transferred data items as *tokens* and we say that idle cycles contain *bubbles*.

Put succinctly, elasticization decouples cycle count from token count. In a conventional synchronous circuit, the i-th token of a wire is transmitted at the i-th cycle, whereas in a synchronous elastic circuit the i-th token is transmitted at some cycle $k \geq i$.

Turning a conventional synchronous adder into a synchronous elastic adder requires a communication discipline that differentiates idle from non-idle cycles (bubbles from tokens). In SELF, this is implemented by a pair of single-bit control wires: *Valid* and *Stop*. Every input or output wire Z in a synchronous component is associated to a *channel* in the elastic version of the same component. The channel is a triple of wires $\langle Z, \mathsf{valid}_Z, \mathsf{stop}_Z \rangle$, with Z carrying the data and the other two wires implementing the control bits, as shown in Figure 2(b). A token is transferred on this channel when $\mathsf{valid}_Z \wedge \neg\mathsf{stop}_Z$: the sender sends valid data and the receiver is ready to accept it; see Figure 4. Additional constraints that guarantee correct elastic behavior are given in Section III. There we define precisely the class of elastic circuits and what it means for a circuit A^{e} to be an elastization of a given circuit A. In particular, our definition implies liveness: A^{e} produces infinite streams of tokens if its environment produces infinite streams of tokens at the input channels and is ready to accept infinite streams at the output channels.

Suppose $\mathcal{N}$ is a network of standard (non-elastic) components, as in Figure 2(a). Suppose we then take elasticizations of

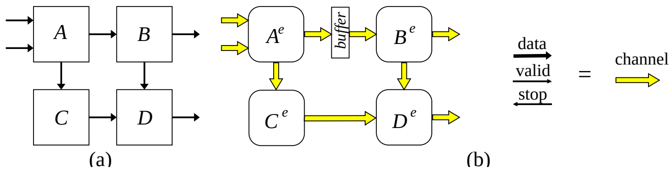

Fig. 2. A synchronous network (a) and its elastic counterpart (b).

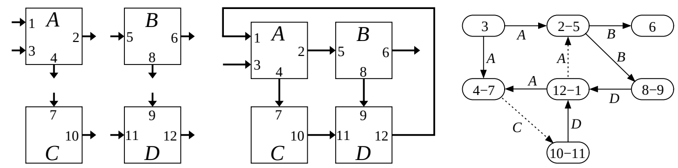

Fig. 3. Four machines (left) put into a network $\mathcal{N}$ (middle), and the network's dependency graph $\Delta(\mathcal{N})$ (right). The nodes of $\Delta(\mathcal{N})$ are wires; internal wires get two labels. The arcs are *non-sequential* input-output wire pairs of component circuits. Dotted arcs indicate that (1,2) and (7,10) are sequential pairs for A and C resp.; they are not part of $\Delta(\mathcal{N})$ so $\Delta(\mathcal{N})$ is acyclic.

these standard components and join their channels accordingly, as in Figure 2(b), ignoring the buffer. Will the resulting network $\mathcal{N}^{\mathrm{e}}$ be an elasticization of $\mathcal{N}$? Will it be elastic at all? These fundamental questions are answered by Theorem 4 of Section IV, which is the main result of the paper. The answers are "yes", provided a certain graph $\Delta^{\mathrm{e}}(\mathcal{N}^{\mathrm{e}})$ associated with $\mathcal{N}^{\mathrm{e}}$ is acyclic. This graph captures the information about paths inside elastic systems that contain no tokens—analogous to combinational paths in ordinary systems. Importantly, $\Delta^{\mathrm{e}}(\mathcal{N}^{\mathrm{e}})$ can be constructed using only local information (the "sequentiality interfaces") of the individual elastic components.

Since elastic networks tolerate any variability in the latency of the components, empty FIFO buffers can be inserted in any channel, as shown in Figure 2(b), without changing the functional behavior of the network. This practically important fact is proved as a consequence of Theorem 4.

Synchronous circuits are modeled in this paper as stream transformers, called *machines*. This well-known technique (see [8] and references therein) appears to be quite underdeveloped. Our rather lengthy preliminary Section II elaborates the necessary theory of networks of machines, culminating with a surprisingly novel combinational loop theorem (Theorem 1).

Figure 3 illustrates Theorem 1 and, by analogy, Theorem 4 as well. It relies on the formalization of the notion of combinational dependence at the level of input-output wire pairs. Each input-output pair of a machine is either *sequential* or not, and the set of sequential pairs provides a machine's "sequentiality interface". When several machines are put together into a network $\mathcal{N}$, their sequentiality interfaces define the graph $\Delta(\mathcal{N})$, the acyclicity of which is a test for the network to be a legitimate machine itself.

Elasticizations of ordinary circuits are not uniquely defined. On the other hand, for every elastic machine A there is a unique standard machine, denoted A^{T}, that corresponds to it. We do not discuss any specific elasticization procedures in this paper, but state our results in the form that only involves elastic machines and their unique standard counterparts. This makes the results applicable to multiple elasticization procedures.

B. Related Work

Carloni et al. [2] pioneered a theory of latency-insensitive circuits based on their notion of *patient processes*. Patient processes are defined at a high level of abstraction that models communication on a channel only by "token or bubble", leaving implementation protocol(s) unspecified. In the companion paper [3], Carloni et al. give an incomplete description of an implementation protocol. Assuming our recovery of that protocol (let us call it LID) is accurate, its transfer condition is more complex than that of SELF (Figure 4) and consequently LID requires significantly more complex implementation. For example, conversion of a regular design into LID form needs a wrapper or registers around every module, increasing the latency of each module's computation by two cycles—a penalty that is not required in the SELF elasticization. There might also be practical challenges in interfacing a LID system with an existing non-LID module, requiring the latter to generate stop signals with complex semantics.

cycle	0	1	2	3	4	5	6	7	8	9	...
data_Z	$*$	A	$\mathbf{B}$	B	B	C	$*$	$*$	D	D	...
valid_Z	0	1	1	1	1	1	0	0	1	1	...
stop_Z	0	0	1	1	0	0	0	1	1	0	...
SELF	□	t	□	□	t	t	□	□	□	t	...
LID	□	t	t	□	t	t	□	□	□	t	...

Fig. 4. Comparing the SELF and LID protocols. The bottom rows show the states of the channel Z, differentiating between bubbles (□) and tokens (t). When $\neg\mathsf{valid}_Z$, the value at the data wire is irrelevant (labelled $*$ in cycles 0, 6 and 7). The receiver can issue a stop_Z even when the sender does not send valid data (cycle 7). In the cycles 3, 4, and 9, the sender persistently maintains the same valid data as in the previous cycle. In SELF, data transfer takes place in cycles 1,4,5,9, so the transferred sequence is $ABCD\ldots$. In LID, the same sequence of values on the channel wires signifies transfer of a different sequence of data: $A\mathbf{B}BCD\ldots$ This is because a token is transferred on the LID channel when $\mathsf{valid}_Z \wedge \neg(\mathsf{stop}_Z \wedge \mathit{pre}(\mathsf{stop}_Z))$, where pre stands for the value during the previous cycle. (The first occurrence of the stop request $\mathsf{stop}_Z = 1$ means "perhaps you will need to stop next cycle" and the data item B sent through the channel during cycle 2 is assumed to be successfully transmitted to the receiver.)

We emphasize that the limitations of LID implementations are not inherent to the concept of patient processes. Regarding latency properties, they do not seem to be more limited than elastic systems. Still, it turns out that patient processes are not general enough to model elastic systems as we define them in Section III. This we prove in Section V where patient processes and elastic systems are compared as alternative formalizations of latency-insensitive circuits.

Suhaib et al. [12] revisited and generalized Carloni's elasticization procedure, validating its correctness by a simulation method based on model checking.

Lee et al. [9] study *causality interfaces* (pairwise input-output dependencies) and are "interested in existence and uniqueness of the behavior of feedback composition", but do not go as far as deriving a combinational loop theorem.

In their work on design of interlock pipelines [6], Jacobson et al. use a protocol equivalent to SELF, without explicitly specifying it.

Manohar and Martin discuss "slack elasticity" of asynchronous implementations in [10]. Their slack elasticity conditions relate to the structure of choices in the asynchronous specification. Unlike [10], in the current paper we deal with synchronous systems and we take a black box view of their control—no information about the control flow (and hence on the structure of choices) is ever used. Instead the connectivity information corresponding to the system data-flow is used for elasticization. Conservatively ignoring control flow may lead to a performance penalty, but simplifies the translation to an elastic system.

II. Circuits as Stream Functions

In this section we introduce *machines* as a mathematical abstraction of *circuits without combinational cycles*. For simplicity, this abstraction implicitly assumes that all sequential elements inside the circuit are *initialized*. Extending to partially initialized systems appears to be trivial. While there is a large body of work studying circuits or equivalent objects with *good (e.g. constructive [1]) combinational cycles* and their composition (e.g. [5]), we deliberately restrict consideration to the fully acyclic objects, since neither logic synthesis nor timing analysis can properly treat circuits with combinational cycles.

Most of the effort in this section goes into establishing modularity conditions guaranteeing that a system obtained as a network of machines (the feedback construction in particular) is a machine itself.

A. Streams

A *stream over* A is an infinite sequence whose elements belong to the set A. The first element of a stream a is referred to by $a[0]$, the second by $a[1]$, etc. For example, the equation $a[i] = 3i + 1$ describes the stream $(1, 4, 7, \ldots)$.

The set of all streams will be denoted A^∞. Occasionally we will need to consider finite sequences too; the set of all, finite or infinite, sequences over A is denoted A^ω.

We will write $a \sim_k b$ to indicate that the streams a and b have a common prefix of length k. The equivalence relations $\sim_0, \sim_1, \sim_2, \ldots$ are progressively finer and have trivial intersection. Thus, to prove two sequences a and b are equal, it suffices to show $a \sim_k b$ holds for every k. Note also that $a \sim_0 b$ holds for every a and b.

We will use the equivalence relations $\sim_k$ to express properties of systems and machines viewed as multivariate stream functions. All these properties will be derived from the following two basic properties of single-variable stream functions $f\colon A^\infty \to B^\infty$.

causality: $\forall a, b \in A^\infty.\ \forall k \geq 0.\ \ a \sim_k b \Rightarrow f(a) \sim_k f(b)$

contraction: $\forall a, b \in A^\infty.\ \forall k \geq 0.\ \ a \sim_k b \Rightarrow f(a) \sim_{k+1} f(b)$

Informally, f is causal if (for every a) the first k elements of $f(a)$ are determined by the first k elements of a, and f is contractive if the first k elements of $f(a)$ are determined by the first $k-1$ elements of a.

Lemma 1: If $f\colon A^\infty \to A^\infty$ is contractive, then it has a unique fixpoint.

Remark. One can define the *distance* $d(a, b)$ between sequences a and b to be $1/2^k$, where k is the length of the largest common prefix of a and b. This gives the sets A^∞ and A^ω the structure of complete metric spaces and Lemma 1 is an instance of Banach Fixed Point Theorem. See the review paper [8] for more details and references about the metric semantics of systems and [13] for "diadic arithmetic of circuits". We choose not to use the metric space terminology in this paper since all "metric reasoning" we need can be as easily done with equivalence relations $\sim_k$ instead. See [11] for principles of reasoning with such "converging equivalence relations" in more general contexts.

B. Systems

Suppose W is a set of typed *wires*; all we know about an individual wire w is a set $\mathsf{type}(w)$ associated to it. A W*-behavior* is a function σ that associates a stream $\sigma.w \in \mathsf{type}(w)^\infty$ to each wire $w \in W$. The set of all W-behaviors will be denoted $[\![W]\!]$. Slightly abusing the notation, we will also write $[\![w]\!]$ for the set $\mathsf{type}(w)^\infty$. Notice that the equivalence relations $\sim_k$ extend naturally from streams to behaviors:

$$\sigma \sim_k \sigma' \qquad \text{iff} \qquad \forall w \in W.\ \ \sigma.w \sim_k \sigma'.w$$

Notice also that a W-behavior σ can be seen as a single stream $(\sigma[0], \sigma[1], \ldots)$ of W*-states*, where a state is an assignment of a value in $\mathsf{type}(w)$ to each wire w.

Definition 1: A W*-system* is a subset of $[\![W]\!]$.

Example. A circuit that at each clock cycle receives an integer as input and returns the sum of all previously received inputs is described by the W-system $\mathcal{S}$, where W consists of two wires u, v of type $\mathbb{Z}$, and $\mathcal{S}$ consists of all stream pairs $(a, b) \in \mathbb{Z}^\infty \times \mathbb{Z}^\infty$ such that $b[0] = 0$ and $b[n] = a[0]+\cdots+a[n-1]$ for $n > 0$. Each stream pair (a, b) represents a behavior σ such that $\sigma.u = a$ and $\sigma.v = b$.

We will use wires as typed variables in formulas meant to describe system properties. The formulas are built using ordinary mathematical and logical notation, enhanced with temporal operators *next*, *always*, and *eventually*, denoted respectively by $(\_)^+, \mathsf{G}, \mathsf{F}$. As an illustration, the system $\mathcal{S}$ in the example above is characterized by the property $v = 0 \wedge \mathsf{G}\,(v^+ = v + u)$. Also, one has $\mathcal{S} \models \mathsf{F}\,\mathsf{G}\,(u > 0) \Rightarrow \mathsf{F}\,\mathsf{G}\,(v > 1000)$, where $\models$ is used to denote that a formula is true of a system.

C. Operations on Systems

If $W' \subseteq W$, there is an obvious projection map $\sigma \mapsto \sigma\!\downarrow\! W'\colon [\![W]\!] \to [\![W']\!]$. These projections are all one needs for the definition of the following two basic operations on systems.

Definition 2: (a) If $\mathcal{S}$ is a W-system and $W' \subseteq W$, then *hiding* W' in $\mathcal{S}$ produces a $(W - W')$-system $\mathsf{hide}_{W'}(\mathcal{S})$ defined by

$$\tau \in \mathsf{hide}_{W'}(\mathcal{S}) \quad \text{iff} \quad \exists \sigma \in \mathcal{S}.\ \ \tau = \sigma\!\downarrow\!(W - W').$$

(b) The *composition* of a W_1-system $\mathcal{S}_1$ and a W_2-system $\mathcal{S}_2$ is a $(W_1 \cup W_2)$-system $\mathcal{S}_1 \sqcup \mathcal{S}_2$ defined by

$$\sigma \in \mathcal{S}_1 \sqcup \mathcal{S}_2 \quad \text{iff} \quad \sigma \downarrow W_1 \in \mathcal{S}_1 \wedge \sigma \downarrow W_2 \in \mathcal{S}_2.$$

If W and W' are disjoint wire sets, $\sigma \in [\![W]\!]$, and $\tau \in [\![W']\!]$, then there is a unique behavior $\vartheta \in [\![W \cup W']\!]$ such that $\sigma = \vartheta \downarrow W$ and $\tau = \vartheta \downarrow W'$. This "product" of behaviors will be written as $\vartheta = \sigma * \tau$. (If W is the empty set, then $[\![W]\!]$ has one element—a "trivial behavior" that is also a multiplicative unit for the product operation $*$.) We will also use the notation $[u \mapsto a, v \mapsto b, \ldots]$ for the $\{u, v, \ldots\}$-behavior σ such that $\sigma.u = a$, $\sigma.v = b$, etc.

Hiding and composition suffice to define complex networks of systems. To model identification of wires, we use simple *connection systems*: by definition, $\mathsf{Conn}(u, v)$ is the $\{u, v\}$-system consisting of all behaviors σ such that $\sigma.u = \sigma.v$.

Now if $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are given systems and $u_1, \ldots, u_n$, $v_1, \ldots, v_n$ are some of their wires, the network obtained from these systems by identifying each wire u_i with the corresponding wire v_i (of equal type) is the system

$$\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,|\, u_1 = v_1, \ldots, u_n = v_n \rangle$$

defined as $\mathsf{hide}_{\{u_1, \ldots, u_n, v_1, \ldots, v_n\}}(\mathcal{S})$, where

$$\mathcal{S} = \mathcal{S}_1 \sqcup \cdots \sqcup \mathcal{S}_m \sqcup \mathsf{Conn}(u_1, v_1) \sqcup \cdots \sqcup \mathsf{Conn}(u_n, v_n).$$

The simplest case ($m = n = 1$) of networks is the construction

$$\langle \mathcal{S} \,|\, u = v \rangle = \mathsf{hide}_{\{u,v\}}(\mathcal{S} \sqcup \mathsf{Conn}(u, v)),$$

used for a *feedback* definition in Section II-E. A behavior σ belongs to $\langle \mathcal{S} \,|\, u = v \rangle$ if and only if $\sigma * [u \mapsto a, v \mapsto a] \in \mathcal{S}$ for some $a \in [\![u]\!]$.

D. Machines

Suppose I and O are disjoint sets of wires, called *inputs* and *outputs*, correspondingly. By definition, an (I, O)*-system* is just an $(I \cup O)$-system. Consider the following properties of an (I, O)-system $\mathcal{S}$.

deterministic:

$$\forall \omega, \omega' \in \mathcal{S}.\;\; \omega \downarrow I = \omega' \downarrow I \;\Rightarrow\; \omega \downarrow O = \omega' \downarrow O$$

functional:

$$\forall \sigma \in [\![I]\!].\, \exists! \tau \in [\![O]\!].\; \sigma * \tau \in \mathcal{S}$$

causal:

$$\forall \omega, \omega' \in \mathcal{S}. \forall k \geq 0.\;\; \omega \downarrow I \sim_k \omega' \downarrow I \;\Rightarrow\; \omega \downarrow O \sim_k \omega' \downarrow O$$

Clearly, functionality implies determinism. Conversely, a deterministic system is functional if and only if it accepts all inputs. Note also that causality implies determinism: if $\omega \downarrow I = \omega' \downarrow I$, then $\omega \downarrow I \sim_k \omega' \downarrow I$ holds for every k, so $\omega \downarrow O \sim_k \omega' \downarrow O$ holds for every k too, so $\omega \downarrow O = \omega' \downarrow O$.

Definition 3: An (I, O)*-machine* is an (I, O)-system that is both functional and causal.

A functional system $\mathcal{S}$ uniquely determines and is determined by the function $F \colon [\![I]\!] \to [\![O]\!]$ such that $F(\sigma) = \tau$ holds if and only if $\sigma * \tau \in \mathcal{S}$. The causality condition for such $\mathcal{S}$ can be also written as follows:

$$\forall \sigma, \sigma' \in [\![I]\!]. \forall k \geq 0.\;\; \sigma \sim_k \sigma' \Rightarrow F(\sigma) \sim_k F(\sigma').$$

The system in the example in Section II-B is a machine if we regard u as an input wire and v as an output wire. The same is true of the system $\mathsf{Conn}(u, v)$: its associated function F is the identity function.

E. Feedback on Machines

We will use the term *feedback* for the system $\langle \mathcal{S} \,|\, u = v \rangle$ as mentioned in Section II-C when $\mathcal{S}$ is a machine and the wires u and v of the same type are an input and output of $\mathcal{S}$ respectively. Our concern now is to understand under what conditions the feedback produces a machine.

To fix the notation, assume $\mathcal{S}$ is an (I, O)-machine given by $F \colon [\![I]\!] \to [\![O]\!]$, with wires $u \in I$, $v \in O$ of the same type A. By the note at the end of Section II-C, we have that for every $\sigma \in [\![I - \{u\}]\!]$ and $\tau \in [\![O - \{v\}]\!]$,

$$\sigma * \tau \in \langle \mathcal{S} \,|\, u = v \rangle$$

if and only if

$$\exists a \in A^\infty.\; F(\sigma * [u \mapsto a]) = \tau * [v \mapsto a]),$$

so $\langle \mathcal{S} \,|\, u = v \rangle$ is functional when the function $F^\sigma_{uv} : A^\infty \to A^\infty$ defined by $F^\sigma_{uv}(a) = F(\sigma * [u \mapsto a]).v$ has a unique fixpoint. By Lemma 1, this is guaranteed if F^σ_{uv} is contractive.

The following definition introduces the key concept of sequentiality that formalizes the intutive notion that there is no combinational dependence of a given output wire on a given input wire. Sequentiality of the pair (u, v) easily implies contractivity of F^σ_{uv} for all σ.

Definition 4: The pair (u, v) is *sequential* for $\mathcal{S}$ if for every $\sigma, \sigma' \in [\![I]\!]$ and every $k \geq 0$

$$\begin{array}{l} \sigma.u \sim_{k-1} \sigma'.u \\ \wedge\;\; \forall x \in I - \{u\}.\, (\sigma.x \sim_k \sigma'.x) \end{array} \;\Rightarrow\; F(\sigma).v \sim_k F(\sigma').v$$

Lemma 2 (Feedback): If (u, v) is a sequential input-output pair for a machine $\mathcal{S}$, then the feedback system $\langle \mathcal{S} \,|\, u = v \rangle$ is a machine too.

Example. Consider the system $\mathcal{S}$ with $I = \{u, v\}$, $O = \{w, z\}$, specified by equations

$$w = u \oplus ((0) \# v) \qquad z = v \oplus v,$$

where all wires have type $\mathbb{Z}$, the symbol $\oplus$ denotes the componentwise sum of streams, and # denotes concatenation. Since z does not depend on u, the pair (u, z) is sequential. The pair (v, w) is also sequential since to compute a prefix of w it suffices to know (a prefix of the same size of u and) a prefix of smaller size of v. The remaining two input-output pairs (u, w) and (v, z) are not sequential.

To find the machine $\langle \mathcal{S} \,|\, v = w \rangle$, we need to solve the equation $v = u \oplus ((0) \# v)$ for v. For each $u = (a_0, a_1, a_2, \ldots)$, the equation has a unique solution $v = \hat{u} = (a_0, a_0 + a_1, a_0 + a_1 + a_2, \ldots)$. Substituting the solution into $z = v \oplus v$, we obtain a description of $\langle \mathcal{S} \,|\, v = w\rangle$ by a single equation that relates its input and output: $z = \hat{u} \oplus \hat{u}$. The other feedback $\langle \mathcal{S} \,|\, u = z\rangle$ is easier to calculate; it is given by equation $w = v \oplus v \oplus ((0)\#v)$.

F. Networks of Machines and the Combinational Loop Theorem

Consider a network $\mathcal{N} = \langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,|\, u_1 = v_1, \ldots, u_n = v_n\rangle$, where $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are machines with disjoint wire sets and the pairs (u_1, v_1),...,(u_n, v_n) involve n distinct input wires u_i and n distinct output wires v_i. (There is no assumption that u_i, v_i belong to the same machine $\mathcal{S}_j$.) Our goal is to understand under what conditions the system $\mathcal{N}$ is a machine.

Note that $\mathcal{N} = \langle \mathcal{S} \,|\, u_1 = v_2, \ldots, u_n = v_n\rangle$, where $\mathcal{S} = \mathcal{S}_1 \sqcup \cdots \sqcup \mathcal{S}_m$. It is easy to check that an input-output pair (u, v) of $\mathcal{S}$ is sequential if either (1) (u, v) is sequential for some $\mathcal{S}_i$, or (2) u and v belong to different machines. Thus, the information about sequentiality of input-output pairs of the "parallel composition" machine $\mathcal{S}$ is readily available from the sequentiality information about the component machines $\mathcal{S}_i$, and our problem boils down to determining when a multiple feedback operation performed on a single machine results in a system that is itself a machine.

Simultaneous feedback specified by a set of two or more input-output pairs of a machine does not necessarily produce a machine even if all pairs involved are sequential. Indeed, in the example in Section II-E, we had a system $\mathcal{S}$ with two sequential pairs (u, z) and (v, w), but (u, z) ceases to be sequential for $\langle \mathcal{S} \,|\, v = w\rangle$. Indeed, if z and u are related by $z = \hat{u} \oplus \hat{u}$, then knowing a prefix of length k of z requires knowing the prefix of the same length of u; a shorter one would not suffice.

To ensure that a multiple feedback construction produces a machine, one needs to show that, in addition to the wire pairs to be identified, sufficiently many other input-output pairs are also sequential. A precise formulation for a *double* feedback is given by a version of the Bekić Lemma: for the system $\langle \mathcal{S} \,|\, u = w, v = z\rangle$ to be a machine, it suffices that *three* pairs of wires be sequential—(u, w), (v, z), and one of (u, z), (v, w). This non-trivial auxiliary result is needed for the proof of Theorem 1 below, and is a special case of it.

Given an (I, O)-machine $\mathcal{S}$, let its *dependency graph* $\varDelta(\mathcal{S})$ have the vertex set $I \cup O$ and directed edges that go from u to v for each pair $(u, v) \in I \times O$ that is *not* sequential. For a network system $\mathcal{N} = \langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,|\, u_1 = v_1, \ldots, u_n = v_n\rangle$, its graph $\varDelta(\mathcal{N})$ is then defined as the direct sum of graphs $\varDelta(\mathcal{S}_1), \ldots, \varDelta(\mathcal{S}_m)$ with each vertex u_i ($1 \le i \le n$) identified with the corresponding vertex v_i (Figure 3).

Theorem 1 (Combinational Loop Theorem): The network system $\mathcal{N}$ is a machine if the graph $\varDelta(\mathcal{N})$ is acyclic.

III. Elastic Machines

In this section we give the definition of elastic machines. Its four parts—input-output structure, persistence conditions, liveness conditions, and the transfer determinism condition—are covered by Definitions 5-8 below.

A. Input-output Structure, Channels, and Transfer

We assume that the set of wires is partitioned into *data*, *valid*, and *stop* wires, so that for each data wire X there exist associated wires valid_X and stop_X of boolean type. (In actual circuit implementations, valid_X and stop_X need not be physical wires; it suffices that they be appropriately encoded.)

Definition 5: Let I, O be disjoint sets of data wires. An $[I, O]$-*system* is an (I', O')-machine, where $I' = I \cup \{\mathsf{valid}_X \,|\, X \in I\} \cup \{\mathsf{stop}_Y \,|\, Y \in O\}$ and $O' = O \cup \{\mathsf{valid}_Y \,|\, Y \in O\} \cup \{\mathsf{stop}_X \,|\, X \in I\}$.

The triples $\langle X, \mathsf{valid}_X, \mathsf{stop}_X\rangle$ (for $X \in I$) and $\langle Y, \mathsf{valid}_Y, \mathsf{stop}_Y\rangle$ (for $Y \in O$) are to be thought of as *elastic input and output channels* of the system.

Let $\mathsf{transfer}_Z$ be a shorthand for $\mathsf{valid}_Z \wedge \neg\mathsf{stop}_Z$ and say that *transfer along Z occurs in a state s* if $s \models \mathsf{transfer}_Z$. Given a behavior $\sigma = (\sigma[0], \sigma[1], \sigma[2], \ldots)$ of an $[I, O]$-system and $Z \in I \cup O$, let σ_Z be the sequence (perhaps finite!) obtained from $\sigma.Z = (\sigma[0].Z, \sigma[1].Z, \sigma[2].Z, \ldots)$ by deleting all entries $\sigma[i].Z$ such that transfer along Z does not occur in $\sigma[i]$. The *transfer behavior* σ^{T} associated with σ is then defined by $\sigma^{\mathsf{T}}.Z = \sigma_Z$. If all sequences σ_Z are infinite, then σ^{T} is an $(I \cup O)$-behavior; in general, however, we only have $\sigma_Z \in \mathsf{type}(Z)^\omega$.

For each wire Z of an $[I, O]$-system $\mathcal{S}$ we introduce an auxiliary *transfer counter variable* tct_Z of type $\mathbb{Z}$. The counters serve for expressing system properties related to transfer. By definition, tct_Z is equal to the number of states that precede the current state and in which transfer along Z has occurred. That is, for every behavior σ of $\mathcal{S}$, we have $\sigma.\mathsf{tct}_Z = (t_0, t_1, \ldots)$, where t_k is the number of indices i such that $i < k$ and transfer along Z occurs in $\sigma[i]$. Note that the sequence $\sigma.\mathsf{tct}_Z$ is non-decreasing and begins with $t_0 = 0$.

The notation $\mathsf{min\_tct}_S$, for any subset S of $I \cup O$ will be used to denote the smallest of the numbers tct_Z, where $Z \in S$.

B. Definition of Elasticity

An elastic component, when ready to communicate over an output channel must remain ready until the transfer takes place.

Definition 6: The *persistence conditions* for an $[I, O]$-system $\mathcal{S}$ are given by

$$\mathcal{S} \models \mathsf{G}\,(\mathsf{valid}_Y \wedge \mathsf{stop}_Y \Rightarrow (\mathsf{valid}_Y)^+ \wedge Y^+ = Y) \qquad (1)$$

for every $Y \in O$.

The conjunct $Y^+ = Y$ can be removed from (1) without affecting the definition of elastic machines (it follows from other conditions). The most useful consequence of persistence is the "handshake lemma":

$$\mathcal{S} \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Y \wedge \mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_Y \;\Rightarrow\; \mathsf{G}\,\mathsf{F}\,\mathsf{transfer}_Y$$

Liveness of an elastic component is expressed in terms of token count: if all input channels have seen k transfers and there is an output channel that has seen less, then the communication on output channels with the minimum amount of transfer must be eventually offered. The following definition formalizes this,

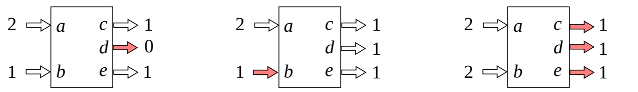

Fig. 5. Liveness: Only the hungriest channels (shaded) are being served. The numbers indicate the current token count at each channel.

together with a similar commitment to eventual readiness on input channels. (See also Figure 5.)

Definition 7: The *liveness conditions* for an $[I, O]$-system are given by

$$\mathcal{S} \models \mathsf{G}\,(\mathsf{min\_tct}_O = \mathsf{tct}_Y \wedge \mathsf{min\_tct}_I > \mathsf{tct}_Y \Rightarrow \mathsf{F}\,\mathsf{valid}_Y) \quad (2)$$

$$\mathcal{S} \models \mathsf{G}\,(\mathsf{min\_tct}_{I\cup O} = \mathsf{tct}_X \Rightarrow \mathsf{F}\,\neg\mathsf{stop}_X) \quad (3)$$

for every $Y \in O$ and every $X \in I$.

In practice, elastic components will satisfy simpler (but stronger) liveness properties; e.g. remove $\mathsf{min\_tct}_O \geq \mathsf{tct}_Y$ from (2) and replace $\mathsf{min\_tct}_{I\cup O} \geq \mathsf{tct}_X$ with $\mathsf{min\_tct}_O \geq \mathsf{tct}_X$ in (3). However, a composition of such components, while satisfying (2) and (3), may not satify the stronger versions of these conditions.

Consider single-channel $[I, O]$-systems satisfying the persistence and liveness conditions: an *elastic consumer* is a $[\{Z\}, \emptyset]$-system $\mathcal{C}$ satisfying (4) below; similarly, an *elastic producer* is a $[\emptyset, \{Z\}]$-system $\mathcal{P}$ satisfying (5) and (6).

$$\mathcal{C} \models \mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_Z \quad (4)$$

$$\mathcal{P} \models \mathsf{G}\,(\mathsf{valid}_Z \wedge \mathsf{stop}_Z \Rightarrow (\mathsf{valid}_Z)^+) \quad (5)$$

$$\mathcal{P} \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Z \quad (6)$$

Let $\mathcal{C}_Z$ be the $\{Z, \mathsf{valid}_Z, \mathsf{stop}_Z\}$-system characterized by condition (4)—the largest (in the sense of behavior inclusion) of the systems satisfying this condition. Similarly, let $\mathcal{P}_Z$ be the $\{Z, \mathsf{valid}_Z, \mathsf{stop}_Z\}$-system characterized by properties (5) and (6). Finally, define *the* $[I, O]$*-elastic environment* to be the system

$$\mathsf{Env}_{I,O} = \textstyle\bigsqcup_{X\in I} \mathcal{P}_X \sqcup \bigsqcup_{Y\in O} \mathcal{C}_Y .$$

Note that $\mathsf{Env}_{I,O}$ is only a system; it is not functional and so is not a machine.

When a system satisfying the persistence and liveness conditions (1-3) is coupled with a matching elastic environment, the transfer on all data wires never comes to a stall:

Lemma 3 (Liveness): If $\mathcal{S}$ satisfies (1-3), then for every behavior ω of $\mathcal{S} \sqcup \mathsf{Env}_{I,O}$, all the component sequences of the transfer behavior ω^{T} are infinite.

As an immediate consequence of Liveness Lemma, if $\mathcal{S}$ satisfies (1-3), then

$$\mathcal{S}^{\mathsf{T}} = \{\omega^{\mathsf{T}} \mid \omega \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}\}$$

is a well-defined (I, O)-system.

Definition 8: An $[I, O]$-system $\mathcal{S}$ is an $[I, O]$*-elastic machine* if it satisfies the properties (1-3) and the associated system $\mathcal{S}^{\mathsf{T}}$ is deterministic.

The liveness conditions (2,3) are visibly related to causality at the transfer level: k transfers on the input channels imply k transfers on the output channels in the cooperating environment. Thus, it is not surprising (even though the proof is not obvious) that the determinism postulated in Definition 8 suffices to derive the causality of $\mathcal{S}^{\mathsf{T}}$:

Theorem 2: If $\mathcal{S}$ is an $[I, O]$-elastic machine, then $\mathcal{S}^{\mathsf{T}}$ is an (I, O)-machine.

In the situation of Definition 8, we say that $\mathcal{S}$ is an *elasticization* of $\mathcal{S}^{\mathsf{T}}$ and that $\mathcal{S}^{\mathsf{T}}$ is the *transfer machine* of $\mathcal{S}$.

IV. Elastic Networks

An *elastic network* $\mathcal{N}$ is given by a set of elastic machines $\mathcal{S}_1, \ldots, \mathcal{S}_m$ with no shared wires, together with a set of channel pairs $(X_1, Y_1), \ldots, (X_n, Y_n)$, where the X_i are n distinct input channels and the Y_i are n distinct output channels. As a network of standard machines, the elastic network $\mathcal{N}$ is defined by

$$\mathcal{N} = \langle \mathcal{S}_1, \ldots, \mathcal{S}_m \mid X_i = Y_i, \mathsf{valid}_{X_i} = \mathsf{valid}_{Y_i}, \mathsf{stop}_{X_i} = \mathsf{stop}_{Y_i}\ (1 \leq i \leq n)\rangle$$

for which we will use the shorter notation

$$\mathcal{N} = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \parallel X_1 = Y_1, \ldots, X_n = Y_n \rangle\!\rangle.$$

We will define a graph that encodes the sequentiality information about the network $\mathcal{N}$ and prove in Theorem 4 that acyclicity of that graph implies that $\mathcal{N}$ is an elastic machine and that $\mathcal{N}^{\mathsf{T}} = \langle \mathcal{S}_1^{\mathsf{T}}, \ldots, \mathcal{S}_m^{\mathsf{T}} \mid X_1 = Y_1, \ldots, X_n = Y_n\rangle$.

A. Elastic Feedback

Elastic feedback is a simple case of elastic network:

$$\langle\!\langle \mathcal{S} \parallel P = Q \rangle\!\rangle = \langle \mathcal{S} \mid P = Q, \mathsf{valid}_P = \mathsf{valid}_Q, \mathsf{stop}_P = \mathsf{stop}_Q\rangle.$$

Definition 9: Suppose $\mathcal{S}$ is an elastic machine. An input-output channel pair (P, Q) will be called *sequential* for $\mathcal{S}$ if

$$\mathcal{S} \models \mathsf{G}\left(\begin{array}{l} \mathsf{min\_tct}_{I\cup O} = \mathsf{tct}_Q \\ \wedge\ \mathsf{min\_tct}_{I-\{P\}} > \mathsf{tct}_Q \end{array} \Rightarrow \mathsf{F}\,\mathsf{valid}_Q \right). \quad (7)$$

Condition (7) is a strengthening of the liveness condition (2) for channel Q. It expresses a degree of independence of the output channel Q from the input channel P; e.g., the first token at Q need not wait for the arrival of the first token at P. This independence can be achieved in the system by storing some tokens inside, between these two channels. Note that (7) does not guarantee that connecting channels P and Q would not introduce ordinary combinational cycles. Therefore the acyclicity condition in the following theorem is required to ensure (by Theorem 1) that the elastic feedback, viewed as an ordinary network, is a machine.

Theorem 3: Let $\mathcal{S}$ be an elastic machine and $\mathcal{F}$ the elastic feedback system $\langle\!\langle \mathcal{S} \parallel P = Q\rangle\!\rangle$. If the channel pair (P, Q) is sequential for $\mathcal{S}$, then: (a) the wire pair (P, Q) is sequential for $\mathcal{S}^{\mathsf{T}}$. If, in addition, $\Delta(\mathcal{F})$ is acyclic, then: (b) $\mathcal{F}$ is an elastic machine, and (c) $\mathcal{F}^{\mathsf{T}} = \langle \mathcal{S}^{\mathsf{T}} \mid P = Q\rangle$.

B. Main Theorems

Sequentiality of two channel pairs $(P, Q), (P', Q)$ of an elastic machine does not imply their "simultaneous sequentiality"

$$\mathcal{S} \models \mathsf{G}\left(\begin{array}{c} \mathsf{min\_tct}_{I\cup O} = \mathsf{tct}_Q \\ \wedge\ \mathsf{min\_tct}_{I-\{P,P'\}} > \mathsf{tct}_Q \end{array} \Rightarrow \mathsf{F}\,\mathsf{valid}_Q\right).$$

This deviates from the situation with ordinary machines, where the analogous property holds and is instrumental in the proof of Combinational Loop Theorem.

To justify multiple feedback on elastic machines, we have thus to postulate that simultaneous sequentiality is true where required. Specifically, we demand that elastic machines come with simultaneous sequentiality information: If $\mathcal{S}$ is an $[I, O]$-elastic machine, then for every $Y \in O$ a set $\delta(Y) \subseteq I$ is given so that

$$\mathcal{S} \models \mathsf{G}\left(\begin{array}{c} \mathsf{min\_tct}_{I\cup O} = \mathsf{tct}_Y \\ \wedge\ \mathsf{min\_tct}_{I-\delta(Y)} > \mathsf{tct}_Y \end{array} \Rightarrow \mathsf{F}\,\mathsf{valid}_Y\right). \quad (8)$$

Note that if $P \in \delta(Q)$, then the pair (P, Q) is sequential, but the converse is not implied. A function $\delta\colon O \to 2^I$ with the property (8) will be called a *sequentiality interface* for $\mathcal{S}$.

For an $[I, O]$-elastic machine $\mathcal{S}$ with a sequentiality interface δ, we define $\varDelta^{\mathrm{e}}(\mathcal{S}, \delta)$ to be the graph with the vertex set $I \cup O$ and directed edges (X, Y) where $X \notin \delta(Y)$. By Theorem 3(a), $\varDelta^{\mathrm{e}}(\mathcal{S}, \delta)$ contains $\Delta(\mathcal{S}^{\mathsf{T}})$ as a subgraph.

Given an elastic network $\mathcal{N} = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,[\!]\, X_1 = Y_1, \ldots, X_n = Y_n \rangle\!\rangle$, where each $\mathcal{S}_i$ comes equipped with a sequentiality interface δ_i, its graph $\varDelta^{\mathrm{e}}(\mathcal{N})$ is by definition the direct sum of graphs $\varDelta^{\mathrm{e}}(\mathcal{S}_1, \delta_1), \ldots, \varDelta^{\mathrm{e}}(\mathcal{S}_m, \delta_m)$ with each vertex X_i ($1 \le i \le n$) identified with the corresponding vertex Y_i.

Theorem 4: If the graphs $\Delta(\mathcal{N})$ and $\varDelta^{\mathrm{e}}(\mathcal{N})$ are acyclic, then the network system $\mathcal{N}$ is an elastic machine, the corresponding non-elastic system $\bar{\mathcal{N}} = \langle \mathcal{S}_1^{\mathsf{T}}, \ldots, \mathcal{S}_m^{\mathsf{T}} \,|\, X_1 = Y_1, \ldots, X_n = Y_n \rangle$ is a machine, and $\mathcal{N}^{\mathsf{T}} = \bar{\mathcal{N}}$.

As in Theorem 3, acyclicity of $\Delta(\mathcal{N})$ is needed to ensure (by Theorem 1) that $\mathcal{N}$ defines a machine. Elasticization procedures (e.g. [4]) will typically produce elastic components with enough sequential input-output wire pairs, so that $\Delta(\mathcal{N})$ will be acyclic as soon as $\varDelta^{\mathrm{e}}(\mathcal{N})$ is acyclic.

Note, however, that cycles in $\varDelta^{\mathrm{e}}(\mathcal{N})$ need not correspond to combinational cycles in $\mathcal{N}$ seen as an ordinary network, since empty buffers with sequential elements cutting the combinational feedbacks may be inserted into $\mathcal{N}$. Even though non-combinational in the ordinary sense, these cycles contain no tokens and therefore no progress along them can be made.

Theorem 4 impies that insertion of empty elastic buffers does not affect the basic functionality of an elastic network, as illustrated in Figure 2(b).

Definition 10: An *empty elastic buffer* is an elastic machine $\mathcal{S}$ such that $\mathcal{S}^{\mathsf{T}} = \mathsf{Conn}(X, Y)$ for some X, Y.

Theorem 5 (Buffer Insertion Theorem): Suppose $\mathcal{B}$ is an empty elastic buffer with channels X, Y. Let $\mathcal{N} = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,[\!]\, X_1 = Y_1, \ldots, X_n = Y_n \rangle\!\rangle$ and $\mathcal{M} = \langle\!\langle \mathcal{B}, \mathcal{S}_1, \ldots, \mathcal{S}_m \,[\!]\, X = Y_1, X_1 = Y, X_2 = Y_2, \ldots, X_n = Y_n \rangle\!\rangle$. If $\Delta(\mathcal{N})$, $\Delta(\mathcal{M})$, and $\varDelta^{\mathrm{e}}(\mathcal{N})$ are acyclic, then $\mathcal{M}$ is an elastic machine, and $\mathcal{M}^{\mathsf{T}} = \mathcal{N}^{\mathsf{T}}$.

The precise relationship between graphs $\Delta(\mathcal{M})$ and $\Delta(\mathcal{N})$ can be easily described. In practice they are at the same time acyclic or not, as a consequence of sequentiality of sufficiently many input-output wire pairs of $\mathcal{B}$.

V. Elastic vs. Patient Systems

Elastic machines and *patient processes* of [2] provide two formalizations of the intuitive concept of latency-insensitive circuits. In this section we address their connections and differences. We begin with an overview of [2], using a minimalistic approach and terminology that differs from the original. We believe, however, that Definition 11 below matches the original definion accurately in most important aspects.

A. Patient Systems

The notation A^* is for the set of finite sequences over A. A *finitary* W-system, by definition, is a set of behaviors σ such that $\sigma.w$ is a finite sequence for every $w \in W$.

A *stalling stream* over A is a stream over $A \cup \{\Box\}$. We will refer to $\Box$ as the *bubble* and to elements of A as *tokens*. We will consider only stalling streams that contain finitely many tokens. If a is such a stream, let $\overline{a} \in A^*$ denote the sequence over A obtained by dropping all bubbles from a. Clearly, a is determined by $\overline{a}$ and the sequence $\partial(a) \in \mathbb{N}^*$ of lengths of bubble sequences between consecutive tokens of a. For example, if

$$a = (\Box, \Box, 7, \Box, 4, 5, \Box, \Box, \Box, 8, \ldots) \quad (9)$$

we have $\overline{a} = (7, 4, 5, 8, \ldots)$ and $\partial(a) = (2, 1, 0, 3, \ldots)$. Two stalling streams a, b are *latency equivalent*, written $a \doteq b$, when $\overline{a} = \overline{b}$. Note that $a \doteq \overline{a}$.

By definition, a *stalling* W*-system* is a set of behaviors σ such that for every $w \in W$, $\sigma.w$ is a stalling stream over $\mathsf{type}(w)$. Latency equivalence extends to W-behaviors and W-systems: $\sigma \doteq \tau$ iff $\sigma.w \doteq \tau.w$ holds for every $w \in W$; $\mathcal{S} \doteq \mathcal{S}'$ iff for every $\sigma \in \mathcal{S}$ ($\sigma \in \mathcal{S}'$) there exists $\tau \in \mathcal{S}'$ ($\tau \in \mathcal{S}$) such that $\sigma \doteq \tau$.

A stalling W-system $\mathcal{S}$ determines a standard finitary W-system $\mathcal{S}^{\mathsf{T}} = \{\overline{\sigma} \mid \sigma \in \mathcal{S}\}$, where $\overline{\sigma}$ is given by $\overline{\sigma}.w = \overline{\sigma.w}$ (for all $w \in W$). Clearly, $\mathcal{S}^{\mathsf{T}} \doteq \mathcal{S}$.

Stalling the k-th token of a by d steps produces a latency equivalent stream that will be denoted $\mathsf{stall}(a, k, d)$. Omitting the easy definition, we give an example: if a is as in (9), then

$$\mathsf{stall}(a, 1, 3) = (\Box, \Box, 7, \Box, \Box, \Box, \Box, 4, 5, \Box, \Box, \Box, 8, \ldots)$$

Definition 11: Let $\prec$ be a well-founded order[1] on W and let $D > 0$. A *patient* W*-system* (relative to $\prec$ and D) is a

[1] Introduction of a well-founded ordering of wires is motivated in [2] with the purpose of modeling combinational dependencies, but such dependencies in patient systems are not discussed in any detail. Moreover, the ordering of wires is implicitly assumed to be *total* in [2], which is somewhat unnatural. For instance, when constructing a patient adder with inputs u, v and output w, one has two ordering choices: $u \prec_1 v \prec_1 w$ and $v \prec_2 u \prec_2 w$. It is not clear that a patient adder in the $\prec_1$-sense will be patient in the $\prec_2$-sense too.

stalling system $\mathcal{P}$ such that for every $\sigma \in \mathcal{P}$, every $u \in W$, and every $k \geq 0$ there exists $\sigma' \in \mathcal{P}$ such that

$$\text{(Pat-1)} \quad \sigma'.u = \mathsf{stall}(\sigma.u, k, 1)$$

and for every $v \neq u$ there exists $d_v \leq D$ such that

$$\text{(Pat-2)} \quad \sigma'.v = \begin{cases} \mathsf{stall}(\sigma.v, k, d_v) & \text{if } u \prec v \\ \mathsf{stall}(\sigma.v, k+1, d_v) & \text{otherwise} \end{cases}$$

The main results of [2] can now be summarized:

1) a theorem saying that the composition of patient systems (with the same W, $\prec$, and D) is a patient system;
2) the definition and analysis of *patient buffers*, i.e. patient systems $\mathcal{B}$ such that $\mathcal{B}^{\mathsf{T}} = \mathsf{Conn}^{\text{fin}}(u, v)$—the finitary connection system;
3) a general construction that, for a given finitary system $\mathcal{M}$ without combinational dependencies (model of a Moore machine), produces a patient system $\mathcal{P}$ such that $\mathcal{P} \mathrel{\mathring{=}} \mathcal{M}$.

B. Comparison

The formalization given by patient systems is at a higher level of abstraction. While elastic machines deal explicitly with handshaking signals between communicating systems, patient systems communicate purely in the token/bubble language.

Given an elastic (as defined in Section III) $[I, O]$-system $\mathcal{E}$, the corresponding stalling $(I \cup O)$-system $\mathcal{E}^{\square}$ is obtained by projecting the finite-transfer behaviors of $\mathcal{E}$ to data wires and replacing data items on each wire with $\square$ at all cycles where transfer along that wire does not occur. Precisely, let $\mathcal{E}^{\mathsf{F}}$ be the subset of $\mathcal{E}$ consisting of all behaviors ω such that $\omega^{\mathsf{T}}.Z$ is finite for all channels Z.[2] Then, given $\omega \in \mathcal{E}^{\mathsf{F}}$, we define a stalling $(I \cup O)$-behavior $\omega^{\square}$ by

$$(\omega^{\square}.Z)[i] = \begin{cases} (\omega.Z)[i] & \text{if } (\omega.\mathsf{valid}_Z)[i] \wedge \neg(\omega.\mathsf{stop}_Z)[i] \\ \square & \text{otherwise} \end{cases}$$

and finally we define the stalling system $\mathcal{E}^{\square}$ as the set of all such behaviors $\omega^{\square}$. Clearly, the system $(\mathcal{E}^{\square})^{\mathsf{T}}$ is the finitary version of the standard machine $\mathcal{E}^{\mathsf{T}}$.

Now we can address some questions pertinent to the comparison of patient processes vs. elastic machines.

Are patient processes more general? The answer is "no" because there exist elastic machines $\mathcal{E}$ such that $\mathcal{E}^{\square}$ is not patient. To see this, consider an elastic machine $\mathcal{E}$ that starts offering new valid outputs on channel u only on even cycles. (The existence of such elastic machines is obvious.) Observe that $\sigma.u = (\square, 7, 9, \ldots)$ is possible for some behavior σ of $\mathcal{E}^{\square}$ (token 7, even though transmitted on cycle 1 was first offered on cycle 0). Then $\mathsf{stall}(\sigma.u, 0, 1) = (\square, \square, 7, 9, \ldots)$ must also be part of a behavior of $\mathcal{E}^{\square}$, by condition (Pat-1) of Definition 11. This implies that token 9 is first offered on cycle 3, contrary to our assumption.

The above example can be viewed as an indication that the condition (Pat-1) is too restrictive. It would be interesting to see if an appropriate modification of (Pat-1) results in a definition of patient processes that captures elastic machines.

[2] One can prove that $\mathcal{E}$ is the set of all limits of behaviors of $\mathcal{E}^{\mathsf{F}}$ and so $\mathcal{E}$ is determined by $\mathcal{E}^{\mathsf{F}}$.

Are elastic machines more general? The answer is an easy "no" since, for example, the set of all possible stalling W-behaviors is a patient system in the sense of Definition 11. However, if one adds to Definition 11 a reasonable requirement that a patient system be a machine, the answer is not immediately clear.

Which formalization is easier to use? Without offering a definitive answer, we would argue that verifying that a low-level design (RTL, say) implements an elastic machine would be easier than verifying that it implements a patient system. The bottom line is that the conditions for a system to be an elastic machine are expressible as temporal properties of suitably constructed infinite-state models. This is not obvious for the determinism condition for $\mathcal{S}^{\mathsf{T}}$ in Definition 8, but can be done by replacing determinism with causality and introducing auxiliary variables for sequences of transferred values over channels. Even though (e.g., because of infinite counters involved) these conditions are not directly checkable by the existing model checking technology, there are palpable opportunities to find manageable stronger conditions that taken together imply elasticity (e.g., postulating a limit on the token count differences between channels eliminates the need for infinite counters). On the other hand, the definition of a patient system, being of the form "for every behavior σ, there exists a behavior σ' such that ..." appears to us to be intrinsically more complex. Our only positive conclusion, however, is that the mechanical checking of either of the definitions is an open problem deserving further study.

VI. Conclusion

We have presented a theory of elastic machines that gives an easy-to-check condition for the compositional theorem of the form "an elasticization of a network of ordinary components is equivalent to the network of components' elasticizations". Verification of a particular implementation is reduced to proving that conditions of Definition 8 are satisfied for all elastic components used, and that the graph $\varDelta^{\mathrm{e}}(\mathcal{N}^{\mathrm{e}})$ is acyclic for every network $\mathcal{N}$ to which the elasticization is applied. While the definition of the graphs $\varDelta^{\mathrm{e}}$ may appear complex because of the sequentiality interfaces involved, it should be noted that the elasticization procedures, e.g. [4], are reasonably expected to completely preserve sequentiality: a channel P belongs to $\delta(Q)$ if the wire-pair (P, Q) is sequential in the original non-elastic machine. This ensures $\varDelta^{\mathrm{e}}(\mathcal{N}^{\mathrm{e}}) = \Delta(\mathcal{N})$ and so testing for sequentiality is done at the level of ordinary networks.

Future work will be focused on proving correctness of particular elasticization methods, on techniques for mechanical verification of elasticity, and on extending the theory to more advanced protocols.

Acknowledgments: Luca Carloni clarified some details of [2]. Ken McMillan pointed out several inaccuracies in a previous version of the paper and further clarified [2] for us. Gerard Berry, Ching-Tsun Chou, John Harrison, and the anonymous reviewers provided useful remarks. We are grateful for all the help we received.

References

[1] G. Berry. *The Constructive Semantics of Pure Esterel*. Draft book, available at http://www.esterel.org, version 3, July 1999.

[2] L. P. Carloni, K. L. McMillan, and A. L. Sangiovanni-Vincentelli. Theory of latency-insensitive design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, 20(9):1059–1076, September 2001.

[3] L. P. Carloni and A. L. Sangiovanni-Vincentelli. Coping with latency in SoC design. *IEEE Micro, Special Issue on Systems on Chip*, 22(5):12, October 2002.

[4] J. Cortadella, M. Kishinevsky, and B. Grundmann. Synthesis of synchronous elastic architectures. In *Proc. Digital Automation Conference (DAC)*, July 2006.

[5] S. A. Edwards and E. A. Lee. The semantics and execution of a synchronous block-diagram language. *Sci. Comput. Program.*, 48(1):21–42, 2003.

[6] H. M. Jacobson et al. Synchronous interlocked pipelines. In *Proc. Int. Symp. on Advanced Research in Asynchronous Circuits and Systems*, pages 3–12, 2002.

[7] S. Krstić, J. Cortadella, M. Kishinevsky, and J. O'Leary. Synchronous elastic networks. Available at `www.lsi.upc.edu/~jordicf/gavina/BIB/reports/fmcad06_ext.pdf`, 2006.

[8] E. A. Lee and A. Sangiovanni-Vincentelli. A framework for comparing models of computation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1217–1229, 1998.

[9] E. A. Lee, H. Zheng, and Y. Zhou. Causality interfaces and compositional causality analysis. Invited paper in *Foundations of Interface Technologies (FIT 2005)*, available at `http://ptolemy.eecs.berkeley.edu/publications`.

[10] R. Manohar and A. J. Martin. Slack elasticity in concurrent computing. In *Proc. 4th Int. Conf. on the Mathematics of Program Construction*, volume 1422 of *Lecture Notes in Computer Science*, pages 272–285, 1998.

[11] J. Matthews. Recursive function definition over coinductive types. In *TPHOLs '99: Proc. the 12th Int. Conf. on Theorem Proving in Higher Order Logics*, pages 73–90, London, UK, 1999. Springer-Verlag.

[12] S. Suhaib, D. Berner, D. Mathaikutty, J.-P. Talpin, and S. Shukla. Presentation and formal verification of a family of protocols for latency insensitive design. Technical Report 2005-02, FERMAT, Virginia Tech, 2005.

[13] J. Vuillemin. On circuits and numbers. *IEEE Transactions on Computers*, 43(8):868–879, 1994.

APPENDIX
PROOFS AND AUXILIARY RESULTS

Contents

A. Proof of Lemma 1

Remark. Lemma 1 and Lemma 4 below are well-known facts. We give their proofs for the sake of completeness.

Suppose a and b are both fixpoints of f. By the contraction property, $a \sim_k b \Rightarrow a \sim_{k+1} b$ holds for every k. Since $a \sim_0 b$, it follows that $a \sim_k b$ holds for every k, so $a = b$. This proves the uniqueness part.

Suppose p is any finite sequence of length k and let S_p be the set of all streams $a \in A^\infty$ such that p is a prefix of a. It follows from the contraction condition that all streams in the set $\{f(a) \mid a \in S_p\}$ have a common prefix of length $k+1$. We will denote that common prefix by $p^\dagger$. Thus, for every $p \in A^*$, there exists $p^\dagger \in A^*$ such that $|p^\dagger| = 1 + |p|$ and

$$\forall a \in A^\infty.\ \ p \preceq a \Rightarrow p^\dagger \preceq f(a), \tag{10}$$

where $\preceq$ stands for the prefix relation. This implies

$$\forall p, q \in A^*.\ \ p \preceq q \Rightarrow p^\dagger \preceq q^\dagger. \tag{11}$$

Consider now the sequence $\epsilon, \epsilon^\dagger, \epsilon^{\dagger\dagger}, \ldots$, where ϵ is the empty sequence. Since $\epsilon \preceq \epsilon^\dagger$, by repeated use of (11), we obtain $\epsilon \preceq \epsilon^\dagger \preceq \epsilon^{\dagger\dagger} \preceq \cdots$. Since the sequences in this chain have increasing length, there is a unique "limit" $a \in A^\infty$ satisfying $\epsilon \preceq \epsilon^\dagger \preceq \epsilon^{\dagger\dagger} \preceq \cdots \preceq a$. By (10), the relations $\epsilon \preceq a, \epsilon^\dagger \preceq a, \epsilon^{\dagger\dagger} \preceq a, \ldots$ imply $\epsilon^\dagger \preceq f(a), \epsilon^{\dagger\dagger} \preceq f(a), \epsilon^{\dagger\dagger\dagger} \preceq f(a), \ldots$ Thus, $\epsilon \preceq \epsilon^\dagger \preceq \epsilon^{\dagger\dagger} \preceq \cdots \preceq f(a)$, which forces $a = f(a)$ (by the uniqueness of the "limit" stream), proving the existence of the fixpoint. This finishes the proof of Lemma 1.

The following fact will be used in the proof of Lemma 2. To state it, we need a definition: A sequence a_n of elements of A^∞ (i.e. a sequence of streams over A) *converges* to $a \in A^\infty$ if every prefix of a is the prefix of all but finitely many of the a_n. More formally,

$$\lim a_n = a \ \text{ iff } \ \forall n \geq 0. \exists k \geq 0. \forall m \geq 0.\ \ m \geq k \Rightarrow a_m \sim_n a.$$

Lemma 4: If $f\colon A^\infty \longrightarrow A^\infty$ is contractive and $\bar{a}$ is an arbitrary element of A^∞ then the sequence of iterations $f^n(\bar{a})$ $(n \geq 0)$ converges to the fixpoint of f.

Proof. By Lemma 1, f has a unique fixpoint a. We prove that $f^m(\bar{a}) \sim_n a$ holds for every m, n such that $m \geq n$. (This clearly implies the lemma.) Our statement is obvious for $n = 0$. Arguing by induction on n, for $n \geq 1$ we have $f^{m-1}(\bar{a}) \sim_{n-1} a$ (induction hypothesis), so applying f to both sides and using the contraction property and $f(a) = a$ we derive $f^m(\bar{a}) \sim_n a$. □

B. Proof of Lemma 2

Sequentiality of the pair (u, v) implies that for every $\sigma \in [\![I - \{u\}]\!]$ there exists a unique fixpoint a_σ of F^σ_{uv}. (We continue with the notation from the paragraph preceding the lemma.) Consequently, $F(\sigma * [u \mapsto a])$ has the form $\tau * [v \mapsto a]$ for a unique $\tau \in [\![O - \{v\}]\!]$. Let $G\colon [\![I - \{u\}]\!] \longrightarrow [\![O - \{v\}]\!]$ be the function that associates τ with σ in the way just described. It is easy to verify (by definition of G) that

$$\sigma * \tau \in \langle \mathcal{S} \mid u = v \rangle \ \text{ iff } \ G(\sigma) = \tau$$

holds for every $\sigma \in [\![I - \{u\}]\!], \tau \in [\![O - \{v\}]\!]$, so it only remains to prove that G satisfies the causality condition. That in turn is a simple consequence of the causality of F and the causality of the parametrized fixpoint operator that associates a_σ with σ. By the latter we mean that $a_\sigma \sim_k a_{\sigma'}$ always follows from $\sigma \sim_k \sigma'$. This we state now and prove as a separate lemma.

Lemma 5: If (u, v) is a sequential pair for F, $\sigma \sim_k \sigma'$, and a_σ and $a_{\sigma'}$ are the fixpoints of F^σ_{uv} and $F^{\sigma'}_{uv}$, then $a_\sigma \sim_k a_{\sigma'}$.

Proof. Let $A = [\![u]\!] = [\![v]\!]$ and let $H\colon [\![I - \{u\}]\!] \times A \longrightarrow A$ be given by $H(\sigma, a) = F(\sigma * [u \mapsto a]).v$. Our sequentiality assumption reads as follows:

$$\forall \sigma, \sigma', a, a'.\ \ \sigma \sim_{k+1} \sigma' \wedge a \sim_k a' \ \Rightarrow\ H(\sigma, a) \sim_{k+1} H(\sigma', a') \tag{12}$$

Starting with an arbitrary $\bar{a} \in A$, define

$$a_n = \begin{cases} \bar{a} & \text{if } n = 0 \\ H(\sigma, a_{n-1}) & \text{if } \ n \geq 1 \end{cases}$$

$$a'_n = \begin{cases} \bar{a} & \text{if } n = 0 \\ H(\sigma', a'_{n-1}) & \text{if } \ n \geq 1 \end{cases}$$

Since a_σ is the fixpoint of the function $F^\sigma_{uv} : a \mapsto H(\sigma, a)$, we have (by Lemma 4) that $a_\sigma = \lim a_n$, and similarly $a_{\sigma'} = \lim a'_n$. By induction, it follows from (12) that $a_\sigma \sim_n a_n$ and $a_{\sigma'} \sim_n a'_n$ hold for every n. Also by induction, it follows from (12) that $a_n \sim_n a'_n$ holds for all $n \leq k$. Therefore, $a_\sigma \sim_k a_{\sigma'}$, as required. □

C. Bekić Lemma

[*Bibliographical information.* Bekić Lemma asserts that the fixpoint of a two-variable function can be computed by iterating the fixpoint operators along the two coordinates. It holds in various contexts. The original result is in: Hans Bekić, *Definable operations in general algebras, and the theory of* *automata and flowcharts*, LNCS, vol. 177, pp. 3–35, Springer 1984.]

We begin with a generalization of the concept of sequentiality.

Definition 12: Let $\mathcal{S}$ be an (I, O)-machine given by $F\colon [\![I]\!] \to [\![O]\!]$ and let $A \subseteq I$, $v \in O$. The pair (A, v) is *sequential* for $\mathcal{S}$ if for every $k \geq 0$, every $\sigma, \sigma' \in [\![I - A]\!]$, and every $\tau, \tau' \in [\![A]\!]$ one has

$$\sigma \sim_{k+1} \sigma \wedge \tau \sim_k \tau' \ \Rightarrow\ F(\sigma * \tau).v \sim_{k+1} F(\sigma' * \tau').v.$$

Lemma 6: (A, v) is sequential if and only if (u, v) is sequential for every $u \in A$.

Proof. The "only if" part is trivial. We prove the "if" part for the case when A has two elements. The general case is only notationally more difficult.

Thus, suppose $A = \{x, y\}$ and both (x, v) and (y, v) are sequential. We need to prove

$$\begin{aligned} & \sigma \sim_{k+1} \sigma \wedge a \sim_k a' \wedge b \sim_k b' \\ \Rightarrow\ & F(\sigma * [x \mapsto a] * [y \mapsto b]).v \\ & \sim_{k+1} F(\sigma' * [x \mapsto a'] * [y \mapsto b']).v. \end{aligned}$$

For the proof, just observe that both $F(\sigma * [x \mapsto a] * [y \mapsto b]).v$ and $F(\sigma' * [x \mapsto a'] * [y \mapsto b']).v$ are equivalent in the sense of $\sim_{k+1}$ with $F(\sigma' * [x \mapsto a'] * [y \mapsto b]).v$ as a consequence of the assumed sequentiality of (x, v) and (y, v). □

Lemma 7 (Bekić Lemma): Suppose (u, w) and (v, z) are sequential pairs for a machine $\mathcal{S}$ and suppose that the four wires involved are distinct. If one of the pairs (u, z), (v, w) is also sequential, then (v, z) is sequential for $\langle \mathcal{S} \mid u = w \rangle$, (u, w) is sequential for $\langle \mathcal{S} \mid v = z \rangle$, and (therefore) the system $\langle \mathcal{S} \mid u = w, v = z \rangle$ is a machine.

Proof. In view of Lemma 6, it suffices to prove that (v, z) is sequential for $\langle \mathcal{S} \mid u = w \rangle$ under either of the following two assumptions:

(i) (u, w) and $(\{u, v\}, z)$ are sequential;
(ii) (v, z) and $(\{u, v\}, w)$ are sequential.

Thus, our goal is to prove

$$\begin{aligned} & \sigma \sim_{k+1} \sigma' \wedge b \sim_k b' \\ \Rightarrow\ & G(\sigma * [v \mapsto b]).z \sim_{k+1} G(\sigma' * [v \mapsto b']).z, \end{aligned}$$

where G is the function corresponding to the system $\langle \mathcal{S} \mid u = w \rangle$. We can restate this in a more convenient form

$$\begin{aligned} & \sigma \sim_{k+1} \sigma' \wedge b \sim_k b' \\ \wedge\ & G(\sigma * [v \mapsto b]) = \tau * [z \mapsto c] \\ \wedge\ & G(\sigma' * [v \mapsto b']) = \tau * [z \mapsto c'] \\ \Rightarrow\ & c \sim_{k+1} c' \end{aligned}$$

By definition of G, this can be further restated as

$$\begin{aligned} & \sigma \sim_{k+1} \sigma' \wedge b \sim_k b' \wedge a \sim_k a' \\ \wedge\ & F(\sigma * [v \mapsto b] * [u \mapsto a]) = \tau * [z \mapsto c] * [w \mapsto a] \\ \wedge\ & F(\sigma' * [v \mapsto b'] * [u \mapsto a']) = \tau * [z \mapsto c'] * [w \mapsto a'] \\ \Rightarrow\ & c \sim_{k+1} c' \end{aligned} \tag{13}$$

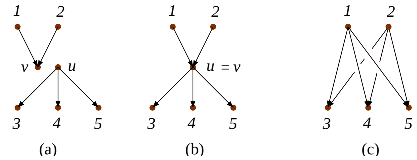

Fig. 6. (a) Γ, (b) $\Gamma_{[u=v]}$, (c) Γ_{uv}

where a and a' are the unique fixpoints of F^{σ}_{uw} and $F^{\sigma'}_{uw}$ respectively. We also added $a \sim_k a'$ to the assumptions; this is justified because $a \sim_k a'$ follows from the other assumptions by Lemma 5.

We finish the proof now by deriving (13) from any of our two assumptions (i),(ii). If (i) holds, then (13) immediately follows. If (ii) holds, we first derive $a \sim_{k+1} a'$ from the assumptions of (13) and sequentiality of $(\{u, v\}, w)$. Then with $a \sim_{k+1} a'$ in place of $a \sim_k a'$ in the assumptions of (13), the conclusion $c \sim_{k+1} c'$ follows by sequentiality of (v, z). □

D. Proof of Theorem 1

The notation $\Gamma_{[u_1=v_1,\ldots,u_n=v_n]}$ will stand for the quotient graph obtained by identifying the vertices u_i and v_i ($i = 1, \ldots, n$) in Γ.

For a given directed graph $\mathit{\Gamma}$, a source vertex[3] u of $\mathit{\Gamma}$, and a sink vertex v of $\mathit{\Gamma}$, let $\mathit{\Gamma}_{uv}$ denote the graph obtained from $\mathit{\Gamma}$ by:

(1) removing u, v and all edges incident with them;
(2) adding an edge (w, z) for all w, z such that (w, u) and (v, z) are edges of $\mathit{\Gamma}$.

(See Figure 6.)

Definition 13: If Γ and Γ' are directed graphs such that every vertex of Γ is a vertex of Γ' and every two vertices joined by an edge in Γ are joined by a path in Γ', then we say that Γ is *immersed* in Γ' and write $\Gamma \hookrightarrow \Gamma'$.

In the main text, we used the same notation Δ for dependency graphs of machines and systems. This is slightly ambiguous, however. For example, if $\mathcal{S}$ is an (I, O)-machine, then $\mathcal{N} = \langle \mathcal{S} \mid u = v \rangle$ is a system, but at the same time an $(I - \{u\}, O - \{v\})$-machine. When $\mathcal{N}$ is regarded as a network, then $\Delta(\mathcal{N})$ is the quotient graph of $\Delta(\mathcal{S})_{[u=v]}$. When $\mathcal{N}$ is regarded as a machine, the vertex set of $\Delta(\mathcal{N})$ is $I \cup O - \{u, v\}$. For the proof, we need to disambiguate this notation.

Definition 14: The *dependency graph* $D(\mathcal{S})$ *of an* (I, O)-*machine* $\mathcal{S}$ has $I \cup O$ as the vertex set, and $\{(u, v) \in I \times O \mid (u, v)$ is not sequential for $\mathcal{S}\}$ as its set of directed edges. The *dependency graph* $\Delta(\mathcal{N})$ *of a network* $\mathcal{N} = \langle \mathcal{S}_1, \ldots, \mathcal{S}_m \mid u_1 = v_1, \ldots, u_n = v_n \rangle$ is $(D(\mathcal{S}_1) \sqcup \cdots \sqcup D(\mathcal{S}_m))_{[u_1=v_1,\ldots,u_n=v_n]}$.

Definition 15: We write $\mathcal{N} \sim \mathcal{N}'$ when the networks $\mathcal{N}$ and $\mathcal{N}'$ describe the same system.

To appreciate Definition 15, note that networks are a form of system descriptions, so that, strictly speaking, a network

[3] A source vertex is a vertex with no incoming edges; a sink vertex is a vertex with no outgoing edges.

is not a system, although it uniquely determines one. The equivalence $\mathcal{N} \sim \mathcal{N}'$ implies $D(\mathcal{N}) = D(\mathcal{N}')$ but it does not imply $\Delta(\mathcal{N}) = \Delta(\mathcal{N}')$.

Lemma 8: If u, v are vertices of Γ and if $\Gamma \hookrightarrow \Gamma'$, then $\Gamma_{[u=v]} \hookrightarrow \Gamma'_{[u=v]}$.

Proof. Trivial. □

Lemma 9: If u is a source of Γ and v is a sink of Γ, then $\Gamma_{uv} \hookrightarrow \Gamma_{[u=v]}$.

Proof. Easy. (Use Figure 6.) □

Lemma 10: If (u, v) is sequential for $\mathcal{S}$, then $D(\langle \mathcal{S} \,|\, u = v\rangle) \hookrightarrow D(\mathcal{S})_{[u=v]}$.

Proof. The proof follows from $D(\langle \mathcal{S} \,|\, u = v\rangle) \hookrightarrow D(\mathcal{S})_{uv}$ and $D(\mathcal{S})_{uv} \hookrightarrow D(\mathcal{S})_{[u=v]}$. The second of these relations is an instance of Lemma 9. For the first, we prove that $D(\langle \mathcal{S} \,|\, u = v\rangle)$ is a subgraph of $D(\mathcal{S})_{uv}$. Notice that these graphs have a common set of vertices $I \cup O - \{u, v\}$. Suppose (w, z) is an edge of $D(\langle \mathcal{S} \,|\, u = v\rangle)$, i.e., (w, z) is not sequential in $\langle \mathcal{S} \,|\, u = v\rangle$. By Bekić Lemma, this implies that either (w, z) is not sequential in $\mathcal{S}$ or that both (w, v), (u, z) are not sequential in $\mathcal{S}$. In both cases, it follows that (w, z) is an edge in $D(\mathcal{S})_{uv}$. □

We strenghten now Theorem 1 as follows:

If $\mathcal{N} = \langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,|\, u_1 = v_1, \ldots, u_n = v_n\rangle$ and $\Delta(\mathcal{N})$ is acyclic, then $\mathcal{N}$ is a machine and $D(\mathcal{N}) \hookrightarrow \Delta(\mathcal{N})$.

We prove this strengthened theorem by induction on n. The case $n = 0$ amounts to the fact mentioned in the main text: if $\mathcal{S}_1, \ldots, \mathcal{S}_m$ are machines with disjoint wire sets, then $\mathcal{S} = \mathcal{S}_1 \sqcup \cdots \sqcup \mathcal{S}_m$ is a machine and an input-output pair (u, v) is sequential for $\mathcal{S}$ if either (1) (u, v) is sequential for some $\mathcal{S}_i$, or (2) u and v belong to different machines $\mathcal{S}_i, \mathcal{S}_j$.

For the induction step, let $\mathcal{N}' = \langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,|\, u_1 = v_1, \ldots, u_{n-1} = v_{n-1}\rangle$. By induction hypothesis, $\mathcal{N}'$ is a machine, so we can write $\mathcal{N} \sim \langle \mathcal{N}' \,|\, u_n = v_n\rangle$. Note that $\Delta(\mathcal{N}) = \Delta(\mathcal{N}')_{[u_n=v_n]}$. It follows from this and the induction hypothesis $D(\mathcal{N}') \hookrightarrow \Delta(\mathcal{N}')$ that $D(\mathcal{N}')_{[u_n=v_n]} \hookrightarrow \Delta(\mathcal{N})$. This in turn, together with acyclicity of $\Delta(\mathcal{N})$, implies that (u_n, v_n) is a sequential pair for $\mathcal{N}'$. Thus, by Lemma 2, $\mathcal{N}$ is a machine. Finally,

$$\begin{array}{rcl} D(\mathcal{N}) & = & D(\langle \mathcal{N}' \,|\, u_n = v_n\rangle) \\ & \hookrightarrow & D(\mathcal{N}')_{[u_n=v_n]} \\ & \hookrightarrow & \Delta(\mathcal{N}')_{[u_n=v_n]} \\ & = & \Delta(\mathcal{N}), \end{array}$$

finishing the proof. The first immersion is justified by Lemma 10 and the second by induction hypothesis together with Lemma 8.

E. Handshake Lemma

Persistence conditions (1) will be used through the following easily derived consequence.

Lemma 11 (Handshake): If $\mathcal{S}$ satisfies (1), then, for every $Y \in O$,

$$\mathcal{S} \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Y \wedge \mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_Y \;\Rightarrow\; \mathsf{G}\,\mathsf{F}\,\mathsf{transfer}_Y.$$

Proof. From (1) we obtain

$$\mathcal{S} \models \mathsf{G}\,(\mathsf{valid}_Y \wedge \mathsf{F}\,\neg\mathsf{stop}_Y \;\Rightarrow\; \mathsf{F}\,\mathsf{transfer}_Y),$$

which implies the lemma. □

F. Proof of Lemma 3

Assuming the contrary, let ω be a behavior of $\mathcal{S} \sqcup \mathsf{Env}_{I,O}$ and let Z be a wire such that the transfer sequence $\omega^{\mathsf{T}}.Z$ is finite and of shortest length. This assumption implies $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{min\_tct}_{I \cup O} \geq \mathsf{tct}_Z$. If $Z \in I$, then from (3) we get $\omega \models \mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_Z$. We also have $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Z$, by (6). Since elastic producers satisfy the persistence condition, Handshake Lemma implies $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{transfer}_Z$, which is a contradiction.

There remains the case $Z \in O$, where we can also assume that there are no $X \in I$ with $\omega^{\mathsf{T}}.X$ of the same length as $\omega^{\mathsf{T}}.Z$. From this additional assumption we have $\omega \models \mathsf{F}\,\mathsf{G}\,(\mathsf{min\_tct}_I > \mathsf{tct}_Z \wedge \mathsf{min\_tct}_O \geq \mathsf{tct}_Z)$, so from (2) we get $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Z$. A call to consumer liveness (4) and then to Handshake Lemma yields the contradictory conclusion $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{transfer}_Z$, as in the previous case.

G. Preliminaries for Proof of Theorem 2

Define an *elastic k-producer* as a system $\mathcal{P}$ that satisfies the producer persistence condition (5) and the following weakened form of the producer liveness condition (6):

$$\mathcal{P} \models \mathsf{G}\,(\mathsf{tct}_Z < k \Rightarrow \mathsf{F}\,\mathsf{valid}_Z). \tag{14}$$

Thus, a k-producer promises to cooperate in creation of at least k transfers.

Similarly, we weaken the definition of the systems $\mathcal{P}_Z$ (for any wire Z) and define $\mathcal{P}_Z^k$ as the system defined by conditions (5) and (14). Then we define the *k-environment* (a system, but not a machine)

$$\mathsf{Env}_{I,O}^k = \textstyle\bigsqcup_{X \in I} \mathcal{P}_X^k \sqcup \bigsqcup_{Y \in O} \mathcal{C}_Y.$$

The following is a finite-transfer version of Lemma 3, proved in much the same way. Note that since $\mathsf{Env}_{I,O} \subseteq \mathsf{Env}_{I,O}^k$ for every k, Lemma 12 actually implies Lemma 3.

Lemma 12: Let $\mathcal{S}$ be an $[I, O]$-system satisfying the conditions (1-3). Then for every behavior $\omega \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}^k$, all the component streams of the transfer behavior ω^{T} have length at least k.

Proof. The proof of Lemma 3 applies almost verbatim. The only change is that, arguing by contraction, we can now assume that, in addition to being finite, the sequence $\omega^{\mathsf{T}}.Z$ has length $< k$. Consequently, the intermediate result $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Z$ is derived from (14), (6), and $\omega \models \mathsf{tct}_Z < k$. □

We will also need two easy lemmas about standard (non-elastic) machines.

Lemma 13: Recall that every network

$$\mathcal{N} = \langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,|\, u_1 = v_1, \ldots, u_n = v_n\rangle$$

is the system obtained by hiding the wires u_i and v_i in the corresponding system

$$\mathcal{N}^\sharp = \mathcal{S}_1 \sqcup \cdots \sqcup \mathcal{S}_m \sqcup \mathsf{Conn}(u_1, v_1) \sqcup \cdots \sqcup \mathsf{Conn}(u_n, v_n).$$

and that $W^\sharp = W \cup \{u_1, v_1, \ldots, u_n, v_n\}$ is the relationship between the wire sets of these systems. Suppose that the graph $\Delta(\mathcal{N})$ acyclic. Then the map $\sigma \mapsto \sigma \downarrow W \colon \mathcal{N}^\sharp \to \mathcal{N}$ is a bijection. In particular, if $W = \emptyset$, then $\mathcal{N}^\sharp$ has exactly one behavior.

Proof. Consider the case $n = 1$. For this case, our lemma says that when $\mathcal{S}$, u, v are as in Lemma 2, then for every behavior $\sigma \in \langle \mathcal{S} \mid u = v \rangle$ there is a unique $a \in [\![u]\!]$ such that $\sigma * [u \mapsto a] * [v \mapsto a] \in \mathcal{S}$. This is clear from considerations in Section II-E, notably Lemma 2.

The general case follows by induction. □

Lemma 14: Suppose $\mathcal{C}$ is a machine with all input-output wire pairs sequential. Suppose also that in the network $\mathcal{N} = \langle \mathcal{C}, \mathcal{C}' \mid X_1 = Y_1, \ldots, X_n = Y_n \rangle$ exactly one wire of each pair (X_i, Y_i) is a wire of $\mathcal{C}$ and the other is a wire of $\mathcal{C}'$. Then $\Delta(\mathcal{N})$ is acyclic.

Proof. By assumption, the graph $D(\mathcal{C})$ has no edges. Thus, $\Delta(\mathcal{N})$ is the union of a bipartite graph $D(\mathcal{C}')$ and a set of vertices, and so is acyclic. □

H. Proof of Theorem 2

We will prove that $\mathcal{S}^\intercal = \{\omega^\intercal \mid \omega \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}\}$ is causal and accepts every input stream.

By assumption, the transfer determinism condition holds for $\mathcal{S}$:

$$\forall \zeta, \vartheta \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}.\ \zeta^\intercal.I = \vartheta^\intercal.I \ \Rightarrow\ \zeta^\intercal.O = \vartheta^\intercal.O. \quad (15)$$

Let us define a *partial W-behavior* (for any given set W of wires) as a function θ that associates to every wire $w \in W$ a (finite or infinite) sequence $\theta.w \in \mathsf{type}(w)^\omega$. Note that every behavior is a partial behavior as well. If all sequences $\theta.w$ ($w \in W$) have length k, we will say that θ is a *partial W-behavior of length k* . We allow $k = \infty$ in this definition, so that a partial behavior of length ∞ is just a behavior.

Our main goal—the causality of $\mathcal{S}^\intercal$—will be proved when we have established the following equivalent property: For every partial I-behavior θ of finite length k, there exists a partial O-behavior χ of length k such that

$$\forall \vartheta \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}.\ \theta \preceq \vartheta^\intercal.I \ \Rightarrow\ \chi \preceq \vartheta^\intercal.O. \quad (16)$$

To prove this, we will need to put our system $\mathcal{S}$ in "eager" environments that are themselves machines, defined next. The eager environments will let us define χ as a function of θ; see (18) below. In the remainder of the proof, we will check that the so obtained χ satisfies (16).

For any wire Z and a (finite or infinite) sequence α of elements of $\mathsf{type}(Z)$, let $\mathcal{P}_Z^\alpha$ be the "eager producer" machine that offers transfer of elements of α in order, and if it succeeds in transferring them all, then it stops offering further transfer. As a system, it is characterized by the following properties:

$$\begin{aligned} \mathcal{P}_Z^\alpha &\models \mathsf{G}\,(\mathsf{tct}_Z = m \wedge m < |\alpha| \ \Rightarrow\ \mathsf{valid}_Z \wedge Z = \alpha[m]) \\ \mathcal{P}_Z^\alpha &\models \mathsf{G}\,(\mathsf{tct}_Z \geq |\alpha| \ \Rightarrow\ \neg\mathsf{valid}_Z \wedge Z = \mathsf{Arb}_Z) \end{aligned}$$

where $|\alpha|$ denotes the length of α and Arb_Z is an arbitrary element of $\mathsf{type}(Z)$. Note that when α is infinite the second condition is vacuously true and the conjunct $m < |\alpha|$ can be removed from the first condition.

Lemma 15: For every behavior $\sigma \in \mathcal{P}_Z^\alpha$, the transfer sequence $\sigma^\intercal.Z$ is a prefix of α.

Define also the "eager consumer" machine $\mathcal{C}_Z^\circ$ (for any wire Z), characterized by the property

$$\mathcal{C}_Z^\circ \models \mathsf{G}\,\neg\mathsf{stop}_Z.$$

For any partial I-behavior θ, define the system

$$\mathcal{S}^\theta = \mathcal{S} \sqcup \mathcal{P}^\theta \sqcup \textstyle\bigsqcup_{Y \in O} \mathcal{C}_Y^\circ, \quad \text{where} \quad \mathcal{P}^\theta = \textstyle\bigsqcup_{X \in I} \mathcal{P}_X^{\theta.X}. \quad (17)$$

Lemma 16: Let θ be a partial I-behavior of length k. The system $\mathcal{S}^\theta$ has exactly one behavior ω and it satisfies $\omega^\intercal.I = \theta$. Moreover, $|\omega^\intercal.Y| \geq k$ for every $Y \in O$.

Proof. It is easy to check that for finite sequences α one has $\mathcal{P}_Z^\alpha \subseteq \mathcal{P}_Z^{|\alpha|}$, while for infinite α one has $\mathcal{P}_Z^\alpha \subseteq \mathcal{P}_Z$. Consequently, if θ is a partial I-behavior of length k, we have that $\mathcal{S}^\theta$ is a subsystem of $\mathcal{S} \sqcup \mathsf{Env}_{I,O}^k$, where we assume that $\mathcal{S} \sqcup \mathsf{Env}_{I,O}^\infty$ is just $\mathcal{S} \sqcup \mathsf{Env}_{I,O}$. By Lemma 12, $|\omega^\intercal.Z| \geq k$ holds for all behaviors ω of $\mathcal{S}^\theta$ and all channels Z.

The eager producer and eager consumer machines have the property that their outputs depend sequentially on all their inputs. Note that $\mathcal{S}^\theta$ is of the form $\mathcal{N}^\sharp$ (see Lemma 13) for a network $\mathcal{N}$ whose one component is $\mathcal{S}$ and all other components are eager producers or consumers. Repeated application of Lemma 14 proves $\Delta(\mathcal{S}^\theta)$ is acyclic. Then Lemma 13 implies that $\mathcal{S}^\theta$ has a unique behavior ω. Since ω is also a behavior of $\mathcal{P}^\theta$, we obtain from Lemma 15 that for every input channel X the sequence $\omega^\intercal.X$ is a prefix of $\theta.X$. Since $|\omega^\intercal.X| \geq k$, this implies $\omega^\intercal.X = \theta.X$ for every X, which is to say that $\omega^\intercal.I = \theta$. □

Lemma 16 implies immediately that for every $\theta \in [\![I]\!]$ there exists $\omega \in \mathcal{S}$ such that $\omega^\intercal.I = \theta$. Thus, $\mathcal{S}^\intercal$ accepts all inputs and it remains to prove that it satisfies the causality property (16). For this proof, we fix a partial I-behavior θ of finite length k and let ω be as provided by Lemma 16. The lemma also implies the existence of a number n such that $\omega[n] \models \mathsf{tct}_Z \geq k$ for all channels Z. Thus, there exist a partial O-behavior χ of length k such that $\chi \preceq \omega^\intercal.O$ and

$$\forall \zeta.\ \zeta \sim_n \omega \ \Rightarrow\ \chi \preceq \zeta^\intercal.O \quad (18)$$

Suppose now $\vartheta \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}$ satisfies $\theta \preceq \vartheta^\intercal.I$. (This is the assumption from the causality property (16).) To finish the proof, we need to verify that $\chi \preceq \vartheta^\intercal.O$. We will do it by exhibiting a behavior $\zeta \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}$ such that

$$\zeta^\intercal.I = \vartheta^\intercal.I \ \text{ and } \ \zeta \sim_n \omega. \quad (19)$$

In view of (15) and (18), these two properties indeed imply $\chi \preceq \vartheta^\intercal.O$.

We proceed to define a behavior ζ satisfying (19). (The definition will take more effort than the proof that it satisfies (19).)

Consider the variation $\mathcal{P}_Z^{\alpha,k,n}$ of the producer machine $\mathcal{P}_Z^{\alpha}$, given by

$$\mathcal{P}_Z^{\alpha,k,n} \models \mathsf{G}\,(\mathsf{ct} \le n \wedge \mathsf{tct}_Z = m \wedge m < k \;\Rightarrow\; \mathsf{valid}_Z \wedge Z = \alpha[m])$$
$$\mathcal{P}_Z^{\alpha,k,n} \models \mathsf{G}\,(\mathsf{ct} \le n \wedge \mathsf{tct}_Z \ge k \;\Rightarrow\; \neg\mathsf{valid}_Z \wedge Z = \mathsf{Arb}_Z)$$
$$\mathcal{P}_Z^{\alpha,k,n} \models \mathsf{G}\,(\mathsf{ct} > n \wedge \mathsf{tct}_Z = m \;\Rightarrow\; \mathsf{valid}_Z \wedge Z = \alpha[m])$$

where the variable ct denotes the position in the stream, i.e., for every $\sigma \in \mathcal{S}$, we set $\sigma[n] \models \mathsf{ct} = n$. The numbers k, n are arbitrary in this definition, although we will use the producers just defined with the values for k and n as defined in previous paragraphs. The sequence α is arbitrary but infinite. Intuitively, $\mathcal{P}_Z^{\alpha,k,n}$ eagerly offers the first k elements of α for transfer while $\mathsf{ct} \le n$. If all k get transfered, then the machine waits until $\mathsf{ct} > n$ and then eagerly offers the rest of α for transfer.

It is easy to see that $\mathcal{P}_Z^{\alpha,k,n}$ is a subsystem of $\mathcal{P}_Z$ and that it is an elastic machine with its two outputs, $\mathsf{data}_Z, \mathsf{valid}_Z$ depending sequentially on its input, stop_Z. Moreover, for every behavior σ of this machine, one has $\sigma^{\mathsf{T}}.Z \preceq \alpha$. In particular, $\sigma^{\mathsf{T}}.Z = \alpha$ for every behavior σ such that σ^{T} is infinite.

Define the system

$$\mathcal{S}^{\vartheta} = \mathcal{S} \sqcup \mathcal{P}^{\vartheta} \sqcup \bigsqcup_{Y \in O} \mathcal{C}_Y^{\circ}, \quad \text{where} \quad \mathcal{P}^{\vartheta} = \bigsqcup_{X \in I} \mathcal{P}_X^{\vartheta^{\mathsf{T}}.X,k,n}.$$

Arguing as in the proof of Lemma 16, we obtain that $\mathcal{S}^{\vartheta}$ has exactly one behavior ζ and that it satisfies $\zeta^{\mathsf{T}}.I = \vartheta^{\mathsf{T}}.I$.

Finally, to deduce $\zeta \sim_n \omega$, note that $\mathcal{P}^{\vartheta}$ and $\mathcal{P}^{\theta}$ are machines with identical behaviors "up to $\mathsf{ct} = n$"; that is, for every behavior of $\mathcal{P}^{\vartheta}$ there is a $\sim_n$-equivalent behavior of $\mathcal{P}^{\theta}$ and vice versa. Therefore, the systems $\mathcal{S}^{\vartheta}$ and $\mathcal{S}^{\theta}$ must be in the same relationship. Since ζ, ω are the only behavors of $\mathcal{S}^{\vartheta}$, $\mathcal{S}^{\theta}$ respectively, it follows that $\zeta \sim_n \omega$.

I. Example: Strong Liveness Not Preserved by Feedback

The forward and backward *strong liveness conditions* for an $[I, O]$-system are given by

$$\mathcal{S} \quad \models \quad \mathsf{G}\,(\mathsf{min\_tct}_I > \mathsf{tct}_Y \Rightarrow \mathsf{F}\,\mathsf{valid}_Y) \qquad (20)$$
$$\mathcal{S} \quad \models \quad \mathsf{G}\,(\mathsf{min\_tct}_O \ge \mathsf{tct}_X \Rightarrow \mathsf{F}\,\neg\mathsf{stop}_X) \qquad (21)$$

for every $X \in I$ and for every $Y \in O$.

In practice, elastic components satisfy these strong liveness conditions. However, a composition of such components, while satisfying weaker liveness conditions (2) and (3), may not satisfy the stronger versions (20) and (21). The fundamental difference between the liveness and the strong liveness is as follows: while the strong liveness promises progress on *all under-served* output (20) or input (21) channels, the liveness only guarantees progress on the *least served*, so to speak the "hungriest", channels.

Figure 7 shows an example demonstrating that feedback does not preserve the strong forward liveness. Figure 7(a) depicts an elastic machine $\mathcal{S}$ with a sequential channel pair (b, e) during a cycle when channel a has seen 2 transfers, channel d - 0 transfers, and all other channels - 1 transfer each. For machine $\mathcal{S}$: $\mathsf{min\_tct}_I = \mathsf{min\_tct}_{\{a,b\}} = 1 > \mathsf{tct}_d$,

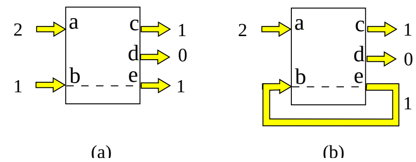

Fig. 7. Example demonstrating that feedback does not preserve the strong forward liveness.

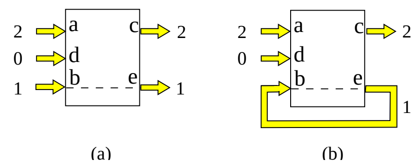

Fig. 8. Example demonstrating that feedback does not preserve the strong backward liveness.

hence (20) implies future progress for d: $\mathsf{F}\,\mathsf{valid}_d$. No progress on c can be guaranteed, since $\mathsf{min\_tct}_{\{a,b\}} = \mathsf{tct}_c$.

To prove the strong forward liveness for $\mathcal{F}(\mathcal{S}) = \langle\!\langle \mathcal{S} \,\|\, b = e \rangle\!\rangle$ shown in Figure 7(b) one needs to prove progress for output channels d and c, since for $\mathcal{F}(\mathcal{S})$ channel b is not an input anymore and $\mathsf{min\_tct}_{I'} = \mathsf{min\_tct}_{\{a\}} = 2 > \mathsf{tct}_c$. Progress on d follows immediately from progress on d in machine $\mathcal{S}$. Progress on c does not. However, if we can show that the second transfer would eventually occur on the feedback channel (b, e), it would imply progress on c. By the definition of transfer and the Handshake Lemma 11 it is necessary and sufficient to demonstrate that $\mathsf{F}\,\mathsf{valid}_e$ and $\mathsf{F}\,\neg\mathsf{stop}_b$ hold. The former holds due to the sequentiality of the input-output channel pair (b, e). The latter does not hold since the strong backward liveness condition for $\mathcal{S}$ cannot be applied until the first transfer on the output channel d occurs. Hence (20) does not hold for $\mathcal{F}(\mathcal{S})$. Note that (2) is satisfied since it only requires progress on the output channel d.

Figure 8 shows an example demonstrating that feedback does not preserve the strong backward liveness either. The counter-example for the strong backward liveness is (almost) symmetric to the forward case. Figure 8(a) depicts an elastic machine $\mathcal{S}$ with a sequential channel pair (b, e) during a cycle when channel a has seen 2 transfers, channel d (an input channel in this example) - 0 transfers, and all other channels - 1 transfer each. For machine $\mathcal{S}$:

$$\mathsf{min\_tct}_O = \mathsf{min\_tct}_{\{c,e\}} = 1 \ge \mathsf{tct}_d = 0$$

hence (21) implies future progress for input d: $\mathsf{F}\,\neg\mathsf{stop}_d$. No progress on a can be guaranteed, since $\mathsf{min\_tct}_{\{c,e\}} < \mathsf{tct}_a$.

To prove the strong backward liveness for $\mathcal{F}(\mathcal{S}) = \langle\!\langle \mathcal{S} \,\|\, b = e \rangle\!\rangle$ shown in Figure 8(b) one needs to prove progress for input channels d and a, since for $\mathcal{F}(\mathcal{S})$ channel e is not an output anymore and

$$\mathsf{min\_tct}_{O'} = \mathsf{min\_tct}_{\{c\}} = 2 \ge \mathsf{tct}_a.$$

Progress on d follows immediately from progress on d in machine $\mathcal{S}$. If we can show that the second transfer would eventually occur on the feedback channel (b, e), it would imply progress on a. By the definition of transfer and the Handshake Lemma 11 it is necessary and sufficient to show that $\mathsf{F}\,\mathsf{valid}_e$ and $\mathsf{F}\,\neg\mathsf{stop}_b$. The latter holds due to the strong backward liveness (21) of the original machine $\mathcal{S}$. The former, however, does not hold since the sequentiality of channel (b, e) cannot be applied until the first transfer on the input channel d occurs. Hence (21) does not hold for $\mathcal{F}(\mathcal{S})$, while (3) holds since it only requires progress on the input channel d.

J. *Proof of Theorem 3* (a)

This proof will use results and arguments from the proof of Theorem 2 above. Let us begin with the appropriate strengthening of Lemma 12.

Lemma 17: Let

$$\mathsf{Env}^k_{I,O,P} = \mathcal{P}^{k-1}_P \sqcup \bigsqcup_{X\in I-P} \mathcal{P}^k_X \sqcup \bigsqcup_{Y\in O} \mathcal{C}_Y$$

and let $\mathcal{S}$ be an $[I, O]$-system satisfying the conditions (1-3) and (7). Then for every behavior $\omega \in \mathcal{S} \sqcup \mathsf{Env}^k_{I,O,P}$ one has $|\omega^\intercal.Q| \geq k$.

Proof. Let $\omega \in \mathcal{S}\sqcup\mathsf{Env}^k_{I,O,P}$. Clearly, $\mathsf{Env}^k_{I,O,P} \subseteq \mathsf{Env}^{k-1}_{I,O}$, so by Lemma 12, $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{tct}_Z \geq k-1$ holds for all $Z \in I \cup O$. Assume $|\omega^\intercal.Q| < k$, the contrary of what we are to prove. Thus,

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\mathsf{min\_tct}_{I\cup O} \geq k-1 \wedge \mathsf{tct}_Q = k-1). \qquad (22)$$

We claim now that $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{tct}_X \geq k$ for every $X \in I-P$. To prove this claim, assume it is not true. Then we have $\omega \models \mathsf{F}\,\mathsf{G}\,(\mathsf{min\_tct}_O \geq k-1 \wedge \mathsf{tct}_X = k-1)$ for some $X \in I-P$. Thus, $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{min\_tct}_{I\cup O} \geq \mathsf{tct}_X$ and since $\omega \in \mathcal{S}$, the liveness of $\mathcal{S}$ implies $\omega \models \mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_X$. On the other hand, since $\omega.\{X, \mathsf{valid}_X, \mathsf{stop}_X\} \in \mathcal{P}^k_X$, from $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{tct}_X = k-1$ and (14) we can derive $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_X$. Since $\mathcal{P}^k_X$ satisfies the persistence condition (1), Lemma 11 implies $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{transfer}_X$, which contradicts our assumption $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{tct}_X = k-1$. This finishes the proof of the claim.

From (22), the claim above, and the elastic sequentiality condition (7), we obtain $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Q$. On the other hand, we have $\omega \models \mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_Q$ since $\mathcal{C}_Q$ is a component of our system. Invoking Lemma 11 again, we obtain $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{transfer}_Q$, which contradicts our starting assumption $|\omega^\intercal.Q| < k$ and so finishes the proof. □

Suppose now θ is a partial I-behavior with $|\theta.P| = k-1$ and $|\theta.X| = k$ for all $X \in I-P$. Define the system $\mathcal{S}^\theta$ in the eager environment by the same equation (17) used in the proof of Theorem 2. In the same way that we derived Lemma 16 from Lemma 12, Lemma 14, and Lemma 13, we can now use Lemma 17 in place of Lemma 12 and prove the following: The system $\mathcal{S}^\theta$ has a unique behavior ω, and that behavior satisfies the properties $\omega^\intercal.I = \theta$ and $|\omega^\intercal.Q| \geq k$.

Let χ be the prefix of length k of $\omega^\intercal.Q$. For the proof that the pair (P, Q) is sequential for $\mathcal{S}^\intercal$, it suffices to prove

$$\forall\vartheta \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}.\ \ \theta \preceq \vartheta^\intercal.I \ \Rightarrow\ \chi \preceq \vartheta^\intercal.Q.$$

This can indeed be derived in the same manner we derived the similar condition (16) in the proof of Theorem 2. For the number n we choose now any integer such that $\omega[n] \models \mathsf{tct}_Q = k$, and for any ϑ such that $\theta \preceq \vartheta^\intercal.I$, we define

$$\mathcal{S}^\vartheta = \mathcal{S} \sqcup \mathcal{P}^\vartheta \sqcup \bigsqcup_{Y\in O} \mathcal{C}^\circ_Y,$$

where

$$\mathcal{P}^\vartheta = \mathcal{P}^{\vartheta^\intercal.P,k-1,n}_P \sqcup \bigsqcup_{X\in I-P} \mathcal{P}^{\vartheta^\intercal.X,k,n}_X.$$

The remaining details of the derivation of $\chi \preceq \vartheta^\intercal.Q$ are now pure repetition of what is done in the proof of Theorem 2 and are therefore omitted.

K. *Proof of Theorem 3* (b)

We will prove here that $\langle\langle \mathcal{S} \,[\![\, P = Q \rangle\rangle$ satisfies the persistence and liveness conditions. The transfer determinism (even functionality) property follows from part (c), proved in Section L below.

Let us start with notation:

$$\mathcal{S}' = \langle\langle \mathcal{S} \,[\![\, P = Q \rangle\rangle \qquad I' = I - \{P\} \qquad O' = O - \{Q\}$$

$$\mu_Z = \mathsf{min\_tct}_{I\cup O} \geq \mathsf{tct}_Z \qquad \mu'_Z = \mathsf{min\_tct}_{I'\cup O'} \geq \mathsf{tct}_Z$$

Recall that for every behavior ω' of $\mathcal{S}'$ there exists a unique behavior ω of $\mathcal{S}$ such that

$$\begin{aligned} &\omega.(I' \cup O') = \omega' \quad \omega.P = \omega.Q \\ &\omega.\mathsf{valid}_P = \omega.\mathsf{valid}_Q \quad \omega.\mathsf{stop}_P = \omega.\mathsf{stop}_Q \end{aligned} \qquad (23)$$

Just from the observation that for each $Z \in I' \cup O'$ the stream triples $(\omega'.Z, \omega'.\mathsf{valid}_Z, \omega'.\mathsf{stop}_Z)$ and $(\omega.Z, \omega.\mathsf{valid}_Z, \omega.\mathsf{stop}_Z)$ are equal, it follows that $\mathcal{S}'$ satisfies the persistence conditions (1).

Turning to the liveness conditions, we first prove an auxiliary result.

Lemma 18: Let $\phi_n = \mathsf{min\_tct}_{I'} > n \wedge \mathsf{min\_tct}_{O'} \geq n$. Then for every $n \geq 0$ and every behavior $\omega \in \mathcal{S}$ as in (23), one has $\omega \models \mathsf{G}\,(\phi_n \Rightarrow \mathsf{F}\,\mathsf{tct}_P > n)$.

Proof. The proof is by induction on n. We can assume

$$\omega \models \mathsf{G}\,(\phi_n \Rightarrow \mathsf{F}\,\mathsf{tct}_P \geq n). \qquad (24)$$

Indeed, this is trivially true for $n = 0$, and for $n > 0$ it follows from the induction hypothesis $\omega \models \mathsf{G}\,(\phi_{n-1} \Rightarrow \mathsf{F}\,\mathsf{tct}_P > n-1)$ and the obvious fact $\omega \models \phi_n \Rightarrow \phi_{n-1}$.

Arguing by contradiction, we can assume

$$\omega \models \mathsf{F}\,(\phi_n \wedge \mathsf{G}\,\mathsf{tct}_P \leq n),$$

which, in view of (24), can be sharpened into

$$\omega \models \mathsf{F}\,(\phi_n \wedge \mathsf{G}\,\mathsf{tct}_P = n).$$

Since $\omega \models \mathsf{G}\,(\phi_n \Rightarrow \mathsf{G}\,\phi_n)$ (by monotonicity of tct_Z), this implies

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\phi_n \wedge \mathsf{tct}_P = n). \qquad (25)$$

Clearly,

$$\omega \models \mathsf{G}\,(\phi_n \wedge \mathsf{tct}_P = n \Rightarrow \mu_P),$$

which with (25) gives $\omega \models \mathsf{F}\,\mathsf{G}\,\mu_P$, which in turn, combined with (3), gives

$$\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_P. \qquad (26)$$

Since $\omega \models \mathsf{G}\,(\mathsf{tct}_P = \mathsf{tct}_Q)$, we have

$$\omega \models \mathsf{G}\,(\phi_n \wedge \mathsf{tct}_P = n \Rightarrow \mu_Q \wedge \mathsf{min\_tct}_{I'} > \mathsf{tct}_Q),$$

which with (25) gives $\omega \models \mathsf{F}\,\mathsf{G}\,(\mu_Q \wedge \mathsf{min\_tct}_{I'} > \mathsf{tct}_Q)$, which in turn, combined with (7) gives

$$\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{F}\,\mathsf{valid}_Q. \tag{27}$$

From (26) and (27), we derive $\omega \models (\mathsf{G}\,\mathsf{F}\,\neg\mathsf{stop}_P) \wedge (\mathsf{G}\,\mathsf{F}\,\mathsf{valid}_P)$. Then by Lemma 11, we have $\omega \models \mathsf{G}\,\mathsf{F}\,\mathsf{transfer}_P$, which implies $\omega \models \mathsf{F}\,\mathsf{tct}_P > n$, contradicting (25). This finishes the proof of Lemma 18. □

The liveness conditions for $\mathcal{S}'$ amount to the following, to be proved for every ω as in (23).

$$\omega \models \mathsf{G}\,(\mu'_Y \wedge \mathsf{min\_tct}_{I'} > \mathsf{tct}_Y \Rightarrow \mathsf{F}\,\mathsf{valid}_Y) \tag{28}$$
$$\omega \models \mathsf{G}\,(\mu'_X \Rightarrow \mathsf{F}\,\neg\mathsf{stop}_X) \tag{29}$$

for every $X \in I'$ and every $Y \in O'$.

We prove the simpler property (29) first. Relying on the backward liveness of $\mathcal{S}$ (3), it suffices to check

$$\omega \models \mathsf{G}\,((\mu'_X \Rightarrow \mathsf{F}\,\neg\mathsf{stop}_X) \vee \mathsf{F}\,\mu_X),$$

which is equivalent to

$$\omega \models \mathsf{G}\,(\mu'_X \wedge \mathsf{G}\,\mathsf{stop}_X \Rightarrow \mathsf{F}\,\mu_X).$$

It is easy to verify that

$$\omega \models \mathsf{G}\,(\mathsf{G}\,\mathsf{stop}_X \Rightarrow \exists n.\ \mathsf{G}\,\mathsf{tct}_X = n),$$

so it suffices to prove

$$\omega \models \mathsf{G}\,(\mu'_X \wedge \mathsf{G}\,\mathsf{tct}_X = n \Rightarrow \mathsf{F}\,\mu_X).$$

Since $\omega \models \mathsf{G}\,(\mu_X \Leftrightarrow (\mu'_X \wedge \mathsf{tct}_P \geq \mathsf{tct}_X))$, our goal can be restated as

$$\omega \models \mathsf{G}\,(\mu'_X \wedge \mathsf{G}\,\mathsf{tct}_X = n \Rightarrow \mathsf{F}\,(\mu'_X \wedge \mathsf{tct}_P \geq \mathsf{tct}_X))$$

and then, in view of $\omega \models \mathsf{G}\,(\mu'_X \wedge \mathsf{G}\,\mathsf{tct}_X = n \Rightarrow \mathsf{G}\,\mu'_X)$, reduced to

$$\omega \models \mathsf{G}\,(\mu'_X \wedge \mathsf{G}\,\mathsf{tct}_X = n \Rightarrow \mathsf{F}\,\mathsf{tct}_P \geq n).$$

Since $\omega \models \mathsf{G}\,(\mu'_X \wedge \mathsf{tct}_X = n \Rightarrow \phi_{n-1})$, Lemma 18 finishes the proof.

For the proof of (28), notice first that, in view of (2), it suffices to prove

$$\omega \models \mathsf{G}\,((\mu'_Y \wedge \psi \Rightarrow \mathsf{F}\,\mathsf{valid}_Y) \vee \mathsf{F}\,(\mu_Y \wedge \mathsf{tct}_P > \mathsf{tct}_Y \wedge \psi)),$$

where $\psi = \mathsf{min\_tct}_{I'} > \mathsf{tct}_Y$. This is equivalent to

$$\omega \models \mathsf{G}\,((\mu'_Y \wedge \psi \wedge \mathsf{G}\,\neg\mathsf{valid}_Y) \Rightarrow \mathsf{F}\,(\mu_Y \wedge \mathsf{tct}_P > \mathsf{tct}_Y \wedge \psi)).$$

Since

$$\omega \models \mathsf{G}\,(\mathsf{G}\,\neg\mathsf{valid}_Y \Rightarrow \exists n.\ \mathsf{G}\,\mathsf{tct}_Y = n),$$

it suffices to prove

$$\omega \models \mathsf{G}\,(\mu'_Y \wedge \psi \wedge \mathsf{G}\,\mathsf{tct}_Y = n \Rightarrow \mathsf{F}\,(\mu_Y \wedge \mathsf{tct}_P > \mathsf{tct}_Y \wedge \psi)).$$

Since $\omega \models \mathsf{G}\,(\mu'_Y \wedge \mathsf{tct}_P > \mathsf{tct}_Y \Rightarrow \mu_Y)$, we can restate this goal as

$$\omega \models \mathsf{G}\,(\mathsf{G}\,\mathsf{tct}_Y = n \Rightarrow \mu'_Y \wedge \psi \Rightarrow \mathsf{F}\,(\mu'_Y \wedge \psi \wedge \mathsf{tct}_P > n)).$$

Now we use $\omega \models \mathsf{G}\,(\mathsf{G}\,\mathsf{tct}_Y = n \Rightarrow \mathsf{G}\,(\mu'_Y \wedge \psi \Leftrightarrow \phi_n))$ to further rewrite our goal as

$$\omega \models \mathsf{G}\,(\mathsf{G}\,\mathsf{tct}_Y = n \Rightarrow \phi_n \Rightarrow \mathsf{F}\,(\phi_n \wedge \mathsf{tct}_P > n)).$$

This finally follows from the already seen $\omega \models \mathsf{G}\,(\phi_n \Rightarrow \mathsf{G}\,\phi_n)$ (true by monotonicity of tct) and Lemma 18.

L. *Proof of Theorem 3* (c)

We continue with the notation $I', O', \mathcal{S}'$ from the previous section.

We need to prove that for every $\omega' \in \mathcal{S}' \sqcup \mathsf{Env}_{I',O'}$ there exists $\omega \in \mathcal{S} \sqcup \mathsf{Env}_{I,O}$ such that $\omega^{\intercal}.(I' \cup O') = \omega'^{\intercal}$ and $\omega^{\intercal}.P = \omega^{\intercal}.Q$. Indeed, we have already seen that, given $\omega' \in \mathcal{S}'$, there exists $\omega \in \mathcal{S}$ such that

$$\omega.(I' \cup O') = \omega' \quad \omega.P = \omega.Q$$
$$\omega.\mathsf{valid}_P = \omega.\mathsf{valid}_Q \quad \omega.\mathsf{stop}_P = \omega.\mathsf{stop}_Q$$

(see (23) above), so it only remains to prove that $\omega \in \mathsf{Env}_{I,O}$ holds for this ω.

By Part *(b)* of the theorem (proved in the previous section) and Lemma 3 we know that $\omega^{\intercal}.Z$ is infinite for every $Z \in I' \cup O'$. It follows then from Lemma 18 that the sequences $\omega^{\intercal}.P$ and $\omega^{\intercal}.Q$ are (equal and) infinite. This implies that $\omega.\{P, \mathsf{valid}_P, \mathsf{stop}_P\} \in \mathcal{P}_P$ and $\omega.\{Q, \mathsf{valid}_Q, \mathsf{stop}_Q\} \in \mathcal{C}_Q$. (The persistence condition in the definition of $\mathcal{P}_P$ is equivalent to the persistence condition for the wire Q for ω and is satisfied because $\omega \in \mathcal{S}$ and Q is an output of $\mathcal{S}$.) Since $\mathsf{Env}_{I',O'} = \mathsf{Env}_{I,O} \sqcup \mathcal{P}_P \sqcup \mathcal{C}_Q$ and $\omega' \in \mathsf{Env}_{I',O'}$, we finally obtain the desired $\omega \in \mathsf{Env}_{I,O}$.

M. *Sequentiality Interface for Elastic Feedback*

Definition 16: Given a function $\delta\colon O \to 2^I$ and $M \in I, N \in O$, the function $\delta_{[M=N]}\colon O - \{N\} \to 2^{I-\{M\}}$ is defined by

$$\delta_{[M=N]}(Y) = \begin{cases} \delta(Y) - \{M\} & \text{if} \quad M \in \delta(Y) \\ \delta(Y) \cap \delta(N) & \text{otherwise} \end{cases}$$

The following formula is another characterization of $\delta_{[M=N]}$.

Lemma 19: For every $X \in I - \{M\}$ and $Y \in O - \{N\}$,

$$X \notin \delta_{[M=N]}(Y) \quad \text{iff} \quad X \notin \delta(Y) \vee (X \notin \delta(N) \wedge M \notin \delta(Y)).$$

Proof. Easy. □

Lemma 20: Suppose $\delta\colon O \to 2^I$ and $P, M \in I$ and $Q, N \in O$, where $P \neq M$ and $Q \neq N$. Then $(\delta_{[M=N]})_{[P=Q]} = (\delta_{[P=Q]})_{[M=N]}$.

Proof. Directly check that for both $\delta' = (\delta_{[M=N]})_{[P=Q]}$ and $\delta' = (\delta_{[P=Q]})_{[M=N]}$ one has $X \notin \delta'(Y)$ if and only if

$$\begin{aligned} & X \rightsquigarrow Y \vee (X \rightsquigarrow N \wedge M \rightsquigarrow Y) \vee (X \rightsquigarrow Q \wedge P \rightsquigarrow Y) \\ \vee \quad & (X \rightsquigarrow N \wedge M \rightsquigarrow Q \wedge P \rightsquigarrow Y) \\ \vee \quad & (X \rightsquigarrow Q \wedge P \rightsquigarrow N \wedge M \rightsquigarrow Y), \end{aligned}$$

where, to avoid clutter, we used the notation $X \rightsquigarrow Y$ for $X \notin \delta(Y)$. □

Lemma 21: If δ is a sequentiality interface for an elastic machine $\mathcal{S}$ and if $M \in \delta(N)$, then $\delta_{[M=N]}$ is a sequentiality interface for $\langle\langle \mathcal{S} \,\|\, M = N \rangle\rangle$.

Proof. Let us start again with notation:

$$\mathcal{S}' = \langle\langle \mathcal{S} \,\|\, M = N \rangle\rangle \qquad \delta' = \delta_{[M=N]}$$

$$I' = I - \{M\} \qquad O' = O - \{N\} \qquad \mu'_Z = \mathsf{min\_tct}_{I' \cup O'} \geq \mathsf{tct}_Z$$

For every $Q \in O'$ we need to prove

$$\omega \models \mathsf{G}\,(\mu'_Q \wedge \mathsf{min\_tct}_{I' - \delta'(Q)} > \mathsf{tct}_Q \Rightarrow \mathsf{F}\,\mathsf{valid}_Q),$$

where ω is a behavior of $\mathcal{S}$ that corresponds to a behavior of $\mathcal{S}'$, i.e.

$$\omega.M = \omega.N \quad \omega.\mathsf{valid}_M = \omega.\mathsf{valid}_N \quad \omega.\mathsf{stop}_M = \omega.\mathsf{stop}_N$$

Consider first the case when $M \in \delta(Q)$. By definition of δ', we have $I' - \delta'(Q) = I - \delta(Q)$ and accordingly restate our goal:

$$\omega \models \mathsf{G}\,(\mu'_Q \wedge \mathsf{min\_tct}_{I - \delta(Q)} > \mathsf{tct}_Q \Rightarrow \mathsf{F}\,\mathsf{valid}_Q).$$

With the shorthand notation $\psi = \mathsf{min\_tct}_{I - \delta(Q)} > \mathsf{tct}_Q$, let us assume the contrary:

$$\omega \models \mathsf{F}\,(\mu'_Q \wedge \psi \wedge \mathsf{G}\,\neg\mathsf{valid}_Q). \tag{30}$$

Since $\omega \models \mathsf{G}\,(\mathsf{G}\,\neg\mathsf{valid}_Q \Rightarrow \exists n.\ \mathsf{G}\,\mathsf{tct}_Q = n)$, we have

$$\omega \models \mathsf{F}\,(\mu'_Q \wedge \psi \wedge \mathsf{G}\,\mathsf{tct}_Q = n)$$

for some n, and therefore

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\mu'_Q \wedge \psi \wedge \mathsf{tct}_Q = n). \tag{31}$$

Note now that

$$\sigma \models \mathsf{G}\,(\mu'_Q \wedge \mathsf{tct}_Q = n \Rightarrow \phi_{n-1}),$$

where ϕ is as in Lemma 18. Lemma 18 is true in the current context, so we have

$$\omega \models \mathsf{G}\,(\mu'_Q \wedge \psi \wedge \mathsf{tct}_Q = n \Rightarrow \mathsf{F}\,\mathsf{tct}_M \geq n). \tag{32}$$

From (31) and (32), we derive

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\mu'_Q \wedge \psi \wedge \mathsf{tct}_Q = n \wedge \mathsf{tct}_M \geq n).$$

This implies

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\mu_Q \wedge \psi)$$

and then, in view of (8),

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\mu_Q \wedge \psi \wedge \mathsf{F}\,\mathsf{valid}_Q),$$

which directly contradicts (30), finishing the first branch of the proof.

Suppose now $M \notin \delta(Q)$. By definition of δ', we have $I' - \delta'(Q) = I' - \delta(Q) \cap \delta(N)$ and our goal is

$$\omega \models \mathsf{G}\,(\mu'_Q \wedge \mathsf{min\_tct}_{I' - \delta(Q) \cap \delta(N)} > \mathsf{tct}_Q \Rightarrow \mathsf{F}\,\mathsf{valid}_Q). \tag{33}$$

Assuming the contrary, we obtain, in a similar fashion as before,

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = n \wedge \neg\mathsf{valid}_Q).$$

for some n, where θ stands for the antecedent part of the implication in (33). Again by Lemma 18, we have

$$\omega \models \mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = n \Rightarrow \mathsf{F}\,\mathsf{tct}_M \geq n),$$

which, combined with the previous formula, implies

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = n \wedge \neg\mathsf{valid}_Q \wedge \mathsf{F}\,\mathsf{tct}_M \geq n).$$

This further implies (by monotonicity of tct_M)

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = n \wedge \neg\mathsf{valid}_Q \wedge \mathsf{tct}_M \geq n).$$

Now, it is clear that one of the following must hold:

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = n \wedge \neg\mathsf{valid}_Q \wedge \mathsf{tct}_M > n), \tag{34}$$

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = n \wedge \neg\mathsf{valid}_Q \wedge \mathsf{tct}_M = n). \tag{35}$$

We proceed to prove that both (34) and (35) lead to contradiction.

Suppose (34) is true. Notice that

$$\omega \models \mathsf{G}\,(\mu'_Q \wedge \mathsf{tct}_M > n \wedge \mathsf{tct}_Q = n \Rightarrow \mu_Q \wedge \mathsf{min\_tct}_{I-\delta(Q)} > \mathsf{tct}_Q),$$

so by combining with (8) we obtain

$$\omega \models \mathsf{G}\,(\mu'_Q \wedge \mathsf{tct}_M > n \wedge \mathsf{tct}_Q = n \Rightarrow \mathsf{F}\,\mathsf{valid}_Q),$$

which (since μ'_Q is a conjunct of θ) contradicts (34).

Suppose (35) is true. Since $\omega \models \mathsf{tct}_M = \mathsf{tct}_N$, we have

$$\omega \models \mathsf{G}\,(\mathsf{tct}_M = \mathsf{tct}_Q \wedge \mu'_Q \Rightarrow \mu_N)$$

. Since $M \in \delta(N)$, we have $I - \delta(N) \subseteq I' - \delta(Q) \cap \delta(N)$. With these observations, we derive

$$\omega \models \mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = \mathsf{tct}_M = n \Rightarrow \mu_N \wedge \mathsf{min\_tct}_{I-\delta(N)} > \mathsf{tct}_N).$$

In view of (8), this implies

$$\omega \models \mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = \mathsf{tct}_M = n \Rightarrow \mathsf{F}\,\mathsf{valid}_N). \tag{36}$$

Since $\omega \models \mathsf{tct}_M = \mathsf{tct}_N$ and since θ has μ_N as one of the conjuncts, by liveness of $\mathcal{S}$ (3) we also get

$$\omega \models \mathsf{G}\,(\theta \wedge \mathsf{tct}_Q = \mathsf{tct}_M = n \Rightarrow \mathsf{F}\,\neg\mathsf{stop}_M). \tag{37}$$

Since $\omega \models \mathsf{valid}_N = \mathsf{valid}_M$, from (36), (37), and (35) we get

$$\omega \models \mathsf{F}\,\mathsf{G}\,(\mathsf{F}\,\mathsf{valid}_N \wedge \mathsf{F}\,\neg\mathsf{stop}_M)$$

By Lemma 11 (and $\omega \models \mathsf{valid}_M = \mathsf{valid}_N$), we obtain $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{transfer}_M$. Thus, $\omega \models \mathsf{F}\,\mathsf{G}\,\mathsf{tct}_M > n$, contradicting (35). □

N. Proof of Theorem 4

In Section D, we disambiguated the use of the dependency symbol Δ and now we do the same for Δ^{e}.

Definition 17: The *dependency graph* $D^{\mathrm{e}}(\mathcal{S}, \delta)$ *of an elastic* $[I, O]$*-machine* $\mathcal{S}$ *with a sequentiality interface* δ has the vertex set $I \cup O$ and a directed edge (X, Y) for every X, Y such that $X \notin \delta(Y)$. The *dependency graph* $\Delta^{\mathrm{e}}(\mathcal{N})$ *of an elastic network* $\mathcal{N} = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,\|\, X_1 = Y_1, \ldots, X_n = Y_n \rangle\!\rangle$ is $(D^{\mathrm{e}}(\mathcal{S}_1, \delta_1) \sqcup \cdots \sqcup D^{\mathrm{e}}(\mathcal{S}_m, \delta_m))_{[X_1=Y_1, \ldots, X_n=Y_n]}$.

Following the analogy with ordinary networks, we will write $\mathcal{N} \sim \mathcal{N}'$ when two elastic network determine the same elastic system.

Lemma 22: Let $\mathcal{S} = \mathcal{S}_1 \sqcup \cdots \sqcup \mathcal{S}_m$, where each $\mathcal{S}_i$ is an $[I_i, O_i]$-elastic machine with a sequentiality interface δ_i. Let $I = \bigcup_{i=1}^m I_i$ and $O = \bigcup_{i=1}^m O_i$, and assume that these unions are disjoint. Define $\delta \colon O \to 2^I$ by $\delta(Y) = \delta_i(Y) \cup \bigcup_{j \neq i} I_j$ for $Y \in O_i$. Then $\mathcal{S}$ is an $[I, O]$-machine, δ is a sequentiality interface for it, and $D^{\mathrm{e}}(\mathcal{S}, \delta)$ is the direct sum of graphs $D^{\mathrm{e}}(\mathcal{S}_i, \delta_i)$.

Proof. Straightforward checking. □

Lemma 23: Under the assumptions of Lemma 22, $\mathcal{S}^{\mathsf{T}} = \mathcal{S}_1^{\mathsf{T}} \sqcup \cdots \sqcup \mathcal{S}_m^{\mathsf{T}}$.

Proof. Straightforward checking. □

Definition 18: Given an elastic network $\mathcal{N} = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,\|\, X_1 = Y_1, \ldots, X_n = Y_n \rangle\!\rangle$ of elastic machines with disjoint inputs and outputs, let δ be as in Lemma 22, and define $\delta_{\mathcal{N}} = \delta_{[X_1=Y_1]\cdots[X_n=Y_n]}$.

Remark. By Lemma 20, the order of the n operations that lead from δ to $\delta_{\mathcal{N}}$ is irrelevant.

Lemma 24: If δ is a sequentiality interface of $\mathcal{S}$ and $X \notin \delta(Y)$, then $D^{\mathrm{e}}(\langle \mathcal{S} \,|\, X = Y \rangle, \delta_{[X=Y]}) \hookrightarrow D^{\mathrm{e}}(\mathcal{S}, \delta)_{[u=v]}$.

Proof. (Mutatis mutandis, the proof is the same as that of Lemma 10.) The lemma follows from $D^{\mathrm{e}}(\langle \mathcal{S} \,|\, X = Y \rangle) \hookrightarrow D^{\mathrm{e}}(\mathcal{S})_{XY} \hookrightarrow D^{\mathrm{e}}(\mathcal{S})_{[X=Y]}$, where the second immersion is an instance of Lemma 9. For the first immersion, we prove that $D^{\mathrm{e}}(\langle\!\langle \mathcal{S} \,\|\, X = Y \rangle\!\rangle)$ is actually a subgraph of $D^{\mathrm{e}}(\mathcal{S})_{XY}$. Notice that these graphs have a common set of vertices $I \cup O - \{X, Y\}$. By Lemma 19,

$$W \notin \delta_{[X=Y]}(Z) \quad \text{iff} \quad W \notin \delta(Z) \vee (W \notin \delta(Y) \wedge X \notin \delta(Z))$$

The left-hand side of this relation says that (W, Z) is an edge of $D^{\mathrm{e}}(\langle \mathcal{S} \,|\, X = Y \rangle, \delta_{[X=Y]})$, while the right-hand side implies that (W, Z) is an edge of $D^{\mathrm{e}}(\mathcal{S}, \delta)_{XY}$. □

Starting the proof of Theorem 4, consider the following claim.

> *If* $\mathcal{N} = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,\|\, X_1 = Y_1, \ldots, X_n = Y_n \rangle\!\rangle$ *and both graphs* $\Delta(\mathcal{N})$, $\Delta^{\mathrm{e}}(\mathcal{N})$ *are acyclic, then* $\mathcal{N}$ *is an elastic machine,* $\delta_{\mathcal{N}}$ *is a sequential interface for it, and* $D^{\mathrm{e}}(\mathcal{N}, \delta_{\mathcal{N}}) \hookrightarrow \Delta^{\mathrm{e}}(\mathcal{N})$.

We prove this claim by induction on n. The case $n = 0$ follows immediately from Lemma 22.

For the induction step, let $\mathcal{N}' = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,\|\, X_1 = Y_1, \ldots, X_{n-1} = Y_{n-1} \rangle\!\rangle$. Note that $\Delta^{\mathrm{e}}(\mathcal{N}) = \Delta^{\mathrm{e}}(\mathcal{N}')_{[X_n=Y_n]}$ and $\delta_{\mathcal{N}} = (\delta_{\mathcal{N}'})_{[X_n=Y_n]}$. Since $\Delta(\mathcal{N}')$ has the acyclic graph $\Delta(\mathcal{N})$ as a quotient, it must be acyclic too. Similarly, $\Delta^{\mathrm{e}}(\mathcal{N}')$ must be acyclic as well. By induction hypothesis, $\mathcal{N}'$ is an elastic machine with a sequentility interface $\delta_{\mathcal{N}'}$, so we can write $\mathcal{N} \sim \langle\!\langle \mathcal{N}' \,\|\, X_n = Y_n \rangle\!\rangle$.

We would like to use Theorem 3(b) to conclude that $\mathcal{N}$ is an elastic machine, but there are two conditions to check: (1) (X_n, Y_n) is a sequential channel pair for (the elastic system determined by) $\mathcal{N}'$, and (2) $\Delta(\langle\!\langle \mathcal{N}' \,\|\, X_n = Y_n \rangle\!\rangle)$ is acyclic.

For (1), it suffices to prove that $X_n \in \delta_{\mathcal{N}'}(Y_n)$. If this were not true, there would be a path from X_n to Y_n in $D^{\mathrm{e}}(\mathcal{N}', \delta_{\mathcal{N}'})$. By induction hypothesis, this graph is immersed in $\Delta^{\mathrm{e}}(\mathcal{N}')$ so there would be a path from X_n to Y_n in $\Delta^{\mathrm{e}}(\mathcal{N}')$, so $\Delta^{\mathrm{e}}(\mathcal{N}) = \Delta^{\mathrm{e}}(\mathcal{N}')_{[X_n=Y_n]}$ would not be acyclic, which it is. This contradiction proves (1).

For (2), we have the derivation

$$\begin{aligned}
&\Delta(\langle\!\langle \mathcal{N}' \,\|\, X_n = Y_n \rangle\!\rangle) \\
&\quad = \quad D(\mathcal{N}')_{[X_n=Y_n, \mathsf{valid}_{X_n}=\mathsf{valid}_{Y_n}, \mathsf{stop}_{X_n}=\mathsf{stop}_{Y_n}]} \\
&\quad \hookrightarrow \quad \Delta(\mathcal{N}')_{[X_n=Y_n, \mathsf{valid}_{X_n}=\mathsf{valid}_{Y_n}, \mathsf{stop}_{X_n}=\mathsf{stop}_{Y_n}]} \\
&\quad = \quad \Delta(\mathcal{N}),
\end{aligned}$$

where the immersion holds because the network $\mathcal{N}'$ is a machine (induction hypothesis) and we found in the proof of Theorem 1 that $D(\mathcal{M}) \hookrightarrow \Delta(\mathcal{M})$ holds for every machine $\mathcal{M}$.

We have now completed the proof that $\mathcal{N}$ is an elastic machine. To finish the induction step, we need to check that $\delta_{\mathcal{N}}$ is a sequentiality interface for $\mathcal{N}$ and that $D^{\mathrm{e}}(\mathcal{N}, \delta_{\mathcal{N}}) \hookrightarrow \Delta^{\mathrm{e}}(\mathcal{N})$. The first of these statements follows from Lemma 21 and the fact $X_n \in \delta_{\mathcal{N}'}(Y_n)$ (proved above). For the second, we have the following derivation:

$$\begin{aligned}
D^{\mathrm{e}}(\mathcal{N}, \delta_{\mathcal{N}}) \quad &= \quad D^{\mathrm{e}}(\langle \mathcal{N}' \,|\, X_n = Y_n \rangle, (\delta_{\mathcal{N}'})_{[X_n=Y_n]}) \\
&\hookrightarrow \quad D^{\mathrm{e}}(\mathcal{N}', \delta_{\mathcal{N}'})_{[X_n=Y_n]} \\
&\hookrightarrow \quad \Delta^{\mathrm{e}}(\mathcal{N}')_{[X_n=Y_n]} \\
&= \quad \Delta^{\mathrm{e}}(\mathcal{N}),
\end{aligned}$$

The first immersion here is justified by Lemma 24 and the second is justified by the induction hypothesis together with Lemma 8.

We have finished the inductive proof of the claim that generalizes part of Theorem 4. It remains to prove that if $\mathcal{N} = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,\|\, X_1 = Y_1, \ldots, X_n = Y_n \rangle\!\rangle$ and both graphs $\Delta(\mathcal{N})$, $\Delta^{\mathrm{e}}(\mathcal{N})$ are acyclic, then $\bar{\mathcal{N}} = \langle \mathcal{S}_1^{\mathsf{T}}, \ldots, \mathcal{S}_m^{\mathsf{T}} \,|\, X_1 = Y_1, \ldots, X_n = Y_n \rangle$ is a machine, and $\mathcal{N}^{\mathsf{T}} = \bar{\mathcal{N}}$.

Observe that Theorem 3(a) implies $D(\mathcal{S}^{\mathsf{T}}) \hookrightarrow D^{\mathrm{e}}(\mathcal{S}, \delta)$ for every elastic machine $\mathcal{S}$ and a sequentiality interface δ for it. As a consequence,

$$\begin{aligned}
&\Delta(\bar{\mathcal{N}}) \\
&\quad = \quad (D(\mathcal{S}_1^{\mathsf{T}}) \sqcup \cdots \sqcup D(\mathcal{S}_m^{\mathsf{T}}))_{[X_1=Y_1, \ldots, X_n=Y_n]} \\
&\quad \hookrightarrow \quad (D^{\mathrm{e}}(\mathcal{S}_1, \delta_1) \sqcup \cdots \sqcup D^{\mathrm{e}}(\mathcal{S}_m, \delta_m))_{[X_1=Y_1, \ldots, X_n=Y_n]} \\
&\quad = \quad \Delta^{\mathrm{e}}(\mathcal{N})
\end{aligned}$$

Since $\Delta^{\mathrm{e}}(\mathcal{N})$ is acyclic, it follows that $\Delta(\bar{\mathcal{N}})$ is acyclic too, and then by Theorem 1 that $\bar{\mathcal{N}}$ is a machine.

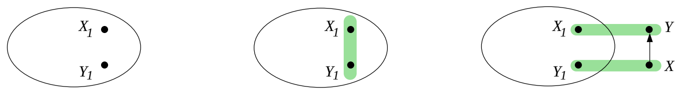

Fig. 9. From left to right: the graphs $\Delta^{\mathrm{e}}(\mathcal{N}')$, $\Delta^{\mathrm{e}}(\mathcal{N})$, and $\Delta^{\mathrm{e}}(\mathcal{M})$. Shaded areas show pairs of vertices that need to be identified. The arc from X to Y should be absent if $\delta_0(Y) = \{X\}$.

Finally, the equality $\mathcal{N}^{\mathsf{T}} = \bar{\mathcal{N}}$ is proved by induction on n. The case $n = 0$ is Lemma 23. For the induction step, we have

$$\begin{aligned} \mathcal{N}^{\mathsf{T}} &= (\langle\!\langle \mathcal{N}' \,\|\, X_n = Y_n \rangle\!\rangle)^{\mathsf{T}} \\ &= \langle (\mathcal{N}')^{\mathsf{T}} \mid X_n = Y_n \rangle \\ &= \langle \bar{\mathcal{N}}' \mid X_n = Y_n \rangle \\ &= \bar{\mathcal{N}}, \end{aligned}$$

where $\bar{\mathcal{N}}' = \langle \mathcal{S}_1^{\mathsf{T}}, \ldots, \mathcal{S}_m^{\mathsf{T}} \mid X_1 = Y_1, \ldots, X_{n-1} = Y_{n-1} \rangle$. The second equality here follows from Theorem 3(b), and the third follows by induction hypothesis.

O. Proof of Theorem 5

We begin with an easy lemma about ordinary machines.

Lemma 25: If $\mathcal{S}$ is a machine with a sequential pair (X_1, Y_1) then

$\langle \mathcal{S} \mid X_1 = Y_1 \rangle = \langle \mathcal{S} \sqcup \mathsf{Conn}(X, Y) \mid X = Y_1, X_1 = Y \rangle$.

Let $\mathcal{N}' = \langle\!\langle \mathcal{S}_1, \ldots, \mathcal{S}_m \,\|\, X_2 = Y_2, \ldots, X_n = Y_n \rangle\!\rangle$. We have

$$\begin{aligned} \Delta^{\mathrm{e}}(\mathcal{N}) &= \Delta^{\mathrm{e}}(\mathcal{N}')_{[X_1 = Y_1]} \\ \Delta^{\mathrm{e}}(\mathcal{M}) &= (\Delta^{\mathrm{e}}(\mathcal{N}') \sqcup D^{\mathrm{e}}(\mathcal{B}, \delta_0))_{[X = Y_1, X_1 = Y]} \end{aligned}$$

where δ_0 is a sequentiality interface for $\mathcal{B}$, which must be given either by $\delta_0(Y) = \{X\}$ or $\delta_0(Y) = \emptyset$. A simple analysis shows that the existence of a cycle in $\Delta^{\mathrm{e}}(\mathcal{M})$ implies the existence of a cycle in $\Delta^{\mathrm{e}}(\mathcal{N})$ (Figure 9).

Theorem 4 now implies

$$\begin{aligned} \mathcal{N}^{\mathsf{T}} &= \langle (\mathcal{N}')^{\mathsf{T}} \mid X_1 = Y_1 \rangle \\ \mathcal{M}^{\mathsf{T}} &= \langle (\mathcal{N}')^{\mathsf{T}}, \mathcal{B}^{\mathsf{T}} \mid X = Y_1, X_1 = Y \rangle \end{aligned}$$

Since $\mathcal{B}^{\mathsf{T}} = \mathsf{Conn}(X, Y)$, Lemma 25 completes the proof of Theorem 5.