

Límite. *Revista Interdisciplinaria de Filosofía y Psicología*
Volumen X, N° X, 2016, pp. x-xx

COGNICIÓN, COMPUTACIÓN Y SISTEMAS DINÁMICOS: VÍAS PARA UNA POSIBLE INTEGRACIÓN TEÓRICA

*COGNITION, COMPUTING AND DYNAMIC SYSTEMS: POSSIBLE WAYS OF
THEORETICAL INTEGRATION*

Mario Villalobos*, Joseph Dewhurst**

Universidad de Tarapacá, Arica, Chile.

School of Philosophy, Psychology and Language Sciences, University of Edinburgh, UK.

Recibido abril de 2016/Received April, 2016
Aceptado mayo de 2016/Accepted May, 2016

RESUMEN

Tradicionalmente, la teoría computacional (computacionalismo) y la teoría de sistemas dinámicos (dinamicismo) se han enfrentado como dos paradigmas opuestos e incompatibles en ciencia cognitiva. Frente a esto, se han hecho esfuerzos conciliatorios e integracionistas que buscan, principalmente, acercar el dinamicismo al computacionalismo a expensas de su postura anti-representacionista. En este trabajo, basándonos en la teoría computacional mecanicista de Piccinini y la noción de clausura funcional, exploramos una vía de integración diferente. Intentamos acercar el computacionalismo al dinamicismo a expensas de su postura representacionista, invitándolo además a reconocer el carácter funcionalmente cerrado de (ciertos) sistemas computacionales.

Palabras Clave: Teoría computacional, teoría de sistemas dinámicos, ciencia cognitiva, representación, clausura funcional.

ABSTRACT

Traditionally, computational theory (CT) and dynamical systems theory (DST) have presented themselves as opposed and incompatible paradigms in cognitive science. There have been some efforts to reconcile these paradigms, mainly, by assimilating DST to CT at the expenses of its anti-representationalist commitments. In this paper, building on Piccinini's mechanistic account of computation and the notion of functional closure, we explore an alternative conciliatory strategy. We try to assimilate CT to DST by dropping its representationalist commitments, and by inviting CT to recognize the functionally closed nature of some computational systems.

Key Words: Computational theory, dynamical systems theory, cognitive science, representation, functional closure.

1. Enmarcando el debate entre Computacionalismo y Dinamicismo

La teoría computacional de la cognición, o computacionalismo, ha jugado desde los inicios de la ciencia cognitiva un rol central en el estudio y modelamiento de los sistemas inteligentes. El

supuesto central de este programa teórico, en términos muy simples, es que los sistemas cognitivos son sistemas computacionales, o que la cognición es fundamentalmente computación. Tradicionalmente, al menos en el área que concierne a la ciencia cognitiva, los sistemas computacionales han sido

* Doctor en Filosofía, School of Philosophy, Psychology and Language Sciences, University of Edinburgh, UK. Académico de la Escuela de Psicología y Filosofía de la Facultad de Ciencias Sociales y Jurídicas de la Universidad de Tarapacá. mario.kirmayr@gmail.com

** PhD Philosophy, School of Philosophy, Psychology and Language Sciences, University of Edinburgh, UK. joseph.e.dewhurst@gmail.com

concebidos como sistemas representacionales funcionalmente abiertos. Es decir, como sistemas cuyas operaciones se realizan sobre entidades que portan alguna forma de contenido semántico, y cuya organización funcional contempla canales de “entrada” (recepción de inputs) y de “salida” (emisión de outputs) (Sprevak, 2010).

Esta caracterización general de los sistemas cognitivos es transversalmente suscrita en casi todas las filas del computacionalismo, independientemente de las arquitecturas o vehículos computacionales hipotetizados. Por ejemplo, en su versión clásica, el computacionalismo identifica los sistemas cognitivos con sistemas computacionales input-output cuyas operaciones se realizan sobre estructuras simbólicas (Fodor, 1981, 1998; Pylyshyn, 1984; Fodor & Pylyshyn, 1988), mientras que en su versión conexionista, el computacionalismo establece que los sistemas cognitivos son sistemas computacionales input-output cuyas operaciones se realizan sobre representaciones no simbólicas (distribuidas o subsimbólicas) (Smolensky, 1986; McClelland, Rummelhart & PDP Research Group, 1986; Clark, 1989). La discusión entre clásicos y conexionistas tiene que ver específicamente con el tipo de formato en el que se implementan los procesos computacionales—simbólico versus no simbólico—, no con el estatus representacional y funcionalmente abierto de los mismos.

Aunque dominante desde los inicios de la ciencia cognitiva, el computacionalismo ha sido criticado desde varios frentes teóricos. Uno de los principales ataques en este contexto ha estado representado por la teoría de sistemas dinámicos. La teoría dinamicista sostiene que los sistemas cognitivos no son sistemas computacionales sino más bien sistemas dinámicos (van Gelder, 1995, 1998; Port & van Gelder, 1995). (Revisaremos pronto qué significa, según estas teorías, decir que los sistemas cognitivos son sistemas dinámicos). Si bien el dinamicismo ofrece varias líneas de objeción frente al computacionalismo, en este trabajo nos centraremos solamente en aquellas relativas al carácter representacional y funcionalmente abierto de los sistemas computacionales.

La primera crítica dinamicista establece que los sistemas cognitivos no son sistemas computacionales porque no operan sobre representaciones. La segunda, que no son computacionales porque su organización funcional no está estructurada en términos de inputs y outputs.

En este artículo intentaremos mostrar que ambas objeciones pueden ser neutralizadas bajo el argumento de que un sistema computacional, para ser tal, no necesita ser representacional ni estar organizado funcionalmente como sistema abierto. Dicho de otra forma, trataremos de mostrar que, contrario a lo creen tanto computacionalistas como dinamicistas, un sistema computacional, y a fortiori un sistema computacional cognitivo del tipo que nos interesa (i.e., el cerebro), puede ser perfectamente y sin contradicción un sistema no representacional funcionalmente cerrado.

La primera objeción, concerniente al estatus representacional de los sistemas computacionales, ha sido desafiada por autores que, apelando a distintos argumentos, sostienen que las representaciones no son esenciales para la realización de procesos computacionales. Esta línea de pensamiento encuentra su versión más contundente y sistemática en la teoría mecanicista genérica de Piccinini (2004, 2007, 2008) (Piccinini & Scarantino, 2010, 2011). La tesis central de Piccinini, dicho en forma muy apretada, es que la computación es un mecanismo funcional que opera sobre el rango de variabilidad de una o más variables (vehículos) que pueden o no involucrar componentes representacionales (los detalles de esta teoría serán revisados en la sección 3). Esta concepción mecanicista, según Piccinini, tendría la doble virtud de ser suficientemente amplia como para reconocer diferentes tipos de computación (semántica, no semántica, digital, analógica, etc.) (Piccinini & Scarantino, 2011), y suficientemente específica como para no caer en indeseables trivializaciones ontológicas (i.e., cualquier mecanismo es ipso facto computación, o alguna otra forma de pancomputacionalismo) (Piccinini, 2007). Lo que resulta interesante en la teoría de Piccinini, al menos para los propósitos del presente trabajo, es que, como se verá en la sección 3, su noción de computación es ontológicamente neutra con respecto al carácter representacional (o no) de las entidades sobre las cuales se realiza la computación.

La segunda objeción, concerniente a la organización funcional de los sistemas computacionales, no ha suscitado, al menos hasta donde sabemos, elaboraciones filosóficas que le salgan al paso. El presente artículo pretende, entre otras cosas, contribuir a llenar ese vacío. La idea de que los sistemas computacionales están inherentemente organizados en términos de inputs y outputs será revisada críticamente a partir del concepto cibernético de clausura

funcional. La noción de clausura funcional, particularmente en los estudios cibernéticos de Maturana (1970/1980, 1975, 2003), y pensada inicialmente para la comprensión de sistemas sensoriomotores, describe un circuito cerrado (circular) de causas y efectos en el que, dada la circularidad del sistema, no hay nodos funcionales que valgan intrínsecamente como “entradas” (inputs) o “salidas” (outputs) del sistema (ahondaremos en esta noción en la sección 4 del presente artículo). Pensada originalmente para entender la dinámica sensoriomotora de los organismos, la noción de clausura funcional ha sido filosóficamente expandida y sistematizada recientemente por Villalobos (2015) (Villalobos & Ward, 2014). Para Villalobos, existe clausura funcional en todo sistema de procesos en el que al menos una parte de la cadena procedimental reentra al sistema cerrándolo funcionalmente. Bajo esta conceptualización, el rango de sistemas que quedan comprendidos dentro de la categoría de sistemas funcionalmente cerrados se amplía considerablemente, sin por ello perder, creemos, las condiciones de borde de la categoría en sí (i.e., sin riesgos de trivialización ontológica). En base a esta conceptualización, argumentaremos que los sistemas computacionales funcionalmente cerrados no solo son concebibles, sino que están de hecho instanciados en modelos computacionales que usualmente tenemos por paradigmáticos.

Mediante esta doble estrategia intentaremos mostrar que, si se hacen ciertos ajustes conceptuales, es posible, al menos en principio, reconciliar las visiones computacional y dinámica de la cognición en un enfoque integrativo.

Para tal propósito tomaremos un modelo paradigmático de sistema computacional, como es la máquina de Turing, y lo examinaremos a la luz de la teoría mecanicista de Piccinini y la noción de clausura funcional. Intentaremos mostrar que la máquina de Turing, o mejor dicho una instancia física y concreta de ella, constituye de hecho un sistema computacional no representacional funcionalmente cerrado.

Si logramos esto, habremos establecido el punto de que un sistema computacional de esas características no solo es posible, sino que existe de hecho en las instancias físicas de la máquina de Turing¹. Habremos dado también un paso pequeño pero importante con miras a la integración de las teorías computacional y dinámica de la cognición. Pero no más que eso. No habremos establecido, ni

es nuestra intención establecer aquí, nótese bien, que el cerebro y el sistema nervioso constituyen un sistema computacional no representacional funcionalmente cerrado análogo a la máquina de Turing (o a cualquier otro sistema computacional).

En este contexto, y antes de comenzar con nuestra elaboración, tal vez es conveniente declarar y explicitar el modesto alcance de este artículo. Lo que este trabajo pretende mostrar es que, contrario a lo que piensan muchos computacionalistas y dinamicistas, un sistema concreto puede ser computacional, no representacional y funcionalmente cerrado al mismo tiempo. Al mostrar esto, el trabajo pretende hacer ver que existe al menos una vía de conciliación posible entre las visiones computacional y dinámica de la cognición. Aunque creemos que esta conciliación, de llevarse a cabo, resultaría beneficiosa para la ciencia cognitiva, este artículo no va a pretender persuadir al lector de aquello. No vamos a intentar convencer al lector de que el cerebro y el sistema nervioso son realmente sistemas computacionales cerrados no representacionales, o que la ciencia cognitiva debiera concebirlos y estudiarlos de esa forma (al menos no por ahora). El presente trabajo, de carácter muy preliminar, solo pretende mostrar que los sistemas computacionales no representacionales y funcionalmente cerrados son una categoría real, y que ello nos abre una posibilidad de conciliación teórica entre los modelos computacional y dinámico. Motivar o justificar la necesidad de dicha conciliación, no obstante, requeriría una argumentación aparte; otro artículo en sí mismo.

Nuestra tesis conciliatoria tampoco reclama (absoluta) originalidad. Posturas y elaboraciones teóricas que se han planteado la compatibilidad entre los modelos computacional y dinámico existen al menos desde que fue lanzado el ataque anti-computacional por parte del dinamicismo (ver la serie de réplicas al artículo target de van Gelder de 1998; especialmente, Dietrich & Markman, 1998, Dennett, 1998, Eliasmith, 1998, Bridgeman, 1998). Se pueden mencionar, por ejemplo, las estrategias que objetan la concepción dinamicista anti-representacional de los sistemas dinámicos, señalando que los sistemas dinámicos (de control) sí son compatibles con propiedades representacionales (Bechtel, 1998; Grush, 2004). También están las estrategias minimalistas, que hacen de los mecanismos computacionales procesos muy cercanos a, o compatibles con, descripciones dinamicistas

(Wheeler, 2005; Clark, 1997). Prácticamente todas estas estrategias, o al menos las más elaboradas, están encaminadas a mostrar que los sistemas cognitivos, e.g., el cerebro y el sistema nervioso, pueden ser vistos como sistemas computacionales y dinámicos en la medida en que son reconocidos como sistemas representacionales. Es decir, incorporan aspectos del dinamicismo pero rechazan su tesis anti-representacional. Por otra parte, ninguna de estas estrategias, hasta donde sabemos, se ha planteado de manera explícita el tema de la organización funcional (abierta o cerrada) de los sistemas computacionales y dinámicos respectivamente.

Contra ese trasfondo, nuestra estrategia se distingue, primero, porque en vez de pedir al dinamicismo que abandone su tesis anti-representacional, invita al computacionalismo a abandonar su compromiso representacional. Y segundo, porque a diferencia de las estrategias precedentes, aborda en forma explícita el tema de la organización funcional.

2. Anti-Computacionalismo Dinamicista

La teoría de sistemas dinámicos establece que los sistemas cognitivos no son sistemas computacionales sino sistemas dinámicos. Un sistema dinámico, en términos generales, es un conjunto de variables cuantificables que cambian en función del tiempo y de ciertos parámetros asociados (van Gelder, 1995, 1998). En ciencia, el ejemplo clásico de sistema dinámico es el sistema solar, expresado como un conjunto de variables físicas (masas, distancias, desplazamientos, fuerzas gravitacionales, etc.) que cambian correlativamente en función del tiempo. En el campo específico de las ciencias cognitivas, y luego del seminal trabajo de van Gelder (1995, 1998), el modelo paradigmático de sistema dinámico lo constituye el gobernador de Watt (un dispositivo de control que regula y mantiene estable el flujo de vapor en un determinado sistema mecánico). De acuerdo a la interpretación de van Gelder, el gobernador de Watt constituye el tipo de modelo adecuado para estudiar y entender los sistemas cognitivos; no porque el gobernador sea en sí mismo un sistema cognitivo, sino porque los sistemas cognitivos funcionan, señala van Gelder, de manera análoga a él.

Van Gelder (1995, 1998) contrapone en forma detallada el funcionamiento de un sistema de control dinámico a un sistema de control computacional. En este artículo no vamos a reproducir el detalle de

dicho análisis, pero sí vamos a centrarnos en aquellos aspectos que son relevantes para nuestra discusión.

Un sistema de control en general, sea computacional o no, opera siempre dentro de un determinado dominio o espacio de tarea. El espacio de tarea corresponde básicamente al problema concreto que se espera que el sistema solucione (e.g., regular la temperatura en un lugar determinado). Para tomar el prototipo propuesto por van Gelder, consideremos el problema de cómo mantener constante, a pesar de variaciones de volumen o presión, el flujo de vapor que alimenta un sistema mecánico determinado.

Una opción es diseñar un sistema de control computacional. Dicho sistema, señala van Gelder (1995), estaría compuesto por dispositivos detectores de las variables de interés (volumen, presión, velocidad de flujo, apertura de válvulas de paso, etc.), dispositivos lectores de dichas mediciones, alguna unidad de procesamiento central en la que dichos valores serían ingresados en algún algoritmo computacional, y unidades ejecutoras que llevarían a cabo los ajustes prescritos por el cómputo. Tal sistema, prosigue van Gelder, operaría en forma secuencial, partiendo con un input, luego un proceso computacional algorítmico intermedio, y finalizando con un output (resultado). Dado que los valores de las magnitudes físicas medidas deben ser “traducidos” a símbolos “legibles” (interpretables) para la unidad de procesamiento, y que dichos símbolos representan sus correspondientes variables físicas, tal sistema operaría realizando computaciones sobre entidades representacionales. Así, a los ojos de van Gelder, tal dispositivo de control sería un sistema que, como todo sistema computacional, operaría sobre representaciones (simbólicas) funcionando en base a inputs y outputs.

La otra opción, que van Gelder favorece, es diseñar un dispositivo mecánico que opere como sistema dinámico acoplado al sistema que se quiere regular. Tal sistema es el gobernador de Watt. El gobernador de Watt (llamado así en honor a su inventor, James Watt) consiste en un ingenioso dispositivo que se acopla al sistema de alimentación de vapor, de tal forma que las mismas variaciones del flujo de vapor llevan mecánicamente a su propia regulación. El detalle ingenieril de tal dispositivo no es relevante para nuestra discusión. Lo que importa es que se trata de un sistema de control que opera en forma estrictamente mecánica, como simple cadena de causas y efectos físicos, sin instancias de medición, transmisión de datos, lectura de símbolos,

aplicación secuencial de pasos algorítmicos, ni nada que pueda interpretarse en términos de lo que van Gelder asume como típicamente computacional. El hecho de que el dispositivo funcione en acoplamiento físico directo con el sistema de alimentación implica, según van Gelder (1995, 1998), que se trata de un sistema continuo de causas y efectos en el que, estrictamente hablando, no existe la distinción entre un “inicio” o “entrada” del sistema (input) y un “final” o “salida” del mismo (output). Se trata de un sistema de control en el que no hay símbolos involucrados (y por tanto representaciones), ni una organización funcional abierta en términos de inputs y outputs.

Van Gelder (1995, 1998) lleva el argumento un paso más allá para defender la tesis de que los sistemas cognitivos generan comportamiento inteligente (i.e., resuelven problemas), de manera análoga al gobernador de Watt, no al sistema computacional. Nosotros, por esta vez, no vamos a seguir el argumento hasta ese punto. Esto básicamente porque, como se anunció en la sección 1 introductoria, nuestro propósito aquí es simplemente mostrar que existe una forma de compatibilidad entre las teorías dinámica y computacional, no demostrar que ellas son, ya sea por separado o en conjunto, correctas o verdaderas.

3. Piccinini y la Concepción Mecanicista de la Computación

Que todo proceso computacional es en el fondo una cierta forma de mecanismo es algo que, al menos dentro del ruedo de las teorías filosóficas, muy pocos estarían dispuestos a objetar. El mismo computacionalismo cognitivista, sin ir más lejos, suscribe esta concepción de la computación en forma implícita y a-problemática (i.e., sin ver ninguna necesidad de justificación).

El consenso parece ser que toda computación implica alguna forma particular de mecanismo. Es decir, que ser un mecanismo de cierto tipo es condición necesaria para ser una computación. Lo que Piccinini agrega, marcando así el disenso con el computacionalismo tradicional, es que ser un mecanismo de cierto tipo es condición no solo necesaria sino también suficiente para ser una computación.

La teoría computacional tradicional, tanto simbólica como conexionista, asume que los procesos computacionales son mecanismos, pero añade inmediatamente que se trata de mecanismos que manipulan entidades o vehículos con contenido

semántico. Sostienen lo que Piccinini (2004) llama una concepción semántica (representacional) de la computación. Piccinini, en cambio, señala que la posesión de contenido semántico es solo una posibilidad, y no un requisito, para los mecanismos computacionales (Piccinini, 2004, 2007, 2008). Para mostrar este argumento necesitamos una breve revisión del aparato conceptual que Piccinini monta en su construcción teórica.

Piccinini señala que, en su sentido más general, y atendiendo también al origen (y uso) técnico del término en la ciencia computacional, la computación consiste simplemente en un “proceso cuya función es manipular vehículos medio-independientes de acuerdo a una regla definida sobre ellos” (Piccinini & Scarantino, 2010, p. 239). Un vehículo medio-independiente es básicamente cualquier variable, discreta o continua, cuyos distintos valores son o pueden ser efectivamente distinguidos como tales por el sistema que los procesa (Piccinini & Scarantino, 2010). La noción de vehículo medio-independiente, que Piccinini toma de Garson (2003), se refiere a que el mecanismo computacional se ejecuta no sobre los detalles de implementación de una determinada variable, sino sobre la variable como tal y su rango de valores. Piccinini plantea esta caracterización tanto para sistemas abstractos (e.g., una máquina de Turing en tanto formalismo matemático), como para sistemas concretos (e.g., un computador de oficina). Aquí, para efectos de exposición, y también porque es el nivel que nos interesa, nos centraremos solamente en sistemas computacionales concretos físicos.

En un sistema computacional físico, los vehículos a los que se refiere Piccinini corresponden a los estados macrofísicos frente los cuales el sistema es computacionalmente sensible. Por ejemplo, si el sistema es computacionalmente sensible a cargas de voltaje eléctrico, lo es con independencia de la particular masa de electrones que genera dichos estados eléctricos. Que el vehículo es medio-independiente quiere decir simplemente que el sistema reacciona y discrimina la variable a un cierto nivel ontológico, siendo funcionalmente “ciego” a sus detalles de implementación.

Ahora bien, estos vehículos pueden corresponder a variables discretas (discontinuas) o continuas. En el primer caso hablamos de vehículos digitales, y por extensión, de computación digital. En el segundo caso hablamos de vehículos analógicos y de computación analógica. Piccinini no descarta que en ciertas formas

de computación puedan encontrarse otros tipos de vehículos, y es particularmente cauto en señalar que el cerebro, cuya complejidad neurofisiológica está todavía lejos de ser enteramente descifrada, bien podría sorprendernos con alguna forma sui generis de vehículo computacional (Piccinini & Scarantino, 2011).

Siguiendo con la definición de Piccinini, tenemos que estas variables o vehículos son manipulados de acuerdo a una cierta regla que se fija sobre ellos. Una regla computacional, señala Piccinini, es simplemente un mapeo o función que establece una cierta relación específica entre un conjunto input y un conjunto output de vehículos. (En la sección 5 mostraremos que esta caracterización funcional de inputs y outputs es en el fondo dispensable, y que el argumento central de Piccinini es igualmente válido para el caso de una organización funcional cerrada). La función o mapeo que constituye la regla computacional, advierte Piccinini, puede o no corresponder a un algoritmo o procedimiento efectivo; lo que cuenta es la especificación de la relación entre ambos conjuntos de vehículos (Piccinini & Scarantino, 2010, 2011).

Con estos conceptos en mente podemos volver al argumento de Piccinini en contra de la concepción semántica de la computación.

La idea de Piccinini, básicamente, es que para que una regla fije una cierta relación específica entre un conjunto de vehículos y otro, i.e., para que haya computación, no hay ninguna necesidad de que tales vehículos se encuentren en alguna relación semántica con algún otro dominio de estados (Piccinini, 2004, 2007, 2008; Piccinini & Scarantino, 2010). Los vehículos pueden, ciertamente, ser soporte de alguna forma de valor semántico (i.e., representar algo), pero ello no es requisito. En el caso de la computación digital, por ejemplo, lo único que el sistema requiere es que los vehículos digitales, o “dígitos” (para seguir la nomenclatura de Piccinini), tengan un rango de variabilidad y distintividad adecuado a la sensibilidad computacional del sistema. Esto es, que el sistema pueda “detectar” las distintas configuraciones de variables sin ambigüedad. Para tomar el ejemplo de Piccinini, el conjunto o serie de dígitos que forman la palabra ‘avocado’ (‘aguacate’, ‘palta’), es tan efectivo para efectos computacionales como la serie de dígitos ‘2#r%h@’ (Piccinini & Scarantino, 2010). En ambos casos, un sistema sensible a diferencias gráficas podrá aplicar igualmente bien su regla de mapeo para transformar

dichos dígitos en aquello que la regla prescriba. El hecho de que en el último caso estemos en presencia de una secuencia de caracteres sin sentido o significado (i.e., sin valor representacional), no impide al sistema la efectiva manipulación de los dígitos. Lo único que importa es que el sistema pueda en efecto discriminar entre los grafos ‘2’, ‘#’, ‘r’, ‘%’, etc., y tratarlos consistentemente. Piccinini sostiene así que un mecanismo computacional, para ser tal, no necesita operar sobre vehículos con valor representacional.

Junto con este argumento, Piccinini ofrece una extensa y exhaustiva taxonomía de sistemas computacionales (Piccinini & Scarantino, 2010, 2011) que aquí, en el acotado marco de la discusión que nos interesa, no necesitamos revisar.

4. Clausura Funcional en Sistemas Sensoriomotores

Aunque con importantes antecedentes y precursores en cibernética clásica (o cibernética de primer orden), la noción de clausura funcional, y especialmente su interpretación epistemológica, aparece explícitamente como tal en la cibernética de segundo orden de Humberto Maturana (1970/1980).

Maturana usa la noción de clausura funcional para caracterizar la organización funcional del sistema sensoriomotor en los seres vivos, y la usa específicamente para defender la tesis de que tal sistema, contrario a lo que comúnmente se asume, no tiene inputs ni outputs como propiedades intrínsecas de su organización funcional (Maturana, 1970/1980, 1975, 2003).

En esta sección, e inspirados en el reciente trabajo de Villalobos (2015), examinaremos la noción de clausura funcional y la aplicaremos a los sistemas computacionales, tomando como caso paradigmático la máquina de Turing. Argumentaremos que una instanciación física literal de la máquina de Turing constituye un caso típico de sistema sensoriomotor con clausura funcional, y por implicación, de computación sin inputs ni outputs.

Un sistema sensoefector es básicamente un sistema funcional compuesto por dos o más transductores cuyas dinámicas están conectadas de una forma u otra (Villalobos, 2015). Un sistema sensoriomotor en un subtipo de sistema sensoefector, a saber, aquél en el que el componente efector exhibe motilidad o alguna forma de movimiento observable.

Los sistemas sensoefectores pueden estar funcionalmente organizados de muchas formas.

Aquí, para nuestros propósitos, distinguiremos dos tipos fundamentales de organización funcional; la organización abierta (o lineal), y la organización cerrada (o circular). Tomemos el ejemplo de un termostato. Un termostato es un sistema funcional compuesto básicamente por dos transductores: un sensor (e.g., una placa bimetalica) y un efector (un calefactor). Si instalamos el sensor en un cuarto de la casa X y lo conectamos a un calefactor de la casa vecina (casa Z), lo que tenemos es un sistema sensoefector abierto o lineal. Decimos que el sistema es abierto o lineal porque la dinámica del sensor afecta la dinámica del efector (a través de alguna conexión o cableado), pero la dinámica de este último no afecta a la del sensor (la cadena de efectos es unidireccional). Generalmente, los termostatos se ensamblan no de esta forma sino como circuitos cerrados. El efector se instala en el mismo cuarto o casa donde está el sensor, de tal forma que su dinámica afecte, a través del aire compartido, la dinámica del sensor. En esta organización funcional cerrada o circular, sensor y efector se afectan mutuamente describiendo una cadena de efectos bidireccional.

Maturana (1970/1980) argumenta que el sistema sensoriomotor de los animales está organizado como un sistema funcional cerrado, tal como lo están los termostatos ordinarios. La dinámica de los órganos sensoriales afecta (a través del sistema nervioso) la dinámica de los órganos efectores (e.g. musculatura estriada), y la dinámica de estos últimos afecta, al cambiar la orientación espacial del cuerpo con respecto a los estímulos ambientales, la dinámica de los órganos sensores. Simplificando un poco, la idea es que el animal se orienta espacialmente hacia el entorno según lo que percibe, y que percibe tal o cual aspecto del entorno según cómo se orienta espacialmente.

Esta circularidad de la dinámica sensoriomotora ya había sido notada y comentada, bajo el concepto de feedback, en la teoría cibernética clásica (Wiener, 1948; Ashby, 1956, 1960), y también en teorías de la percepción de corte fenomenológico (Merleau-Ponty, 1963). Sin embargo, hasta antes de Maturana (1970/1980) no se había reparado en el siguiente punto epistemológico.

Si un sistema sensoefector, desde el punto de vista estrictamente funcional, es cerrado o circular, ¿dónde están su “entrada” y su “salida”? ¿Dónde están los “inputs” y los “outputs”? ¿Son éstos elementos intrínsecos de su organización, o se trata

más bien de propiedades adscritas por el observador? Maturana levanta y contesta estas preguntas en el ámbito específico del sistema nervioso de los organismos. Nosotros, siguiendo el trabajo de Villalobos (2015), vamos a realizar el análisis en un plano más general.

Retomemos el ejemplo del termostato. Tomando la temperatura ambiente como punto de referencia, nosotros los usuarios interpretamos típicamente el termostato como recibiendo entradas (inputs) a través de su sensor bimetalico, y emitiendo salidas (outputs) a través del calefactor. Lo que pasamos por alto, generalmente, es que la temperatura del ambiente es solo un link más a través del cual se cierra la circularidad funcional del termostato. El termostato, en tanto circuito funcional, no está “abierto” a la temperatura del ambiente, sino que se cierra funcionalmente sobre sí mismo a través de ella. Puesto que el circuito funcional es circular, valdría igualmente decir que el calefactor, como transductor, es la “entrada” de los impulsos eléctricos del cableado hacia el ambiente, y que el sensor bimetalico es la “salida” de la temperatura ambiente hacia el cableado. Usualmente no vemos al termostato de esta forma, porque nuestro interés está puesto en el uso para el cual ha sido diseñado como artefacto doméstico (i.e., regular la temperatura del ambiente). No obstante, desde el punto estrictamente físico y funcional, ambas descripciones son igualmente válidas (Villalobos, 2015). Esto es, el circuito también puede ser visto como manteniendo una cierta correlación de actividad en el cableado que une los transductores, valiéndose, para tal efecto, de la conexión que ellos establecen a través del aire de la pieza.

Ahora bien, que ambas descripciones sean igualmente válidas indica que la asignación de inputs y outputs se trata más bien de una convención que fija el observador, no de propiedades intrínsecas al sistema. Tal es el punto central que Villalobos, basándose en el análisis neurofisiológico de Maturana, identifica y extiende para todo sistema funcional cerrado. En un sistema funcional cerrado no existen inputs y outputs como propiedades intrínsecas del sistema, sino solo como adscripciones hechas por el observador (Villalobos, 2015).

5. Computación, Representación, Inputs-Outputs

En las secciones anteriores hemos revisado, por separado, el argumento de Piccinini en contra de

la concepción representacional de la computación y la noción de clausura funcional. En esta sección vamos a proponer una articulación de estas dos líneas argumentativas, aplicándolas sobre un sistema computacional paradigmático; la máquina de Turing.

Alan Turing propuso un sistema abstracto de cálculo lógico en el marco de sus trabajos sobre fundamentos de teoría matemática (Turing, 1937). La máquina imaginada por Turing está compuesta, básicamente, por un dispositivo (o “cabeza”) de lectura-escritura que opera sobre una cinta de datos o caracteres (inscripciones, símbolos), y un procesador autómatá móvil. La “cabeza” lee un dato o carácter por vez, y el autómatá, siguiendo un algoritmo establecido, computa a partir de su estado interno y el dato leído ciertas “acciones”. Entre las acciones posibles está el escribir o borrar símbolos en la cinta, moverse a lo largo de la cinta en ambas direcciones, y pasar de un estado interno a otro (Turing, 1937).

Nosotros, que estamos interesados no en los formalismos matemáticos probados por Turing sino en la utilidad que los modelos computacionales pueden tener para el estudio de sistemas cognitivos concretos, vamos a concentrarnos únicamente en ejemplares físicos de la máquina de Turing. Nuestras preguntas guía serán: 1) ¿Son los vehículos computacionales de la máquina de Turing inherentemente representacionales? Y 2) ¿Posee la organización funcional de la máquina de Turing inputs y outputs como componentes intrínsecos?

Lo primero que debemos notar con respecto a la máquina de Turing es que se trata de un sistema de computación digital, es decir, un sistema que opera sobre un dominio de variables discretas. Estas variables, que Piccinini llama “dígitos”, son todos los estados macrofísicos del sistema que cuentan como vehículos computacionales. Una máquina de Turing, en tanto sistema físico concreto, posee un sinnúmero de variables macrofísicas que son comunes a cualquier cuerpo físico (e.g., masa total, longitud y ancho del sistema, etc.), y que no cuentan como ítems de computación. Los dígitos, recuérdese, son aquellas variables macrofísicas del sistema que resultan relevantes para los mecanismos computacionales, o, alternativamente, aquellas que un observador identifica como participando en algún mecanismo computacional (e.g., forma física de los caracteres en la cinta de datos, umbrales de voltaje en el circuito eléctrico interno del autómatá, etc.). Lo importante, para nuestro análisis, es notar que

estas variables macrofísicas pueden asumir cualquier forma, siempre y cuando el sistema las distinga y las trate consistentemente; esto es, siempre y cuando entren en el dominio de la función o regla de mapeo del sistema.

Lo que hace a un sistema X un sistema computacional, si hemos de seguir la definición de Piccinini, es que manipula conjuntos de variables medio-independientes bajo la forma de algún mecanismo o regla establecida; nada más (y nada menos tampoco). Tal como vimos en la sección 2 de este artículo, para que una regla de mapeo opere efectivamente sobre un conjunto de vehículos, no es requisito que dichos vehículos guarden algún tipo de relación semántica con algún otro dominio de estados o variables. En el caso de la máquina de Turing, un usuario u observador puede, ciertamente, interpretar los vehículos como significando algo (prácticamente cualquier cosa; de ahí su poder y atractivo), pero dicha asignación semántica no es lo que convierte a dicho sistema en un sistema computacional.

El mismo Turing interpretó, para los propósitos específicos de su teoría matemática, los vehículos como representando números reales (Turing, 1937), pero en ningún momento planteó la tesis de que su máquina abstracta era computacional porque sus vehículos representaban números reales, o porque podían representar cualquier otro dominio de variables. Su máquina era computacional, esencialmente, porque podía generar o calcular, a través de un procedimiento efectivo o algoritmo (i.e., un conjunto finito de pasos), secuencias definidas (potencialmente infinitas) de dígitos. Por supuesto, lo que a Turing le interesaba, en el contexto de la teoría matemática de números computables, era que estos dígitos podían interpretarse como las secuencias de expansión decimal total de números reales. Sin embargo, lo que le daba al sistema su carácter computacional no era esta interpretación sino su naturaleza algorítmica, su operación como procedimiento efectivo.

Una máquina de Turing física, de acuerdo a este análisis, ostenta su estatus computacional previo a cualquier asignación interpretativa o semántica con respecto a las variables macrofísicas que constituyen los vehículos del sistema. Dicho de otro modo, y para contestar la primera pregunta de esta sección, los vehículos computacionales de la máquina de Turing no son inherentemente representacionales.

Veamos ahora la organización funcional de la máquina de Turing. En el plano de la teoría matemática original, el proceso lector, la acción de escritura, o el carácter móvil del autómatas no son aspectos relevantes en sí, pues estamos hablando de una entidad meramente pensada para propósitos demostrativos lógico-matemáticos. El asunto cambia, sin embargo, cuando consideramos una instanciación física concreta de la máquina de Turing.

El dispositivo lector es básicamente un sensor que detecta los patrones gráficos que exhibe la cinta en cada celda, y tanto la acción de escritura como la acción de moverse a lo largo de la cinta son efectuadas por algún dispositivo motor. En el plano de la realización física, la máquina de Turing encarna así un sistema sensoriomotor que detecta patrones gráficos y ejecuta ciertos movimientos.

Lo interesante es que este sistema sensoriomotor está organizado como un circuito funcional cerrado, tal como lo está el termostato de nuestro ejemplo anterior, y como lo está, según Maturana, el sistema sensoriomotor de los organismos. El sensor, a través del cableado interno del autómatas, afecta la dinámica del dispositivo motor, y el dispositivo motor, al exponer el lector a distintas celdas de la cinta, afecta al dispositivo sensor. Dicho de otro modo, lo que hace el dispositivo motor depende, en parte, de lo que lee el sensor, y lo que lee el sensor depende, en parte, de si el dispositivo motor ha desplazado el autómatas hacia la izquierda o derecha de la cinta. La dependencia es bidireccional y el circuito funcional es cerrado o circular.

Tal como en el caso del termostato, el observador puede, si quiere o le es útil, asignar inputs y outputs al sistema, tomando como base su punto de referencia o sus intereses como usuario. Típicamente, el observador o usuario está interesado en lo que pasa en la cinta de caracteres, que es donde él espera ver el resultado de las computaciones (de manera análoga a como en el caso del termostato está interesado en la regulación de la temperatura ambiente). La cinta, así, es vista como aquello que proporciona los inputs para, y recibe los outputs desde, la máquina computacional.

Sin embargo, como hemos visto anteriormente, en un sistema funcionalmente cerrado la distinción de inputs y outputs no refleja ninguna propiedad intrínseca del sistema, sino solo una convención descriptiva del observador. Sería igualmente válido, aunque probablemente poco útil para los intereses del observador, cambiar de foco y centrarse en los

mecanismos que conectan la dinámica del sensor con la dinámica del dispositivo motor, y ver la cinta simplemente como aquello que, recibiendo ciertos inputs desde el autómatas, emite ciertos outputs sobre él manteniendo así una cierta correlación de actividad entre el sensor y el dispositivo motor.

De acuerdo a este análisis, y para contestar la segunda pregunta de esta sección, una máquina de Turing física y concreta, en tanto sistema funcional cerrado, carece de inputs y outputs como componentes intrínsecos de su organización.

6. Conclusiones

Computacionalismo y dinamicismo, tradicionalmente, se han presentado a sí mismos como paradigmas cognitivos alternativos e incompatibles. El computacionalismo asume que los sistemas cognitivos son sistemas computacionales, y que los sistemas computacionales son sistemas representacionales funcionalmente abiertos. A partir de esas premisas, el computacionalismo extrae la conclusión válida (aunque no necesariamente verdadera), de que los sistemas cognitivos son sistemas representacionales funcionalmente abiertos. El dinamicismo, por su parte, señala que los sistemas cognitivos no son sistemas computacionales sino sistemas dinámicos, y que los sistemas dinámicos, a diferencia de los computacionales, no son sistemas representacionales ni están funcionalmente organizados en términos de inputs y outputs. A partir de estas premisas el dinamicismo extrae la conclusión válida (aunque no necesariamente verdadera), de que los sistemas cognitivos son sistemas no representacionales funcionalmente cerrados.

Más allá de sus diferencias, ambos enfoques teóricos suscriben gruesamente la concepción que el otro tiene de su propio modelo cognitivo. Es decir, el computacionalismo acepta sin mayores objeciones la concepción dinamicista de los sistemas dinámicos, y el dinamicismo acepta sin mayores objeciones la concepción computacionalista de los sistemas computacionales. Sobre ese acuerdo mutuo, lo que los enfoques ponen en discusión, de modo disyuntivo, es si los sistemas cognitivos son computacionales ó dinámicos.

En el presente artículo hemos intentado mostrar que el carácter disyuntivo de esta disputa es solo aparente. Hemos defendido la tesis de que un sistema computacional, para ser tal, no necesita ser representacional ni estar funcionalmente abierto en

términos de inputs y outputs. Esto es, que tanto el estatus representacional como la organización funcional abierta no son rasgos definitorios de un sistema computacional, y que si los sistemas cognitivos son concebidos como sistemas computacionales (como lo quiere el computacionalismo), perfectamente pueden ser concebidos al mismo tiempo como sistemas no representacionales funcionalmente cerrados (como lo quiere el dinamicismo).

Si esta propuesta conceptual es consistente, entonces computacionalismo y dinamicismo cuentan al menos con una alternativa para buscar vías de integración.

Nótese, eso sí, que si los programas computacional y dinamicista estimaran posible, conveniente o necesario trabajar bajo un paradigma unificado, la estrategia conceptual de acercamiento no tendría por qué ser necesariamente la que se ha propuesto

en este artículo. Son posibles, y de hecho existen, otras formas alternativas de conciliación que aquí, por razones de espacio, y también porque no tenemos aún una evaluación comparativa de ellas, no han sido revisadas.

Nótese también, para recordar el alcance de este artículo, que aquí no hemos proporcionado los argumentos necesarios para pensar que una conciliación teórica y programática entre los enfoques computacional y dinámico, de ser posible, sería deseable o fructífera para las ciencias cognitivas. Mostrar que X es posible no es mostrar que X es recomendable o deseable. Muchos computacionalistas y dinamicistas, y también muchos observadores externos, podrían estimar que ambos programas de investigación están bien así como están trabajando en veredas opuestas. Nosotros creemos que no, pero hemos dejado para otra instancia esa discusión.

Referencias

- Ashby, W. R. (1956). *An introduction to cybernetics*. London: Chapman & Hall.
- (1960). *Design for a brain*. London: Chapman & Hall.
- Bechtel, W. (1998). Representations and cognitive explanations: assessing the dynamicist's challenge in cognitive science. *Cognitive Science*, 22, 295-318.
- Bridgeman, B. (1998). The dynamical model is a Perceptron. *Behavioral and Brain Sciences*, 21, 631-632.
- Clark, A. (1989). *Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing*. Cambridge: MIT Press.
- (1997). *Being there: Putting brain, body and world together again*. Cambridge: MIT Press.
- Denet, D. C. (1998). "Revolution, no! Reform, si!". *Behavioral and Brain Sciences*, 21, 636-637.
- Dietrich, E. & Markman, A. B. (1998). All information processing entails computation. or, If R. A. Fisher had been a cognitive scientist. *Behavioral and Brain Sciences*, 21, 637-638.
- Eliasmith, C. (1998). Dynamical models and van Gelder's dynamicism: Two different things. *Behavioral and Brain Sciences*, 21, 616-665.
- Fodor, J. A. & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture. *Cognition*, 28, 3-71.
- Fodor, J. A. (1998). *Concepts*. Oxford: Clarendon Press.
- Garson, J. (2003). The introduction of information into neurobiology. *Philosophy of Science*, 70(5), 926-936.
- Grush, R. (2004). The emulation theory of representation: Motor control, imagery, and perception. *Behavioral and Brain Sciences*, 27, 377-442.
- Maturana, H. (1970/1980). Biology of cognition. En H. Maturana & F. Varela, (Eds.), *Autopoiesis and Cognition: The Realization of the Living* (pp. 5-56). Dordrecht, Holland: Kluwer Academic Publishers.
- (1975). The organization of the living: A theory of the living organization. *International Journal of Man-Machine Studies*, 7(3), 313-332.
- (2003). The Biological foundations of self-consciousness and the physical domain of existence. En N. Luhmann, H. Maturana, M. Namiki, V. Redder & F. Varela (Eds.), *Beobachter: Convergenz der erkenntnistheorien?* (pp. 47-117). München: Wilhelm Fink Verlag.
- McClelland, J., Rumerhalt, D. & the PDP research group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge: MIT Press.
- Merleau-Ponty, M. (1963). *The structure of behavior*. Boston: Beacon Press.
- Piccinini, G. (2004). Functionalism, computationalism, and mental content. *Canadian Journal of Philosophy*, 34(3), 375-410.
- (2007). Computing mechanisms. *Philosophy of Science*, 74(4), 501-526.
- (2008). Computation without representation. *Philosophical studies*, 137(2), 205-241.
- Piccinini, G & Scarantino, A. (2010). Computation vs. Information processing: why their difference matters to cognitive science. *Studies in History and Philosophy of Science*, 41, 237-246.
- Piccinini, G. & Scarantino, A. (2011). Information processing, computation and cognition. *Journal of Biological Physics*, 37(1), 1-38.
- Port, R. & Van Gelder, T. J. (1995). *Mind as Motion: Explorations in the Dynamics of Cognition*. Cambridge: MIT Press.
- Pylyshyn, Z.W. (1984). *Computing and Cognition*. Cambridge: MIT Press.
- Smolensky, P. (1986). The constituent structure of connectionist mental states: A reply to Fodor and Pylyshyn. *Southern Journal of Philosophy*, 26, 37-163.
- Sprevak, M. (2010). Computation, individuation, and the received view on representation. *Studies in History and Philosophy of Science*, 40, 260-270.
- Turing, A. M. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42(2), 230-265.
- Van Gelder, T. J. (1995). What Might Cognition Be, If Not Computation?. *Journal of Philosophy*, 92, 345-381.
- (1998). The Dynamical Hypothesis in Cognitive Science. *Behavioral and Brain Sciences*, 21, 615-65.
- Villalobos, M. (2015). *The Biological Roots of Cognition and the Social Origins of Mind* (PhD Thesis). Edinburgh: University of Edinburgh.
- Villalobos, M & Ward, D. (2014). Living systems: autopoiesis, autonomy and enaction. *Philosophy and Technology*, 28, 225-239. doi: 10.1007/s13347-014-0154-y
- Wheeler, M. (2005). *Reconstructing the Cognitive World*. Cambridge: MIT Press.
- Wiener, N. (1948). *Cybernetics: or Control and Communication in the Animal and the Machine*. Cambridge: MIT Press.

Notas

- ¹ Las implementaciones físicas de la máquina de Turing, tal como es ideada por Turing (1937) en su formulación original (salvo la extensión infinita de la cinta), son escasas, pero las hay. Véase por ejemplo el modelo de Anders Nissen, Martin Have, Mikkel Vester y Sean Geggie (Aarhus University) (<http://legoofdoom.blogspot.com>), y el de Jeroen van den Bos y Davy Landman (Centrum Wiskunde & Informatica, Amsterdam) (<http://www.legoturingmachine.org>). Véase también el (impresionantemente fiel) modelo de Mike Davey (<http://aturingmachine.com/index.php>).