GENÉTICO *versus* GENERATIVO,

O DE LA GENERACIÓN AUTOMÁTICA DE FORMAS

LEMA: **MENDEL**

GENÉTICO *versus* GENERATIVO, O DE LA GENERACIÓN AUTOMÁTICA DE FORMAS

LEMA: MENDEL

Resumen

La confusión entre los conceptos de ***genético*** y ***generativo***, dos conceptos relacionados pero distintos, ejemplifica perfectamente el estancamiento de la metodología creativa arquitectónica, y en especial muestra las dificultades para superar los esquemas del paradigma de la Modernidad.

Mediante una precisa e informada descripción del origen y significado de estos dos conceptos, y de su relación con la disciplina de la arquitectura, se pretende mostrar cuáles son los fundamentos de la arquitectura generativa, afincados en la tradición Moderna; y desde la reinterpretación de la genética, más ajustada a la realidad, se propone una posible agenda de la arquitectura para la conformación de la arquitectura al pensamiento contemporáneo.

La generación automática de formas, geometrías o edificios es una técnica vieja, que data por lo menos del renacimiento. Según Wittkower, por ejemplo, León Battista Alberti usaba una especie de *gramática* formal que le permitía combinar columnas con entablamentos, y pilares con arcos, pero nunca al revés; y Andrea Palladio, siempre según Wittkower, generó las plantas y los alzados de sus famosas *villas* con una técnica basada en su *deducción lógica* a partir de una generatriz inicial y siguiendo una serie de reglas más o menos formales[1].

Esta técnica, que denominamos ***generativa*** (la explicaremos con mayor detalle más adelante en el artículo), ha gozado históricamente de gran popularidad en la arquitectura desde el s.XV, y la han utilizado arquitectos y tratadistas de Palladio a Le Corbusier, de Durand a Alexander. Tejiendo una tupida red de influencias, y cruzando estilos y épocas, Colin Rowe (claramente influenciado él mismo por Wittkower) la identificó en Palladio y en le Corbusier[2], a lo que yo añadiría a Peter Eisenman[3], alumno en Cornell del mismísimo Rowe. De hecho, como veremos, el paradigma generativo sigue la filosofía del movimiento moderno: la lógica y la razón al servicio de la técnica y la creatividad.

La metodología generativa consiste sencillamente en la recombinación de elementos arquitectónicos y su transformación recursiva: dada una o varias formas iniciales y un conjunto de reglas de transformación preestablecidas, nuevas formas pueden generarse por la aplicación de las reglas a las formas iniciales, y pueden aplicarse estas u otras reglas a cada una de las formas intermedias sucesivamente, hasta alcanzar las formas finales. Esta maquinaria generativa es lo que se denomina un ***sistema formal***.

Más recientemente, una nueva generación de arquitectos jóvenes ha usado la potencia de cálculo de los ordenadores para dar un salto cuántico en la aplicación de las técnicas generativas a la creación de "*formas generadas por ordenador*". Son ejemplos paradigmáticos de esta tendencia las arquitecturas de formas organicistas (apodadas *blobs*) de arquitectos como Greg Lynn, Hani Rashid (Asymptote), Ben van Berkel (UN Studio), Lars Spuybroek (NOX) o el mismo Peter Eisenman. El uso de software de las industrias automovilística, aeronáutica y cinematográfica, de gran poder geométrico, les ha permitido generar formas totalmente innovadoras, rompiendo supuestamente con la tradición racionalista de la caja y del "*form follows function*".

La universalización del ordenador digital personal y su creciente potencia de cálculo (pero en especial la coincidencia entre su lógica interna y la agenda de estos nuevos diseñadores) ha permitido aumentar significativamente el ámbito de aplicación de los sistemas generativos. La concurrencia de nuevos formulismos geométricos (como las *mallas*, las NURBS[4], los *patches*, y otras herramientas de geometría diferencial) sobre la redoblada potencia de los ordenadores ha significado una auténtica revolución formal. Sin embargo, esta mayor *potencia generativa* y la libertad representativa ha sido interpretada en muchas ocasiones como un cambio de paradigma y como la apertura de nuevos horizontes epistemológicos, como el advenimiento de una *nueva arquitectura*. Esta ausencia de fronteras en la práctica, ha permitido también que se ensalce a la tecnología misma, bautizándose a la "*nueva*" arquitectura como arquitectura *digital*, *virtual*, *cyber-*

arquitectura o *hyper-arquitectura*. Aunque formalmente existe en estas corrientes una ruptura con la tradición de la Modernidad, no hay en el fondo tal fractura, sino tan solo una inmensa expansión del *espacio de posibilidades* facilitada de forma decisiva por el uso de ordenadores digitales. Es decir, el ordenador ha permitido explorar, de forma hasta ahora ilimitada, el espacio teórico de los sistemas formales.

Pero la utilización generalizada de estos aparatos digitales no ha *liberado* al arquitecto de la tradición Moderna como se pretende, sino más bien al contrario. Por su propia lógica interna, y por la aceptación de una serie de premisas, el ordenador ha sumido a estas arquitecturas generativas aun más adentro del modelo de la Modernidad.

Irónicamente, la herramienta que ha provocado la zozobra de los pilares más sólidos del racionalismo científico, y que ha hecho caer los muros del paradigma de la modernidad, ha encerrado a nuestra disciplina en su propia tradición de la que pretendía ser liberada por ella. Así que, como veremos, aún hay mucho de lineal, simple, racional, formal, estático, cerrado, de orden y de lógico en las actuales corrientes generativas a las que he eludido, y muy poco aún de complejo, dinámico, caótico, informal, adaptativo, abierto, diferente o rizomático. La arquitectura parece que no ha conseguido entrar aún, a pesar de lo que se ha podido pensar, en la superación[5] de la Modernidad.

Uno de los ejemplos que podría ilustrar convincentemente este desfase (aunque no sea necesariamente el más relevante) es la confusión existente, desde hace tiempo, entre dos conceptos próximos pero distintos que podríamos designar como lo ***genético*** y lo ***generativo***. Aunque ambos términos comparten una lejana raíz de origen protoindoeuropeo ([*genə–*]: producir, engendrar, nacer, de la cual derivan multitud de palabras), ambos términos aluden a dos paradigmas distintos referidos a la generación más o menos automática o mecánica de objetos. Cada uno de estos paradigmas tiene detrás una lógica bien definida, aunque no siempre bien entendida.

Este problema es especialmente considerable en la arquitectura (y en otras áreas *productivas* del conocimiento, como el arte, el diseño y la computación), donde han salido desde hace tiempo arquitectos que afirman que trabajan sobre “arquitecturas genéticas”, o entienden la “arquitectura como un programa”, o que “generan formas mediante tal o cual programa”.

Es por esta razón que parece muy importante (y lo será más en el futuro) ser capaces de identificar y distinguir estas dos maneras de generación de forma. Nos proponemos, pues, hacer una investigación a fondo de lo que significan estos dos paradigmas, y de cuáles son sus aplicaciones a la arquitectura.

¿Que es generativo?

El concepto de *generativo* (en la acepción que se le va a dar al término en este artículo) va indisociablemente ligado al concepto de ***sistema formal***. Un *sistema formal* es un dispositivo mental, un invento de la lógica y la matemática, que intenta capturar las particularidades esenciales de un contexto determinado, ya sea conceptual o del mundo real. Esta normalmente formado por una serie

de elementos, a saber: **(a)** un conjunto finito de *símbolos*, que son las piezas constitutivas del sistema, como los átomos de las moléculas o las letras de las palabras. Mediante la agregación de varios símbolos podemos formar **(b)** *cadenas de símbolos*, que serían las moléculas o las palabras del ejemplo anterior. En un sistema formal, generalmente, no todas las cadenas son aceptables: para saber cuáles son las normas de formación de cadenas necesitaremos unas **(c)** reglas de formación de cadenas de símbolos válidas, así como alguna forma de identificarlas como válidas; y también es imprescindible que tengamos una o más **(d)** reglas de producción, que nos permitan generar nuevas cadenas a partir de las que ya tenemos, y otras más a partir de estas últimas, y a sí sucesivamente. Pero la primera o primeras cadenas tienen que ser dadas, por lo que es menester uno o varios **(e)** *axiomas* o cadenas de símbolos iniciales, que no son producidas sino asumidas como parte del sistema por la propia definición.

A modo de ejemplo, un caso simple de sistema formal es el de los números naturales, que todos conocemos bien. En el sistema de números naturales, podemos usar *cifras* árabes {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} (símbolos) para formar *números* (cadenas de símbolos), siguiendo algunas reglas simples que no hace falta detallar. Son números válidos todos aquellos que incluyan uno o más símbolos (por ejemplo '7', '12345', y '9999'), y no lo son las cadenas que incluyan símbolos extraños (ejemplo: '34n3') o signos de puntuación que serían correctos en cadenas de otros sistemas ('103.654,326'). Además debemos proporcionar una regla o reglas para la producción de nuevas cadenas dentro del sistema: supongamos la regla $S(n) = n+1$ que obtiene el "numero siguiente" para cualquier número dado n. Y para que nuestro sistema sea completo, lo proveemos de algún *axioma* (en este caso será la cadena '1'), o cadena inicial, como *semilla* del proceso de generación. Armados con este "instrumental" podemos ***generar***, con la suficiente paciencia todos y cada uno de las cadenas de nuestro sistema, es decir todos los números naturales excepto el cero (y la razón por la que no podemos generar el cero, en este sistema en particular es porque no hay ninguna regla que nos permita alcanzarlo ya que se que se ha dado como axioma el '1').

Es importante destacar, sin embargo, que el sistema formal sólo se define basándose en su lógica interna, y que en él sólo se establece un funcionamiento mecánico para su autogeneración. No hay, pues, ninguna indicación del *significado* que tiene cada uno de los miembros del sistema, sino que más bien ese significado es una interpretación externa al sistema. Los sistemas formales son a menudo modelos de fenómenos reales, y muy utilizados para describir conjuntos de objetos. Sin embargo, no hay en ellos ningún indicio de la relación que tienen con el fenómeno que representan, es decir son *arbitrarios* e *inmotivados*[6].

Los conceptos de símbolo, cadena y regla pueden extenderse al ámbito formal de la arquitectura. Un ejemplo de tal sistema formal es el de los órdenes clásicos (dórico, jónico, corintio, etc.), donde los *elementos* (columnas, pilastras, muros, capiteles y molduras varias) se pueden combinar unas con otras formando piezas mayores, tales cómo pórticos, naves, fachadas, frontones, etc., y finalmente edificios enteros. El sistema queda plenamente determinado por la elección de un orden específico, que incluye las reglas de formación de los elementos, la combinatoria entre ellos y

las proporciones entre las partes. El axioma de un orden clásico sería el *módulo* que da la pauta para generar todos los elementos proporcionalmente.

Algoritmos y Máquinas de Turing

Sin embargo no es lo mismo establecer la estructura de un sistema (sus elementos y cadenas) que su *praxis*: la aplicación de reglas concretas sobre cadenas específicas para la obtención de resultados. La descripción del concepto de *regla* es más abstracta: se refiere al método o proceso formalmente definido (denominado ***algoritmo***[7] o computación) mediante el cual una cadena de símbolos se transforma en otra. En su forma más simple, un *algoritmo* es el proceso secuencial sistemático por el que símbolos o números de entrada (*input*) son convertidos en símbolos o números de salida (*output*), en un numero finito de pasos. La formulación precisa y genérica del concepto de algoritmo es del siglo XX, y su especificación más importante es en los términos de una ***máquina de Turing***.

Una *máquina de Turing* es un aparato imaginario, una máquina teórica capaz de efectuar operaciones elementales precisas sobre símbolos. Funciona como una ecuación o una fórmula, convirtiendo una cadena de símbolos en otra siguiendo un conjunto de reglas. El mecanismo definido por Turing[8] puede leer y escribir sobre una cinta de papel (posiblemente infinita) sobre la que puede haber o bien un **0** bien un **1** sobre cada una de las celdillas en las que está dividida. La máquina posee además una tabla de instrucciones finita, y un *estado* interno (una única posición de memoria) en cada momento, y actúa por pasos discretos. Cuando la máquina empieza su proceso, lee el contenido de una celda sobre la cinta de papel y realiza una acción (borrar y escribir, desplazar la cinta una posición a la derecha o a la izquierda, pararse, etc.), que viene unívocamente determinada en la tabla de instrucciones dependiendo de: (a) el estado interno de la máquina, y (b) la cifra en la celdilla escaneda.

La máquina de Turing es, de hecho, un sistema formal. Existen, además, infinitas máquinas posibles, una para cada proceso imaginable, o dicho de otra manera, una para cada *algoritmo*. Además, Turing introdujo posteriormente la ***máquina universal de Turing (UTM)*** que engloba y reproduce a todas las maquinas de Turing individuales posibles. Imagine que cada algoritmo es descrito por una máquina de Turing de manera estándar: el proceso de leer, interpretar y ejecutar las instrucciones de ese algoritmo es un proceso mecánico en sí mismo, y puede ser personificado en otra máquina de Turing especial a saber, la *máquina universal de Turing*. Podemos hacer que una máquina universal haga cualquier tarea que una maquina de Turing particular pudiera hacer: ¡Una máquina para todas las tareas posibles! Para conseguir esta proeza sólo sería necesario suministrar la descripción de la máquina particular a la maquina universal, codificada en forma de una cadena de símbolos sobre la cinta de lectura: la máquina universal de Turing reúne pues, en un solo lugar (sobre la cinta de papel), la descripción de las instrucciones y los datos a procesar. No es difícil pensar una maquina de Turing como un ***programa*** (de hecho ya hemos dicho que es un algoritmo), y que la máquina universal es un ***computador***, encargado de interpretar y obedecer el programa. La

máquina universal encarna la idea esencial del ordenador: una máquina que puede realizar cualquier tarea bien definida si se le suministra el programa adecuado y los datos necesarios.

La máquina de Turing, junto a la *arquitectura von Neumann* tiene una importancia crucial en la definición e invención de los ordenadores electrónicos. La ***arquitectura von Neumann*** es un modelo de ordenador que usa una única estructura de almacenamiento para el conjunto de instrucciones sobre cómo realizar los cálculos (el programa) y los datos requeridos o producidos por él. John von Neumann[9] creó este modelo como un ejemplo de una máquina de proceso universal, un concepto que también está implícito en la máquina universal de Turing. Al tratar de forma idéntica a las instrucciones y los datos, la máquina puede fácilmente cambiar su propio código, en otras palabras es reprogramable. La nueva dimensión que von Neumann dio al ordenador digital ha llegado hasta nuestros días en su forma actual.

Podríamos decir, sin faltar a la verdad, que el ordenador digital tal y como lo conocemos hoy tiene sus inicios y sus fundamentos en al menos tres ideas fundamentales de tres grandes científicos de nuestro tiempo: las *máquinas de Turing* y la *arquitectura von Neumann*, que ya hemos visto, y las *gramáticas generativas* de Noam Chomsky.

Autómatas y Gramáticas Generativas

Pero von Neumann también merece nuestra atención en relación a la ***teoría de autómatas***. En general, un autómata es cualquier máquina artificial que funciona "por si sola", lo que denominaríamos un *"robot"*. En informática, sin embargo, un ***autómata de estados finitos*** (FSM), o *máquina de estados finitos*, es una máquina abstracta, parecida a la máquina de Turing. De nuevo, como en el caso de ésta última, el concepto de *máquina* se refiere abstractamente al *proceso* que realizaría dicha máquina imaginaria de existir, es decir, ejecutar una serie de pasos.

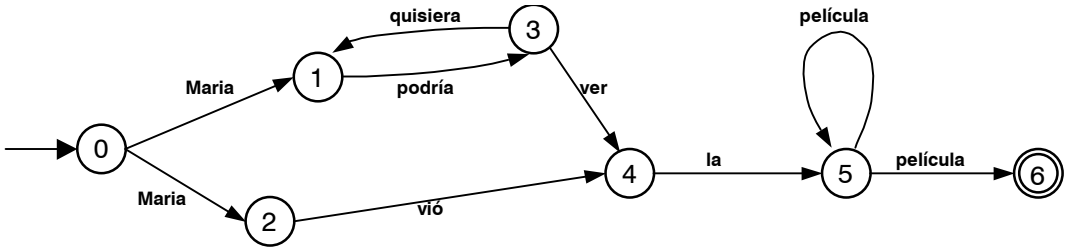


Figura 1 – Grafo de transición finita (ejemplo)

Supongamos que definimos un grafo como el de la Figura 1, que determina un conjunto (*vocabulario*) de cadenas de símbolos y los posibles caminos entre ellas cruzando los nodos de la red. Cada camino completo (empezando en el nodo 0 y terminando en el 6) a través del grafo representa una *frase*, y el conjunto de frases posibles es el *lenguaje* definido por la red.

Una *máquina de estados finitos* consiste en un grafo como el anterior, a lo que se le añade además el proceso algorítmico que describe como y bajo que condiciones pasar de un nodo a otro, es decir, las reglas de *fabricación* (o *generación*) de frases del lenguaje: empezando en el nodo

inicial, la *máquina* lee una cadena del alfabeto, y usa las reglas de generación para determinar cuál es la siguiente cadena.

Este modelo descriptivo opera sobre *lenguajes* de vocablos definidos y finitos, aunque también puede usarse en lenguajes menos estructurados o infinitos, pero siempre formalizados. Por ello, como es comprensible, los autómatas de estados finitos son muy usados en la descripción de lenguajes, especialmente la de lenguajes formales que son los que usan los ordenadores, por lo que también son muy usados en informática teórica.

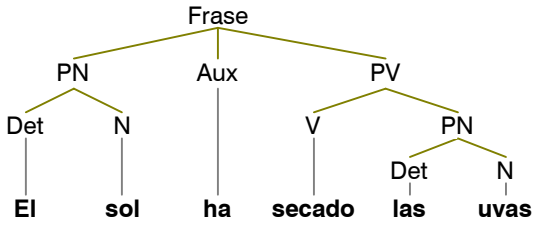
En el campo de la lingüística, por fin, Noam Chomsky[10] es el autor e introductor de las influyente y potentes ***gramáticas generativas*** (o *gramáticas transformativas*) con su tesis doctoral[11],. Esta teoría explica como las locuciones del discurso lingüístico (palabras, frases) corresponden a *estructuras superficiales* abstractas que a su vez corresponden a *estructuras profundas* aún más abstractas. La creación (y la interpretación) de las palabras, frases y locuciones del lenguaje depende en todo momento de las *reglas transformacionales generativas*, que junto con un conjunto finito de términos(raíces, morfemas, etc.) pueden producir un número infinito de frases en ese lenguaje. Chomsky cree que la capacidad gramatical es innata en los seres humanos, y que estamos dotados desde el nacimiento de una *gramática universal*.

Una *gramática generativa* (o también llamada *gramática formal*) ve la lingüística como una operación combinatoria de fabricación del lenguaje. Consiste en una serie de reglas mediante las cuales es posible decidir (generar o interpretar) si una frase dada es correcta o incorrecta. En el fondo, no es más que un ***algoritmo*** que permite *generar* todas y sólo las frases gramaticales de un lenguaje.

En su forma más simple, la gramática generativa es una *máquina de estados finitos* (ver más arriba, dónde cada regla de producción tiene la forma A → b (dónde *A* es un símbolo no terminal y *b* es una cadena de símbolos terminales y no terminales). Así, por ejemplo, podemos definir la siguiente gramática:

(a). Frase → PN Aux PV
(b). PN → Det N
(c). PV → V PN

Con esta simple gramática podemos, en tres o cuatro pasos, generar e interpretar frases del tipo:



Las gramáticas generativas son sistemas formales y los sistemas formales son gramáticas generativas. La gramática generativa describe un lenguaje formal, el conjunto (posiblemente infinito) de palabras (finitas) extraídas de un alfabeto (finito) de símbolos. Y un lenguaje formal puede ser

especificado de muchas maneras, de las cuales hemos visto algunas: producido por una gramática generativa, por un autómata de estados finitos o por una máquina de Turing.

La arquitectura de los arquitectos denominados *New York Five* (Peter Eisenman, Robert Graves, Charles Gwathmey, John Hejduk y Richard Meier)[12] fue la primera y más importante aplicación del concepto de gramática generativa a la generación de formas. Su trabajo, especialmente los diseños de Eisenman para las *Casas I-X*, parten de formas geométricas autónomas, supuestamente de origen racionalista (Terragni, Le Corbusier, neo-cubismo, etc.) que se deforman y transforman según los dictados de una *gramática*, tal y como la muestran sus propios diagramas generativos.

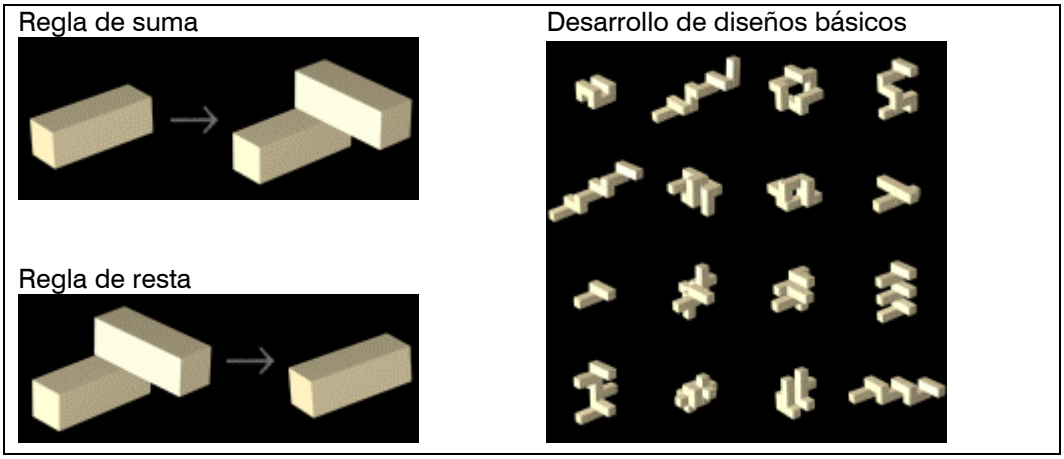


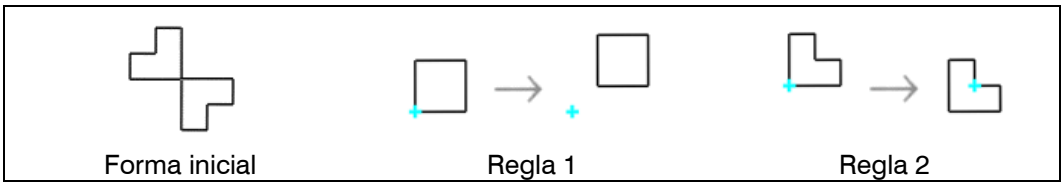
Figura 2 – Ejemplo de una *shape grammar*.

Las gramáticas generativas de Chomsky han sido decisivamente influyentes en muchas áreas productivas, tal es su adaptación a la lógica del proceso, la estructura y el algoritmo. Como muestra del maridaje entre las gramáticas y la forma geométrica presentamos las *gramáticas de forma*(***shape grammars***) descritas por primera vez por George Stiny[13]. Las *shape grammars* son gramáticas generativas que proponen una aproximación computacional a la generación de diseños (según el autor son "la maquinaria formal para la definición algorítmica de lenguajes de diseños espaciales en dos y tres dimensiones"). En las *shape grammars* (ver Figura 2 y Figura 3), los *símbolos* son primitivas geométricas sencillas (líneas, puntos o arcos, y en tres dimensiones, cubos, cilindros o superficies NURBS), y las *cadenas de símbolos* son diseños de objetos arquitectónicos, formados por agregación de entidades primitivas. Las reglas de transformación pueden provocar el cambio de una forma en otra, generando diseños por la recursiva aplicación de reglas sobre formas.

Las arquitecturas descritas más arriba (basadas en sistemas formales) o las del principio del texto (las generadas por ordenador) siguen invariablemente el mismo patrón: la utilización de gramáticas (como las *shape grammars*) para transformar unas formas axiomáticas y arbitrarias en otras de nuevas, sucesivamente, siguiendo reglas geométricas de transformación: es por esta razón que podemos clasificarlas como ***arquitecturas generativas***.

También en los programas de modelado geométrico (diseño asistido por ordenador o *CAD)* el usuario puede transformar unas formas en otras mediante la aplicación de reglas preestablecidas, de

las que hay un numero finito. Algunas reglas sencillas, aplicadas manual o automáticamente, son transformaciones afines como extrusiones, traslaciones o rotaciones, aunque otras son más complejas al modificar los valores de los distintos parámetros que definen cada una de las formas (factores de las funciones de base, red de puntos de control, etc.). Como todos los programas de software, los programas de modelado geométrico son máquinas de Turing, y por consiguiente sistemas formales; concretamente, son la encarnación de *shape grammars* [paramétricas] lo que nos refiere directamente a las gramáticas generativas descritas anteriormente. Así pues, la creación automática de formas mediante programas de modelado geométrico sigue el espíritu y la letra de una gramática y también pueden ser denominadas, sin riesgo a equivocarse como ***generativas***.



Ejemplos de diseños:

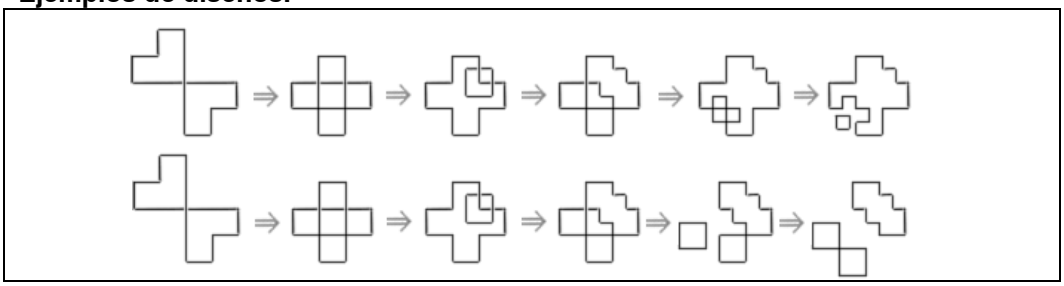


Figura 3 – Ejemplo de una ***shape grammar*.**

Vemos pues que aquellas arquitecturas que querían autodenominarse como avanzadas, nuevas o reformadoras no son sino la extensión, mediante nuevos mecanismos, de procesos generativos que tienen que ver con la racionalidad, el formalismo y el proceso. No hay en ellas nada nuevo. No implican una evolución positiva de la metodología de la creación arquitectónica. En resumen, no hacen más que acentuar el paradigma de la Modernidad que pretenden superar.

¿Que es genético?

El concepto de ***genético*** es a mi modo de ver, sustancialmente diferente y puede sugerir caminos totalmente inexplorados y distintos del anteriormente expuesto. Colocada en su correcto contexto, investigando cuidadosamente qué significa *genético* y trasladando el concepto al campo de la arquitectura, esta investigación puede revelarse sugerente y aportar ideas nuevas a la metodología y el proceso creativo.

El descubrimiento por parte de Watson y Crick[14], en 1953, de la molécula y la estructura del ácido desoxirribonucleíco (ADN) explicó y dio forma física a los *genes*, la unidad abstracta de herencia descrita por Mendel en 1866 para definir funcionalmente las características internas, externas y el aspecto de un ser vivo cualquiera (***fenotipo***). La molécula de ADN está constituida por

una doble espiral helicoidal de parejas de nucleótidos (Adenina–Tiamina, Citosina–Guanina) o *bases,* que funcionan como las *letras* del ***código genético***: cogidas de tres en tres, las bases expresan la formación de un *aminoácido* concreto, y largas cadenas de aminoácidos forman las *proteínas*, auténticos fundamentos de la construcción y funcionamiento de los organismos vivos, y que determinan (en parte, como veremos) el fenotipo. Para la fabricación de proteínas (la actividad más fundamental de la célula), fragmentos de la cadena de bases de ADN se transcriben en una cadena de mARN[15], una molécula similar que viaja fuera del núcleo de la célula hasta los *ribosomas*, orgánulos celulares encargados de transcribir el código genético y sintetizar las moléculas de proteína.

Gracias al descubrimiento de este bello mecanismo nació la atractiva idea de que en el genotipo se encuentra ***la información que codifica el fenotipo***, es decir, que existe un isomorfismo entre genes y aspecto. De ahí que se haya impuesto la idea, tanto en círculos científicos como en el saber popular, de que existe un *código de la vida* que determina unívocamente (eso es, *determinísticamente*) las características y rasgos de un ser vivo al ejecutarse. Este es, en resumen, el denominado *dogma central* de la biología molecular, idea central del denominado *determinismo genético*.

Es por esta razón que el término genético se ha asociado siempre al de generativo, pues existe cierto paralelismo entre el dogma central y un sistema formal: las bases (A-T-C-G) forman cadenas (genes), que se copian en el mARN, que es llevado a los ribosomas donde, como si de una máquina de Turing se tratara, se fabrican las proteínas según una tabla de instrucciones preexistente: el ***código genético***[16]. Y también por esta razón la idea de *genético* es tan atractiva para los amantes de la generación automática de formas: el sistema generativo resultante muestra las virtudes de ser fácilmente formalizable, al mismo tiempo que es una metáfora importada de la biología para generar formas orgánicas.

Sin embargo la biología celular ha sabido desde hace años (aunque no siempre lo ha precisado) que en la célula existen no uno, sino dos sistemas de información: el ***genético*** y el ***epigenético***. El familiar sistema genético (ADN → mARN → Proteína → Fenotipo) es aplicable directamente a un pequeño numero de fenotipos, que en los humanos se cuantifica en un 2%[17]. Sin embargo, el sistema *epigenético* es el gran desconocido y el gran ignorado, y vale la pena pararnos un momento a ilustrarlo.

La trascripción del ADN en mARN, la traducción del mARN según el código genético y la producción de proteínas en los ribosomas no ocurre en el vacío, sino en una *sopa* de compuestos moleculares (el ***sistema epigenético***) que interactúan entre sí, se potencian o inhiben entre ellos, o se retroalimentan continuamente en una compleja dinámica no lineal. Así, los genes y sus productos (las proteínas) son juez y parte de la mayoría de reacciones. Desde la trascripción del ADN a la formación de las intrincadas estructuras moléculas tridimensionales de las proteínas, el resultado de todas las reacciones químicas de la célula está condicionado por la presencia de determinadas proteínas (las *enzimas*); el propio código genético está formado por proteínas (tARN), ellas mismas fabricadas por el mismo proceso epigenético, en un complicado argumento circular. Una multitud de

procesos paralelos superpuestos hacen que la visión del dogma central diste mucho de describir fidedignamente la realidad. El sistema epigenético es un sistema inestable de gran complejidad interpuesto entre los elementos genéticos unitarios y el fenotipo final, sin duda un sistema dinámico no lineal de naturaleza caótica, del que es posible esbozar un mapa de sus componentes pero muy difícil, si no imposible, determinar su resultado[18].

La célula no es, por lo tanto, un sistema cerrado y lineal, equiparable a un sistema formal, sino un sistema complejo abierto, dinámico y autorregulado. Y los organismos vivos son sistemas de sistemas muy diferentes de meras máquinas fisiológicas. Si bien el sistema genético es lineal, los fenotipos dependen de un contexto (el sistema epigenético) no-lineal y complejo. En los seres vivos, la expresión del *fenotipo* no es un fenómeno determinista, sino la emergencia de propiedades generadas por la interacción compleja de multitud de interacciones simples.

Visto de esta manera, podemos vislumbrar nuevas vías de generación de formas basada no ya en sistemas generativos, sino en una aproximación a lo complejo, de forma que aprendamos a crear (no ya *forma*, sino *arquitectura*) en contextos no-lineales, dinámicos y abiertos. No existe en la actualidad ningún paralelismo entre el sistema genético-epigenético y la arquitectura. No se ha hecho ninguna aproximación a la genética de la forma, ni a la generación de arquitectura compleja (recordemos que si *forma* corresponde a *fenotipo*, *arquitectura* equivale al organismo complejo).

Cabe la posibilidad, si se me permite la licencia, de contemplar la arquitectura (y en especial el proceso creativo) como un sistema de información complejo, donde una multitud de elementos simples interactúan entre ellos hasta hacer emerger un sistema de mayor complejidad. Tan sólo tenemos que ser capaces de redefinir *arquitectura* como un sistema dinámico, y entender cuáles son las partes y su esquemática de interacciones. Este simple cambio de perspectiva situaría a la arquitectura (como disciplina) en al ámbito de la teoría de la información, abriéndose un sinnúmero de posibilidades procesuales y creativas.

Para ello tenemos, por irónico que parezca, un aliado excepcional: el ordenador. Más allá de su probada capacidad como ejecutor de gramáticas generativas, el ordenador dispone de la capacidad de ***simular*** (y nosotros de crear los modelos). Aun que la simulación es un sistema formal cerrado, su interpretación y significado (como en el caso de los lenguajes) es una organización de nivel superior. Tal vez éste sea la agenda para una revisión de la arquitectura, que nos permita por fin superar el paradigma de la Modernidad.

NOTAS

[1] Rudolph Wittkower: *Architectural principles in the Age of Humanism,* London, Warburg Institute, University of London, 1949.

[2] Colin Rowe, en su artículo "Las matemáticas de la vivienda ideal", aparecido en castellano en *Manierismo y Arquitectura Moderna, y otros escritos*. (Barcelona: Gustavo Gili: 1978. pp.9s), agradece explícitamente la influencia de Rudolph Witkower.

NOTAS

[3] Peter Eisenman fué uno de los cinco integrantes de la exposición y el libro-catálogo *Five architects: Eisenman, Graves, Gwathmey, Hejduk, Meier (*New York: Wittenborn: 1972), prologado por Colin Rowe; juntamente con la interpretación matamática de Rowe de la arquitectura, la influencias más importante en la obra de estos arquitectos fueron las gramáticas generativas chomskianas (que veremos más adelante) apelando a la naturaleza lingüística de la arquitectura.

[4] Siglas de las *Non-Uniform Rational B-Splines*, que son la base de los *blobs*.

[5] No deseo usar el término *posmodernidad* por su abultada lista de acepciones: *à la* Lyotard, *à la* Jameson, *à la* Jencks, etc. Desearía usarlo en su sentido real, de *superación de la Modernidad*.

[6] Ferdinand de Saussure: *Curso de lingüística general*.

[7] La palabra *algoritmo* deriva del nombre del matemático persa Abu Ja'far Mohammed ibn Mûsâ al-Khowârizm (s.IX).

[8] **Alan Turing** (Londres 1912- Wilmslow 1954) fue uno de los más grandes matemáticos de nuestro siglo. Nacido en Inglaterra, y educado en matemáticas en la universidad de Cambridge (Reino Unido), se introdujo desde joven en el complejo mundo de la lógica matemática por inspiración de los *Principia Mathematica* de Bertrand Russell. En 1936, se propuso resolver un problema planteado por Hilbert: ¿puede existir, al menos en principio, un método preciso por el cual se pudiese decidir si cualquier afirmación matemática puede ser probada? La solución de Turing (aunque L. Church se adelantó unos meses en la publicación de los resultados) se ha considerado siempre más original y diferente, y su escrito de 1936 ha dejado además el concepto de ***máquina de Turing***, que se ha convertido en la herramienta básica de la teoría moderna de la computación. Para resolver tal reto, Turing necesitaba una definición de *método* o *proceso* mediante el cual abordar el problema de Hilbert con precisión y contundencia, e ideó una máquina teórica capaz de efectuar operaciones elementales precisas sobre símbolos, y argumentó convincentemente de que el alcance de dicha máquina cubría todo lo que Hilbert consideraba como 'método preciso' (lo que actualmente denominaríamos ***algoritmo***).

[9] **John von Neumann** (Budapest,1903-Washington DC, 1957) fue un científico relevante en muchos campos. Entre 1921 y 1926, estudió en Budapest, Berlín y en el prestigioso politécnico de Zürich, e impartió clases en Berlín, Hamburgo y Götingen, donde trabajó con David Hilbert; desde 1930 fue profesor de Princeton, donde fue nombrado uno de los seis profesores, junto a Albert Einstein entre otros, del recientemente creado *Institute for Advanced Study*, posición que conservó hasta el final de sus días.

Al principio, su carrera estuvo centrada alrededor de la lógica matemática y la teoría de conjuntos, pero su trabajo incansable influyó decisivamente en áreas tan diversas como la mecánica cuántica, donde contribuyó decisivamente al desarrollo del campo; la mecánica estadística que él contribuyó a inventar; la creación de las álgebras de von Neumann; la teoría de los juegos, importantísima para el estudio del moderno comportamiento económico; y el estudio de las complicadísimas ecuaciones diferenciales parciales no-lineales, cuyo manejo le indujeron al estudio de los métodos numéricos, lo que a su vez le llevó a interesarse por las nuevas posibilidades del cálculo numérico mediante computadores electrónicos. Durante la II Guerra Mundial, puso a disposición de distintos proyectos de investigación militar su valiosa experiencia en hidrodinámica, balística, meteorología, teoría de los juegos y estadística, y como coordinador de algunos equipos de ingenieros, haciendo de correa de transmisión entre las necesidades del Los Alamos National Laboratory (y del *Proyecto Manhattan*, código en clave del equipo que desarrolló la bomba atómica), y de los ingenieros de la Universidad de Pennsylvania que estaban construyendo el ENIAC, el primer computador electrónico. En 1944, él mismo empezó a construir con su propio equipo un ordenador en el *Institute for Advanced Studies* el *IAS*. Tras la guerra, se concentró en el desarrollo del IAS y en la definición de la ***teoría de autómatas*** .

[10] Nacido en 1928, el Dr. **Avram Noam Chomsky** es profesor de lingüística del Massachusetts Institute of Technology (MIT), además de un teórico de la izquierda socialista americana y un polémico analista político.

11 Noam Chomsky: The Logical Structure of Linguistic Theory. PhD tesis, University of Pennsylvania: 1955 (publicada por Plenum Press, New York: 1975).

[12] Ver nota 3.

[13] George Stiny: Introduction to shape and shape grammars. En E*nvironment and Planning B, 1980, vol. 7 pp. 343-351.*

[14] James D. Watson: *The Double Helix: A Personal Account of the Discovery of the Structure of DNA,* New York : Atheneum, 1968.

[15] ARN: ácido ribonucleíco; mARN: ácido ribonucleíco *mensajero*.

[16] Para una explicación no excesivamente técnica del funcionamiento de la célula como un sistema formal, véase: Hofstadter, Douglas R.: *Gödel, Escher, Bach : an eternal golden braid*. New York : Basic Books, 1979.

[17] Strohman, Richard C.: Linear Genetics, Non-Linear Epigenetics: Complementary Approaches To Understanding Complex Diseases. En *Integrative Physiological & Behavioral Science*, Sep-Dec 1995, Vol. 30 num. 4, pp.273s.

[18] Esto es debido a la naturaleza misma de los sistemas caóticos. Véase: James Gleick: *Chaos: Making a New Science*. Penguin, 1987.