El rol de las reglas heurísticas en los problemas intratables

*Víctor Rodríguez**

Cómo es sabido, existe una amplia tradición de investigación que intenta caracterizar a la actividad científica como una actividad de planteamiento y de resolución de problemas. Tomando este enfoque general, lo que hago aquí son algunas reflexiones sobre el concepto mismo de problema científico, bajo la suposición de que nos encontramos con uno de esos conceptos de considerable vaguedad intrínseca, e intento acercar a la filosofía de la ciencia una línea de investigación que está aportando en mi opinión elementos de interés para una mejor elucidación del mismo. Me refiero al área de la complejidad computacional, aunque en realidad podríamos pensarla como un emergente de una trama de relaciones entre varias disciplinas: la lógica matemática, la teoría de autómatas, las teorías de lenguajes formales, y varias ramas de las ciencias de la computación, especialmente la teoría abstracta de la computabilidad. Por razones de espacio y también de eventual impacto sobre la filosofía de la ciencia, he recortado el análisis de esta temática a los llamados problemas intratables, ya que los mismos brindan sugerencias interesantes para un proyecto de taxonomía de problemas, a la vez que generan interesantes desafíos a la epistemología. El concepto de regla heurística subyace con diferentes grados de generalidad a todo el trabajo.

Si bien el concepto de problema científico ha sido trabajado por los filósofos de la ciencia, también lo ha sido por científicos de varias disciplinas. Polya sentó una tradición y un estilo en lo referente al abordaje de problemas en matemáticas. Las escuelas rusa, francesa, y otras, en varias ramas de la ciencia, generaron sus propias maneras de caracterizar un problema y también de algún modo sus propias estrategias para buscar la solución. Normalmente esto se reflejó en un estilo de gradación de problemas atendiendo a diferentes tipos de dificultades, desde cuestiones de complejidad operativa hasta las raíces conceptuales de las teorías subyacentes. De este modo, tanto el planteamiento, como la resolución de problemas en diversas disciplinas científicas oscilaron entre la sensibilidad pedagógica y la profundidad filosófica. Al parecer, esta tendencia se mantiene.

Pero la matemática es un caso especial, por el nivel técnico involucrado en el planteamiento de los problemas. En la mayoría de las áreas de investigación que no son tan dependientes de lenguajes semi-formales, gran parte del uso del concepto de problema está asociado a un estilo de formulación de preguntas y a intentos de articulación de respuestas sobre bases teoréticas determinadas. Esto es realmente un lugar común en el contexto de la filosofía, y también lo es en mi opinión en buena parte de la filosofía de la ciencia contemporánea.

Resulta bastante atractiva en este sentido la propuesta de Hintikka y la escuela finlandesa, de explorar las consecuencias de un enfoque asociado a preguntas "Por qué", las llamadas "Why Questions". Como es sabido, este es un programa de investigación bastante consolidado. Otra línea que ha generado muchos subproductos, y sobre la que ya hemos

* Facultad de Filosofía y Humanidades, Universidad Nacional de Córdoba.

expuesto en varias ocasiones, es la que asocia a los descubrimientos científicos con el planteamiento y la resolución de problemas en contextos de diversos grados de generalidad y con diferentes matices procedurales. Me refiero al extendido programa de Herbert Simon y colaboradores. Un tercer enfoque bastante considerado está relacionado con la dinámica teórica en el contexto de la ciencia normal en sentido kuhniano. A pesar de lo trillado de este enfoque, da la impresión de que todavía merece consideración e investigación, dado que subyace algo similar al mismo en numerosas prácticas científicas y en mi opinión estamos muy lejos de tener una clara idea de lo que es resolver un problema dentro de la fase de ciencia normal. Finalmente, para nuestros fines, llama también la atención el destino del proyecto de Larry Laudan de comparar teorías por su potencial eficacia en la resolución de problemas. En mi opinión asistimos en este caso a una situación curiosa entre una línea sumamente atractiva de investigación en filosofía de la ciencia, que parece más que estar agotada, recién estar en sus comienzos, y una aparentemente irremediable vaguedad en la caracterización de la noción de problema científico. Menciono estos enfoques porque son representativos de la investigación filosófica en torno del concepto de problema y porque en ninguno de ellos tiene todavía un lugar la caracterización de los problemas dada por la complejidad computacional, ni aún en un programa de investigación tan cercano como el de Simon, que ha contribuido en buena medida al acercamiento de la computación a la filosofía de la ciencia. En relación con el título de nuestro trabajo, esto también ha contribuido a que no tengamos una noción robusta a la vez que de suficiente generalidad de regla heurística.

Pero al margen de la computación, si estos programas de investigación siguen gozando de buena salud, al menos para los filósofos de la ciencia, y si podemos captar a una buena parte de la naturaleza de la ciencia a través del análisis de sus problemas e intentos de soluciones, el concepto de problema dentro de cada uno de ellos requiere un análisis mucho más detallado. Normalmente esos abordajes generales no satisfacen las expectativas epistemológicas y es natural que así sea, dado la vasta gama de problemas con sus respectivos intentos de solución. Uno alude al concepto de problema de manera similar a como lo hace con el concepto de teoría. Todo el mundo parece saber de qué está hablando cuando usa estos términos, a pesar de que la vaguedad y ambigüedad acechan en cualquier inferencia que los invoque. Podría pensárselos como ingredientes que conforman el sentido común de los filósofos de la ciencia. Algo así como elementos indispensables en una filosofía *folk* de la ciencia.

Para intentar avanzar un poquito en esta cuestión, pretendo elaborar una primera aproximación al contexto epistemológico en torno de la noción de problema desde el contexto científico particular mencionado. Naturalmente, este intento tiene sus riesgos, como la mayoría de las extrapolaciones de contextos específicos a otros más generales; además, la elección de paradigmas computacionales suele considerarse como demasiado restrictiva para buenos sectores de la comunidad filosófica, pero hasta donde conozco el tema es el único sector de la investigación científica que ha producido importantes avances en torno del concepto de problema.

En lo que sigue vamos a considerar que un problema es una pregunta general que posee usualmente varios parámetros cuyos valores quedan sin especificar. Seguimos aquí el tratamiento standard que es usual en los textos y artículos clásicos de teoría de la computación y áreas afines. Para caracterizar un problema, se da una descripción general de todos sus parámetros y un enunciado de qué propiedades debe satisfacer la respuesta o solución. Tenemos una instancia del mismo cuando especificamos valores particulares para todos los parámetros del problema.

Hay varios modos usuales de encontrar la solución para un problema determinado. Aún cuando la línea divisoria no siempre es nítida, podemos esbozar cuatro ámbitos principales de abordaje sistemático de un problema:

a) La aplicación de procedimientos asociados a reglas heurísticas particulares, tales como ensayo y error, o la ponderación exhaustiva de casos, usualmente asociada a variantes de enumeración. También son frecuentes las reglas de búsqueda guiadas por conjeturas conservadoras o audaces.
b) El uso de un algoritmo que lleva a la solución del problema.
Entendemos aquí algoritmo en general como un procedimiento explícito paso por paso para resolver problemas. Es común que se asocie a los algoritmos con programas computacionales escritos en algún lenguaje de computación preciso.
c) El uso de una definición recursiva.
d) La aplicación de una fórmula explícita que da la solución.

Usualmente cuando no se da esta última situación, la de hallar una fórmula explícita, se construyen algoritmos de aceptable complejidad para abordar los problemas. Dicha complejidad generalmente se mide tanto por el esfuerzo de cómputo que se necesita para evaluar la fórmula o su equivalente en versión algorítmica, como por la relación entre esta capacidad de cómputo y la longitud o tamaño de los datos de entrada del problema.

Decimos que un algoritmo resuelve un problema si se puede aplicar a cualquier instancia del mismo y se puede garantizar siempre que produce una solución para esa instancia dada. En general se desea encontrar el algoritmo más eficiente para resolver un problema, siendo esta noción de eficiencia fuertemente dependiente de los recursos que se necesitan para ejecutar el algoritmo. Esto ha generado varios términos técnicos que son necesarios debido a que hay diferentes modos de describir las instancias de un problema dado y es natural intentar poner un poco de orden entre los mismos. Sólo tomaremos algunos conceptos básicos necesarios para lo que sigue:

a) Un concepto de medida de una instancia de un problema, que pretende reflejar la cantidad de datos de entrada necesarios para describir la instancia.
b) Una función de complejidad temporal para un algoritmo dado, que expresa los requerimientos temporales para dar para cada longitud de entrada posible, la cantidad mayor de tiempo que se necesita para resolver una instancia del problema de esa medida. Lo de la cantidad mayor aquí significa que la función pretende ser sensible al peor de los casos y no al más corto. De lo que se trata es de evaluar al algoritmo de acuerdo con su desempeño sobre la peor entrada posible que se le puede presentar.
c) Se define un algoritmo de tiempo polinómico como aquel cuya función de complejidad está acotada por otra función polinómica.
d) Un algoritmo cuya función de complejidad temporal no puede ser acotada de este modo, se llama algoritmo de tiempo exponencial.

La mayoría de los algoritmos de tiempo exponencial son variaciones de búsquedas exhaustivas, mientras que los de tiempo polinómico generalmente captan algo más profundo de la estructura de un problema. La práctica en estos campos ha llevado a considerar que un problema puede ser bien resuelto si puede ser abordado exitosamente con un algoritmo de tiempo polinómico. A los que no se los puede tratar de este modo se los llama problemas intratables.

El concepto de intratabilidad ha resultado ser mucho más fructífero de lo que se había imaginado hace algunas décadas. En particular, es considerablemente independiente del esquema de codificación particular del problema y del modelo de computadora que se use para determinar la complejidad temporal. Se ha mostrado que los esquemas de codificación standard que se usan en la práctica para cualquier problema particular siempre difieren a lo sumo polinómicamente uno de otro. Algo similar sucede con los modelos realistas de computadoras estudiados hasta ahora. Las máquinas de Turing, las máquinas de Turing de cintas múltiples y las máquinas de acceso aleatorio, son equivalentes con respecto a la complejidad temporal polinómica. La clase de problemas intratables no queda afectada por el modelo de computadora que se usa, al menos en las versiones conocidas de computadoras abstractas.

Con respecto a la intratabilidad, hay dos causas diferentes para la misma a tener en cuenta. Una es cuando el problema es tan difícil que se necesita un tiempo exponencial para descubrir la solución. La otra es que la solución en si misma deba ser tan extensa que no puede ser descrita con una expresión que tenga longitud acotada por una función polinómica de la longitud de entrada. De ambas, la intratabilidad más común es la primera. Aquí conviene establecer una distinción entre intratabilidad e indecidibilidad. Recordemos que problemas decidibles son aquellos que se pueden resolver por sí o por no.

La indecidibilidad es anterior a la computación. Su primer gran resultado fue el problema del halting, resuelto por Turing. La entrada para este problema es un programa de computadora, junto con sus datos de entrada; el problema es decidir si el programa eventualmente parará. La dificultad aparece debido a la posibilidad de búsqueda no acotada. La solución obvia es correr el programa hasta que se pare, pero no es claro cuando es correcto abandonar la búsqueda y decidir que el programa no parará. Turing demostró usando la técnica de diagonalización heredada de Cantor, que no existe un algoritmo que pueda controlar exitosamente todas las instancias del problema. Otro notable ejemplo es la solución del llamado décimo problema de Hilbert. Se trata de un ejemplo tomado de la teoría de números y consiste en resolver ecuaciones diofánticas. Esto es, dada una ecuación polinómica, averiguar si existe una solución en números enteros. Hilbert planteó en 1900 el problema de encontrar un procedimiento de decisión general para resolver tales ecuaciones diofánticas. Matiyasevich demostró en 1971 que no existe tal procedimiento de decisión.

Los primeros problemas intratables, pero decidibles se obtuvieron en la década del 60 en relación con jerarquías de complejidad. Se suele decir que estos primeros problemas fueron construcciones bastante artificiales. En los 70 se arribó a problemas intratables más cercanos a la práctica científica. Hay aquí problemas de teoría de autómatas, de lógica matemática y de teoría de lenguajes formales. Las demostraciones muestran que estos problemas no se pueden resolver en tiempo polinómico usando aún un modelo computacional "no determinista", que puede realizar un número no acotado de sucesiones independientes en paralelo.

Uno de los principales subproductos de esta práctica ha sido el hallazgo de interesantes relaciones entre problemas, que a su vez ayudan en la construcción de nuevos algoritmos. Una técnica importante es la de reducción de un problema a otro, dando una transformación constructiva que proyecta cualquier instancia del primer problema en una instancia equivalente del segundo. Esto proporciona los medios para convertir a un algoritmo que resuelve el segundo problema en un algoritmo correspondiente para resolver el primero. Dicho de otro modo, un problema X se dice que es reducible a un problema Y si, dada una subrutina capaz de resolver el problema Y, puede construirse un algoritmo para resolver el problema X. Se demostró, por ejemplo, que el problema del halting es reducible al décimo problema de Hilbert, de lo que se sigue que este último debe ser indecidible.

Hay muchos ejemplos de reducciones exitosas que anticiparon la revolución que se produjo en 1971 con el trabajo de Stephen Cook *"The complexity of theorem proving procedures"*, en el que quedó explicitada la importancia de la reducibilidad en tiempo polinómico. Esto significa que si tenemos una reducción de tiempo polinómico de un problema a otro, esto asegura que cualquier algoritmo de tiempo polinómico para el segundo problema puede convertirse en un correspondiente algoritmo de tiempo polinómico para el primer problema. Por otra parte Cook trabajó la clase de problemas de decisión NP que pueden resolverse en tiempo polinómico por una computadora no determinista. La mayoría de los problemas aparentemente intratables que se encuentran en la práctica, cuando se escriben como problemas de decisión, pertenecen a esta clase. Y finalmente demostró que un problema particular en NP, llamado el problema de la satisfacibilidad, tiene la propiedad que todo otro problema en NP puede ser reducido polinómicamente a él. Si el problema de la satisfacibilidad puede resolverse con un algoritmo de tiempo polinómico, entonces así también todo problema en NP, y si cualquier problema en NP es intratable, entonces el problema de la satisfacibilidad debe ser también intratable. En este sentido y en la jerga de los computacionalistas, el problema de la satisfacibilidad es el problema "más duro" en NP. A modo de corolario mostró que otros problemas pueden ser tan duros como el anterior entre los miembros de NP. Esta conjetura general sobre la "dureza" de los problemas fue ilustrada a través de muchos ejemplos por Richard Karp desde 1972. Hay numerosos problemas que son tan duros como el problema de la satisfacibilidad. A esta clase de equivalencia entre los problemas mas duros en NP se le ha llamado la clase de problemas NP-completos.

La pregunta importante que se desprende de todo este contexto es si son intratables los problemas NP-completos. Esta pregunta es considerada como una de las más importantes de las ciencias de la computación. Y es poco lo que se ha logrado hasta ahora en torno de una respuesta efectiva a la misma. No sabemos si NP-completitud implica intratabilidad.

Pero esta ausencia de demostraciones generales convincentes no ha impedido ensayos de taxonomías, dado que ya se conocen varios centenares de problemas NP-completos pertenecientes a diversas ramas de investigación. Garey y Jonhson han elaborado una lista realmente impactante de tales problemas organizada bajo las entradas siguientes:
Teoría de grafos, diseño de redes, conjuntos y particiones, almacenaje y recuperación, secuenciación e inventariado, programación matemática, álgebra y teoría de números, juegos y rompecabezas, lógica, lenguajes y autómatas, optimización de programas, y misceláneas. Si tenemos en cuenta el lugar importante que tienen estas especialidades en la sintaxis de numerosos modelos en uso en diferentes disciplinas científicas, es fácil reconocer que la NP-completitud está invadiendo a las disciplinas científicas.

Cuando hablamos antes de una función de complejidad polinómica, hemos usado un concepto de complejidad que ha sido objeto de numerosas interpretaciones. El tratamiento de la complejidad se ha extendido claramente fuera del dominio de las ciencias de la computación; por ello vamos a aclarar su significado dentro de nuestro contexto. En un trabajo ahora clásico de 1965, Hartmanis y Stearns dieron una definición precisa de "clase de complejidad", consistente de todos los problemas resolubles en un número de pasos acotado por una función dada de la longitud de entrada, y usando la técnica de diagonalización probaron muchos resultados importantes acerca de la estructura de las clases de complejidad. A través de investigaciones posteriores se han encontrado numerosas relaciones de inclusión que muestran una jerarquía en la complejidad de los problemas.

LOG-Tiempo $\subseteq$ LOG-espacio $\subseteq$ P-Tiempo $\subseteq$ P-espacio $\subseteq$ EXP-tiempo $\subseteq$
$\subseteq$ EXP-espacio $\subseteq$ 2EXP-tiempo ...

Si además consideramos categorías no deterministas, por ej. NP-tiempo, obtenemos muchas más clases y muchas preguntas interesantes.

En la actualidad un número creciente de problemas que aparecen en aplicaciones no triviales resultan ser NP-completos o peor. En tales casos, a los fines de abordarlos con algún éxito, se recurre a algoritmos de aproximación, a algoritmos probabilistas y a heurísticas. En general se logran por estos medios soluciones que pueden garantizarse como buenas en el promedio. Por razones de tiempo voy a hacer un solo comentario en relación con uno de estos casos: los algoritmos de aproximación de tiempo polinómico y la situación conflictiva en que se encuentra el estado del arte de la construcción de los mismos. Dado que se trata de algoritmos de aproximación, se acepta la presencia de un error relativo, algo así como una pequeña diferencia entre la solución hallada y la solución ideal. Para algunos problemas este error puede hacerse tan pequeño como se desee. Para otros parece haber una cota en la aproximación que no puede superarse. Para otros problemas hasta ahora no se ha hallado un algoritmo con error relativo acotado. Y peor aún, hay problemas para los cuales la existencia de tal algoritmo implicaría que P = NP. Pero esta es la conjetura más importante no resuelta de todo el campo bajo análisis.

Sea cual fuere la relación entre P y NP, está vinculada con la distinción entre la habilidad para resolver un problema y la habilidad para chequear una solución. Una consecuencia sorprendente de la igualdad será que todo problema para el cual sus soluciones son fáciles de chequear, sería también fácil de resolver. Podría erradicarse el problema de la explosión combinatoria. O dicho de otro modo, siempre que un problema tuviera una demostración corta, un procedimiento podría encontrar esa demostración rápidamente. Dado que P es parte de NP, la pregunta es si pueden ser iguales. Desde 1971, esto significa que o todos esos problemas pueden ser resueltos por un procedimiento mecanizado de cómputo o ninguno de ellos puede. La creencia es que P es distinto de NP, significando que los problemas NP-completos son inherentemente intratables, pero nadie lo sabe con seguridad hasta ahora.

Continuando con nuestro periplo por la intratabilidad, recordemos que hemos asociado problemas difíciles de resolver con funciones de complejidad exponencial. Pues bien, hay problemas aún más difíciles. Hay funciones exponenciales dobles, triples,..., hasta Q, del tipo 2 a la 2 a la 2,..., a la 2 a la N, con Q apariciones de 2. Y por si esto fuera poco, hay ejemplos realmente monstruosos, como un formalismo llamado WS1S, que es extremadamente difícil de analizar. Se ha demostrado que no admite un algoritmo exponencial de orden Q, para cualquier Q. Esto significa que para cualquier algoritmo A que determina la verdad de las fórmulas de WS1S (y hay tales algoritmos) y para cualquier número fijo Q, habrá fórmulas de longitud N, para N más y más grandes, que requerirán que A corra para unidades de tiempo más largos que 2 a la 2 a la 2,..., a la 2 a la N, con Q apariciones de 2. Este caso no es sólo intratable en el sentido standard; su desempeño temporal es peor que cualquier exponencial de orden Q, razón por la cual se suele hablar de intratabilidad ilimitada (Harel, 1992).

Una observación que es conveniente hacer aquí: a pesar de lo intrincados de estos problemas, o como a veces se les ha llamado, de ser intrínsecamente difíciles, debemos recordar que hay dos tipos de cuestiones involucradas, cuestiones teóricas y prácticas. En buena medida los resultados asociados a cuestiones teóricas que se han incorporado a la formación básica de cualquier especialista en computación son mayormente producto de la lógica matemática. El núcleo mas fuerte es la indecidibilidad. La intratabilidad está mayormente referida a la imposibilidad práctica de lograr una solución para problemas bien planteados. Ambas tienen fuertes implicaciones para la epistemología. La esfera práctica, que es de la que nos ocupamos aquí, está contribuyendo a cambiar nuestras apreciaciones acerca de las relaciones entre el planteamiento de un problema y su eventual solución. La esfera teórica puede contribuir a cambiar nuestra imagen de los alcances de ciertas metodologías que están fuertemente tramadas entre algoritmos y heurísticas.

Esto sugiere una primera gradación de los problemas en tres grandes categorías: problemas que se pueden resolver, problemas que no se pueden resolver debido al enorme costo de tiempo y espacio de máquina, totalmente fuera del alcance de cualquier procedimiento de cómputo imaginable en la práctica, y problemas que se puede demostrar que son insolubles aún en teoría. El físico matemático Robert Geroch lo expresó a esto en los siguientes términos: "Ya no hay más preguntas para las cuales sabemos que podemos determinar la respuesta, y aquellas para las cuales no sabemos sí o no podemos determinar la respuesta. Ahora hay además preguntas para las cuales sabemos que no podemos determinar la respuesta." Uno podría pensar que se trata de un campo interesante pero muy limitado en su alcance. En ese caso no habría demasiado interés en profundizar los alcances de estos teoremas de limitación y de asociarlos a otros ámbitos más cotidianos de la práctica científica. Es por ello que he seleccionado esta referencia a Geroch.

Remarco este trabajo, cuyo título es "problemas insolubles", porque se trata de una producción dentro del ámbito de la física teórica. Este es en mi opinión el principal indicador de que la intratabilidad ya merece un serio análisis dentro del campo de la filosofía de la ciencia. En el lapso de 60 años hemos sutilmente pasado, por un lado, de la emergencia de la indecidibilidad en el dominio de la lógica matemática, a la demostración de notables teoremas de indecidibilidad dentro de la matemática, como es el caso del décimo problema de Hilbert resuelto por esta vía, y por otro lado, hemos pasado de la emergencia de la complejidad computacional a las primeras incursiones de estas cuestiones en el terreno de disciplinas empíricas con alto grado de abstracción.

Ejemplos como el de Geroch y muchos otros muestran que la indecidibilidad está golpeando las puertas de la actividad científica y que la intratabilidad está dentro del edificio ortodoxo de la ciencia, ocupando un lugar importante aunque haya pasado desapercibida en gran medida. Si bien las analogías en la historia de la ciencia suelen ser peligrosas al momento de extraer inferencias con conclusiones de alcances similares, no puedo evitar asociar esta situación con una anécdota atribuida a Lord Kelvin en relación con la situación de la física hacia fines del siglo pasado. En cierta ocasión declaró que la física le parecía un conjunto perfectamente armónico y en lo esencial acabado; decía que no veía en el horizonte más que dos pequeñas nubes oscuras: el resultado negativo del experimento de Michelson y Morley, y la catástrofe ultravioleta. Sucede que estas dos pequeñas nubes sólo se disiparon con la construcción de la teoría de la relatividad y la teoría de los cuantos. Podríamos estar tentados en pensar que la intratabilidad y la indecidibilidad son sólo dos pequeñas nubes oscuras en el firmamento de los problemas científicos. Pero en realidad ya no parecen ser tan pequeñas, y aunque debamos reconocer que no sabemos hacer todavía buenas predicciones sobre la evolución del concepto de problema ni su destino en relación con las limitaciones internas de los formalismos, los avances realizados en relación con la estructura de numerosos problemas, las clases de equivalencia encontradas, los mecanismos para comparaciones eficaces y el hallazgo de ciertos problemas arquetípicos, obligan ya a una revisión considerable de las relaciones entre metodología y heurísticas e invitan a ejercicios de taxonomías que sirvan de base para una mejor comprensión del concepto.

Nota: Este trabajo forma parte de un proyecto subsidiado por Secyt-U.N.C., CONICOR y FONCYT.

Bibliografía

Davis M.: *Computability and Unsolvability*, Dover P. Co., N. York, 1982.

Garey M., Johnson D.: *Computers and Intractability*, Freeman and Co. N. York, 1979.

Harel D.: *Algorithmics*, 2da ed., Addison Wesley, Reading, Mass., 1992.

Hopcroft J., Ullman J.: *Introduction to automata theory, languages, and computation*, Addison-Wesley, Reading, Mass., 1979.

Lauriere J.: *Problem solving and artificial intelligence*, Prentice Hall, N. York, 1989.

Lewis H., Papadimitriou C.: *Elements of the theory of computation*, 2da Ed. Prentice Hall, N. Jersey, 1998.

Mandrioli D., Ghezzi C.: *Theoretical foundations of computer science*, J. Wiley and Sons, N. York, 1987.

Motwani R., Raghavan P.: *Randomized algorithms*, Cambridge UP 1995.

Rogers H., Jr.: *Theory of recursive functions and effective computability*, The MIT Press, 1992.

Taylor R.G.: *Models of Computation and formal languages*, Oxford U.P. 1998.

Van Leewen J., (Ed.): *Handbook of theoretical computer science, Vol. A. Algorithms and complexity*, The MIT Press, Cambridge, 1990.