

## EDUCACIÓN EN INFORMÁTICA



# Computing Curricula en América Latina

Fotografía gentileza Johan Fabry.

En este artículo presentamos la experiencia de haber creado un modelo curricular basado en la Computing Curricula [Chang et al., 2001], que luego fue extendido y automatizado para soportar adaptaciones en cualquier país de la región o del mundo, en español o en otros idiomas, respetando las peculiaridades de cada lugar pero siguiendo las recomendaciones internacionales.

## INTRODUCCIÓN

Todos los que hemos tenido la suerte de vivir en esta época podemos observar el avance de la ciencia y tecnología con una mezcla de sentimientos que van desde la admiración, por la cantidad de novedades que aparecen todos los días, pasando por el orgullo de ser protagonistas y llegando a una inevitable sensación de susto por la avalancha acelerada de información que se genera a diario y que parece venirse encima de todos nosotros.

Existen varios factores que han provocado esta aceleración entre los cuales debemos obligatoriamente citar la aparición del

computador, la Ciencia de la Computación e Internet. El computador, que fue creado con fines bélicos, pronto fue participando en diversas áreas del conocimiento debido a su poder de resolver problemas que el ser humano demoraba mucho en hacer manualmente. La Ciencia de la Computación, que nació con participación de diversas áreas profesionales, no demoró mucho en ganarse un espacio y hacer sentir que es una disciplina que vino para quedarse y cambiar el mundo tal y como era visto hace seis décadas. Internet, a su vez, se ha convertido en el medio de comunicación de las nuevas generaciones por excelencia. Hoy en día hablamos de los nativos digitales para referirnos a aquellos niños que nacieron



### **Ernesto Cuadros-Vargas**

*Doctor en Ciencia de la Computación, ICMC, Universidad de Sao Paulo (2004). Fundador y miembro de la Sociedad de Computación Peruana. Secretario Ejecutivo del Centro Latinoamericano de Informática (CLEI) (2009-2010) y miembro del Educational Activities Board de IEEE (2006-2009).*

en medio de este mundo digital. Para ellos hablar de tecnología es tan natural como respirar.

Estos tres inventos del ser humano han provocado un círculo virtuoso donde se genera una gran cantidad de información que afecta a todas las áreas del conocimiento, entre las cuales también se encuentra la propia Ciencia de la Computación.

En este artículo presentaremos una propuesta para poder acompañar este avance en el ámbito latinoamericano, siguiendo la filosofía japonesa de imitar, igualar y superar.

## SOBRE LA CARRERA EN AMÉRICA LATINA

El área de computación llegó a América Latina como una novedad importada necesaria. La mayoría de las primeras carreras de este tipo en nuestra región se remontan a la década de los '70 y llegan con diversas tendencias y nombres como Ciencia de la Computación (influencia estadounidense), Informática (influencia europea) e Ingeniería de Sistemas, cuyo origen no tiene un claro consenso pero parece referirse también al área de computación.

La Ingeniería de Sistemas es, probablemente, la más confusa por la falta de consenso en definir de forma clara su identidad. Existen mezclas de nombres generadas por diversas influencias. En los EE.UU. se encuentran instituciones como INCOSE (International Council on Systems Engineering, <http://www.incose.org>) y el ISSS (International Society for the Systems Sciences, <http://iss.org>), además de las carreras denominadas System Engineering no vinculadas a computación, como la de Massachusetts Institute of Technology (MIT) que presenta su Engineering Systems Division (<http://esd.mit.edu>). Un reflejo de la falta de consenso en relación a este tema es que la agencia acreditadora ABET (<http://www.abet.org>) clasifica a la carrera Systems Engineering en el rubro de Others, mientras INCOSE trabaja para que tenga su propio espacio.

Por si fuera poco, existen combinaciones de nombres como Ingeniería de Computación y

Sistemas, Ingeniería Informática y Sistemas, etc. los cuales, en muchos casos, tienen el mismo nombre pero se refieren a una carrera distinta. Asimismo hay países donde la carrera se da como licenciatura y en otros como Ingeniería. Una opinión personal del autor es que toda esta confusión es sólo una señal para que continuemos trabajando por aumentar el espacio de la disciplina Computación, frente a otras carreras más antiguas y tradicionales.

Los responsables de ayudar a cambiar esta situación somos, en primer lugar, aquellos involucrados directamente en la formación de profesionales. Necesitamos acortar la distancia en relación a otras regiones y ese es el espíritu de esta propuesta.

## SOBRE LA PROPUESTA INTERNACIONAL

A nivel mundial existen carreras que podríamos denominar "tradicionales" como Ingeniería Civil o Medicina, sobre las que difícilmente se tendrían discrepancias acerca de los cursos básicos que hay que dictar para un estudiante universitario. Esto ocurre debido a la antigüedad de esas carreras y a la madurez que ya tienen, así como a una cierta estabilidad y consenso en sus contenidos.

En el caso de la Ciencia de la Computación y todo lo referente al hardware estamos hablando de disciplinas que tienen un poco más de medio siglo de existencia pero han tenido un crecimiento de tipo exponencial. Ambas áreas, hardware y software, se han acelerado mutuamente. Por esta razón, desde hace varias décadas los profesionales de la Ciencia de la Computación, agrupados en instituciones como la ACM y la Sociedad de Computación del IEEE, trabajan arduamente en darles un contenido estándar a las carreras del área.

Los primeros esfuerzos datan de la década de los '60, en las que la ACM publicó el documento [ACM65], que fueron continuados en [COSINE67] y [ACM68]. El crecimiento del área en los siguientes años provocó que esta última recomendación

quedase rápidamente obsoleta generando entonces las recomendaciones [EC77, ACM78]. Éstas son actualizadas cada diez años aproximadamente, llegando a [Tuc91] y [CS2001]. La previsión es que deberíamos tener una siguiente actualización en 2011. Pero la velocidad con que las cosas cambian ha obligado a acortar el tiempo para el siguiente reporte y ya vemos actualizaciones en [CC2005] y especialmente en [CS2008].

Basados en todos estos cambios tenemos un gran desafío por delante: ¿Cómo acortar la distancia en nuestra región considerando que muchas veces tenemos menos recursos? La respuesta está en nuestra propia área: utilizar de forma más eficiente la propia Ciencia de la Computación, para poder generar nuevas curricula que respeten las propuestas internacionales, pero que al mismo tiempo nos permitan adaptarlas a las exigencias de cada una de nuestras instituciones.

Los documentos [CS2001, CS2008] presentan los siguientes puntos positivos:

- Presentan un contenido muy detallado del Cuerpo de Conocimiento recomendado para una carrera en Ciencia de la Computación.
- Presentan recomendaciones generales sobre la secuencia de cursos básicos de la carrera.
- El equipo involucrado dispone de los recursos adecuados para generar una recomendación de esta magnitud.
- Si bien es cierto que ambos documentos tienen una fuerte participación estadounidense, también es cierto que no resulta difícil adaptarlos a otras regiones del mundo debido a que la Computación es única y los conceptos dictados son similares.
- El documento [CS2008] hace un especial énfasis en algunos cambios como agregar contenido referente a Seguridad y Paralelismo de forma transversal en toda la carrera y no sólo como cursos aislados. Esto significa que, por ejemplo, conceptos de paralelismo deben ser tocados con diferentes niveles de profundidad, desde que el alumno entra



al primer semestre y presente de forma transversal en cursos como algoritmos, programación, ingeniería de software, base de datos, entre otros.

A pesar de ser un excelente documento de referencia para carreras de pregrado en Ciencia de la Computación, existen varios aspectos que aún no han sido cubiertos por la forma de trabajo que se tuvo. Entre estos puntos podemos mencionar:

- Para la acreditación es necesario relacionar cada curso con un grupo de resultados esperados. Este documento menciona el punto pero no muestra algún tipo de tabla donde sea posible observar esta relación curso por curso.
- La acreditación también solicita que sea posible analizar: ¿Cuántas horas y en qué curso se cubre cada uno de los tópicos del Cuerpo de Conocimiento? sin tener que analizar manualmente cada uno de los cursos de la carrera. Esta información representa una tabla de millares de celdas que contiene las más de cien unidades del Cuerpo de Conocimiento vs todos los cursos de la carrera. Este tipo de información existe pero habría que recurrir al análisis individual y manual de cada curso, a fin de detectar si cumplimos con el mínimo de horas para cada tópico.

- No ofrece una distribución de créditos por áreas que ayude a entender el espectro de una determinada carrera, para poder compararlo con las recomendaciones de máximos y mínimos de créditos por áreas que pueden ser observadas en [CC2005].
- Nuestras universidades suelen ofrecer la malla curricular en forma de cursos clasificados por semestres. Sin embargo, la visión gráfica de los cursos interconectados a través de prerrequisitos ayuda mucho a que el alumno entienda qué líneas forman el camino crítico en la carrera y a cuáles hay que prestarle mayor atención. La consecuencia de esto es que podemos reducir la deserción a través de una mayor cantidad de información de carácter preventivo.
- La propuesta de la [CS2008] tiene un excelente contenido. Pero al mismo tiempo es un documento grande escrito manualmente en LaTeX. Considerando la importancia del documento, y su impacto en nuestra sociedad, lo ideal sería poder generar toda la información antes mencionada de forma automática a partir de la información contenida en el cuerpo de conocimiento y en los cursos.

## SOBRE NUESTRA PROPUESTA

Todos estos cambios desafían frontalmente la forma en que planteamos y enseñamos en una carrera de Ciencia de la Computación. Considerando las ventajas en el documento [CS2008] y sus predecesores, la idea fue crear un programa que pudiese generar, de forma muy rápida, toda una malla curricular con las siguientes características:

- Contenido sólido coherente con el cuerpo de conocimiento de Ciencia de la Computación.
- Actualización respecto a los cambios planteados en [CS2008].
- Visión gráfica de la malla curricular para poder detectar cuellos de botella, caminos críticos, etc., como por ejemplo en la Figura 1. Este grafo es generado utilizando el programa.

Semestre	CS	HU	ET	CB	Total
Primer	8	9		5	22
Segundo	12	5		5	22
Tercero	10	4		8	22
Cuarto	8			14	22
Quinto	15	3		4	22
Sexto	22				22
Séptimo	13	3		6	22
Octavo	16	3	3		22
Noveno	14	3	2		19
Décimo	11	9	2		22
Total	129	39	7	42	

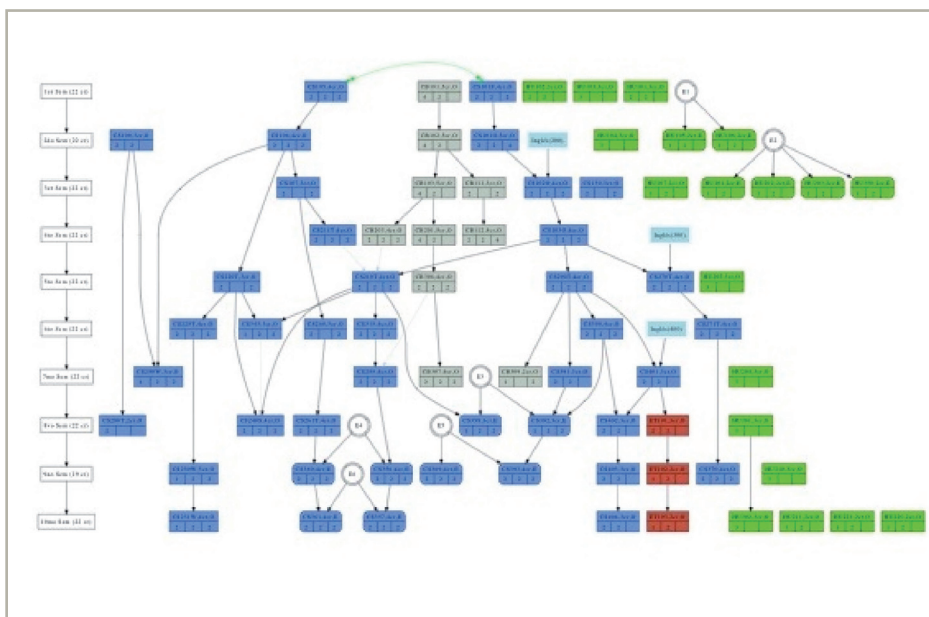


Figura 1: Visualización malla curricular.

- Distribuciones de cursos por áreas (Figura 2) y por niveles (Figura 3).
- Generar los gráficos comparativos del espectro de una carrera en relación a las propuestas internacionales como Computer Science (CS) (Figura 4), Computer Engineering (CE) (Figura 5), Information Systems (IS) (Figura 6), Software Engineering (SE) (Figura 7), o Information Technology (IT).
- Distribución tabular de cursos por áreas como Ciencia de la Computación (CS), Humanidades (HU), formación de Empresas de Base Tecnológica (ET) y Ciencias Básicas (CB).

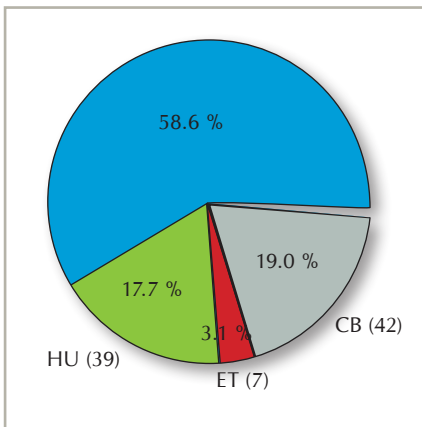


Figura 2: Distribución de créditos por área.

- Distribución de tópicos del Cuerpo de Conocimiento por curso para acreditación y control interno del programa profesional. Esta tabla no puede ser mostrada debido a que supera las nueve mil celdas.
- El docente no entrega el sílabo de cada curso a la jefatura del departamento. Por el contrario, el programa de generación de esta malla también genera los sílabos. Y los cambios en el contenido deben previamente ser coordinados con los docentes del área considerando la visión gráfica de la malla presentada en la Figura 1.
- Para un seguimiento efectivo del avance y desempeño docente, el programa también crea páginas Web para cada curso, de tal forma que el avance es controlado de manera efectiva y los problemas detectados se pueden corregir con rapidez. Esta parte aún no está concluida en 100 por ciento.

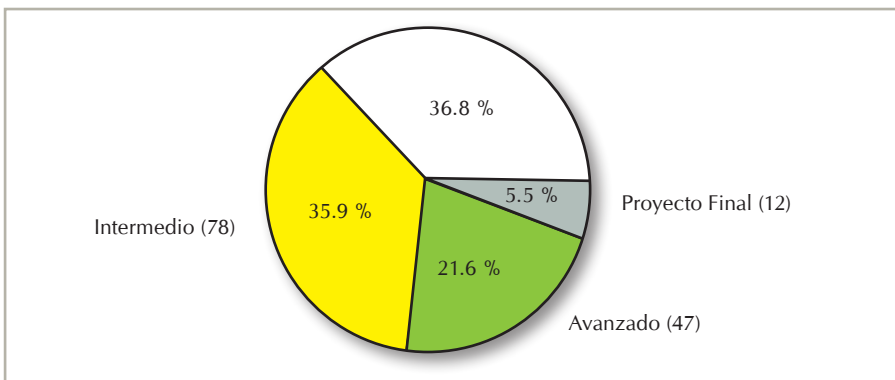


Figura 3: Distribución de créditos por nivel.

- El programa también genera un libro de sílabos, uno con las descripciones cortas de cada curso y un tercero con el listado bibliográfico por curso. Todo esto permite coordinar con la biblioteca una sola vez y para toda la carrera en caso de no existir un libro disponible.

En base a esta propuesta es posible acortar mucho el tiempo dedicado a este tipo de tareas de gestión de la carrera y aumentar la calidad y el control de lo que se dicta, así como detectar problemas antes de que sucedan.

## EXTENDIENDO EL MODELO PARA CUALQUIER OTRA CARRERA

Gracias a la abstracción inherente a nuestra formación en Ciencia de la Computación es fácil entender que el programa sería fácilmente adaptable a diversas otras carreras. En este sentido, para extender este modelo necesitamos seguir los siguientes pasos:

- Definir en forma de macros LaTeX el Cuerpo de Conocimiento del área.
- Lista de resultados Outcomes de acuerdo a la acreditación del área.
- Definir el contenido de cada curso en función del cuerpo del conocimiento respectivo. Es aquí donde relacionaremos cada curso con un conjunto de resultados esperados.
- Redactar la información básica de la carrera tal como: visión, misión, títulos otorgados, relevancia en la sociedad, etc.

Probablemente el mayor problema sería pedir que otras carreras llenasen el contenido de los cursos en LaTeX. Para resolver ese problema se está trabajando en una interfase vía Web que facilite la interacción de colegas de otras carreras.

## CONCLUSIONES Y TRABAJOS FUTUROS

Este documento presenta una propuesta para generar y mantener actualizada una malla curricular de acuerdo a cualquier propuesta internacional.

La cantidad de tiempo invertido en llenar la información para una nueva carrera reduce al orden de algunos pocos días el trabajo que antes podía tomar fácilmente meses.

El control que es posible tener sobre la carrera, basado en los diversos indicadores, abre una gran cantidad de posibilidades y nos ayuda a detectar tempranamente múltiples problemas y mejorar la calidad de la educación ofrecida.

Una característica que se adicionará es que, en base al seguimiento por curso, será posible recolectar los datos con el objetivo de generar estadísticas históricas y analizar el desempeño académico de un docente en todos los cursos que dicta.

Asimismo será posible comparar todos los cursos que dicta un único profesor (a), a fin de analizar en cuáles se desempeña mejor y detectar posibles puntos débiles para reforzar.

Teniendo tabulados los resultados esperados por cada curso es posible comparar de manera efectiva si el (la) docente está cumpliendo con los mismos, y generar indicadores que retroalimenten a los propios docentes y permitan que él/ella mejore su cátedra.

La visión gráfica de la malla permite detectar posibles incoherencias en la malla curricular, lo que es mucho más difícil de hacer sobre una visión tabular tradicional de cursos distribuidos por semestre. BITS

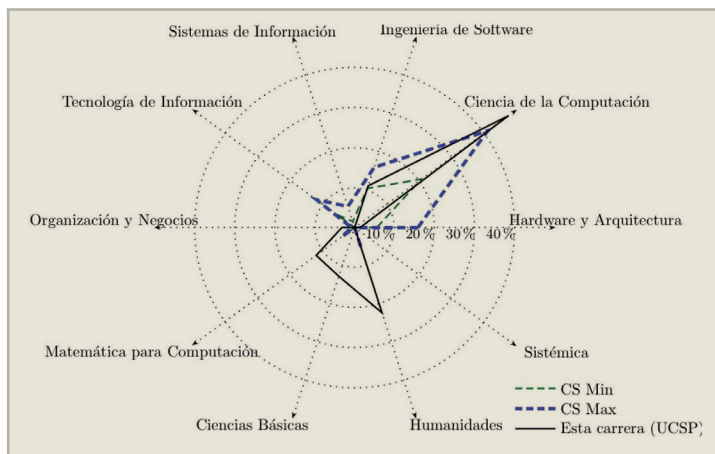


Figura 4: Comparación de la malla generada contra la recomendación de CS.

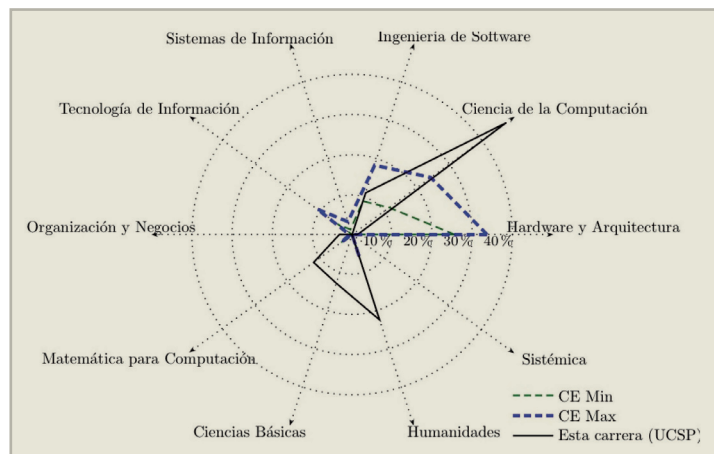


Figura 5: Comparación de la malla generada contra la recomendación de CE.

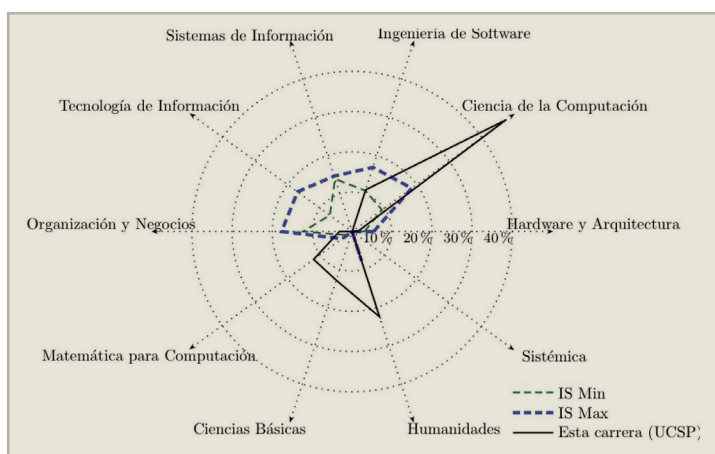


Figura 6: Comparación de la malla generada contra la recomendación de IS.

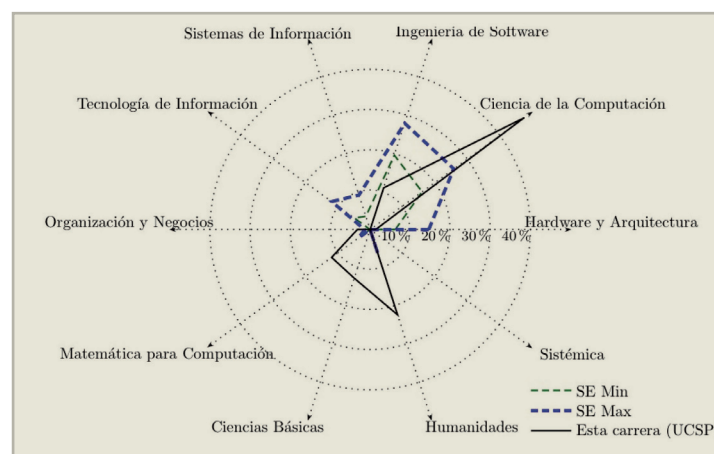


Figura 7: Comparación de la malla generada contra la recomendación de SE.

## REFERENCIAS

[ACM65] ACM (1965). Curriculum Committee on Computer Science. An Undergraduate Program in Computer Science. Communications of the ACM, 8(9):543–552.

[ACM68] ACM (1968). Curriculum Committee on Computer Science. Curriculum '68: Recommendations for the Undergraduate Program in Computer Science. Communications of the ACM, 11(3):151–197.

[ACM78] ACM (1978). Curriculum Committee on Computer Science. Curriculum '78: Recommendations for the Undergraduate Program in Computer Science. Communications of the ACM, 22(3):147–166.

[CS2008] Cassel, L., Clements, A., Davies, G., Guzdial, M., McCauley, R., McGettrick, A., Roberts, E., Sloan, B., Snyder, L., Tymann, P., and Weide, B. W. (2008). Computer science curriculum 2008: An interim revision of cs2001. Technical report, ACM/IEEE-CS.

[CS2001] Chang, C., Denning, P. J., II, J. H. C., Engel, G., Sloan, R., Carver, D., Eckhouse, R., King, W., Lau, F., Mengel, S., Srimani, P., Roberts, E., Shackelford, R., Austing, R., Cover, C. F., Davies, G., McGettrick, A., Schneider, G. M., and Wolz, U. (2001). Computing Curricula 2001 Computer Science. Technical report, ACM/IEEE,

<http://www.computer.org/education/cc2001/steelman/cc2001/index.htm>. Last visited March 2004.

[COSINE67] COSINE Committee (1967). Computer Science in Electrical Engineering. Commission on Engineering Education, Washington, DC.

[EC77] IEEE-CS (1977). A Curriculum in Computer Science and Engineering. Technical Report, IEEE-CS. Education Committee of the IEEE-CS. Publication EHO119-8.

[CC2005] Shakelford, R., Cross, J. H., Davies, G., Impagliazzo, J., Kamali, R., LeBlanc, R., Lunt, B., McGettrick, A., Sloan, R., and Topi, H. (2005). Computing curricula 2005. Technical report, ACM/IEEE, <http://www.acm.org/education>.

[Tuc1991] Tucker, A. B., Barnes, B. H., Aiken, R. M., Barker, K., Bruce, K. B., Cain, J. T., Conry, S. E., Engel, G. L., Epstein, R. G., Lidtke, D. K., Mulder, M. C., Rogers, J. B., Spafford, E. H., and Turner, A. J. (1991). Computing curricula '91. Technical report, Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers.