CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL INSTITUTO POLITÉCNICO NACIONAL

Programas de Posgrado en Ciencias de la Computación

El Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional a través del Departamento de Computación y del Laboratorio de Tecnologías de Información ofrece estudios de posgrado a nivel maestría y doctorado en la especialidad de Ciencias de la Computación. Actualmente se admiten anualmente en las dos sedes alrededor de 42 estudiantes para su programa de maestría y 10 estudiantes en el programa de doctorado. El programa de posgrado atiende anualmente un promedio de 90 estudiantes. Los estudiantes de nacionalidad mexicana no pagan colegiatura.

El Programa de Posgrado en Ciencias de la Computación tiene adscritos a 27 investigadores de tiempo completo con el grado de doctor, 15 de ellos adscritos oficialmente al Departamento de Computación ubicado en la Sede Zacatenco y 12 investigadores adscritos al Laboratorio de Tecnologías de Información ubicado en la Unidad Tamaulipas. Además en cada periodo académico se cuenta con la colaboración de profesores asociados al programa y con profesores visitantes y/o en estancia posdoctoral. Las líneas de investigación que se cultivan en el programa se encuentran las siguientes:

1. Fundamentos de la Computación e Inteligencia Artificial
2. Bases de Datos y Sistemas de Información
3. Programación de Sistemas, Sistemas Operativos, Sistemas Distribuidos y Sistemas de Tiempo Real
4. Criptografía, Arquitectura de Computadoras y Hardware Reconfigurable
5. Graficación, Visualización y Procesamiento de Imágenes
6. Tecnologías de Información
7. Ingeniería Computacional: Robótica, Sistemas Embebidos, Cómputo Móvil y Cómputo Reconfigurable

El Departamento de Computación y el Laboratorio de Tecnologías de Información cuentan con varios laboratorios con sistemas heterogéneos que permiten a investigadores y estudiantes desarrollar los proyectos de cursos y de tesis. Entre las plataformas utilizadas en los laboratorios se encuentran mallas (grids) de servidores de trabajo y computadoras personales con varios tipos de sistemas operativos. La información más relevante de nuestros programas puede consultarse en las páginas web: www.cs.cinvestav.mx y www.tamps.cinvestav.mx o bien puede pedirse a:

Coordinación Académica
Departamento de Computación
CINVESTAV-IPN
Avenida Instituto Politécnico Nacional 2508,
07360 México, D. F.,
México,
Teléfono (55) 5061-3758
Fax (55) 5061-3757
coordinacion@cs.cinvestav.mx

Laboratorio de Tecnologías de Información
CINVESTAV-Tamaulipas
Parque Científico y Tecnológico Tecnotam
Km. 5.5 Carretera a Soto La Marina,
Ciudad Victoria, Tamaulipas
Teléfono (834) 107-0220
Fax (834) 107-0224
admin@tamps.cinvestav.mx

1. Programa de Doctorado en Ciencias de la Compuación

El programa de doctorado tiene como objetivo preparar especialistas con un conocimiento profundo y amplio de la disciplina computacional y con la capacidad de generar conocimiento en la misma. Tiene una duración promedio de 4 años, y puede iniciar en el mes de enero, mayo o septiembre de cada año.

El programa está dirigido a personas que han obtenido el grado de maestro en ciencias en la especialidad de Computación, o un área afín. Está enfocado a desarrollar, bajo la supervisión de un director de tesis, un proyecto de investigación que contribuya al desarrollo del conocimiento en las ciencias de la computación.

2. Proceso y requisitos de admisión

El proceso de admisión para ingreso al programa de doctorado considera tres partes fundamentales:

- Presentación del examen teórico de ingreso al posgrado equivalente al que se aplica a los aspirantes al programa de maestría.
- Presentación de un examen oral de conocimientos generales en dos áreas fundamentales de la propuesta de investigación.
- Presentación de una propuesta de investigación, también denominada propuesta doctoral, avalada por un profesor adscrito al programa.

A continuación se presentan las consideraciones para cada una de las tres partes del proceso de admisión.

El aspirante deberá solicitar que un profesor del Programa Institucional de Computación que acepte participar como su asesor de estudios. Con este propósito y a petición del aspirante, el coordinador académico del programa calendarizará una entrevista con cada uno, o sólo algunos, de los profesores del programa. Los investigadores participantes en el programa son responsables de dirigir los trabajos de tesis. Sin embargo, para fomentar la multidisciplinariedad o abordar temas de aplicación de la computación a la ciencia o la tecnología, investigadores de otros departamentos podrán participar como codirectores de tesis. En casos excepcionales, profesores de otras instituciones, previamente aprobados por el Colegio de Profesores, con la debida justificación, podrán participar como codirectores de tesis de doctorado.

El proceso de admisión inicia cuando el director de tesis potencial solicita por escrito al Coordinador Académico la evaluación de los conocimientos del aspirante, entregando su currículum vitae y solicitud de ingreso al programa de doctorado. La solicitud de ingreso deberá describir brevemente los motivos del aspirante para realizar un doctorado en Computación. Asimismo, el asesor deberá establecer las dos áreas principales en las cuales se desarrollará la tesis.

El aspirante presentará entonces el examen de ingreso a la maestría, debiendo obtener

como mínimo la calificación promedio de la generación más reciente de maestría que haya realizado dicho examen. De no alcanzar la calificación mínima aceptable, no se autorizará que el aspirante ingrese al programa de doctorado. Todo estudiante que no logre aprobar el examen de admisión de maestría en su primer intento, no podrá volver a tomar dicho examen sino hasta que haya transcurrido un período de al menos 12 meses. Se permitirán un máximo de 2 intentos para aprobar este examen.

Todos los aspirantes a ingresar al doctorado, sin excepción, deberán aprobar un examen oral en dos áreas principales que cubra su propuesta de tesis doctoral. Se designará a dos profesores del programa para aplicar los exámenes orales. Dicha designación la realizará el Coordinador Académico, en común acuerdo con el asesor y con los profesores involucrados. Los dos profesores del programa seleccionados para realizar la evaluación oral del sustentante, deberán reportar el resultado de la evaluación directamente al Coordinador Académico. Bajo ninguna circunstancia, el asesor (o asesores) de tesis del aspirante a ingresar al doctorado, podrá(n) aplicar dicho examen.

En caso de que el sustentante lo solicite, los examinadores deberán proporcionar la lista de temas y/o sugerir bibliografía que incluya referencias relevantes a los temas a ser evaluados. El resultado del examen oral es “Aprobado” o “Reprobado”. La aprobación del sustentante en el examen oral deberá ser unánime de parte de los dos examinadores. El caso contrario implicará la no aceptación del candidato.

Sólo tras haber aprobado el examen oral, y, en su caso, el examen general de conocimientos, la propuesta doctoral será evaluada por un comité de admisión conformado por cinco investigadores, los cuales serán designados por el Coordinador Académico, en común acuerdo con el asesor. El comité deberá estar conformado por profesores del programa y el asesor(e)s formará(n) parte de él.

En caso de que se falle en cualquiera de las tres partes del proceso de admisión, la Coordinación Académica elaborará un acta oficial en la que conste que el aspirante ha sido rechazado, indicando lo siguiente:

- Título de la propuesta presentada
- Fecha
- Fase del proceso de admisión
- En su caso, nombres de los integrantes del comité de evaluación
- Descripción de las causas del rechazo
- En su caso, copia de la propuesta presentada

El aspirante admitido deberá entregar al Departamento de Servicios Escolares del Cinvestav los siguientes documentos:

- Solicitud de Admisión al Cinvestav.
- Constancia de examen TOEFL (o equivalente) con al menos 500 puntos.
- Certificado completo de estudios profesionales del ciclo de licenciatura o ingeniería con el promedio general de aprovechamiento.
- Certificado completo de estudios de maestría (en su caso) con el promedio general de aprovechamiento.
- Copia del acta del examen final o de su título de maestría.
- Copias de constancias o certificados de otros estudios cursados.
- Cuatro fotografías tamaño infantil.

- Dos cartas de recomendación (original y copia) de profesores o investigadores que lo conozcan.
- Dos copias del acta de nacimiento.
- Copias de constancias o certificados de los seminarios, cursos, congresos y conferencias en los que haya participado.
- Currículum Vitae único (CVU) de acuerdo al formato establecido por el CONACyT
- Copia de la Clave Única de Registro de Población (CURP).
- Copia de constancias o certificados de los seminarios, congresos y/o conferencias en los que ha participado.

3. Programa de estudios

El programa doctoral de un estudiante varía de acuerdo al proyecto de investigación, a los intereses y experiencia del estudiante, a su supervisor de estudios y a las recomendaciones del comité de admisión. Sin embargo, un programa típico incluye las siguientes fases, desarrolladas en durante cuatro años:

- Preparación mediante cursos. Se deben tomar mínimo cuatro cursos de nivel de especialidad.
- A lo más la mitad de los cursos, previa autorización del comité de admisión, pueden cursarse en otros programas del Cinvestav o fuera del Cinvestav.
- Cursar los Seminarios de Tesis de Doctorado una vez por año, durante el cuatrimestre septiembre-diciembre.
- Obtención de resultados preliminares.
- Presentación de un examen predoctoral (no después de 9 cuatrimestres de haber iniciado el programa).
- Obtención de resultados definitivos.
- Escritura de la tesis.
- Presentación del examen doctoral (defensa pública de la tesis).

El desarrollo del trabajo de investigación podrá llevarse a cabo por medio de estancias industriales o en otras instituciones, nacionales o extranjeras, si, de acuerdo con la evaluación correspondiente del Colegio de Profesores, se considera necesario.

4. Requisitos de permanencia

El periodo mínimo de residencia es de dos años académicos dedicados de tiempo completo a la investigación que conducirá a la elaboración de la tesis doctoral. Se estima que, en general, los candidatos requieren de cuatro años para completar su preparación y su proyecto de tesis. El candidato deberá reportar periódicamente sus avances a la comunidad académica del Departamento de Computación o del Laboratorio de Tecnologías de Información mediante reportes y a través de los Seminarios de Tesis de Doctorado.

5. Requisitos para la obtención del grado académico

Antes de solicitar la presentación de la tesis, el estudiante deberá sustentar un examen predoctoral que versará sobre tópicos fundamentales de la computación y el área principal que el alumno elija. El candidato deberá presentar una tesis desarrollada bajo

la supervisión de su asesor de estudios. Asimismo, el candidato deberá presentar las publicaciones que acrediten la originalidad de su trabajo lo cual puede hacerse de la siguiente manera:

1. Al menos dos publicaciones en congresos internacionales arbitrados y de prestigio en el área de especialización, o
2. Un artículo aceptado o publicado en una revista periódica con arbitraje estricto y listada en el Science Citation Index.

Una vez aceptada la tesis por el Comité de Graduación, el candidato presentará un examen final ante el comité y el asesor de estudios sobre el contenido de su tesis. Si la defensa es exitosa de acuerdo con los criterios del Comité de Graduación, el Cinvestav otorgará al estudiante el grado de Doctor en Ciencias en la especialidad de Computación.

6. Líneas de generación y/o aplicación del conocimiento

A. LGAC 1: Fundamentos de la Computación e Inteligencia Artificial (CA-FCIA).

Comprende los fundamentos teóricos de la metodología de la computación y los modelos de razonamiento usados para el desarrollo de sistemas inteligentes. El área de Computación Evolutiva, la cual se refiere al uso de sistemas bio-inspirados para la solución de problemas computacionales difíciles, se considera también parte de esta línea, si bien su orientación ha sido más hacia la optimización que hacia la inteligencia artificial.

B. LGAC 2: Bases de Datos y Sistemas de Información (CA-BDSI):

Comprende el desarrollo e integración de sistemas de software basado en la descomposición funcional y el desarrollo de herramientas de software. Dentro de esta área, y con una fuerte componente tecnológica, se considera el desarrollo de aplicaciones, protocolos y herramientas para sistemas web. Esta área ha tenido gran impacto dentro del Cinvestav (p.ej., a través del desarrollo del sistema de control escolar que se usa en el Centro).

C. LGAC 3: Programación de Sistemas, Sistemas Operativos, Sistemas Distribuidos y Sistemas de Tiempo Real (CA-PSDT):

Comprende el diseño y desarrollo de software para administrar los recursos de sistemas de cómputo y desarrollar software de aplicación. Es de destacar en esta área la importancia cada vez mayor de los mecanismos de seguridad informáticos a nivel de computadoras y redes de computadoras, los cuales requieren tomar como base estrategias generales para integrar soluciones ad hoc para un problema específico.

D. LGAC 4: Criptografía, Arquitectura de Computadoras y Hardware Reconfigurable (CA-ACHR):

Comprende el estudio, análisis y diseño en hardware reconfigurable de algoritmos criptográficos, algoritmos para la comprensión/descomprensión de información y algoritmos aplicados a visión por computadora. Como parte de esta línea de investigación se encuentra el cómputo reconfigurable, el cual se refiere al uso de dispositivos de hardware reconfigurable que permiten construir soluciones hardware/software a problemas computacionales altamente

demandantes.

E. LGAC 5: Graficación, Visualización y Procesamiento de Imágenes (CA-GVPI):

Comprende la integración de herramientas computacionales diversas para resolver problemas de visión computacional, procesamiento de señales/video y visualización. Un área dominante es esta disciplina es la de sistemas empotrados, esto es, el desarrollo de dispositivos que tienen componentes de software empotrados en hardware. Por ejemplo, dispositivos tales como teléfonos celulares, agendas personales digitales, reproductores de audio digital, grabadoras de video digital, sistemas de alarma, máquinas de rayos X y herramientas médicas láser. Todas ellas requieren de integración de hardware y software empotrado.

F. LGAC 6: Tecnologías de Información (CA-TI)

Comprende un conocimiento técnico amplio de diversas áreas de Ciencias de la Computación enfocadas a la realización de investigación aplicada. Combina aspectos de las líneas de investigación anteriores haciendo énfasis en la resolución de problemas prácticos de Ingeniería de Software, Ingeniería Computacional, Sistemas de Información, Bases de Datos, Redes de Computadoras, Procesamiento de Información y Seguridad Computacional. La línea proporciona los conocimientos ingenieriles de Ciencias de la Computación necesarios para la aplicación y manejo de las nuevas Tecnologías de Información en diversos contextos de la vida moderna.

G. LGAC 7: Ingeniería Computacional: Robótica, Sistemas Embebidos y Cómputo Reconfigurable (CA-IC)

Comprende la integración de herramientas computacionales diversas para resolver problemas de automatización, robótica. Un área dominante en esta disciplina es la de sistemas empotrados, esto es, el desarrollo de dispositivos que tienen componentes de software empotrados en hardware. Por ejemplo, dispositivos tales como teléfonos celulares, agendas personales digitales, reproductoras de audio digital, grabadoras de video digital, sistemas de alarma, máquinas de rayos X, herramientas médicas láser. Todas ellas requieren integración de hardware y software empotrado. También considera el estudio, análisis y diseño de prototipos en hardware reconfigurable de algoritmos criptográficos, algoritmos para compresión/descompresión de información y algoritmos aplicados a visión por computadora. Como parte de esta línea de investigación se encuentra el cómputo reconfigurable, el cual se refiere al uso de dispositivos de hardware reconfigurable que permiten construir soluciones hardware/software a problemas computacionales altamente demandantes.

7. Cursos

Los cursos a acreditar son los acordados por el comité de admisión conformado para el estudiante, los cuales dependerán del tema de tesis y de la formación académica previa del estudiante. Estos pueden variar entre 4 y 6 cursos a lo largo del doctorado. La elección de los cursos busca para el estudiante una formación con los conocimientos esenciales de la Computación, con la mayor amplitud posible en las diferentes áreas de la Computación, y con la mayor profundidad posible en las áreas relacionadas con el tema de tesis del estudiante.

Los cursos están agrupados bajo un núcleo y siete áreas de especialidad. Todos los cursos del núcleo se ofrecen todos por lo menos una vez al año en los primeros dos cuatrimestres del año lectivo. Los cursos en las áreas de especialidad se ofrecen tomando en cuenta la demanda y la planta de profesores. Los cursos de cada área de especialidad se dividen en formativos y de especialización. Los formativos proporcionan amplitud de conocimientos, y los de especialización proporcionan profundidad en alguna línea de investigación.

7.1 Cursos del Núcleo

El núcleo comprende los conocimientos básicos que cualquier egresado del programa de Doctorado en Ciencias de la Computación debe saber. Los cursos del núcleo son ocho (8):

- Matemáticas Discretas
- Análisis y Diseño de Algoritmos
- Programación Orientada a Objetos
- Ingeniería de Software
- Sistemas Operativos
- Arquitectura de Computadoras
- Bases de Datos
- Lenguajes de Programación

7.2 Cursos Formativos y de Especialización

Los cursos se toman de las siguientes áreas de especialización. Los cursos formativos y de especialización se presentan por línea de investigación en la Tabla I.

8. Trabajo de tesis

El trabajo de tesis puede iniciarse desde el primer año del doctorado, dependerá de la disponibilidad de cursos que el alumno debe cursar. La carga de trabajo entre los cursos y el trabajo de tesis debe ser equilibrada.

En el primer cuatrimestre de cada ciclo académico el estudiante debe cursar “Seminario de Doctorado”, que se acredita con la misma escala de calificación con que se acreditan los cursos (esta escala es descrita más adelante). Durante este tiempo el alumno podrá hacer estancias académicas o de investigación en otra Institución de investigación.

Tabla I. Mapa curricular de los cursos formativos y de especialización por línea de investigación.

Nivel	*Fundamentos Teóricos de la Computación e Inteligencia Artificial*	*Bases de Datos y Sistemas de Información*	*Programación de Sistemas, Sistemas Distribuidos y Sistemas de Tiempo Real*	*Criptografía, Arquitectura de Computadoras y Hardware Reconfigurable*	*Graficación, Visualización y Procesamiento de Imágenes*	*Tecnologías de Información*	*Ingeniería Computacional: Robótica, Sistemas Embebidos y Cómputo Reconfigurable*
Formativos	Inteligencia Artificial	Lógica y Bases de Datos	Sistemas de Tiempo Real	Aritmética Computacional	Graficación	Inteligencia Artificial	Redes de Computadoras
	Introducción a la Computación Evolutiva	Minería de Datos	Redes de Computadoras	Códigos y Criptografía	Procesamiento de Imágenes	Sistemas Distribuidos	Cómputo Móvil
	Optimización Numérica		Computación Paralela	Cómputo Móvil		Minería de Datos	Computación Paralela
	Optimización Combinatoria		Sistemas Distribuidos			Geometría Computacional	Procesamiento de Imágenes
Especialización	Optimización en Ingeniería	Seguridad en Sistemas de Información	Sistemas Colaborativos Distribuidos	Cómputo Reconfigurable	Visión por Computadora	Bioinformática	Robots Móviles Inteligentes
	Computabilidad y Complejidad	Tópicos selectos de IA: Sistemas de Soporte a la Toma de Decisiones	Cómputo Móvil y Ubicuo	Tópicos Selectos en Criptografía	Reconocimiento de Patrones	Optimización Combinatoria	Sistemas Empotrados
	Tópicos Selectos de Computación Científica I		Tópicos Selectos de Sistemas Distribuídos	Tópicos Avanzados en Criptografía Simétrica		Optimización en Ingeniería	Cómputo Reconfigurable
	Tópicos selectos en IA: Sistemas de Agente y Multiagentes			Tópicos Selectos en Sistemas Digitales: VHDL		Redes Neuronales Artificiales	Visión por Computadora
	Tópicos Selectos en Inteligencia Artificial: Teoría de Juegos.			Tópicos Selectos en Teoría de Códigos		Seguridad en Sistemas de Información	Seguridad en Sistemas de Información
							Tópicos Selectos en Redes de Computadoras

9. Descripción de los cursos

Los cursos que de ofertan en el programa son listados a continuación.

Cursos del Núcleo

- Análisis y diseño de algoritmos
- Arquitectura de Computadoras
- Ingeniería de software
- Matemáticas discretas
- Programación Orientada a Objetos
- Sistemas operativos
- Base de datos
- Lenguajes de Programación

Cursos Formativos y de Especialización

- Aritmética computacional
- Bioinformática
- Códigos y Criptografía
- Computabilidad y complejidad
- Computación Paralela
- Cómputo basado en redes
- Cómputo Móvil
- Fundamentos Algebraicos de la Computación
- Geometría Computacional
- Graficación
- Inteligencia Artificial
- Introducción a la computación evolutiva
- Introducción al Cómputo Reconfigurable
- Introducción a la Teoría de la Probabilidad
- Lógica y Base de Datos
- Lógica matemática
- Minería de Datos
- Modelación y simulación de sistemas de eventos discretos
- Optimización Combinatoria
- Optimización en ingeniería
- Programación Concurrente
- Procesamiento digital de imágenes
- Realidad virtual
- Reconocimiento de Patrones
- Redes de Computadoras
- Redes Neuronales Artificiales
- Robots Móviles Inteligentes
- Seguridad en sistemas de información
- Sistemas Colaborativos Distribuidos
- Sistemas Distribuidos
- Sistemas Empotrados
- Sistemas de tiempo real
- Sistemas de tiempo real
- Teoría de Autómatas
- Topicos Selectos de Base de Datos
- Tópicos Selectos de Complejidad Computacional

- Tópicos Selectos de Computación Científica I
- Tópicos Selectos de Computación Científica II
- Tópicos Selectos en Criptografía
- Tópicos Selectos en Inteligencia Artificial: Agentes y Multiagentes
- Tópicos selectos en Inteligencia Artificial: Introducción a la optimización evolutiva multiobjetivo
- Tópicos Selectos en Inteligencia Artificial: Razonamiento Aproximado
- Tópicos selectos de Inteligencia Artificial: Sistemas de Soporte a la Toma de Decisiones
- Tópicos Selectos en Redes de Computadoras
- Tópicos Selectos en Redes Neuronales Artificiales
- Tópicos selectos en sistemas digitales: VHDL
- Tópicos Selectos en Teoría de Códigos
- Tópicos Selectos en Visualización
- Visión por computadora
- Seminario de Doctorado

A continuación se presentan los objetivos de cada curso ofertado en el programa

Cursos del Núcleo

Análisis y diseño de algoritmos

Presentar las técnicas para analizar y diseñar algoritmos y revisar la teoría computacional relacionada con la clasificación de problemas. En este curso se revisará el proceso de análisis de algoritmos así como las técnicas utilizadas para diseñar algoritmos eficientes. En la primera parte se introducirán algunos conceptos matemáticos necesarios para el análisis de algoritmos. Se revisarán los modelos computacionales más utilizados y se definirá de manera breve lo que se entiende por complejidad computacional. En la segunda parte se ejemplificará la complejidad de los algoritmos mediante el análisis de algoritmos típicos como ordenamiento, búsquedas, algoritmos sobre gráficas, etc. En la tercera parte se presentarán técnicas de diseño de algoritmos generales como programación dinámica, algoritmos ávidos y métodos branch-and-bound. Finalmente, en la cuarta parte se presentarán algunos resultados que determinan las clases de complejidad. Se introducirá la clasificación de los problemas de decisión, los problemas difíciles y los problemas completos, los problemas polinomiales y no-polinomiales.

Arquitectura de Computadoras

Los avances en el rendimiento de los procesadores modernos son dramáticos. Aun cuando buena parte del rendimiento actual se debe a los avances en la tecnología de computadoras, esto mismo ha permitido que la arquitectura de los procesadores evolucione y se pueden ejecutar más y mejores funciones directamente sobre un procesador. La disponibilidad a bajo costo de los microprocesadores, hace que el estudio de la arquitectura de computadoras sea necesario para aquel interesado en conocer cómo explotar al máximo el rendimiento de los procesadores actuales. En el curso se revisa la organización de las computadoras modernas y sus diferentes componentes. Se revisa la arquitectura de los procesadores modernos, su conjunto de instrucciones y la jerarquía de memoria sobre la cual estos han sido diseñados.
Se revisan los aspectos más relevantes de la arquitectura de computadoras los cuales le permiten ofrecer mejores rendimientos. Después de revisar los aspectos para evaluar el rendimiento de un procesador, se revisan los avances en el diseño de conjuntos de instrucciones. Posteriormente, se revisa la organización de la jerarquía

de memoria y los diferentes aspectos sobre la ejecución paralela de varias instrucciones.

Ingeniería de software

Debido a la complejidad de los sistemas de software, actualmente requerimos de metodologías para el desarrollo de los mismos. El curso de Ingeniería de Software tiene por objetivo explicar los principios que se usan en el ciclo de vida y los métodos para un desarrollo eficiente y de calidad. Se tratan los métodos clásicos (funcionales y estructurados) del análisis y diseño de sistemas, considerando las metodologías para el diseño de bases de datos y modelos de información.

Matemáticas discretas

Ofrecer al estudiante un panorama general de las Matemáticas que son particularmente útiles a las Ciencias de la Computación. Se inicia presentando las ideas básicas del principio de conteo y el razonamiento combinatorio elemental. A continuación se ofrece una introducción general a la lógica matemática, un estudio riguroso de la teoría de conjuntos, el principio de la inducción matemática y los métodos recursivos. Posteriormente se estudian las relaciones y funciones y se termina con lenguajes y máquinas de estados finitos. El curso no supone conocimientos matemáticos profundos previos y se enfoca principalmente a desarrollar la capacidad del estudiante para resolver problemas.

Programación Orientada a Objetos

Dar un repaso a los conceptos de la Programación Orientada a Objetos para obtener un nivel básico y avanzado de programación. Se hace énfasis en el lenguaje Java. El curso inicia con el desarrollo de aplicaciones de consola pasando por aplicaciones de escritorio mediante interfaces gráficas hasta llegar a la programación de aplicaciones vía web.

Sistemas operativos

Este curso aborda el diseño y la implementación de un sistema operativo: el software maestro que administra y controla los recursos tantos físicos como lógicos de una computadora. En este curso se estudia 1) las características de los dispositivos físicos, tales como: el procesador, la memoria principal, los dispositivos secundarios (terminales, discos, red, etc.) y 2) el diseño, los problemas y las técnicas de implementación de los componentes de software principales tales como: el núcleo de multiprogramación del procesador (procesos), el sistema de archivos, el módulo de sincronización y de comunicación entre procesos, el módulo de administración de memoria virtual, las técnicas de reservación/liberación dinámica de memoria, el sistema de entrada/salida de datos, la reservación/liberación de recursos.

Base de datos

En este curso se presentan diversos modelos de datos que son abstracciones matemáticas para representar la información del mundo real en datos y conocimiento. El curso cubre también los aspectos de la organización física de los datos, con detalles de implantación para cada uno de los modelos lógicos. Los diversos modelos son unificados mediante el modelo ente-vinculo de Chen que incorpora importante información semántica correspondiente al mundo real. Tomando como punto de

partida el modelo de Chen, se tratan los modelos semánticos de datos que incluyen técnicas de Representación de Conocimiento. Finalmente, considerando la corriente de extender los modelos basados en entidades y abstracción en base de datos, tratamos el enfoque Orientado a Objetos.

Lenguajes de Programación

Que el alumno conozca los conceptos más importantes asociados al diseño de lenguajes de programación, así como los principales paradigmas de programación, para permitirle realizar una evaluación critica de los lenguajes existentes y futuros, y desarrollar criterios para elegir el lenguaje más apropiado para resolver un determinado problema de programación.

Cursos Formativos y de Especialización

Aritmética computacional

Estudiar los métodos, algoritmos y técnicas de mejora de desempeño necesarias para obtener implementaciones eficientes de operaciones aritméticas en sistemas computacionales con recursos restringidos y en dispositivos de hardware reconfigurable FPGAs. Los conceptos y técnicas a ser revisados en este curso harán un énfasis especial en algoritmos de la aritmética de campos finitos y su correspondiente implementación en dispositivos FPGAs.

Bioinformática

El de este curso es que el estudiante adquiera los conocimientos y habilidades necesarios para efectuar el análisis computacional de datos biológicos, con el fin de enunciar predicciones funcionales o estructurales sobre los mismos. Para ello durante el curso se abordarán las principales áreas de la Bioinformática: bases de datos biológicas, alineamiento de secuencias, descubrimiento de genes y promotores, filogenética molecular, bioinformática estructural, genómica y proteómica. Simultáneamente, se analizarán algunas de las principales aplicaciones de la Bioinformática.

Códigos y Criptografía

Estudiar algunos de los principales métodos, algoritmos, técnicas y herramientas necesarias para la implementación de aplicaciones criptográficas y de seguridad de datos. El contenido del curso está orientado a la resolución eficiente del problema de cómo establecer una comunicación segura entre dos o más entidades de manera tal que se garantice un alto grado de confidencialidad, integridad y autenticidad en los datos y documentos intercambiados. El contenido del curso incluye el estudio de diversos aspectos teóricos relacionados con los algoritmos y esquemas criptográficos, así como la aplicación de dichas técnicas para resolver problemas prácticos en el envío de datos y comunicación de manera segura.

Computabilidad y complejidad

Conocer los diversos paradigmas de computación formal, entre estos las funciones recursivas y las máquinas de Turing. Examinar las limitaciones de la noción de computabilidad y los criterios diversos de clasificación de problemas atendiendo a la

complejidad de sus procedimientos de solución. Se presenta a las funciones recursivas siguiendo el enfoque de máquinas de Turing, de programas-while y el puramente formal. Luego se presenta los grados de irresolubilidad para pasar después las jerarqu´ıas temporal y espacial de los problemas tratables. Ya para terminar, presentaremos una colección de problemas completos-NP. Al último presentamos la noción de complejidad abstracta debida a Kolmogorov.

Computación Paralela

El propósito de este curso es discutir técnicas y aplicaciones de la computación paralela y de forma en que se construyen aplicaciones para este paradigma. En este curso nos concentraremos en el uso de varias computadoras que se comunican entre sí ya sea a través de una memoria compartida o mediante el envío de mensajes.

Cómputo basado en redes

Entender los conceptos fundamentales y desarrollar las habilidades de programación requeridos para la construcción de sistemas distribuidos basados en la arquitectura cliente-servidor de Internet. Estudiar algunas de las tecnologías de Internet incorporadas al lenguaje de programación Java tales como Berkley Sockets, Servlets, Java Server Pages, Remote Method Invocation, CORBA y Enterprise Java Beans las cuales facilitan el desarrollo de aplicaciones distribuidas. Desarrollar prácticas y proyectos de programación que enfaticen conceptos como sistemas abiertos, interoperabilidad, portabilidad, seguridad e integración.

Cómputo Móvil

La necesidad de información en cualquier momento y lugar, conjuntamente con el surgimiento de dispositivos de cómputo portátiles y los avances en las tecnologías de comunicación inalámbrica e Internet, han hecho a la Computación Móvil una realidad. Esta tiene como finalidad, el tratamiento automático de información por medio de dispositivos computacionales con capacidad de movilidad y con acceso digital a fuentes de información via una infraestructura de comunicaci ón inalámbrica. Los ambientes de cómputo móvil se
caracterizan por restricciones importantes de recursos y cambios frecuentes en las condiciones de operación lo cual impone desafios que involucran diversas areas de la ciencias computacionales, ingenieríaś computacional, eléctrica y de telecomunicaciones. El principal de este curso es el de proporcionar un marco general de los fundamentos, métodos y tecnologías para la compresión de problemáticas y el desarrollo de aplicaciones en el área de computación móvil. Adicionalmente a la formación teórica, el curso tendrá una fuerte orientación práctica al desarrollo de aplicaciones en dispositivos móviles, articularmente en teléfonos celulares con soporte para Android.

Fundamentos Algebraicos de la Computación

El principal de este curso es el de introducir los conceptos básicos de las estructuras algebraicas. El conocimiento de estas estructuras es indispensable para el estudio de las ciencias computacionales modernas, por lo que el material a ser cubierto en este curso sirve como base teórica para el estudio de temas tales como teoría de códigos, criptología, etc.

Geometría Computacional

En este curso se estudiarán los conceptos básicos de la Geometría Computacional

(GC) así como sus aplicaciones. El material aborda las técnicas necesarias para el diseño y análisis de algoritmos eficientes para resolver problemas en geometría, tales como: cubiertas convexas (*convex hulls*), intersecciones geométricas, diagramas de Voronoi, triangulaciones de Delaunay, estructuras de datos geométricas, etc.

Graficación

El del curso es estudiar las diversas técnicas de la literatura para el trazado de una escena y de las formas bi- y/o tridimensionales que la componen, además de las técnicas para manipularla y visualizarla. La herramienta sugerida de trabajo, para desarrollar las tareas del curso, es un sistema de desarrollo de interfaces gráficas (GUI, por sus siglas en inglés) basado en objetos, llamado Qt (www.trolltech.com) y OpenGL (www.opengl.org) o Mesa (www.mesa3d.org) para la construcción y manipulación de escenas tridimensionales.

Inteligencia Artificial

Los trabajos desarrollados en Inteligencia Artificial (IA) versan sobre la ciencia del conocimiento y las máquinas inteligentes. La primera modela y desarrolla estructuras que permiten explicar los mecanismos utilizados por el ser humano para resolver problemas. En cambio, las máquinas inteligentes son sistemas que ejecutan funciones similares a las desarrolladas por un especialista. Estas últimas constituyen las ramas de aplicación de la IA: Resolución de problemas, Sistemas Expertos, Procesamiento de Lenguaje Natural, Aprendizaje automático, Demostración automática de teoremas, Visión, Robótica, etc. Este curso se orienta principalmente a este último enfoque: las aplicaciones de la IA.

Introducción a la computación evolutiva

En este curso se estudiaran los conceptos básicos de las técnicas más importantes de computación evolutiva, haciendo especial énfasis en los algoritmos genéticos. Inicialmente, se harán un recorrido histórico en el que se resumirán los logros más importantes en torno a la simulación de los procesos evolutivos como una herramienta para el aprendizaje y la optimización. Posteriormente, se analizaran y compararan de manera general los 3 paradigmas principales que se utilizan hoy en día en la computación evolutiva: las estrategias evolutivas, la programación evolutiva y los algoritmos genéticos. En cada caso se abordara su inspiración biológica, su motivación, su funcionamiento y algunas de sus aplicaciones. Finalmente, se estudiara a mayor detalle el funcionamiento, fundamentos teóricos, implementación y operación de los algoritmos genéticos, que es actualmente el paradigma evolutivo más utilizado por los investigadores que trabajan en esta disciplina.

Introducción al Cómputo Reconfigurable

El curso presenta los fundamentos y formalismo del Cómputo Reconfigurable así como metodologías y técnicas de cómo diseñar y aplicar arquitecturas bajo este enfoque en la solución de problemas que involucran una alta demanda computacional. Se analizan las aspectos centrales para implementar cálculos bajo este paradigma y se abren perspectivas de cómo construir sistemas de cómputo cuya arquitectura, maximalmente concurrente, refleje la naturaleza estructural de su descripción algorítmica. Adicionalmente en el curso se analizan y estudian técnicas y herramientas de diseño, análisis y optimización de sistemas de cómputo reconfigurable basado en tecnología FPGA. Finalmente se estudia y analiza el estado del arte bajo la perspectiva hardware y software a fin de identificar líneas potenciales de investigación.

Introducción a la Teoría de la Probabilidad

Se estudian los principios básicos de la teoría de la probabilidad junto con algunas aplicaciones de ´esta. Se familiariza al estudiante con distintas distribuciones de probabilidad las cuales son utilizadas frecuentemente en otras áreas de las ciencias computacionales.

Lógica y Base de Datos

Base de datos y lógica se interrelacionan como fundamentación y como un enfoque de sistema deductivo en la solución de problemas visto como consultas en la base de datos. Los primeros temas consisten en una revisión de la lógica de predicados de primer orden y la demostración automática de teoremas para, posteriormente, encontrar una fundamentación para las bases de datos en el lenguaje relacional, restricciones de integridad y diseño de base de datos. La segunda parte es usar la aproximación de la lógica como sistema deductivo a las bases de datos con aplicación de algunos sistemas como CORAL a un sistema relacional Posgress. Algunos temas selectos se presentan con la finalidad de una investigación posterior más profunda.

Lógica matemática

Presentar los conocimientos básicos de Lógica Matemática para poder trabajar en Programación Lógica, en Inteligencia Artificial y, en general, en Ciencias de la Computación.

Minería de Datos

Conocer de manera general las técnicas y enfoques del proceso general de Minería de Datos. Se conocen los fundamentos y conceptos necesarios de cada una de las etapas del proceso. Se explora el uso de fuentes de datos para análisis y toma de decisiones resultantes de tareas de clasificación, predicción o agrupamiento.

Modelación y simulación de sistemas de eventos discretos

Es curso provee una introducción al modelado de sistemas usando tanto simulaciones de computadora como técnicas matemáticas. Se espera que al final de curso el alumno tenga el conocimiento suficiente para la realización de simulaciones efectivas. El software usado para las simulaciones es ARENA.

Optimización Combinatoria

El de este curso es que el estudiante conozca y aplique a problemas teóricos y del mundo real el paradigma de solución basado en la optimización combinatoria. El paradigma de solución de la optimización combinatoria descansa en fundamentos teóricos para modelar y representar problemas, y en algoritmos de búsqueda local para resolver dichos problemas. El fundamento teórico de la optimización combinatoria se apoya en la teoría de la complejidad computacional y en el uso de objetos matemáticos para representar problemas de optimización combinatoria. Los objetos matemáticos que serán presentados incluyen; grafos, árboles, permutaciones, particiones, cadenas de crecimiento restringido, polinomios diagonales, polinomios caja, polinomios “mayor que”. Es altamente relevante el énfasis que se hace sobre la representación, generación, y enumeración. Los algoritmos de búsqueda local usan la modelización/representación basada en objetos matemáticos para resolver instancias de problemas teóricos y prácticos. Los algoritmos de búsqueda local que serán tratados en detalle incluyen: a) algoritmos de mutación/selección; b) algoritmo de recocido simulado; c) algoritmos genéticos; d) búsqueda tabú; y e) algoritmo de colonia de hormigas.

Al final del curso se espera que el estudiante tenga los conocimientos y habilidades necesarias para modelar problemas de la vida real usando al menos los objetos matemáticos vistos en el curso , y resolver dichos problemas usando algoritmos de optimización combinatoria básicos o mezclas de ellos.

Optimización en ingeniería

Presentar al estudiante un repaso histórico, teórico y práctico de los diversos métodos de optimización global, enfatizando sus ventajas y desventajas. Así mismo, generar habilidades para decidir y modificar técnicas según las demandas de la aplicación específica. En este curso se estudian diversos métodos de programación matemática para resolver problemas de optimización lineal y no lineal (principalmente sin restricciones). El curso enfatizará aspectos algorítmicos y de implementación sobre los aspectos teóricos, por lo que es necesario tener al menos conocimientos básicos de programación. También se requieren conocimientos de cálculo, trigonometría, geometría y álgebra.

Programación Concurrente

Se pretende:

- Estudio de los mecanismos para compartir y controlar recursos.
- Estudio de los mecanismos basados en paso de mensajes.
- Estudio de lenguajes académicos de programación concurrente: Pascal – S, SR.
- los conceptos fundamentales en el diseño e implementación de aplicaciones multitarea.
- Uso de bibliotecas para el desarrollo de aplicaciones multi-hilo: Pthreads, Java-Threads

Procesamiento digital de imágenes

El curso está enfocado a introducir los conceptos teóricos y prácticos asociados con el procesamiento de imágenes digitales bidimensionales. Los siguientes tópicos principales son cubiertos: preprocesamiento, segmentación, análisis y reconocimiento de la imagen. Al completar el curso el estudiante será capaz de aplicar los conceptos aprendidos en problemas reales, así como comprender textos avanzados de investigación en el campo de procesamiento de imágenes.

Realidad virtual

Se estudian la tecnología actual para realizar realidad virtual, realidad aumentada y uso de interfaces hápticas. Se analizaran los aspectos de ingeniera, científicos y aspectos funcionales de sistemas de realidad virtual y los fundamentos de modelado de mundos virtuales y su programación. Se tratan las técnicas de imagen con OpenGL para la generación de ambientes tridimensionales (3D): creación de modelos, diseño de escenarios, iluminación, texturas, interiores, exteriores y perspectiva.

Reconocimiento de Patrones

Conocer los procesos en ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades de o entre conjuntos de dichos objetos. El reconocimiento de patrones es un proceso fundamental en casi todas las acciones humanas. Lo esencial de él es la clasificación. Clasificar patrones en base a un conocimiento a priori o información estadística extraída de los patrones. Los patrones a clasificar suelen ser

grupos de medidas u observaciones, definiendo puntos en un espacio multidimensional apropiado. Entre las aplicaciones del reconocimiento de patrones tenemos: el reconocimiento de voz, clasificación de documentos, reconocimiento de la escritura, reconocimiento de caras humanas, etc.

Redes de Computadoras

En este curso el alumno conocerá las tecnologías involucradas en las redes de computadoras analizadas a través de la arquitectura por capas siguiendo un enfoque descendente. Presentar los principios básicos de la arquitectura TCP/IP y su implementación en Internet. Se revisaran distintas alternativas de interconexión de redes, la función y problemática de cada una de las capas del modelo ISO/OSI. Se dará especial énfasis a las capas de aplicación, transporte y red del modelo de referencia de Internet. El alumno conocerá los protocolos básicos dentro de cada capa. Complementara los estudios teóricos con implementaciones básicas de algunos de los algoritmos y protocolos analizados. Analizara algunos aspectos generales de gestión de redes y de nuevas tecnologías de redes inalámbricas.

Redes Neuronales Artificiales

Conocer, entender y aplicar algunos modelos clásicos de Redes Neuronales Artificiales (RNA) para la resolución de problemas y descubrir la estrecha relación entre la neurobiología, la probabilidad, la estadística y la computación para el modelado con RNA inspiradas en el sistema nervioso. Las Redes de Neuronas Artificiales son un paradigma de aprendizaje y de procesamiento automático inspirado en el funcionamiento del sistema nervioso. Éstas simulan las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, una computadora o un conjunto de válvulas). El es conseguir que las máquinas den respuestas robustas y de generalización similares a las que es capaz de dar nuestro cerebro. En este curso se busca entender el uso de las RNA como herramienta de análisis de patrones y su posible aplicación al estudio de la codificación de la información en diversos sistemas sensoriales, además de aplicar las RNA en el modelado de sistemas biológicos y simular el alto nivel de paralelismo en el procesamiento de la información visual. Para ello empezaremos revisando sus fundamentos biológicos. Seguiremos con la modelización clásica de RNA (características, topología, reglas de aprendizaje, etc.) y haremos una clasificación de las mismas, analizaremos las RNA con alimentación hacia adelante, aquellas con alimentación hacia atrás, las memorias asociativas, ejemplos de aplicación y una introducción al modelado bio-inspirado : conexionismo neuromimético.

Robots Móviles Inteligentes

Conocer los fundamentos de los aspectos computacionales utilizados en robots móviles autónomos. Estudiar los componentes de un robot móvil: percepción, visión, planeación, navegación, construcción de mapas y localización. Aplicar los conocimientos adquiridos en el desarrollo de proyectos con robots móviles.

Seguridad en sistemas de información

El curso está organizado en tres partes fundamentales. En la primera se revisan los aspectos fundamentales para construcción de un cortafuegos. En la segunda parte se revisan los fundamentos criptográficos para proveer servicios de seguridad y algunas aplicaciones. En la tercera parte se revisan algunas herramientas que ayudan a detectar y corregir problemas de seguridad informática en un sistema de información.

Sistemas Colaborativos Distribuidos

Dar a conocer al alumno los fundamentos teóricos y prácticos del campo de investigación multidisciplinario denominado ”Trabajo Cooperativo Asistido por Computadora” (*CSCW por sus siglas en inglés*), haciendo énfasis en el estudio de los sistemas computacionales (*Groupware* por su denominación en inglés) que soportan grupos de personas comprometidas en un proyecto común y que proveen una interfaz a un entorno compartido. En particular, se analizan las arquitecturas de distribución fundamentales para permitir a personas físicamente distribuidas comunicar, colaborar y coordinar sus actividades como si estuvieran cara a cara. Asimismo, se estudian los principales mecanismos propuestos para administrar la compartición de la información, tanto a nivel de interfaz de grupo como a nivel de núcleo funcional. Este dominio de investigación ha contribuido a la evolución de diversos dominios de aplicación, entre los que se encuentran los sistemas de mensajes, los editores de grupo, los sistemas de soporte para la toma de decisiones en grupo, las salas de reuniones virtuales, las conferencias por computadora, los agentes inteligentes, los sistemas de coordinación (*workflows*) y la enseñanza/aprendizaje colaborativo.

Sistemas Distribuidos

Dotar al alumno de conocimientos para que pueda comprender y aplicar los sistemas distribuidos, tanto en el área de base de datos, como en el área de aplicaciones de red, como manejo de protocolos, sistemas operativos, bajo diferentes tipos de enlaces, diferentes arquitecturas de cómputo distribuido(cliente/servidor). Se analizarán las principales tecnologías de programación para sistemas distribuidos utilizando Middlewares como RPCs, RMIs, Corba y Servicios Web.

Sistemas Empotrados

Particularmente el curso persigue los siguientes s:

- Establecer un marco de las tecnologías y componentes hardware y software para el desarrollo de sistemas empotrados, así como las herramientas que asistan en la administración de diseños
- Evaluar compromisos de diseño entre diferentes opciones tecnológicas para la implementación de sistemas empotrados

Aunado al contenido teórico de carácter general, el curso tiene una fuerte orientación al desarrollo de aplicaciones prácticas. Los aspectos teóricos se reforzarán fundamentalmente mediante la realización de un proyecto de diseño hardware/software de un sistema empotrado. En este sentido, es deseable, tener conocimientos previos de Lenguajes de descripción de hardware, Síntesis de alto nivel y programación.

Sistemas de tiempo real

En los sistemas de tiempo real existen 3 componentes principales que los caracterizan, así como la interrelación entre estos. En primer lugar está el tiempo, el cual es el recurso más valioso a tratar y gestionar en un sistema en tiempo real. Las tareas deben ser asignadas y planificadas a fin de que cumplan unos plazos (deadlines). La correcta ejecución del sistema de tiempo real depende tanto de la validez lógica de la respuesta, como del instante de tiempo en que se produce. Como segundo componente se encuentra la confiabilidad, debido a que un fallo en el sistema

en tiempo real podría causar serias consecuencias. Estos dos componentes son la principal causa de una mayor dificultad en el diseño de estos sistemas respecto a los sistemas informativos de propósito general. El tercer componente involucra al ambiente en el cual el computador opera, el cual es un componente activo en cualquier sistema de tiempo real. Por ejemplo en un sistema de avión, no tiene sentido considerar el sistema de cómputo por separado de la aeronave, ya que ambos interactúan entre sí.

Sistemas de tiempo real

En los sistemas de tiempo real existen 3 componentes principales que los caracterizan, así como la interrelación entre estos. En primer lugar está el tiempo, el cual es el recurso más valioso a tratar y gestionar en un sistema en tiempo real. Las tareas deben ser asignadas y planificadas a fin de que cumplan unos plazos (deadlines). La correcta ejecución del sistema de tiempo real depende tanto de la validez lógica de la respuesta, como del instante de tiempo en que se produce. Como segundo componente se encuentra la confiabilidad, debido a que un fallo en el sistema en tiempo real podría causar serias consecuencias. Estos dos componentes son la principal causa de una mayor dificultad en el diseño de estos sistemas respecto a los sistemas informativos de propósito general. El tercer componente involucra al ambiente en el cual el computador opera, el cual es un componente activo en cualquier sistema de tiempo real. Por ejemplo en un sistema de avión, no tiene sentido considerar el sistema de cómputo por separado de la aeronave, ya que ambos interactúan entre sí.

Teoría de Autómatas

Conocer los elementos básicos de los lenguajes formales, de los dispositivos formales que los reconocen y conocer también a la Jerarquía de Chomski.

Topicos Selectos de Base de Datos

Familiarizar a los estudiantes con los fundamentos, las bondades, problemas y retos que hay detrás de las BBDD actuales al momento de relacionarse con tecnologías como las Redes de Computadoras, la Orientación a Objetos, las tecnologías de Georeferenciación, el manejo de Almacenes de Datos, la Web y el manejo de datos semi estructurados (XML) y no estructurados. El curso de divide en temas principales en los que se profundizará acorde a las necesidades particulares de investigación en el área.

Tópicos Selectos de Complejidad Computacional

Revisión de los aspectos clásicos de la teoría de los problemas NP-completos y la investigación reciente relacionada con la caracterización de problemas NP-completos.

Tópicos Selectos de Computación Científica I

Estudiar la teoría y los métodos matemáticos-computacionales para la resolución de algunos problemas de ingeniería, física, química y biología. Se hace énfasis en ecuaciones lineales de orden *n* y sistemas ecuaciones diferenciales de primer orden. Abordando el problema de condiciones iniciales y el problema de valores a la frontera. El problema de Sturm-Liouville es tratado como un problema de valores a la frontera regular y como un problema de valores propios. Se estudia el método de Monte Carlo para simulación y solución de algunos problemas científicos y de ingeniera.

Tópicos Selectos de Computación Científica II

Estudiar los modelos y métodos matemáticos computacionales haciendo énfasis en aspectos algebraicos, geométricos y de visualización. Se estudian teoría de grupos de matrices de rotación con aplicación a algunos problemas físicos. Visualización de geometría del espacio fase en las soluciones de ecuaciones diferenciales. Autómatas Celulares y sistemas dinámicos discretos es tratdo en este curso como un nuevo ambiente para modelar problemas de ecosistemas y físicos.

Tópicos Selectos en Criptografía

Este curso presenta una selección de los descubrimientos más recientes en criptografía. El curso inicia con un análisis y recuento de las primitivas usadas en criptografía, seguido por una discusión de los esquemas que han sido propuestos recientemente para realizar criptografía simétrica y de llave pública.

Tópicos Selectos en Inteligencia Artificial: Agentes y Multiagentes

Los sistemas multiagentes surgieron en el campo de la investigación de tecnología de la información en la década de los 90. Un agente es un sistema o componente de software, el cual es capaz de cooperar para resolver problemas específicos. El del curso es dar una visión introductoria a los agentes autónomos y a los sistemas multiagentes desde el punto de vista teórico como práctico. Se explicarán las diferentes arquitectura de agente (reactiva, deliberativa e hibrida), así como los mecanismos de interacción, coordinación y cooperación entre sistemas multiagentes. Las aplicaciones son diversas: control de procesos industriales, comercio electrónico, subastas, etc.

Tópicos selectos en Inteligencia Artificial: Introducción a la optimización evolutiva multiobjetivo

En este curso se estudiarán los conceptos básicos de la optimización multi, así como el uso de los algoritmos evolutivos en esta área. El material cubierto abarca desde los orígenes de la optimización multi (en economía y planeación), hasta los avances más recientes. Además de analizar las técnicas evolutivas multi de mayor uso en la actualidad, se estudiarán otras heurísticas que también han sido extendidas para lidiar con problemas multi (p.ej., la colonia de hormigas), discutiendo sus ventajas y limitantes principales. Adicionalmente, se revisará el trabajo teórico realizado en esta área y se discutirán algunos de los temas de investigación futura que han permanecido poco explorados durante los últimos años.

Tópicos Selectos en Inteligencia Artificial: Razonamiento Aproximado

El tratamiento de la incertidumbre constituye uno de los campos fundamentales de la Inteligencia Artificial (IA), ya que esta presente en todas las aplicaciones de la IA (sistemas expertos, control automático, aprendizaje por una maquina, etc). En el curso se estudiaran los diferentes modelos de razonamiento aproximado; se desarrollara una base de conocimientos; y se diseñara y construirá un prototipo que muestre el razonamiento aproximado, de acuerdo con alguno de los modelos estudiados.

Tópicos selectos de Inteligencia Artificial: Sistemas de Soporte a la Toma de Decisiones

1. Estudio de metodologías y herramientas, matemáticas y computacionales, para el análisis, diseño y desarrollo de sistemas de soporte a la Toma de Decisiones.
2. Estudio de las metodologías y herramientas para el análisis, diseño y desarrollo de los procesos involucrados en la Toma de Decisiones.
3. Estudio de los lenguajes para el análisis, diseño y desarrollo de sistemas de soporte a la Toma de Decisiones.
4. Estudio las técnicas de Inteligencia Artificial, de Teoría de Juegos y de Administración del Conocimiento, con las cuales se fundamenta el desarrollan de sistemas inteligentes para la Toma de Decisiones.

Tópicos Selectos en Redes de Computadoras

En este curso se estudian las técnicas más recientes para la construcción de redes de computadoras y protocolos de comunicación. Se revisan las tecnologías y protocolos de comunicación utilizadas en redes inalámbricas y móviles. Se analizan los requisitos de servicio que demandan las nuevas aplicaciones multimedia. Se estudian las arquitecturas y tecnologías para el aprovisionamiento de calidad de servicio en redes de próxima generación. Se describen los conceptos fundamentales de gestión de redes y servicios, como un instrumento para el aprovisionamiento de calidad de servicio. Se describen las tecnologías emergentes para realizar la automatización de las tareas de gestión de red en ambientes dinámicos. Se realizan simulaciones mediante software especializado con el fin de aplicar los conocimientos adquiridos en el curso.

Tópicos Selectos en Redes Neuronales Artificiales

Comprender y aplicar los métodos neuronales a la resolución de problemas complejos inspirándonos en el funcionamiento de nuestro cerebro para su concepción modular, local, distribuida y paralela con el fin de crear sistemas fácilmente adaptables e integrables a otros.

Tópicos selectos en sistemas digitales: VHDL

Se describen los varios estilos de diseño en VHDL así como sus niveles de abstracción. Se enfatizan, en forma muy particular, las características que permiten al lenguaje VHDL diseñar circuitos digitales, como también la capacidad de crear diseños transportables. Se analizan la sintaxis y la síntesis como el análisis sobre su mapeo en FPGA y la posibilidad de rediseño para corregir errores y mejorar sus características funcionales.

Tópicos Selectos en Teoría de Códigos

En la última década hemos presenciado numerosos y significativos avances en la teoría de códigos. El material de este curso se propone motivar el conocimiento de la teoría de códigos, así como presentar algunos de los últimos avances alcanzados en esta disciplina. El curso inicia con una introducción a la teoría de la información de Shannon para después discutir y analizar las propiedades y cotas teóricas de códigos específicos de corrección de error.

Tópicos Selectos en Visualización

En este curso se aplicarán las nociones de visión por computadora (VC) en tres dimensiones. La meta de VC es deducir las propiedades y estructura de un mundo tridimensional a partir de una o más vistas bidimensionales. Primero se estudiarán

algunas técnicas para procesamiento y análisis de imagen y también se tratarán temas de visualización 3D, animación y realidad virtual, para la creación de modelos tridimensionales y para tener la habilidad de "navegar" a través de ellos. Las herramientas de trabajo serán la librería de procesamiento de imágenes scimagen, y Qt (www.trolltech.com) para el desarrollo de las interfaces gráficas y Mesa (www.mesa3d.org) para interactuar con objetos tri-dimensionales.

Visión por computadora

Se revisará la teoría para la reconstrucción tridimensional de escenas a partir de una o varias imágenes bidimensionales, tomadas por una cámara convencional. Se hará énfasis en los métodos para obtener la reconstrucción a partir de las correspondencias de puntos entre las imágenes, lo que se conoce como *autocalibración de la cámara*.

Seminario de Doctorado

En este curso se revisa el avance de cada investigación doctoral que se está llevando a cabo en el programa. El trabajo del seminario de doctorado consiste de dos exposiciones, participación en el seminario y la entrega de un reporte escrito en inglés con la investigación desarrollada en el último año de actividades. Para el caso de estudiantes que tienen menos de un año, será a partir de su fecha de ingreso. Cada estudiante de doctorado formalmente inscrito deberá participar en el seminario exponiendo los avances en su investigación doctoral.

Contenido de cursos

A continuación se presentan los contenidos detallados por cada curso ofertado en el programa

Análisis y diseño de algoritmos

Objetivo

Presentar las técnicas para analizar y diseñar algoritmos y revisar la teoría computacional relacionada con la clasificación de problemas. En este curso se revisará el proceso de análisis de algoritmos así como las técnicas utilizadas para diseñar algoritmos eficientes. En la primera parte se introducirán algunos conceptos matemáticos necesarios para el análisis de algoritmos. Se revisarán los modelos computacionales más utilizados y se definirá de manera breve lo que se entiende por complejidad computacional. En la segunda parte se ejemplificará la complejidad de los algoritmos mediante el análisis de algoritmos típicos como ordenamiento, búsquedas, algoritmos sobre gráficas, etc. En la tercera parte se presentarán técnicas de diseño de algoritmos generales como programación dinámica, algoritmos ávidos y métodos branch-and-bound. Finalmente, en la cuarta parte se presentarán algunos resultados que determinan las clases de complejidad. Se introducirá la clasificación de los problemas de decisión, los problemas difíciles y los problemas completos, los problemas polinomiales y no-polinomiales.

Contenido

1. El papel de la teoría en las ciencias de la computación
    a. Historia breve sobre la Teoría de la Computación
    b. Preliminares matemáticos
    c. El rango de crecimiento de las funciones
    d. Modelos computacionales
    e. Complejidad computacional

2. Análisis de Algoritmos
    a. Historia breve sobre la Teoría de la Computación
    b. Ecuaciones de recurrencia
    c. Métodos de búsqueda
    d. Métodos de ordenamiento
    e. Estructuras de datos elementales y tablas de dispersión
    f. Arboles rojinegros y estructuras aumentadas
    g. Algoritmos elementales sobre teoría de gráficas

3. Técnicas para Diseño de Algoritmos
    a. Algoritmos de Fuerza Bruta
    b. Divide y Vencerás
    c. Programación Dinámica
    d. Algoritmos ávidos
    e. Backtracking
    f. Branch-and-bound

4. Complejidad computacional
    a. Lenguajes y problemas
    b. Recursos acotados
    c. Clasificación de los problemas de decisión
    d. Problemas difíciles y problemas completos
    e. Problemas P y NP

Bibliografía

1. A.V. Aho, J.E. Hopcroft, and J.D. Ullman, The Analysis and Design of Computer Algorithms. Addisson-Wesley. Reading, Mass. ISBN: 0201000296. 1974.
2. A.V. Aho, and J.D. Ullman, Foundations of Computer Science (Principles of Computer Science Series). W H Freeman & Co. ISBN: 0716782847. 1995.
3. Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. Introduction to Algorithms (MIT Electrical Engineering and Computer Science Series). The MIT Press, ISBN: 0262031418. 1990. (Libro de Texto).
4. Garey, M. and Johnson, D.S., Computers and untractability: A guide to the theory of NP-completeness. W H Freeman & Co.; ISBN: 0716710455. 1979.
5. Graham, R., Knuth, D. E. and Patashnik, O. Concrete Mathematics. Addison-Wesley, 1989.
6. Savage, John, E. Models of Computation: Exploring the Power of Computing. Addisson-Wesley. Reading, Mass. 1998. ISBN: 0201895390.

Arquitectura de Computadoras

Objetivo

Revisar la organización y arquitectura de los sistemas de cómputo modernos que permiten mejorar su rendimiento.

Descripción

Los avances en el rendimiento de los procesadores modernos son dramáticos. Aun cuando buena parte del rendimiento actual se debe a los avances en la tecnología de computadoras, esto mismo ha permitido que la arquitectura de los procesadores evolucione y se pueden ejecutar más y mejores funciones directamente sobre un procesador. La disponibilidad a bajo costo de los microprocesadores, hace que el estudio de la arquitectura de computadoras sea necesario para aquel interesado en conocer cómo explotar al máximo el rendimiento de los procesadores actuales. En el curso se revisa la organización de las computadoras modernas y sus diferentes componentes. Se revisa la arquitectura de los procesadores modernos, su conjunto de instrucciones y la jerarquía de memoria sobre la cual estos han sido diseñados.

Se revisan los aspectos más relevantes de la arquitectura de computadoras los cuales le permiten ofrecer mejores rendimientos. Después de revisar los aspectos para evaluar el rendimiento de un procesador, se revisan los avances en el diseño de conjuntos de instrucciones. Posteriormente, se revisa la organización de la jerarquía de memoria y los diferentes aspectos sobre la ejecución paralela de varias instrucciones.

Contenido

1. Introducción. Los microprocesadores de inicios del siglo XX1.
    a. Organización y arquitectura de las computadoras modernas
    b. La evolución y rendimiento de las computadoras

2. Conjuntos de instrucciones: características y funciones.
    a. Tipos de operandos y tipos de operaciones
    b. Formatos y modos de direccionamiento
    c. Ciclos de instrucciones
    d. Ejecución pipeline
    e. Conjuntos de instrucciones reducidos

3. Jerarquía de Memoria
    a. Memoria caché
    b. Organización de DRAM
    c. Memoria principal

4. Paralelismo a nivel de instrucciones y procesadores superescalares.
    a. Revisión
    b. Aspectos de diseño
    c. Predicción de saltos
    d. Ejecución fuera de orden

Bibliografía

1. Culler, David E., Sing, Jaswinder Pal and Gupta, Anoop. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kauffmann Publishers. San Francisco, CA. 1996. ISBN 1-55860-343-3.
2. Hennesy, John L. and Patterson, David A. Computer Architecture: A Quantitative Approach: Third Edition. Morgan Kauffmann Publishers. San Francisco, CA. 2003. ISBN 1-55860-724-2.
3. Hill, Mark D., Jouppi, Norman P. and Sohi, Furindar P Eds. Readings in Compuer Architecture. Morgan Kauffmann Publishers. San Francisco, CA.

2000. ISBN 1-55860-539-8.
4. Morris Mano, M. and Kime, Charles R. Logic and Computer Design Fundamentals: Second Edition. Prentice Hall. New Jersey. ISBN 0-13-012468-0.
5. Patterson, David A and Hennesy, John L. Computer Organization & Design: The Hardware/Software Interface: Third Edition. Morgan Kauffmann Publishers. San Francisco, CA. 2005. ISBN 1-55860-604-1.
6. Stallings,William. Computer Organization and Architecture: Fifth Edition. Prentice Hall. New Jersey. ISBN 0-13-081294-3.
7. Dezso Sima, Terence Fountain, Peter Kacsuk, ”Advanced Computer Architectures: A Design Space Approach”, Addison-Wesley, 1998. ISBN: 0-201-42291-3.
8. IEEE Computer Magazine.

Ingeniería de software

Objetivo

Este curso tiene como objetivo que los estudiantes adquieran los conocimientos y competencias en el área de Ingeniería de Software necesarios para llevar a cabo exitosamente proyectos complejos de desarrollo de software.

Contenido

1. Introducción a la Ingeniería de Software
    a. Fallas en ingeniería de software
    b. ¿Qué es la ingeniería de software?
    c. Conceptos básico de ingeniería de software
    d. Actividades de la ingeniería de desarrollo de software
    e. La gestión del desarrollo de software
    f. Los costos de la ingeniería de software
    g. Desafíos clave que enfrenta la ingeniería del software
    h. Ética profesional y responsabilidad

2. Procesos de software
    a. El proceso de software
    b. Modelo de proceso de Software
    c. El modelo de cascada
    d. Modelos incrementales de proceso
    e. Modelos evolutivos de proceso
    f. El Proceso Unificado (RUP)
    g. Modelos ágiles de procesos

3. Modelado con UML
    a. Conceptos básicos de modelado
    b. Diagramas de caso de uso
    c. Diagramas de clases
    d. Diagramas de interacción
    e. Diagramas de estado
    f. Diagramas de actividades

4. Ingeniería de requerimientos
    a. Obtención de requerimientos

   b. Análisis de requerimientos
   c. Validación de requerimientos
   d. Gestión de requerimientos

5. Modelado de análisis
   a. Análisis de requerimientos
   b. Conceptos y enfoques de modelado de análisis
   c. Modelado de datos
   d. Conceptos de modelado orientado a objetos
   e. Modelado basado en escenarios
   f. Modelado basado en clases

6. Ingeniería de diseño
   a. Proceso de diseño
   b. Conceptos de diseño
   c. Modelo de diseño
   d. Arquitectura del software
   e. Diseño de datos
   f. Arquitecturas y patrones
   g. Diseño de arquitectura
   h. Diseño de la interfaz de usuario

7. Verificación y validación del software
   a. Estrategias de prueba para software tradicional
   b. Estrategias de prueba para software orientado a objetos
   c. Pruebas de validación
   d. Pruebas de sistema
   e. Depuración
   f. Fundamentos de pruebas de software
   g. Pruebas de caja blanca
   h. Pruebas de ruta básica de ejecución
   i. Pruebas de estructura de control
   j. Pruebas de caja negra
   k. Métodos de prueba orientados a objetos

8. Métricas para el software
   a. Calidad del software
   b. Métrica para el modelo de análisis
   c. Métricas para el modelo de diseño
   d. Métricas para código fuente
   e. Métricas para pruebas de software
   f. Métricas para el mantenimiento

Bibliografía

1. Roger S Pressman. Software Engineering: A Practitioner's Approach, 2004, 6th edition, McGraw-Hill; ISBN-10: 007301933X
2. Ian Sommerville. Software Engineering, 2006, 8th Edition, Addison Wesley; ISBN-10: 0321313798
3. Kent Beck. Extreme Programming Explained: Embrace Change, 2000, Addison-Wesley; ISBN-10: 0201616416
4. Bernd Bruegge and Allen H. Dutoit. Object-Oriented Software Engineering: Using UML, Patterns and Java, 2003, 2nd edition, Prentice Hall; ISBN-10: 0130471100

Matemáticas discretas

Objetivo

Ofrecer al estudiante un panorama general de las Matemáticas que son particularmente útiles a las Ciencias de la Computación. Se inicia presentando las ideas básicas del principio de conteo y el razonamiento combinatorio elemental. A continuación se ofrece una introducción general a la lógica matemática, un estudio riguroso de la teoría de conjuntos, el principio de la inducción matemática y los métodos recursivos. Posteriormente se estudian las relaciones y funciones y se termina con lenguajes y máquinas de estados finitos. El curso no supone conocimientos matemáticos profundos previos y se enfoca principalmente a desarrollar la capacidad del estudiante para resolver problemas.

Contenido

1. Relaciones binarias y gráficas.
    a. Relaciones binarias.
        1) Relaciones y gráficas.
        2) Interpretación matricial.
    b. Clases de relaciones.
        1) Relaciones de equivalencia.
        2) Relaciones de orden total y parcial.
        3) Retículas y conjuntos parcialmente ordenados.
        4) Orden parcial de relaciones de equivalencia.
    c. Gráficas y sus aplicaciones.
    d. Gráficas de subconjuntos.
    e. Gráficas de De Bruijn.
    f. Computación de gráficas con LGraph.
    g. Aplicaciones a ingeniería de software: análisis de datos, dependencias funcionales, etc.

2. Semigrupos, monoides e ideales.
    a. Propiedades fundamentales.
        1) Semigrupos y monoides: Mapeos, orden parcial en semigrupos.
        2) Ideales: Izquierdos, derechos y principales.
        3) Semigrupos: Semigrupos simples y semigrup-0, teorema de Jordan-Holder.
    b. Aplicaciones de semigrupos.
        1) Semigrupos de relaciones binarias.
        2) Semigrupos libres y de transformación: Lenguajes y teoría de máquinas.
        3) Semigrupo de substituciones.

3. Teoría de números y computabilidad.
    a. Números naturales y enteros.
        1) Postulados de Peanoy principio de inducción matemática.
        2) Orden total y buen orden.
        3) Sistema algebraico de los números naturales y enteros.
        4) Algoritmo de la división. Representación de los números enteros.
    b. Números, proporción y geometría.
        1) Números de Fibonacci.
        2) Números de Lucas.
        3) Representaciones geométricas.
    c. Computabilidad
        1) Recursividad.
        2) Cardinalidad del continuo y conjunto de Cantor.

3) Palabras infinitas y computación digital infinita.
4) Curvas de Hilbert y Peano.
5) Generación de números aleatorios: variedades de números aleatorios.

4. Grupos y sus aplicaciones
    a. Axiomas de grupos.
    b. Generadores y gráficas de grupos.
    c. Grupos de permutación.
    d. Grupos de simetría.
    e. Aplicaciones: geometría, cristalografía y códigos.

Bibliografía

1. Semigrupos y aplicaciones a la computación. Sergio V. Chapa Vergara. Depto. de Ingeniería Eléctrica. Sección de Computación. Notas de clase 2000.
2. Introducción a teoría de grupos. Sergio V. Chapa Vergara. Depto. de Ingeniería Eléctrica. Sección de Computación. Notas de clase 2003.
3. Teoría de números y geometría. Sergio V. Chapa Vergara. Depto. de Ingeniería Eléctrica. Secc. de Computación. Notas de clase 2004.
4. Libro de texto: Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, McGraw-Hill Education (ISE Editions); 6a edition 2007, ISBN: 007288008.

Libros complementarios

1. Ralph P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, Pearson Addison Wesley, 5a edition 2003 ISBN: 0201726343.
2. Susanna S. Epp, *Discrete Mathematics with Applications*, Thomson Brooks/Cole, 3a edition 2004, ISBN: 0534359450.
3. Judith L. Gersting, *Mathematical structures for Computer Science: A modern approach to Discrete Mathematics*, W. H. Freeman and Company, 6a edition 2006, ISBN: 071676864X.
4. Edgar G. Goodaire y Michael M. Parmenter, *Discrete Mathematics with Graph Theory*, Pearson Prentice Hall, 3a edition 2006, ISBN: 0131679953.
5. Rubin H. Landau, *A first course in Scientific Computing: Symbolic, graphic, and numeric modeling using Maple, Java, and Fortran 90*, Princeton Universtiy Press, 1a edition 2005, ISBN: 0691121834.
6. Edward A. Bender, *Mathematics for algorithm and systems analysis*, Dover Publications, 1a edición 2005, ISBN: 0486442500.

Programación Orientada a Objetos

Objetivo

Dar un repaso a los conceptos de la Programación Orientada a Objetos para obtener un nivel básico y avanzado de programación. Se hace énfasis en el lenguaje Java. El curso inicia con el desarrollo de aplicaciones de consola pasando por aplicaciones de escritorio mediante interfaces gráficas hasta llegar a la programación de aplicaciones vía web.

Contenido

1. Introducción
    a. Características del Enfoque de Orientación a Objetos
    b. Características generales de Java

c. Creación, compilación y ejecución de programas

2. Aspectos fundamentales de Java
    a. Tipos de datos
    b. Clases
    c. Objetos, miembros y referencias
    d. Métodos
    e. Miembros estáticos
    f. Método main
    g. Inicializadores estáticos
    h. Clases - Otros aspectos
    i. Ámbito de las variables
    j. Recogida de basura
    k. Sobrecarga de métodos
    l. La referencia this
    m. La referencia null
    n. Ocultamiento de variables
    o. El API de Java

3. Particularidades de Java
    a. La clase String
    b. Cláusula package
    c. Modificadores de acceso
    d. Herencia
        1. Redefinición de métodos. El uso de super
        2. Inicialización de clases derivadas
        3. El modificador de acceso protected
        4. Up-casting y Down-casting
        5. Operador cast
    e. La clase Object
    f. La cláusula final
    g. Excepciones
        1. Generación de excepciones
        2. Captura de excepciones
        3. Cláusula finally
    h. Clases abstractas
        1. Declaración e implementación de métodos abstractos
        2. Referencias y objetos abstractos
    i. Interfaces
        1. Referencias a Interfaces
        2. Agrupaciones de constantes
    j. Clases empotradas
    k. Threads
        1. La Clase Thread
        2. Interface Runnable

4. Temas Avanzados
    a. Documentación de código
    b. Clase Vector
    c. Entrada y Salida
        1. InputStream y OutputStream
        2. Readers y Writers
        3. Serialización
    d. Tokenizing
        1. Clase StreamTokenizer

2. Clase StringTokenizer

e. Colecciones
    1. Interfaces
    2. La interface Collection
    3. La interface Set
    4. La interface List
    5. Interface Map

5. Swing
    a. Crear una aplicación básica
    b. Clasificación de componentes gráficos en Swing
    c. Manejo de Eventos
        6. Adaptadores y clases internas para manejo de eventos
        7. Threads y el manejo de eventos
    d. Controladores de distribución

6. JDBC
    a. Creación de una tabla de una base de datos
    b. Extracción de información de una tabla
    c. Conectar un programa Java con una base de datos
    d. La comunicación con una base de datos
    e. Actualización de una base de datos
    f. Consultas a una base de datos
    g. Sentencias preparadas
    h. Metadatos
    i. Transacciones
    j. Procedimientos en tablas
    k. Equivalencia de tipos entre Java y SQL
    l. Excepciones JDBC

7. JavaBeans
    a. Conceptos y propiedades
    b. La información de un Bean
    c. Propiedades de un Bean
        8. Propiedades compartidas
    d. Eventos

8. Servlets
    a. Clasificación de servlets
    b. La interface servlet
    c. Estructura básica de un servlet
    d. Instalación de las clases servlet
    e. Implementación de un servlet

9. Java Server Pages
    a. Directivas
    b. Estructura
    c. Sesiones
    d. JavaBeans

Bibliografía

1. Thinking in Java, 4th Edition. Bruce Eckel http://mindview.net/Books/TIJ4
2. Introducción a la Programación con Java. Un enfoque Orientado a Objetos.

Arnow, D., G. Weiss, AddisonWesley, 2000.
3. Cómo programar en Java. H. M. Deitel y P. J. Deitel. Editorial Prentice Hall. 2005

Sistemas operativos

Objetivo

Este curso aborda el diseño y la implementación de un sistema operativo: el software maestro que administra y controla los recursos tantos físicos como lógicos de una computadora.

En este curso se estudia 1) las características de los dispositivos físicos, tales como: el procesador, la memoria principal, los dispositivos secundarios (terminales, discos, red, etc.) y 2) el diseño, los problemas y las técnicas de implementación de los componentes de software principales tales como: el núcleo de multiprogramación del procesador (procesos), el sistema de archivos, el módulo de sincronización y de comunicación entre procesos, el módulo de administración de memoria virtual, las técnicas de reservación/liberación dinámica de memoria, el sistema de entrada/salida de datos, la reservación/liberación de recursos.

Contenido

1. Introducción
    a. Organización básica de un sistema de cómputo: procesador, memoria, cache, disco, etc.
    b. Funciones y organización de un sistema operativo
    c. c) Historia del desarrollo de sistemas operativos: introducción a los conceptos de proceso, programa, sistemas batch, sistemas monousuario, sistemas multiprogramados, sistema de tiempo compartido.

2. Programación a nivel máquina
    a. Introducción
    b. Flujo de control
    c. Procedimientos
    d. Datos estructurados

3. Jerarquía de memoria y memoria cache

4. Cadena de producción de programa: compiladores, cargadores y encadenadores de módulos.

5. Núcleo de multiprogramación del procesador
    a. Programas y procesos
    b. Principios de un núcleo de procesos
    c. Programación de hilos (”Threads”)

6. Mecanismos de sincronización entre procesos
    a. Monitores y semáforos
    b. Estudio de problemas básicos: productores/consumidores, filósofos, problema del Barbero durmiente, etc.

7. Memoria virtual de un proceso
    a. Noción de memoria virtual: segmentación y paginación
    b. Sistema de memoria P6/Linux

c. Reservación/liberación dinámica de memoria

8. Comunicaciones entre procesos
    a. Comunicación por mensajes
    b. Memoria compartida
    c. Puertos de comunicaciones y comunicación por ”sockets”

9. Administración de la memoria secundaria: sistemas de archivos
    a. Sistema de entrada/salida de datos desde/hacia dispositivos físicos
    b. Formateo físico de un dispositivo de memoria secundaria (disco duro, disco flexible, etc.)
    c. Organización de un sistema de archivos

10. Comunicación entre sistemas operativos
    a. Redes locales (LAN) y a gran distancia (WAN)
    b. Internet: direcciones y DNS
    c. Comunicaciones en Internet

Bibliografía

1. Andrew S. Tanenbaum and Albert S. Woodhull, “Operating Systems Design and Implementation”, 3rd Edition, Prentice Hall, 2006.
2. Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, “Operating System Concepts”, John Wiley & Sons, 2005.
3. Sistemas Operativos Stalling, William Ed. Prentice-Hall

Base de datos

Objetivo

En este curso se presentan diversos modelos de datos que son abstracciones matemáticas para representar la información del mundo real en datos y conocimiento. El curso cubre también los aspectos de la organización física de los datos, con detalles de implantación para cada uno de los modelos lógicos. Los diversos modelos son unificados mediante el modelo ente-vinculo de Chen que incorpora importante información semántica correspondiente al mundo real. Tomando como punto de partida el modelo de Chen, se tratan los modelos semánticos de datos que incluyen técnicas de Representación de Conocimiento. Finalmente, considerando la corriente de extender los modelos basados en entidades y abstracción en base de datos, tratamos el enfoque Orientado a Objetos.

Contenido

1. Introducción a las bases de datos.
    a. Retrospectiva y perspectiva del desarrollo de las bases de datos.
        1. El enfoque CODASYL.
        2. El programador como navegador.
        3. Tecnología y sistemas manejadores de bases de datos.
        4. La perspectiva de las bases de datos hacia el siglo XXI.
    b. b) Representación y semántica.
        1. Representación basada en entidades.
        2. Abstracción de datos: agregación y generalización.
        3. El modelo semántico de datos de Chen: Entidad-Asociación.
        4. El modelo relacional de datos de Codd.
        5. El modelo orientado a objetos.

2. El modelo relacional de datos
    a. Esquemas de relación y formas normales.
        1. Esquema y estructuras de relación.
        2. Dependencias funcionales y llaves.
        3. Formas normales.
    b. Lenguajes del modelo relacional.
        1. Algebra relacional.
        2. Arboles de consulta
        3. Calculo relacional de tuplas y de dominios.
    c. Completitud relacional.
        1. Equivalencias de lenguajes: Algebraico vs cálculo de tuplas vs cálculo de dominios.
        2. Expresiones Alpha.

3. Diseño de bases de datos.
    a. El proceso de diseño de base de datos.
        1. Un panorama de la metodología DATA1.
        2. Recolección y análisis de requerimientos.
        3. Base de datos, meta base de datos y diseño.
        4. Calidad del diseño.
    b. El modelo entidad-vinculo-extendido.
        1. El modelo estructural.
        2. Problemas y ejemplos.
        3. Extensiones tipos de: entidades, generalizaciones y asociaciones.
    c. Teoría de diseño.
        1. Dependencias funcionales.
        2. Axiomas de inferencia.
        3. Axiomas de Amstrong.
        4. Secuencias de derivación y gráficas dirigidas acíclicas.

4. Lenguajes y operaciones
    a. Lenguajes EVEX
        1. EVEX como lenguaje visual de consulta.
        2. Algebra ERC+.
        3. SQL/ER
    b. Lenguaje visual de dominios.
        1. Query-by-example
        2. QBE/SQL.
        3. Query basado en formas.
    c. Lenguaje LIDA
        1. Lenguaje de flujo de datos.
        2. LIDA/SQL
        3. Modelo para base de datos paralela.
    d. Aplicaciones PosgresSQL.

5. Base de datos geográficas.
    a. El modelo objeto-relacional/entidad-vinculo
        1. Modelación de entidades y objetos geométricos.
        2. Representación y semántica de datos en sistemas geográficos.
    b. Consultas espaciales y temáticas.
        1. Consultas temáticas.
        2. Consultas espaciales.
        3. Análisis estadístico.
    c. Análisis de datos espaciales.
        1. Operaciones topológicas-geométricas.

2. Interpolación espacial.
3. Kriging.
4. Mínimos cuadrados.

d. Aplicación: Grass y Postgis.

6. Base de datos activa.
   a. Análisis y modelo.
      1. Modelo de conocimiento: Evento, condición, acción.
      2. Herramienta de descripción gráfica.
   b. Arquitectura de sistemas de base de datos activas.
      1. Grados de integración.
      2. Arquitectura cliente-servidor.
      3. Especificación de reglas.
      4. Procesamiento de reglas.

Bibliografía

1. C.J. Date. INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS, VOL. I, 6TA. EDICION, ADDISON WESLEY.
2. H. S. KORTH Y A. SILBERSCHATZ, FUNDAMENTOS DE BASES DE DATOS. MC GRAW HILL.
3. Ramez Elmasri; Shamkant Navathe; Alfredo Goni Sarriguren; Arturo Jaime Elizondo; Tomas Pérez Fernández. FUNDAMENTOS DE SISTEMAS DE BASES DE DATOS. Pearson Educacion. 3a edition ISBN: 8478290516.
4. Garcia-Molina, Hector; Ullman, Jeffrey D.; Widom, Jennifer D. (Prentice Hall) DATABASE SYSTEMS. Stanford. ISBN: 0130980439.
5. Jeffrey D. Ullman - Jennifer Widom. INTRODUCCION A LAS BASES DE DATOS Prentice Hall. ISBN: 9701702565.
6. Toby J. Teorey, Sam S. Lightstone, and Tom Nadeau. Database Modeling and Design: Logical Design. Morgan Kaufmann; 4 edition (September 6, 2005). ISBN: 0126853525
7. Ozsu, M. Tamer; Valduriez, Patrick. PRINCIPLES OF DISTRIBUTED DATA BASE SYSTEMS. 2a edition (1997). Prentice-Hall. ISBN: 0136597076.
8. Alex Kriegel and Boris M. Trukhnov. SQL Bible. Editorial: Wiley; edition (April 1, 2003) ISBN: 0764525840
9. C.J. Date. DATABASE IN DEPTH. O'REILLY & ASSOCIATES. ISBN: 0596100124. (May 2005).
10. Houlette. FUNDAMENTOS DE SQL. 1a edición. Editorial McGraw-Hill. 2003 ISBN: 9701038959.
11. Larry Ullman. GUIA DE APRENDIZAJE: MYSQL. Pearson Education. 2003. ISBN: 8420538434.
12. Taylor, Allen G. SQL FOR DUMMIES. WILEY & Sons. 2003. ISBN: 0764540750. (Ag 2003).

Lenguajes de Programación

Objetivo

Que el alumno conozca los conceptos más importantes asociados al diseño de lenguajes de programación, así como los principales paradigmas de programación, para permitirle realizar una evaluación critica de los lenguajes existentes y futuros, y desarrollar criterios para elegir el lenguaje más apropiado para resolver un determinado problema de programación.

Contenido

1. Introducción
2. Computabilidad
3. Descripción de sintaxis y semántica
4. Análisis léxico y sintáctico
5. Tipos de variables y tipos de inferencia
6. Funciones, recursión, listas
7. Administración de almacenamiento
8. Control en lenguajes secuenciales
9. Abstracción de datos y modularidad
10. Objetos y eficiencia en tiempo de ejecución
11. Concurrencia y programación distribuida
12. El paradigma de programación funcional
13. El paradigma de programación orientada a Objetos
14. El paradigma de programación lógica

Bibliografía

1. John Mitchell. Concepts in Programming Language. Cambridge University Press, 2003. ISBN: 0-521-78098-5.
2. Robert W. Sebesta. Concepts of Programming Languages 8a edition (18 de April de 2007). Addison Wesley. ISBN: 0321330250.
3. R. Kent Dybvig. The Scheme Programming Language: Third Edition. 3a edition (1 de October de 2003). The MIT Press. ISBN: 0262541483.
4. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi. How to Design Programs: An Introduction to Programming and Computing (Hardcover). The MIT Press (12 de Febrero 12 2001). ISBN-10: 0262062186.
5. Daniel P. Friedman, Matthias Felleisen. The Little Schemer - 4th Edition. The MIT Press. ISBN 0262560992.
6. The Little Schemer - 4th Edition ISBN 0262560992 Daniel P. Friedman, Matthias Felleisen The MIT Press 4th edition (December 21, 1995)
7. Essentials of Programming Languages 2nd ed. ISBN: 0-262-06217-8 Daniel P. Friedman, Michell Wand, and Christopher T. Haynes The MIT Press 2nd Edition (2001)
8. Concepts in Programming Languages ISBN: 0-521-78098-5 John Mitchell Cambridge University Press, 2003
9. Concepts of Programming Languages (7th Edition) ISBN: 0321330250 Robert W. Sebesta Publisher: Addison Wesley; 7 edition (April 22, 2005)
10. The Scheme Programming Language: Third Edition ISBN: 0262541483 R. Kent Dybvig The MIT Press 3rd edition (October 1, 2003)
11. Programming Ruby: The Pragmatic Programmers' Guide, Second Edition ISBN: 0974514055 Dave Thomas, Chad Fowler, Andy Hunt Pragmatic Bookshelf 2nd edition (October 1, 2004)
12. Structure and Interpretation of Computer Programs - 2nd Edition (MIT Electrical Engineering and Computer Science) ISBN-10: 0262692201 ISBN-13: 978-0262692205 Harold Abelson , Gerald Jay Sussman The MIT Press; 2 edition (July 25, 1996)
13. How to Design Programs: An Introduction to Programming and Computing (Hardcover) ISBN-10: 0262062186 ISBN-13: 978-0262062183 Matthias Felleisen , Robert Bruce Findler , Matthew Flatt , Shriram Krishnamurthi The MIT Press (February 12, 2001)

C. Contenido desarrollado de los cursos formativos y de especialidad

Aritmética computacional

Objetivo

Estudiar los métodos, algoritmos y técnicas de mejora de desempeño necesarias para obtener implementaciones eficientes de operaciones aritméticas en sistemas computacionales con recursos restringidos y en dispositivos de hardware

reconfigurable FPGAs. Los conceptos y técnicas a ser revisados en este curso harán un énfasis especial en algoritmos de la aritmética de campos finitos y su correspondiente implementación en dispositivos FPGAs.

Contenido

1. Introducción a los dispositivos modernos de hardware reconfigurable.
2. Teoría Elemental de Números: definiciones y teoremas, números primos y compuestos, teorema fundamental de la aritmética, máximo común divisor, etc.
3. Definiciones de campos finitos primos (GF(P)).
4. Campos finitos binarios (GF(2m)).
5. Algoritmos Eficientes para aritmética GF(P).
   a. Suma y substracción modular.
   b. Multiplicación Modular y operación de elevar al cuadrado.
   c. Inversión modular.
   d. Pruebas de Primalidad.
   e. Exponenciación Modular. Métodos Binarios LSB-First y MSB-Fist. Métodos M-arios.
6. Técnicas alternativas.
7. Aplicaciones a Teoría de Códigos.
   a. Introducción al algoritmo de RSA.
   b. Curvas elípticas sobre campos finitos.
   c. Introducción a códigos Reed-Solomon.

Bibliografía

1. Francisco Rodriguez-Henríquez, N.A. Saqib, Arturo Diaz Perez and Cetin Kaya Koc¸, “Cryptographic Algorithms on Reconfigurable Hardware´´, Springer First Edition, November 2006, 362 pages, ISBN: 0387338837.
2. Handbook of Applied Cryptography, Menezes, Oorschot, Vanstone. CRC Press, New York, fifth edition (2001) Portal Internet del libro.
3. The Art of Computing Programming, Donald E Knuth, Addison-Wesley, Boston, third edition (2001) Portal Internet del Libro.
4. C. K. Koc. High-Speed RSA Implementation. TR 201, RSA Laboratories, 73 pages, November 1994.
5. Elementary Number Theory and its applications, Fourth Edition, K. Rosen, AT&T Bell Laboratories, 2000. Portal Internet del libro.
6. Presentaciones de Xilinx sobre diseños en FPGAs.

Bioinformática

Objetivo

El objetivo de este curso es que el estudiante adquiera los conocimientos y habilidades necesarios para efectuar el análisis computacional de datos biológicos, con el fin de enunciar predicciones funcionales o estructurales sobre los mismos. Para ello durante el curso se abordarán las principales áreas de la Bioinformática: bases de datos biológicas, alineamiento de secuencias, descubrimiento de genes y promotores, filogenética molecular, bioinformática estructural, genómica y proteómica. Simultáneamente, se analizarán algunas de las principales aplicaciones de la Bioinformática.

Contenido

1. Introducción a la Bioinformática
    a. Panorama histórico de la Bioinformática
    b. Análisis de secuencias de proteínas
    c. Análisis de secuencias de DNA
    d. Análisis de secuencias de RNA
    e. Regiones de codificación del DNA
    f. Genómica y Proteómica

2. Bases de datos (BD) biológicas
    a. Tipos de BD biológicas
    b. Problemas en las BD biológicas
    c. Recuperación de información de BD biológicas

3. Alineamiento de secuencias
    a. Alineamiento de pares
    b. Búsqueda de similitud en BD
    c. Alineamiento múltiple de secuencias
    d. Perfiles y modelos ocultos de Markov
    e. Descubrimiento de motivos y dominios de proteínas

4. Descubrimiento de genes y promotores
    a. Descubrimiento de genes en Procariotas
    b. Descubrimiento de genes en Eucariotas
    c. Descubrimiento de promotores en Procariotas
    d. Descubrimiento de promotores en Eucariotas

5. Filogenética molecular
    a. Conceptos básicos de filogenética
    b. Árboles filogenéticos
    c. Métodos de construcción de árboles filogenéticos

6. Bioinformática estructural
    a. Fundamentos de estructura de proteínas
    b. Visualización, comparación y clasificación de estructura de proteínas
    c. Predicción de estructura de proteínas
    d. Predicción de estructura de RNA

7. Genómica y proteómica
    a. Cartografía genética
    b. Ensamble genómico
    c. Genómica funcional
    d. Proteómica

Bibliografía

1. Neil C. Jones and Pavel A. Pevzner. *An Introduction to Bioinformatics Algorithms*. The MIT Press; 1st edition (August 1, 2004), ISBN-10: 0262101068.
2. Cynthia Gibas and Per Jambeck. *Developing Bioinformatics Computer Skills*. O'Reilly Media, Inc., 1st edition (April 15, 2001), ISBN-10: 1565926641.
3. Jin Xiong. *Essential Bioinformatics* Cambridge University Press, 1st edition (March 13, 2006), ISBN-10: 0521600820.
4. A. Malcolm Campbell and Laurie J. Heyer. *Discovering Genomics, Proteomics and Bioinformatics*. Benjamin Cummings, 2nd edition (March 12, 2006), ISBN-10: 0805382194.

5. Arthur M. Lesk. *Introduction to Bioinformatics*. Oxford University Press, USA, 3rd edition (June 2, 2008), ISBN-10: 0199208042.
6. Dan Gusfield. *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1st edition (May 28, 1997), ISBN-10: 0521585198.

Códigos y Criptografía

Objetivo

Estudiar algunos de los principales métodos, algoritmos, técnicas y herramientas necesarias para la implementación de aplicaciones criptográficas y de seguridad de datos. El contenido del curso está orientado a la resolución eficiente del problema de cómo establecer una comunicación segura entre dos o más entidades de manera tal que se garantice un alto grado de confidencialidad, integridad y autenticidad en los datos y documentos intercambiados. El contenido del curso incluye el estudio de diversos aspectos teóricos relacionados con los algoritmos y esquemas criptográficos, así como la aplicación de dichas técnicas para resolver problemas prácticos en el envío de datos y comunicación de manera segura.

Contenido

1. Introducción a Criptosistemas clásicos
    a. Definición de comunicación segura,
    b. Ataques a criptosistemas.
    c. Técnicas y algoritmos clásicos en criptografía moderna.
    d. Principales esquemas criptográficos.
2. Criptografía de llave secreta.
    a. DES. AES. RC4.
    b. Modos de operación.
3. Criptografía de llave pública
    a. Funciones de solo ida y sus pasadizos secretos (trapdoors).
    b. Criptosistemas de clave pública.
    c. RSA, Diffie-Hellman, El Gamal
    d. Criptosistemas de curvas elípticas.
4. Firmas digitales y autenticación
    a. Funciones de digestión de mensajes (Hash functions).
    b. Firmas digitales.
    c. Protocolos de autenticación.
    d. Protocolos: dinero digital.
    e. Protocolos de identificación y autenticación.
5. Protocolos de Seguridad para autenticación.
6. Firmas a ciegas y sus aplicaciones
    a. Elecciones electrónicas
    b. Monedero digital

Bibliografía

1. Francisco Rodriguez-Henriquez, N.A. Saqib, Arturo Diaz Perez and Cetin Kaya Koc¸,"Cryptographic Algorithms on Reconfigurable Hardware´´, Springer First Edition, November 2006, 362 pages, ISBN: 0387338837.
2. W. Trappe&L. C.Washington. Introduction to Cryptography with Coding Theory, Prentice- Hall, 2002. ISBN: 0-13-061814-4.
3. Guide to Elliptic Curve Cryptography, D. Hankerson, A. Menezes, S. Vanstone, Springer- Verlag, 2004. ISBN 0-387-95273-X

4. International Workshop on Cryptographic Hardware and Embedded Systems, CHES 99-CHES 2006.
5. Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition, John Wiley & Sons, 1996 ISBN 0-471-11709-9.
6. Handbook of Applied Cryptography, Menezes, Oorschot, Vanstone. CRC Press, New York, fifth edition (2001) Portal Internet del libro.
7. The Art of Computing Programming, Donald E Knuth, Addison-Wesley, Boston, third edition (2001) Portal Internet del Libro
8. The Design of Rijndael, Joan Daemen, Vincent Rijmen, Springer, Heidelberg, 2002 Portal Internet del Libro
9. Elementary Number Theory and its applications, Fourth Edition, K. Rosen, AT&T Bell Laboratories, 2000. Portal Internet del libro.
10. Jorge Ramió Aguirre, Seguridad Informática y Criptografía, Tercera edición 3.1 2003, disponible en: http://www.criptored.upm.es.

Computabilidad y complejidad

Objetivo

Conocer los diversos paradigmas de computación formal, entre estos las funciones recursivas y las máquinas de Turing. Examinar las limitaciones de la noción de computabilidad y los criterios diversos de clasificación de problemas atendiendo a la complejidad de sus procedimientos de solución. Se presenta a las funciones recursivas siguiendo el enfoque de máquinas de Turing, de programas-while y el puramente formal. Luego se presenta los grados de irresolubilidad para pasar después las jerarqu´ıas temporal y espacial de los problemas tratables. Ya para terminar, presentaremos una colección de problemas completos-NP. Al ´ultimo presentamos la noción de complejidad abstracta debida a Kolmogorov.

Contenido

1. Conceptos básicos
2. Pruebas por contradicción
3. Inducción matemática
4. Enumeración de Cantor
5. Lenguajes formales de programación
6. Funciones computables
7. Máquinas de Turing
8. Tesis de Church
9. Propiedades de cerradura de funciones computables
10. Numerabilidad de las funciones computables
11. Primera lista de ejercicios
12. Primera lista de programas
13. Funciones elementales
14. Jerarquía de Grzegorczyk
15. Función de Ackermann
16. Segunda lista de ejercicios
17. Funciones de apareamiento
18. Función beta de Gödel
19. Universalidad
20. Funciones recursivas
21. Problema de la parada
22. Decidibilidad
23. Teoremas de recursión

24. Teoremas de autorreproducción
25. Virus en programas-while
26. Tercera lista de ejercicios
27. Segunda lista de programas
28. Teorema de Rice
29. Problema de la correspondencia de Post
30. Irresolubilidad de problemas en GLC
31. Algunos otros problemas irresolubles
32. Irresolubilidad en la Aritmética Aritmética de Peano
33. El teorema de Goodstein
34. La conjetura de Catalán
35. Clasificación de problemas irresolubles
36. Computaciones con oráculos
37. Enumerabilidad recursiva relativa a oráculos
38. Ordenes de funciones
39. Algunas clases de funciones
40. Limites inferiores
41. Clases de problemas: Problemas de decisión, problemas de solución
42. Comprobadores y resolvedores
43. Complejidades de tiempo y espacio
44. Jerarquía en espacio
45. Jerarquía en tiempo
46. Algunas relaciones entre clases
47. Jerarquías en clases no-deterministas
48. Teorema de Borodin y consecuencias
49. Clases de problemas
50. Reducibilidades
51. Problemas difíciles y completos en clases polinomiales
52. Algunos problemas principales completos-NP
53. Tercera lista de programas
54. Nociones básicas
55. Problemas de acotación
56. Problemas de acceso
57. Problema de acceso generalizado
58. Algoritmos probabilísticos
59. Expresiones polinomiales nulas
60. Método de Monte-Carlo para probar primicidad
61. Máquinas probabilísticas
62. Máquinas del tipo 1
63. Máquinas-p
64. Computabilidad con mTp’s
65. Algunas inclusiones entre clases
66. Complejidad de Kolmogorov
67. Conjuntos ralos

Bibliografía

1. Aho, Ullman: Foundations of computer science, W.H. Freeman & Co., 1992
2. Aho, Hopcroft, Ullman: The design and analysis of computer algorithms, Addison-Wesley, Reading, MA, 1974
3. Anderson, Turing et al: Mentes y máquinas, en la colección Problemas científicos y filosóficos, Universidad Nacional Autónoma de México, 1970
4. Arbib, Kfoury, Moll: A basis for theoretical computer science, Springer-Verlag, 1981
5. Balcazar. Diaz, Gabarro: Structural complexity I, Springer-Verlag, 1988
6. Barwise: Handbook of mathematical logic, North-Holland, 1977

7. Chaitin: Algorithmic information theory, Cambridge University Press, 1987
8. Cutland: Computability: An introduction to recursive function theory, Cambridge University Press, 1980
9. Davis: Computability and unsolvability, Mc-Graw Hill, 1958
10. Davis: The undecidable, Raven Press, 1965
11. Epstein: Degrees of unsolvability: Structure and theory, Lect. Notes in Comp. Sci. Nr. 759, Springer-Verlag, 1979.
12. Garey, Johnson: Computers and intractability: A guide to the theory of NP-completeness, Freeman, 1979
13. Grzegorczyk: Some classes of recursive functions, Rosprawy matematyczne Nr. 4, IMPAN, Warszawa, 1953
14. Hermes: Aufzahlbarkeit, Entscheidbarkeit, Berechenbarkeit, Springer-Verlag, 1965 (English translation: Enumerability, decidability and computability, Springer-Verlag, 1969, traduzione italiana Enumerabilit´a, decidabilit´a computabilit´a, Boringhieri, 1975)
15. Hinman: Recursion-theoretic hierarchies, Springer-Verlag, 1978

Computación Paralela

Objetivo

El propósito de este curso es discutir técnicas y aplicaciones de la computación paralela y de forma en que se construyen aplicaciones para este paradigma. En este curso nos concentraremos en el uso de varias computadoras que se comunican entre sí ya sea a través de una memoria compartida o mediante el envío de mensajes.

Contenido

1. Computación Paralela
   a. ¿Por qué computación paralela?
   b. Tipos de computadoras paralelas
   c. Fundamentos del diseño de computadoras paralelas

2. Programas paralelos
   a. Aplicaciones como casos de estudio
   b. El proceso de paralelización
   c. Un caso de studio

3. Programación paralela para mejorar el rendimiento
   a. Descomposición
   b. Mapeo
   c. Calendarización
   d. Evaluación del rendimiento
   e. Medidas de eficiencia

4. Programación con memoria compartida
   a. Procesos
   b. Threads

5. Programación paralela mediante intercambio de mensajes
   a. Principios básicos
   b. Herramientas de software
      1) PVM
      2) MPI

6. Estrategias generales de programación paralela
   a. Particionamiento
   b. Divide y vencerás
   c. Aplicaciones tipo pipeline

d. Aplicaciones de tipo síncrono
e. Balance de carga

7. Algoritmos y aplicaciones
   a. Algoritmos de ordenamiento
   b. Algoritmos numéricos
   c. Algoritmos para procesamiento de imágenes
   d. Algoritmos de búsqueda y optimización

Bibliografía

1. Culler, David, Singh, J. P. And Gupta, A. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kauffman Publishers, 1998. ISBN 1-55860-343-3.
2. Foster, Ian T. Designing and Building Parallel Programs. Addisson-Wesley, 1995. ISBN 0-201-57594-9.
3. Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta. An Introduction to Parallel Computing: Design and Analysis of Algorithms (2nd Ed.). Addisson-Wesley 2003. ISBN 0201648652.
4. Pacheco, Peter. Parallel Programming With MPI. Morgan Kaufmann. ISBN 1558603395.
5. Quinn, Michael J. Parallel Programming in C with MPI and OpenMP. McGraw-Hill. 2003. ISBN 0072822562 .
6. Wilkinson, Barry and Allen, Michael. Parallel Programming: Techniques and Aplications Using Networked Workstations and Parallel Computers (2nd Ed.). Prentice-Hall Inc. 2004, ISBN 0131405632.

Cómputo basado en redes

Objetivo

Entender los conceptos fundamentales y desarrollar las habilidades de programación requeridos para la construcción de sistemas distribuidos basados en la arquitectura cliente-servidor de Internet. Estudiar algunas de las tecnologías de Internet incorporadas al lenguaje de programación Java tales como Berkley Sockets, Servlets, Java Server Pages, Remote Method Invocation, CORBA y Enterprise Java Beans las cuales facilitan el desarrollo de aplicaciones distribuidas. Desarrollar prácticas y proyectos de programación que enfaticen conceptos como sistemas abiertos, interoperabilidad, portabilidad, seguridad e integración.

Contenido

1. Arquitectura de Internet
   a) Protocolo de Internet IP
   b) Protocolos de Transporte TCP y UDP
   c) Aplicaciones usando Sockets y DNS
   d) Programación concurrente con Java Threads
   e) Comunicación entre aplicaciones con Java Sockets
2. La WWW como ejemplo de computación cliente/servidor
   a) Introducción a la programación cliente/servidor
   b) Diseño de clientes y servidores
   c) Conceptos básicos de la Web
      1) Localizadores universales de recursos (URL)
      2) Protocolo de transferencia de hipertexto (HTTP)
      3) Lenguaje de marcado de hipertexto (HTML)
      4) Java applets

d) Sistemas multimedia en red
e) Criptografía de llaves públicas y certificados digitales
f) Programación CGI del Servidor

3. Construcción de aplicaciones Web
   a) Arquitectura de tres partes (three-tier architecture)
   b) Acceso a bases de datos con Java Database Connectivity (JDBC)
   c) Programación de servidores con Java Servlets
   d) Programación de guiones (scripting) con Java Server Pages (JSP)
4. Sistemas de objetos distribuidos
   a) Llamadas a procedimientos remotos
   b) Modelo de objetos distribuidos de Java (Java Remote Method Invocation, RMI)
   c) Modelo de componentes distribuidos de Java (Enterprise Java Beans)
5. Sistemas de agentes
   a) Agentes de software
   b) Sistemas de agente único
   c) Sistemas multi-agente
   d) Comercio electrónico
   e) Agentes móviles

Bibliografía

1. Douglas E Comer. El libro de Internet. Prentice-Hall. 1995.
2. Dougals E Comer. Computer Networks and Internets. Prentice-Hall. 2001.
3. David Flanagan, Jim Farley, William Crawford & Kris Magnusson. Java Enterprise in a Nutshell. O RReilly, 1999.
4. Andrew S Tanembaum. Redes de computadoras. Prentice-Hall. 1993.
5. Dawna Trevis Dewire. Client/server computing. Prentice-Hall. 1993.
6. Alex Berson. Client/server architecture. McGraw-Hill. 1992.
7. Michel Morrison et al. Java 1.1. Unleashed. 3th edition. Sams.net Publishing. 1997.
8. Robert Orfali, Dan Harkey & Jeri Edwards. The essential client/server survival guide. 2nd edition. John Wiley. 1996.
9. Robert Orfali, Dan Harkey & Jeri Edwards. The essential distributed objects survival guide. John Wiley. 1996.

Cómputo Móvil

Objetivo

La necesidad de información en cualquier momento y lugar, conjuntamente con el surgimiento de dispositivos de cómputo portátiles y los avances en las tecnologías de comunicación inalámbrica e Internet, han hecho a la Computación Móvil una realidad. Esta tiene como finalidad, el tratamiento automático de información por medio de dispositivos computacionales con capacidad de movilidad y con acceso digital a fuentes de información via una infraestructura de comunicaci ón inalámbrica. Los ambientes de cómputo móvil se
caracterizan por restricciones importantes de recursos y cambios frecuentes en las condiciones de operación lo cual impone desafios que involucran diversas areas de la ciencias computacionales, ingenieríaś computacional, eléctrica y de telecomunicaciones. El objetivo principal de este curso es el de proporcionar un marco general de los fundamentos, métodos y tecnologías para la compresión de problemáticas y el desarrollo de aplicaciones en el área de computación móvil. Adicionalmente a la formación teórica, el curso tendrá una fuerte orientación práctica al desarrollo de aplicaciones en dispositivos móviles, articularmente en teléfonos celulares con soporte para Android.

Contenido

1. Introducción y motivación
2. Fudamentos de cómputo móvil
3. Tecnologías móviles
4. Desarrollo de aplicaciones
5. Administracion de recursos móviles
6. Localización
7. Redes y seguridad

Bibliografía

1. Reza B'Far, Mobile Computing Principles, Cambridge University Press, 2005.
2. Tommi Mikkonen, Programming Mobile Devices, John Wiley and Sons, 2007.
3. K. Pahlavan and Allen H. Levesque, Wireless Information Networks, Wiley-Interscience, Second edition, 2005.
4. Roy Ben Hayun, Java ME on Symbian OS, Wiley, 2009.
5. Ed Burnette, Hello Android: Introducing Google's Mobile Development Platform, Pragmatic Bookshelf, 3a edition.
6. IEEE Computer Magazine.
7. IEEE Transactions on Mobile Computing.

Fundamentos Algebraicos de la Computación

Objetivo

El principal objetivo de este curso es el de introducir los conceptos básicos de las estructuras algebraicas. El conocimiento de estas estructuras es indispensable para el estudio de las ciencias computacionales modernas, por lo que el material a ser cubierto en este curso sirve como base teórica para el estudio de temas tales como teoría de códigos, criptología, etc.

Contenido

1. Conjuntos, operaciones en conjuntos, relaciones, relaciones de equivalencia, funciones.
2. Propiedades básicas de los enteros, algoritmo extendido de Euclides, Congruencias, resolución lineal de congruencias, teorema chino del residuo, teorema de Fermat y Euler.
4. Grupos, Subgrupos, cosets, teorema de Lagrange, orden de un grupo, subgrupos normales, grupos cocientes, homomorfismos e isomorfismos, enunciado del teorema de Sylow's y aplicaciones.
5. Anillos, dominios enteros y campos por división, ideales, dominios euclideanos, dominios de principio ideal, dominios de factorización única y anillos polinomiales.
6. Campos, espacios vectoriales, transformaciones lineales y matrices, Determinantes, Rango, sistemas de ecuaciones lineales.
7. campos de extensión, campos de separación, caracterización de campos finitos, aplicación a teoría de códigos.

Bibliografía

1. Victor Shoup: A Computational Introduction to Number Theory and Algebra. Cambridge University Press 2005.

2. I. N. Herstein: Topics in Algebra. Wiley 1975.
3. John B. Fraleigh: A First Course in Abstract Algebra. Addison-Wessley 2002

Geometría Computacional

Objetivo

En este curso se estudiarán los conceptos básicos de la Geometría Computacional (GC) así como sus aplicaciones. El material aborda las técnicas necesarias para el diseño y análisis de algoritmos eficientes para resolver problemas en geometría, tales como: cubiertas convexas (*convex hulls*), intersecciones geométricas, diagramas de Voronoi, triangulaciones de Delaunay, estructuras de datos geométricas, etc.

Contenido

1. Introducción a la GC
   a. Qué es la GC
   b. Ejemplo de un problema típico de GC
   c. Limitaciones de la GC
   d. Tendencias de la GC
   e. Dominios de aplicación

2. Cubiertas convexas
   a. Algoritmo de Graham
   b. Algoritmo de Chan
   c. Algoritmo QuickHull

3. Intersección de segmentos de línea
   a. Intersecciones geométricas
   b. Intersección de segmentos de línea
   c. Algoritmo de barrido del plano (plane-sweep)
   d. Representación e intersección de subdivisiones planares

4. Triangulación de polígonos
   a. Grafos planares
   b. El problema de la galería de arte
   c. Partición de un polígono es sus piezas monótonas
   d. Triangulación de polígonos monótonos

5. Programación lineal
   a. Intersección de sub-planos (half-plane)
   b. Algoritmo divide y venceras
   c. Algoritmo determinístico incremental
   d. Algoritmo aleatorio

6. Búsqueda de rango ortogonal
   a. Consultas de rango en una dimensión
   b. Árboles kd
   c. Árboles de rango
   d. Árboles de rango en dimensiones mayores
   e. Conjuntos generales de puntos

7. Localización de puntos en el plano
   a. El método de Kirkpatrick
   b. Descomposiciones trapezoidales
   c. Estructuras de datos para localización de puntos

8. Diagramas de Voronoi
   a. Definición y propiedades básicas
   b. Construcción de diagramas de Voronoi
   c. Algoritmo divide y venceras
   d. Algoritmo aleatorio incremental
   e. Algoritmo de Fortune

9. Arreglos de líneas y dualidad
   a. Cálculo de la discrepancia
   b. Dualidad
   c. Construcción incremental de arreglos
   d. Niveles y discrepancia

10. Triangulación de Delaunay
   a. Triangulación de conjuntos de puntos
   b. Propiedades de la triangulación de Delaunay
   c. Construcción incremental

Bibliografía

1. Mark de Berg, Otfried Cheong, Marc van Kreveld and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition (April 16, 2008), ISBN-10: 3540779736.
2. Joseph O'Rourke. *Computational Geometry in C*. Cambridge University Press; 2000 edition (February 15, 2001), ISBN-10: 0521649765.
3. Jacob E. Goodman and Joseph O'Rourke (editors). *Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, 2nd edition (April 15, 2004), ISBN-10: 1584883014.
4. Franco P. Preparata and Michael Ian Shamos. *Computational Geometry an Introduction*. Springer, (August 6, 1993), ISBN-10: 0387961313.

Graficación

Objetivo

El objetivo del curso es estudiar las diversas técnicas de la literatura para el trazado de una escena y de las formas bi- y/o tridimensionales que la componen, además de las técnicas para manipularla y visualizarla. La herramienta sugerida de trabajo, para desarrollar las tareas del curso, es un sistema de desarrollo de interfaces gráficas (GUI, por sus siglas en inglés) basado en objetos, llamado Qt (www.trolltech.com) y OpenGL (www.opengl.org) o Mesa (www.mesa3d.org) para la construcción y manipulación de escenas tridimensionales.

Contenido

1. Introducción.
   a. Definición y temas que estudia graficación.
   b. Arquitectura para despliegue tipo raster
   c. Marco de trabajo conceptual

2. Algoritmos básicos para trazo de primitivas en 2D
   a. Trazo líneas con el algoritmo incremental de punto medio
   b. Trazo de círculos con el algoritmo incremental de punto medio

c. Rellenado de polígonos.
d. Trazado de fractales con el lenguaje LOGO
e. Primitivas gruesas: líneas, círculos y polígonos.
f. Cortado (clipping).

3. Transformaciones geométricas
   a. Translación, escalamiento, rotación y sesgado (sheared)
   b. Las transformaciones en coordenadas homogéneas.
   c. Composición de transformaciones 2D
   d. Representación matricial de transformaciones 3D
   e. Composición de transformaciones 3D
   f. Las transformaciones como un cambio en el sistema de coordenadas.

4. Visión en 3D
   a. Proyecciones: paralelas, en perspectiva.
   b. Especificación de una vista arbitraria.
   c. Deducción de ecuaciones de las proyecciones geométricas planas.

5. Representación de curvas y superficies
   a. Mallas de polígonos
   b. Curvas cúbicas paramétricas: Hermite, Bézier y B-splines.
   c. Superficies cúbicas paramétricas
   d. Superficies cuádricas.

6. Determinación de la superficie visible
   a. Funciones de dos variables
   b. El algoritmo de buffer z

7. Modelado de sólidos
   a. Operaciones Booleanas regularizadas
   b. Instanciamiento de Primitivas
   c. Representaciones de barrido
   d. Representaciones que particionan el espacio: Descomposición de celdas, enumeración de ocupancia espacial y octrees.
   e. Geometría sólida constructiva.

8. Iluminación y sombreado
   a. Modelos de iluminación
   b. Modelos de sombreado para polígonos.
   c. Sombras

9. Dibujado por trazo de rayo
   a. Algoritmo básico para el trazo de rayos
   b. Cálculo de las intersecciones rayo-superficie

Bibliografía

1. J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes, Computer Graphics: Principles and Practice 2nd edition in C. Addison Wesley
2. Neider, Jackie, Tom Davis, and Mason Woo, OpenGL Programming Guide: The Official Guide to Learning OpenGL Version 2 (5th Edition). Addison-Wesley
3. D. Hearn, M. P. Baker Computer Graphics, C version 2nd edition, Prentice Hall

Inteligencia Artificial

Objetivo

Los trabajos desarrollados en Inteligencia Artificial (IA) versan sobre la ciencia del conocimiento y las máquinas inteligentes. La primera modela y desarrolla estructuras que permiten explicar los mecanismos utilizados por el ser humano para resolver problemas. En cambio, las máquinas inteligentes son sistemas que ejecutan funciones similares a las desarrolladas por un especialista. Estas últimas constituyen las ramas de aplicación de la IA: Resolución de problemas, Sistemas Expertos, Procesamiento de Lenguaje Natural, Aprendizaje automático, Demostración automática de teoremas, Visión, Robótica, etc. Este curso se orienta principalmente a este último enfoque: las aplicaciones de la IA.

Contenido

1. Representación del Conocimiento y mecanismo de inferencia en: Lógica simbólica: Cálculo proposicional
    a) Redes semánticas
    b) Reglas de producción
    c) Objetos estructurados
2. Resolución de problemas
    a) Métodos clásicos en resolución. Descripción de estados y operadores. Procesos de Búsqueda en grafos
    b) Métodos de propagación y de enumeración.
        1) Métodos de resolución por enumeración.
        2) Método del gradiente (escalada).
        3) Programación dinámica.
        4) Coloración optimal.
    c) Heurísticas admisibles. Funciones de evaluación. Algoritmos de búsqueda optima
3. Deducción automática empleando Calculo Proposicional
    a) Algoritmo de unificación de acuerdo con la teoría de la demostración
    b) Encadenamiento hacia adelante y unificación
    c) Principio de resolución
    d) Ejemplos de aplicaciones
4. Sistemas Expertos
    a) Diferentes modelos de razonamiento
        1) Arquitectura de sistemas expertos
        2) Motores de inferencia Cero
        3) Motores de inferencia con variable
    b) Razonamiento aproximado
        1) Origen de la incertidumbre
        2) Tratamiento de la incertidumbre
        3) Ejemplos de SE
5. Aprendizaje automático
    a) Taxonomía del aprendizaje
    b) Aprendizaje Inductivo de conceptos
    c) Ejemplos de sistemas de aprendizaje
6. Procesamiento de lenguaje natural escrito
    a) Diversas técnicas de análisis sintáctico y semántico. ATN, RTN. Descriptores funcionales
    b) Ejemplos de sistemas de comprensión de LN

Bibliografía

1. Stuart Russell y Peter Norvig. Artificial Intelligence: A Modern Approach, 2nd Edition. Prentice Hall. 2003. ISBN: 0137903952.
2. Inteligencia Artificial, 2a Edición Elaine Rich y Kevin Knight Mc Graw Hill, 1994
3. Inteligencia Artificial, 3a Edición Winston, P. H. Addison-Wesley Iberoamericana, 1994
4. Bolc, Leonard, Natural language parsing systems, 1987
5. Brooke, Deborha L., Natural language processor as a universal front end to expert systems, 1983.
6. C.L. Chang, R. Lee. Symbolic Logic and Mechanical Theorem Proving. Academic Press, 1973.
7. Charniak, Eugene, Introduction to artificial intelligence, 1985
8. Clancey, William J. de. Artificial intelligence and learning environments,
9. Farreny, ”Les Syst`emes Experts: Principes et Examples”. Cepadues, eds., 1985
10. N. J. Nilson, Principios de Inteligencia Artificial. Díaz de Santos, 1987.
11. P.H. Winston. Artificial Intelligence (Second Edition). Addison Wesley, Iberoamericana, eds., 1992
12. Thayse, Andre, From natural language processing to logic for expert systems : a logic based approach to artificial intelligence, 1991.

Introducción a la computación evolutiva

Objetivo

En este curso se estudiaran los conceptos básicos de las técnicas más importantes de computación evolutiva, haciendo especial énfasis en los algoritmos genéticos. Inicialmente, se harán un recorrido histórico en el que se resumirán los logros más importantes en torno a la simulación de los procesos evolutivos como una herramienta para el aprendizaje y la optimización. Posteriormente, se analizaran y compararan de manera general los 3 paradigmas principales que se utilizan hoy en día en la computación evolutiva: las estrategias evolutivas, la programación evolutiva y los algoritmos genéticos. En cada caso se abordara su inspiración biológica, su motivación, su funcionamiento y algunas de sus aplicaciones. Finalmente, se estudiara a mayor detalle el funcionamiento, fundamentos teóricos, implementación y operación de los algoritmos genéticos, que es actualmente el paradigma evolutivo más utilizado por los investigadores que trabajan en esta disciplina.

Contenido

1. Técnicas heurísticas
   a) Problemas P y NP
   b) Técnicas clásicas de búsqueda y optimización
   c) Lo que el mundo real demanda
   d) ¿Que es una heurística?
   e) ¿Realmente necesitamos técnicas heurísticas?
   f) Ejemplos de técnicas heurísticas: Búsqueda tabú, recocido simulado, escalando la colina.
2. Nociones de optimización
   a) Optimización Global
   b) Optimización Numérica
   c) Optimización Combinatoria
   d) Espacios de búsqueda convexos y cóncavos

e) Restricciones explicitas e implícitas
f) Restricciones de igualdad y desigualdad
g) Zona factible y no factible

3. Antecedentes históricos
   a) Inspiración biológica
   b) Primeros intentos
   c) Cronología de descubrimientos importantes
4. Paradigmas principales
   a) Estrategias evolutivas
   b) Programación evolutiva
   c) Algoritmo genético
   d) Comparaciones
5. La computación evolutiva en el contexto de la Inteligencia Artificial
   a) Criticas (IA Clásica vs. Técnicas Heurísticas)
   b) ¿Un nuevo paradigma?
6. Terminología biológica vs. Terminología usada en la computación evolutiva: glosario básico y comparaciones
7. Algoritmos genéticos.
   a) Generalidades: Definición. Componentes básicos. Funcionamiento
   b) Representación: Binaria, códigos de gray real, programación genética, algoritmos genéticos desordenados (Messy GAs), otras propuestas.
   c) Selección: Proporcional, torneo, estado uniforme, uso de jerarquías.
   d) Cruza: Importancia, un punto, dos puntos, uniforme, casos especiales.
   e) Mutación: Importancia, forma básica, no uniforme, casos especiales
   f) Función de aptitud. Definición. Uso de “cajas negras”. Manejo de restricciones usando funciones de penalización. Ejemplos
   g) Ajuste de parámetros: Estudios empíricos. Auto-adaptación
   h) Implementación: Software propio, software de dominio público, sistemas comerciales.
   i) Operadores avanzados: Diploides y dominancia. Inversión. Micro-operadores: segregación, traslocación, duplicación y borrado.
   j) Teoría: Teorema de los esquemas. Modelos exactos. Teoría de convergencia. No Free Lunch Theorems. Criticas. Decepción
   k) ¿Cuándo aplicar un algoritmo genético? Limitaciones. Ventajas. Áreas de aplicación
8. Áreas abiertas de investigación. Inspiración biológica. Paralelismo. Teoría. Representación. Operadores. Co-evolución.
9. Algoritmos culturales. Ajuste de parámetros. Auto-adaptación
10. Otras
11. Temas avanzados (opcionales) Funciones multimodales. Nichos. Repartición de aptitud. Ejemplos
12. Funciones con objetivos múltiples. Técnicas básicas
13. Manejo de restricciones. Función de penalización. Pena de muerte. Separación de objetivos y restricciones. Uso de representaciones y operadores especiales. Otras propuestas

Bibliografía

1. A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer, Berlin, 2003, ISBN 3-540-40184-9 (libro de texto).
2. David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.
3. David Corne, Marco Dorigo&Fred Glover (editores), New Ideas in Optimization, McGraw- Hill, London, 1999.
4. Sadiq M. Sait & Habib Youssef, Iterative Computer Algorithms with Applications in Engineering, IEEE Computer Society, Los Alamitos, California, 1999.

5. Melanie Mitchell, An Introduction to Genetic Algorithms, MIT Press, Cambridge, Massachusetts, 1996.
6. Zbigniew Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer-Verlag, Second Edition, 1992.
7. John H. Holland, Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, MIT Press, Cambridge, Massachusetts, Second Edition, 1992.
8. David B. Fogel, Evolutionary Computation. Toward a New Philosophy of Machine Intelligence, The Institute of Electrical and Electronic Engineers, New York, 1995.
9. Thomas Back, Evolutionary Algorithms in Theory and Practice, Oxford University Press, New York, 1996.
10. David B. Fogel, Evolutionary Computation: The Fossil Record, The Institute of Electrical and Electronic Engineers, New York, 1998.
11. John R. Koza, Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, Massachusetts, 1992.
12. Colin B. Reeves (editor), Modern Heuristic Techniques for Combinatorial Problems, John Wiley & Sons, Great Britain, 1993.
13. Zbigniew Michalewicz & David B. Fogel, How to Solve It: Modern Heuristics, Springer, Berlin, 2000.

Introducción al Cómputo Reconfigurable

Objetivo

El curso presenta los fundamentos y formalismo del Cómputo Reconfigurable así como metodologías y técnicas de cómo diseñar y aplicar arquitecturas bajo este enfoque en la solución de problemas que involucran una alta demanda computacional. Se analizan las aspectos centrales para implementar cálculos bajo este paradigma y se abren perspectivas de cómo construir sistemas de cómputo cuya arquitectura, maximalmente concurrente, refleje la naturaleza estructural de su descripción algorítmica. Adicionalmente en el curso se analizan y estudian técnicas y herramientas de diseño, análisis y optimización de sistemas de cómputo reconfigurable basado en tecnología FPGA. Finalmente se estudia y analiza el estado del arte bajo la perspectiva hardware y software a fin de identificar líneas potenciales de investigación.

Contenido

1. Introducción
   a. Convergencia digital
   b. Cómputo físico
   c. Modelado de sistemas
   d. Diseño asistido por computadora

2. Tendencias tecnológicas
   a. Evolución de la tecnología
   b. Escala de integración
   c. Crisis y paradigmas de diseño
   d. Hardware reconfigurable: tecnología FPGA

3. Principios de diseño de computadoras
   a. Modelo computacional
   b. Sistemas basados en microprocesadores
   c. Abstracción hardware/software
   d. Modelos de procesamiento paralelo

4. Cómputo reconfigurable
    a. Fundamentos
    b. Localidad funcional y temporal
    c. Cómputo espacial
    d. Sistemas de cómputo reconfigurable
    e. Clasificación de arquitecturas reconfigurables
    f. Reconfiguración estática y dinámica parcial

5. Implementación de aplicaciones computacionalmente intensivas
    a. Relación entre arquitectura de computadoras y diseño de algoritmos
    b. Naturaleza estructural de un algoritmo
    c. Dependencias de datos y control
    d. Representaciones topológicas abstractas
    e. Problemáticas

6. Métricas de comparación de plataformas computacionales
    a. Fundamentos: chip ≠ chip
    b. Algunas métricas de desempeño
    c. Normalización de métricas
    d. Caracterización de recursos de implementación
    e. Densidad computacional

7. Diseño a nivel de sistema y síntesis de alto nivel
    a. Niveles de abstracción
    b. Síntesis de alto nivel
    c. Optimización multiobjetivo
    d. Síntesis lógica y mapeo tecnológico

8. Aplicaciones y casos de estudio
    a. Arquitecturas para la percepción visual. Procesamiento de imágenes basado en ventanas
    b. Sistemas de procesamiento de señal. Procesamiento de señales adaptivo
    c. Arquitecturas para compresión/descompresión de información
    d. Implementación de sistemas biológicamente inspirados
    e. Modelos de percepción visual: detección de movimiento y segmentación de imágenes

Bibliografía

1. Ramancgandran Vaidyanathan, and Jerry L. Trahan, Dynamic Reconfiguration Architectures and Algorithms, Kluwer Academic Publishers, Series in Computer Science, 2004.
2. Nikolaos S. Voros and Konstantinos Masselos, System Level Design of Reconfigurable Systems on Chip, Springer Verlag, 2005.
3. Hesham el-Rewini, and Mostafa Abd-El-Bar, Advanced Computer Architecture and Parallel
4. Processing, Wiley, 2005.
5. Scott Hauck, Andre DeHon, Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation (Systems on Silicon), Morgan Kaufmann (November 9, 2007)
6. Christophe Bobda, Introduction to Reconfigurable Computing: Architectures, Algorithms, and Applications, Springer; 1 edition (November 9, 2007)

7. Morris Mano, M. and Kime, Charles R. Logic and Computer Design Fundamentals: Second Edition. Prentice Hall. New Jersey. ISBN 0-13-012468-0.
8. Rodriguez-Henriquez, F., Saqib, N., Diaz-Perez, A. and Koc, C. Cryptographic Algorithms on Reconfigurable Hardware. Springer. 2006. ISBN 0-387-33883-7.
9. Wolfe, Wayne. FPGA-Based System Design. Prentice-Hall, New Jersey. 2004. ISBN 0- 13-142461-0.

Introducción a la Teoría de la Probabilidad

Objetivo

Se estudian los principios básicos de la teoría de la probabilidad junto con algunas aplicaciones de ´esta. Se familiariza al estudiante con distintas distribuciones de probabilidad las cuales son utilizadas frecuentemente en otras áreas de las ciencias computacionales.

Contenido

1. Probabilidad clásica, espacio muestral, eventos, probabilidad condicional, independencia,
2. regla de Bayes y aplicaciones.
3. variable discreta aleatoria, esperanza, momentos, varianza, desigualdad de Chebyshev. Dsitribuciones de probabilidad binomial, Poisson, geométrica, e hipergeometrica. Aproximación de Poisson a la distribución binomial.
4. Variable aleatoria continua. Densidades normales y exponenciales.
5. Distribución multivariable, densidad normal multivariable.
6. Teorema del límite central, caminatas aleatorias y proceso de Poisson.

Bibliografía

1. Hoel, Port and Stone. Introduction to Probability Theory.
2. Hoel, Port and Stone. Introduction to Stochastic Process.
3. Feller. An Introduction to Probability Theory and its Application. Volume-1 and 2.

Lógica y Base de Datos

Objetivo

Base de datos y lógica se interrelacionan como fundamentación y como un enfoque de sistema deductivo en la solución de problemas visto como consultas en la base de datos. Los primeros temas consisten en una revisión de la lógica de predicados de primer orden y la demostración automática de teoremas para, posteriormente, encontrar una fundamentación para las bases de datos en el lenguaje relacional, restricciones de integridad y diseño de base de datos. La segunda parte es usar la aproximación de la lógica como sistema deductivo a las bases de datos con aplicación de algunos sistemas como CORAL a un sistema relacional Posgress. Algunos temas selectos se presentan con la finalidad de una investigación posterior más profunda.

Contenido

1. Introducción a la lógica matemática
   a) Lógica proposicional
   b) Lógica de predicados de primer orden
   c) Demostración automática de teoremas

2. La lógica como fundamento a base de datos
    a) El modelo relacional y lógica de predicados de primer orden
    b) Restricciones de identidad y lógica
    c) Suposición de un mundo cerrado en base de datos
    d) Lógica para descripción de datos
3. Fundamentación lógica en el modelo entidad-relación
    a) Estructura jerárquica de la lógica de predicados
    b) Restricciones estáticas
    c) Dinámica de las bases de datos
    d) Generalización de las restricciones relacionales
    e) Teoría de normalización de base de datos

4. Base de datos deductivas
    a) Base de datos deductivas y base de datos lógicas
    b) Base de datos deductivas definidas
    c) Base de datos deductivas indefinidas
    d) Base de datos lógicas
5. Sistemas de base de datos deductivas
    a) Sistema R: aproximación relacional a bases de datos
    b) Sistema CORAL
    c) Estrategias para procesamiento de consultas recursivas
6. Temas selectos en Base de Datos
    a) Lógica no monotonía
    b) Suposición del mundo cerrado vs negación bajo falla
    c) Retículas y unificación

Bibliografía

1. S. V. Chapa Vergara; *"Lógica clásica"*; Departamento de Ingeniería Eléctrica, CINVESTAV, México, 2000.
2. S. V. Chapa Vergara; *"Lógica e inteligencia artificial"* Departamento de Ingeniería Eléctrica, CINVESTAV, México, 2000.
3. S. V. Chapa Vergara; *"Lógica y base de datos"* Departamento de Ingeniería Eléctrica, CINVESTAV, México, 2000.
4. B. Thalheim; *"Entity-Relationship"*; Springer-Verlag, 2000.
5. Minker J.; *"Foundations of Deductive Databases and Logic Programming"*, Morgan Kaufmann, Los Altos, CA, 1988.
6. J. Myloupolus, M. L. Brodie; *"Artificial Intelligence and Databases"*; Morgan Kaufmann, Los Altos, CA, 1988.

Lógica matemática

Objetivo

Presentar los conocimientos básicos de Lógica Matemática para poder trabajar en Programación Lógica, en Inteligencia Artificial y, en general, en Ciencias de la Computación.

Contenido

1. Algebras booleanas
    a) Órdenes parciales y retículos
    b) Retículos booleanos
    c) Manipulación algebraica
    d) Representación de ´algebras booleanas finitas

2. Calculo de proposiciones
    a) Semántica
        1) Tablas de verdad
        2) Formas normales conjuntivas y disyuntivas
        3) Algoritmos de minimización
    b) Sintaxis
        1) Proposiciones bien formadas
        2) Axiomas lógicas, reglas de inferencia, pruebas, teoremas
        3) Teorema de Deducción
    c) Teorema de consistencia
    d) Teorema de completitud
    e) Demostración automática de teoremas
        1) Algoritmos de Wang
3. Calculo de predicados
    a) Sintaxis
        1) Términos y formulas bien formadas
        2) Axiomas lógicas, reglas de inferencia, pruebas, teoremas
        3) Teorema de deducción
    b) Formas clausulares
    c) El cálculo de predicados y PROLOG
    d) Semántica
        1) Interpretación y modelos
        2) Teorema de Lowenheim-Skolem
        3) Teorema de Compacidad
    e) Teorema de completitud de Godel
4. Demostración automática de teoremas
    a) Davis-Putnam.
    b) Unificación y resolución
    c) Estrategias de resolución
5. Lógicas modales y polivalentes
    a) Lógica modal
    b) Lógica difusa
    c) Verificación de programas

Bibliografía

1. Barwise (ed): *Handbook of Mathematical Logic*, Studies in Logic and the Foundations of Mathematics, Vol. 90. North-Holland Publishing Co., 1977.
2. Bauer, P., Nouak, S.,Winkler, R. *A brief course in fuzzy logic and fuzzy control*, disponible en ftp.flll.uni-linz.ac.at/pub/info/, 1996.
3. Bell, Machover: *Introduction to Mathematical Logic*, North-Holland, 1978.
4. Bolc, Borowik: *Many-valued Logics*, Springer-Verlag, 1992.
5. Boyer, Moore: *A Computational Logic*. Academic Press, 1979.
6. Clocksin, Mellish: *Programming in Prolog*, Springer-Verlag, 1981.
7. Cuena: *Lógica Informática*, Alianza Informática, 1987.
8. Chang, Lee: *Symbolic Logic and Mechanical Theorem Proving*. Academic Press, 1973.
9. Enderton, *A Mathematical Introduction to Logic*, Academic Press, 1972.
10. Gabbay, Hogger, Robinson, (editors): *Handbook of Logic in Artificial Intelligence and Logic Programming*, Oxford University Press, 1993.
11. Gallier: *Logic for Computer Science*. John Wiley, 1987.
12. Genesereth, Nilson: *Logical foundations of Artificial intelligence*. Morgan Kauffman, 1987.
13. Hamilton: *Logic for Mathematicians*. Cambridge University Press, 1978.
14. Hajek, P., Godo Ll. *Deductive systems of fuzzy logic*, unpublished manuscript, 1997.

15. Huth, Ryan: *Logic in Computer Science: Modelling and Reasoning about Systems*, Cambridge University Press, 1999.
16. Kowalski R.: *Logic for Problem Solving*, North Holland, 1979.
17. Lu: *Mathematical Logic for Computer Science*. World Scientific, 1989.
18. Mendelson: *Introduction to Mathematical Logic*. Van Nostrand (Third Edition), 1987.
19. Nilsson: *Principles of Artificial Intelligence*. Morgan Kauffman, 1980.
20. Schoning: *Logic for Computer Scientists*. Birkhauser,1989.
21. Shoenfield: *Mathematical Logic*, Addison-Wesley Publishing Co., 1967.
22. Sterling, Shapiro: *The Art of Prolog: Advanced Programming Techniques*. MIT Press, 1986.

Minería de Datos

Objetivo

Conocer de manera general las técnicas y enfoques del proceso genearl de Minería de Datos. Se conocen los fundamentos y conceptos necesarios de cada una de las etapas del proceso. Se explora el uso de fuentes de datos para análisis y toma de decisiones resultantes de tareas de clasificación, predicción o agrupamiento.

Contenido

1. Introducción a minería de datos
   a. Objetivos de la minería de datos
   b. Problemas aptos para minería de datos
   c. Aplicaciones comerciales
   d. Aplicaciones no-comerciales
   e. Técnicas generales para análisis de los datos (predicción, clasificación, clustering).

2. Selección de Fuentes de Datos y Calidad de Datos
   a. Datos esperados para minería de datos
   b. Calidad de los datos
   c. Fuentes posibles de datos.

3. Preprocesamiento y Preparación de Datos
   a. Operaciones sobre los datos
   b. Problemas en el manejo de datos reales
   c. Selección de variables
   d. Muestreo, selección de registros
   e. Análisis de correlación
   f. Creación de nuevas variables, agregación de variables

4. Técnicas de Análisis
   a. Principales técnicas para el análisis de datos.
   b. Aplicación de las técnicas de análisis de datos
   c. Técnicas para la identificación de características, tendencias y relaciones en los datos
   d. Visualización
   e. Técnicas estadísticas (correlación, análisis factorial)

5. Creación de modelos de datos: clasificación y predicción
   a. Principales técnicas para clasificación y predicción de datos

b. Aplicación de las técnicas de clasificación y predicción de datos
c. Inducción de reglas: C4.5.
d. Redes neuronales.
e. Técnicas estadísticas: regresión.

6. Creación de modelos de datos: clustering
   a. Conceptos fundamentales de clustering
   b. Principales técnicas para el clustering
   c. Aplicación de técnicas de clustering
   d. Redes neuronales: Kohonen SOM
   e. Técnicas estadísticas: K-means
   f. Clustering difuso: Fuzzy c-Means

7. Evaluación de modelos
   a. Conceptos fundamentales
   b. Técnicas de evaluación de modelos
   c. Aplicación de las técnicas de evaluación de modelos

Bibliografía

1. Data Mining: Concepts and Techniques, 2nd ed. Jiawei Han and Micheline Kamber Morgan Kaufmann, 2006
2. Data Mining: Practical Machine Learning Tools and Techniques I.H. Witten and E. Frank Morgan Kaufmann, 1999
3. Machine Learning Tom M. Mitchell Prentice Hall, 2003
4. Inteligencia Artificial Russell y Norvig Prentice Hall, 2009
5. Pang-Ning Tan, Michael Steinbach and Vipin Kumar. Introduction to Data Mining. Addison- Wesley. 2006. ISBN: 0321321367.
6. Anil K. Jain, Richard C. Dubes. Algorithms for Clustering Data. Prentice Hall. 1988. ISBN: 013022278X.
7. T. Hastie, R. Tibshirani and J. Friedman. Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer-Verlag. 2001. ISBN: 0387952845.

Modelación y simulación de sistemas de eventos discretos

Objetivo

Es curso provee una introducción al modelado de sistemas usando tanto simulaciones de computadora como técnicas matemáticas. Se espera que al final de curso el alumno tenga el conocimiento suficiente para la realización de simulaciones efectivas. El software usado para las simulaciones es ARENA.

Contenido

1. Modelado de sistemas para simulaciones
2. Verificación y validación del modelo
3. Colección y análisis de datos
4. Simulación del análisis de salida
5. Experimentación con el modelo de simulación
6. Técnicas optimas de búsqueda
7. Comparación y evaluación de sistemas de diseño alternativos
8. Implementación de resultados
9. Generación de números aleatorios

10. Generación de variaciones aleatorias

Bibliografía

1. A.M. Law and W.D. Kelton, Simulation, Modeling and Analysis, 3rd edition, McGrawHill, 2000.
2. W.D. Kelton, R.P. Sadowski, D.T. Sturrock, Simulation with ARENA, 3rd edition, Mc-Graw Hill, 2004
3. Notas: http://delta.cs.cinvestav.mx/lixo/teach.html
4. Discrete Event System Simulation (Banks, J., Carson, J.S., Nelson, B.L., D.M. Nicol), Prentice Hall Publishing Company, 3rd edition, 2000 (T57.62.b35, 2000)
5. Introduction to Discrete Event Systems (Christos G. Cassandras and Stephane Lafortune), Kluwer Academic Publishers, 1999
6. Object-oriented Computer Simulation of DES (Jury Tyszer), Kluwer Academic Publishers, April, 1999
7. Journals: 1. Simulation; 2. IEEE Transactions on: System, Man and Cybernetics, Robot and Automation, Knowledge and Data Engineering; 3. International Journal of: Production Research, Computer Integrated Manufacturing.

Optimización Combinatoria

Objetivo

El objetivo de este curso es que el estudiante conozca y aplique a problemas teóricos y del mundo real el paradigma de solución basado en la optimización combinatoria. El paradigma de solución de la optimización combinatoria descansa en fundamentos teóricos para modelar y representar problemas, y en algoritmos de búsqueda local para resolver dichos problemas. El fundamento teórico de la optimización combinatoria se apoya en la teoría de la complejidad computacional y en el uso de objetos matemáticos para representar problemas de optimización combinatoria. Los objetos matemáticos que serán presentados incluyen; grafos, árboles, permutaciones, particiones, cadenas de crecimiento restringido, polinomios diagonales, polinomios caja, polinomios “mayor que”. Es altamente relevante el énfasis que se hace sobre la representación, generación, y enumeración. Los algoritmos de búsqueda local usan la modelización/representación basada en objetos matemáticos para resolver instancias de problemas teóricos y prácticos. Los algoritmos de búsqueda local que serán tratados en detalle incluyen: a) algoritmos de mutación/selección; b) algoritmo de recocido simulado; c) algoritmos genéticos; d) búsqueda tabú; y e) algoritmo de colonia de hormigas.

Al final del curso se espera que el estudiante tenga los conocimientos y habilidades necesarias para modelar problemas de la vida real usando al menos los objetos matemáticos vistos en el curso , y resolver dichos problemas usando algoritmos de optimización combinatoria básicos o mezclas de ellos.

Contenido

1. Teoría de complejidad computacional.
   a. Teoría de NP-Completez
   b. Definición de las clases de complejidad computacional
   c. Procedimiento para la demostración de la NP-Completez

2. Representación y Generación (RG) de objetos matemáticos
   a. RG de grafos
   b. RG de árboles
   c. RG de permutaciones.
   d. RG de particiones
   e. RG de cadenas de crecimiento restringido
   f. RG de polinomios diagonales, polinomios caja, y polinomios “mayor que”

3. Algoritmos de Mutación y Selección
    a. Introducción a los Algoritmos de mutación y selección
    b. Algoritmo de 1 padre – 1 hijo
    c. Algoritmo de 1 padre – m hijos
    d. Algoritmo de n padres – m hijos selección uniforme
    e. Algoritmo de n padres – m hijos selección proporcional
    f. Sintonización de parámetros

4. Algoritmo de Recocido simulado
    a. Algoritmos de aceptación por umbral
    b. La analogía física
    c. Modelos Markovianos
    d. Un modelo homogéneo
    e. Un modelo no-homogéneo
    f. Comportamiento asintótico
    g. Calendarización de enfriamiento
    h. Sintonización de parámetros

5. Algoritmos Genéticos
    a. Algoritmos evolutivos
    b. Bases teóricas de los algoritmos genéticos
    c. Genética de poblaciones
    d. Selección artificial
    e. Selección natural
    f. Recombinación de genes
    g. Sintonización de parámetros

6. Búsqueda Tabú
    a. Inspiración del algoritmo de búsqueda tabú
    b. Métodos de solución con un enfoque iterativo
    c. Uso eficiente de la memoria
    d. Listas tabú de tamaño variable
    e. El criterio de aspiración
    f. Ajuste de la diversificación
    g. Sintonización de parámetros

7. Algoritmo de Colonia de Hormigas (ACH)
    a. Teoría de la optimización basada en colonia de hormigas
    b. Representación ad-hoc para ACH
    c. Problemas de enrutamiento
    d. Problemas de calendarización

8. Aplicación a problemas reales
    a. Modelización de problemas reales
    b. Análisis de la representación y espacio de búsqueda
    c. Funciones de vecindad
    d. Sintonización de parámetros

Bibliografía

1. Local Search in Combinatorial optimization, E. Aarts, J, Karel, ISBN 0-691-11522-2, Princeton University Press, 2003
2. Ant Colony Optimization, M. Dorigo, T. Stützle, ISBN 81-203-2684-9, Prentice Hall, 2004
3. Combinatorial Algorithms Generation, Enumeration, and Search, D.L. Kreher, D,R, Stinson, CRC Press, 1998
4. Tabu Search, F. Glover. M. Laguna, Kluwer Academic Publishers, ISBN 0-7923-9965-X,

1997
5. The Design of Innovation (Genetic Algorithms and Evolutionary Computation), D. Goldberg, ISBN 1402070985, Springer 2002

Optimización en ingeniería

Objetivo

Presentar al estudiante un repaso histórico, teórico y práctico de los diversos métodos de optimización global, enfatizando sus ventajas y desventajas. Así mismo, generar habilidades para decidir y modificar técnicas según las demandas de la aplicación específica. En este curso se estudian diversos métodos de programación matemática para resolver problemas de optimización lineal y no lineal (principalmente sin restricciones). El curso enfatizará aspectos algorítmicos y de implementación sobre los aspectos teóricos, por lo que es necesario tener al menos conocimientos básicos de programación. También se requieren conocimientos de cálculo, trigonometría, geometría y álgebra.

Contenido

1. Introducción a la optimización.
2. Técnicas clásicas.
3. Programación lineal y el método simplex.
4. Programación no lineal: métodos unidimensionales.
5. Programación no lineal: métodos multidimensionales.
6. Programación no lineal: métodos de optimización con restricciones.
7. Repaso de métodos modernos de optimización.

Bibliografía

1. Singiresu S. Rao. Engineering Optimization: Theory and Practice. Ed. John Wiley & Sons, Inc., 1996.
2. Kalyanmoy Deb. Optimization for Engineering Design: Algorithms and Examples. Ed. Prentice-Hall of India, 1995.
3. David M. Himmelblau. Applied Nonlinear Programming. Ed. McGraw-Hill, 1972.
4. G.V. Reklaitis, A. Ravindran y K.M. Ragsdell. Engineering Optimization: Methods and Applications. Ed. John Wiley & Sons, Inc., 1983.
5. Jorge Nocedal y Stephen J. Wright. Numerical Optimization. Ed. Springer, 1999.
6. Garrett N. Vanderplats. Numerical Optimization Techniques for Engineering Design with Applications. Ed. McGraw-Hill, 1984.
7. J. Frédéric Bonnans, J. Charles Gilbert, Claude Lemaréchal y Claudia A. Sagastizábal. Numerical Optimization: Theoretical and Practical Aspects. Ed. Springer, 2003.
8. P. Venkataraman. Applied Optimization with MATLAB Programming. Ed. John Wiley & Sons, 2002.

Programación Concurrente

Objetivo

1. Estudio de los mecanismos para compartir y controlar recursos.
2. Estudio de los mecanismos basados en paso de mensajes.
3. Estudio de lenguajes académicos de programación concurrente: Pascal – S, SR.
4. los conceptos fundamentales en el diseño e implementación de aplicaciones multitarea.
5. Uso de bibliotecas para el desarrollo de aplicaciones multi-hilo: Pthreads, Java-

Threads

Contenido

1. Conceptos básicos
    a) Diferencia entre programación secuencial y programación concurrente.
    b) Conceptos de: Proceso, programa multitarea, multiproceso, granularidad, etc.
    c) Ejecución atómica, principio de ejecución concurrente.
    d) Sincronización, no interferencia.
    e) Propiedades: vivacidad, seguridad, exactitud.
2. Exclusión mutua
    a) Introducción al problema de la exclusión mutua.
    b) Propuestas clásicas de solución: una bandera, dos banderas, cesión voluntaria, asignación de turnos.
    c) Especificación de ejecución concurrente: cobegin/coend, fork/join, corutinas.
    d) Concurrencia y sincronización: ejecución intercalada.
    e) Operaciones atómicas: principio de no interferencia
    f) Invariantes globales
3. Mecanismos de control de concurrencia basados en variables compartidas
    a) Semáforos.
    b) Barreras.
    c) Lista de esperas acotadas.
    d) Algoritmos clásicos de la programación concurrente: Productores y consumidores 5 filósofos, barbero dormido, etc.
4. Pascal – S
    a) Importancia histórica de Pascal – S en la evolución de la programación concurrente.
    b) Primer trabajo práctico.
5. Paso de mensajes
    a) Rendez-vous
    b) Llamados a procesos remotos
    c) Llamado a métodos remotos
    d) Comunicación síncrona y asíncrona
6. Lenguaje de programación SR (Sincronizing Resources)
    a) Presentación de SR, como lenguaje académico para el desarrollo de aplicaciones concurrentes
    b) Segundo trabajo práctico
7. Programación multi-hilo
    a) Introducción a la programación multi-hilo
    b) Diferencia entre hilos y procesos
    c) Gestión de recursos de maquina al programar con hilos
    d) Bibliotecas para la programación multi-hilo
    e) Tercer trabajo práctico

Bibliografía

1. Ben-Ari, M., Principles of Concurrent and Distributed Programming, Prentice Hall, 1990.
2. Lewis, B,. Berg, D., Pthread Primer – A guide to Multithreaded Programming, SunSoft Press 1996.
3. Hoare, C., Communicating Sequential Process, June 2004
4. Goetz, B., Peierls, T., Bloch, J., Bowbeer, J., Holmes, D., Lea, D., Java Concurrency in Practice, Addison Wesley Professional, 2006

Procesamiento digital de imágenes

Objetivo

El curso está enfocado a introducir los conceptos teóricos y prácticos asociados con el procesamiento de imágenes digitales bidimensionales. Los siguientes tópicos principales son cubiertos: preprocesamiento, segmentación, análisis y reconocimiento de la imagen. Al completar el curso el estudiante será capaz de aplicar los conceptos aprendidos en problemas reales, así como comprender textos avanzados de investigación en el campo de procesamiento de imágenes.

Contenido

0. Revisión general de matrices y teoría de conjuntos
1. Fundamentos de la imagen digital
    a. Concepto de imagen
    b. Adquisición de imágenes
    c. Muestreo y cuantificación
        1. Imágenes en escala de grises
        2. Imágenes a color
    d. Formato de imágenes
        1. Profundidad del color
        2. Compresión
        3. Tipos de gráficos
        4. Formatos de almacenamiento
    e. 1.5. Relaciones básicas entre píxeles
        1. Operaciones interpíxeles
        2. Operaciones artimético-lógicas
        3. Vecindad
        4. Conectividad
        5. Etiquetado
        6. Métricas
    f. Transformaciones geométricas de la imagen
        1. Transformaciones básicas
        2. Coordenadas homogéneas
        3. Transformaciones compuestas

2. Transformadas básicas de la imagen
    a. Transformada de Fourier
        1. Transformada continua de Fourier (1D y 2D)
        2. Transformada discreta de Fourier (1D y 2D)
        3. Transformada rápida de Fourier
    b. Convolución
        1. Convolución continua
        2. Convolución discreta
    c. Otras transformadas de la imagen separables
        1. Transformada Hadamard
        2. Transformada Walsh

3. Mejoramiento de la imagen
    a. Fundamentos
        1. Métodos en el dominio espacial
        2. Métodos en el dominio de la frecuencia
    b. Transformaciones de intensidad simples
        3. Normalización
        4. Negativos
    c. Compresión del rango dinámico

1. Procesamiento del histograma
   d. Ecualización del histograma
   e. Ecualización adaptativa del histograma
      1. Mejora local
      2. Sustracción de imágenes
      3. Promediado de imágenes
      4. Técnicas difusas (Fuzzy)

4. Filtrado de imágenes
   a. Tipos de ruido en imágenes
   b. Filtrado espacial
      1. Fundamentos
      2. Filtros suavizantes
      3. Filtros realzantes
      4. Detección de bordas
   c. Filtrado en el dominio de la frecuencia
      1. Fundamentos
      2. Filtro pasa bajas (Butteworth)
      3. Filtro pasa banda (Gabor)
      4. Filtro pasa altas (Butteworth)
      5. Filtro homomórfico

5. Morfología matemática
   a. Fundamentos
   b. Erosión y dilatación
   c. Apertura y cerradura
   d. Gradientes morfológicos
   e. Esqueletos

6. Segmentación de imágenes
   a. Umbralado global simple
   b. Umbralado óptimo
      1. Método de Otsu
      2. Método Ni-Black
      3. Basado en las características del límite
   c. Segmentación orientada a regiones
      1. Transformada Watershed

Bibliografía

1. Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing,” 3rd. Edition, Prentice Hall, 2008. ISBN: 9780131687288
2. Rafael C. González, Richard E. Woods, and Steven L. Eddins, “Digital Image Processing Using MATLAB,” 2nd. Edition, Gatesmark Publishing, 2009. ISBN: 9780982085400
3. Pierre Soille, “Morphological Image Analysis. Principles and Applications,” 2nd. Edition, Springer-Verlag, 2004. ISBN: 3-540-42988-3.
4. Libro de texto: *Digital Image Processing*, R.C. Gonzalez, and R.E.Woods (1992), AddisonWesley.
5. The image processing handbook Russ, John C. CRC Press. 1998
6. Digital image processing Pratt, William K. 1991.

Realidad virtual

Objetivo

Se estudian la tecnología actual para realizar realidad virtual, realidad aumentada y

uso de interfaces hápticas. Se analizaran los aspectos de ingeniera, científicos y aspectos funcionales de sistemas de realidad virtual y los fundamentos de modelado de mundos virtuales y su programación. Se tratan las técnicas de imagen con OpenGL para la generación de ambientes tridimensionales (3D): creación de modelos, diseño de escenarios, iluminación, texturas, interiores, exteriores y perspectiva.

Contenido

1. Introducción a la realidad virtual
2. Algoritmos de geometría computacional.
3. Dibujado de escenas en 3D con OpenGL
    a) Mallas de triángulos
    b) Estructuras de datos para manejas datos masivos
    c) Iluminación
    d) Manejo de textura
4. Sistemas de realidad aumentada
    a) Adquisición de video
    b) Procesamiento de los marcos de video
    c) Dibujado de la escena virtual
5. Interfaces hápticas
    a) Modelado
    b) Adquisición de parámetros
    c) Colisión de detecciones en 3D
6. Modelos deformables con mallas de simplejos
7. Hardware para ambientes inmersivos
8. Creación del mapa virtual de la República Mexicana.

Bibliografía

1. W. R. Sherman and A. Craig, Understanding Virtual Reality: Interface, Application, and Design. The Morgan Kaufmann Series in Computer Graphics. 2004.
2. G. van den Bergen, Collision detection in interactive 3D enviroments. Morgan Kaufmann. 2004.
3. T. McReynolds and D. Blythe. Advanced Graphics Programming Using OpenGL. Morgan Kaufmann. 2005.
4. F.P. Preparata and M.I. Shamos. Computational Geometry, Springer-Verlag. 1985.

Reconocimiento de Patrones

Objetivo

Conocer los procesos en ingeniería, computación y matemáticas relacionados con objetos físicos y/o abstractos, con el propósito de extraer información que permita establecer propiedades de o entre conjuntos de dichos objetos. El reconocimiento de patrones es un proceso fundamental en casi todas las acciones humanas. Lo esencial de él es la clasificación. Clasificar patrones en base a un conocimiento a priori o información estadística extraída de los patrones. Los patrones a clasificar suelen ser grupos de medidas u observaciones, definiendo puntos en un espacio multidimensional apropiado. Entre las aplicaciones del reconocimiento de patrones tenemos: el reconocimiento de voz, clasificación de documentos, reconocimiento de la escritura, reconocimiento de caras humanas, etc.

Contenido

1. Introducción.

a. ¿ Qué es el reconocimiento de patrones ?
b. Ejemplos de aplicación.
c. Formulación de un problema en Reconocimiento de Patrones.
d. Sistemas de Reconocimiento de Patrones.
e. Conceptos básicos.
f. Líneas de investigación y/o enfoques en Reconocimiento de Patrones.

2. Extracción de características.
   a. Características geométricas.
   b. Características cromáticas.
   c. Otras características.
   d. Normalización de características.

3. Selección de características.
   a. Análisis de componentes principales.
   b. Análisis multivariado.
   c. Discriminante de Fisher.
   d. Búsquedas.

4. Clasificación estadística.
   a. Introducción.
   b. Aprendizaje supervisado paramétrico.
   c. Aprendizaje supervisado no paramétrico.
   d. Aprendizaje no supervisado.

5. Clasificación neuronal.
   a. Introducción.
   b. Modelo neuronal básico.
   c. Modelos clásicos en redes neuronales.
   d. Perceptrón multinivel.
   e. Memorias asociativas.

6. Categorización.
   a. Conceptos básicos.
   b. Medidas de proximidad.
   c. Algoritmos secuenciales.
   d. Algoritmos jerárquicos.
   e. Esquemas basados en funciones de optimización.
   f. Validez de una categorización.

7. Evaluación del desempeño de la clasificación.
   a. Matriz de confusión.
   b. Sensibilidad y especificidad.
   c. Curvas ROC.
   d. Validación cruzada.
   e. Intervalos de confianza.

Bibliografía

Libro de texto

1. J. P. Marques de Sá, "Pattern recognition : Concepts, methods and applications", 318 pages, Springer-Verlag 2001, ISBN: 3-540-42297-8.

Libros complementarios

1. Sergios Theodoridis, Konstantinos Koutroumbas, "Pattern Recognition", 856 pages, Academic Press, 3 edition (February 24, 2006) ISBN: 0123695317
2. R.O. Duda, P.E. Hart, D.G. Stork, "Pattern classification", 680 pages, Ed. Wiley 2000, ISBN: 0-471-05669-3.
3. Bishop, "Pattern Recognition and Machine Learning", 738 pages, Springer 2006, ISBN. 9780387310732.
4. Evangelia Micheli-Tzanakou, "Supervised and unsupervised pattern recognition", 367 pages, CRC 2000, ISBN: 0849322782.
5. Phiroz Bhagat, "Pattern recognition in industry", Elsevier 2005, ISBN: 0080445381
6. Dietrich W. R. Paulus, Joachim Hornegger, "Applied Pattern Recognition, Fourth Edition: Algorithms and Implementation in C++", 372 pages, GWV-Vieweg, 4 edition (December 2, 2003) ISBN: 3528355581
7. M. Friedman and A. Kandel, "Introduction to Pattern Recognition, statistical, structural, neural and fuzzy logic approaches", World Scientific, Signapore, 1999
8. Donald Tveter, "The Pattern Recognition Basis of Artificial Intelligence", 388 pages March 1998, Wiley-IEEE Computer Society Press, ISBN: 0-8186-7796-1
9. R. Schalkoff, "Pattern Recognition: Statistical, Structural and Neural Approaches", 384 pages, Ed. Wiley, 1 edition (June 7, 1991) ISBN: 0471529745

Redes de Computadoras

Objetivo

En este curso el alumno conocerá las tecnologías involucradas en las redes de computadoras analizadas a través de la arquitectura por capas siguiendo un enfoque descendente. Presentar los principios básicos de la arquitectura TCP/IP y su implementación en Internet. Se revisaran distintas alternativas de interconexión de redes, la función y problemática de cada una de las capas del modelo ISO/OSI. Se dará especial énfasis a las capas de aplicación, transporte y red del modelo de referencia de Internet.

El alumno conocerá los protocolos básicos dentro de cada capa. Complementara los estudios teóricos con implementaciones básicas de algunos de los algoritmos y protocolos analizados. Analizara algunos aspectos generales de gestión de redes y de nuevas tecnologías de redes inalámbricas.

Contenido

1. Introducción
    a) Elementos de Internet (hosts, routers, conexiones, proveedores, etc.)
    b) Tipos de protocolos (orientación a conexión y sin conexión)
    c) Tipos de redes (difusión, conmutación)
    d) Acceso a Internet. Medios de transmisión.
    e) Rutas y retardos en Internet.
    f) Arquitectura. Modelo de capas. TCP/IP
2. Capa de aplicación
    a) Protocolos básicos
    b) El world wide web: HTTP
    c) Transferencia de archivos: FTP
    d) Correo electrónico y noticias: SMTP, POP3, IMAP, NNTP
    e) Protocolos especializados
    f) Protocolo de configuración dinámica: DHCP
    g) Servicio de directorio: DNS

h) Protocolo de administración de redes: SNMP
i) Seguridad: SSL, HTTPS
j) Programación de Sockets (TCP e UDP)

3. Capa de transporte
   a) Características generales y clasificación
   b) Servicio sin conexión. UDP
   c) Fundamentos de la transferencia fiable
   d) Control de flujo: retransmisión adaptativa, ventana deslizante
   e) Servicio orientado a conexión. TCP
   f) Control de la congestión
4. Capa de red
   a) Técnicas de conmutación: datagramas, circuitos virtuales
   b) Algoritmos de estado de enlaces y vector de distancias
   c) Estrategias de encaminamiento. RIP, OSPF, BGP
   d) El protocolo Internet (IP e IPv6)
   e) El protocolo de mensajes de control de Internet (ICMP)
   f) Estructura de un router
5. Capa de enlace
   a) Función de la capa de enlace.
   b) Protocolos de acceso al medio. Ethernet.
   c) Hubs, bridges y switches.
   d) El protocolo punto a punto (PPP)
   e) Modo de transferencia asíncrona (ATM)
6. El nivel físico
   a) La transmisión de datos
   b) Señales para la transmisión de datos
   c) Medios de transmisión
   d) Codificación y modulación
   e) Tecnologías de módem
7. Presentaciones adicionales:

a) Gestión de Redes
b) Redes Inalámbricas.

Bibliografía

1. J.F. Kurose y K.W. Ross, Redes de Computadores. Un enfoque Descendente Basado en Internet", 2004, Pearson Addison Wesley. ISBN: 8478290613.
2. D.E. Comer, D.L. Stevens y M. Evangelista, "Internetworking with TCP/IP, Vol. III: Client- Server Programming and Applications, Linux/Posix Sockets Version", 2001, Prentice Hall. ISBN: 0130320714.
3. A.U. Black, "Tecnologías emergentes para redes de computadoras", 1999, Pearson Prentice Hall. ISBN: 9701702689.
4. A.S. Tanenbaum, Redes de computadoras", 2003, Pearson Prentice-Hall. ISBN: 9702601622.
5. W. Stallings, Comunicaciones y redes de computadores", 2004, Person Prentice-Hall. ISBN: 8420541109,
6. D. E. Comer y D.L. Stevens, "Internetworking with TCP/IP Vol. III Client-Server Programming and Applications-Windows Sockets Version", Prentice-Hall. 2000. ISBN: 0138487146.
7. M. Donahoo y K. Calvert, "TCP/IP Sockets in C: Practical Guide for Programmers (The Practical Guides Series)", Morgan Kaufmann. 2000. ISBN: 1558608265.
8. Computer Networking: A Top-Down Approach, 5th Edition. James F. Kurose, Amhersteith W. Ross. ISBN-10: 0136079679, ISBN-13: 9780136079675. Ed. Addison-Wesley 2010
9. Jim Kurose, Keith Ross. Computer Networking: A Top Down Approach

Featuring the Internet", 3rd edition, Addison-Wesley
10. Lopez, A. Novo. Protocolos de Internet. Diseño e implementación en sistemas UNIX. Ed. Alfa Omega
11. W. Stevens, TCP/IP Illustrated, Vol. 1: The Protocols, Addison-Wesley, 1994
12. D.E. Comer, Interworking with TCP/IP. Vol1: Principles, protocols and architectures, Ed. Prentice Hall International, 1995
13. Página del Internet Engineering Task Force http://www.ietf.org
14. Página web del Network Simulator 2. http://www.isi.edu/nsnam/ns/ns-build.html

Redes Neuronales Artificiales

Objetivo

Conocer, entender y aplicar algunos modelos clásicos de Redes Neuronales Artificiales (RNA) para la resolución de problemas y descubrir la estrecha relación entre la neurobiología, la probabilidad, la estadística y la computación para el modelado con RNA inspiradas en el sistema nervioso. Las Redes de Neuronas Artificiales son un paradigma de aprendizaje y de procesamiento automático inspirado en el funcionamiento del sistema nervioso. Éstas simulan las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos artificiales (como un circuito integrado, una computadora o un conjunto de válvulas). El objetivo es conseguir que las máquinas den respuestas robustas y de generalización similares a las que es capaz de dar nuestro cerebro. En este curso se busca entender el uso de las RNA como herramienta de análisis de patrones y su posible aplicación al estudio de la codificación de la información en diversos sistemas sensoriales, además de aplicar las RNA en el modelado de sistemas biológicos y simular el alto nivel de paralelismo en el procesamiento de la información visual. Para ello empezaremos revisando sus fundamentos biológicos. Seguiremos con la modelización clásica de RNA (características, topología, reglas de aprendizaje, etc.) y haremos una clasificación de las mismas, analizaremos las RNA con alimentación hacia adelante, aquellas con alimentación hacia atrás, las memorias asociativas, ejemplos de aplicación y una introducción al modelado bio-inspirado : conexionismo neuromimético.

Contenido

1. Introducción.
   a. ¿Qué son las redes neuronales artificiales y cuáles sus aplicaciones?
   b. Fundamentos biológicos.
   c. Elementos de las RNA.
   d. Procesos de aprendizaje.
2. RNA tipo perceptrón.
   a. Problema del filtrado adaptativo.
   b. Filtros y algoritmos por mínimos cuadrados
   c. El perceptrón y su regla de aprendizaje.
   d. Algoritmo de retropropagación.
   e. Generalización del algoritmo de retropropagación.
   f. El perceptrón multinivel y su regla de aprendizaje.

3. Memorias asociativas.
   a. Noción de sistemas dinámicos.
   b. Atractores.
   c. Modelo Hopfield.
   d. Modelo BAM.
   e. Modelo de las máquinas de Boltzmann.

4. RNA recurrentes y RNA híbridas.
   a. Arquitecturas de red recurrente.
   b. Mapas auto-organizados.
   c. Arquitecturas híbridas.

5. Aplicaciones.
   a. En visión.
   b. En control.
   c. En tratamiento de señales.
   d. En la toma de decisiones.

6. Líneas abiertas de investigación en RNA.
   a. Aprendizaje neuro-conjugado.
   b. Programación neurodinámica.
   c. RNA binarias.
   d. RNA difusas.

7. Introducción al conexionismo neuromimético.
   a. La bio-inspiración en la neurocomputación.
   b. Localidad y distribución.
   c. Modularidad y escalabilidad.
   d. Conexionismo bio-inspirado.

Bibliografía

Libro de texto

1. Haykin S., "Neural Networks: A Comprehensive Foundation" (2nd Edition) (Hardcover)" , Hardcover: 842 pages Publisher: Prentice Hall; 2nd edition (July 6, 1998) ISBN: 0132733501.

Libros complementarios

1. Madan M. Gupta, Liang Jin, Noriyasu Homma, "Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory" (Hardcover), Hardcover: 700 pages Publisher: Wiley-IEEE Press; 1 edition (April 1, 2003) ISBN: 0471219487.
2. Raul Rojas, J. Feldman, "Neural Networks: A Systematic Introduction" (Paperback), Paperback: 502 pages Publisher: Springer; 1 edition (July 12,1996) ISBN: 3540605053.
3. Gérard Dreyfus, "Neural Networks", Springer 2005, ISBN 9783540229803.
4. Ke Lin Du y M. N. S. Swamy, "Neural Networks in a Softcomputing Framework", Springer 2006, ISBN 9781846283024.

Robots Móviles Inteligentes

Objetivo

Conocer los fundamentos de los aspectos computacionales utilizados en robots móviles autónomos. Estudiar los componentes de un robot móvil: percepción, visión, planeación, navegación, construcción de mapas y localización. Aplicar los conocimientos adquiridos en el desarrollo de proyectos con robots móviles.

Contenido

1. Introducción

a. ¿Qué es un robot?
b. Historia y motivaciones
c. Clasificación de robots
d. Componentes principales y conceptos básicos
e. Problemáticas
f. Aplicaciones

2. Arquitectura de robots móviles
   a. Diseño de robots móviles
   b. Mecánica
   c. Modelos matemáticos
   d. Clasificación de robots móviles

3. Percepción
   a. Sensores propioceptivos
   b. Sensores exteroceptivos
   c. Ruido
   d. Imprecisión
   e. Caracterización de sensores
   f. Filtros

4. Modelos de comportamiento
   a. Definiciones
   b. Motivaciones biológicos
   c. Vehículos de Braitenberg

5. Control de robots móviles
   a. Sistemas lineales
   b. Control en lazo abierto y en lazo cerrado
   c. Elementos de control
   d. Sistemas no holonómicos
   e. Principios de lógica difusa
   f. Principios de programación en tiempo real

6. Visión por computadora
   a. Principios
   b. Filtros
   c. Características invariantes
   d. Color
   e. Reconocimiento de patrones

7. Modelado del entorno
   a. Definiciones
   b. Representaciones geométricas
   c. Representaciones topológicas
   d. Espacio de configuraciones
   e. Localización
   f. Construcción de mapas

8. Planificación de trayectorias
   a. Definiciones. Enfoque local y global del problema de planificación
   b. Grafos de visibilidad y de tangentes
   c. Planificación por frentes de onda
   d. Campos de potencial artificial
   e. Planificadores cinemáticas

f. Algoritmos bio-inspirados

9. Robots inteligentes
   a. Procesos de decisión de Markov
   b. Métodos básicos de solución
   c. Clasificadores
   d. Aprendizaje por refuerzo

10. Temas avanzados en robótica móvil
   a. Interacción hombre-máquina
   b. Sistemas multirobots
   c. Robots humanoides
   d. Aplicaciones específicas

Bibliografía

1. Steven M. LaValle, Planning Algorithms, Cambridge University Press, 2006, disponible en http://msl.cs.uiuc.edu/planning/
2. G. Dudek, M. Jenkin, Computacional principles of mobile robots, Cambridge University Press, 2000 (texto)
3. R.C. Arkin, Behavior-based robotics, The MIT Press, 2000

Seguridad en sistemas de información

Objetivo

El curso está organizado en tres partes fundamentales. En la primera se revisan los aspectos fundamentales para construcción de un cortafuegos. En la segunda parte se revisan los fundamentos criptográficos para proveer servicios de seguridad y algunas aplicaciones. En la tercera parte se revisan algunas herramientas que ayudan a detectar y corregir problemas de seguridad informática en un sistema de información.

Contenido

1. Parte 1
   a. Introducción a TCP/IP
   b. Conceptos de seguridad en redes
   c. Configuración de red en el sistema GNU/Linux
   d. Configuración de una puerta (gateway)
   e. Uso de IPTables
   f. Configuración de cortafuegos
   g. Zonas militarizadas y redireccionamiento de servicios
   h. Configuración y compilación del núcleo de GNU/Linux
   i. Cómo crear un cortafuegos sin un disco duro. Arranque desde CDROM.
   j. LDAP
   k. Seguridad en redes inalámbricas:
   b. autenticación de usuarios y un cortafuego dinámico: NoCAT y WifiDog
   a. Redes virtuales

2. Parte 2
   a. Servicios de Seguridad en Sistemas de Información: Introducción.
   b. Servicios de Seguridad en Sistemas de Información: Aplicaciones.
   c. Casos de Estudio: Notaría Digital, Elecciones electrónicas, dinero digital
   d. Protocolo IKE de IPSec: Sigma.

e. Certificados e Infraestructura de llave Pública (PKI)
f. Caso de estudio 1: Certificados con información biométrica.
g. Caso de estudio 2: Facturas electrónicas del SAT
h. Protocolo SSL/TLS.
i. Seguridad en el cómputo nube
j. Seguridad en Ambientes Computacionales Altamente Restringidos.
k. Seguridad en Redes Inalámbricas de Sensores.

3. Parte 3
   a. Análisis de Tráfico en Redes Locales
   b. SNMP
   c. Control de Acceso
   d. Autentificación mediante Kerberos y otros protocolos
   e. Instalación y configuración de un servidor de correo electrónico con políticas de seguridad
   f. Mecanismos Anti-SPAM para el Correo Electrónico
   g. Servicios de seguridad en SMS
   h. Sistemas de Detección de Intrusos

Bibliografía

1. De la Fraga, L. G. Seguridad en Redes de Computadoras Usando GNU/Linux. Notas del curso impartido el 9 de septiembre 2004 en el 1st International Conference on Electrical and Electronics Engineering. Acapulco, Guerrero. September 8-10, 2004.
2. Gerhard Mourani, Securing & Optimizing Linux: The Ultimate Solution gmourani@openna.com, version 2.0, July 2002 http://www.tldp.org
3. Thomas Wadlow and Vlad Gorelik, "Security in the Browser", Communications of the ACM, Vol. 52 No. 5, May 2009.
4. Ryan West, "The Psychology of Security", Communications of the ACM, Vol. 51 No. 4, April 2008.
5. "Cloud Computing", Brian Hayes, Communications of the ACM July 2008 Vol. 51 No 7.
6. William Stallings. Cryptography and Network Security: Principles and Practice, 5th Edition. Prentice Hall. 2011.
7. Alexander Clemm. Network Management Fundamentals. Cisco Press. 2006.
8. Securing & Optimizing Linux: The Ultimate Solution Gerhard Mourani, gmourani@openna.com, version 2.0, July 2002 http://www.tldp.org
9. Sitio de OpenSSL (www.openssl.org)
10. Cryptography and Network Security. Principles and Practice, Third Edition.W. Stallings, Prentice-Hall, 2003.

Sistemas Colaborativos Distribuidos

Objetivo

Dar a conocer al alumno los fundamentos teóricos y prácticos del campo de investigación multidisciplinario denominado ”Trabajo Cooperativo Asistido por Computadora” (*CSCW por sus siglas en inglés*), haciendo énfasis en el estudio de los sistemas computacionales (*Groupware* por su denominación en inglés) que soportan grupos de personas comprometidas en un proyecto común y que proveen una interfaz a un entorno compartido. En particular, se analizan las arquitecturas de distribución fundamentales para permitir a personas físicamente distribuidas comunicar, colaborar y coordinar sus actividades como si estuvieran cara a cara. Asimismo, se estudian los principales mecanismos propuestos para administrar la compartición de la información, tanto a nivel de interfaz de grupo como a nivel de núcleo funcional. Este dominio de investigación ha contribuido a la evolución de diversos dominios de aplicación, entre los que se encuentran los sistemas de mensajes, los editores de grupo, los sistemas

de soporte para la toma de decisiones en grupo, las salas de reuniones virtuales, las conferencias por computadora, los agentes inteligentes, los sistemas de coordinación (*workflows*) y la enseñanza/aprendizaje colaborativo.

Contenido

1. Introducción al Trabajo Cooperativo Asistido por Computadora (TCAC)
    a) Motivación, historia y objetivos
    b) Perspectivas del TCAC
        1) Sociológica
        2) Sistemas distribuidos y bases de datos
        3) Redes de comunicaciones
        4) Interacción hombre-máquina
        5) Inteligencia artificial
    c) Diferencias entre aplicaciones mono-usuario, multi-usuario y colaborativas
    d) Modelo conceptual de sistemas colaborativos
        1) Espacio de comunicación
        2) Espacio de producción
        3) Espacio de coordinación
2. Topologías de sistemas colaborativos
    a) Interacciones entre colaboradores
        1) Cara a cara
        2) Síncronas-remotas
        3) Asíncronas
        4) Asíncronas-remotas
    b) Ejemplos representativos de aplicaciones
3. Requerimientos de los sistemas colaborativos
    a) Sincronía
    b) Grano de la información compartida y del tiempo
    c) Administración de la información compartida
        1) Arquitecturas de distribución
        2) Mecanismos de actualización
        3) Políticas de control de acceso
        4) Estrategias de control de concurrencia
        5) Persistencia
    d) Infraestructuras (frameworks) vs cajas de herramientas (toolkits)
4. Problemas en el diseño de sistemas colaborativos
    a) Interfaces de grupo
        1) Vistas WYSIWIS estrictas y relajadas
        2) Foco de atención y distracción
        3) Adaptabilidad a la dinámica del grupo
        4) Administración del espacio de despliegue
        5) Ejemplos de elementos de interfaz de grupo (widgets)
    b) Procesos de grupo
        1) Acceso secuencial vs. Paralelo
        2) Protocolos tecnológicos y sociales
        3) Operaciones síncronas vs. Asíncronas
    c) Mecanismos de control de concurrencia
        1) Candados explícitos
        2) Transacciones
        3) Copias maestras/esclavas
        4) Candados flexibles
        5) Toma de turno
        6) Detección de dependencias
        7) Ejecución reversible
        8) Transformación de operaciones

d) Control de acceso
e) Notificación de eventos

5. Arquitecturas de software para sistemas colaborativos
   a) Presentación-Abstracción-Control (PAC*)
   b) Abstracción-Liga-Vista (ALV)
   c) Modelo-Vista-Controlador (MVC) para sistemas multi-usuario
6. Arquitecturas de distribución
   a) Centralizada
   b) Totalmente replicada
   c) Hibrida
   d) Semi-replicada
   e) Asimétrica
   f) Arquitecturas adaptables

Bibliografía

1. G. Calvary, J, Coutaz, and L. Nigay, From Single-User Architectural Design to PAC*: a Generic Software Architecture Model for CSCW, in Proceedings of the Conference on Human-Computer Interaction (CHI'97), ACM Press, pp. 242-249, Atlanta GA (USA), March 1997.
2. Dix, Challenges for Cooperative Work on the Web: An Analytical Approach, Computer Supported Cooperative Work: The Journal of Collaborative Computing, Kluwer Academic Publishers, pp. 135-156, Netherlands, 1997.
3. A. Ellis, S. J. Gibbs, and G. L. Rein. Groupware: Some Issues and Experiences, in Communications of the ACM, 34(1): 38-58, 1991.
4. Ellis and J. Wainer, A Conceptual Model of Groupware, in Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'94), ACM Press, pp. 79- 88, Chapel Hill, NC (USA), 1994.
5. S. Greenberg and D. Marwood, Real-Time Groupware as a Distributed System: Concurrency Control and its Effect on the Interface, in Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'94), ACM Press, pp. 207-218, Chapel Hill, NC (USA), 1994.
6. Gutwin and S. Greenberg, A Descriptive Framework of Workspace Awareness for Real-Time Groupware, Computer Supported Cooperative Work 11(3-4): 411-446, 2002.
7. R. D. Hill, The abstraction-link-view paradigm: using constraints to connect user interfaces to applications, in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press, pp. 335 - 342, Monterey, CA (USA), 1992.
8. S. Lukosch, M. Hellweg, and M. Rasel, CSCL, Anywhere and Anytime, In Proceedings of the 12th International Workshop on Groupware (CRIWG'2006), Springer, LNCS 4154, pp. 326-340, Medina del Campo (Spain), 2006.
9. J. Munson and P. Dewan, A concurrency control framework for collaborative systems, in Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'96), ACM Press, pp. 278-287, Boston, MA (USA), November 1996.
10. W. G. Phillips, Architectures for Synchronous Groupware, Technical Report 1999-425, Department of Computing and Information Science, Queen's University, Kingston, Ontario (Canada), 1999.
11. 11. M. Romero Salcedo and D. Decouchant, Structured Cooperative Authoring for the World Wide Web, Computer Supported Cooperative Work, 6(2/3): 157-174, 1997.
12. 12. J. Tam and S. Greenberg, A framework for asynchronous change awareness in collaborative documents and workspaces, International Journal of Man-Machine Studies 64(7): 583-598, 2006.

Sistemas Distribuidos

Objetivo

Dotar al alumno de conocimientos para que pueda comprender y aplicar los sistemas distribuidos, tanto en el área de base de datos, como en el área de aplicaciones de red, como manejo de protocolos, sistemas operativos, bajo diferentes tipos de enlaces, diferentes arquitecturas de cómputo distribuido(cliente/servidor). Se analizarán las principales tecnologías de programación para sistemas distribuidos utilizando Middlewares como RPCs, RMIs, Corba y Servicios Web.

Contenido

1. Introducción a los Sistemas Distribuidos
    a. Caracterización de los sistemas distribuidos
    b. Ejemplos de sistemas distribuidos
    c. Recursos compartidos
    d. La Web

2. Modelos de sistema
    a. Modelos arquitectónicos
    b. Modelos fundamentales

3. Comunicación en sistemas distribuidos
    a. Mecanismos básicos de comunicación (IPC)
    b. Comunicación cliente/servidor
    c. Sockets y RPCs
    d. Presentación externa de datos y empaquetado (xdr, cdr, xml)
    e. Comunicación en grupo
    f. Middlewares
    g. Manejo de procesos e hilos

4. Sistemas de archivos distribuidos
    a. Conceptos básicos y estructura
    b. Servicios de directorio y archivos
    c. Replicación
    d. Manejo de transacciones
    e. Caso de estudio: NFS, AFS

5. Sincronización y coordinación
    a. Mecanismos de sincronización entre procesos
    b. Algoritmos de sincronización de relojes
    c. Manejo de estado global
    d. Coordinación y acuerdo: exclusión mutua, elección y multidifusión

6. Objetos distribuidos e invocación remota (middlewares)
    a. Introducción
    b. Comunicación entre objetos distribuidos
    c. Llamadas a procedimientos/métodos remotos (RPC/RMI)
    d. Eventos y notificaciones
    e. RMIs

7. Aplicaciones Web
    a. Aplicaciones con clientes pesados (desarrollo del lado del cliente: DHTML)
    b. Aplicaciones con clientes ligeros (desarrollo del lado del servidor: JSP)
    c. Arquitectura de aplicaciones Web usando MVC
    d. Servicios Web

8. Seguridad
    a. Introducción
    b. Técnicas de seguridad: firewalls, criptografía, certificados
    c. Algoritmos criptográficos: simétricos y asimétricos
    d. Firmas digitales

Bibliografía

1. Distributed Systems: Concepts and Design** . G. Coulouris, J. Dollimore, T. Kindberg, Editorial: Addison Wesley,2005, 4th edition. ISBN: 0321263545
2. Sistemas Distribuidos (español -versión anteior -). George Coulouris; Jean Dollimore; Sebastián Dormido; Tim Kindberg Editorial: Addison Wesley | 3era Edición Idioma: Español ISBN: 8478290494.
3. From P2P to Web Services and Grids: Peers in a Client/Server World Ian J. Taylor and Andrew Harrison Editorial Springer; 1 edición (Octubre 21, 2004) ISBN-10: 1852338695
4. ISBN-13: 978-1852338695
5. Distributed Systems: Principles and Paradigms** Andrew S. Tanenbaum, Maarten van Oteen
6. Editorial: Prentice Hall; United States 2nd edition (Oct 2, 2006) ISBN: 0132392275
7. Concurrent and Distributed Computing in Java ** Vijay K. Garg Editorial: Wiley-IEEE Press (February 4, 2004) ISBN: 047143230X
8. Java in Distributed Systems: Concurrency, Distribution and Persistence Marko Boger
9. Editorial: John Wiley & Sons; 1 edition (May 16, 2001) ISBN: 0471498386
10. Sistemas Operativos Distribuidos** Andrew S. Tanenbaum Editorial: Prentice Hall; ISBN: 9688806277
11. Distributed Operating Systems Andrew S. Tanenbaum Editorial: Prentice Hall; US Ed edition August 25, 1994) ISBN: 0132199084
12. Distributed Operating Systems: Concepts and Practice Doreen L. Galli Editorial: Prentice Hall; 1st edition (August 31, 1999) ISBN: 0130798436
13. Vijay K. Garg. Concurrent and Distributed Computing in Java. Editorial: Wiley-IEEE Press (February 4, 2004). ISBN: 047143230X.
14. Marko Boger. Java in Distributed Systems: Concurrency, Distribution and Persistence. Editorial: John Wiley & Sons; 1 edition (May 16, 2001). ISBN: 0471498386.
15. Doreen L. Galli Distributed Operating Systems: Concepts and Practice Editorial: Prentice Hall; 1st edition (August 31, 1999) ISBN: 0130798436

Sistemas Empotrados

Objetivo

Particularmente el curso persigue los siguientes objetivos:

- Establecer un marco de las tecnologías y componentes hardware y software para el desarrollo de sistemas empotrados, así como las herramientas que asistan en la administración de diseños
- Evaluar compromisos de diseño entre diferentes opciones tecnológicas para la implementación de sistemas empotrados

Aunado al contenido teórico de carácter general, el curso tiene una fuerte orientación al desarrollo de aplicaciones prácticas. Los aspectos teóricos se reforzarán fundamentalmente mediante la realización de un proyecto de diseño hardware/software de un sistema empotrado. En este sentido, es deseable, tener conocimientos previos de Lenguajes de descripción de hardware, Síntesis de alto nivel y programación.

Contenido

1. Introducción
    a. Concepto de sistema
    b. Definición de un sistema embebido
    c. Características principales
    d. Tecnologíaas: microprocesadores, IC, y de diseño
2. Modelado de sistemas
    a. Niveles de abstracción
    b. Lenguages de descripción de hardware
3. Diseño asistido por computadora
    a. Conceptualización y modelado
    b. Síntesis y optimización
    c. Validación y verificación
4. Sistemas embebidos y tecnología FPGA
    a. Tecnología FPGA
    b. Procesadores “soft”
    c. Perifericos y coprocesadores
5. Co-diseño hardware/software
    a. Preliminares
    b. Administración de diseño
    c. Particionamiento hardware y software
    d. Establecimiento de interfaces
6. Diseño de sistemas empotrados
    a. Requerimientos funcionales y no funcionales
    b. Fases en el diseño
    c. Sistema de buses y de comunicación
7. Sistemas operativos embebidos
    a. Características
    b. Administración de tareas y calendarización
    c. Manejo de interrupciones
    d. Comunicación entre procesos y sincronización
    e. Administración de memoria
8. Desarrollo de proyecto

Bibliografía

1. Giovanni de Micheli, Synthesis and Optimization of Digital Circuits, McGraw-Hill, New York, 1994.
2. D. D, Gajski, N. Dutt, A. Wu, and S. Lin, High-level Synthesis, Kluwer, Boston, 1992.
3. Peter Gun, Kikil Dutt, and Alexandru Nicolau, Memory Architecture Exploration for Programmable Embedded Systems, Kluwer Academic Publishers, 2003.

4. Joseph A. Fisher, Paolo Faraboschi, and Cliff Young, Embedded Computing: A VLIW Approach to Architecture, Compilers and Tools, Morgan Kauffmann, 2005.
5. IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems
6. IEEE Transaction on Very Large Scale Integrated Circuits
7. IEEE Computer Magazine
8. Marwedel, P. Embedded System Design. Ed. Springer, 2003 ISBN: 978-1-4020-7690-9.
9. Frank Vahid, Tony D. Givargis. Embedded System Design: A Unified Hardware/Software Introduction Ed. Wiley, 2002. ISBN: 978-0-471-38678-0.
10. John Catsoulis. Designing Embedded Hardware. Ed. O'Reilly Media. ISBN: 978-0596007553.
11. Qing Li, Caroline Yao. Real-Time Concepts for Embedded Systems. Ed. CMP Books, 2003. ISBN: 978-1578201242.
12. Arnold S. Berger. Embedded Systems Design: An Introduction to Processes, Tools and Techniques. Ed. CMP Books, 2001. ISBN: 978-1578200733

Sistemas de tiempo real

Objetivo

En los sistemas de tiempo real existen 3 componentes principales que los caracterizan, así como la interrelación entre estos. En primer lugar está el tiempo, el cual es el recurso más valioso a tratar y gestionar en un sistema en tiempo real. Las tareas deben ser asignadas y planificadas a fin de que cumplan unos plazos (deadlines). La correcta ejecución del sistema de tiempo real depende tanto de la validez lógica de la respuesta, como del instante de tiempo en que se produce. Como segundo componente se encuentra la confiabilidad, debido a que un fallo en el sistema en tiempo real podría causar serias consecuencias. Estos dos componentes son la principal causa de una mayor dificultad en el diseño de estos sistemas respecto a los sistemas informativos de propósito general. El tercer componente involucra al ambiente en el cual el computador opera, el cual es un componente activo en cualquier sistema de tiempo real. Por ejemplo en un sistema de avión, no tiene sentido considerar el sistema de cómputo por separado de la aeronave, ya que ambos interactúan entre sí.

Contenido

1. Introducción a los Sistemas de Tiempo Real (* nuevo)
2. Aplicaciones de Sistemas de Tiempo Real.
3. Sistemas Operativos.
4. Sistemas Operativos: Componentes y Ejecutivo de Tiempo Real.
5. Procesos Concurrentes.
6. Sincronización de Procesos.
7. Modelo del Sistema
8. Métodos de Planificación
9. Planificación Cíclica
10. Planificación por Prioridades Fijas
11. Planificación Dinámica de Tareas
12. Planificación de Tareas Aperiódicas
13. Overload Scheduling in Real-Time Systems
14. Interacción entre Tareas
15. Tolerancia a Fallos
16. Introduction to Embedded Systems

Teoría de Autómatas

Objetivo

Conocer los elementos básicos de los lenguajes formales, de los dispositivos formales que los reconocen y conocer también a la Jerarquía de Chomski.

Contenido

1. Introducción a la noción de autómata
2. Fundamentos matemáticos
    a) Cadenas y lenguajes
    b) Semigrupo de cadenas
    c) Propiedades de lenguajes
3. Gramáticas formales
    a) Representación de lenguajes
    b) Conceptos básicos de gramáticas
    c) Gramáticas formales
    d) Tipos de gramáticas
    e) Arboles de derivación
    f) Ambigüedad
4. Maquinas de estados finitos
    a) Máquinas de transición y con estados asignados
    b) Conversión entre ambos tipos de máquinas
    c) Equivalencia de máquinas de estados finitos
    d) Reducción de estados
    e) Lenguajes de estados finitos
    f) Reconocedores de estados finitos
    g) Reconocedores deterministicos e indeterministicos y conversión entre ellos
    h) Aplicaciones al diseño de máquinas
    i) Reconocedores de estados finitos y su relación con las gramáticas regulares
    j) Propiedades de los lenguajes de estados finitos
    k) Ambigüedad
5. Limitaciones de autómatas finitos
    a) Limitaciones de reconocedores
    b) Limitaciones de generadores
    c) Limitaciones de traductores
6. Autómatas de cinta
    a) Reconocedores de cinta y gramáticas formales
    b) Máquinas secuenciales generalizadas
    c) Reconocedores de dos modos
    d) Relación de reconocedores de estados finitos
7. Autómatas de pila
    a) Reconocedores de pila
    b) Reconocedores de pila para gramáticas libres de contexto
    c) Construcción de analizadores de sintaxis de pila
    d) Propiedades de cerrado de los lenguajes libres de contexto
    e) Reconocedores de pila determisticos
    f) Reconocedores de conteo
8. Lenguajes libres de contexto
    a) Introducción
    b) Transformación de gramáticas
    c) Formas canonícas de gramáticas
    d) Estructura de los lenguajes libres de contexto
9. Analisis de sintaxis

a) Analisis top-down y bottom-up
b) Análisis de precedencia
c) Análisis gereralizado de izquierda a derecha

10. Máquinas de Turing
    a) Conceptos fundamentales
    b) Simulación de otros autómatas
    c) Transductores de Turing
    d) Reconocedores de Turing
    e) Funciones computables y predicados
    f) Algoritmos y computabilidad efectiva
    g) Máquina universal de Turing

Bibliografía

1. Aho, Hopcroft, Ullman: Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979
2. Denning, JB Dennis, JE Qualitz: Machines, languages and computation, Prentice-Hall, 1978
3. Ginzburg: Algebraic theory of automata, Academic Press, 1968
4. Harrison: Introduction to formal languages theory, Addison Wesley, 1978
5. Salomaa: Formal languages, Academic Press, 1973
6. Salomaa: Automata theory, Pergamon Press, 1969.

Notas del curso:

1. El libro “Teoría de Autómatas”, de Guillermo Morales-Luna, es el texto del curso. Está disponible en varios formatos: PostScript Comprimido, DVI y HTML.

Topicos Selectos de Base de Datos

Objetivo

Familiarizar a los estudiantes con los fundamentos, las bondades, problemas y retos que hay detrás de las BBDD actuales al momento de relacionarse con tecnologías como las Redes de Computadoras, la Orientación a Objetos, las tecnologías de Georeferenciación, el manejo de Almacenes de Datos, la Web y el manejo de datos semi estructurados (XML) y no estructurados. El curso de divide en temas principales en los que se profundizará acorde a las necesidades particulares de investigación en el área.

Contenido

1. Bases de Datos Distribuidas (BDD)
    a. Fundamentos de los Sistemas de BDD (SBDD)
    b. Arquitectura Cliente/Servidor
    c. Diseño de BDD
    d. Procesamiento de consultas
    e. Manejo de Transacciones

2. Bases de Datos Federadas (BDF)
    a. Características de las BDF
    b. Arquitectura de un Sistema de BDF (SBDF)
    c. Construcción de un SBDF

3. Bases de Datos Semi-estructuradas (XML)
    a. Definición de datos semi-estructurados
    b. Introducción a XML

c. Consultando datos con XPath, XQuery y SQL/XML
d. Motores de consulta XML

4. Creación y Explotación de Almacenes de Datos
    a. Introducción a los Almacenes de Datos
    b. Procesamiento Analítico en Línea (OLAP)
    c. Modelos de Datos: ROLAP, MOLAP, HOLAP
    d. Operadores Multidemsionales
    e. Minería de Datos

5. Sistemas de Información Geográfica (SIG)
    a. Fundamentos
    b. Componentes de un SIG
    c. Estándares
    d. Aplicaciones

6. Bases de Datos Objetos-Relacional
    a. Introducción a los Sistemas de Bases de Datos Orientadas a Objetos
    b. Abstracción de Datos
    c. Modelo Orientado a Objetos
    d. Propuestas de Sistemas Gestores de BDOO

Bibliografía

2. Tamer Özsu and Patrick Valduriez. Principles of Distributed Database Systems, 3rd Edition Springer. ISBN-10: 1441988335. Mar 12, 2011.
3. Garcia-Molina, Hector; Ullman, Jeffrey D.; Widom, Jennifer D.DATABASE SYSTEMS The Complete Book. Prentice Hall. ISBN: 0130980439. 2008.
4. Gavin Powell. Beginning XML Databases. Wrox Publisher. ISBN-10: 0471791202. ISBN-13: 978-0471791201. 2006.
5. Dunren Che, Karl Aberer and Tamer Özsu. Query optimization in XML structured-document databases. The International Journal on Very Large Data Bases. Volume 15 , Issue 3 (September 2006). Pages: 263 - 289. ISSN:1066-8888. Springer-Verlag New York, Inc. Secaucus, NJ, USA . 2006
6. Surajit Chaudhuri and Kyuseok Shim. Storage and Retrieval of XML Data Using Relational Databases. Proceedings of the 27th International Conference on Very Large Data Bases. pag.730-740. 2001. ISBN:1-55860-804-4. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA
7. Paul Geoffrey Brown. Object-Relational Database Development: A Plumber's Guide. Prentice Hall. ISBN-10: 0130194603. ISBN-13: 978-0130194602. 2000.
8. Öszu, Tamar and Valduriez, P. Principles of Distributed Database Systems 2nd Ed. Prentice Hall, 1998.
9. Bell, D. A. and Grimson, J. B. Distributed Database Systems. Addisson-Wesley, 1992.
10. N. Mamoulis, T. Seidl, Torben B. Pedersen, K. Torp and I. Assent. Advances in Spatial and Temporal Databases: 11th International Symposium, SSTD 2009 Aalborg, Denmark. Lectures Notes in Computer Science. Springer. 2009.
11. Ceri, S. and Pelagatti, G. Distributed Databes - Principles and Systems. McGraw Hill, 1984.
12. Elmagarmid, A. K. and Pu, C. (guest eds.) ACM Computing Surveys. Special issue on heterogeneous databases, September 1990, 22(3).
13. Öszu, Tamar and Valduriez, P. "Distributed Database Systems: where are we now?", IEEE Computer, August 1991, 24(8).

Tópicos Selectos de Complejidad Computacional

Objetivo

Revisión de los aspectos clásicos de la teoría de los problemas NP-completos y la investigación reciente relacionada con la caracterización de problemas NP-completos.

Contenido

1. Teoría clásica de complejidad computacional tocando aspectos relacionados con la teoría de los problemas NP-Completos, y adicionalmente la revisión actualizada de los avances logrados en la resolución polinomial de versiones restringidas de estos problemas.
2. Teoría de caracterización de problemas NP-Completos, en esta parte se revisaran aspectos relacionados con:
    a) Fenómenos de transición de fase en problemas computacionales (por ejemplo teoría de percolación).
    b) Grafos aleatorios.
    c) Caracterización de redes complejas.
    d) Avances recientes en la caracterización de algunos problemas NP-Completos.

Bibliografía

1. Dietrich Stauffer, Ammon Aharony ”Introduction to Percolation Theory”, CRC Second Edition, 1994, ISBN: 0748402535.
2. Guido Caldarelli, ”Scale-Free Networks: ComplexWebs in Nature and Technology”, Oxford University Press, 2007, ISBN: 0199211515.
3. Mark Newman, Albert-Laszlo Barabasi, Duncan J. Watts, ”The Structure and Dynamics of Networks: (Princeton Studies in Complexity)”, Princeton University Press, 2006, ISBN: 0691113572.
4. Rick Durrett , Random Graph Dynamics”, Cambridge University Press, 2006, ISBN: 0521866561.
5. Bela Bollobas, Random Graphs”, Cambridge University Press Second Edition, 2001, ISBN: 0521797225.
6. M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness”, W. H. Freeman, 1979, ISBN: 0716710455.

Tópicos Selectos de Computación Científica I

Objetivo

Estudiar la teoría y los métodos matemáticos-computacionales para la resolución de algunos problemas de ingeniería, física, química y biología. Se hace énfasis en ecuaciones lineales de orden *n* y sistemas ecuaciones diferenciales de primer orden. Abordando el problema de condiciones iniciales y el problema de valores a la frontera. El problema de Sturm-Liouville es tratado como un problema de valores a la frontera regular y como un problema de valores propios. Se estudia el método de Monte Carlo para simulación y solución de algunos problemas científicos y de ingeniera.

Contenido

1. Espacios lineales y grupo de matrices
    a) Geometría de los espacios lineales
        1) Espacios lineales con producto interior

2) Espacio euclidiano, unitario
3) Espacio, espacio-tiempo de Minkowsky y espacio simplectico

b) Estructura de matrices
1) Espacio vectorial de matrices
2) Polinomios de matrices
3) Problema de valores propios

c) Grupo de matrices
1) El grupos general y subgrupos
2) Topologías de grupos de matrices y grupos de matrices de Lie

2. Herramienta básica del algebra lineal
a) Representación y solución de sistemas lineales
b) Factorización LU
c) Sistema tridiagonal y pentagonal de ecuaciones
d) Solución al problemas de valores propios

3. Ecuaciones diferenciales ordinarias
a) El problema de valores iníciales de EDO
1) El método de Picard
2) Ecuaciones diferenciales de orden n
3) Sistemas de ecuaciones diferenciales

b) El problema de valores a la frontera y de Sturm-Liouville
1) Problema regular de Sturm-Liouville
2) Problema singular de Sturm-Liouville

4. Métodos numéricos para ecuaciones diferenciales
a) Métodos para el problema de valores iniciales
1) Métodos de diferencias finitas
2) Método de Euler
3) Método de Runge-Kutta
4) Método de extrapolación

b) Computación de la exponencial matricial
c) Métodos para el problema de valores a la frontera
1) Método de “Shooting”
2) Método de equilibrio
3) El problema de valores propios

5. Método Monte Carlo
a) Generación de números aleatorios
b) Generación de diversas variables aleatorias continúas
c) Generación de diversas variables aleatorias discretas
d) Pruebas Estadísticas
e) Importancia de muestreo

6. Experimentación y modelos
a) Modelos basados en ecuaciones diferenciales de segundo orden
1) Ecuación Bessel
2) Ecuación de Laguerre
3) Ecuación de Legendre
4) Ecuación de Tchevychev
5) Ecuación de Lotka-Volterra
6) Ecuación de Schrödinger
7) Ecuación de Dirac

b) Modelos basados en Monte Carlo
1) Cadenas de Markov
2) Integración numérica vía método Monte Carlo
3) Camino aleatorio y movimiento browinano
4) Integrales de Wiener
5) Integrales de Ito

6) Integrales de Feyman

Bibliografía

1. Sergio V. Chapa Vergara, Harold V. McIntosh; *"Ecuaciones Diferenciales Ordinarias y Teoría de Weyl, Tomo I: Ecuaciones Diferenciales Escalares y Problema de Sturm- Liouville"*; Ed. Lagares, México D.F. 2005.
2. Sergio V. Chapa Vergara, Harold V. McIntosh, Amílcar Meneses Viveros; *"Ecuaciones Diferenciales Ordinarias y Teoria de Weyl, Tomo III: Teoría de Weyl y aplicaciones a la mecánica cuántica"*; por publicar.
3. Bernd Thaller; *"Visual Quantum Mechanics"*; Springer-Verlag, July 2000.
4. Bernd Thaller; *"Advanced Visual Quantum Mechanics"* Springer-Verlag, 2005.
5. David P. Landau, Kurt Binder; *"A Guide to Monte Carlo Simulations in Statistical Physics"*; Cambridge University Press; 2000.
6. J.D. Hoffman; "Numerical Methods for Engineers and Scientists"; Marcel Dekker; Second Edition, 2001.
7. S.E. Arge, A.M. Bruaset, H.P. Langtangen; *"Modern Software Tools for Scientific Computing"*; Birkhauser; 1997.
8. Dæhlen, A. Tveito; *"Numerical Method and Software Tools in Industrial Mathematics"*; Birkhauser; 1997.

Tópicos Selectos de Computación Científica II

Objetivo

Estudiar los modelos y métodos matemáticos computacionales haciendo énfasis en aspectos algebraicos, geométricos y de visualización. Se estudian teoría de grupos de matrices de rotación con aplicación a algunos problemas físicos. Visualización de geometría del espacio fase en las soluciones de ecuaciones diferenciales. Autómatas Celulares y sistemas dinámicos discretos es tratdo en este curso como un nuevo ambiente para modelar problemas de ecosistemas y físicos.

Contenido

1. Grupo de matrices de rotación
    a) Función de una matriz
    b) Series de Campbell-Hausdorff
    c) Mapeo exponencial de matrices simétricas y antisimétricas
    d) Geometría del grupo de rotación en 3 dimensiones
2. Visualización de Variable compleja
    a) Funciones complejas
    b) Transformación de Mobius
    c) Esfera de Riemann
    d) Funciones especiales
3. Visualización y geometría del espacio fase
    a) Matrices unimodulares y diagramas de bifurcación
    b) Partición en el espacio fase R3
    c) Llenado del espacio del campo direccional en R3
    d) Secuencias de rotación en el espacio fase
    e) Geometría simpectica y el espacio fase
4. Sistemas dinámicos y Autómatas celulares
    a) Autómatas celulares lineales
        1) Análisis de Autómatas Celulares
        a) Diagramas de parejas
        b) Diagramas de subconjuntos
        c) Diagramas de de Bruijn
        b) Autómatas celulares reversibles

c) Computabilidad y autómatas celulares
d) Comportamiento no trivial y Chate Maneville
e) Modelos de reacción y difusión
f) Modelos de coexistencia de dos especies

5. Temas de interés en computación científica
   a) Bases de datos científicas
      1) Modelos de datos
      2) Base de datos deductivas
      3) Consultas dinámicas y análisis exploratorio de datos
   b) Computación simbólica
      1) CONVERT
      2) Bases de Gröbner
   c) Ambientes de visualización científica
      1) Clusters de visualización
      2) Aplicaciones distribuidas para clusters de visualización
      3) Aplicaciones paralelas para clusters de visualización
      4) Modelos para visualización en 3 y 4 dimensiones
      5) GEOM, PHOC y SHOC
   d) Ambientes de computación de alto rendimiento
      1) LAPAC
      2) SLEDGE
      3) SECO y SERO

Bibliografía

1. Sergio V. Chapa Vergara, Harold V. McIntosh; *"Ecuaciones Diferenciales Ordinarias y Teoría de Weyl, Tomo I: Ecuaciones Diferenciales Escalares y Problema de Sturm-Liouville"*; Ed. Lagares, México D.F. 2005.
2. Sergio V. Chapa Vergara, Harold V. McIntosh; *"Ecuaciones Diferenciales Ordinarias y Teoría de Weyl, Tomo II: Sistemas lineales de ecuaciones diferenciales. Un enfoque Geométrico-Matricial"*; por publicar.
3. Sergio V. Chapa Vergara, Harold V. McIntosh, Amílcar Meneses Viveros; *"Ecuaciones Diferenciales Ordinarias y Teoría de Weyl, Tomo III: Teoría de Weyl y aplicaciones a la mecánica cuántica"*; por publicar.
4. Bernd Thaller; *"Visual Quantum Mechanics"*; Springer-Verlag, July 2000.
5. Bernd Thaller; *"Advanced Visual Quantum Mechanics"* Springer-Verlag, 2005.
6. Tristan Needham; *"Visual Complex Analysis"*; Oxford University Press; 1997.
7. Harold V. McIntosh; *"Complex Analysis"*; Universidad Autónoma de Puebla; Puebla, México; April 20, 2001.
8. Harold V. McIntosh; *"Linear Cellular Automata"*; Universidad Autónoma de Puebla; Puebla, México; May 20, 1987
9. Harold V. McIntosh; *"Linear Cellular Automata via de Bruijn Diagrams"*; Universidad Autónoma de Puebla; Puebla, México; May 20, 1990.
10. T. Toffoli, N. Margolus; *"Cellular Automata Machines"*; MIT Press, 1987.
11. S. Wolfram; *"A New Kind of Science"*; Wolfram Media, 2002.
12. J.D. Hoffman; "Numerical Methods for Engineers and Scientists"; Marcel Dekker; Second Edition, 2001.
13. S.E. Arge, A.M. Bruaset, H.P. Langtangen; *"Modern Software Tools for Scientific Computing"*; Birkhauser; 1997.
14. Dæhlen, A. Tveito; *"Numerical Method and Software Tools in Industrial Mathematics"*; Birkhauser; 1997.

Tópicos Selectos en Criptografía

Objetivo

Este curso presenta una selección de los descubrimientos más recientes en criptografía. El curso inicia con un análisis y recuento de las primitivas usadas en criptografía, seguido por una discusión de los esquemas que han sido propuestos recientemente para realizar criptografía simétrica y de llave pública.

Contenido

1. Marco Teórico: Funciones de solo ida, funciones y permutaciones pseudo-aleatorias, generadores pseudo-aleatorios.
2. Criptografía de llave simétrica: Nociones y definiciones de seguridad, modos de operación en cifradores por bloque, códigos de autenticación de mensajes.
3. Criptografía de la llave pública: RSA-OAEP, el problema del logaritmo discreto e hipótesis de Diffie Hellman, criptografía de curva elíptica y criptografía basada en emparejamientos, esquemas de firma.

Bibliografía

1. D. Stinson: Cryptography- Theory and Practice, CRC Press, 2006
2. Menezes, P. van Oorschot, S. Vanstone: Handbook of Applied Cryptography, CRC Press, Boca Raton, 1996
3. Goldreich: Foundations of Cryptography. Cambridge University Press, 2004
4. Michael George Luby: Pseudorandomness and Cryptographic Applications. Princeton University Press, 1996.

Tópicos Selectos en Inteligencia Artificial: Agentes y Multiagentes

Objetivo

Los sistemas multiagentes surgieron en el campo de la investigación de tecnología de la información en la década de los 90. Un agente es un sistema o componente de software, el cual es capaz de cooperar para resolver problemas específicos. El objetivo del curso es dar una visión introductoria a los agentes autónomos y a los sistemas multiagentes desde el punto de vista teórico como práctico. Se explicarán las diferentes arquitectura de agente (reactiva, deliberativa e hibrida), así como los mecanismos de interacción, coordinación y cooperación entre sistemas multiagentes. Las aplicaciones son diversas: control de procesos industriales, comercio electrónico, subastas, etc.

Objetivos específicos:

- Comprender los diferentes enfoques de la Inteligencia Artificial distribuida
- Estudiar los diferentes modelos de sistemas de agentes
- Diseñar y construir un prototipo que muestre el razonamiento llevado a cabo para resolver
- problemas de acuerdo con alguno de los modelos estudiados.

Contenido

1. Introducción
   a) Concepto de agente.
   b) Agentes y objetos.
   c) Agentes y sistemas expertos.
   d) Agentes y sistemas distribuidos.

e) Campos de aplicación típicas.

2. Agentes Inteligentes
   a) Arquitecturas abstractas para agentes.
   b) Diseño de agentes inteligentes.
   c) Mecanismo de razonamiento.
   d) Agentes como sistemas reactivos
   e) Arquitectura hibrida.

3. Mecanismos de Inferencia
   a) Demostración de teoremas
   b) Programación orientada a agentes
   c) Lógicas para sistemas multiagentes
   d) Lógica modal
4. Sistemas multiagentes
   a) Interacción entre agentes: principios de la cooperación.
   b) Sistemas cooperación vs. No cooperativos.
   c) Heurísticas para la cooperación
   d) Coherencia y coordinación
   e) Negociación y argumentación
   f) Aplicaciones: subasta, comercio electrónico
   g) Criterios de evaluación
5. Comunicación

a) Lenguajes de comunicación de agentes
b) Protocolos KQML /KIF
c) Ejemplos de aplicación

Bibliografía

1. Bordini, Rafael H et al., Multiagent programming: Languages, platforms, and applications, Springer, New York, 2005.
2. Jaques Ferber, Multiagent Systems. An introduction to Distributed Artificial Intelligence, Addison Wesley, NY, 1999.
3. Fagin Ronald, Reasoning about knowledge, Cambridge Mass: MIT Press, 1995.
4. Fisher, M., and Wooldrige, M., “On the formal specification and verification of multiagent systems”, International Journal of Cooperative Information Systems, 6(1), pp. ,1997.
5. Mas Ana, Agentes de software y sistemas multiagentes, Pearson-Prentice Hall, UK, 2005.
6. Michael Wooldridge, An Introduction to Multiagent Systems, John Wiley, , 2002.
7. FIFA, “Fipa specification version 2.0”, Technical report, FIPA, part (2), Foundation for Intelligent Physical Agtent 1998.
8. Wooldrige M., Jennings N. R., “Intelligent Agents: Theory and Practice”, The Knowledge Engineering Review, Vol 10(2), pp. 115-152, 1995.

Tópicos selectos en Inteligencia Artificial: Introducción a la optimización evolutiva multiobjetivo

Objetivo

En este curso se estudiarán los conceptos básicos de la optimización multiobjetivo, así como el uso de los algoritmos evolutivos en esta área. El material cubierto abarca desde los orígenes de la optimización multiobjetivo (en economía y planeación), hasta

los avances más recientes. Además de analizar las técnicas evolutivas multiobjetivo de mayor uso en la actualidad, se estudiarán otras heurísticas que también han sido extendidas para lidiar con problemas multiobjetivo (p.ej., la colonia de hormigas), discutiendo sus ventajas y limitantes principales. Adicionalmente, se revisará el trabajo teórico realizado en esta área y se discutirán algunos de los temas de investigación futura que han permanecido poco explorados durante los últimos años.

Contenido

1. Conceptos Básicos
    a) Atributos, metas, criterios y objetivos
    b) Definición de un problema multiobjetivo
    c) Tipos de problemas multiobjetivo
    d) Vector ideal
    e) Convexidad y concavidad
    f) Optimo de Pareto
    g) Dominancia de Pareto y conjunto optimo de Pareto
    h) Frente de Pareto
2. Antecedentes Históricos
    a) Orígenes de la optimización multiobjetivo
    b) Clasificación de técnicas
    c) Revisión rápida de enfoques usados en investigación de operaciones
3. Algoritmos Evolutivos
    a) Motivación para resolver problemas multiobjetivo
    b) Técnicas basadas en funciones de agregación (lineales o no lineales)
    c) Técnicas poblacionales
    d) Técnicas basadas en jerarquización de Pareto
    e) Otras técnicas
4. Técnicas para Mantener Diversidad
    a) Nichos y compartición de aptitud
    b) Operadores de agrupamiento (crowding)
    c) Otros esquemas
5. Funciones de Prueba
    a) ¿Cómo diseñarlas adecuadamente?
    b) Ejemplos sin restricciones
    c) Ejemplos con restricciones
    d) Optimización combinatoria
    e) Problemas del mundo real
    f) Problemas que no se han abordado
6. Métricas
    a) ¿Cómo comparar dos algoritmos multiobjetivo?
    b) Cantidad de elementos del conjunto de Pareto
    c) Dispersión
    d) Cercanía al verdadero frente de Pareto
    e) Métodos estadísticos
    f) Otro tipo de métricas
    g) Limitantes de las métricas
7. Teoría
    a) Conjuntos parcialmente ordenados
    b) Convergencia de algoritmos evolutivos multiobjetivo
    c) Nichos y otros métodos para mantener diversidad
    d) Restricciones a la cruza
    e) Análisis de complejidad de los principales algoritmos evolutivos multiobjetivo
    f) Costo computacional
8. Algoritmos Evolutivos Multiobjetivo Paralelos
    a) Filosofía

b) Paradigmas
c) Ejemplos

9. Toma de Decisiones Multicriterio
   a) Actitud del tomador de decisiones
   b) Incorporación de preferencias en algoritmos evolutivos multiobjetivo
   c) Puntos a tomar en consideración
10. Otras heurísticas multiobjetivo
   a) Recocido simulado
   b) Búsqueda tabú
   c) La colonia de hormigas
   d) Aprendizaje por refuerzo
   e) Algoritmos meméticos
   f) Optimización mediante cúmulos de partículas
   g) Técnicas adicionales (algoritmos culturales, sistema inmune artificial, búsqueda cooperativa, etc.)
11. Áreas de investigación futura
   a) Toma de decisiones
   b) Nuevos algoritmos
   c) Teoría
   d) Nuevas heurísticas
   e) Nuevas métricas
   f) Búsqueda local
   g) Estructuras de datos espaciales para poblaciones secundarias
   h) Eficiencia
   i) Ideas no exploradas

Bibliografía

1. (Libro de texto): Coello Coello, Carlos A.; Van Veldhuizen, David A. & Lamont, Gary B. “Evolutionary Algorithms for Solving Multi-Objective Problems”, Kluwer Academic Publishers, New York, ISBN 0-3064-6762-3, May 2002.
2. http://delta.cs.cinvestav.mx/ccoello/EMOO

Tópicos Selectos en Inteligencia Artificial: Razonamiento Aproximado

Objetivo

El tratamiento de la incertidumbre constituye uno de los campos fundamentales de la Inteligencia Artificial (IA), ya que esta presente en todas las aplicaciones de la IA (sistemas expertos, control automático, aprendizaje por una maquina, etc). En el curso se estudiaran los diferentes modelos de razonamiento aproximado; se desarrollara una base de conocimientos; y se diseñara y construirá un prototipo que muestre el razonamiento aproximado, de acuerdo con alguno de los modelos estudiados.

Contenido

1. Introducción
   a) Fuentes de incertidumbre
   b) Evolución histórica del razonamiento incierto
   c) Conjuntos, algebra y estadística borrosa
2. Modelo probabilista clásico
   a) Conceptos sobre probabilidad
   b) Independencia, correlación y causalidad
   c) Teorema de Bayes
   d) Ejemplos de sistemas

3. Redes bayesianas
    a) Definición de red bayesiana
    b) Algoritmos de propagación
    c) Tipo de reglas válidas
    d) Prospector como ejemplo clásico
4. Modelo de factores de certeza
    a) Definición de los factores de certeza
    b) Propagación de la incertidumbre en una red de inferencia
    c) Creencia absoluta y actualización de creencias
    d) MYCIN como ejemplo clásico
5. Modelo posibilista
    a) Teoría de la posibilidad
    b) Proposiciones clásicas, vagas y compuestas
    c) Patrones de inferencia
    d) Tratamiento de la vaguedad e incertidumbre en las reglas
    e) Teoría de la posibilidad vs teoría de la probabilidad
    f) Milord como ejemplo clásico

Bibliografía

1. Gerla, Giangiacomo (2001) Fuzzy logic: mathematical tools for approximate reasoning,
2. Leondes, Cornelius T. ed. (2000) Knowledge-based systems: techniques and applications,
3. Lopez de Mantaras R. (1990) Approximate Reasoning Models, Ellis Horwood Limited, John Wiley & Sons, 1990
4. Pearl, Judea, (1986) A constraint- propagation approach to probabilistic reasonning, in Kanal L. N. & Lemmer J.F. (eds) Uncertainty in Artificial Intelligence, North-Holland, Amsterdam, 357-369
5. Zadeh L. A. (1983) The Role of Fuzzy Logic in the Management of Uncertainty in Expert Systems, Fuzzy Sets and Systems, 11, pp 199-277.
6. Zadeh L. A. (1979) A Theory of Approximate Reasoning, Machine Intelligence, 9. pp. 149-194.

Tópicos selectos de Inteligencia Artificial: Sistemas de Soporte a la Toma de Decisiones

Objetivo

4. Estudio de metodologías y herramientas, matemáticas y computacionales, para el análisis, diseño y desarrollo de sistemas de soporte a la Toma de Decisiones.
5. Estudio de las metodologías y herramientas para el análisis, diseño y desarrollo de los procesos involucrados en la Toma de Decisiones.
6. Estudio de los lenguajes para el análisis, diseño y desarrollo de sistemas de soporte a la Toma de Decisiones.
7. Estudio las técnicas de Inteligencia Artificial, de Teoría de Juegos y de Administración del Conocimiento, con las cuales se fundamenta el desarrollan de sistemas inteligentes para la Toma de Decisiones.

Contenido

1. Introducción y antecedentes de la Toma de Decisiones
2. Modelación formal de la Toma de Decisiones

a. álgebras de conjuntos para la Toma de Decisiones
b. Lógicas multi-valuadas para la Toma de Decisiones

3. Análisis y diseño de Ontologías para la Toma de Decisiones
4. Análisis, diseño y programación de procesos dinámicos para la Toma de Decisiones
5. Técnicas de Inteligencia Artificial para la Toma de Decisiones
6. Técnicas de Teoría de Juegos para la Toma de Decisiones
7. Técnicas de Administración del Conocimiento para la Toma de Decisiones
8. Desarrollo de casos de estudio
9. Análisis y diseño de pruebas
10. Puesta en marcha de sistemas de Toma de Decisiones

Bibliografía

1. *Mathematics of Decision Making*

a. Vol. I: Statistical Modeling & Analysis for Complex Data Problems;
b. Vol. II: Logistics Systems-Design & Optimization;
c. Vol. III: Energy & Environment;
d. Vol. IV: Analysis, Control & Optimization of Complex Dynamic Systems;
e. Vol. V: Column Generation;
f. *f*) Vol. VI: Performance Evaluation & Planning Methods for the Next Generation Internet;
g. Vol. VII: Essays & Surveys in Global Optimization;
h. Vol. VIII: Graph Theory & Combinatorial Optimization;
i. Vol. IX: Numerical Methods in Finance;
j. Vol. X: Dynamic Games: Theory & Applications. *Zaccour, Georges (Ed.) Springer Verlag, 2005, 3242 p. (10-volume-set). ISBN: 978-0-387-26117-1.*

Tópicos Selectos en Redes de Computadoras

Objetivo

En este curso se estudian las técnicas más recientes para la construcción de redes de computadoras y protocolos de comunicación. Se revisan las tecnologías y protocolos de comunicación utilizadas en redes inalámbricas y móviles. Se analizan los requisitos de servicio que demandan las nuevas aplicaciones multimedia. Se estudian las arquitecturas y tecnologías para el aprovisionamiento de calidad de servicio en redes de próxima generación. Se describen los conceptos fundamentales de gestión de redes y servicios, como un instrumento para el aprovisionamiento de calidad de servicio. Se describen las tecnologías emergentes para realizar la automatización de las tareas de gestión de red en ambientes dinámicos. Se realizan simulaciones mediante software especializado con el fin de aplicar los conocimientos adquiridos en el curso.

Contenido

1. Redes Inalámbricas y Móviles.
    a. Características de redes inalámbricas,
    b. redes inalámbricas 802.1X,
    c. estándares y tecnología celular,
    d. movilidad en redes celulares

2. Transmisión de Datos Multimedia
    a. Aplicaciones multimedia,

   b. streaming de audio y video,
   c. protocolos para aplicaciones de tiempo real interactivas

3. Calidad de Servicio en Redes de Próxima Generación.
   a. Parámetros de calidad de servicio,
   b. mecanismos de “scheduling” y “policing”,
   c. arquitectura de servicios diferenciados (DiffServ) y de servicios integrados (IntServ).

4. Gestión de Redes.
   a. Conceptos generales de gestión de redes y servicios,
   b. infraestructuras y marcos de referencia para gestión de red,
   c. gestión de red y aprovisionamiento de calidad de servicio en redes de próxima generación.

5. Automatización en la Gestión de Redes y Servicios.
   a. Introducción a las comunicaciones y computación autónomas,
   b. gestión basada en políticas,
   c. caso de uso de automatización de gestión de redes y servicios de próxima generación.

Bibliografía

1. Computer Networking: A Top-Down Approach, 5th Edition. James F. Kurose, Amhersteith W. Ross. ISBN-10: 0136079679, ISBN-13: 9780136079675. Ed. Addison-Wesley 2010
2. New Horizons in Mobile and Wireless Communications, Volume 1: Radio Interfaces. Ramjee Prasad and Albena Mihovska. Ed. Artech House. ISBN 978-1-60783-967-5, 2009
3. New Horizons in Mobile and Wireless Communications, Volume 2: Networks, Services and Applications. Ramjee Prasad and Albena Mihovska. Ed. Artech House. ISBN 978-1-60783-969-9, 2009
4. New Horizons in Mobile and Wireless Communications, Volume 3: Reconfigurability. Ramjee Prasad and Albena Mihovska. Ed. Artech House. ISBN 978-1-60783-971-2, 2009
5. New Horizons in Mobile and Wireless Communications, Volume 4: Ad Hoc Networks and PANs. Ramjee Prasad and Albena Mihovska. Ed. Artech House. ISBN 978-1-60783-973-6, 2009
6. SIP: Understanding the Session Initiation Protocol, Third Edition. Alan B. Johnston. Ed. Artech House. ISBN 978-1-60783-995-8, 2009
7. Broadband Wireless Access & Local Networks: Mobile WiMAX and WiFi. Byeong Gi Lee and Sunghyun Choi. Ed. Artech House. ISBN 978-1-59693-294-4, 2008
8. Practical Radio Resource Management in Wireless Systems. Sofoklis Kyriazakos and George Karetsos. Ed. Artech House. ISBN 978-1-58053-633-2, 2004
9. Artificial Intelligence in Wireless Communications. Thomas W. Rondeau and Charles W. Bostian. Ed. Artech House. ISBN 978-1-60783-235-5, 2009
10. Utility Computing Strategies, Standards, and Technologies. Alfredo Mendoza. ISBN 978-1-59693-025-4, 2007
11. OPNET IT Guru, OPNET Modeler, OPNET Modeler Wireless Suite, and Specialized modules for Flow Analysis, MPLS, UMTS and WiMax and Documentation www.opnet.com

Tópicos Selectos en Redes Neuronales Artificiales

Objetivo

Comprender y aplicar los métodos neuronales a la resolución de problemas complejos inspirándonos en el funcionamiento de nuestro cerebro para su concepción modular, local, distribuida y paralela con el fin de crear sistemas fácilmente adaptables e integrables a otros.

Contenido

1. Neuromimetismo (modelación en neurociencias)
    a) Mecanismos inspirados biológicamente
    b) Sistemas ópticos biológicamente inspirados
    c) Campos receptivos
2. Niveles de percepción
    a) Filtrado de bajo nivel (brillo, iluminación y contraste)
    b) Multi-filtrado espacial (ilusiones ópticas)
    c) Niveles de atención
3. Codificación y decodificación neuronal
    a) El sistema visual
    b) Métodos de correlación inversa: células simples
    c) No linearidades estáticas: células complejas
    d) Campos receptivos en la retina y en el LGN
    e) Construcción de los campos receptivos en V1
4. Neuronas y circuitos neuronales
    a) Modelos de integración y disparo
    b) Conductancias dependientes del voltaje
    c) Modelo de Hodgkin-Huxley
    d) Redes recurrentes
    e) Redes excitatorias-inhibitorias
    f) Redes estocásticas
5. Adaptación y aprendizaje
    a) Reglas de plasticidad sináptica
    b) Modelos causales para la estimación de la densidad

Bibliografía

Libros de base

1. Peter Dayan y L. F. Abbott, *Theoretical neuroscience: Computational and mathematical modelling of neural systems*, 460 pages, MIT Press, 2001, ISBN: 9780262541855
2. G. N. Reeke, R.R. Poznanski, K. A. Lindsay , J.R. Rosenberg y O. Sporns, *Modeling in the Neurosciences: From Biological Systems to Neuromimetic Robotics*, CRC; 2 edition (March 29, 2005), ISBN-10: 0415328683, ISBN-13: 978-0415328685.
3. Yoseph Bar-Cohen, *Biomimetics: Biologically Inspired Technologies*, CRC (November 2, 2005), ISBN-10: 0849331633, ISBN-13: 978-0849331633.
4. Anthony Brabazon y Michael O'Neill, *Biologically Inspired Algorithms for Financial Modelling* (Natural Computing Series), Springer; 1 edition (February 10, 2006), ISBN-10: 3540262520, ISBN-13: 978-3540262527.
5. Eugene M. Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*, The MIT Press ISBN 978-0-262-09043-8, 2007
6. *f*) Chris Eliasmith and Charles H. Anderson, *Neural Engineering: Computation, Representation and Dynamics in Neurobiological Systems*, The MIT Press 2003.
7. Bin He, *Neural Engineering*, 488 pages, Springer 1 edition (March 4, 2005), ISBN- 13: 978-0306486098.

Profundizar :

1. Chistof Koch, Biophysics of computation: Information processing in single neurons, 562 pages, Oxford University Press, 1999, ISBN: 9780195181999.
2. Arjen van Ooyen, Modelling Neural Development, A Bradford Book, The MIT Press, USA, 2003, ISBN 0-262-22066-0.
3. Moutier, Sylvain, Inhibition neuronale et cognitive, Lermes Science Publications, Lavoisier, 2003, ISBN 2-7462-0771-0.
4. Klaus Ober-mayer and Terrence J. Sejnowski, Self-Organizing Map Formation: Foundation of Neural Computation, The MIT Press 2001.

Tópicos selectos en sistemas digitales: VHDL

Objetivo

Se describen los varios estilos de diseño en VHDL así como sus niveles de abstracción. Se enfatizan, en forma muy particular, las características que permiten al lenguaje VHDL diseñar circuitos digitales, como también la capacidad de crear diseños transportables. Se analizan la sintaxis y la síntesis como el análisis sobre su mapeo en FPGA y la posibilidad de rediseño para corregir errores y mejorar sus características funcionales.

Contenido

1. Introducción a la aplicación y al lenguaje VHDL
2. Aplicación de VHDL y estilo de uso
3. Conceptos importantes del lenguaje y sus constructores principales
4. Proceso de diseño electrónico
5. Especificaciones del modelo
6. Estilos de VHDL
7. Abstracción para definir detalles.
8. Nivel de layout.
9. Nivel lógico.
10. Nivel de transferencia entre registros.
11. Nivel de comportamiento.
12. Conceptos principales del lenguaje.
13. Sentencias concurrentes y secuenciales.
14. Entidad y arquitectura.
15. Asignación de señales e instancia de componentes.
16. Procesos y tipos.
17. Compilación y análisis.
18. Configuración y mapeo.

Bibliografía

1. Allen Dewey, Analisis and design the Digital Systems with VHDL, XILINX ISE.
2. Roth, Charles H. Digital design using VHDL.
3. Navabi, Zanalabcdin, Analysis and modelling of design systems.

Tópicos Selectos en Teoría de Códigos

Objetivo

En la última década hemos presenciado numerosos y significativos avances en la

teoría de códigos. El material de este curso se propone motivar el conocimiento de la teoría de códigos, así como presentar algunos de los últimos avances alcanzados en esta disciplina. El curso inicia con una introducción a la teoría de la información de Shannon para después discutir y analizar las propiedades y cotas teóricas de códigos específicos de corrección de error.

Contenido

1. Entropía su caracterización y sus propiedades, códigos Huffman, códigos Shanon-Fano, robustez de las técnicas de codificación, codificación libre de ruido, canal discreto sin memoria y capacidad de canal, teorema fundamental de la teoría de la información.
2. Códigos de corrección de error, principio de la mínima distancia, códigos lineales, cotas de Hamming, cota de Singleton, códigos Reed, códigos BCH, decodificación BCH, decodificación por lista.

Bibliografía

1. F.J. MacWilliams and Neil J.A. Sloane: Theory of Error Correcting Codes. Elsevier/North Holland, Amsterdam, 1981.
2. Vera S. Pless and W. Carry Huffman (Eds.): Handbook of Coding Theory (2 volumes), Elsevier 1998
3. Jacobus H. van Lint: Introduction to Coding Theory. Graduate Texts in Mathematics 86, Springer- Verlag, Berlin, 1999
4. Richard E. Blahut: Theory and practice of error control codes. Addison-Wesley, Reading Massachusetts, 1983.

Tópicos Selectos en Visualización

Objetivo

En este curso se aplicarán las nociones de visión por computadora (VC) en tres dimensiones. La meta de VC es deducir las propiedades y estructura de un mundo tridimensional a partir de una o más vistas bidimensionales. Primero se estudiarán algunas técnicas para procesamiento y análisis de imagen y también se tratarán temas de visualización 3D, animación y realidad virtual, para la creación de modelos tridimensionales y para tener la habilidad de "navegar" a través de ellos. Las herramientas de trabajo serán la librería de procesamiento de imágenes scimagen, y Qt (www.trolltech.com) para el desarrollo de las interfaces gráficas y Mesa (www.mesa3d.org) para interactuar con objetos tri-dimensionales.

Contenido

1. Introducción al Procesamiento de Imagen
   a) Representación de una Imagen digital
   b) Modelo general para el procesamiento de imágenes
   c) Elementos de un sistema de procesamiento digital de imágenes: adquisición, almacenamiento, una computadora, comunicación, despliegue y software.
2. Fundamentos de Imágenes Digitales
   a) Un modelo simple de imagen
   b) Muestreo y cuantización
   c) Relaciones entre pixels: vecinos, conectividad, distancia, operaciones aritméticas/lógicas.
   d) Geometría de imágenes: transformaciones y proyecciones.
   e) Filtrado espacial.
3. Segmentación de Imagen
   a) Detección de discontinuidades
   b) Umbralización
4. Representación de la forma y reconocimiento del objeto

a) Esquemas de representación. El esqueleto de una región. Códigos de cadena.
b) Descriptores de fronteras: momentos.
c) Morfología.
d) Elementos de análisis de imagen
e) Métodos de decisión teórica: emparejamiento (matching), clasificadores óptimos estadísticos, redes neuronales.

5. Algoritmos geométricos
   a) Algoritmo optimo para encontrar el par de puntos más cercano.
   b) Algoritmos para encontrar la cubierta convexa (convex hull)
   c) La lista de aristas doblemente ligada
   d) Algoritmo optimo para encontrar el diagrama de Voronoi.
6. Reconstrucción del volumen a partir de líneas de contornos
   a) Extracción de las líneas de contorno
   b) Muestreo de las curvas extraídas
   c) Triangulación y visualización.
7. Reconstrucción a partir de dos o más vistas
   a) Calibración de una cámara. Parámetros intrínsecos y extrínsecos
   b) Estimación de los parámetros de una cámara
   c) Detección de esquinas basado en la trasformada de Hough (5)
   d) Triangulación para recobrar la tercera dimensión
   e) Geometría equipolar. El tensor trifocal. Soluciones no-lineales

Bibliografía

1. F.P. Preparata and M.I. Shamos. Computational Geometry, Springer-Verlag. 1985.
2. E. Trucco and A. Verri, Introductory Techniques for 3D Computer Vision 1998, Prentice Hall.
3. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision 2nd edition, 2003, Cambridge
4. R. Johnsonbaugh. Discrete Mathematics. 4th ed. Prentice Hall. 1997
5. Corner detection based on modified Hough transform F. Shen and H. Wang, Pattern Recognition Letters 23 (2002) 1039-1049
6. R.C. Gonzalez and R.E. Woods, Digital Image Processing 1992, Addison Wesley
7. Abigail Martínez Rivas, Reconstrucción del volumen a partir de su mapa de contornos. Tesis de Maestría. 2005 Sección de Computación, Cinvestav.

Visión por computadora

Objetivo

Se revisará la teoría para la reconstrucción tridimensional de escenas a partir de una o varias imágenes bidimensionales, tomadas por una cámara convencional. Se hará énfasis en los métodos para obtener la reconstrucción a partir de las correspondencias de puntos entre las imágenes, lo que se conoce como *autocalibración de la cámara*.

Contenido

1. Geometría proyectiva
2. Métodos numéricos

a) Propiedades de la descomposición en valores singulares (SVD)
b) El método de Gauss-Newton
c) El método Levenberg-Marquardt
d) Heurísticas para optimización nolineal

3. Modelo para la cámara obscura. Parámetros intrínsecos y extrínsecos de la cámara
4. Autocalibración de la cámara con homografías.
5. Autocalibración de la cámara usando cuboides.
6. La geometría epipolar
7. El tensor trifocal
8. Reconstrucción usando varias cámaras
9. Visualización de la reconstrucción con mapeo de texturas

Bibliografía

1. R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision 2nd edition, 2003, Cambridge
2. E. Trucco and A. Verri, Introductory Techniques for 3D Computer Vision 1998, Prentice Hall.

Seminario de Doctorado

Objetivo

Revisar los avances de los proyectos de investigación de todos los estudiantes inscritos al programa de doctorado.

Descripción

En este curso se revisa el avance de cada investigación doctoral que se está llevando a cabo en el programa. El trabajo del seminario de doctorado consiste de dos exposiciones, participación en el seminario y la entrega de un reporte escrito en inglés con la investigación desarrollada en el último año de actividades. Para el caso de estudiantes que tienen menos de un año, será a partir de su fecha de ingreso. Cada estudiante de doctorado formalmente inscrito deberá participar en el seminario exponiendo los avances en su investigación doctoral.

Organización

1. La primera exposición será de 20 minutos y presentará solo los aspectos más relevantes del trabajo de investigación hasta llegar al planteamiento del problema. Se programarán cuatro exposiciones por sesión. El contenido de la presentación deberá incluir al menos los puntos siguientes:
    a. Presentación
    b. Antecedentes
    c. Motivación
    d. Planteamiento del problema
    e. Metodología para resolver el problema
    f. Calendario de actividades desde el inicio de la investigación doctoral.

2. En la segunda ronda se programarán dos exposiciones por sesión. Éstas serán de 40 minutos + 15 minutos para preguntas y deberán organizarse de la siguiente manera:
    a. Presentación
    b. Revisión del Planteamiento del Problema y Metodología destacando la situación en la que actualmente se encuentra la investigación.
    c. Soluciones o enfoques reportados por otros autores

d. Solución(es) o enfoque(es) propuesto(s) en la investigación
e. Resultados más importantes
f. Publicaciones que han sido originadas por la investigación
g. Trabajo a futuro (actividades a realizar para concluir la investigación incluido un calendario)
h. Conclusiones

NOTA: Dependiendo del grado de avance, cada estudiante junto con su supervisor podrán hacer una breve presentación del funcionamiento de algún prototipo computacional.

3. En la segunda exposición se contará al menos con la presencia del Supervisor de Tesis, de un profesor invitado y del titular del seminario. La invitación queda abierta para que todos los profesores asistan a todas las presentaciones. Cada uno de los tres profesores mencionados llenará al final de cada exposición un formato con sus impresiones acerca de la calidad y profundidad de la presentación, calidad de los resultados y recomendaciones u observaciones sobre el trabajo. Las hojas de evaluación se harán llegar al supervisor del trabajo.

4. Con el propósito de que los propios estudiantes de doctorado conozcan los aspectos que están siendo evaluados, se pedirá también que uno de los estudiantes de doctorado llene una hoja de evaluación.

5. Las presentaciones se colocarán en una página electrónica dedicada al seminario de estudiantes de doctorado.

6. La asistencia al seminario de los estudiantes de doctorado es obligatoria para todas las presentaciones.