Development of Computational Thinking in Discrete Mathematics Training

Desenvolvimento do Pensamento Computacional em la Formação em Matemática Discreta

Desarrollo del Pensamiento Computacional en la Formación en Matemática Discreta

Augusto Flores P.
Politécnico Internacional
augustoflores@politecnico.cr

(Artículo de **REFLEXIÓN**. Recibido el 11/02/2011. Aprobado el 28/04/2011)

Abstract
This document regards to the development of computational thinking in the formation in discrete mathematics. In the first place, details four main components of computational thinking: abstract thought, logical thought, modeling thought and constructive thought. In the second place, is describing part of the content of discrete mathematics that has close relationship *with the computational thinking, through a correspondent example of application. Finally, is make a mapping of knowledge units of discrete mathematics with details of the correspondent computational thinking.*

Keywords: Discrete Mathematics, computational thinking, training in mathematics, unit of knowledge, constructive thinking.

Resumo
Este documento refere-se ao desenvolvimento do pensamento computacional na formação em matemática discreta. Primeiro, detalhada quatro componentes principais do pensamento computacional, pensamento abstrato, raciocínio lógico, *pensamento*, modelagem e pensamento construtivo. Em segundo lugar, descreve alguns dos conteúdos de matemática discreta, *que* está intimamente relacionado com o pensamento computacional, através de uma aplicação exemplo. Finalmente, existe um mapeamento de unidades de conhecimento de matemática discreta com detalhes do pensamento correspondente computacional.

Palavras-chave*:* Matemática discreta, *pensando* computacionais, formação matemática, unidade de conhecimento, pensamento construtivo.

Resumen
Este documento se refiere al desarrollo del pensamiento computacional en la formación en matemáticas discretas. En primer lugar, se detallan cuatro componentes principales del pensamiento computacional: pensamiento abstracto, pensamiento lógico, pensamiento modelado y pensamiento constructivo. En segundo lugar, se describe parte del contenido de las matemáticas discretas, que tiene estrecha relación con el pensamiento computacional, a través de un ejemplo de aplicación correspondiente. Por último, se hace un mapeo de las unidades de conocimiento de las matemáticas discretas con los detalles subsecuentes del pensamiento computacional.

Palabras clave: Matemática discreta, pensamiento computacional, formación matemática, unidad de conocimiento, pensamiento constructivo.

1. Introducción

La matemática discreta es una rama de la matemática aplicada que se ocupa de los arreglos de objetos discretos que están separados unos de otros, tales como números enteros, números reales, proposiciones, conjuntos, relaciones, funciones y grafos [1]. Tiene muchas aplicaciones en las ciencias computacionales y la ingeniería del software, por ejemplo, cómo buscar información útil para un ingeniero, cómo describir la estructura estática y el comportamiento dinámico de un sistema de software, y cómo verificar una especificación de software mediante declaraciones lógicas, entre otras. Con el fin de que el maestro logre los objetivos de formación en matemáticas discretas, y específicamente para los estudiantes que se especializan en informática, en este trabajo se presenta una idea para introducir el "pensamiento computacional" en los procesos formativos de las matemáticas discretas.

El término "pensamiento computacional" ha sido acuñado para describir la forma como piensa un científico computacional [2]. El pensamiento computacional se ha convertido en una habilidad fundamental, clasificado junto a la lectura, la escritura y la aritmética, que se pueden encontrar en todas las temáticas [3]. En el caso de los sistemas biológicos, significa la habilidad para reunir las múltiples abstracciones que la biología molecular ha acumulado. En las ciencias computacionales, puede ayudar a las personas a comprender y construir un sistema informático para resolver un determinado problema.

En este trabajo, se introduce “el pensamiento computacional” en la formación en matemática discreta. En la sección II, se describe brevemente la aplicación de las matemáticas discretas y el pensamiento computacional; en la sección III, se ilustra en detalle el contenido del pensamiento computacional, y se da una definición para la matemática discreta; en la sección IV, se presenta el contenido del currículo de matemáticas discretas, diseñado por ACM/IEEE Computing Curricula 2005 [4].

En la sección V, se entregan un mapeo de la Matemática Discreta para el Pensamiento Computacional.

2. Contenido del pensamiento computacional

2.1 Pensamiento abstracto

El pensamiento abstracto es fundamental en la informática y la tecnología para comprender el cuerpo principal del problema de los computadores. Pensar en abstracto es una interesante heurística de propósito muy general que puede ayudar a enfrentar la solución de un problema. Informalmente, el pensamiento abstracto se puede considerar como el mapeo de una representación base para una nueva pero más simple representación [5]. La representación abstracta es más sencilla porque el mapeo por lo general ofrece detalles pero conserva ciertas propiedades deseables, y traduce el problema viejo en un problema nuevo que puede resolverse con nuestro conocimiento [6].

Definición 1. Sistema formal: Un sistema formal Σ es una tripleta (L, Ω, Δ), en la que L es el lenguaje del dominio especificado, Ω es el conjunto de axiomas acerca de las reglas utilizadas, y Δ es el mecanismo deductivo de Σ.

Por lo general, un lenguaje es definido por el alfabeto, y es un conjunto de términos bien formados y el conjunto de fórmulas bien formadas que pueden construir el lenguaje de dominio. Los axiomas son las fórmulas de base bien formadas, es decir $\Omega \subseteq L$. El mecanismo deductivo es el conjunto de reglas de inferencia que puedan inducir nuevos teoremas desde los ya existentes.

Definición 2. Abstracción: Una abstracción es una tripleta, $A = (\Sigma_1, \Sigma_2, f)$, en la que Σ_1 y Σ_2 son un sistema formal, y f es una función que mapea el lenguaje de Σ_1 en el de Σ_2.

Definición 3. Pensamiento abstracto: El pensamiento abstracto es un método de pensamiento que resuelve un nuevo problema mediante abstracción, y que traduce el problema fuente $\ell_1 \in \Sigma_1$ en un problema fuente $\ell_2 = f(\ell_1 \in \Sigma_2)$, y luego utiliza algunos axiomas y el mecanismo deductivo de Σ_2 para resolver ℓ_2.

2.2 Pensamiento lógico

El pensamiento lógico es el proceso en el que se utiliza la consistencia de razonamiento para llegar a una conclusión. Algunos problemas o estados de computador –situaciones– que involucran al pensamiento lógico siempre invocan la estructura matemática, para las relaciones entre algunas hipótesis y las declaraciones dadas, y para la secuencia de razonamiento que hace alguna conclusión más razonable.

El núcleo y la base de todo pensamiento lógico es el pensamiento secuencial que organiza una serie de declaraciones en una cadena, en la que el primer elemento representa la conclusión anterior; el proceso de pensamiento secuencial consiste en tomar algunas declaraciones en una progresión como una cadena que adquiere un significado en y de la misma. Pensar lógicamente es construir paso a paso algunos enfoques.

Se ha demostrado que la formación lógica de la matemática discreta puede hacer más inteligentes y meticulosos a los estudiantes de computación. Un estudiante que tiene la capacidad de pensar lógicamente no concibe las respuestas rápidas para algunos problemas computacionales, tal como "es demasiado difícil", o "no sé". Por el contrario, aplicará el pensamiento lógico para profundizar en el problema propuesto y comprender mejor el método y llegar a una solución. El pensamiento lógico no es un proceso mágico o una cuestión de herencia genética, sino un sabio proceso mental que se imparte en el proceso formativo en matemáticas discretas.

2.3 Pensamiento modelado

Este pensamiento, en el uso técnico del término, se refiere a la traducción de objetos o fenómenos del mundo real en ecuaciones matemáticas, y/o relaciones computacionales. Consiste en seleccionar una representación apropiada o modelar los aspectos relevantes de un problema para hacerlo manejable. El modelado computacional es la representación de objetos reales en un computador. Un problema que será resuelto utilizando el computador debe ser modelado mediante un modelo de software correspondiente.

El modelado computacional es un método matemático y computacional para resolver problemas del mundo real. En virtud del pensamiento modelado, los estudiantes pueden experimentar procesos de resolución de problemas. Además, pueden aprender cómo identificar un problema, construir o seleccionar modelos apropiados, averiguar qué datos deben recopilar, probar la validez de un modelo, calcular soluciones e implementar el modelo. Con el fin de promover la creatividad de los estudiantes y demostrar el vínculo entre la informática teórica y las aplicaciones del mundo real, es necesario enfatizar en la construcción de modelos. Los modelos computacionales pueden proporcionar explicaciones que soporten múltiples perspectivas, que van desde el nivel conceptual, al nivel lógico y finalmente a nivel físico; todos estos niveles puede saciar la sed de cualquiera por conocimiento.

2.4 Pensamiento constructivo

La meta de la teoría es lograr la práctica en la realidad. Así que resolver algunos problemas –como la realización aritmética, la organización de datos, la representación gráfica de la información, jugar ajedrez con otra persona– es una habilidad importante. El pensamiento constructivo puede ayudar a resolver estos problemas mediante algoritmos y programas; muchos programas interesantes y útiles requieren un mayor esfuerzo, y desarrollar algunos ejercicios algorítmicos en matemáticas discretas.

Los computadores trabajan con algoritmos. Un algoritmo es un proceso mecánico paso a paso sin ambigüedades que no requiere de agudeza o ingenio para llevarlo a cabo. Tal es el caso de alguien a quien le gustan las recetas culinarias, la elaboración de una buena receta puede ser una tarea difícil y creativa, pero si sigue los pasos de que describe la receta se convertirá en una tarea sencilla y rutinaria. Realizar estas operaciones requiere de un pensamiento innovador.

Definición 4. Pensamiento constructivo: Es cualquier procedimiento computacional bien definido que tiene algún valor, o conjunto de valores, como entrada y produce un valor, o conjunto de valores, como salida. Informalmente:

Pensamiento constructivo:= (Q, I, Ω, F), donde Q es un conjunto de estados computacionales, tanto I como Ω son subconjuntos de Q. Entre los cuales, I representa el conjunto de entradas computacionales, y Ω el conjunto de salidas. Al utilizar la función F es posible definir un orden que se organiza de la siguiente manera:

x_0, x_1, x_2,… x_k, donde $x_k = F(x_{k-1})$. El orden "x_0, x_1, x_2,… x_k" los pasos computacionales construidos comenzando con el paso x_0 y terminando con x_k.

Como portador de pensamiento constructivo, un algoritmo o programa es una secuencia de pasos computacionales que transforman las entradas en salidas. También se puede visualizar un algoritmo como una herramienta para resolver un problema computacional bien especificado. El algoritmo describe un procedimiento computacional específico para lograr que las entradas/salidas se relacionen.

3. Contenido de la matemática discreta

La matemática discreta tiene aplicaciones en todos los campos de las ciencias computacionales, se utiliza ampliamente en las telecomunicaciones y el procesamiento de la información. En matemática discreta se trabaja con objetos discretos, tales como números enteros, proposiciones, conjuntos, relaciones y funciones, y se aplican conceptos asociados con ellos, y propiedades y relaciones entre ellos [7]. Además, incluye conjuntos, funciones y relaciones, álgebra matricial, probabilidad finita y combinatoria, teoría de grafos, diferencias finitas y las relaciones de recurrencia, lógica, inducción matemática, y pensamiento algorítmico [8, 9]. Debido a esta diversidad de temas, a veces se prefiere estudiarlos todos como contenido de la matemática discreta, pero, la matemática discreta tiene un conjunto mínimo de temas como condición necesaria para comprender el pensamiento computacional.

A finales de 1990 se conformó un Join ACM/IEEE Task Force para revisar los planes de estudios de pregrado en informática. Como resultado reportaron seis tópicos como la base de conocimientos para estructuras discretas: 1) funciones, relaciones y conjuntos, 2) lógica básica, 3) técnicas de pruebas, 4) conceptos básicos de conteo, 5) grafos y árboles, y 6) probabilidad discreta. Llegaron a la conclusión de que los procesos formativos en matemática discreta deben incluir ejemplos y aplicaciones desde las ciencias computacionales [1], ya que las aplicaciones pueden mejorar la comprensión de estos tópicos.

3.1 Lógica matemática

La lógica es un lenguaje para razonar sobre algunas aserciones. Se trata de un conjunto de reglas que se pueden utilizar cuando se hace un razonamiento lógico. El razonamiento humano ha sido observado durante siglos desde los tiempos de los griegos, y los patrones que aparecen en el razonamiento han sido extraídos y abstraídos. El fundamento de la lógica fue establecido por el matemático inglés G. Boole a mediados del siglo XIX. La lógica matemática se interesa por lo verdadero o falso de las declaraciones, y cómo esa verdad/falsedad de una declaración se puede determinar a partir de otras declaraciones. Nosotros usamos símbolos para representar declaraciones arbitrarias para que los resultados se puedan utilizar en muchas situaciones similares pero diferentes, por lo que la lógica puede promover la claridad de pensamiento y eliminar ambigüedades y errores.

Existen varios tipos de lógica: lógica de oraciones —lógica proposicional—, lógica de objetos —lógica de predicados—, lógica de incertidumbres, lógica difusa, lógica modal, lógica temporal, entre otras. Pero para un curso de matemática discreta, sólo concierne la lógica proposicional y de predicados, que son fundamentales para comprender otras lógicas.

Ejemplo 1. Lógica para efectuar búsquedas booleanas: En lógica proposicional hay varios conectores (¬, ∧, ∨) que se utilizan ampliamente en la búsqueda de información en la Internet. Por ejemplo, el motor de búsqueda de Google soporta la técnica de búsquedas booleanas, que usualmente pueden ayudar a encontrar páginas web sobre un tema en particular. En Google, "+" se utiliza como el conector lógicos "∧", "–" como el conector lógico "¬", y "OR" como "∨". Si se escribe la frase de entrada "pensamiento computacional" en Google, se van a buscar todas las páginas web acerca de pensamiento computacional; si se escribe "pensamiento – computacional" se buscan todas las páginas que incluyan "pensamiento", pero no "computacional", por lo que la búsqueda se reduce un poco.

3.2 Teoría de conjuntos

El concepto de conjunto es fundamental para las ciencias computacionales. Por ejemplo, la relación entre dos objetos se representan como un conjunto de pares ordenados de objetos, el concepto de par ordenado se define utilizando conjuntos; los números naturales, que son la base de otros números, también se definen a través de conjuntos; el concepto de función, siendo un tipo especial de relación, se basa en conjuntos, y los grafos y los dígrafos consistente en líneas y puntos se describen como un par ordenado de conjuntos.

La relación es un conjunto especial que consiste de dos tuplas, que son una abstracción de las relaciones que se observan en la vida cotidiana, como las que existen entre padre e hijo, dirección y número de teléfono, etc. En la teoría de conjuntos, la atención se centra en las propiedades de esas relaciones, tales como la reflexividad, irreflexivilidad, simetría, antisimetría y transitividad.

Una función es algo que asocia cada elemento de un conjunto con un elemento de otro conjunto. Se utiliza bastante a menudo incluso en contextos no técnicos. Por ejemplo, un número de seguro social identifica a una única persona; la tasa de impuesto sobre la renta varía en función de los ingresos, y así sucesivamente. En resumen, una función es absolutamente como una relación, pero un elemento en una función no se relaciona con muchos elementos.

Ejemplo 2. Aplicación de una relación transitiva en telefonía móvil: La red de telefonía móvil cuenta con centros de datos en varias ciudades de cada país, que están unidas con cables de una vía. Es posible modelar esta situación mediante relaciones. Sea R la relación, $(a, b) \in R$ si hay un cable óptico desde el centro de datos a a otro en b. ¿Cómo se puede garantizar que existe algún tipo de enlace de una ciudad a otra compuesto por uno o más cables? Aunque R no se puede utilizar directamente para responder a esto, sin embargo, es posible encontrar todos los pares de centros de datos que tienen un enlace construyendo una relación transitiva de R.

3.3 Álgebra abstracta

El álgebra abstracta es una aplicación típica de pensamiento abstracto. Los matemáticos trabajan y se deleitan con números bizarros, tales como i —raíz cuadrada de -1—, al que parece asociársele una mezcla de misticismo irracional. No obstante, las operaciones básicas aritmética —suma, resta, multiplicación, división, etc.— se aplican extraordinariamente bien a muchas situaciones disparatadas de la vida real, tales como el balance de una chequera, cuidar la puntuación en un juego de cartas, o medir los ingredientes para una receta. En todas estas situaciones, se utiliza el operador suma y multiplicación para modelar perfectamente la situación.

En álgebra abstracta, frecuentemente las estructuras se mueven rápidamente de un punto de vista a otro más general. A menudo se mueven de un lado a otro entre los conceptos abstractos generales y ejemplos concretos específicos; este movimiento entre estructuras abstractas e instancias concretas de estas estructuras es el corazón del álgebra abstracta. La abstracción a menudo se siente como un procedimiento sin intuición, guiada por la lógica desnuda, en la que la prueba se convierte en crucial. Pero después de estudiar cuidadosamente algunos ejemplos es posible trasladarse a un caso general más claro. Hay dos razones para ello: en primer lugar, hacer lo contrario tomaría mucho tiempo, en segundo lugar, el punto de vista abstracto es mucho más fácil que el concreto.

Ejemplo 3. El software es un álgebra abstracta: El software es una estructura algebraica SW = (Lenguaje, Strcat, Strcomp, Strcpy,…), donde Lenguaje es un conjunto de cadenas que han sido procesado por el software, Strcat, Strcomp y Strcpy son operadores de cadena. Por ejemplo, Strcomp puede comparar dos cadenas si son iguales. En la naturaleza, el software es una máquina que puede procesar y traducir cadenas; además, es una estructura algebraica cuyo objeto de operación es la cadena.

3.4 Teoría de grafos

Un grafo G consiste de dos conjuntos disjuntos V —vértices, nodos— y E —bordes, aristas—, y una relación de incidencia que asocia un par de nodos de cada arista. Sus aplicaciones son múltiples en diversas disciplinas como en Biología —árboles filogenéticos—, Ciencias Computacionales —programa de control de punto muerto, modelización de Internet—, Económica —redes sociales—, Ingeniería —redes de computadores—, y otras ramas como los deportes —modelado de torneos.

Utilizando la teoría de grafos es posible explorar algunas de sus numerosas aplicaciones, especialmente en las ciencias computacionales, como el análisis del camino crítico, problemas de coloración de gráficas, árboles de expansión mínima, y técnicas de empaquetamiento binario. Hay dos objetivos importantes para la formación en teoría de grafos: formar a los estudiantes para escribir pruebas completas y concisas, y comprender la aplicación de la teoría de grafos en informática e ingeniería de software.

Ejemplo 4. Aplicación CP-net: Extraer la información sobre las preferencias de los usuarios generalmente es un proceso arduo, y los analistas de decisiones humanas han desarrollado técnicas sofisticadas para ayudar a elicitar esta información; CP-net es un modelo gráfico importante y útil para representar esas preferencias [10].

Definición 5. Una CP-net sobre las variables $V = \{X_1, X_2,..., X_n\}$ es un grafo dirigido G sobre $X_1, X_2,..., X_n$ cuyos nodos son anotados en tablas de preferencia condicionada $CPT(X_i)$ para cada $X_i \in V$. Cada tabla de preferencia condicional $CPT(X_i)$ asocia un orden total $>$ con cada instancias u de los padres X_i $Pa(X_i) = U$.

La Fig. 1 ilustra un CP-net que representa las preferencias de alguien para el vestido de noche. Se compone de tres variables J, P y S, para chaqueta, pantalones y camisa, respectivamente. Esta persona incondicionalmente prefiere el negro al blanco como color para camisa y pantalones, mientras que su preferencia entre camisas rojas y blancas está condicionada por la combinación de chaqueta y pantalón: si tienen el mismo color, entonces prefiere una camisa roja. De lo contrario, si la chaqueta y los pantalones son de colores diferentes, entonces prefiere una camisa blanca.

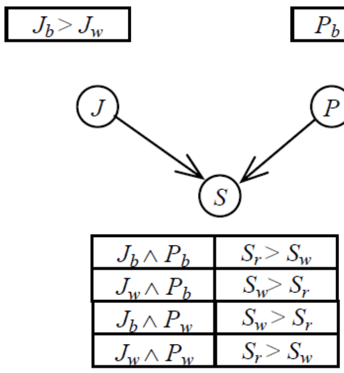

Fig. 1. CP-net para "traje de gala": chaqueta, pantalón y camisa

Como se indicó anteriormente, el pensamiento computacional tiene algunas clases de pensamientos y la matemática discreta estudia los objetos discretos y sus relaciones. Con el fin de comprender el contenido de la matemática discreta, y captar la idea de pensamiento computacional en matemáticas discretas, a continuación se describe la relación entre pensamiento computacional y matemáticas discretas.

3.5 Abstracción mediante estructuras algebraicas y grafos

Muchas de las unidades de conocimiento de las matemáticas discretas pueden incluirse en pensamiento abstracto. La propuesta formal de la lógica matemática es una abstracción que pueda describir algunas declaraciones sobre las ciencias computacionales y el mundo real. Tales como "protocolo TCP es un protocolo de Internet", nos dice que en la tecnología de Internet, TCP es un protocolo importante; además, conjunto, relación, función, puede verse como una abstracción de algunos objetos discretos que son estudiados por el científico computacional; el UML de la ingeniería del software, es una serie diagramas —de clases, de objetos, de estado, de casos de uso,…— que se abstraen de componentes de software.

3.6 Pensamiento lógico mediante lógica computacional

El pensamiento lógico discurre a través de todos los procesos de formación en matemática discreta, entre los cuales, la lógica matemática sienta una creación de la razón. En la teoría de conjuntos, la estructura algebraica y la teoría de grafos, muchas pruebas de teoremas se pueden ver como pensamiento lógico. En la unidad lógica, se esboza la estructura básica y se dan ejemplos de cada técnica de la prueba. Mediante el pensamiento lógico, es posible discutir qué tipo de prueba es la mejor para un problema dado. Además, es posible relacionar las ideas de inducción matemática para la recursividad y las estructuras definidas recursivamente.

3.7 Pensamiento modelado mediante teoría y relación de conjuntos

La matemática discreta tiene aplicaciones en casi todas las áreas imaginables de las ciencias computacionales. Modelar con matemáticas discretas es una habilidad extremadamente importante para resolver problemas, que da la capacidad para desarrollar algunos programas para resolver los problemas computacionales mediante pensamiento constructivo. Las herramientas para modelar son: proposición, conjunto, permutación, relación, grafo, árbol, máquina de estados finitos, operador, y estructura algebraica —como grupo, anillo, álgebra de Boole.

3.8 Pensamiento constructivo mediante algoritmo y prueba

En un curso de matemáticas discretas, existen muchas unidades de conocimiento sobre prueba de teoremas y construcción de algoritmos. Al asumir la naturaleza de la prueba, las pruebas directas e indirectas, la prueba por contradicción, los contraejemplos, las pruebas de existencia y constructivas, es posible decir que la construcción de pensamiento es un enfoque computacional importante. Por otra parte, la inducción matemática —débil, fuerte, estructural—, el principio del buen orden, el estándar de búsqueda y los algoritmos de ordenamiento, los argumentos de correctitud algorítmica, las definiciones recursivas, los algoritmos iterativos y recursivos son algunos enfoques constructivos concretos.

REFERENCIAS

[1] ACM/IEEE. "Task Force Report on Computing Curricula 2001". Computer Science, Vol. 2001. http://www.acm.org/education/curricula.html, Sep. 2010.
[2] J. M. Wing. "Computational thinking". Communications of the ACM, Vol. 49, No. 3, pp. 33-35. Mar. 2006.
[3] M. Guzdial. "Paving the way for computational thinking". Communications of the ACM, Vol. 51, No. 8, pp. 25-27, Aug. 2008.
[4] ACM/IEEE. "Computing Curricula 2005". http://www.acm.org/education/education/curricula-recommendations, Oct. 2010.
[5] J. S. Warford. "An experience teaching formal methods in discrete mathematics". ACM SIGCSE Bulletin, Vol. 27, No. 3, pp. 60-64. Sept. 1995.

[6] M. Shaw. “Software Engineering for the 21st Century: A Basis for Rethinking the Curriculum”. CMU-ISRI-05-108. School of Computer Science, Carnegie Mellon University, Pittsburgh PA. 2005.
[7] P. Dourish, G. R. Hayes, L. Irani, C. P. Lee, S. Lindtner, B. Nardi, D. J. Patterson & B. Tomlinson. “Informatics at UC Irvine”. Abstracts on Human Factors in Computing Systems. CHI '08, Florence. Italy, pp. 3651-3656. Apr. 2008.
[8] N. Crisler & P. Fisher. “Discrete mathematics through Applications”. Ney York: W. H. Freeman and Company, 544 p. 1999.
[9] B. Marion. “Final Oral Report on the SIGCSE Committee on the Implementation of a Discrete Mathematics Course”. In SIGCSE Technical Symposium on Computer Science Education, Houston, Texas, pp. 268-9. Mar. 2006.
[10] C. Boutilier & R. Brafman. “CP-nets: A tool for representing and reasoning about conditional ceteris paribus preference statements”. Journal of Artificial Intelligence Research, Vol. 21, pp. 135–191. 2004. Ω