

## A TOMADA DE DECISÃO EM RECURSOS HUMANOS COM DADOS REPLICADOS E INCONSISTENTES: UMA APLICAÇÃO DA TEORIA DOS CONJUNTOS APROXIMATIVOS

**Ayrton Benedito Gaia do Couto\***

BNDES

Rio de Janeiro – RJ, Brasil

[acouto@bndes.gov.br](mailto:acouto@bndes.gov.br)

**Luiz Flavio Autran Monteiro Gomes**

Ibmec/RJ

Rio de Janeiro – RJ, Brasil

[autran@ibmecrj.br](mailto:autran@ibmecrj.br)

\* *Corresponding author* / autor para quem as correspondências devem ser encaminhadas

*Recebido em 12/2008; aceito em 03/2010*

*Received December 2008; accepted March 2010*

### Resumo

Este estudo trata da tomada de decisão com dados replicados e inconsistentes, relativos ao universo de Recursos Humanos, em uma instituição financeira nacional. A replicação ocorre por questões técnicas e/ou econômicas, e visa o atendimento de necessidades corporativas e departamentais dessa instituição. Como metodologia de pesquisa, utilizou-se a observação direta das inconsistências e a simulação com base em dados reais que refletissem a replicação com inconsistências. Os autores apresentam a aplicação de um método multicritério para tornar racional o processo de tomada de decisão, e que se transformou em elemento de motivação deste estudo. O método utilizado foi a Teoria dos Conjuntos Aproximativos (TCA). Um algoritmo foi desenvolvido para indicar as relações de equivalência, aproximações inferior e superior, região de fronteira, medida da exatidão, reduções e núcleo e, posteriormente, implementado em um *software*.

**Palavras-chave:** tomada de decisão; inconsistência; teoria dos conjuntos aproximativos.

### Abstract

This study deals about decision-making with replicated and inconsistent data, relating to the universe of Human Resources, within a domestic financial institution. Replication occurs because of technical and/or economic questions, and seeks to meet corporate and departmental requirements of such an institution. As research methodology, direct observation of such inconsistencies was used as well as a simulation based on actual data which would reflect replication with inconsistencies. The authors introduce an application of a multi-criteria method to render the decision-making process rational, and was transformed into an element that stimulated this study. The method used was the Rough Set Theory (RST). An algorithm was developed to indicate the equivalence relations, lower and higher approximations, borderline region, accuracy measure, reducts and core and was subsequently implemented into a software.

**Keywords:** decision-making; inconsistency; rough set theory.

## 1. Introdução

O presente estudo tem como ponto de partida, a situação encontrada em pesquisas (ex. consulta de quantitativo de pessoal) a banco de dados replicados e inconsistentes de Recursos Humanos (RH), de uma instituição financeira nacional. Dado que, as inconsistências ocorriam esporadicamente, utilizou-se da simulação em planilha eletrônica para ilustrar as situações de replicação comumente encontradas.

Por possuir base matemática para o tratamento da imprecisão de dados e, por não haver necessidade de informação preliminar sobre os dados em questão, os autores propõem mostrar a utilização da Teoria dos Conjuntos Aproximativos (*Rough Set Theory*) ou TCA como método multicritério de apoio à tomada de decisão, ante a situação de replicação de dados com inconsistência. De forma resumida, são apresentados os fundamentos principais, exemplos de aplicações e restrições da TCA e, por último, a sua aplicação a um problema prático em um ambiente de RH. Para tornar racional essa tomada de decisão, desenvolveu-se um algoritmo (e posteriormente implementado em software) para indicar as aproximações inferior e superior e a região de fronteira, calcular a medida de exatidão (sistema de informação), indicar as fontes de dados (in)dispensáveis, sugerir as fontes essenciais (“reduções”) e, possivelmente, a fonte de dados principal (“núcleo”), como opções ao conjunto de dados replicados.

## 2. Definição do Problema

Ao se elaborarem pesquisas (consultas) a bancos de dados (sistemas de armazenamento de dados em computador), uma situação particularmente encontrada é a replicação de dados, isto é, quando múltiplas cópias de um mesmo conjunto de dados são disponibilizadas para consulta, por exemplo, com o objetivo de descentralizar o seu acesso (Date, 1984; Son, 1988). Mas, se a atualização dessas cópias não for efetuada sob algum controle (redundância controlada), haverá ocasiões em que as cópias não serão concordantes, isto é, quando no mínimo uma cópia não tiver sido atualizada integralmente. Nesse caso, o banco de dados é dito estar “inconsistente” (Codd, 1970; Date, 1984; Son, 1988).

É com base na situação de replicação concomitante com a inconsistência de dados que este estudo foi desenvolvido. Em uma determinada instituição financeira nacional, especificamente com relação a dados de RH, constatou-se que, esporadicamente, havia discordâncias (inconsistências) nos resultados de uma mesma pesquisa (ex. quantitativo de pessoal), proveniente de dados replicados. Essa replicação deve-se a questões técnicas, sendo a principal o atendimento a demandas corporativas e departamentais, provenientes de plataformas (ambientes) tecnológicas distintas.

Com o objetivo de ilustrar a aplicação de um método multicritério de apoio à tomada de decisão – neste contexto, “como escolher uma fonte de dados ante a constatação de que os dados estão replicados e inconsistentes?”, os autores desse estudo optaram pela TCA, pelas seguintes razões: (a) existência de dado impreciso (inconsistente) e indiscernível; (b) possibilidade de processar os dados sob a perspectiva (matemática) de um “sistema de informação”; (c) possibilidade de tratar a imprecisão por meio de aproximações “inferior” e “superior”; (d) possibilidade de obter as fontes essenciais (“reduções”) e a fonte principal (“núcleo”) de dados como opções ao conjunto de dados original; (e) inexistência de qualquer informação preliminar sobre os dados em questão (ex. frequência de ocorrência das replicações com inconsistência). Outras teorias poderiam ser utilizadas – ex. *Fuzzy Set Theory* ou Teoria dos Conjuntos Nebulosos, proposta por Lotfi Asker Zadeh, em 1965, como uma extensão da

lógica convencional (booleana) para introduzir o conceito de verdade não absoluta (Gomes, Gomes & Almeida, 2006). Assim, A TCA e a *Fuzzy Set Theory* são abordagens independentes para o tratamento do conhecimento imperfeito (incompleto) e impreciso (vago, indeterminado) (Pawlak *et al.*, 1995). Ademais, um algoritmo foi desenvolvido e implementado em um programa de computador (pelo primeiro autor desse estudo), em linguagem Borland® Delphi/Pascal, que tornou racional a tomada de decisão pela indicação das possíveis “reduções” de dados e do “núcleo”, caso exista, além das relações de equivalência, aproximações inferior e superior, região de fronteira e cálculo da medida da exatidão (sistema de informação).

### 3. A Teoria dos Conjuntos Aproximativos

A TCA, proposta pelo matemático polonês, Zdzislaw Pawlak, em 1982, destina-se ao tratamento da imprecisão de dados, por meio de “aproximações” (inferior e superior) de um conjunto de dados (Pawlak, 1991). Tem como ponto inicial, a relação de “indiscernibilidade” (*indiscernibility*), isto é, aquela que identifica os objetos com a mesma propriedade. Objetos de interesse que possuem as mesmas propriedades são “indiscerníveis” e, conseqüentemente, são tratados como idênticos ou similares (“grânulos”). Esses grânulos são conhecidos como “conjuntos elementares” (*elementary sets*) e constituem os “conceitos” (*concepts*) de conhecimento sobre algo de interesse (Pawlak, 2000). A “granularidade” (*granularity*) na representação da informação, segundo Pawlak & Slowinski (1994), pode ser origem de inconsistências nas decisões, dada à ambigüidade para explicar e prescrever com base em informação inconsistente. Por sua vez, Grzymala-Busse (1988) cita que os fenômenos do mundo real, quando são representados por sistemas de informação, incluem inconsistências, que têm origem, por exemplo, nas ações distintas de diferentes especialistas para o mesmo objeto em questão.

Em Pawlak (1991) encontra-se um exemplo que ilustra alguns conceitos. Dado o conjunto  $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$  de brinquedos, classificado de acordo com a cor (vermelha, azul, amarela), forma (quadrada, redonda, triangular) e tamanho (pequeno, grande), ter-se-ia:

	<u>cor</u>		<u>forma</u>		<u>tamanho</u>
x1, x3, x7	vermelha,	x1, x5	redonda,	x2, x7, x8	grande,
x2, x4	azul,	x2, x6	quadrada,	x1, x3, x4, x5, x6	pequeno.
x5, x6, x8	amarela,	x3, x4, x7, x8	triangular,		

e foram definidas três relações de equivalência,  $R_1$ ,  $R_2$  e  $R_3$ , para cor, forma e tamanho, respectivamente, tendo as seguintes classes de equivalência:

$$U/R_1 = \{\{x_1, x_3, x_7\}, \{x_2, x_4\}, \{x_5, x_6, x_8\}\} \quad (1)$$

$$U/R_2 = \{\{x_1, x_5\}, \{x_2, x_6\}, \{x_3, x_4, x_7, x_8\}\} \quad (2)$$

$$U/R_3 = \{\{x_2, x_7, x_8\}, \{x_1, x_3, x_4, x_5, x_6\}\} \quad (3)$$

as quais são conceitos (categorias) elementares na base de conhecimento  $K = (U, \{R_1, R_2, R_3\})$ .

Assim, de acordo com Pawlak (1991), o conhecimento apóia-se na habilidade em classificar objetos. Nesse caso, um objeto pode ser algo real ou abstrato. Assim, dado um conjunto finito  $U \neq \emptyset$  de objetos (o universo), um subconjunto  $X \subseteq U$  é referenciado como um “conceito” ou “categoria” em  $U$ , podendo ser qualquer família de conceitos em  $U$  denominada “conhecimento abstrato” ou, de forma abreviada, “conhecimento” sobre  $U$ . Ademais, não é usual tratar com uma única classificação, mas com uma família de classificações básicas (ex. cor, temperatura etc.) sobre  $U$ . Neste ponto, “relações de equivalência” e

“classificações” têm o mesmo significado, indistintamente. De acordo com Grzymala-Busse (1988) e Ziarko (1993), a “relação de equivalência” também é conhecida como “relação de indiscernibilidade”; e “classes de equivalência” são conhecidas como “conjuntos elementares”. Assim, se  $R$  é uma relação de equivalência sobre  $U$ , então  $U/R$  significa a família de todas as classes de equivalência de  $R$  (Pawlak, 1991). Ainda de acordo com Pawlak (1991), se  $P \subseteq R$  e  $P \neq \emptyset$ , então  $\cap P$  (interseção de todas as relações de equivalência pertencentes a  $P$ ) é também uma relação de equivalência, e é indicado por  $IND(P)$ , e é conhecido como “relação de indiscernibilidade” sobre  $P$ . Assim,  $U/IND(P)$  significa a família de todas as classes de equivalência da relação de equivalência  $IND(P)$ , e representa o conhecimento associado com a família de relações de equivalência de  $P$ . No Anexo I, encontra-se um exemplo de aplicação dos conceitos anteriormente definidos.

De acordo com a TCA, um conjunto de conceitos elementares pode originar um conceito “preciso” (*crisp* ou *precise*), ou “aproximativo”, ou “impreciso” (*rough* ou *imprecise*). Assim, os conceitos aproximativos não podem ser expressos em termos de conceitos elementares, mas expressos com a utilização de aproximações “inferior” (*lower*) e “superior” (*upper*) de um conceito. A aproximação “inferior” de um conceito é a união de todos os conceitos elementares que estão inclusos naquele conceito; enquanto que a aproximação “superior” é a união de todos os conceitos elementares que têm alguma intersecção (não vazia) com o conceito em questão. A diferença entre essas aproximações é conhecida como “região de fronteira” (*boundary region*) (Pawlak, 2000). Assim, segundo Pawlak (2000), ficaria fácil identificar se um conceito é “aproximativo”: se a região de fronteira não for vazia ou se as aproximações inferior e superior forem diferentes. Uma outra definição importante no contexto dessa Teoria, refere-se a “sistema de informação”. Um sistema de informação (*information system*) ou sistema de representação do conhecimento (*knowledge representation system*) ou base de dados (*database*) é uma tabela finita, em que as linhas são identificadas pelos objetos e as colunas, pelos atributos. Assim, um sistema do conhecimento pode ser visto como uma coleção de objetos descritos pelos valores dos atributos (Pawlak, 1991; Pawlak & Slowinski, 1994; Pawlak, 2000). Segundo Pawlak & Slowinski (1994), por sistema de informação entende-se como uma tupla  $S = (U, Q, V, f)$ , onde  $U$  é um conjunto finito de objetos,  $Q$  é um conjunto finito de atributos,  $V = \cup_{q \in Q} V_q$ , onde  $V_q$  é o domínio do atributo  $q$  e,  $f: U \times Q \rightarrow V$  é uma função total tal que,  $f(x, q) \in V_q$  para cada  $q \in Q$ ,  $x \in U$ , conhecida como “função de informação”. Ainda de acordo com Pawlak & Slowinski (1994), dado um sistema de informação,  $S = (U, Q, V, f)$ , e  $P \subseteq Q$ , e  $x, y \in U$ , é dito que  $x$  e  $y$  são “indiscerníveis” pelo conjunto de atributos  $P$  em  $S$ , se  $f(x, q) = f(y, q)$  para todo  $q \in P$ . Portanto, todo  $P \subseteq Q$  gera uma relação binária em  $U$ , conhecida como “relação de indiscernibilidade”, denotada por  $IND(P)$ . Dado que,  $P \subseteq Q$  e  $Y \subseteq U$ , a aproximação inferior ( $\underline{P}Y$ ) e a aproximação superior ( $\overline{P}Y$ ) são definidas como

$$\underline{P}Y = \cup \{X \in U / P: X \subseteq Y\} \quad \text{e} \quad \overline{P}Y = \cup \{X \in U / P: X \cap Y \neq \emptyset\} \quad (4)$$

Assim,  $Y$  é um conjunto “aproximativo” (*rough*) com relação a  $P$ , se e somente se,  $\underline{P}Y \neq \overline{P}Y$  (Pawlak, 1991). A “região de fronteira” (*borderline region*) de um conjunto  $Y$  é definida como

$$Bn_P(Y) = \overline{P}Y - \underline{P}Y \quad (5)$$

Para todo conjunto  $Y \subseteq U$ , pode-se associar a “precisão” (*accuracy*) de aproximação do conjunto  $Y$  por  $P$ , em  $S$  (Pawlak & Slowinski, 1994):

$$\alpha_P(Y) = \text{card}(\underline{P}Y) / \text{card}(\overline{P}Y) \quad (6)$$

A precisão mede o “quanto está completo” (*completeness*) o conhecimento sobre o conjunto  $Y$ . A falta de exatidão de um conjunto é devida à existência da região de fronteira. Quanto maior a região de fronteira, menor é a precisão (Pawlak, 1991). Retornando-se ao exemplo das lojas (Quadro 3, no Anexo I), pode-se então calcular a “precisão” para o conjunto lucro:

$$\text{de: } \underline{P} Y = \{1, 6\} \text{ (aproximação inferior)} \quad (7)$$

$$\overline{P} Y = \{1, 2, 3, 6\} \text{ (aproximação superior)} \quad (8)$$

$$\text{então: } \alpha_p(Y) = \text{card}(\underline{P} Y) / \text{card}(\overline{P} Y) = 2 / 4 = 0,5 \quad (9)$$

#### 4. Redução e núcleo de um Sistema do Conhecimento

Dois importantes conceitos: a “redução” (*reduct*) e o “núcleo” (*core*) de um sistema do conhecimento. A redução é a sua parte essencial, isto é, o conjunto de atributos que fornece a mesma qualidade de classificação que o conjunto original de atributos; é o conjunto mínimo de atributos de condição que permite tomar as mesmas decisões caso houvesse todos os atributos de condição (Pawlak, 1991; Pawlak & Slowinski, 1994; Pawlak, 2000). Segundo Ziarko (1993), a redução de atributos é uma das mais úteis idéias da TCA. O núcleo pode ser interpretado como a parte mais importante desse conhecimento, ou seja, a coleção dos atributos mais importantes de um sistema do conhecimento (Pawlak, 1991; Pawlak & Slowinski, 1994; Pawlak, 2000). Considere-se que  $\mathbf{R}$  seja uma família de relações e  $R \in \mathbf{R}$ . Diz-se que  $R$  é “dispensável” em  $\mathbf{R}$  se  $\text{IND}(\mathbf{R}) = \text{IND}(\mathbf{R} - \{R\})$ ; de outra forma,  $R$  é “indispensável” em  $\mathbf{R}$ . A família  $\mathbf{R}$  é “independente” se cada  $R \in \mathbf{R}$  é indispensável em  $\mathbf{R}$ ; caso contrário,  $\mathbf{R}$  é “dependente” (Pawlak, 1991).

Pawlak (1991) define as seguintes proposições:

- Se  $\mathbf{R}$  é independente e  $\mathbf{P} \subseteq \mathbf{R}$ , então  $\mathbf{P}$  é também independente.
- $\text{CORE}(\mathbf{P}) = \bigcap \text{RED}(\mathbf{P})$ , onde  $\text{RED}(\mathbf{P})$  é a família de todas as “reduções” de  $\mathbf{P}$ .

Em Pawlak (1991) encontra-se um exemplo que ilustra como obter as reduções e o núcleo de um sistema do conhecimento: dada a família  $\mathbf{R} = \{P, Q, R\}$  de três relações de equivalência  $P, Q$  e  $R$ , com as seguintes classes de equivalência:

$$U/P = \{\{x_1, x_4, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_6, x_7\}\} \quad (10)$$

$$U/Q = \{\{x_1, x_3, x_5\}, \{x_6\}, \{x_2, x_4, x_7, x_8\}\} \quad (11)$$

$$U/R = \{\{x_1, x_5\}, \{x_6\}, \{x_2, x_7, x_8\}, \{x_3, x_4\}\} \quad (12)$$

Assim, a relação  $\text{IND}(\mathbf{R})$  tem as seguintes classes de equivalência:

$$U/\text{IND}(\mathbf{R}) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} \quad (13)$$

A relação  $P$  é indispensável em  $\mathbf{R}$ , dado que:

$$U/\text{IND}(\mathbf{R} - \{P\}) = \{\{x_1, x_5\}, \{x_2, x_7, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}\} \neq U/\text{IND}(\mathbf{R}) \quad (14)$$

Para a relação  $Q$ , tem-se que:

$$U/\text{IND}(\mathbf{R} - \{Q\}) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} = U/\text{IND}(\mathbf{R}), \quad (15)$$

assim, a relação  $Q$  é dispensável em  $\mathbf{R}$ .

Similarmente, para a relação R, obtém-se:

$$U/IND(\mathbf{R} - \{R\}) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} = U/IND(\mathbf{R}), \quad (16)$$

a relação R é também dispensável em R.

Isto significa que a classificação definida pelas três relações de equivalência P, Q e R é a mesma que a classificação definida pela relação P e Q ou P e R. Com o intuito de encontrar as reduções da família  $\mathbf{R} = \{P, Q, R\}$ , verifica-se se cada par de relação “P,Q” e “P,R” são independentes ou não. Dado que  $U/IND(\{P,Q\}) \neq U/IND(Q)$  e  $U/IND(\{P,Q\}) \neq U/IND(P)$ , as relações P e Q são independentes e, conseqüentemente,  $\{P,Q\}$  é uma redução de R. Procedimento semelhante é utilizado para encontrar a redução formada pela relação  $\{P,R\}$ . Assim, há duas reduções na família R,  $\{P,Q\}$  e  $\{P,R\}$ , e a interseção dessas reduções ( $\{P,Q\} \cap \{P,R\}$ ) é o núcleo  $\{P\}$  (Pawlak, 1991).

Tendo como referência o exemplo anterior (“família R”), um algoritmo foi desenvolvido pelo primeiro autor desse estudo para obter as relações de equivalência (in)dispensáveis, as reduções e o núcleo:

1- Para cada relação de equivalência, coloca-se uma ordem (início em “1”), para cada classe de equivalência encontrada. Das expressões abaixo:

$$U/P = \{\{x_1, x_4, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_6, x_7\}\}$$

$$U/Q = \{\{x_1, x_3, x_5\}, \{x_6\}, \{x_2, x_4, x_7, x_8\}\}$$

$$U/R = \{\{x_1, x_5\}, \{x_6\}, \{x_2, x_7, x_8\}, \{x_3, x_4\}\}$$

obtém-se o Quadro 4:

**Quadro 4** – Relações de equivalência.

Fonte: Adaptado de Pawlak (1991).

	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>
U/P	1	2	3	1	1	4	4	2
U/Q	1	3	1	3	1	2	3	3
U/R	1	3	4	4	1	2	3	3

2- Obtém-se a relação principal R, da seguinte forma: com base no quadro anterior (Quadro 4) e, com início em “x<sub>1</sub>” (classe de ordem “1,1,1”, respectivamente U/P, U/Q e U/R), busca-se outra classe que possua essa mesma ordem (“x<sub>5</sub>”). Neste caso, encontra-se a classe  $\{x_1, x_5\}$ ; essa é a classe de ordem “1,1,1”. Repete-se o processo para as demais classes. Para a relação R, obtém-se a ordem mostrada no Quadro 5:

**Quadro 5** – Obtenção da relação principal R.

Fonte: Adaptado de Pawlak (1991).

	x <sub>1</sub>	x <sub>2</sub>	x <sub>3</sub>	x <sub>4</sub>	x <sub>5</sub>	x <sub>6</sub>	x <sub>7</sub>	x <sub>8</sub>
U/R	1	2	3	4	1	5	6	2

3- A relação principal é então obtida de acordo com a ordem apresentada no Quadro 5:

$$U/IND(\mathbf{R}) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} \quad (17)$$

4- Repete-se o processo para a obtenção das demais relações conforme mostram os Quadros 6, 7 e 8:

a) Obtenção de  $\{\mathbf{R} - \mathbf{P}\}$ . De:

**Quadro 6** – Obtenção da relação  $\{\mathbf{R} - \mathbf{P}\}$ .

Fonte: Adaptado de Pawlak (1991).

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
U/Q	1	3	1	3	1	2	3	3
U/R	1	3	4	4	1	2	3	3
U/R – P	1	2	3	4	1	5	2	2

$$\text{então: } U/IND(\mathbf{R} - \{\mathbf{P}\}) = \{\{x_1, x_5\}, \{x_2, x_7, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}\} \quad (18)$$

Essa relação é obtida da seguinte forma: com base no quadro anterior (Quadro 6) e, com início em “ $x_1$ ” (classe de ordem “1,1”, respectivamente U/Q e U/R), busca-se outra classe que possua essa mesma ordem (“ $x_5$ ”). Nesse caso, encontra-se a classe  $\{x_1, x_5\}$ ; essa é a classe de ordem “1”, relativa a U/R – P. Para a ordem seguinte (“3,3”, respectivamente U/Q e U/R), encontra-se a classe  $\{x_2, x_7, x_8\}$ ; essa é a classe de ordem “2” (relativa a U/R – P). Repete-se o processo para as demais classes.

b) Obtenção de  $\{\mathbf{R} - \mathbf{Q}\}$ . De:

**Quadro 7** – Obtenção da relação  $\{\mathbf{R} - \mathbf{Q}\}$ .

Fonte: Adaptado de Pawlak (1991).

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
U/P	1	2	3	1	1	4	4	2
U/R	1	3	4	4	1	2	3	3
U/R – Q	1	2	3	4	1	5	6	2

$$\text{então: } U/IND(\mathbf{R} - \{\mathbf{Q}\}) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} \quad (19)$$

Essa relação é obtida da seguinte forma: com base no quadro anterior (Quadro 7) e, com início em “ $x_1$ ” (classe de ordem “1,1”, respectivamente U/P e U/R), busca-se outra classe que possua essa mesma ordem (“ $x_5$ ”). Neste caso, encontra-se a classe  $\{x_1, x_5\}$ ; essa é a classe de ordem “1”, relativa a U/R – Q. Para a ordem seguinte (“2,3”, respectivamente U/P e U/R), encontra-se a classe  $\{x_2, x_8\}$ ; essa é a classe de ordem “2” (relativa a U/R – Q). Repete-se o processo para as demais classes.

c) Obtenção de  $\{\mathbf{R} - \mathbf{R}\}$ :

**Quadro 8** – Obtenção da relação  $\{\mathbf{R} - \mathbf{R}\}$ .

Fonte: Adaptado de Pawlak (1991).

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
U/P	1	2	3	1	1	4	4	2
U/Q	1	3	1	3	1	2	3	3
U/R – R	1	2	3	4	1	5	6	2

$$\text{Obtendo-se então: } U/IND(\mathbf{R} - \{\mathbf{R}\}) = \{\{x_1, x_5\}, \{x_2, x_8\}, \{x_3\}, \{x_4\}, \{x_6\}, \{x_7\}\} \quad (20)$$

Essa relação é obtida da seguinte forma: com base no quadro anterior (Quadro 8) e, com início em “ $x_1$ ” (classe de ordem “1,1”, respectivamente U/P e U/Q), busca-se outra classe que possua essa mesma ordem (“ $x_5$ ”). Neste caso, encontra-se a classe  $\{x_1, x_5\}$ ; essa é a classe de ordem “1”, relativa a U/R – R. Para a ordem seguinte (“2,3”, respectivamente U/P e U/Q), encontra-se a classe  $\{x_2, x_8\}$ ; essa é a classe de ordem “2” (relativa a U/R – R). Repete-se o processo para as demais classes.

5- Comparam-se as ordens obtidas, conforme mostrado no Quadro 9:

**Quadro 9** – Comparação das relações.

Fonte: Adaptado de Pawlak (1991).

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
U/R	1	2	3	4	1	5	6	2
U/R – P	1	2	3	4	1	5	2	2
U/R – Q	1	2	3	4	1	5	6	2
U/R – R	1	2	3	4	1	5	6	2

Como a ordem de  $\{R - P\}$  é diferente da ordem de R, P é “indispensável”. Como as ordens de  $\{R - Q\}$  e  $\{R - R\}$  são iguais à ordem de R, Q e R são “dispensáveis”.

6- Verificam-se as possíveis reduções entre “P,Q” e “P,R”, de acordo com os Quadros 10 e 11:

a) para  $\{P,Q\}$ :

**Quadro 10** – Verificação de redução para  $\{P,Q\}$ .

Fonte: Adaptado de Pawlak (1991).

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
U/P	1	2	3	1	1	4	4	2
U/Q	1	3	1	3	1	2	3	3
U/P,Q	1	2	3	4	1	5	6	2

Como a ordem de  $\{P,Q\}$  é diferente das ordens de P e de Q,  $\{P,Q\}$  é uma redução.

b) para  $\{P,R\}$ :

**Quadro 11** – Verificação de redução para  $\{P,R\}$ .

Fonte: Adaptado de Pawlak (1991).

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$
U/P	1	2	3	1	1	4	4	2
U/R	1	3	4	4	1	2	3	3
U/P,R	1	2	3	4	1	5	6	2

Como a ordem de  $\{P,R\}$  é diferente das ordens de P e de R,  $\{P,R\}$  é uma redução.

7- Pela interseção das reduções, verifica-se se há um núcleo:

$$\{P,Q\} \cap \{P,R\} = \{P\} \quad (21)$$

## 5. Exemplos de aplicação da TCA

Desde o início de sua criação em 1982, encontra-se na literatura várias aplicações da TCA, tais como: análise e simplificação de circuitos digitais (Pawlak, 1991); inteligência artificial (Pawlak, 1991; Pawlak *et al.*, 1995); análise de conflito no Oriente Médio (Pawlak, 1991; Pawlak & Slowinski, 1994); pesquisa de mercado (Ziarko, 1999); especificação de produtos (Shao *et al.*, 2006); recuperação de informações (Das-Gupta, 1988; Huang *et al.*, 2006; Xiaoyue & Rujiang, 2006); *data mining* (Flinkman *et al.*, 2000); tratamento da imprecisão em sistemas de informação (Gomes & Gomes, 2001); processamento de grandes bases de dados (Lin, 2008); aprendizagem indutiva (Ziarko & Wong, 1986); sistemas de robótica (Bit & Beaubouef, 2008); medicina (Tsumoto, 2000; Hassanien, Abdelhafez & Own, 2008); construção civil (Tam, Tong & Chan, 2006).

## 6. Restrições da TCA

Ziarko (1993; 1993a) aponta algumas restrições da TCA, quando aplicada a um conjunto de informações (“classificações”):

- 1- A TCA necessita que essas classificações estejam completas e corretas, ou seja, a TCA é sensível a pequenos erros de classificação, causados por problemas de dependência de atributos.
- 2- As conclusões derivadas desse conjunto são aplicáveis somente a esse conjunto, o que na prática, limita a generalização das conclusões para um conjunto maior de informações.

Como alternativa a essas restrições, Ziarko (1993; 1993a) propõe o uso de um modelo, VP (*variable precision*), para o reconhecimento da presença de dependência de dados em situações em que os mesmos seriam considerados independentes.

Nowicki (2008) também aponta restrições quando as classificações estão incompletas e propõe um modelo alternativo que combine arquiteturas *Neuro-Fuzzy* com a TCA.

Ademais, Greco, Matarazzo & Slowinski (2005) observam que o princípio da indiscernibilidade não é suficiente para cobrir toda a semântica de um conjunto de informações. Por exemplo: duas empresas,  $\alpha$  e  $\beta$ , são avaliadas pelo critério do risco de falência, dado pela “relação de débito” (total de débito/total de ativos). Se a empresa  $\alpha$  tem uma baixa relação de débito e a empresa  $\beta$  tem uma alta relação de débito, então, dentro da TCA,  $\alpha$  é diferente (discernível) de  $\beta$  no que diz respeito a essa relação. Entretanto, do ponto de vista da avaliação do risco de falência, a relação de débito da empresa  $\alpha$  não é simplesmente diferente da relação de débito da empresa  $\beta$ , mas a empresa  $\alpha$  está em melhor condição do que a empresa  $\beta$ . Agora, considere-se que outros atributos (rentabilidade, qualidade gerencial, posição de mercado, etc.) sejam iguais para ambas as empresas e, por algum motivo, um decisor tenha feito uma avaliação de que a empresa  $\alpha$  tem um risco de falência maior que o da empresa  $\beta$ . Nesse caso, pela TCA, as empresas  $\alpha$  e  $\beta$  são discerníveis e uma relação de débito menor está associada a um risco maior de falência enquanto que uma relação de débito maior está associada a um risco menor de falência. Isso é contraditório. Dentro do Apoio Multicritério à Decisão, o princípio da indiscernibilidade tem que ser substituído pelo “princípio da dominância”: se  $x$  domina  $y$ , isto é, se  $x$  é ao menos tão bom quanto  $y$  com relação a todos os critérios considerados, então  $x$  deveria pertencer a uma classe não pior do que a classe de  $y$ ; se não, há uma inconsistência entre  $x$  e  $y$  (Roy & Bouyssou, 1993). Aplicando-se o princípio da dominância na análise das empresas  $\alpha$  e  $\beta$ ,

constata-se uma inconsistência entre a relação de débito e o risco de falência, o que leva ao seguinte paradoxo: “menor relação de débito, risco maior de falência”. Assim, para que seja possível tratar com problemas de apoio multicritério à decisão (*multiple-criteria decision aiding*), deve-se estender a TCA com a substituição da relação de indiscernibilidade pela “relação de dominância”, utilizando-se de “modelos de preferência”, isto é, de regras de decisão do tipo “se ..., então ...” (“*if ..., then ...*”) (Greco, Matarazzo & Slowinski, 2005).

## 7. Aplicação prática na empresa

Observou-se que, em uma instituição financeira nacional, especificamente com relação a RH, na replicação de dados oriundos de um “banco de dados principal” (neste contexto, armazenado em um computador central), para outros bancos de dados em plataformas tecnológicas distintas, ocorriam esporadicamente inconsistências. Até porque essa replicação não era tempestiva, isto é, a replicação não acontecia imediatamente após a ocorrência de um evento (ex. designação de um empregado para a função de chefe de departamento). As inconsistências eram detectadas quando os resultados obtidos dos bancos de dados replicados eram comparados entre si. Ou seja, por algum motivo que foge ao escopo deste estudo, o processo de replicação não era perfeito. Um procedimento de conferência de dados foi então montado: a mesma pesquisa deveria ser realizada em dois ou mais bancos de dados. Caso houvesse alguma discordância, uma pesquisa mais ampla deveria ser realizada para dirimir as diferenças encontradas nos resultados. Essa replicação ocorria por questões técnicas, isto é, para atender necessidades específicas (corporativas e departamentais). A partir da fonte principal (banco de dados principal), diariamente, os dados eram replicados para um banco de dados que servia de suporte ao correio eletrônico e ao fluxo eletrônico de documentos, para um segundo banco de dados, para atender necessidades específicas (departamentais) e, por último, para atender necessidades de agregações de dados (*business intelligence*). A Figura 1 ilustra a arquitetura assim constituída.

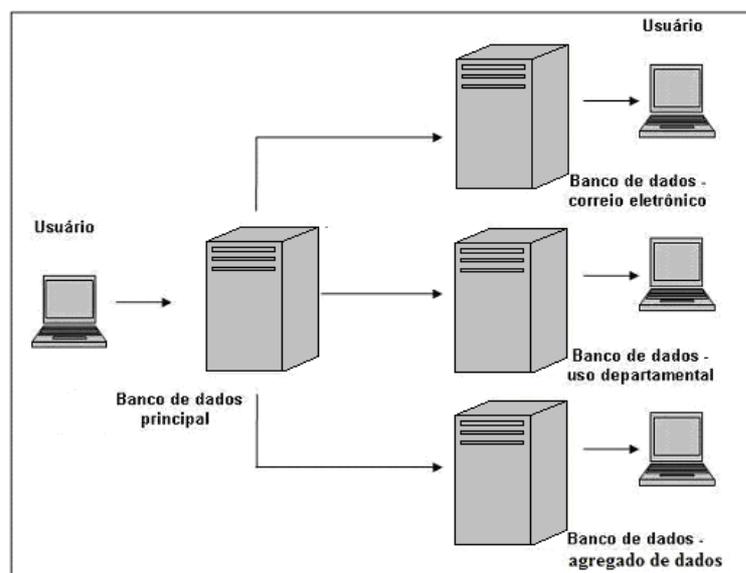
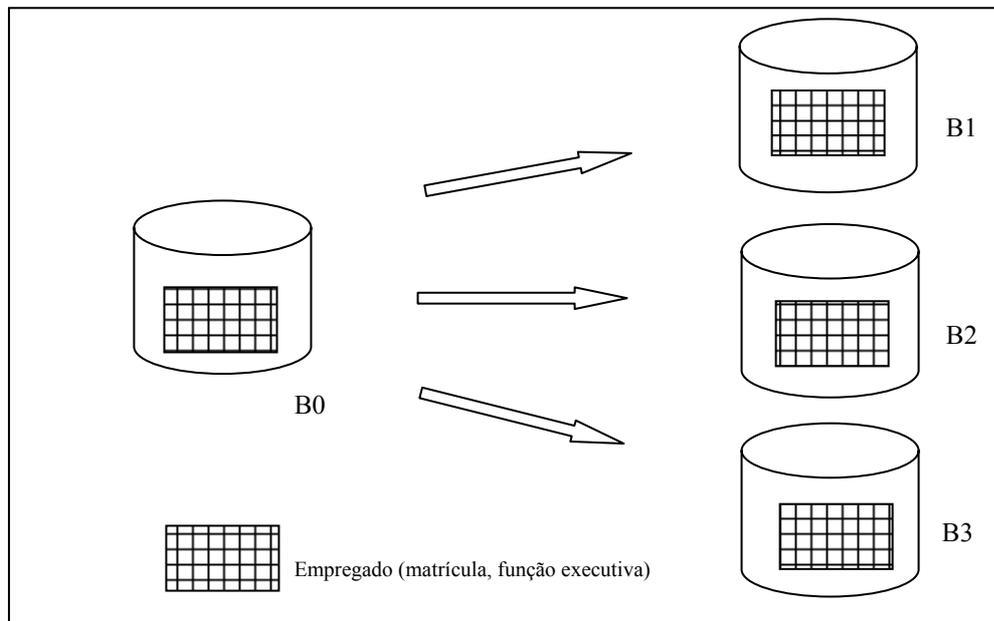


Figura 1 – Arquitetura de replicação de dados.

Ressalve-se que, no dia-a-dia, não há uma visão global dos dados, isto é, a visão é exclusivamente do banco de dados em uso (correio eletrônico, uso departamental ou agregado de dados). Mas, geralmente, ao se elaborar qualquer consulta, tinha-se o cuidado de comparar o resultado obtido com os resultados de consultas provenientes das demais fontes (bancos de dados). Cabia ao “decisor” (executivo responsável pela gestão e uso das informações), identificar a fonte mais confiável. O foco de atenção foi no “quantitativo de empregados” no exercício da função executiva de “chefe de departamento”.

## 8. Aplicação da TCA

Neste contexto, cada registro de um determinado arquivo de banco de dados, descreve um “empregado” (entidade) e tem, além de vários outros atributos, a matrícula que o identifica e a função executiva em exercício. Para simplificar, serão considerados apenas os atributos “matrícula” e a “função executiva”, ilustrados na Figura 2.



**Figura 2** – Replicação da entidade “Empregado”.

A atualização é efetuada no banco de dados B0 e os dados replicados para os demais bancos de dados B1, B2 e B3, conforme a sua finalidade (correio eletrônico, uso departamental e agregado de dados, respectivamente). Quando se realiza uma consulta aos dados replicados, tem-se uma visão única do banco de dados pesquisado (B1, B2 ou B3), o que não permite, de imediato, saber se a replicação ocorreu de forma perfeita. A consulta é restrita à fonte de dados pesquisada. Para simular a replicação com inconsistência do atributo “função executiva”, utilizou-se uma planilha eletrônica (Microsoft Excel), com a função matemática ALEATÓRIOENTRE. Foram considerados dados reais: em um universo aproximado de 2.000 (dois mil) empregados, há 69% (ou 1.381 empregados) com nível superior de instrução, no exercício (ou não) de função executiva: coordenador de serviços (CD), gerente

(GR), chefe de departamento (CH) e superintendente (SD). Na planilha, “NN” indica que um empregado não exerce função executiva. Ressalve-se que há 116 empregados que exercem a função executiva de “chefe de departamento”. Na planilha, cada linha simula um determinado empregado – a matrícula original foi substituída por um numeral seqüencial (coluna “Empr.”), e uma coluna para cada atributo “função executiva”, originário dos bancos de dados B0, B1, B2 e B3. Foram selecionados os dez primeiros registros (ou empregados), mostrados no Quadro 12.

**Quadro 12** – Simulação da replicação de dados.

<b>Empr.</b>	<b>B0</b>	<b>B1</b>	<b>B2</b>	<b>B3</b>
0001	GR	GR	GR	X
0002	CH	CH	CH	CH
0003	CH	CH	X	CH
0004	CH	CH	X	CH
0005	CH	CH	CH	CH
0006	CH	CH	CH	CH
0007	CH	X	CH	CH
0008	SD	SD	SD	SD
0009	NN	NN	NN	NN
0010	GR	GR	GR	GR

Essa simulação foi obtida pelo uso da função ALEATÓRIOENTRE, em dois momentos: para selecionar o registro (ALEATÓRIOENTRE(1;10)) e, para selecionar a base (ou banco de dados) B1, B2 ou B3, a ser replicada com inconsistência (ALEATÓRIOENTRE(1;3)), pela indicação de um “X” (“xis”). Esse procedimento foi executado quatro vezes. As inconsistências foram geradas nos registros “0001”, “0003”, “0004” e “0007” (Quadro 12). Para o universo considerado – empregados que exercem função executiva de “chefe de departamento” (CH), atribuiu-se “1” (um) para aqueles que foram replicados perfeitamente e “0” (zero) para aqueles que não o foram ou que não pertencem a esse universo, o que se indica no Quadro 13.

**Quadro 13** – Simulação da replicação de dados.

<b>Empr.</b>	<b>B0</b>	<b>B1</b>	<b>B2</b>	<b>B3</b>
0001	GR	0	0	0
0002	CH	1	1	1
0003	CH	1	0	1
0004	CH	1	0	1
0005	CH	1	1	1
0006	CH	1	1	1
0007	CH	0	1	1
0008	SD	0	0	0
0009	NN	0	0	0
0010	GR	0	0	0

Pelo Quadro 13, constata-se que há uma inconsistência para os empregados “0003”, “0004” e “0007”, com relação às replicações em B2, B2 e B1, respectivamente. Pela análise de “B1”, por exemplo, há uma relação de “indiscernibilidade” quanto aos empregados “0002”, “0003”, “0004”, “0005” e “0006”, tendo em vista que todos possuem o valor “1”. Para o conjunto de empregados (E), foram identificados os que pertencem ao conjunto aproximativo inferior ( $\underline{P}$ ), ou seja, aqueles que, com certeza, foram replicados corretamente (“1” em B1, B2 e B3):

$$\underline{P} E = \{E2, E5, E6\} \quad (22)$$

Para aqueles que podem ter sido replicados corretamente em B1, B2 e B3, obtém-se o conjunto aproximativo superior:

$$\overline{P} E = \{E1, E2, E3, E4, E5, E6, E7, E8, E9, E10\} \quad (23)$$

A região de fronteira é, portanto:

$$\overline{P} E - \underline{P} E = \{E1, E3, E4, E7, E8, E9, E10\} \quad (24)$$

Como os conjuntos aproximativos inferior ( $\underline{P} E$ ) e superior ( $\overline{P} E$ ) são distintos, deduz-se que o conjunto de empregados em questão pode ser tratado pela TCA. Para o conjunto de empregados considerados, foram estabelecidas as seguintes relações de equivalência:

$$R_{B1} = \{\{E2, E3, E4, E5, E6\}, \{E1, E7, E8, E9, E10\}\} \quad (25)$$

$$R_{B2} = \{\{E2, E5, E6, E7\}, \{E1, E3, E4, E8, E9, E10\}\} \quad (26)$$

$$R_{B3} = \{\{E2, E3, E4, E5, E6, E7\}, \{E1, E8, E9, E10\}\} \quad (27)$$

Essas relações foram obtidas da seguinte forma: primeiro, um subconjunto foi formado para aqueles empregados que possuíam o valor “1” (um) e um outro subconjunto para aqueles com valor “0” (zero). Tendo como base as relações anteriores e o algoritmo desenvolvido, foi estabelecida uma relação principal (**R**), identificando-se primeiro os empregados com valor “1” e, em seguida, os empregados subsequentes:

$$R = \{\{E2, E5, E6\}, \{E1, E8, E9, E10\}, \{E3, E4\}, \{E7\}\} \quad (28)$$

Para saber se cada relação  $R_{B1}$ ,  $R_{B2}$  ou  $R_{B3}$  é indispensável em relação a **R** (Pawlak, 1991), estabeleceu-se uma nova relação comum (**RR**), suprimindo-se, consecutivamente, as relações  $R_{B1}$ ,  $R_{B2}$  e  $R_{B3}$ :

$$RR_{B1} = \{R - \{R_{B1}\}\} = \{\{E2, E5, E6, E7\}, \{E1, E8, E9, E10\}, \{E3, E4\}\} \quad (29)$$

$$RR_{B2} = \{R - \{R_{B2}\}\} = \{\{E2, E3, E4, E5, E6\}, \{E1, E8, E9, E10\}, \{E7\}\} \quad (30)$$

$$RR_{B3} = \{R - \{R_{B3}\}\} = \{\{E2, E5, E6\}, \{E1, E8, E9, E10\}, \{E3, E4\}, \{E7\}\} \quad (31)$$

Como as relações  $RR_{B1}$  e  $RR_{B2}$  são diferentes de **R**,  $R_{B1}$  e  $R_{B2}$  são indispensáveis. Como  $RR_{B3}$  é igual à relação **R**,  $R_{B3}$  é dispensável. Para encontrar as “reduções”, deve-se identificar uma nova relação (**RT**) para cada par de relações  $\{B1, B3\}$  e  $\{B2, B3\}$ :

$$RT_{B1B3} = \{R - \{R_{B2}\}\} = \{\{E2, E3, E4, E5, E6\}, \{E1, E8, E9, E10\}, \{E7\}\} \quad (32)$$

$$RT_{B2B3} = \{R - \{R_{B1}\}\} = \{\{E2, E5, E6, E7\}, \{E1, E8, E9, E10\}, \{E3, E4\}\} \quad (33)$$

Como  $RT_{B_1B_3} \neq R_{B_1}$  e  $RT_{B_1B_3} \neq R_{B_3}$ , a relação  $\{B_1, B_3\}$  é uma “redução”. Como  $RT_{B_2B_3} \neq R_{B_2}$  e  $RT_{B_2B_3} \neq R_{B_3}$ , também a relação  $\{B_2, B_3\}$  é uma “redução”.

Dadas as reduções  $\{B_1, B_3\}$  e  $\{B_2, B_3\}$ , tem-se que:

$$\{B_1, B_3\} \cap \{B_2, B_3\} = \{B_3\} \quad (34)$$

ou seja, identifica-se que  $B_3$  é o “núcleo” desse sistema de informação, de acordo com a proposição  $CORE(P) = \cap RED(P)$ , onde  $RED(P)$  é a família de todas as “reduções” de  $P$  (Pawlak, 1991). Pelo Quadro 13, observa-se que não houve inconsistência na replicação para o banco de dados  $B_3$ , no que se refere ao universo de empregados com função executiva de chefe de departamento (CH). Por uma análise de sensibilidade, o banco de dados  $B_3$  representa a melhor alternativa ante as inconsistências detectadas nos outros bancos de dados ( $B_1$  e  $B_2$ ), e que está alinhada com a sugestão obtida pela aplicação da TCA. Para facilitar a aplicação da TCA, com a identificação das relações, aproximações inferior e superior, região de fronteira, medida da exatidão, indicação das reduções e do núcleo, o algoritmo foi implementado em linguagem de computador, Borland® Delphi/Pascal, versão 2007, pelo primeiro autor desse estudo. A Figura 3 ilustra a interface utilizada pelo programa desenvolvido.

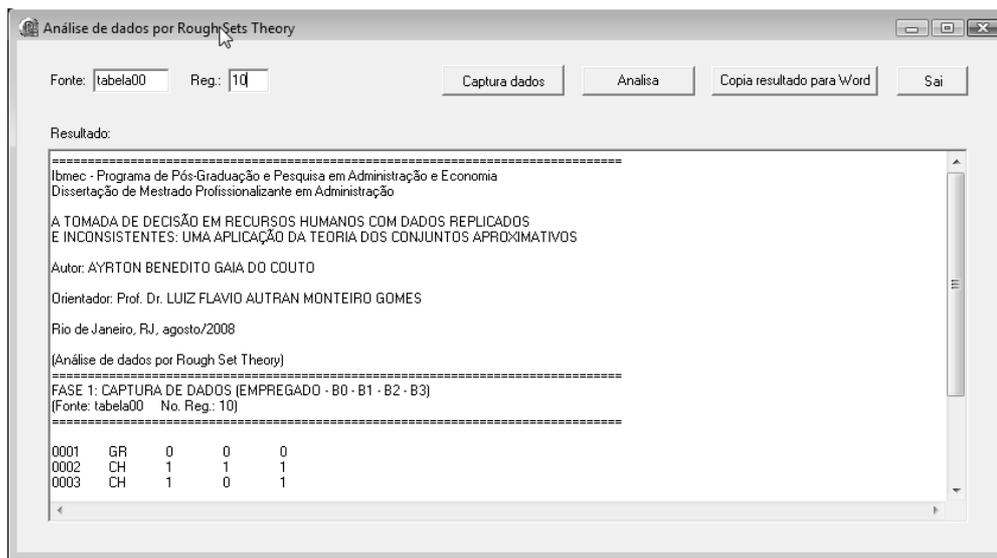
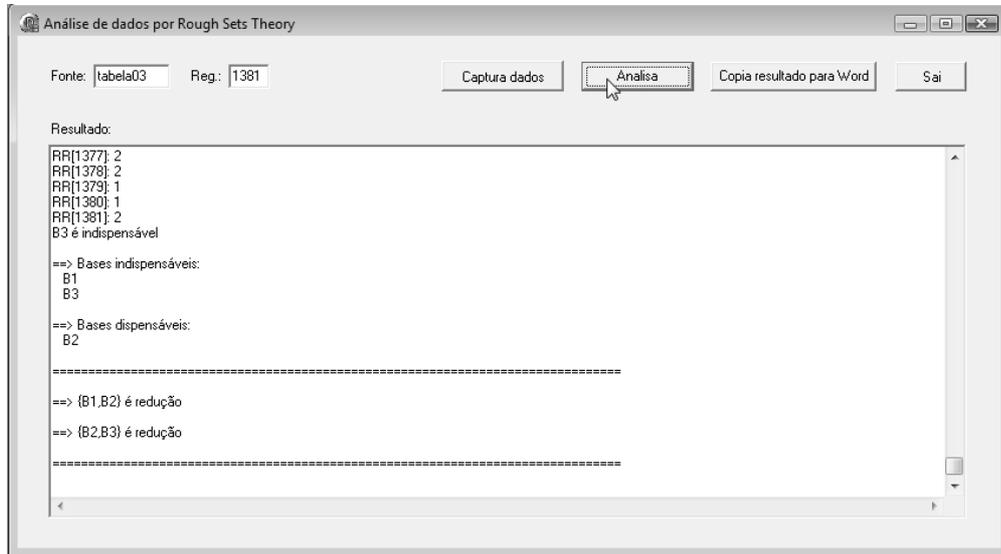


Figura 3 – Programa para captura e análise de dados.

Ademais, foram realizadas outras duas simulações:

1- Com 1.381 registros e 3 ocorrências de inconsistência: empregados “1313” (em  $B_1$ ), “0055” (em  $B_3$ ) e “0501” (em  $B_1$ ) (Figura 4). Nesse caso,  $B_2$  foi replicado corretamente.



**Figura 4** – Simulação com 3 ocorrências de inconsistência – fase: resultados.

Por uma análise de sensibilidade, constata-se que o banco de dados B2 foi perfeitamente replicado, comparando-se as indicações de atualização (“1”) com o que está registrado no banco de dados principal B0: há 116 empregados com função executiva de chefe de departamento (CH). Assim, a indicação do núcleo (B2) pela TCA – interseção dos conjuntos {B1, B2} e {B2, B3}, está de acordo com essa constatação. Ademais, esse resultado corrobora com o somatório de células com valor “1”, na coluna “B2”: há também 116 empregados com a função executiva de chefe de departamento (Quadro 14).

**Quadro 14** – Simulação com 3 ocorrências de inconsistência – somatório de “chefes de departamento”.

Empr	B0	B1	B2	B3
1368	CH	1	1	1
1369	CH	1	1	1
1370	GR	0	0	0
1371	GR	0	0	0
1372	GR	0	0	0
1373	CH	1	1	1
1374	CH	1	1	1
1375	CH	1	1	1
1376	CH	1	1	1
1377	GR	0	0	0
1378	SD	0	0	0
1379	CH	1	1	1
1380	CH	1	1	1
1381	GR	0	0	0
	<b>Σ</b>	<b>114</b>	<b>116</b>	<b>115</b>

Por questões de simplificação, o Quadro 14 reproduz somente o final da planilha.

2- Com 1.381 registros e 4 ocorrências de inconsistência: empregados “1313” (em B1), “0055” (em B3), “0501” (em B1) e “0202” (em B2) (Quadro 15 e Figura 5).

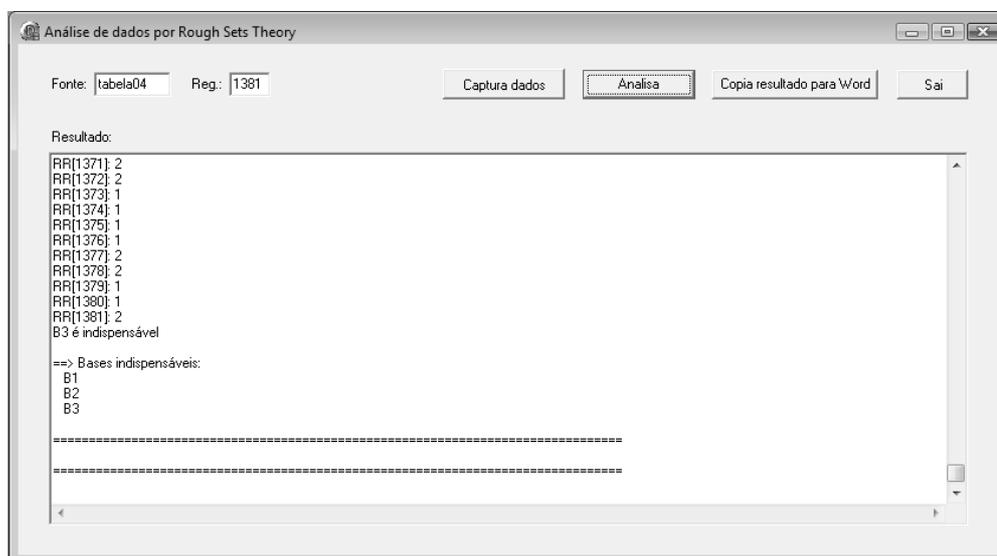
**Quadro 15** – Simulação com 4 ocorrências de inconsistência.

	A	B	C	D	E	F	G	H	I
1	Empr	B0	B1	B2	B3				
2	0001	GR	GR	GR	GR				
3	0002	CH	CH	CH	CH				
4	0003	CH	CH	CH	CH				
5	0004	CH	CH	CH	CH				
6	0005	CH	CH	CH	CH				
7	0006	CH	CH	CH	CH				
8	0007	CH	CH	CH	CH				
9	0008	SD	SD	SD	SD				
10	0009	NN	NN	NN	NN				
11	0010	GR	GR	GR	GR				
12	0011	CH	CH	CH	CH				
13	0012	NN	NN	NN	NN				
14	0013	CH	CH	CH	CH				
15	0014	NN	NN	NN	NN				

Empr	Base
1313	1
0055	3
0501	1
0202	2

Por questões de simplificação, o Quadro 15 reproduz somente o início da planilha.



**Figura 5** – Simulação com 4 ocorrências de inconsistência – fase: resultados.

Como houve inconsistência na replicação em B1, B2 e B3, o resultado sugerido indica que, todas as bases são “indispensáveis” (Figura 5). Nesse caso, também por uma análise de sensibilidade, não havia como indicar uma base de referência para consulta, dada à inconsistência gerada em cada base. Provavelmente, o melhor resultado da pesquisa

(quantitativo de chefes de departamento) deveria ser alguma “composição” dos resultados obtidos de cada base (Quadro 16), ou, simplesmente, ser descartado e se aguardar por uma nova replicação.

**Quadro 16** – Simulação com 4 ocorrências de inconsistência – somatório de “chefes de departamento”.

Empr	B0	B1	B2	B3
1368	CH	1	1	1
1369	CH	1	1	1
1370	GR	0	0	0
1371	GR	0	0	0
1372	GR	0	0	0
1373	CH	1	1	1
1374	CH	1	1	1
1375	CH	1	1	1
1376	CH	1	1	1
1377	GR	0	0	0
1378	SD	0	0	0
1379	CH	1	1	1
1380	CH	1	1	1
1381	GR	0	0	0
<b>Σ</b>		<b>114</b>	<b>115</b>	<b>115</b>

Por questões de simplificação, o Quadro 16 reproduz somente o final da planilha.

Com referência à complexidade computacional, de acordo com Toscani & Veloso (2008), o cálculo da complexidade (de um algoritmo) concentra-se em determinar a ordem de magnitude do número de operações fundamentais na execução do algoritmo. A operação escolhida como fundamental deve ser tal que, a contagem do número de vezes que ela é executada sobre uma entrada expresse a quantidade de trabalho (“esforço computacional” ou custo) do algoritmo, dispensando outras medidas.

Para o algoritmo de pesquisa em questão, o seu desenvolvimento consistiu de 6 (seis) fases (Quadro 17). Para obter o custo de complexidade para cada fase, conforme Anexo II, determinou-se que a operação fundamental seria o “loop” de pesquisa (“faça enquanto ... fim-faça”) utilizado em cada fase, para o “pior caso”.

**Quadro 17** – Fases e custos da complexidade do algoritmo de pesquisa.

Fase do algoritmo	Custo da complexidade
1- Captura dados e obtém aproximações inferior e superior	$n$
2- Obtém região de fronteira; calcula exatidão (accuracy)	$n^2$
3- Preenche <i>arrays</i> BB1, BB2 e BB3, em suporte à obtenção da relação de equivalência principal “R”	$n$
4- Obtém a relação principal (R)	$n + n^2$
5- Identifica bases (in)dispensáveis	$3[2n + n^2]$
6- Verifica se há reduções	$3n$

Segundo Wang *et al.* (2006), encontrar reduções (mínimas) para sistemas de informação ou tabelas de decisão tornou-se um problema “complexo” (“NP-complete problem”), tendo em vista a possibilidade da existência de mais de uma redução. Assim, para o contexto dessa pesquisa, o algoritmo desenvolvido limitou-se a pesquisar as aproximações inferior e superior, a região de fronteira, calcular a medida da exatidão, obter a relação de equivalência principal, e identificar as bases de dados (in)dispensáveis e as reduções e possível núcleo para um sistema de informação com 3 (três) “atributos” (função executiva “chefe de departamento” oriunda das bases replicadas B1, B2 e B3), face a dificuldade conhecida de se pesquisar reduções para muitos (“n”) atributos de um sistema de informação.

## 9. Conclusões e sugestões para pesquisas futuras

Para o ambiente da empresa em questão (instituição financeira nacional) e, ante a situação encontrada – bancos de dados replicados e inconsistentes, cuja inconsistência é de natureza desconhecida para o estudo em questão e de ocorrência esporádica, coube ao “decisor” (executivo responsável pela gestão e uso das informações), adotar, inicialmente, um padrão de conferência dos resultados obtidos: de forma manual, cada resultado de consulta (ex. quantitativo de pessoal) era confrontado com consultas extraídas de mais de um banco de dados. Ressalve-se que, na simulação, há uma visão completa dos dados replicados e da sua origem (B0). Porém, no ambiente da empresa em estudo, a consulta aos dados replicados é restrita, ou seja, é realizada por meio de uma aplicação específica com acesso a uma única fonte de dados (B1, B2 ou B3). Não há como comparar o resultado de uma consulta com a origem dos dados (B0), que não seja de forma manual. A TCA, como método de apoio à decisão, mostrou-se adequada para o tratamento da “indiscernibilidade”, mediante as indicações das “reduções” e de um “núcleo” de dados, quando foi possível. A sua aplicabilidade provém da própria natureza dos dados pesquisados (da área de RH da empresa): neste caso, não há qualquer informação adicional sobre a ocorrência das inconsistências, mas tão somente, o próprio dado. Nesse contexto, este estudo revelou-se uma alternativa viável para a tomada de decisão com dados replicados e inconsistentes, dentre as prováveis alternativas de solução, mesmo que não seja possível obter uma fonte de referência para consulta (Quadro 16). Nesse caso, torna-se uma ferramenta indicativa de que uma nova replicação deve ser efetuada com o intuito de dirimir as inconsistências observadas.

Ao abordar o problema de decisão com dados replicados e inconsistentes, esta pesquisa restringiu-se a um cenário com três bancos de dados replicados, 1.381 registros (ou empregados) e um atributo (função executiva) em questão. Inicialmente, uma oportunidade que se deriva é a TCA aplicada a um número maior de casos de replicação com inconsistência, para um universo maior de empregados e para um conjunto mais amplo de atributos, com o objetivo de averiguar se os resultados correspondem às expectativas. Nesse caso, deve-se adaptar o algoritmo e, conseqüentemente, o *software* desenvolvido, para a captura e análise dos dados.

Um outro papel para a TCA que se conjectura seria como ferramenta de auxílio ao monitoramento das replicações de dados, necessário à minimização ou eliminação das conseqüências indesejáveis, que podem advir de uma decisão com dados replicados e inconsistentes. Como, pela própria abordagem da TCA, não é preciso qualquer informação adicional sobre os dados em questão (ex. frequência de ocorrência das inconsistências), sugere-se o mesmo tipo de pesquisa por meio de outras abordagens (ex. *Fuzzy Set Theory*, *Neuro-Fuzzy* com TCA etc.), com o objetivo de avaliar a relação “custo versus benefícios” para cada abordagem utilizada.

A extensão da TCA proposta por Ziarko (1993; 1993a), pelo uso do “modelo VP” (*variable precision*), mostra-se uma alternativa interessante para o tratamento da indiscernibilidade de dados com dependência entre si. As restrições apontadas na TCA e a proposta de um modelo alternativo (“relação de dominância”), por Greco, Matarazzo & Slowinski (2005), mostraram-se, também, uma alternativa à TCA clássica, que pode ser aplicada ao estudo em questão.

### Agradecimento

Os autores agradecem ao CNPq pelo apoio na elaboração deste artigo através do Processo Número 310603/2009-9.

### Referências Bibliográficas

- (1) Bit, M. & Beaubouef, T. (2008). Rough set uncertainty for robotic systems. *Journal of Computing Sciences in Colleges, ACM*, **23**(6), 126-132.
- (2) Codd, E.F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, **13**(6), 377-387.
- (3) Das-Gupta, P. (1988). Rough sets and information retrieval. SIGIR'88: Proceedings of the 11<sup>th</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, p. 567-581.
- (4) Date, C.J. (1984). *Introdução a sistemas de bancos de dados*. Campus, Rio de Janeiro.
- (5) Flinkman, M.; Michalowski, W.; Nilsson, S.; Slowinski, R.; Susmaga, R. & Wilk, S. (2000). Use of rough sets analysis to classify siberian forest ecosystems according to net primary production of phytomass. *Information Systems & Operational Research*, **38**(3), 145-160.
- (6) Gomes, L.F.A.M. & Gomes, C.F.S. (2001). Uma técnica de data mining: princípios básicos dos conjuntos aproximativos e suas aplicações. *Revista ANGRAD*, **2**(1), 13-22.
- (7) Gomes, L.F.A.M.; Gomes, C.F.S. & Almeida, A.T. (2006). *Tomada de decisão gerencial: enfoque multicritério*. Atlas, São Paulo.
- (8) Greco, S.; Matarazzo, B. & Slowinski, R. (2005). Decision rule approach. **In: Multiple criteria decision analysis state of the art surveys** [edited by J. Figueira, S. Greco and M. Ehrgott]. Springer Science, New York + Business media, cap. 13, p. 507-561.
- (9) Grzymala-Busse, J.W. (1988). Knowledge acquisition under uncertainty – a rough set approach. *Journal of Intelligent and Robotic Systems*, **1**, 3-16.
- (10) Hassanien, A.E.; Abdelhafez, M.E. & Own, H.S. (2008). *Rough sets data analysis in knowledge discovery: a case of kuwaiti diabetic children patients*. Hindawi Publishing Corporation, Advances in Fuzzy Systems, a. id 528461, p. 1-13.
- (11) Huang, C.-C.; Tseng, T.-L.; Chuang, H.-F & Liang, H.-F. (2006). Rough-set-based approach to manufacturing process document retrieval. *International Journal of Production Research*, **44**(14), 2889-2911.
- (12) Lin, T.Y. (2008). *Rough set theory in very large databases*. Disponível em: [http://www.cs.sjsu.edu/~tylin/publications/paperList/82\\_rs\\_dm8.pdf](http://www.cs.sjsu.edu/~tylin/publications/paperList/82_rs_dm8.pdf).

- (13) Nowicki, R. (2008). On combining neuro-fuzzy architectures with the rough set theory to solve classification problems with incomplete data. *IEEE Transactions on Knowledge and Data Engineering*. Disponível em: <<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/69/4358933/04487067.pdf?tp=&arnumber=4487067&isnumber=4358933>>.
- (14) Pawlak, Z. (2000). Rough sets and decision analysis. *Information Systems & Operational Research*, **38**(3), 132-144.
- (15) \_\_\_\_\_. (1991). *Rough sets. Theoretical aspects of reasoning about data*. Kluwer Academic Publishers, Dordrecht.
- (16) Pawlak, Z.; Grzymala-Busse, J.; Slowinski, R. & Ziarko, W. (1995). Rough sets. *Communications of the ACM*, **38**(11), 89-95.
- (17) Pawlak, Z. & Slowinski, R. (1994). Rough set approach to multi-attribute decision analysis. *European Journal of Operational Research*, Invited Review, **72**, 443-459.
- (18) Roy, B. & Bouyssou, D. (1993). *Aide Multicritère à la Décision: Méthodes et Cas*. Economica, Paris.
- (19) Shao, X.-Y.; Wang Z.-H.; Li, P.-G. & Feng, C.-X.J. (2006). Integrating data mining and rough set for customer group-based discovery of product configuration rules. *International Journal of Production Research*, **44**(14), 2789-2811.
- (20) Son, S.H. (1988). Replicated data management in distributed database systems. *Sigmod Record*, **17**(4), 62-69.
- (21) Tam, C.M.; Tong, T.K.L. & Chan, K.K. (2006). Rough set theory for distilling construction safety measures. *Construction Management and Economics*, **24**(24), 1199-1206.
- (22) Toscani, L.V. & Veloso, P.A.S. (2008). *Complexidade de algoritmos*. Bookman, São Paulo.
- (23) Tsumoto, S. (2000). Automated knowledge discovery in clinical databases based on rough set model. *Information Systems & Operational Research*, **38**(3), 196-207.
- (24) Xiaoyue, W. & Rujiang, B. (2006). An effective hybrid classifier based on rough sets and neural networks. *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*.
- (25) Wang, R.; Miao, D. & Hu, G. (2006). Discernibility matrix based algorithm for reduction of attributes. *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*.
- (26) Ziarko, W. (1993). Analysis of uncertain information in the framework of variable precision rough sets. *Foundations of Computing and Decision Sciences*, **18**(3-4), 381-396.
- (27) \_\_\_\_\_. (1999). Discovery through rough set theory. *Communications of the ACM*, **42**(11), 55-57.
- (28) \_\_\_\_\_. (1993a). Variable precision rough set model. *Journal of Computer and System Sciences*, **46**(1), 39-59.
- (29) Ziarko, W. & Wong, S.K.M. (1986). On learning and evaluation of decision rules in the context of rough sets. *Proceedings of the ACM SIGART International Symposium on Methodologies for Intelligent Systems*, Knoxville, p. 308-324.

### Anexo I – Exemplo de aplicação de conceitos da TCA

Em Pawlak (2000), utiliza uma disposição de dados em forma de tabela (*database*) – com linhas e colunas, para exemplificar outros conceitos. Este exemplo (Quadro 1) é composto de seis lojas e quatro atributos (aspectos quantitativos ou qualitativos):

**Quadro 1** – Tabela-exemplo.  
Fonte: Adaptado de Pawlak (2000).

Loja	E	Q	L	P
1	Alta	Boa	Não	Lucro
2	Média	Boa	Não	Prejuízo
3	Média	Boa	Não	Lucro
4	Sem	Média	Não	Prejuízo
5	Média	Média	Sim	Prejuízo
6	Alta	Média	Sim	Lucro

No Quadro 1 tem-se: E – autonomia dos vendedores; Q – qualidade da mercadoria; L – localização com trânsito intenso; P – resultado (lucro ou prejuízo). Cada loja é caracterizada pelos atributos E, Q, L e P. Assim, todas as lojas são “discerníveis” pelo emprego das informações disponibilizadas por esses atributos. Porém, as lojas 2 e 3 são “indiscerníveis” em termos dos atributos E, Q e L, tendo em vista que possuem os mesmos valores para esses atributos. Cada subconjunto de atributos determina uma “partição” (“classificação”) de todos os objetos em “classes”, que têm a mesma descrição em termos daqueles atributos. Por exemplo, os atributos Q e L agregam todas as lojas nas classes {1,2,3}, {4} e {5,6}, conforme indicado no Quadro 2.

**Quadro 2** – Tabela-exemplo.  
Fonte: Adaptado de Pawlak (2000).

Loja	Q	L
1	Boa	Não
2	Boa	Não
3	Boa	Não
4	Média	Não
5	Média	Sim
6	Média	Sim

Considere-se agora, o seguinte problema: quais são as características das lojas que realizaram lucro (ou tiveram prejuízo) em termos dos atributos E, Q e L? Isto é, o interesse está em descrever o conjunto (conceito) {1,3,6} (ou {2,4,5}), mostrado no Quadro 1. Identifica-se facilmente que essa questão não pode ser respondida de modo único, porque as lojas 2 e 3 têm as mesmas características quanto aos atributos E, Q e L, mas a loja 2 teve prejuízo, enquanto a loja 3 realizou lucro. Com base no Quadro 1, pode-se afirmar que: as lojas 1 e 6 realizaram lucro, as lojas 4 e 5 tiveram prejuízo e as lojas 2 e 3 não podem ser classificadas (em lucro ou prejuízo). O Quadro 3 mostra o resultado dessa análise.

**Quadro 3** – Tabela-exemplo.  
Fonte: Adaptado de Pawlak (2000).

Loja	E	Q	L	P	Resultado
1	Alta	Boa	Não	Lucro	LUCRO
2	Média	Boa	Não	Prejuízo	?
3	Média	Boa	Não	Lucro	?
4	Sem	Média	Não	Prejuízo	PREJUÍZO
5	Média	Média	Sim	Prejuízo	PREJUÍZO
6	Alta	Média	Sim	Lucro	LUCRO

Empregando os atributos E, Q e L, deduz-se que: as lojas 1 e 6 certamente realizaram lucro, isto é, certamente pertencem ao conjunto {1,3,6}; enquanto as lojas 1, 2, 3 e 6 possivelmente realizaram lucro, isto é, possivelmente pertencem ao conjunto {1,3,6}. Os conjuntos {1,6} e {1,2,3,6} representam, respectivamente, as aproximações “inferior” e “superior” do conjunto {1,3,6}. O conjunto {2,3} representa a diferença entre as aproximações superior e inferior e caracteriza a “região de fronteira” do conjunto {1,3,6}.

## Anexo II – Fases e custos da complexidade do algoritmo de pesquisa

### 1- Fase 1: Captura dados e obtém aproximações inferior e superior

```
n_quant    ←      “quantidade de registros informada”
cont       ← 0      // contador de registros
n_reg_inf  ← 0      // número de elementos da aproximação inferior
n_reg_sup  ← 0      // número de elementos da aproximação superior
```

faça enquanto (cont < n\_quant)

```
  cont ← cont + 1
```

```
  obter registro[cont]
```

```
  v_empregado[cont] ← matrícula
```

```
  v_b0[cont] ← dado // obtém dado da base 0
```

```
  v_b1[cont] ← dado // obtém dado da base 1
```

```
  v_b2[cont] ← dado // obtém dado da base 2
```

```
  v_b3[cont] ← dado // obtém dado da base 3
```

```
  se (v_b1[cont]+v_b2[cont]+v_b3[cont]) = “111” então // não há inconsistência
```

```
    n_reg_inf ← n_reg_inf + 1
```

```
    v_reg_inf[n_reg_inf] ← v_empregado[cont] // obtém aprox. inferior
```

```
  senão
```

```
    n_reg_sup ← n_reg_sup + 1
```

```
    v_reg_sup[n_reg_sup] ← v_empregado[cont] // obtém aprox. superior
```

```
  fim-se
```

fim-faça

- **Cálculo da complexidade:** nessa fase, há a captura de dados (de matrícula, da função executiva (base B0) e se houve (“1”) ou não (“0”) atualização nas bases B1, B2 e B3), e a identificação da aproximação: se é inferior (“111”, para as bases B1, B2 e B3, respectivamente) ou superior; a ordem de magnitude do número de operações fundamentais é “n”.

### **2- Fase 2: Obtém região de fronteira; calcula exatidão (accuracy)**

```
i ← 0
r_exatidao ← 0,00

faça enquanto (i < n_reg_sup)
  i ← i + 1
  k ← 1
  regioao_frenteira ← true

  faça enquanto (k <= n_reg_inf) e (regiao_frenteira)
    se (v_reg_sup[i] = v_reg_inf[k]) então
      regioao_frenteira ← false
    fim-se
    k ← k + 1
  fim-faça

  se regioao_frenteira então
    exibir(v_reg_sup[i]) // exibe elemento da região de fronteira
  fim-se

fim-faça

r_exatidao ← (n_reg_inf / n_reg_sup)
```

- **Cálculo da complexidade:** como nessa fase há a comparação dos elementos das aproximações inferior e superior, a ordem de magnitude do número de operações fundamentais é “n x n” (no pior caso) ou “n<sup>2</sup>”.

### **3- Fase 3: Preenche arrays BB1, BB2 e BB3, em suporte à obtenção da relação de equivalência principal “R”**

```
i ← 0

faça enquanto (i < n_quant)
  i ← i + 1

  se v_b1[i] = '0' então
    v_BB1[i] ← '2'
  senão
    se v_b1[i] = '1' então
```

```
    v_BB1[i] ← '1'
  fim-se
fim-se

se v_b2[i] = '0' então
  v_BB2[i] ← '2'
senão
  se v_b2[i] = '1' então
    v_BB2[i] ← '1'
  fim-se
fim-se

se v_b3[i] = '0' então
  v_BB3[i] ← '2'
senão
  se v_b3[i] = '1' então
    v_BB3[i] ← '1'
  fim-se
fim-se
```

fim-faça

**- Cálculo da complexidade: nessa fase, a ordem de magnitude do número de operações fundamentais é “n”.**

#### **4- Fase 4: Obtém a relação principal (R)**

```
i ← 0

faça enquanto (i < n_quant)
  i ← i + 1
  se (v_BB1[i] = '1') e (v_BB2[i] = '1') e (v_BB3[i] = '1') então
    v_R[i] ← '1'
  fim-se
fim-faça

k ← 2
b1 ← "
b2 ← "
b3 ← "
preenchido ← false

faça enquanto não (preenchido)

  preenchido ← true
  i ← 0

  faça enquanto (i < n_quant)
```

```

i ← i + 1

se (v_R[i] = ") então
  preenchido ← false
  se (b1 = ") então
    v_R[i] ← inttostr(k) // converte inteiro em "string"
    b1 ← v_BB1[i]
    b2 ← v_BB2[i]
    b3 ← v_BB3[i]
  senão
    se (v_BB1[i] = b1) e (v_BB2[i] = b2) e (v_BB3[i] = b3) então
      v_R[i] ← inttostr(k)// converte inteiro em "string"
    fim-se
  fim-se
fim-se

fim-faça

k ← k + 1
b1 ← "

fim-faça

```

- **Cálculo da complexidade:** nessa fase, a ordem de magnitude do número de operações fundamentais é “n” + “n x n” (no pior caso) ou “n + n<sup>2</sup>”.

### **5- Fase 5: Identifica bases (in)dispensáveis**

// ==> **Analisa se B1 é indispensável**

```

i ← 0

faça enquanto (i < n_quant)
  i ← i + 1
  se (v_BB2[i] = '1') e (v_BB3[i] = '1') então
    v_RR[i] ← '1'
  fim-se
fim-faça

k ← 2
b2 ← "
b3 ← "
preenchido ← false

faça enquanto não (preenchido)
  preenchido ← true
  i ← 0

```

```

faça enquanto (i < n_quant)

    i ← i + 1
    se (v_RR[i] = "") então
        preenchido ← false
        se (b2 = "") então
            v_RR[i] ← inttostr(k)
            b2 ← v_BB2[i]
            b3 ← v_BB3[i]
        senão
            se (v_BB2[i] = b2) e (v_BB3[i] = b3) então
                v_RR[i] ← inttostr(k)
            fim-se
        fim-se
    fim-se

fim-faça

k ← k + 1
b2 ← ""

fim-faça

i ← 0
B1_indispensavel ← false

faça enquanto (i < n_quant) e não (B1_indispensavel)
    i ← i + 1
    v_b2_b3[i] ← v_RR[i]
    se (v_R[i] # v_RR[i]) então
        B1_indispensavel ← true
    fim-se
fim-faça

se B1_indispensavel então
    exhibe '==> B1 é indispensável'
senão
    exhibe '==> B1 é dispensável'
fim-se

// ==> Analisa se B2 é indispensável

i ← 0
faça enquanto (i < n_quant)
    i ← i + 1
    se (v_BB1[i] = '1') e (v_BB3[i] = '1') então
        v_RR[i] ← '1'
    fim-se
fim-faça

```

```
k ← 2
b1 ← "
b3 ← "
preenchido ← false

faça enquanto não (preenchido)
  preenchido ← true
  i ← 0

  faça enquanto (i < n_quant)

    i ← i + 1
    se (v_RR[i] = "") então
      preenchido ← false
      se (b1 = "") então
        v_RR[i] ← inttostr(k)
        b1 ← v_BB1[i]
        b3 ← v_BB3[i]
      senão
        se (v_BB1[i] = b1) e (v_BB3[i] = b3) então
          v_RR[i] ← inttostr(k)
        fim-se
      fim-se
    fim-se

  fim-faça

  k ← k + 1
  b1 ← "

fim-faça

i ← 0
B2_indispensavel ← false

faça enquanto (i < n_quant) e não (B2_indispensavel)
  i ← i + 1
  v_b1_b3[i] ← v_RR[i]
  se (v_R[i] # v_RR[i]) então
    B2_indispensavel ← true
  fim-se
fim-faça

se B2_indispensavel então
  exhibe '==> B2 é indispensável'
senão
  exhibe '==> B2 é dispensável'
fim-se
```

// ==> **Analisa se B3 é indispensável**

```
i ← 0
faça enquanto (i < n_quant)
    i ← i + 1
    se (v_BB1[i] = '1') e (v_BB2[i] = '1') então
        v_RR[i] ← '1'
    fim-se
fim-faça

k ← 2
b1 ← ""
b2 ← ""
preenchido ← false

faça enquanto não (preenchido)
    preenchido ← true
    i ← 0

    faça enquanto (i < n_quant)

        i ← i + 1
        se (v_RR[i] = "") então
            preenchido ← false
            se (b1 = "") então
                v_RR[i] ← inttostr(k)
                b1 ← v_BB1[i]
                b2 ← v_BB2[i]
            senão
                se (v_BB1[i] = b1) e (v_BB2[i] = b2) então
                    v_RR[i] ← inttostr(k)
                fim-se
            fim-se
        fim-se

    fim-faça

    k ← k + 1
    b1 ← ""

fim-faça

i ← 0
B3_indispensavel ← false

faça enquanto (i < n_quant) e não (B3_indispensavel)
    i ← i + 1
    v_b1_b2[i] ← v_RR[i]
    se (v_R[i] ≠ v_RR[i]) então
        B3_indispensavel ← true
```

```
fim-se
fim-faça

se B3_indispensavel então
  exhibe '==> B3 é indispensável'
senão
  exhibe '==> B3 é dispensável'
fim-se
```

- **Cálculo da complexidade:** nessa fase, a ordem de magnitude do número de operações fundamentais para as três bases de dados é  $3 \times [“n” + “n \times n”]$  (no pior caso) + “n” (no pior caso) ou  $“3[2n + n^2]”$ .

#### **6- Fase 6: Verifica se há reduções**

**// verifica se {b1,b2} é redução**

```
se (B1_indispensavel e não B2_indispensavel) ou (não B1_indispensavel e
B2_indispensavel) então
```

```
b0 ← "
b1 ← "
b2 ← "
redução ← false
```

```
i ← 0
faça enquanto (i < n_quant) e não (redução)
  i ← i + 1
  b0 ← b0 + v_b1_b2[i]
  b1 ← b1 + v_BB1[i]
  b2 ← b2 + v_BB2[i]
  se (b0 # b1) e (b0 # b2) então
    exhibe '==> {B1,B2} é redução'
    redução ← true
fim-se
fim-faça
```

```
fim-se
```

**// verifica se {b1,b3} é redução**

```
se (B1_indispensavel e não B3_indispensavel) ou (não B1_indispensavel e
B3_indispensavel) então
```

```
b0 ← "
b1 ← "
b3 ← "
redução ← false
```

```
i ← 0
faça enquanto (i < n_quant) e não (redução)
  i ← i + 1
  b0 ← b0 + v_b1_b3[i]
  b1 ← b1 + v_BB1[i]
  b3 ← b3 + v_BB3[i]
  se (b0 # b1) e (b0 # b3) então
    exibe '==> {B1,B3} é redução'
    redução ← true
  fim-se
fim-faça

fim-se
```

**// verifica se {b2,b3} é redução**

se (B2\_indispensavel e não B3\_indispensavel) ou (não B2\_indispensavel e B3\_indispensavel) então

```
b0 ← "
b2 ← "
b3 ← "
redução ← false
```

```
i ← 0
faça enquanto (i < n_quant) e não (redução)
  i ← i + 1
  b0 ← b0 + v_b2_b3[i]
  b2 ← b2 + v_BB2[i]
  b3 ← b3 + v_BB3[i]
  se (b0 # b2) e (b0 # b3) então
    exibe '==> {B2,B3} é redução'
    redução ← true
  fim-se
fim-faça

fim-se
```

**- Cálculo da complexidade: nessa fase, a ordem de magnitude do número de operações fundamentais para as três reduções é “3 x n (no pior caso)” ou “3n”.**