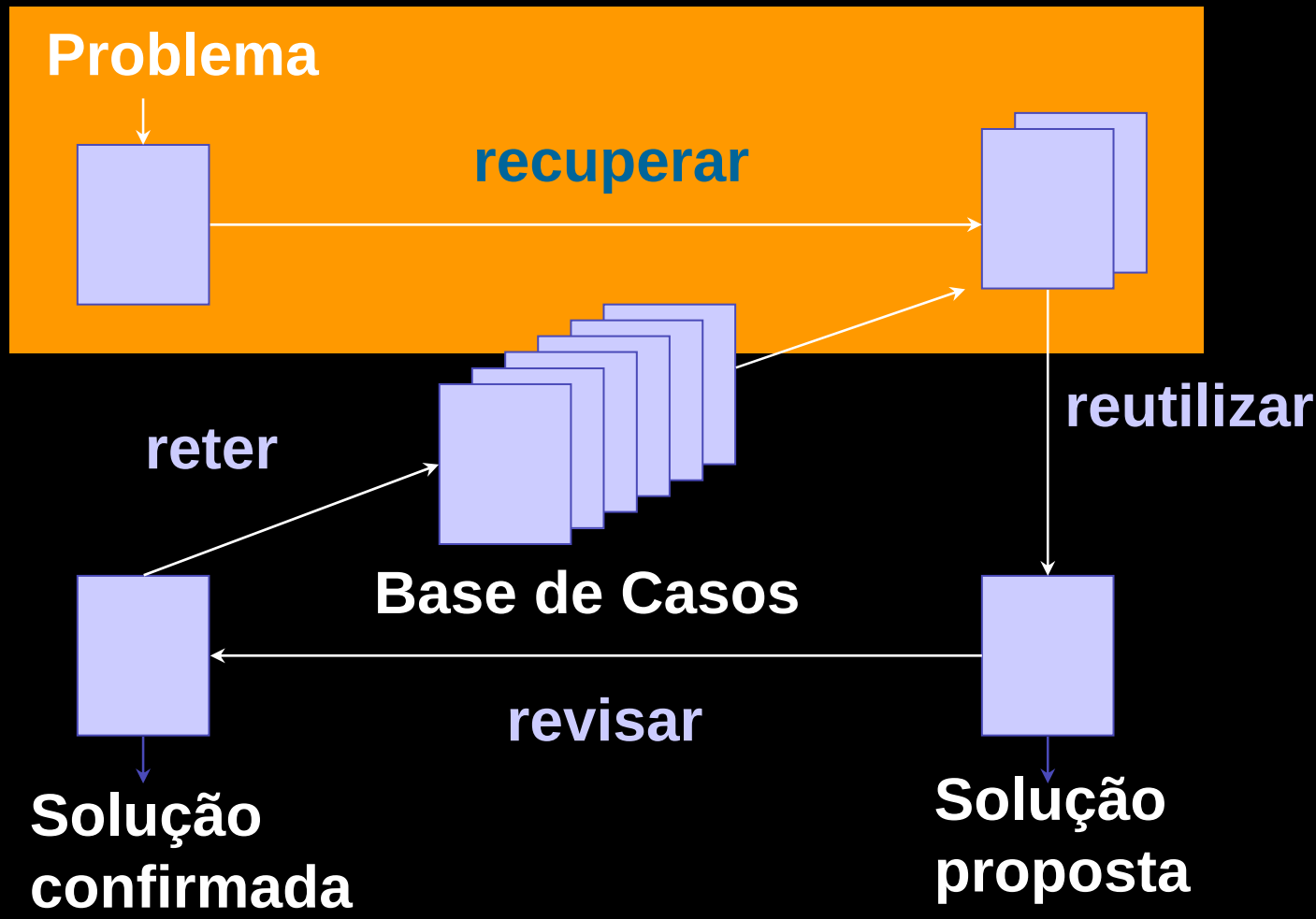
Raciocínio Baseado em Casos
3. Recuperação de Casos

Prof. Aldo von Wangenheim
Disciplinas:
- Raciocínio Baseado em Casos - PPGCC/INE/UFSC
- Sistemas de Raciocínio e Gestão Baseados em Casos - EGC/UFSC

Ciclo de RBC - Recuperação



Como identificar casos uteis?

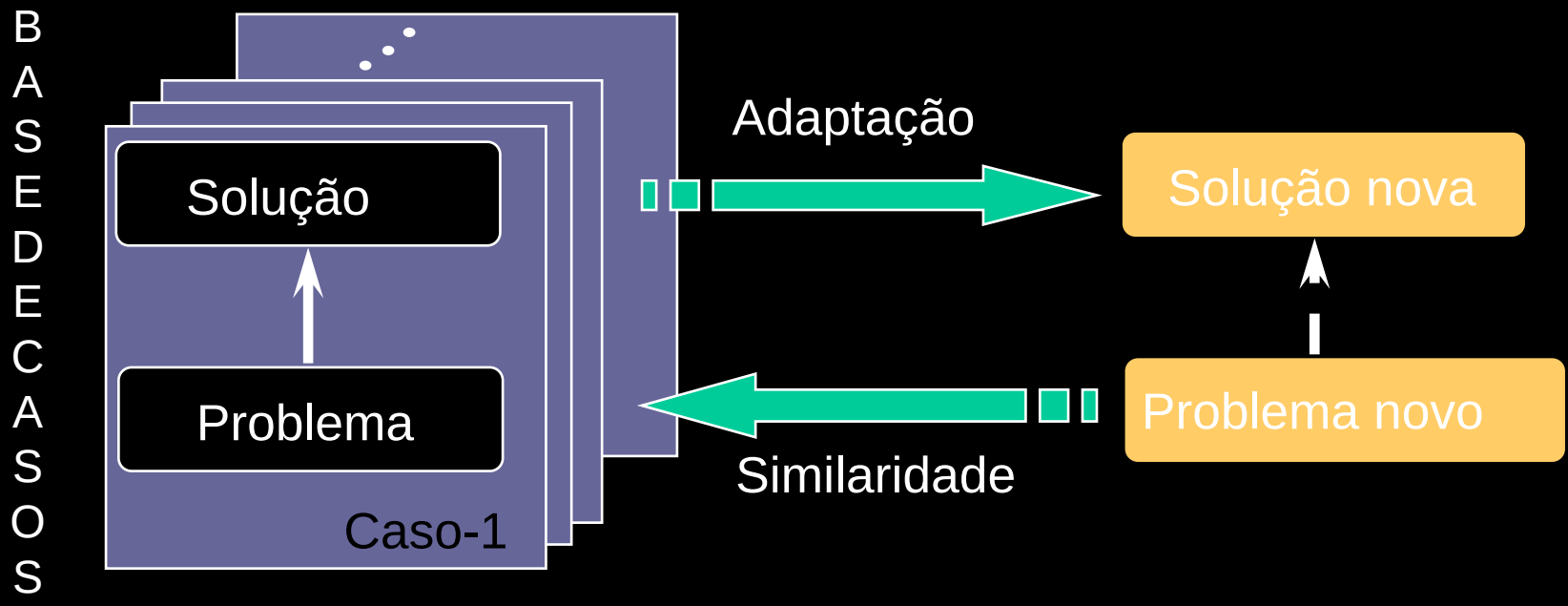
- Procura-se por caso(s) na base, que, na situação atual é **útil** para determinar a sua solução.
- O que significa um caso ser útil para solucionar um determinado problema?
- Hipótese: **Problemas similares possuem soluções semelhantes**

⇨O critério *a posteriori* da **utilidade de soluções** passa a ser reduzido ao critério *a priori* **similaridade de descrições de problema**: Um caso é útil se ele é similar à questão atual.

⇨Busca de casos **similares**

- Objetivo da similaridade:
  - Selecionar casos que possam ser facilmente adaptados para o problema atual
  - Selecionar casos que quase têm a mesma solução como o problema atual

Recuperação de casos similares



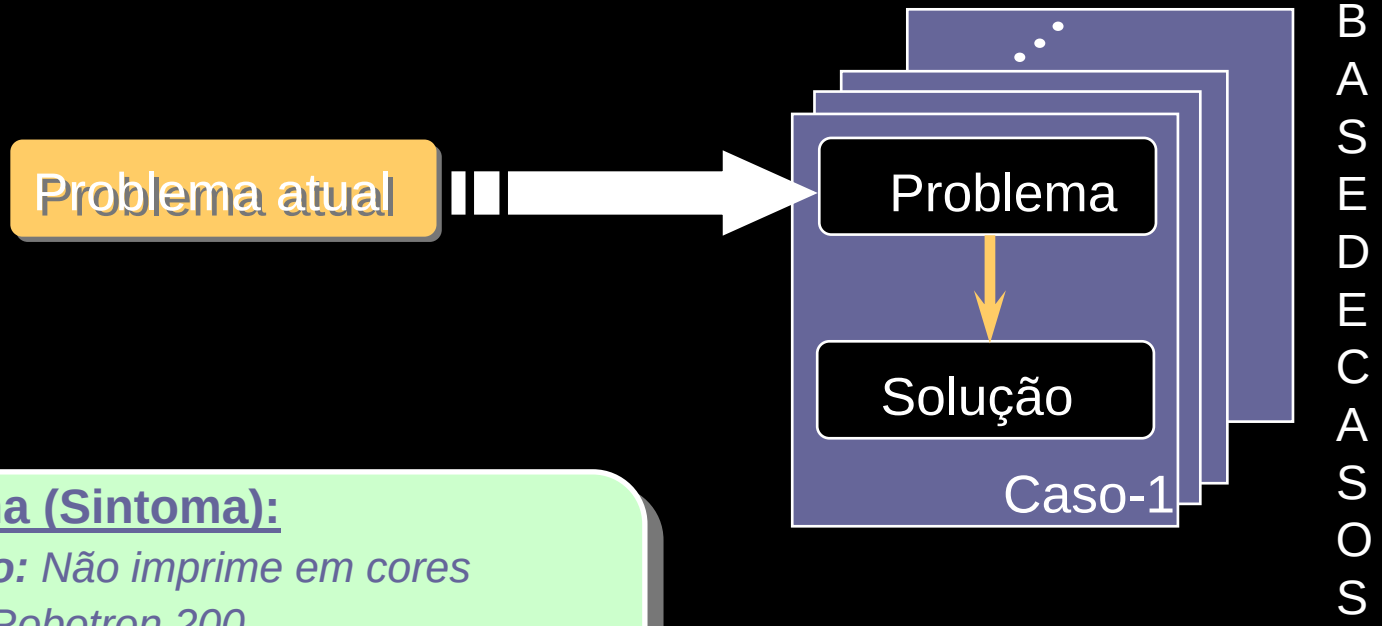
- Como buscar os casos?
- Como comparar os casos com a situação atual?
- Como determinar a similaridade de casos?
- Como identificar o(s) caso(s) mais similar(es)?

Tarefa central da recuperação

- **Dado:**
  - Base de casos BC = $\{C_1,...,C_n\}$ e uma medida de similaridade *sim*
  - Busca: Q (problema novo)
- **Queremos achar:**
  1. o caso mais similar C_i OU
  2. os *m* casos mais similares $\{C_1,...,C_m\}$ (ordenados ou sem ordem) OU
  3. todos os casos $\{C_1,...,C_m\}$ que têm pelo menos a similaridade sim_{min} com o problema novo Q
- **Processo de recuperação:**
  - 1. Descrição do problema/situação atual
  - 2. Busca na base de caso
  - 3. Comparação parcial dos casos da base com o problema atual
  - 4. Ordenação dos casos com base no valor da similaridade

Descrição da situação atual

- Depende da aplicação
- Com base na representação dos casos na base



Problema (Sintoma):
Descrição: *Não imprime em cores*
Modelo: *Robotron 200*
Luz de estado do papel: *apagada*
Luz de estado da tinta colorida: *acesa*
Luz de estado da tinta preta: *apagada*
Estado do interruptor: *não conhecido*

Semântica da similaridade

- Grau da similaridade = utilidade/reusabilidade da solução
- Não existe uma similaridade absoluta - sempre depende da meta de recuperação na aplicação específica
  - dois carros são similares quando a velocidade máxima é similar?
  - dois carros são similares quando o preço é similar?
- A **meta de recuperação** explicitamente define o objeto a ser reutilizado, a finalidade de sua reutilização, a tarefa relacionada à reutilização, o ponto de vista específico e o contexto particular.
  - P.ex.: recupere o relatório de problema para o diagnóstico relativa ao conserto de impressoras do ponto de vista do pessoal do SAC na empresa IntelliPrinters.
- Meta da modelagem da similaridade: prover uma aproximação boa
  - perto da utilidade real
  - fácil de computar

Modelar similaridade

- **Várias abordagens dependendo da representação de casos**
- **Medidas de similaridade:**
  - Funções para comparar dois casos *sim*: Caso x Caso $\rightarrow$ [0..1]
    Supõe (P_1, S_1) e (P_2, S_2) são dois casos e P o problema atual.
    Se $sim(P, P_1) \geq sim(P, P_2)$ então não preferimos a solução S_2 em cima da solução S_1 para o problema atual P.
  - Similaridades são geralmente normalizadas em uma faixa de 0 a 1, onde 0 é a dissimilaridade total e 1 a coincidência absoluta, ou através de porcentagens, onde 100% é um casamento exato.
- **Similaridade global vs. local**
  - Medidas de similaridade local: similaridade no nível de atributos
  - Medidas de similaridade global: similaridade no nível de casos
    - combinação de medidas de similaridade local
    - consideração de importância/pesos diferentes de atributos

Similaridade global: Nearest Neighbor

- Ocorrências em uma base de casos são vistas como pontos em um espaço multidimensional.
- A distância espacial entre as respectivas representações dos caso reflete a similaridade entre estes.
- A busca reduz-se à determinação do vizinho geometricamente mais próximo, após definição de uma medida de distância d.

Similaridade global: *Nearest Neighbor*

Caso 1:

Modelo: Robotron Matrix 600

Luz da tinta: vermelha ...

Caso 2:

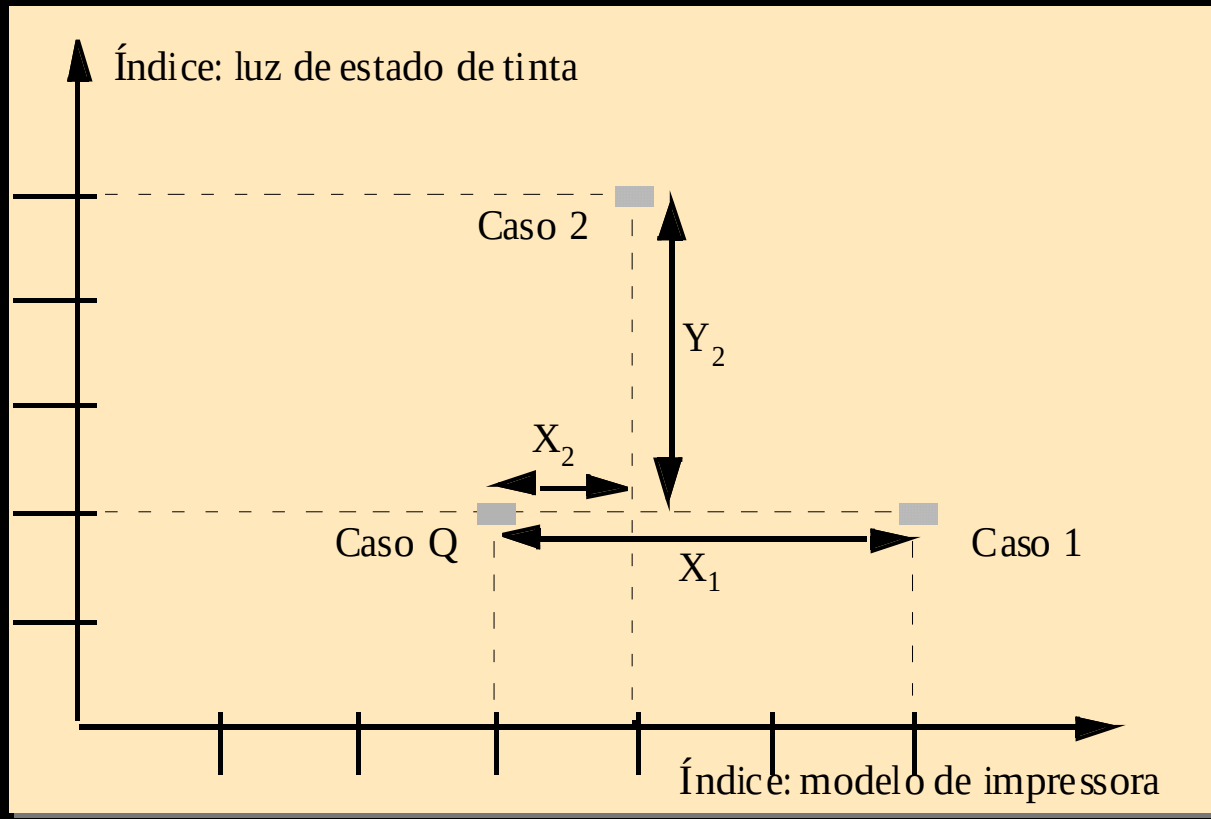
Modelo: Robotron 400

Luz da tinta: verde ...

Situação atual Q:

Modelo: Robotron 200

Luz da tinta: vermelha ...



a distância de Q ao caso 1: $d_1 = X_1 + Y_1 = 3 + 0 = 3$

a distância de Q ao caso 2: $d_2 = X_2 + Y_2 = 1 + 3 = 4$

⇨ caso 1 é o vizinho mais próximo.

Medidas de similaridade global

- Exemplo: *nearest neighbor ponderada*
- Dado duas descrições de problemas C1, C2 com os atributos y_1, ..., y_p utilizado para representação

$$SIM(C1,C2) = \sum_{j=1}^{p} w_j sim_j(C1,C2)$$

  - sim_j: Medida da similaridade local para atributo j
  - peso w_j : indica a importância do atributo j para a determinação da similaridade
  - peso $w_j = 0$ ⇨ atributo não considerado para recuperação
  - normalização é realizada com a divisão do valor de similaridade pela soma total dos pesos dos índices

Medidas de similaridade local

- Devem ser definidas em relação:
  - ao tipo específico de um atributo, e
  - no contexto de aplicação específico.

- Exemplos
  - Número
  - Símbolo binário
  - Símbolo (ordenado, não-ordenado, taxonômico)
  - *String*

Medida de similaridade local: números

- Com base na diferença dos valores: $sim_j(x,y) = f(x-y)$
- Exemplo: $sim_j(x,y) = |x-y|$
- com f em geral:
  - f: IR $\rightarrow$ [0..1] oder N $\rightarrow$ [0..1]
  - f(0) = 1 (Reflexidade)
  - f(x): decrescente monótono para x>0 e crescente monótono para x< 0
- Exemplos:
  - Função escada: só quando um caso é completamente inútil antes de uma certo grau de similaridade
  - polinominal (fator 1: linear)
- Simetria/Assimetria
  - medida simétrica: |(caso - busca)|
  - medida assimétrica (caso-busca)
    Exemplo: preço máximo

Medida de similaridade local: símbolos ordenados

- Símbolos ordenados representam valores simbólicos em uma determinada ordem.
- Exemplo: «*febre*»:{*baixa*: temperatura entre 36C e 37C, *média*: temperatura entre 37C e 38.5C, *alta*: temperatura acima de 38.5C} em ordem crescente.
- Medida de similaridade local:
  - Assinar um valor Integer a cada símbolo preservando a ordem
  - Exemplo:
    - baixa --> 1
    - média --> 2
    - alta --> 3
  - sim_j: Uso das mesmas medidas do Tipo Número
- Normalmente, a distância entre estes valores ordinais será igual, mas pode-se também definir valores não eqüidistantes.

Medida de similaridade local: símbolos não ordenados

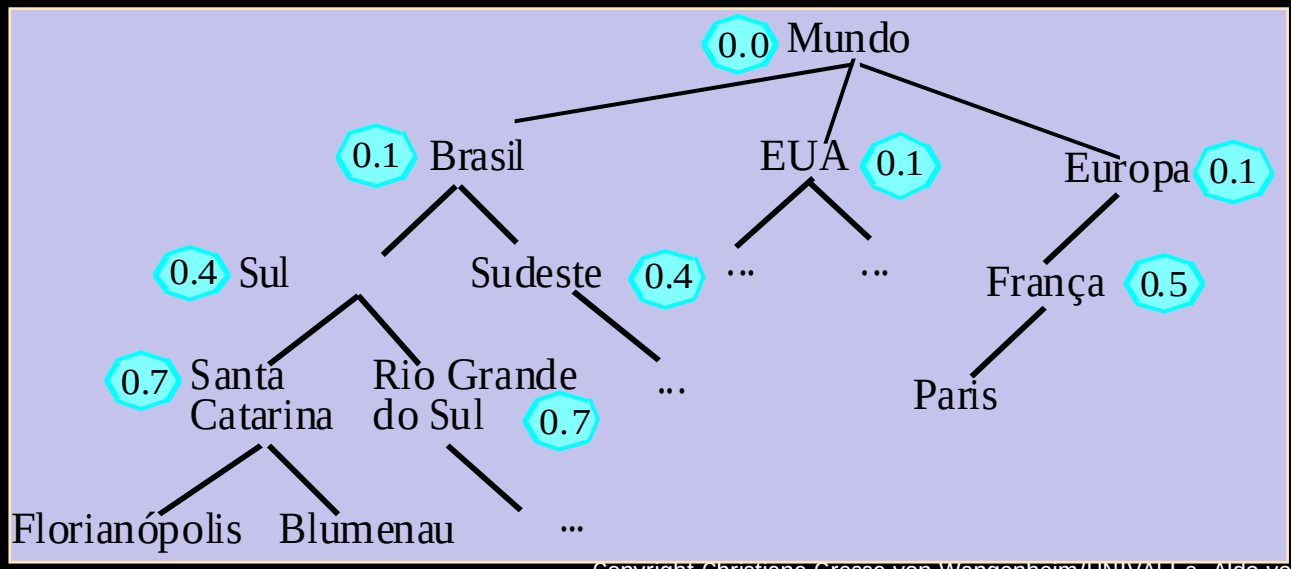
- Símbolos não ordenados representam valores sem qualquer ordem definida.
- Exemplo: lista de destinos de uma agência de viagens: *(Rio de Janeiro, Brasília, Miami, Paris ...)*
- Tabela de similaridade: $sim_j(x,y) = s[x,y]$
- Tipo do símbolo $T_A=\{v_1,...,v_k\}$

s[x,y]	v_1	v_2	...	v_k
v_1	s[1,1]	s[1,2]		s[1,k]
v_2	s[2,1]	s[2,2]		s[2,k]
...				
v_k	s[k,1]	s[k,2]		s[k,k]

- Valores na diagonal = 1
- Medidas simétricas: Triângulo da matriz de cima = Triângulo da matriz de baixo

Medida de similaridade local: taxonomias

- Uma taxonomia é uma árvore n-ária na qual os nodos representam valores simbólicos descrevendo o relacionamento entre os valores e sua posição na taxonomia.
- $sim_j(x,y)$
  - Assinar um valor de similaridade para cada nodo interno
  - Valores de similaridade crescente para os nodos sucessor
  - Similaridade entre dois nodos de folha é calculada pelo valor de similaridade do precedente comum mais próximo.



Medida de similaridade local: *strings*

- *Strings* descrevem valores de forma textual
- Exemplo «*problema da impressora*»:"*impressora não imprime em preto*".
- Calcular a similaridade entre *strings* considerando uma semântica razoável é uma tarefa extremamente difícil ⇨ substituir *strings* por valores simbólicos sempre que possível.
- **correspondência exata:** dois *strings* são similares se são escritos da mesma forma
  - P.ex. sim_j("printer","printer")=1.0; sim_j("printer","print")=0.0.
- **correção ortográfica:** compara o número de caracteres que são idênticos, ponderado pelo número total de caracteres no *string*-consulta.
  - P.ex. sim_j("printer","print")=5/7= 0.7
- **contagem de palavras:** conta o número de palavras idênticas em dois casos, normalizado por meio da divisão pelo número total de palavras no *string*-consulta.
  - sim_j("*impressora não imprime preto*","*impressora não imprime texto azul*")=4/6 = 0.67

Processo de recuperação - 1

- O processo de recuperação de casos pode ser comparado com um problema de busca massiva.
- Idealmente, a medida de similaridade é aplicada a todos os casos gerado um conjunto-resposta **completo** e **correto**.
  - **Completeza:** O método de recuperação é denominado completo, se toda relação de similaridade representada no modelo do sistema também se encontra no resultado deste método de recuperação.
  - **Correção do método de recuperação**: Um método de recuperação de casos é correto, se uma relação de similaridade definida pelo método entre um caso e o problema atual também existe no conceito de similaridade desenvolvido para a aplicação.
- Mas, pelo problema de **eficiência** em grandes bases de casos: optimização de técnicas de busca otimizadas, que não analisam toda a base de casos para cada consulta, mas apenas um subconjunto desta considerado por alguma heurística

Processo de recuperação - 2

- Várias abordagens:
  - Recuperação seqüencial
  - Recuperação de dois níveis
  - Recuperação usando árvores k-d
  - Recuperação usando redes
  - ...
- Abordagem depende do tamanho da base e da representação dos casos
- Organização da base de casos:
  - listas lineares (somente para bases pequenas)
  - estruturas indexadas para grandes bases:
    - *kd-trees*, redes de recuperação, etc.

Recuperação seqüencial

Estrutura de dados

```
TIPOS:
  TipoCaso = ...
  SimCaso = REGISTRO
        case: TipoCaso;
        similaridade: [0..1]
                        FIM;
VARIAVEIS:
  ListaCasoSim: VETOR [1..m] DE SimCaso
  CaseBase: ARRAY [1..n] DE TipoCaso  (* base de casos *)
  Consulta: TipoCaso
```

Algoritmo

```
FUNÇÃO SelecRel(CaseBase,Consulta,m): ListaCasoSim
INÍCIO
  ListaCasoSim[1..m].similaridade := 0
  PARA i:=1 TO n FAÇA
    SE sim(Consulta,CaseBase[i])>ListaCasoSim[m].similaridade
      ENTÃO insira CaseBase[i] em ListaCasoSim
  RETORNE ListaCasoSim
FIM
```

Características da recuperação seqüencial

- Complexidade: O(n)
- O processo é **completo** e **correto**, como o conceito de similaridade representado no sistema é aplicado de forma seqüencial a todos os exemplos de casos da base de casos:
- Desvantagens:
  - Lento, se a base é muito grande
  - Esforço da recuperação é independente da busca
  - Esforço da recuperação é independente do número dos casos a serem recuperados (m)
- Vantagens:
  - Implementação simples
  - Nenhuma estrutura de indexação necessária
  - Qualquer medida de similaridade pode ser utilizada

Recuperação utilizando Árvores k-d

- Estruturas de indexação que nos permite indexar um caso por 30 ou mais chaves secundárias e nenhuma chave primária
  - Técnicas mais comuns como árvores-B, árvores-B+ ou tabelas de hash são inadequadas para uma aplicação assim.
  - Uma das estruturas de indexação que mais tem sido utilizada é a Árvore k-d.
- Uma árvore k-d é:
  - Árvore binária para indexação multichave. Também chamada Árvore de Pesquisa Binária Multidimensional.
  - Foi "redescoberta" pela Inteligência Artificial no início da década de 1990 como mecanismo de indexação de casos.
  - Sistemas de RBC muito conhecidos como PATDEX (PATtern Directed EXpert System - o antecessor do CBR-Works e uma componente da bancada de IA MOLTKE) e INRECA utilizaram esta técnica.

Estrutura da Árvore k-d

- Uma árvore k-d é uma árvore onde k é o número de chaves para cada registro.
  - Assim, chamamos uma árvore com três chaves de árvore 3-d (tridimensional), etc.
- Cada registro em uma árvore k-d possui:
  - dados e ponteiros, como qualquer árvore,
  - um conjunto ordenado de **k** valores de chaves (v_0,...,v_{k-1}).
- Associado a cada nodo P está um discriminador DISC(P), que é utilizado para especificar qual chave da k-tupla v_0,...,v_{k-1} será utilizada neste nodo para tomar uma decisão de ramificação.
  - Geralmente o discriminador é função da profundidade do nodo (resto da divisão da profundidade pelo número de chaves).
- O filho à esquerda é chamado **`loson(P)`** e o à direita **`hison(P)`**.

Funcionamento da Árvore k-d

- Caminhamento e Inserção (nas folhas) são realizados seguindo-se a seguinte regra de decisão (dados Q = chaves do registro procurado ou registro a ser incluído e P = nodo):

SE $K_j(Q) < K_j(P)$ ENTÃO
o registro Q está em loson(P)

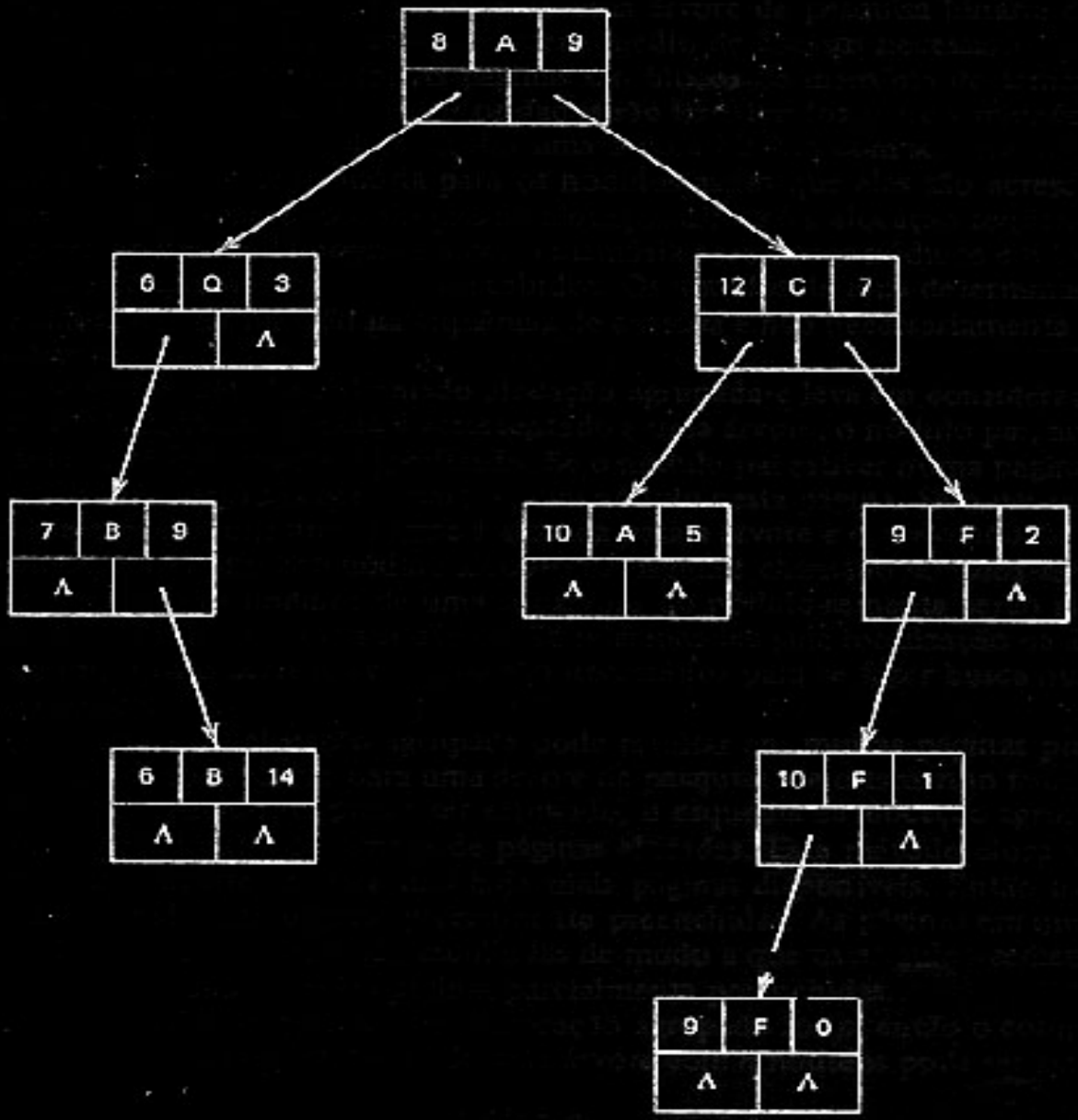
SE $K_j(Q) > K_j(P)$ ENTÃO
o registro Q está em hison(P)

- Se dois valores de chave são iguais, a decisão é tomada com base nos demais valores de chave na seguinte ordem (superchave):

$$S_j(P) = K_j(P), K_{j+1}(P), \ldots, K_{k-1}(P), K_0(P), \ldots, K_{j-1}(P)$$

- Adiante um exemplo de inserção.

8 A 9, 6 Q 3, 12 C 7, 7 B 9, 6 B 14,
10 A 5, 9 F 2, 10 F 1, e 9 F 0



Vantagens das Árvores k-d

- Árvores k-d podem ser utilizadas diretamente para todos os 3 tipos de pesquisa: simples, com limites e booleana.
- O tempo médio de acesso a um registro não é pior do que o da árvore binária. Todas as características de tempo de pesquisa, complexidade, etc da árvore binária valem, no que diz respeito à pesquisa, também para a árvore k-d: $O(1{,}4 \log_2 n)$.
- Inserção (não balanceada) de um nodo requer também tempo O(log2 n).
- Flexibilidade: Aplicável a qualquer tipo de aplicação onde se queira fazer recuperação de chaves secundárias ou recuperação multichaves.

Desvantagens das Árvores k-d

- Gera árvores de profundidade extremamente grande.
  - Solução: Pode ser resolvido através da paginação.
- Inserção balanceada é extremamente cara.
- Rebalanceamento (apos várias inserções ou exclusões) também extremamente caro.

Atividade curricular: Modelagem da recuperação

- Revisão da descrição do problema:
  - todos os atributos são relevantes para a recuperação de casos similares?
  - A descrição inclui todos os atributos relevantes?
- Definição das medidas de similaridade local para cada tipo de um atributo utilizado na recuperação
- Definição da medida de similaridade global (inclusive definição do peso para cada atributo)