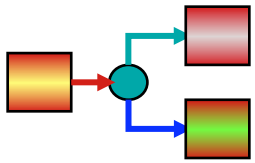


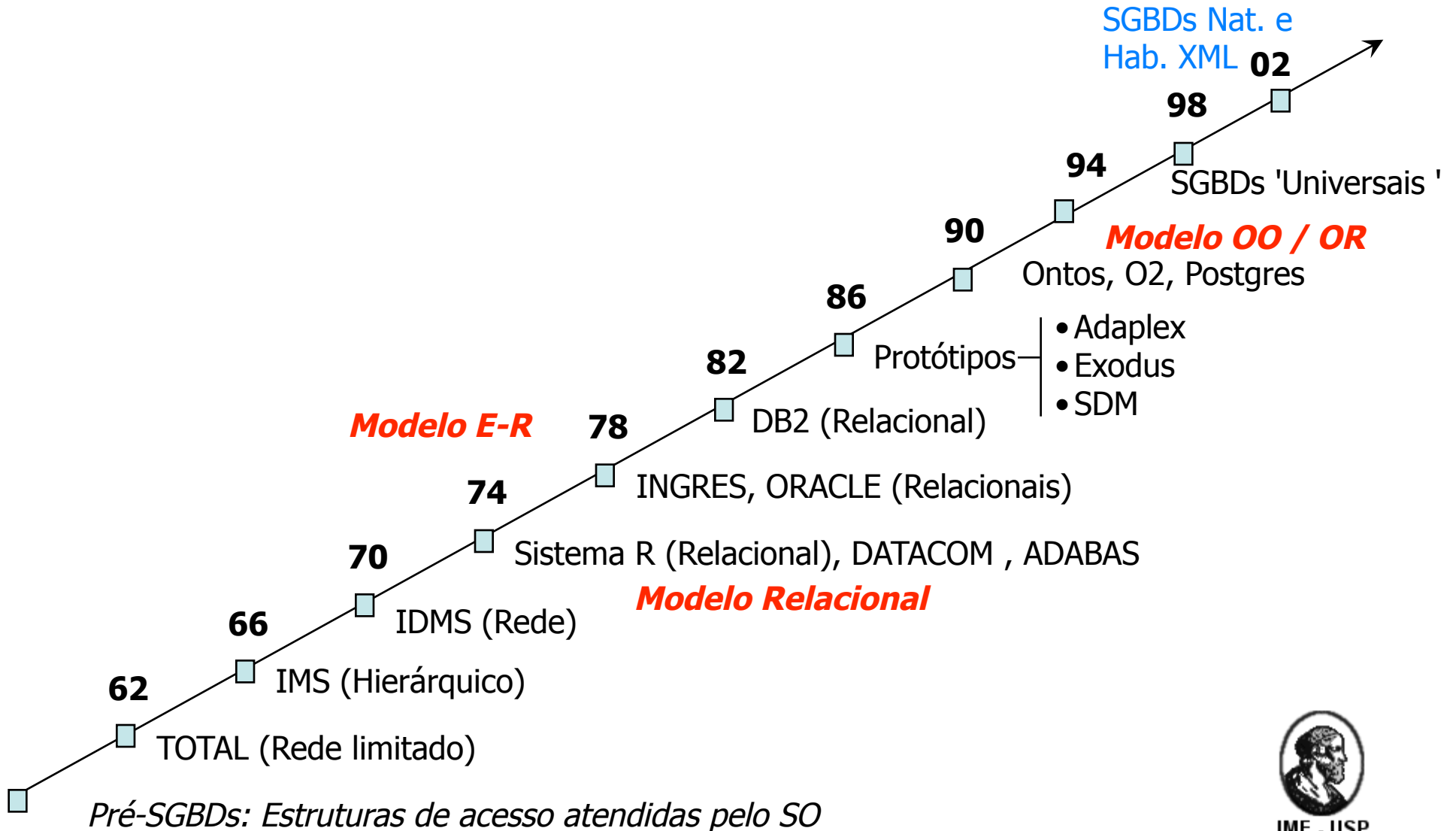
Soluções clássicas, seus limites e novos desafios em Bancos de Dados

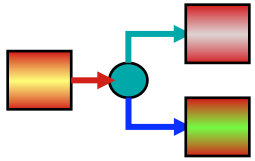
João Eduardo Ferreira (jef@ime.usp.br)



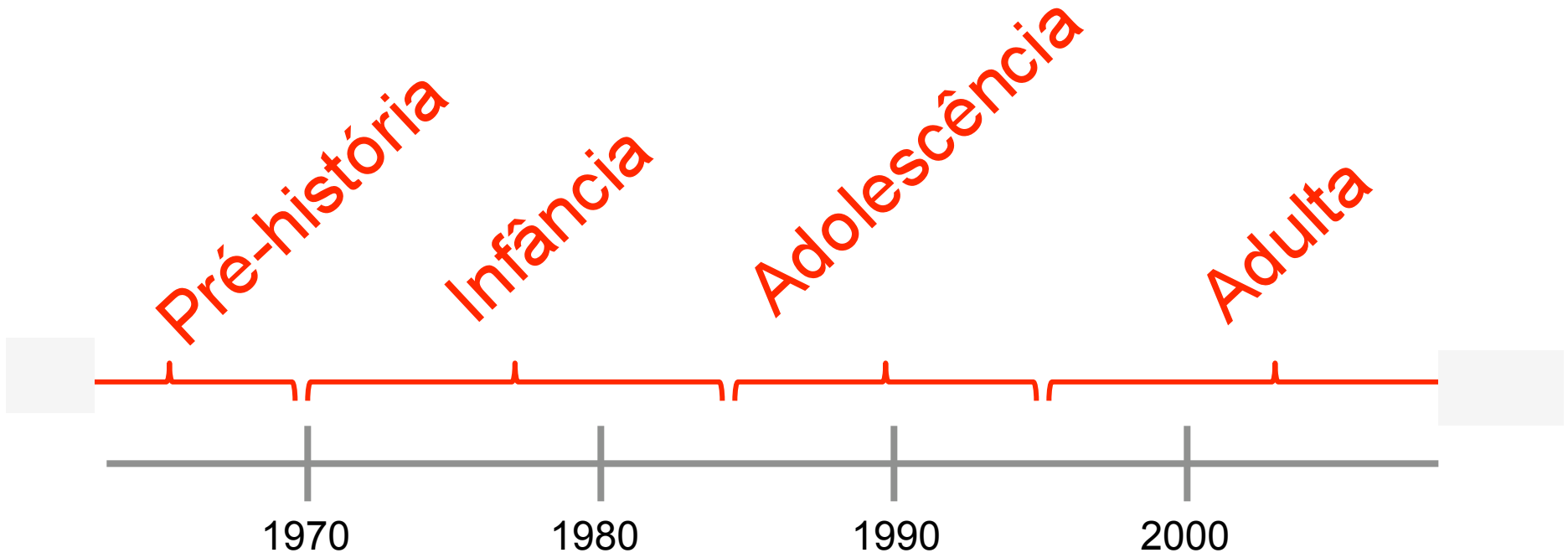


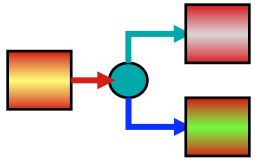
Evolução dos Modelos e SGBDs





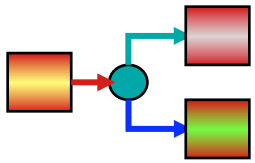
Linha de tempo de BD





Pré-história (antes de 1970)

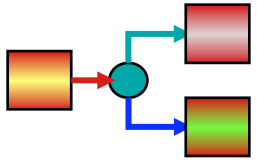
- Limitações de **hardware** →
 - Limitações de **software**
- Baixíssimo nível de abstração
- Sistemas de gerência de arquivos



Como era a área (70-85)

- Muitos **recursos** para pesquisa em BD.

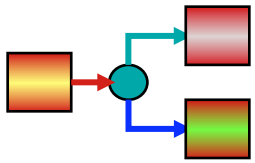




Como era a área (70-85)

- Muitos **recursos** para pesquisa em BD.
- Muitos **grupos** de BD espalhados pelo mundo.





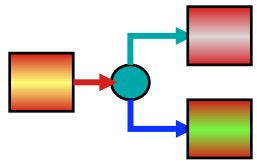
Como era a área (70-85)

- Principal resultado:
 - Modelo de dados **relacional** de Codd (IBM)
 - Técnicas para implementá-lo

PubID	Publisher	PubAddress
03-4472822	Random House	123 4th Street, New York
04-7733903	Wiley and Sons	45 Lincoln Blvd, Chicago
03-4859223	O'Reilly Press	77 Boston Ave, Cambridge
03-3920886	City Lights Books	99 Market, San Francisco

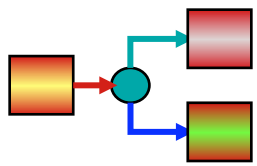
AuthorID	AuthorName	AuthorBDay
345-28-2938	Haile Selassie	14-Aug-92
392-48-9965	Joe Blow	14-Mar-15
454-22-4012	Sally Hemmings	12-Sept-70
663-59-1254	Hannah Arendt	12-Mar-06

ISBN	AuthorID	PubID	Date	Title
1-34532-482-1	345-28-2938	03-4472822	1990	Cold Fusion for Dummies
1-38482-995-1	392-48-9965	04-7733903	1985	Macrame and Straw Tying
2-35921-499-4	454-22-4012	03-4859223	1952	Fluid Dynamics of Aquaducts
1-38278-293-4	663-59-1254	03-3920886	1967	Beads, Baskets & Revolution



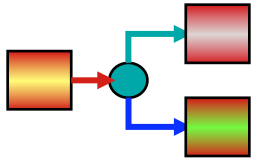
Principais resultados da pesquisa (70 a 85)

- Teoria de banco de dados
 - resultou na **formalização** do modelo relacional
- SQL
 - linguagem **declarativa** de programação



Principais resultados da pesquisa (70 a 85)

- **Otimização de consultas**
 - primeira abordagem realista de **programação automática**
- **Gerência de transações**
 - algoritmos eficientes para **concorrência** e **reconstrução** apoiando grandes bases e grande número de usuários



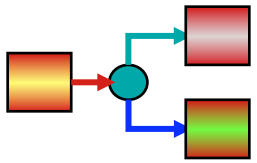
Primeiros protótipos e produtos

- **System R** (IBM)



- **INGRES** (Berkeley)





Primeiros protótipos e produtos

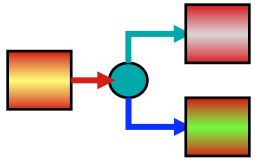
- **System R** (IBM)



ORACLE®

- **INGRES** (Berkeley)





Primeiros protótipos e produtos

- **System R** (IBM)



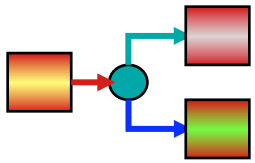
ORACLE®

- **INGRES** (Berkeley)

Informix
SOFTWARE
The database company™

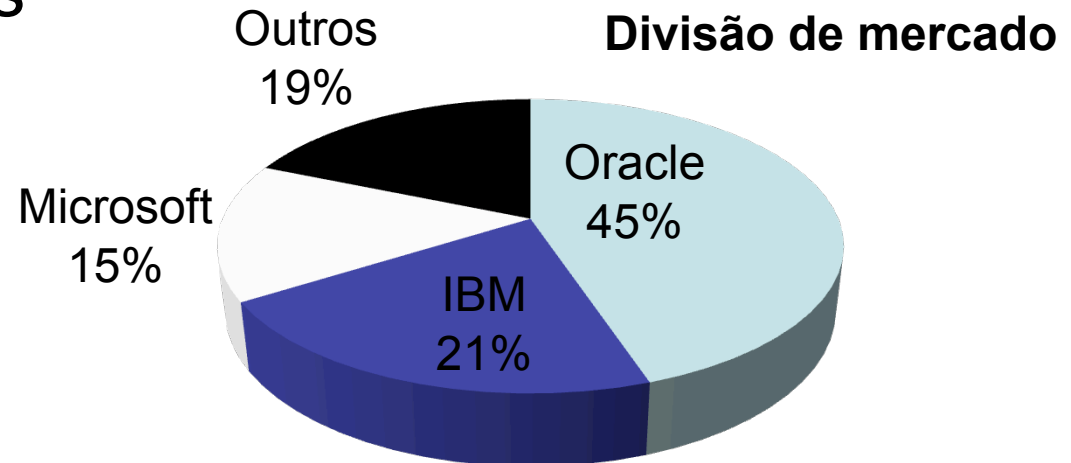


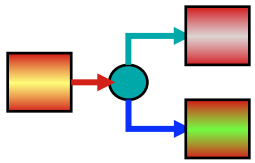
Microsoft®
SQL Server 2008



Indústria de software

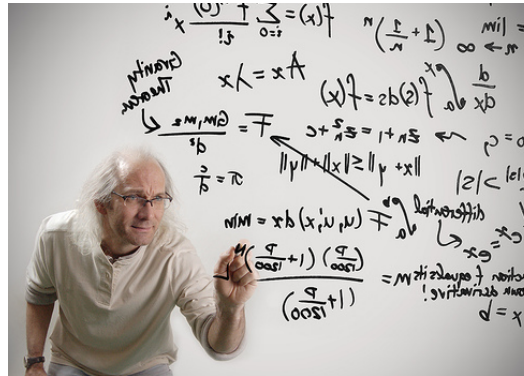
- Pesquisas na área de BD resultaram em uma importante indústria de software.
- 2007 (IDC):
 - US\$ 19 bilhões





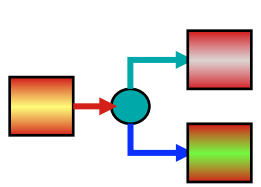
Perfil dos pesquisadores

- *Theory guys*



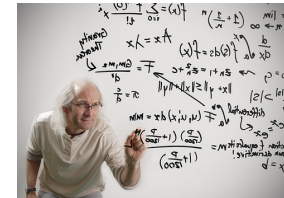
- *System guys*



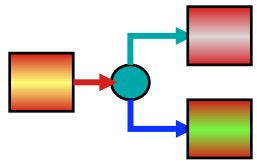


O que era necessário saber para pesquisar na área

- *Theory guys*



- Uma boa base de **lógica matemática**
- Muitos pesquisadores de teoria de BD oriundos da área de **teoria da computação**

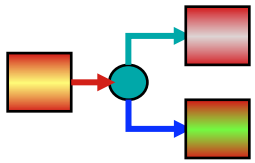


O que era necessário saber para pesquisar na área

- *System guys*



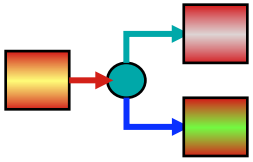
- Sistemas operacionais
- Rudimentos de **arquitetura de computadores** (principalmente, dispositivos de **armazenamento**)
- **Linguagens** de programação e **compiladores** (?)
- Base (tênue) de **matemática discreta**



Áreas de pesquisa compartimentadas

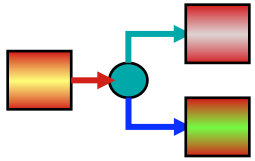
- Áreas de pesquisa eram bem **compartimentadas**
- **Pouca cooperação** entre pesquisadores de áreas diferentes
- **Mesmo** ocorria **em outras áreas** clássicas da Computação (Redes, Linguagens de programação, Sistemas distribuídos, . . .).





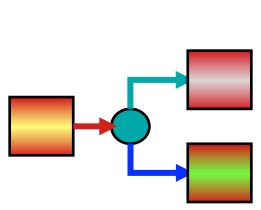
85-95 - Aplicações

- Software de banco de dados para aplicações “**não-convencionais**”:
 - aplicações de projeto (engenharia),
 - bases de dados espaciais,
 - aplicações de tempo real,
 -



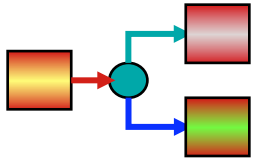
85-95 – Para atender aplicações...

- Software de banco de dados **dedutivo**
- Software de banco de dados **orientado a objetos**
 - Vários produtos no mercado, desde o fim da década de 80
- **Fracasso retumbante** de mercado:
 - Em 2004, todos produtos chamados **pós-relacionais** tinham ao redor de **1% do mercado**
 - Sem perspectivas de crescimento



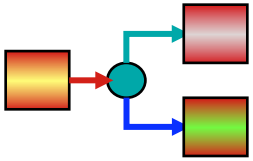
O que era necessário saber para pesquisar na área

- *Theory guys*
 - Uma boa base de **lógica matemática**
 - Teoria de **BD**
- *System guys*
 - O **mesmo que antes** + conhecimentos de **BD** relacional



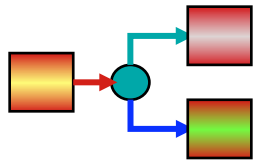
E no mercado de SGBD?

- Mercado de SGBDs **parou de crescer**.
- Grandes usuários de SGBD já o adquiriram: **sistemas legados**.
- Novos usuários:
 - Grande crescimento de bases de dados baseadas em **software livre** (MySQL, PostgreSQL).
- Oracle e Microsoft já distribuem **versões gratuitas**.
- Novo foco das divisões de banco de dados de IBM e Oracle: **Integração de informações**.



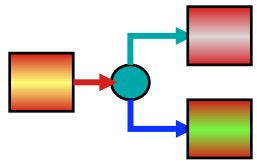
SGBD onipresente

- Originalmente:
 - SGBD era utilizado em **sistemas de informação gerenciais**
- Hoje:
 - SGBD tornou-se **sistema de arquivos de alto-nível**
 - Aparece em **todo tipo** de aplicações



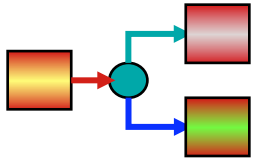
Papéis no mercado (extremos)

- Função: **operário** da fábrica de sistemas de informação
 - Tarefas repetitivas
 - Uso de ferramentas existentes
 - Pouca criatividade
- EUA prevê **declínio no seu mercado de trabalho** de TI nesta área:
 - Migração para países em desenvolvimento
 - Softwares que automatizam o desenvolvimento
 - Até quanto vai?



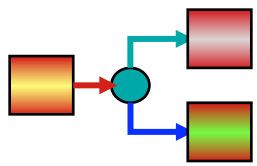
Papéis no mercado (extremos)

- “Engenheiro” de dados
 - Aplicações complexas envolvendo grandes massas de dados
 - Web
 - Data-center



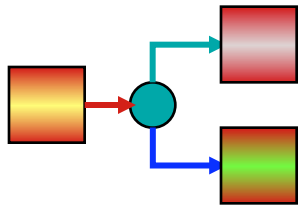
O que é necessário saber para trabalhar e pesquisar na área

- **Tudo que já era necessário** conhecer anteriormente:
 - Uma boa base de Lógica Matemática
 - Teoria de BD
 - Sistemas operacionais
 - Compiladores
 - Base de matemática discreta
 - Rudimentos de arquitetura de computadores (principalmente, dispositivos de armazenamento)
 - Conhecimentos de BD relacional (linguagens, processamento de consultas, processamento de transações)



O que é necessário saber para trabalhar e pesquisar na área

- ..., e **muito mais**:
 - Uma base mais ampla de Matemática (volumes de dados envolvidos = **métodos aproximados**):
 - álgebra, cálculo, **estatística**...
 - **Algorítmica** (foco não é mais somente o número de acessos a disco);
 - **Paralelismo**;
 - **Recuperação de informações**;
 - **Aprendizagem de máquina**, . . .



Nosso Ponto de Partida

- Dado Semi-Estruturado

Um objeto OEM é uma quádrupla
(label, oid, type, value)

- label: cadeia de caracteres

- oid: identificador do objeto

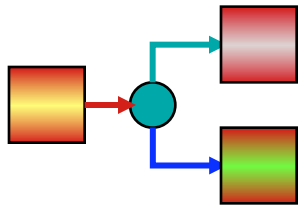
- type: *complexo* ou *atômico*(*int*, *string*, *gif*, *jpeg* ...)

- value:

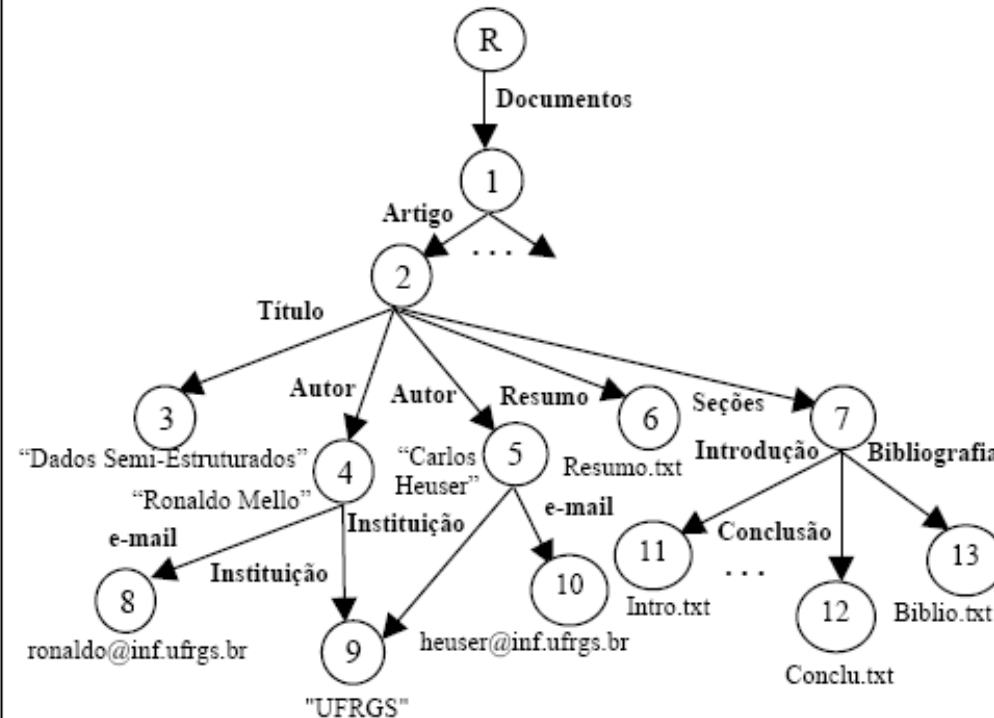
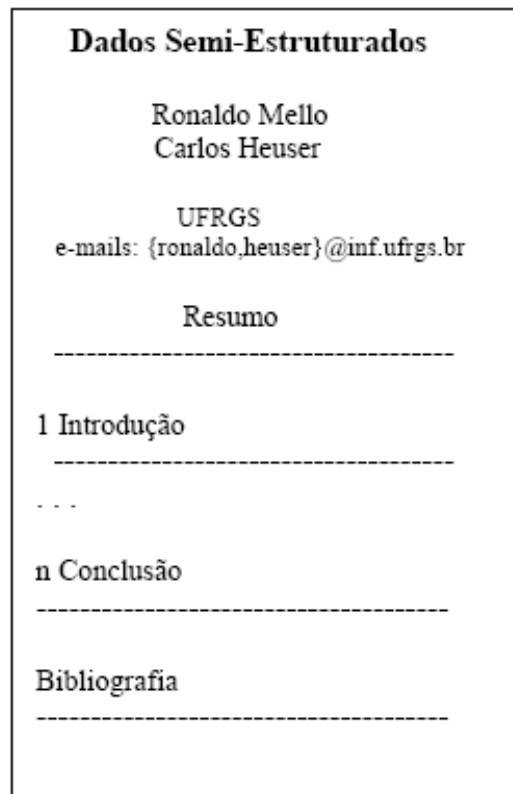
- se o tipo é complexo, conjunto de oid 's

- senão um valor atômico





Nosso Ponto de Partida



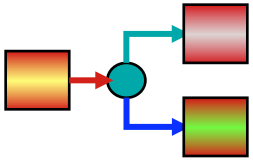
(a)

(b)

Figura 2.1 - Exemplo de um objeto semi-estruturado (a) e sua representação em um modelo de dados baseado em grafo (b)

Fonte: Mello et al: <http://www.dcc.ufrj.br/~braganholo/artigos/tutorial.pdf>- 2008

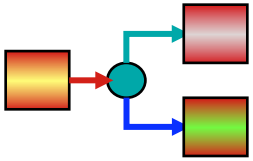




Modelo de Dados

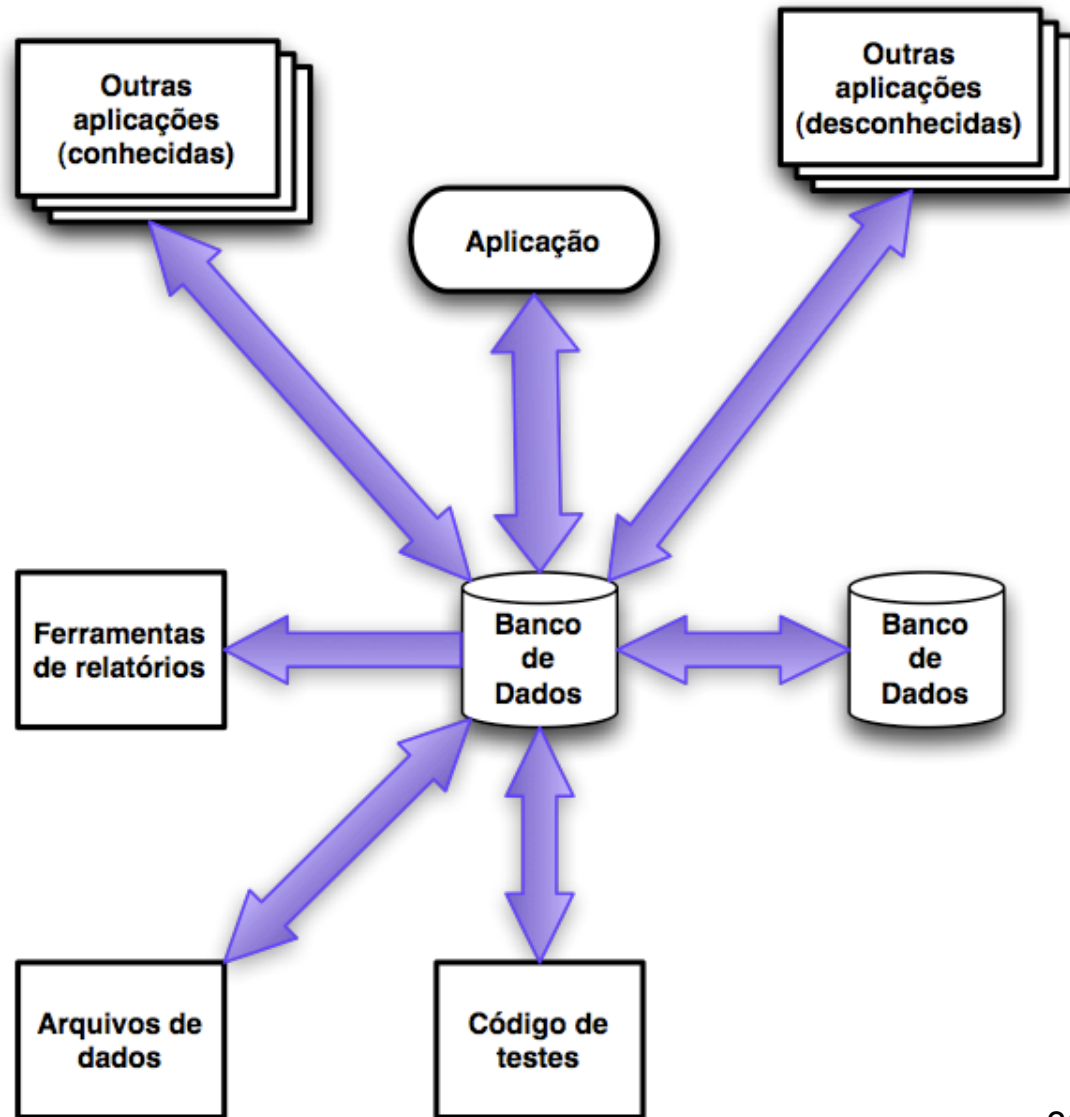
Refatoração de BD

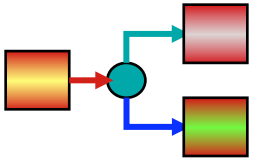
- 48 refatorações definidas por Scott Ambler
- Não acrescenta funcionalidade
- Melhora o modelo
- Preserva a **semântica informacional**



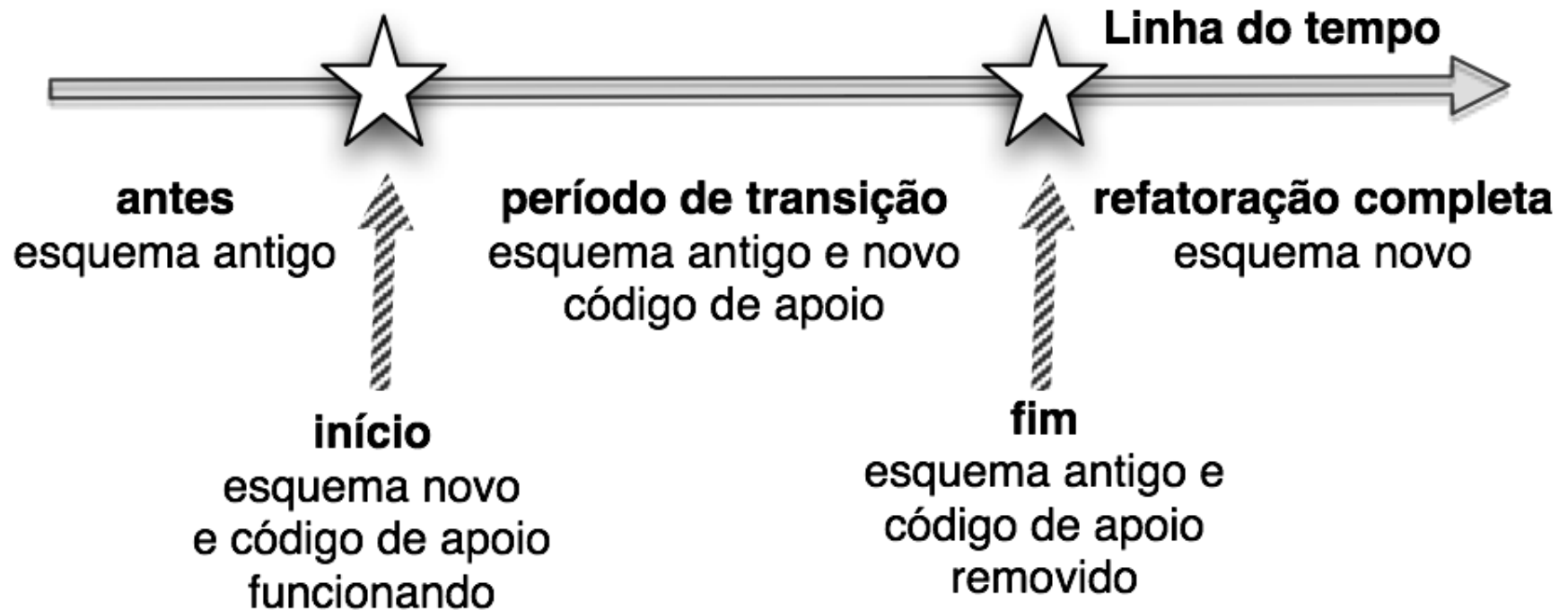
Modelo de Dados

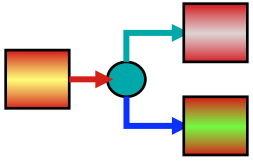
- Se preocupa com uma arquitetura complexa





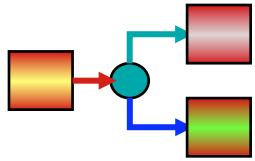
Modelo de Dados





Modelo de Dados (triggers)

- Codificação específica para cada refatoração
- Código para evitar circularidade
- Transação lenta, caso muito processamento
- Possível erro desconhecido para a aplicação



Exemplo de refatoração

- Dividir tabela

Antes

Paciente
Id
Nome
Profissão
Salário
Paga Aluguel?

Durante

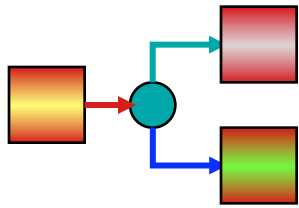
Paciente	Dados Sócio Econômicos
Id	paciente_id
Nome	Profissão
Profissão	Salário
Salário	Paga aluguel?
Paga Aluguel?	

Depois

Paciente	Dados Sócio Econômicos
Id	paciente_id
Nome	Profissão
	Salário
	Paga aluguel?

Período de transição

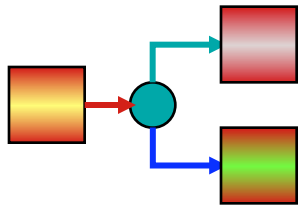
- esquema antigo obsoleto
- esquema novo disponível



Transações

- **Propriedades de uma transação clássica**
- **Atomicidade**
- **Consistência**
- **Isolamento**
- **Durabilidade**





Nosso Ponto de Partida

Definição 2.3.1 Uma *ordem parcial (irreflexiva)* $<$ sobre um conjunto Σ é uma relação binária em $\Sigma \times \Sigma$ (ou seja, é um conjunto de pares ordenados), representada por $(\Sigma, <)$, satisfazendo as seguintes restrições:

- A ordem $<$ é transitiva: para todo $x, y, z \in \Sigma$ se $x < y$ e $y < z$ então $x < z$.
- A ordem $<$ é irreflexiva: para todo $x \in \Sigma$, $x \not< x$.

O exemplo da Figura 2.4 ainda nos mostra que uma ordem parcial pode ser vista como um grafo dirigido; e, mais do que isso, como a ordem parcial é irreflexiva, concluímos que o grafo dirigido deve ser *acíclico*. A representação da Figura 2.4 é um grafo dirigido acíclico.

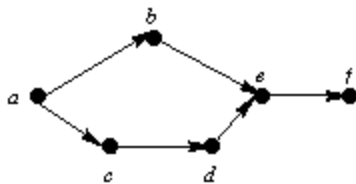
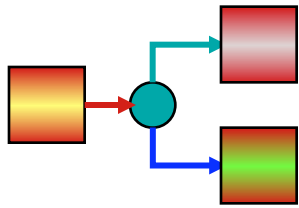


Figura 2.4: Ordem parcial

Fonte Ferreira & Finger 2000 : <http://www.ime.usp.br/~jef/ec2000.ps>





Nosso Ponto de Partida

Definição 2.3.2 Uma *transação* T_i é uma ordem parcial com ordenação $<_i$ tal que:

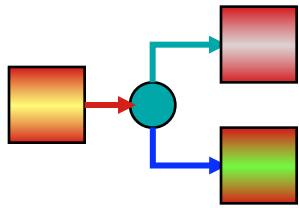
1. $T_i \subseteq \{r_i[x], w_i[x] | x \text{ é um dado}\} \cup \{a_i, c_i\}$.

Ou seja, T_i é um conjunto finito composto apenas de operações de leitura, operações de escrita, e operações de terminação, a_i ou c_i .

2. T_i possui uma e apenas uma operação de terminação: $c_i \in T_i \iff a_i \notin T_i$.

3. A operação de terminação t_i é a última operação da transação: $p_i <_i t_i$ para toda operação p_i , $t_i = a_i$ ou $t_i = c_i$.

4. Se $r_i[x], w_i[x] \in T_i$, então $r_i[x] <_i w_i[x]$ ou $w_i[x] <_i r_i[x]$ (operações sobre o mesmo dado devem estar ordenadas).



Nosso Ponto de Partida

Definição 2.3.2 Uma *transação* T_i é uma ordem parcial com ordenação $<_i$ tal que:

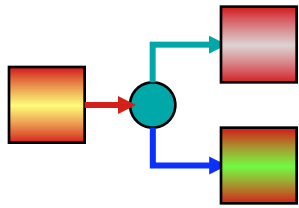
1. $T_i \subseteq \{r_i[x], w_i[x] | x \text{ é um dado}\} \cup \{a_i, c_i\}$.

Ou seja, T_i é um conjunto finito composto apenas de operações de leitura, operações de escrita, e operações de terminação, a_i ou c_i .

2. T_i possui uma e apenas uma operação de terminação: $c_i \in T_i \iff a_i \notin T_i$.

3. A operação de terminação t_i é a última operação da transação: $p_i <_i t_i$ para toda operação p_i , $t_i = a_i$ ou $t_i = c_i$.

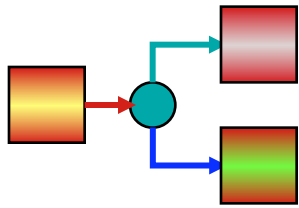
4. Se $r_i[x], w_i[x] \in T_i$, então $r_i[x] <_i w_i[x]$ ou $w_i[x] <_i r_i[x]$ (operações sobre o mesmo dado devem estar ordenadas).



Nosso Ponto de Partida

Definição 2.3.3 Um *escalonamento completo* sobre o conjunto de transações $\{T_1, T_2, \dots, T_n\}$ é uma ordem parcial $(E, <_E)$ tal que:

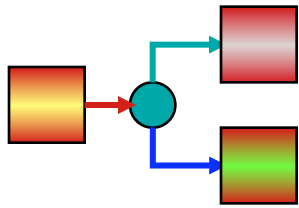
- $E = T_1 \cup T_2 \cup \dots \cup T_n$;
- $<_E \supseteq <_1 \cup <_2 \cup \dots \cup <_n$; ou seja, a ordenação do escalonamento estende a ordenação das operações internas a cada transação;
- para cada par de operações conflitantes p_i e q_j , $p_i <_E q_j$ ou $q_j <_E p_i$.



Nosso Ponto de Partida

Definição 2.3.3 Um *escalonamento completo* sobre o conjunto de transações $\{T_1, T_2, \dots, T_n\}$ é uma ordem parcial $(E, <_E)$ tal que:

- $E = T_1 \cup T_2 \cup \dots \cup T_n$;
- $<_E \supseteq <_1 \cup <_2 \cup \dots \cup <_n$; ou seja, a ordenação do escalonamento estende a ordenação das operações internas a cada transação;
- para cada par de operações conflitantes p_i e q_j , $p_i <_E q_j$ ou $q_j <_E p_i$.

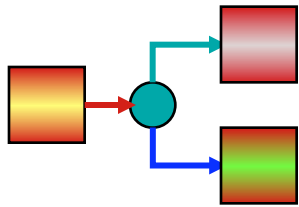


Nosso Ponto de Partida

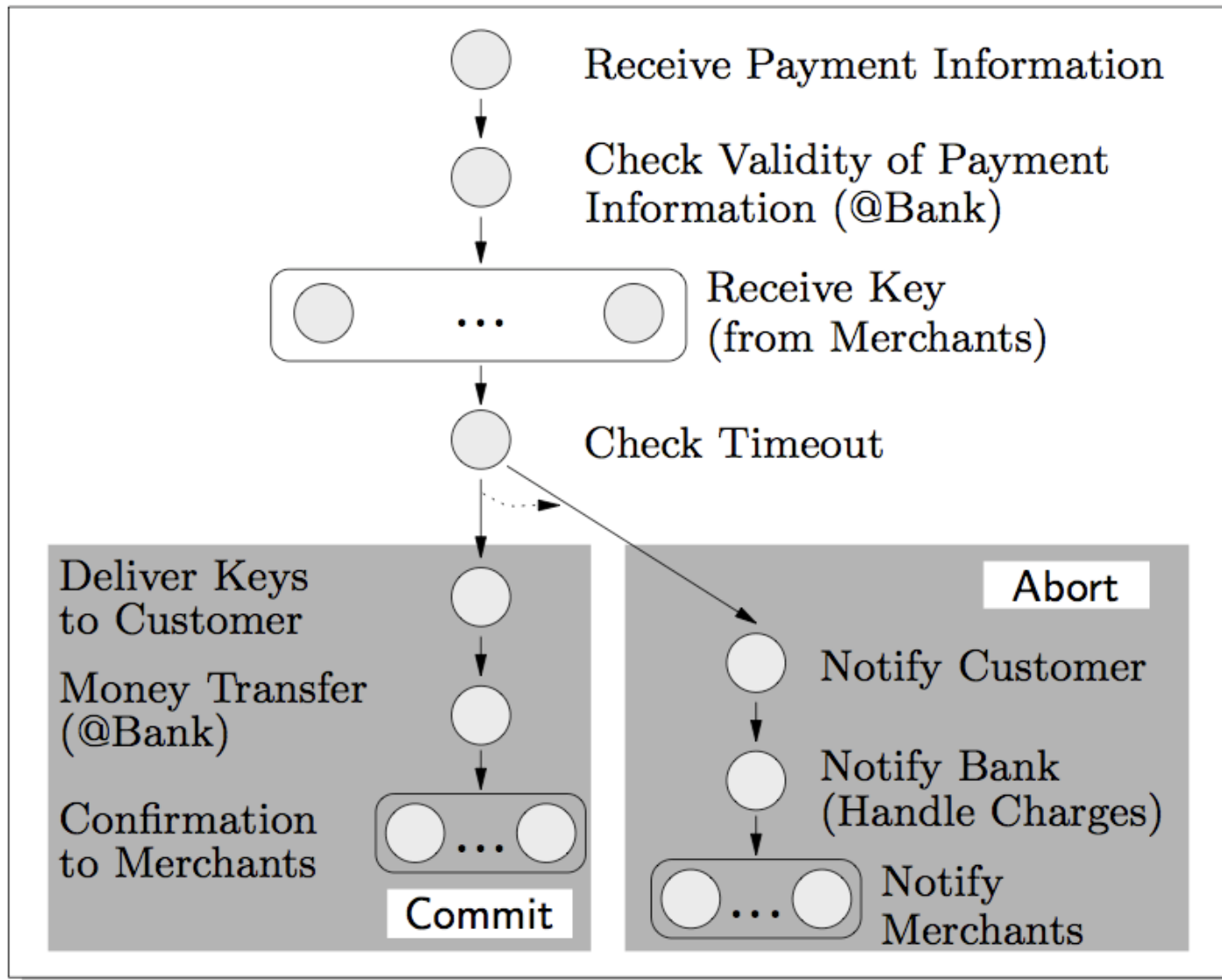
Felizmente existe uma forma bem fácil de se determinar se um dado escalonamento E é seriável ou não, que depende da construção do *grafo de seriação* para o escalonamento $GS(E)$. O grafo $GS(E)$ é um grafo dirigido construído da seguinte maneira:

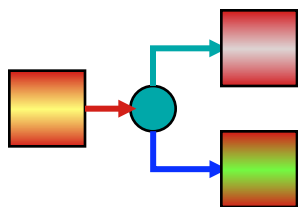
- os nós de $GS(E)$ são as transações confirmadas de E ;
- para cada par de operações conflitantes p_i e q_j tal que $p_i <_E q_j$, adicionamos a aresta $T_i \rightarrow T_j$ a $GS(E)$.

Teorema 2.3.7 (Condição de Seriação) *Um escalonamento E é seriável se e somente se $GS(E)$ é acíclico.*

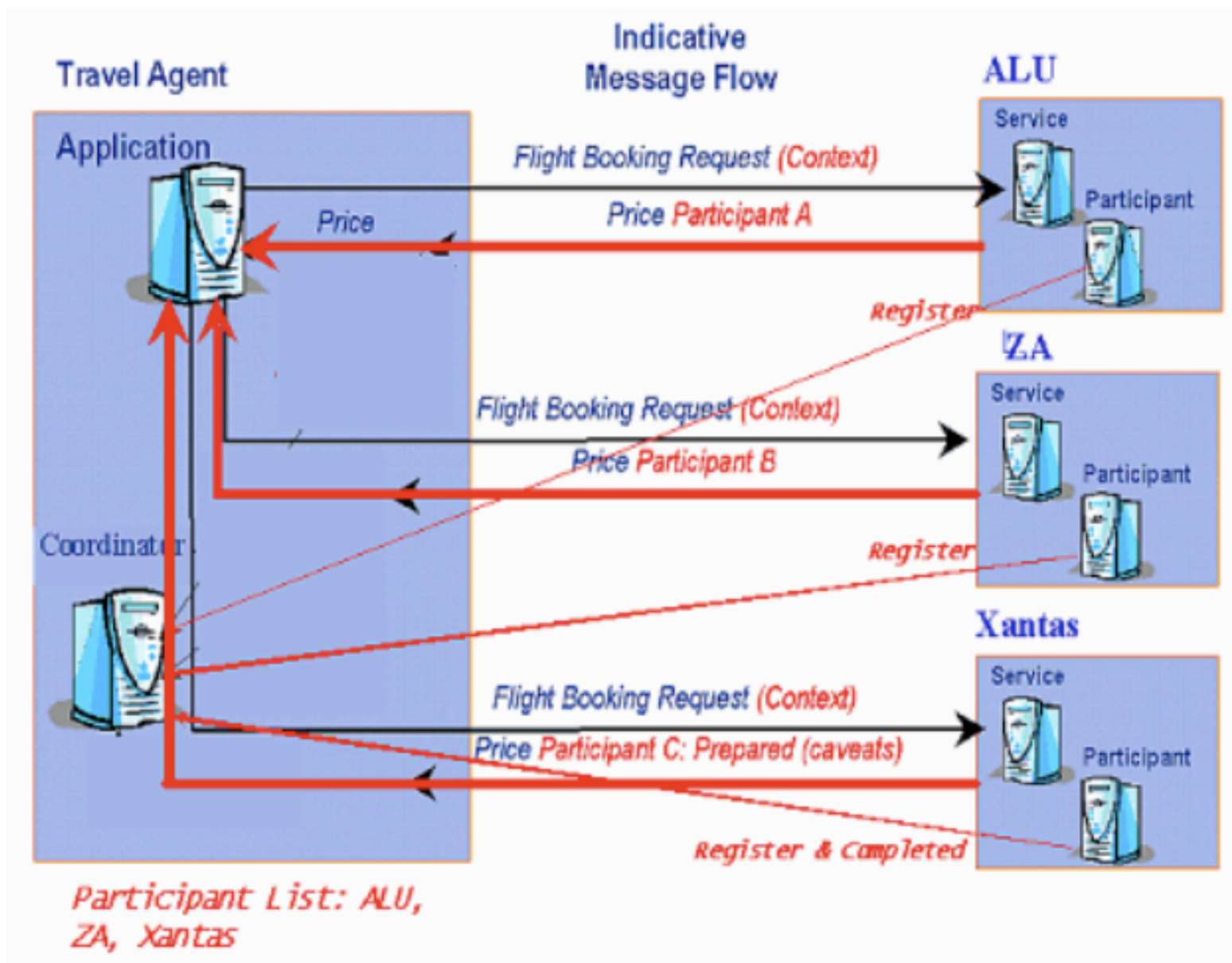


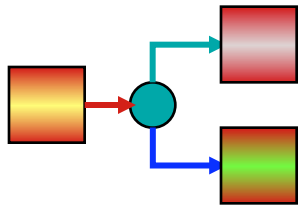
Limites das Transações





Limites das Transações

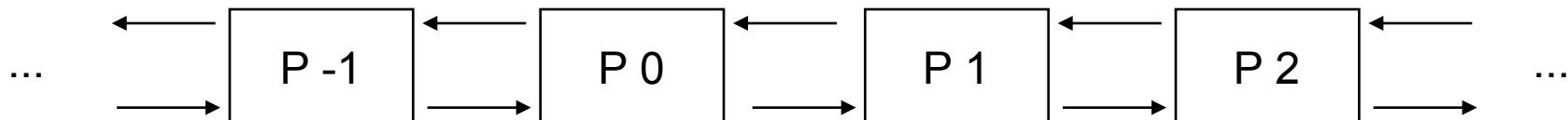


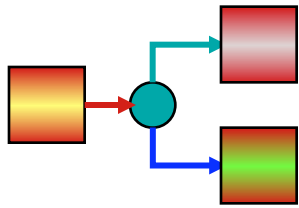


Nosso Ponto de Partida

Álgebra de Processos

- Comportamento de Sistemas: processos e dados;
- Processos: mecanismos de controle para manipulação de dados;
- Vários processos executando concorrentemente;





Nosso Ponto de Partida

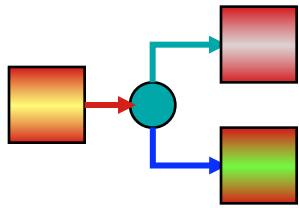
Utilização

- . Representação textual;
- . Semântica formal;
- . Mecanismos de análise.

Componentes básicos

- . Conjunto de ações;
- . Operador de composição seqüencial (“.”): $(a . b)$;
- . Operador de composição alternativa (“+”): $(a + b)$;
- . Axiomas de equivalência de processos;
- . Regras de transição.





Nosso Ponto de Partida

Axiomas de equivalência

$$x + y = y + x ;$$

$$x + x = x ;$$

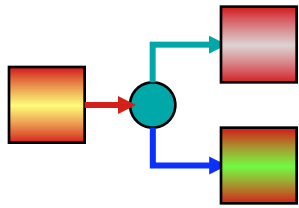
$$(x . y) . z = x . (y . z)$$

Processos bissimilares

- . Devem executar exatamente as mesmas cadeias ações;
- . Possuem a mesma estrutura de ramificação;

$z.(x + y)$ e $z.x + z.y$ não são bissimilares.





Nosso Ponto de Partida

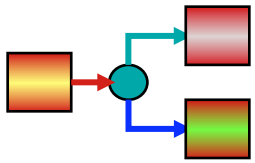
ACP

- Novos operadores foram definidos;
- Operador entrelaçamento (“||”): $(a||b)$

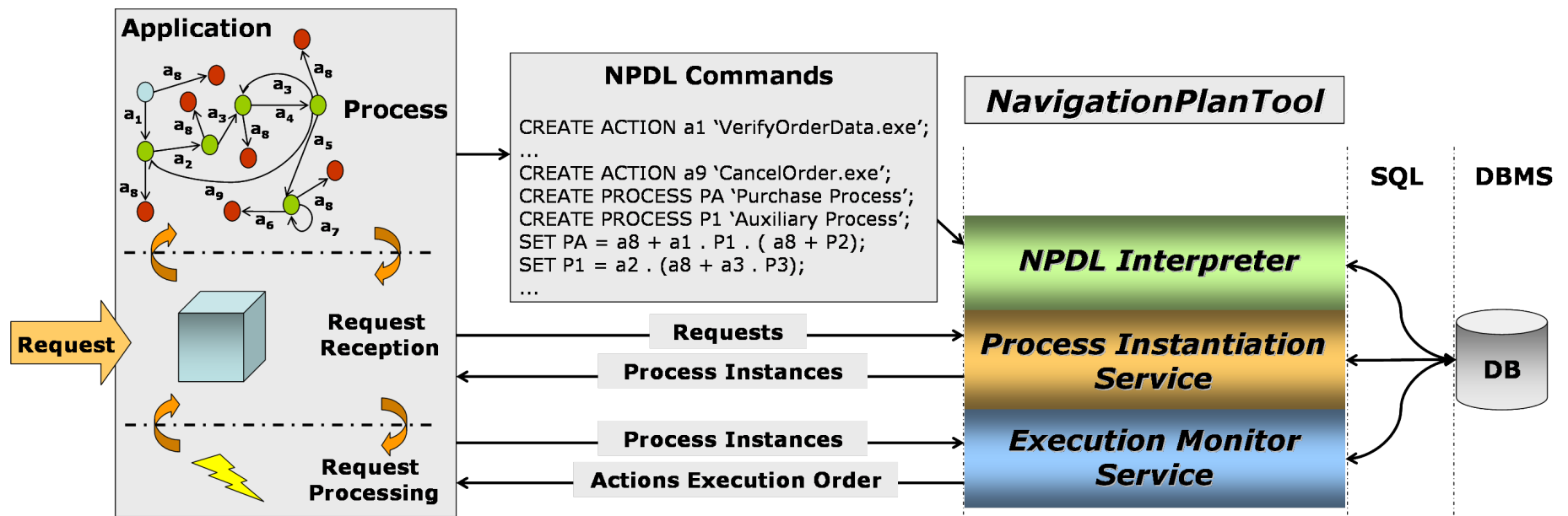
Recursão

- $M = a \cdot M + b$.



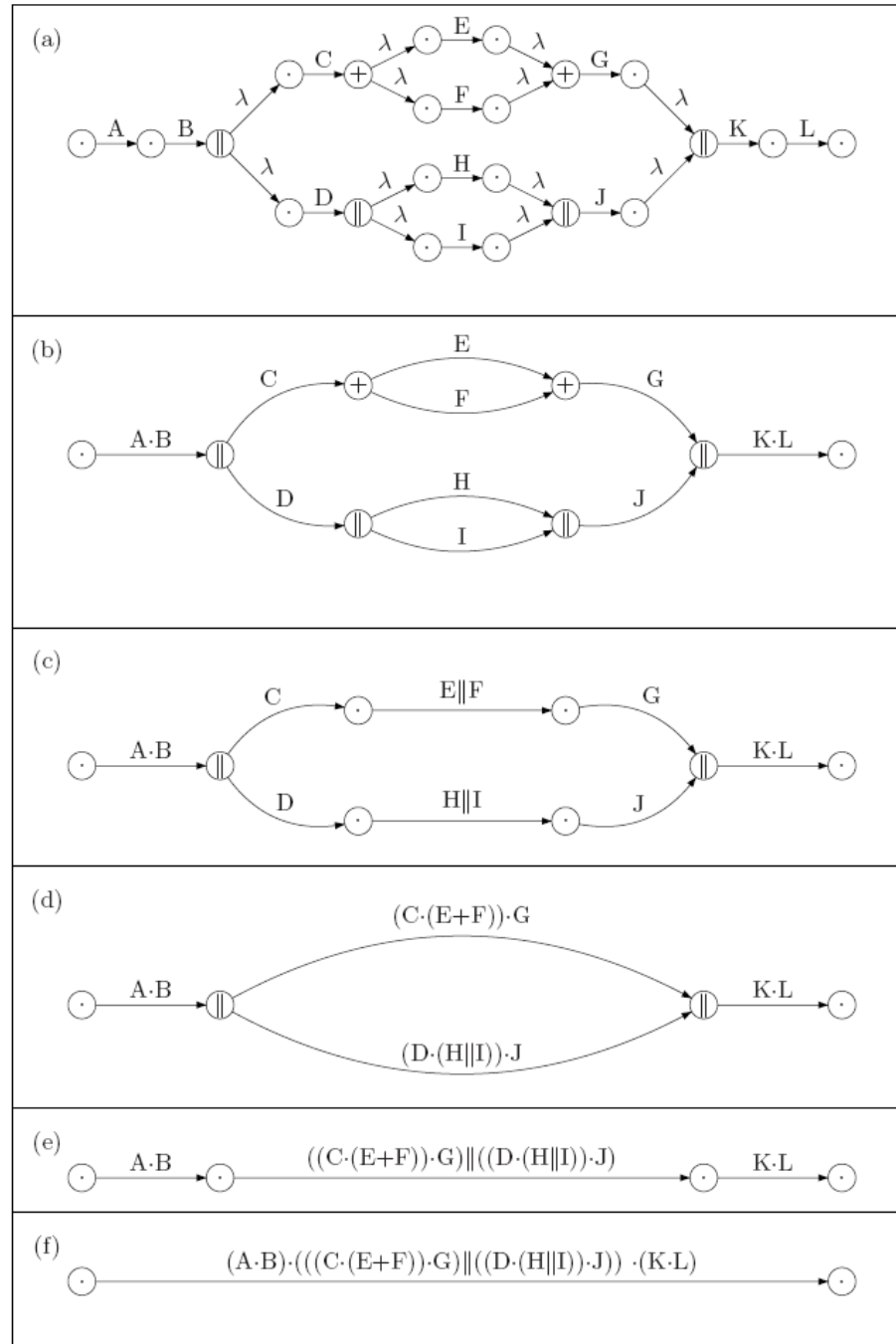
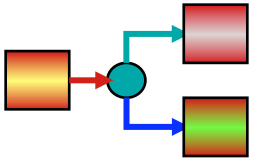


Contribuições: Grupo de pesquisa



Fonte: Braghetto 2008

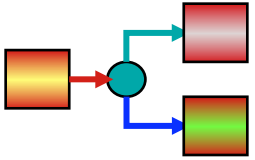


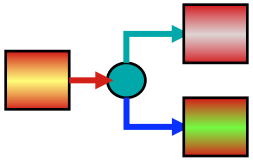


Fonte: Oikawa 2008



IME - USP





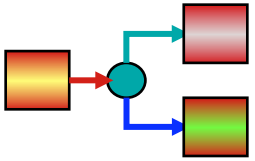
Transactional model based on data states

- bo_1 registration received
- bo_2 key received
- bo_3 money sent
- data states

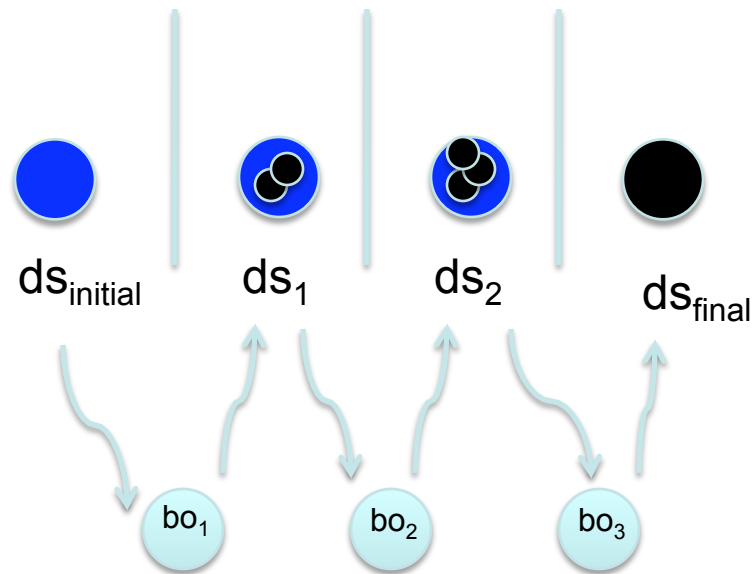
$ds_{initial}, ds_1, ds_2, ds_{final}$

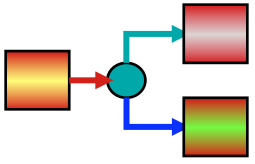
$ds_{initial}$ initial data state (generated by request)

ds_{final} (final data state) = ds_3

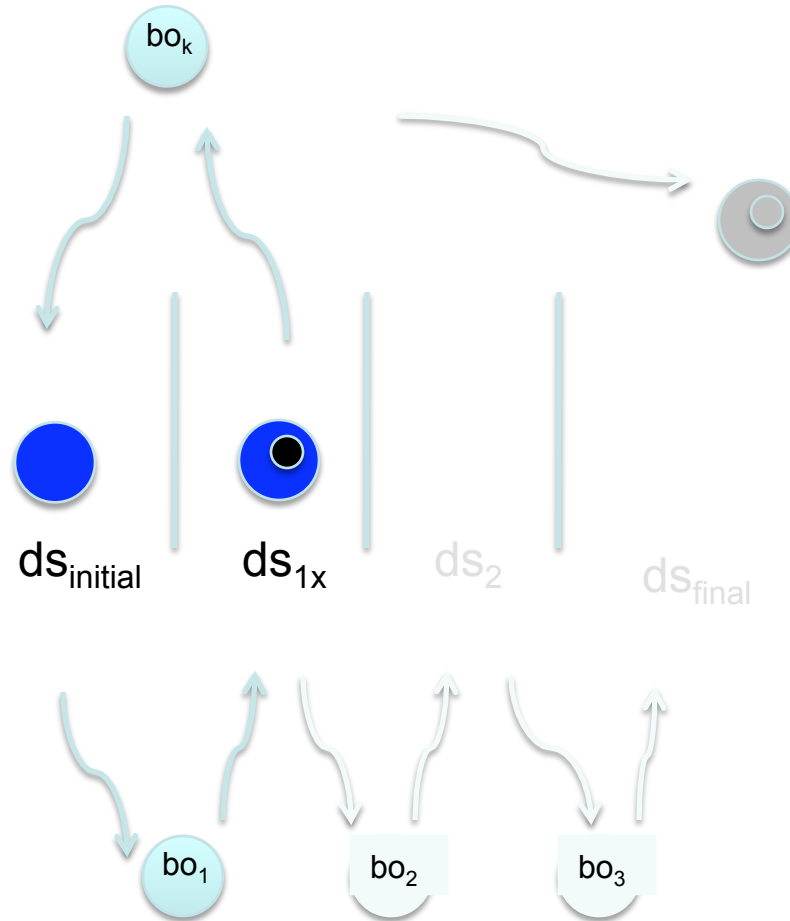


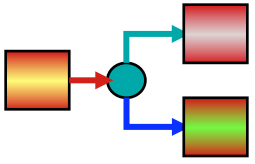
Transactional model based on data state



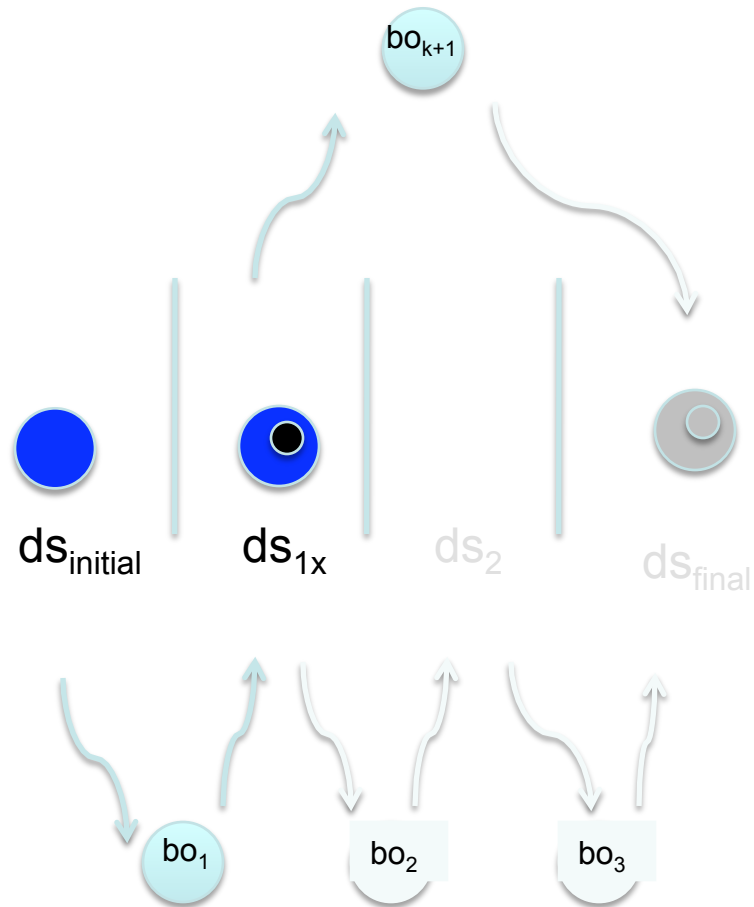


If $ds_{1x} \neq ds_1$ and there is bo_k

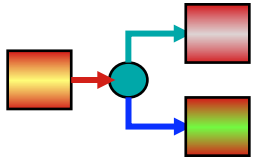




If $ds_{1x} \neq ds_1$ and there is bo_{k+1}

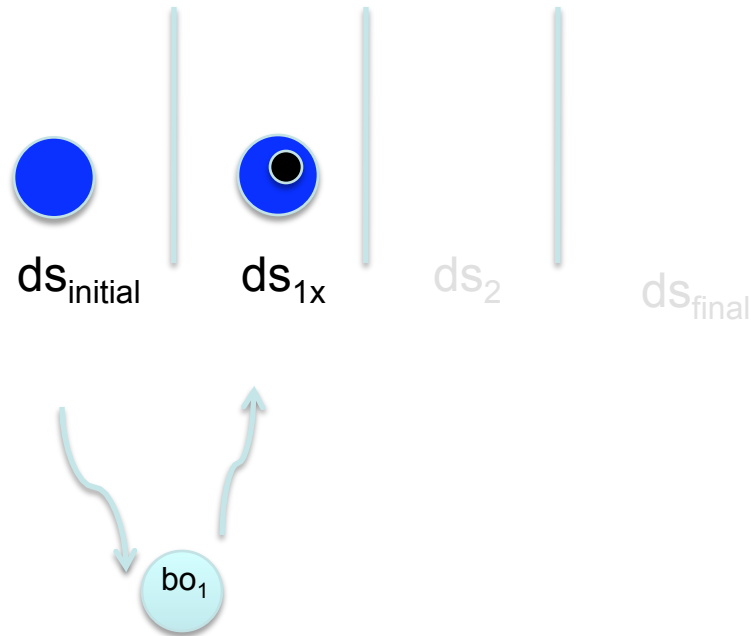


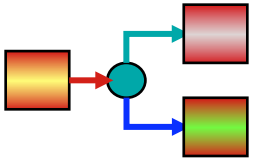
However, if



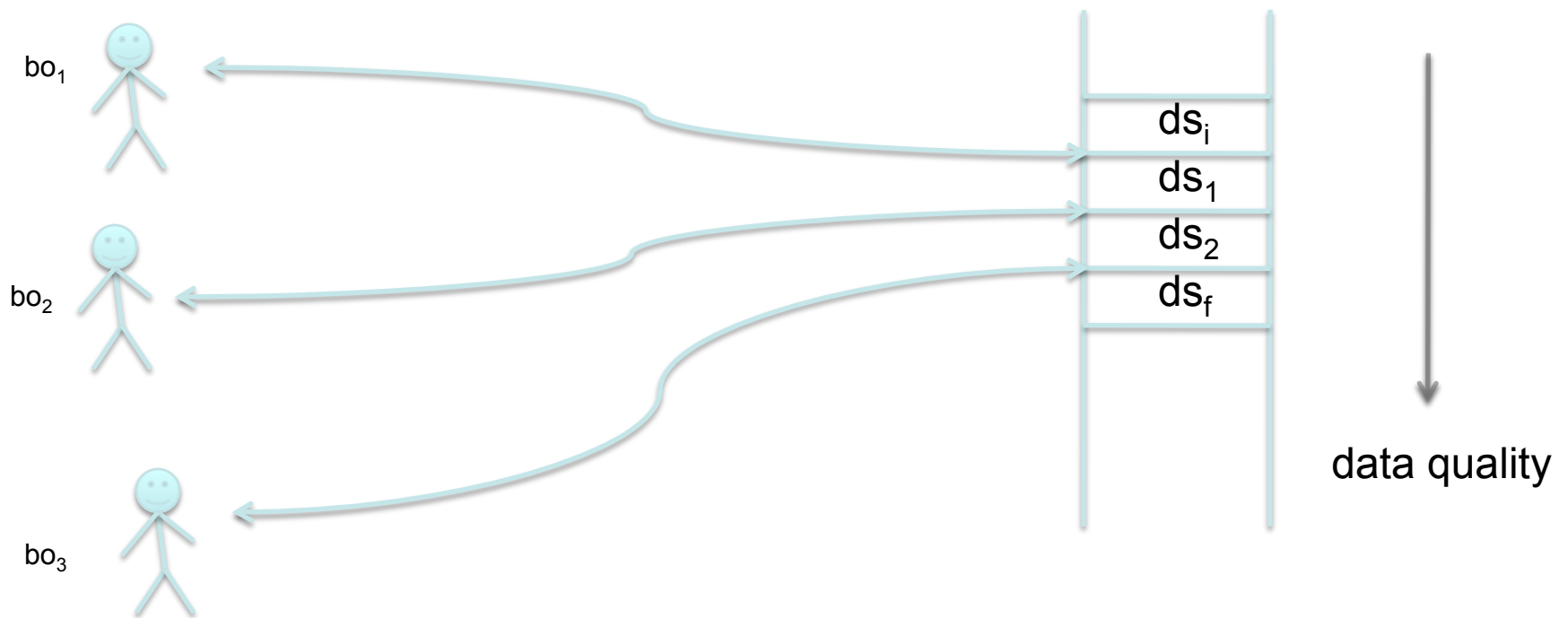
$$ds_{1x} \neq ds_1$$

and the ds_{1x}
is non-expected
data state ...
exceptions

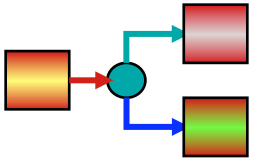




Scheduling data states: input-output data states



$$ds_{initial} < ds_1 < ds_2 < ds_{final}$$



Queries

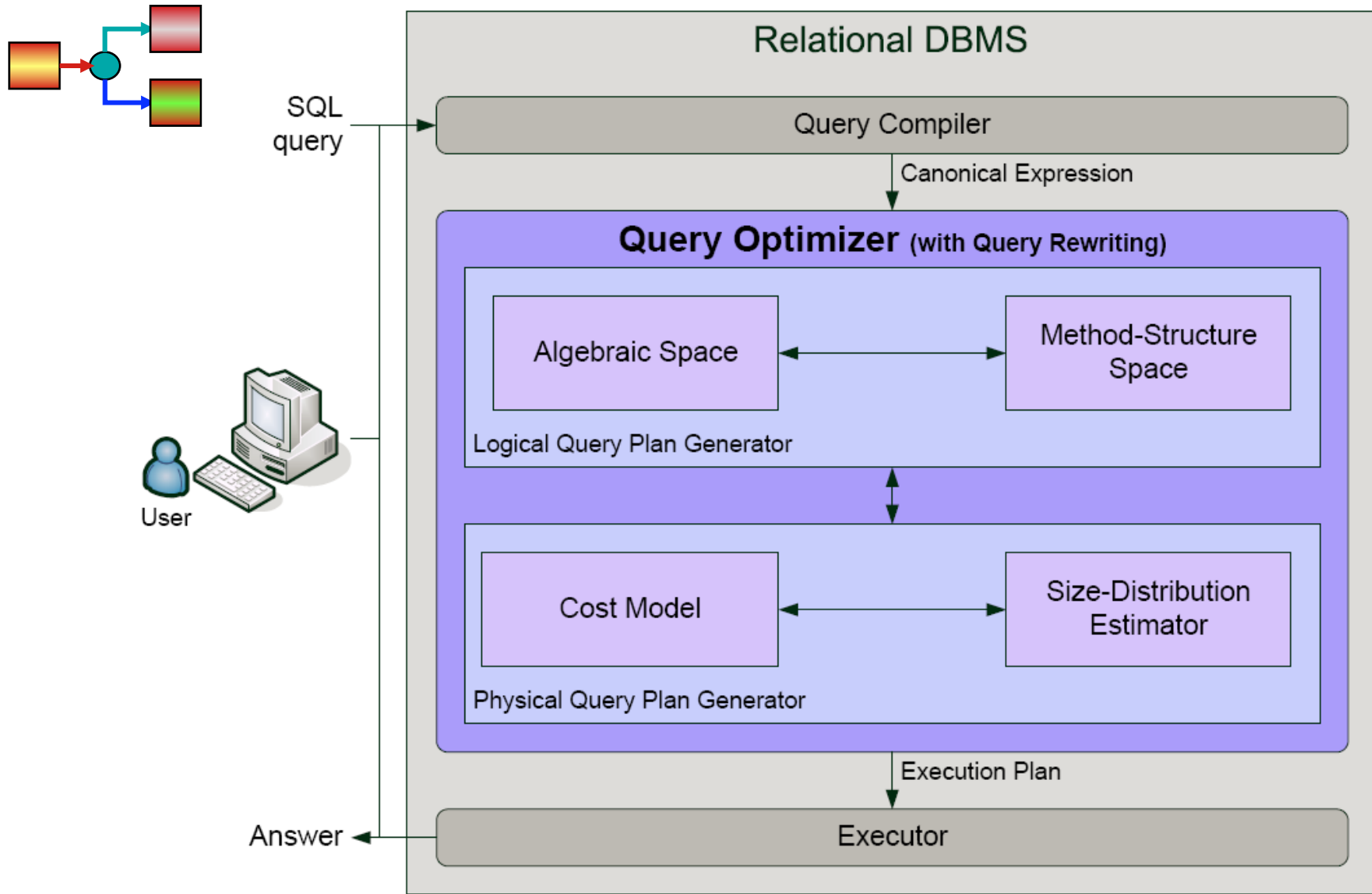
Relational Database Management Systems (DBMSs):

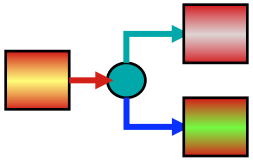
- **Traditional Data:** numerical and string
 - Traditional operators
 - exact matching ($=$ and \neq)
 - relational - TOR ($<$, \leq , $>$ and \geq)

Example

- $4.9 \neq 5.0$
- algebra $<$ algebraic

Relational DBMS architecture:





Queries

Dataset example

- Relation:

`EuropeanCountries = {Abbreviation, Country, Capital, Latitude, Longitude}`

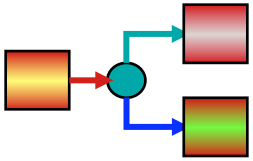
- 50 tuples
- 100 bytes
- disk block - 1 Kb

SQL query

Select the European capitals that have the latitude and longitude less than Paris latitude and longitude, respectively.

Relational algebra

$\sigma_{(\text{Latitude} < 48.87 \text{ AND Longitude} < 2.33)} \text{EuropeanCountries}$

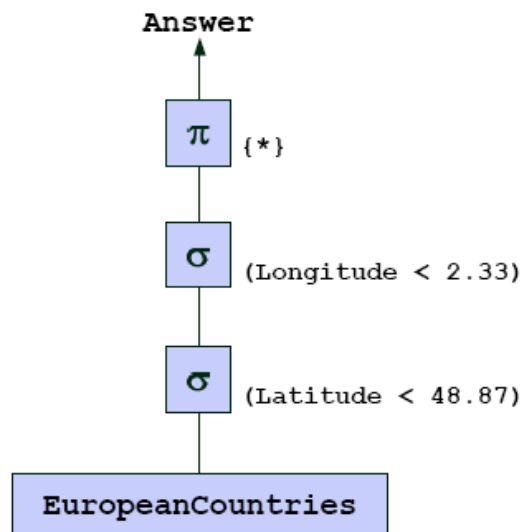


Queries

Relational algebra

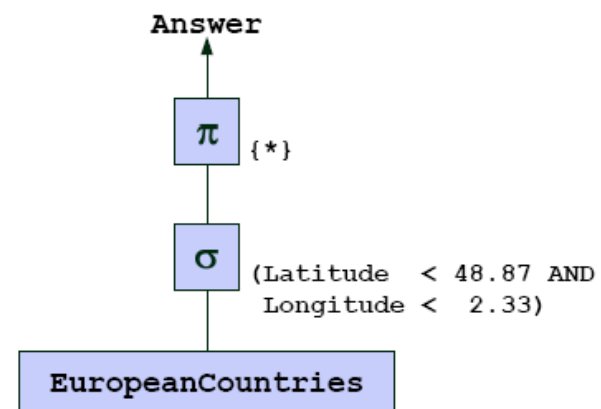
$\sigma(\text{Latitude} < 48.87 \text{ AND } \text{Longitude} < 2.33) \text{ EuropeanCountries}$

- Canonical plan

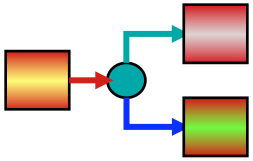


14 disk accesses.

- Alternative plan



8 disk accesses.



Queries

- **Complex data:** image, audio, multi-dimensional data, time series, genomic sequences, . . .

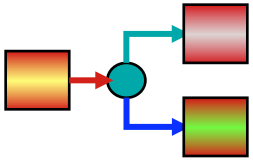


Tom Hanks

?

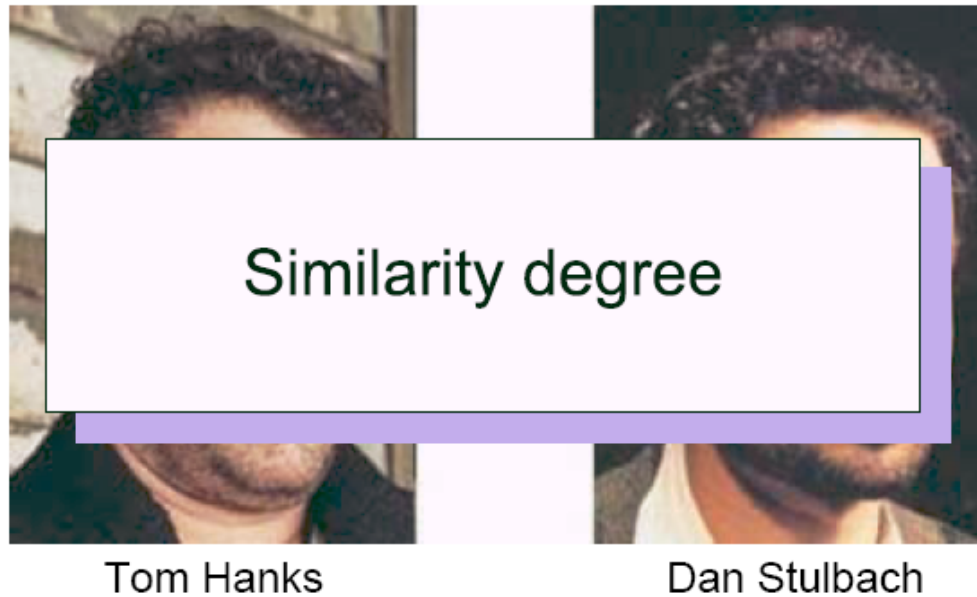


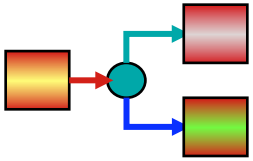
Dan Stulbach



Queries

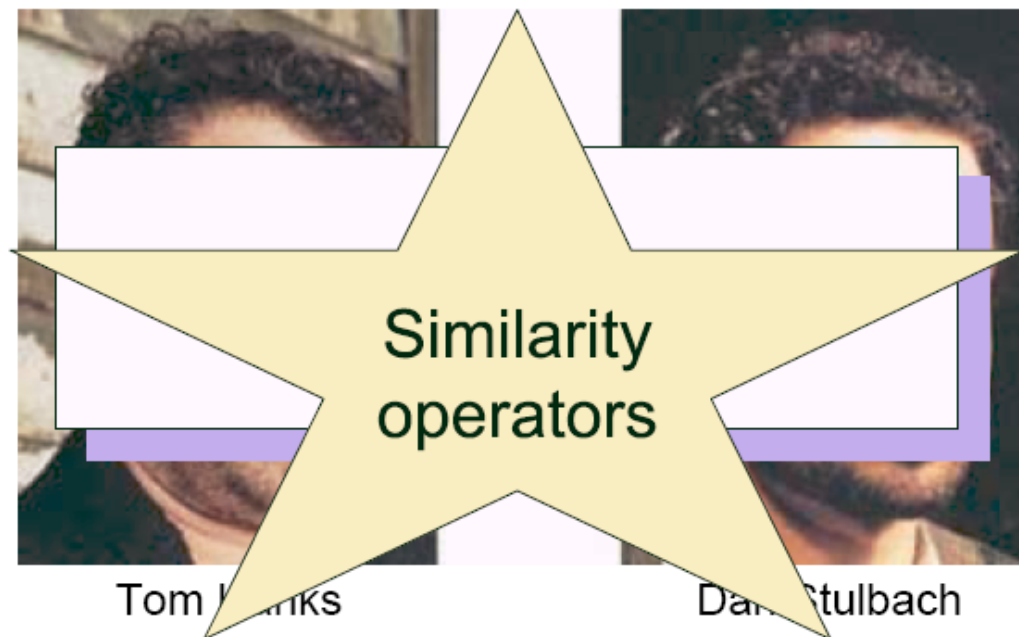
- **Complex data:** image, audio, multi-dimensional data, time series, genomic sequences, ...

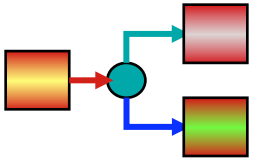




Queries

- **Complex data:** image, audio, multi-dimensional data, time series, genomic sequences, ...





Queries

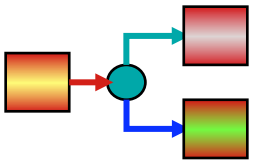
Distance function or metric $\Rightarrow d$

Quantifies the similarity between two elements.

Metric space

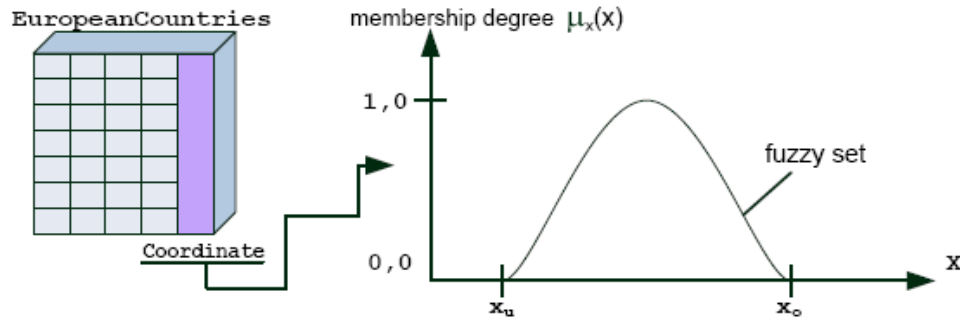
Definition: A **metric space** M is defined by the pair $\langle \mathbb{S}, d \rangle$ if the metric d , associated with an element domain \mathbb{S} , satisfying the properties:

- symmetry: $d(s_1, s_2) = d(s_2, s_1)$;
- non-negativity: $0 < d(s_1, s_2) < \infty$ if $s_1 \neq s_2$ and $d(s_1, s_1) = 0$;
- triangular inequality: $d(s_1, s_2) \leq d(s_1, s_3) + d(s_3, s_2)$



Queries

- Fuzzy



- Rank

EuropeanCountries

						Score
						1
						2
						3
						4
						5
						6

- Exact

EuropeanCountries

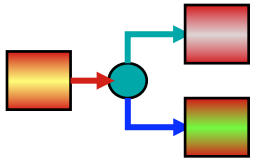
Abbreviation	Country	Capital	Latitude	Longitude	Coordinate

Traditional Complex

- Hybrid = Rank + Fuzzy

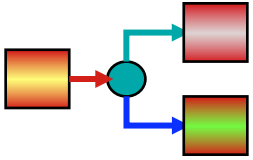
EuropeanCountries

						Score
						1
						2
						3
						4
						5
						6



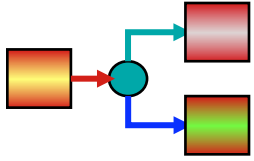
Conclusões

- Banco de dados não é mais uma área isolada
- Antes:
 - conhecimentos necessários para atuar na área vinham de alguns **poucos livros-texto**
- Hoje:
 - **amplo espectro** de conhecimentos necessários para atuar com sucesso na área
 - projetos bem sucedidos tendem a incluir pesquisadores de **várias áreas** da computação (IR, ML, IA, paralelismo, ...)



Conclusões

- A área de banco de dados **deixou de tratar apenas o armazenamento de grandes bases de dados.**
- O foco da pesquisa na área é:
 - tratamento de **grandes massas de dados em geral** (não necessariamente persistentes);
 - tratamento de dados que tenham **alguma estrutura.**



Conclusões

- Fomos de
 - **Gerência de Banco de Dados** (*database management*)
para
 - **Gerência de grandes volumes de Dados** (*data management*).

Conclusões

- Banco de dados deixou de ser uma área de pesquisa **compartimentada** e **isolada** das demais, para tornar-se uma área de pesquisa **multi-disciplinar**.
- **Base matemática** necessária para atuar na área **aumentou** muito.
- Para pesquisar com **sucesso** na área e necessário montar **equipes multi-disciplinares**.