

Este exemplar corresponde à redação final da  
Tese/Dissertação devidamente corrigida e defendida  
por: Bei Yi  
e aprovada pela Banca Examinadora.  
Campinas, 25 de AGOSTO de 2004  
*[Assinatura]*  
COORDENADOR DE PÓS-GRADUAÇÃO  
CPG-10

**Um modelo de dados para objetos móveis**

*Bei Yi*

**Dissertação de Mestrado**

UNICAMP  
BIBLIOTECA CENTRAL  
DESENVOLVIMENTO DE COLEÇÃO

## Um modelo de dados para objetos móveis

Bei Yi

09 de Junho de 2004

---

### Banca Examinadora:

- Profa. Dra. Claudia Bauzer Medeiros, IC-UNICAMP (Orientadora)
- Prof. Dr. Joao Eduardo Ferreira, IME-USP
- Profa. Dra. Anamaria Gomide, IC-UNICAMP
- Profa. Dra. Maria Beatriz Felgar de Toledo (Suplente), IC-UNICAMP

UNIDADE	BC
1ª CHAMADA	I/UNICAMP
	B396m
	EX
COMBO, BC/	61021
PROC.	16-117-04
	C <input type="checkbox"/> D <input checked="" type="checkbox"/>
PREÇO	11,00
DATA	19.11.04
1ª CPD	

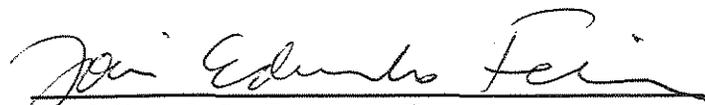
pb Id 332651

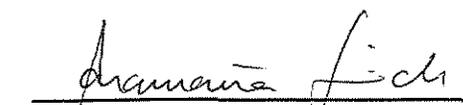
**FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DO IMECC DA UNICAMP**

Bei, Yi
B396m Um modelo de dados para objetos móveis / Bei Yi -- Campinas, [S.P. :s.n.], 2004.
Orientador : Claudia Bauzer Medeiros
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.
1. Banco de dados temporais. 2. Banco de dados orientados a objetos. 3. Sistemas de informação geográfica. I. Medeiros, Claudia Bauzer. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

## TERMO DE APROVAÇÃO

Tese defendida e aprovada em 07 de julho de 2004, pela Banca examinadora composta pelos Professores Doutores:

  
\_\_\_\_\_  
**Prof. Dr. João Eduardo Ferreira**  
IME - USP

  
\_\_\_\_\_  
**Prof. Dra. Anamaria Gomide**  
IC - UNICAMP

  
\_\_\_\_\_  
**Prof. Dra. Claudia Maria Bauzer Medeiros**  
IC - UNICAMP

2004.20934

# Um modelo de dados para objetos móveis

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Bei Yi e aprovada pela Banca Examinadora.

Campinas, 07 de Julho de 2004.



---

Profa. Dra. Claudia Bauzer Medeiros,  
IC-UNICAMP (Orientadora)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

---

© Bei Yi, 2004.  
Todos os direitos reservados.

# Agradecimentos

A Deus pela sua sabedoria infinita em me guiar nesta estrada da vida.

À minha eterna mestra Claudia Bauzer Medeiros, sem a qual este trabalho não teria iniciado nem terminado. Não há palavras nesse mundo para expressar a minha gratidão pela sua paciência e dedicação em me ajudar superar cada empecilho.

Aos meus pais, pela filosofia de vida e todo alicerce moral que me apoiaram em todos momentos difíceis.

Ao Bismark, pelo carinho, paciência, correções ortográficas e longas discussões técnicas.

Aos Professores Pedro Rezende e Jorge Stolfi pelas discussões interessantes de geometria computacional.

---

Aos meus amigos do grupo de Banco de Dados do IC-UNICAMP: Daniel Andrade, Nielsen Simões, Flávio, André Santache.

Aos meus amigos da Motorola: Eduardo Seiti, João Loureiro, Silvio Leite, Elaine Haddad, pelas discussões e pelo apoio. Aos meus gerentes da Motorola, Rosana Jamal, Alice Arashiro, Rose Hecksher, José Barletta pelo suporte e visão de investimento profissional dos funcionários.

Aos funcionários do IC, pela orientação e atendimento atencioso.

À minha banca examinadora, pelas revisões e contribuições valiosas.

Este trabalho foi parcialmente financiado pelos projetos PRONEX-MCT SAI, CNPq WebMaps e Agroflow.

# Resumo

A popularização dos dispositivos como GPS tem possibilitado o surgimento de aplicações que fazem coleta e análise de objetos móveis. Os sistemas de bancos de dados tradicionais não apóiam o gerenciamento de dados relativos a objetos móveis, caracterizados por gerarem um grande número de informações continuamente. As pesquisas nesta área são recentes, com relativamente poucos resultados específicos à gerencia de dados móveis.

A dissertação propõe um modelo de dados orientado a objetos que incorpora duas características: tratamento uniforme de objetos estáticos, espaciais, temporais, espaço-temporais e móveis; e um conjunto básico de operadores e os algoritmos destes operadores podem ser compostos progressivamente para criar um grande variedade de operações para consultas ao banco de dados. Este modelo se diferencia das demais propostas por considerar não apenas objetos móveis 1D, mas também aquelas com geometria 2D.

# Abstract

The popularization of devices like GPS and wireless network have enabled new applications that collect and analyze mobile objects. Traditional database system do not support the management of mobile object data, since a great amount of information is generated continuously. Research in this area is recent with relatively few works on mobile data management.

This Msc thesis proposes an object-oriented mobile object data model that incorporates two characteristics: it supports modeling of static, spatial, temporal, spatio-temporal and mobile objects in a homogeneous way; it specifies a set of basic operators and their algorithmic specification. These operators can be composed to obtain a wide variety of complex operators to query a mobile object database. Unlike most other proposals, this mode supports not only 1D object, but also those with 2D geometric descriptions.

---

# Sumário

<b>1</b>	<b>Motivação e Objetivos da Dissertação</b>	<b>1</b>
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>3</b>
2.1	Introdução de tempo em bancos de dados . . . . .	3
2.1.1	Representação dos dados temporais . . . . .	4
2.1.2	Semântica dos dados temporais . . . . .	7
2.2	Introdução do conceito de espaço em bases de dados . . . . .	
2.2.1	Representação dos dados espaciais . . . . .	
2.2.2	Semântica dos dados espaciais . . . . .	7
2.3	Espaço e Tempo . . . . .	8
2.3.1	Analogias entre espaço e tempo . . . . .	9
2.3.2	Combinação de espaço e tempo . . . . .	9
2.4	Objetos Móveis . . . . .	12
2.4.1	Caracterização de sistemas de objetos móveis . . . . .	12
2.4.2	Representação de objetos móveis em banco de dados . . . . .	13
2.4.3	Desafios de bancos de dados de objetos móveis . . . . .	14
2.4.4	Modelos de dados para objetos móveis . . . . .	14
2.5	Resumo . . . . .	17
<b>3</b>	<b>Modelo de dados para objetos móveis</b>	<b>19</b>
3.1	Conceitos básicos . . . . .	19
3.1.1	O que se consulta sobre objetos móveis . . . . .	20
3.1.2	Movimento e Trajetória . . . . .	21
3.1.3	O Modelo de Dados Geográfico Temporal de Faria . . . . .	22
3.2	Apresentação do modelo proposto . . . . .	25
3.2.1	Idéia básica . . . . .	25
3.2.2	Modelo estendido e classes principais . . . . .	25
3.2.3	Objetos móveis e as classes derivadas . . . . .	27
3.2.4	Descrição de um objeto móvel . . . . .	29

3.3	Resumo . . . . .	30
<b>4</b>	<b>Operadores Móveis</b>	<b>32</b>
4.1	Introdução . . . . .	32
4.2	Consultas a bancos de dados móveis . . . . .	32
4.2.1	Operadores Espaciais . . . . .	33
4.2.2	Operadores Temporais . . . . .	35
4.2.3	Operadores Espaço-temporais . . . . .	37
4.3	Operadores espaço-temporais para objetos móveis . . . . .	38
4.3.1	Introdução . . . . .	38
4.3.2	Preprocessamento das trajetórias . . . . .	39
4.3.3	Operadores Localização-temporais Móveis . . . . .	39
4.3.4	Operadores Métrico-temporais móveis . . . . .	40
4.3.5	Operadores Topológico-temporais móveis . . . . .	41
4.3.6	Operadores compostos móveis . . . . .	58
4.4	Resumo . . . . .	60
<b>5</b>	<b>Consultas a objetos móveis</b>	<b>61</b>
5.1	Introdução . . . . .	61
5.2	Definição de Classes . . . . .	62
5.3	Predicado Temporal . . . . .	63
5.4	Predicado Espacial . . . . .	65
5.5	Predicado espaço-temporal . . . . .	65
5.5.1	Predicados métrico-temporais . . . . .	65
5.5.2	Predicados topológico-temporais . . . . .	67
5.6	Consultas sobre o futuro . . . . .	69
5.7	Resumo . . . . .	69
<b>6</b>	<b>Conclusões e Extensões</b>	<b>70</b>
	<b>Bibliografia</b>	<b>72</b>

# Lista de Figuras

3.1	Trajétórias traçadas por ponto, linha e polígono. . . . .	22
3.2	Modelo de dados proposto pela Faria. . . . .	24
3.3	Modelo de dados proposto . . . . .	26
4.1	Exemplo 1 (mapeamento 3D-2D): Intersecção de $S \times \text{TrajM} \rightarrow M$ tem intersecção com $S$ . . . . .	44
4.2	Exemplo 2 (mapeamento 3D-2D): Intersecção de $C \times \text{TrajM} \rightarrow M$ tem intersecção com $C$ . . . . .	44
4.3	Exemplo 3 (mapeamento 3D-2D): Intersecção de $\text{TrajM1} \times \text{TrajM2} \rightarrow M1$ não tem intersecção com $M2$ . . . . .	45
4.4	Exemplo 4 (mapeamento 3D-2D): Intersecção de $\text{TrajM1} \times \text{TrajM2} \rightarrow M1$ tem intersecção com $M2$ . . . . .	46
4.5	Exemplo 5 (mapeamento 3D-2D): Intersecção de $\text{TrajM1} \times \text{TrajM2} \rightarrow M1$ não tem intersecção com $M2$ . . . . .	46
4.6	Exemplo 6 - Dois objetos móveis (Carro $M1$ e Carro $M2$ ) . . . . .	47
4.7	Operador $M\_DISJOINT$ . . . . .	49
4.8	Operador $M\_OVERLAP$ . . . . .	51
4.9	Operador $M\_MEET$ . . . . .	54
4.10	Operador $M\_INSIDE$ . . . . .	57
4.11	Grafo do relacionamento topológico . . . . .	59

# Capítulo 1

## Motivação e Objetivos da Dissertação

O surgimento das novas tecnologias para obtenção de dados espaciais e de meios de comunicação tornou possível coletar dados de objetos móveis e transmití-los em tempo real. A popularização dos dispositivos como o GPS (Global Position System) possibilitou usos pessoais e comerciais antes restritos ao uso militar. Isso, por sua vez, abriu uma vasta gama de novas aplicações, chamadas de *aplicações móveis*. Exemplos destas aplicações são: sistemas de monitoramento ambiental e biológico; controle e monitoramento de veículos terrestres; estudo de correntes migratórias.

O interesse principal neste tipo de aplicação são as características de *movimento*. A velocidade pode variar e, conseqüentemente, as taxas de amostragem também - veja-se, por exemplo, o contraste entre trajetória de veículos e migração de seres humanos. Além disso, certas aplicações envolvem solução em tempo real e requerem previsão do futuro (monitoramento de veículos) enquanto outras não requerem um acompanhamento em tempo real mas envolvem análise mais complexa de padrões (monitoramento ambiental)[Peu01].

Tais aplicações introduzem um novo tipo de conceito que são os *objetos móveis*. Estes trazem alguns desafios antes não enfrentados na pesquisa de banco de dados ou de sistemas geográficos. Primeiramente, ao contrário dos objetos normalmente gerenciados nesses sistemas, cuja localização costuma permanecer constante, os dados de objetos móveis mudam continuamente com o tempo. Bancos de dados espaço-temporais são adequados para armazenar mudanças espaciais discretas de objetos em função do tempo, mas não são adequados para suportar mudança contínua [EGSV97, SWCD97]. Em segundo lugar, modelos de dados propostos para objetos móveis são raros e maioria só trata de objetos pontuais. Em terceiro lugar, tais modelos raramente vêm acompanhados por descrição de operadores que explorem efetivamente as novas possibilidades que os objetos móveis podem oferecer.

O objetivo desta dissertação é tratar dos três problemas levantados no parágrafo anterior, propondo um modelo para objetos móveis georreferenciados (relacionados a algum local na superfície da Terra). Este modelo é orientado a objetos e pode ser implementado utilizando um SGBD que provê mecanismos básicos de gerenciamento de consultas.

O modelo proposto caracteriza-se pelas seguintes propriedades:

- estende o trabalho de Faria [Far98], o qual modela objetos espaço-temporais discretos. A extensão facilita o tratamento genérico de objetos em mudança contínua (móveis) e de objetos em mudança discreta;
- usa a idéia de agrupar as versões dos objetos móveis em cada intervalo de tempo para formar a trajetória do objeto em função do tempo;
- trata não só os objetos pontuais mas também objetos extensos (linhas e polígonos)
- propõe operadores específicos que exploram as novas possibilidades de consultas, necessários a aplicações móveis.

Com isso, as principais contribuições são:

- Proposta de um modelo de objetos que consideram objetos móveis pontuais e extensos e, trata de maneira uniforme objetos de mudança discreta e objetos de mudança contínua (vide Capítulo 3), através de amostragens subsequentes.
- Especificação de um conjunto básico de operadores que leva em consideração as características de mobilidade. Estes operadores são apresentados junto com o detalhe dos algoritmos respectivos (vide Capítulo 4).
- Validação da proposta pela instanciação de exemplos de consultas utilizando o modelo e operadores propostos (vide Capítulo 5).

Esta dissertação já deu origem a um artigo publicado no IV GeoInfo[YM02].

O restante da dissertação está organizado da seguinte forma. O capítulo 2 dá uma visão geral de pesquisa em bancos de dados de objetos móveis. O capítulo 3 descreve o modelo de objetos proposto. O capítulo 4 especifica um conjunto básico de operadores espaço-temporais, e depois, a partir desses operadores, especifica os operadores específicos para objetos móveis e os algoritmos para implementar cada um deles. O capítulo 5 descreve exemplos de consultas sobre os objetos definidos no capítulo 3 usando os operadores especificados no capítulo 4. Finalmente, o capítulo 6 apresenta as conclusões e possíveis extensões do trabalho.

# Capítulo 2

## Revisão Bibliográfica

Modelos de dados por definição são formalismos e ferramentas que descrevem os dados do mundo real para um conjunto de usuários e aplicações. Parte dessas ferramentas permite o mapeamento do mundo modelado para o banco de dados. Os primeiros tipos de dados para os quais foram propostos modelos são chamados de dados convencionais. Esses dados são tipicamente alfanuméricos e tratam de propriedades descritivas e estáticas de objetos. Com o advento de novas necessidades e de avanços tecnológicos, as variáveis de tempo e espaço foram introduzidos nos sistemas de banco de dados.

Este capítulo apresenta uma visão geral da bibliografia correlata, em especial, trabalhos relativos a modelagem de dados espaciais, temporais e objetos móveis.

---

### 2.1 Introdução de tempo em bancos de dados

A introdução do fator tempo em bancos de dados deu origem à pesquisa sobre SGBD *temporais*. Nos sistemas que contêm apenas dados convencionais, eventuais atributos envolvendo tempo são manipulados unicamente por programas de aplicação e o sistema de gerenciamento de banco de dados interpreta o tempo como valores de tipos de dados básicos [Sno92]. Tradicionalmente os sistemas que armazenam dados convencionais capturam uma imagem instantânea da realidade. Sendo assim, não se adequam a aplicações que requerem informações passadas, presentes e futuras [MRSS01].

O conceito de tempo foi aos poucos incorporado a modelos de dados para suportar consultas temporais complexas. O padrão atualmente consolidado é denominado TSQL2 [Sno95]. Além de permitir formulação de consultas complexas, permite também atualizações de estados passados (se algum erro for detectado ou quando alguma informação torna-se disponível) e consultas sobre estados futuros (para propósitos de planejamento).

Vale caracterizar aqui os dados temporais quanto aos aspectos de representação e semântica e esclarecer quais são principais diferenças em relação aos dados convencionais.

### 2.1.1 Representação dos dados temporais

A representação dos dados temporais precisa levar em consideração sua natureza, unidade de medida e duração.

- Natureza

O tempo é uma variável de natureza contínua, ou seja, é isomorfo aos números reais. A maioria dos dados convencionais, ao contrário, são discretos, ou seja, isomorfos aos números naturais. Apesar disso, quando a dimensão temporal é incorporada a um modelo de dados, a sua representação se baseia num modelo discreto. Segundo Faria [Far98], existem quatro razões para tal. Primeiro, as medidas do tempo são imprecisas. Segundo, a maioria das referências a tempo da linguagem natural são compatíveis com o modelo discreto de tempo. Terceiro, muitos eventos não são instantâneos, pois têm duração. Finalmente, a implementação de um modelo de dados necessariamente precisa de uma codificação discreta para tempo.

Por causa da discrepância entre o modelo discreto e a natureza contínua do tempo, podem existir informações que não possuem valor de tempo preciso ou mesmo ausência do valor. Isso gera a chamada *indeterminância temporal*.

- Unidades de medida

Num modelo discreto do tempo, a menor unidade de tempo costuma ser chamada de *chronon*. Ela é uma unidade indivisível de tempo com uma duração fixa definida por cada aplicação. Uma vez definido o *chronon*, ele é utilizado pela aplicação para representar um instante de tempo. Existe um instante que recebe uma denominação especial de *Now* que representa o instante atual.

A noção de intervalo temporal é utilizado para representar eventos que não são instantâneos. Um intervalo temporal é caracterizado pelo tempo decorrido entre dois instantes. Em um sistema que suporta uma linha de tempo composta por *chronons*, um intervalo pode ser representado por um conjunto destes *chronons* contíguos. Um intervalo é representado pelos instantes que o delimitam. Dependendo da pertinência ou não dos instantes limitantes do intervalo, este pode ser aberto (os limitantes não pertencem ao intervalo), semi aberto (um dos limitantes pertence ao intervalo) ou fechado (os limitantes pertencem ao intervalo). Quando um dos limites é representado pelo instante atual (*now*) tem-se a representação de um intervalo particular cujo tamanho varia com a passagem do tempo.

Tanto os instantes quanto os intervalos de tempo são denominados de marcas de tempo. Uma marca de tempo é um valor de tempo que pode ser associado a um atributo, tupla ou relação em um banco de dados relacional, ou a um atributo ou um objeto em bancos de dados orientado a objetos.

- Duração

Duração é um valor de tempo relativo a um instante inicial, podendo ser fixa ou variável. O *tempo relativo* tem sua posição relativa a alguma entidade ou a uma variável de tempo como *now*. O *tempo absoluto* define a posição de uma entidade no eixo temporal. Esta posição não depende nem do tempo de outra entidade, nem de variáveis especiais como *now*.

Uma duração pode ser fixa ou variável. Uma duração fixa é constituída por um número determinado de chronons. Um exemplo disso é a duração de um minuto, constituído de 60 segundos. Já o mês é um exemplo de duração variável, pois dependendo do mês e ano, o número de dias que compõem o mês é diferente. Quando a duração for positiva, ela denota passagem de tempo para o futuro, quando negativa, denota volta para o passado.

## 2.1.2 Semântica dos dados temporais

A semântica temporal está associada ao tipo de dados temporais utilizados. Em geral, supõe-se que dados temporais são dados descritivos associados a marcas de tempo que têm semânticas específicas. Banco de dados temporais tradicionalmente consideram dois tipos de tempo: tempo de transação e tempo de validade[JCE<sup>+</sup>94].

O *tempo de Transação* de um dado no banco de dados é a marca de tempo em que aquele dado foi efetivamente armazenado. O tempo de transação associado a um conjunto de dados identifica o tempo das transações que inseriram/atualizaram o conjunto do banco de dados. Os valores de tempo de transação costumam ser supridos automaticamente pelo próprio SGBD.

O *tempo de Validade* de um fato é o tempo em que o fato é verdade na realidade modelada. Este valor é fornecido pelo usuário na maioria das vezes. Os tempos de transação e de validade são ortogonais.

A partir dessa classificação de tempos, os bancos de dados são classificados como:

- históricos (que utilizam tempo de validade);
- rollback (que utilizam tempo de transação);
- bitemporal (que utilizam tanto tempo de validade quanto de transação).

Operadores temporais agem sobre marcas de tempo para comparar registros em bancos de dados históricos, de rollback ou bitemporal. Exemplo dos operadores temporais são: BEFORE, OVERLAP, EQUAL, CONTAINS, MEETS, com as semânticas associadas. Por exemplo, (t1 BEFORE t2) indica que a marca de tempo t1 referencia valores que são menores que aquelas da marca de tempo t2.

## 2.2 Introdução do conceito de espaço em bases de dados

A partir dos anos 90, o espaço tornou-se objeto de estudo em banco de dados. Sistemas de Informação Geográfica(SIG) são sistemas automatizados usados para armazenar, analisar, manipular e visualizar dados geográficos, ou seja, dados que representam objetos e fenômenos em que a localização geográfica é uma característica inerente à informação e indispensável para analisá-la [CCH<sup>+</sup>96].

### 2.2.1 Representação dos dados espaciais

Segundo [CCH<sup>+</sup>96], dados espaciais têm apresentação *matricial* ou *vetorial*. O formato *matricial* é caracterizado por uma matriz de células de tamanhos regulares, onde a cada célula é associado um valor representando as características geográficas da região correspondente. O formato *vetorial* descreve dados espaciais com uma combinação de formas geométricas (pontos, linhas e polígonos). O termo *raster* designa células retangulares, mas na maioria das vezes é usado como termo genérico para a representação matricial.

O uso dos modelos matricial ou vetorial é dependente da aplicação. Segundo [Eas90], como o modelo matricial modela o mundo em termos de amostras regularmente espaçadas (cada célula possui o mesmo tamanho), ele é mais apto para capturar fenômenos de pouca variação espacial. Por exemplo, o modelo matricial é usual em aplicações que usam imagens de satélite.

Já o modelo vetorial modela o mundo por entidades que são apresentadas como objetos discretos, sendo mais adequado para representar fenômenos dinâmicos e identificáveis como objetos individuais. Por exemplo, são usados para representar artefatos construídos pelos homens - ruas, estradas, cidades.

Os dados matriciais podem ser processados mais rapidamente para a maioria das questões envolvendo sobreposição e aproximação usando operações algébricas de matrizes. Nestas situações o modelo vetorial requer um processamento mais pesado pois a topologia vetorial é complexa. Essas dificuldades se acentuam quando o espaço não se restringe mais a uma dimensão (ponto) mas a duas e três dimensões.

A dissertação adota o modelo vetorial como modelo de representação dos dados espaciais pois o modelo vetorial se adequa mais a descrever objetos com taxa de amostragem variada e dinâmica.

### 2.2.2 Semântica dos dados espaciais

O termo *dado espacial* denota qualquer tipo de dado que descreve fenômenos aos quais esteja associada alguma dimensão espacial. A dissertação considera apenas dados espaciais georreferenciados. Os dados geográficos são geralmente caracterizados a partir de dois componentes fundamentais: atributo e localização. O componente *atributo* (também chamado atributo convencional ou atributo não-espacial) descreve as propriedades temáticas de uma entidade geográfica, tais como o nome. O componente *localização* informa a localização espacial do fenômeno, ou seja, o seu georreferenciamento, associada a propriedades geométricas e topológicas. Por exemplo, a região da bacia amazônica possui atributos descritivos tais como nome, clima dominante, mas também os atributos espaciais que determinam sua posição geográfica, sua extensão e o seu formato.

Muitos SGBDs comerciais como Oracle Spatial, Geodetic Datablade da Informix e DB2 Extensão Espacial da IBM já suportam dados espaciais.

#### Operadores Espaciais

Os operadores mais usuais de busca por dados convencionais são os de comparação entre valores (maior, menor, e igual). Já os operadores temporais são relacionados a comparações entre marcas de tempo (OVERLAP, MEETS). Os operadores espaciais são bem mais complexos, envolvendo frequentemente relações geométricas. Segundo [Güt94], os relacionamentos espaciais são classificados em métrico, de orientação, e topológico.

Os relacionamentos *métricos* geram, a partir de um ou mais objetos geográficos, um escalar que representa uma propriedade intrínseca aos objetos geográficos analisados. Estes relacionamentos podem ser classificados em:

- inter-objetos: baseado principalmente no cálculo de um valor escalar usando mais de um objeto como, por exemplo, a distância.
- intra-objetos: baseado principalmente no cálculo de área, perímetro ou comprimento de um objeto.

Os relacionamentos de *orientação* descrevem onde os objetos estão posicionados em relação a outros. Para este fim, é preciso um marco de referência, um objeto de referência e o objeto em questão. O marco de referência é a orientação que determina a direção na qual o objeto em questão está localizado em relação a um objeto de referência.

Segundo Cilia [Cil96], há várias alternativas para a definição dos relacionamentos de orientação entre dois objetos, que dependem da quantidade de pontos que os representam. Por exemplo, quando um objeto poligonal é representado por um ponto, pode-se utilizar o centróide ou mediana. Por outro lado, quando um objeto é representado por um polígono, pode-se usar o limite inferior esquerdo e o limite superior direito do seu MBR (Minimum Bounding Rectangle).

Normalmente consideram-se quatro primitivas de orientação: *norte*, *sul*, *leste* e *oeste*. Com base nestas primitivas, são definidos formalmente outros relacionamentos, por exemplo acima, abaixo, e mesma posição.

Relacionamentos *topológicos* representam o relacionamento de intersecção e disjunção das fronteiras e do interior entre dois objetos geográficos. [ESM93] apresenta uma teoria que provê um estudo completo sobre as restrições topológicas binárias entre regiões sem buracos, baseada em conceitos algébricos e na teoria dos conjuntos. O estudo compara as intersecções das fronteiras e interiores de duas regiões. Os resultados válidos para estas intersecções são *vazio* ou *não vazia*. Existem 16 possíveis combinações para estes relacionamentos. Destas combinações, oito não são válidas e duas são simétricas, resultando assim seis relacionamentos válidos entre regiões: *disjoint*, *in*, *touch*, *equal*, *cover* e *overlap*. Este método é conhecido como método das 4-intersecções.

Estes trabalhos não consideraram a dimensão da intersecção, mas tão somente se esta é vazia ou não. O trabalho de Clementini et al. [CFvO93]. A dimensão da intersecção pode ser *vazia* (null), pontual (dimensão 0D), linha (1D) ou região (2D), gerando um total de 256 relacionamentos, dos quais somente 52 são aplicáveis. Ele mostrou que os 52 relacionamentos podem ser especificados pela combinação de operadores de fronteira e os relacionamentos topológicos mutuamente exclusivos *disjoint*, *touch*(ou *meet*), *overlap*, *inside* e *cross*.

Esses conceitos serão utilizados no Capítulo 4 para a descrição de operadores móveis.

## 2.3 Espaço e Tempo

O título desta seção lembra a teoria de Einstein que descreve a relação entre o espaço e o tempo. É necessário esclarecer que a seção não pretende tratar da teoria de relatividade, mas lembrar que as entidades espaço e tempo são estritamente relacionadas entre si. Os acontecimentos, a história do mundo são relacionamentos de espaço e tempo, o tempo e espaço não podem existir isoladamente. Perguntas frequentes como "O que? Onde? Quando?" mostram que para uma informação ser completa precisa considerar espaço e tempo juntos. Tais fatores serão separados no texto para facilitar o estudo do efeito de cada variável nos fenômenos do mundo.

Domínio Espacial	Domínio Temporal
campo	linha de tempo
Objetos possuem posição no espaço	Objetos (marcas de tempo) existem no tempo
Posições de objetos podem ser pontos, linhas, regiões	Marcas de tempo no tempo podem ser instantes de tempo, intervalo de tempo
Objetos espaciais são ortogonais a campos	Objetos temporais e linhas de tempo são ortogonais

Tabela 2.1: Analogias entre tempo e espaço

### 2.3.1 Analogias entre espaço e tempo

Segundo Pfooser [Pfo00], o espaço possui atributos espaciais que são representados como campos. A extensão dos objetos dentro de um campo é apresentada como conjuntos de pontos, linhas e polígonos, com valores associados que são valores de atributos espaciais apresentados. Objetos espaciais ocupam uma posição no espaço. Objetos espaciais são ortogonais a campos. Analogamente, o tempo é representado em linhas de tempo. Na verdade, podemos considerar linhas de tempo com granularidades e marcas de tempo diferentes para cada ambiente de aplicação. Eventos e estados posicionam-se na linha de tempo como instante de tempo e intervalo de tempo, respectivamente. Objetos são ortogonais a linhas de tempo. A tabela 2.1 resume as principais analogias entre o espaço e tempo:

### 2.3.2 Combinação de espaço e tempo

Integrar as dimensões temporal e espacial aumenta a complexidade de sua representação e de gerenciamento. Em sistemas que lidam com objetos móveis, estas dimensões são intrinsecamente relacionadas. Portanto, é imprescindível o entendimento de bancos de dados espaço-temporais antes de tratarmos de objetos móveis, que são alvo da presente dissertação.

Um *objeto espaço-temporal* é um objeto que possui pelo menos um atributo espacial e um temporal. Os outros atributos são chamados de atributos convencionais ou descritivos (que descrevem as propriedades temáticas do objeto). Tanto os atributos convencionais quanto os espaciais podem variar com tempo. Considere por exemplo uma fazenda com atributos “Nome”, “Localização” e “Tempo”. O atributo “Localização” é composto por um conjunto de coordenadas que situa a fazenda e, ao mesmo tempo, permite inferir sua

geometria. Tanto nome quanto a geometria podem variar ao longo do tempo. Um *objeto espacial* é um objeto que possui pelo menos um atributo espacial, contendo localização - no mesmo exemplo, o atributo “Tempo” não existe. Um *objeto temporal* é um objeto que possui pelo menos um atributo temporal, que pode ser unidimensional (instante de tempo) ou bidimensional (intervalo de tempo). No mesmo exemplo, o atributo “Localização” deixaria de existir.

Bancos de dados espaço-temporais têm sido alvo de muitas pesquisas recentemente - vide, por exemplo, [Peu01]. Muitos tópicos ainda continuam em aberto. Há várias razões para isto, baseadas principalmente no fato que tanto espaço quanto tempo são fatores complexos e sua captura, armazenamento e gerenciamento envolvem muitos desafios.

### Sistemas espaço-temporais

O termo sistemas *espaço-temporais* é uma designação abrangente que referencia uma grande variedade de sistemas que lidam com a evolução temporal de fenômenos espaciais.

Segundo Pfooser [Pfo00], podemos considerar combinações de um objeto espacial com instantes de tempo e com intervalos de tempo. Armazenar o estado de um objeto espacial num instante de tempo resulta no chamado *snapshot*. O termo *snapshot* significa uma captura da posição de um objeto espacial em um instante de tempo. Armazenar o estado de um objeto espacial num intervalo de tempo é capturar a evolução do objeto no tempo. Dependendo do tipo de mudança, diferentes conceitos espaço-temporais são envolvidos:

- se a mudança for contínua, o *movimento* dos objetos é capturado. Por exemplo, um carro que muda sua posição continuamente está se *movendo*.

- se a mudança for discreta, a *mudança* dos objetos nesse intervalo é capturada. Os estados no início do intervalo e o final de intervalo são capturados.

Segundo Moreira[MRSS01], a classificação dos sistemas espaço-temporais pode ser baseada nas propriedades dos objetos armazenados e gerenciados pelo sistema. Dois aspectos devem ser considerados: a mudança das propriedades espaciais dos objetos e a maneira como essas propriedades são modificadas ao longo do tempo.

Propriedades espaciais que se modificam conforme o tempo são *localização, direção, tamanho e forma*. As propriedades de localização e direção capturam o movimento dos objetos. Essa evolução pode ser modelada em transformações geométricas como translação e rotação. A evolução de tamanho e forma descreve a mutação dos objetos espaciais. A evolução de tamanho captura variações em escala, i.e. expansão e contração dos objetos, enquanto que variação de forma captura informações sobre deformação. As deformações mais complicadas são as que envolvem agregação ou fragmentação de objetos complexos.

A maneira como essas propriedades são modificadas conforme o tempo pode ser representada segundo três tipos, apresentados a seguir:

- Mudança contínua:

Os atributos espaciais com variação contínua podem ser modelados com um gráfico em função de tempo representado por uma curva contínua.

- Registro de eventos:

Quando os valores dos atributos espaciais são alterados em momentos precisos, a mudança é dita descontínua. O registro de eventos é um modelo usado para registrar essas mudanças descontínuas dos atributos espaciais. Ele registra fatos num intervalo válido de tempo de um evento. Fora dos intervalos, os atributos são indeterminados ou recebem algum valor padrão. Esse modelo geralmente utiliza gerenciamento de versões para tratar cada registro de evento como uma versão daquele objeto num determinado intervalo. Registros de eventos são normalmente considerados em banco de dados espaço-temporais.

- Mudança de estado:

Mudança de estado é semelhante ao registro de eventos. A única diferença é que o valor dos atributos espaciais é mantido constante até a ocorrência do próximo evento.

A complexidade dos sistemas espaço-temporais varia conforme o número de propriedades físicas envolvidas assim como o requisito da taxa de atualização dos dados. Sistemas que trabalham com objetos móveis se preocupam apenas com a localização e a direção dos objetos, por exemplo, rastreamento de veículos. Sistemas que tratam de mudanças discretas se concentram em monitorar a deformação (mudança em tamanho e forma), por exemplo, evolução da geometria de uma ilha com a erosão. Já sistemas como monitoramento de áreas de poluição podem lidar tanto com movimento quanto com a deformação. Existem vários estudos sobre a evolução de formas e de tamanhos [EAT92], mas esta dissertação não detalhará esses estudos pois apenas o movimento é o interesse principal.

O grau de continuidade das mudanças de propriedades dos objetos espaço-temporais depende da aplicação. Cada sistema tem seu requisito da taxa de atualização dos dados. Por exemplo, uma empresa de monitoramento de veículos pode necessitar da atualização de hora em hora da posição de caminhões em estradas, mas a posição em cada instante é importante para o sistema, ou seja, não pode haver lacunas entre uma coleta e outra de dados. Os dados devem aparecer como contínuos mesmo que a coleta não seja feita infinitas vezes. Já uma aplicação de controle de limite de propriedade do Ministério de Agricultura coleta os dados em determinadas épocas, pois as mudanças de limites espaciais são muito menos frequentes e descontínuas. Enfim, a escolha entre mudanças contínuas e descontínuas deve ser determinada conforme a natureza do objeto, da aplicação e da

análise custo das atualizações de dados. A barreira entre mudança contínua e descontínua é dependente da aplicação.

A seguir, os objetos móveis são apresentados como uma especialização dentro de sistemas espaço temporais.

## 2.4 Objetos Móveis

No mundo real, somos rodeados por objetos que se movimentam, desde os objetos naturais como animais ou vento até objetos artificiais como carros, navios e satélites. Esses objetos, apesar das diferenças que possuem, todos são móveis.

Um *objeto móvel* é qualquer objeto, pontual ou com extensão, que muda sua posição geográfica *continuamente* conforme o passar do tempo. Como um objeto móvel possui atributos espaciais e temporais, ele pode ser considerado especialização de um objeto espaço-temporal.

Por exemplo, um táxi pode ser modelado como um objeto pontual cuja localização varia com o tempo. Já um cardume de golfinhos pode ser modelado como um objeto 2D. No último caso, não só a localização, mas também a geometria do conjunto de golfinhos varia. Objetos móveis não pontuais podem sofrer rotação. Portanto, além da velocidade linear, pode-se calcular também a velocidade angular de um objeto.

### 2.4.1 Caracterização de sistemas de objetos móveis

Os objetos móveis podem ser caracterizados por cenários de movimento e pela maneira como seus dados são adquiridos e utilizados.

#### Infra-estrutura

Geralmente, as aplicações que lidam com objetos móveis podem ser agrupadas de acordo com o cenário de movimento. Segundo Pfoser [Pfo00], existem três cenários: movimentos livres (jogadores num campo de futebol, variação de poluição num lago), movimentos limitados (carros trafegando, pessoas andando nos prédios) e movimentos numa rede. A última categoria é um caso especial de movimentos limitados que considera o posicionamento relativo ao espaço limitante. Por exemplo, no caso de carros, só há interesse no seu posicionamento em relação à rede de rua, ao invés das suas coordenadas absolutas. Na verdade, no entanto, todo movimento tem alguma limitação.

Objetos que limitam o movimento de outros são chamados de *infra-estrutura*. Para objetos móveis do tipo veicular, exemplos de infra-estrutura são prédios, lagos e zonas de pedestre, mas também bloqueios de tráfego ou tráfego em baixo movimento. Infra-estruturas podem ser subdivididas em simples e dinâmicas.

Infra-estruturas simples são objetos espaciais que não mudam conforme o tempo. Infra-estruturas dinâmicas são elementos que podem aparecer e desaparecer (tais como bloqueios de tráfego) ou que mudam (tais como fluxo de tráfego em baixo movimento).

### Método de Aquisição de dados

Existem basicamente dois tipos de método de aquisição de dados para aplicações móveis:

- detecção automática da mudança de direção de velocidade, chamados de sistemas orientados a eventos.
- captura de dados de localização numa sequência ordenada de pontos discretos no tempo, chamados de sistemas de monitoramento, em que a captura é baseada em GPS.

O GPS (Sistema Global de Posicionamento) nos permite determinar posições na Terra em qualquer hora, qualquer lugar, e em qualquer condição de tempo. O sistema consiste em 24 satélites que orbitam em torno da Terra acima de 20000 km. Os satélites transmitem sinais que podem ser detectados por receptores de GPS, que são capazes de determinar suas localizações com precisão. O princípio do funcionamento do GPS é a medida de distância entre o receptor e vários satélites. Um total de quatro distâncias, de quatro satélites, são necessárias para resolver um conjunto de quatro equações que expressam latitude, longitude, altura e tempo. A maior parte dos GPS utiliza de 3 a 6 satélites. A distância do satélite ao receptor pode ser calculada multiplicando o tempo que leva o sinal para chegar ao receptor pela velocidade da luz.

### 2.4.2 Representação de objetos móveis em banco de dados

Como os objetos móveis podem ter dimensões variáveis, eles podem ser divididos em basicamente dois tipos:

#### Pontos móveis

Muitos objetos móveis não requerem toda a funcionalidade de objetos espaço-temporais. Para uma categoria específica de objetos móveis, tamanho e forma não importam. Nessas condições, apenas a localização do objeto em determinado tempo é importante para o sistema. Tais objetos podem ser simplificados como *pontos móveis*.

Exemplos desse tipo de aplicação são os sistemas que lidam com posições de carros ou animais, para usos de controle de tráfego, monitoramento de atividades de pesca e de proteção a animais. Um exemplo real é o projeto denominado de "Sistemas Avançados de Transporte" da Caltrans na Califórnia. Nessa aplicação, os veículos equipados com GPS

transmitem suas posições para um computador central usando comunicação por rádio ou celular. Na central o dado é processado para controlar fluxo de transporte [cal].

### Regiões móveis

Pontos móveis são casos particulares de uma abstração genérica das denominadas *regiões móveis*. Uma região móvel permite a captura do movimento e das mudanças de forma e tamanho de um objeto. Além disso, mesmo que a região móvel não considere a mudança de forma e tamanho, ela considera a extensão do objeto quando essa propriedade não pode ser ignorada. Por exemplo, o tamanho do avião não pode ser desconsiderado quando se deseja saber se haverá colisão de aviões. A representação de região móvel também possibilita obtenção de orientação e de rotação. Segundo Moreira [MRSS01], até hoje, o estudo de representação de regiões móveis é apenas no nível de modelo de dados abstratos. Não existem propostas de modelos de dados físicos que sejam adequados para representar regiões móveis. As operações geométricas se tornam muito mais complexas e isso requer soluções a serem especializadas para cada domínio de aplicação, levando em conta particularidades da aplicação.

Os sistemas de monitoramento ambiental muitas vezes lidam com objetos heterogêneos com dimensão superior a um. Exemplo desses sistemas são evolução de mancha de poluição ou migração de grupos de animais. Em geral, são sistemas que precisam saber o tamanho e a forma do objeto móvel além da sua posição.

### 2.4.3 Desafios de bancos de dados de objetos móveis

Um *banco de dados de objetos móveis* é uma especialização de bancos de dados espaço-temporais. A diferença é que os objetos móveis modificam continuamente a sua posição. Essa diferença apresenta novos desafios. Primeiramente, o número de registros cresceria muito se cada posição exigisse o armazenamento um registro novo. Em segundo lugar, a complexidade algorítmica dos operadores é muito maior pois precisam considerar a continuidade do movimento. Em terceiro lugar, pode haver intervalo de tempo sem informação amostrada. Isso pode apresentar "lacunas" de informações, o que exige técnicas de "reconstituição" da continuidade, usando técnicas de aproximação tais como interpolação. Esses novos conceitos exigem uma modelagem de dados nova para representá-los adequadamente e que seja ao mesmo tempo viável para fins computacionais.

### 2.4.4 Modelos de dados para objetos móveis

Existem duas frentes de pesquisa em modelagem de objetos móveis. Uma das vertentes está orientada a aplicações altamente dinâmicas e as consultas visam o presente e o futuro

próximo. A outra está mais orientada às aplicações em que o enfoque das consultas é sobre o histórico do objeto. Esse tipo de aplicação geralmente é usado para extrair informação de padrão de comportamento, ou seja, mineração de dados. Em outras palavras, uma vertente cuida de dados que alimentam o banco de dados em tempo real, enquanto que a outra examina o conteúdo armazenado, mas sem a preocupação de seu monitoramento em tempo real.

### Sistemas de tempo real e o Modelo MOST

Sistemas de tempo real para objetos móveis preocupam-se com a antecipação dos eventos do futuro, fornecendo respostas para perguntas do tipo “Quais são os postos de gasolina mais próximos de um objeto móvel?” ou “Quais são os taxis que entrarão na região central dentro de 5 minutos?”

Sistemas desse tipo envolvendo objetos móveis são chamados de sistemas MOD (Mobile Object Database). Para contornar o problema de atualização de dados, Wolfson et al [SWCD97] propuseram o modelo MOST (Moving Objects Spatial Temporal). Junto ao modelo, é apresentado uma linguagem de consulta, chamada de Lógica Temporal Futura (FTL). Essa linguagem traz uma sintática específica para lidar com atributos que mudam continuamente no tempo e o protótipo DOMINO implementa essas idéias. A idéia básica do modelo é representar a posição geográfica do objeto pontual usando um vetor de movimento. Todo objeto espacial tem coordenadas  $x$ ,  $y$  e  $z$ . Cada coordenada ( $x$ ,  $y$  ou  $z$ ) é representada por um atributo dinâmico.

Um atributo dinâmico  $A$  é um atributo composto por três campos:

$\langle A.value, A.updatetime, A.function \rangle$ ,

onde  $A.updatetime$  é o instante em que o vetor de movimento é atualizado.  $A.value$  é o valor da coordenada no instante da atualização ( $A.updatetime$ ).  $A.function$  é a função que descreve o valor de  $A$  para cada  $t$  depois do  $A.updatetime$  e antes da próxima atualização. Se  $V(t)$  é o valor da coordenada no tempo  $t$ , então:

$$V(t) = A.value + A.function(t), \forall t \in (A.updatetime, \infty]$$

O modelo MOST baseia-se na hipótese que é possível atualizar o vetor de movimento automaticamente quando ocorrer uma mudança na velocidade e na direção.

A linguagem de consulta FTL é uma extensão para ser usada sobre um SGBD e permitir consultas sobre estados futuros. As respostas para essas consultas são tentativas; baseiam-se no conhecimento do mundo daquele estado, mas esse conhecimento pode mudar. Dois operadores básicos usado no FTL são: *Until* e *NextTime*. Considere dois estados  $f$  e  $g$ . A formula *f Until g* é satisfeita se o  $f$  for satisfeito naquele estado ou em algum estado futuro enquanto  $g$  for satisfeito. A fórmula *NextTime f* é satisfeita num estado, se  $f$  for satisfeito no estado seguinte do sistema. Outros operadores como *Eventually* e *Always* são composições dos operadores básicos.

O modelo MOST introduz um intervalo de incerteza nos atributos dinâmicos. Especificamente, o intervalo de incerteza do objeto móvel é limitado pela infra-estrutura na qual ele se movimenta. Por exemplo, numa infra-estrutura fixa de vias de transporte, considera-se apenas uma seção da via. Se o movimento for livre, considera-se um círculo. Baseado nesse intervalo de incerteza, as variantes de consulta *May* e *Must* (*pode e deve*) são introduzidas. Estas palavras-chave permitem fazer a consulta "selecionar todos objetos que estão dentro do polígono P". A semântica de *May* retorna os possíveis objetos que podem/devem estar dentro do polígono P. Já *Must* permite solicitar o conjunto de objetos que certamente *Must* estão no polígono P.

No MOST, no pior caso, o número de atualizações é igual ao da abordagem tradicional. Como a maioria dos movimentos permanece uniforme por um tempo, esta abordagem torna-se uma solução interessante. Outra vantagem do modelo é que consegue satisfazer consultas sobre o futuro próximo. Uma questão importante a ser considerada é o balançamento entre o custo de atualização e a imprecisão das posições.

A desvantagem desse modelo é que ele não descreve a trajetória completa dos objetos móveis. A informação de trajetória só é armazenada desde o tempo da última atualização até o futuro próximo. Além do mais, só trata de objetos pontuais e ignora estruturas espaciais mais complexas como regiões. O atributo posição é agora um atributo derivado pois o valor de cada coordenada é calculado através das definições de  $V(t)$ . Isso gera um custo a mais ao fazer consultas cujo predicado envolve a posição.

## Sistema de Monitoramento e Modelo do projeto CHOROCHRONOS

Sistemas de monitoramento preocupam-se com a análise posterior dos eventos, fornecendo respostas para consultas como "Quais navios pesqueiros foram proibidos numa região" ou "Quando um navio pesqueiro entrou num porto?"

O grupo CHOROCHRONOS [EGSV97] propôs um modelo conceitual de dados baseado em definir os tipos base de dados espacial e temporal, e utilizar construtores espaciais para construir outros tipos de dados. No modelo conceitual específico para objetos móveis, os tipos base são *ponto*, *região* e *tempo*. A nomenclatura usada nas descrições seguintes abrevia a palavra "móvel" com a letra 'M'. O tipo **Mponto** mapeia um valor de tempo  $t$  do domínio tempo para um ponto  $P$  do domínio espaço. O tipo **Mregião** mapeia um valor de tempo  $t$  do domínio tempo para uma região  $R$  do domínio espaço. As funções que mapeiam entre os tipos são chamadas de construtores. Os valores do tipo **Mponto** descrevem a posição como uma função de tempo, que pode ser representado como uma curva no espaço tridimensional  $(x, y, t)$  e o valor do tipo **Mregião** é um conjunto de volumes no espaço 3D  $(x, y, t)$ . A tabela 2.2 apresenta exemplos de operadores.

O modelo conceitual do CHOROCHRONOS é mais genérico e abrangente que o modelo MOST, por considerar regiões e não apenas pontos móveis. A desvantagem desse modelo

Operador	Mapeamento
Começa	Mponto $\mapsto$ tempo
Visita	Mregião x Mponto $\mapsto$ Mponto

Tabela 2.2: Exemplos de operadores

é que ele é abstrato demais e se preocupa com os objetos móveis em si, esquecendo os objetos espaço-temporais que mudam discretamente.

Há vários outros aspectos considerados na pesquisa de bancos de dados de objetos móveis e que não são abordados nesta dissertação. Exemplos são estudos de eficiência de armazenamento [CAA01], indexação [SJLL99] e gerenciamento de objetos armazenados.

Um exemplo concreto de implementação é o trabalho de Moreira [SM01], que implementou um sistema de monitoramento de navio pesqueiro. Ele se concentrou em estudar basicamente duas áreas: geração de dados de objetos móveis para simulação; e como lidar com incertezas em banco de dados que não possuem dados suficientes mas precisam "simular" a continuidade. O artifício usado para incerteza é definir uma área de possibilidade baseada na velocidade máxima do objeto. Ele também define semânticas de incerteza nas consultas: *Possível*, *Certamente* e *Provável*.

## 2.5 Resumo

Este capítulo apresentou os conceitos básicos de dados temporais, espaciais e espaço-temporais assim como dos bancos de dados que os implementam. Muitos pesquisadores têm tentado fazer convergir o estudo de tempo com o espaço, já que ambos apresentam semelhanças e são complementares em termos de representação dos fenômenos naturais.

Os sistemas espaço-temporais foram classificados segundo uma série de critérios. Uma classe apresenta natureza altamente dinâmica, contínua, caracterizando sistemas de objetos móveis. Essas características trazem desafios aos bancos de dados tradicionais. Existem duas frentes de propostas de modelagem: orientados a sistemas de monitoramento e de tempo real. Suas vantagens e desvantagens foram analisadas neste capítulo.

O artigo [Peu01] apresenta uma análise da comunidade de pesquisa de sistemas espaço-temporais incluindo objetos móveis. Segundo o artigo, tanto os fenômenos discretos quanto contínuos envolvem objetos espaciais e temporais que possuem características de parte-de ou é-um, o que corresponde muito bem à visão de modelos orientados a objeto. No entanto, um modelo de dados orientado a objeto que possa englobar essas duas facetas de sistemas espaço-temporais ainda é inexistente.

Em função dessa necessidade, a presente dissertação apresentará no próximo capítulo

um modelo de dados orientado a objetos que trata tanto de objetos de mudança discreta quando de mudança contínua.

---

# Capítulo 3

## Modelo de dados para objetos móveis

Este capítulo descreve o modelo de dados proposto para objetos móveis. O modelo explora os conceitos de objetos móveis, reduzindo um sistema originalmente 3D para 2D, visando representá-los de maneira que facilite as consultas. O modelo proposto é orientado a objetos e trata uniformemente os objetos de mudança discreta e contínua. Vale lembrar que na literatura correlata, o estudo sobre objetos móveis é restrito a objetos pontuais móveis [MRSS01, EGSV97, SWCD97]. O modelo aqui proposto, ao contrário, considera também objetos lineares e poligonais.

Um objeto móvel é composto de atributos convencionais e um componente espaço-temporal. Um objeto móvel genérico pode ser descrito por:

$$O = \langle \text{atributos convencionais, componente espaço-temporal} \rangle$$

Embora o modelo permita modificação de atributos convencionais ao longo do tempo, isto não será considerado na dissertação, pois foge ao assunto de mobilidade. A mudança de atributos convencionais ao longo do tempo é um assunto já bastante discutido na literatura e em nada contribuirá à questão de mobilidade. Desta forma, de agora em diante, apenas os componentes espacial e temporal serão discutidos na dissertação. Subentende-se que sempre haverá possibilidade de variação de componentes convencionais.

Este capítulo inicialmente discute os conceitos básicos e, a seguir, apresenta o modelo.

### 3.1 Conceitos básicos

Partindo do princípio que um banco de dados deve armazenar informações que possam ser consultadas, deve-se perguntar que tipo de consultas devem ser oferecidas por um banco de dados de objetos móveis. Em seguida, os conceitos de movimento e trajetória serão explorados pois têm grande importância em relação ao tipo de informação armazenada

em banco de dados de objetos móveis.

### 3.1.1 O que se consulta sobre objetos móveis

No contexto de banco de dados de objetos móveis, vamos considerar que os objetos são de três tipos, de acordo com o seu comportamento temporal: estático, temporal discreto e temporal contínuo. Tratando-se de objetos móveis, o interesse maior são as consultas sobre a história do movimento do objeto, considerando o passado ou evolução futura.

Classificando as consultas de acordo com os principais atributos que são tempo e espaço, elas basicamente são divididas em três tipos, de acordo com o predicado considerado: consultas temporais, espaciais e espaço-temporais. Sejam as seguintes consultas:

- Consulta 1: No tempo  $t_1$ , onde estava (coordenada  $x, y$ ) o carro A?
- Consulta 2: Quando o carro A estava na coordenada  $x_1, y_1$ ?
- Consulta 3: No tempo  $t_1$ , o carro A estava fora da região central (demarcada pelo polígono X)?
- Consulta 4: Quando o carro A estava fora da região central (demarcada pelo polígono X)?
- Consulta 5: Qual é a distância entre os carros A e B no tempo  $t_1$ ?
- Consulta 6: Determine o intervalo de tempo em que os carros A e B mantiveram uma distância de 3km?
- Consulta 7: Quando o carro A entrará na região central?
- Consulta 8: Quando os carros A e B se encontrarão?
- Consulta 9: Quando um objeto sofreu rotação e qual é a sua velocidade angular?

As consultas 1 e 2 são consultas típicas em bancos de dados espaço-temporais, visando um único objeto. Consultas mais complexas envolvem a interação entre os objetos, variando no tempo e espaço.

Observe que as consultas 3 e 4 tratam do relacionamento entre um objeto móvel e um objeto estático (região central). As consultas 5 e 6 tratam de relacionamento métrico entre dois objetos móveis. As consultas 7 e 8 são deduções sobre o futuro baseados em dados históricos de cada objeto móvel. A consulta 9 envolve um objeto não pontual.

Os exemplos anteriores dão uma idéia básica de alguns tipos de consultas que um SGBD com objetos móveis deveria suportar. Uma descrição mais detalhada e completa sobre as consultas de objetos móveis será apresentada no Capítulo 4.

### 3.1.2 Movimento e Trajetória

O *movimento* consiste num relacionamento entre espaço e tempo. Não existe conceito do movimento apenas com espaço, pois sem tempo não se tem memória, e sem memória, não se percebe a mudança. Se o espaço representasse os estados do nosso mundo, o tempo seria uma espécie de índice para estados diferentes desse mundo. Podemos observar o papel de índice do tempo pelos exemplos de consultas, pois as consultas 1 e 3 retornam só um resultado, enquanto que as consultas 2 e 4 podem retornar um conjunto de resultados. Ou seja, dado um tempo, só existe um estado de relacionamento espacial entre dois objetos, mas esse mesmo estado pode acontecer em vários instantes. O movimento de um objeto é a história da mudança da sua posição. O retrato dessa história é a *trajetória* do objeto. Dessa maneira, pode-se considerar que a trajetória é a definição do movimento armazenado como uma composição particular de tempo e espaço.

A *trajetória* é o registro do movimento. Ela demonstra a evolução da posição de um objeto móvel. Observando o rastro de um caramujo, pode-se saber algumas informações sobre o seu movimento. Sabe-se que ele saiu de um ponto de coordenada  $x_1$  e  $y_1$  para um outro ponto de coordenada  $x_2$  e  $y_2$ , e o rastro indica um deslocamento espacial que liga os dois pontos. Ou seja, pela simples observação do rastro, pode-se perceber a localização espacial do caramujo, mas não se sabe quando cada pedaço da marca é feita. Mas sempre é possível marcar o tempo inicial e o tempo final para o ponto inicial e final do rastro. Não se pode esquecer, no entanto, que o movimento do caramujo pode não ser uniforme durante esse intervalo. Uma solução aproximada seria dividir o rastro em intervalos pequenos suficientes para garantir que cada intervalo contempla movimentos uniformes e a aceleração ocorre no início de cada pequeno intervalo. Essa solução apresenta o inconveniente que o número desses intervalos pequenos pode ser grande demais. Nesse contexto, pode-se acrescentar um valor de tolerância para diminuir o número de divisões. O caramujo pode até voltar para trás pelo mesmo caminho. Isso não seria problema pois ele pode estar "equipado" com o GPS que monitora sua posição. Assim, mesmo que o trajetória seja sobre uma mesma reta, ela é registrado temporalmente.

A figura 3.1 ilustra a trajetória de objetos móveis. A trajetória de um objeto pontual é um conjunto de pontos que formam uma reta; a trajetória de um objeto linear é um conjunto de linhas que formam um polígono; e a trajetória de um polígono é um conjunto de polígonos que formam um poliedro, cuja projeção no plano espacial 2D é também um polígono. Se o objeto ficar em repouso por um intervalo de tempo, sua trajetória pode ser representada pela geometria do próprio objeto. Mais detalhes sobre este enfoque de redução de dimensão estão na seção 4.3.5.

Dessa maneira, a projeção do movimento no plano espacial pode ser vista como um objeto geométrico (pontos, linhas e polígonos). Conseqüentemente, uma operação espaço-temporal sobre um objeto móvel pode não precisar envolver o atributo temporal. Isso pode

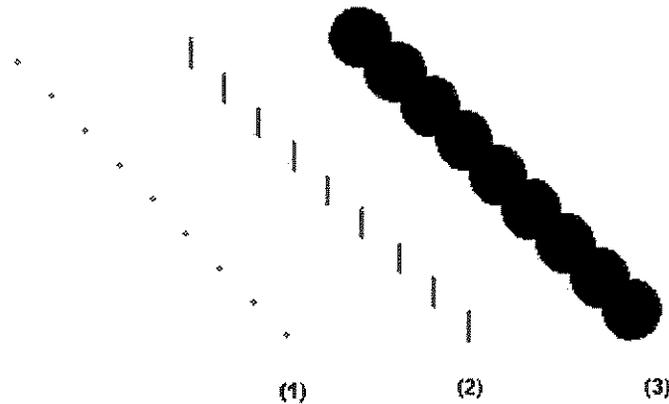


Figura 3.1: Trajetórias traçadas por ponto, linha e polígono.

ser vantajoso, pois alguns operadores podem ser implementados de maneira similar aos operadores estritamente espaciais.

### 3.1.3 O Modelo de Dados Geográfico Temporal de Faria

O modelo proposto na dissertação é baseado no modelo espaço-temporal proposto pela dissertação de Faria[Far98]. Este último é um modelo de dados geográfico-temporal que não considera apenas a visão estática, mas também a visão de *mudanças*. Em outras palavras, não somente os fatos geográficos são modelados, mas também os processos de mudança geográfica.

O espaço geográfico é modelado como um conjunto de geo-objetos que se relacionam, cada um com seu estado (atributos) e comportamento (métodos). As restrições de integridade sobre os objetos geográficos são formalizadas como métodos.

No modelo, um processo geográfico é representado por um objeto geográfico discreto composto, cuja lista de geo-objetos componentes contém as entidades geográficas participantes deste processo, e cujos atributos contém os valores (ou intervalos de valores) que caracterizam a transição de estado dessas entidades. A transição de estado de objeto geográfico composto é representada como um intervalo temporal, pois cada um dos componentes é modificado em tempos distintos.

Fazendo um paralelo com os modelos descritos por Moreira[MRSS01], pode se observar que o modelo de Faria pode ser enquadrado no modelo de *mudança de estado* citado no Capítulo 2.

A figura 3.2 apresenta a descrição das classes do modelo de Faria em notação OMT, composta de objetos espaciais, temporais e espaço-temporais, detalhados a seguir.

### Objetos espaciais

A classe SpatialObject representa os geo-objetos atemporais. O componente espacial de um SpatialObject (SP) é definido por uma lista de objetos da classe Geom. A classe Geom é composta por atributos não-espaciais para guardar informações específicas da representação (tipo de projeção, escala) e um atributo espacial para guardar objetos geométricos que definem a geometria da representação:

Sp: list (Geom)

Geom: <atributos convencionais, list(Obj\_Geom)>

O componente espacial da classe Geom é uma lista de objetos da classe abstrata: Obj\_Geom. Esta é uma generalização das classes geométricas básicas: Ponto, Linha e Polígono. A classe Polígono é composta por uma lista de linhas. A classe Linha é composta por uma lista de pontos.

### Objeto Temporal

A classe TempElement é composta por uma lista de objetos da classe Interval. A classe Interval é composta por uma lista de objetos do tipo Event. Objetos do tipo Event representam instantes (pontos) de tempo.

Objetos temporais (instâncias de TempObject) são aqueles que têm um componente temporal associado (ou seja, objeto da classe Time). As classes derivadas da classe Time (Event, Interval e TempElement) são utilizadas para representar as marcas de tempo associadas aos objetos temporais.

### Objeto Espaço-Temporal

A classe SpatioTempObject representa geo-objetos temporais, tendo assim componentes espaciais e temporais. A sua componente espacial é definida por uma lista de objetos da classe GeomT. Assim como a classe Geom, GeomT corresponde a uma representação espacial do objeto geográfico. O que a torna diferente de Geom é fato de seus atributos serem temporalizados. Segundo Botelho[Bot95], a temporalização consiste em anexar à classe os atributos necessários para armazenar toda a história do objeto e dos seus componentes. As classes geométricas (Geom) são temporalizados em PointT, LineT e PolygonT. Faria salientou que o componente espacial de um objeto do tipo SpatialTempObject não pode ser obtido pela simples composição de objetos das classes Sp e Time. Na composição simples, a localização de um geo-objeto temporal ficaria assim:

< Sp, T >, onde Sp: list (Geom)

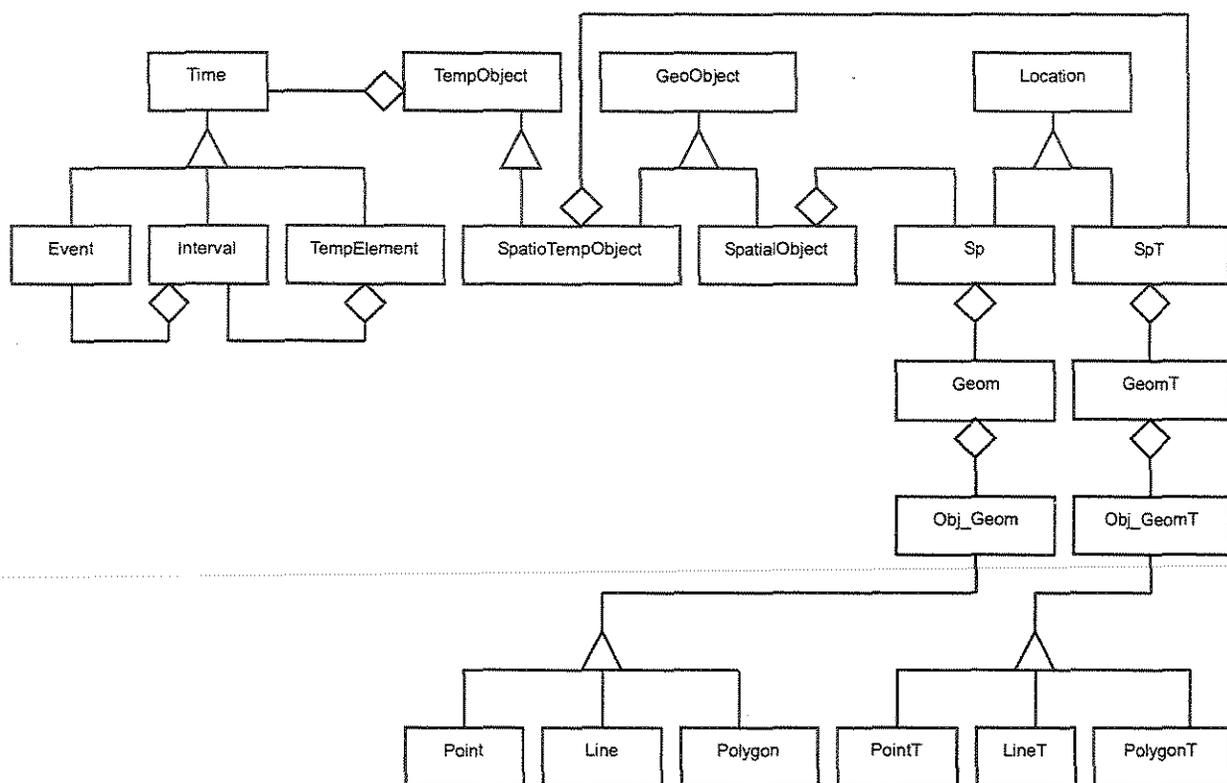


Figura 3.2: Modelo de dados proposto pela Faria.

Essa alternativa não permite que a evolução da localização de um geo-objeto seja tratada isoladamente do geo-objeto em si. Da mesma forma, é necessária a definição da classe `GeomT`. Isso permite ter uma história para cada representação espacial do objeto espaço-temporal.

## 3.2 Apresentação do modelo proposto

### 3.2.1 Idéia básica

O modelo proposto é baseado em modificações do modelo de Faria. A idéia principal é estendê-lo de forma a armazenar as trajetórias dos objetos móveis. Pela figura 3.1 podemos perceber que a trajetória (área varrida pelo objeto) é dependente da geometria do objeto móvel. Além da geometria, é necessário também o tempo do percurso. Caso a velocidade não seja constante no trecho, é preciso ter uma função da variação de velocidade. Desta forma, para descrever o movimento de um objeto móvel é preciso armazenar:

- a geometria do percurso do objeto móvel
- a variação de velocidade armazenada em cada trecho do percurso
- o tempo para percorrer o percurso

---

Os objetos tratados são objetos rígidos (não sofrem fragmentação nem composição nem mudam de tamanho). Além do mais, são objetos 2D.

### 3.2.2 Modelo estendido e classes principais

A figura 3.3 mostra o diagrama de classes proposto para o modelo. Este diagrama estende a proposta de Faria [Far98] apresentada na seção anterior. A extensão consiste numa hierarquia de classes cuja raiz é `SpatioTempObject` (ou seja, objetos espaço-temporais). Como o modelo de Faria é para mudança discreta, esta proposta abrange tanto mudanças discretas quanto contínuas.

Partindo do princípio que um objeto móvel é também um objeto espaço-temporal, ele é apresentado como derivado da classe `SpatioTempObject`. Dessa maneira, o objeto `SpatioTempObject` é especializado em `MotionObject` e `ChangeObject`.

Os objetos espaço-temporais de mudanças discretas eram representados pela classe `SpatioTempObject` no modelo de Faria[Far98]. Como as novas classes de objetos móveis foram introduzidas, a classe `SpatioTempObject` tornou-se classe base. A classe `ChangeObject` passa a representar os objetos de mudança discreta.

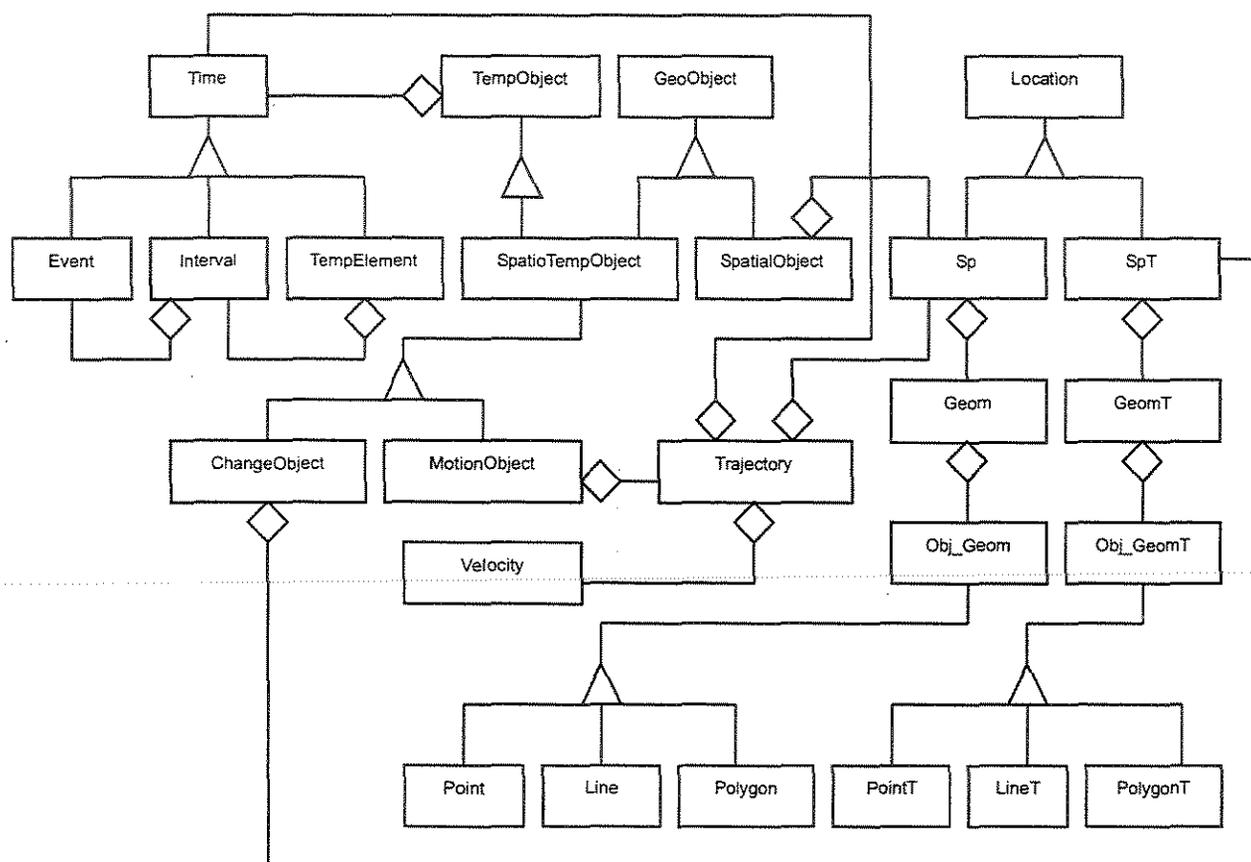


Figura 3.3: Modelo de dados proposto

### 3.2.3 Objetos móveis e as classes derivadas

A classe *MotionObject* descreve as características de um objeto móvel rígido (não mutável) que possui um movimento contínuo. Um objeto móvel rígido é um caso comum para maioria dos objetos móveis: por exemplo, um carro em movimento (objeto pontual) ou um trem andando no trilho (objeto com extensão). Todos componentes desses objetos se locomovem juntos e andam na mesma velocidade. Se a velocidade não variasse, a composição simples de *Sp* e *Time* seria suficiente para representar a trajetória, a geometria e o tempo percorrido dentro de um intervalo de tempo.

O componente espaço-temporal de cada objeto móvel da classe *MotionObject* é composto por vários trechos de trajetórias *Trajectory* que o percurso do objeto traçou ao longo de tempo. Cada trecho de trajetória é capaz de recompor toda história de movimento do objeto naquele trecho. O método *t\_trajectory* do objeto *Trajectory* retorna a geometria da trajetória descrita pelo objeto no plano 2D. Um objeto da classe *Trajectory* é obtido pela composição das classes *Sp*, *Time* e *Velocity*.

A classe *Sp* corresponde à representação geométrica do objeto, composta pelos atributos que descrevem a projeção e a escala do objeto e a geometria em si. Esse objeto geométrico pode ser um ponto, uma linha ou um polígono.

A classe *Time* corresponde a objetos que armazenam o intervalo de tempo gasto pelo objeto para percorrer um trecho de trajetória.

Para descrever a componente que representa o movimento não uniforme (com velocidade variável), foi necessário a criação de uma classe *Velocity*. *Velocity* contém parâmetros de uma função que descreve a variação da velocidade no tempo.

A função *Velocity* foi incluída para diminuir amostras de movimento armazenados, diminuindo, assim, o espaço ocupado. Esta função pode, inclusive, considerar a variação da velocidade angular. Este último caso não é tratado, pois esta dissertação não considera a rotação dos objetos. Esse componente pode deixar de existir caso a velocidade seja constante (movimento uniforme ou objeto em repouso). Em princípio, um objeto móvel deveria ser uma especialização de *SpatioTempObject*. No entanto, a classe *SpatioTempObject* definida no modelo de Faria prevê mudanças de geometria e forma, o que não é contemplado na dissertação. O objetivo da antiga classe *SpatioTempObject* é representar uma entidade geo-objeto que é composta por geo-objetos menores que possuem sua evolução espaço-temporal própria. Isso não é o caso dos objetos móveis rígidos, pois a geometria do objeto móvel não é alterada durante o seu movimento.

```
class MotionObject inherit SpatioTempObject
    list (trajectory: Trajectory)
method
```

```

    t_trajectories (t:Time) /* retorna a lista de trajetórias num
        determinado intervalo de tempo.*/
end;

class Trajectory
    tuple (l: Sp, t: Time, v: Velocity)
method
    t_sp (t: Time) /* retorna a geometria e a posição de um objeto
        móvel num determinado tempo t. */
    geom_trajectory (t:Time) /* retorna a trajetória geométrica descrita
        pelo objeto móvel no plano xy. */
end;

class Velocity type
    tuple (initial_velocity: real,
        acceleration: real)
method
    t_velocity (t:Time) /* retorna a velocidade num determinado tempo
        t.*/
end;

```

---

### Infra-estruturas associadas

Um objeto móvel pode ter seu movimento limitado pela sua infra-estrutura associado. Para associar os objetos móveis a suas infra-estruturas, o modelo permite que cada objeto móvel possua uma lista de objetos da classe GeoObject. Sendo GeoObject a superclasse de SpatialObject e SpatioTempObject, a infra-estrutura é fixa ou dinâmica. No modelo, a infra-estrutura pode ser considerada um objeto estático e interação de um objeto móvel com a infra-estrutura será o equivalente à interação entre dois objetos (móvel e outro estático).

Por associar infra-estrutura, o modelo permite que se possa introduzir uma tolerância de erros quando os dados não sejam completos. Por exemplo, dada a posição de um objeto com movimento livre, a posição provável desse objeto no próximo intervalo de tempo é limitada por uma circunferência onde o raio é a velocidade máxima multiplicada pela duração do intervalo de tempo. Com a infra-estrutura associada, a posição provável é limitada pela composição da geometria da infra-estrutura e a circunferência.

### Trecho da trajetória

Antes de inserir os dados no banco de dados de objetos móveis, é importante definir o que é um "trecho de trajetória". Supõe-se que em cada trecho da trajetória, o objeto não muda de direção e que cada segmento da trajetória é um trecho retilíneo. Desta forma, o movimento de um objeto será armazenado ao longo do tempo da forma  $\langle Sp_1, t_1, v_1 \rangle, \langle Sp_2, t_2, v_2 \rangle, \langle Sp_3, t_3, v_3 \rangle \dots \langle Sp_n, t_n, v_n \rangle$ , descrevendo os componentes espacial, temporal e velocidade a cada instante  $t_j$  amostrado. Um trecho de trajetória corresponde assim ao par:

$$\langle Sp_i, t_i, v_i \rangle, \langle Sp_{i+1}, t_{i+1}, v_{i+1} \rangle$$

onde  $Sp$  pode ter geometria pontual, linear ou poligonal. Se a geometria é pontual, então  $\langle Sp_i, Sp_{i+1} \rangle$  é uma reta. Se linear ou poligonal, sejam  $x_i$  e  $x_{i+1}$  os centróides de  $Sp_i$  e  $Sp_{i+1}$ . Então  $\langle x_i, x_{i+1} \rangle$  é um segmento de reta.

### 3.2.4 Descrição de um objeto móvel

Esta seção decreve o componente espaço-temporal de um objeto móvel no modelo proposto. O componente é descrito para situações de repouso e de movimento dos objetos móveis representados por ponto, linha e polígono.

#### Objeto pontual

Um objeto pontual espaço-temporal de mudança discreta pode ser representado a cada intervalo de tempo por um objeto pontual em repouso. O componente espaço-temporal  $\langle Sp, T \rangle$  é:

$$(\langle x_i y_i \rangle, [t_i, t_f]),$$

onde  $x_i$  e  $y_i$  descrevem a localização, e  $[t_i, t_f]$  descreve o intervalo de tempo em que o objeto permaneceu no mesmo lugar.

Quando um objeto pontual se locomove, o seu movimento contínuo pode ser descrito por uma lista de trechos de trajetória. Cada trecho é representado por pares de tuplas da seguinte maneira:

$$(\langle x_i y_i \rangle, v_i, t_i) (\langle x_f y_f \rangle, v_f, t_f),$$

onde  $x_i y_i$  e  $x_f y_f$  são as posições inicial e final do trecho e  $t_i$  e  $t_f$  denota o intervalo de tempo para percorrer o trecho (objetos da classe Time) e  $v_i$  e  $v_f$  são as velocidades do objeto (objeto da classe Velocity).

#### Objeto linear

O componente espaço-temporal de um objeto linear em repouso pode ser descrito da seguinte maneira:

$$(\langle x_{1i}y_{1i} \rangle, \langle x_{2i}y_{2i} \rangle, [t_i, t_f]),$$

onde os dois primeiros elementos são os pontos inicial e final da geometria e  $t_i$  e  $t_f$  corresponde ao intervalo de tempo em que ele está em repouso.

Quando um objeto linear entra em movimento, cada trecho de sua trajetória pode ser descrito da seguinte maneira:

$$(\langle x_{1i}y_{1i} \rangle, \langle x_{2i}y_{2i} \rangle, v_i, t_i) (\langle x_{1f}y_{1f} \rangle, \langle x_{2f}y_{2f} \rangle, v_f, t_f),$$

Onde  $x_{1i}y_{1i}$  e  $x_{2i}y_{2i}$  denotam a posição inicial da reta no início do movimento e  $x_{1f}y_{1f}$  e  $x_{2f}y_{2f}$  denotam a posição da reta no final do movimento. Os valores  $t_i$  e  $t_f$  denotam o intervalo de tempo para percorrer o trecho e  $v_i$  e  $v_f$  são objetos da classe Velocity.

### Objeto poligonal

De forma similar, um objeto poligonal em repouso pode ser descrito da seguinte maneira:

$$(\langle x_{1i}y_{1i} \rangle, \dots, \langle x_{ni}y_{ni} \rangle, [t_i, t_f])$$

Quando tal objeto poligonal estiver em movimento, a sua trajetória projetada no plano  $xy$  pode ser descrita da seguinte forma:

$$(\langle x_{1i}y_{1i} \rangle, \dots, \langle x_{ni}y_{ni} \rangle, v_i, t_i) (\langle x_{1f}y_{1f} \rangle, \dots, \langle x_{nf}y_{nf} \rangle, v_f, t_f)$$

Note que em todos casos, a posição do objeto no intervalo de tempo entre  $t_i$  e  $t_f$  não é indeterminada, mas calculada através da interpolação baseada na função de velocidade do objeto naquele trecho.

Um problema neste modelo é que armazena a geometria do objeto a cada trecho de trajetória. Em princípio, sendo a geometria rígida, bastaria definir um ponto específico e apenas armazenar a geometria inicial. O restante da trajetória referenciaria apenas tal ponto. No entanto, isso impediria posteriormente de considerar a rotação dos objetos. Por este motivo, optou-se por armazenar toda a geometria a cada trecho de trajetória, apesar de que esta dissertação não trata a rotação.

## 3.3 Resumo

Este capítulo apresentou a modelagem dos objetos móveis como uma especialização de objeto espaço-temporal. O modelo proposto utiliza o conceito de trajetória para descrever a história de um objeto móvel. Utilizando a trajetória, a dimensão tempo é embutida na geometria descrita pela trajetória, o que traz duas principais vantagens:

- representar os dados de forma uniforme junto aos objetos espaço-temporais de mudança discreta;
- facilitar o cálculo dos operadores;

A primeira vantagem tornou possível descrever o modelo proposto estendendo o modelo de Faria. A segunda vantagem será detalhada no capítulo 4.

---

# Capítulo 4

## Operadores Móveis

### 4.1 Introdução

O objetivo deste capítulo é especificar as operações das classes de objetos móveis do modelo proposto no Capítulo 3. Objetos móveis requerem novos operadores espaço-temporais para explorar as possibilidades de consultas. Para tal, primeiramente as consultas são classificadas e posteriormente são descritos os tipos de operadores propostos.

A formulação de consultas em bancos de dados geográficos pode utilizar operadores espaciais, temporais e espaço-temporais. Os seus parâmetros (operandos) são valores, objetos (temporais, espaciais, ou espaço-temporais) ou conjuntos destes. Os operadores são representados por  $OPER(A,B,C \dots)$ , sendo  $OPER$  o nome do operador e  $A, B, C$ , etc., os seus operandos. Os operandos podem ser combinados ou compostos para formular operações mais complexas.

### 4.2 Consultas a bancos de dados móveis

Como visto no Capítulo 2, os atributos de objetos em bancos de dados espaço-temporais são divididos em componentes temáticos (ou convencionais), espaciais e temporais. Em geral, as consultas podem ser igualmente divididas em: consultas *temáticas*, consultas *espaciais*, consultas *temporais*, e consultas *espaço-temporais*. As consultas temáticas se referem ao conteúdo de atributos não espaciais; as consultas espaciais podem ser de localização, métricas, de orientação, ou topológicas. As consultas temporais são aquelas onde o tempo é parâmetro, e que consideram relações temporais entre ocorrência de fenômenos. Finalmente as consultas espaço-temporais combinam características das consultas espaciais e temporais.

Os objetos móveis, por serem objetos espaço-temporais contínuos, possuem uma história

mais completa e rica em termos de mudanças espaço-temporais. Eles armazenam o seu *comportamento* espaço-temporal e não apenas os *eventos* dos objetos tradicionais. [EGSV97] define que um relacionamento espaço-temporal é uma seqüência de relacionamentos espaciais que dura um intervalo de tempo, chamando-o de *desenvolvimento*.

Vamos tomar o exemplo de um carro durante uma enchente, e queremos saber se ele atravessou uma região alagada. Isso significa que é preciso verificar se o carro entrou na região, se ele ficou durante algum tempo dentro dela e se ele saiu. Ao mesmo tempo, a área alagada da enchente pode mudar de geometria o que complica o processamento. A consulta evidencia características típicas de consulta de objetos móveis: é preciso verificar diferentes predicados espaciais durante uma série de eventos e períodos, numa dada ordem temporal.

O enfoque adotado neste trabalho é de consultas sobre *transições* de relacionamentos espaciais entre um objeto móvel e (1) um fixo; (2) um objeto de mudança discreta (3) ou um outro objeto móvel durante, uma série de eventos e períodos. Os objetos considerados neste trabalho estão situados no espaço topológico  $R^2$ . A geometria do componente espacial dos objetos envolvidos corresponde a elementos simples que podem ser: um ponto; uma linha simples, que não se intercepta a si mesma e possui apenas dois pontos terminais; uma região simples, que é conectada, ou seja, que não é a união de conjuntos disjuntos de pontos, e que não contém buracos, representada por um polígono convexo.

Os operadores espaciais, temporais e espaço-temporais desta seção foram retirados da proposta de Faria [Far98].

---

### 4.2.1 Operadores Espaciais

Os operadores espaciais podem ser de localização, de orientação, métricos, e topológicos. Os operadores espaciais têm como operandos objetos geográficos (operador localização) ou seus componentes espaciais (operadores métricos, orientação e topológicos). Os operadores de orientação não serão estudados na dissertação.

#### Operador Localização

O operador de localização retorna o componente espacial do objeto passado como parâmetro. A localização consiste de uma lista de objetos geométricos que descrevem a geometria e a localização do objeto associada a atributos não-espaciais (escala e projeção por exemplo). A sintaxe básica é SP(A).

## Operadores Métricos

Os operadores métricos geram, a partir de um ou mais objetos, um escalar que representa uma propriedade intrínseca aos objetos analisados [Cil96]. Os operadores métricos são operadores quantitativos, e podem ser classificados por número de parâmetros:

- Unários:

AREA(A): calcula a área ocupada pelo polígono A.

PERIMETER(A): calcula o perímetro do polígono A.

LENGTH(A): calcula o comprimento da linha A.

- Binários:

DISTANCE (A,B): calcula a distância entre dois conjuntos (A e B) de objetos geométricos (pontos, linhas ou polígonos).

Tendo como base o trabalho de [TP95], define-se aqui a distância entre dois objetos geométricos  $p$  e  $q$  como a distância euclidiana entre os centróides dos dois objetos.

## Operadores Topológicos

Os operadores topológicos retornam um valor lógico (verdadeiro ou falso) se há um determinado relacionamento topológico entre dois conjuntos de objetos geométricos.

Os relacionamentos topológicos são invariantes face a transformações de escala, translação e rotação [EGSV97, CCH<sup>+</sup>96]. Essas mudanças são ditas qualitativas e não quantitativas pois não são expressas em medidas convencionais.

Os tradicionais modelos de relacionamento topológico são os modelos de 4-Intersection e a sua extensão 9-Intersection. Ambos utilizam o conceito de *fronteira* e o *interior* do objeto; o método 9-Intersection considera também o *exterior* do objeto. No método 4-Intersection, temos 2 variáveis (fronteira e o interior), sua combinação gerando uma matriz de 2x2. O método 9-Intersection, como tem 3 variáveis, gera uma matriz de 3x3. Nesta dissertação, será utilizado a notação de 9-Intersection: seja um objeto geométrico A,  $A^0$  denota o seu interior e  $\delta A$  a sua fronteira.

Como já mencionado no Capítulo 2, são basicamente cinco operadores: *DISJOINT*, *MEET*, *INSIDE*, *OVERLAP* E *CROSS*. Sejam A e B dois objetos geométricos em  $R^2$  :

- $DISJOINT(A,B) \leftrightarrow (A \cap B = \emptyset)$
- $MEET(A,B) \leftrightarrow (A \cap B \neq \emptyset) \wedge (A^0 \cap B^0 = \emptyset)$
- $INSIDE(A,B) \leftrightarrow (A \cap B = A) \wedge (A^0 \cap B^0 \neq \emptyset)$

- $OVERLAP(A,B) \leftrightarrow (A \cap B \neq A) \wedge (A \cap B \neq B) \wedge (A^0 \cap B^0) \neq \emptyset \wedge dim(A^0) = dim(B^0)$
- $CROSS(A,B) \leftrightarrow (A \cap B \neq A) \wedge (A \cap B \neq B) \wedge dim(A^0 \cap B^0) = \max(dim(A^0), dim(B^0)) - 1$

Um operador  $op$  é simétrico se e somente se  $(A \text{ op } B) \leftrightarrow (B \text{ op } A)$ . Com a exceção de *inside*, todos outros operadores são simétricos.

### 4.2.2 Operadores Temporais

Os operadores temporais podem ser unários ou binários. Os operandos podem ser valores de tempo ou objetos (temporais ou espaço temporais). O valor retornado pode ser valor de tempo, valores booleanos, ou objetos (temporais ou espaço-temporais).

Os valores de tempo considerados no presente trabalho são: instante de tempo, intervalo de tempo ou elemento temporal. A notação utilizada nesta seção é a seguinte:  $O$  denota um objeto geométrico e  $T$  componentes temporais (eventos ou intervalos).

#### Operadores Unários

Os operadores unários retornam valores de tempo. São definidos os seguintes:

- $TV(O)$ : retorna o componente temporal do objeto  $O$ , que designa o tempo de validade do objeto  $O$ .
- $BEGIN(T)$ : retorna o valor de tempo que representa o início de  $T$ . Ou seja:  
 $BEGIN(T) = \{t \mid t \in T \wedge \neg \exists t_1 (t_1 \in T \wedge t_1 < t)\}$  onde  $t$  é um instante de tempo }.
- $END(T)$ : retorna o valor de tempo que representa o instante final do  $T$ . Ou seja:  
 $END(T) = \{t \mid t \in T \wedge \neg \exists t_1 (t_1 \in T \wedge t_1 > t)\}$  onde  $t$  é um instante de tempo }.
- $DAY(T)$ ,  $MONTH(T)$  e  $YEAR(T)$ : retornam, respectivamente, dia, mês e ano considerando que  $T$  tenha uma granularidade composta formada pelo elemento (ano, mês, dia).
- $VWHEN(X)$ : retorna um conjunto de valores tempo no qual um relacionamento espaço-temporal é válido.

## Operadores Binários

Os seguintes operadores temporais binários de relacionamento são considerados, para  $T1$  e  $T2$  intervalos de tempo e  $t1$  e  $t2$  instantes.

- $T\_BEFORE(T1, T2)$ : retorna o valor lógico verdadeiro quando o último instante de  $T1$  é anterior ao primeiro instante de  $T2$ , e falso caso contrário. Ou seja:

$T\_BEFORE(T1, T2) \leftrightarrow END(T1) < BEGIN(T2)$ , onde  $T_S$  denota o tempo inicial e  $T_E$  denota o tempo final.

- $T\_OVERLAPS(T1, T2)$ : retorna o valor lógico verdadeiro quando os dois operandos se sobrepõem em algum valor de tempo, e falso caso contrário. Ou seja:

$T\_OVERLAPS(T1, T2) \leftrightarrow \exists t(t \in T1 \wedge t \in T2)$

- $T\_EQUAL(T1, T2)$ : retorna o valor lógico verdadeiro quando a duração de  $T1$  é igual à apresentada por  $T2$ , e falso caso contrário. Ou seja:

$T\_EQUAL(T1, T2) \leftrightarrow T1 = T2$

- $T\_CONTAINS(T1, T2)$ : retorna o valor lógico verdadeiro quando  $T1$  contém todos os valores de tempo pertencentes a  $T2$ , e falso caso contrário. Ou seja:

$T\_CONTAINS(T1, T2) \leftrightarrow \forall t \in T2, t \in T1$

- $T\_MEETS(T1, T2)$ : retorna o valor lógico verdadeiro quando  $T2$  inicia no instante seguinte ao instante final de  $T1$ , e falso caso contrário. Ou seja:

$T\_MEETS(T1, T2) \leftrightarrow BEGIN(T2) - END(T1) = 1$ , onde 1 representa um chronon

Os operadores de tempo anteriormente definidos são todos binários e verificam o relacionamento temporal entre os dois objetos temporais. Os operadores  $T\_EQUAL$  e  $T\_OVERLAP$  são simétricos enquanto que os outros não. Segundo [Far98], outros relacionamentos como *after*, *during*, *starts*, *finishes*, *disjoint* podem ser expressos utilizando os operadores acima.

Há ainda operadores temporais que criam objetos temporais, a saber:

- $EVENT(x)$ : retorna um objeto *Event* a partir de um valor numérico  $x$ .
- $INTERVAL(T1, T2)$ : retorna um intervalo de tempo tendo início  $T1$  e fim  $T2$ . Esse operador cria um objeto *Interval* a partir de dois valores.
- $T\_INTERVAL(t1, t2)$ : retorna um intervalo de tempo que corresponde à intersecção entre os instantes de tempo  $t1$  e  $t2$ .

- $VSLICE(O,T)$ : retorna os estados de  $O$  (objeto espaço-temporal) reduzidos apenas àqueles tempos especificados pelo parâmetro  $T$  (objeto temporal). Esse operador é semelhante a uma projeção de  $O$  no tempo.

### 4.2.3 Operadores Espaço-temporais

Os operadores espaço-temporais estendem os operadores espaciais, definidos na seção 4.2.1, pela adição da dimensão temporal. Seus operandos são objetos espaço-temporais, ou um objeto espaço-temporal e outro temporal. Os operadores espaço-temporais podem ser localização-temporal, métrico-temporais e topológico-temporais. Os operadores apresentados a seguir são aplicáveis a objetos espaço-temporais com mudanças discretas.

#### Operador Localização-temporal

O operador localização-temporal é denotado por  $ST\_SP(O,T)$ , e retorna a representação espacial do objeto  $O$  válida no tempo  $T$ . Informalmente tem-se:

$$ST\_SP(O,T) = \{ \langle Sp_i, T_i \rangle \}$$

onde  $Sp_i$  é o conjunto de objetos geométricos que descrevem a geometria de  $O$  no tempo  $T_i$  e todos  $T_i$  são disjuntos e contidos em  $T$ .

#### Operadores Métrico-temporais

Os operadores métrico-temporais recebem o tempo e o objeto como parâmetro e retornam uma lista de valores associados a valores métricos para aquele objeto, dentro daquele tempo.

- $ST\_AREA(O,T)$ : retorna pares (área, valor de tempo)
- $ST\_LENGTH(O,T)$ : retorna pares (comprimento, valor de tempo)
- $ST\_PERIMETER(O,T)$ : retorna pares (perímetro, valor de tempo)
- $ST\_DISTANCE(O1,O2,T)$ : retorna pares (distância entre  $O1$  e  $O2$ , valor de tempo), e a distância a cada valor de tempo é dada por  $DISTANCE(O1,O2)$ , ou seja,  $ST\_DISTANCE(O1, O2, T) = \{ \langle DISTANCE(O1, O2), T_i \rangle \}$

Os primeiros três operadores retornam pares de valores para cada estado do componente espacial do objeto  $O$  válido em  $T$ .  $ST\_DISTANCE$  retorna a distância entre os componentes espaciais de  $O1$  e  $O2$  (válidos em  $T$ ) para todos intervalos de intersecção temporal entre estes componentes.

Por exemplo,  $ST\_DISTANCE(O1,O2,[t1,t20]) = \{ \langle 10, [t1, t5] \rangle; \langle 15, [t6, t7] \rangle; \langle 18, [t8, t20] \rangle \}$  mostra que a distância entre O1 e O2 varia no tempo de 10 para 15 para 18 nos intervalos indicados.

### Operadores Topológicos-temporais

Os operadores topológico-temporais retornam uma lista de valores lógicos associados aos respectivos intervalos em que a localização de um objeto O1 e a localização de um objeto O2 são constantes no tempo T. Os valores lógicos indicam se o relacionamento topológico é verdadeiro ou falso.

Os operadores são:  $ST\_DISJOINT(O1,O2)$ ,  $ST\_MEETS(O1,O2)$ ,  $ST\_OVERLAP(O1,O2)$ ,  $ST\_INSIDE(O1,O2)$  e  $ST\_CROSS(O1,O2)$ .

Por exemplo,  $ST\_MEET(O1,O2) = \langle true, [t1, t2] \rangle; \langle false, [t3, t5] \rangle; \langle true, [t9, Now] \rangle$  indica que O1 e O2 são adjacentes nos intervalos  $[t1,t2]$  e  $[t9, Now]$ .

## 4.3 Operadores espaço-temporais para objetos móveis

### 4.3.1 Introdução

Os operadores espaço-temporais tradicionais são aplicados apenas em intervalos de tempo nos quais a localização do objeto permanece a mesma. Já os operadores espaço-temporais móveis se aplicam aos objetos cuja localização muda continuamente. Se estes operadores retornassem pares de tuplas  $\langle valores, tempo \rangle$ , geraria uma quantidade grande de tuplas de resultados. A idéia então, é retornar, tuplas que descrevem a trajetória dos objetos entre dois instantes amostrados. Os trabalhos de [EGSV98, GBE<sup>+</sup>00] sugerem que os operadores espaço-temporais que operam em objetos espaço-temporais de mudança contínua retornem valores que descrevem o movimento.

Esta seção define os operadores espaço-temporais móveis propostos. A maioria tem a sintaxe dos operadores espaço-temporais tradicionais, mas o resultado e os algoritmos de processamento são diferentes. Por convenção, M denota objetos móveis, C denota um objeto espaço-temporal de mudança discreta (*ChangeObject*) e que S denota um objeto fixo (*SpatialObject*).

A diferença dos operadores móveis em relação aos operadores espaço-temporais de mudança discreta não é apenas em sintaxe, mas também em semântica. Conceitos tais ENTRAR, SAIR, ATRAVESSAR, CONTORNAR, COLIDIR passaram a existir pela característica móvel dos objetos. Esses conceitos podem ser traduzidos em operadores móveis que dão possibilidades para novas consultas.

Apenas lembrando, para os operadores definidos a seguir, o modelo proposto considera que um objeto móvel M descreve um conjunto de trechos de trajetórias. Em cada trecho

de trajetória, a localização e geometria são descritos pela classe  $SP$ , a velocidade é descrita por  $V$  e o tempo da trajetória é descrito por  $T$ . O componente espaço-temporal de  $M$  é descrito por  $\langle SP_i, V_i, T_i \rangle$ . Se  $\langle SP_i, V_i, T_i \rangle, \langle SP_{i+1}, V_{i+1}, T_{i+1} \rangle$  é um trecho de trajetória e  $x$  o contróide de  $Sp$ , então  $\langle x_i, x_{i+1} \rangle$  é um segmento de reta.

### 4.3.2 Preprocessamento das trajetórias

Antes de calcular os operadores, é preciso fazer algumas etapas de preprocessamento das trajetórias para padronizá-las. Todos os operadores, desta forma, supõem que tenha havido esta padronização.

Os passos anteriores ao processamento de operadores são os seguintes:

- padronização das unidades de espaço
- padronização das unidades de tempo
- compatibilização dos intervalos de tempo

A padronização das unidades de espaço se dá através da padronização dos atributos de escala e projeção armazenados nos objetos geométricos que compõem as trajetórias, convertendo as unidades de medida espacial para uma única granularidade - no caso, a maior possível. Assim, havendo um espaço medido em metro e outro em quilômetro, a unidade a ser usada será quilômetro.

A padronização das unidades de tempo consiste em conversão de medidas de tempo para a mesma granularidade (a maior). Por exemplo, se a trajetória de caminhão é medida de hora em hora e a trajetória de outro caminhão é medida de minuto e minuto, então a medida de minutos deve ser convertida em horas.

A compatibilização de intervalos de tempo é necessária para operadores que envolvem dois objetos espaço-temporais. Ela consiste em duas etapas:

- a) Para operadores binários, escolher apenas os intervalos de tempo em que há intersecção temporal entre os objetos. Os intervalos de tempo fora da intersecção são desconsiderados.
- b) Dividir os intervalos de tempos dentro desta intersecção em intervalos comuns iguais.

### 4.3.3 Operadores Localização-temporais Móveis

Operadores localização-temporais retornam a localização de um objeto em um dado tempo. A forma geral de especificá-los é  $M\_SP(M,T)$ , onde  $M$  é um objeto móvel e  $T$  é um instante ou um intervalo de tempo.

$M\_SP(M,T)$  retorna uma tupla  $\langle Sp, T \rangle$ , se  $T$  é um objeto da classe *Event* (instante de tempo). Para cálculo desse valor, existe três casos: (1) o valor de  $T$  coincide com

o instante inicial (ou final) de uma trajetória. Nesse caso, só é necessário retornar a localização; (2) o valor de  $T$  está entre o instante inicial e o final de uma trajetória. Nesse caso, é necessário utilizar o cálculo da trajetória para obter a posição. (3) o valor de  $T$  não pertence a nenhum intervalo de tempo das trajetórias. Nesse caso, pode-se estimar um valor baseado no histórico do objeto.

Caso  $T$  seja um objeto da classe `Interval`, o operador retorna um conjunto de trajetórias (objeto da classe `Trajectory`) com o formato  $\langle Sp_i, T_i, V_i \rangle$ .

O operador auxiliar, `Geom_Traj(Traj, T)`, que retorna a geometria do objeto `Trajectory` no tempo  $T$ , o parâmetro de entrada `Traj` é um objeto da classe `Trajectory`. Esse operador é útil nos processamento dos operadores topológicos temporais. Lembrando que `Trajectory` de um ponto é linear, `Trajectory` de linha é linear ou poligonal e `Trajectory` dos polígonos é poligonal. Observe que no diagrama de classes proposto no Capítulo 3, este operador é um método da classe `Trajectory`.

#### 4.3.4 Operadores Métrico-temporais móveis

Os operadores métrico-temporais recebem um ou dois objetos móveis  $M$  e um objeto temporal  $T$  como parâmetro. O valor retornado é um conjunto de números reais no tempo. Num modelo onde os objetos sofrem deformações, perímetro, comprimento e área mudam com o tempo, e podem ser expressos em funções temporais ou em pares de valores temporais dependendo da continuidade da mudança. No caso do modelo proposto, os objetos considerados são apenas objetos rígidos que não sofrem mudança de forma. Desta maneira, esses operadores são definidos da seguinte maneira:

$$M\_PERIMETER(M,T) = ST\_PERIMETER(M,T) = PERIMETER(M) \text{ para } \forall t \in T$$

$$M\_LENGTH(M,T) = ST\_LENGTH(M,T) = LENGTH(M) \text{ para } \forall t \in T$$

$$M\_AREA(M,T) = ST\_AREA(M,T) = AREA(M) \text{ para } \forall t \in T$$

Surge aqui a necessidade de um novo operador `M_VELOCITY(M,T)` tipicamente para objetos móveis. Retorna as funções da velocidade no formato de distância/tempo (km/h por exemplo) do objeto no instante ou no intervalo de tempo  $T$ . Análogo ao operador localização temporal, quando o  $T$  é um instante, o operador `M_VELOCITY(M,T)` pode ser calculado em três casos: (1) o valor de  $T$  coincide com o instante inicial ou final de uma trajetória. Nesse caso, só é necessário retornar a velocidade inicial ou final ; (2) o valor de  $T$  está entre o instante inicial e o final de uma trajetória. Nesse caso, é necessário utilizar a função da velocidade para calculá-lo. (3) o valor de  $T$  não pertence a nenhum intervalo de tempo das trajetórias. Nesse caso, pode-se estimar um valor baseado no histórico. Quando  $T$  for um intervalo, o operador retorna um conjunto de pares de velocidade e tempo com o formato  $\langle V_i, T_i \rangle$ .

O operador `M_DISTANCE(A,B,T)` tem três parâmetros: dois objetos espaço-temporais

(A,B) e tempo (T). O valor da distância deve considerar 3 casos devido aos tipos de objetos envolvidos para cálculo de distância.

A função da distância é calculada baseado na localização do centróide:

$$D = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

onde  $x_A$  e  $x_B$  denotam coordenadas no eixo x do centróide de A e B e  $y_A$  e  $y_B$  denotam coordenadas no eixo y do centróide de A e B.

Sabe-se que para um ponto fixo (objeto S da classe SpatialObject), as coordenadas x e y não mudam. Para um objeto de mudança discreta (objetos C da classe ChangeObject), as coordenadas x e y permanece constantes durante um intervalo de tempo. Finalmente no caso de um objeto móvel M (MotionObject), as coordenadas x e y mudam conforme uma função de tempo em cada trecho da trajetória. Substituindo esses valores na equação de distância, temos:

#### Caso 1: MotionObject $\times$ SpatialObject

M\_DISTANCE(M,S,T) retorna uma lista de tuplas  $\langle D_i, T_i \rangle$  onde  $D_i$  calcula a distância entre o centróide de cada trecho de trajetória de M1 e a localização constante do objeto espacial fixo S.  $T_i$  são os intervalos de tempo de cada trecho da trajetória de M.

#### Caso 2: MotionObject $\times$ ChangeObject

M\_DISTANCE(M,C,T) retorna uma lista de  $\langle D_i, T_i \rangle$  onde  $D_i$  calcula a distância entre o centróide de M e C de cada trecho da trajetória de M em que a localização de C é constante.  $T_i$  são os intervalos de tempo em que há localização constante do Change-Object C.

#### Caso 3: MotionObject $\times$ MotionObject

M\_DISTANCE(M1,M2,T) retorna uma lista de  $\langle D_i, T_i \rangle$  onde  $D_i$  calcula a distância dos centróides de cada trecho da trajetória de M1 com cada trecho da trajetória de M2 em que os intervalos de tempo sejam iguais. Ou seja, os trechos de M1 e M2 passaram por processo de compatibilização de tempo.  $T_i$  são os intervalos de tempos gerado pelo processo de compatibilização.

### 4.3.5 Operadores Topológico-temporais móveis

Os operadores topológicos são complexos e conseqüentemente mais caros. No caso de operadores topológico-temporais, temos uma terceira dimensão que é tempo. Os trabalhos

[Pfo00, EGSV97] sugerem que os objetos móveis sejam modelados em espaço 3D. Neste espaço, os objetos móveis em questão estão no espaço  $R^2$  e o tempo é a terceira dimensão no qual os objetos móveis desenham o volume conforme passa o tempo. Considerando o modelo proposto, temos:

- Um objeto estático S (SpatialObject) é representado por um prisma com as duas bases paralelas, definidas pelo formato geométrico do objeto no espaço  $R^2$ . A altura do prisma é o tempo de validade do objeto.
- Um objeto de mudança discreta C (ChangeObject) é representado por um conjunto de prismas retos com as duas bases paralelas e definidas pelo formato geométrico do objeto no espaço  $R^2$ . A altura de cada prisma é definida pelo intervalo de tempo em que o objeto permaneceu na mesma localização.
- Um objeto móvel M (MotionObject) é representado por um conjunto de prismas oblíquos com bases paralelas e definidas pelo formato geométrico do objeto no espaço  $R^2$ . A altura de cada prisma conexo é definida pelo intervalo de tempo de cada trecho de trajetória.

A solução mais direta para implementar os operadores topológico-temporais móveis seria usar algoritmos topológicos para objetos 3D. Esses algoritmos são muito complexos e difíceis de implementar [KHM<sup>+</sup>98].

Para diminuir a complexidade de implementar os operadores móveis, a dissertação propõe outra abordagem: mapear os objetos de 3D para 2D e calcular os operadores sobre tais projeções.

Para entender melhor o algoritmo proposto, é importante lembrar que um ponto em movimento desenha uma trajetória de segmentos de linha. Uma linha em movimento desenha uma trajetória de polígonos e um polígono em movimento desenha um volume mas sua projeção em 2D é também um polígono.

### Algoritmo 1. mapeamento 3D para 2D

O algoritmo de mapeamento de 3D para 2D é o seguinte: dado os objetos  $O_1$  e  $O_2$ , deseja-se saber se o relacionamento topológico *OPER* existe para  $O_1$  e  $O_2$  no intervalo de tempo  $T$ , onde  $T$  foi dividido em intervalos menores  $I_i$  de acordo com o processo de compatibilização. O algoritmo é genericamente descrito em seguintes passos, para cada intervalo  $I_i$ :

1. executar o operador topológico espaço-temporais correspondente em 2D sobre os segmentos de trajetórias  $Traj_{1_i}$  e  $Traj_{2_i}$  percorridas pelos objetos no intervalo

- $I_i$ . Cada segmento de trajetória é uma reta nos casos de objeto do tipo ponto ou quando a projeção 2D é reta. Quando a projeção é um polígono, considera-se que um segmento de trajetória é definido quando o centróide do polígono percorre uma reta.
2. no intervalo de tempo  $I_i$ , se o relacionamento topológico não for verdadeiro, retorna falso para aquele intervalo. Caso o relacionamento topológico seja verdadeiro, determinar a região no espaço 2D  $R_i$  em que o relacionamento topológico se aplica e continua o passo seguinte.
  3. calcular os intervalos de tempo ( $I1_j$  e  $I2_j$ ) em que cada objeto esteve em  $R_i$  e verificar se  $I1_j \subseteq I2_j$ . Se for, o operador móvel retorna verdadeiro para  $I_i$ . Caso contrário, determine a intersecção temporal de ( $I1_j$  e  $I2_j$ ), onde  $T\_INTERVAL(I1_j, I2_j) = (I_{j+1})$ . Repita os passos anteriores utilizando o novo intervalo  $I_{j+1}$  até atingir uma tolerância  $I$ . Note que  $j$  denota o número de interações para determinar a validade do relacionamento no intervalo  $I_i$ .
  4. O algoritmo deve ser repetido para todos intervalos de tempo  $I_i$  que compõem o tempo de vida do objeto móvel.

Para ilustrar o algoritmo, podemos usar o cálculo da relação de intersecção entre as trajetórias dos objetos. Foi escolhida essa relação pois ela é a base para cálculo dos demais operadores já que tanto a teoria de 4-Intersection quanto a de 9-Intersection baseiam-se na informação da intersecção do interior, fronteira e exterior dos objetos.

A figura 4.1 mostra que para um objeto móvel(M) e outro estático(S), apenas uma interação do algoritmo (passo 1-3) é aplicada a cada trecho de trajetória. A explicação intuitiva é que se a trajetória do objeto móvel cruza com a posição do objeto fixo, então certamente em algum momento objeto móvel teve intersecção com o estático. Um objeto fixo fica na mesma posição no intervalo de tempo  $[-\infty, +\infty]$ . Dessa maneira, a verificação do intervalo temporal do passo (3) sempre será satisfeita pois sempre é verdadeiro afirmar que: o intervalo de tempo em que o objeto M esteve no  $R_i \subseteq [-\infty, +\infty]$ .

A figura 4.2 ilustra que um objeto móvel(M) e outro de mudança discreta(C), o algoritmo é aplicado após o pré-processamento e também apenas uma interação do algoritmo (passo 1-3) é aplicada para cada intervalo compatibilizado. Em cada intervalo compatibilizado, a posição do objeto C é fixa, e por isso o procedimento é igual ao do objeto estático S.

Os exemplos subsequentes ilustram caso de dois objetos móveis (M1 e M2). No caso da figura 4.3, apenas os passos 1 e 2 são executados. Isso acontece porque as trajetórias não se sobrepõem no mesmo intervalo. Os casos das figuras 4.4 e 4.5 ilustram que para dois objetos móveis que possuem intersecção nas trajetórias não se pode afirmar se

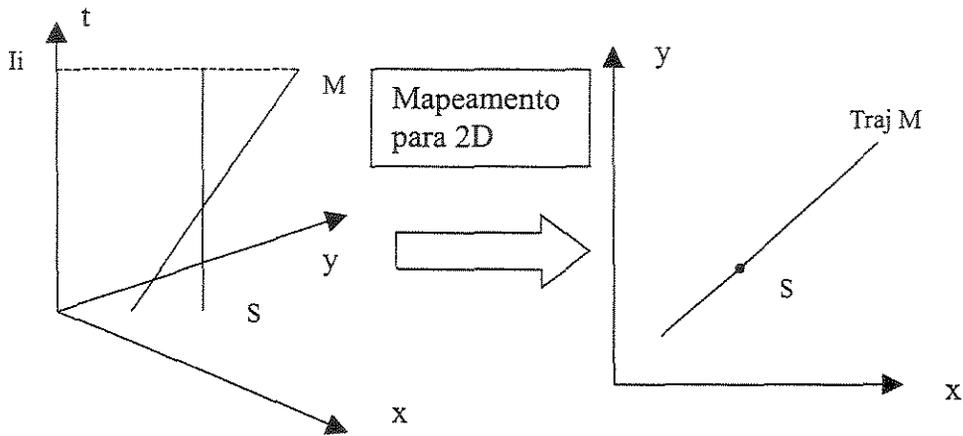


Figura 4.1: Exemplo 1 (mapeamento 3D-2D): Intersecção de  $S \times TrajM \rightarrow M$  tem intersecção com  $S$

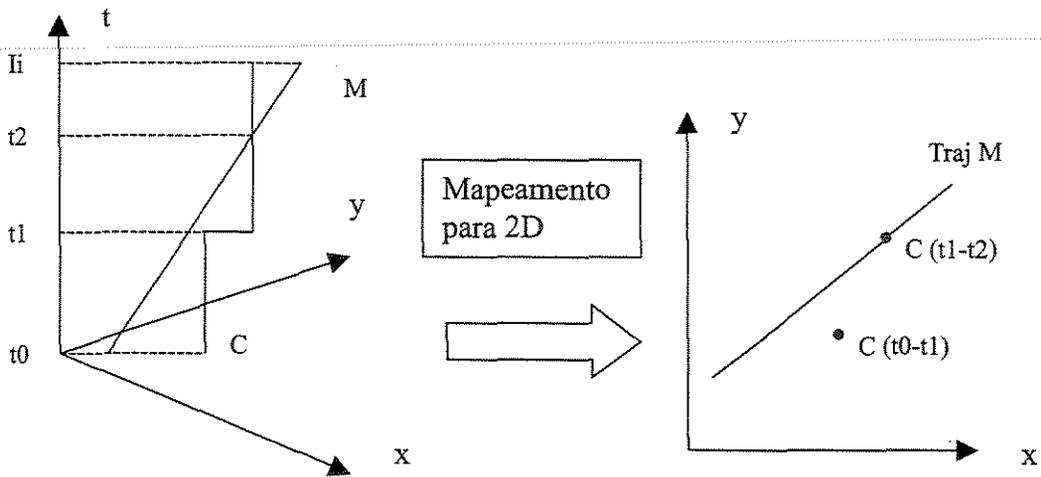


Figura 4.2: Exemplo 2 (mapeamento 3D-2D): Intersecção de  $C \times TrajM \rightarrow M$  tem intersecção com  $C$

houve intersecção ou não simplesmente baseado na geometria da trajetória. É necessário verificar a informação do intervalo temporal e executar o passo 4 caso necessário.

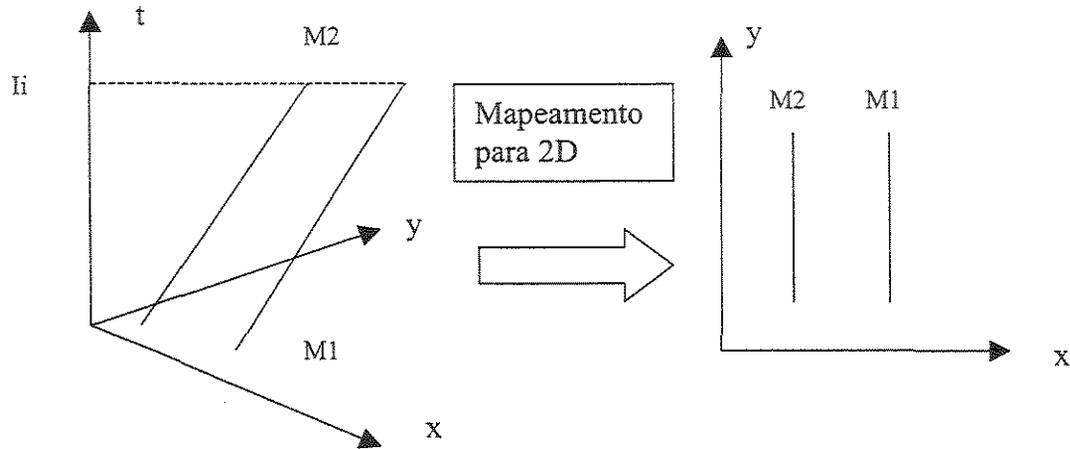


Figura 4.3: Exemplo 3 (mapeamento 3D-2D): Intersecção de  $TrajM1 \times TrajM2 \rightarrow M1$  não tem intersecção com  $M2$

A figura 4.6 mostra um caso mais complexo, onde todos os passos do algoritmo são aplicados repetidamente. Isso acontece porque o carro ( $M1$ ) praticamente "segue" o carro ( $M2$ ) no mesmo caminho com diferença temporal muito pequena. Nesse caso, deve-se analisar "fotografias" momentâneas para verificar existência de intersecção.

Pelos exemplos, pode-se perceber que o algoritmo analisa o componente espacial fixando o componente temporal. Caso uma "fotografia" permita calcular o relacionamento espacial, o algoritmo termina. Caso contrário, é necessário repetir o ciclo diminuindo o intervalo. O critério da diminuição de intervalo baseia-se na análise do componente espacial feito no ciclo anterior. Isso funciona como se usássemos um "zoom temporal" para verificar o relacionamento espaço-temporal.

A seguir, mostramos como cada operador móvel pode ser implementado usando o algoritmo. É importante ressaltar que não é objetivo da dissertação fazer análise algorítmica das possíveis soluções para operadores móveis. O objetivo porém, é mostrar que o modelo proposto permite que os operadores possam ser implementados e além disso, implementados de maneiras diferentes, cabendo a cada aplicação escolher o que mais lhe convém.

Considerando os operadores topológicos Disjoint, Overlap, Meet e Inside, a forma geral de um operador topológico-temporal móvel é:

$M\_TOP(M,O,T)$ , onde  $M$  é um objeto móvel,  $O$  é um objeto do tipo  $S$ ,  $C$  ou  $M$ ,  $TOP$  é um operador topológico e  $T$  um intervalo de tempo.

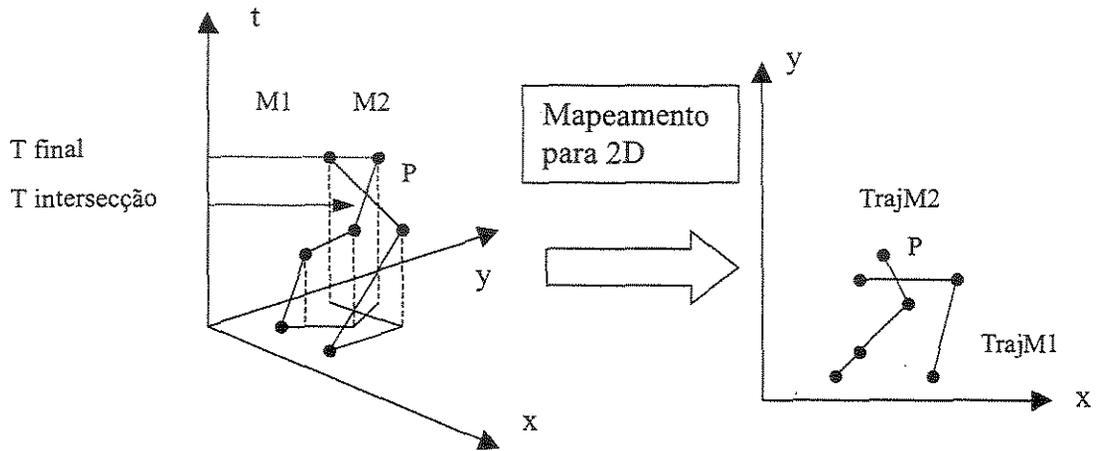


Figura 4.4: Exemplo 4 (mapeamento 3D-2D): Intersecção de TrajM1 x TrajM2 → M1 tem intersecção com M2

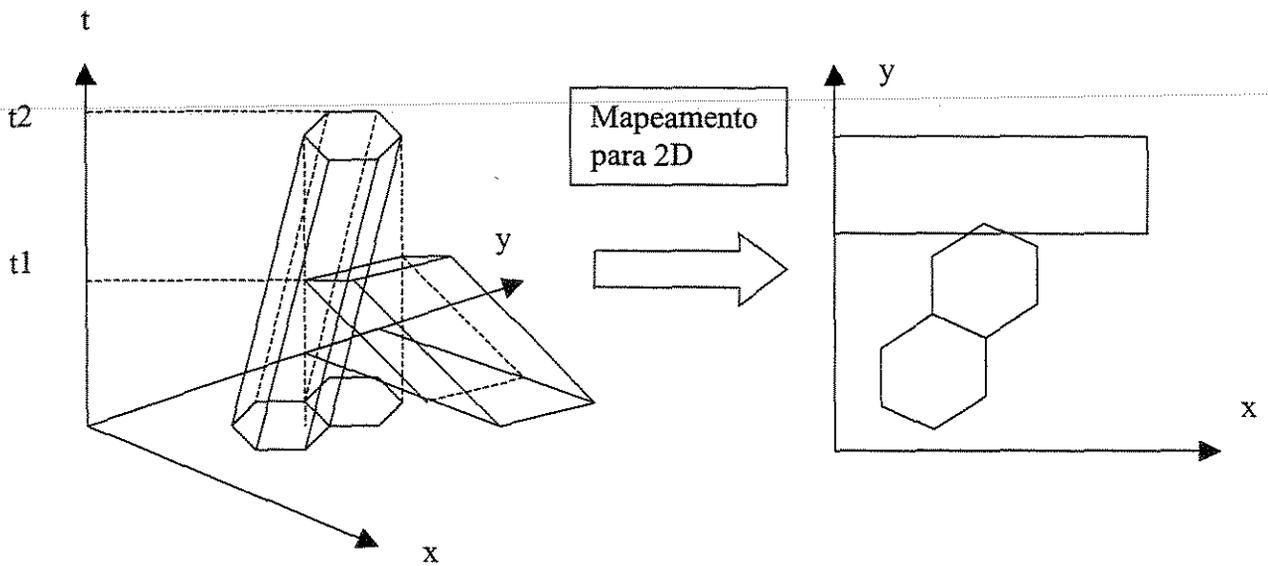
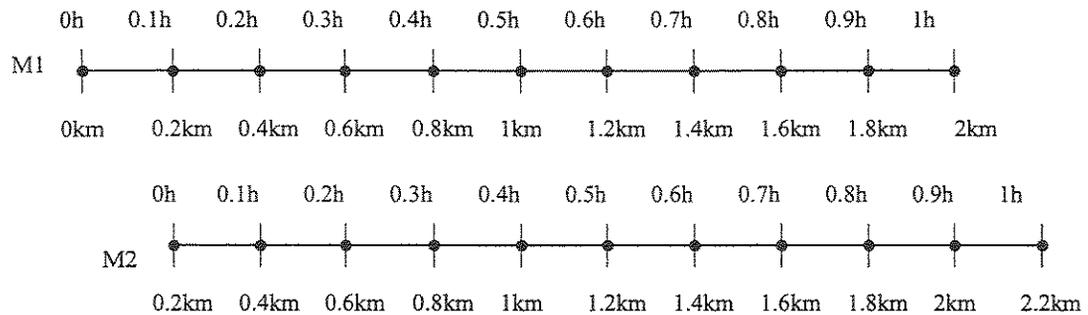


Figura 4.5: Exemplo 5 (mapeamento 3D-2D): Intersecção de TrajM1 x TrajM2 → M1 não tem intersecção com M2



$T = \{0h, 1h\}$ , sejam dois carros M1 e M2 que andam ambos 20km/h e distam 20 km no tempo 0h. Determinar se objetos pontuais M1 e M2 se interceptam aplicando o algoritmo de mapeamento.

- $j = 0, I_i = \{0h, 1h\}$ :
- (I) TrajM1 se intercepta com TrajM2 no trecho  $\{0.2km, 2km\}$
  - (II) M1 estava no trecho entre  $\{0.1h, 1h\}$  e M2 estava no trecho entre  $\{0h, 0.9h\}$
  - (III)  $\{0.1h, 1h\}$  não está contido nem igual a  $\{0h, 0.9h\}$ , continua a interação, o próximo intervalo temporal é  $\{0.1h, 0.9h\}$
- $j = 1, I_i = \{0.1h, 0.9h\}$ :
- (I) TrajM1 se intercepta com TrajM2 no trecho  $\{0.4km, 1.8km\}$
  - (II) M1 estava no trecho entre  $\{0.2h, 0.9h\}$  e M2 estava no trecho entre  $\{0.1h, 0.8h\}$
  - (III)  $\{0.2h, 0.9h\}$  não está contido nem igual a  $\{0.1h, 0.8h\}$ , continua a interação, o próximo intervalo temporal é  $\{0.1h, 0.8h\}$
- $j = 2, I_i = \{0.2h, 0.8h\}$ :
- (I) TrajM1 se intercepta com TrajM2 no trecho  $\{0.6km, 1.6km\}$
  - (II) M1 estava no trecho entre  $\{0.3h, 0.8h\}$  e M2 estava no trecho entre  $\{0.2h, 0.7h\}$
  - (III)  $\{0.3h, 0.8h\}$  não está contido nem igual a  $\{0.2h, 0.7h\}$ , continua a interação, o próximo intervalo temporal é  $\{0.3h, 0.7h\}$
- $j = 3, I_i = \{0.3h, 0.7h\}$ :
- (I) TrajM1 se intercepta com TrajM2 no trecho  $\{0.8km, 1.6km\}$
  - (II) M1 estava no trecho entre  $\{0.4h, 0.7h\}$  e M2 estava no trecho entre  $\{0.3h, 0.6h\}$
  - (III)  $\{0.4h, 0.7h\}$  não está contido nem igual a  $\{0.3h, 0.6h\}$ , continua a interação, o próximo intervalo temporal é  $\{0.4h, 0.6h\}$
- $j = 4, I_i = \{0.4h, 0.6h\}$ :
- (I) TrajM1 se intercepta com TrajM2 no trecho  $\{1km, 1.2km\}$
  - (II) M1 estava no trecho entre  $\{0.5h, 0.6h\}$  e M2 estava no trecho entre  $\{0.4h, 0.5h\}$
  - (III)  $\{0.5h, 0.6h\}$  não está contido nem igual a  $\{0.4h, 0.5h\}$ , continua a interação, o próximo intervalo temporal é o instante 0.5h
- $j = 5, I_i = \{0.5\}$ :
- No instante 0.5h, M1 estava na posição 1km e M2 na posição 1.2 km.
  - Fim do algoritmo com INTERSECT = falso

Figura 4.6: Exemplo 6 - Dois objetos móveis (Carro M1 e Carro M2)

A forma geral destes operadores visa obter o cálculo do relacionamento topológico TOP durante T e o resultado são tuplas  $\langle \text{boolean}, I_k \rangle$  onde  $I_k \subseteq T$ . Para simplificar a explicação dos algoritmos, supõe-se que T é infinito portanto será omitido das definições e figuras. Ou seja, a notação usada é simplificada para M\_TOP(M,O).

### Operador M\_DISJOINT(A,B)

O operador M\_DISJOINT(A,B) retorna verdadeiro para o instante/intervalo de tempo onde há intersecção entre dois objetos A e B. O operador M\_DISJOINT utiliza o conceito do operador espacial DISJOINT, a saber:

**Caso 1:** MotionObject(M)  $\times$  SpatialObject(S)

Para cada intervalo compatibilizado de tempo  $T_i$ , deve-se aplicar o algoritmo 1 da seguinte maneira:

1. Recuperar a geometria da trajetória de M para cada intervalo  $T_i$ , ou seja,  $GeomTrajM = GeomTraj(M\_SP(M, T_i))$ .
2. Se GeomTrajM for um ponto, uma linha ou um polígono e se DISJOINT(GeomTrajM, SP(S)) = Verdadeiro, então retorna Verdadeiro. Caso DISJOINT(GeomTrajM, SP(S)) = Falso, verifique a região de intersecção  $R_i$ .
3. como o objeto S fica na mesma posição no tempo  $[-\infty, +\infty]$ , o intervalo de tempo que objeto M entrou/saiu na região de intersecção será o intervalo de tempo onde M\_DISJOINT(M,S) é Falso.

As figuras a) até i) 4.7 mostram todos os 9 casos de disjunção entre objeto M(ponto, linha ou polígono) e objeto S(ponto, linha ou polígono).

Note que SP(S) pode ser um ponto, uma linha ou um polígono, dependendo da geometria de S. Note também que o Algoritmo 1 é aplicado apenas por uma interação (passos 1-3) já que se houve intersecção de um objeto móvel com uma região fixa, certamente houve a intersecção em algum instante.

**Caso 2:** MotionObject(M)  $\times$  ChangeObject(C)

O operador ST\_SP(C, $T_i$ ) é aplicado para recuperar a posição do objeto C em cada intervalo  $T_i$  de tempo em que permaneceu na mesma posição. Como resultado, temos uma lista de  $\langle Sp_i, T_i \rangle$ .

Para cada intervalo  $T_i$ , aplicar DISJOINT( $GeomTraj(M\_SP(M, T_i))$ ,  $SP_i$ ) da mesma maneira como é feito no caso de MotionObject(M)  $\times$  SpatioObject(S).

**Caso 3:** MotionObject(M1)  $\times$  MotionObject (M2)

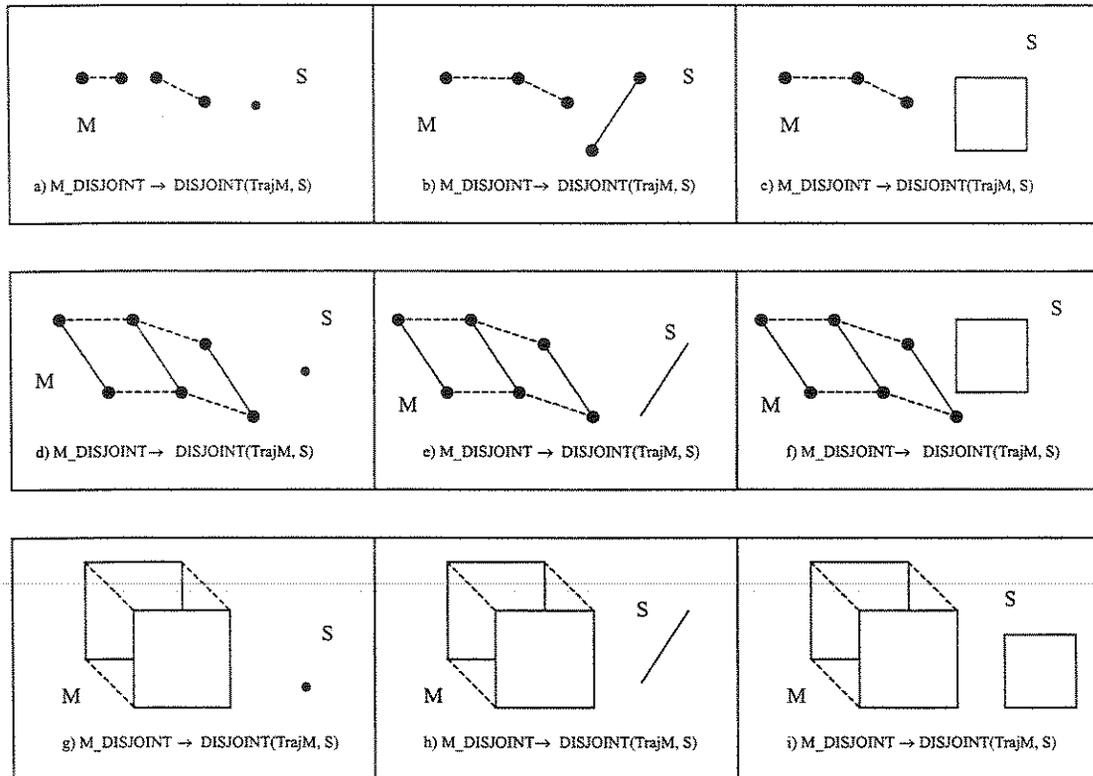


Figura 4.7: Operador  $M\_DISJOINT$

Para cada intervalo compatibilizado de tempo  $T_i$ , deve-se aplicar o algoritmo da seguinte maneira. O algoritmo retorna Verdadeiro se a comparação der Verdadeiro para todos intervalos compatibilizados. Caso contrário, termina com Falso ao primeiro resultado de Falso obtido.

1. Recuperar a geometria das trajetórias do M1 e M2 para cada trecho  $T_i$ , ou seja:  $GeomTrajM1 = Geom.Traj(M\_SP(M1, T_i))$  e  $GeomTrajM2 = Geom.Traj(M\_SP(M2, T_i))$ .
2. Se  $DISJOINT(GeomTrajM1, GeomTrajM2) = Verdadeiro$  Então retorna Verdadeiro. Caso  $DISJOINT(GeomTrajM1, GeomTrajM2) = Falso$ , verifique a região de intersecção  $R_i$  (Repetir até atingir a tolerância desejada de intervalo de tempo).

Note que mesmo que  $DISJOINT(GeomTrajM1, GeomTrajM2)$  retorne Falso, não necessariamente significa que houve a intersecção dos objetos. Para isso, é preciso que verificar se o intervalo temporal em que M1 entrou e saiu da região  $R_i$  está contido ou igual ao intervalo temporal em que M2 entrou e saiu da região  $R_i$ .

### Operador M\_OVERLAP(A,B)

O operador M\_OVERLAP(A,B) retorna verdadeiro para todo instante/intervalo de tempo onde há sobreposição entre dois objetos A e B. Note que OVERLAP só faz sentido entre objetos geométricos da mesma dimensão, excluindo os pontos: linha  $\times$  linha e polígono  $\times$  polígono.

#### Caso 1: MotionObject(M) $\times$ SpatioObject (S)

Um objeto linear em movimento gera uma trajetória GeomTrajM poligonal ou linear e um polígono em movimento gera uma trajetória poligonal. O algoritmo a seguir funciona para os casos em que M é linear e S é linear e GeomTrajM é linear, ou quando M é poligonal e S também é poligonal. Nestes casos, é possível definir M\_OVERLAP utilizando a definição padrão de OVERLAP:

1. Para cada intervalo compatibilizado de tempo  $T_i$ , recuperar a geometria da trajetória do M para cada trecho  $T_i$ , ou seja,  $GeomTrajM = Geom.Traj(M\_SP(M, T_i))$ .
2. Se  $OVERLAP(GeomTrajM, SP(S)) = Falso$ , então retorna Falso. Caso  $OVERLAP(GeomTrajM, SP(S)) = Verdadeiro$ , verifique a região de sobreposição  $R_i$ .
3. como o objeto S fica na mesma posição no tempo  $[-\infty, +\infty]$ , o intervalo de tempo em que o objeto M entrou/saiu da região de sobreposição será o intervalo de tempo onde M\_OVERLAP(M,S) é Verdadeiro.

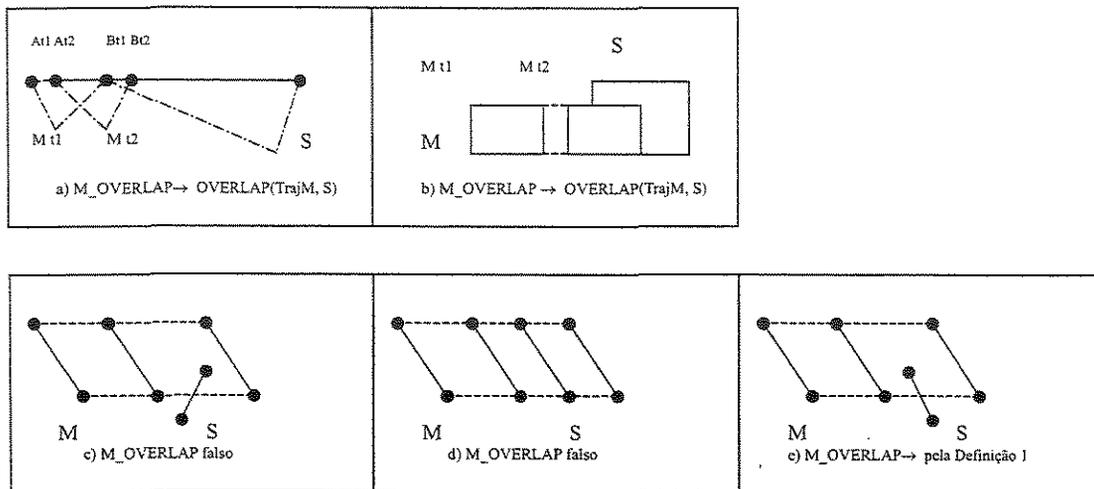


Figura 4.8: Operador M-OVERLAP

A figura 4.8a mostra o caso de objeto linear  $M$  com movimento linear que tem uma relação de OVERLAP com objeto linear  $S$  no tempo  $t_2$ . A figura 4.8b mostra o caso de um objeto poligonal e a relação de OVERLAP com o objeto poligonal  $S$  no tempo  $t_2$ .

Se  $M$  é uma linha e sua trajetória é um polígono, surge um problema, porque não há definição do operador espacial de OVERLAP entre polígono e linha. Sendo assim, a dissertação propõe um novo conceito de OVERLAP para um objeto linear móvel com outro objeto linear.

Definição 1.  $M\_OVERLAP(M,S) \leftrightarrow (GeomTrajM \cap SP(S) \neq SP(S)) \wedge (TrajM \cap SP(S) \neq M\_SP(M, T_{int}) \wedge dim(GeomTrajM \cap SP(S)) = 1 \wedge (S \parallel M\_SP(M, T_{int})))$ , onde  $T_{int}$  é o tempo de intersecção do  $GeomTrajM$  e  $S$ .

Intuitivamente, a definição 1 significa que:

1.  $GeomTrajM$  se sobrepõe a  $S$  e a intersecção é uma linha
2.  $M$  deve estar sempre paralelo a  $S$  mas em nenhum instante da sua trajetória,  $M$  será idêntico a  $S$ .

A figura 4.8c mostra que apesar do  $GeomTrajM$  interceptar  $S$ ,  $M$  não é paralelo a  $S$  e portanto não há sobreposição. A figura 4.8d mostra o caso de que apesar de  $GeomTrajM$  interceptar  $S$  e  $M$  ser paralelo a  $S$ , a intersecção de  $GeomTrajM$  e  $S$  é idêntica a  $S$  (i.é, de novo, não há sobreposição). Só a figura 4.8e mostra um caso em que  $M\_OVERLAP(M,S)$  é verdadeiro.

Caso 2:  $MotionObject(M) \times ChangeObject(C)$

O operador  $ST\_SP(C, T_i)$  é aplicado para recuperar a posição do objeto  $C$  em cada intervalo  $T_i$  de tempo em que permaneceu na mesma posição. Como resultado, temos uma lista de  $\langle Sp_i, T_i \rangle$  e para cada intervalo  $T_i$ , aplicamos  $M\_OVERLAP(GeomTrajM, SP_i)$  da mesma maneira como é feito no caso de  $MotionObject(M) \times SpatioObject(S)$ .

### Caso 3: $MotionObject(M1) \times MotionObject(M2)$

$M1$  e  $M2$  podem ter trajetórias lineares ou poligonais. Os passos 1 e 2 são semelhantes ao Caso 1. A exceção é o passo 3 para cada intervalo  $T_i$ :

1. Recuperar a geometria das trajetórias do  $M1$  e  $M2$  para cada trecho  $T_i$ , ou seja:  $GeomTrajM1 = Geom\_Traj(M\_SP(M1, T_i))$  e  $GeomTrajM2 = Geom\_Traj(M\_SP(M2, T_i))$ .
2. Se  $OVERLAP(GeomTrajM1, GeomTrajM2) = \text{Falso}$ , então retorna Falso. Caso  $OVERLAP(GeomTrajM1, GeomTrajM2) = \text{Verdadeiro}$ , verifique a região de sobreposição  $R_i$ .
3. Mesmo que  $OVERLAP(TrajM1, TrajM2)$  retornasse Verdadeiro, não necessariamente significa que houve a intersecção dos objetos. Para isso, precisa verificar se o intervalo temporal em que objeto  $M1$  entrou e saiu da região de sobreposição ( $R_i$ ) está contido ou igual ao intervalo temporal em que objeto  $M2$  entrou e saiu da região.

### Operador $M\_MEET(A, B)$

O operador  $M\_MEET(A, B)$  retorna verdadeiro para cada instante/intervalo de tempo onde o objeto  $A$  toca o objeto  $B$ . O relacionamento espacial  $MEET$  acontece em 5 casos: 1) um ponto e uma linha 2) um ponto e uma região 3) uma linha e uma região 4) duas linhas 5) duas regiões.

#### Caso 1: $MotionObject(M) \times SpatioObject(S)$

1. Para cada intervalo compatibilizado de tempo  $T_i$ , recuperar a geometria da trajetória de  $M$  para cada trecho  $T_i$ , ou seja  $GeomTrajM = Geom\_Traj(M\_SP(M, T_i))$ .
2. Existem variadas condições de verificação da existência da relação  $M\_MEET(A, B)$ , basicamente divididos em três grupos para geometria de  $M$  e  $S$ .
  - Para casos (ponto, linha) e (linha, polígono) quando  $M$  for linha e  $TrajM$  for polígono e  $S$  for polígono, então verifique  $MEET(TrajM, SP(S))$  e  $OVERLAP(TrajM, S)$ . Se ambos operadores retornarem Falso, retorna Falso. Caso ambos operadores retornem Verdadeiro, verifique a região de sobreposição  $R_i$ . Veja os exemplos representados pelas figuras 4.9a, 4.9b, 4.9p, 4.9q, 4.9t

e 4.9u. Por exemplo, a figura 4.9a mostra  $M\_MEET(M,S)$  como  $MEET(\text{ponto}, \text{linha})$ . A figura 4.9q mostra  $M\_MEET(M,S)$  como o cálculo de  $M\_OVERLAP(\text{linha}, \text{polígono})$  usando a nova noção introduzida na dissertação de  $M\_OVERLAP$  de linha, polígono, em que a geometria de trajetória de  $M$  é um polígono.

- Para casos (ponto, polígono), (linha, linha), (linha, polígono), (polígono, linha), então verifique  $MEET(\text{TrajM}, SP(S))$  e  $CROSS(\text{TrajM}, S)$ . Se ambos operadores retornarem Falso, então retorna Falso. Caso ambos operadores retornem Verdadeiro, verifique a região de sobreposição  $R_i$ . Veja os exemplos representados pelas figuras 4.9f, 4.9g, 4.9j, 4.9k, 4.9l, 4.9m, 4.9n, 4.9o, 4.9r e 4.9s, que cobrem todos os casos.
- Se  $M$  for linha e  $\text{TrajM}$  for linha e  $S$  for ponto; ou se  $M$  for polígono e  $S$  ponto; então verifique  $MEET(\text{TrajM}, SP(S))$  e  $INSIDE(\text{TrajM}, S)$ . Se ambos operadores retornarem Falso, então retorna Falso. Caso ambos operadores retornarem Verdadeiro, verifique a região de sobreposição  $R_i$ . Veja os exemplos representados pelas figuras 4.9c, 4.9d, 4.9h e 4.9i de  $M\_MEET(A,B)$ , que cobrem todos os casos.

3. como o objeto  $S$  fica na mesma posição no tempo  $[-\infty, +\infty]$ , o intervalo de tempo em que objeto  $M$  entrou/saiu na região de sobreposição será o intervalo de tempo onde  $M\_MEET(M,S)$  é Verdadeiro.

É natural que consideremos o caso de  $MEET(\text{TrajM}, S)$ , mas por que considerar também  $OVERLAP(\text{TrajM}, S)$  e  $CROSS(\text{TrajM}, S)$ ? Na verdade,  $CROSS(\text{TrajM}, S)$  e  $OVERLAP(\text{TrajM}, S)$  indicam que em algum momento houve um toque ( $MEET$ ) entre os dois objetos, mas o objeto  $A$  continuou a sua trajetória e cruzou ( $CROSS$ ) ou sobrepôs ( $OVERLAP$ ) ao objeto  $B$ .

#### Caso 2: $\text{MotionObject}(M) \times \text{ChangeObject}(C)$

Em todos casos pode-se aplicar o algoritmo descrito para o Caso 1 para cada trecho em que a posição de  $C$  se mantém a mesma.

#### Caso 3: $\text{MotionObject}(M1) \times \text{MotionObject}(M2)$

Análogo ao Caso 1,  $M1$  pode ter terem trajetórias lineares ou poligonais assim como  $M2$ . Os passos 1 e 2 são semelhantes ao Caso 1. A diferença é o passo 3:

1. Recuperar a geometria das trajetórias do  $M1$  e  $M2$  para cada trecho  $T_i$ , ou seja:  $\text{GeomTrajM1} = \text{GeomTraj}(M\_SP(M1, T_i))$  e  $\text{GeomTrajM2} = \text{GeomTraj}(M\_SP(M2, T_i))$ .
2. A seguir, computar o Caso 1, calculando para  $\text{GeomTrajM1}$  e  $\text{GeomTrajM2}$ .

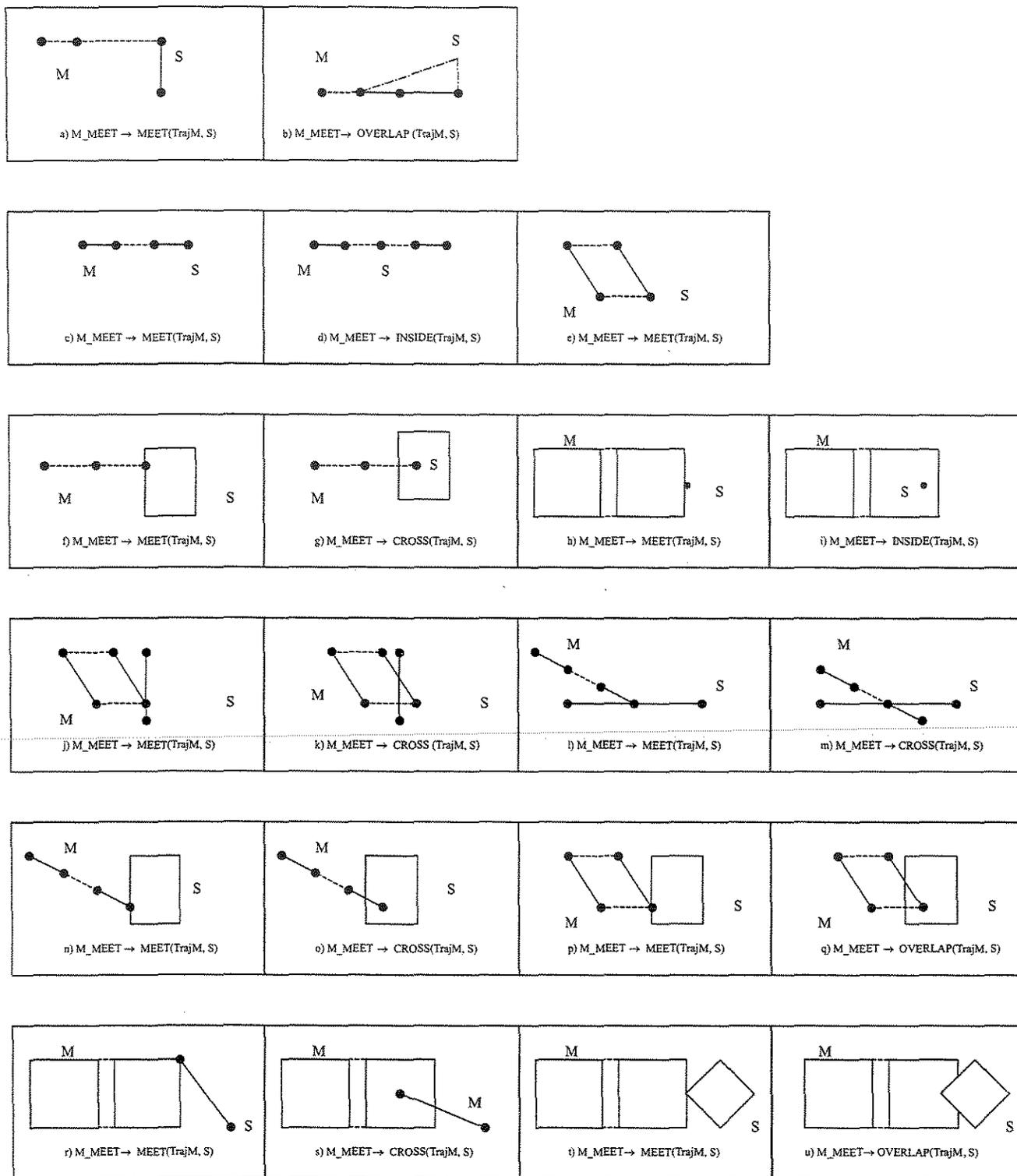


Figura 4.9: Operador M\_MEET

3. Retornar que a relação é verdadeiro no passo 2 não necessariamente significa que houve a intersecção dos objetos. Para isso, precisa verificar se o intervalo temporal em que o objeto M1 entrou e saiu da região de sobreposição ( $R_i$ ) está contido ou igual ao intervalo temporal em que objeto M2 entrou e saiu da região  $R_i$ .

### Operador M\_INSIDE(A,B)

O operador M\_INSIDE(A,B) retorna verdadeiro para cada instante/intervalo de tempo onde o objeto A está dentro do objeto B. Situações de INSIDE acontecem em 6 casos: 1) um ponto dentro do outro ponto 2) um ponto dentro de uma linha 3) um ponto dentro de um polígono 4) uma linha dentro da outra linha 5) uma linha dentro de um polígono 6) um polígono dentro do outro polígono.

#### Caso 1: MotionObject(M) $\times$ SpatioObject(S)

1. Para cada intervalo compatibilizado de tempo  $T_i$ , recuperar a geometria da trajetória de M para cada trecho  $T_i$ , ou seja  $GeomTrajM = Geom\_Traj(M\_SP(M, T_i))$ .
2. Existem variadas condições de verificação da existência da relação M\_INSIDE(A,B):
  - Para M ponto e S ponto, se  $(INSIDE(GeomTrajM, S) \wedge MEET(GeomTrajM, S)) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura 4.10a e 4.10b.
  - Para M ponto e S linha, se  $(INSIDE(GeomTrajM, S) \wedge MEET(GeomTrajM, S) \wedge CROSS(GeomTrajM, S) \wedge OVERLAP(GeomTrajM, S)) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura 4.10c, 4.10d e 4.10e.
  - Para M linha e S ponto, se  $(INSIDE(GeomTrajM, S) \wedge MEET(GeomTrajM, S)) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura 4.10f, 4.10g, e 4.10h.
  - Para M ponto e S polígono, se  $(INSIDE(GeomTrajM, S) \wedge CROSS(GeomTrajM, S)) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura 4.10i e 4.10j.
  - Para M ponto e S polígono, se  $INSIDE(GeomTrajM, S) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura 4.10k.
  - Para M linha e S linha, se  $(INSIDE(GeomTrajM, S) \wedge MEET(GeomTrajM, S) \wedge OVERLAP(GeomTrajM, S)) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura 4.10l, 4.10m e 4.10n.

- Para M linha e S polígono, se  $(\text{INSIDE}(\text{GeomTrajM}, S) \wedge \text{OVERLAP}(\text{GeomTrajM}, S)) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura 4.10o e 4.10p.
  - Para M polígono e S polígono, se  $\text{INSIDE}(\text{GeomTrajM}, S) = \text{Falso}$ , retorne Falso. Se verdadeiro, executa passo 3. Veja a figura q) do M\_INSIDE (figura 4.10).
3. como o objeto S fica na mesma posição no tempo  $[-\infty, +\infty]$ , o intervalo de tempo em que objeto M entrou/saiu na região de sobreposição será o intervalo de tempo onde  $\text{M\_INSIDE}(M,S)$  é Verdadeiro.

**Caso 2:**  $\text{MotionObject}(M) \times \text{ChangeObject}(C)$

Em todos casos pode-se aplicar o algoritmo descrito para o Caso 1 para cada trecho em que a posição de C se mantém a mesma.

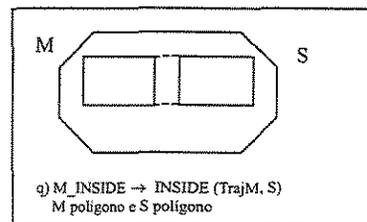
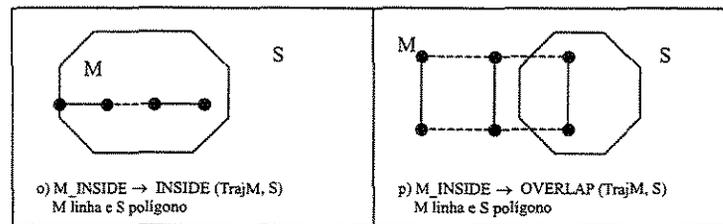
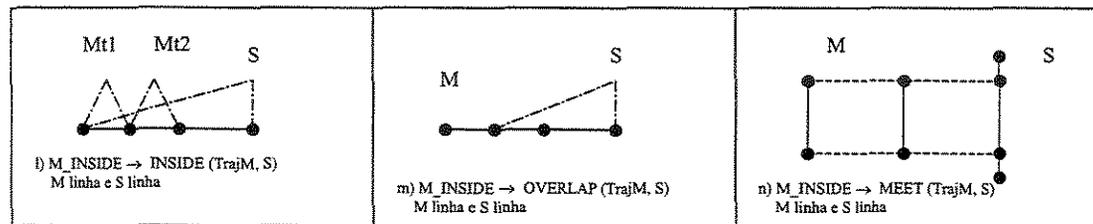
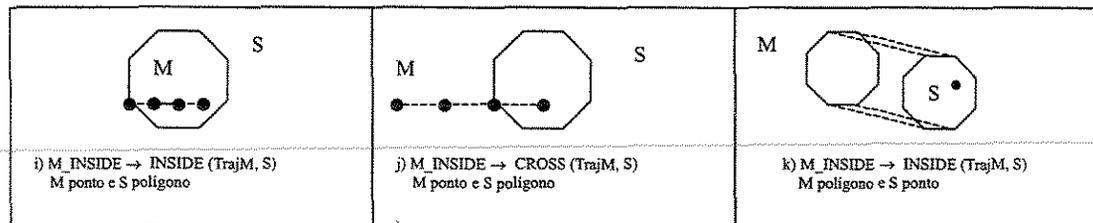
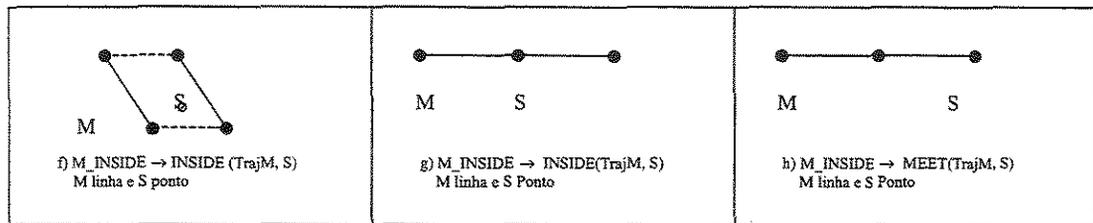
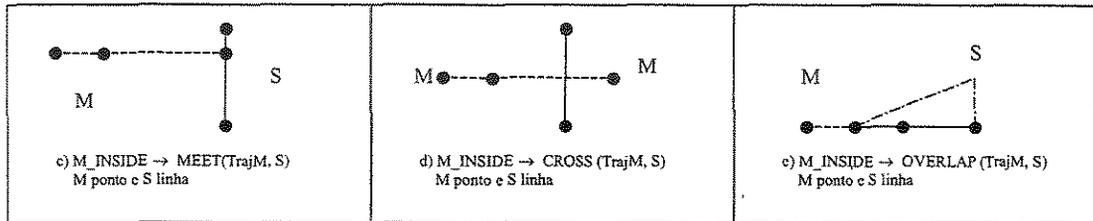
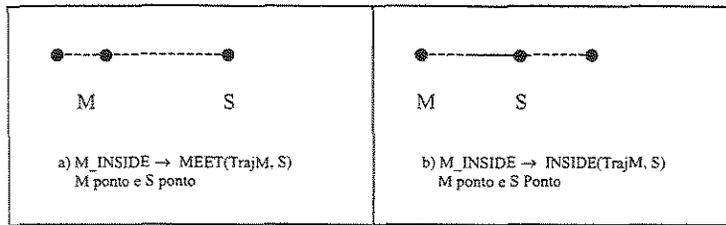
**Caso 3:**  $\text{MotionObject}(M1) \times \text{MotionObject}(M2)$

Análogo ao Caso 1, M1 pode ter trajetórias lineares ou poligonais assim como M2. Os passos 1 e 2 são semelhantes ao Caso 1. E o passo 3 é o seguinte:

1. Recuperar a geometria trajetória de M1 e M2 para cada trecho  $T_i$ , ou seja:  $\text{GeomTrajM1} = \text{Geom.Traj}(M\_SP(M1, T_i))$  e  $\text{GeomTrajM2} = \text{Geom.Traj}(M\_SP(M2, T_i))$ .
2. Esse passo é análogo ao Caso 1, para  $\text{GeomTrajM1}$  e  $\text{GeomTrajM2}$ .
3. Retornar que a relação é verdadeiro no passo 2 não necessariamente significa que houve a intersecção dos objetos. Para isso, precisa verificar se o intervalo temporal em que objeto M1 entrou e saiu da região onde a geometria do objeto M1 contém a geometria do objeto M2 ( $R_i$ ) está contido ou igual do intervalo temporal que objeto M2 entrou e saiu da região  $R_i$ .

**Observação**

Como pode se observar, no caso de relacionamento topológico entre um objeto estático e um móvel ou de um objeto de mudança discreta e móvel, só é preciso executar uma interação do Algoritmo 1. Já no caso de 2 objetos móveis, mais interações precisam ser feitas.



### 4.3.6 Operadores compostos móveis

Os operadores espaço-temporais móveis descritos na seção anterior são bastante semelhantes aos operadores espaço-temporais tradicionais. A principal diferença é que o processo é realizado sobre trajetórias. O que caracteriza os objetos móveis é a noção de trajetória, ou seja, uma seqüência de mudanças contínuas de relacionamento (métrico ou topológico) ao longo do tempo. A composição dos operadores anteriores permite o cálculo de novos tipos de função.

#### Operador métrico-temporal composto

Operadores métrico-temporais compostos envolvem aplicação de dois operadores métrico-móveis em seqüência. Os operadores  $APPROACH(M1,M2)$  e  $MOVEAWAY(M1,M2)$  são utilizados para descobrir se o objeto M1 se aproxima ou se afasta de M2, respectivamente. A métrica usada para esses operadores é a distância. Seja T um intervalo de tempo  $(T_S, T_E)$ , em que  $T_S$  BEFORE  $T_E$ .

- $APPROACH(M1,M2, T)$ : retorna verdadeiro quando  $M\_DISTANCE(M1, M2, T_S) > M\_DISTANCE(M1, M2, T_E)$
- $MOVEAWAY(M1,M2)$ : retorna verdadeiro quando  $M\_DISTANCE(M1, M2, T_S) < M\_DISTANCE(M1, M2, T_E)$

Uma sugestão de implementação desses operadores é calcular o mínimo(ou os mínimos locais) das funções de distância. Partindo desses mínimos, procura-se a seguir os trechos de expansão/redução e aproximação/afastamento.

#### Operador topológico-temporal composto

Operadores topológico-temporais compostos envolvem aplicação de dois operadores móveis topológicos em seqüência. Os operadores  $M\_ENTER(M,O)$  e  $M\_LEAVE(M,O)$  são aplicados a dois objetos, ao menos um móvel.

- $M\_ENTER(M,O,T)$ : retorna verdadeiro quando  $M\_DISJOINT(A, B, T_S) \wedge M\_INSIDE(M, O, T_E)$
- $M\_LEAVE(M,O,T)$ : retorna verdadeiro quando  $M\_INSIDE(M, O, T_S) \wedge M\_DISJOINT(M, O, T_E)$
- $M\_CROSS(M,O,T)$ : retorna verdadeiro quando  $M\_ENTER(M, O, T_S) \wedge M\_LEAVE(M, O, T_E)$

Observe que os operadores `M_ENTER` e `M_LEAVE` poderiam ter envolvido o operador `M_MEET(A,B)` na seqüência entre `M_DISJOINT` e `M_INSIDE`, pois se um objeto entra ou sai de um outro objeto se ambos se encontram na borda(`M_MEET`). No entanto, baseado na teoria do Engenhofer[[EAT92](#)], pode-se concluir que sendo um movimento contínuo, um objeto não poderia entrar em outro objeto sem se encontrar com o outro na borda.

A partir dos operadores espaço-temporais móveis básicos é possível construir outros operadores móveis compostos. Basta combinar os operadores temporais com os operadores espaço-temporais móveis.

### Mudança gradual de relacionamento topológico

O trabalho de Engenhofer et al. [[EAT92](#)] propôs um modelo que fornece uma medida quantitativa para calcular quanto um relacionamento topológico difere do outro. Relacionamentos topológicos são sensíveis a transformações tais como translação, rotação, redução e expansão. Essas transformações são mapeadas num grafo, que também demonstra a “distância” entre dois relacionamentos de forma gráfica, ou seja, a transição de estado dos relacionamentos topológicos. A figura 4.11 mostra transições entre relacionamentos topológicos.

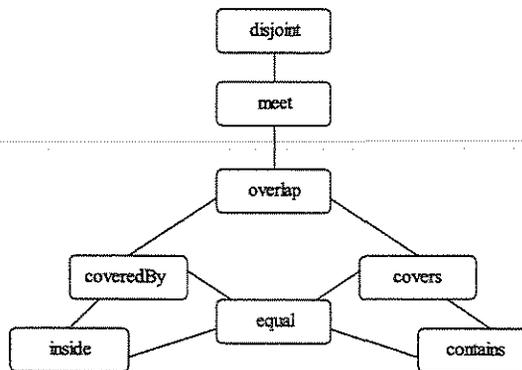


Figura 4.11: Grafo do relacionamento topológico

O grafo descreve o relacionamento topológico entre duas regiões (polígonos) que possuem mudanças graduais da sua localização e formato. Cada nó representa um relacionamento entre as duas regiões. O grafo não é direcional, ou seja, um relacionamento pode transitar para outro relacionamento que está conectado a ele, mas nunca poderia pular para outro que não esteja diretamente conectado. Por exemplo, dados dois objetos que se encontram(`MEET`), as únicas possíveis evoluções(próximo estado)seriam `DISJOINT` ou

OVERLAP. Por isso, um relacionamento ENTER obrigatoriamente passa de DISJOINT para MEET e posteriormente para OVERLAP, COVEREDBY e finalmente INSIDE. Cada possível caminho do grafo é um padrão de comportamento, e cada aplicação deveria escolher a combinação de relacionamentos que lhe interesse. Esse modelo também pode ser usado para predição de relacionamentos já que os possíveis relacionamentos são os nós que estão conectados no presente relacionamento.

## 4.4 Resumo

Este capítulo descreveu os operadores espaciais, temporais e espaço-temporais para objetos de mudança discreta e para objetos móveis. Propôs um algoritmo de mapeamento de operadores 3D para 2D, utilizando trajetórias descritas pelo objeto para simplificar e otimizar implementação de operadores móveis. Em seguida, foram levantados alguns exemplos de operadores compostos para objetos móveis, explorando as possibilidades de novas consultas. O mais importante é mostrar que os operadores móveis podem ser implementados sobre o modelo proposto pelo Capítulo 3. O próximo capítulo mostra como esses operadores podem ser utilizados em consultas.

---

# Capítulo 5

## Consultas a objetos móveis

### 5.1 Introdução

O capítulo 4 apresentou os operadores de objetos móveis. Este capítulo mostra uma aplicação exemplo, ilustrando o uso e combinação dos operadores em especificação de consultas. Os exemplos utilizarão a sintaxe do SGBD OO O2, para facilitar o entendimento. Embora este seja um sistema que não mais exista, sua linguagem de consultas serviu como base para as atuais linguagens de consulta de SGBD objeto-relacional.

Uma consulta de banco de dados seleciona conjuntos de objetos ou atributos de objetos que satisfazem determinadas condições. Então pode-se decompor consultas em duas partes:

- 
- selecionar um conjunto de objetos que satisfazem determinadas condições
  - selecionar atributos desses objetos

O exemplo a ser usado no capítulo envolve um sistema fictício de monitoramento de transportes da Cidade de São Paulo. Nesse sistema, os objetos estáticos são modelados como objetos pontuais (semáforos), lineares (ruas) ou poligonais (zonas da cidade). Existem também objetos de mudança discreta: lineares (linhas de bloqueio) ou poligonais (lagos e rios que inundam periodicamente). Os objetos móveis considerados podem ser: pontuais (carros), lineares (comboio de caminhão).

As consultas têm basicamente três objetivos:

- avaliar as propriedades e o comportamento dos objetos móveis do sistema - por exemplo, determinar a trajetória dos veículos.
- avaliar a interação dos objetos móveis com outros objetos - por exemplo, a interferência da malha viária (estático) na trajetória de um carro (móvel) ou o impacto de uma enchente (mudança discreta) na trajetória de um caminhão (móvel).

- monitorar objetos móveis para relatar infrações - por exemplo, determinar os carros que trafegaram nos dias em que a placa não era permitida.

As consultas estão organizadas segundo predicados temporal, espacial ou espaço-temporal.

## 5.2 Definição de Classes

Antes de desenvolver uma aplicação é necessário definir o esquema de banco de dados e das classes. Definindo os objetos da aplicação exemplo segundo o modelo proposto no Capítulo 3, temos:

- Objetos estáticos: os objetos estáticos(semáforo, rua, zona) pertencem a classes derivadas de SpatialObject. Os componentes espaciais (classe obj\_Geom) são: Semáforo: Point; Rua: Line; Zona: Polygon;
- Objetos de mudança discreta: os objetos de mudança discreta (linha de bloqueio, Rio/Lago) pertencem a classes derivadas de ChangeObject, que é uma especialização de SpatioTemporalObject. Os componentes geométricos (classe obj\_GeomT) são: Linha de Bloqueio: Line; Rio/Lago: Polygon;
- Objetos de mudança contínua: os objetos de mudança contínua (carro, comboio) pertencem a classes derivadas de MotionObject, que é uma especialização de SpatioTemporalObject. Parte espacial de um MotionObject é formada por um conjunto de objetos Trajectory. Cada objeto Trajectory tem componentes SP, Time e Velocity. O componente SP é do tipo obj\_Geom, sendo Carro:Point; e Comboio:Line.

A seguir são definidas as classes para os objetos presentes na aplicação exemplo.

```
class Semaforo inherit SpatioObject type
  tuple(numero: string,
        estado: list(tuple(cor:string, time:Time)),
        SP:Point)

end;

class Rua inherit SpatioObject type
  tuple(nome: string,
        estado: list(tuple(interditada:boolean, time:Time)),
```

```

        zona: Zona,
        SP:Line)
end;

class Zona inherit SpatioObject type
    tuple(nome: string,
          SP:Polygon)
end;

class Linha_Bloqueio inherit ChangeObject type
    tuple(numero: string,
          SpT: LineT)
end;

class Rio_Lago inherit ChangeObject type
    tuple(nome: string,
          tipo: integer,
          SpT: PolygonT)
end;

class Carro inherit MotionObject type
    tuple(placa: string,
          proprietario: list(tuple(RG:string, time:Time)),
          traj: Trajectory(tuple(l:Point, t:Time, v:Velocity)))
end;

class Comboio inherit MotionObject type
    tuple(placa: string,
          proprietario: list(tuple(RG:string, time:Time)),
          traj: Trajectory(tuple(l:Point, t:Time, v:Velocity)))
end;

```

## 5.3 Predicado Temporal

Um predicado temporal é aquele que expressa uma relação temporal. Ele pode ser aplicado sobre qualquer componente temporal de objetos estáticos, de mudança discreta e móveis. Esta seção trata apenas destes componentes temporais, sendo os espaço-temporais deixa-

dos para a seção seguinte.

Exemplos:

1. objeto estático (pontual) - "Selecione os semáforos que ficaram vermelho no período entre 12:00 01/01/2000 e 13:00 01/01/2000?"

```
select s
from s in Semaforo, e in s.estado
where e.cor = "vermelho" and
e.time->t_contains(interval(event("12:00 01/01/2000"),
event("13:00 01/01/2000")))
```

O predicado temporal usado nessa consulta é T\_CONTAINS.

2. objeto estático (linear) - "Quais são as ruas interditadas no dia 01/01/2000?"

```
select r
from r in Rua, e in r.estado
where e.interditado = true
and e.time->t_equals(day("01/01/2000"))
```

O predicado temporal usado nessa consulta é T\_EQUALS.

3. objeto móvel - "Qual é o proprietário do carro A no ano 2004?"

```
select p.RG
from c in Carro, p in c.proprietario
where p.time->t_equals(year(2004))
```

O predicado temporal usado nessa consulta é T\_EQUALS.

Estas três consultas envolvem um atributo temporal: na primeira consulta, a cor do objeto semáforo; na segunda consulta, o estado de trânsito do objeto rua; na terceira, o proprietário do carro. Todos estes atributos são temporais pois mudam com tempo.

## 5.4 Predicado Espacial

As consultas com predicados espaciais selecionam objetos que satisfazem um relacionamento estritamente espacial. Como objetos móveis e de mudança discreta são objetos espaço-temporais, aqui só será apresentado um exemplo para objetos fixos.

1. objetos fixos - "Quantos semáforos existem na zona Norte da cidade?"

```
select count(s)
from s in Semaforo, z in Zona
where s->sp_inside(z->sp) and z.nome="Norte"
```

O predicado espacial utilizado nessa consulta é SP\_INSIDE.

## 5.5 Predicado espaço-temporal

Predicados espaço-temporais podem ser divididos em métrica-temporal e topológica-temporal

### 5.5.1 Predicados métrico-temporais

Predicados métrico-temporais podem ser:

Unário: retornam área, comprimento, perímetro ou velocidade que satisfaçam uma certa condição num dado tempo;

Binário: retornam a distância entre dois objetos num dado tempo.

Exemplos:

1. objeto móvel e objeto fixo - "Quais são os carros que em ano 2004 já ultrapassaram 120km/h nas ruas que ultrapassam 5km de comprimento?"

Essa consulta pode ser feita em dois passos:

- 1) q1 seleciona as ruas que ultrapassam 5km de comprimento (predicado métrico)

```
define q1 as select r from Rua where r->sp_length() > 5
```

- 2) seleciona os carros que já ultrapassaram 120km/h em q1 no ano 2004.

```
select c from c in Carro, r in q1 where
c->m_velocity(c,vwhen(c->m_cross(r, YEAR(2004)))) > 120
```

O operador métrico-temporais usado nesta consulta é M\_VELOCITY, o operador espacial usado é SP\_LENGTH e operador temporal usado é VWHEN.

2. dois objetos móveis - "Qual é a velocidade do carro de placa A no dia 01/03/2002, sempre que sua distância ao carro B foi menor que 1km?"

Fazendo essa consulta em três passos:

- 1) Selecionar os carros A e B.

```
define q1 as element (select c
                      from c in Carro
                      where c.placa ="A")
```

```
define q2 as element (select c
                      from c in Carro
                      where c.placa ="B")
```

- 2) determinar os intervalos de tempo onde o carro A dista menos de 1km do carro B no dia 01/03/2002

```
define q3 as vwhen(q1->m_distance(q2, DAY(01/03/2002)) < 1)
```

- 3) determinar a velocidade do carro A quando dista menos de 1km do carro B no dia 01/03/2002.

```
select set(tuple(vel: c->m_velocity(q1, t), time:t)) from t in q3
```

Os operadores métrico-temporais usados nessa consulta são M\_DISTANCE e M\_VELOCIT

Os exemplos 1 e 2 demonstram o uso de operadores métrico-temporais móveis: M\_VELOCIT M\_DISTANCE. O operador M\_VELOCITY é estritamente para objetos móveis; já no caso do operador M\_DISTANCE, os objetos A e B podem ser combinação entre objetos móveis, de mudança discreta e objeto estático.

## 5.5.2 Predicados topológico-temporais

Os operadores topológico-temporais para objetos móveis são todos binários. Os básicos são M\_DISJOINT, M\_MEET, M\_OVERLAP, M\_INSIDE. Utilizando os operadores básicos, pode-se usar operadores compostos tais como M\_ENTER, M\_LEAVE e M\_CROSS.

Exemplos:

1. objeto móvel(pontual) e objeto fixo - "Quantos carros (C) cruzaram a zona Sul durante ano de 2003?"

Essa consulta pode ser feita em dois passos:

- 1) Selecionar a zona sul

```

:
define q1 as select z
from z in Zona
where z.name = "sul"

```

- 2) Selecionar o número de carros

```

select count(c)
from c in Carro,
z in q1 where c->m_cross(z, YEAR(2003))

```

O predicado topológico temporal (móvel composta) utilizado nessa consulta é M\_CROSS.

2. objeto móvel(linear) e objeto fixo - "Quais são as zonas(Z) de São Paulo que não são frequentadas os comboios de caminhão(C) desde 2000?"

```
select z from z in Zona, c em Caminhao where
c->m_disjoint(z,interval(year(2000), NDW))
```

O predicado topológico temporal utilizado nessa consulta é M\_DISJOINT.

3. objeto móveis - "Determine se o lugar em que o carro de placa A colidiu com carro de placa B no dia 03/03/2003 pertence à Zona Leste."

Essa consulta pode ser feita em três passos:

- 1) Determinar os dois carros e a zona Leste:

```
define q1 as element (select c
                        from c in Carro
                        where c.placa ="A")

define q2 as element (select c
                        from c in Carro
                        where c.placa ="B")

define q3 as element (select z
                        from z in Zona
                        where z.name="Leste")
```

- 2) determinar o lugar onde o carro A colidiu com o carro B

```
define q4 as m_sp(q1, vwhen(q1->m_meet(q2,DAY(01/03/2002)))
```

- 3) determinar se o local de colisão pertence a zona Leste

```
select l
from l in q4
where l->sp_inside(q3)
```

O predicado topológico-temporal utilizado nessa consulta é `M_MEET`. Se `l` é vazio, a resposta é negativa (o local de colisão não pertence a zona Leste).

Os exemplos 1 e 2 demonstram o uso simples de `M_CROSS` e `ST_DISJOINT`. O exemplo 3 mostra que os operadores espaço-temporais podem ser aninhados. Ele mostra que o operador `ST_SP(Obj, OPER_ESPAÇO-TEMPORAL())` retorna um resultado espacial para servir de parâmetro de entrada para o operador espacial `INSIDE`.

## 5.6 Consultas sobre o futuro

Todas as consultas temporais descritas anteriormente podem se referir a um tempo futuro. As estimativas são baseadas no histórico das trajetórias. Os métodos usados para estimativa de trajetória são tratados em [SM01, Pfo00]. Além disso, métodos estatísticos de estimativa também podem ser utilizados.

## 5.7 Resumo

Este capítulo apresentou exemplos de consultas que podem ser feitas envolvendo uma base de objetos móveis. As consultas foram classificadas de acordo com o critério de validação, usa o modelo descrito no Capítulo 3 e os operadores descritos no Capítulo 4. Os exemplos ilustraram que os operadores e o modelo proposto permitem que objetos estáticos, de mudança discreta e contínua podem estar presentes na mesma consulta e que os operadores temporais, espaciais, espaço-temporais e móveis podem ser aninhados para criar consultas complexas.

# Capítulo 6

## Conclusões e Extensões

As pesquisas feitas na área de modelo de dados para objetos móveis têm sido focadas em objetos pontuais, já os objetos fixos e de mudança discreta têm sido descartados no seu domínio de problemas. Além disso, a maioria dos modelos omite detalhes sobre os operadores e ainda mais raramente discute os algoritmos associados.

No entanto, objetos móveis só existem com os objetos fixos e de mudança discreta. Mais ainda, só se percebe a mudança dos objetos móveis quando observamos a sua interação com os demais.

Para preencher esta lacuna, a dissertação propôs um modelo de dados que engloba de forma homogênea os objetos fixos, de mudança discreta e móveis. Isso permite consultar dados sobre objetos que possuem forte interação mas de naturezas diferentes.

A dissertação também especifica um conjunto básico de operadores sobre o banco de dados e detalha os algoritmos associados. Mostra, igualmente, como estes operadores podem ser combinados para especificar consultas complexas sobre movimento e interação dos objetos.

A solução mais direta de resolver os problemas de construção de operadores para objetos móveis é modelar o tempo como terceira dimensão e definir os operadores utilizando algoritmos 3D. A análise de operadores para objetos espaço-temporais de mudança discreta mostra que muitas definições são semelhantes para objetos móveis. A diferença é que como o tempo é contínuo, ele representa uma dimensão a mais,. O problema é então reduzir esta dimensão e tornar o cálculo das operadores mais simples.

A solução encontrada pela dissertação foi a utilização das trajetórias como um registro espaço-temporal, o qual mapeia o espaço 3D em 2D. A outra dificuldade consiste em utilizar as informações das trajetórias para calcular os operadores. A dissertação propõe um algoritmo que utiliza os operadores de objetos espaço-temporal discreta para calcular os operadores de objetos móveis. Isso pode melhorar a complexidade de processamento e implementação.

Sendo assim, a dissertação apresenta as seguintes contribuições:

- Proposta de um modelo de objetos que considera objetos 1D e 2D e trata de maneira uniforme objetos de mudança discreta e objetos móveis (vide Capítulo 3).
- Especificação de um conjunto básico de operadores que leva em consideração as características de mobilidade. Estes operadores são apresentados junto com o detalhe dos algoritmos respectivos (vide Capítulo 4).
- Validação da proposta pela instanciação de exemplos de consultas utilizando o modelo e operadores propostos (vide Capítulo 5).

Bancos de dados de objetos móveis constituem um assunto de pesquisa recente e muitos tópicos ainda precisam ser estudados e explorados. Sabemos que a definição de um modelo de dados é apenas um começo do trabalho. No contexto desta dissertação, possíveis extensões futuras são:

- descrever os algoritmos dos operadores móveis para objetos que possuem rotação assim como objetos mutáveis que se fragmentam e se fundem.
  - estender o modelo para objetos tri-dimensionais.
  - incluir conceitos de incerteza nas consultas sobre futuro.
  - estudar sobre a complexidade dos algoritmos propostos.
- 
- implementar a proposta para um sistema real.
  - analisar o uso de índices espaciais para melhorar a performance das consultas sobre objetos móveis.

# Referências Bibliográficas

- [Bar99] D. Barbará. Mobile computing and databases-a survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):108–117, 1999.
- [Bot95] M.A. Botelho. Incorporação de facilidades espaço-temporais em banco de dados orientado a objetos. Dissertação de Mestrado, Universidade Estadual de Campinas, 1995.
- [CAA01] H.D. Chon, D. Agrawal, e A. El Abbadi. Storage and retrieval of moving objects. Em *Proceedings of the Second International Conference on Mobile Data Management*, páginas 173–184. Springer-Verlag, 2001.
- [cal] Caltrans new technology and research program. advanced transportation system program plan.
- [CCH<sup>+</sup>96] G. Câmara, M.A. Casanova, A.S. Hemerly, G.C. Magalhães, e C.B. Medeiros. Anatomia de sistema de informação geográfica. *X Escola de Computacao*, 1996.
- [CFvO93] E. Clementini, P. Di Felice, e P. van Oosterom. A small set of formal topological relationships suitable for end-user interaction. Em *Proceedings of the Third International Symposium on Advances in Spatial Databases*, páginas 277–295. Springer-Verlag, 1993.
- [Cil96] M. Cilia. Bancos de Dados Ativos com Suporte a Restrições Topológicas em Sistemas de Informação Geográfica. Dissertação de Mestrado, UNICAMP, march 1996.
- [Eas90] C.M. Eastman. Vector versus raster: a functional comparison of drawing technologies. *IEEE Comput. Graph. Appl.*, 10(5):68–80, 1990.
- [EAT92] M.J. Egenhofer e K.K. Al-Taha. Reasoning about gradual changes of topological relationships. Em *Proceedings of the International Conference GIS - From Space to Territory: Theories and Methods of Spatio-Temporal Reasoning*

- on *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, páginas 196–219. Springer-Verlag, 1992.
- [EGSV97] M. Erwig, R.H. Güting, M. Schneider, e M. Vazirgiannis. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. Relatório Técnico, FemUniversit t Hagen, Dec. 1997.
- [EGSV98] M. Erwig, R.H. Güting, M. Schneider, e M. Vazirgiannis. Abstract and discrete modeling of spatio-temporal data types. Em *Proceedings of the sixth ACM international symposium on Advances in geographic information systems*, páginas 131–136. ACM Press, 1998.
- [ES99] M. Erwig e M. Schneider. Developments in spatio-temporal query languages. Em *Proceedings of the 10th International Workshop on Database & Expert Systems Applications*, páginas 441–449. IEEE Computer Society, 1999.
- [ESM93] M. Egenhofer, J. Sharma, e D. Mark. A critical comparison of the 4-intersection and 9-intersection models for spatial relations. *Autocarto 11, Minneapolis, MN*, páginas 1–11, 1993.
- [Far98] G. Faria. Um banco de dados espaço-temporal para desenvolvimento de aplicações em sistemas de informação geográfica. Dissertação de Mestrado, Universidade Estadual de Campinas, 1998.
- [FGNS00] L. Forlizzi, R.H. Güting, E. Nardelli, e M. Schneider. A data model and data structures for moving objects databases. *SIGMOD Rec.*, 29(2):319–330, 2000.
- [GBE+00] R.H. Güting, M.H. Böhlen, M. Erwig, C.S. Jensen, N.A. Lorentzos, M. Schneider, e M. Vazirgiannis. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.*, 25(1):1–42, 2000.
- [JCE+94] C.S. Jensen, J. Clifford, R. Elmasri, S. Gadia, P. Hayes, e S. Jajodia. A consensus glossary of temporal database concepts. *SIGMOD Rec.*, 23(1):52–64, 1994.
- [KHM+98] J.T. Klosowski, M. Held, J.S.B. Mitchell, H. Sowizral, e K. Zikan. Efficient collision detection using bounding volume hierarchies of  $k$ -DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, /1998.
- [MRSS01] J. Moreira, C. Ribeiro, J.M. Saglio, e M. Scholl. Issues in spatio-temporal database systems: data models, languages and moving objects. Em *The encyclopedia of life support systems*. UNESCO-EOLSS, 2001.

- [Peu01] D.J. Peuquet. Making space for time: Issues in space-time data representation. *GeoInformatica*, 5(1):11–29, 2001.
- [Pfo00] D. Pfoser. *Issues in the Management of Moving Poing Objects*. Tese de Doutorado, Department of Computer Science The Faculty of Engineering and Science Aalborg University, September 2000.
- [RA] K.H. Ryu e Y.A. Ahn. Application of moving objects and spatiotemporal reasoning.
- [SJLL99] S. Saltenis, C.S. Jensen, S.T. Leutenegger, e M.A. Lopez. Indexing the position of coninuously moving objects. Relatório Técnico, A TimeCenter Technical Report, 1999.
- [SM01] J.M. Saglio e J. Moreira. Oporto: A realistic scenario generator for moving objects. *Geoinformatica*, 5(1):71–93, 2001.
- [Sno92] R. Snodgrass. Temporal Databases. Em *Proc Intl. Conf. on GIS - From Space to Territory: Theories and Methods of Spatial Reasoning*, Springer Verlag Lecture Notes in Computer Science 639, páginas 22–63, 1992.
- [Sno95] R.T. Snodgrass. The tsq!@ temporal query language. *Kluwer Academic Publishers*, 1995.
- [SWCD97] A.P. Sistla, O. Wolfson, S. Chamberlain, e S. Dao. Modeling and querying moving objects. Em *Proc. IEEE Intl. Conf. On Data Engeneering*, páginas 422–432, Birmingham, UK, 1997.
- [SWCD98] A.P. Sistla, O. Wolfson, S. Chamberlain, e S. Dao. Querying the uncertain position of moving objects. *Lecture Notes in Computer Science*, 1399:310–321, 1998.
- [TP95] Y. Theodoridis e D. Papadias. Range Queries Involving Spatial Relations: A Performance Analysis. Em *Lectures Notes in Computer Science*, páginas 537–551, 1995.
- [WXCJ98] O. Wolfson, B. Xu, S. Chamberlain, e L. Jiang. Moving objects databases: Issues and solutions. Em *Statistical and Scientific Database Management*, páginas 111–122, 1998.
- [YM02] B. Yi e C.B. Medeiros. Um modelo de dados para objetos móveis. *GeoInfo*, páginas 33–40, 2002.