

Indexação e métodos de busca de objetos espaço-temporais: um estudo de caso

ETORE MARCARI JUNIOR

16/06/2016



Sumário

- Motivação
 - Ideia inicial
 - O que existe hoje: R-Tree
 - Indexação de dados espaço-temporais
 - Estudo de caso: TB-Tree
 - Metodologia e métodos estudados
 - Considerações
- 

Motivação

- Apesar de não aparecer para os usuários, estruturas de dados estão presentes internamente nas diferentes aplicações e SIGs.

Motivação

- Apesar de não aparece presentes internamente



estruturas de dados estão ações e SIGs.

Motivação

- Apesar de não aparecer para os usuários, estruturas de dados estão presentes internamente nas diferentes aplicações e SIGs
- Indexadores: estruturas que:
 - Otimizam acesso aos dados
 - Melhoram o desempenho de consultas (quando em bancos de dados)
 - Facilitam a “manipulação”

Ideia inicial

- Procurar na literatura um tipo de indexador de dados espaço temporais e verificar sua viabilidade de implementação

Ideia inicial

- Procurar na literatura um tipo de indexador de dados espaço temporais e verificar sua viabilidade de implementação
- Qual escolher?
- Como escolher?
- Para qual finalidade?

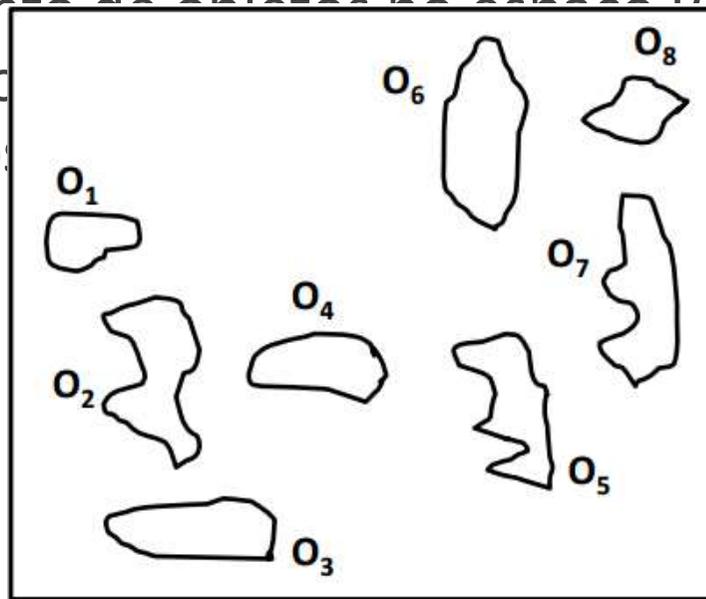
O que existe hoje: R-Tree

- R-Tree: estrutura de dados
- Dado um conjunto de objetos no espaço R^k ($k > 1$), uma R-tree organiza o espaço subjacente em uma hierarquia de intervalos k -dimensionais (possivelmente com sobreposições)

O que existe hoje: R-Tree

- R-Tree: estrutura de dados

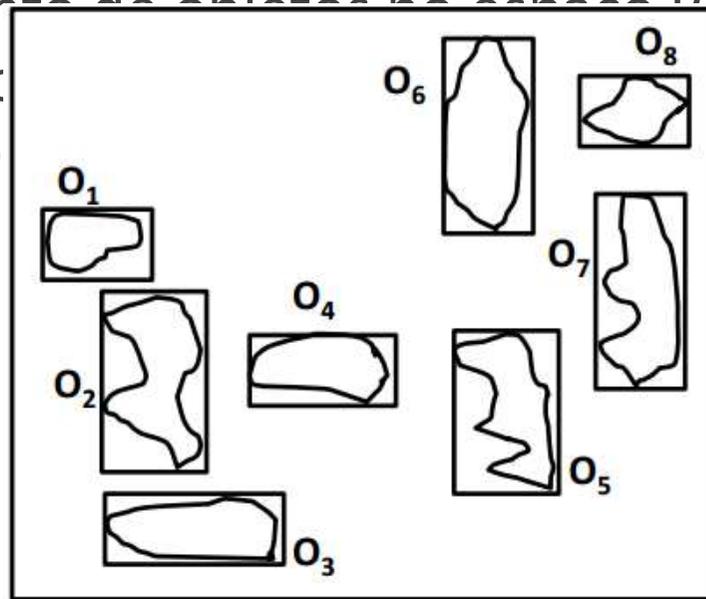
- Dado um conjunto de objetos no espaço R^k ($k > 1$), uma R-tree organiza o espaço em intervalos k -dimensionais (porções)



O que existe hoje: R-Tree

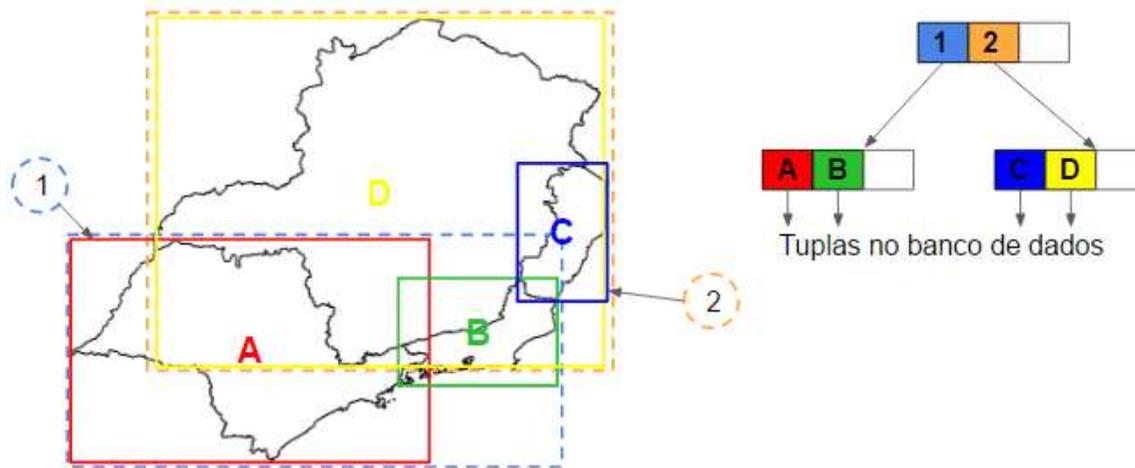
- R-Tree: estrutura de dados

- Dado um conjunto de objetos no espaço D^k ($k > 1$), uma R-tree organiza o espaço em intervalos k -dimensionais (porções)



O que existe hoje: R-Tree

- Intervalos são organizados em uma estrutura de árvore
- Diferentes tipos de nós:
 - o Nós internos
 - o Nós folha

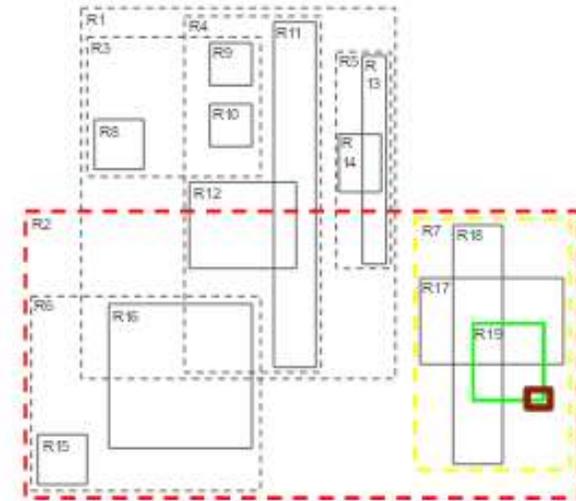
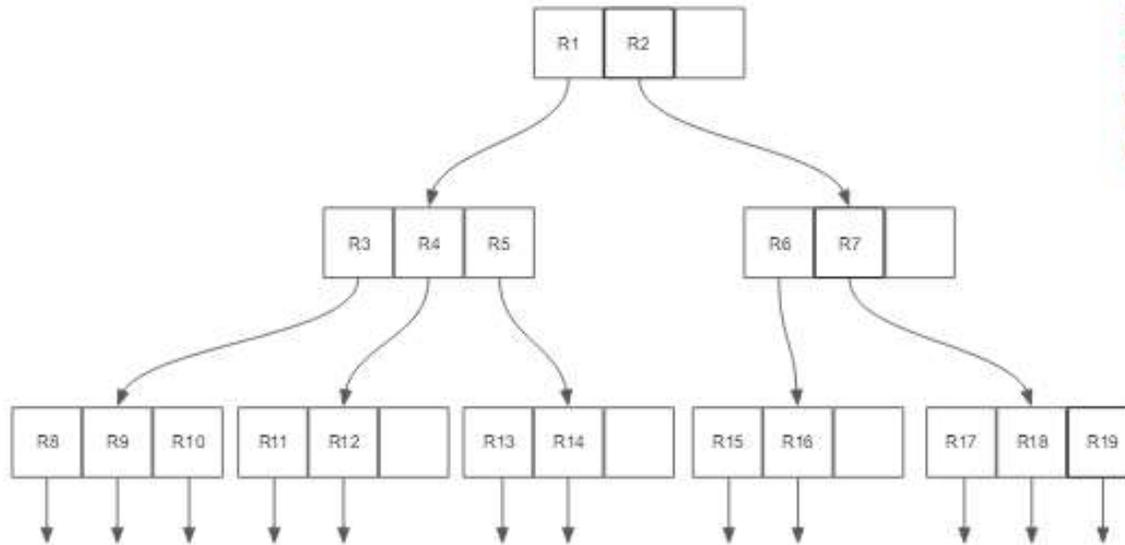


O que existe hoje: R-Tree

- Segue uma série de regras
- Deve estar sempre balanceada

O que existe hoje: R-Tree

- Segue uma série de regras
- Deve estar sempre balanceada



O que existe hoje: R-Tree

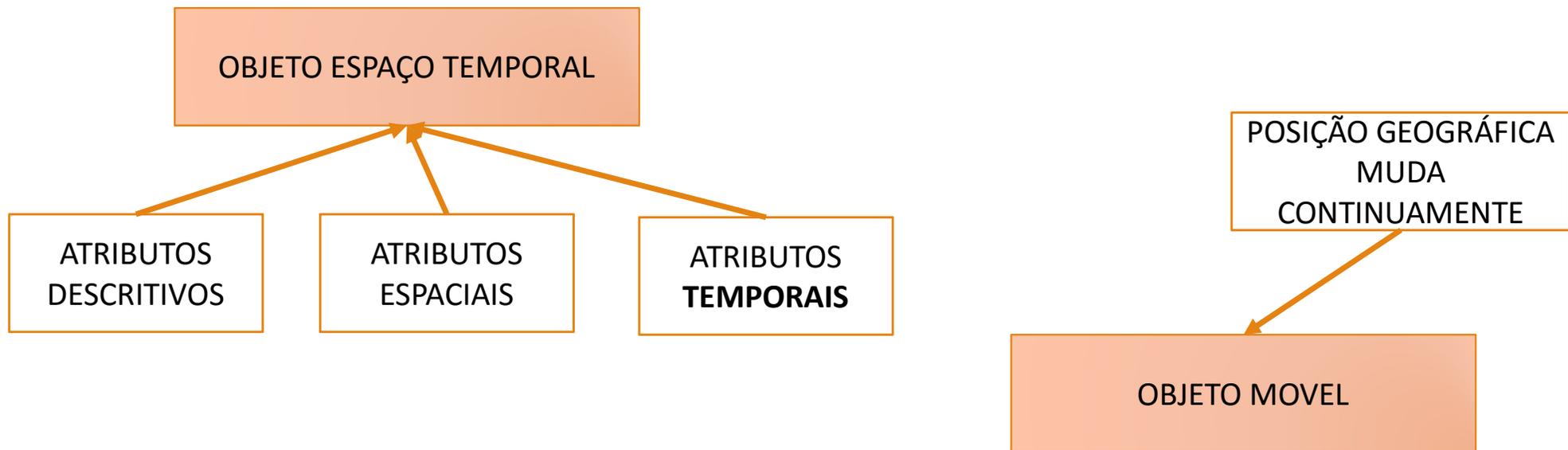
- Segue uma série de regras
- Deve estar sempre balanceada
- Possui algoritmos bem definidos e diferentes implementações
- Amplamente utilizada pela comunidade
- Mas...

O que existe hoje: R-Tree

- Segue uma série de regras
- Deve estar sempre balanceada
- Possui algoritmos bem definidos e diferentes implementações
- Amplamente utilizada pela comunidade
- Mas...

funciona bem para dados estáticos, que não variam
com o decorrer do tempo

Objetos espaço-temporais e objetos móveis



Indexação de objetos espaço-temporais

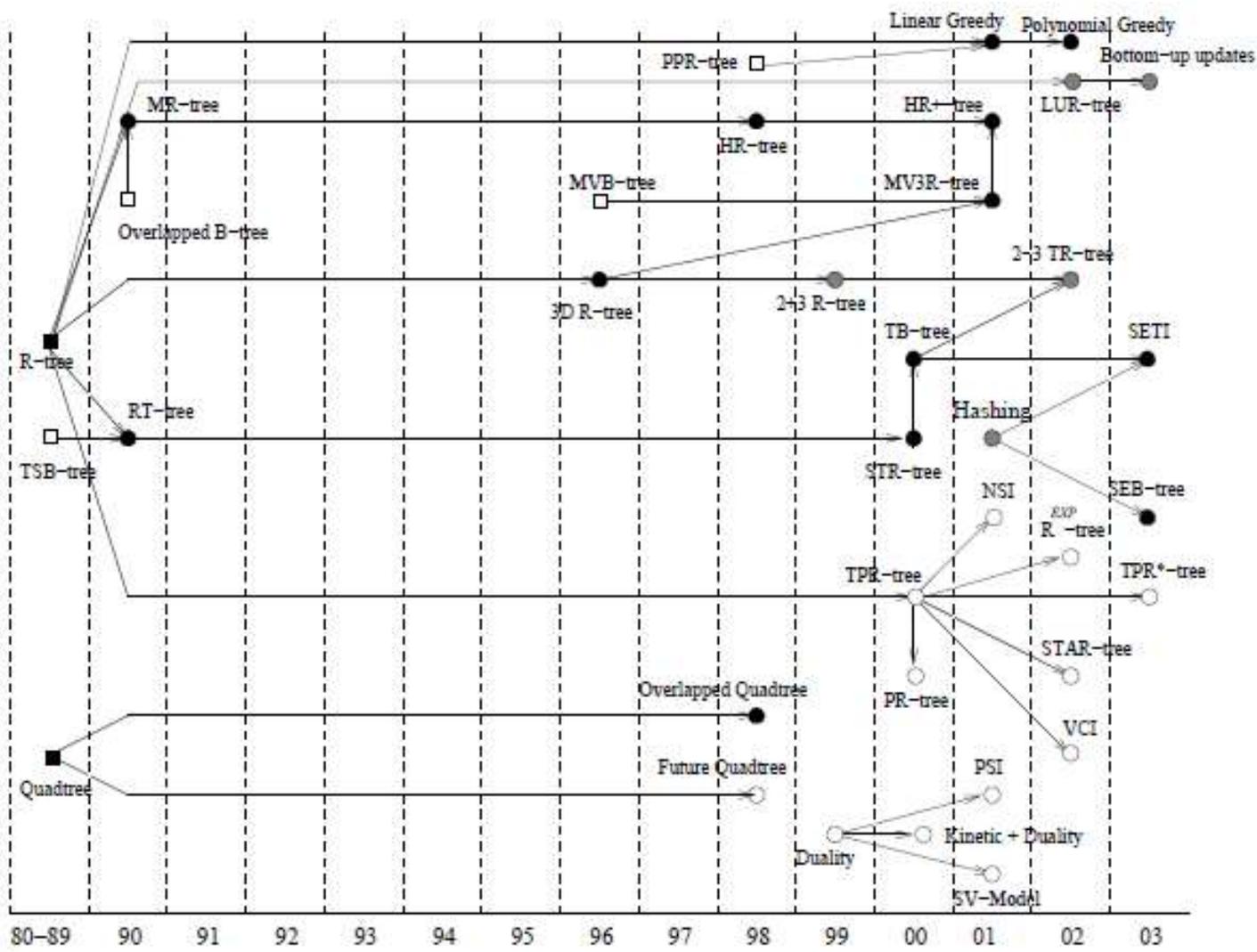
- Buscam responder três grupos de perguntas:



- Sobre a posição de objetos no passado
- Sobre a posição dos objetos no “agora” (tempo corrente)
- Sobre onde estarão os objetos no futuro (predição)

In

- B

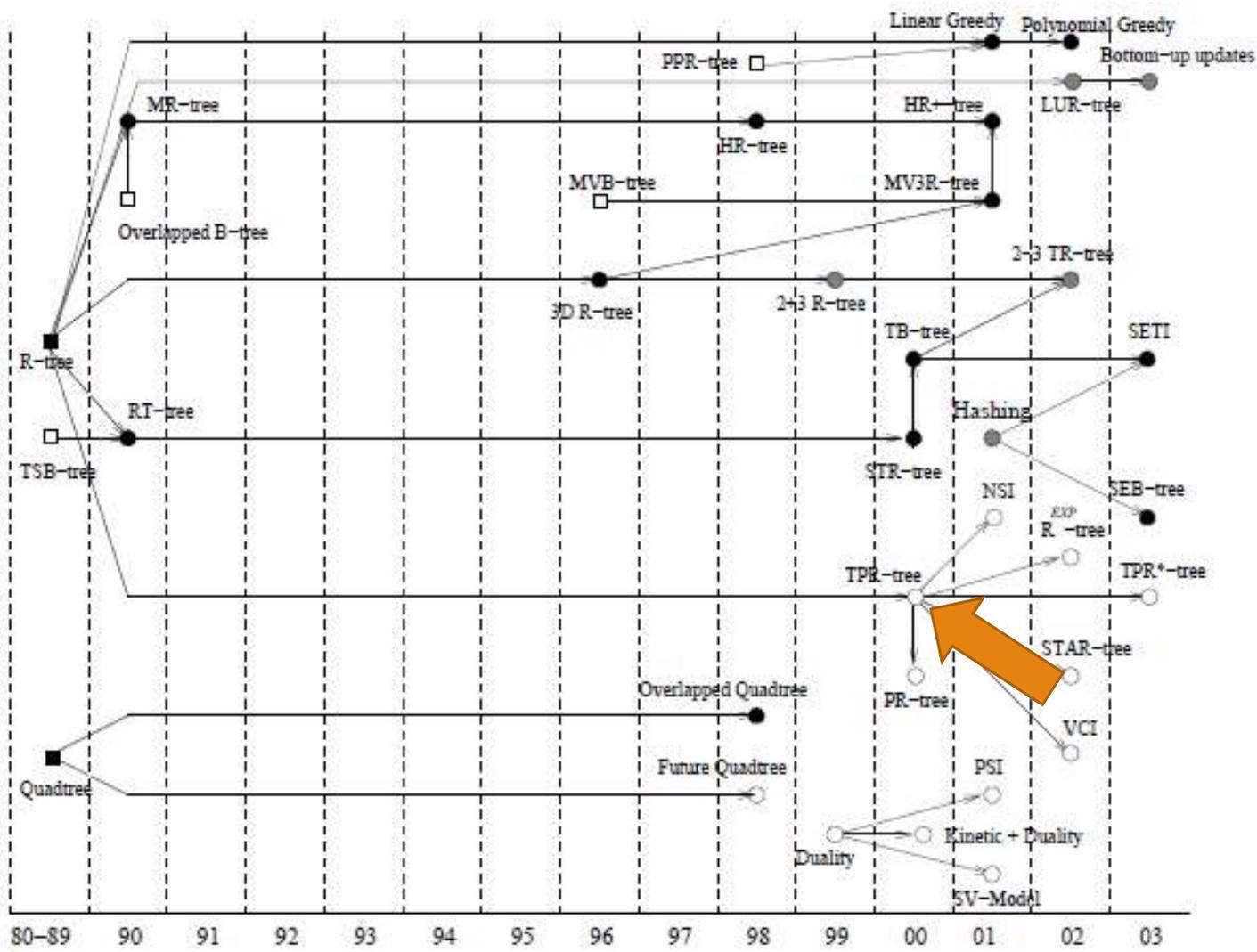


o rais

- Passado
- Presente
- Futuro
- Espacial
- Temporal

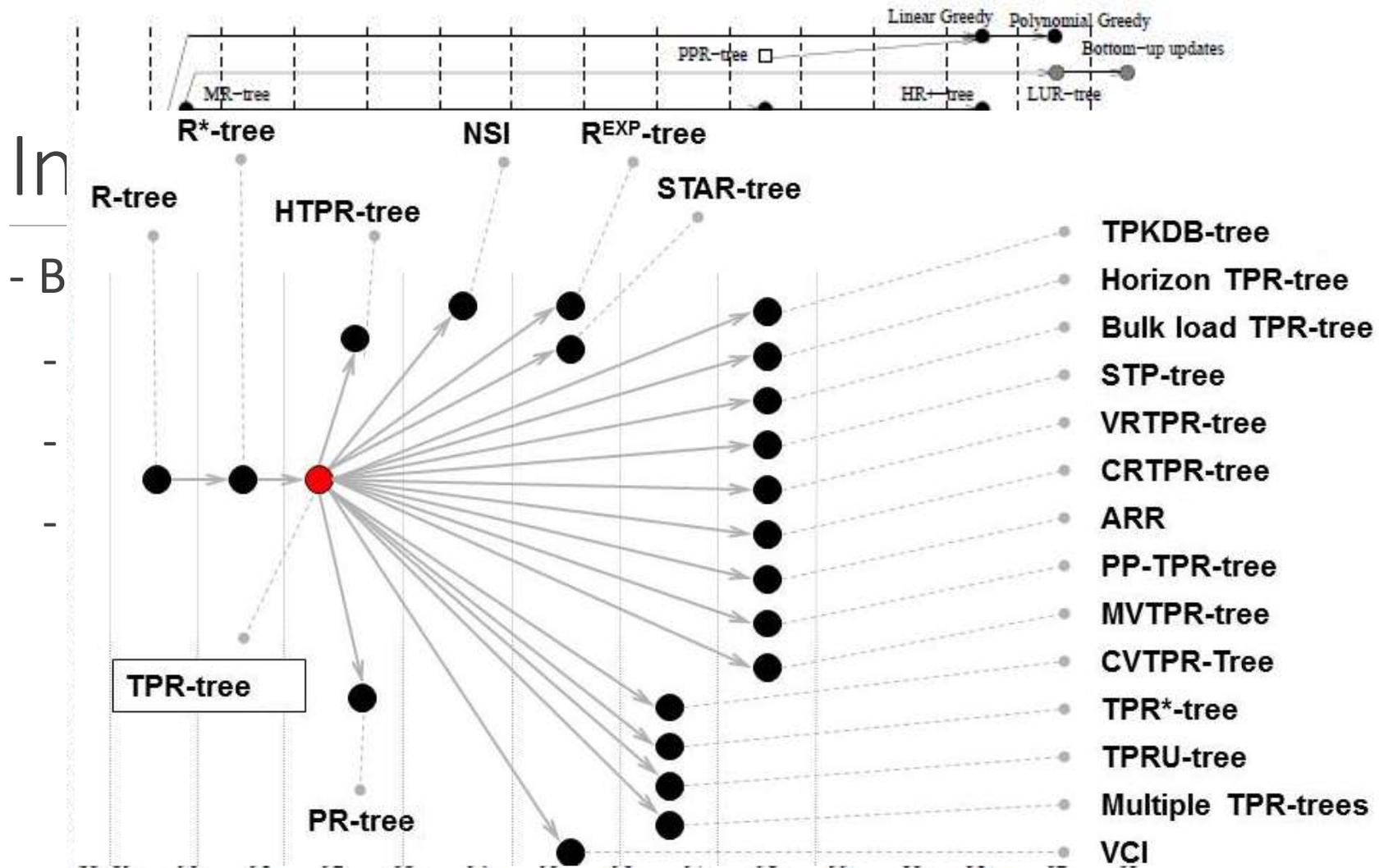
In

- B

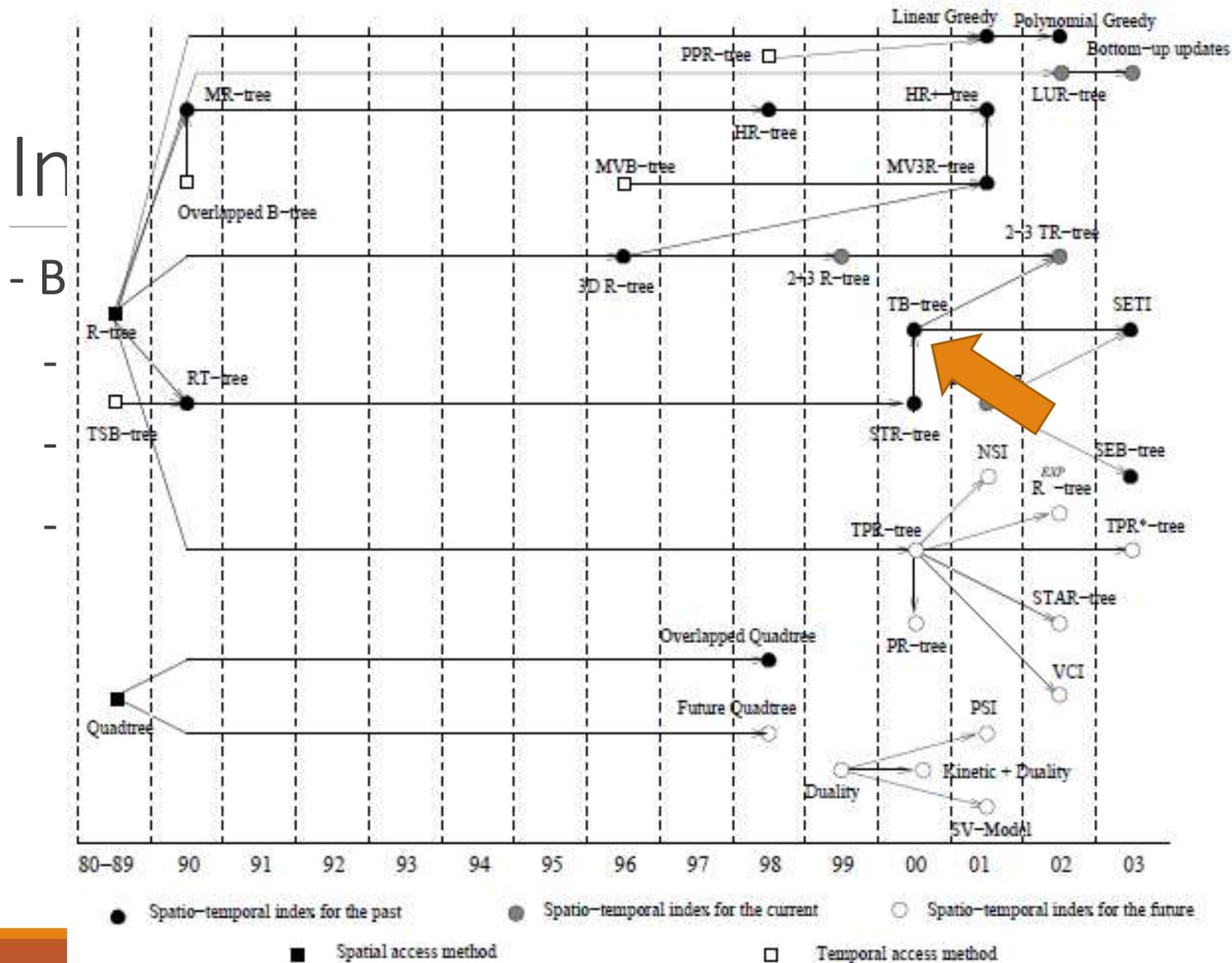


o rais

- Passado
- Presente
- Futuro
- Espacial
- Temporal



Horais



Indexação de objetos espaço-temporais

- Além das 3 perguntas anteriores..

Outra pergunta fundamental: o que desejamos indexar?

Indexação de objetos espaço-temporais

- Além das 3 perguntas anteriores..

Outra pergunta fundamental: o que desejamos indexar?

Trajetória de objetos (barcos) na costa do país

Estudo de caso

- Necessidades a serem atendidas:
 - Qual a posição de um barco X durante determinado período de tempo?
 - Retorne a trajetória do barco X no intervalo de tempo (t_i, t_f)
 - Retorne todos os barcos que estavam em determinada posição em um intervalo de tempo

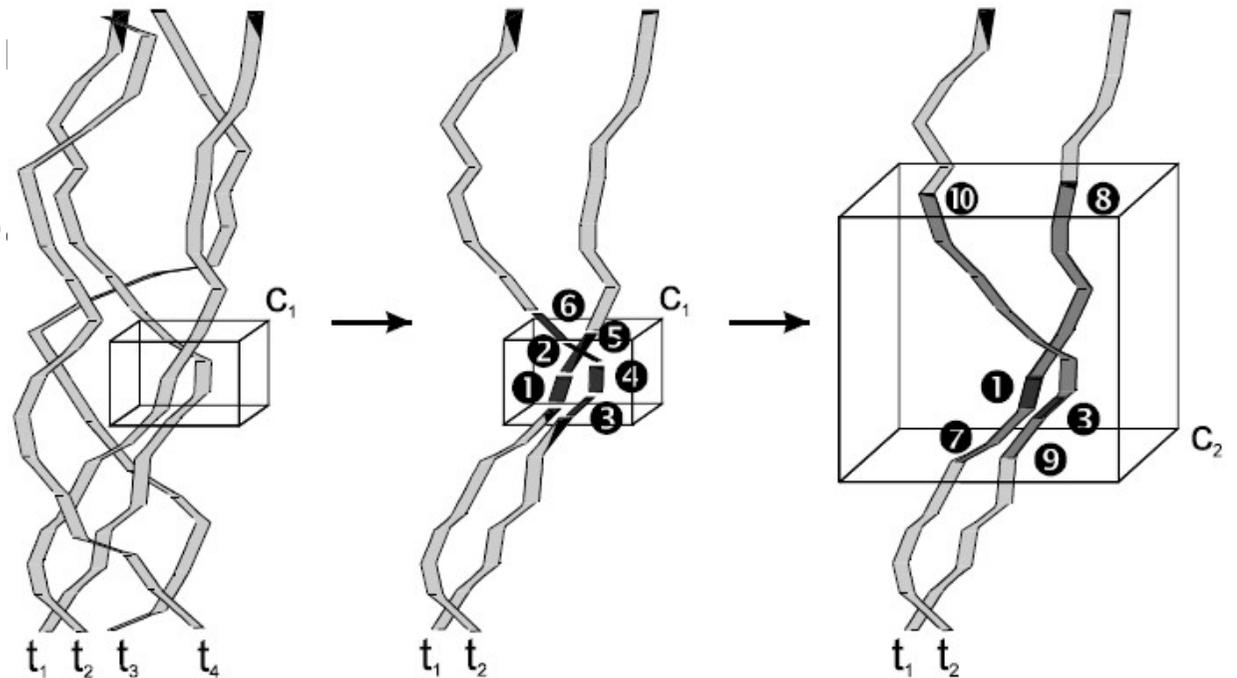
Estudo de caso

- Necessidades a serem atendidas:

- Qual a posição de um barco
tempo?

- Retorne a trajetória do b

- Retorne todos os barcos
em um intervalo de tempo



Estudo de caso

- Estrutura de dados de indexação escolhida:

TB-Tree

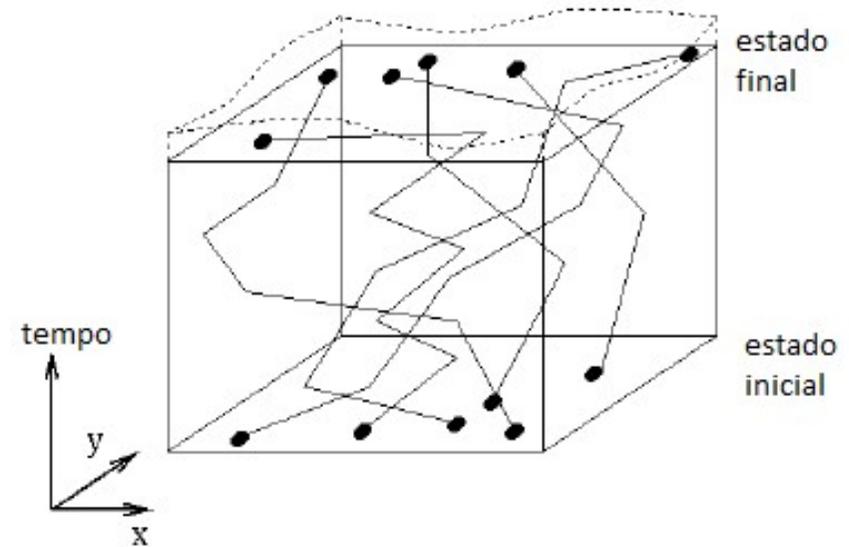
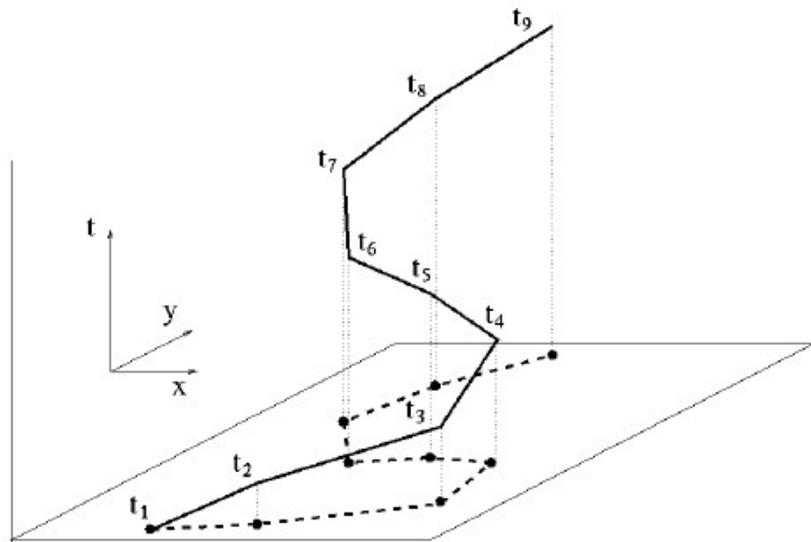
Trajectory-Bundle Tree

Estudo de caso: TB-Tree

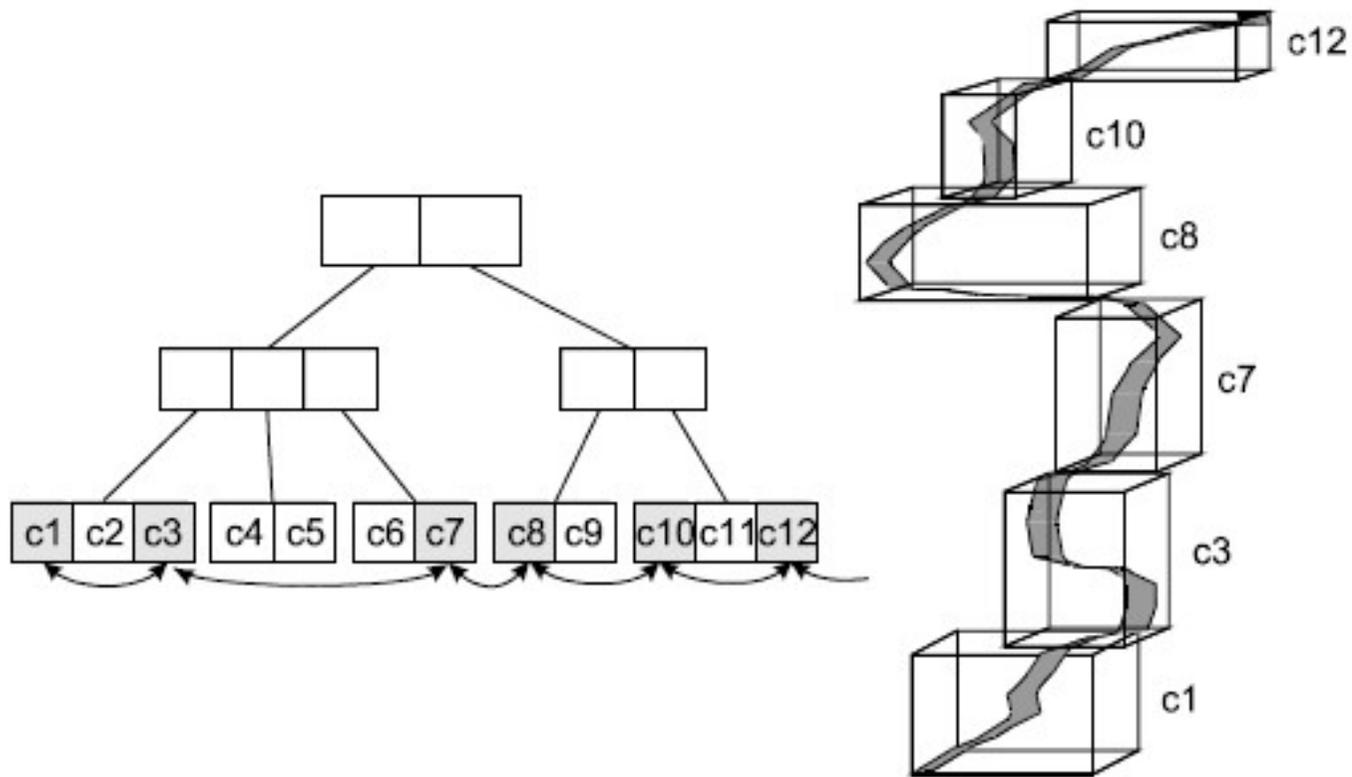
Características especiais:

- Possui o conceito de relação entre diferentes nós existentes nas folhas
- Usa artifício de “linkar” os nós folhas que representam a mesma trajetória
- Na teoria, apresenta melhor desempenho que estruturas criadas anteriormente
- Ignora algumas características da R-Tree convencional, como o número mínimo de nós e proximidade entre os itens

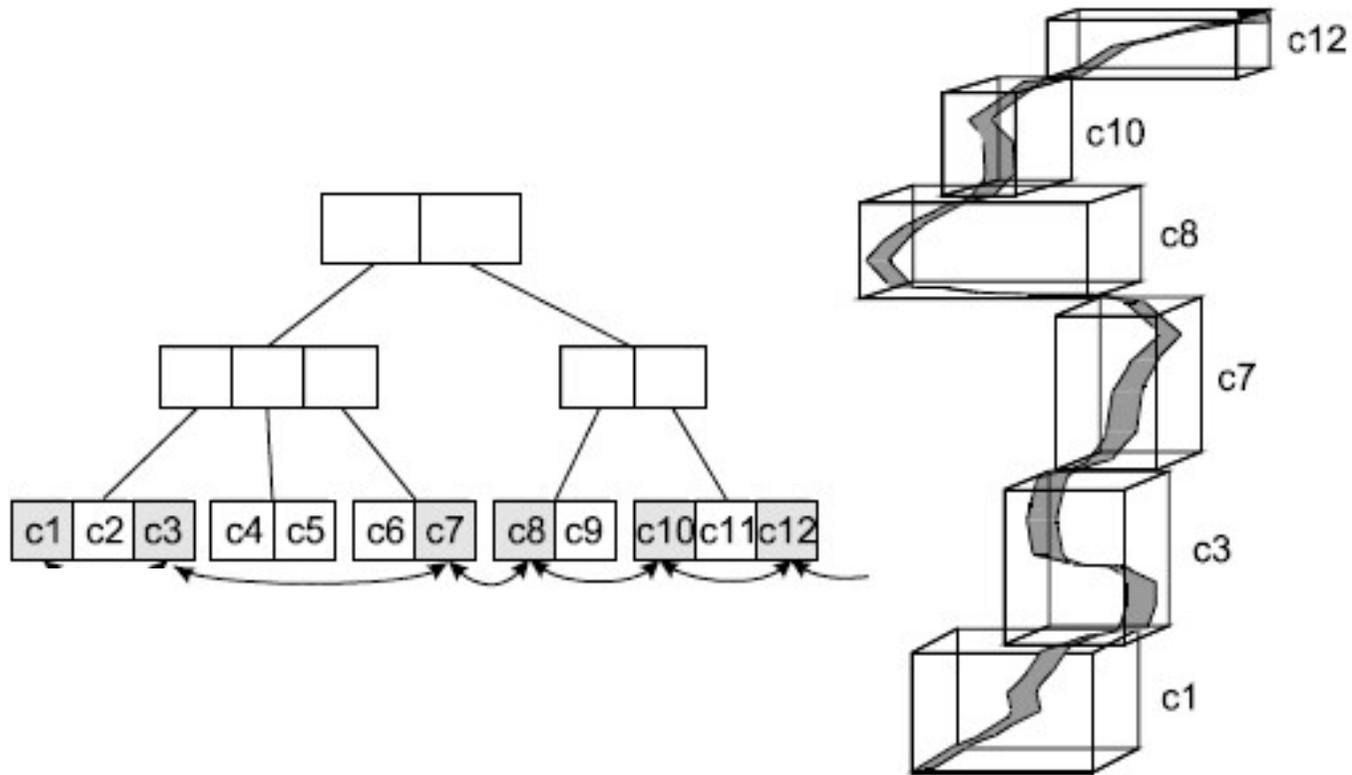
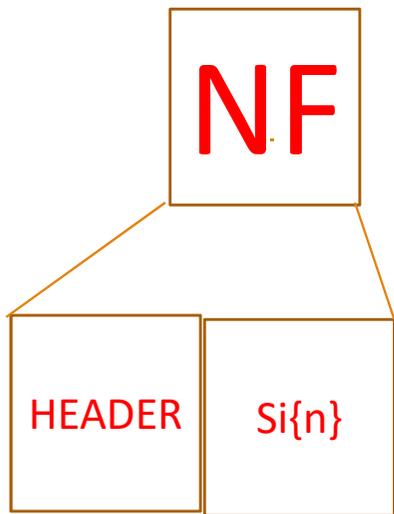
Estudo de caso: TB-Tree



Estudo de caso: TB-Tree

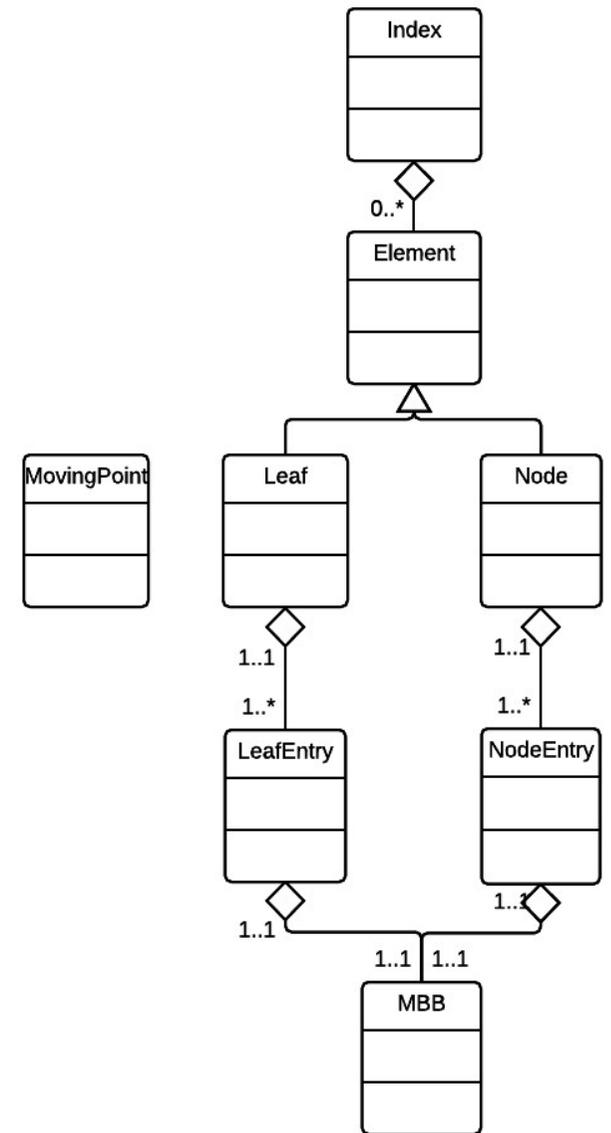
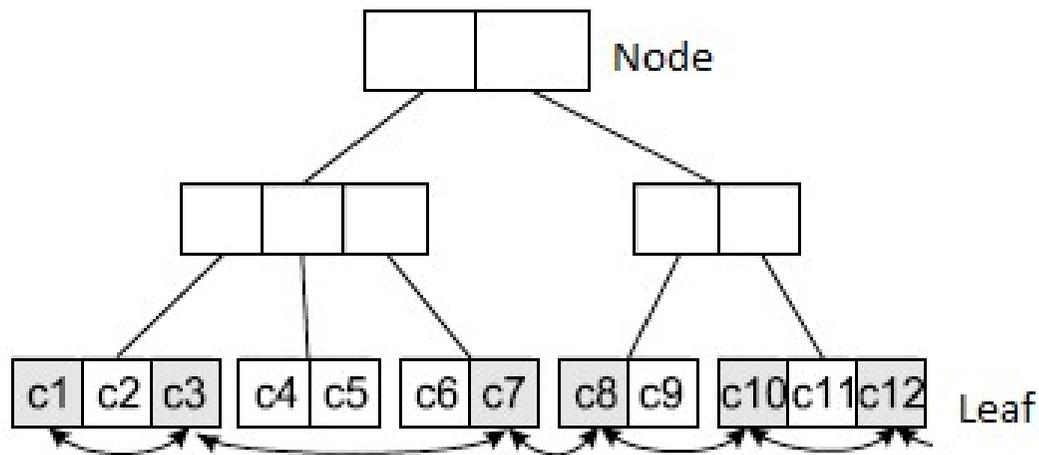


Estudo de caso: TB-Tree



Id
#entradas
ptrCorr
ptrPai
ptrProximo (com id igual)
ptrAnterior (com id igual)

Estudo de caso: TB-Tree



Métodos a serem implementados

Classe tbtree

- Insert(N, E)
- FindNode(N, E)
- CombinedSearch(N, range1, range2)
- SplitNode(N)
- RemoveNode(N)
- Clear()

Considerações

- Apresentação de diversos indexadores feita por artigos científicos
 - Algumas vezes incompletos, ou sem apresentar os algoritmos
- Vasto campo de estudo
- Não é possível destacar um padrão seguido pelo mercado e academia
- Decisão sobre qual indexador aplicar a determinada aplicação pode levar a maus resultados

Trabalhos futuros

- Avaliar necessidade de real implementação na TerraLib5
- Implementar de fato, segundo padrão da biblioteca, o indexador
- Integrar com o que existe hoje de análise espaço temporal

Módulo st



FIM

