

Escalonamento de Transações

Relatório do Segundo Trabalho da Matéria

CI1218 - Banco de Dados

Jorge Lucas Vicilli Jabczenski
GRR20190372

22 de Abril de 2022

1 Introdução

Nesse relatório será descrito brevemente a implementação dos algoritmos de Visão Equivalente e Serialidade Por Conflito utilizados para a detecção de conflitos de escalonamento de transações concorrentes.

2 Algoritmos

2.1 Serialidade Por Conflito

Para a implementação desse algoritmo, foi utilizada uma matriz de adjacência para simular um grafo direcional. A matriz é preenchida através de um loop aninhado procurando relações de Escrita/Escrita, Escrita/Leitura e Leitura/Escrita entre as entradas.

A detecção de ciclos é feita percorrendo essa matriz e verificando recursivamente se existe uma aresta conectando os vértices (que são as representações da transações) e assim é possível dizer se existe ou não um ciclo naquele grafo.

2.2 Equivalência por Visão

O algoritmo de Equivalência por Visão foi implementado da seguinte forma: para cada permutação possível daquele agendamento, primeiro é verificado se a última escrita ainda é feita pela transação correta, ou seja, a transação que realizou a última escrita no agendamento original.

Após isso, é verificado se as ordens das relações de Escrita/Leitura são mantidas para todas as transações presentes naquele agendamento. Ou seja, se houve uma leitura de um atributo em uma transação após a escrita desse atributo em outra transação, essa ordem deve permanecer na permutação que está sendo analisada.

Caso essas duas exigências sejam atendidas, podemos dizer que o agendamento é equivalente por visão, portanto não há a necessidade de continuar percorrendo as demais permutações.

3 Estruturas de Dados

3.1 `input_t`

Estrutura que mapeia uma linha da entrada. Pode ser interpretada como uma operação dentro daquele agendamento.

3.2 `schedule_t`

Estrutura que guarda todos os dados de um agendamento. Os algoritmos de Conflito por Serialidade e Equivalência por Visão são rodados individualmente para cada agendamento.

3.3 `graph_t`

Estrutura que representa um grafo. Para guardar as relações entre os vértices foi utilizada uma matriz de adjacência.

3.4 `key`, `last_write` e `writes`

Não são exatamente estruturas de dados, mas sim mapeamentos que facilitam imensamente encontrar e guardar as relações necessárias para executar o algoritmo de Equivalência por Visão.

4 Saída

A saída do programa é uma lista de agendamentos, apresentados por sua ordem de aparição, transações que compõem aquele agendamento e duas classificações: a de serialidade por conflito (SS ou NS) e equivalência por visão (SV ou NV).