

УДК 004.621.3

## МЕТОДИКА ПРИВЕДЕННЯ РОЗКЛАДУ ТРАНЗАКЦІЇ ДО ЕКВІВАЛЕНТНОЇ ПОСЛІДОВНОЇ ФОРМИ

Іванченко І.С.

## THE METHOD OF BRINGING THE TRANSACTION TO A SCHEDULE OF EQUIVALENT SEQUENTIAL FORM

Ivanchenko I.

*Розроблено узагальнену методику приведення довільних розкладів транзакцій до еквівалентної послідовної форми. Пропонується застосування циклічного виконання певних процесів для забезпечення здатності до впорядкування будь-якого набору транзакцій, а отже забезпечення можливості їх паралельного (переміжного) виконання. А для забезпечення коректності паралельної обробки інформації необхідно організувати динамічне формування еквівалентного послідовного розкладу операцій у рамках означеного фрейму, який зсувається під час роботи системи. Вище наведені операції покладені в основу розроблених алгоритмів та дає можливість не лише достовірно розпізнати виникнення тупиків у процесі паралельного використання транзакцій, але й автоматично ліквідувати такі конфлікти і, таким чином, зводити будь-які розклади до еквівалентної послідовної форми.*

**Ключові слова:** інформаційні ресурси, каскадне скасування, транзакція, послідовна обробка, паралельна обробка, фрейми.

**Вступ.** У зв'язку з чітко вираженою тенденцією до розширення обробки інформації у режимі колективного маніпулювання інформацією, а також інтенсивним розвитком розподілених інформаційних ресурсів, особливої актуальності набуває одна із задач підтримки цілісності – задача управління процесами паралельної обробки.

Методи забезпечення цілісності інформаційних ресурсів при паралельній обробці інформації розглянуті зокрема, в [1, 2]. Найбільш поширений і надійний метод управління паралельною обробкою, пов'язаний з поняттями транзакції та розклад.

У більшості методів паралельної обробки використовується деяка форма перевірки того, чи результат виконання транзакції такий, яким він був би тоді, коли б усі транзакції, що обробляються в системі у даний інтервал часу, виконувалися послідовно. Ці методи ґрунтуються на теорії зведення процесу паралельного виконання до послідовної форми [2]. У випадку, коли транзакції виконуються послідовно, тобто виконуються усі сукупності складових операцій однієї транзакції, а потім – наступної, паралельна обробка відсутня. Послідовне виконання зумовлює неможливість взаємовпливу транзакцій,

коли результат виконання операцій однієї транзакції, спричиняв би зміну результатів виконання операцій іншої транзакції у процесі її виконання.

**Основна частина.** На підставі рекомендацій, які наведені у літературі [1, 2] розроблено узагальнену методику приведення довільних розкладів транзакцій до еквівалентної послідовної форми, що передбачає детектування в аналізованих розкладах пунктів та їх ліквідацію і полягає в наступному.

Для забезпечення коректності паралельної обробки інформації у будь-який фіксований момент часу розклад виконання операцій транзакцій (надалі – розклад), що паралельно обробляють набір елементів інформаційних ресурсів (ІР), повинен бути строго послідовним. Для забезпечення відновлюваності означеного розкладу необхідним є також дотримання умови при відповідальності даного розкладу ("... будь-яка транзакція повинна завершувати виконання з фіксацією результатів лише після того, як виконання завершили всі транзакції, що попередньо модифікували елементи інформації, якими маніпулювала дана транзакція").

Доцільним (характер цієї вимоги не є критичним для забезпечення коректності обробки інформації ІР, а швидше, стосується удосконалення роботи системи управління інформаційними ресурсами) є також забезпечення можливості запобігання каскадним скасуванням (ЗКС) транзакцій.

Вимогам послідовності може задовольнятись у двох випадках:

— операції транзакцій відразу надходять на вхід планувальника у порядку, що не викликає конфліктів під час доступу до елементів інформації, це послідовний розклад – ідеальний випадок, коли втручання планувальника в процес виконання є мінімальним;

— операції транзакцій надходять на вхід планувальника у порядку, що викликає конфлікт між цими операціями під час доступу до елементів інформації; у цій ситуації розклад може бути:

• таким, що потенційно зводиться до еквівалентного послідовного – функції

планувальника полягають у встановленні такого порядку передачі операцій до адміністратора інформації, який би усував можливі конфлікти між ними;

• таким, що до еквівалентного послідовного не зводиться – в цьому випадку необхідно застосувати методику детектування та усунення взаємоблокувань між операціями в розкладі (тупиків), після чого розклад стає послідовним або еквівалентно послідовним.

1. Набір операцій, що підлягають аналізу на предмет конфліктів та взаємоблокувань, у різні моменти часу відрізняються. Це відбувається в наслідок надходження запитів на виконання нових транзакцій, завдяки чому загальний розклад змінюється, а також, завершення деяких або всіх старих транзакцій, після чого вони перестають бути потенційною причиною виникнення нових конфліктів. Враховуючи це, доцільним є періодичний аналіз не загального розкладу операцій, що будь-коли надходили на виконання, а лише тієї його частини, що містить операції активних (не завершених на даний момент) транзакцій [3]. Таку частину загального розкладу називатимемо фреймом активних транзакцій (надалі, просто – фреймом).

2. Початковим моментом роботи планувальника є надходження набору запитів на виконання деяких транзакцій; тобто "... одночасно надійшли запити на виконання таких транзакцій ..." –  $T_1, T_2, \dots, T_n$  – це початкова позиція фрейму – надалі по мірі виконання та завершення транзакцій фрейм зсуватиметься по відношенню до глобального потоку операцій, що будь-коли надходили на виконання. Зсув фрейму відбувається наступним чином: зафіксовані або скасовані транзакції, а відтак, і їх операції, будуть видалятися з розкладу, внаслідок чого на результати виконання наступних транзакцій вони не матимуть впливу. Слід зазначити, що в момент після фіксації даної конкретної транзакції відбувається зняття усіх блокувань, що їх ця транзакція встановила (вимоги протоколу двофазного блокування).

3. Натомість у процесі роботи системи надходитимуть все нові запити на виконання нових транзакцій, причому операції, що повинні виконуватись в контексті даної нової транзакції надходять паралельно з операціями інших транзакцій, у тому числі і тих, що вже почали виконуватись до цього моменту. Оскільки надходження їх є асинхронним по відношенню до решти операцій у розкладі, останній буде динамічно змінюватись в тому розумінні, що повний набір операцій нової транзакції не обов'язково буде додано одним суцільним блоком у кінець існуючого розкладу, а буде розосереджено в ньому.

4. Таким чином, для забезпечення коректності паралельної обробки інформації необхідно організувати динамічне формування еквівалентного послідовного розкладу операцій у рамках означеного фрейму, який зсувається під час роботи системи. Цього можна досягти наступним чином:

1) відтворити поточну структуру фрейму на орієнтованому графі, а граф очікувань – визначається в термінах транзакцій, операції яких виконуються в рамках аналізованого фрейму;

2) детектувати в побудованому графі очікувань всі контури, а отже, виявити в аналізованому розкладі всі конфліктні ситуації;

3) визначити набір вершин, що призводять до появи контурів в графі очікувань, а отже, набір транзакцій, операції яких спричиняють появу конфліктів в аналізованому розкладі;

4) із визначеного набору вершин, що призводять до появи контурів в графі очікувань, вибрати вершину – кандидат для видалення з графа з метою його повної деконтуризації, тобто ця вершина відповідає так званій, транзакції – жертві, скасування якої призведе до усунення всіх конфліктних ситуацій і вимагатиме найменших витрат ресурсів; слід зауважити, що таких вершин може бути декілька в залежності від структури графа очікувань;

5) видалити вершину – кандидат з графа очікувань, тобто, скасувати транзакцію – жертву, що призведе до зміни змісту аналізованого розкладу;

6) базуючись на результатах виконання топологічного сортування графа очікувань реструктурувати зміст аналізованого розкладу так, щоб отримати еквівалентний йому послідовний розклад.

Циклічним виконання вище згаданих процесів досягається забезпечення здатності до впорядкування будь-якого набору транзакцій, а отже забезпечується можливість їх паралельного (переміжного) виконання.

Наведені вище міркування покладені в основу розробленого алгоритму автоматизованої розробки (рис. 1 - 3), який дає можливість не лише достовірно розпізнати виникнення тупиків у процесі паралельного використання транзакцій, але й автоматично ліквідувати такі конфлікти і, таким чином, зводити будь-які розклади до еквівалентної послідовної форми.

**Висновки.** Запропонована методика є узагальненою в тому розумінні, що способи формування еквівалентного послідовного розкладу операцій для розкладів із різним ступенем деталізації моделей транзакцій є аналогічними і відрізняються лише процедурою побудови графу очікувань для аналізованого фрейму.

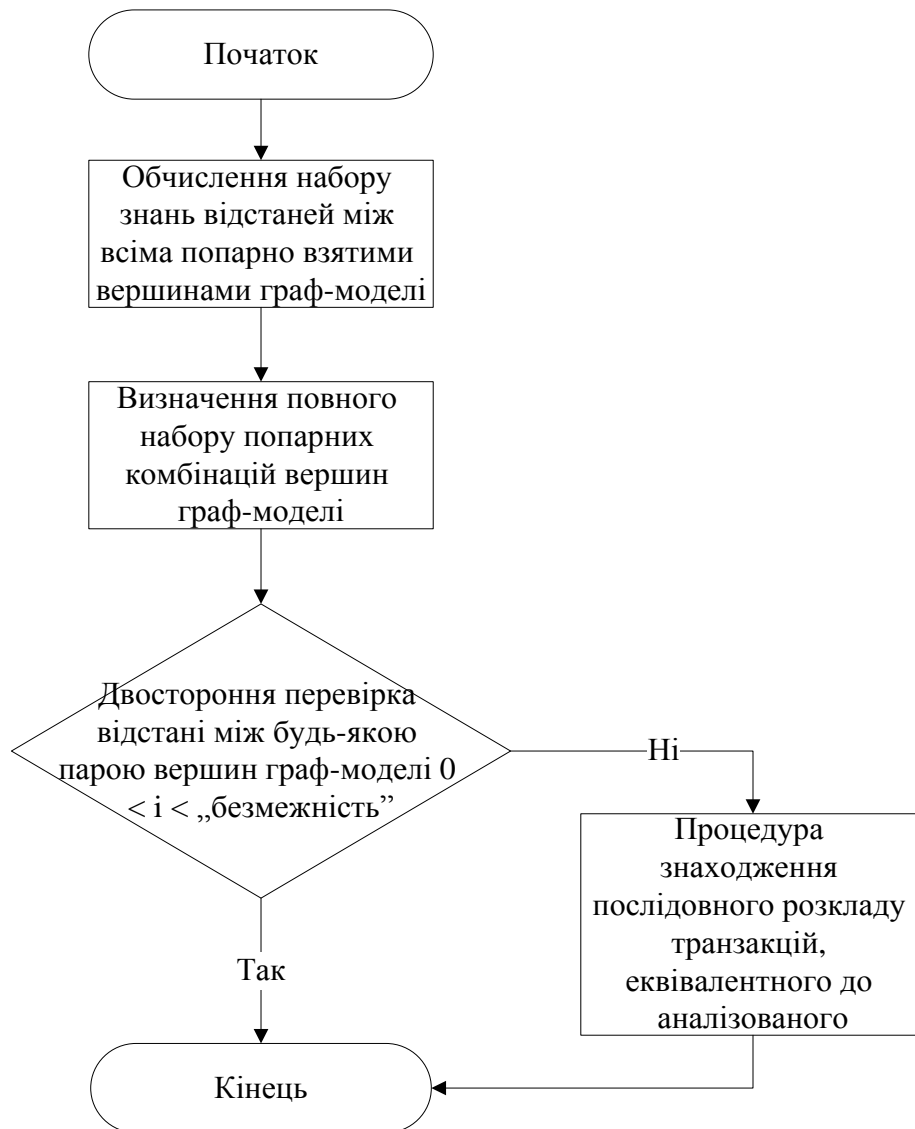


Рис. 1. Структура алгоритму процедури аналізу граф-моделі розкладу транзакцій на наявність контурів

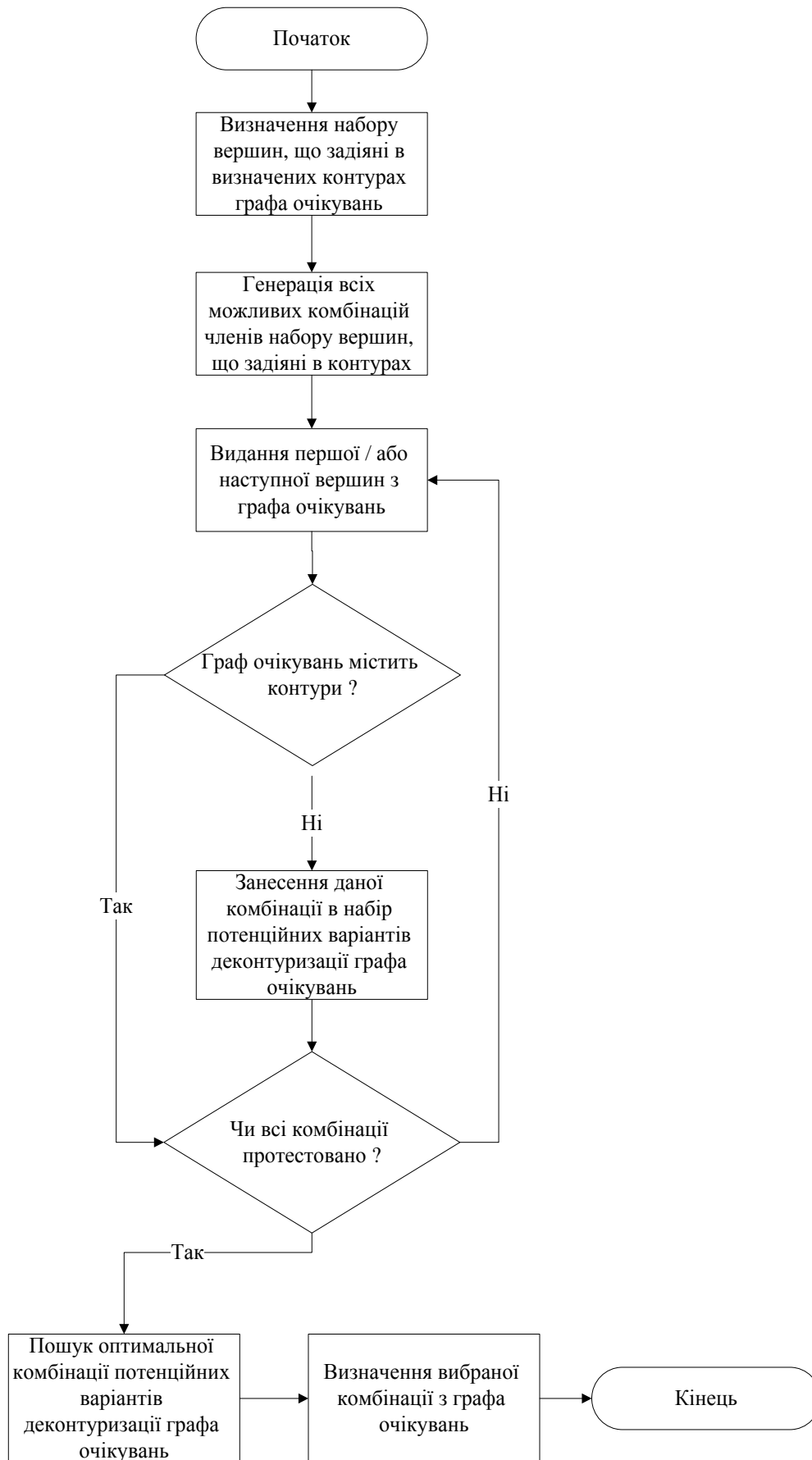


Рис.2. Структура алгоритму процедури аналізу граф-моделі розкладу транзакцій на наявність контурів

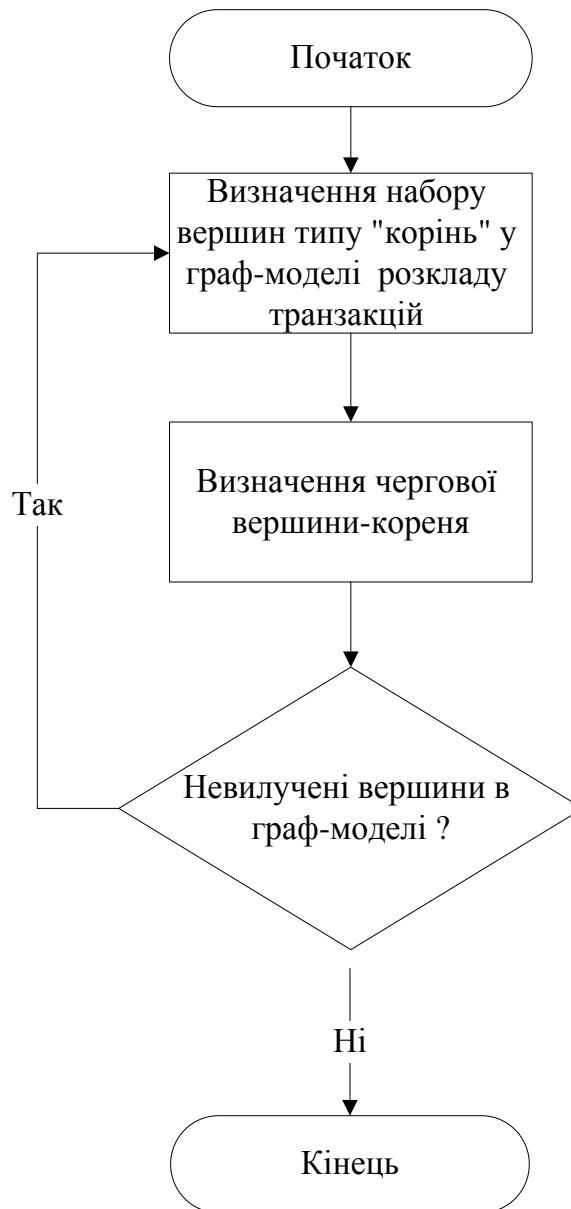


Рис.3. Структура алгоритму процедури зведення розкладу транзакцій до еквівалентного послідовного

#### Література

1. Дейт К.Дж. Введение в системы баз данных / Дейт К.Дж. – М.: СПб: Изд. дом "Вильямс", 1991. – 848 с.
2. Хансен Г. Базы данных разработка и управление / Хансен Г., Хансен Д. – М.: ЗАО "Изд. БИНОМ", 1999. – 699 с.
3. Капустян М.В. Модели передачи информации с учетом обнаружения, недопущения и устранения тупиковых ситуаций / Капустян М.В., Олешко Т.И., Хорошко В.А. // Вісник ДУІКТ, т. 4, № 3, 2006. – с. 156-162.

#### References

1. Dejt K.Dzh. Vvedenie v sistemy baz dannyh / Dejt K.Dzh. – М.: SPB: Izd. dom "Vil'jams", 1991. – 848 s.
2. Hansen G. Bazy dannyh razrabotka i upravlenie / Hansen G., Hansen D. – М.: ZAO "Izd. BINOM", 1999. – 699 s.
3. Kapustjan M.V. Modeli peredachi informacii s uchetom obnaruzhenija, nedopushhenija i ustraneniya

tupikovyh situacij / Kapustjan M.V., Oleshko T.I., Horoshko V.A. // Visnik DUİKT, t. 4, № 3, 2006. – s. 156-162.

#### Іванченко І.С.

#### МЕТОДИКА ПРИВЕДЕНИЯ РАСПИСАНИЯ ТРАНЗАКЦИИ К ЭКВИВАЛЕНТНОЙ ПОСЛЕДОВАТЕЛЬНОЙ ФОРМЕ

*Разработана обобщенная методика приведения произвольных расписаний транзакций к эквивалентной последовательной форме. Предлагается применение циклического выполнения определенных процессов для обеспечения способности к упорядочению любого набора транзакций, а значит обеспечения возможности их параллельного (перемежающейся) выполнения. А для обеспечения корректности параллельной обработки информации необходимо организовать динамическое формирование эквивалентного последовательного расписания операций в рамках указанного фрейма, который сдвигается во время работы системы. Выше*

указанные операции положены в основу разработанных алгоритмов и позволяют не только достоверно распознать возникновения тупиков в процессе параллельного использования транзакций, но и автоматически ликвидировать такие конфликты и, таким образом, сводить любые расклады к эквивалентной последовательной форме.

**Ключевые слова:** информационные ресурсы, каскадная отмена, транзакция, последовательная обработка, параллельная обработка, фреймы.

**Ivanchenko I.S.**

#### **THE METHOD OF BRINGING THE TRANSACTION TO A SCHEDULE OF EQUIVALENT SEQUENTIAL FORM**

*Developed general method of bringing arbitrary scheduling transactions to an equivalent serial form. It is proposed the use of looping specific processes to the ability to regulate any transaction, and thus enable their parallel (intermittent) performance. And to ensure the correctness of*

*the parallel processing of information is necessary to organize the formation of a dynamic equivalent serial schedule operations within the specified frame that moves during operation. The above operations form the basis for the proposed algorithms, and can not only reliably detect the impasse in the process of using both transactions, but also automatically eliminate such conflicts and thus reduce any decomposed to an equivalent sequential form.*

**Keywords:** information resources, cascaded revocation, transaction, serial processing, parallel processing, frames.

**Іванченко І.С.** – Національний авіаційний університет, м. Київ

**Рецензент:** Петров Олександр Степанович – докт. техн. наук, професор, завідувач кафедри безпеки інформаційних систем, Східноукраїнський національний університет імені Володимира Даля, м. Луганськ.