wykład

Przetwarzanie transakcyjne

Cz.2

Plan wykładu

Celem niniejszego wykładu jest omówienie problematyki związanej z transakcjami w bazie danych.

W szczególności zostaną omówione:

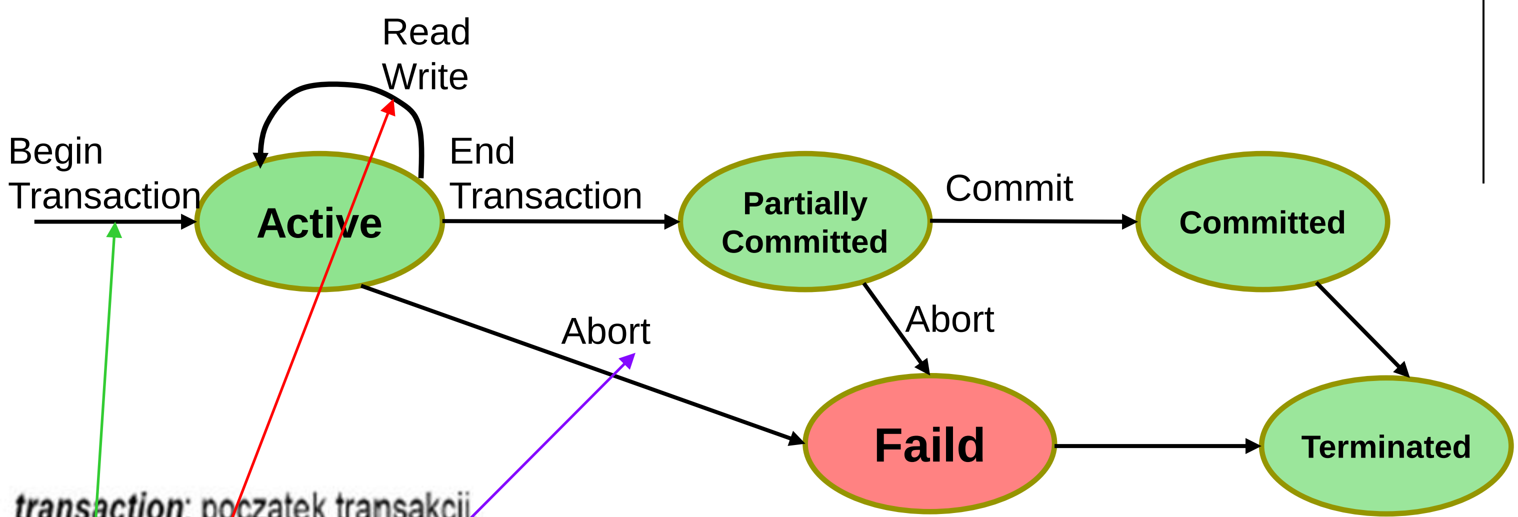
- transakcja i jej własności,
- formalny model transakcji,
- sekwencyjne i współbieżne realizacje zbioru transakcji,
- uszeregowalność transakcji.

transakcja (3)

Transakcja jest: **Cechy transakcji**

1. Atomowa: jeżeli pieniądze zostaną poprawnie przetransferowane z konta A do B
2. Spójna: jeżeli kwota odjęta z konta A jest równa kwocie dodanej do konta B
3. Izolowana: jeżeli inne transakcje wykonywane współbieżnie, czytające i modyfikujące konta A i B, nie mają wpływu na transakcję
4. Trwała: jeżeli po zakończeniu transakcji, baza danych trwale odzwierciedla nowe stany kont A i B

diagram stanów transakcji



Begin_transaction: początek transakcji.
Read, ***Write***: operacje odczytu i zapisu danych w bazie danych.
End_transaction: koniec transakcji:
    Commit: zatwierdzenie (akceptacja) wyników transakcji.
    Rollback: wycofanie wyników transakcji

1. (**Begin_Transaction**) uruchamia transakcję, która jest aktywna.
2. (***Read, Write***) dokonuje się w stanie aktywnym transakcji.
3. (***Abort***) przeprowadza transakcję ze stanu **Active** do stanu **Failed**, a następnie **Terminate**.
4. Kończenie transakcji z jej zatwierdzeniem przeprowadza ją ze stanu **Active** do **Partially committed** transakcja jest gotowa do zatwierdzenia.

model transakcji (2)

Każda transakcja może być reprezentowana przez graf skierowany:
$G = (V, A)$, gdzie:
V - jest zbiorem węzłów odpowiadających operacjom transakcji T_i;
A - jest zbiorem krawędzi reprezentujących porządek na zbiorze operacji.

Przykład:

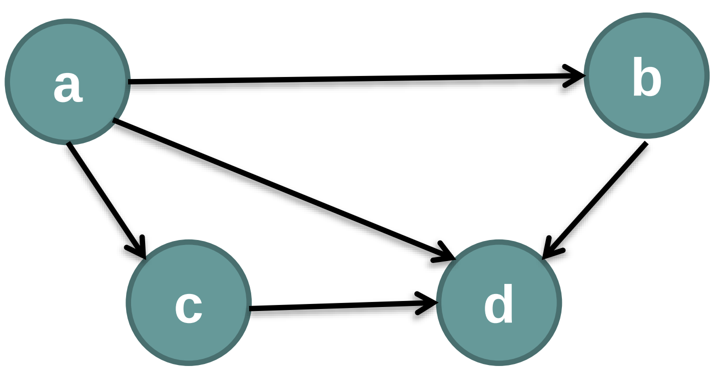
$r_1(x) \rightarrow w_1(x) \rightarrow r_2(y) \rightarrow w_2(y) \rightarrow c_1$

sekwencyjne wykonywanie transakcji

$r_1(x) \rightarrow w_1(x) \rightarrow w_2(y) \rightarrow c_1$

$r_2(y)$

współbieżne wykonywanie transakcji



Graf skierowany, digraf - DG

klasyfikacja transakcji

Ze względu na porządek operacji:

- transakcja sekwencyjna
- transakcja współbieżna

1

Ze względu na zależność operacji:

- transakcja zależna od danych
- transakcja niezależna od danych

2

Ze względu na typy operacji:

- zapytania lub transakcja odczytu (read only)
- transakcja aktualizująca - transakcja (read/write)

3

Trzy kryteria podziału transakcji: **porządek operacji**, **zależność operacji**, **typ operacji**.

realizacje transakcji (1)

- Częściowo uporządkowaną sekwencją operacji należących do zbioru współbieżnie wykonywanych transakcji nazywamy realizacją (historią). Realizacja modeluje, formalnie, współbieżne wykonanie zbioru transakcji

- Formalnie, realizacją S zbioru n transakcji T1, T2, ..., Tn nazywamy takie uporządkowanie operacji współbieżnie wykonywanych transakcji, w którym, dla każdej transakcji Ti w realizacji S, porządek wykonania operacji transakcji Ti jest taki sam jak porządek < *Ti*

W praktyce, w jednym systemie bazy danych działa równocześnie wiele transakcji.

realizacje transakcji (2)

Formalna notacja realizacji S zbioru transakcji.

$$S(\tau) = (\overline{T}_r(\tau), <r)$$

POS: *partially ordered set*

- gdzie:

1. $\boxed{\overline{T}_r(\tau)}$ zbiór operacji wszystkich transakcji należących do zbioru τ
2. $\boxed{<r}$ relacja częściowego porządku na zbiorze $\overline{T}_r(\tau)$,
3. Dla dowolnej pary operacji o_i, $o_j \in \overline{T}_r(\tau)$, takich, że żądają one dostępu do tej samej danej i co najmniej jedna z nich jest operacją zapisu, zachodzi $o_i <_r o_j$ lub $o_j <_r o_i$

realizacje transakcji (3)

Realizacja zawierająca tylko operacje zatwierdzonych transakcji nazywana jest ***zaakceptowaną projekcją***

(Dalsze rozważania dotyczyć będą tyko realizacji spełniających powyższy warunek)

T_0 T_1 T_2 T_1 T_2 T_1

Przykład:

Realizacja r:

r: $w_0(x), w_0(y), c_0, r_1(x), r_2(x), w_1(x), r_1(y), w_2(x), c_2, w_1(y),$
$c_1, r_f(x), r_f(y), c_f;$

T_1 T_f

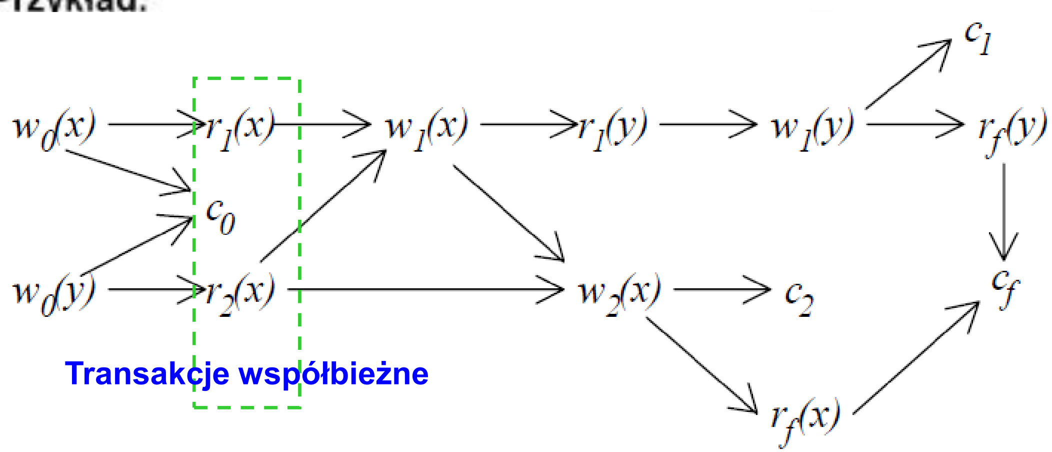
ZAAKCEPTOWANA PROJEKCJA

[Przykład zaakceptowanej projekcji]

realizacje transakcji (4)

- Dowolną realizację można przedstawić w postaci grafu skierowanego, nazywanego grafem realizacji, $GR(s(\tau)) = (V, A)$. Węzły grafu odpowiadają operacjom ze zbioru $\overline{T}_r(\tau)$, natomiast krawędzie grafu reprezentują relację częściowego porządku $< r$
- **Przykład:**

Przykład grafu realizacji dla transakcji T_0, T_1, T_2, T_f przedstawiono na slajdzie.



Przykład:

r: $w_0(x)$, $w_0(y)$, c_0, $r_1(x)$, $r_2(x)$, $w_1(x)$, $r_1(y)$, $w_2(x)$, c_2, $w_1(y)$, c_1, $r_f(x)$, $r_f(y)$, c_f;

[Przykład zaakceptowanej projekcji]

realizacje sekwencyjne i współbieżne

1. Mówimy, że dana realizacja jest sekwencyjna jeżeli, dla każdych dwóch transakcji, wszystkie operacje jednej z nich poprzedzają wszystkie operacje drugiej

2. W przeciwnym wypadku realizacja jest współbieżna

Stan i obraz bazy danych

Stan bazy danych - zbiór wartości wszystkich danych w bazie danych

Obraz bazy danych - widziany przez transakcję Ti zbiór wartości danych odczytywanych przez transakcję Ti

uszeregowalność realizacji (1)

Założenie 1

każda realizacja sekwencyjna jest poprawna

Założenie 2

każda realizacja współbieżna równoważna dowolnej realizacji sekwencyjnej tego samego zbioru transakcji jest również poprawna

Przykład:

Dane (początkowe wartości): a=50; b=50
Transakcja T1: sumuje konta a i b
Transakcja T2: przelewa 30 z konta a na konto b

Dana realizacja postaci:

s: ...r2(a, 50) w2(a, 20) r1(a,20) r1(b, 50) r2(b,50) w2(b, 80) c1 c2

Czy dana realizacja jest poprawna?

Czy realizacja jest poprawna?

uszeregowalność realizacji (1)

Przykład:

Dane (początkowe wartości): a=50; b=50
Transakcja T1: sumuje konta a i b
Transakcja T2: przelewa 30 z konta a na konto b

Dana realizacja postaci:

s: ...r2(a, 50) w2(a, 20) r1(a,20) r1(b, 50) r2(b,50) w2(b, 80) c1 c2

Czy dana realizacja jest poprawna?

Przedstawiona na slajdzie realizacja **nie jest poprawna** ponieważ obraz bazy danych widziany przez transakcję **T_1 to a+b=70, zamiast 100.**

uszeregowalność realizacji (1)

Realizacje sekwencyjne transakcji T1 i T2:

s1:...r1(a, 50) r1(b, 50) c1 r2(a, 50) w2(a, 20) r2(b, 50) w2(b, 80) c2 .....

końcowy stan bazy danych: a= 20; b= 80

obraz bazy danych widziany przez T2: a = 50; b = 50
obraz bazy danych widziany przez T1: a = 50; b = 50

W przykładzie ze slajdu transakcje **T_1** i **T_2** są realizowane sekwencyjnie.
W tym przypadku obraz bazy danych widziany przez obie transakcje **jest poprawny**.

konflikt (1)

pojęcia konfliktu dwóch operacji

Dwie **operacje** $o_i(x)$, $o_j(y)$ współbieżnej realizacji są **konfliktowe**, wtedy i tylko wtedy, gdy są spełnione następujące trzy warunki:

1. $x = y$ Operacje na różnych danych nigdy nie są konfliktowe

2. $i \neq j$ Operacje konfliktowe muszą należeć do różnych transakcji

3. Jedna z dwóch operacji o_i lub o_j musi być operacją zapisu

1

2

3

konflikt (2)

Rozszerzenie pojęcia konfliktu na zbiór (dwie) transakcji

- Dwie transakcje T_i, T_j są konfliktowe, jeżeli zawierają wzajemnie konfliktowe operacje
- Mówimy, że operacja $o_i(x)$ poprzedza operację $o_j(y)$ w realizacji $r(\tau)$, co zapisujemy jako $o_i(x) \rightarrow o_j(y)$, jeżeli operacje te są konfliktowe i operacja $o_i(x)$ poprzedza $o_j(y)$ w $r(\tau)$.
- Następujące pary operacji mogą znajdować się w konflikcie:
  - $r_i(x)$ $w_j(x)$
  - $w_i(x)$ $r_j(x)$
  - $w_i(x)$ $w_j(x)$

Pojęcie konfliktu można rozszerzyć na zbiór transakcji.
Dwie transakcje T_i oraz T_j są konfliktowe, jeżeli zawierają wzajemnie konfliktowe operacje.

Relacja poprzedzania jest antysymetryczna i przechodnia, a więc jest relacją częściowego porządku.

konfliktowa równoważność

Rozszerzenie relacji poprzedzania na zbiór (dwie) transakcji

• Mówimy, że transakcja T_i poprzedza transakcję T_j w realizacji $r(\tau)$, co zapisujemy jako $T_i \rightarrow T_j$, jeżeli zawierają odpowiednio operacje $o_i(x)$ i $o_j(x)$, między którymi zachodzi związek poprzedzania

• Mówimy, że dwie realizacje $r(\tau) = (T_r(\tau), <_r)$ i $r'(\tau) = (T_r(\tau), <_{r'})$ są konfliktowo równoważne, jeżeli dla każdej pary operacji $o_i(x)$ i $o_j(y)$ w realizacji $r(\tau)$, takich, że ... $\rightarrow o_j(y)$ w realizacji $r'(\tau)$

2. $i \neq j$ Operacje konfliktowe muszą należeć do różnych transakcji

Relacje poprzedzania można rozszerzyć na zbiór transakcji.

konfliktowa równoważność

Pojęcie równoważności dwóch realizacji

- Mówimy, że transakcja T_i poprzedza transakcję T_j w realizacji $r(\tau)$, co zapisujemy jako $T_i \rightarrow T_j$, jeżeli zawierają odpowiednio operacje $o_i(x)$ i $o_j(x)$, między którymi zachodzi związek poprzedzania

- Mówimy, że dwie realizacje $r(\tau) = (T_r(\tau), <r)$ i $r'(\tau) = (T_r(\tau), <r')$ są konfliktowo równoważne, jeżeli dla każdej pary operacji $o_i(x)$ i $o_j(y)$ w realizacji $r(\tau)$, takich, że $o_i(x) \rightarrow oj(y)$ zachodzi również $o_i(x) \rightarrow o_j(y)$ w realizacji $r'(T)$

Po wprowadzeniu relacji poprzedzania, można formalnie zdefiniować pojęcie równoważności dwóch realizacji.

Obecnie, sformułowane zostanie kryterium poprawności współbieżnej realizacji zbioru transakcji nazywane **kryterium konfliktowej uszeregowalności**.

kryterium konfliktowej uszeregowalności

Kryterium konfliktowej uszeregowalności

Realizacja $r(\tau)$ zbioru transakcji τ jest **konfliktowo uszeregowalna** wtedy i tylko wtedy, gdy jest ona konfliktowo równoważna dowolnej sekwencyjnej realizacji τ

definicja

Weryfikacja konfliktowej uszeregowalności

Grafem konfliktowej-uszeregowalności realizacji $r(\tau)$

nazywamy skierowany graf CSRG($r(\tau)$) = (V, A), taki, w którym zbiór wierzchołków V odpowiada transakcjom ze zbioru , natomiast zbiór krawędzi $A = \{(T_i, T_j) : T_i \rightarrow T_j\}$

graf konfliktowej uszeregowalności realizacji

twierdzenie konfliktowej uszeregowalności

Realizacja $r(\tau)$ zbioru transakcji jest
konfliktowo-uszeregowalna
wtedy i tylko wtedy, gdy jej graf konfliktowej uszeregowalności CSRG($r(\tau)$) jest acykliczny

Korzystając z grafu konfliktowej uszeregowalności można sformułować twierdzenie, pozwalające
w sposób algorytmiczny weryfikować, czy dana realizacja współbieżna jest poprawna, tj. **konfliktowo uszeregowalna**.

Realizacja r(τ) zbioru transakcji T jest konfliktowo uszeregowalna wtedy i tylko wtedy, gdy jej graf konfliktowej uszeregowalności CSRG(r(τ)) jest acykliczny.

1. **każda realizacja sekwencyjna zbioru transakcji zachowuje spójność bazy danych**
2. **z definicji grafu konfliktowej uszeregowalności wynika, że graf ten, dla dowolnej realizacji sekwencyjnej, musi być acykliczny**
3. **z definicji równoważności realizacji wynika, że graf konfliktowej uszeregowalności realizacji współbieżnej musi być również acykliczny**

realizacje odtwarzalne (1)

- Czy własność uszeregowalności gwarantuje wolność od anomalii ?

Przykład:

$H = r_1[x]\ w_1[x]\ r_1[y]\ r_2[x]\ w_1[y]\ r_2[y]\ c_2\ r_1[z]\ w_1[z]$ <crash> c_1

- Historia H jest uszeregowalna, ale nie jest wolna od anomalii (brudny odczyt). Po restarcie systemu transakcja T2 nie zostanie poprawnie odtworzona

Czy własność uszeregowalności gwarantuje poprawność dowolnej realizacji transakcji, w szczególności, czy gwarantuje wolność od anomalii współbieżnego wykonywania transakcji?

1 **awaria** **2** **wycofanie** **3** **zakleszczenie**

realizacje odtwarzalne (1)

- Czy własność uszeregowalności gwarantuje wolność od anomalii ?

Przykład:

T_2 T_2

$H = r_1[x]\ w_1[x]\ r_1[y]\ \boxed{r_2[x]}\ w_1[y]\ \boxed{r_2[y]}\ c_2\ r_1[z]\ w_1[z]$ <crash> c_1

- Historia H jest uszeregowalna, ale nie jest wolna od anomalii (brudny odczyt). Po restarcie systemu transakcja T2 nie zostanie poprawnie odtworzona

Jeżeli rozważamy realizacje, które zawierają operacje wycofywanych transakcji, to odpowiedź na postawione na wstępie pytanie **jest negatywna**.

definicje

- Potrzebna jest definicja nowych własności realizacji wykluczających anomalie będące wynikiem awarii systemu
- Mówimy, że transakcja T_i czyta daną x z transakcji T_j w realizacji H jeżeli

$w_j[x] < r_i[x]$

$a_j < r_i[x]$

jeżeli istnieje operacja $w_k[x]$ taka, że $w_j[x] < w_k[x] < r_i[x]$, wtedy $a_k < r_i[x]$

Jeżeli rozważamy szerszą klasę realizacji, które zawierają operacje zatwierdzonych jak i wycofywanych, na skutek awarii, transakcji, **potrzebne są definicje nowych własności realizacji, wykluczających anomalie będące wynikiem awarii systemu**.

realizacje odtwarzalne (2)

RC

- Realizacja *H* jest **odtwarzalna** (ang. *recoverable*) (*RC*) wówczas, jeżeli transakcja T_i *czyta* z transakcji T_j ($i \neq j$) w realizacji *H* i $c_i \in H$, to $c_j < c_i$

ACA

- Realizacja *H* **unika kaskadowych wycofań** (ang. *avoids cascading aborts*) (*ACA*) wówczas, jeżeli transakcja T_i *czyta* z transakcji T_j ($i \neq j$), to $c_j < r_i[x]$

ST

- Realizacja *H* jest **ścisła** (ang. *strict*) (*ST*) wówczas, jeżeli $w_j[x] < o_i[x]$ ($i \neq j$), zachodzi $a_j < o_i[x]$ lub $c_j < o_i[x]$, gdzie $o_i[x]$ jest jedną z operacji $r_i[x]$ lub $w_i[x]$

realizacje uszeregowalne

- Realizacja r zbioru transakcji jest poprawna (uszeregowalna) jeżeli jest ona obrazowo i stanowo równoważna jakiejkolwiek sekwencyjnej realizacji tego zbioru transakcji. Realizację taką nazywamy realizacją uszeregowalną (SR)

Przedstawiona na poprzednich slajdach definicja kryterium konfliktowej uszeregowalności stanowi zmodyfikowaną wersję podstawowego kryterium poprawności współbieżnej realizacji transakcji, które nosi nazwę kryterium uszeregowalności.

Zasadnicza różnica pomiędzy definicją **kryterium uszeregowalności a kryterium konfliktowej uszeregowalności** kryje się w **definicji równoważności realizacji transakcji**.

graf uszeregowalności (1)

- **Grafem uszeregowalności** realizacji $r(\tau)$ nazywamy skierowany graf $SG(r(\tau)) = (V, A)$, taki, w którym zbiór wierzchołków V odpowiada transakcjom ze zbioru τ, natomiast zbiór krawędzi jest zdefiniowany następująco:
- Jeżeli istnieje dana x, i operacje $T_i : r(x)$, $T_j : w(x) \in T_r(\tau)$, takie, że Ti : r(x) czyta wartość danej x zapisanej przez operację Tj : w(x), to:

1. $(T_j, T_i) \in A$
2. Jeżeli $T_j \neq T_0$, $T_i \neq T_f$ i istnieje operacja $T_k : w(x) \in T_r(\tau)$, $T_k \neq T_0$, to $(T_k, T_j) \in A$ lub $(T_i, T_k) \in A$
3. Jeżeli $T_j \neq T_0$, to $(T_0, T_j) \in A$

W celu weryfikacji uszeregowalności realizacji konstruujemy graf uszeregowalności realizacji.

graf uszeregowalności (2)

4. Jeżeli $T_j = T_0$, $T_i \neq T_f$ i istnieje operacja $T_k : w(x) \in T_r(\tau)$, $T_k \neq T_0$, to $(T_i, T_k) \in A$;
5. Jeżeli $T_i = T_f$, i istnieje operacja $Tk : w(x) \in T_r(\tau)$, to $(T_k, T_j) \in A$

Dana realizacja $r(\tau)$ jest uszeregowalna wtedy i tylko wtedy, gdy można skonstruować dla niej acykliczny skierowany graf uszeregowlaności SG($r(\tau)$)

definicja

2. Jeżeli $T_j \neq T_0$, $T_i \neq T_f$ i istnieje operacja $T_k : w(x) \in T_r(\tau)$, $T_k \neq T_0$, to $(T_k, T_j) \in A$ lub $(T_i, T_k) \in A$

Można pokazać, że dana realizacja r(T) jest uszeregowalna wtedy i tylko wtedy, gdy można skonstruować dla niej **acykliczny skierowany graf uszeregowalności** SG(r(T)).

Problem NP zupełny

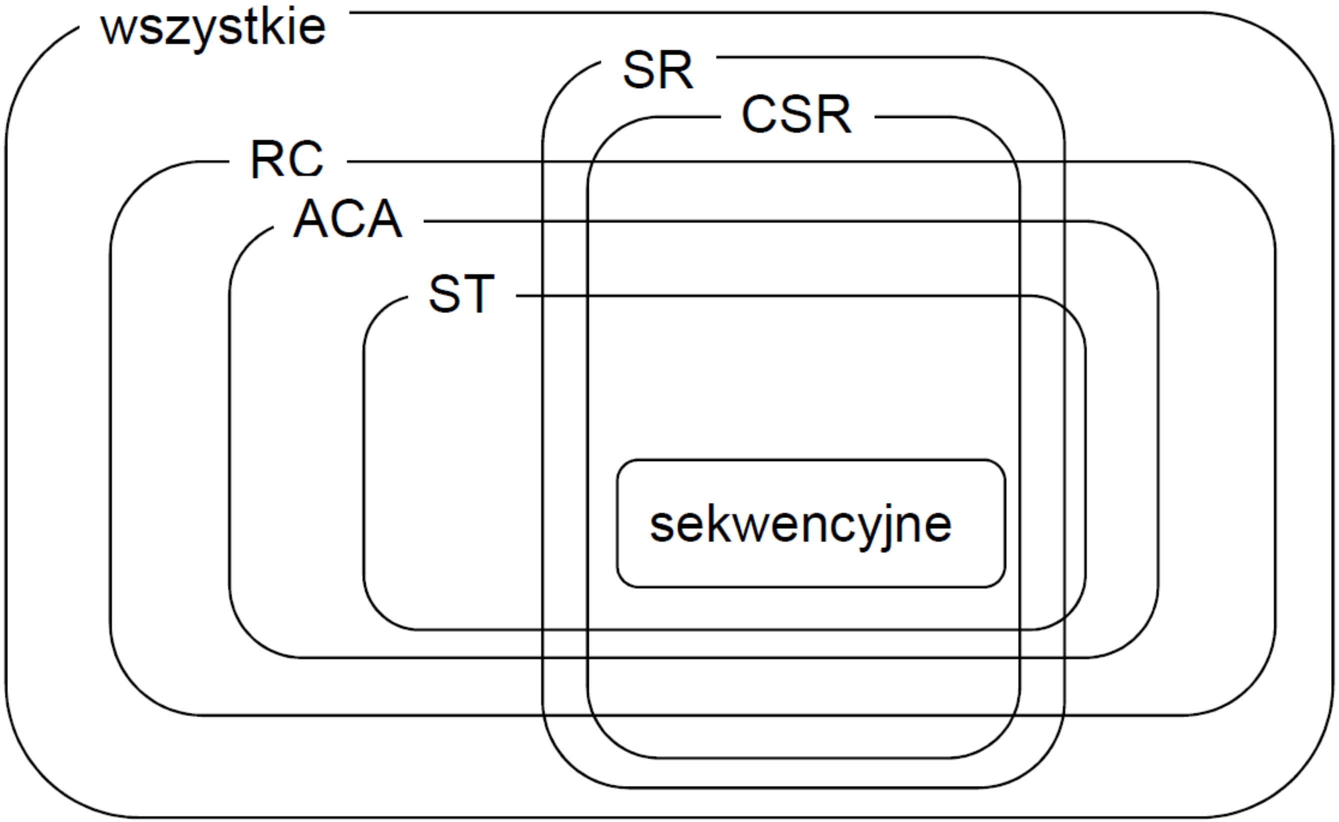
graf uszeregowalności (2)

4. Jeżeli $T_j = T_0$, $T_i \neq T_f$ i istnieje operacja $T_k : w(x) \in T_r(\tau)$, $T_k \neq T_0$, to $(T_i, T_k) \in A$;
5. Jeżeli $T_i = T_f$, i istnieje operacja $Tk : w(x) \in T_r(\tau)$, to $(T_k, T_j) \in A$

Dana realizacja $r(\tau)$ jest uszeregowalna wtedy i tylko wtedy, gdy można skonstruować dla niej acykliczny skierowany graf uszeregowlaności $SG(r(\tau))$

Z teorii grafów wynika, że weryfikacja czy dany poligraf zawiera cykl jest problemem NP zupełnym.

Uszeregowalność transakcji - klasyfikacja



KONIEC WYKŁADU

Cz. 2