# 10. tétel: Trace alapú hibadetektálás n-gramokkal

**Kidolgozta: Ribli János, 2010. december**

## Prekoncepciók:

1. Topológiák ismertek
    a. erőforrás függőségi (függőleges)
    b. adatfolyam / szolgáltatás kompozíció (vízszintes)
2. Ismert a komponensek hibaterjedési karakterisztikája (a hibák „ritka” események)

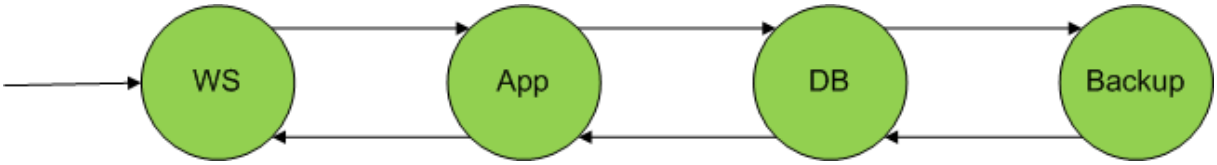
**Ez alapján modellezzük a helyes működést. A modellezés egyik lehetősége a tracek használatán alapul.**

## Tracing

(egyszerű példa: printStackTrace() metódus)

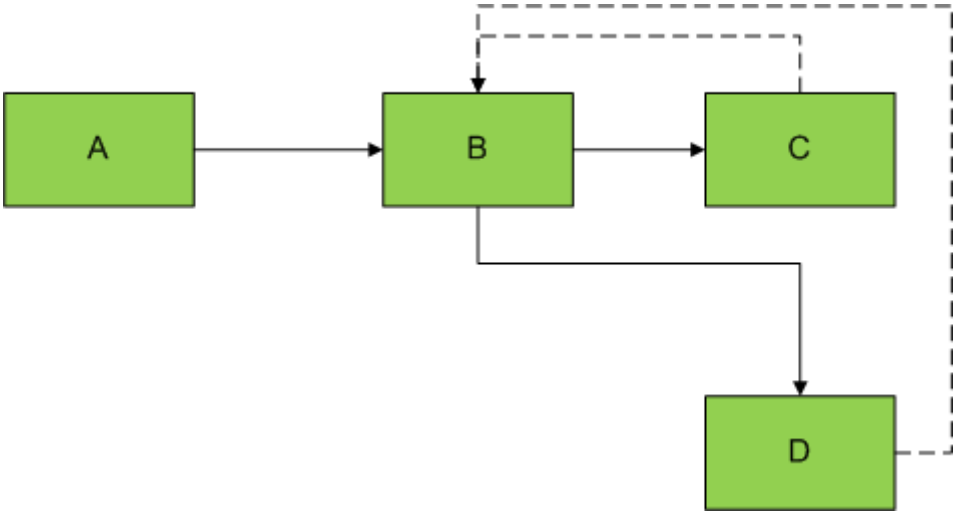
Motiváció: Egy művelet végrehajtása közben az alkalmazás egymás után több komponensen is futhat. Ennek követésekor keletkeznek trace-ek.

Példa (webes kiszolgálás):




*Application Response Management (ARM)*

Példa trace-ekre (metódus hívások szintjén):



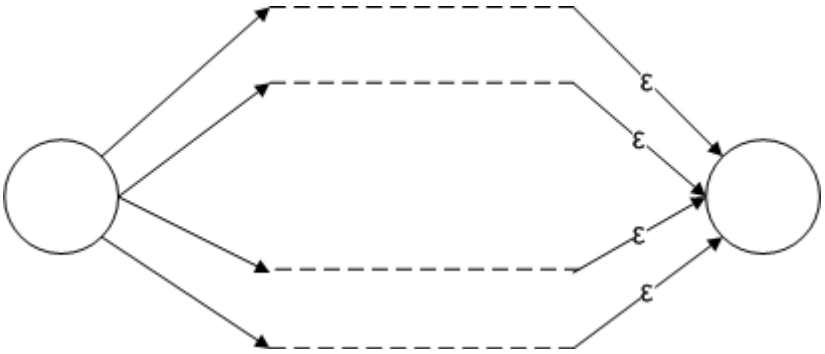

Ennek a nyoma: ABCbDb
A kis betű azt jelenti, hogy oda tér vissza.

Trace korlátok/jellemzők:

1. lokális korlátok (A-ból mindig meghívjuk B-t)
2. globális korlátok (egymás után hogyan következnek az összefüggő szakaszok)

Helyes működés modellezése tapasztalati úton lehetséges (a rögzített tracek alapján).

Modellezés véges automatákkal




Problémák:

- Nem generalizálható
- Nagy méret

Modellezés **n-Gram**mal

Egy mondatban a következő szó valószínűsége csak az előző n-1 szótól függ. Ilyen jellegű összefüggéseket kihasználva egyszerűsíteni lehet az automatát (összevont állapotok használatával).

**Def:** An n-gram is a subsequence of n items from a given sequence.

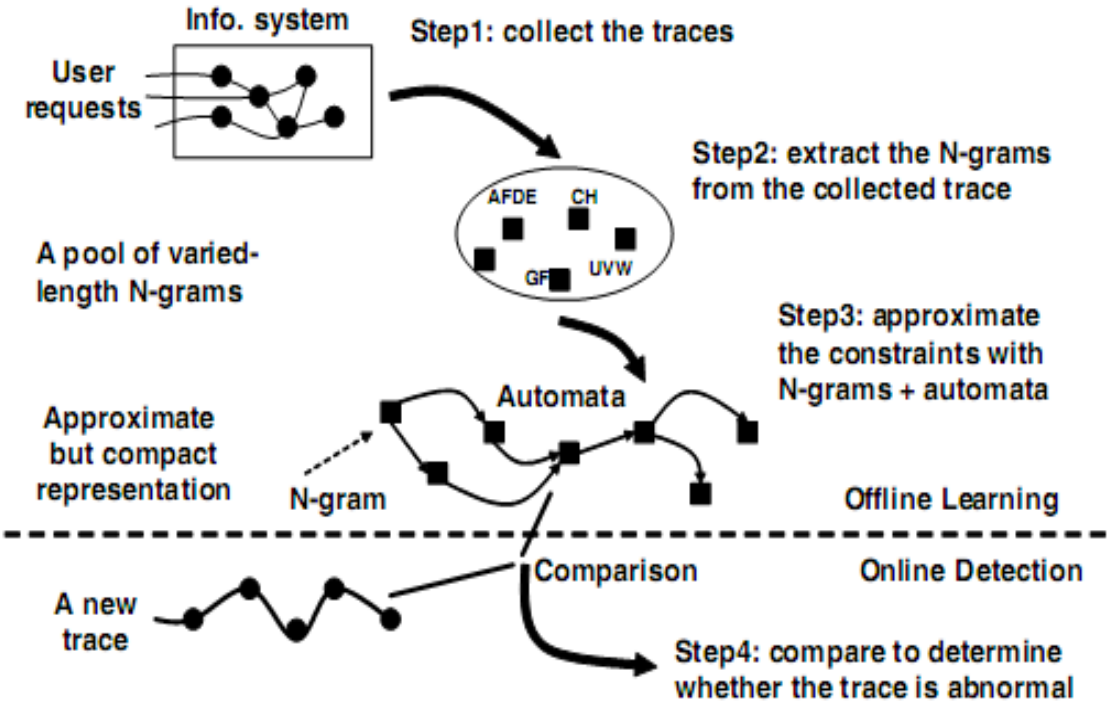
**n-Gram példa:**


## Trace alapú analízis

A trace alapú analízis folyamata a következő:

1. Tracek összegyűjtése a rendszerből
2. n-Gramok kinyerése a tracekből
3. Kényszerek, feltételek megadása (közelítés) n-Grammokkal és automatákkal (felismerő automata építése)
4. Illesztéssel meghatározni, hogy egy trace helyes-e (detektálás)

Ezt a folyamatot szemléletesen az alábbi ábra mutatja be.




Az egyes lépések megfelelő algoritmusokkal valósíthatók meg.

**n-Gramok kinyerése (az algoritmus pszeudo kódja a függelékben olvasható):**

**Alapvető működés:** Felírjuk a tanulóhalmazban előforduló különböző hosszúságú szekvenciákat. A szekvenciák mellé feljegyezzük az f(s) értéket is, ami megadja s szekvencia gyakoriságát a tanulóhalmazon. Ezt követően az alfa határérték alapján kiválasztjuk azokat a szekvenciákat, amelyek kellően sokszor fordulnak elő. (A szekvenciák felírását a hossz alapján iteratívan végezzük el a gyakoiriság szerinti szűrést követően.) Ezek a szekvenciák lesznek az n-Gramok.

**Példa:**

Bemenet:

Trace-ek: ABC, CD, ABD, AB, CBD

$\alpha = 0{,}7$

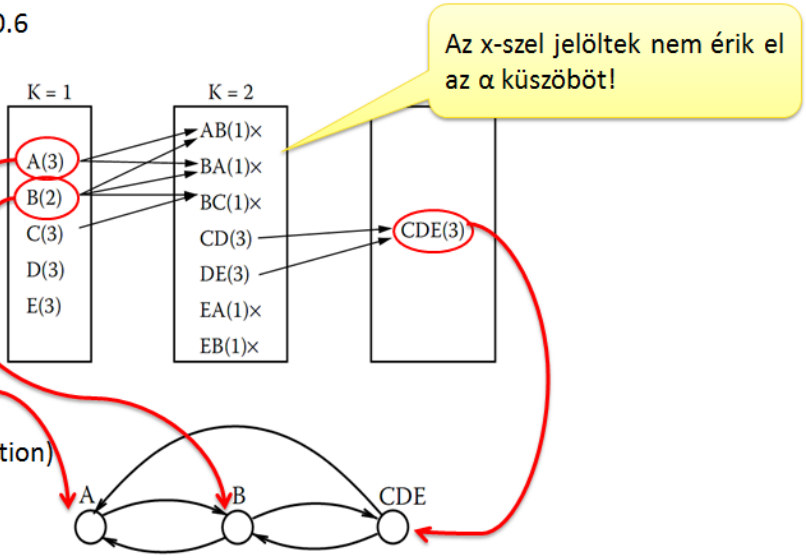
| k=1 | k=2 | k=3 |
|---|---|---|
| A (3)<br>B (3)<br>C (2)<br>D(2) | AB (3)<br>BD(2)<br>~~BC (1)~~<br>~~CB(1)~~<br>~~CD(1)~~ | ~~ABD(1)~~ |

**Felismerő automata építése (az algoritmus pszeudo kódja a függelékben olvasható):**

**Alapvető működés:** Az előző algoritmussal meghatároztuk az n-Gramokat valamint rendelkezésre állnak a tracek.
Ezt követően az egyes traceket előállítjuk az n-Gramokból. Először a leghosszabb n-Gramot vesszük és megnézzük, hogy szerepel-e a traceben. Ha igen, akkor ezt használjuk fel az adott trace megadására. Ha a leghosszabb n-Gramokból nem tudjuk előállítani a tracet, akkor az eggyel rövidebbekkel próbálkozunk tovább (és így tovább). Az azonos hosszú n-Gramok közül a leggyakoribbal próbálkozunk először. A felhasznált n-Gramokból automatát építünk. Az automatában akkor követ egy állapot egy másikat (akkor rajzolunk nyilat), ha valamely tracet úgy állítottuk elő, hogy az egyik n-Gram után a másik következik.

**Péda (n-Gramok kinyerése és automata építése):**

- Three traces: ABCDE, CDEA, CDEBA
  (capital letters are the components visited during the transaction)
- Threshold: α = 0.6
- Algorithm 1:
(n-gram extraction)




- Algorithm 2:
(automata construction)




**Detektálás (az algoritmus pszeudo kódja a függelékben olvasható):**

**Alapvető működés:**

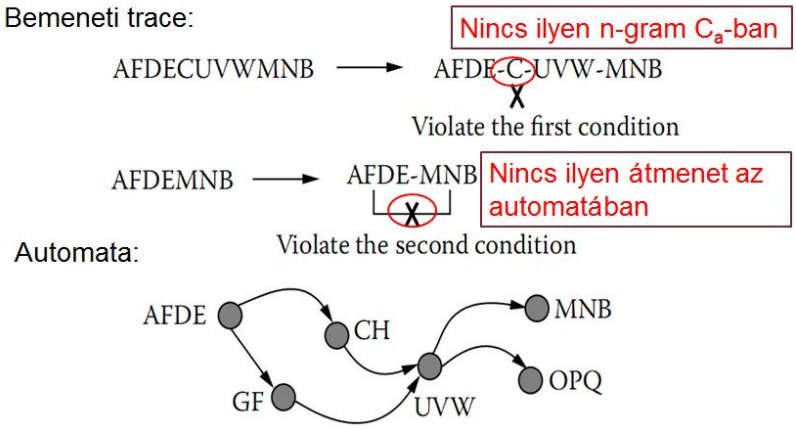
Definiáljuk a következő két szabályt:

1. szabály: a különféle n-gram halmazok közül mindig a hosszabb n-gramokat tartalmazó halmazt választjuk
2. szabály: az ugyanazon halmazban lévő n-gramok közül a gyakoribbat válasszuk

Egy trace elfogadható, ha

1. feltétel: a trace felvágható az 1. és 2. szabály szerint a meghatározott ($C_a$) halmazokban található n-gramokra
2. feltétel: ezen n-gramok átmenetei követik az automata átmeneteit (részhalmaz)

**Detektálási példa az előbbi eljárás alapján:**




---

# FÜGGELÉK:

**Algoritmus 1. (n-Gramok kinyerése)**

**Input:** the set of unique traces
**Output:** the sets of varied-length n-grams.

$C_1 = \{$the set of single components $c_1^i$ with $f(c_1^i) > 0\}$.
$k = 1$
**do**
  **for each** two elements $c_k^i$, $c_k^j$ from the set $C_k$,
    **if** the last $k-1$ component sequence of $c_k^i$ equals
      the first $k-1$ component sequence of $c_k^j$,
      **then** generate a new sequence
        $s = c_k^i$ plus the last component of $c_k^j$;
        count $f(s)$, the number of times that $s$ appears in the trace data;
        **if** $f(s) > \alpha \cdot min(f(c_k^i), f(c_k^j))$,
          **then** put $s$ into the set $C_{k+1}$.
  $k = k + 1$.
**while** $C_k$ is not empty
**return** all $C_j$, for $1 \leq j \leq k-1$

**$C^i_k$** *az* i-edik k-*gram*
**f(s)** mutatja, hogy a szekvencia hányszor jelenik meg a tanítóhalmazban

ha egy hosszabb $s$ szekvencia helyettesíteni tudja valamely szülőjének α százalékát, akkor egy új $s$ szekvenciát kell bevezetni mint (k+1)-gram

**Algoritmus 2. (automata építése)**

**Input:** the set of unique traces and the sets of n-grams
**Output:** the automaton $E$

set $E[m][n] = 0$ for any two n-grams $m, n$
**for each** trace $T$
  set $k = L$ and $l = T$'s length
  **do**
    **for each** $k$-gram $c_k^i$ selected from $C_k$ according to the sorted order (with the most frequent one first),

      search and replace all $c_k^i$ in $T$ with the assigned state number;

      **if** the length of the replaced part equals $l$,
        **then break** from the inner loop.
    $k = k - 1$.
  **while** the length of the replaced part $\neq l$ and $k \geq 1$.

  from left to right, set $E[m][n] = 1$ if an n-gram $n$ follows another n-gram $m$ contiguously in the trace $T$

remove the unused n-grams/states from $E$
**return** the matrix $E$

L = a leghosszabb n-gram hossza

**Algoritmus 3. (detektálás)**

**Input:** the automaton and the new trace $T$
**Output:** true (normal) or false (abnormal)

$C_a$ *azon n-gramok halmaza, amelyet tartalmaz az automata*
$N_a$ *a* $C_a$–ban lévő n-gramok száma

set $R$ = true, $l = T$'s length, and $m = 0$
**for each** n-gram $c^i_a \in C_a$ selected according to rules 1 and 2,

  search and replace all $c^i_a$ in $T$ with the state number;
  $m = m + 1$;
  **if** the length of the replaced part equal to $l$,
    **then** break the loop;

  **else if** $m == N_a$
      **then** $R$ = false.
**if** $R ==$ true, **then**
  from left to right, compare each state transition of $T$ against the automaton;
  **if** any transition not found in the automaton,
    **then** $R$ = false.
**return** $R$