

SZAKDOLGOZAT



MISKOLCI EGYETEM

Rutinszerű feladatok automatizálása grafikus felhasználói felületek esetében

Készítette:

Pázmándi Erik

Programtervező informatikus

Témavezető:

Dr. Kovács Béla

Konzulens:

Piller Imre

MISKOLC, 2022

MISKOLCI EGYETEM

Gépészmérnöki és Informatikai Kar

Alkalmazott Matematikai Intézeti Tanszék

Szám:

SZAKDOLGOZAT FELADAT

Pázmándi Erik (GXN833) programtervező informatikus jelölt részére.

A szakdolgozat tárgyköre: Folyamatelemzés, RPA

A szakdolgozat címe: Rutinszerű feladatok automatizálása grafikus felhasználói felületek esetében

A feladat részletezése:

A számítógépek kifejlesztésének és használatának egyik fő motivációja, hogy a segítségével az automatizált módon végrehajtható folyamatok emberi beavatkozás nélkül is végrehajthatóak legyenek. Ennek ellenére számos esetben tapasztalhatjuk, hogy az alkalmazások felhasználói felületén rutinszerűen, repetitíven hajtanak végre műveleteket.

A dolgozat azt vizsgálja, hogy ezek a folyamatok a korábban rögzített eseménysorok alapján hogyan ismerhetők fel. Bemutatja az RPA (Robotic Process Automation) eszközkészletét, többek között a folyamatelemzés elterjedt módszereit, alkalmazási lehetőségeit, a grafikus felhasználói felületekhez kapcsolódó speciális eseteket. Az elemzésekhez, automatizálást segítő eszköz elkészítéséhez Microsoft Windows platformon Delphi programozási nyelv kerül felhasználásra.

Témavezető: Dr. Kovács Béla (egyetemi docens)

Konzulens: Piller Imre (egyetemi tanársegéd)

A feladat kiadásának ideje: 2021. Szeptember 23.

.....
szakfelelős

EREDETISÉGI NYILATKOZAT

Alulírott **Pázmándi Erik**; Neptun-kód: **GXN833** a Miskolci Egyetem Gépészmérnöki és Informatikai Karának végzős Programtervező informatikus szakos hallgatója ezennel büntetőjogi és fegyelmi felelősségem tudatában nyilatkozom és aláírással igazolom, hogy *Rutinszerű feladatok automatizálása grafikus felhasználói felületek esetében* című szakdolgozatom saját, önálló munkám; az abban hivatkozott szakirodalom felhasználása a forráskezelés szabályai szerint történt.

Tudomásul veszem, hogy szakdolgozat esetén plágiumnak számít:

- szó szerinti idézet közlése idézőjel és hivatkozás megjelölése nélkül;
- tartalmi idézet hivatkozás megjelölése nélkül;
- más publikált gondolatainak saját gondolatként való feltüntetése.

Alulírott kijelentem, hogy a plágium fogalmát megismertem, és tudomásul veszem, hogy plágium esetén szakdolgozatom visszautasításra kerül.

Miskolc, év hó nap

.....

Hallgató

1.

szükséges (módosítás külön lapon)

A szakdolgozat feladat módosítása

nem szükséges

.....

dátum

.....

témavezető(k)

2. A feladat kidolgozását ellenőriztem:

témavezető (dátum, aláírás):

konzulens (dátum, aláírás):

.....

.....

.....

.....

.....

.....

3. A szakdolgozat beadható:

.....

dátum

.....

témavezető(k)

4. A szakdolgozat szövegoldalt

..... program protokollt (listát, felhasználói leírást)

..... elektronikus adathordozót (részletezve)

.....

..... egyéb mellékletet (részletezve)

.....

tartalmaz.

.....

dátum

.....

témavezető(k)

5.

bocsátható

A szakdolgozat bírálatra

nem bocsátható

A bíráló neve:

.....

dátum

.....

szakfelelős

6. A szakdolgozat osztályzata

a témavezető javaslata:

a bíráló javaslata:

a szakdolgozat végleges eredménye:

Miskolc,

.....

a Záróvizsga Bizottság Elnöke

Tartalomjegyzék

1. Bevezetés	1
2. Konceptió	2
2.1. Alpha-algoritmus	2
2.1.1. Rövid leírása	2
2.1.2. Eseménynapló	2
2.1.3. Minták	3
2.1.4. Példa	4
2.1.5. Korlátozások	6
2.2. Robotic Process Automation	6
2.3. Delphi	6
3. Tervezés	7
3.1. Bevezetés	7
3.2. Az Alpha-algoritmus	8
3.2.1. Alkalmazása	9
4. Megvalósítás	11
5. Tesztelés	12
6. Összefoglalás	13
Irodalomjegyzék	14

1. fejezet

Bevezetés

A dolgozat azt vizsgálja, hogy a számítógépek felhasználói felületén miféle sokszor elismételt folyamatok zajlanak le, ezek egy robot szempontjából hogyan is néznek ki, hogyan lehet ezeket utánozni, illetve automatizálni. Bemutat folyamatelemzési módszereket, ■
... BŐVÍTENI! (1-2 oldal)

2. fejezet

Koncepció

2.1. Alpha-algoritmus

Az Alpha algoritmus (vagy Alpha bányász) mint folyamatelemzési algoritmus célja, hogy eseménysorozatok halmazából egy ok-okozat rendszert építsen fel. Először van der Aalst, Weijters és Märušter hozta be a köztudatba. A működésében az eseménysorok halmazát nevezhetjük eseménynaplónak is. Ez az eseménynapló úgynevezett trace halmazoknak a halmaza, egy trace pedig adott tevékenységnek a sorozata.

Az Alpha bányász volt a legelső folyamatbányászati módszer amit valaha javasoltak és egy egész jó rálátást biztosít a folyamatbányászat céljára, valamint arra, hogy a folyamatokban lévő különböző tevékenységek hogyan is vannak végrehajtva. Emelett, az Alpha bányász szolgáltat számos újabb folyamatbányászati technika (pl.: Heurisztikus bányász, genetikus bányászat) alapjaként.

2.1. definíció. (*Munkafolyamati trace*) Egy string a T ábécé feladatai közül.

2.2. definíció. (*Munkafolyamati napló*) Munkafolyamati tracek halmaza.

2.1.1. Rövid leírása

Az algoritmus egy munkafolyamati naplót $W \subseteq T^*$ kap bemenetként, és eredményként egy munkafolyamati hálót épít fel.

Ezt az alapján csinálja meg, hogy megvizsgálja az általános kapcsolatokat az egyes feladatok között. Például egy adott feladat lehet, hogy minden esetben megelőz egy másik feladatot, ami egy hasznos információ.

2.1.2. Eseménynapló

Az eseménynapló az elsődleges szükséglet bármely folyamatbányászati algoritmus alkalmazásához. Az eseménynapló a következőket tartalmazza: egyedi azonosító az esethez, tevékenység megnevezése valamint egy időbélyeg. Egy eseménynaplót akár tevékenységek halmazának halmazaként is lehet ábrázolni.

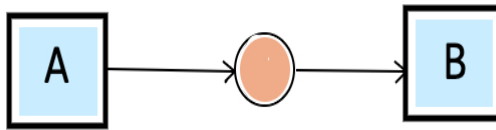
Az Alpha bányász szabályai szerint az egyes tevékenységek között az alábbi 4 féle kapcsolat egyike lehetséges:

1. **Közvetlen sorrend:** $x > y$ akkor és csakis akkor ha az x eseményt közvetlenül követi y .

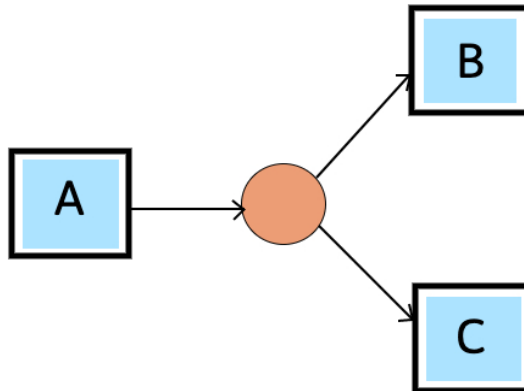
2. **Okozat:** $x \rightarrow y$ ha $x > y$ és nem $y > x$.
3. **Párhuzam:** $x \parallel y$ ha $x > y$ és $y > x$.
4. **Választás:** $x \# y$ ha nem $(x > y)$ és nem $(y > x)$.

2.1.3. Minták

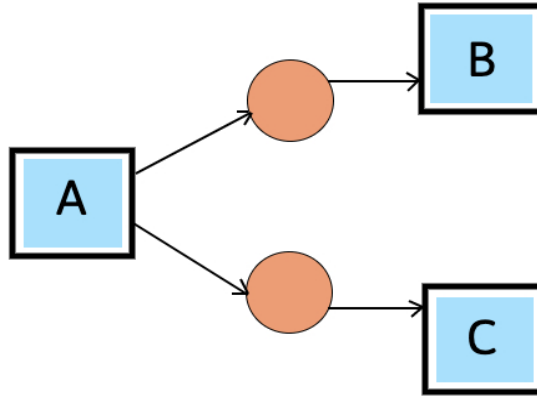
2.1. ábra. Szekvencia: $A \rightarrow B$



2.2. ábra. XOR-elágazás: $A \rightarrow B$, $A \rightarrow C$ és $B \# C$



2.3. ábra. **ÉS-elágazás:** $A \rightarrow B$, $A \rightarrow C$ és $B \parallel C$



2.1.4. Példa

Vegyük példának a következő eseménynaplót:

2.4. ábra. Példa eseménynapló

ID	Tevékenység	Időbélyeg
1	A	2022-10-05 13:50:40.000
1	B	2022-10-05 16:30:12.000
1	C	2022-10-05 16:57:31.000
1	D	2022-10-06 13:50:41.000
2	A	2022-10-06 15:30:27.000
2	C	2022-10-06 16:23:33.000
2	B	2022-10-07 08:33:02.000
2	D	2022-10-07 12:41:11.000
3	A	2022-10-07 13:02:57.000
3	E	2022-10-07 14:11:21.000
3	D	2022-10-07 14:59:22.000

Ebben az esetben az eseménynaplót az alábbi módon tudjuk jelölni:

$$L_1 = [< A, B, C, D >, < A, C, B, D >, < A, E, D >]$$

Az Alpha bányász úgy kezdi a munkát, hogy az eseménynaplót közvetlen-sorrend, okozat, párhuzam és választás relációkra alakítja és ezeket felhasználva létrehoz egy petri hálót ami leírja a folyamat modellét.

Első lépésként létrehoz egy lenyomati mátrixot:

2.5. ábra. Példa lenyomati mátrix

	A	B	C	D	E
A	#	\rightarrow	\rightarrow	#	\rightarrow
B	\leftarrow	#	\parallel	\rightarrow	#
C	\leftarrow	\parallel	#	\rightarrow	#
D	#	\leftarrow	\leftarrow	#	\leftarrow
E	\leftarrow	#	#	\rightarrow	#

Y_W az összes (A, B) pár halmaza a feladatok maximális halmazából úgy, hogy:

- Egyik $A \times A$ és $B \times B$ sem tagja \succ -nek, és
- $A \times B$ részhalmaza \rightarrow -nek.

P_W tartalmazza az egyes Y_W -hez tartozó helyeket $p_{(A,B)}$, plussz a beviteli i_W helyet és a kimeneti o_W helyet. A folyamati reláció F_W az alábbiak uniójából áll össze:

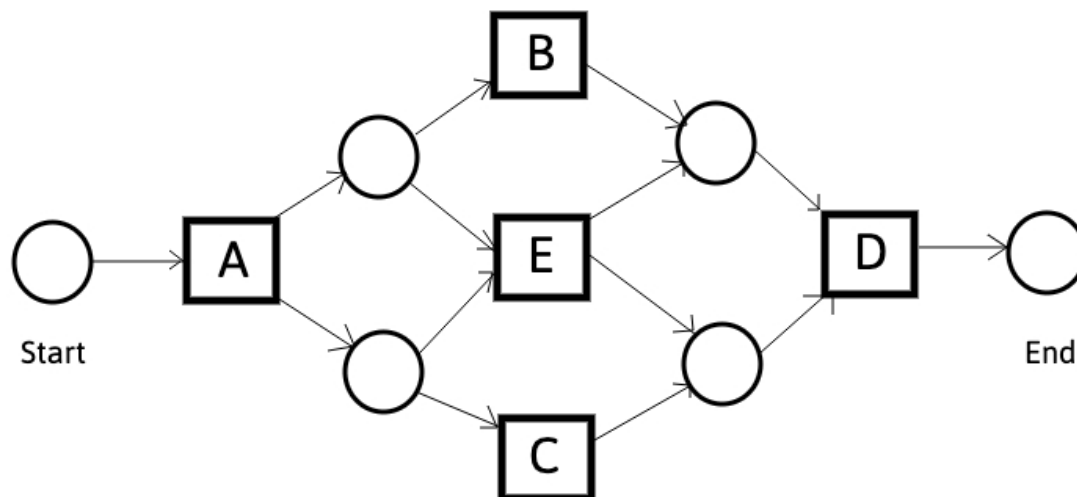
- $\{(a, p_{(C,B)}) | (A, B) \in Y_W \wedge a \in A\}$
- $\{(p_{(A,B)}, b) | (A, B) \in Y_W \wedge b \in B\}$
- $\{(i_W, t) | t \in T_1\}$
- $\{(t, i_0) | t \in T_0\}$

Az eredmény

- egy Petri háló struktúra $\alpha(W) = (P_W, T_W, F_W)$
- egy beviteli hellyel i_W és egy kimeneti hellyel o_W
- mivel minden T_W átmenet F_W -úton van i_W -ből o_W -be, így valóban egy munkafolyamati háló.

Ehhez a példához az alábbi petri háló jön létre az Alpha bányász használatával

2.6. ábra. Példa kimeneti petri háló



2.1.5. Korlátozások

- **Implicit helyek:** Az Alpha bányász nem tud különbséget tenni az implicit és a szükséges helyek között, így a felfedezett petri hálóban előfordulhatnak plusz szükségtelen helyek.
- **Ciklusok:** Az Alpha bányász nem képes 1-gyes és 2-tes hosszúságú ciklusok felismerésére a folyamatmodellben.
- A helyi függőségeket gyakran nem veszi észre az Alpha bányász.

Forrás: (Alpha-algoritmus, 2022).

2.2. Robotic Process Automation

Mi is ez, Alkalmazási területek, Meglévő rendszerek

2.3. Delphi

Delphi mint programozási nyelv, Miért Delphi a dolgozathoz

3. fejezet

Tervezés

3.1. Bevezetés

A dolgozat által vizsgált témához egy komplex multifunkciós szoftver került megtervezésre, mely a dolgozati téma elemzési részében kifejezetten nagy szerepet tölt be. Összetettsége révén rengeteg időt és odafigyelést igényelt már maga a tervezési fázis is. Számos ábra és tervezet került megalkotásra, melynek a túlnyomó része rendkívül jelentősnek bizonyult az implementáció során.

A legmagasabb szinten az alábbi ábra nyújtja a legtisztább áttekintését a különböző funkcióknak és a szoftver sokszínűségének.

3.1. ábra. High-level áttekintő ábra

Felhasználói interfész					
Különböző, felhasználói felülethez kapcsolódó szálak	Láncolt lista kezelő	Folyamat rögzítő	Folyamat generáló	Adatbányász	Eredmény ábrázoló
		Folyamat lejátszó	Folyamat időzítő		Petri háló kezelő
Kiegészítő funkciótár			Konfiguráció kezelő		

- **Felhasználói interfész:** A kezelőfelület, amivel a felhasználó eléri és kezelni tudja az egyes szoftverfunkciókat.
- **Felhasználói felülethez kapcsolódó szálak:** Fontos - a felhasználó számára nem látható - szálak, amelyek feldata bizonyos billentyűkombinációk figyelése anélkül, hogy a program reszponzivitását kártékonyan befolyásolnák.
- **Láncolt lista kezelő:** Az egyes folyamatok láncolt listaként vannak kezelve a szoftverbe, ez az alrendszer felel a megfelelő értelmezésükért.
- **Folyamat rögzítő:** Figyeli és rögzíti a perifériák általi beviteli értékeket.

- **Folyamat generáló:** Előre meghatározott forgatókönyvek alapján úgy generál folyamatokat mintha azt egy felhasználó végezte volna el.
- **Folyamat lejátszó:** A folyamatokat játsza vissza, egy felhasználót szimulál.
- **Folyamat időzítő:** A Windowsba integrált rendszert felhasználva ütemez / időzít folyamatokat.
- **Adatbányász:** Az Alpha-algoritmust implementálva folyamatelemzést hajt végre több folyamaton.
- **Eredmény ábrázoló:** Az Adatbányász által elvégzett folyamatelemzés eredményeit jeleníti meg.
- **Petri háló kezelő:** A petri-hálót mint struktúra, valamint a hozzá tartozó függvényeket a szoftver számára értelmezhető módon implementálja.
- **Kiegészítő funkciótár:** Számos hasznos funkció gyűjteménye, melyet a többi alrendsze használ.
- **Konfiguráció kezelő:** Futásidők között felhasználói preferenciák és beállítások tárolásáért és betöltéséért felel.

3.2. Az Alpha-algoritmus

Az Alpha-algoritmust mint folyamatelemzési módszert hasznosan lehet alkalmazni a dolgozati témában. Az algoritmus, jelentőségét tekintve elengedhetetlen részét képezi a dolgozatnak. Jelen esetben a folyamatokat azok jellegétől és céljától függetlenül lehet elemezni, akár cél nélküli beviteli sorozatra is alkalmazható az Alpha-algoritmus.

3.2.1. Alkalmazása

Ebben az alfejezetben bemutatásra kerül, hogy hogyan is kapcsolódik pontosan az Alpha-algoritmus a dolgozat témájához.

Az alább található példában három előre létrehozott folyamaton kerül alkalmazásra az Alpha-algoritmus. A folyamatok egyszerűek, hogy szemléletes legyen a példa, viszont ugyanezzel a módszerrel több száz vagy akár több ezer hosszú folyamaton is alkalmazható az algoritmus.

Maguk a folyamatok szolgálnak bemenetként, részeredményeként eseménynapló és lenyomati mátrix jön létre, kimentként pedig egy olyan petri háló kerül generálásra mely leírja a folyamat modelljét.

3.2. ábra. Beviteli folyamatok

Folyamat	ID	Típus	Érték	Érték típusa	Eltelt idő
1	1	Key	Left Alt	WM_SYSKEYDOWN	0
1	2	Key	F4	WM_SYSKEYDOWN	100
1	3	Key	F4	WM_SYSKEYUP	150
1	4	Key	Left Alt	WM_KEYUP	612
2	1	Key	Left Alt	WM_SYSKEYDOWN	0
2	2	Key	F4	WM_SYSKEYUP	80
2	3	Key	F4	WM_SYSKEYDOWN	51
2	4	Key	Left Alt	WM_KEYUP	152
3	1	Key	Left Alt	WM_SYSKEYDOWN	0
3	2	Key	F12	WM_SYSKEYDOWN	151
3	3	Key	Left Alt	WM_KEYUP	188

Az Alpha-algoritmus alkalmazásában, mint bármely folyamatbányászati algoritmusnál, első lépésként ezekből az eseményekből fel kell építeni az eseménynaplót amiből később dolgozik az algoritmus. Ez a lépés konkrétan arról szól, hogy a már meglévő folyamatok az Alpha-algoritmusnak szükséges formátumra kerülnek átalakításra.

Ez jelen esetben az alábbi három szabály alapján történik:

1. A "**Folyamat**" elnevezésű oszlop alapján triviális módon meghatározásra kerül az esethez tartozó egyedi azonosító,
2. A "**Típus**", "**Érték**" és "**Érték típusa**" oszlophármas értékeiből létrejön a tevékenység megnevezése, ami a továbbiakban „*Activity_x*”-ként lesz feltüntetve,
3. Az "**Eltelt idő**" oszlop alapján (az előző esemény óta eltelt időt mutatja) pedig létrejön egy relatív-időbélyeg az "**ID**" oszlop segítségével, hiszen az utóbbi alapján határozható meg az események szekvenciája.

Ezeknek megfelelően az alábbi eseménynaplót kapjuk:

3.3. ábra. Eseménynapló

Azonosító	Tevékenység	Relatív időbélyeg
1	Activity_0	0
1	Activity_1	100
1	Activity_2	250
1	Activity_3	762
2	Activity_0	0
2	Activity_2	80
2	Activity_1	131
2	Activity_3	283
3	Activity_0	0
3	Activity_4	151
3	Activity_3	339

Miután megvan az eseménynapló, az Alpha-algoritmus elkezdei a munkát, azaz az eseménynaplót közvetlen-sorrend, okozat, párhuzam és választás relációkra alakítja, ezzel létrehozva az alábbi lenyomati mátrixot:

3.4. ábra. Lenyomati mátrix

	A_0	A_1	A_2	A_3	A_4
A_0	#	→	→	#	→
A_1	←	#		→	#
A_2	←		#	→	#
A_3	#	←	←	#	←
A_4	←	#	#	→	#

4. fejezet

Megvalósítás

Program megvalósítása, Unitok bővebb leírása, 1-2 kódrészlet...

5. fejezet

Tesztelés

UI bemutatása, felhasználói kézikönyv

6. fejezet

Összefoglalás

ELKÉSZÍTENI (1-2 oldal)

Irodalomjegyzék

Alpha-algoritmus. (2022). *Alpha-algoritmus — Wikipedia, the free encyclopedia*. Retrieved from https://en.wikipedia.org/wiki/Alpha_algorithm ([Online, 2022-Október-07])

CD Használati útmutató

Ennek a címe lehet például *A mellékelt CD tartalma* vagy *Adathordozó használati útmutató* is.

Ez jellemzően csak egy fél-egy oldalas leírás. Arra szolgál, hogy ha valaki kézhez kapja a szakdolgozathoz tartozó CD-t, akkor tudja, hogy mi hol van rajta. Jellemzően elég csak felsorolni, hogy milyen jegyzékek vannak, és azokban mi található. Az elkészített programok telepítéséhez, futtatásához tartozó instrukciók kerülhetnek ide.

A CD lemezre mindenképpen rá kell tenni

- a dolgozatot egy `dolgozat.pdf` fájl formájában,
- a LaTeX forráskódját a dolgozatnak,
- az elkészített programot, fontosabb futási eredményeket (például ha kép a kimenet),
- egy útmutatót a CD használatához (ami lehet ez a fejezet külön PDF-be vagy Markdown fájlként kimentve).