Állapot minimalizálás

Sorrendi hálózatok állapot minimalizálása

A megtervezett sorrendi hálózat bonyolultsága függ attól, hogy hány állapotú az állapotgráf, amely alapján készült.

A specifikáció alapján nem biztos, hogy a minimális állapotszámú állapotgráfot sikerül előállítani.

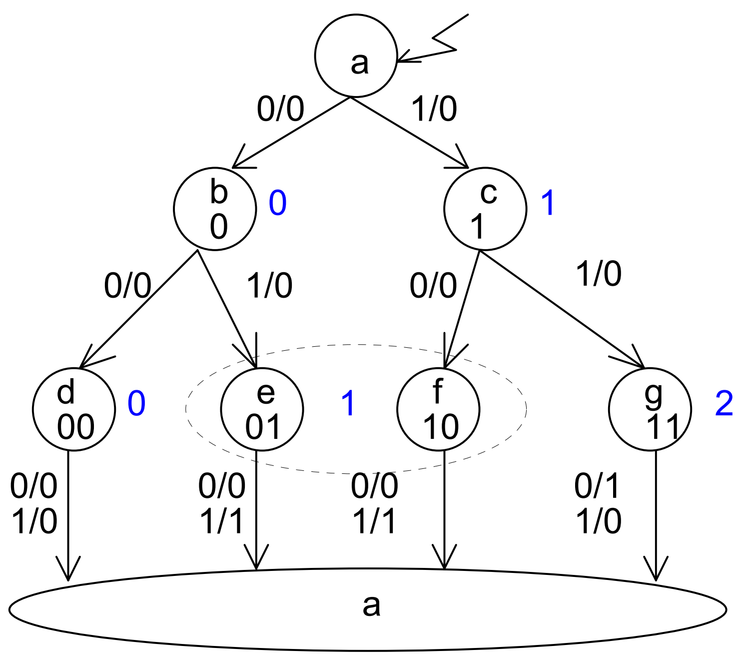
Példa:

Egy sorrendi hálózat egyetlen bemenetére sorosan érkeznek 3 bites számok, az órajellel szinkronban. A megtervezendő automatának fel kell ismerni, ha a 3 bites számokban pontosan 2db 1-es van, és jelezni a 3. bit beérkezésekor.

x: 010 011 111 101 110 100 001 011

y: 00**0** 00**1** 00**0** 00**1** 00**1** 00**0** 00**0** 00**1**

Szöveg alapján előzetes állapotgráf:



Itt e és f összevonható, mivel csak azt kell megjegyezni, hány 1-es jött, a sorrendjük mindegy.

Általában a több állapotból álló állapotgráf megvalósítása bonyolultabb hálózatot eredményez. Ha az állapotszám átlép egy 2 hatvány határt (pl. 8 helyett 9), akkor több flip-flop szükséges a megvalósításához (itt 3 helyett 4). A többlet flip-flopnak vezérlő függvénye is lesz, ami újabb kombinációs hálózatrész megvalósítást igényli.

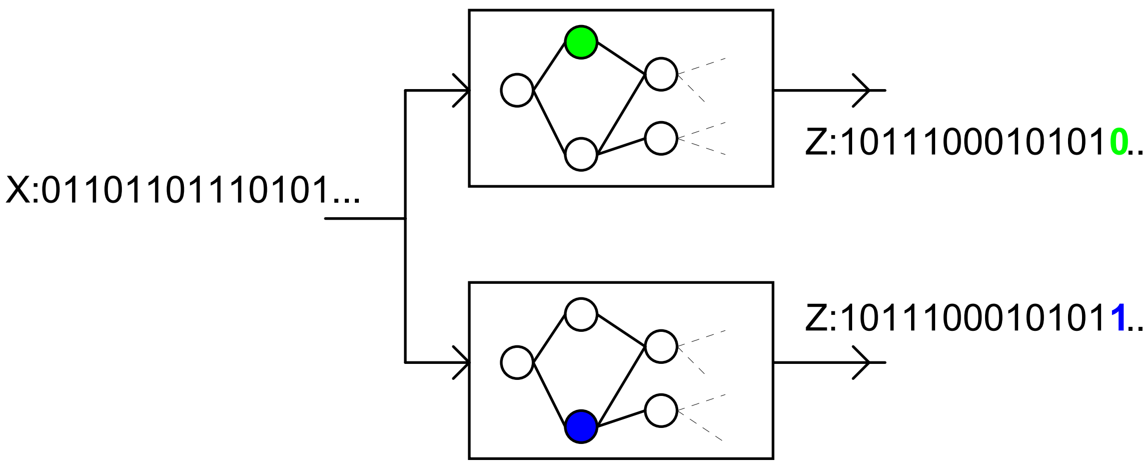
Ha az állapotszám nem lépi át az előbbi határt, de az eredetinél kevesebb állapottal is megvalósítható az automata, akkor is sok esetben kifizetődőbb a kevesebb állapotból kiinduló elkészítése. Ezért állapotminimalizálásra van szükség.

Az állaptszám minimalizálására szisztematikus módszerek léteznek. Ezeket fogjuk áttekinteni az alábbiakban.

Teljesen specifikált hálózatok minimalizálása

Egy automata ***teljesen specifikált (TSH)***, ha az összes következő állapota (Q_{t+1}) és kimenete (Z) specifikált.

Egy automata két állapota ***megkülönböztethető***, ha létezik olyan bemeneti sorozat, melyre adott válaszsorozat különböző a két állapotból indulva.



Egy automata két állapota *nem* ***megkülönböztethe-tő, ha nem létezik a fenti sorozat.***

Egy automata két állapota ***ekvivalens***, ha nem megkülönböztethető. (explicit definíció).

Ez a definíció nem alkalmas az ekvivalens állapotok megkeresésére, ezért egy erre használható definícióra is szükségünk van.

Egy automata két állapota ***ekvivalens***, ha (rekurzív definíció)
a. Mindkét állapotban, tetszőleges de azonos bemenetre a kimenetek azonosak.

b. A két állapotból tetszőleges de azonos bemenetre a következő állapotok is ekvivalensek.

Jelölése: a=b

Az ekvivalencia reláció

- reflexív a=a
- szimmetrikus a=b ↔ b=a
- tranzitív a=b és b=c ↔ a=c

Ekvivalencia osztály: páronként ekvivalens állapotok halmaza.

Maximális ekvivalencia osztály: Egy ekvivalencia osztály maximális, ha nem bővíthető újabb állapottal.

A TSH minimalizálás teljes algoritmusa

A továbbiakban egy mintapéldán mutatjuk be a minimalizálás algoritmusait.

x	0	1
a	b/0	f/1
b	c/0	f/1
c	b/0	f/1
d	d/0	a/1
e	d/0	c/1
f	b/1	f/0

Első lépés az ***összes antivalens és ekvivalens állapotpár***t keressük meg az un. lépcsős tábla segítségével.

1.

	a	b	c	d	e
b	**bc**				
c	✓	✓			
d	bd af	cd af	bd af		
e	bd cf	cd cf	bd cf	ac	
f	✗	✗	✗	✗	✗

2.

	a	b	c	d	e
b	✓				
c	✓	✓			
d	✗	✗	✗		
e	✗	✗	✗	✓	
f	✗	✗	✗	✗	✗

A továbbiakban a ***maximális ekvivalenciaosztályo-kat*** kell megkeresnünk. Erre több módszer is létezik.

1. ***Maximális ekvivalencia osztályok meghatározása az ekvivalensek alapján összevonással***

Az ekvivalens állapotpárok: a=b, a=c, b=c, d=e
Az ekvivalencia tranzitivitása miatt: a=b, a=c, b=c:
a=b=c ⇒ (abc), d=e ⇒ (de)
f nem ekvivalens egyetlen más állapottal sem: (f)
Tehát a max.ekv.osztályok:(abc), (de), (f)
Ez a módszer akkor célszerű, ha kevés az ekvivalens állapotpár.

2. Maximális ekvivalencia osztályok meghatározása az antivalensek alapján szétszedéssel

Az előbbi eredményre úgy is eljuthatunk, ha az antivalens állapot párok alapján az összes állapotból álló halmazt olyan csoportokra osztjuk, ahol az egymással antivalens állapotok nem szerepelhetnek egy csoportban. Az összes antivalens állapotpárral elvégezve a felosztást, s a végén elhagyva a nem maximálisakat, szintén megkapjuk a maximális ekvivalencia osztályokat. Ezt az algoritmust a nem teljesen specifikált hálózatok minimalizálásánál ismertetjük részletesen. A fő különbség az, hogy a lépcsős táblát itt az ekvivalencia, ott pedig a kompatibilitás figyelembe vételével kell kitölteni, továbbá TSH esetén az így kapott maximális kompatibilitási osztályok adják a végeredményt, míg NTSH esetén még további lépések következnek.
Ez a módszer akkor célszerű, ha kevés az antivalens állapotpár.

3. Maximális ekvivalencia osztályok megkeresése partíció finomítással

Itt először az állapotok halmazát a kimenet alapján partícionáljuk (diszjunkt csoportokra osztjuk). Azon állapotok kerülnek egy csoportba, melyek kimenete azonos bemenetre azonos kimenetet ad (az ekvivalencia első feltétele teljesül).

Ezután a csoportokat megszámozzuk, majd állapotonként feljegyezzük, hogy az egyes bemeneti kombinációk hatására mely sorszámú csoportban lesz a következő állapot.

Ez alapján tovább finomítjuk az egyes csoportokat. Azon állapotok maradnak egy csoportban, melyek azonos bemenetre azonos csoportba kerülnek, hiszen ahol ez nem teljesül, azon állapotok antivalens állapotokba kerülnek azonos bemenet hatására, vagyis nem teljesül az ekvivalencia második feltétele (hogy azonos bemenetre a következő állapotok is ekvivalensek).

x	0	1
a	**b/0**	**f/1**
b	**c/0**	**f/1**
c	**b/0**	**f/1**
d	**d/0**	**a/1**
e	**d/0**	**c/1**
f	**b/1**	**f/0**

Az előbbi példánál maradva:

1. felosztás:			2. felosztás		
	0.	1.	0.	1.	2.
	abcde	f	**abc**	**de**	f
x:0	**00000**	0	**000**	**11**	0
x:1	**11100**	1	**222**	**00**	2

Tovább nem lehet finomítani, ezért a maximális ekvivalencia osztályok: (abc)(de)(f)

Többnyire ez jár a legkevesebb munkával, mivel nem kell kitölteni a lépcsős táblát. Viszont ebben a formában csak teljesen specifikált hálózat esetén alkalmazható.

A maximális ekvivalencia osztályoknak feleltetjük meg a minimalizált sorrendi hálózat állapotait, majd kitöltjük a minimalizált állapottáblát.
Az új táblát az eredeti figyelembe vételével töltjük ki. Adott bemenetre egy ekvivalencia osztály állapotaira az eredeti táblán rákövetkező állapotok szintén egy ekvivalencia osztályban szerepelnek, az ***ezt képviselő új állapotot*** írjuk az új tábla megfelelő sorába. Pl. az A(abc) állapotaira rákövetkező állapotok x=0 esetén bcb, ezeket az A(abc) ekvivalencia osztály tartalmazza, ezért az A sorába az x=0 oszlopba A kerül.
Amely állapotokat egy ekvivalencia osztály képvisel, azon állapotokhoz tartozó kimenetek egyformák az eredeti táblán (ha nem hibáztunk), ezt kell beírni a minimalizált táblába. Pl. az A(abc) ekvivalencia osztály állapotaiban az x=0 bemenetre 0 lesz a kimenet, ezért ezt írjuk az A sorában az x=0 oszlopba.

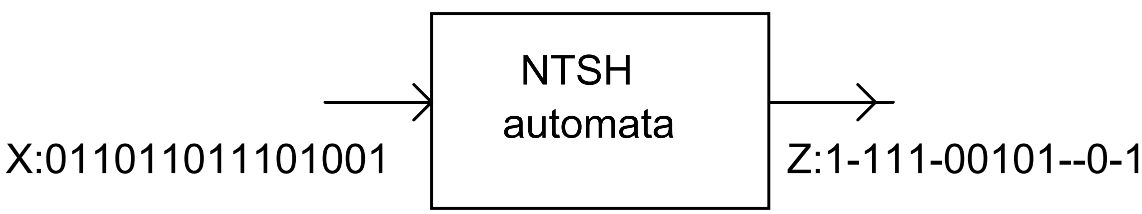
x	0	1
a	**b/0**	**f/1**
b	**c/0**	**f/1**
c	**b/0**	**f/1**
d	**d/0**	**a/1**
e	**d/0**	**c/1**
f	**b/1**	**f/0**

X	0	1
A(abc)	**A/0**	**D/1**
B(de)	**B/0**	**A/1**
D(f)	**A/1**	**D/0**

Nem teljesen specifikált hálózatok (NTSH) állapot minimalizálása

NTSH:

Van az állapottáblában olyan kimenet vagy következő állapot, amely nem specifikált.
(Itt a megtervezett áramkör tetszöleges kimenetet adhat ill. tetszőleges állapotba léphet. Normál működés során az NTSH hálózat nem kaphat olyan bemeneti sorozatot aminek hatására nem definiált állapotba lépne, hiszen innetől kezdve nem definiált a működése.)

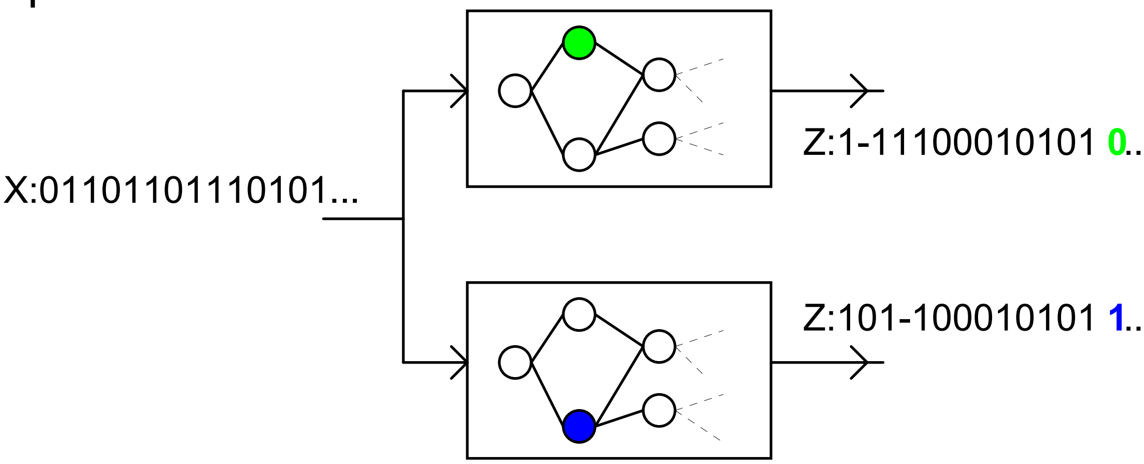


A Z-n közömbössel jelölt kimenetek esetén a *megtervezett* automata tetszőleges értéket adhat.

A feladat egy lehetőleg minimális állapotszámú NTSH automata tervezése.

Ún. NP-teljes feladat, egzakt megvalósítása nagyon nagy számítási igényű, ezért erre a gyakorlatban heurisztikus módszereket alkalmaznak. Több azonos állapotszámú megoldás is létezhet.

Az NTSH két állapota megkülönböztethető:
Ha létezik olyan bemeneti sorozat, melynél a létrejövő állapotok specifikáltak, s a két állapotból indulva különböznek a kimenetek valamely mindkettőben specifikált esetben.



Az NTSH két állapota nem megkülönböztethető:
Ha nem létezik a fenti sorozat.

Az NTSH nem megkülönböztethető állapotait ***kompatibilis állapotok***nak nevezzük.

Jelölése: ~ Pl: a~b

Az TSH két állapota kompatibilis (rekurzív definíció):
a. Ha mindkét állapotban, tetszőleges de azonos bemenetre a kimenetek azonosak, ha mindkettő specifikált.

b. A két állapotból tetszőleges de azonos bemenetre a következő állapotok is kompatibilisek, ha mindkettő specifikált.

Kompatibilitási osztály:
Az egymással kompatibilis állapotok halmaza.

Pl. Ha a~b, a~c, b~c akkor (abc) egy kompatibilitási osztályt alkot.

A kompatibilitási reláció
- reflexív a~a
- szimmetrikus a~b ↔b~a
- de ***nem tranzitív*** a~b és b~c-ből *nem* következik a~c

Maximális kompatibilitási osztály:
Nem bővíthető újabb állapottal. (Nincs több olyan állapot, amely az osztály összes tagjával kompatibilis lenne.)

Az NTSH minimalizálás teljes algoritmusa

A teljes algoritmust egy mintapéldán mutatjuk be.

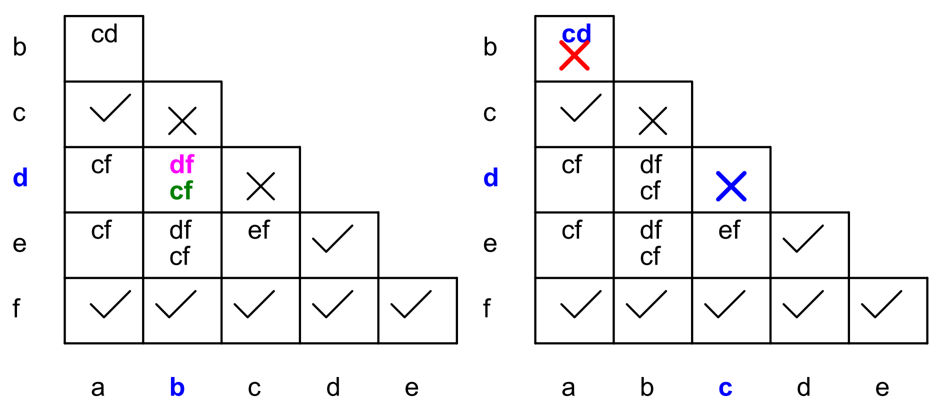
x1x2	00	01	11	10
a	c/0	-/-	a/0	-/-
b	**d**/0	**c**/0	-/-	-/-
c	-/-	e/1	a/0	-/-
d	**f**/-	**f**/0	-/-	-/-
e	f/0	f/-	-/-	-/-
f	-/-	-/-	a/0	-/-

1. lépés

Az összes kompatibilis és inkompatibilis állapotpár megkeresése, lépcsős tábla segítségével.

Első lépésben a kompatibilitás definíciója alapján biztosan kompatibilis (pipa), nem kompatibilis (kereszt) és feltételesen kompatibilis (feltételek) bejegyzések kerülhetnek a rubrikákba.

1. 2.



Második lépésben az inkompatibilis bejegyzésekből indulva inkompatibilissá tesszük azokat a bejegyzéseket, amelyek feltételéről már tudjuk, hogy nem teljesül.

Kompatibilisek: a~c, a~d, a~e, a~f, b~d, b~e, b~f, c~e, c~f, d~e, d~f, e~f.

Inkompatibilisek: ab, bc, cd.

Kompatibilisek: a~c, a~d, a~e, a~f, b~d, b~e, b~f, c~e, c~f, d~e, d~f, e~f.
Inkompatibilisek: ab, bc, cd.

2. Lépés

A maximális kompatibilitási osztályok megkeresése.

a. Kompatibilis párok alapján, bővítéssel:

1.

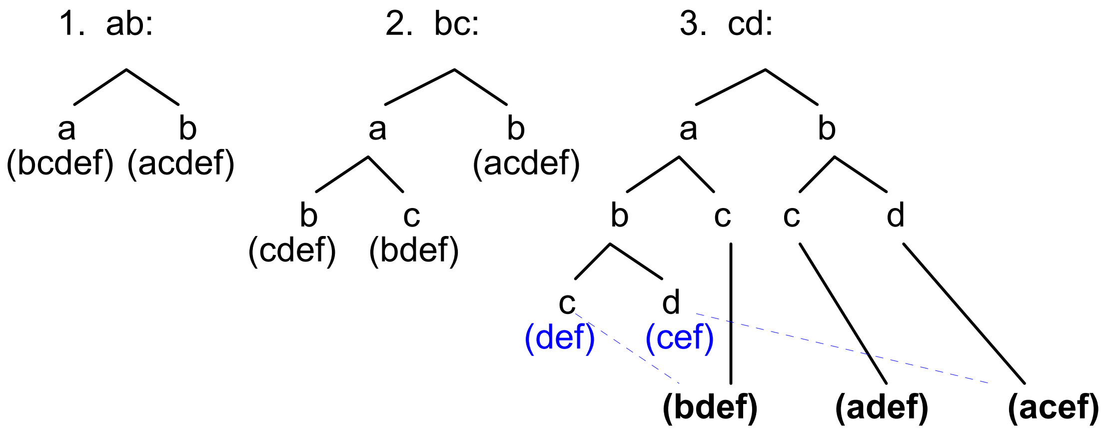
b	X				
c		X			
d			X		
e					
f					
	a	b	c	d	e

e: (ef)
d: de, df, (ef)->(def)
c: cf, ce, (ef)->(cef)
b: bf, be, (ef)->(bef)
bd, be, bf, (def)-> **(bdef)**
a: af, ae, (ef)-> (aef)
ad, ae, af, (def)-> **(adef)**
ac, ae, af, (cef)-> **(acef)**

A nem maximálisakat elhagyva, a ***maximális kompatibilitási osztályok***:
(acef), (adef), (bdef)

b. Inkompatibilis párok alapján szétszedéssel:

Inkompatibilis párok: ab, bc, cd
Az alábbi ábra lépésenként mutatja a szétszedést.



A kékkel jelöltek nem maximális kompatibilisek, elnyeli amely tartalmazza őket.

A bővítés célszerű, ha kevés a kompatibilis pár.

A szétszedés célszerű, ha kevés az inkompatibilis pár.

A maximális kompatibilisek száma jóval nagyobb is lehet, mint az eredeti hálózat állapotszáma.

3. Lépés

Ki kell választani a (maximális) kompatibilisek közül minimális számút, mely teljesít két feltételt:

a. ***Lefedik*** az eredeti hálózat összes állapotát

b. A kiválasztott kompatibilisek ***zárt halmazt alkotnak***. Vagyis a kiválsztottak közül egy tetszőleges kompatibilis tetszőleges 2 állapotából azonos bemenetekre a következő állapotok is együtt szerepelnek valamely kiválasztott komatibilisben.

Tehát ***minimális zárt fedést*** keresünk.

Egy-egy kompatibilishez az eredeti állapottáblából lehet meghatározni a zártság feltételeit. (Melyek lesznek a következő állapotok egy-egy bemeneti kombináció esetén.)

x1x2	00	01	11	10
a	**c**/0	-/-	**a**/0	-/-
b	d/0	c/0	-/-	-/-
c	-/-	**e**/1	**a**/0	-/-
d	f/-	f/0	-/-	-/-
e	**f**/0	**f**/-	-/-	-/-
f	-/-	-/-	**a**/0	-/-

	00	01	11
(acef):	**(cf),**	**(ef),**	**(aaa)**
(adef):	cf,	(f),	(aa)
(bdef):	(df), cf,		-

Ha ugyanaz az állapot a következő, azt figyelmen kívül hagyjuk, mivel minden állapot kompatibilis önmagával és biztosan szerepel valamely osztályban. Ha a feltételt a kompatibilis önmaga teljesíti, azt is figyelmen kívül hagyjuk (itt zárójelbe tettük).

Fedjük le az eredeti állapotokat, minimális számú kompatibilissel, **fedési táblázat** segítségével:

	a	b	c	d	e	f
(acef)	x		**x**		x	x
(adef)	x			x	x	x
(bdef)		**x**		x	x	x

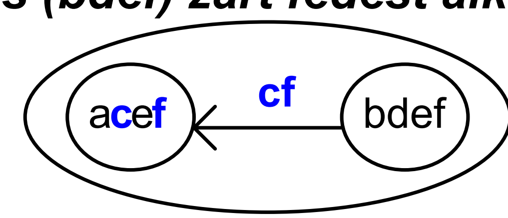
Az (acef) és (bdef)-re mindenképpen szükség van, mert c-t és b-t csak ők fedik le.
Ellenőrizzük, hogy zárt fedést alkotnak-e.

(acef): (cf), (ef)

Önmaga teljesíti a zártságot, mert cf és ef együtt szerepel (acef)-ben

(bdef): (df), cf

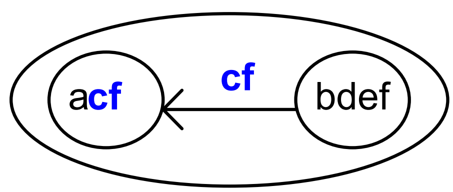
A df szerepel (bdef)-ben, cf pedig (acef)-ben.
Tehát (acef) és (bdef) zárt fedést alkotnak!



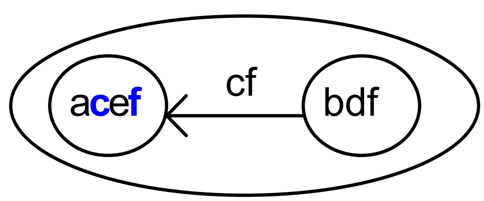
Vizsgáljuk meg, elhagyható-e állapot valamelyikből, a zártsági és fedési feltételek megsértése nélkül?
A fedési feltétel miatt csak olyan állapot hagyható el, amely legalább 2 kompatibilisben szerepel.

x1x2	00	01	11	10
a	c/0	-/-	a/0	-/-
b	d/0	c/0	-/-	-/-
c	-/-	e/1	a/0	-/-
d	f/-	f/0	-/-	-/-
e	f/0	f/-	-/-	-/-
f	-/-	-/-	a/0	-/-

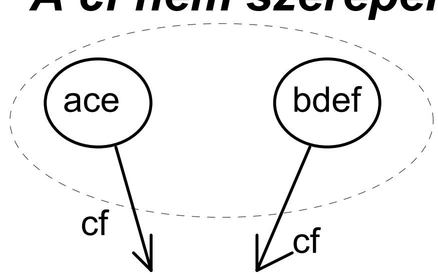
Pl: e-t elhagyva (acef)-ből ***zárt fedést*** kapunk.
(acf): -
(bdef): (df), cf



e-t elhagyva (bdef)-ből szintén ***zárt fedést*** kapunk.
(acef): (cf)
(bdf): (df), cf



De pl. f-et elhagyva (acef)-ből ***nem zárt a fedés***!
(ace): **cf**
(bdef): (df), **cf** ***A cf nem szerepel együtt sehol!***



A zárt fedés keresésének algoritmusa:

1. A maximális kompatibilisek alapján minimális fedést keresünk a fedési táblázat segítségével.

2. Ellnőrizzük, hogy zárt-e a fedés. Ha igen, vagy ezekhez a kompatibilisekhez rendeljük a minimalizált állapottábla állapotait, vagy megpróbálunk állapotokat elhagyni a fedés és zártság feltételeinek megsértése nélkül.

3. Ha nem zárt a fedés, megpróbáljuk további kompatibilisek hozzávételével vagy állapotok elhagyásával zárttá tenni.

4. lépés
A minimalizált állapottábla kitöltése

Ehhez szükség van az eredeti állapottáblára, és a kiválsztott zárt fedésre.

Az eredeti állapottábla:

x1x2	00	01	11	10
a	***c/0***	***-/-***	***a/0***	***-/-***
b	d/0	c/0	-/-	-/-
c	***-/-***	***e/1***	***a/0***	***-/-***
d	f/-	f/0	-/-	-/-
e	***f/0***	***f/-***	***-/-***	***-/-***
f	***-/-***	***-/-***	***a/0***	***-/-***

Válasszuk az (acef), (bdef) zárt fedést. Ekkor a minimalizált állapottábla:

x1x2	00	01	11	10
A(acef)	**A/0**	**-/1**	**A/0**	-/-
B(bdef)	B/0	A/0	A/0	-/-

Az **A** állapot **(acef)**-kompatibilisben szereplő állapotokat képviseli. x1x1=00 bemenetre **(acef)**-ből egy **cf**-et tartalmazó kompatibilisbe kell menni. Ezért ekkor a következő állapot szintén **A**. Mivel van olyan állapot, az a,**c**,e,**f** között, melyre a kimenet specifikált (**0**), ezért az új kimenet is ez az érték. Hasonló a többi rubrika kitöltése is.