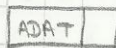
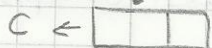
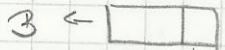
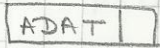


U csomópont legyen olyan elemmel reprezentálva, aminek egy mutatója van.

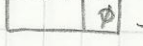
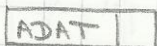


Mindkét elem mutató



Minden egyes csomópont mellé egy listát rendelünk. A lista feje maga a csomópont.

Egy n oldalú sokszög átlóinak száma: $\frac{n(n-1)}{2}$
 Ugyannyira a teljes gráf éleinek száma.



$n-1$

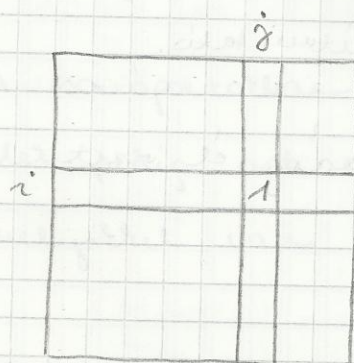
A adatkészletének értéke kezdésként.

ez az egyik

ez a másik

és a gráf egy megvalósítása ↑

A gráf egy másik megvalósítása:



$n \times n$
csúsmátrix

A i csúcs össze van kötve a j -déllel. (oda 1-et írunk)

Mérete: $C_{H,H}$

$$H^H - C^H - 2H$$

13
H

	1	2	3	4	5
1		1	1	1	1
2	1				
3	1				
4	1				
5	1				

A főátlóra szimmetrikus, de a főátlót és az alatti elemeket tárolni.

Ha felkennéljük a főátló alatti és feletti részt is, akkor lehet ábrázolni az irányított gráfot.

	1	2	3
1		1	1
2			
3	1		

Ez a fajta ábrázolás a gráfnál nem helytálló.

Ennél a megvalósításnál csak az utat tároljuk. Ha a szomszédos cella elemet akarjuk tárolni, azt külön kelljük meg.

Egy ilyen mátrix akkor van kikennélve, ha tele van 1-vel.

A mátrixba vehetjük fel i, j elemet (vagyis a gráfba).

Ha csak 3×3 -mátrixot tárolunk:

$$\text{ha } i < j \text{ akkor } G(i, j) = \text{igaz}$$

$$\text{különben } G(j, i) = \text{igaz};$$

A gráfot pontokkal nem tudjuk bontani

Fajtuk be a mátrix főátló feletti részét, és injuk

éi a szomszédos pontokat, amik össze vannak kötve.

1, 2

1, 3

1, 4

1, 5

Ilyenkor egy zámpár egy élt ír le.

Esztet tárolhatom olyan mátrixban, amiben

2 onlopa van.

Ez az elszomszédossági mátrix, de itt sem lehet új csúcsot felvenni.

Ez a zámpár megmondja, hogy a főátló felett hány 0-tól különböző érték van.

Gráfoknál van izolált pontja, ilyenkor ez nem szerepel az elvadászban.

Csak abban az esetben lehet egy gráf részait megadni az elvadászban, ha nincs izolált pontja.

Programok szerkezete

1, strukturált programtervezés

- blokk
- feltételes elágazás
- ismétlés

2, program matematikai fogalmai

- mat-i értelemben leírás

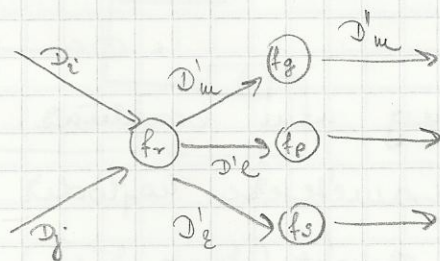
3, a program szerkezetének ábrázolása

4, a program olyan irányított gráf, amelynek csomópontjai

a, - predikátum-csomópontok 1 befutó és 2 kifutó él

b, - adatkezelő csomópont (függvény csomópont)

c, - gyűjtő csomópont



4/a, - erre tudunk elágazást megoldani

- felső ág, ha hamis

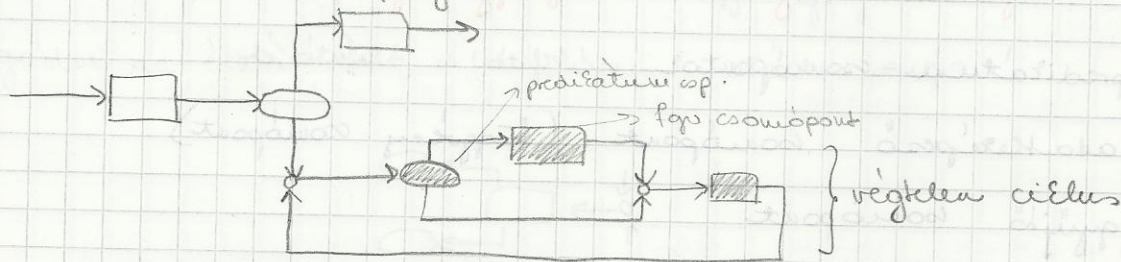
- alsó ág, ha igaz.

4/b, - 1 befutó és 1 kimenő él van

5. A valódi program olyan program, amelynek
- a₁ - vég es számú nem zeros beemenőele és kiemenőele van
 - b₁ - csomópontjait predikátum - csomópontok, fgv - csomópontok, gyűjtő - csomópontok alkotják.
 - c₁ - minden csomópontján legalább egy beemenő - éllel kezdődő és kiemenő éllel végződő útvonal vezet.

Mj.:
1; itt a legkisebb a bonyolultság

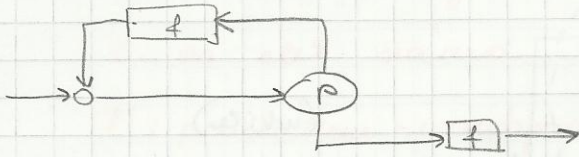
ÁBRA: Valódi program-e?



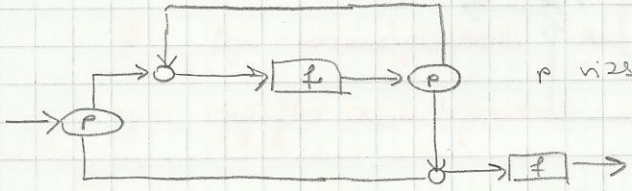
- 5/a; - nem lehet \emptyset , negatívól meg nem beindul.
 - a program a beemenő paraméterekre kiadja a végre, és ad valamilyen megoldást (kiemenő él)
 és végmond a működés eredménye

az ábra megvalósít az 5/c - t, mert a végtelen ciklusban nincs kiemenő él.

Ékvivalens programgráfok:



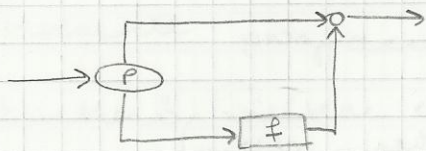
if p then ... és kilép



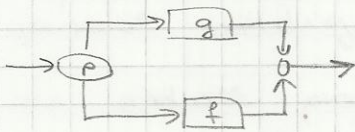
p vizsgál. f-cel visszatér kilépés

p, f, p, f, p ... f kilép

Strukturált programcskezetek: (1 bemenő és 1 kimenő él)

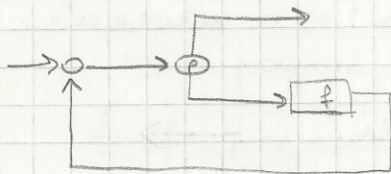


if p then f



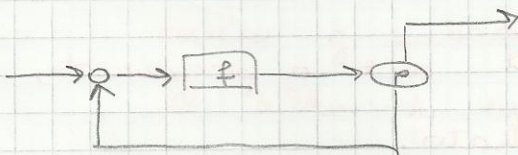
if p then f
else g

Szűrés



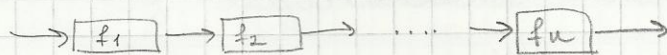
while p do f

iteráció



repeat f until p

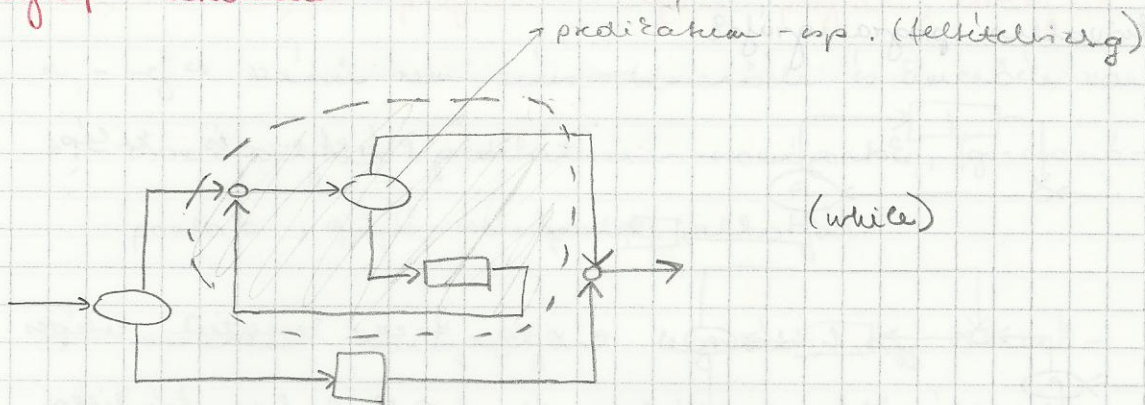
nem alapstruktúra, de minden olyan invariánsra rendelhető



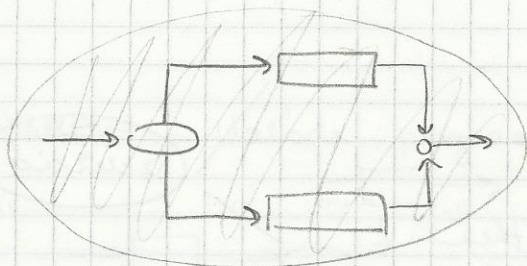
begin f1, f2 ... fn end

szekvencia

Reszlograf lebontás:



lebontás: ha felfedeztük benne 1 alopformát, helyettesítjük egy fog-ismérvontal



(if ... then ... else ...)



az alopforma! \Rightarrow a nyelvünk

Strukturált program lebontás célja! (\rightarrow)

Strukturált programmal nevezett azt a programot, amelynek részlográfia lebontható.

Tétel: (Böhm - Jacopini)

\forall valódi programhoz meg lehet konstruálni a vele ekvivalens strukturált programot.

Lemma: $\epsilon = 3\pi + \phi + 1$

$\pi = \gamma$

ϵ : az él szám

π : predikátum - csomópontok száma

ϕ : fgv - csomópontok száma

γ : gyűjtő csomópontok száma

\neq lépésért megvalósító f program felülől történő kifejtése az az eljárás, amely elsőként az f programot a bevezetett programegység segítségével valósítja meg, majd a következőként a programegységben szereplő lépéseket ismét valamilyen programegységgel valósítja meg, és az eljárást addig ismételi, míg a bevezetett lépéssel az adott programozási nyelv függvényeivel, utasításával kifejezhetővé nem válik.

Tétel: (A program felülől lefelé történő kifejtéséről kétértelműségi feltétel)

az f program felülől lefelé történő kifejtésekor

a) $f = B(g, k)$ begin fgv-csomópontok

ezek g szabadon választható, mint f hurokja, de akkor k egyértelműen meghatározott

• végrehajtás sorrendje nyilvánvaló: 1; g 2; k

• g megválasztása determinálja a k -t.

b) $f = ITE(p; g, k)$ diff. then else... predikátum igaz és hamis ág fgv-csomópontjai

ezek p szabadon választható, de akkor g és k egyértelműen meghatározott.

c.) $f = WD(p, g)$

esetén g szabadon választható, de akkor p egyértelműen meghatározott.

A nem strukturáltság jellemzői

Tétel: Egy program \Leftrightarrow nem strukturált, ha részgráfiaként előfordul a több E_i -, ill. beépő éllel rendelkező állus vagy döntés.

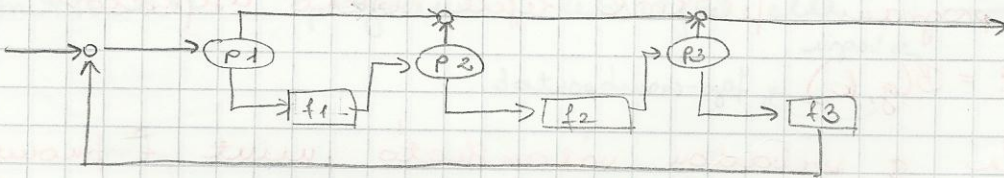
tétel

feltétel \Rightarrow állítás

$f \Leftrightarrow a \rightarrow$ megfordítható, kihat $a \Rightarrow f$.

pl.: $c^2 = a^2 + b^2 \Rightarrow \triangle$

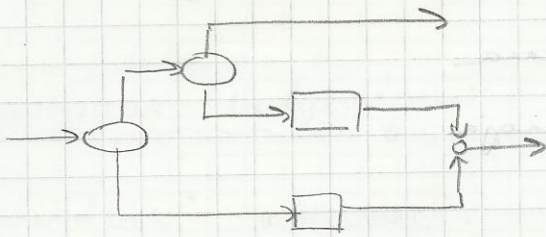
(Böhm-Jacopini példája nem strukturált programra)



- előrenkelt állusnak kiírtató, de innen körben is E_i lehet lépni \Rightarrow a program végét jelöli

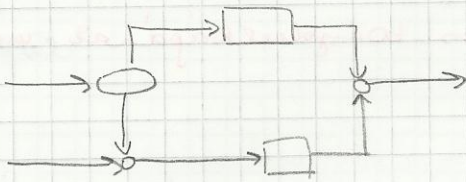
Tétel: A nem strukturált program, a nem strukturált alaptörvény gráfjai közül legkevesebb 2-t tartalmaz.

ilyenes: - döntés több kilepő éllel

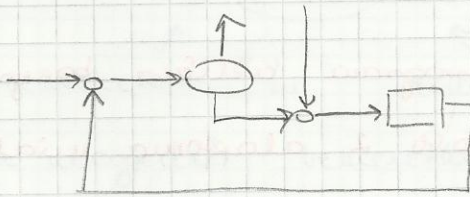


(elágasítás)

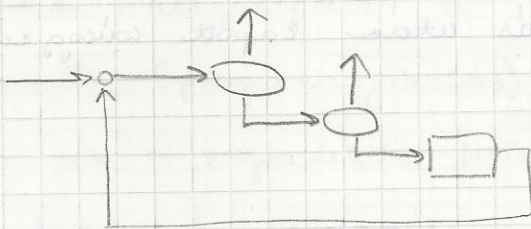
- döntés, több bemenő éllel



- átlus több bemenő éllel



- átlus több kilepő éllel



Program szerkeszti komplexitása

összefüggő gráf:

Egy G összefüggő gráfban a lineárisan független élek száma maximuma:

$$w(G) = N - C + 1$$