

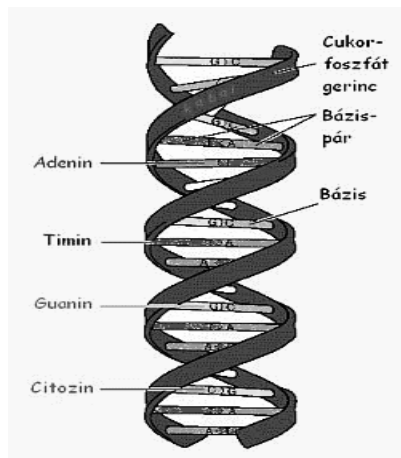
Az ember legfontosabb energiaforrására a cukorra is ugyanez érvényes, csak fordítva, hiszen az élő szervezet csak „jobbra forgató” cukrokat gyárt és képes felhasználni, míg a balra forgatók az élő szervezetben nem hasznosulnak. Valószínű ezzel magyarázható, hogy a DNS molekulák csak egyféle, jobbsaváros hélixet képeznek (3. ábra).

Létezik-e mélyebb kapcsolat az élő szervezetek és a részecskék szimmetriatlajdonsága közt, vagyis a neutrínó-aminosav, antineutrínó-cukor párok viselkedése mögött nem húzódik-e meg egy általános érvényű törvényszerűség? Talán érdekes kérdésfeltevés lehet a szaktudósok számára, de az is lehet, hogy pusztán véletlen a hasonló viselkedés.

Talán önkéntelenül is megfogalmazódik bennünk a kérdés, hogy miért csak közelítőleg szimmetrikus a természet? A válasz ma még nem ismeretes, de mivel bevezetőnek Feynman nyilatkozatát választottam, ezért záróként is őt idézem:

„A természeti törvények csak közelítően szimmetrikusak, nehogy féltékenyek legyünk a természet tökéletességére!” (Feynman: Mai fizika)

Borbély Éva



3. ábra

Algoritmusok tervezése

II. rész

Algoritmusok, programok leírására, tervezésére a következő grafikus vagy szöveges ábrázolási módokat szokás használni:

- folyamatábrák
- struktogramok (box diagram, Chapin chart, Nassi-Shneiderman chart, program struktúra diagram)
- pszeudokód
- Warnier-Orr-diagram
- Jackson-diagram

A cikk második részében a Warnier-Orr, valamint a Jackson-diagramokat mutatjuk be.

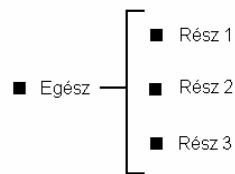
Warnier-Orr-diagram

A Warnier-Orr-diagramok elegánsan és egyszerűen ábrázolják a komplex számítási folyamatokat, algoritmusokat.

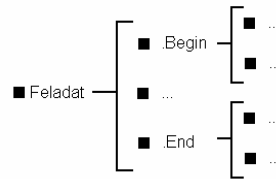
A diagramok nyolc egymáshoz hasonló elemből, blokkból épülnek fel, ezek a következők:

- Hierarchia, struktúra

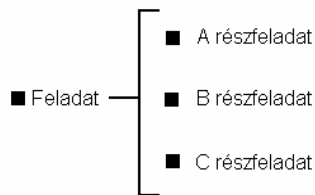
- Begin-End blokkok
- Szekvencia
- Ismétlés, jelölése: $(szám)$, vagy (k, v) , $szám$ -szor, vagy k -től, v -ig ismétli
- Rekurzió
- Párhuzamosság, logikai és, jelölése: $+$, vagy AND
- Elágazás, választás, logikai vagy, jelölése: \oplus , vagy OR
- Komplementer-képzés, logikai tagadás, jelölése: entitás



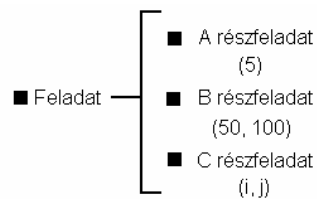
Hierarchia (az egész három részből áll)



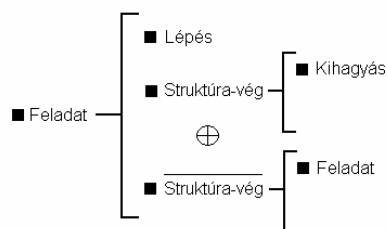
Begin-End blokk



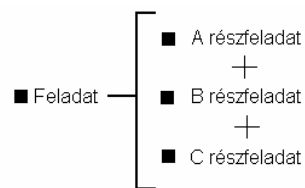
Szekvencia (először az A, azután a B, majd a C)



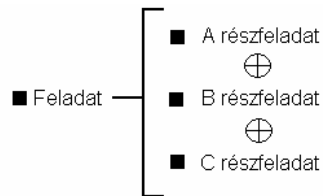
Ismétlés (5-ször végrehajtja az A részfeladatot, azután 50-től 100-ig a B részfeladatot, azután pedig i-től j-ig a C részfeladatot)



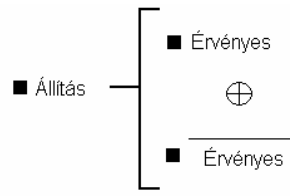
Rekurzió



Párhuzamosság (az A, B és C részfeladatok párhuzamosan hajtódnak végre)

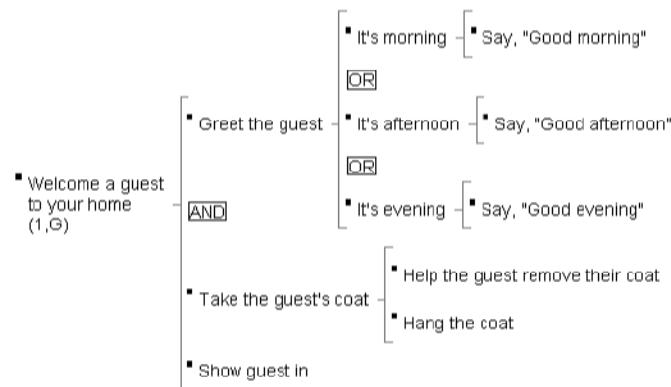


*Elágazás
(választani lehet A, B és C közül)*



*Tagadás
(az állítás lehet érvényes vagy nem érvényes)*

Példa: A alábbi Warnier-Orr-diagram egy vendég fogadását ábrázolja otthonunkban.



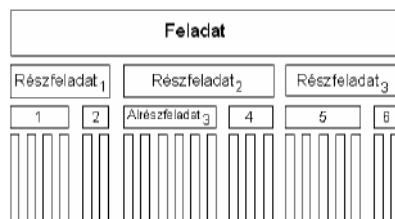
Jackson-diagram

A Jackson-diagram a procedurális absztrahálás logikai szerkezetének ábrázolására, könnyű áttekintésére, elemzésére kialakult technika.

A programozási feladatok nagy része részfeladatokra bontható. A részfeladatoktól függően a felosztás lehet:

- minden részfeladat független a többitől és önmagában is egy feladatot képez (pl.: írjunk egy olyan programot, amely 10 adott fraktálfüggvény esetén megrajzolja a fraktál képét a képernyőn)
- a részfeladatok függetlenek, de a megoldásuk kombinációjából alakul ki a feladat megoldása (pl.: írjunk egy olyan rajzolóprogramot, amely rendelkezik a következő rajzoló funkciókkal: vonalrajzolás, téglalaprajzolás, ellipszisrajzolás, satírozott téglalap, satírozott ellipszis rajzolása, adott kerületű sokszög kitöltése stb.)
- létezik néhány alaprészfeladat, ezekre épül egy néhány komplexebb részfeladat és így tovább (pl.: objektumhierarchia tervezése)

Absztrahálás esetén különböző részfeladatokra egy közös megoldást próbálunk keresni.



Procedurális absztrahálás

A Jackson-diagram (Michael Jackson – nem az énekes) által megalkotott egyszerű ábrázolásmód, nyilakat, téglalapokat és a téglalapokban bizonyos szimbólumokat használ az algoritmus leírására.

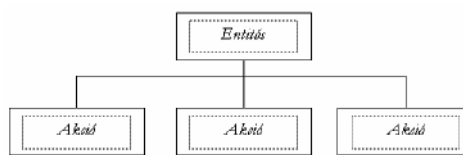
A szimbólumok a következők (ezeket a téglalapok jobb-felső sarkába kell rajzolni):

- *: iteráció, ciklus
- o: elágazás, választási lehetőség (o – *option*)
- -: null-komponens (pl. egy *if-then-else* típusú elágazásnál, ha nincs *else-ág*)

A téglalapokba kerülnek az *akciók* és az *entitások*. Az *entitásokat* a rendszer működteti, az *akciókat* pedig az entitások és ezek más entitásokat érinthetnek. Az akciók és az entitások hierarchiába szervezhetők, így alakulnak ki a struktúrák vagy szekvenciák.



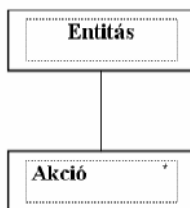
Akcio és Entitas



Entitas és akcio hierarchia: szekvencia, struktura



Elagazás, választás



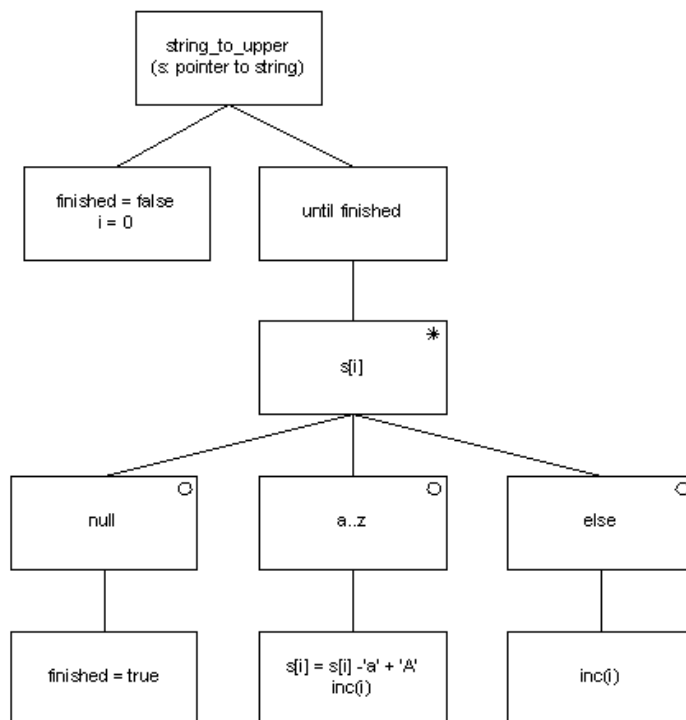
Ciklus, iteracio



Null-komponens

Példa: Rajzoljuk fel egy karaktersorozat nagybetűssé alakításának függvényét Jackson-diagram segítségével. A karaktersorozat C-típusú null-terminál karaktersorozat. Egy ciklussal bejárjuk a karaktersorozatot (a 0 indextól 'null'-ig). Ha a karaktersorozat

aktuális karaktere kisbetű (a..z), akkor nagybetűssé alakítjuk (kivonjuk a kis 'a' betű kódját és hozzáadjuk a nagy 'A' betű kódját), ha nem kisbetű, akkor továbblépünk.



Számos olyan automatikus segédeszköz, alkalmazás létezik, amelyek megkönnyítik a cikkünkben bemutatott diagramok rajzolását. Némelyek képesek arra is, hogy a megrajzolt diagram alapján az algoritmust programmá alakítsák, automatikusan kódot generáljanak pl. *Pascal* vagy *C nyelvre*.

Egy ilyen alkalmazás pl. a *B-liner 2002*, amely elérhető a www.varatek.com honlapon.

Kovács Lehel