

IX. Komputeralgebra rendszerek

Programozás

Czirbusz Sándor
czirbusz@gmail.com

Komputeralgebra Tanszék
ELTE Informatika Kar

2010-2011 őszi

Index

- 1 Értékkadás
 - Maple-ban
 - Sage-ben
- 2 Feltételes utasítás
- 3 Az IF a Maple-ban
 - IF a Sage-ben
- 4 Ciklusok
 - Ciklusok a Maple-ban
 - for/from ciklus
 - for/in ciklus
 - while ciklus
 - Általános ciklusok
 - Ciklusok a Sage-ben

Az értékadás és a Maple

- `name := expr;`
`(name1, ..., namen) := (expr1, ..., exprn);`
- Tesztelés : **assigned**(name)
- Hozzárendelés visszavonása (a szimbólum azért marad):
 - `name:='name'`
 - **unassign**('name1','name2',...)
 - **evaln**(name);
- Az assign utasítás :`assign(a,b)`, `assign(A=B)`, `assign(L)` : mindkét oldal kiértékelése
 - „a,b” paraméterek: `a = b` (egyszerű változókra)
 - „A,B” paraméterek: A,B azonos hosszú kifejezés-sorozat
 - „L” paraméter: L egyenletek listája vagy halmaza.

Az értékkadás és a Sage

- Az „x” változót kivéve kötelező a `[varlist =] var('[varlist]')` deklaráció, ez egyben a meglévő változók értékeit meg is szünteti
- Értékkadás : `var = kif`
- Ugyanúgy megy a `var1, var2, ... = kif1, kif2, ...` forma is. Azonos számú változó és kifejezést esetén mindegyik változóba a megfelelő érték kerül.
- Ha csak egy változó és több kifejezés van, akkor a változóba egy „tuple” kerül

Feltételes utasítás I

```
if cond1 then  
    statement_sequ1  
elif cond2 then  
    statement_sequ2  
elif cond3 then  
    statement_sequ3  
    .....  
else  
    statement_sequN  
end if
```

Feltételes utasítás II

Feltételes kifejezések konstruálása és kiértékelése:

- Relációs műveletek : $<$, $<=$, $=$, $>=$, $>$, $<>$
- Logikai műveletek - **and**, **or**, **xor**, **implies**, **not**
- Logikai nevek- **true**, **false**, **FAIL**
- Kiértékelés : **evalb**, **testeq**, vagy **is**

Megjegyzés A logika logika háromértékű : true, false és FAIL

Feltételes utasítás

```
if cond1:
    statement_sequ1
elif cond2:
    statement_sequ2
elif cond3:
    statement_sequ3
    .....
else:
    statement_sequN
```

Nincs blokkvége, a kettőspont kötelező

Feltételes kifejezések konstruálása

- Feltételvizsgálat : $\text{kif1} == \text{kif2}$ vagy kif1 **is** kif2
- Logikai operátorok : **and**, **or**, **not**
- A logikai kiértékelések természetesebbek a Maple-énál

for/from ciklus

```
for counter from init by increment to final do  
    statement_seq  
end do
```

A **from**, a **by** és a **to** klauza opcionális, sorrendjük tetszőleges.
A default értékek: **by** =1, **from** = 0.

for/in ciklus

```
for variable in expr do  
    statement_seq  
end do
```

Az "expr" (iterálható struktúra) minden komponesén végigszalad a változó és végrehajtja az utasításokat. Ilyen pl a PHP **foreach** ciklusa.

while ciklus

```
while cond do  
    statement_seq  
end do
```

Általános ciklusok

```
for counter from init by increment to final  
while cond do  
statement_seq  
end do
```

```
for variable in expr while cond do  
    statement_seq  
end do
```

Ciklusfutás befolyásolása : **break,next**

A for iteráció I

```
for counter in range( ) :  
    statement_seq
```

A **range** paraméterei a szokásos kezdőérték (a default 0), végérték és léptetés (default = 1).

A más nyelvekben szokásostól, a végértékre már nem fut le a ciklus, mert itt futások számát jelenti.

Variánsai:

range : egy listát állít elő

xrange : nem listát készít, hanem iterátort, ezért gyorsabb

srange : a leglassab, szintén listát készít,
továbbparaméterezhető

A for iteráció II

```
for counter in obj:  
    statement_seq
```

Itt az „obj” egy iterálható objektum, többnyire lista. A kettőspont itt is kötelező, a ciklustörzs addig tart, míg ki nem lépünk az indentálásból.

A while ciklus

```
while logikai_felt:  
    statement_seq
```

A szokásos **while** szerkezet a Pythonos-szintaxissal.

Példák

Lásd a munkalapokon