

Állapotgépek, véges automaták

Dr. Schuster György

OE-KVK-MAI
schuster.gyorgy@kvk.uni-obuda.hu

2018. július 3.

A kérdés:

A kérdés:
hány állapota van egy standard
golyóstollnak?

A kérdés:

hány állapota van egy standard golyóstollnak?

A válasz:

A kérdés:

hány állapota van egy standard golyóstollnak?

A válasz:

4

A kérdés:
hány állapota van egy standard
golyóstollnak?

A válasz:

4



Hogyan? Miért?

0.



0.



1.



Golyóstoll

0.



1.



2.



Golyóstoll

0.



1.



2.



3.



Golyóstoll

0.



1.



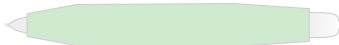
2.



3.



0.



Golyóstoll

0.



1.



2.



3.



0.



1.



2.



3.



Hát így!

0.



1.



2.



3.



Hát így!

A golyóstoll egy **állapógép**.

Golyóstoll

0.



1.



2.



3.



Hát így!

A golyóstoll egy **állapotgép**.



Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.
Az éleken a "változások" szerepelnek.

Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.
Az éleken a "változások" szerepelnek.
A szögpontokban az állapot "neve" van.

Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Az éleken a "változások" szerepelnek.

A szögpontokban az állapot "neve" van.

Vannak visszahajló élek is. Ezek általában az állapotban maradás feltételei.

Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Az éleken a "változások" szerepelnek.

A szögpontokban az állapot "neve" van.

Vannak visszahajló élek is. Ezek általában az állapotban maradás feltételei.

Tehát:

Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

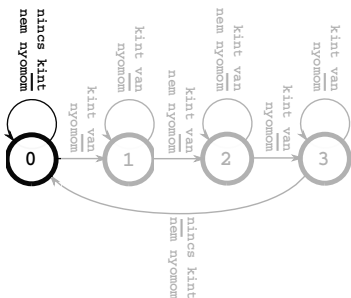
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Az éleken a "változások" szerepelnek.

A szögpontokban az állapot "neve" van.

Vannak visszahajló élek is. Ezek általában az állapotban maradás feltételei.

Tehát:



Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

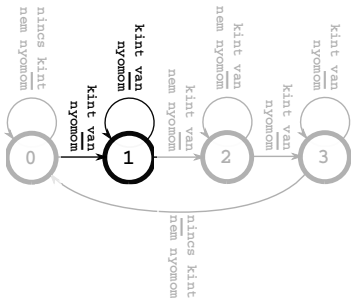
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Az éleken a "változások" szerepelnek.

A szögpontokban az állapot "neve" van.

Vannak visszahajló élek is. Ezek általában az állapotban maradás feltételei.

Tehát:



Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

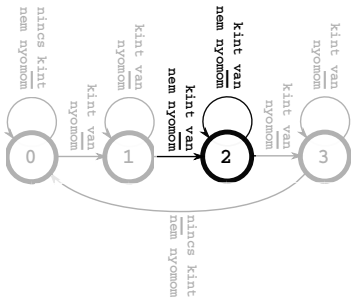
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Az éleken a "változások" szerepelnek.

A szögpontokban az állapot "neve" van.

Vannak visszahajló élek is. Ezek általában az állapotban maradás feltételei.

Tehát:



Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

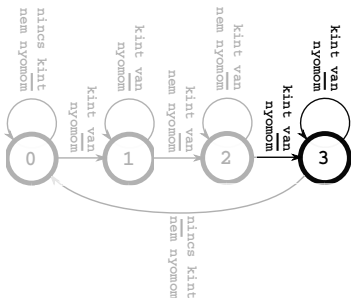
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Az éleken a "változások" szerepelnek.

A szögpontokban az állapot "neve" van.

Vannak visszahajló élek is. Ezek általában az állapotban maradás feltételei.

Tehát:



Állapotgráf

Az állapotgép leírásához a legegyszerűbb eszköz az állapotgráf.

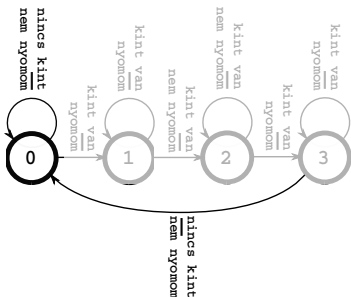
Ez egy gráf, tehát szögpontjai és élei vannak. Az élek irányítottak.

Az éleken a "változások" szerepelnek.

A szögpontokban az állapot "neve" van.

Vannak visszahajló élek is. Ezek általában az állapotban maradás feltételei.

Tehát:



Az állapot

Az állapot szimbóluma egy kör.



Az állapot

Az állapot szimbóluma egy kör.
Az állapot neve a kör belsejében.



Az állapot

- Az állapot szimbóluma egy kör.
- Az állapot neve a kör belsejében.
- "Mitől marad az állapotban" él.



Az állapot

Az állapot szimbóluma egy kör.

Az állapot neve a kör belsejében.

"Mitől marad az állapotban" él.

***B** bemenetek állapota, **K** kimenetek állapota.*



Az állapot

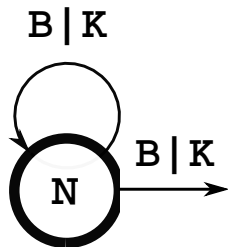
Az állapot szimbóluma egy kör.

Az állapot neve a kör belsejében.

"Mitől marad az állapotban" él.

B bemenetek állapota, *K* kimenetek állapota.

"Következő állapotba" él.



Az állapot

Az állapot szimbóluma egy kör.

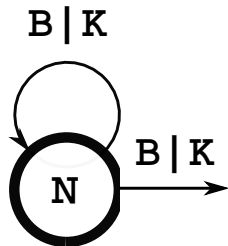
Az állapot neve a kör belsejében.

"Mitől marad az állapotban" él.

B bemenetek állapota, *K* kimenetek állapota.

"Következő állapotba" él.

B és *K* mint az előbb.



Az állapot

Az állapot szimbóluma egy kör.

Az állapot neve a kör belsejében.

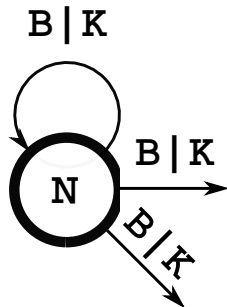
"Mitől marad az állapotban" él.

B bemenetek állapota, K kimenetek állapota.

"Következő állapotba" él.

B és K mint az előbb.

Lehet többféle kimeneti irány is.



Az állapot

Az állapot szimbóluma egy kör.

Az állapot neve a kör belsejében.

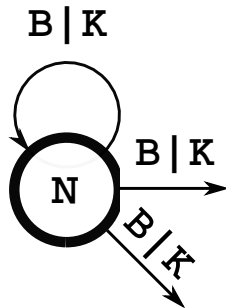
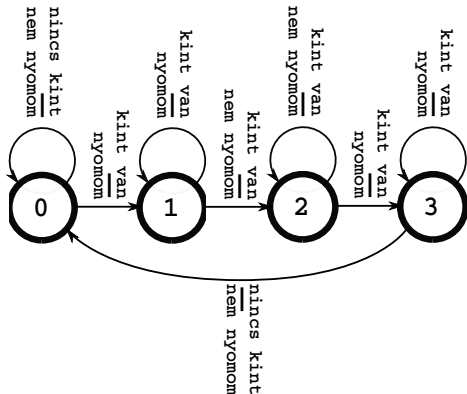
"Mitől marad az állapotban" él.

B bemenetek állapota, **K** kimenetek állapota.

"Következő állapotba" él.

B és **K** mint az előbb.

Lehet többféle kimeneti irány is.



Hogyan programozzuk le?

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Két alapvető lehetőség van:

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Két alapvető lehetőség van:

- "gyengén fogott állapotok",
- "erősen fogott állapotok",
- keverhetjük ezeket.

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Két alapvető lehetőség van:

- "gyengén fogott állapotok",
- "erősen fogott állapotok",
- keverhetjük ezeket.

Erősen fogott állapot:

- az állapot a saját magára jellemző kimeneteket állítja.

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Két alapvető lehetőség van:

- "gyengén fogott állapotok",
- "erősen fogott állapotok",
- keverhetjük ezeket.

Erősen fogott állapot:

- az állapot a saját magára jellemző kimeneteket állítja.

Gyengén fogott állapot:

- az állapot a következő állapotra jellemző kimeneteket állítja.

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Két alapvető lehetőség van:

- "gyengén fogott állapotok",
- "erősen fogott állapotok",
- keverhetjük ezeket.

Erősen fogott állapot:

- az állapot a saját magára jellemző kimeneteket állítja.

A következőképpen:

Gyengén fogott állapot:

- az állapot a következő állapotra jellemző kimeneteket állítja.

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Két alapvető lehetőség van:

- "gyengén fogott állapotok",
- "erősen fogott állapotok",
- keverhetjük ezeket.

Erősen fogott állapot:

- az állapot a saját magára jellemző kimeneteket állítja.

A következmény:

Gyengén fogott állapot:

- az állapot a következő állapotra jellemző kimeneteket állítja.

- A kimenetet egy másik állapot tartósan átírhatja.

Hogyan programozzuk le?

A véges automaták mindig végtelen ciklusban futnak.

Két alapvető lehetőség van:

- "gyengén fogott állapotok",
- "erősen fogott állapotok",
- keverhetjük ezeket.

Erősen fogott állapot:

- az állapot a saját magára jellemző kimeneteket állítja.

A következőképpen:

- A kimenetet egy másik állapot csak pillanatokra írhatja át.

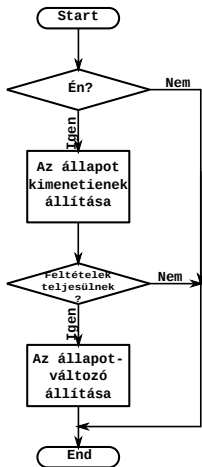
Gyengén fogott állapot:

- az állapot a következő állapotra jellemző kimeneteket állítja.

- A kimenetet egy másik állapot tartósan átírhatja.

”Erősen fogott” állapot

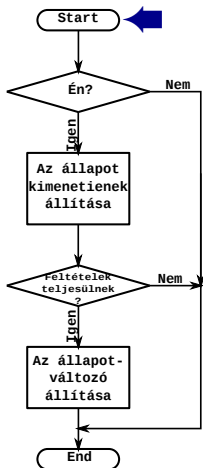
Működése:



”Erősen fogott” állapot

Működése:

- a program belép az állapotba,



”Erősen fogott” állapot



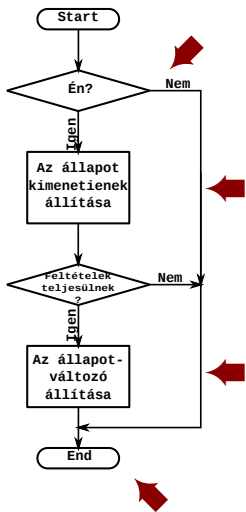
Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,

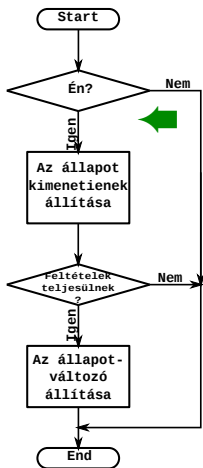
”Erősen fogott” állapot

Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,



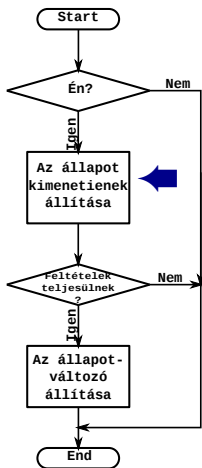
”Erősen fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,

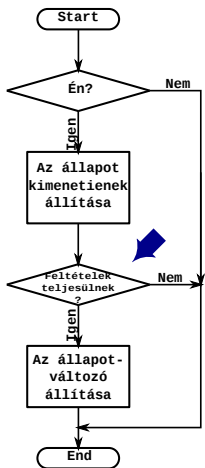
”Erősen fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- állítja a kérdéses kimeneti változókat,

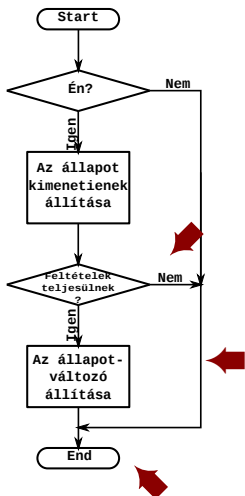
”Erősen fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- állítja a kérdéses kimeneti változókat,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e.

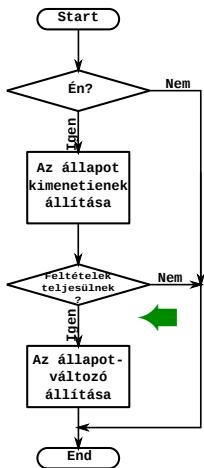
”Erősen fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- állítja a kérdéses kimeneti változókat,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,

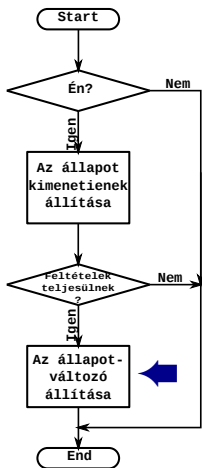
”Erősen fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- állítja a kérdéses kimeneti változókat,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,
- ha igen, folytatja,

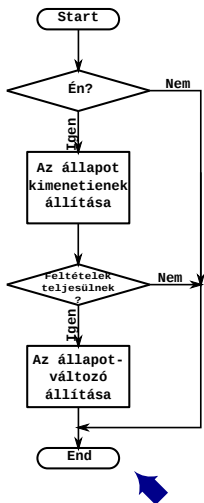
”Erősen fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- állítja a kérdéses kimeneti változókat,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,
- ha igen, folytatja,
- módosítja az állapot változót,

”Erősen fogott” állapot

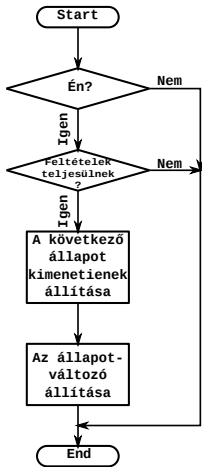


Működése:

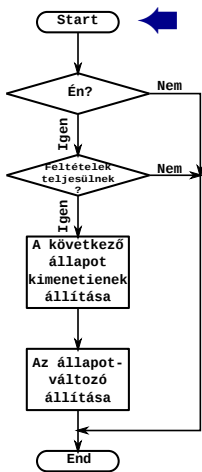
- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- állítja a kérdéses kimeneti változókat,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,
- ha igen, folytatja,
- módosítja az állapot változót,
- kilép az állapotból.

”Gyengén fogott” állapot

Működése:



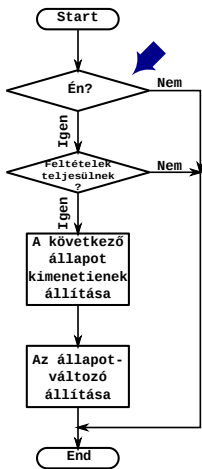
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,

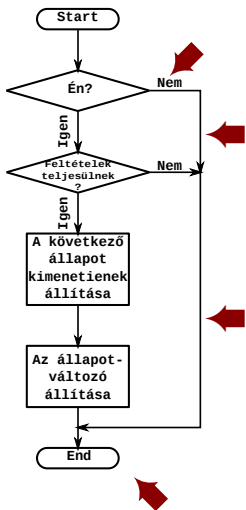
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,

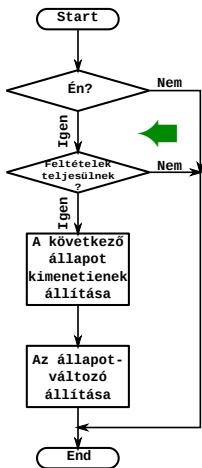
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,

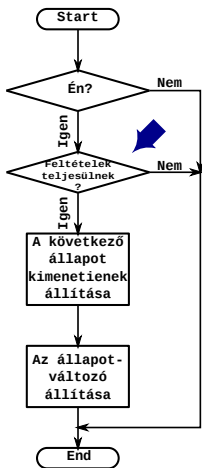
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,

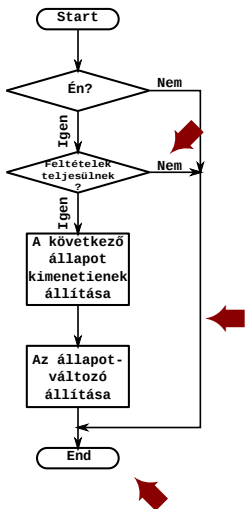
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e.

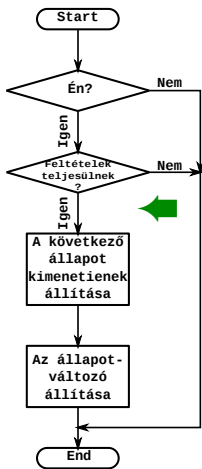
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,

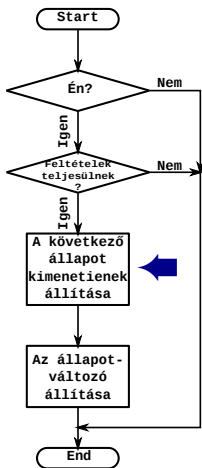
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,
- ha igen, folytatja,

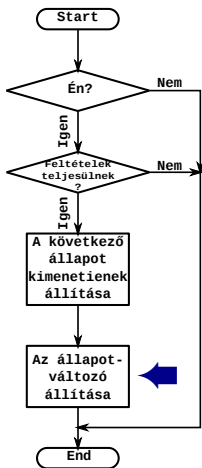
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,
- ha igen, folytatja,
- beállítja a következő állapot kimeneteit,

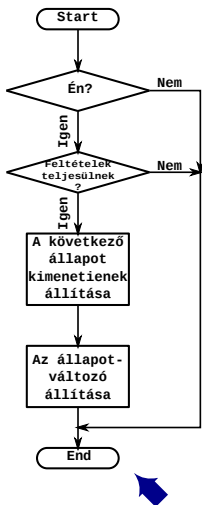
”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,
- ha igen, folytatja,
- beállítja a következő állapot kimeneteit,
- módosítja az állapot változót,

”Gyengén fogott” állapot



Működése:

- a program belép az állapotba,
- megvizsgálja, hogy az állapot aktív-e,
- ha nem kilép az állapotból,
- ha igen folytatja,
- megvizsgálja, hogy a továbblépés feltételei teljesültek-e,
- ha nem, kilép az állapotból,
- ha igen, folytatja,
- beállítja a következő állapot kimeneteit,
- módosítja az állapot változót,
- kilép az állapotból.

Az állapotgép működése

Alapelvek:

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,

Magyarázat:

a bemeneti változókat és állapotokat folyamatosan ellenőrizni kell.

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,

Magyarázat:

így a legegyszerűbb kezelni a szekvenciát.

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,

Magyarázat:

az állapotváltozónak meg kell maradnia a függvényből kilépés után is.

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,

Magyarázat:

egy várakozási ciklus leállítja a főciklus futását.

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,
5. egy állapotgép mindig csak egy állapotot szolgál ki egyszerre,

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,
5. egy állapotgép mindig csak egy állapotot szolgál ki egyszerre,

Magyarázat:

hagyni kell a többi állapotgépnek is időt.

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,
5. egy állapotgép mindig csak egy állapotot szolgál ki egyszerre,
6. az összes állapotgép lehetőleg egyetlen ciklusba legyen futtatva.

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,
5. egy állapotgép mindig csak egy állapotot szolgál ki egyszerre,
6. az összes állapotgép lehetőleg egyetlen ciklusba legyen futtatva.

Magyarázat:

így egyszerűbb az állapotgépek kezelése.

Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,
5. egy állapotgép mindig csak egy állapotot szolgál ki egyszerre,
6. az összes állapotgép lehetőleg egyetlen ciklusba legyen futtatva.

```
    :  
while (1)  
{  
    stm0 ();  
    stm1 ();  
    stm2 ();  
    :  
    stmX ();  
}  
    :
```


Az állapotgép működése

Alapelvek:

1. az állapotgép mindig ciklus belsejében fut,
2. egy állapotgép egy szubrutin, vagy függvény,
3. az állapotváltozó statikus, egész jellegű változó,
4. az állapotgép soha nem tartalmazhat várakozási ciklust,
5. egy állapotgép mindig csak egy állapotot szolgál ki egyszerre,
6. az összes állapotgép lehetőleg egyetlen ciklusba legyen futtatva.

```
⋮  
while (1)  
{  
  stm0();  
  stm1();  
  stm2();  
  ⋮  
  stmX();  
}
```

Az **stm** függvények az állapotgépeket megvalósító függvények.

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:saját
            kimetek
            állítása
            if(feltétel)
            {
                st=új_állapot;
            }
            break;
        case 1:
            :
            :
            break;

        :
        :
        case N:
            :
            :
            break;
    }
}
```

”Erősen fogott” C-ben

Programkód

```
void stm0(void) ←  
{  
  switch(st)  
  {  
    case 0:saját  
      kimetek  
      állítása  
      if(feltétel)  
      {  
        st=új_állapot;  
      }  
      break;  
    case 1:  
      :  
      :  
      break;  
  
      :  
      :  
    case N:  
      :  
      :  
      break;  
  }  
}
```

- az állapotgép kezelő függvény,

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
  ➔ switch(st)
  {
    case 0:saját
      kimetek
      állítása
      if(feltétel)
      {
        st=új_állapot;
      }
      break;
    case 1:
      :
      :
      break;
    :
    :
    case N:
      :
      :
      break;
  }
  ➔
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:saját
            kimetek
            állítása
            if(feltétel)
            {
                st=új_állapot;
            }
            break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        ➡ case 0: saját
            kimetek
            állítása
            if(feltétel)
            {
                st=új_állapot;
            }
        ➡ break;
        case 1:
            :
            :
            break;

            :
            :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0: saját kimenetek állítása
                if(feltétel)
                {
                    st=új_állapot;
                }
                break;
        case 1:
                :
                :
                break;

        :
        :
        case N:
                :
                :
                break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- a kimenetek állítása,

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:saját
            kimetek
            állítása
            ➔ if(feltétel)
                {
                    st=új_állapot;
                }
            break;
        case 1:
            :
            :
            break;

        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- a kimenetek állítása,
- továbblépési feltétel vizsgálat,

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:saját
            kimenetek
            állítása
            if(feltétel)
            {
                ➡ st=új_állapot;
            }
            break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok `switch` szerkezete,
- az állapotváltó,
- az állapot (0.),
- a kimenetek állítása,
- továbblépési feltétel vizsgálat,
- új érték az állapotváltóznak,

”Erősen fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:saját
            kimetek
            állítása
            if(feltétel)
            {
                st=új_állapot;
            }
            break;
        case 1:
            :
            :
            break;

        :

        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- a kimenetek állítása,
- továbblépési feltétel vizsgálat,
- új érték az állapotváltóznak,
- a **break;** (mindenképpen),

"Erősen fogott" C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:saját
            kimetek
            állítása
            if(feltétel)
            {
                st=új_állapot;
            }
            break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- a kimenetek állítása,
- továbblépési feltétel vizsgálat,
- új érték az állapotváltóznak,
- a **break;** (mindenképpen),
- a többi állapot.

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:if(feltétel)
            {
                következõ
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
            break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

”Gyengén fogott” C-ben

Programkód

```
void stm0(void) ←  
{  
  switch(st)  
  {  
    case 0:if(feltétel)  
      {  
        következő  
        állapot  
        kimeteinek  
        állítása  
        st=új_állapot;  
      }  
    break;  
    case 1:  
      .  
      .  
    break;  
    .  
    .  
    case N:  
      .  
      .  
    break;  
  }  
}
```

- az állapotgép kezelő függvény,

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    ➔ switch(st)
    {
        case 0:if(feltétel)
            {
                következő
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
        break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
    ➔ }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:if(feltétel)
            {
                következő
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
        break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        ➔ case 0: if(feltétel)
            {
                következő
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
        ➔ break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:if(feltétel) ←
            {
                következő
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
        break;
        case 1:
            :
            :
        break;
        :
        :
        case N:
            :
            :
        break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- továbblépési feltétel vizsgálat,

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:if(feltétel)
            {
                következõ
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
        break;
        case 1:
            :
            :
        break;
        :
        :
        case N:
            :
            :
        break;
    }
}
```



- az állapotgép kezelõ függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- továbblépési feltétel vizsgálat,
- a kimenetek állítása,

”Gyengén fogott” C-ben


Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:if(feltétel)
            {
                következő
                állapot
                kimeteinek
                állítása
                ➡ st=új_állapot;
            }
        break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- továbblépési feltétel vizsgálat,
- a kimenetek állítása,
- új érték az állapotváltónak,

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:if(feltétel)
            {
                következő
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
            break; 
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- továbblépési feltétel vizsgálat,
- a kimenetek állítása,
- új érték az állapotváltóznak,
- a **break;** (mindenképpen),

”Gyengén fogott” C-ben

Programkód

```
void stm0(void)
{
    switch(st)
    {
        case 0:if(feltétel)
            {
                következő
                állapot
                kimeteinek
                állítása
                st=új_állapot;
            }
        break;
        case 1:
            :
            :
            break;
        :
        :
        case N:
            :
            :
            break;
    }
}
```

- az állapotgép kezelő függvény,
- az állapotok **switch** szerkezete,
- az állapotváltó,
- az állapot (0.),
- továbblépési feltétel vizsgálat,
- a kimenetek állítása,
- új érték az állapotváltónak,
- a **break;** (mindenképpen),
- a többi állapot.

