2. tétel

a. Bitmozgató műveletek, Neumann ciklus

Bitenkénti léptetés

Léptető utasításokat használ több alapfunkció (soros-párhuzamos konverzió), és a szorzás művelete is.

Léptetés (SHIFT)

A bitek értékeit egy helyi értékkel balra vagy jobbra léptetjük.

Jobbra léptetés esetén a balszélső bit értéke vagy nulla, vagy az előjelbittel megegyező lesz (aritmetikai shift), a jobb oldali bit értéke egy átviteli bitként jelenik meg, amit a gép eltárol.

Balra léptetés esetén mindig nulla lesz a jobb oldali bit értéke, a balról kilépő érték pedig átvitelként jelenik meg.

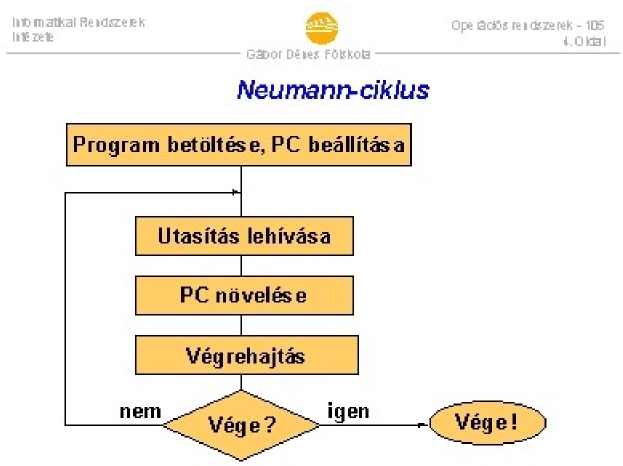
A fixpontos számoknál a balra léptetés kettővel szorzásnak, a jobbra léptetés kettővel osztásnak felel meg.

Forgatás (ROTATE)

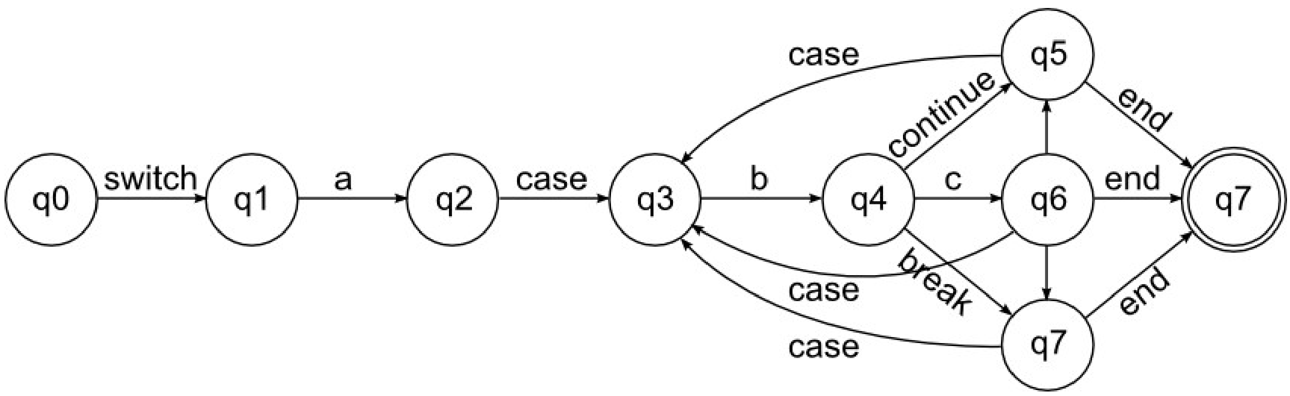
Balra rotálás esetén a kilépő bit vagy a legkisebb helyi értékre kerül, vagy eltárolódik átvitelként, és az előzőleg eltárolt átviteli bit kerül a legkisebb helyi értékre.

Jobbra rotálás esetén a kilépő bit vagy a legnagyobb helyi értékre kerül, vagy eltárolódik átvitelként, és az előzőleg eltárolt átviteli bit kerül a legnagyobb helyi értékre.

Neumann-ciklus:



b. Ismertesse a CASE szelekciót felismerő automatát



c. Az LL(k) grammatika

Az LL(1) grammatika

A G grammatikát egyszerű LL(1) grammatikának nevezzük, ha

1. ε-mentes
2. A helyettesítési szabályok jobboldala terminális szimbólummal kezdődik.
3. Alternatívák esetén a jobboldalak kezdő terminális páronként különbözőek, azaz $A \rightarrow a_1\alpha_1 \mid a_2\alpha_2 \mid ... \mid a_k\alpha_k$, ahol $a_i \neq a_j$, ha $i \neq j$.

Példa: G=({a, b, c, d}, {S,A}, S, P) ahol a helyettesítési szabályok a következők:

- $S \rightarrow aS|bAc$
- $A \rightarrow bAc|d$

Ez egy egyszerű LL(1) grammatika.

ε-mentes LL(1) grammatika

A FIRST(α)={A|α→a(β)}, azaz a FIRST(α) halmaz azokat a terminális szimbólumokat tartalmazza, amelyek az α-ból levezethető szimbólumsorozatok baloldalán állnak.
A G grammatikát ε-mentes LL(1) grammatikának nevezzük, ha minden A nemterminális szimbóluma $A \rightarrow \alpha_1 \mid \alpha_2 \mid ... \mid \alpha_k$ esetén $FIRST(\alpha_i) \cap FIRST(\alpha_j) = 0$, ha $i \neq j$.
Ha a G grammatika egyszerű LL(1) grammatika, akkor G egy ε-mentes LL(1) grammatika.
Ha a G grammatika ε-mentes LL(1) grammatika, akkor megoldható egy vele ekvivalens G' egyszerű LL(1) grammatika.
Példa: Legyen G a következő grammatika: G=({a,b,c,d,e}, {A,B,C,D,S}, S, P), ahol a helyettesítési szabályok a következők:

- $S \rightarrow ABC$
- $A \rightarrow a|Bbc|Ccd$
- $B \rightarrow bBb|cCc$
- $C \rightarrow dDd|Dd$
- $D \rightarrow \varepsilon$

Ez a G grammatika egy &epsilon-mentes LL(1) grammatika.

Az LL(k) grammatika

Az $S \Rightarrow^* wx$ legbaloldalibb levezetés építése során eljutunk a $S \Rightarrow^* wA\beta$ mondatformáig és az $A\beta \Rightarrow^* x$-et szeretnénk elérni, akkor az A-ra alkalmazható A > a helyettesítést egyértelműen meghatározhatjuk az x első k db szimbólumának előre olvasásával (ekkor és csak ekkor LL(k) nyelvtanról beszélhetünk).

Legyen $FIRST_k(\alpha)$ az α-ból levezethető szimbólumsorozatok k hosszúságú kezdő terminális sorozatainak halmaza, azaz $FIRST_k(\alpha) = \{x \mid \alpha \Rightarrow^* x\beta \text{ és } |x| = k\} \cup \{x \mid \alpha \Rightarrow^* x \text{ és } |x| < k\}$. Tehát a $FIRST_k(\alpha)$ az x első k darab szimbólumát, $|x|<k$ esetén pedig a teljes x-et tartalmazza. Ha $\alpha \Rightarrow^* \varepsilon$, akkor természetesen $\varepsilon \in FIRST_k(\alpha)$.

A G grammatika LL(k) grammatika, ha tetszőleges $S \Rightarrow^* wA\beta \Rightarrow w\alpha_1\beta \Rightarrow^* wx$ és $S \Rightarrow^* wA\beta \Rightarrow w\alpha_2\beta \Rightarrow^* wx$ levezetéspárra $FIRST_k(x) = FIRST_k(y)$ esetén $\alpha_1 = \alpha_2$. Eszerint tehát ha egy grammatika LL(k) grammatika, akkor a már elemzett w utáni k darab terminális szimbólum az A-ra alkalmazható helyettesítési szabályt egyértelműen meghatározza.