

Esim: html-listarakenteiden jäsennyys

Tehtävä: Laadi jäsennin, jolla voi tutkia, onko annettu merkkijono laillinen html:n ``- tai ``-listarakenne! Voit olettaa, että sisäkkäisiä listoja saa olla rajoittamattoman paljon, ja jokaiseen listaan on tallettu vähintään yhden kirjaimen alkio (ei siis sallita tyhjiä listoja).

Listarakenteet kuvaava kielioppi on seuraava:

$$S \rightarrow \langle ol \rangle L \langle /ol \rangle \mid \langle ul \rangle L \langle /ul \rangle \mid T$$
$$L \rightarrow \langle li \rangle S \mid \langle li \rangle SL$$
$$T \rightarrow AT \mid A$$
$$A \rightarrow a \mid b \mid \dots \mid z$$

(A :n säännöissä voi luetella kaikki tekstissä sallitut merkit.)

Otetaan käyttöön uudet apusymbolit: $B = \langle ol \rangle$, $C = \langle /ol \rangle$, $D = \langle ul \rangle$, $E = \langle /ul \rangle$, $F = \langle li \rangle$. Nyt kielioppi on seuraava:

$$S \rightarrow BLC|DLE|T$$

$$L \rightarrow FS|FSL$$

$$T \rightarrow AT|A$$

$$A \rightarrow a|b|\dots|z$$

$$B \rightarrow \langle ol \rangle$$

$$C \rightarrow \langle /ol \rangle$$

$$D \rightarrow \langle ul \rangle$$

$$E \rightarrow \langle /ul \rangle$$

$$F \rightarrow \langle li \rangle$$

Tämä voidaan muuntaa LL-muotoon (itse asiassa tarvitsemme LL(2)-muotoa: on tutkittava enintään 2 seuraavaa merkkiä, jotta tiedetään, mikä sääntö valitaan):

$$S \rightarrow BLC|DLE|T$$

$$L \rightarrow FSL'$$

$$L' \rightarrow L|\epsilon$$

$$T \rightarrow AT'$$

$$T' \rightarrow T|\epsilon$$

$$A \rightarrow a|b|\dots|z$$

$$B \rightarrow \langle ol \rangle$$

$$C \rightarrow \langle /ol \rangle$$

$$D \rightarrow \langle ul \rangle$$

$$E \rightarrow \langle /ul \rangle$$

$$F \rightarrow \langle li \rangle$$

Jäsennin pseudokoodina:

```
/*Apurutiinit: */  
token getnext; /* lue seuraava "sana" välilyöntiin */  
/* rivinvaihtoon, '>'- tai '<'-merkkiin saakka. Hyppää */  
/* alussa olevien välilyöntien yli */  
void Virhe; /* virheenkäsittely */  
/* Seuraava osanen: */  
char* next;
```

```
function S {  
if(next == '< ol >') { /* rule  $S \rightarrow BLC$  */  
    next = getnext;  
    L;  
    if(next != '</ol>') Virhe;  
}  
else  
    if (next == '< ul >') { /* rule  $S \rightarrow ELF$  */  
        next = getnext;  
        L;  
        if(next != '</ul>') Virhe;  
    }  
    elseT; /* rule  $S \rightarrow T$  */  
}
```

```

function L {
if (next == '< li >') { /* rule L → FS */
    next = getnext;
    S;
    if(next == '< li >') /* rule L → FSL */
        L;
}
else Virhe;
}

```

```

function T {
while((next! = '< /ol >') || (next! = '< ul >') || (next! = '< li >'))
    next = getnext;
}

```