A Paice–Husk-szótövező

Ahogy a könyvben bemutattuk, a szótövező változó számú iteratív lépésben hajtja végre az eljárást. A szabályokat a könnyebb kereshetőség érdekében a szó utolsó karaktere szerint rendezett csoportokban tárolják. A szabályok az alábbi komponensekből állnak:

- egy vagy több karakterből álló végződés, fordított sorrendben;
- opcionális eredetiséget jelölő bit: `*`;
- a levágandó karakterek számát meghatározó érték (≥ 0);
- opcionális, egy vagy több karakterből álló helyettesítő végződés;
- folytatást kódoló szimbólum: `>` jelzi a folytatást, `.` a terminálást.

A szabálykészlet mindössze 115 szabályt tartalmaz.

A PC-szótövező túltövezésre való hajlamát a szabályok alkalmazásával előálló szóalakra vonatkozó elfogadhatósági kritériumok korlátozzák. Ezek alapján, ha egy szó magánhangzóval kezdődik, akkor legalább még egy betűt kell a redukált alaknak tartalmaznia; ha viszont mássalhangzóval kezdődik, akkor a redukált alaknak legalább három betűsnek kell lennie, és a három betűből az egyiknek magánhangzónak kell lennie. Természetesen ezek a feltételek sem oldják meg teljes mértékben a problémát, hiszen a tövező vét alul- és túltövezési hibákat is, de jelentősen korlátozzák a túltövezések számát.

PÉLDA. Nézzünk két példát a PC-tövező működésére! A *provision* szó illeszkedik `nois4j>` szabályra, amit a következőképpen kell értelmezni. Az első négy karakter a fordított szóvégződés. Ennél a szabálynál nincs megkövetelve a szóalak eredeti, érintetlen volta — bár ez itt teljesülne —, hiszen eredetiségbitet a szabály nem tartalmaz. A redukálás során 4 karaktert vág le, és a *j*-t illeszti be, azaz a *provij* szóalakot állítja elő, amire az elfogadhatósági kritériumok is teljesülnek. A szabály utolsó szimbóluma azt jelzi, hogy ezzel az előállított szóalakkal még nem terminál a tövező.

A következő illeszkedő szabályt a j-csoportban találjuk, ezek közül a leghosszabban a `ji1d.` szabály illeszkedik. Ez előírja a szóvégi *j* cseréjét *d*-re, majd terminál a tövezés a *provid* szóalakkal. Érdemes megjegyezni, hogy a *provide* szóra is ezt a tövet állítja elő az algoritmus. Összefoglalva:

$$\textit{provision} \xrightarrow{\texttt{nois4j>}} \textit{provij} \xrightarrow{\texttt{ji1d.}} \textit{provid.}$$

A *multiplications* szó először az `s*1>` szabályra illeszkedik, és mivel ez az eredeti és sértetlen alak, ezért alkalmazható rá, az eredmény a többes szám levágásával előálló *multiplication* alak. Ez a `noi3>` szabályra illeszkedik, amely levágja az

utolsó három karaktert, így kapjuk a *multiplicat* alakot. Erre a tacilp4y. terminál szabály alkalmazható, ami előállítja a *multiply* szótövet. Összefoglalva:

$$\textit{multiplications} \xrightarrow{\text{s*1>}} \textit{multiplication} \xrightarrow{\text{noi3>}} \textit{multiplicat} \xrightarrow{\text{tacilp4y.}} \textit{multiply}.$$

Az algoritmusról további részletek és implementációk találhatóak a Paice–Husk-tövező hivatalos honlapján.