A Porter-szótövező

Az algoritmus részletes bemutatásához szükség van néhány definícióra (Porter, 1980). *Mássalhangzó* minden olyan betű, amely nem *A, E, I, O, U*, illetve mássalhangzó utáni *Y* (ebben a részben a szóalakokat, illetve azok részeit nagybetűvel írjuk). A *toy* szóban a *T* és az *Y* is mássalhangzó, a *syzygy*-ben *S, Z* és *G* a mássalhangzó. Ha egy betű nem mássalhangzó, akkor *magánhangzó*. A mássalhangzókat *c*-vel (consonant), a magánhangzókat *v*-vel (vowel) jelöljük. A mássalhangzók, illetve magánhangzók nem üres sorozatát rendre *C*, illetve *V* jelöli, azaz reguláris kifejezéssel $C = c^+$, és $V = v^+$. Evvel a jelölésmóddal minden szó az alábbi négy alak egyikét veheti fel:

$$CVCV\ldots V, \qquad CVCV\ldots C, \qquad VCVC\ldots C, \qquad VCVC\ldots V,$$

illetve ha $[X]$-szel jelöljük az X opcionális előfordulását, akkor ez egyszerűsíthető a

$$[C]VCVC\ldots[V], \quad \text{ill.} \quad [C](VC)^m\ldots[V]$$

alakokra, ahol a második kifejezésben az $m \geq 0$ kitevő a (VC) sorozat ismétlődésének számát jelzi.

Nézzünk néhány példát *m* különböző értékeire:

$m = 0$ *be, a, free*
$m = 1$ *this, trouble, eat*
$m = 2$ *lines, capote, eaten*

Ahogy azt a könyvben is megadtuk, az átírószabályok alakja

$$(\textit{feltétel})\ S_1 \rightarrow S_2.$$

A *feltétel* az S_1-től megfosztott szótő *m* értékére vonatkozó feltételen kívül az alábbi típusú megszorításokat tartalmazhatja:

- *S – a szóalak az adott betűvel végződik
- *v* – a szóalak magánhangzót tartalmaz
- *d – a szótő dupla mássalhangzóra végződik (pl. *-TT, -SS*).
- *o – a szóalak *cvc* mintára illeszkedik, ahol a második mássalhangzó nem *W, X*, ill. *Y* (pl. *-WIL, -HOP*).

A *feltétel* logikai operátorokat is tartalmazhat.

Az algoritmus öt fő lépésből áll. Az egyes lépéseken belül, ha egy szóalak több átírószabályra is illeszkedik, akkor az kerül végrehajtásra, amelyik a leghosszabban illeszkedik a szóalakra.

Az első lépés több részlépésből áll, amelyek a többes szám és a melléknévi igenevek jelének a levágását végzik. Az 1a. lépésben a többes szám jelének feldolgozása és újrakódolása történik. Az 1b. lépésben a múlt (*-D*) és jelen idejű melléknévi igenév (*-ING*) képzője kerül lemetszésre.

1a. lépés

1.	SSES → SS	*caresses → caress*
2.	IES → I	*ponies → poni, ties → ti*
3.	SS → SS	*caress → caress*
4.	S →	*cats → cat*

1b. lépés

5.	$(m>0)$ EED → EE	*feed → feed, agreed → agree*
6.	(*v*) ED →	*plastered → plaster, bled → bled*
7.	(*v*) ING →	*motoring → motor, sing → sing*

A következő lépést akkor kell végrehajtani, ha a 6. vagy a 7. szabályt sikeresen alkalmaztuk. Ezek a szabályok az 1b. lépés után megmaradt szóalakvégek újrakódolását végzik el, amire azért van szükség, hogy a későbbi szabályokat uniform módon alkalmazhassuk. Végül az 1. lépés utolsó eleme (1c) a szóvégi *Y*-t írja át *I*-re, ha van a szóalakban másik magánhangzó.

1b'. lépés

8.	AT → ATE	*conflat(ed) → conflate*
9.	BL → BLE	*troubl(ed) → trouble*
10.	IZ → IZE	*siz(ed) → size*
11.	$(*d \wedge \neg(*L \vee *S \vee *Z))$ → egybetűsre	*hopp(ing) → hop, tann(ed) → tan, fall(ing) → fall* *hiss(ing) → hiss, fizz(ed) → fizz*
12.	$(m=1 \wedge *o)$ → E	*fail(ing) → fail, fil(ing) → file*

1c. lépés

13.	(*v*) Y → I	*happy → happi, sky → sky*

A 2. lépés a dupla képzők második tagját vágja le. Az utolsó előtti karakter indexelése alapján a lépés hatékonyan megvalósítható. Hasonló indexelő technika alkalmazható a következő lépések során is (ld. még a könyv 2.2. ábráját).

2. lépés

14.	$(m>0)$ ATIONAL → ATE	*relational → relate*
15.	$(m>0)$ TIONAL → TION	*conditional → condition, rational → rational*
16.	$(m>0)$ ENCI → ENCE	*valenci → valence*
17.	$(m>0)$ ANCI → ANCE	*hesitanci → hesitance*

18. $(m > 0)$ IZER → IZE — *digitizer → digitize*
19. $(m > 0)$ ABLI → ABLE — *conformabli → conformable*
20. $(m > 0)$ ALLI → AL — *radicalli → radical*
21. $(m > 0)$ ENTLI → ENT — *differentli → different*
22. $(m > 0)$ ELI → E — *vileli → vile*
23. $(m > 0)$ OUSLI → OUS — *analogousli → analogous*
24. $(m > 0)$ IZATION → IZE — *vietnamization → vietnamize*
25. $(m > 0)$ ATION → ATE — *predication → predicate*
26. $(m > 0)$ ATOR → ATE — *operator → operate*
27. $(m > 0)$ ALISM → AL — *feudalism → feudal*
28. $(m > 0)$ IVENESS → IVE — *decisiveness → decisive*
29. $(m > 0)$ FULNESS → FUL — *hopefulness → hopeful*
30. $(m > 0)$ OUSNESS → OUS — *callousness → callous*
31. $(m > 0)$ ALITI → AL — *formaliti → formal*
32. $(m > 0)$ IVITI → IVE — *sensitiviti → sensitive*
33. $(m > 0)$ BILITI → BLE — *sensibiliti → sensible*

A 3. és 4. lépésben különböző képzők levágását végzi az algoritmus. A különbség a szóalak *m* értékére vonatkozó feltételben és az implementálásnál alkalmazandó karakter vizsgálatában van: a 3. lépésben az utolsó, míg a 4. lépésben az utolsó előtti karakter szerint kell elágazni.

3. lépés

34. $(m > 0)$ ICATE → IC — *triplicate → triplic*
35. $(m > 0)$ ATIVE → — *formative → form*
36. $(m > 0)$ ALIZE → AL — *formalize → formal*
37. $(m > 0)$ ICITI → IC — *electriciti → electric*
38. $(m > 0)$ ICAL → IC — *electrical → electric*
39. $(m > 0)$ FUL → — *hopeful → hope*
40. $(m > 0)$ NESS → — *goodness → good*

4. lépés

41. $(m > 1)$ AL → — *revival → reviv*
42. $(m > 1)$ ANCE → — *allowance → allow*
43. $(m > 1)$ ENCE → — *inference → infer*
44. $(m > 1)$ ER → — *airliner → airlin*
45. $(m > 1)$ IC → — *gyroscopic → gyroscop*
46. $(m > 1)$ ABLE → — *adjustable → adjust*
47. $(m > 1)$ IBLE → — *defensible → defens*

48. $(m > 1)$ ANT → *irritant → irrit*
49. $(m > 1)$ EMENT → *replacement → replac*
50. $(m > 1)$ MENT → *adjustment → adjust*
51. $(m > 1)$ ENT → *dependent → depend*
52. $(m > 1 \wedge (*S \vee *T))$ ION → *adoption → adopt*
53. $(m > 1)$ OU → *homologou → homolog*
54. $(m > 1)$ ISM → *communism → commun*
55. $(m > 1)$ ATE → *activate → activ*
56. $(m > 1)$ ITI → *angulariti → angular*
57. $(m > 1)$ OUS → *homologous → homolog*
58. $(m > 1)$ IVE → *effective → effect*
59. $(m > 1)$ IZE → *bowdlerize → bowdler*

Az utolsó lépésben csak az eredmények egységesítését végzi a módszer, a szóvégi *E* levágásával, illetve a kettős *LL* rövidítésével.

5a) lépés

60. $(m > 1)$ E → *probate → probat, rate → rate*
61. $(m = 1 \wedge \neg *o)$ E → *cease → ceas*

5b) lépés

62. $(m > 1 \wedge *d \wedge *L)$ → egybetűsre
controll → control, roll → roll

PÉLDA. Nézzük meg először, hogyan működik az algoritmus a *generalizations* szó esetén. Az alábbiakban a nyilakon feltüntettük az alkalmazott szabályok számát.

$$\textit{generalizations} \xrightarrow[4.\text{ szabály}]{1a.\text{ lépés}} \textit{generalization} \xrightarrow[24.\text{ szabály}]{2.\text{ lépés}} \textit{generalize}$$
$$\xrightarrow[36.\text{ szabály}]{3.\text{ lépés}} \textit{general} \xrightarrow[41.\text{ szabály}]{4.\text{ lépés}} \textit{gener}$$

A következő példán az *oscillators* szó szótövezését követhetjük végig.

$$\textit{oscillators} \xrightarrow[4.\text{ szabály}]{1a.\text{ lépés}} \textit{oscillator} \xrightarrow[26.\text{ szabály}]{2.\text{ lépés}} \textit{oscillate}$$
$$\xrightarrow[55.\text{ szabály}]{4.\text{ lépés}} \textit{oscill} \xrightarrow[62.\text{ szabály}]{5.\text{ lépés}} \textit{oscil}$$

A Porter-szótövező egy implementációja online kipróbálható ezen a honlapon.

Forrás:
M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
M. F. Porter. Snowball: A language for stemming algorithms, 2001.