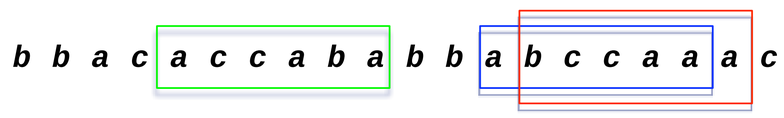
Kevert mintaillesztés – elmélet, algoritmusok és alkalmazások

Burcsi Péter
Eötvös Loránd Tudományegyetem, Informatikai Kar, Komputeralgebra Tanszék

Bevezetés

A mintaillesztési feladatokban egy szövegben kell megkeresnünk egy minta előfordulásait. A kevert mintaillesztésnél ugyanez a feladat, de most akkor is találatot jelzünk, ha a minta és a szövegrészlet nem pontosan ugyanabban a sorrendben tartalmazza a betűket.

Formálisan a következő feladatról van szó. Egy w szó $P(w)$ Parikh-vektora vagy abelianizáltja az a σ hosszúságú vektor, melynek k-adik komponense megegyezik a Σ ábécé k-adik elemének a w-beli előfordulásainak számával. Például az $\{a, b, c\}$ ábécé felett a *abbacabbab* szó abelianizáltja vagy Parikh-vektora $(4, 5, 1)$, ha megállapodunk abban, hogy az ábécé betűinek sorrendje a szokásos. A w szó összes Parikh-vektorának $PS(w)$ halmaza a w Parikh-halmaza; és ha az előfordulások multiplicitását is tekintjük, akkor Parikh multihalmazról beszélünk.
A vizsgált kevert mintaillesztési feladatban bemenetként adott egy szövegsztring és egy minta Parikh-vektor, kimenetként pedig arra a kérdésre kell válaszolnunk, hogy előfordul-e a minta Parikh-vektor a szövegsztringben – azaz eleme-e a Parikh-halmaznak. A kibővített változatban az előfordulások helyét is meg kell adni. Például az *abbabaaab* szóban a $(3, 1)$ vektor előfordul, a $(5, 1)$ pedig nem (az ábécé kételemű).



A teljes szövegben több helyen is előfordul a $(3, 1, 2)$ Parikh-vektor.

Rekonstrukciós feladat

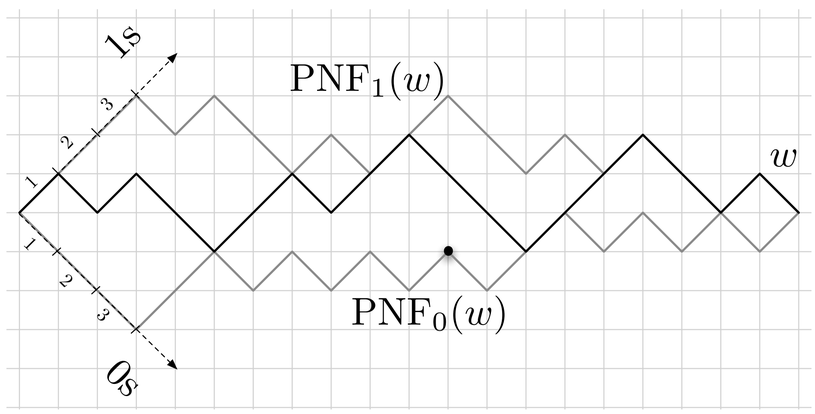
A témakör vizsgálatának eredeti motivációja többek között az volt, hogy tömegspektrometriai mérésekből hogyan lehet fehérjemolekulák szerkezetére, az aminosavak sorrendjére következtetni. A fenti formalizmust használva ez azzal a kérdéssel rokon, hogy a Parikh-halmazából vagy Parikh-multihalmazából rekonstruálható-e egy w szó. Szerzőtársaimmal ezt a kérdéskört vizsgáltuk. Néhány kutatási eredményünk:

- Végtelen sok példából álló családot konstruáltunk nem rekonstruálható sztringekre és fákra, különböző fajta rekonstrukciós feladatokhoz.
- Polinomok felhasználásával hatékony algoritmust adunk két fa Parikh-multihalmazának összehasonlítására, illetve speciális esetekben a rekonstrukcióra.

A fenti két fa („molekula") Parikh-halmazából, sőt Parikh-multihalmazából nem lehet felismerni a kettő közötti különbséget. Vagyis minden minteillesztési feladatra ugyanúgy és ugyanannyi találatot adnak. Például az $(1, 2)$ Parikh-vektor mindegyikben háromszor fordul elő.

Matematikai kapcsolatok

A témakör kapcsolódik a szavak kombinatorikájához. Formális definíciók helyett az alábbi ábrával szemléltetjük az ún. prefix-normális alakok és prefix-normális szavak fogalmát. E szavak mind kombinatorikailag, mind algoritmikusan érdekes kihívást jelentenek.



A középen húzódó cikkcakkos vastag vonal egy w 0-1 sorozatot (sztringet, szót) jelképez, 0 esetén lefelé, 1 esetén felfelé lépünk. A két szélső vonal közti rész a szó Parikh-halmazát ábrázolja. A két szélső vonal is felfogható szavakként, ezeket hívjuk a w prefix-normális alakjainak. Egy prefix-normális alak jelemzője, hogy minden k hosszú prefixe legalább annyi 1-est tartalmaz, mint bármely más k hosszú része (ez a felső vonalra igaz, az alsóra 0-val teljesül).

- A prefix-normális szavak kombinatorikai és formális nyelvi jellemzői.
- Algoritmusok prefix-normális szavak ellenőrzésére, előállítására és az összes adott hosszúságú prefix-normális szó generálására.

Algoritmusok a kevert mintaillesztési feladatra

A kevert mintaillesztési feladat lineáris időben – azaz a szó végigolvasásával – egyszerűen megoldható. Szerzőtársaimmal közösen ezen a naiv algoritmuson a következő javításokat tudtuk elérni:

- Wawelet tree adatszerkezet alkalmazásával várható értékben a lineárisnál gyorsabb algoritmus.
- Amennyiben előfeldolgozásra is lehetőségünk van, elérhető konstans idejű lekérdezés kételemű (0-1) ábécé esetén, ehhez hatékony előfeldolgozás.

Alkalmazások; jelenlegi és tervezett munka

A témakör részben a bioinformatikának a szekvenálással, molekula-rekonstrukcióval foglalkozó területeiből merít inspirációt. Doktoranduszaimmal közösen gyakorlati alkalmazásokon dolgozunk, ám ezek többsége még folyamatban levő munka.
Jelenleg elsősorban azon dolgozunk, hogy az elméleti eredményeknek és a kidolgozott algoritmusoknak alkalmazhatóságát vizsgáljuk meg kémiai és biológiai adatbázisokban történő keresési feladatokban.

- Lehetséges-e a kevert mintaillesztési feladatok eredményének felhasználása egzakt mintaillesztési feladatoknál történő szűréshez? Elég hatékonyan szeli-e részekre az adatbázist egy kevert keresés? Megfelelően szűri-e a ki a negatív találatok nagy részét? Nem hoz-e be túl sok fals pozitív találatot?
- Vannak-e olyan ipari alkalmazások, ahol egy kevert mintaillesztési feladat megoldása adja a releváns találatok nagy részét?
- Használható-e a spektrometria szerkezetmeghatározás más szekvenálási módok kiegészítésére pl. fehérjéknél vagy DNS-molekuláknál?

Ezen kívül mind a rekonstrukciós feladatnál, mind a prefix-normális szavak vizsgálatánál számos elméleti kérdésre keressük a választ.

- Mi történik, ha a címkéket nem egy véges ábécé, hanem egy algebrai struktúra (csoport vagy gyűrű) elemei közül választjuk?
- Le tudjuk-e írni a nem rekonstruálható sztringeket, fákat általánosan?
- Van-e olyan kevert mintaillesztési adathalmaz, mely már elégséges a teljes rekonstrukcióhoz?
- Mi a prefix-normális szavak aszimptotikusan pontos száma?
- Mekkora a prefix-normális szavak ekvivalenicaosztályainak mérete (átlagosan, illetve milyen az eloszlása)?

Hivatkozások

A fent leírt eredmények az alábbi publikációkban jelentek meg:

[1] D. Bartha and P. Burcsi.
Reconstructibility of trees from subtree size frequencies.
Stud. Univ. Babeş-Bolyai Math., 59(4):435–442, 2014.

[2] D. Bartha, P. Burcsi, and Zs. Lipták.
Reconstruction of Trees from Jumbled and Weighted Subtrees.
In R. Grossi and M. Lewenstein, editors, *27th Annual Symposium on Combinatorial Pattern Matching (CPM 2016)*, volume 54 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:13, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[3] P. Burcsi, F. Cicalese, G. Fici, and Zs. Lipták.
On Table Arrangements, Scrabble Freaks, and Jumbled Pattern Matching.
In *Proc. of the 5th International Conference on Fun with Algorithms (FUN 2010)*, volume 6099 of *LNCS*, pages 89–101, 2010.

[4] P. Burcsi, F. Cicalese, G. Fici, and Zs. Lipták.
Algorithms for jumbled pattern matching in strings.
Int. J. Found. Comput. Sci., 23(2):357–374, 2012.

[5] P. Burcsi, F. Cicalese, G. Fici, and Zs. Lipták.
On approximate jumbled pattern matching in strings.
Theory Comput. Syst., 50(1):35–51, 2012.

[6] P. Burcsi, G. Fici, Zs. Lipták, F. Ruskey, and J. Sawada.
Normal, abby normal, prefix normal.
In *Proc. 7th Int. Conf. Fun with Algorithms (FUN 2014)*, volume 8496 of *LNCS*, pages 74–88, 2014.

[7] P. Burcsi, G. Fici, Zs. Lipták, F. Ruskey, and J. Sawada.
On combinatorial generation of prefix normal words.
In *Proc. 25th Ann. Symp. on Comb. Pattern Matching (CPM 2014)*, volume 8486 of *LNCS*, pages 60–69, 2014.

[8] P. Burcsi, G. Fici, Zs. Lipták, F. Ruskey, and J. Sawada.
On prefix normal words and prefix normal forms.
Theoretical Computer Science, page megjelenés alatt, 2016.