

Példaalapú programszintézis fastruktúrájú kompozíciókon

Mucsányi Bálint, Gyarmathy Bálint, Czapp Ádám

ELTE Eötvös Loránd Tudományegyetem
Informatikai Kar,
Programozáselmélet és Szoftvertchnológiai Tanszék

{jlv5ae, mzobld, plg78q}@inf.elte.hu



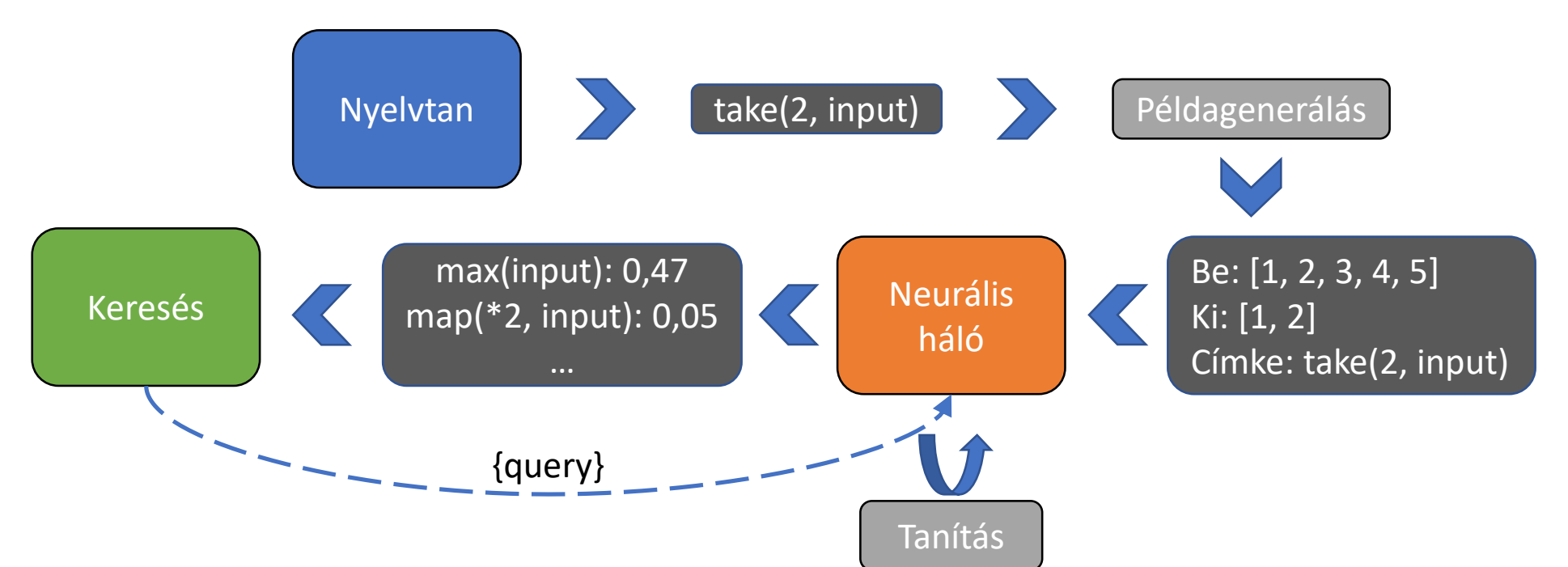
Bevezetés

Bemutatjuk a FlexCoder névre hallgató programszintézis-rendszerünket, melynek feladata függvénykompozíciók generálása, melyek a rendszernek megadott bemeneteket a szintén specifikált kimenetekre transzformálják. A bemenetek egész számokat tartalmazó listák lehetnek, a kimenetek pedig vagy egész számok, vagy szintén egész számok listái.

Célunk egy olyan rendszer megalkotása volt, amely az ismert state-of-the-art rendszerekkel ellentétben nem szab mesterséges korlátot a bemeneti listák hosszára, illetve melyben egy jóval flexibilisebb környezetfüggetlen nyelvtanon alapulnak a generálható függvénykompozíciók. Szintén fontosnak tartottuk a rendszer kiterjesztését fastruktúrájú függvénykompozíciókra is, melyek több bemeneti listát is paraméterül kaphatnak.

Komponensek

Egy általános neurális programszintézis-rendszer három fő komponensből épül fel. A megadott *nyelvtan* képi a rendszer alapját, hiszen ez szabja meg, milyen lehetséges programokat szintetizálhat a rendszer. A *keresőalgorithmus* végzi magát a programszintézist. A feltárt problémátér óriási, ezért ennek hatékony bejárásának érdekében szokásos heurisztikákat alkalmazni, melynek szerepkörét az elmúlt években sikeresen töltik be a *mély neurális háló*k. Rendszerünkben a beam search algoritmust használjuk. Az *adatgenerálás* feladata, hogy a neurális háló számára jó minőségű és nagy mennyiségű tanítópéldát generáljon, a nyelvtan által megszabott szabályok alapján.



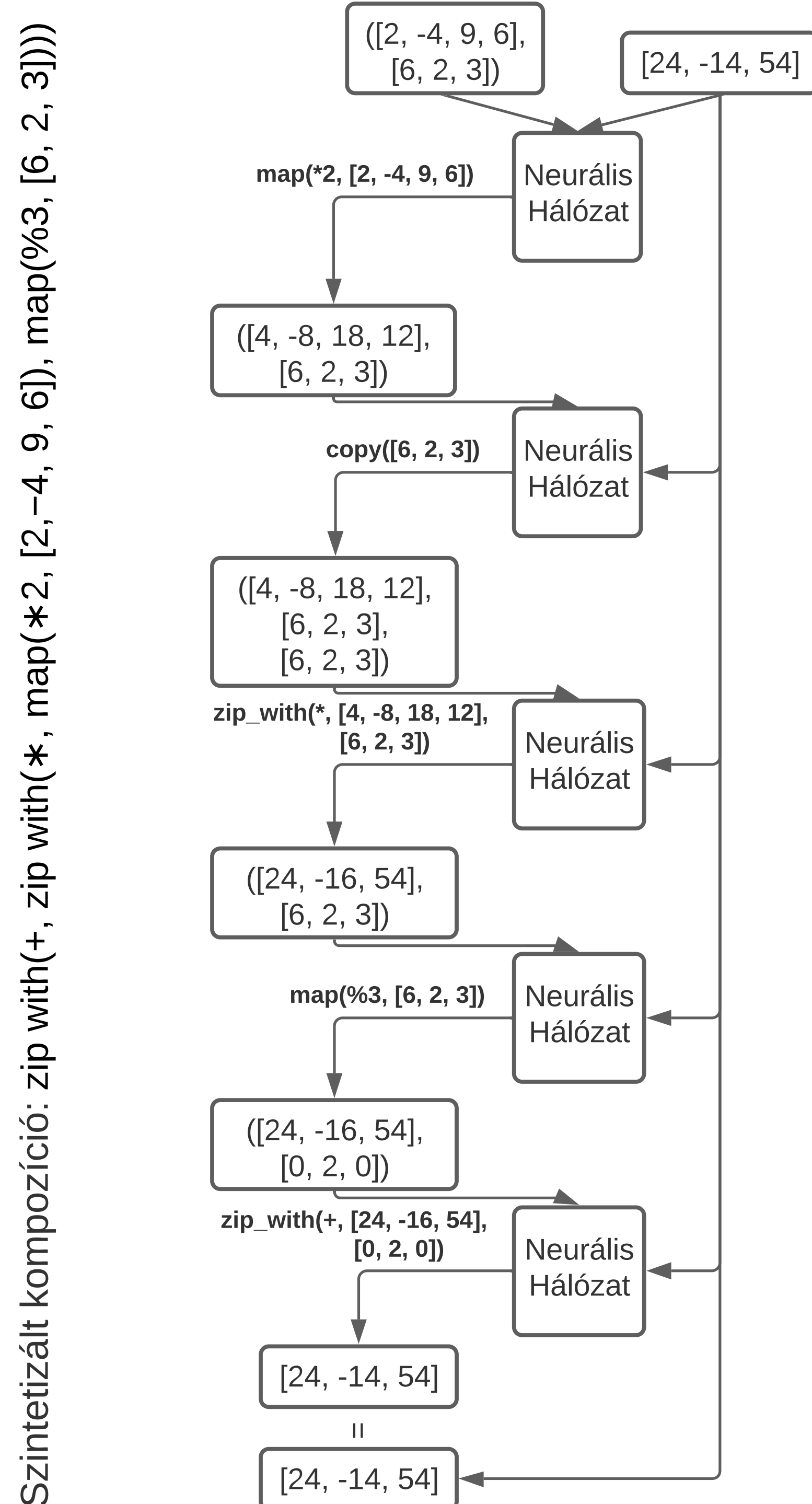
Hivatkozások

[1] Gyarmathy, B., Mucsányi, B., Czapp, Á., Szilágyi, D., & Pintér, B. (2021, January). *Flexcoder: Practical Program Synthesis with Flexible Input Lengths and Expressive Lambda Functions*. In *ICPRAM* (pp. 386-395).

<https://www.scitepress.org/Papers/2021/102378/102378.pdf>

[2] Mucsányi, B., Gyarmathy, B., Czapp, Á., & Pintér, B. *Flexible Example-Based Program Synthesis on Tree-Structured Function Compositions*. In *SN Computer Science*.

Szintetizálási folyamat

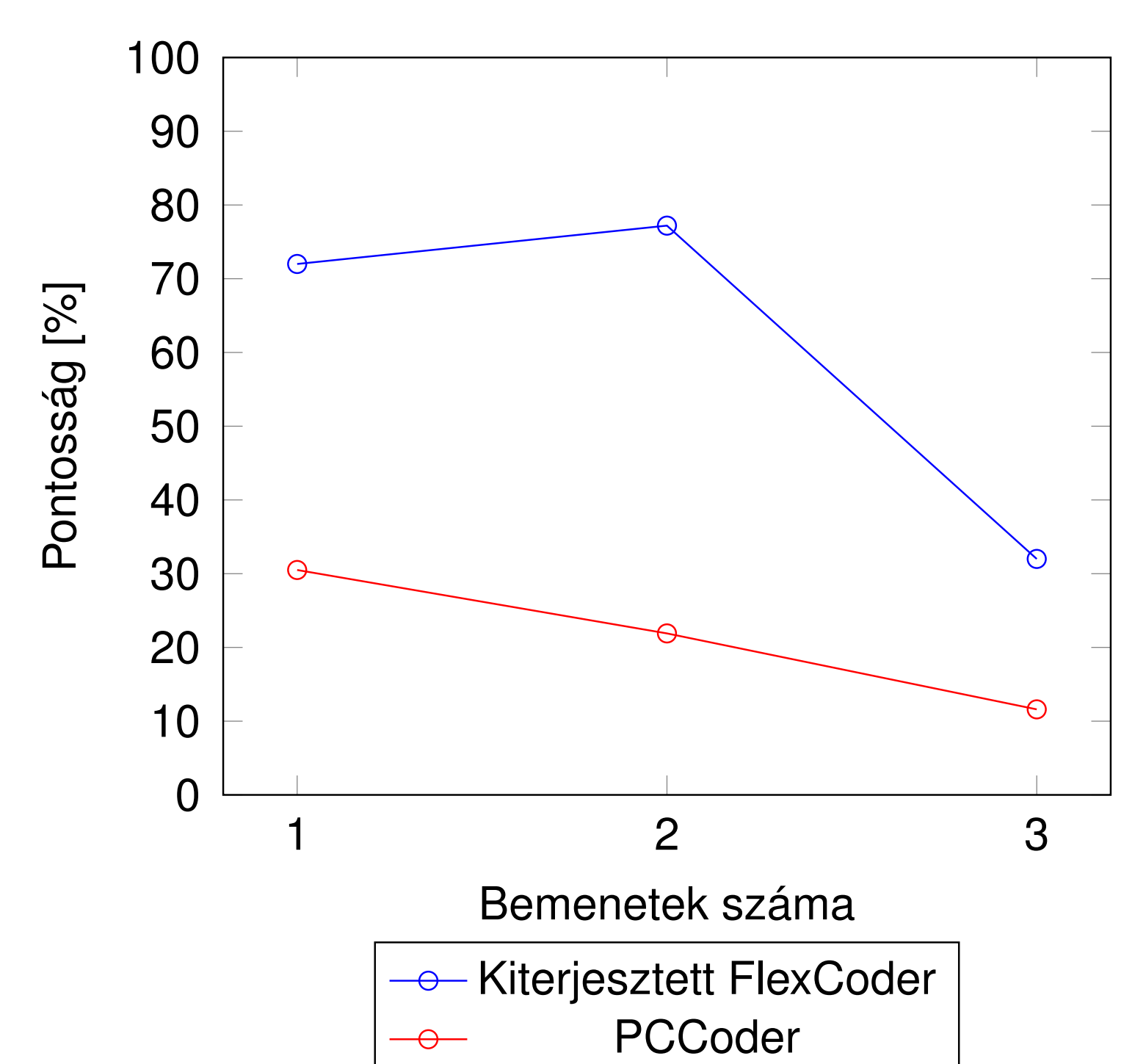
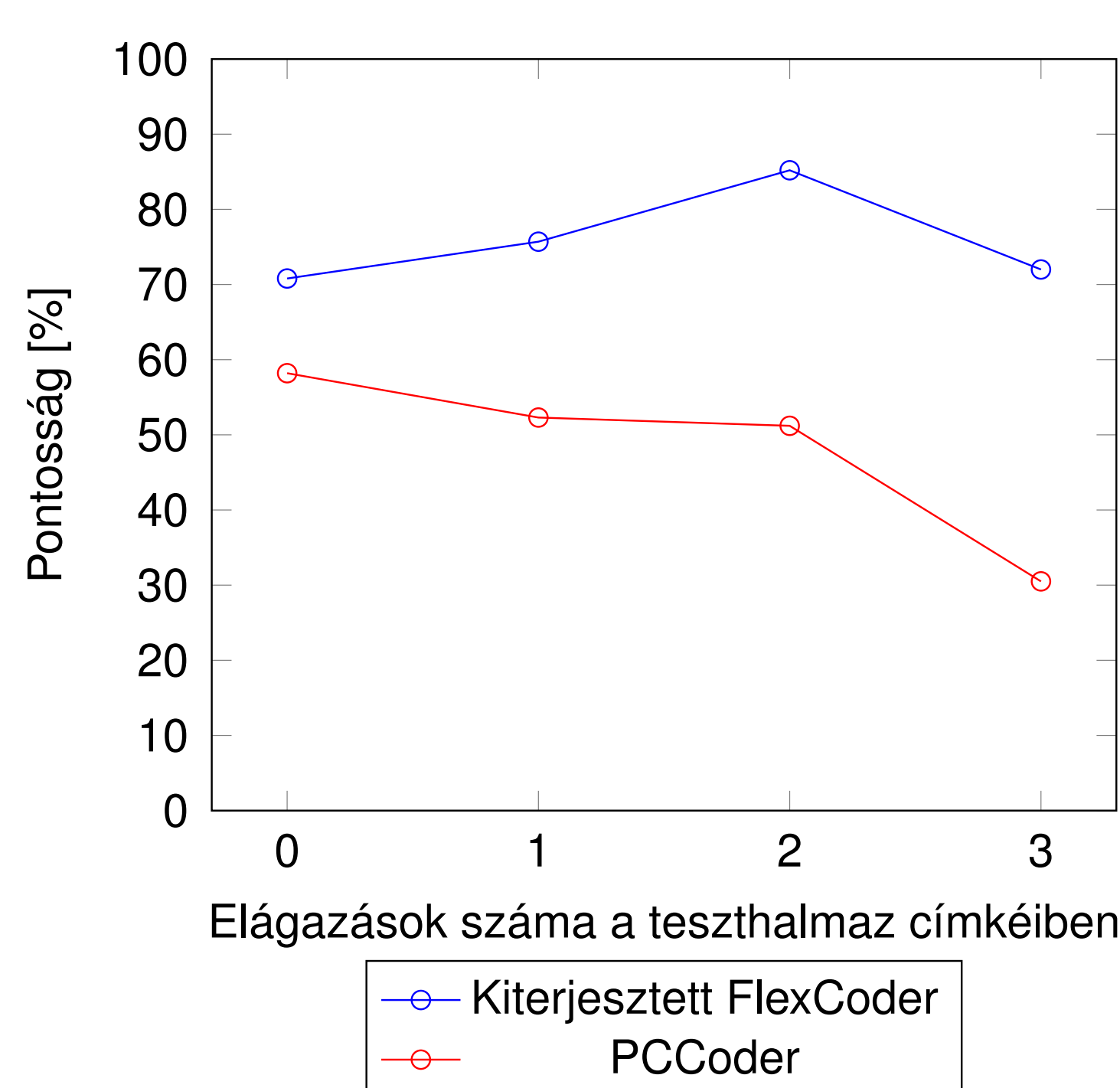


Szintetizált kompozíció: zip with(+, zip with(*, map(*2, [2, -4, 9, 6]), map(%3, [6, 2, 3]))

A bal oldali ábra a programszintézis folyamatát mutatja be. A bemenetek és az ezekhez tartozó kimenetek a feladatban megadottak. A neurális háló minden lépésben kiválaszt bemenetéből egy listát, illetve meghatároz egy felparaméterezett függvényt, amelyet a kiválasztott listára alkalmaz. Ezen iteratív algoritmus addig folyik, amíg a rendszer meg nem talál egy megoldást adó függvénykompozíciót, vagy el nem ér egy időlimitet.

Fontos hangsúlyozni, hogy ez egy ideális eset, ugyanis a neurális háló természetükből adódóan nagyon ritkán tudják csak teljes pontossággal végzni feladatukat. Ezért alkalmazzuk a beam search keresési algoritmust, amely ezt a bizonytalanságot ellensúlyozza több lehetséges megoldás egyidejű számoltatásával. A beam size határozza meg a keresés minden szintjén, hogy hány félkész függvénykompozíciót tart számon az algoritmus. A keresési fában elérhetünk egy mélységi korlátot, mely esetben a beam size-ok megduplázásával újraindul az algoritmus. Az eljárás az előző esettel megegyezően vagy a megoldás megtalálásakor, vagy az időlimit elérésekor terminál.

Eredmények



Rendszerünk lényegesen jobban alkalmazkodik a bemeneti listák hosszának változtatásához, mint a PCCoder, amely egy state-of-the-art programszintézis-rendszer. Szerintünk ez a tulajdonság kiemelten fontos a valóélet-beli alkalmazhatósághoz. A fejlesztett rendszer nyelvtana lényegesen gazdagabb, mint a PCCoder-é. A fastruktúrájú kompozíciókra kiterjesztett FlexCoder számottevően felülmúlta a PCCoder-t az általunk megalkotott tesztadathalmazokon.

A fenti ábrákon látható, hogy a rendszer jobban is skálázódik az elágazások számának növeletével, mint a PCCoder. A rendszerről írt első cikkünket [1] elfogadták a 2021-es ICPRAM konferencián és jelölték is azt a konferencia legjobb cikkének járó díjra. Felkértek minket egy bővített cikk [2] megírására is a Springer Nature Computer Science kiadványba, melyben a kiterjesztett rendszert mutattuk be.

Támogatás

EFOP-3.6.3-VEKOP-16-2017-00001: Tehetséggondozás és kutatói utánpótlás fejlesztése autonóm járműirányítási technológiák területén – A projekt a Magyar Állam és az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósul meg.