|  |  |
| --- | --- |
|  | **EÖTVÖS LORÁND TUDOMÁNYEGYETEM**  Informatikai Kar  Programozási Nyelvek és Fordítóprogramok Tanszék |

**Praktikus programozás SKI kombinátor kalkulussal**

**Témavezető: Készítette:**

Kaposi Ambrus Fülöp Olivér - PG8SJ5

egyetemi docens programtervező informatikus MSc szak

**Budapest, 2023**

# 1. Tartalomjegyzék

[1. Tartalomjegyzék 1](#_Toc140620536)

[2. Bevezetés 2](#_Toc140620537)

[3. A kutatás ismertetése 3](#_Toc140620538)

[3. 1. SKI kombinátor kalkulus 3](#_Toc140620539)

[3. 1. 1. Története 3](#_Toc140620540)

[3. 1. 2. Elemei és működése 3](#_Toc140620541)

.

# 2. Bevezetés

Diplomamunkámként egy gyakorlatban is használható, az SKI kombinátorokra épülő típusos kifejezésnyelvet készítettem, amely nem tartalmaz változókat. A nyelvhez parsert, típusellenőrzőt és egy kiértékelőt is implementáltam Java nyelven.

Az elkészült nyelvben példaként egyszerű programokat adtam meg és értékeltem ki és azt vizsgáltam, hogy egy ilyen nyelvben milyen módon lehet nagyobb programokat minél egyszerűbben és olvashatóan megadni. A kutatásban felhasznált módszerek elsősorban a típusparaméterek használata és a típuskikövetkeztetés, valamint összehasonlítottam a nyelv két különböző, mély és sekély implementációját.

# 3. A kutatás ismertetése

## 3. 1. SKI kombinátor kalkulus

### 3. 1. 1. Története

Az SKI kombinátor kalkulus az elméleti számítástudomány és matematikai logika egyik alapvető számítási módszere. Az 1920-as években Moses Schönfinkel mutatta be és később az 1930-as években Haskell Curry kiegészítette.

Történelmileg, az SKI kalkulus eredete David Hilbert munkájára vezethető vissza. Egy formális rendszert szeretett volna létrehozni matematikai bizonyítások kifejezésére, amely a lehető legkevesebb axiomát és következtetési szabályt tartalmazza. Schönfinkelnek Hilbert munkája adta az ihletet az SKI kalkulus elkészítéséhez, amellyel változók használata nélkül tudott függvényeket kifejezni. Az SKI elnevezés a három alapkombinátorból következik: S, K és I.

### 3. 1. 2. Elemei és működése

A kombinátor egy magasabb rendű függvény, amely csak függvény alkalmazásokat és korábban meghatározott kombinátorokat használ az eredmény kiszámításához. Az SKI kalkulus primitív kombinátorai az S, K és I, melyekkel az alapvető müveletek végezhetőek el. E három alapkombinátor kombinációjával és a paramétereken való alkalmazásával, komplex számításokat lehet kifejezni és kiértékelni. A fő ötlet, hogy bármely kiszámítható függvény reprezentálható a három alapkombinátorral, mivel a lambda kalkulus esetén a Church-Rosser tétel garantálja, hogy a számítások eredményei nem függenek az alkalmazott redukciós lépések sorrendjétől és, hogy a lambda kalkulus és a kombinátor logika algebrailag ekvivalens egymással, tehát bármely lambda kifejezésnek van egy ekvivalens SKI kombinátor kifejezése, és fordítva, bármely SKI kombinátor kifejezésnek van egy ekvilvalens lambda kifejezése.

Az S kombinátor függvények kompozícióját és alkalmazását fejezi ki. Három paramétert vár, az első paramétert alkalmazza a harmadikra, majd ezt alkalmazza a második és a harmadik paraméterek alkalmazásának eredményére.

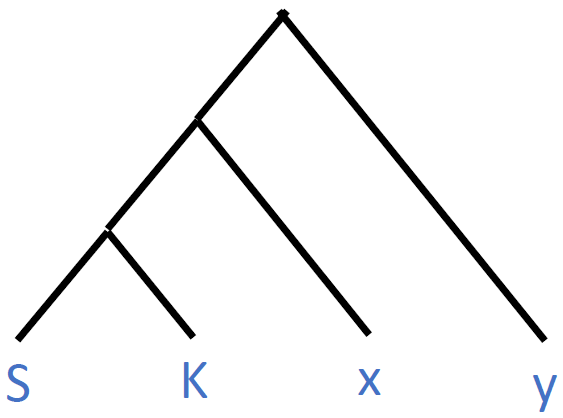
A példákban az egyes termek közötti szóközök a függvényalkalmazás műveletét jelölik.

A K kombinátor a konstans függvényként viselkedik. A két paramétere közül mindig az elsővel tér vissza, a másodikat figyelmen kívül hagyja.

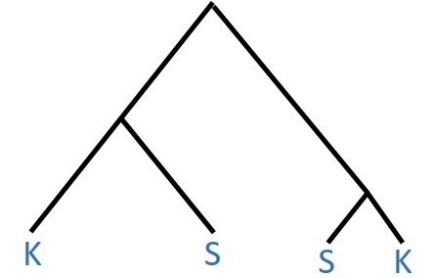
Az I kombinátor pedig az identitás függvényt reprezentálja, vagyis egyszerűen csak a kapott paraméterével tér vissza.

Az SKI kalkulus nem a minimális rendszer, ami megegyezik a lambda kalkulus számítási képességeivel, mivel az I kombinátor kifejezhető az S K K vagy S K S kifejezésekkel. I kombinátort akár el is lehetne hagyni, csak kényelemből használjuk és azért, hogy a kifejezések rövidebbek legyenek.

Az SKI kifejezések bináris fákként reprezentálhatóak, amelynek a gyökere és csúcsai az egyes függvény alkalmazások, levelei pedig a kombinátorok és paramétereik. A közös csúcsokhoz tartozó részfák az írásos formában bezárójelezhetőek, de az olvashatóság kedvéért csak abban az esetben írjuk ki a zárójeleket ha azok tényleg szükségesek. Zárójelek nélkül a kifejezések bal asszociatívak, tehát az kifejezést -ként kell értelmezni.



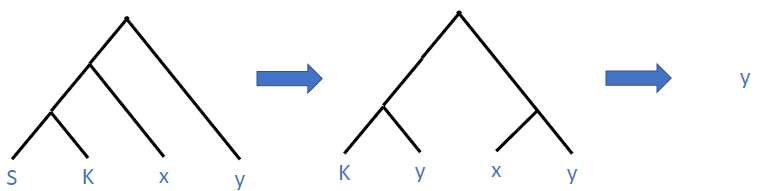
Például a helyett elég csak -t írni.

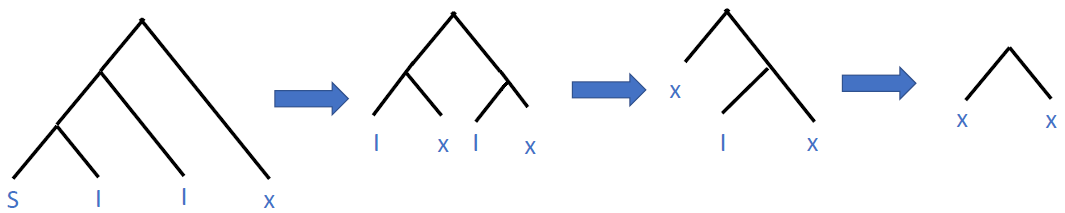


Az egyes kombinátorokon csak akkor lehet átírási szabályokat alkalmazni, ha azok a megfelelő számú paramétert kapták, S: 3 db., K: 2 db. és I: 1 db.

Például a következő kifejezések esetén nem tudunk átírási szabályt alkalmazni: vagy , de a esetben már igen: . Ha egy kombinátornak a szükségesnél több paraméter áll rendelkezésére, akkor csak a kellő számút használja fel az átírási szabály alkalmazásához, a maradék paremétereket pedig helyben hagyja.

Néhány további példa az SKI kifejezések kiértékelésére az átírási szabályok használatával:





lecture02: 8. o. példa ábrák

számokra, churck kódolás, + wiki: Typed skinál , pl termek ismertetésénél

S, K, I, előnyei, hátrányai, használata

Gyakorlati használathoz: the SKI calculus has deep connections to lambda calculus and serves as a foundation for functional programming languages.

... gpt + ambrus + practical ski

History, def , lambda, turing (Kaposi Ambrus: Type systems (course notes) -https://bitbucket.org/akaposi/typesystems/src/master/src/main.pdf )

The properties of SKI were known before any computers were built

First combinator-based programming language was APL, designed by Ken Iverson in the 1960’s

The SKI calculus has found applications in various areas, including theoretical computer science, formal logic, and functional programming. It serves as a foundational system for understanding the principles of computation and the expressive power of functional languages. The simplicity and elegance of the SKI calculus make it a fascinating topic for research and exploration, shedding light on the nature of computation and the limits of formal systems.