

# Típuselmélet 1. MintaZH

Az elméleti kérdéseket az alattuk dedikált mezőben kell megválaszolni. A gyakorlati kérdések megoldását egy `.agda` fájlban kell feltölteni a TMS rendszerbe. A gyakorlaton használt könyvtáron kívül mást nem lehet használni.

Név: \_\_\_\_\_

Neptun kód: \_\_\_\_\_

Oktató, Csoport szám: \_\_\_\_\_

## Elméleti Kérdések

1. **5 pont** Vegyük az alábbi definíciókat:

$$f : \mathbb{N} \rightarrow \mathbb{N}$$
$$f = \lambda x \rightarrow x + 1$$
$$g : (\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}$$
$$g = \lambda h \rightarrow h \ (h \ 14)$$

Normalizáld a  $g \ f$  kifejezést lépésenként! Mindegyik lépéshez írd oda, hogy milyen szabályt használtál! (Egy  $\beta$ -,  $\eta$ -szabály használata, illetve definíció szerinti átalakítás számít egy lépésnek.)

**Feladattípushoz megjegyzés:** A ZH-n nem csak lambdás kifejezések lehetnek a függvények, hanem tetszőleges megadott függvény lehet, akár mintaillesztéssel is. Minden függvény definíciója meg lesz adva itt, amelyek alapján kell átalakítani, ez eltérhet a Lib-ben látott definícióktól, mint például `_+_` vagy `_*_` esetén, ezért figyeljünk oda! Az "általános iskola" szabály csak akkor használható, ha a definíciója valamelyik alapműveletnek nincs megadva a feladatban (mint ahogy például ebben a feladatban is van). A számos konstansok természetesen a `suc`-os, `zero`-s formát jelentik, csak röviden.

2. **4 pont** Lépésenként vezesd le, hány darab eleme van az alábbi típusnak! (Egy lépésnek számít egy típuskonstruktor felbontása vagy konkrét típus átírása. Az azonos dolgok egy lépésben átírhatók.)

**Feladattípushoz megjegyzés:** Előfordulhat olyan is, hogy végtelen lesz az eredmény. Az egy lépésnek számít, hogy ezen kifejezésben amikor eljutunk oda szabályosan, akkor a két nyilat egyszerre írjuk át a hatványozásra.

$$| (\text{Bool} \rightarrow \top \uplus \text{Bool}) \times (A \rightarrow \perp) | =$$

3. Vegyük az alábbi típust:

```
data X (A : Set) : Set where
```

```
  X1 : A → X A → X A
```

```
  X2 : ℕ → X A
```

- (a) **2 pont** Írd le az  $X$  típus iterátorának a típusát!

- (b) **2 pont** Írd le az  $X$  típus  $\beta$ -szabályait!

**Feladattípushoz megjegyzés:** Előfordulhat, hogy itt koinduktív típus szerepel és a koiterátorát kell megadni a  $\beta$ -szabályokkal.

4. **4 pont** Vegyük az alábbi függvényt:

```
function :  $\mathbb{N} \rightarrow \mathbb{N}$ 
function zero = 2
function (suc n) = 3 + function n
```

Definiáld újra a függvényt a természetes számok iterátorának segítségével! A természetes számok iterátorát hívjuk `iteN`-nek.

**Feladattípushoz megjegyzés:** Más típus is állhat itt, például lista, Bool, rendezett pár, összeg típus; ezeknek is van iterátoruk, akkor értelemszerűen azok segítségével kell megadni a feladatban szereplő függvényt.

5. **3 pont** Mit jelent az, hogy egy típus szigorúan pozitív? Írj példát egy olyan típusra, amely nem szigorúan pozitív és elrontja a típuselmélet konzisztenciáját!

**Feladattípushoz megjegyzés:** Ez nagyon joker, hogy milyen kérdés van itt; kérdés mintaillesztésről, komintaillesztésről, rekurzióról, korekurzióról, stb.

## Gyakorlati Kérdések

6. **4 pont** Definíálj bijekciót a  $\text{Bool} \rightarrow \text{Bool}$  és a  $\text{Bool} \times \text{Bool}$  típusok között `bij6` néven. Ügyelj arra, hogy `fst bij6 o snd bij6 = id` és `snd bij6 o fst bij6 = id` teljesüljön!

**Feladattípushoz megjegyzés:** Biztos, hogy ilyen jellegű feladat lesz. Ekkor a két oldal elemszáma meg kell egyezzen. Ha ez nem történik meg, akkor ezen fajta feladat hibás és jelezni szükségeltetik.

7. **4 pont** Definiálj egy `TriEither` adattípust, `A`, `B` és `C` típusparaméterekkel. Három konstruktora legyen:

- `left : A → TriEither A B C`
- `middle : B → TriEither A B C` és
- `right : C → TriEither A B C`

Definiáld a típus iterátorát mintaillesztéssel és (ha szükséges) rekurzióval!

**Feladattípushoz megjegyzés:** Itt lehet induktív vagy koinduktív típus is.

8. **4 pont** Definiálj egy `f8` nevű függvényt amely a  $f(x) = 1 + x^2$  függvényt számolja ki rekurzívan! Az implementáció során használd indukciós hipotézist! Összeadás és szorzás műveletek használata megengedett.

**Feladattípushoz megjegyzés:** Itt természetes számot kell feldolgozni, de az eredmény bármilyen induktív típus lehet (lista, rendezett pár, természetes szám, stb).

9. **4 pont** Implementálj egy `concat` függvényt listákra, amely egy listák listáját várja paraméterül és egy listát ad vissza, amely tartalmazza az összes allistát összekonkatenálva.

**Feladattípushoz megjegyzés:** Itt bármilyen induktív feladat előfordulhat, paraméter, eredmény induktív értékekkel vegyítve.

10. **4 pont** Definiáld az `intersperse` függvényt `Stream`-ek fölött, amely egy elemet beszúr minden elem közé.

**Feladattípushoz megjegyzés:** Itt bármilyen koinduktív feladat előfordulhat, paraméter, eredmény bármilyen módon vegyítve.

## Pontozótábla

[illegible]