

Funkcionális programozás 13. gyakorlat

A feladat összefoglaló leírása

A feladatban az akasztófajátékot fogjuk megvalósítani. A játék során a játékos kap egy feladványt és az a cél, hogy különböző betűk tippelésével eljusson a megfejtéshez a megadott hibahatáron belül.

A játék szabályai a következők:

- Kezdetben csak a feladvány hossza és tagoltsága látható.
- Minden lépésben csak egy betűt lehet tippelni.
- A helyesen megtippelt betűk a feladványban a megfelelő pozíciókon láthatóvá válnak.
- A rosszul tippelt betűk feljegyzésre kerülnek.

A játéknak vége szakad, ha:

- a játékosnak sikerült kitalálnia a feladványt (azaz minden feladványban szereplő betűt megadott) és legfeljebb öt rossz tippje volt, vagy
- a játékos ötnél többször tippelt rosszul.

A feladványhoz megadunk egy ábécét. A feladvány szövege csak a megadott ábécé elemeiből állhat.

```
type ABC = [Char]
```

```
abc :: ABC  
abc = ['A'..'Z']
```

1. Játék állapota

A játék aktuális állapotát (**State**) az eredeti feladvány szövege (**Riddle**), a helyesen tippelt betűk (**RightGuesses**) és a rosszul tippelt betűk (**WrongGuesses**) rendezett hármasa adja meg.

Az állapot mindhárom elemének közös invariáns tulajdonsága, hogy csak nagybetűket tartalmazhatnak! Ezt a függvények helyes megadásával fogjuk kikényszeríteni.

```
type Riddle      = String
```

```
type RightGuesses = [Char]
type WrongGuesses = [Char]
```

Definiáld a játék állapotát leíró adatszerkezetet (**State**)! Legyen egy **St** adatkonstruktor! A teszteléshez szükséges lesz az egyenlőségvizsgálat az állapotokon.

2. Megfelelő-e a megadott betű?

A tippelt betűk egyaránt lehetnek kis- és nagybetűk, valamint egyéb karakterek is. Definiáld egy függvényt, amely kis- és nagybetűkre való tekintet nélkül megvizsgálja, hogy az megtalálható-e a paraméterül kapott ábécében!

```
isValidLetter 'a' abc
isValidLetter 'X' abc
not $ isValidLetter ' ' abc
isValidLetter '$' "]-$><"
isValidLetter 'E' ['a'..'z']
```

2. A kezdeti állapot

Definiáld egy függvényt, mely inicializálja a kezdeti állapotot! Amennyiben a feladványban csak az adott ábécében is szereplő elemek és szóköz szerepel, a függvény állítsa elő a kezdeti állapotot leíró rendezett hármast! A rendezett hármast első tagja a feladvány, második és harmadik tagja pedig az eddig jól és rosszul tippelt betűket tárolja.

```
startState abc "" == Just (St [] [] [])
startState abc "SOS" == Just (St "SOS" [] [])
startState abc "Save Our Souls" == Just (St "SAVE OUR SOULS" [] [])
startState abc "Save Our Souls!" == Nothing
```

3. Betűk tippelése

Definiáld a `guessLetter` függvényt, amely egy tippelt betű alapján módosítja a játék állapotát!

Ekkor a következő eseteket különböztetjük meg:

- Ha a tippelt betű (függetlenül attól, hogy az kis- vagy nagybetű) nem szerepel a megadott ábécében, az eredmény **Nothing**.
- Ha a tippelt betű már szerepel a korábban tippelt betűk között, ne változtassunk az állapoton!
- Ha a tippelt betű szerepel a szóban, szűrjük be a betűt a jól tippelt betűk listájának elejére!
- Ha a tippelt betű nem szerepel a szóban, szűrjük be a betűt a rossz tippek listájának elejére!

```
guessLetter :: ABC -> Char -> State -> Maybe State
```

```
guessLetter abc 'a' (St "SAVE OUR SOULS" [] []) == Just (St "SAVE OUR SOULS" "A" [])
guessLetter abc 'A' (St "SAVE OUR SOULS" [] []) == Just (St "SAVE OUR SOULS" "A" [])
```

```

guessLetter abc 'k' (St "SAVE OUT SOULS" [] []) == Just (St "SAVE OUR SOULS" [] "K")
guessLetter abc 'a' (St "SAVE OUT SOULS" "A" []) == Just (St "SAVE OUR SOULS" "A" [])
guessLetter abc 'K' (St "SAVE OUR SOULS" [] "K") == Just (St "SAVE OUR SOULS" [] "K")
guessLetter abc 'v' (St "SAVE OUR SOULS" "A" []) == Just (St "SAVE OUR SOULS" "VA" [])
guessLetter abc 'k' (St "SAVE OUR SOULS" "VA" []) == Just (St "SAVE OUR SOULS" "VA" "K")
guessLetter abc 'á' (St "SAVE OUR SOULS" "VA" []) == Nothing

```

4. A feladvány megjelenítése

Definiálj egy `showRiddle` függvényt, amellyel a feladvány jelenlegi állapotát lehet megjeleníteni a játékos számára!

A feladványban a helyesen tippelt karakterek jelenjenek meg, a többi betű helyére a '_' karaktert illesszük! Ügyeljünk arra, hogy a feladvány tagoltsága a játékos számára is látható maradjon!

Megjegyzés: próbáld meg megoldani a `showState` függvény nélkül!

```

showRiddle (St "SAVE OUR SOULS" [] []) == "____ _"
showRiddle (St "SAVE OUR SOULS" "AL" []) == "_A_ _ _L_"
showRiddle (St "SAVE OUR SOULS" "SAL" []) == "SA_ _ _S_LS"
showRiddle (St "SAVE OUR SOULS" "SALO" []) == "SA_ _ O_ _SO_LS"
showRiddle (St "SOS" "SO" "AL") == "SOS"

```

5. Sikerült-e megfejteni a feladványt?

Definiálj egy függvényt, amely a játék egy adott állapotához megadja, hogy a játékosnak sikerült-e megfejtenie a feladványt! A feladványt akkor tekintjük megfejtettnek, ha a tippetek figyelembe véve minden betűje ismert.

Megjegyzés: Ebben az esetben még nem kell foglalkozni a rosszul tippelt betűk számával.

```

not $ isRiddleComplete (St "SOS" [] [])
isRiddleComplete (St "SOS" "SALO" [])
not $ isRiddleComplete (St "SOS" "ALO" [])
not $ isRiddleComplete (St "SOS" [] "LKHJIG")
isRiddleComplete (St "SAVE OUR SOULS" "SAVEOURL" "KZTW")

```

6. Vége van-e a játéknak?

Definiáljuk az `isGameOver` függvényt, amely eldönti, hogy a megadott állapot a játék végét jelenti-e!

A játéknak akkor van vége, ha:

- a játékos megfejtette a feladványt és legfeljebb öt betűt hibázott el, vagy
- a játékos ötnél több betűt tippelt rosszul.

```
not $ isGameOver (St "SOS" [] [])
isGameOver (St "SOS" [] "LKHJIG")
isGameOver (St "SAVE OUR SOULS" "SAVEOURL" "KZTW")
```

7. A játék lejátszása

Definiálj egy `play` függvényt, amely egy állapot és egy karakterlista segítségével lejátszsa a játékot! A lejátszásnak vége szakad, ha:

- a játék végét jelentő feltételek valamelyike teljesül, vagy
- ha a játéknak még nincs vége, de a megadott karakter sorozat elfogyott.

```
play abc "SAkZVETWOURL" (startState abc "Save Our Souls") == St ("SAVE OUR SOULS" "LRUOEVAS" "WTZK")
play abc "PjiSAkZVETWOURL" (startState abc "Save Our Souls") == St ("SAVE OUR SOULS" "EVAS" "TZKIJP")
play abc "WOUrL" (startState abc "Save Our Souls") == St ("SAVE OUR SOULS" "LRUO" "W")
play abc "ESAKIVTLZVSETSWVOUYRL" (startState abc "Save Our Souls") == St ("SAVE OUR SOULS" "UOLVASE" "YWZTIK")
play abc "EpSAKIVTLZVSETSWVOUYRL" (startState abc "Save Our Souls") == St ("SAVE OUR SOULS" "LVASE" "WZTIKP")
```