

Zadanie J: SML - Unifikacja pierwszego rzędu

Termy pierwszego rzędu będziemy reprezentowali następująco

```
type name = char list
datatype term = Fun of name * term list | Var of name
```

(term jest albo aplikacją funkcji o określonej nazwie do listy argumentów (termów), albo zmienną o określonej nazwie) W nazwach mogą wystąpić tylko znaki alfanumeryczne, tzn. takie dla których predykat Char.isAlphaNum zwraca true.

Podstawienie to funkcja, przypisująca zmiennym termy, która prawie wszędzie jest identycznością. Każde podstawienie naturalnie rozszerza się na wszystkie termy. Podstawienie nazywamy jednopunktowym jeśli nie jest identycznością dla dokładnie jednej zmiennej. Każde nietrywialne podstawienie można reprezentować jako złożenie skończonej liczby podstawień jednopunktowych – taka reprezentacja nazywa się formą trójkątną. Formy trójkątne podstawień będziemy reprezentowali następująco:

```
type substitution = (name * term) list
```

Obiekt typu substitution postaci $[\sigma_1, \sigma_2, \dots, \sigma_n]$ wyznacza podstawienie $\lambda t. \sigma_n(\dots \sigma_2(\sigma_1(t)) \dots)$.

Zaimplementuj funkcję, która dla zadanej pary termów znajdzie najbardziej ogólne podstawienie unifikujące, lub stwierdzi że termów nie da się zunifikować. Funkcja powinna mieć sygnaturę $\text{unify} : \text{term} \rightarrow \text{term} \rightarrow \text{substitution option}$. W przypadku gdy termów nie da się zunifikować funkcja powinna zwrócić NONE

W pliku .sml powinna znaleźć się definicja:

```
fun unify t1 t2 = ...
```

Wymagany czas działania: $O(n \cdot \alpha(n))$

Przykład

```
val t1 = Fun ([#"f"], [Var [#"x"], Fun ([#"g"], [Var [#"x"], Var [#"y"]])]) : term
val t2 = Fun ([#"f"], [Fun ([#"h"], [Var [#"y"]]), Var [#"z"]]) : term
val t3 = Fun ([#"f"], [Fun ([#"h"], [Var [#"z"]]), Var [#"z"]]) : term
- unify t1 t2;
val it =
  SOME
    [(#"z"], Fun ([#"g"], [Var [#"x"], Var [#"y"]]),
      ([#"x"], Fun ([#"h"], [Var [#"y"]])]) : (name * term) list option
- unify t1 t3;
val it = NONE : (name * term) list option
```