

## Zadanie na pracownię nr 2

### Eliminacja wspólnych podwyrażeń

Eliminacja wspólnych podwyrażeń (CSE) to optymalizacja, którą może przeprowadzić kompilator, gdy wyrażenie zawiera dwa równoważne podwyrażenia. Można ją zaimplementować przez transformację programu w równoważny program, w którym wspólne wyrażenie obliczane jest przy użyciu let-wyrażenia. Przykładowo, wyrażenie

$$(2 * x) + (1 + 2 * x)$$

może zostać zastąpione wyrażeniem

```
let v0 = 2 * x in v0 + (1 + v0)
```

a wyrażenie

```
if x + y > 0 then x + y else 0
```

wyrażeniem

```
let v0 = x + y in if v0 > 0 then v0 else 0
```

W tym zadaniu trzeba zaimplementować eliminację wspólnych podwyrażeń dla języka LET.

### Zadanie

Dwa podwyrażenia  $e_1$  i  $e_2$  wyrażenia  $e$  z języka LET uznajemy na potrzeby tego zadania za równoważne, jeśli następujące dwa warunki są spełnione:

1.  $e_1$  i  $e_2$  są  $\alpha$ -równoważne,
2. Wolne wystąpienia zmiennych w wyrażeniach  $e_1$  i  $e_2$  są wolne także w  $e$ .

Przykładowo, w wyrażeniu

```
(let x = 1 in x + z) + (10 * (let y = 1 in y + z))
```

wyrażenia `let x = 1 in x + z` oraz `let y = 1 in y + z` są równoważne, a w wyrażeniu

`x * 10 + (let x = 3 in x * 10)`

dwa wystąpienia podwyrażenie `x * 10` nie są równoważne. W wyrażeniu

`let y = 3 in y * 10 + y * 10`

dwa wystąpienia wyrażenia `y * 10` nie są równoważne, ale w wyrażeniu

`y * 10 + y * 10`

dwa wystąpienia wyrażenia `y * 10` są równoważne.

Wyrażenie jest *nietrywialne*, jeśli w drzewie składni abstrakcyjnej ma przynajmniej jeden węzeł niebędący liściem.

CSE to transformacja, która zamienia wyrażenie `e`, w którym znajdują się dwa równoważne nietrywialne podwyrażenia `e1` i `e2`, na wyrażenie

`let v = e3 in e'`

gdzie:

- `v` to świeża zmienna (nie występująca w `e`),
- `e3` to dowolne wyrażenie  $\alpha$ -równoważne `e1` i `e2`,
- `e'` to wyrażenie `e`, w którym wystąpienia podwyrażeń `e1` i `e2` zamieniono na wyrażenie będące zmienną `v`.

Zaimplementuj całkowitą funkcję

`cse : expr -> expr option`

taką, że:

- Jeśli `cse e = Some e'`, to `e'` jest wynikiem jednokrotnego zastosowania CSE na wyrażeniu `e` lub którymkolwiek z jego podwyrażeń.
- Jeśli `cse e = None`, oznacza to, że ani `e`, ani żadne z jego podwyrażeń nie zawierają dwóch równoważnych podwyrażeń.

W szczególności:

- Jeśli w wyrażeniu `e` znajdują się trzy lub więcej wzajemnie równoważnych podwyrażeń, w jednym kroku można uprościć je wszystkie,
- Jeśli w `e` znajduje się więcej niż jeden zestaw równoważnych podwyrażeń, funkcja `cse` powinna wyeliminować tylko jeden z nich, bez znaczenia który.
- Wyrażenie `e` może zawierać zmienne wolne.

## Przykłady

Przekształcenie wyrażenia

```
if x then z * 10 + z * 10 else 0
```

w wyrażenie

```
let v = z * 10 in if x then v + v else 0
```

z punktu widzenia wydajności działania programu niekoniecznie jest dobrym pomysłem, ale z punktu widzenia zadania jest to poprawne zastosowanie CSE. Transformacja w

```
if x then let v = z * 10 in v + v else 0
```

jest również poprawnym rozwiązaniem.