

# Compositional logics project

Task: Довести лему, що множина ключів результату злиття є об'єднанням множин ключів аргументів.

There are several fundamental ways in Isabelle to write "Dict" type, the first is recursion

```
datatype ('key, 'value) Map =  
  Empty  
| Entry 'key 'value "('key, 'value) Map"
```

And the second is using set theory. We will use "set theory" approach.

Let's declare the type for dictionary:

```
type_synonym ('key, 'value) Dict = "('key * 'value) set"
```

The dict(map) is created from basic type "set" which provides the set of unique pairs.

To prove that set of keys after merge is equal to merged set of keys from two maps we need to define two functions:

- Keys - get keys of dictionary
- Merge - to merge two maps

```
fun keys :: "('key, 'value) Dict ⇒ 'key set" where  
  "keys d = {k. ∃v. (k, v) ∈ d}"  
  
fun merge :: "('key, 'value) Dict ⇒ ('key, 'value) Dict ⇒ ('key, 'value) Dict" where  
  "merge d1 d2 = d1 ∪ {kv ∈ d2. fst kv ∉ keys d1}"
```

The keys function accepts Dict and returns the set of first elements from Dict pairs. The merge function accepts two Dicts and performs disjunction of first Dict and the elements from the second Dict, where the first element(key) is not presented in keys of Dict1.

The automatic mathematic induction used to prove the task lemma:

```
lemma merge_keys_disjunction: "keys (merge d1 d2) = keys d2 ∪ keys d1"  
  apply auto  
  done
```