Homework 3

Functional Programming (ITI0212)

Deadline: Thursday 14th May 2026

Task 1

A decidable proposition is one which is definitively true or false: it is a proposition equipped with the extra data of a concrete witness or counterexample, either `h : p` or `h : ¬ p`.

```
inductive Decidable (p : Prop) where
  | isFalse (h : ¬ p) : Decidable p
  | isTrue (h : p) : Decidable p
```

This definition is part of the Lean standard library.

Prove that

- If propositions `p` and `q` are `Decidable`, then so is their conjunction `p ∧ q`
- If propositions `p` and `q` are `Decidable`, then so is their disjunction `p ∨ q`.
- If a proposition `p` is `Decidable` then so is its negation `¬ p`.
- If a proposition `p` is `Decidable` then `p ∨ ¬ p` is true.

Task 2

Write functions,

`sum_to_sigma : (α ⊕ β) → (Σ (b : Bool), (if b then α else β))`

`sigma_to_sum : (Σ (b : Bool), (if b then α else β)) → (α ⊕ β),`

that are mutually inverse, i.e. `sum_to_sigma (sigma_to_sum s) = s` and vice-versa.

Optional extra: prove that your functions are mutually inverse.

Task 3

Prove that `∀ (xs : List α), List.map id xs = xs`.

Hint: the `congrArg` function may be useful.