

hack.result

Result types.

Some times computations lead to error. The type `result.t A E` captures computations of value `A` which might result in an error `E`. The transformations `map`, `mapE` and `bimap` gives ways to transform the results type. The `collect` type gives a way to collect the results of a list of computations.

```
Definition t A E := (A + list E)%type.

Definition map {A B}(f : A -> B) {E} (ta : t A E) : t B E :=
  match ta with
  | inl a => inl (f a)
  | inr es => inr es
  end.

Definition map2 {A B C}(f : A -> B -> C) {E}(ta : t A E)(tb : t B E) : t C E :=
  match ta, tb with
  | inl a, inl b => inl (f a b)
  | inr ea, inr eb => inr (ea ++ eb)
  | inr ea, _ => inr ea
  | _, inr eb => inr eb
  end%list.

Definition collect {A E} : list (t A E) -> t (list A) E :=
  let combine {A E}(x : t A E) (y : t (list A) E) : t (list A) E :=
    match x, y with
    | inl a, inl yas => inl (a :: yas)
    | inr e, inr yerr => inr (e ++ yerr)
    | inl a, yerr => yerr
    | inr r, _ => inr r
    end%list in
  List.fold_right combine (inl []%list).

Module option.

Definition from {A E} (e : E) (oA : option A) : t A E :=
  match oA with
  | Some a => inl a
  | _ => inr [e]
  end%list.

Definition app {u v}(f : u -> option v) : u -> t v u
:= fun u => from u (f u).

End option.
```

[Index](#)

This page has been generated by [coqdoc](#)