v2 · Latest

Copy Publish

```
;; ACL2 Translation of Specware Workflow Specification
;; Using defcoproduct for polymorphic types and category theory constructs

(in-package "ACL2")

;; Include necessary books for polymorphic types and sum types
(include-book "std/util/define" :dir :system)
(include-book "std/util/defrule" :dir :system)
(include-book "centaur/fty/deftypes" :dir :system)
(include-book "centaur/fty/basetypes" :dir :system)
(include-book "std/util/defsum" :dir :system)
;; Note: defcoproduct depends on defsum infrastructure

;; Polymorphic list type using defcoproduct
(defcoproduct list-type (element-type)
  (empty-list)
  (cons-list (head element-type) (tail (list-type element-type))))

;; Basic list operations (polymorphic)
(define empty-list-p ((lst (list-type element-type)))
  :returns (booleanp)
  (case-match lst
    (('empty-list) t)
    (& nil)))

(define list-cons ((x element-type) (lst (list-type element-type)))
  :returns (list-type element-type)
  (cons-list x lst))

(define list-concat ((lst1 (list-type element-type))
                     (lst2 (list-type element-type)))
  :returns (list-type element-type)
  (case-match lst1
    (('empty-list) lst2)
    (('cons-list head tail)
     (list-cons head (list-concat tail lst2)))))
```