;;Domain for cleaning floor tiles

;; A domain file for CMP2020M assignment 2016/2017

;; Define the name for this domain (needs to match the domain definition

;; in the problem files)

(define (domain floor-tile)

;; We only require some typing to make this plan faster. We can do without!

(:requirements :typing)

;; We have two types: robots and the tiles, both are objects

(:types robot tile - object)

;; define all the predicates as they are used in the probem files

(:predicates

;; described what tile a robot is at

(robot-at ?r - robot ?x - tile)

;; indicates that tile ?x is above tile ?y

(up ?x - tile ?y - tile)

;; indicates that tile ?x is below tile ?y

(down ?x - tile ?y - tile)

;; indicates that tile ?x is right of tile ?y

(right ?x - tile ?y - tile)

;; indicates that tile ?x is left of tile ?y

(left ?x - tile ?y - tile)

;; indicates that a tile is clear (robot can move there)

(clear ?x - tile)

;; indicates that a tile is cleaned

(cleaned ?x - tile)

)

;; ACTIONS that need to be defined:

(:action clean-up

:parameters(?r - robot ?pos - tile ?targ - tile)

:precondition(and

(robot-at ?r ?pos )

(up ?targ ?pos)

(not(cleaned ?targ))

)

:effect(and

(cleaned ?targ)

(not(clear ?targ))

)

)

(:action clean-down

:parameters(?r - robot ?pos - tile ?targ - tile)

:precondition(and

(robot-at ?r ?pos )

(down ?targ ?pos)

(not(cleaned ?targ))

)

:effect(and

(cleaned ?targ)

(not(clear ?targ))

)

)

(:action up

:parameters(?r - robot ?from - tile ?to - tile)

:precondition(and

(robot-at ?r ?from )

(not(robot-at ?r ?to))

(up ?to ?from)

(clear ?to)

(not(cleaned ?to))

)

:effect(and

(robot-at ?r ?to)

(not(robot-at ?r ?from))

)

)

(:action down

:parameters(?r - robot ?from - tile ?to - tile)

:precondition(and

(robot-at ?r ?from )

(not(robot-at ?r ?to))

(down ?to ?from)

(clear ?to)

(not(cleaned ?to))

)

:effect(and

(robot-at ?r ?to)

(not(robot-at ?r ?from))

)

)

(:action right

:parameters(?r - robot ?from - tile ?to - tile)

:precondition(and

(robot-at ?r ?from )

(not(robot-at ?r ?to))

(right ?to ?from)

(clear ?to)

(not(cleaned ?to))

)

:effect(and

(robot-at ?r ?to)

(not(robot-at ?r ?from))

)

)

(:action left

:parameters(?r - robot ?from - tile ?to - tile)

:precondition(and

(robot-at ?r ?from )

(not(robot-at ?r ?to))

(left ?to ?from)

(clear ?to)

(not(cleaned ?to))

)

:effect(and

(robot-at ?r ?to)

(not(robot-at ?r ?from))

)

)

)