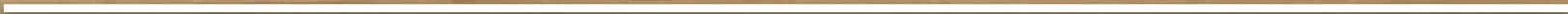# Woodworking

Bruno Ribeiro - Igor Penha - Lucas Gobbi - Rafael Nobre

01
Introdução

O que é?

Woodworking em português significa Marcenaria. Esta atividade consiste em um trabalho, podendo ser tanto artesanal quanto industrial, de criação de mobílias e outros objetivos destinados ao design de interiores.

O Domínio

O Domínio tenta simular uma marcenaria, onde a "entrada" do problema são partes ou tábuas de madeira, com o objetivo final de tratar esses materiais chegando a um tipo de superfície, tratamento, cor e tamanho desejado.

O código simula uma marcenaria por ter diversas máquinas disponíveis para uso onde cada uma delas retorna o objeto de maneira diferente, podendo mudar tamanho com a máquina de serra ou mudar a cor com a máquina de spray, dentre outros.

02 O Domínio

(:types
    acolour awood woodobj machine
    surface treatmentstatus
    aboardsize apartsize - object
    highspeed-saw glazer grinder
immersion-varnisher
    planer saw spray-varnisher - machine
    board part - woodobj)
 (:constants
        verysmooth smooth rough - surface
        varnished glazed untreated
colourfragments - treatmentstatus
        natural - acolour
        small medium large - apartsize)

Tipos:
Existem tipos específicos de máquina, tipos específicos de objetos de madeira e tipos mais gerais como cor, superfície, tratamento e tamanho dos objetos de madeira.

Constantes:
Os objetos podem variar em superfície, status do tratamento, cor e tamanho.

```
(:predicates
    (unused ?obj - part)
    (available ?obj - woodobj)
    (surface-condition ?obj - woodobj ?surface - surface)
    (treatment ?obj - part ?treatment - treatmentstatus)
    (colour ?obj - part ?colour - acolour)
    (wood ?obj - woodobj ?wood - awood)
    (boardsize ?board - board ?size - aboardsize)
    (goalsize ?part - part ?size - apartsize)
    (boardsize-successor ?size1 ?size2 - aboardsize)
    (in-highspeed-saw ?b - board ?m - highspeed-saw)
    (empty ?m - highspeed-saw)
    (has-colour ?machine - machine ?colour - acolour)
    (contains-part ?b - board ?p - part)
    (grind-treatment-change ?old ?new - treatmentstatus)
    (is-smooth ?surface - surface))
(:functions (total-cost) - number
    (spray-varnish-cost ?obj - part) - number
    (glaze-cost ?obj - part) - number
    (grind-cost ?obj - part) - number
    (plane-cost ?obj - part) - number)
```

Predicados:
Wood -> qual a madeira do objeto

has -colour -> se a máquina possui essa cor para utilizar

Grind-treatment-change -> se é possível trocar o tratamento para outro

Function:
Para determinados problemas algumas atividades podem variar de custo.

```
(:action do-immersion-varnish
    :parameters (?x - part ?m - immersion-varnisher
            ?newcolour - acolour ?surface - surface)
    :precondition (and
            (available ?x)
            (has-colour ?m ?newcolour)
            (surface-condition ?x ?surface)
            (is-smooth ?surface)
            (treatment ?x untreated))
    :effect (and
            (increase (total-cost) 10)
            (not (treatment ?x untreated))
            (treatment ?x varnished)
            (not (colour ?x natural))
            (colour ?x ?newcolour)))
```

Action do-immersion-varnish:

Essa action seria para realizar a envernização de uma parte de madeira.

```
(:action do-spray-varnish
  :parameters (?x - part ?m - spray-varnisher
        ?newcolour - acolour ?surface - surface)
  :precondition (and
      (available ?x)
      (has-colour ?m ?newcolour)
      (surface-condition ?x ?surface)
      (is-smooth ?surface)
      (treatment ?x untreated))
  :effect (and
      (increase (total-cost) (spray-varnish-cost ?x))
      (not (treatment ?x untreated))
      (treatment ?x varnished)
      (not (colour ?x natural))
      (colour ?x ?newcolour)))
```

do-spray-varnish:

Essa action seria para realizar a envernização a partir da utilização de um spray em uma parte de madeira.

```
(:action do-glaze
    :parameters (?x - part ?m - glazer
            ?newcolour - acolour)
    :precondition (and
        (available ?x)
        (has-colour ?m ?newcolour)
        (treatment ?x untreated))
    :effect (and
        (increase (total-cost) (glaze-cost ?x))
        (not (treatment ?x untreated))
        (treatment ?x glazed)
        (not (colour ?x natural))
        (colour ?x ?newcolour)))
```

do-glaze:

Essa action seria para realizar a esmaltação de uma parte de madeira.

```
(:action do-grind
    :parameters (?x - part ?m - grinder ?oldsurface - surface
            ?oldcolour - acolour
            ?oldtreatment ?newtreatment - treatmentstatus)
    :precondition (and
            (available ?x)
            (surface-condition ?x ?oldsurface)
            (is-smooth ?oldsurface)
            (colour ?x ?oldcolour)
            (treatment ?x ?oldtreatment)
            (grind-treatment-change ?oldtreatment
?newtreatment))
    :effect (and
            (increase (total-cost) (grind-cost ?x))
            (not (surface-condition ?x ?oldsurface))
            (surface-condition ?x verysmooth)
            (not (treatment ?x ?oldtreatment))
            (treatment ?x ?newtreatment)
            (not (colour ?x ?oldcolour))
            (colour ?x natural)))
```

do-grind:

Essa action seria para moer uma parte de madeira.

```
(:action do-plane
    :parameters (?x - part ?m - planer ?oldsurface - surface
            ?oldcolour - acolour ?oldtreatment - treatmentstatus)
    :precondition (and
        (available ?x)
        (surface-condition ?x ?oldsurface)
        (treatment ?x ?oldtreatment)
        (colour ?x ?oldcolour))
    :effect (and
        (increase (total-cost) (plane-cost ?x))
        (not (surface-condition ?x ?oldsurface))
        (surface-condition ?x smooth)
        (not (treatment ?x ?oldtreatment))
        (treatment ?x untreated)
        (not (colour ?x ?oldcolour))
        (colour ?x natural)))
```

do-plane:

Essa action seria para planificar uma parte de madeira.

```
(:action load-highspeed-saw
  :parameters (?b - board ?m - highspeed-saw)
  :precondition (and
      (empty ?m)
      (available ?b))
  :effect (and
      (increase (total-cost) 30)
      (not (available ?b))
      (not (empty ?m))
      (in-highspeed-saw ?b ?m)))
(:action unload-highspeed-saw
  :parameters (?b - board ?m - highspeed-saw)
  :precondition (in-highspeed-saw ?b ?m)
  :effect (and
      (increase (total-cost) 10)
      (available ?b)
      (not (in-highspeed-saw ?b ?m))
      (empty ?m)))
```

load/unload-highspeed-saw:

Action para carregar e descarregar uma serra de alta velocidade com tábuas de madeira.

```
(:action cut-board-small
    :parameters (?b - board ?p - part ?m - highspeed-saw ?w -
awood
            ?surface - surface ?size_before ?size_after -
aboardsize)
    :precondition (and
        (unused ?p)
        (goalsize ?p small)
        (in-highspeed-saw ?b ?m)
        (wood ?b ?w)
        (surface-condition ?b ?surface)
        (boardsize ?b ?size_before)
        (boardsize-successor ?size_after ?size_before))
    :effect (and
        (increase (total-cost) 10)
        (not (unused ?p))
        (available ?p)
        (wood ?p ?w)
        (surface-condition ?p ?surface)
        (colour ?p natural)
        (treatment ?p untreated)
        (boardsize ?b ?size_after)))
```

cut-board-small :

Esta ação representa o corte de uma tábua utilizando uma serra rápida

Possui variações que mudam somente o tamanho da tábua:
- Cut-board-medium
- cut-board-large

```
(:action do-saw-small
    :parameters (?b - board ?p - part ?m - saw ?w - awood
            ?surface - surface ?size_before ?size_after -
aboardsize)
    :precondition (and
        (unused ?p)
        (goalsize ?p small)
        (available ?b)
        (wood ?b ?w)
        (surface-condition ?b ?surface)
        (boardsize ?b ?size_before)
        (boardsize-successor ?size_after ?size_before))
    :effect (and
        (increase (total-cost) 30)
        (not (unused ?p))
        (available ?p)
        (wood ?p ?w)
        (surface-condition ?p ?surface)
        (colour ?p natural)
        (treatment ?p untreated)
        (boardsize ?b ?size_after)))
```

do-saw-small :

Esta ação representa o corte de uma tábua utilizando uma serra comum

Possui variações que mudam somente o tamanho da tábua:
- do-saw medium
- do-saw -large

03
O Problema

Exemplo de Problema

❖

```
(define (problem wood-prob)
 (:domain woodworking)
 (:objects
   grindero - grinder
   glazero - glazer
   immersion-varnishero - immersion-varnisher
   planero - planer
   highspeed-sawo - highspeed-saw
   spray-varnishero - spray-varnisher
   sawo - saw
   blue mauve - acolour
   beech mahogany - awood
   po p1 p2 - part
   so - aboardsize  )
```

```
(:init
   (grind-treatment-change varnished
colourfragments)
   (grind-treatment-change glazed untreated)
   (grind-treatment-change untreated
untreated)
   (grind-treatment-change colourfragments
untreated)
   (is-smooth smooth)
   (is-smooth verysmooth)
   (= (total-cost) o)
   (has-colour glazero natural)
   (has-colour immersion-varnishero blue)
   (empty highspeed-sawo)
   (has-colour spray-varnishero mauve)
```

❖

Exemplo de Problema

(available p0)
   (colour p0 blue)
   (wood p0 mahogany)
   (surface-condition p0 verysmooth)
   (treatment p0 colourfragments)
   (goalsize p0 small)
   (= (spray-varnish-cost p0) 5)
   (= (glaze-cost p0) 10)
   (= (grind-cost p0) 15)
   (= (plane-cost p0) 10)

(available p1)
   (colour p1 natural)
   (wood p1 mahogany)
   (surface-condition p1 smooth)
   (treatment p1 colourfragments)
   (goalsize p1 small)
   (= (spray-varnish-cost p1) 5)
   (= (glaze-cost p1) 10)
   (= (grind-cost p1) 15)
   (= (plane-cost p1) 10)

Exemplo de Problema

```
(available p2)
   (colour p2 mauve)
   (wood p2 beech)
   (surface-condition p2 verysmooth)
   (treatment p2 colourfragments)
   (goalsize p2 small)
   (= (spray-varnish-cost p2) 5)
   (= (glaze-cost p2) 10)
   (= (grind-cost p2) 15)
   (= (plane-cost p2) 10)
)
```

```
(:goal
   (and
      (available p0)
      (wood p0 mahogany)
      (surface-condition p0 smooth)
      (available p1)
      (surface-condition p1 smooth)
      (treatment p1 glazed)
      (available p2)
      (colour p2 natural)
      (treatment p2 glazed)
   )
)

(:metric minimize (total-cost))
```

Exemplo de Solução

(do-grind p2 grindero verysmooth mauve
colourfragments untreated)
(do-glaze p2 glazero natural)
(do-plane po planero verysmooth blue
colourfragments)
(do-grind p1 grindero smooth natural
colourfragments untreated)
(do-plane p1 planero verysmooth natural
untreated)
(do-glaze p1 glazero natural)
; cost = 70 (general cost)

04
A Competição

IPC 2011

Este domínio participou da sétima competição internacional de planejamento (IPC 2011), em que 6 planejadores competiram. Dentre eles, apenas um não foi capaz de resolver todos os problemas do domínio, o planejador SASE.

Curiosamente o planejador SASE foi o de melhor eficiência dentre os 6 da competição, conseguindo resolver 48.5% dos problemas de todos os domínios somados (570). Entretanto, para o domínio Woodworking, resolveu apenas 5 dos 30

05

Benchmark

problem1	Fast Downward		Fast Downward - fdss23		Scorpion maidu	
lama-first	0.342s	12 actions cost 235	0.411	12 actions cost 235	0.547s	12 actions cost 235
lama	8.955s	plan.1 = 12 cost 235 plan.5 = 13 cost 195	9.143	plan.1 = 12 cost 235 plan.5 = 13 cost 195	9.280s	plan.1 = 12 cost 235 plan.5 = 13 cost 195

problem20	Fast Downward		Fast Downward - fdss23		Scorpion maidu	
lama-first	0.628s	30 actions cost 685	0.646s	30 actions cost 685	0.766s	30 actions cost 685
lama	2m46.2s	plan.1 = 30 cost 685 plan.2 = 30 cost 675	2m8.2s	plan.1 = 30 cost 685 plan.2 = 30 cost 675	1m54.6s	plan.1 = 30 cost 685 plan.2 = 30 cost 675

Obrigado