Gramáticas da Forma

(implementações)

Rodrigo Correia

Faculdade de Arquitectura

Dezembro2012

PROGRAMA

Gramática Formulação

PROGRAMA + DESIGNA

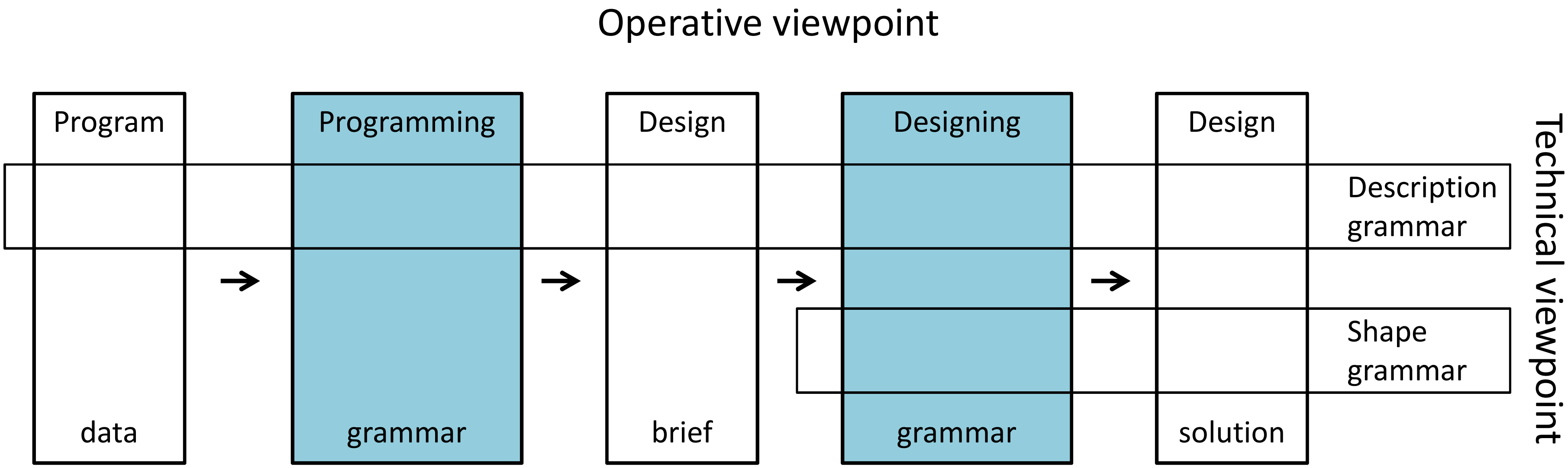
Gramática Formulação

Gramática Projecto

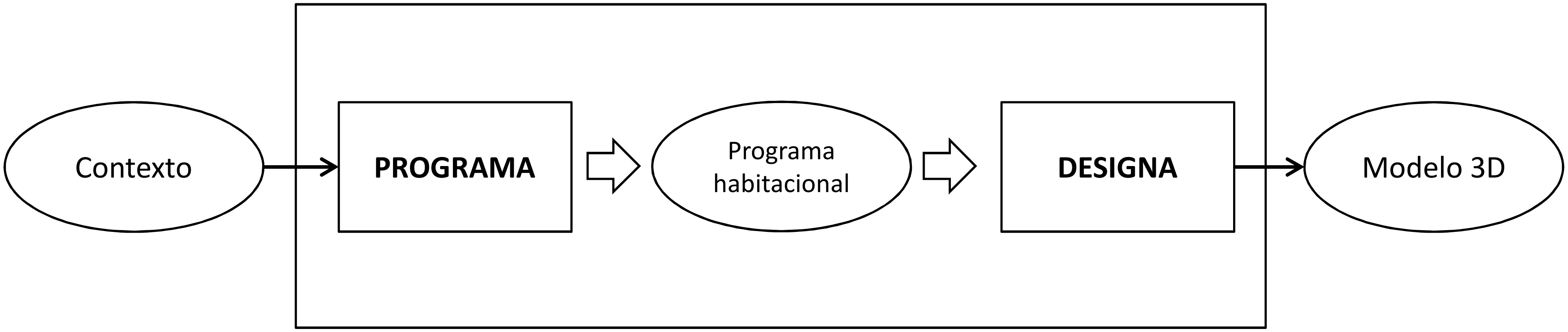
PROGRAMA + DESIGNA = MALAGUEIRA

Gramática Formulação Gramática Projecto Gramática Discursiva

Gramática Discursiva



Gramática Discursiva



Contexto

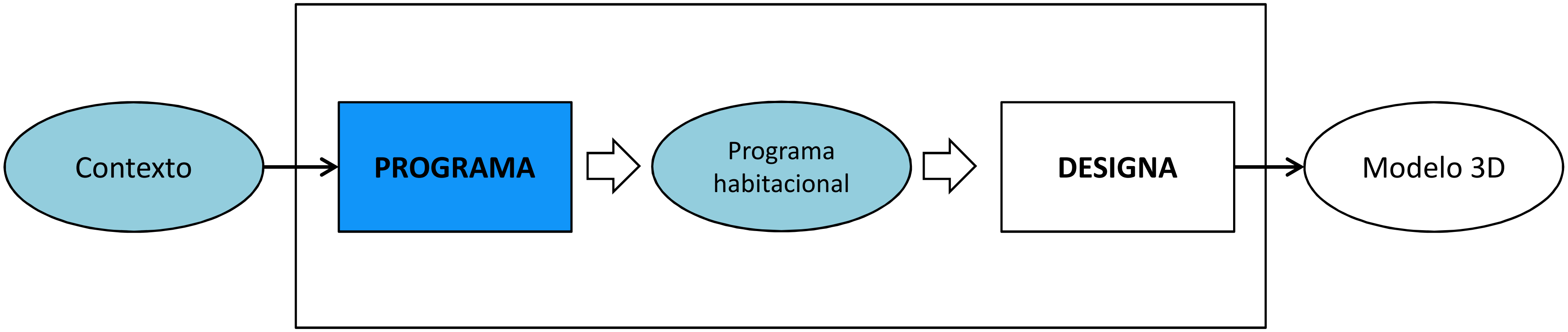
PROGRAMA

Programa habitacional

DESIGNA

Modelo 3D

Gramática Discursiva



Contexto

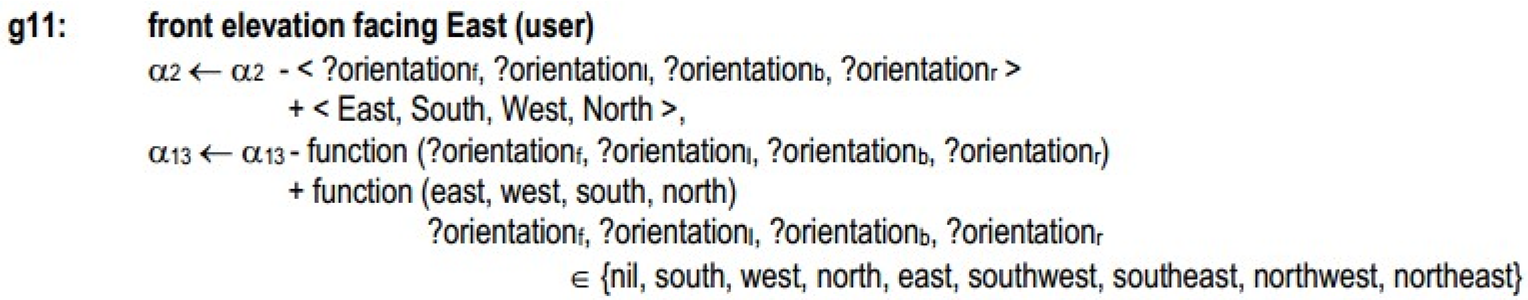
PROGRAMA

Programa habitacional

DESIGNA

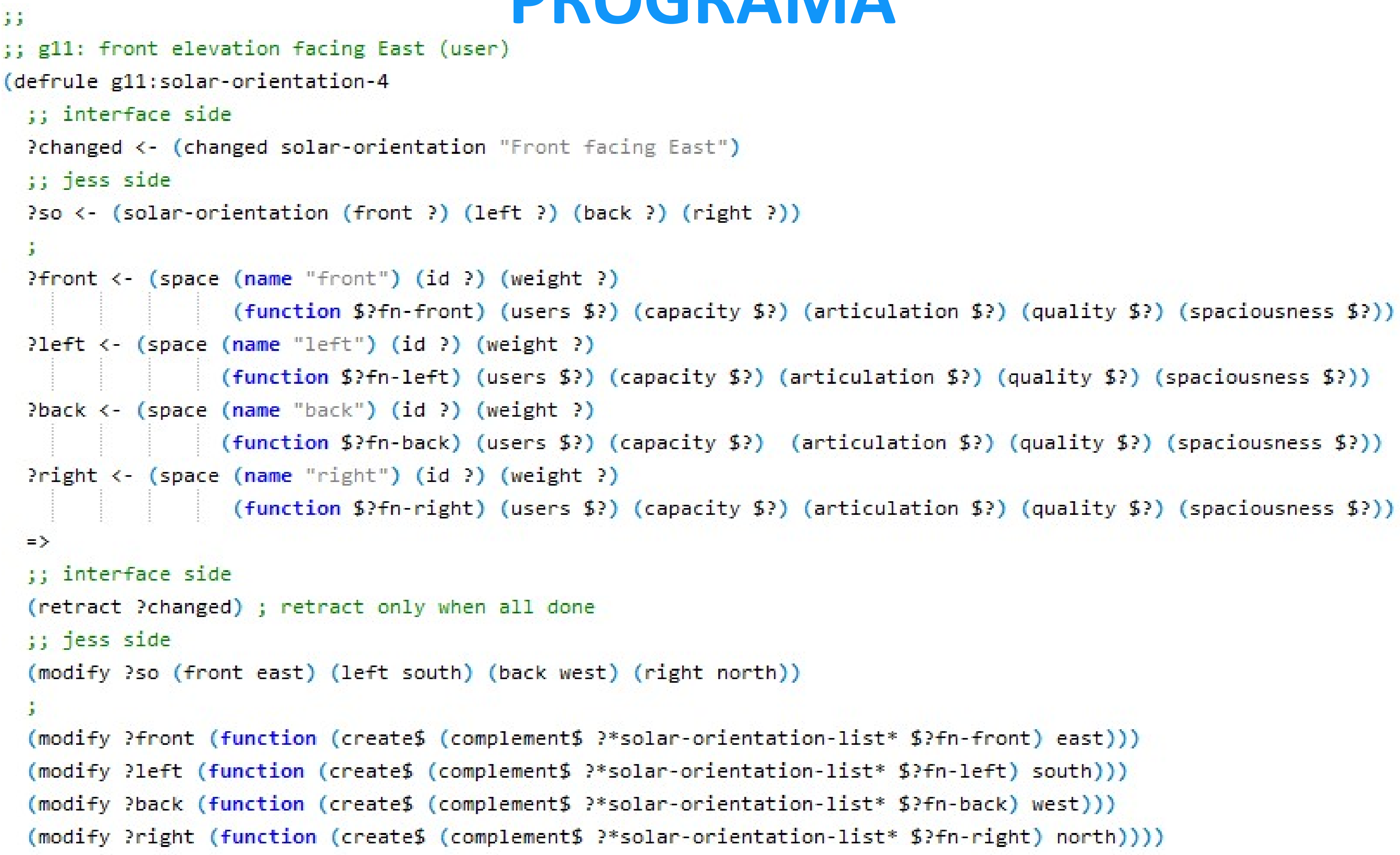
Modelo 3D

PROGRAMA



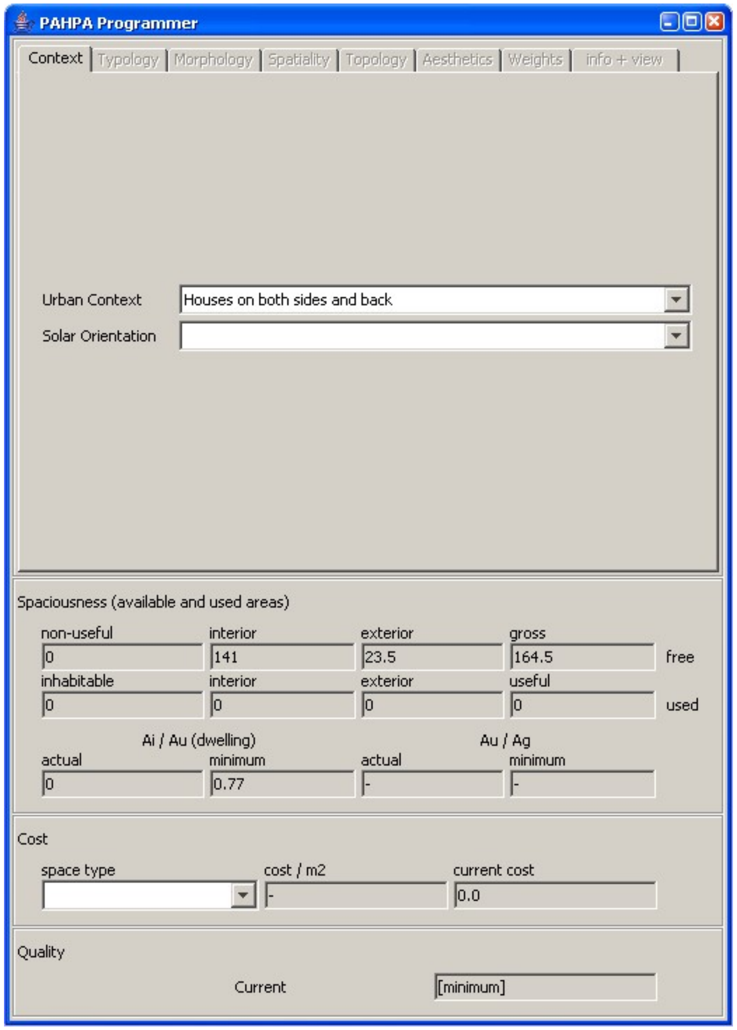
```
g11:    front elevation facing East (user)
        $\alpha_2 \leftarrow \alpha_2$ - < ?orientation$_f$, ?orientation$_l$, ?orientation$_b$, ?orientation$_r$ >
                  + < East, South, West, North >,
        $\alpha_{13} \leftarrow \alpha_{13}$ - function (?orientation$_f$, ?orientation$_l$, ?orientation$_b$, ?orientation$_r$)
                  + function (east, west, south, north)
                          ?orientation$_f$, ?orientation$_l$, ?orientation$_b$, ?orientation$_r$
                                  $\in$ {nil, south, west, north, east, southwest, southeast, northwest, northeast}
```

PROGRAMA



```
;;
;; g11: front elevation facing East (user)
(defrule g11:solar-orientation-4
  ;; interface side
  ?changed <- (changed solar-orientation "Front facing East")
  ;; jess side
  ?so <- (solar-orientation (front ?) (left ?) (back ?) (right ?))
  ;
  ?front <- (space (name "front") (id ?) (weight ?)
                   (function $?fn-front) (users $?) (capacity $?) (articulation $?) (quality $?) (spaciousness $?))
  ?left <- (space (name "left") (id ?) (weight ?)
                  (function $?fn-left) (users $?) (capacity $?) (articulation $?) (quality $?) (spaciousness $?))
  ?back <- (space (name "back") (id ?) (weight ?)
                  (function $?fn-back) (users $?) (capacity $?)  (articulation $?) (quality $?) (spaciousness $?))
  ?right <- (space (name "right") (id ?) (weight ?)
                   (function $?fn-right) (users $?) (capacity $?) (articulation $?) (quality $?) (spaciousness $?))
  =>
  ;; interface side
  (retract ?changed) ; retract only when all done
  ;; jess side
  (modify ?so (front east) (left south) (back west) (right north))
  ;
  (modify ?front (function (create$ (complement$ ?*solar-orientation-list* $?fn-front) east)))
  (modify ?left (function (create$ (complement$ ?*solar-orientation-list* $?fn-left) south)))
  (modify ?back (function (create$ (complement$ ?*solar-orientation-list* $?fn-back) west)))
  (modify ?right (function (create$ (complement$ ?*solar-orientation-list* $?fn-right) north)))
```

PROGRAMA



Urban Context: Houses on both sides and back

Solar Orientation:

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	141	23.5	164.5	free
inhabitable	interior	exterior	useful	
0	0	0	0	used

Ai / Au (dwelling)		Au / Ag	
actual	minimum	actual	minimum
0	0.77	-	-

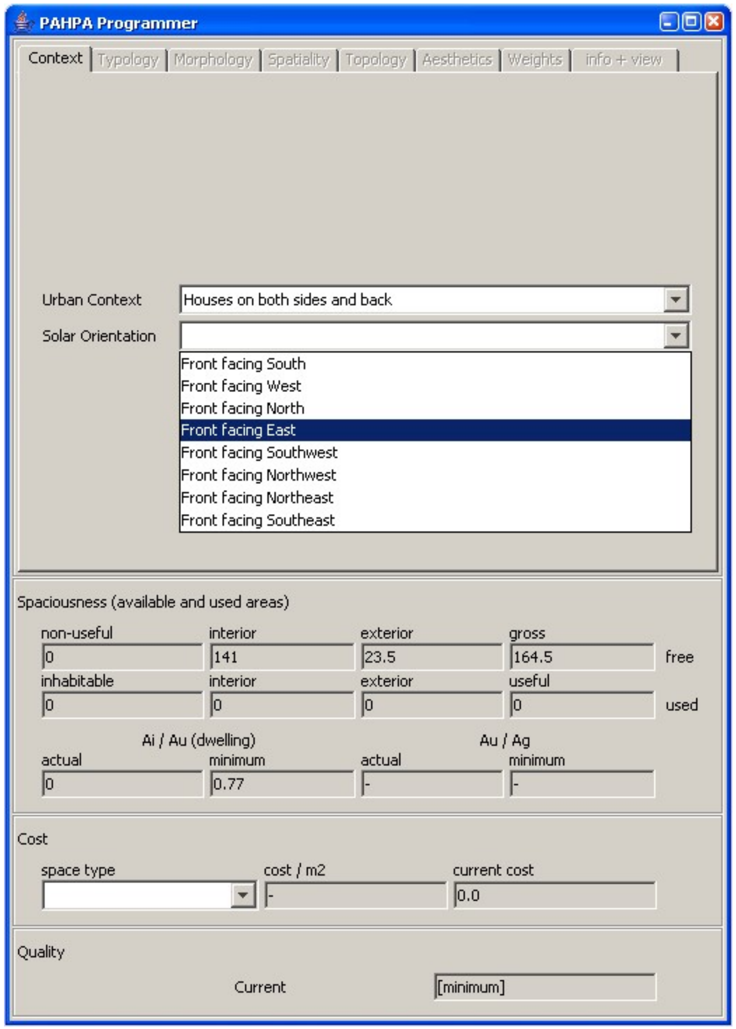
Cost

space type	cost / m2	current cost
	-	0.0

Quality

Current: [minimum]

PROGRAMA



Urban Context: Houses on both sides and back

Solar Orientation:

Front facing South
Front facing West
Front facing North
Front facing East
Front facing Southwest
Front facing Northwest
Front facing Northeast
Front facing Southeast

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	141	23.5	164.5	free
inhabitable	**interior**	**exterior**	**useful**	
0	0	0	0	used

Ai / Au (dwelling)		Au / Ag	
actual	minimum	actual	minimum
0	0.77	-	-

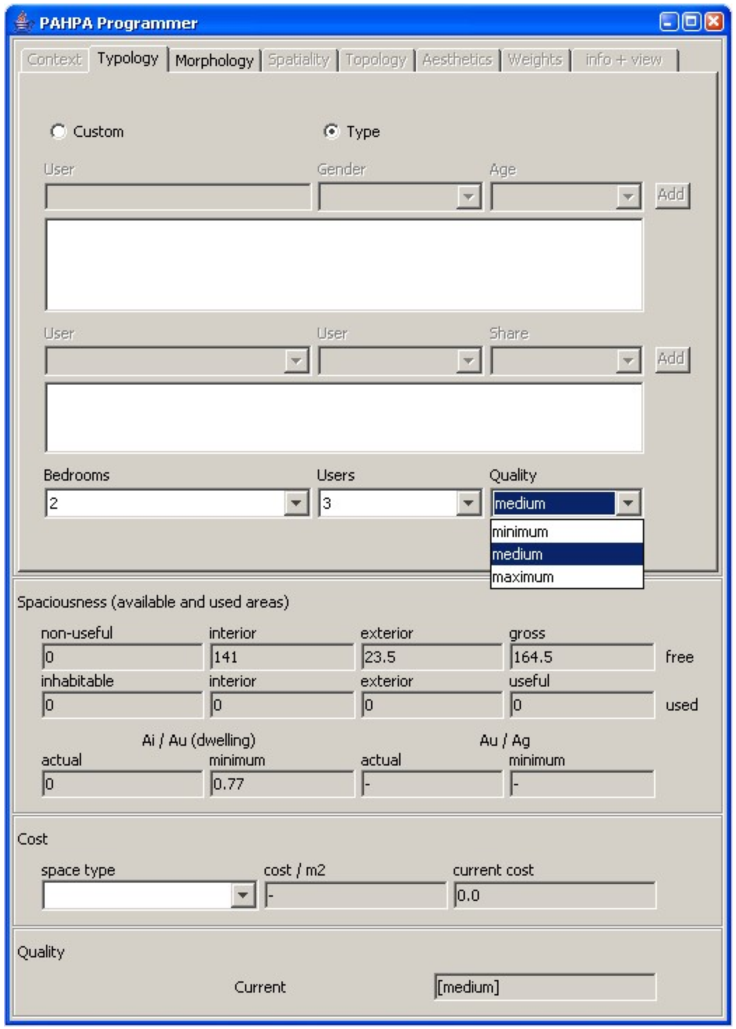
Cost

space type	cost / m2	current cost
	-	0.0

Quality

Current: [minimum]

PROGRAMA



○ Custom ● Type

User

Gender

Age

User

User

Share

Bedrooms: 2

Users: 3

Quality: medium

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	141	23.5	164.5	free
inhabitable	interior	exterior	useful	
0	0	0	0	used

Ai / Au (dwelling)		Au / Ag	
actual	minimum	actual	minimum
0	0.77	-	-

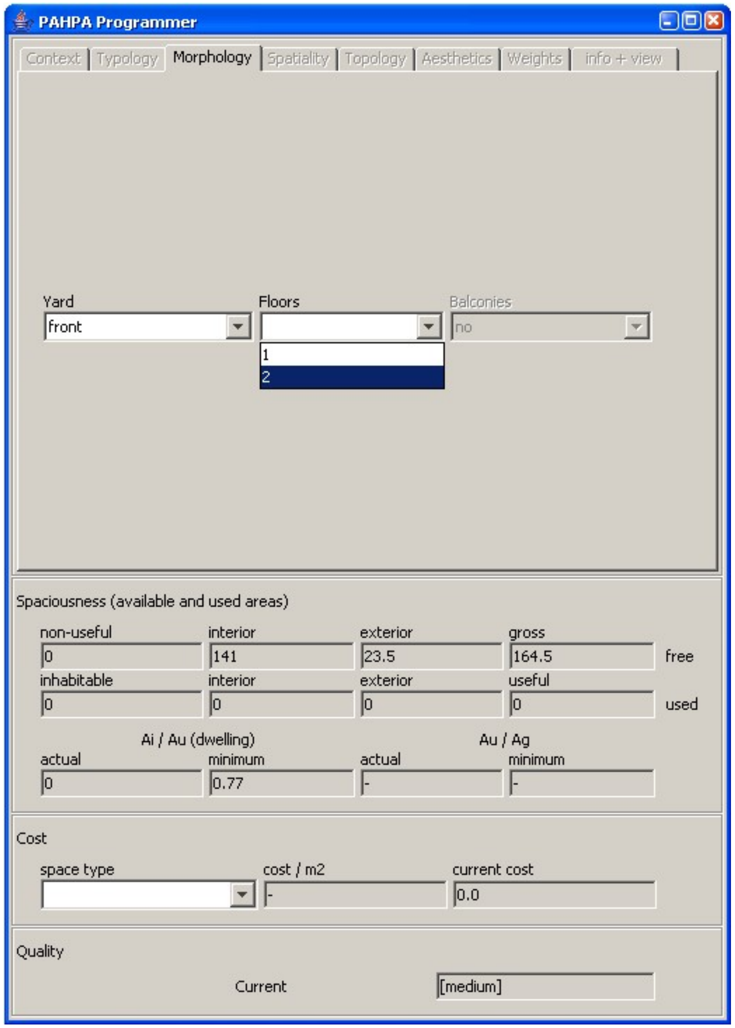
Cost

space type	cost / m2	current cost
	-	0.0

Quality

Current: [medium]

PROGRAMA



Yard: front

Floors: 1, 2

Balconies: no

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	141	23.5	164.5	free
inhabitable	interior	exterior	useful	
0	0	0	0	used

Ai / Au (dwelling)		Au / Ag	
actual	minimum	actual	minimum
0	0.77	-	-

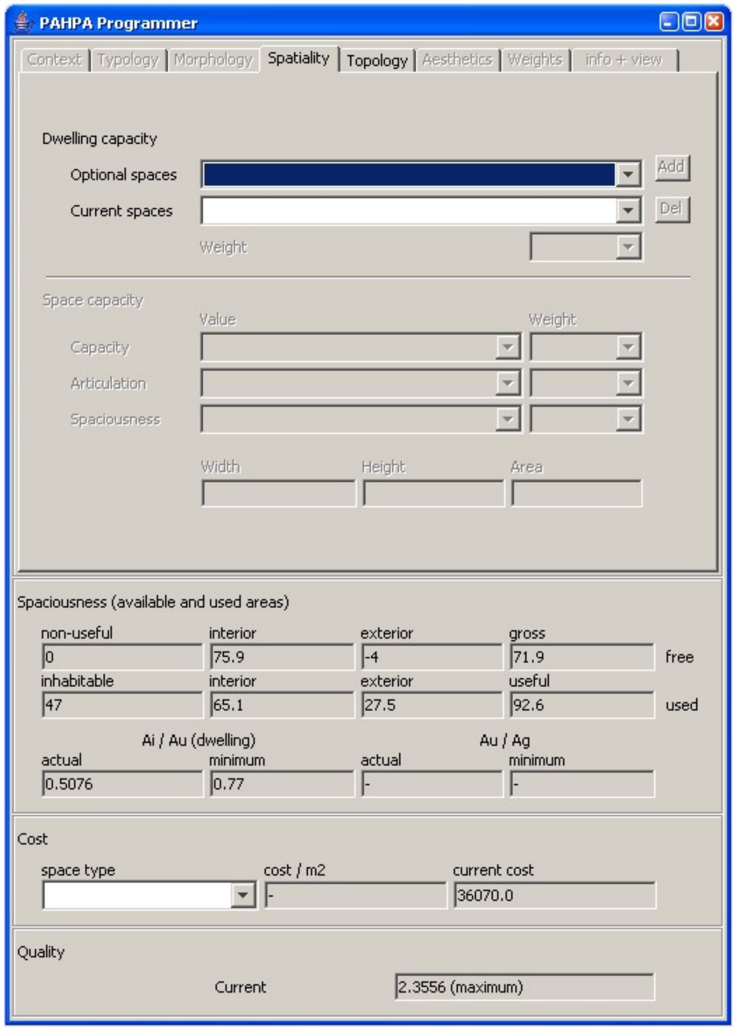
Cost

space type	cost / m2	current cost
	-	0.0

Quality

Current: [medium]

PROGRAMA



PAHPA Programmer
Context | Typology | Morphology | Spatiality | Topology | Aesthetics | Weights | info + view
Add
Del

Dwelling capacity

Optional spaces

Current spaces

Weight

Space capacity

	Value	Weight
Capacity		
Articulation		
Spaciousness		

Width	Height	Area

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	75.9	-4	71.9	free
inhabitable	interior	exterior	useful	
47	65.1	27.5	92.6	used

Ai / Au (dwelling)		Au / Ag	
actual	minimum	actual	minimum
0.5076	0.77	-	-

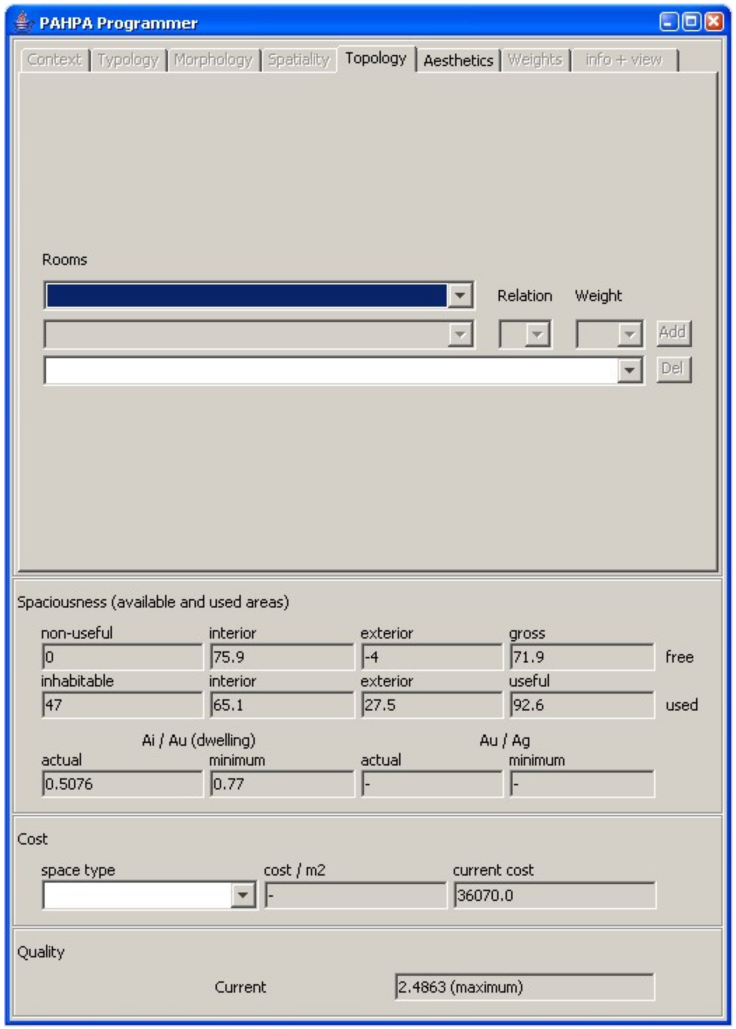
Cost

space type	cost / m2	current cost
	-	36070.0

Quality

Current 2.3556 (maximum)

PROGRAMA



Rooms

Relation

Weight

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	75.9	-4	71.9	free
inhabitable	interior	exterior	useful	
47	65.1	27.5	92.6	used

Ai / Au (dwelling)

Au / Ag

actual	minimum	actual	minimum
0.5076	0.77	-	-

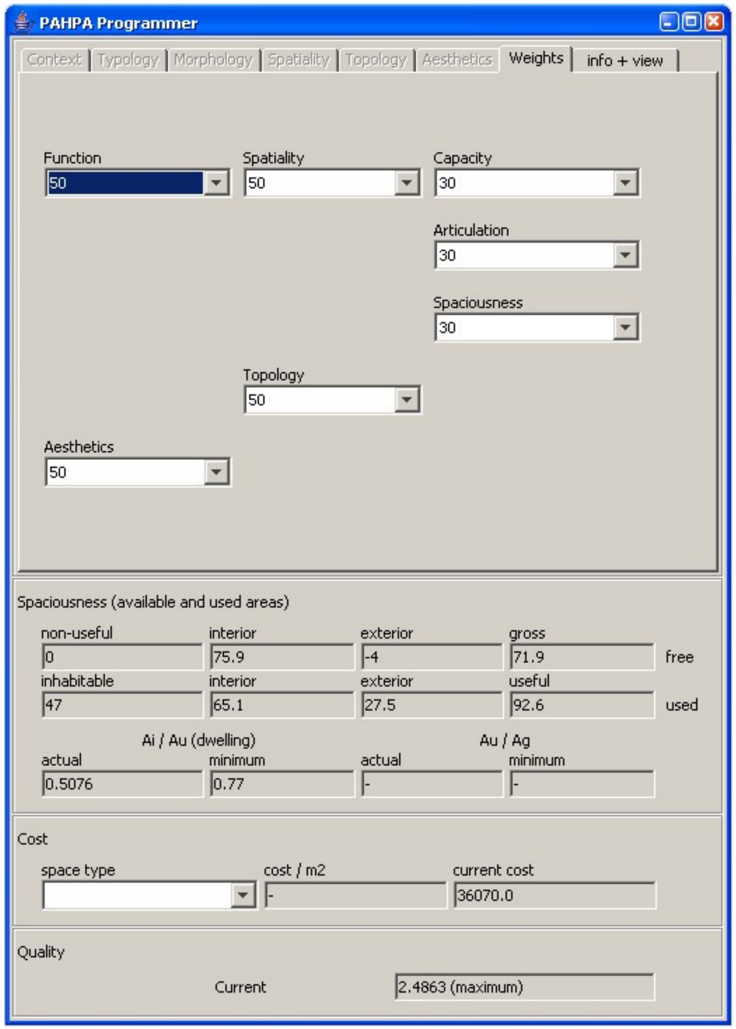
Cost

space type	cost / m2	current cost
	-	36070.0

Quality

Current 2.4863 (maximum)

PROGRAMA



Function: 50
Spatiality: 50
Capacity: 30
Articulation: 30
Spaciousness: 30
Topology: 50
Aesthetics: 50

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	75.9	-4	71.9	free
inhabitable	interior	exterior	useful	
47	65.1	27.5	92.6	used

Ai / Au (dwelling)		Au / Ag	
actual	minimum	actual	minimum
0.5076	0.77	-	-

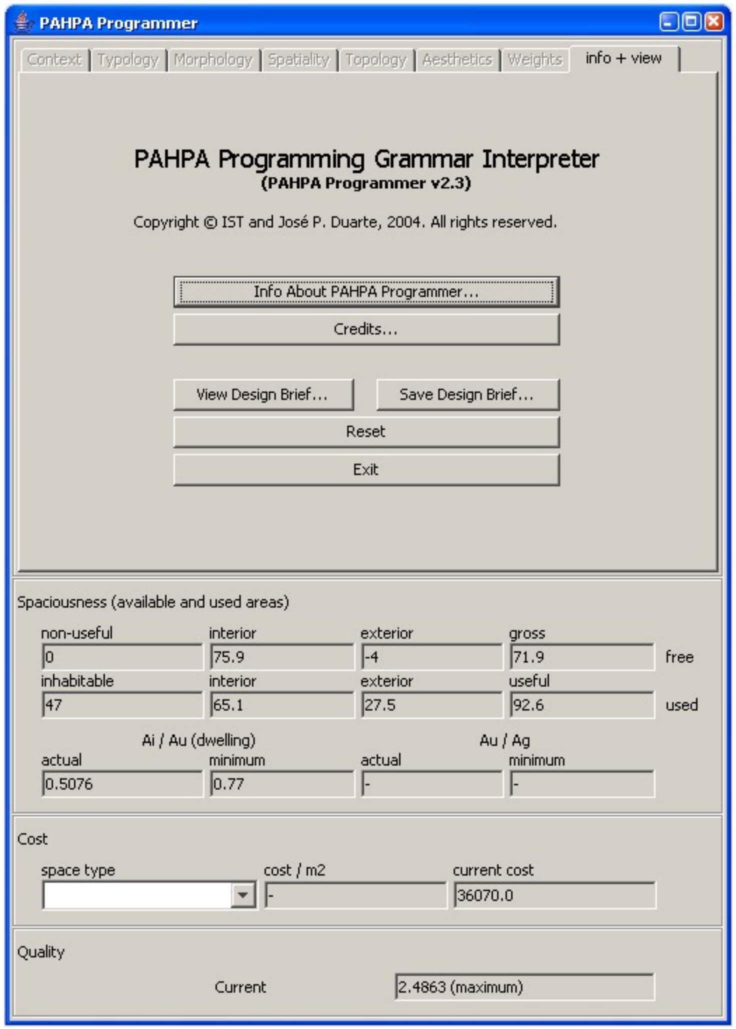
Cost

space type	cost / m2	current cost
	-	36070.0

Quality

Current: 2.4863 (maximum)

PROGRAMA



PAHPA Programming Grammar Interpreter
(PAHPA Programmer v2.3)

Copyright © IST and José P. Duarte, 2004. All rights reserved.

Spaciousness (available and used areas)

non-useful	interior	exterior	gross	
0	75.9	-4	71.9	free
inhabitable	interior	exterior	useful	
47	65.1	27.5	92.6	used

Ai / Au (dwelling)		Au / Ag	
actual	minimum	actual	minimum
0.5076	0.77	-	-

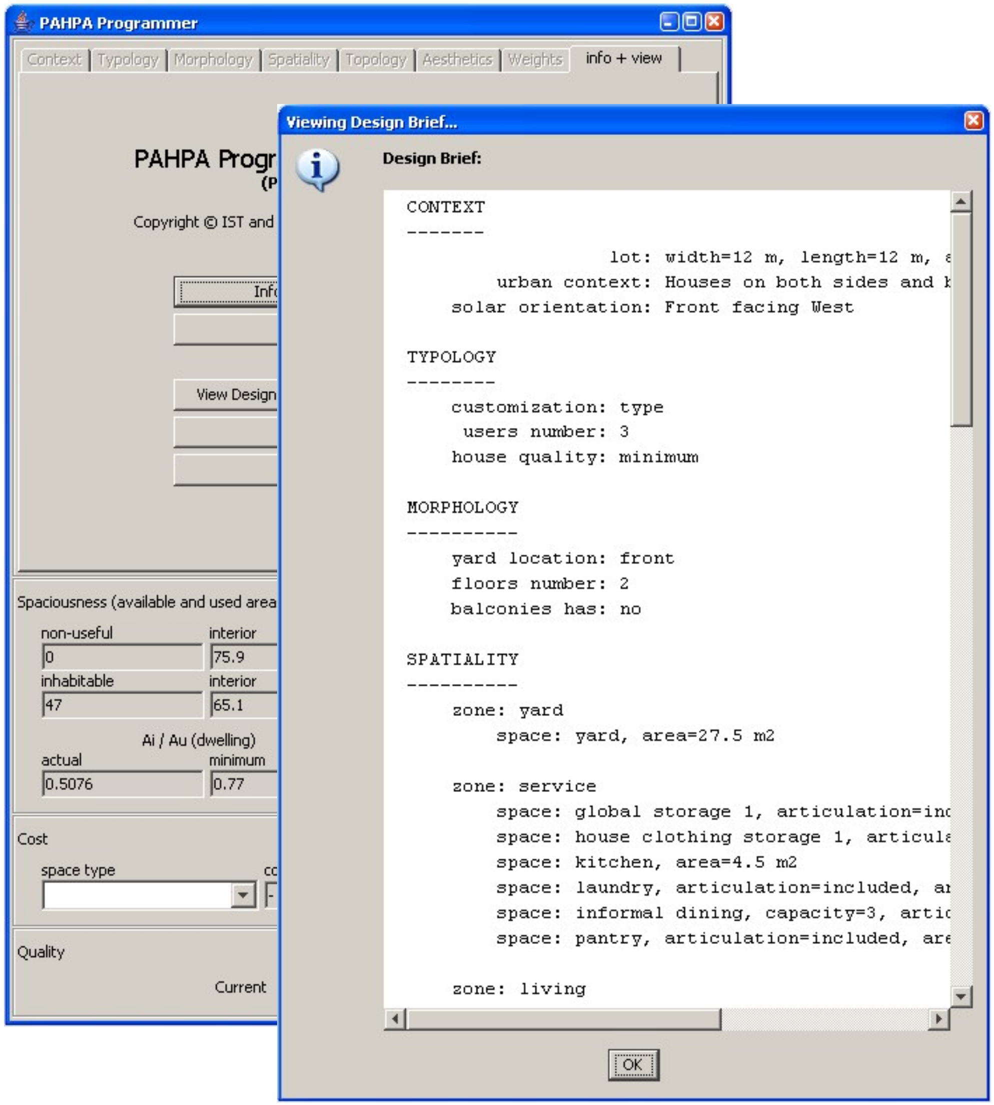
Cost

space type	cost / m2	current cost
	-	36070.0

Quality

Current: 2.4863 (maximum)

PROGRAMA



Spaciousness (available and used area

non-useful: 0 | interior: 75.9

inhabitable: 47 | interior: 65.1

Ai / Au (dwelling)

actual: 0.5076 | minimum: 0.77

Cost

space type

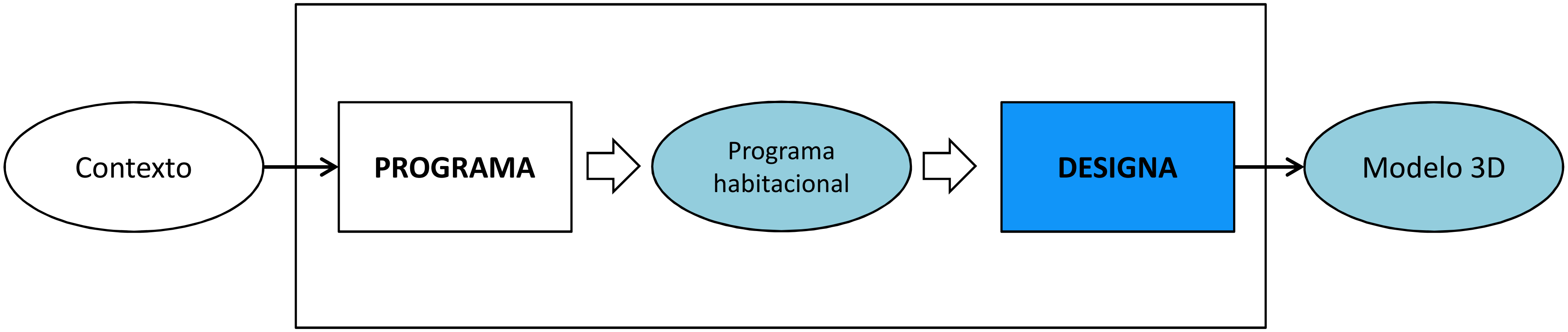
Quality

Current

Design Brief:

```
CONTEXT
-------
                  lot: width=12 m, length=12 m, a
        urban context: Houses on both sides and b
    solar orientation: Front facing West

TYPOLOGY
--------
    customization: type
     users number: 3
    house quality: minimum

MORPHOLOGY
----------
    yard location: front
    floors number: 2
    balconies has: no

SPATIALITY
----------
    zone: yard
        space: yard, area=27.5 m2

    zone: service
        space: global storage 1, articulation=inc
        space: house clothing storage 1, articula
        space: kitchen, area=4.5 m2
        space: laundry, articulation=included, ar
        space: informal dining, capacity=3, artic
        space: pantry, articulation=included, are

    zone: living
```

Gramática Discursiva



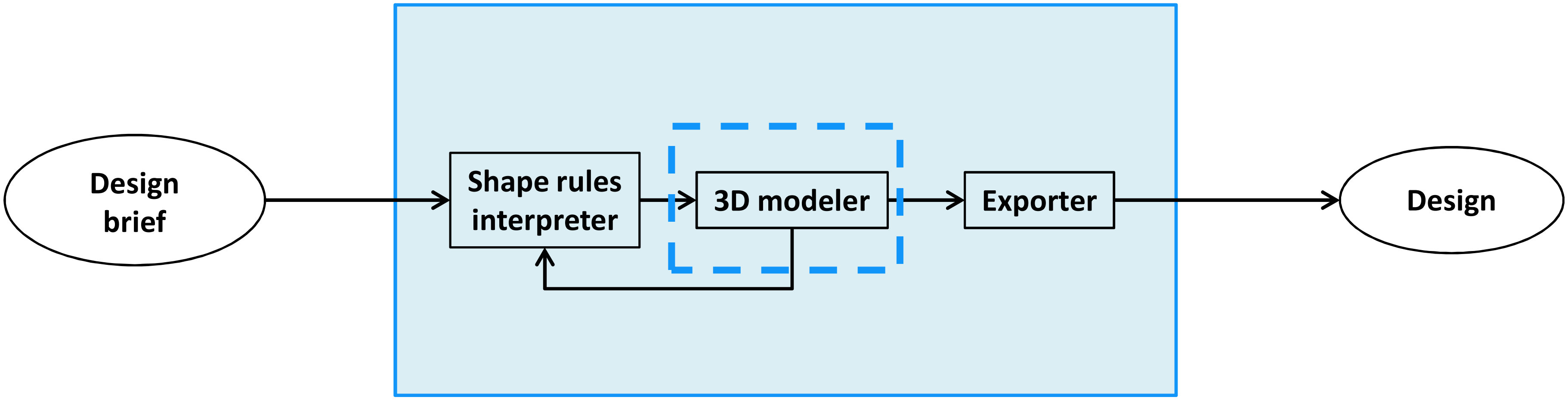
DESIGNA

- Interpretador de gramáticas da forma

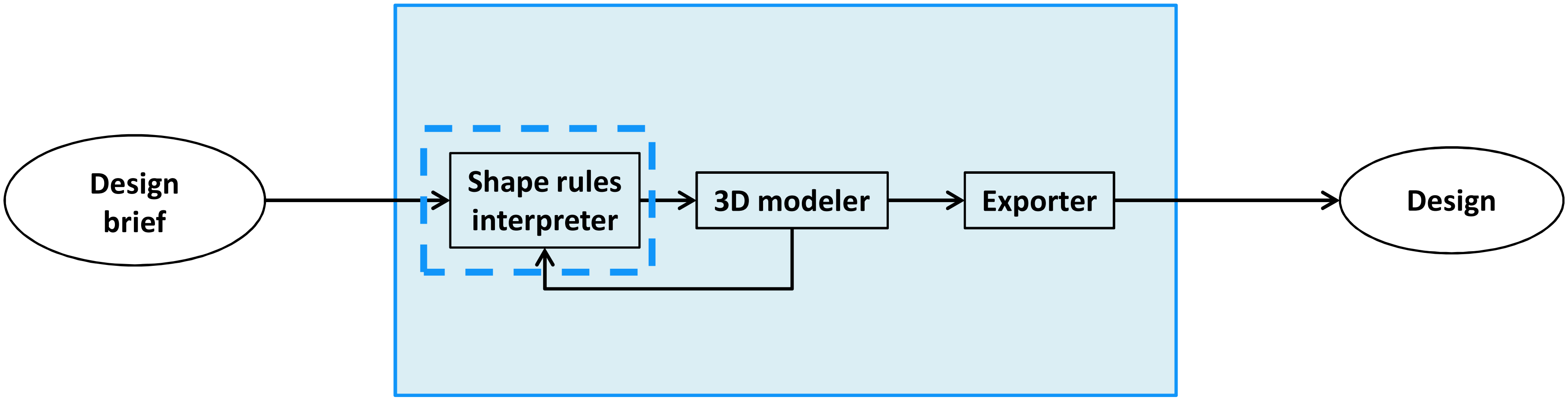
DESIGNA

- Interpretador de gramáticas de forma
  - Formas: Representação + Criação e Modificação
  - Regras: Representação + Aplicação e Controlo
  - CAD

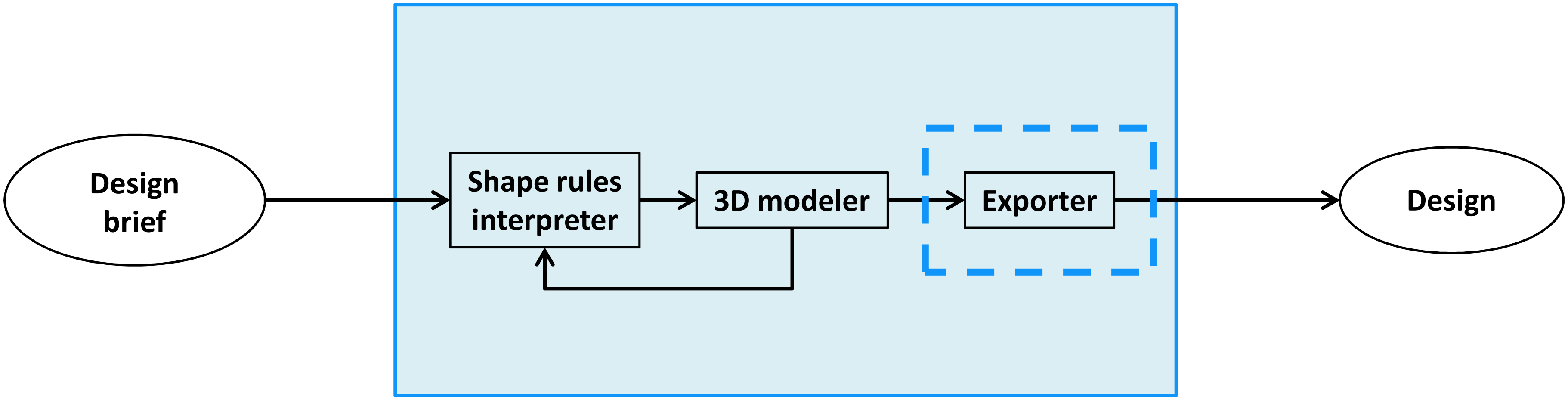
DESIGNA


Design brief
Shape rules interpreter
3D modeler
Exporter
Design

DESIGNA


Design brief
Shape rules interpreter
3D modeler
Exporter
Design

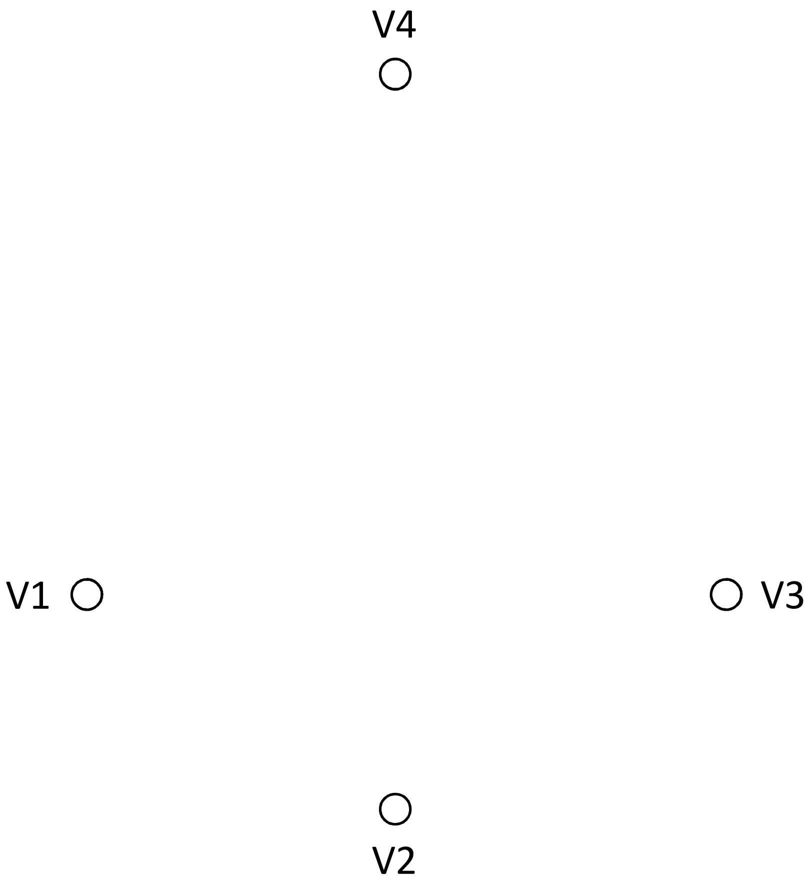
DESIGNA


Design brief
Shape rules interpreter
3D modeler
Exporter
Design

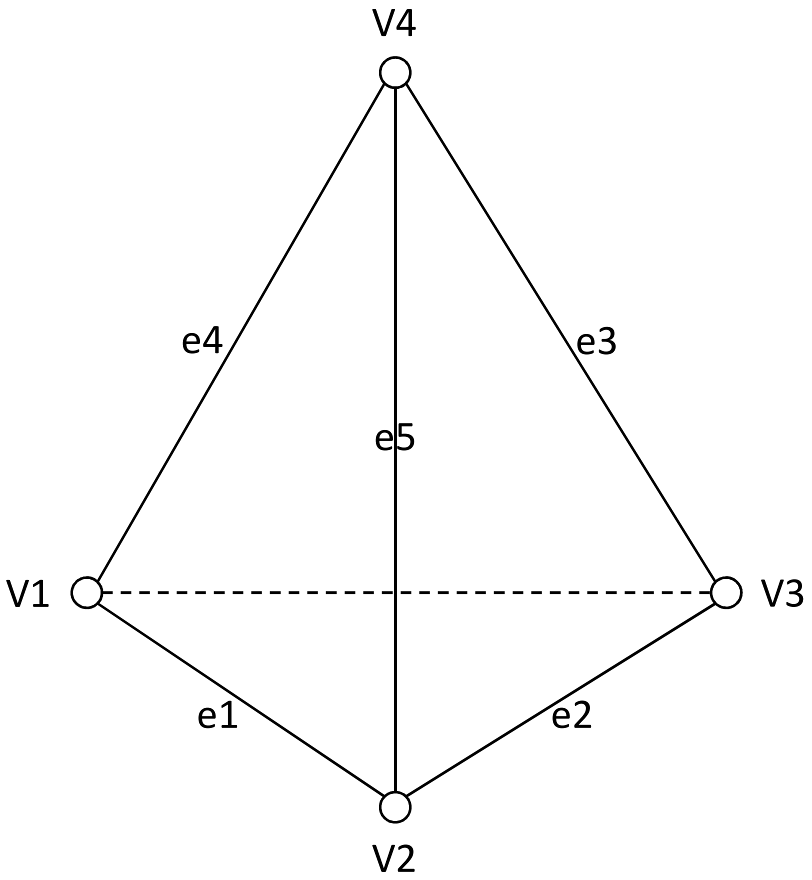
Formas

Representação

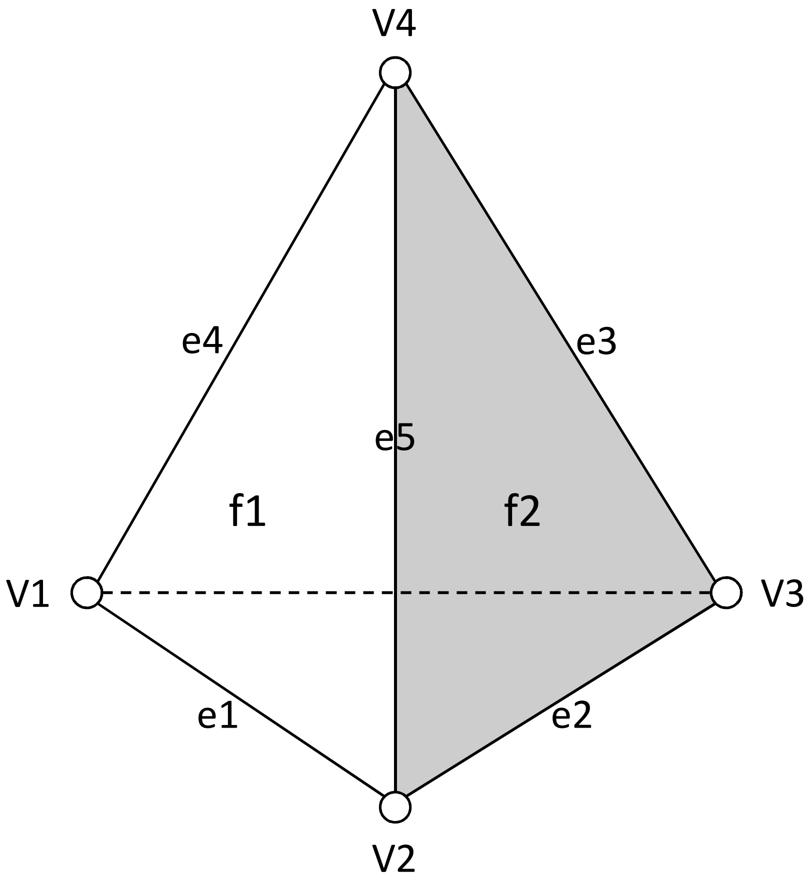
Formas



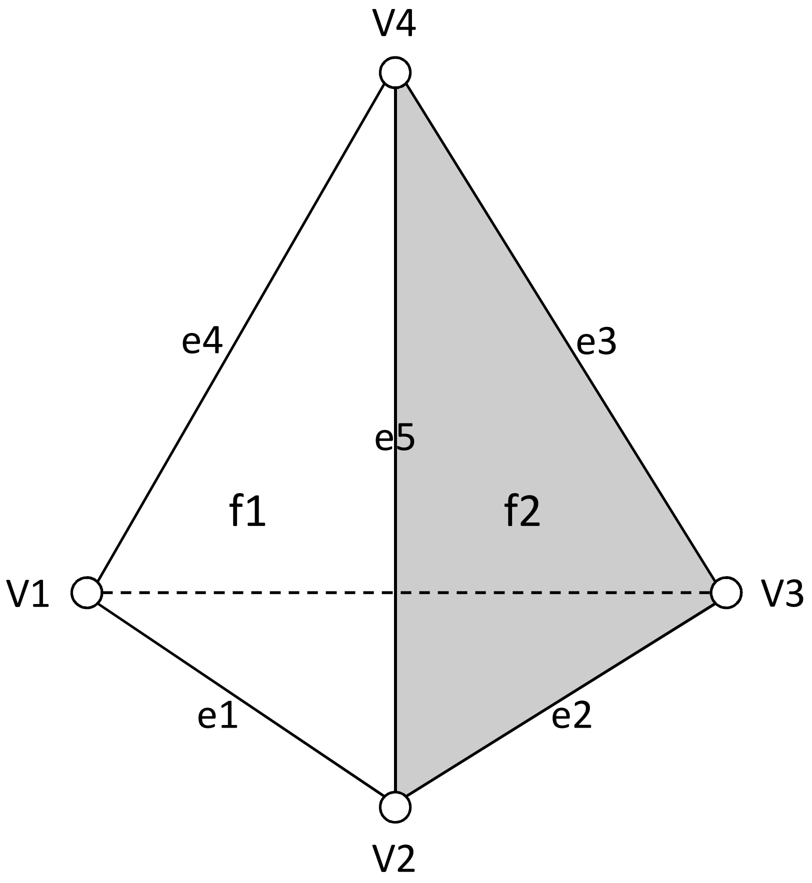
Formas

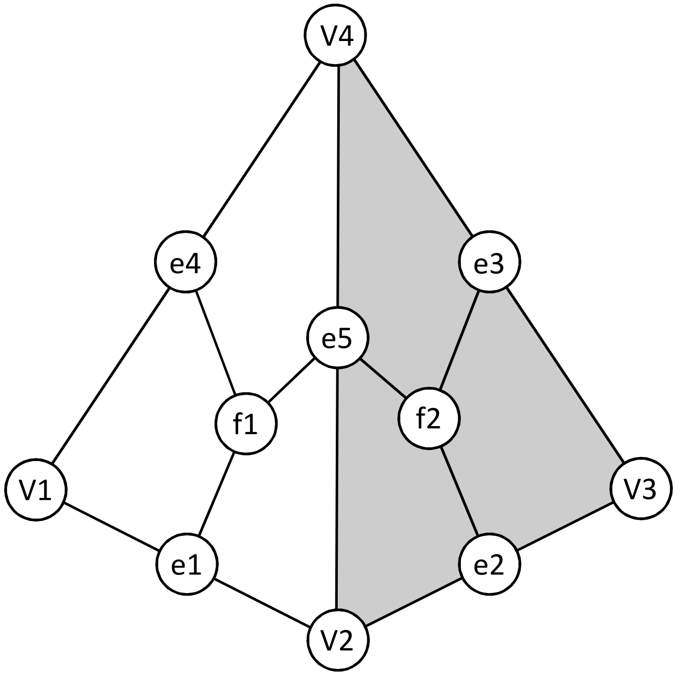


Formas

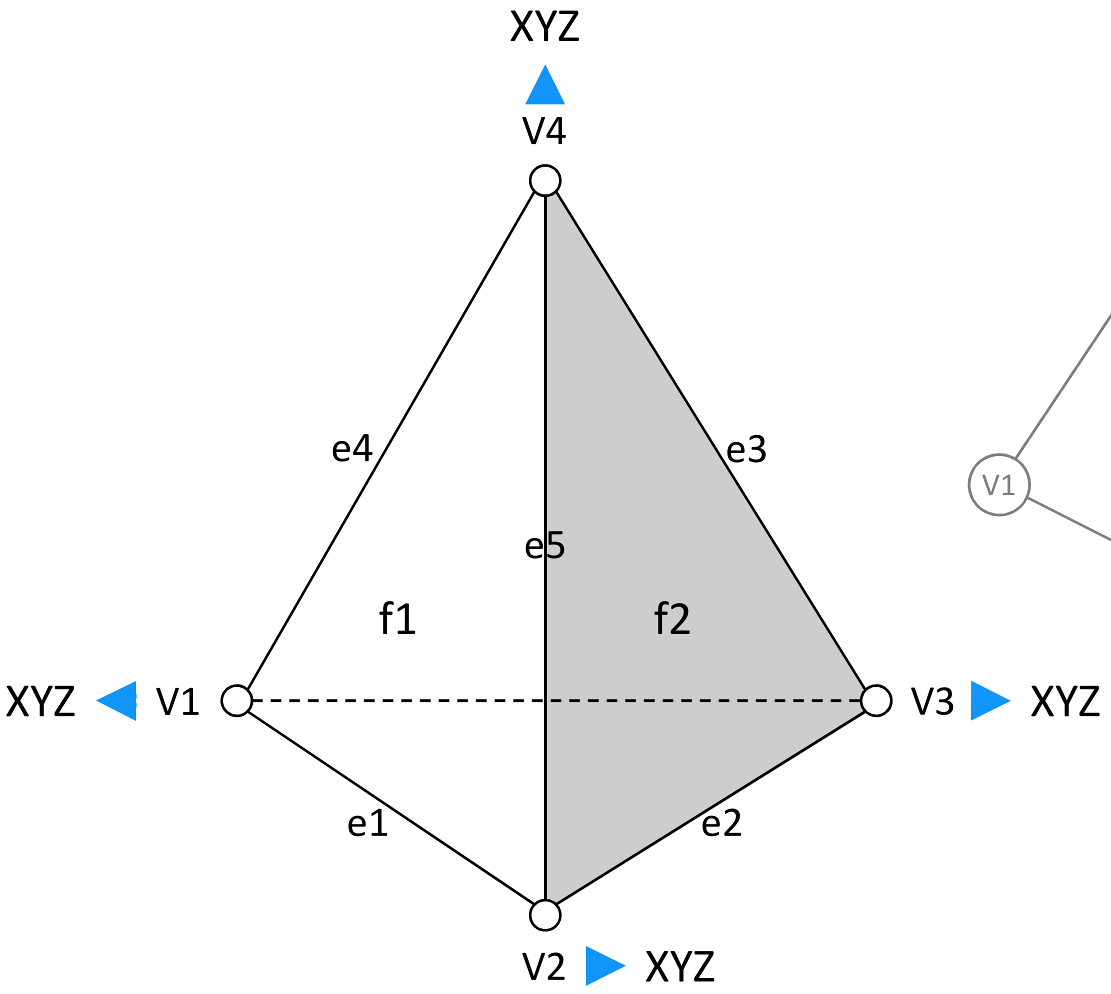


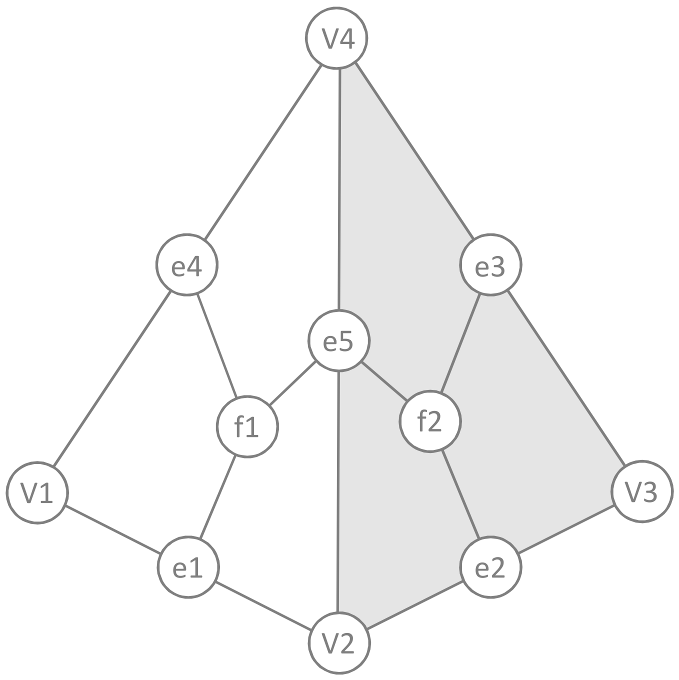
Topologia



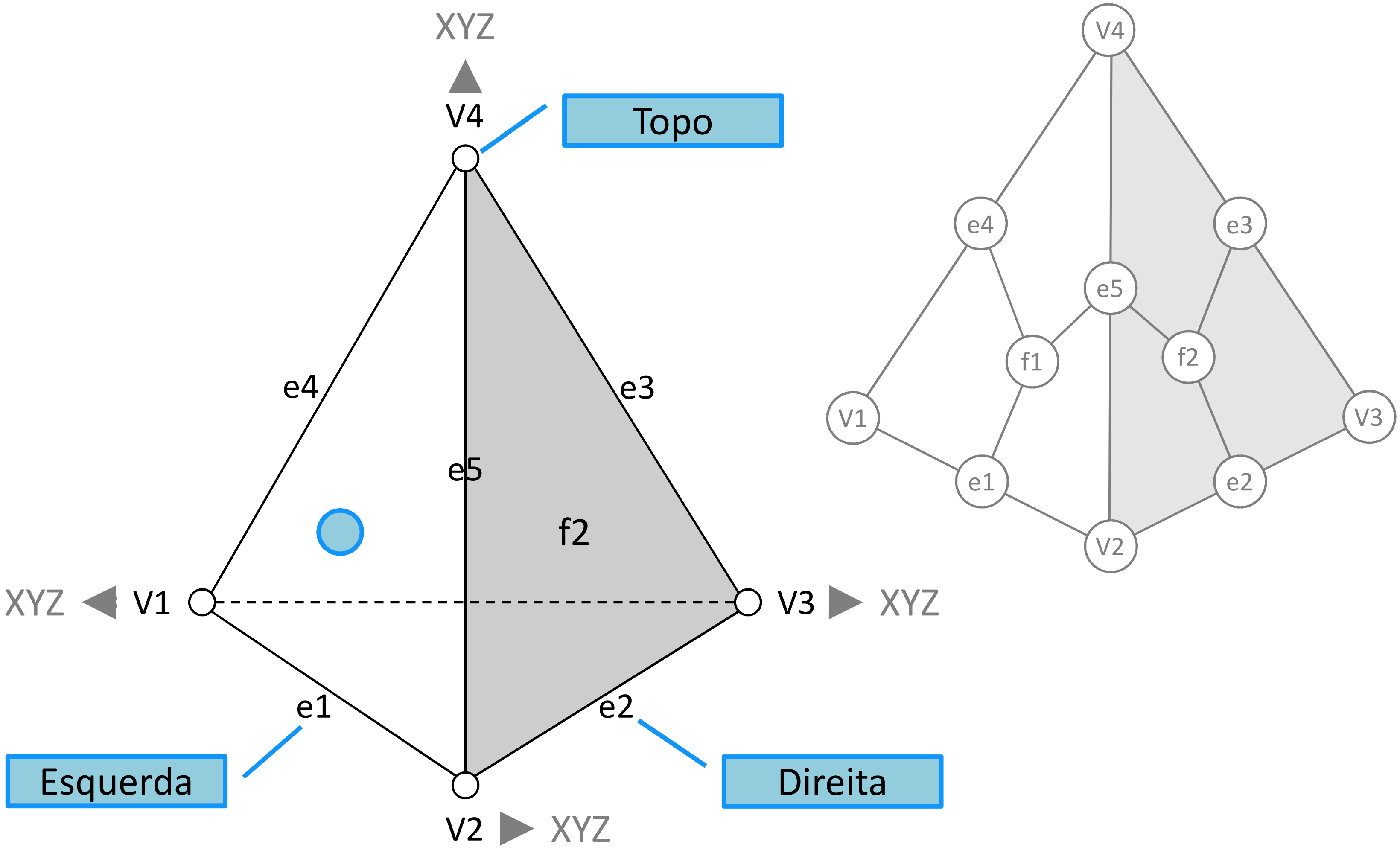


Geometria


XYZ
V4
e4
e3
e5
f1
f2
XYZ
V1
V3
XYZ
e1
e2
V2
XYZ


V4
e4
e3
e5
f1
f2
V1
V3
e1
e2
V2

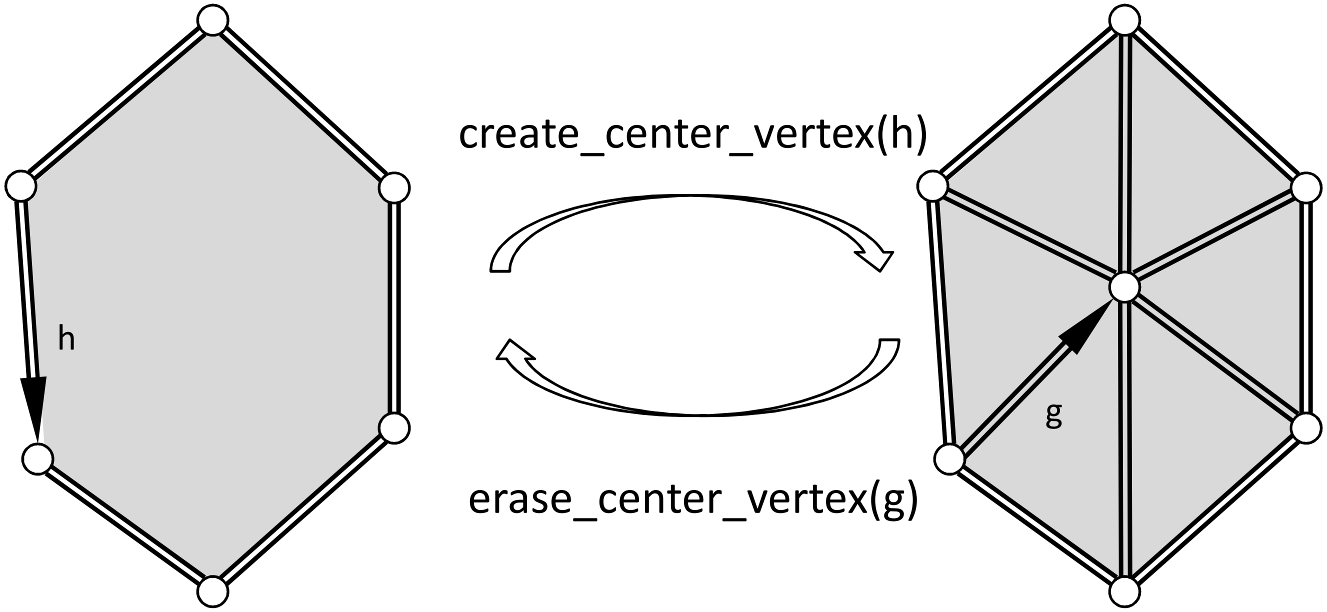
Rótulos


XYZ
V4
Topo
e4
e3
e5
f2
XYZ
V1
V3
XYZ
e1
e2
Esquerda
Direita
V2
XYZ
V4
e4
e3
e5
f1
f2
V1
V3
e1
e2
V2

Formas

Criação e Modificação

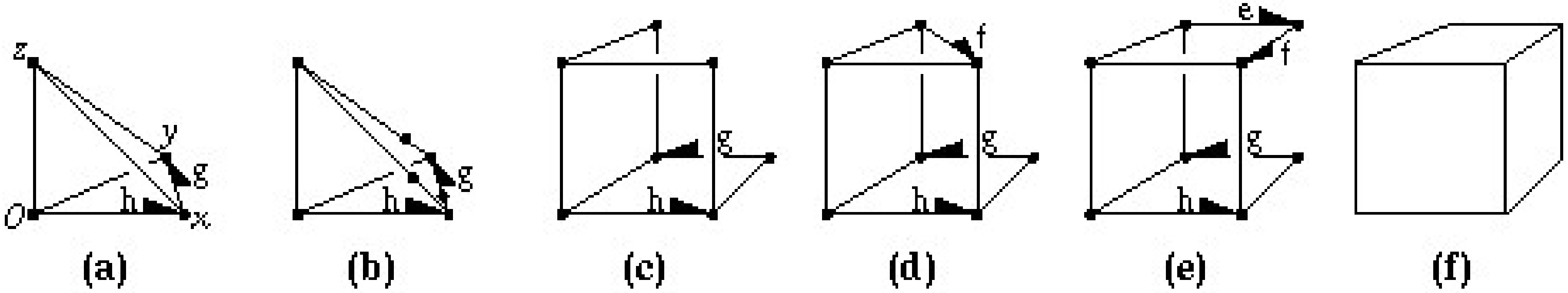
Formas



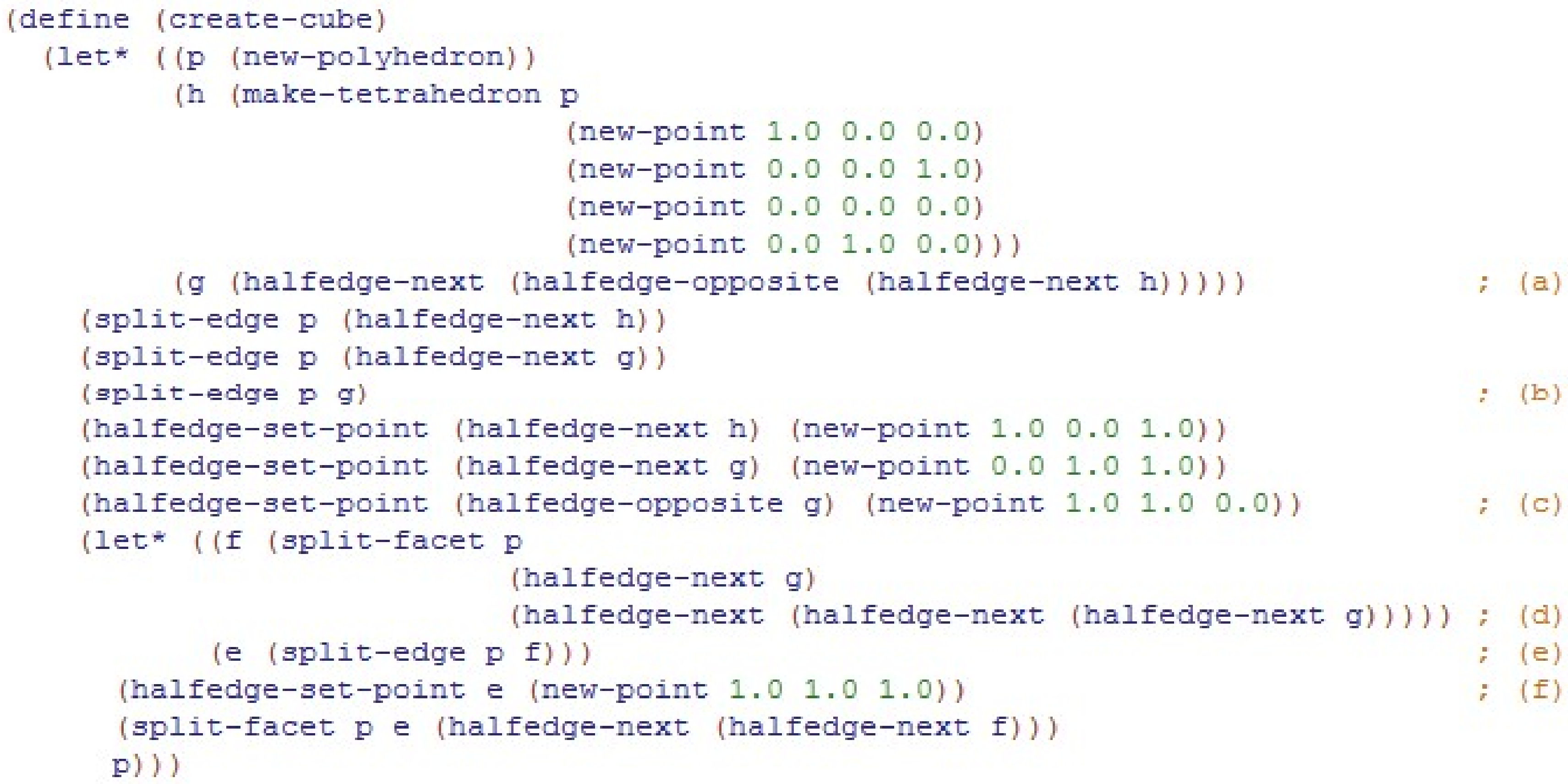
create_center_vertex(h)

erase_center_vertex(g)

Formas



Formas



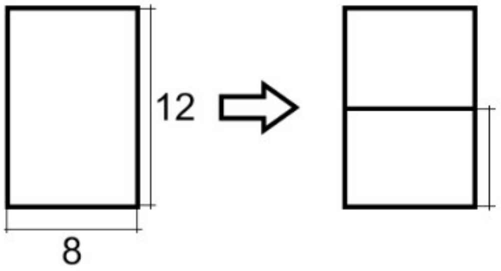
```
(define (create-cube)
  (let* ((p (new-polyhedron))
         (h (make-tetrahedron p
                              (new-point 1.0 0.0 0.0)
                              (new-point 0.0 0.0 1.0)
                              (new-point 0.0 0.0 0.0)
                              (new-point 0.0 1.0 0.0)))
         (g (halfedge-next (halfedge-opposite (halfedge-next h)))))            ; (a)
    (split-edge p (halfedge-next h))
    (split-edge p (halfedge-next g))
    (split-edge p g)                                                           ; (b)
    (halfedge-set-point (halfedge-next h) (new-point 1.0 0.0 1.0))
    (halfedge-set-point (halfedge-next g) (new-point 0.0 1.0 1.0))
    (halfedge-set-point (halfedge-opposite g) (new-point 1.0 1.0 0.0))         ; (c)
    (let* ((f (split-facet p
                           (halfedge-next g)
                           (halfedge-next (halfedge-next (halfedge-next g))))) ; (d)
           (e (split-edge p f)))                                               ; (e)
      (halfedge-set-point e (new-point 1.0 1.0 1.0))                           ; (f)
      (split-facet p e (halfedge-next (halfedge-next f)))
      p)))
```

Regras

Representação

Regras

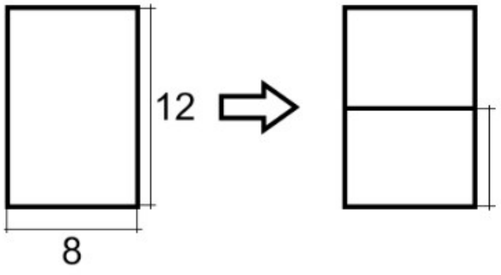
Regra gramática da forma:



onde d = 6 ou d = 7

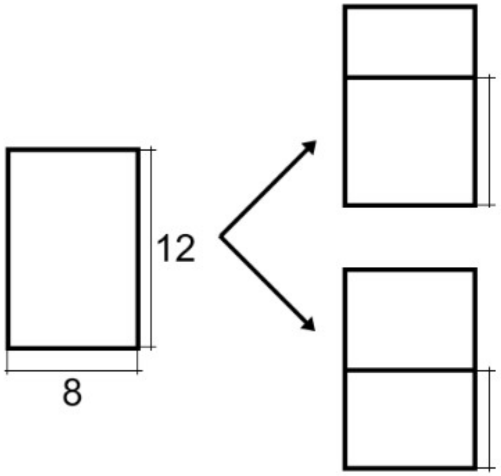
Regras

Regra gramática da forma:



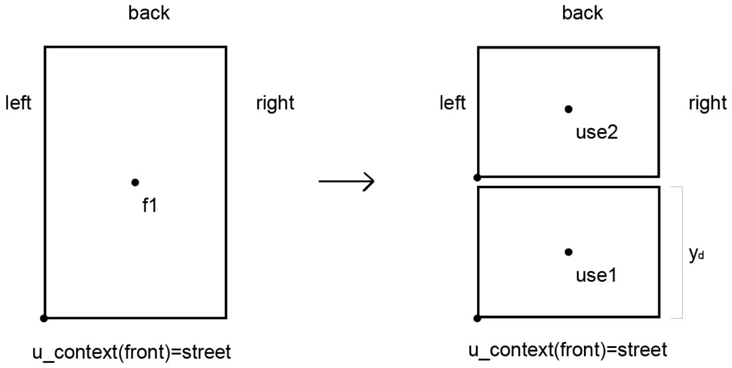
onde d = 6 ou d = 7

Operador DESIGNA:



Regras

R5: **Locate inside/outside zones on the first floor**



$\alpha_1 \leftarrow \alpha_1$

$\alpha_8 \leftarrow \alpha_8$

$\alpha_9 \leftarrow \alpha_9 , \forall\ \alpha_1 , \alpha_8 = \text{frontyard} \wedge \alpha_9 = \text{true}$

$\quad\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 6.00 \wedge a_{in} = a_{use2} \wedge a_{ou} = a_{use1}$

$\quad \forall\ \alpha_1 , \alpha_8 = \text{frontyard} \wedge \alpha_9 = \text{false}$

$\quad\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 7.00 \wedge a_{in} = a_{use2} \wedge a_{ou} = a_{use1}$

$\quad \alpha_1 = < \text{street, ?use, street, ?use} > , \forall\ \text{?use} \wedge \alpha_8 = \text{backyard} , \forall\ \alpha_9$

$\quad\quad \Rightarrow \text{use1} = \text{inside1} \wedge \text{use2} = \text{outside1} \wedge y_d = 7.00 \wedge a_{in} = a_{use1} \wedge a_{ou} = a_{use2}$

$\quad \alpha_1 = < \text{street, ?use, street, ?use} > , \forall\ \text{?use} \wedge \alpha_8 = \text{backyard} , \alpha_9 = \text{true}$

$\quad\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 6.00 \wedge a_{in} = a_{use1} \wedge a_{ou} = a_{use2}$

$\quad \alpha_1 = < \text{street, ?use, street, ?use} > , \forall\ \text{?use} \wedge \alpha_8 = \text{backyard} , \alpha_9 = \text{false}$

$\quad\quad \Rightarrow \text{use1} = \text{outside1} \wedge \text{use2} = \text{inside1} \wedge y_d = 5.00 \wedge a_{in} = a_{use1} \wedge a_{ou} = a_{use2}$

$\delta_{13} \leftarrow \delta_{13} - < [(\text{f1}, id_{f1}, \varnothing, ((x_{f1}, y_{f1}, z_{f1}) , dx_{f1}, dy_{f1}, dz_{f1}, a_{f1}] >$

$\quad + < [(\text{use1}, id_{f1}, \varnothing, ((x_{f1}, y_{f1}, z_{f1}) , dx_{f1}, dy_{f1} - (dx_{f1} - y_d + 2 \cdot 0.10), dz_{f1}, dx_{f1} \cdot dy_{f1} - (\text{f1}_{dy} - y_d + 2 \cdot 0.10)],$

$\quad\quad [(\text{use2}, \max (id) + 1, \varnothing, ((x_{f1}, y_{f1} + y_d, z_{f1}) , dx_{f1}, dy_{f1} - y_d, dz_{f1}, dx_{f1} . dy_{f1} - y_d] >$

$\delta_{15} \leftarrow \delta_{15} + < \text{available}, (\text{f1}_{dx} \cdot 0.20, a_{in}, - (a_{in} + \text{f1}_{dx} \cdot 0.20), - \text{f1}_{dx} \cdot 0.20), - A_u / A_g + A_u - \text{f1}_{dx} \cdot 0.20 / A_g >$

$\delta_{17} \leftarrow \delta_{17} - < [id_{f1}, id_{?space}, \text{adjacent}] , \text{?space} \in \{\text{front, left, back, right}\}$

$\quad + < [id_{inside1}, id_{?left}, \text{adjacent}],$

$\quad\quad [id_{inside1}, id_{?right}, \text{adjacent}],$

$\quad\quad [id_{outside1}, id_{?left}, \text{adjacent}],$

$\quad\quad [id_{outside1}, id_{?right}, \text{adjacent}],$

$\quad\quad [id_{inside1}, id_{?space1}, \text{adjacent}]$

$\quad\quad [id_{outside1}, id_{?space2}, \text{adjacent}]$

$\quad\quad\quad \alpha_8 = \text{frontyard} \Rightarrow \text{?space}_1 = \text{back} \wedge \text{?space}_2 = \text{front}$

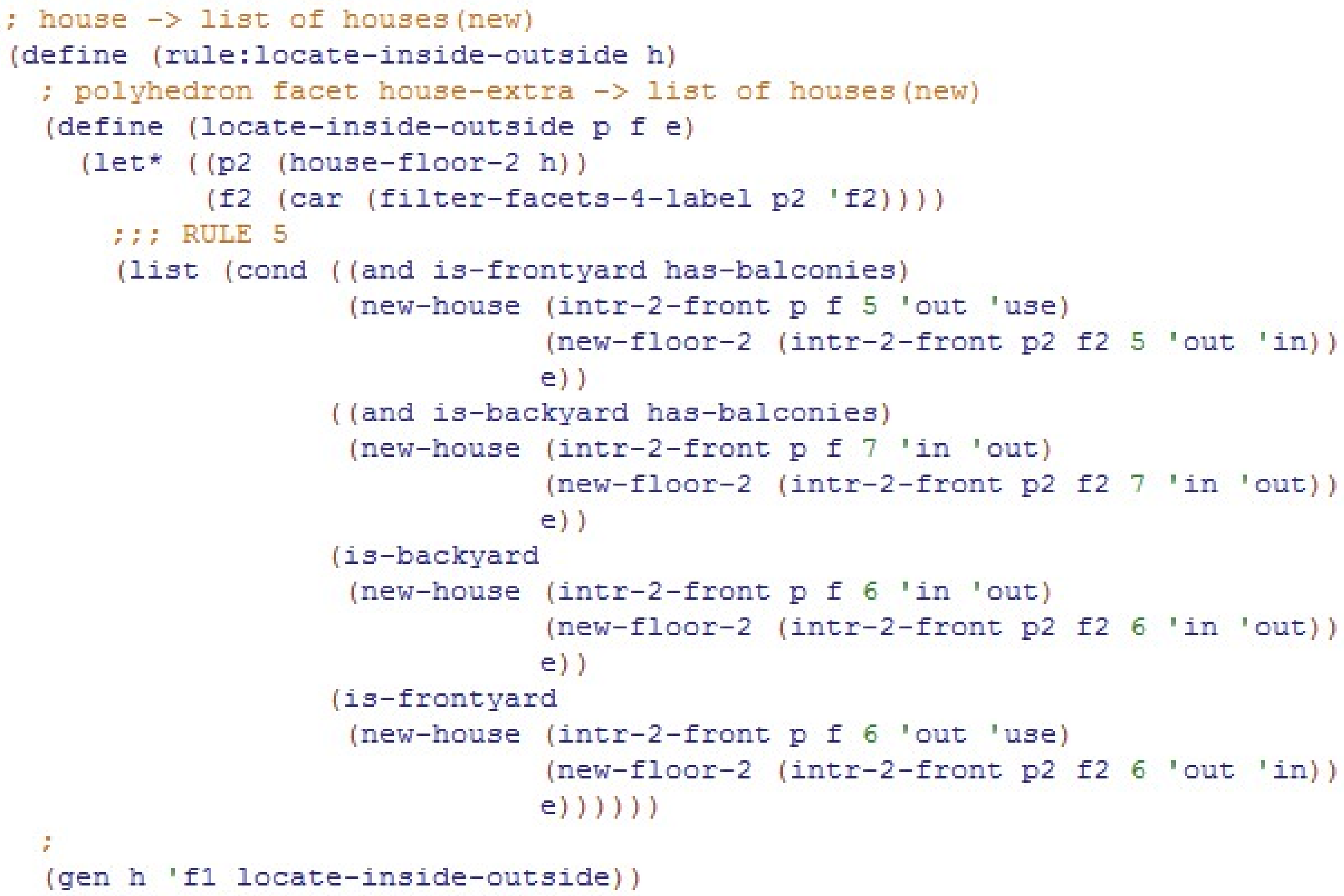
$\quad\quad\quad \alpha_8 = \text{backyard} \Rightarrow \text{?space}_1 = \text{front} \wedge \text{?space}_2 = \text{back}$

$\delta_{20} \leftarrow \delta_{20} + < [\text{wall}, \max(id) + 1, (\text{inside, outside}), ((x_{f1}, y_d - 0.10, z_{f1}), dx_{f1}, 0.20, dz_{f1}, dx_{f1} \cdot dz_{f1})] >$

$\delta_{24} \leftarrow \delta_{24} + \text{wall\_cost} (dx_{f1} \cdot dz_{f1}, \text{unit\_cost (wall, 0.20, material)})$

$\alpha_{25} \leftarrow \alpha_{25} + < [\text{R4}, 0] >$

Regras

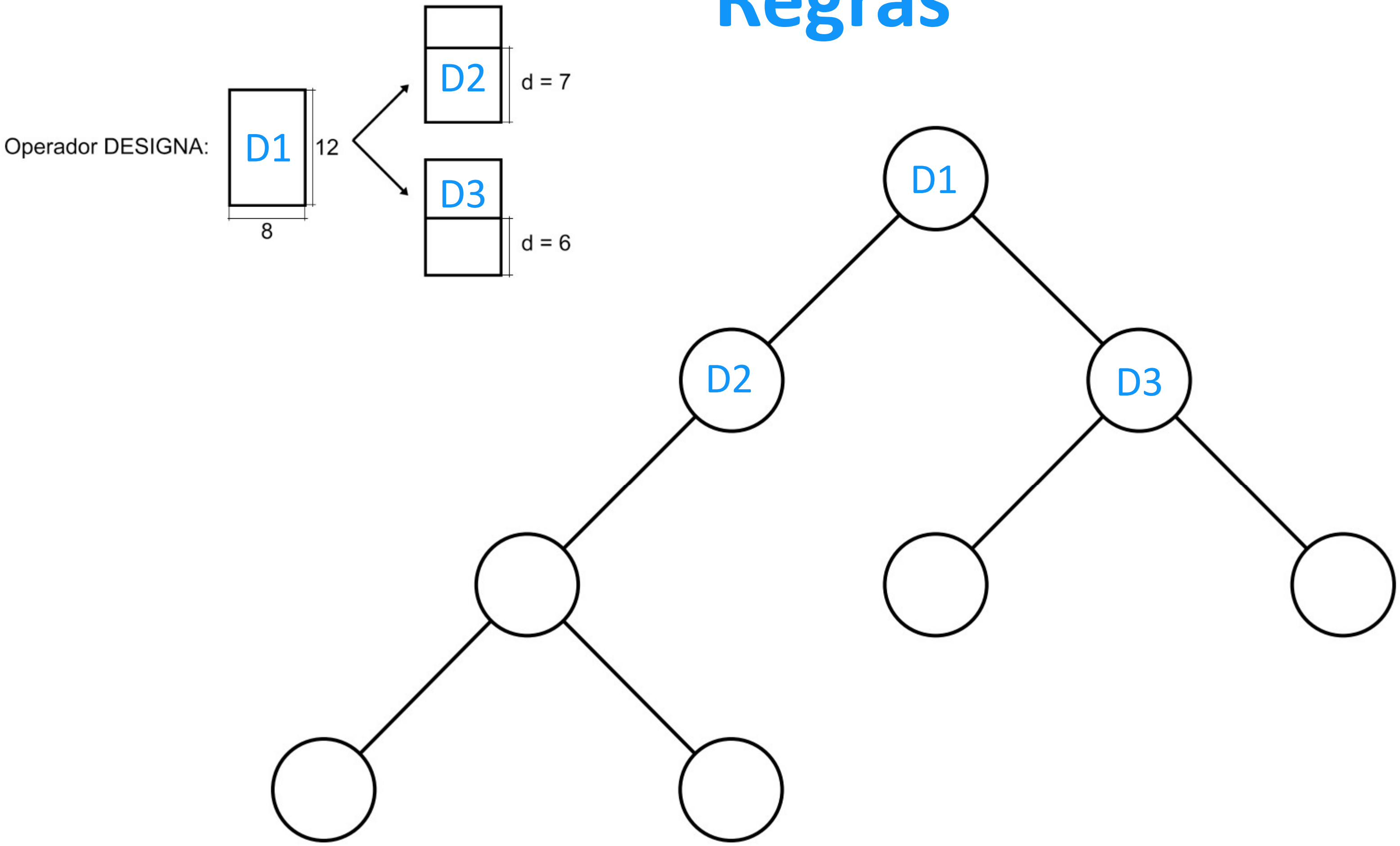


```
; house -> list of houses(new)
(define (rule:locate-inside-outside h)
  ; polyhedron facet house-extra -> list of houses(new)
  (define (locate-inside-outside p f e)
    (let* ((p2 (house-floor-2 h))
           (f2 (car (filter-facets-4-label p2 'f2))))
      ;;; RULE 5
      (list (cond ((and is-frontyard has-balconies)
                   (new-house (intr-2-front p f 5 'out 'use)
                              (new-floor-2 (intr-2-front p2 f2 5 'out 'in))
                              e))
                  ((and is-backyard has-balconies)
                   (new-house (intr-2-front p f 7 'in 'out)
                              (new-floor-2 (intr-2-front p2 f2 7 'in 'out))
                              e))
                  (is-backyard
                   (new-house (intr-2-front p f 6 'in 'out)
                              (new-floor-2 (intr-2-front p2 f2 6 'in 'out))
                              e))
                  (is-frontyard
                   (new-house (intr-2-front p f 6 'out 'use)
                              (new-floor-2 (intr-2-front p2 f2 6 'out 'in))
                              e))))))
  ;
  (gen h 'f1 locate-inside-outside))
```

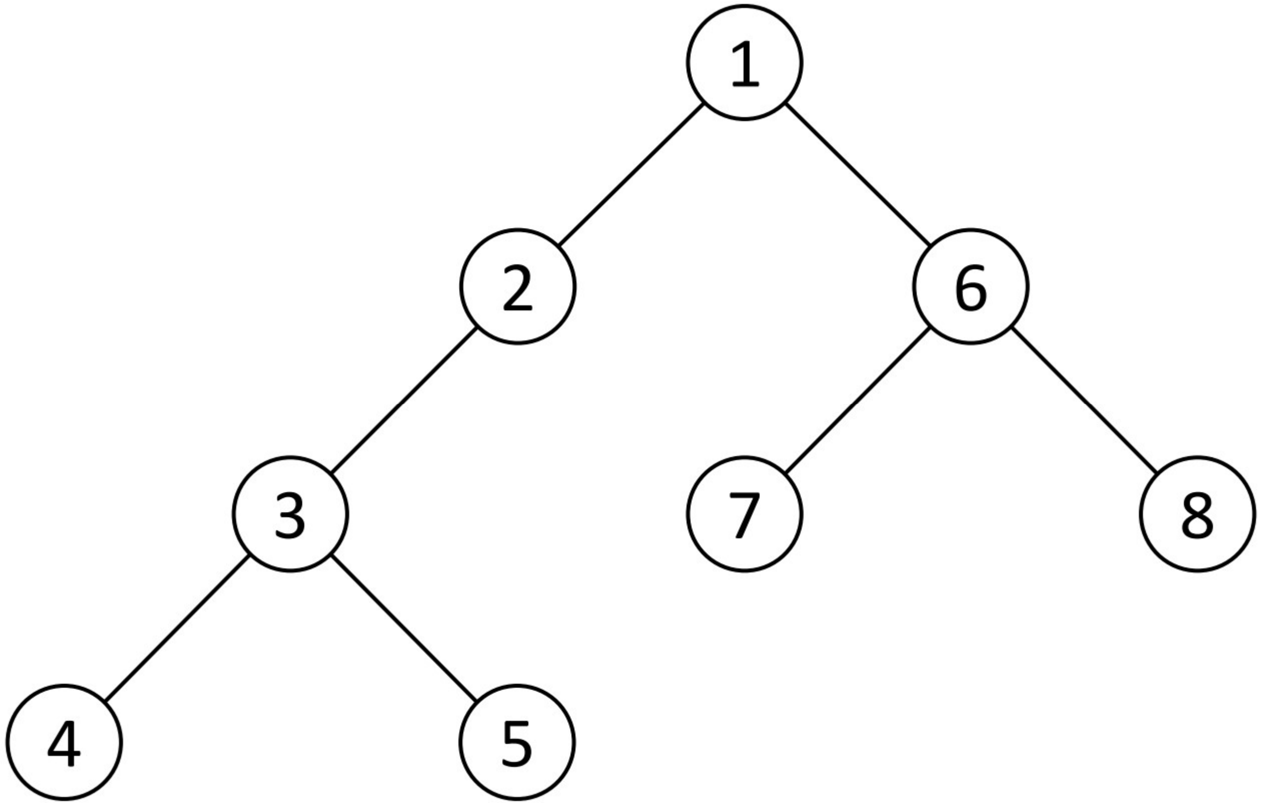
Regras

Aplicação & Controlo

Regras

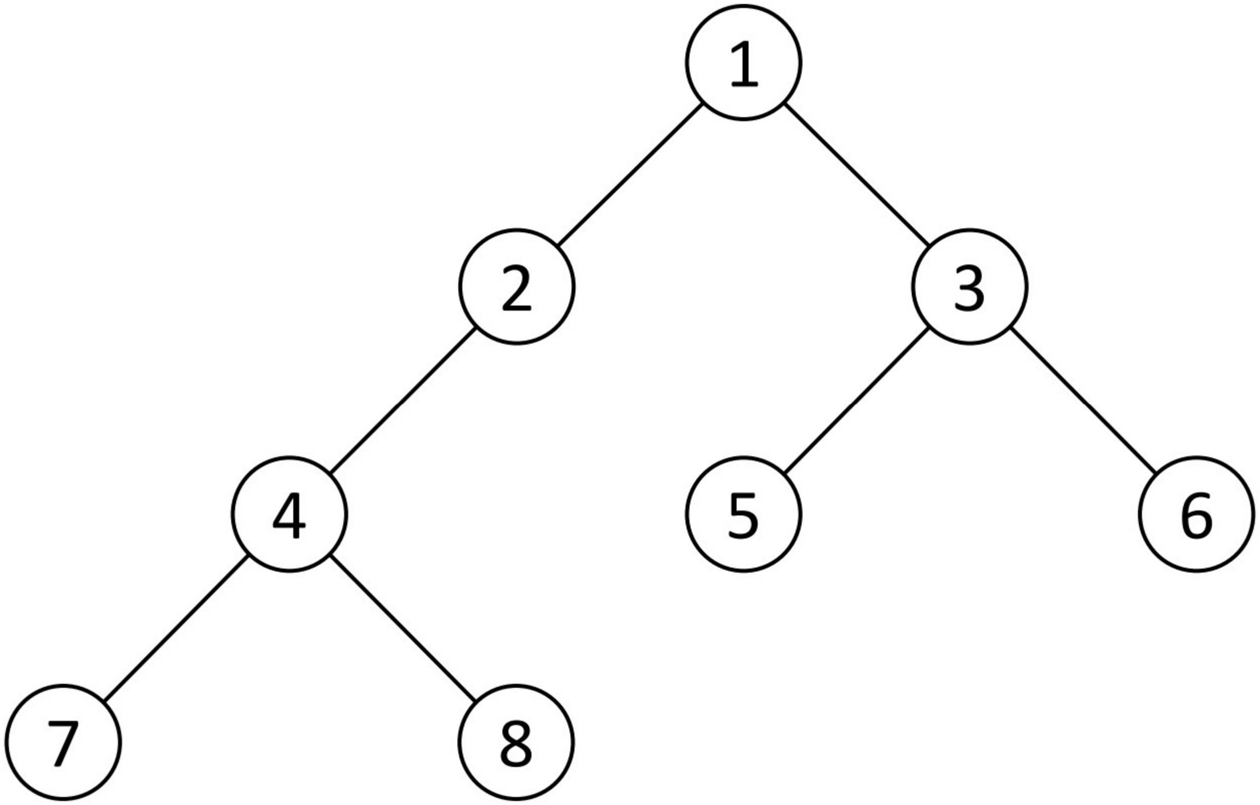


Regras



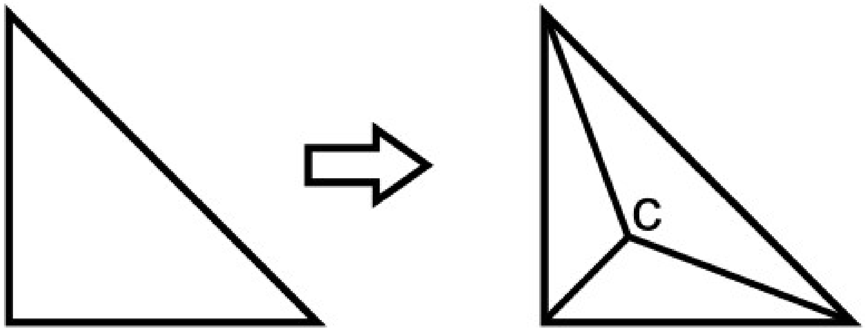
Procura em profundidade

Regras



Procura em largura

Gramática 3D



facet f

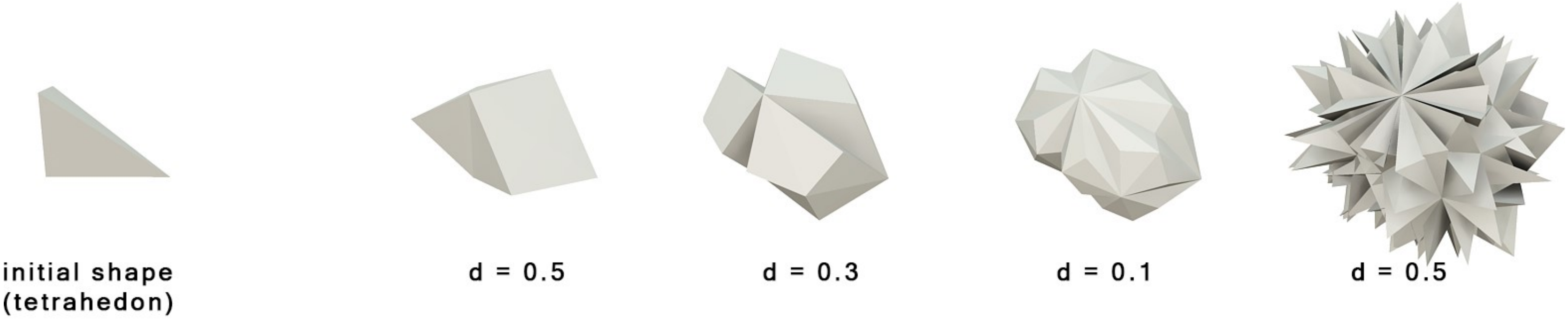
3 new facets, f1, f2, f3

c = f center + f normal * d

Gramática 3D

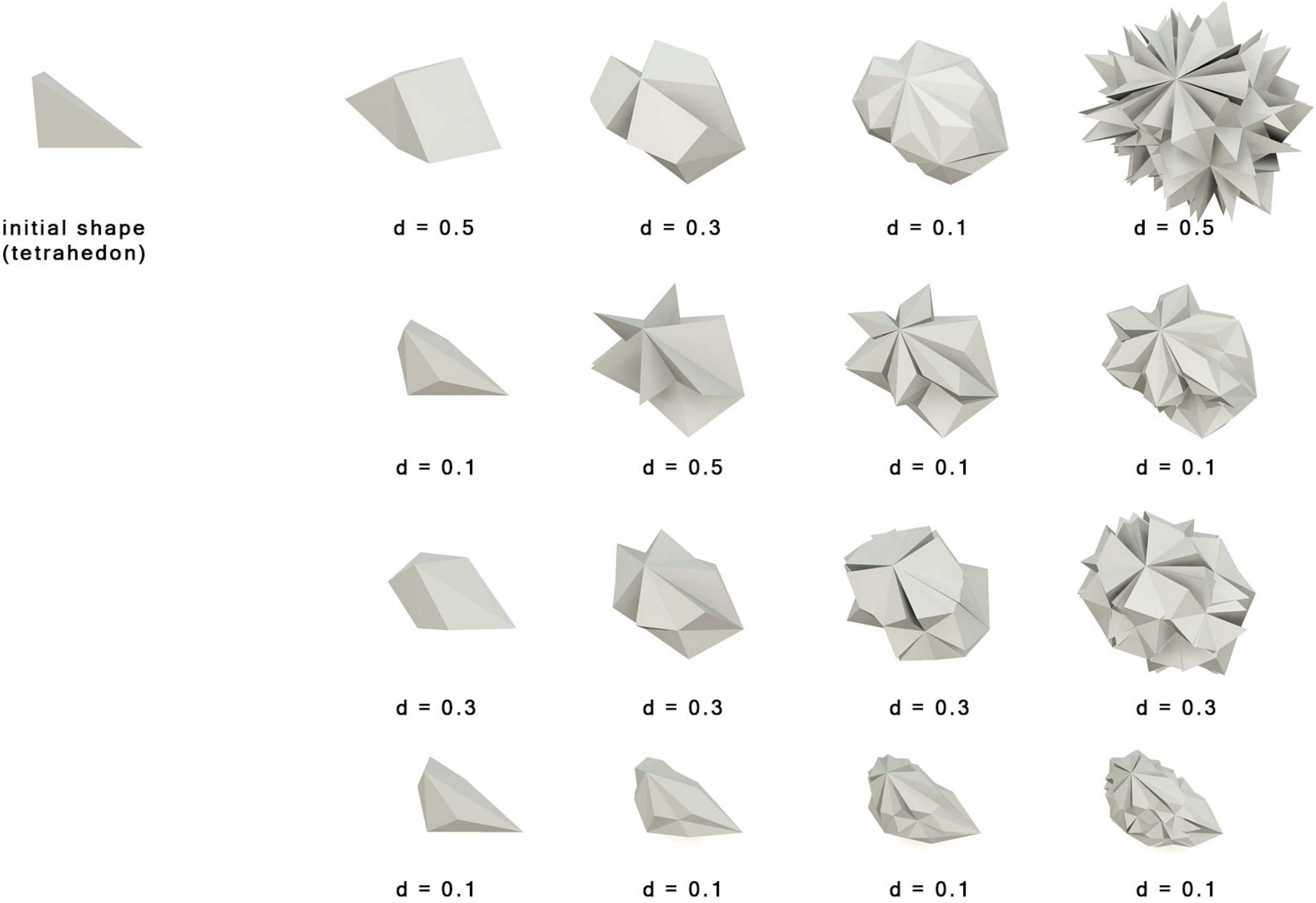
Procura em profundidade

Gramática 3D



Procura em largura

Gramática 3D



initial shape
(tetrahedon)

d = 0.5

d = 0.3

d = 0.1

d = 0.5

d = 0.1

d = 0.5

d = 0.1

d = 0.1

d = 0.3

d = 0.3

d = 0.3

d = 0.3

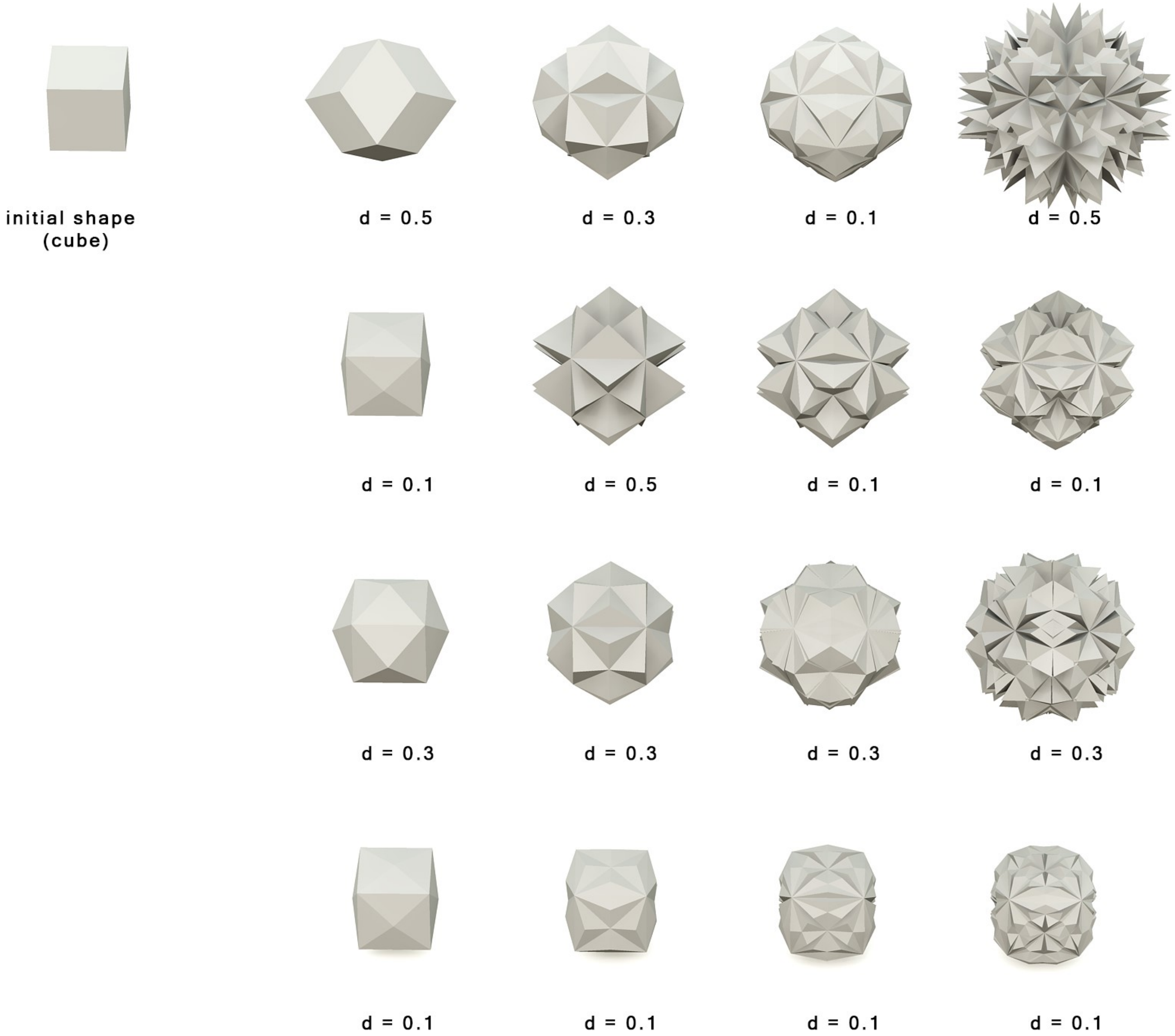
d = 0.1

d = 0.1

d = 0.1

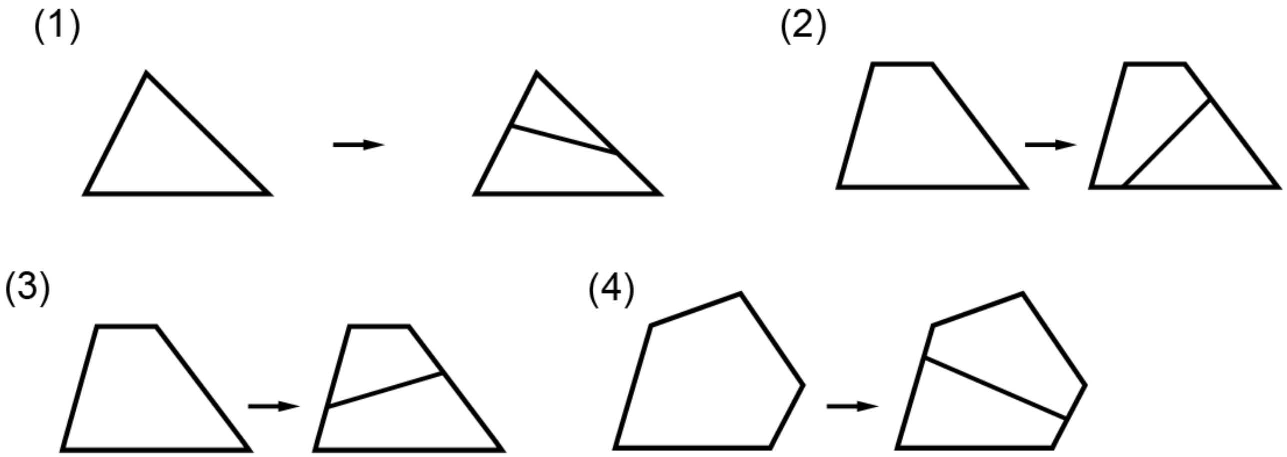
d = 0.1

Gramática 3D

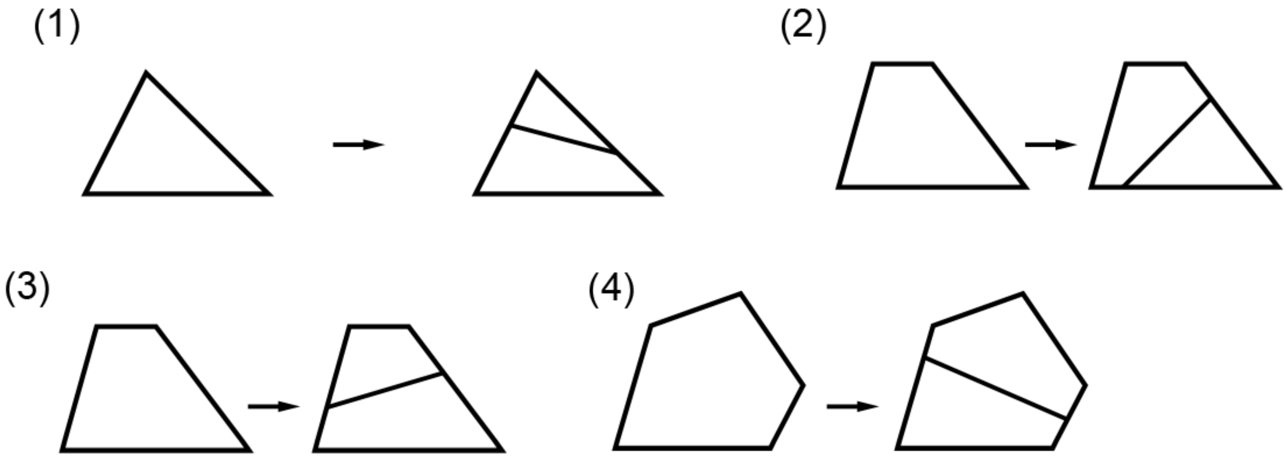


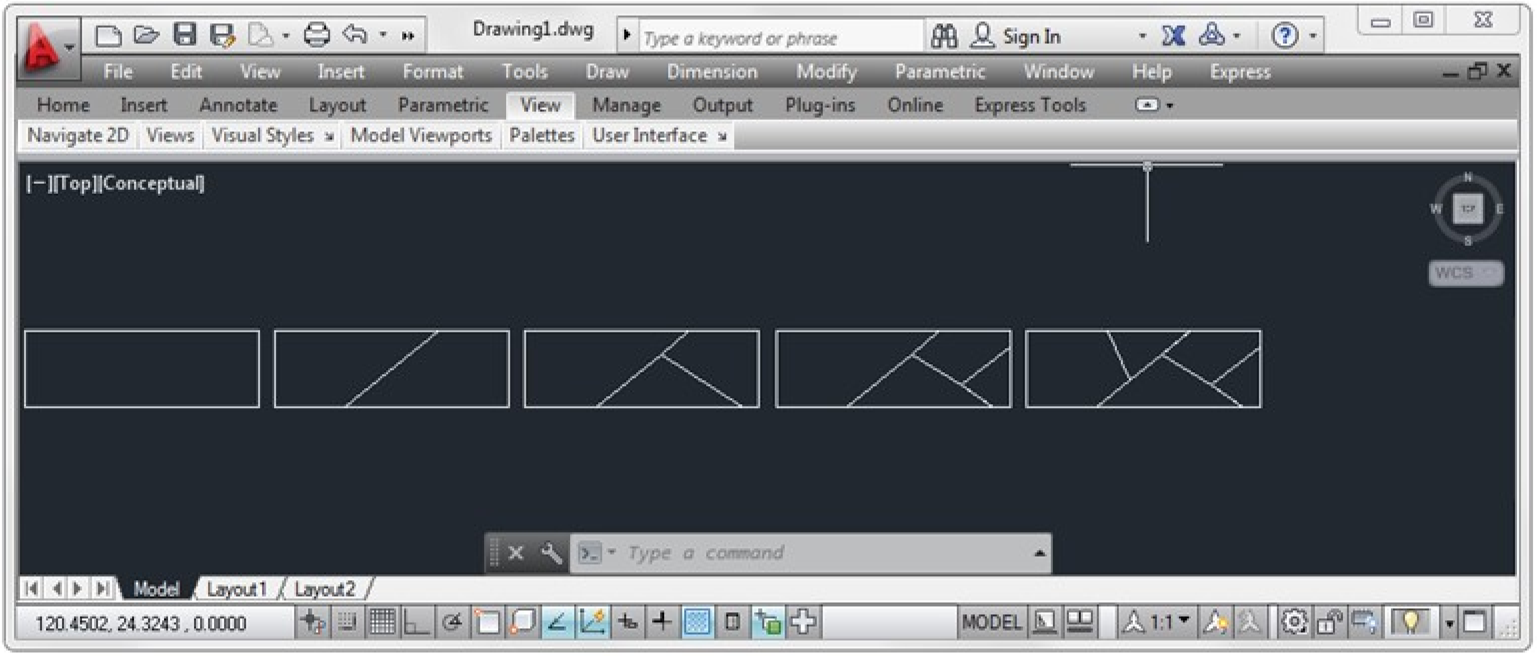
Exemplos

Ice-ray



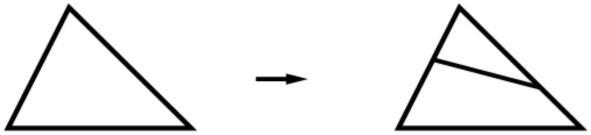
Ice-ray


(1)
(2)
(3)
(4)

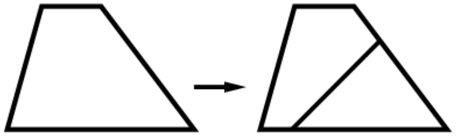

Drawing1.dwg
Type a keyword or phrase
Sign In
File Edit View Insert Format Tools Draw Dimension Modify Parametric Window Help Express
Home Insert Annotate Layout Parametric View Manage Output Plug-ins Online Express Tools
Navigate 2D Views Visual Styles Model Viewports Palettes User Interface
[−][Top][Conceptual]
WCS
Type a command
Model Layout1 Layout2
120.4502, 24.3243 , 0.0000
MODEL
1:1

Ice-ray

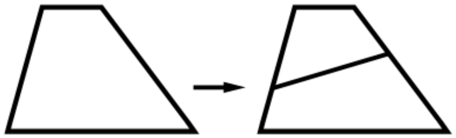
(1)



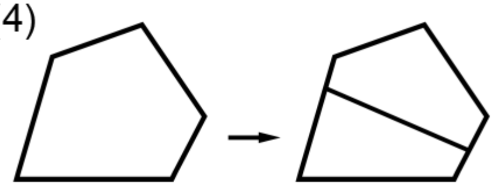
(2)

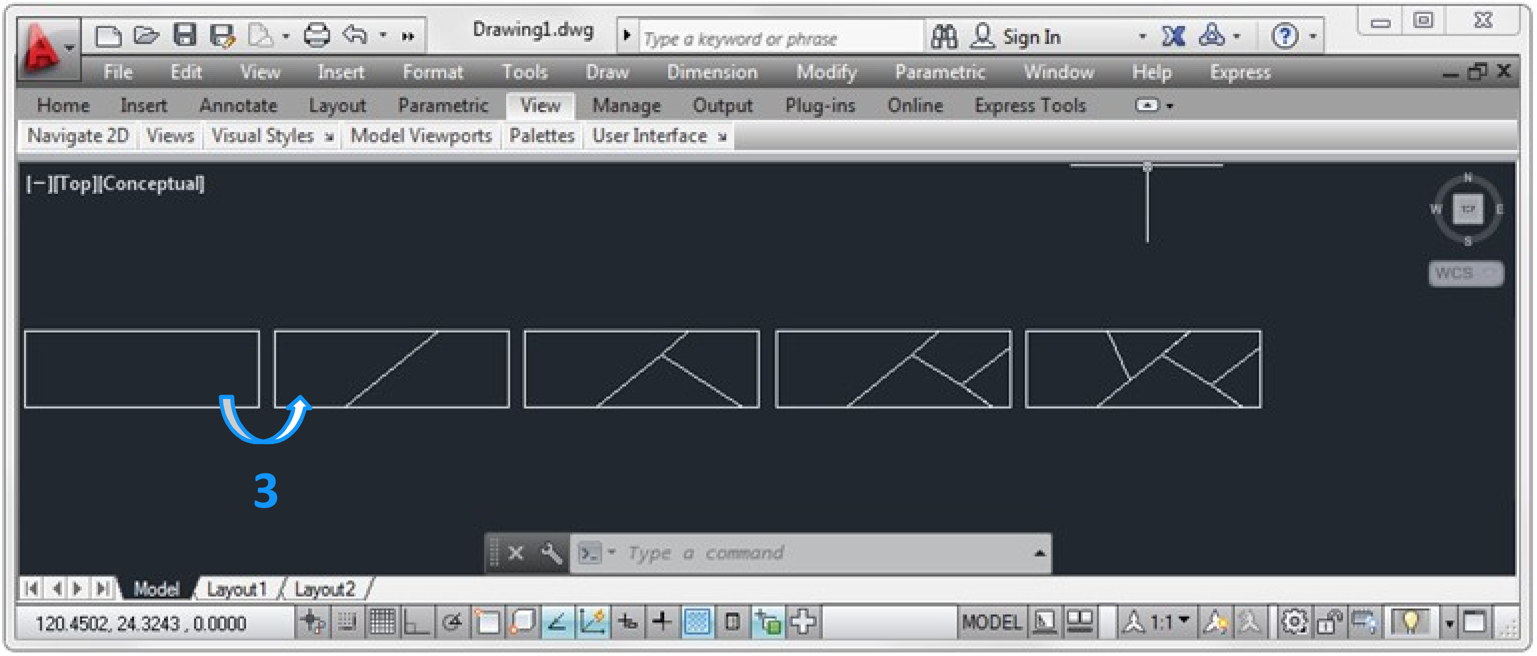


(3)

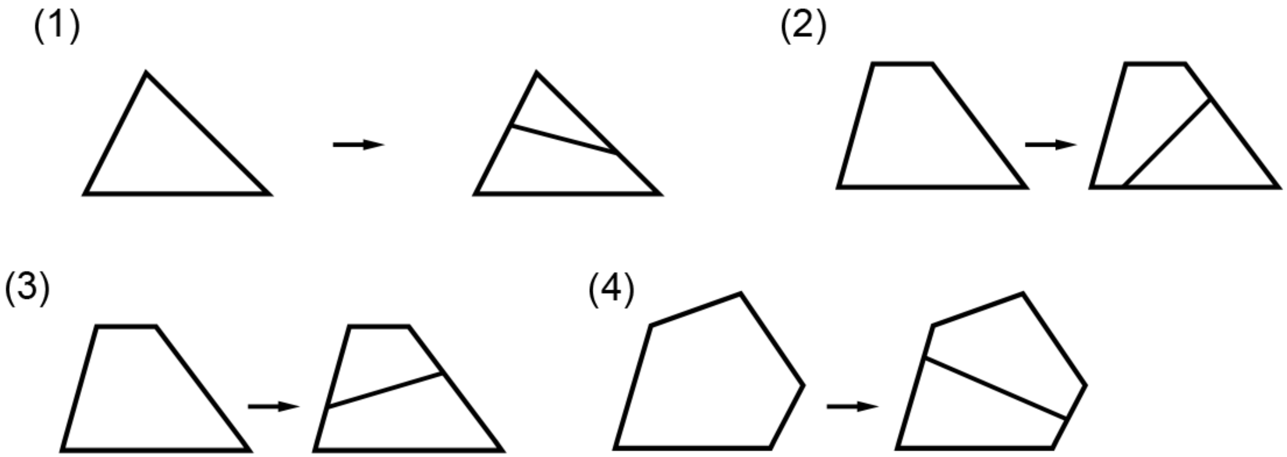


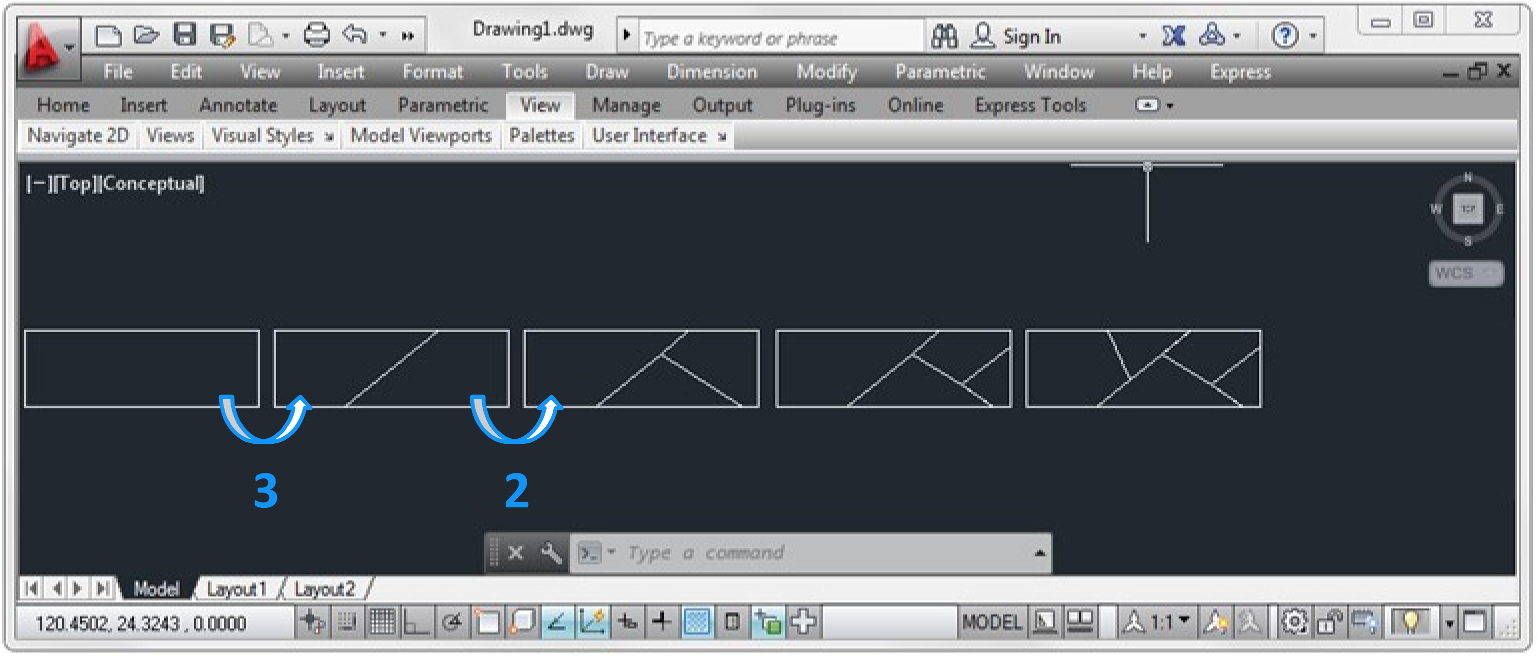
(4)



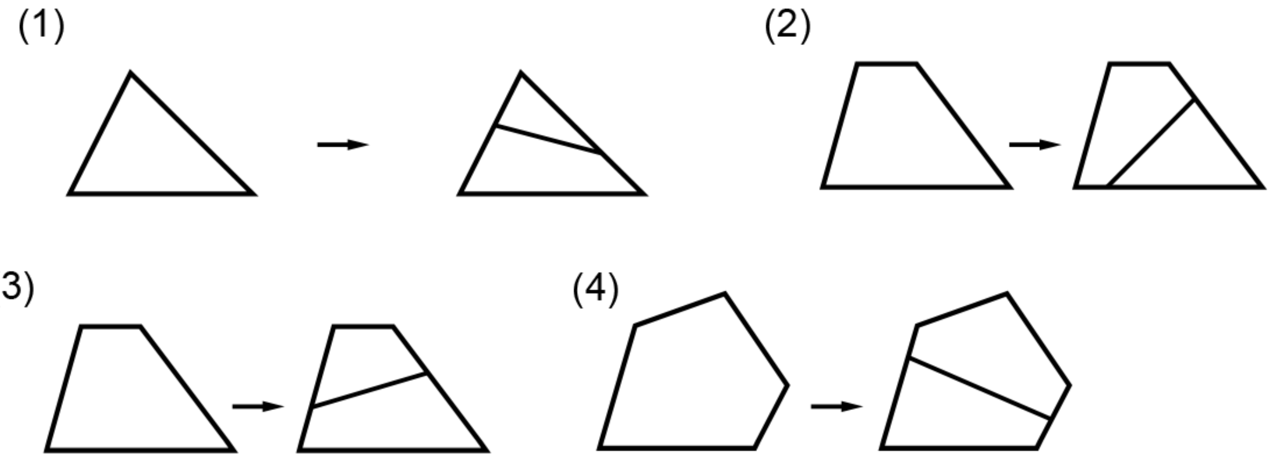

Drawing1.dwg
Type a keyword or phrase
Sign In
File Edit View Insert Format Tools Draw Dimension Modify Parametric Window Help Express
Home Insert Annotate Layout Parametric View Manage Output Plug-ins Online Express Tools
Navigate 2D Views Visual Styles Model Viewports Palettes User Interface
[−][Top][Conceptual]
N
W
E
S
TOP
WCS
3
Type a command
Model Layout1 Layout2
120.4502, 24.3243 , 0.0000
MODEL
1:1

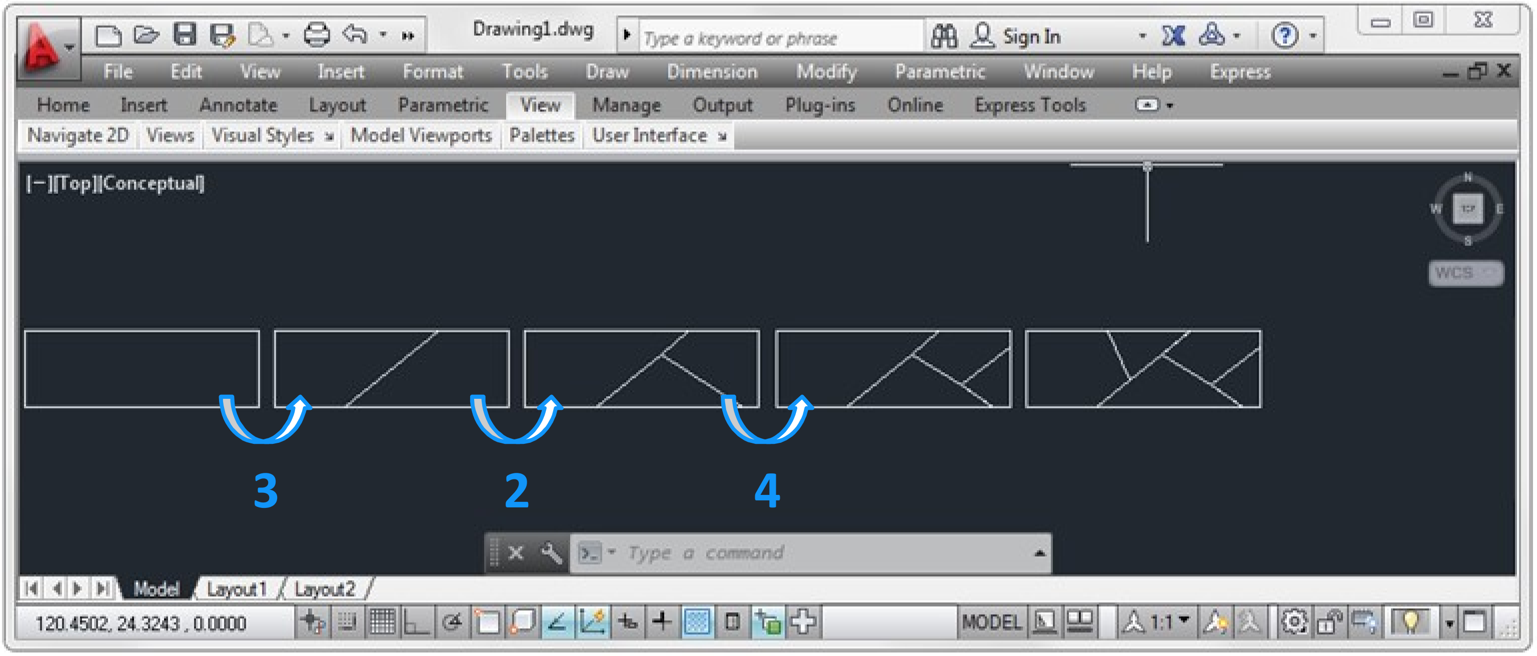
Ice-ray


(1)
(2)
(3)
(4)


Drawing1.dwg
Type a keyword or phrase
Sign In
File Edit View Insert Format Tools Draw Dimension Modify Parametric Window Help Express
Home Insert Annotate Layout Parametric View Manage Output Plug-ins Online Express Tools
Navigate 2D Views Visual Styles Model Viewports Palettes User Interface
[−][Top][Conceptual]
WCS
3
2
Type a command
Model Layout1 Layout2
120.4502, 24.3243 , 0.0000
MODEL
1:1

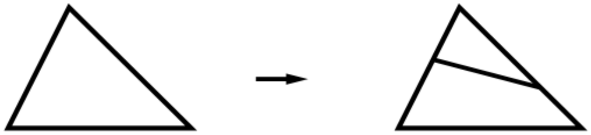
Ice-ray


(1)
(2)
(3)
(4)

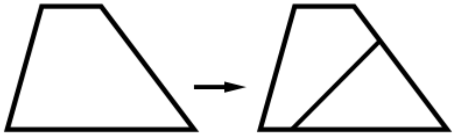

Drawing1.dwg
Type a keyword or phrase
Sign In
File
Edit
View
Insert
Format
Tools
Draw
Dimension
Modify
Parametric
Window
Help
Express
Home
Insert
Annotate
Layout
Parametric
View
Manage
Output
Plug-ins
Online
Express Tools
Navigate 2D
Views
Visual Styles
Model Viewports
Palettes
User Interface
[−][Top][Conceptual]
WCS
3
2
4
Type a command
Model
Layout1
Layout2
120.4502, 24.3243 , 0.0000
MODEL
1:1

Ice-ray

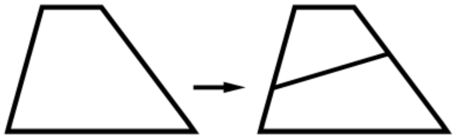
(1)



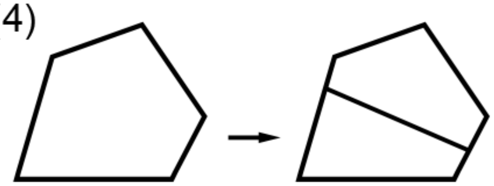
(2)

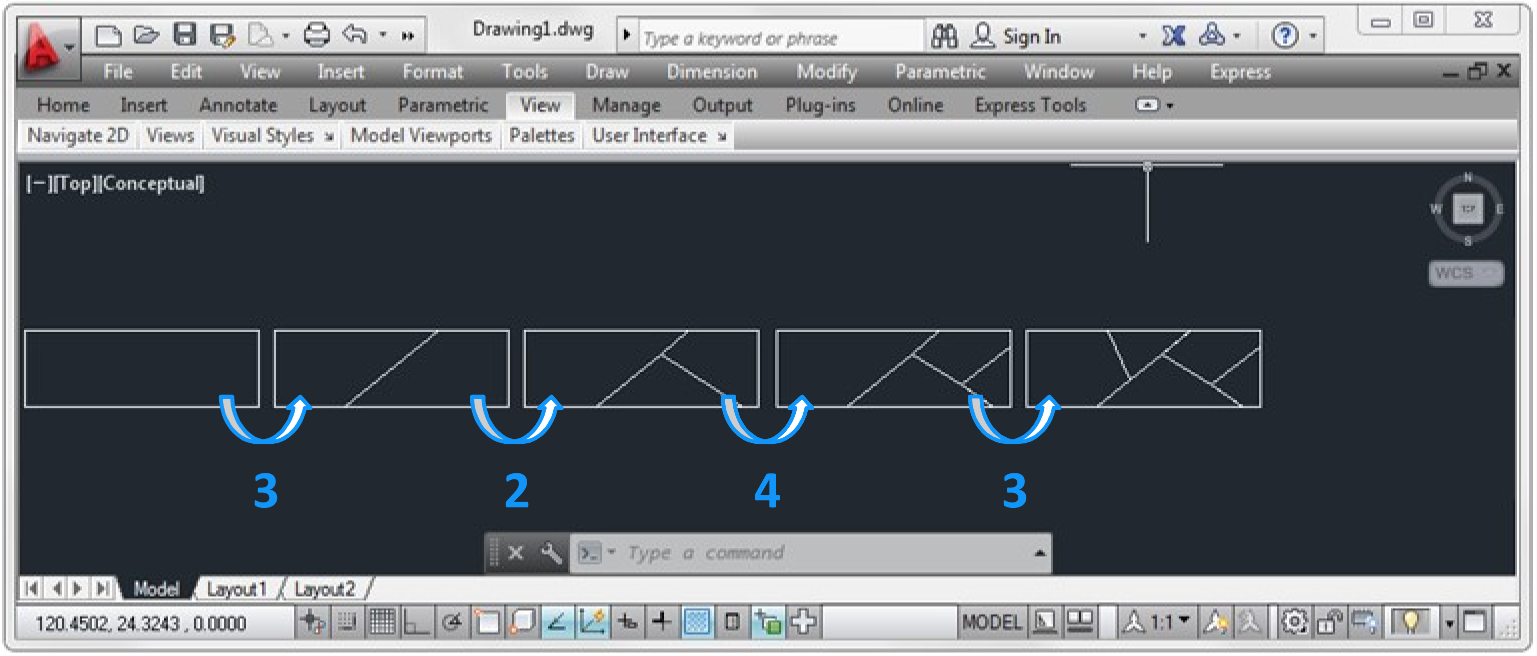


(3)



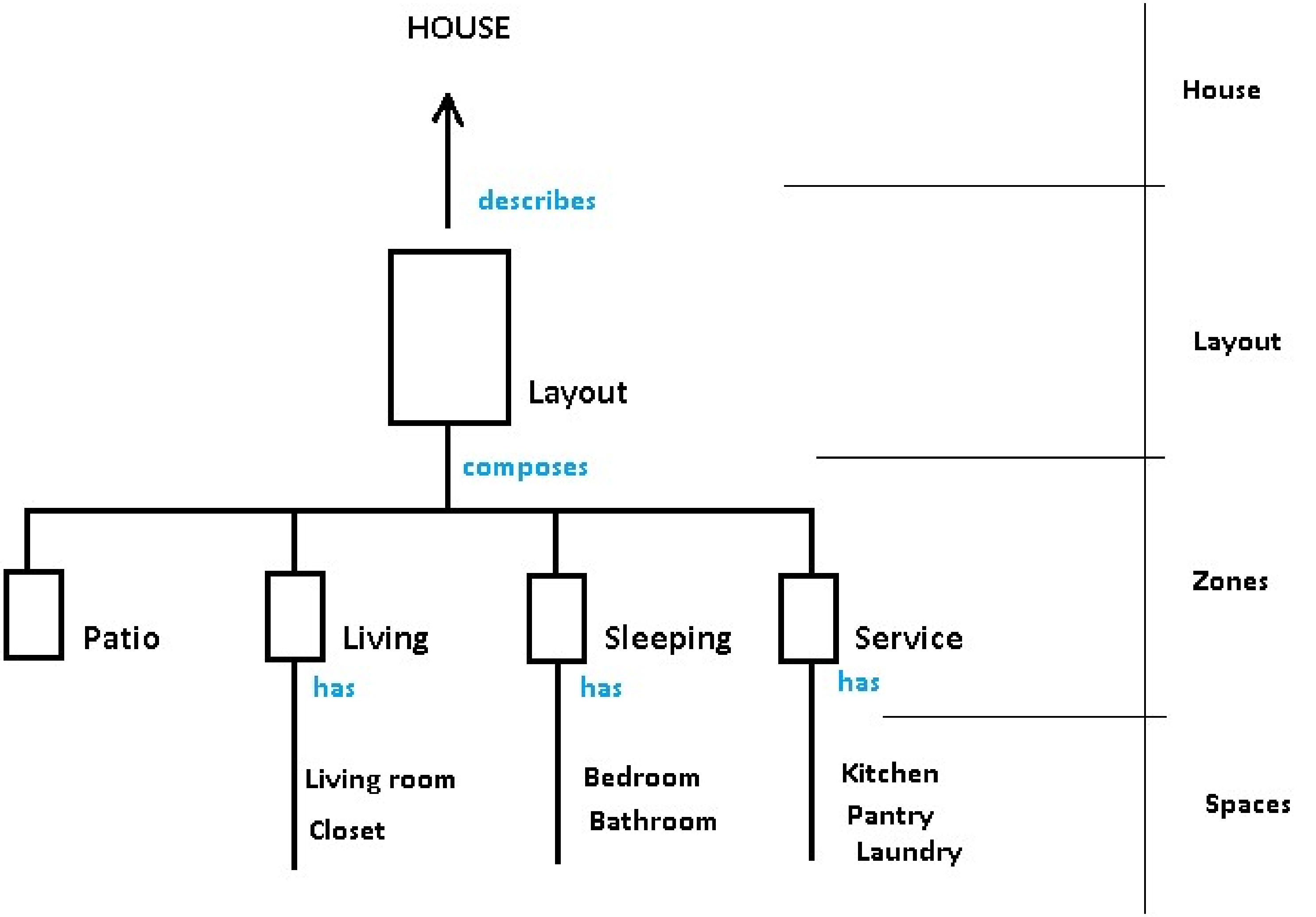
(4)




Drawing1.dwg
Type a keyword or phrase
Sign In
File
Edit
View
Insert
Format
Tools
Draw
Dimension
Modify
Parametric
Window
Help
Express
Home
Insert
Annotate
Layout
Parametric
View
Manage
Output
Plug-ins
Online
Express Tools
Navigate 2D
Views
Visual Styles
Model Viewports
Palettes
User Interface
[−][Top][Conceptual]
WCS
3
2
4
3
Type a command
Model
Layout1
Layout2
120.4502, 24.3243 , 0.0000
MODEL
1:1

Malagueira

Malagueira



HOUSE

describes

Layout

composes

Patio

Living

has

Living room

Closet

Sleeping

has

Bedroom

Bathroom

Service

has

Kitchen

Pantry

Laundry

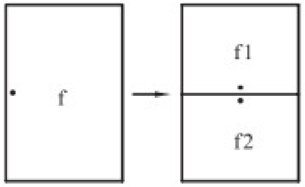
House

Layout

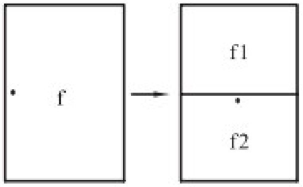
Zones

Spaces

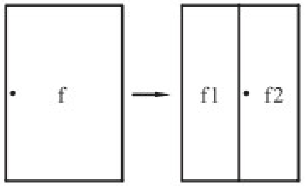
Malagueira



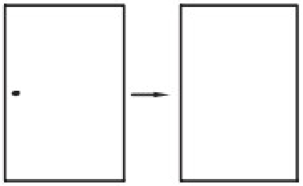
Rule A



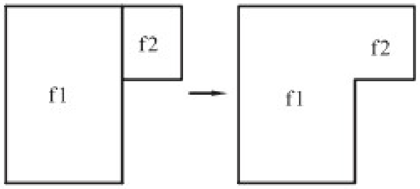
Rule B



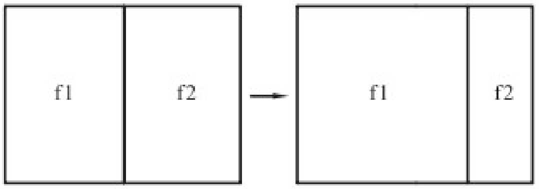
Rule C



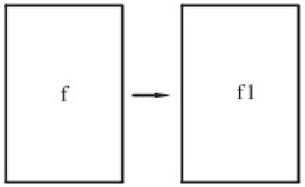
Rule D



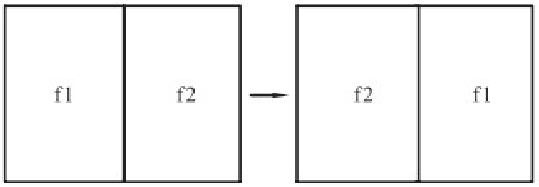
Rule E



Rule F

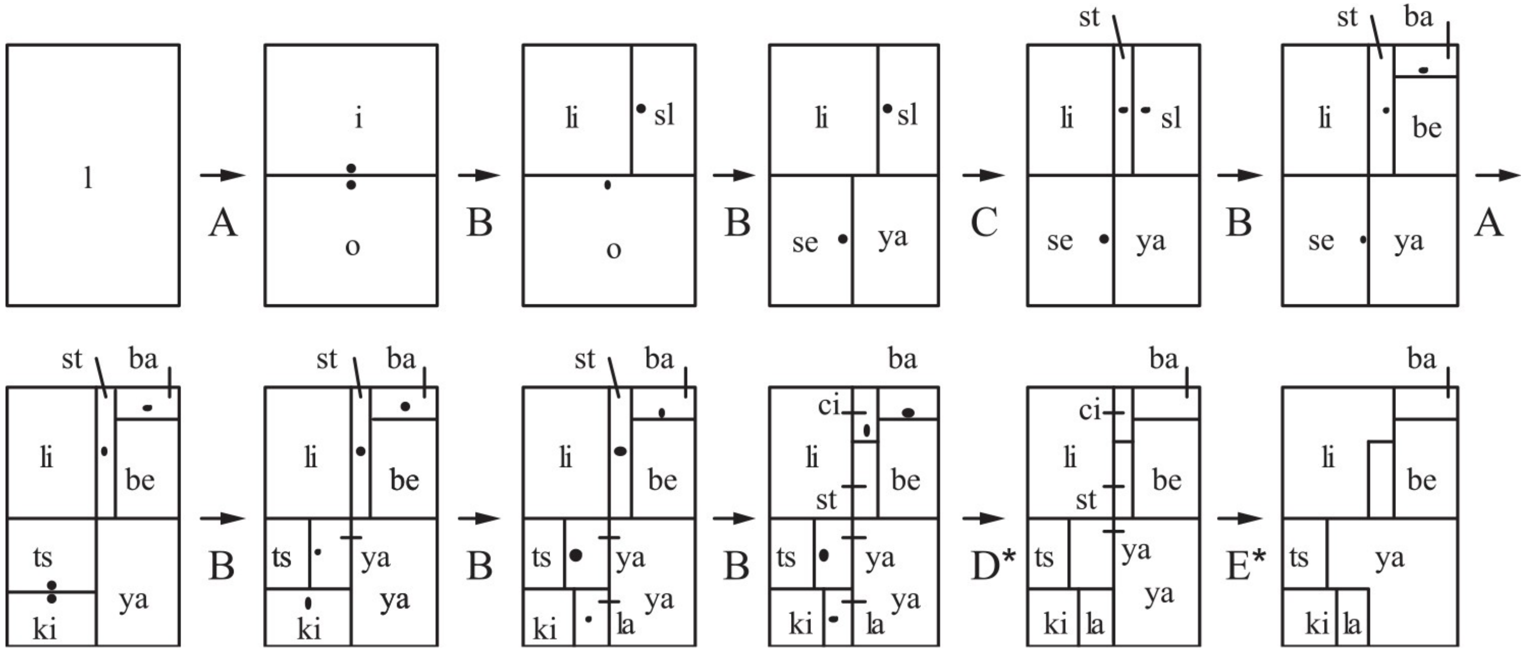


Rule G

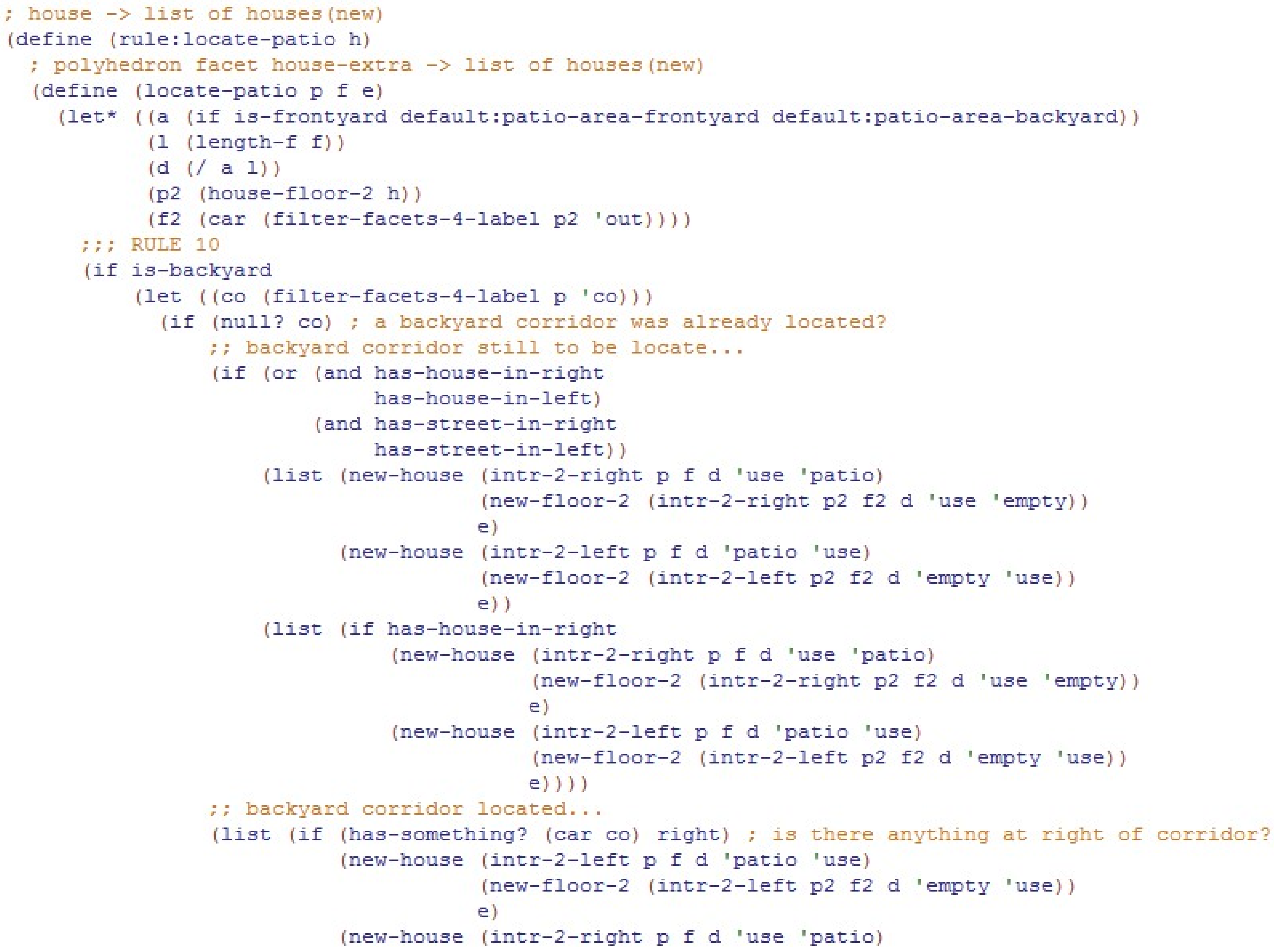


Rule H

Malagueira

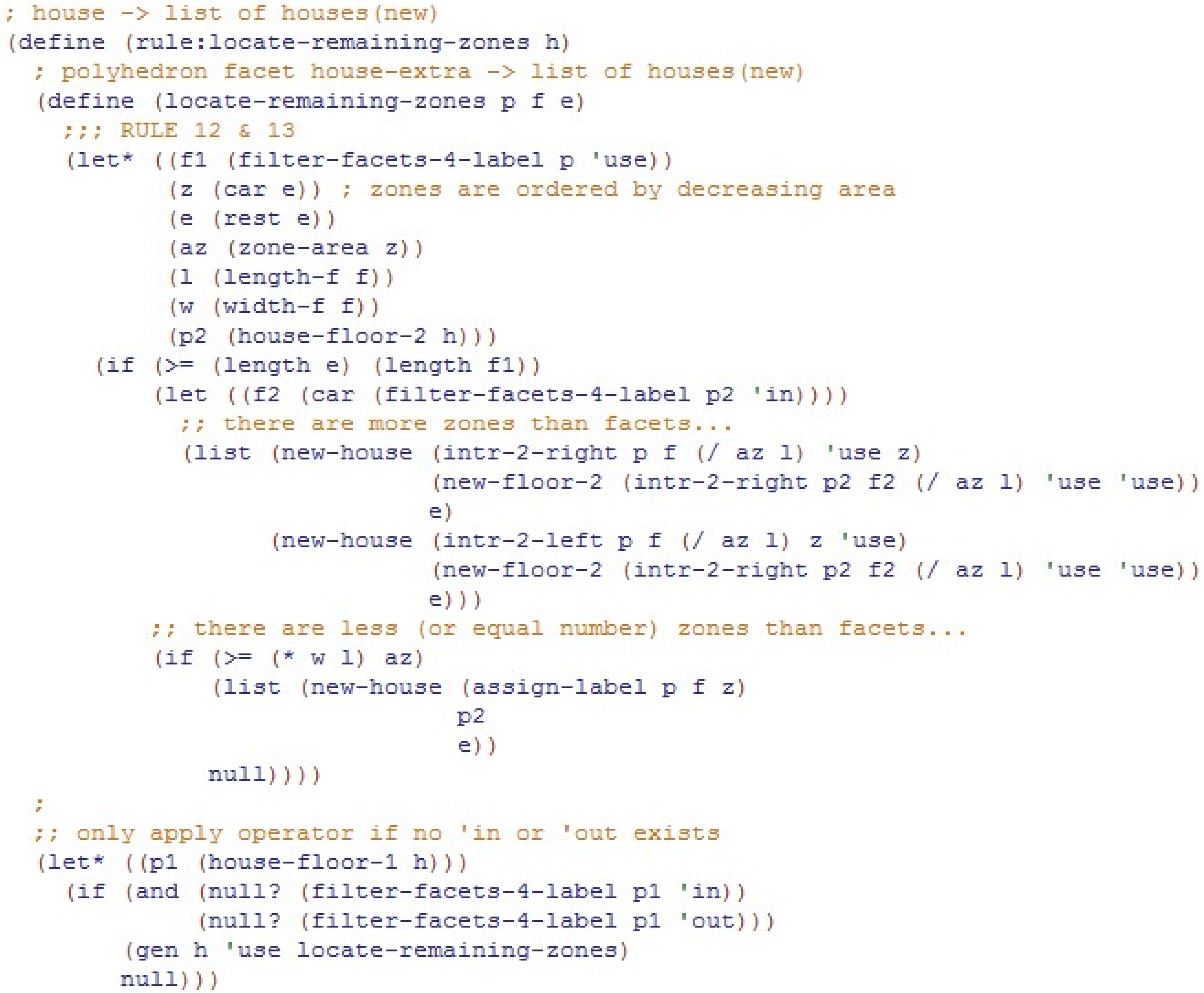

l
A
i
o
B
li
sl
o
B
li
sl
se
ya
C
st
li
sl
se
ya
B
st
ba
li
be
se
ya
A
st
ba
li
be
ts
ki
ya
B
st
ba
li
be
ts
ya
ki
ya
B
st
ba
li
be
ts
ya
ki
la
ya
B
ba
ci
li
st
be
ts
ya
ki
la
ya
D*
ba
ci
li
st
be
ts
ya
ki
la
ya
E*
ba
li
be
ts
ya
ki
la

Malagueira



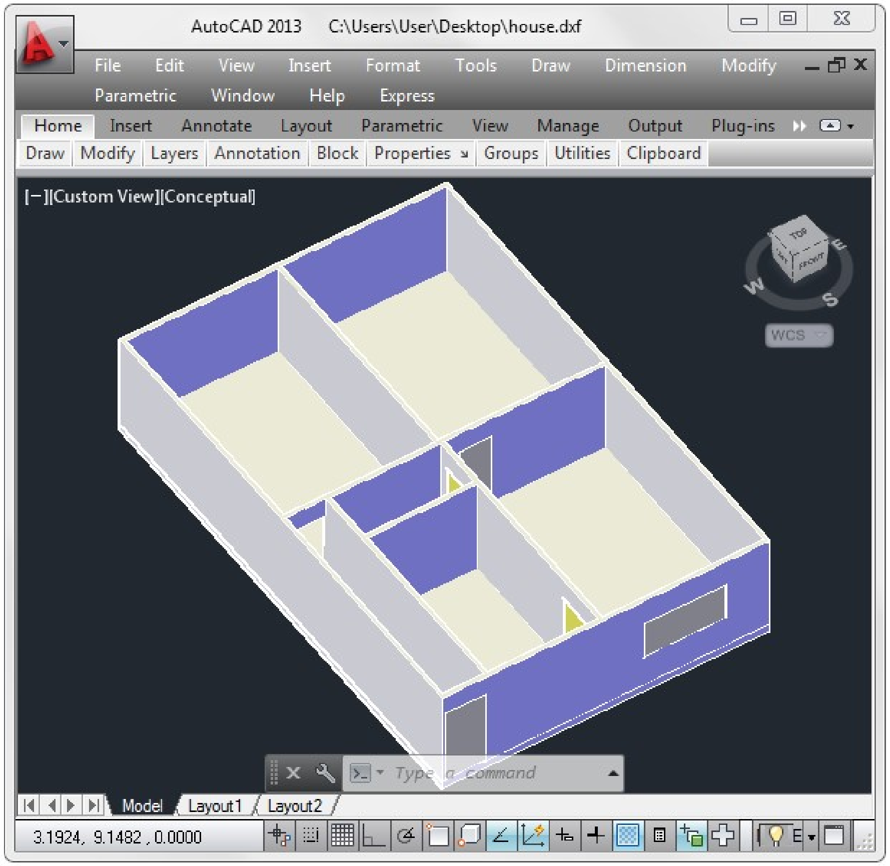
```scheme
; house -> list of houses(new)
(define (rule:locate-patio h)
  ; polyhedron facet house-extra -> list of houses(new)
  (define (locate-patio p f e)
    (let* ((a (if is-frontyard default:patio-area-frontyard default:patio-area-backyard))
           (l (length-f f))
           (d (/ a l))
           (p2 (house-floor-2 h))
           (f2 (car (filter-facets-4-label p2 'out))))
      ;;; RULE 10
      (if is-backyard
          (let ((co (filter-facets-4-label p 'co)))
            (if (null? co) ; a backyard corridor was already located?
                ;; backyard corridor still to be locate...
                (if (or (and has-house-in-right
                             has-house-in-left)
                        (and has-street-in-right
                             has-street-in-left))
                    (list (new-house (intr-2-right p f d 'use 'patio)
                                     (new-floor-2 (intr-2-right p2 f2 d 'use 'empty))
                                     e)
                          (new-house (intr-2-left p f d 'patio 'use)
                                     (new-floor-2 (intr-2-left p2 f2 d 'empty 'use))
                                     e))
                    (list (if has-house-in-right
                              (new-house (intr-2-right p f d 'use 'patio)
                                         (new-floor-2 (intr-2-right p2 f2 d 'use 'empty))
                                         e)
                              (new-house (intr-2-left p f d 'patio 'use)
                                         (new-floor-2 (intr-2-left p2 f2 d 'empty 'use))
                                         e))))
                ;; backyard corridor located...
                (list (if (has-something? (car co) right) ; is there anything at right of corridor?
                          (new-house (intr-2-left p f d 'patio 'use)
                                     (new-floor-2 (intr-2-left p2 f2 d 'empty 'use))
                                     e)
                          (new-house (intr-2-right p f d 'use 'patio)
```

Malagueira



```
; house -> list of houses(new)
(define (rule:locate-remaining-zones h)
  ; polyhedron facet house-extra -> list of houses(new)
  (define (locate-remaining-zones p f e)
    ;;; RULE 12 & 13
    (let* ((f1 (filter-facets-4-label p 'use))
           (z (car e)) ; zones are ordered by decreasing area
           (e (rest e))
           (az (zone-area z))
           (l (length-f f))
           (w (width-f f))
           (p2 (house-floor-2 h)))
      (if (>= (length e) (length f1))
          (let ((f2 (car (filter-facets-4-label p2 'in))))
            ;; there are more zones than facets...
            (list (new-house (intr-2-right p f (/ az l)) 'use z)
                             (new-floor-2 (intr-2-right p2 f2 (/ az l) 'use 'use))
                             e)
                  (new-house (intr-2-left p f (/ az l) z 'use)
                             (new-floor-2 (intr-2-right p2 f2 (/ az l) 'use 'use))
                             e)))
          ;; there are less (or equal number) zones than facets...
          (if (>= (* w l) az)
              (list (new-house (assign-label p f z)
                               p2
                               e))
              null))))
  ;
  ;; only apply operator if no 'in or 'out exists
  (let* ((p1 (house-floor-1 h)))
    (if (and (null? (filter-facets-4-label p1 'in))
             (null? (filter-facets-4-label p1 'out)))
        (gen h 'use locate-remaining-zones)
        null)))
```

Malagueira

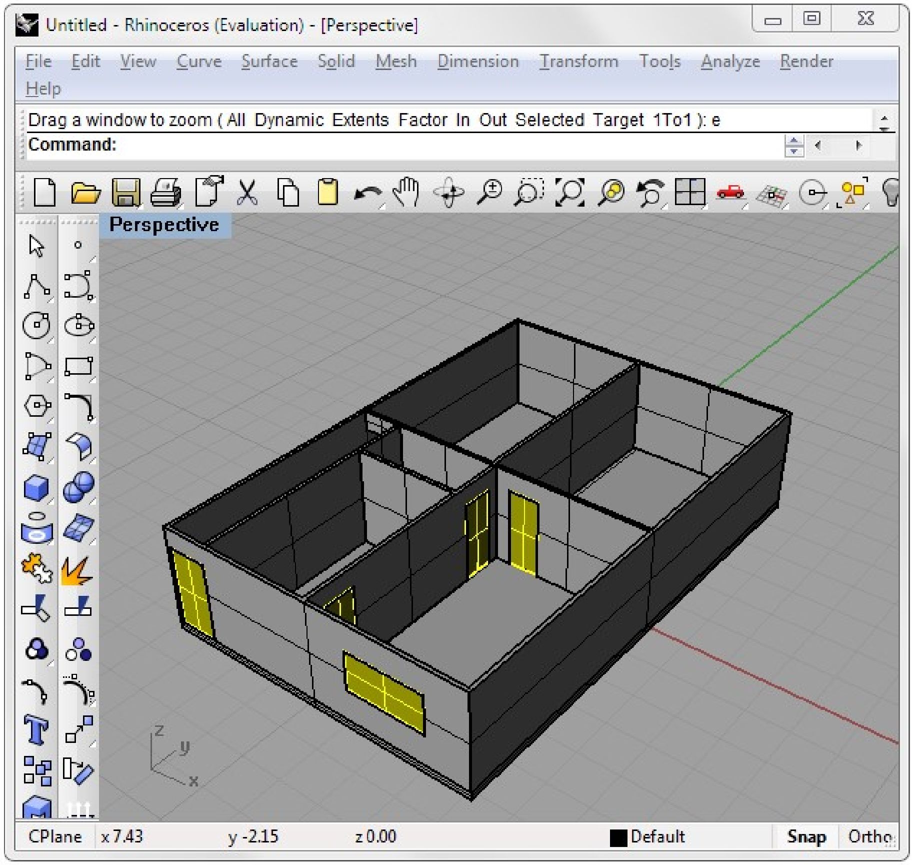


AutoCAD 2013 C:\Users\User\Desktop\house.dxf

[−][Custom View][Conceptual]

3.1924, 9.1482 , 0.0000

File Edit View Insert Format Tools Draw Dimension Modify
Parametric Window Help Express
Home Insert Annotate Layout Parametric View Manage Output Plug-ins
Draw Modify Layers Annotation Block Properties Groups Utilities Clipboard
TOP
FRONT
W
E
S
WCS
Type a command
Model
Layout1
Layout2



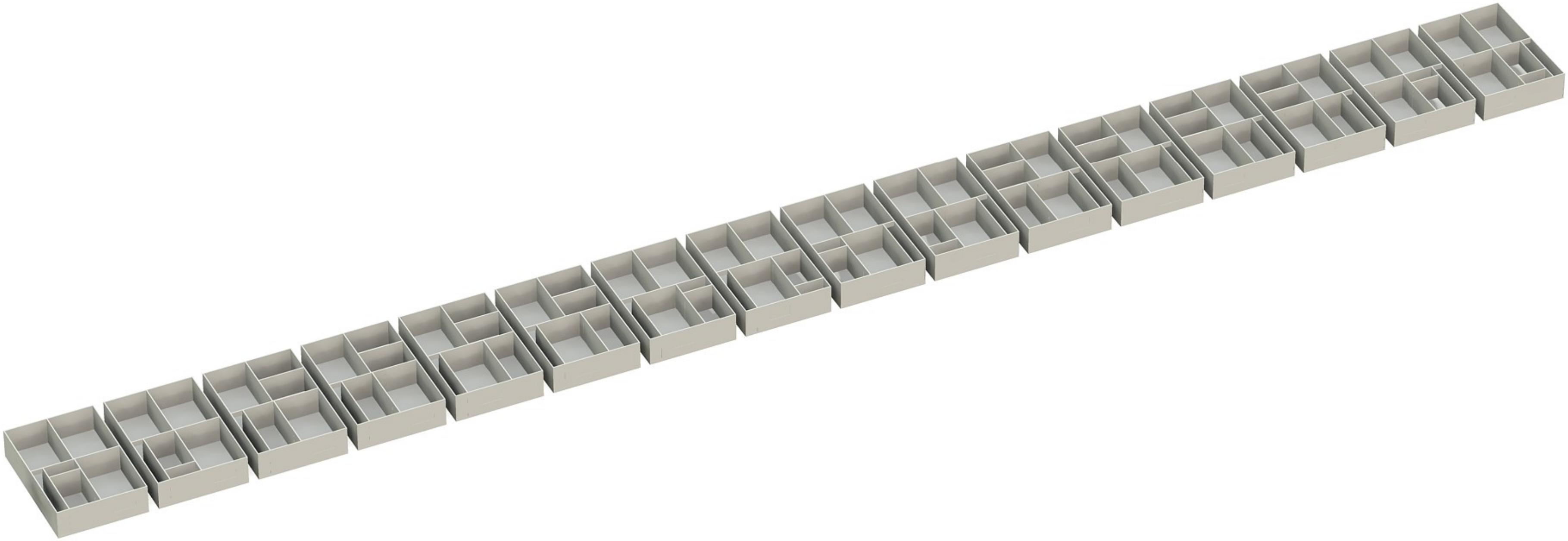
Untitled - Rhinoceros (Evaluation) - [Perspective]

```
Drag a window to zoom ( All  Dynamic  Extents  Factor  In  Out  Selected  Target  1To1 ): e
Command:
```

CPlane x 7.43 y -2.15 z 0.00 Default Snap Ortho

File Edit View Curve Surface Solid Mesh Dimension Transform Tools Analyze Render
Help
Perspective
z
y
x

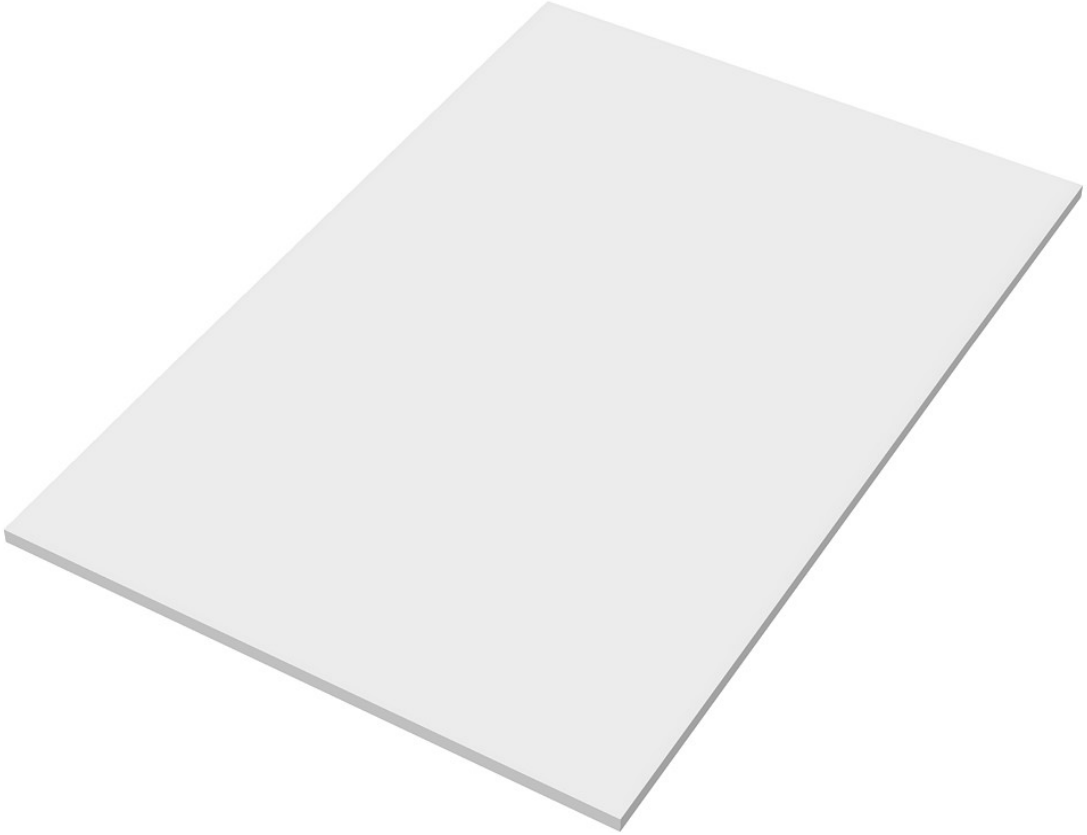
Malagueira



Malagueira

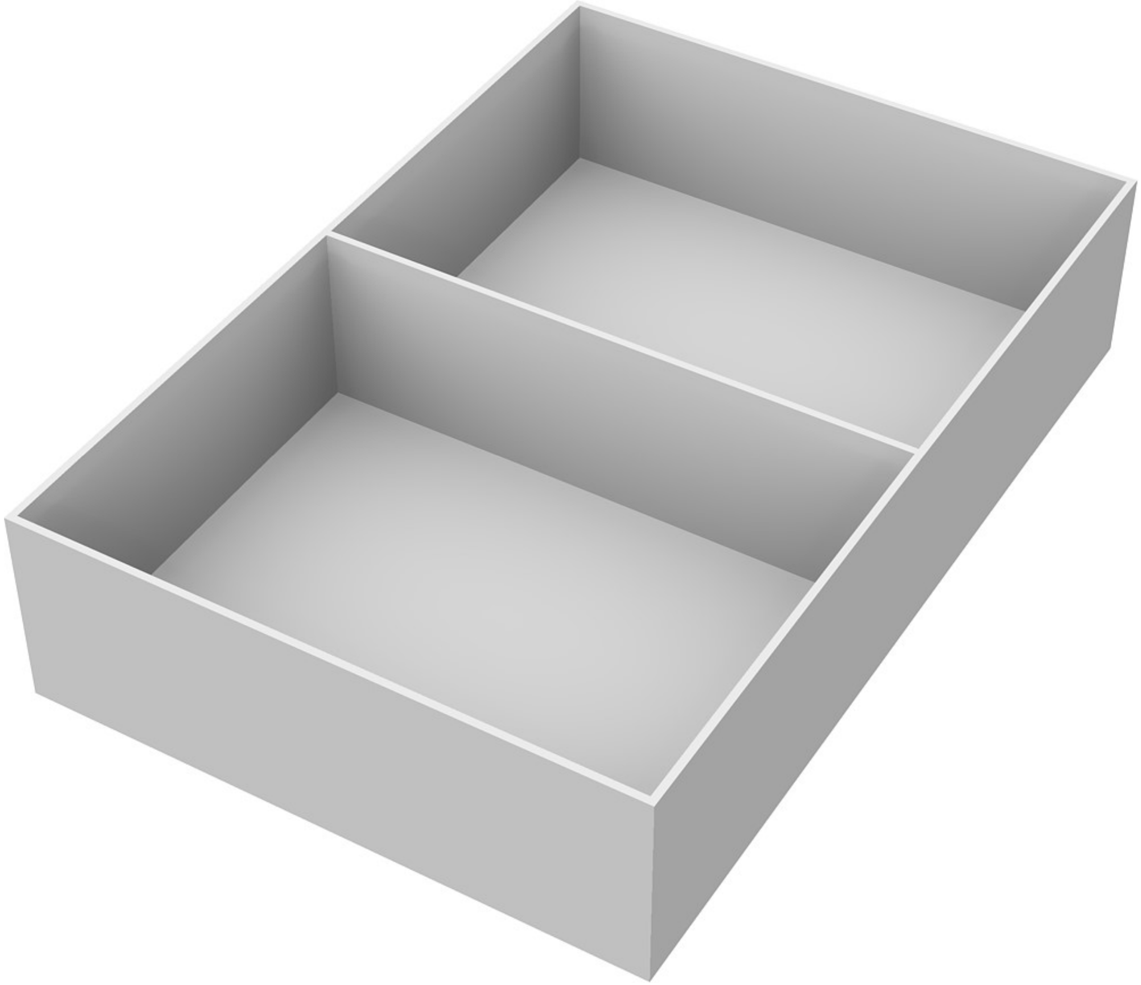
Desenho vazio

Malagueira



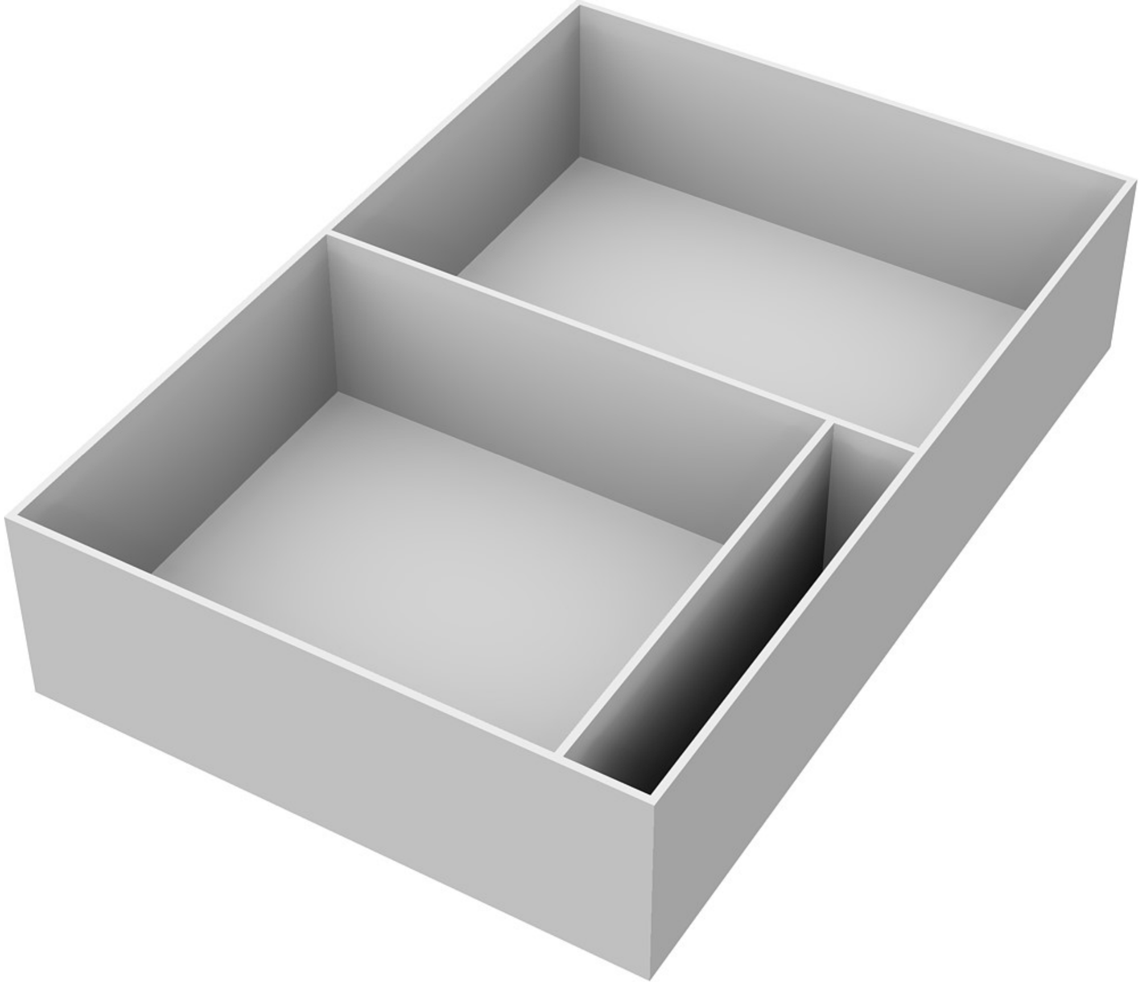
Introduzir laje

Malagueira



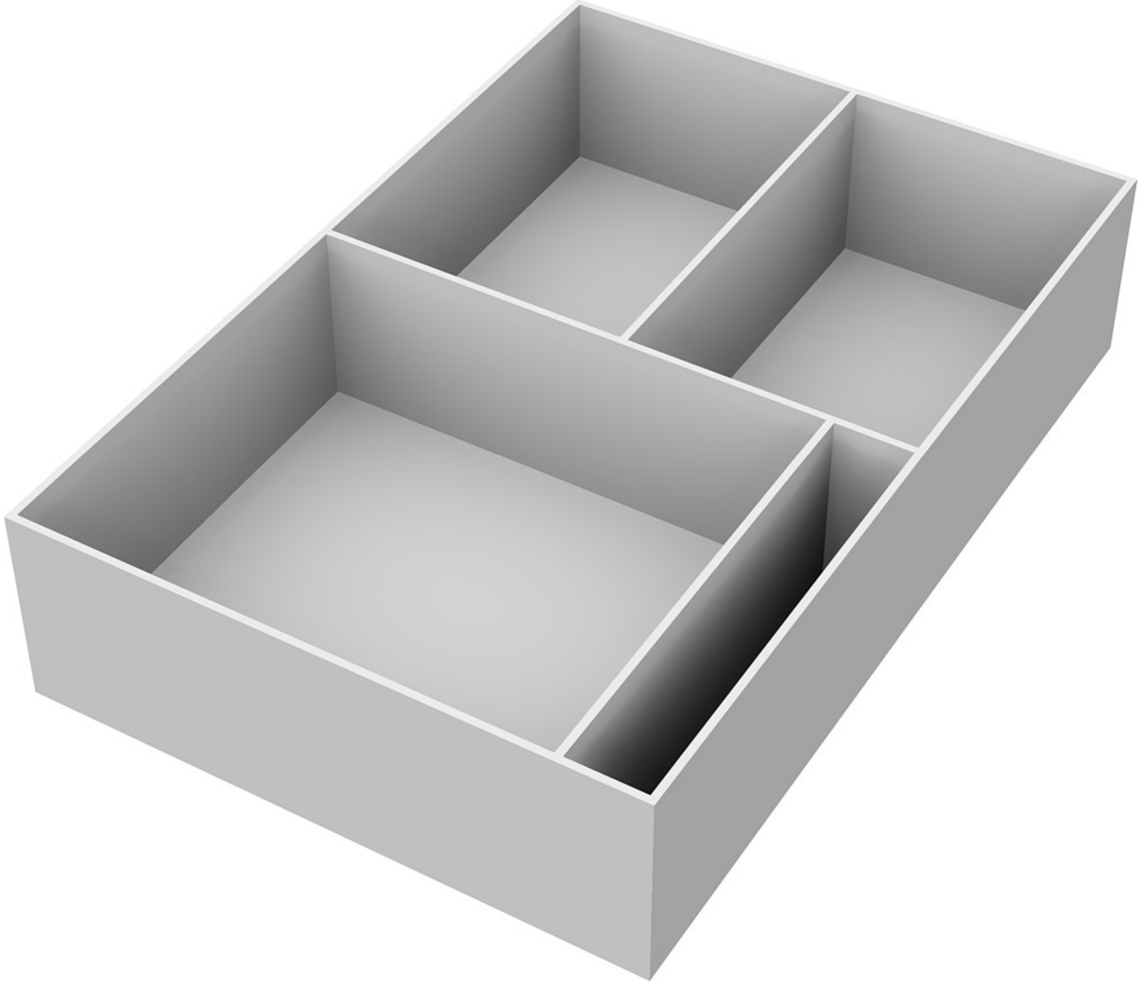
Subir paredes e 1ª divisão

Malagueira



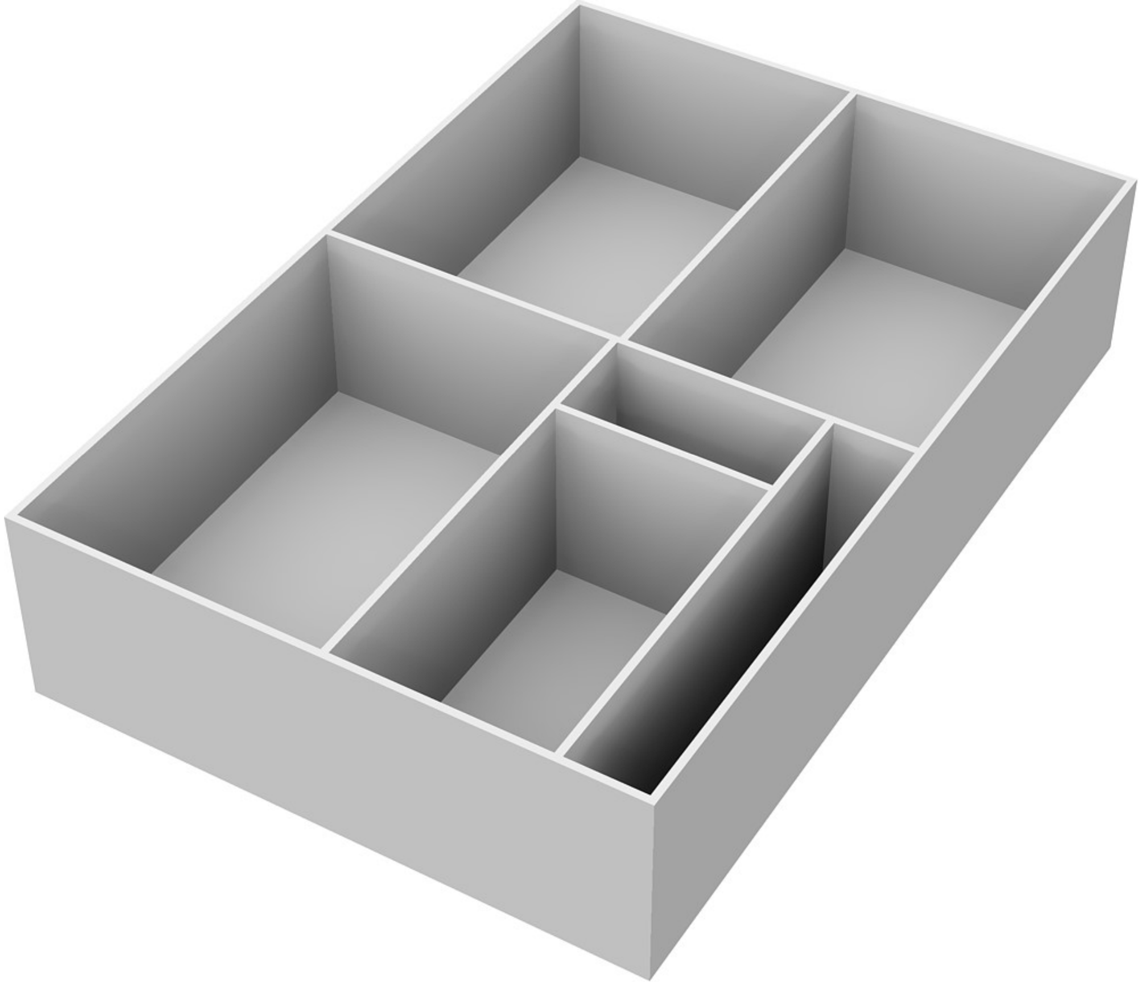
Localizar corredor para pátio traseiro

Malagueira



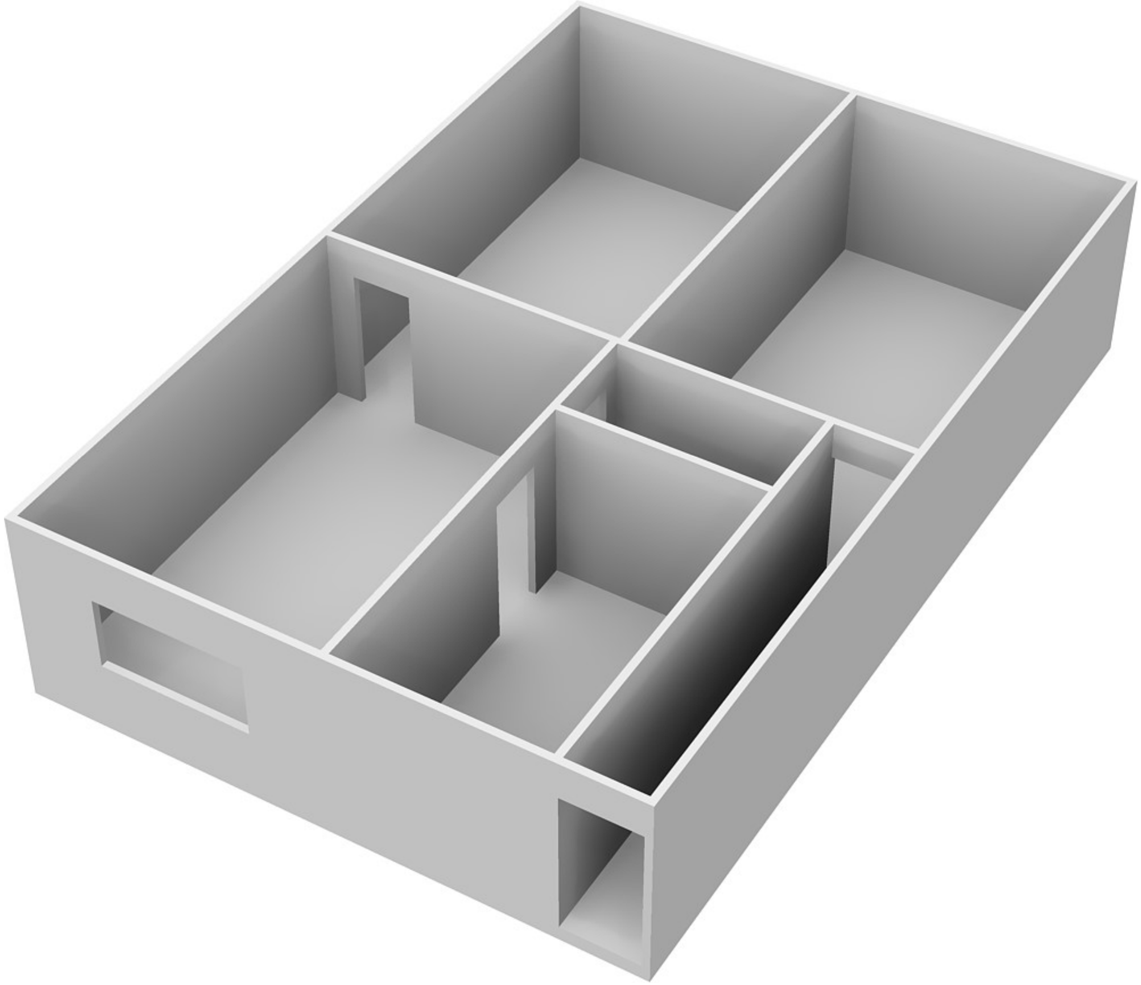
Localizar restantes áreas funcionais

Malagueira



Localizar espaços

Malagueira



Introduzir pormenores

Obrigado

Perguntas?