Spatial Inference –
Learning vs. Constraint Solving

Carsten Gips, Petra Hofstedt, Fritz Wysotzki

{cagi,ph,wysotzki}@cs.tu-berlin.de.

Berlin University of Technology

Spatial Relations

Constraint Solving

Machine Learning

Introduction.

- Starting point: textual descriptions of spatial scenes.

Introduction.

- Starting point: textual descriptions of spatial scenes.
- Aims:

Introduction.

- Starting point: textual descriptions of spatial scenes.
- Aims:
  - Checking the texts for consistency.

Introduction.

- Starting point: textual descriptions of spatial scenes.
- Aims:
  - Checking the texts for consistency.
  - Generating appropriate depictions.

Introduction.

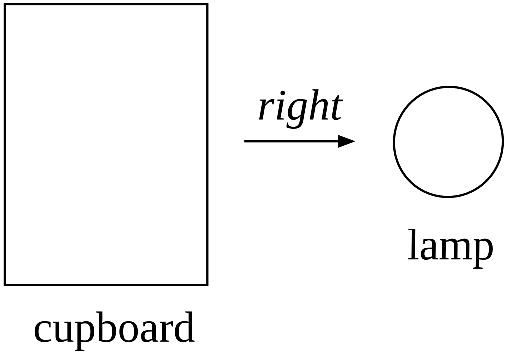
- Starting point: textual descriptions of spatial scenes.
- Aims:
  - Checking the texts for consistency.
  - Generating appropriate depictions.
  - Inferring new information (not initially given in the texts).

Introduction.

- Starting point: textual descriptions of spatial scenes.
- Aims:
  - Checking the texts for consistency.
  - Generating appropriate depictions.
  - Inferring new information (not initially given in the texts).
- Here: consider only depiction generation.

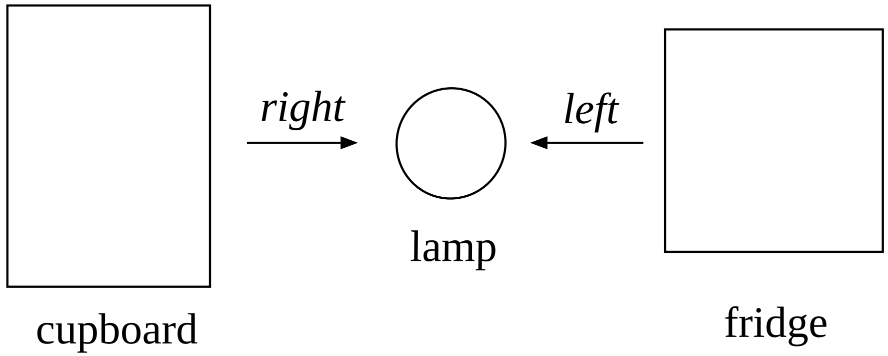
Introduction: Spatial Relations.

(1) (2) (3)



`right(cupboard,lamp)`

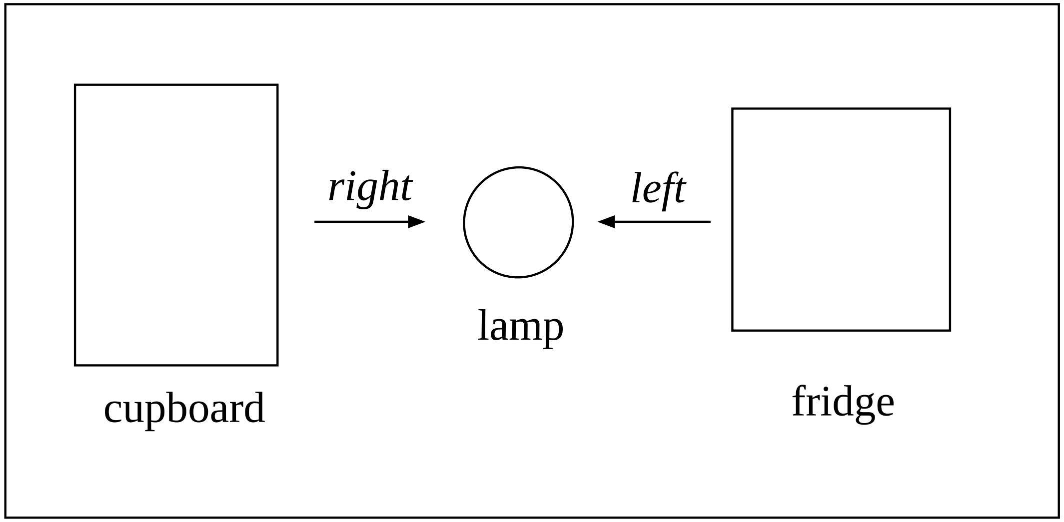
Introduction: Spatial Relations.




```
right(cupboard,lamp)
left(fridge,lamp)
```

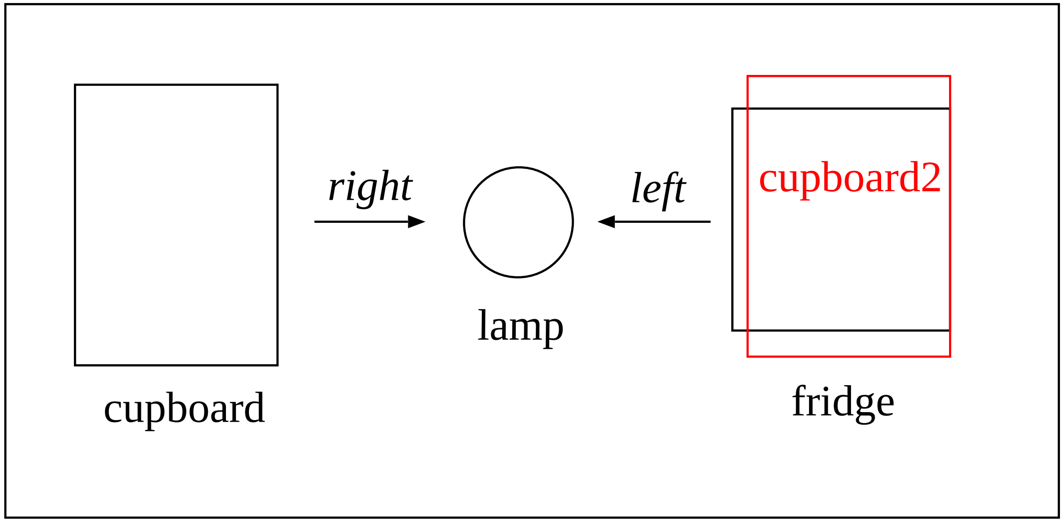
Introduction: Spatial Relations.

(1) (2) (3)



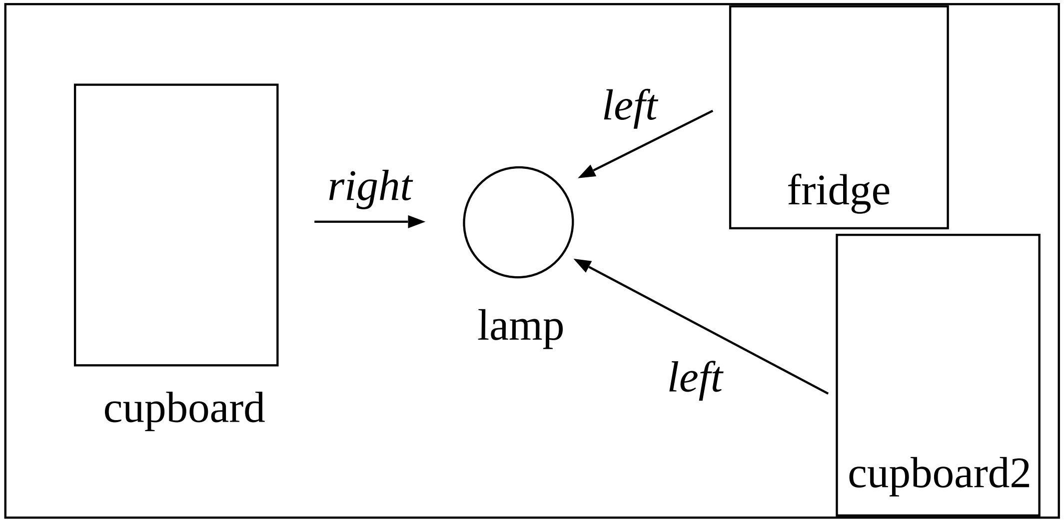

```
right(cupboard,lamp)
left(fridge,lamp)
in_room(fridge),in_room(lamp),in_room(cupboard)
```

Introduction: Spatial Relations.




```
right(cupboard,lamp)
left(fridge,lamp)
in_room(fridge),in_room(lamp),in_room(cupboard)
left(cupboard2,lamp),in_room(cupboard2)
```

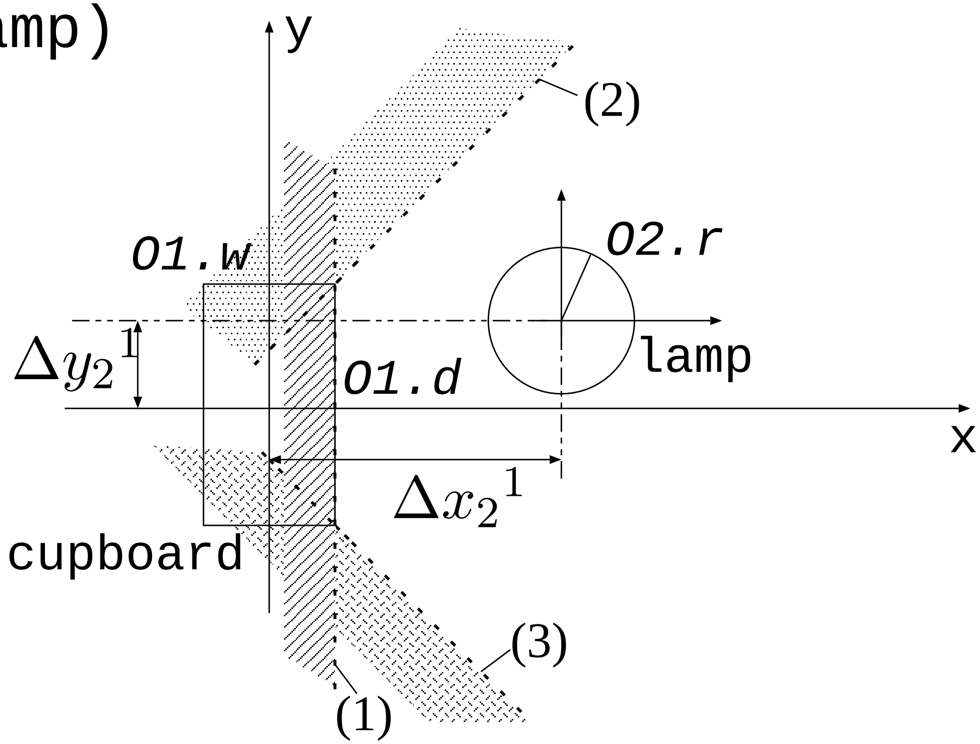
Introduction: Spatial Relations.




```
right(cupboard,lamp)
left(fridge,lamp)
in_room(fridge),in_room(lamp),in_room(cupboard)
left(cupboard2,lamp),in_room(cupboard2)
```

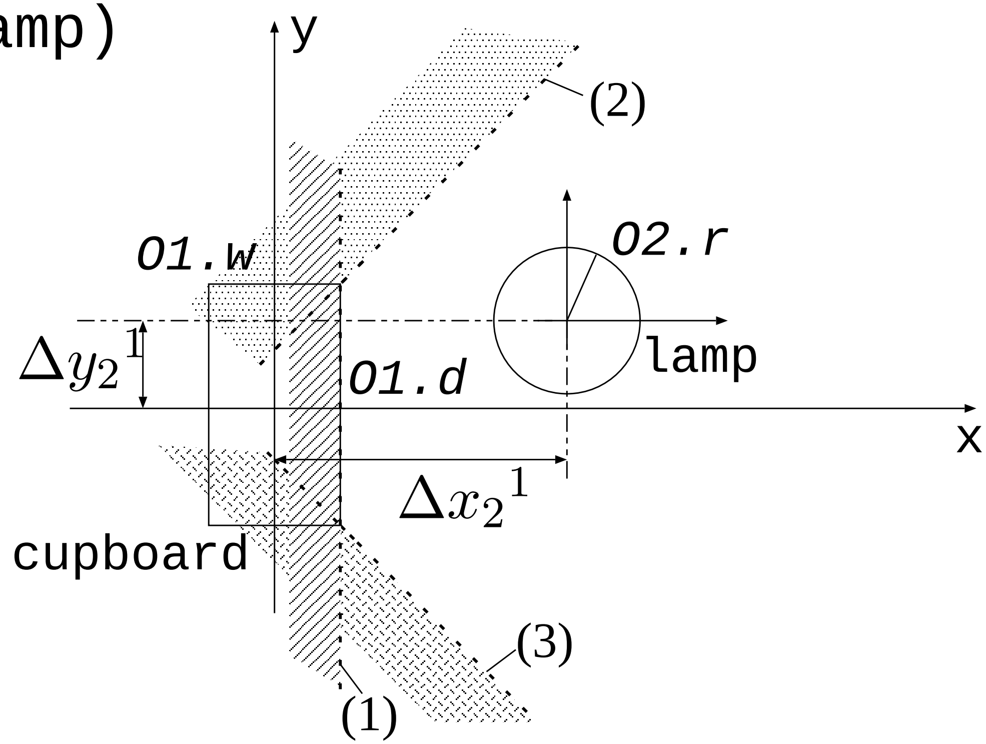
Introduction: Spatial Relations. A Metric Approach. **(1) (2) (3)**

```
right(cupboard, lamp)
```



Introduction: Spatial Relations. A Metric Approach. **(1) (2) (3)**

`right(cupboard, lamp)`



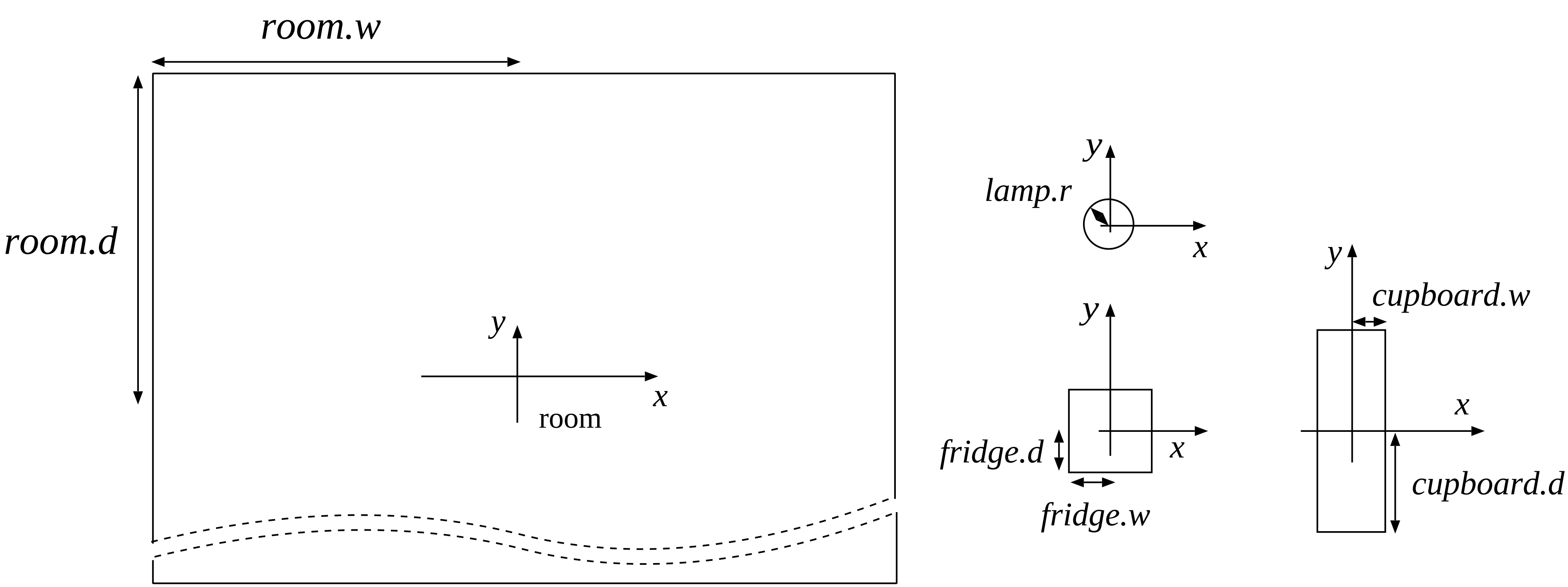
Mathematical description: `right(`$O1$`, `$O2$`)`

$$\Delta x_2{}^1 \geq O1.w + O2.r \qquad (1)$$

$$\Delta x_2{}^1 \geq \Delta y_2{}^1 + O1.w - O1.d + \sqrt{2}O2.r \qquad (2)$$

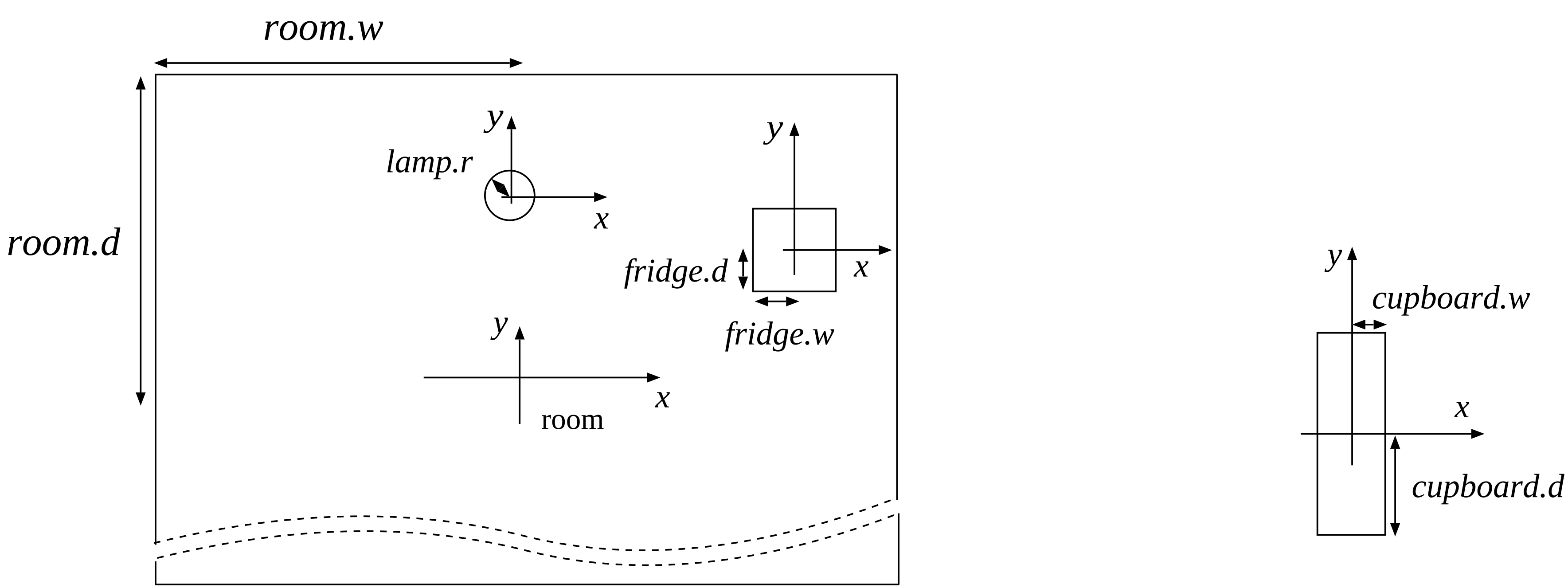
$$\Delta x_2{}^1 \geq -\Delta y_2{}^1 + O1.w - O1.d + \sqrt{2}O2.r \qquad (3)$$

Introduction: Spatial Relations. Example Scene. **(1) (2) (3)**



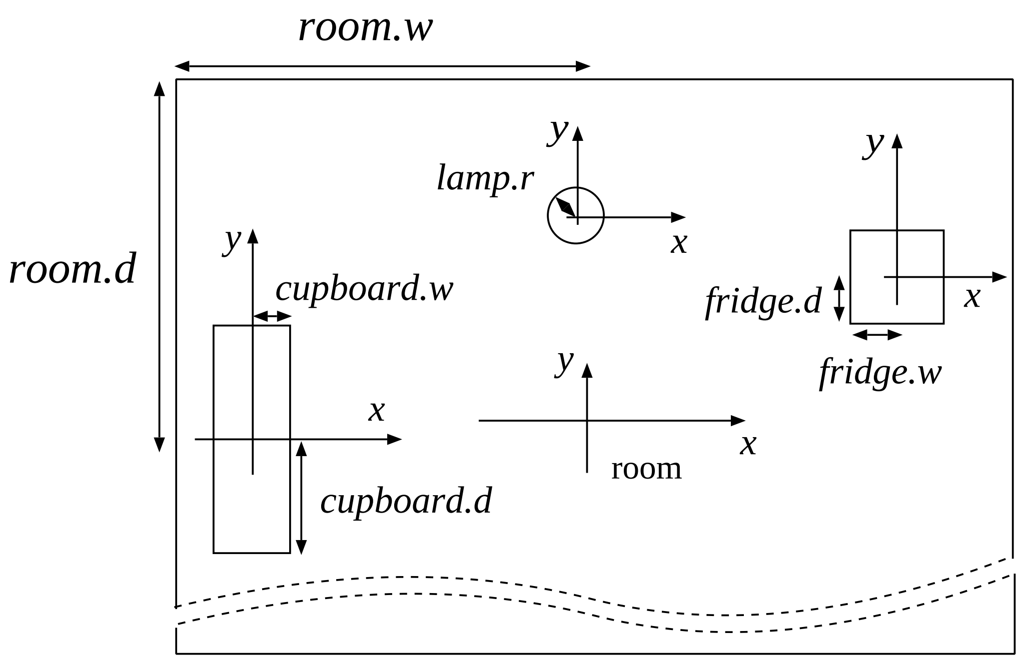

```
room.w∈[4.0,4.5],room.d∈[4.0,4.5],lamp.r=0.3,
fridge.w∈[0.4,0.5],fridge.d∈[0.4,0.5],
cupboard.w=0.4, cupboard.d∈[1.0,1.2]
in_room(fridge),...,not_overlap(fridge,lamp,...),
```

Introduction: Spatial Relations. Example Scene.



room.w∈[4.0,4.5],room.d∈[4.0,4.5],lamp.r=0.3,
fridge.w∈[0.4,0.5],fridge.d∈[0.4,0.5],
cupboard.w=0.4, cupboard.d∈[1.0,1.2]
in_room(fridge),...,not_overlap(fridge,lamp,...),
left(fridge,lamp),

Introduction: Spatial Relations. Example Scene.



room.w∈[4.0,4.5],room.d∈[4.0,4.5],lamp.r=0.3,
fridge.w∈[0.4,0.5],fridge.d∈[0.4,0.5],
cupboard.w=0.4, cupboard.d∈[1.0,1.2]
in_room(fridge),...,not_overlap(fridge,lamp,...),
left(fridge,lamp),right (cupboard, lamp)

Outline

- Constraint Solving:

Outline

- Constraint Solving:
  - Express spatial relations mathematically by constraints.

Outline

- Constraint Solving:
  - Express spatial relations mathematically by constraints.
  - Determine constraint domains.

Outline

- Constraint Solving:
  - Express spatial relations mathematically by constraints.
  - Determine constraint domains.
  - Solve constraints by cooperating constraint solvers.

Outline

- Constraint Solving:
  - Express spatial relations mathematically by constraints.
  - Determine constraint domains.
  - Solve constraints by cooperating constraint solvers.
- Machine Learning:

Outline

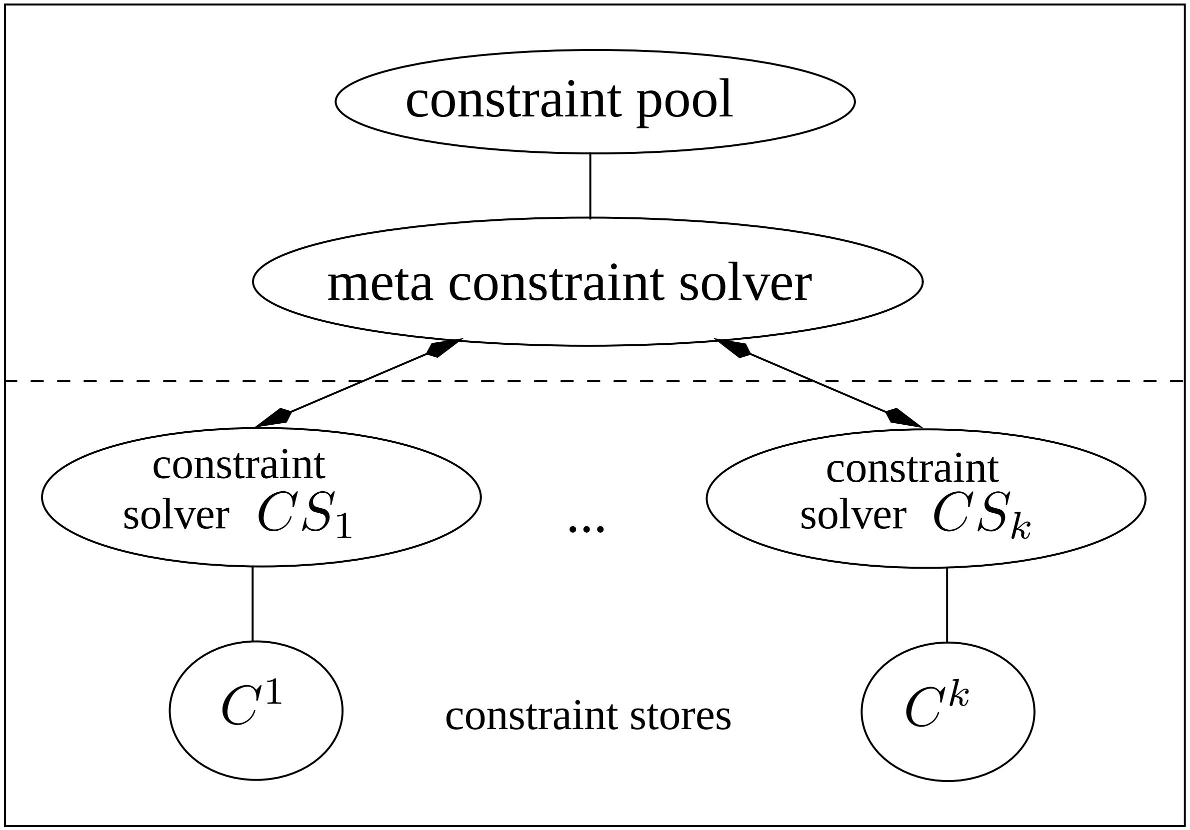
- Constraint Solving:
  - Express spatial relations mathematically by constraints.
  - Determine constraint domains.
  - Solve constraints by cooperating constraint solvers.
- Machine Learning:
  - Learning spatial relations.

Outline

- Constraint Solving:
  - Express spatial relations mathematically by constraints.
  - Determine constraint domains.
  - Solve constraints by cooperating constraint solvers.
- Machine Learning:
  - Learning spatial relations.
  - Deciding constraints by depiction generation.

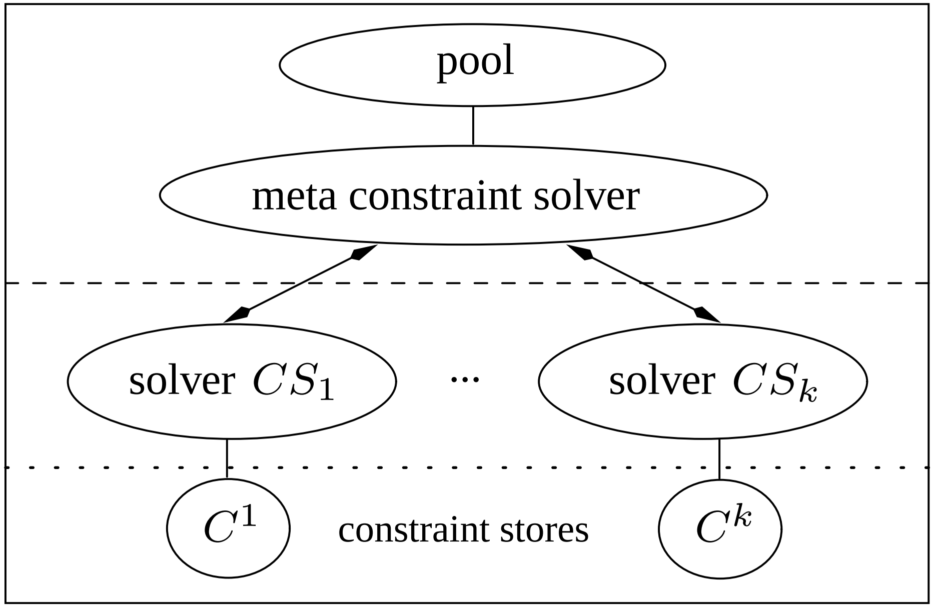
Spatial Inference: Constraint Solving.

Cooperative Constraint Solvers



Spatial Inference: Constraint Solving.

Cooperative Constraint Solvers



- Implementation: [HGS01],
- Integration of solvers, e.g. an interval arithmetic solver [Bra02], a solver for linear arithmetic [Krz97], a finite domain constraint solver [CSP01].

Constraint Solving. Constraint-File.


```
[global]
  verbose = 1
  pool    = meta.pool.Simple
  output  = all
  outvars = {roomw, roomd, dx10, dy10, dx20, dy20, dx30, dy30}
[solver]
  ISolver = solver.brandeis.Brandeis
```

Constraint Solving. Constraint-File. (1) (2) (3)

```
[global]
  verbose = 1
  pool    = meta.pool.Simple
  output  = all
  outvars = {roomw, roomd, dx10, dy10, dx20, dy20, dx30, dy30}
[solver]
  ISolver = solver.brandeis.Brandeis

[constraints]
  room.w, room.d in [4, 4.5];      # width and depth of the room
  O1.w = 0.4; O1.d in [1.0, 1.2]; # size of the cupboard
  O3.w, O3.d in [0.4, 0.5];       # size of the fridge
  O2.r = 0.3;                     # radius of the lamp
```

Constraint Solving. Constraint-File.


```
[global]
  verbose = 1
  pool    = meta.pool.Simple
  output  = all
  outvars = {roomw, roomd, dx10, dy10, dx20, dy20, dx30, dy30}
[solver]
  ISolver = solver.brandeis.Brandeis

[constraints]
  ...  # sizes of objects
  # the relation right(cupboard, lamp)
  dx21 >= O1.w + O2.r;
  dx21 >= dy21 + O1.w - O1.d + (2 ^ 0.5) * O2.r;
  dx21 >= - dy21 + O1.w - O1.d + (2 ^ 0.5) * O2.r;
```

Constraint Solving. Constraint-File. (1) (2) (3)

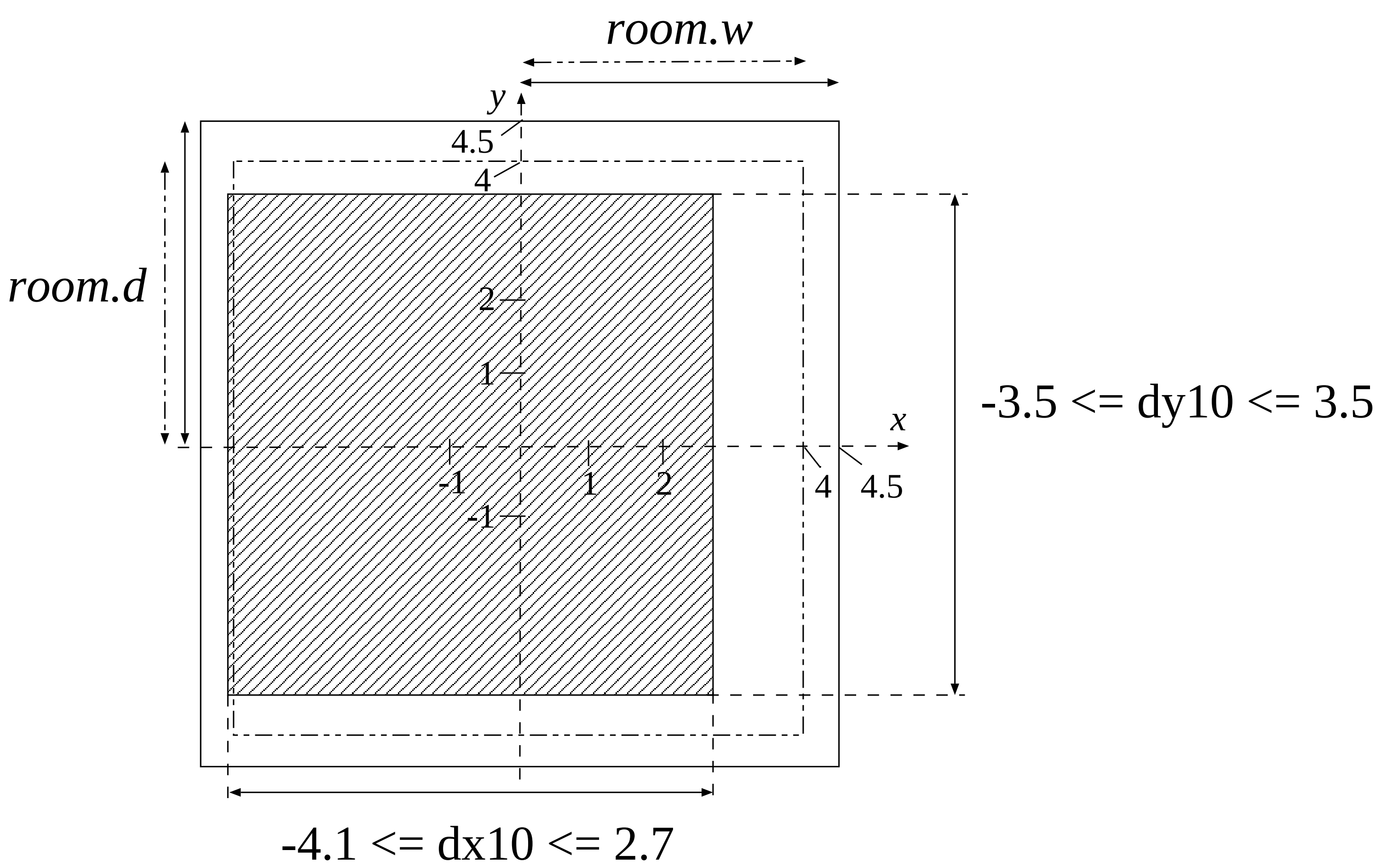
```
[global]
  verbose = 1
  pool    = meta.pool.Simple
  output  = all
  outvars = {roomw, roomd, dx10, dy10, dx20, dy20, dx30, dy30}
[solver]
  ISolver = solver.brandeis.Brandeis

[constraints]
  ...  # sizes of objects
  ...  # the relation right(cupboard, lamp)

  # the coordinates of the lamp wrt.  the cupboard (transformation)
  dx21 = dx20 - dx10;
  dy21 = dy20 - dy10;
```

Constraint Solving. Solution space.

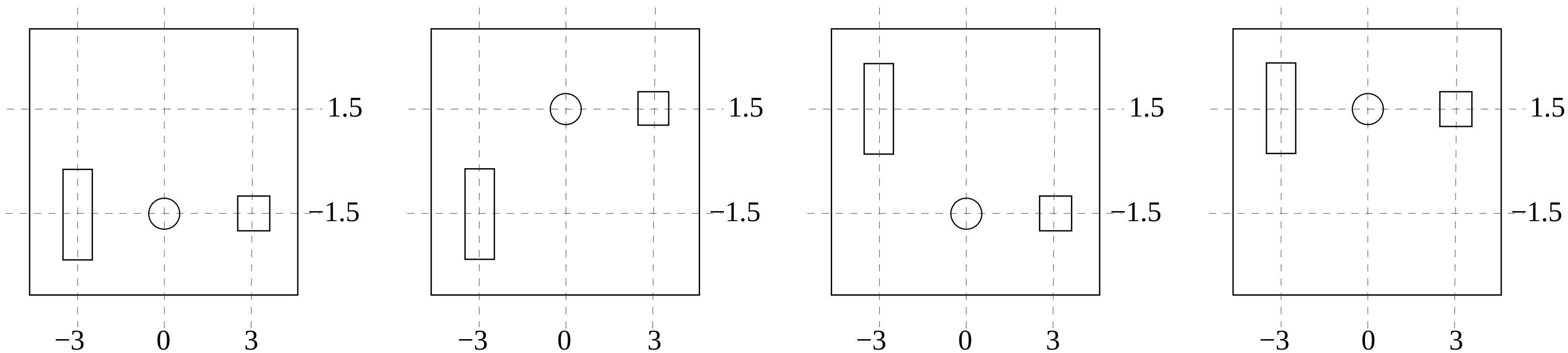
Placement of the cupboard. Solution space:



Constraint Solving. Solution space.

Solution space. Depictions:

```
...
[solver]
  ISolver = solver.brandeis.Brandeis
  FDSolver= solver.csplib.CSPlib
[constraints]
  ...
  dx10, dx20, dx30 inn {-3.0, 0.0, 3.0};
  dy10, dy20, dy30 inn {-1.5, 1.5};
```



Machine Learning Approach

- Constraint solvers are often incomplete.

Machine Learning Approach

- Constraint solvers are often incomplete.
- 2-Step algorithm:

Machine Learning Approach

- Constraint solvers are often incomplete.
- 2-Step algorithm:
    - Learning the problem, i.e. the decision boundary.

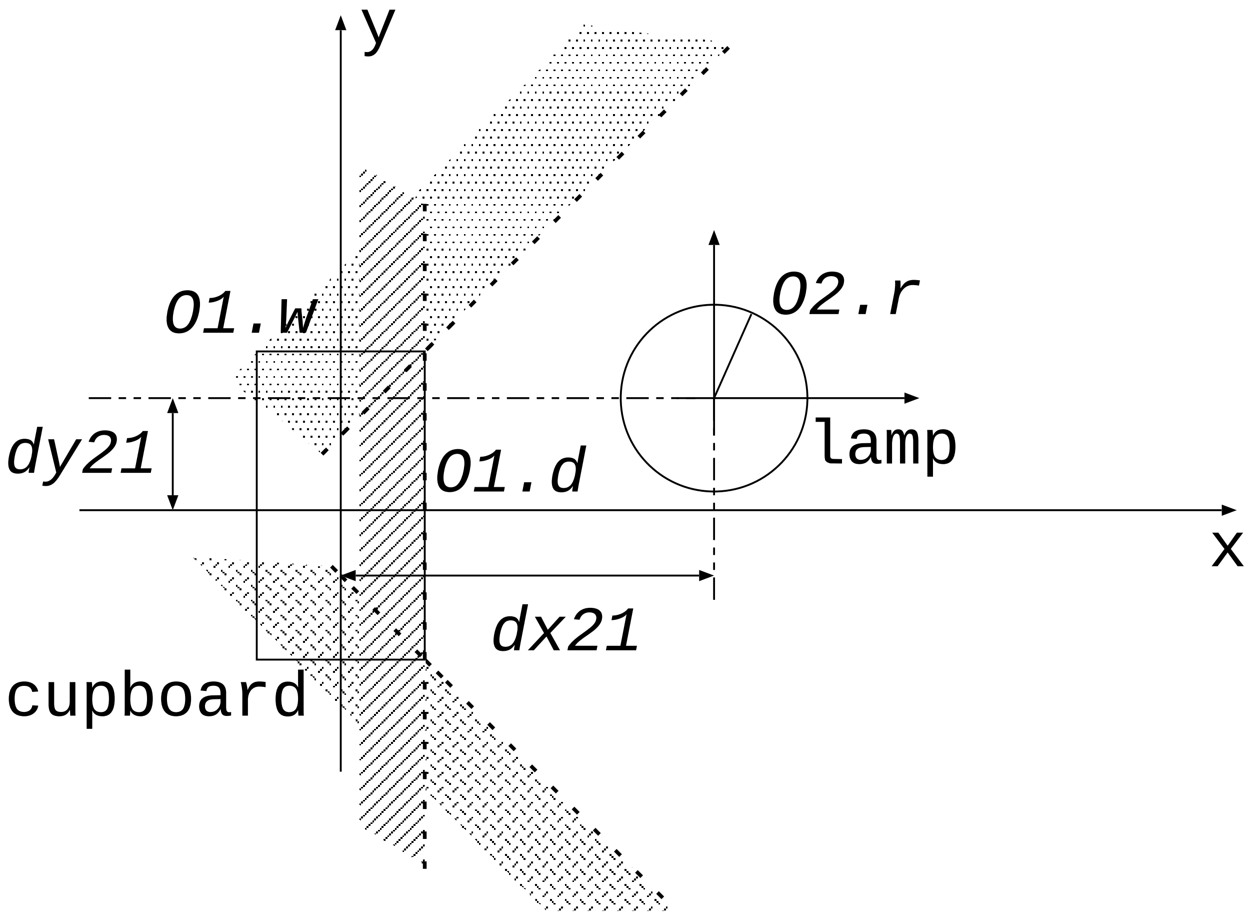
Machine Learning Approach

- Constraint solvers are often incomplete.
- 2-Step algorithm:
  - Learning the problem, i.e. the decision boundary.
  - Use results for solving the constraints,
    i.e. generating depictions.

Machine Learning Approach

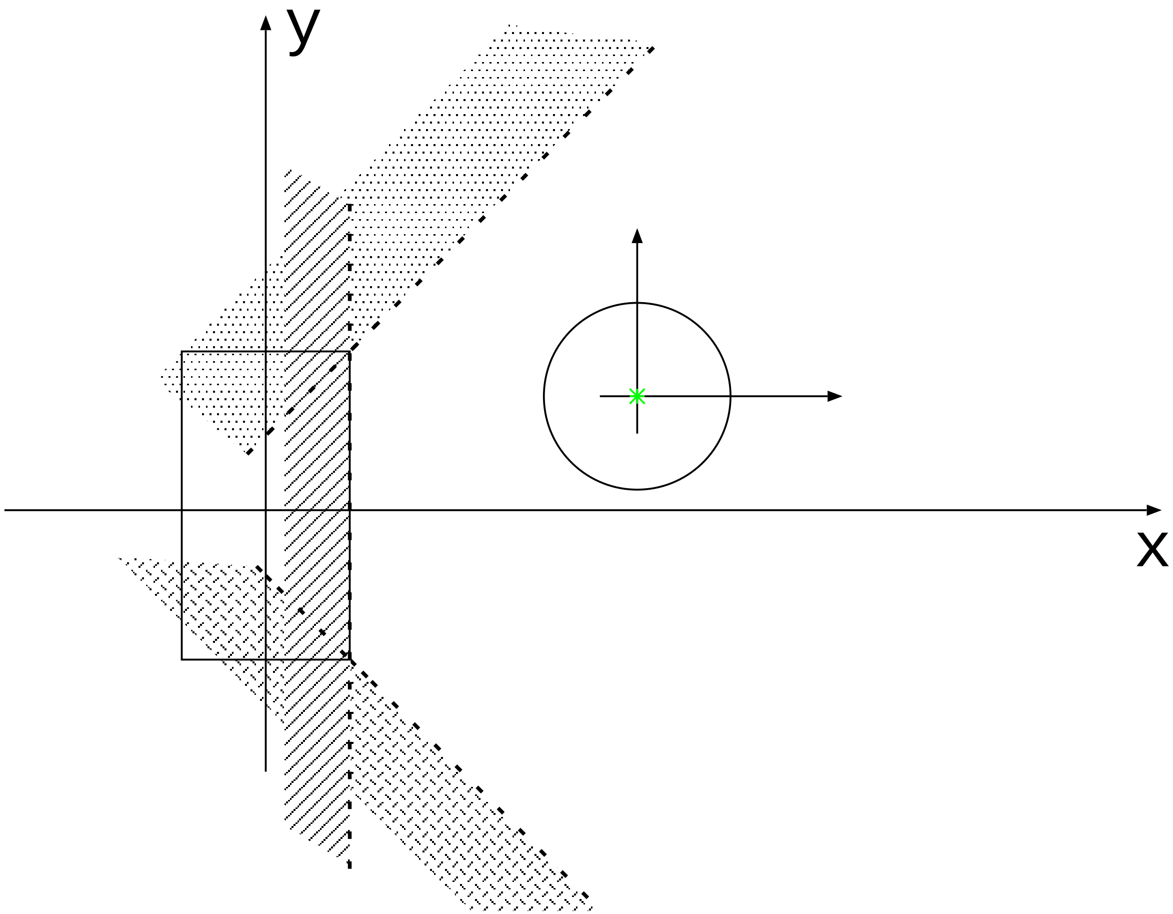
- Constraint solvers are often incomplete.
- 2-Step algorithm:
  - Learning the problem, i.e. the decision boundary.
  - Use results for solving the constraints,
    i.e. generating depictions.
- Learning step is done only once.

Machine Learning. Learning.

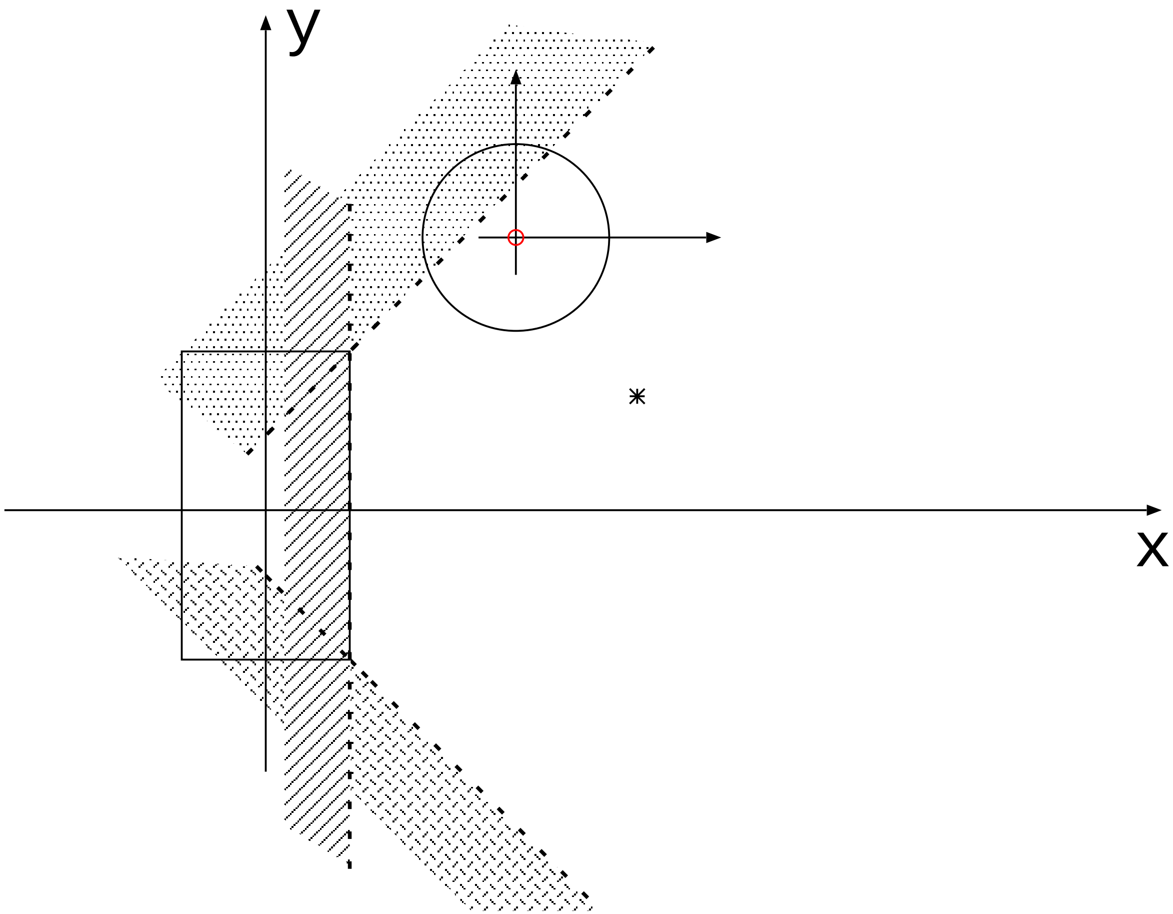


right(cupboard, lamp).

Machine Learning. Learning.

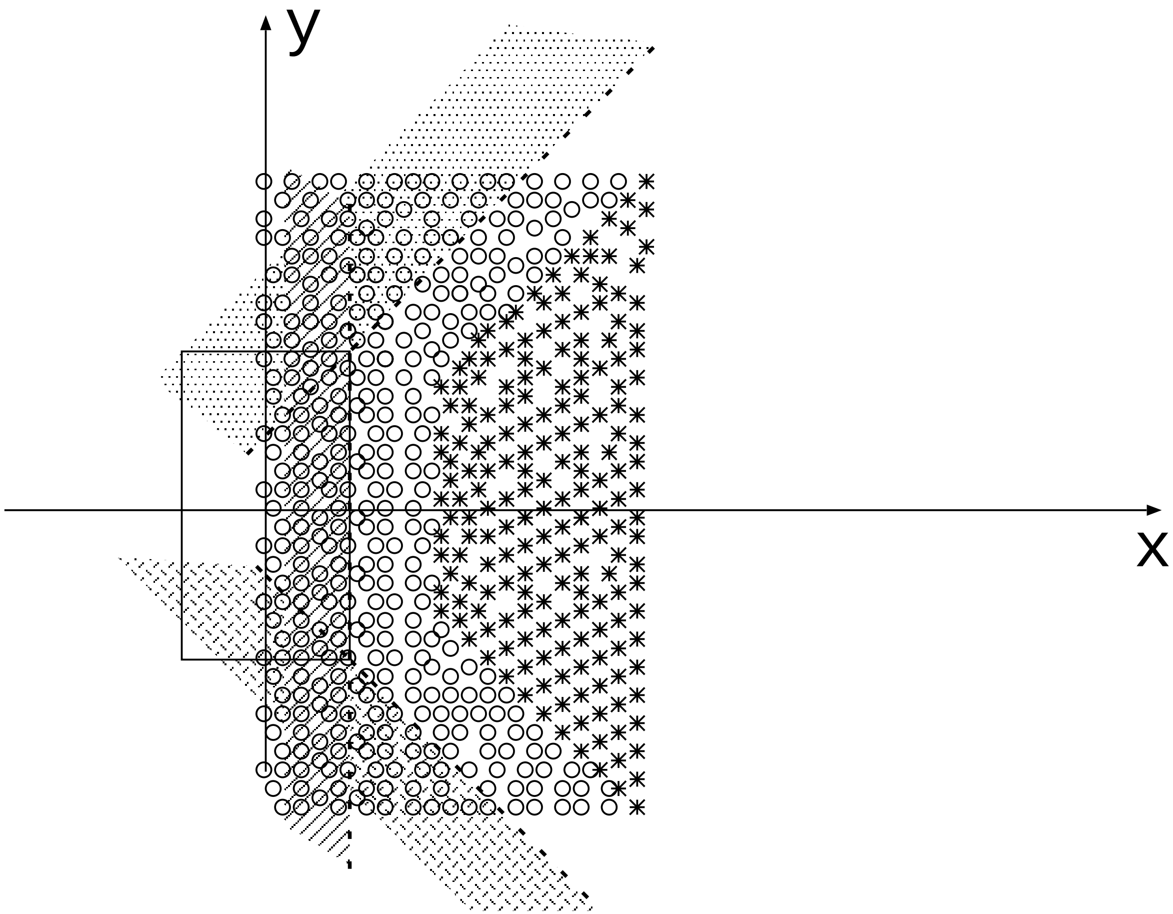


Slice of configuration space: parameters w, d, r clamped.



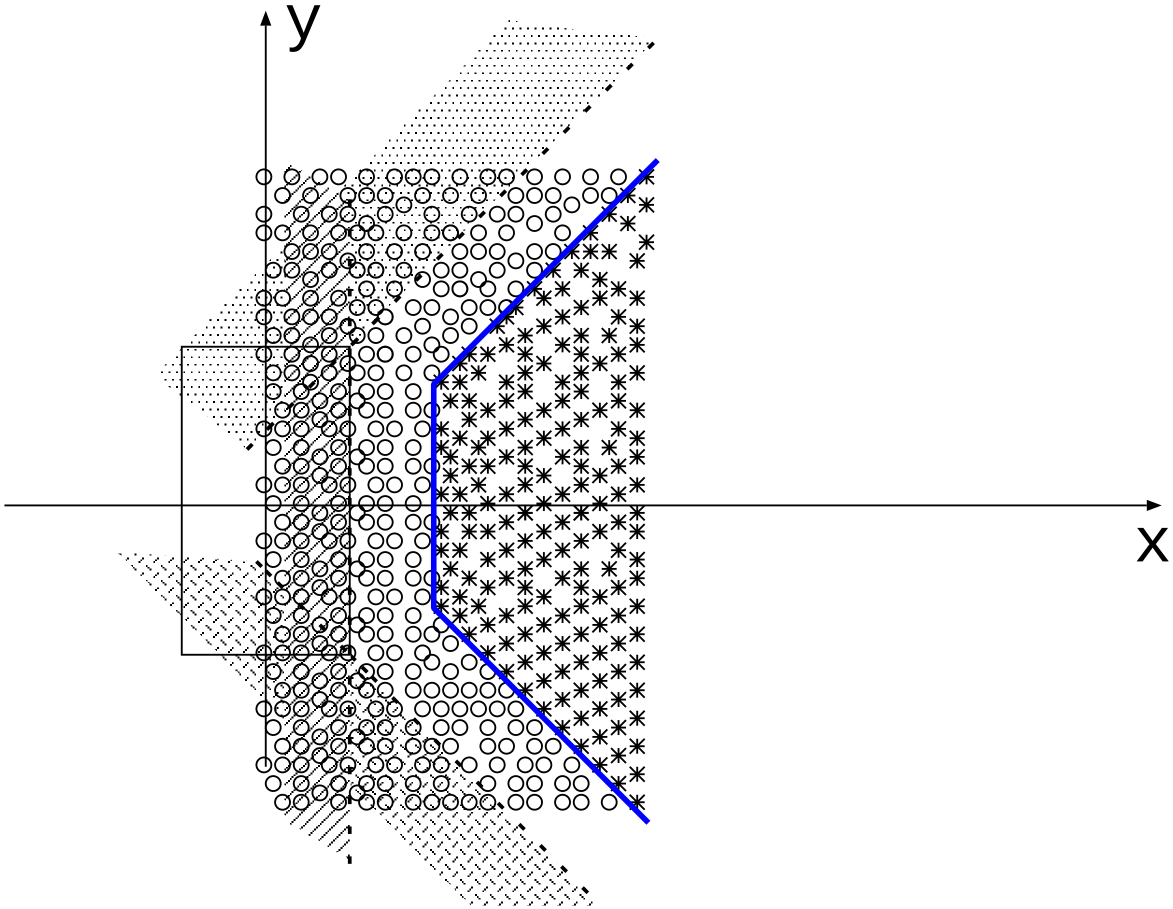
Slice of configuration space: parameters w, d, r clamped.

Machine Learning. Learning.



Dataset for training with CAL5.

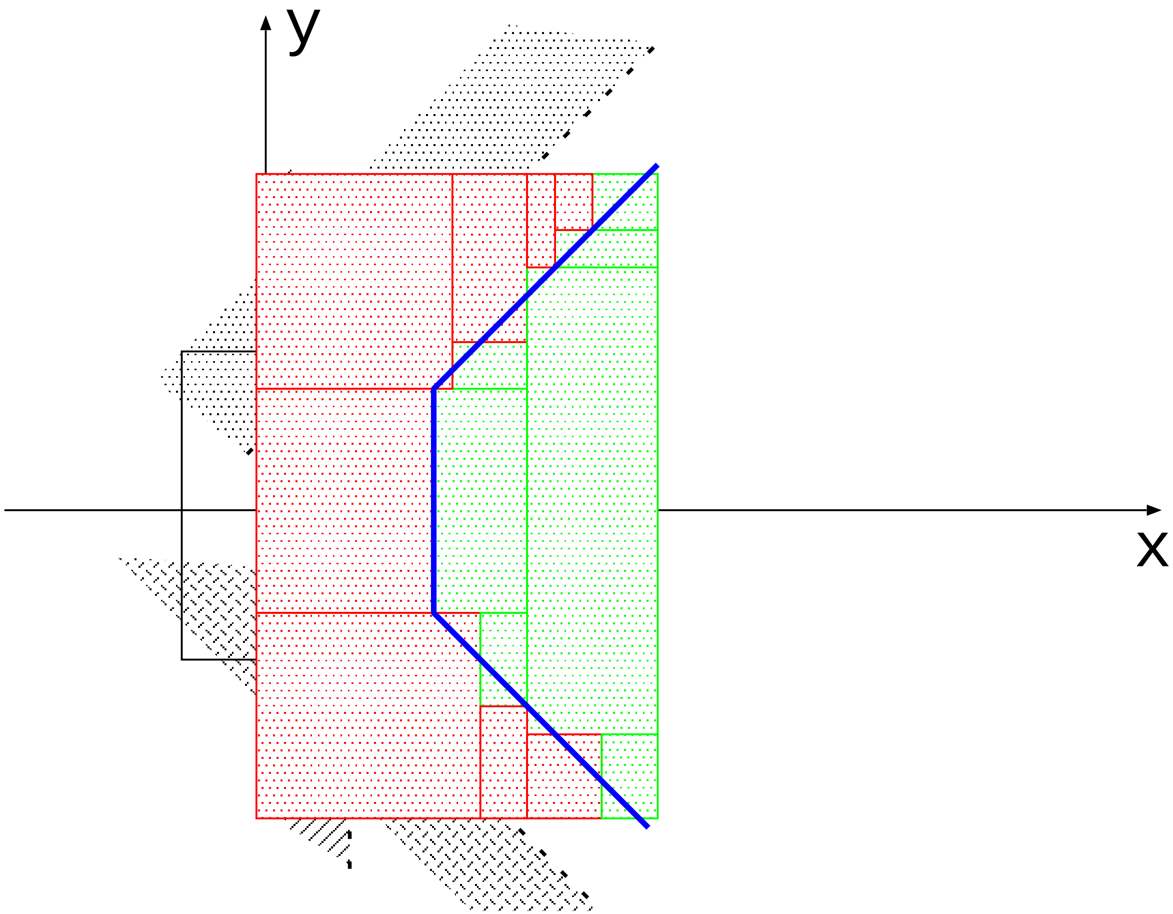
Machine Learning. Learning.



Decision boundary (to be learned).

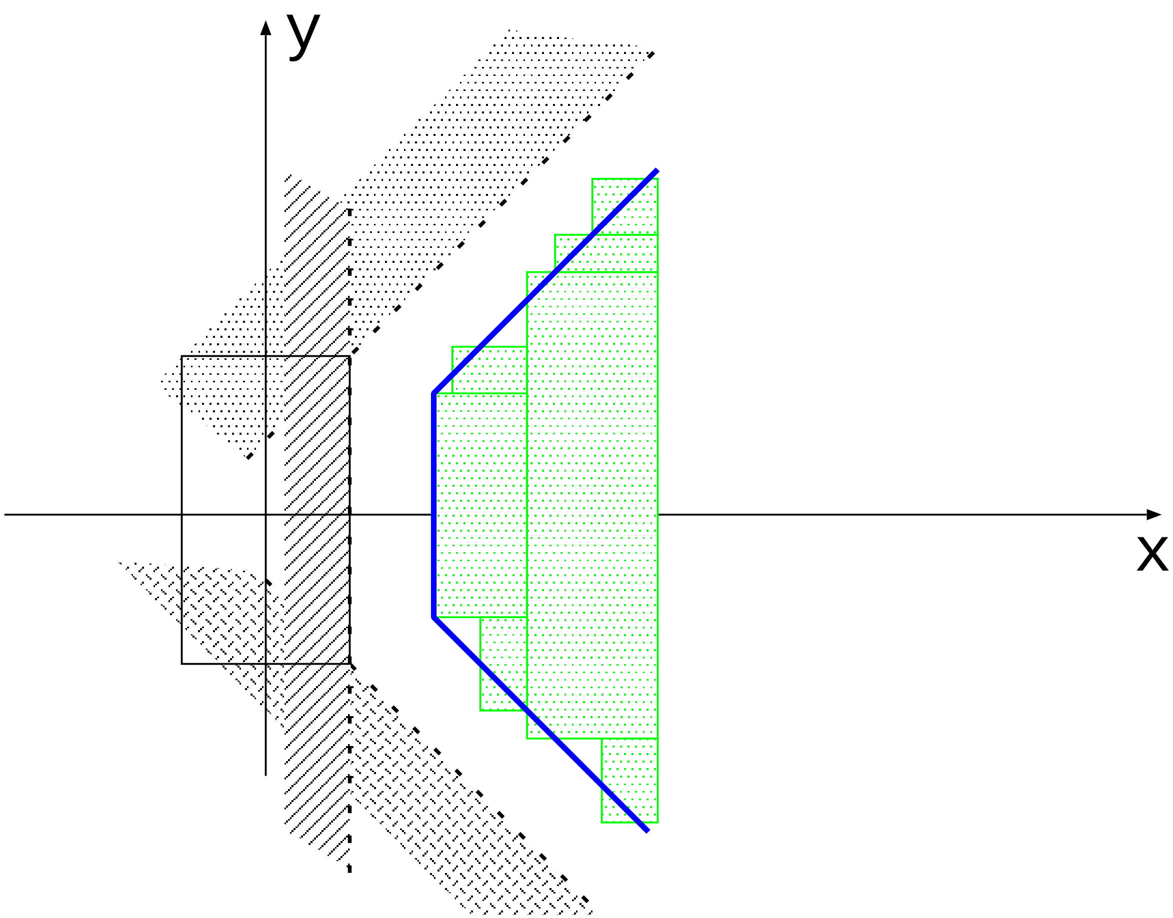
Machine Learning. Learning.

(1) (2) (3)



Resulting CAL5 decision tree.

Machine Learning. Learning.



"Pruned" decision tree.

Step 2

- Solving the constraints: generating depictions.

Machine Learning. Depiction Generation.

Step 2

- Solving the constraints: generating depictions.
- Use results of the learning step.

Step 2

- Solving the constraints: generating depictions.
- Use results of the learning step.
- Binary relations: always two objects.

Machine Learning. Depiction Generation.

Step 2

- Solving the constraints: generating depictions.
- Use results of the learning step.
- Binary relations: always two objects.
  - Both objects unknown.

Machine Learning. Depiction Generation.

Step 2

- Solving the constraints: generating depictions.
- Use results of the learning step.
- Binary relations: always two objects.
  - Both objects unknown.
  - First object already placed.

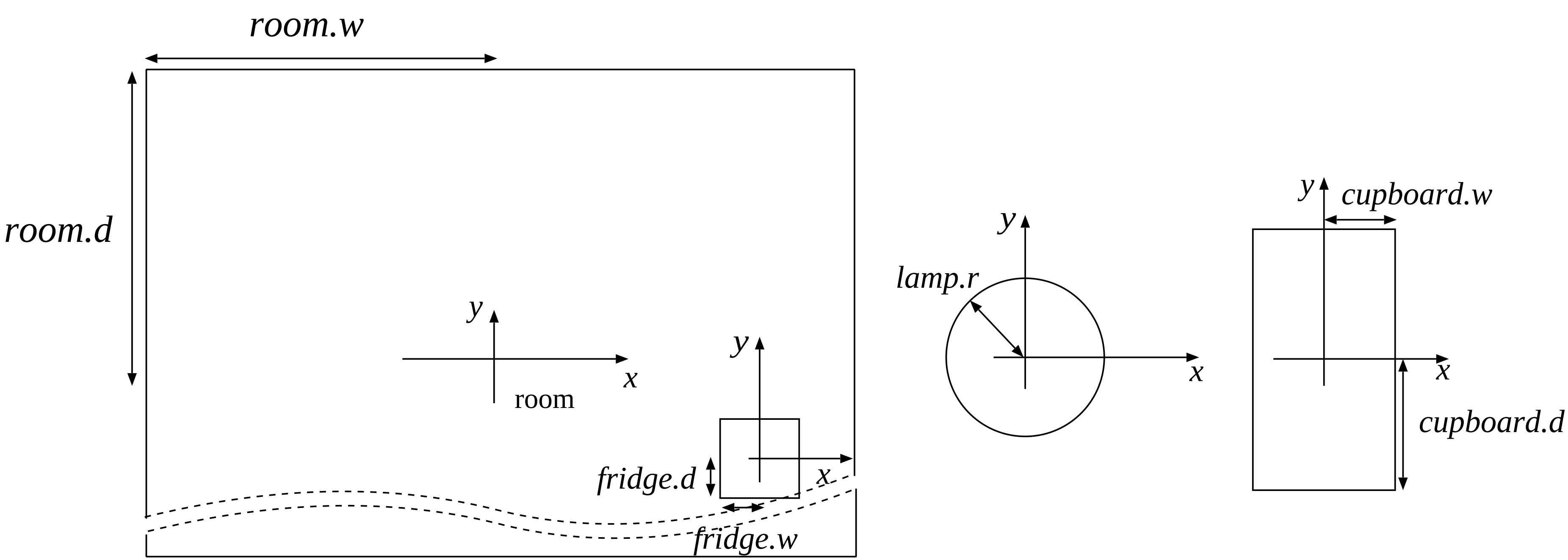
Step 2

- Solving the constraints: generating depictions.
- Use results of the learning step.
- Binary relations: always two objects.
  - Both objects unknown.
  - First object already placed.
  - Second object placed.

Step 2

- Solving the constraints: generating depictions.
- Use results of the learning step.
- Binary relations: always two objects.
  - Both objects unknown.
  - First object already placed.
  - Second object placed.
  - Both objects already placed.

Machine Learning. Depiction Generation: 1.

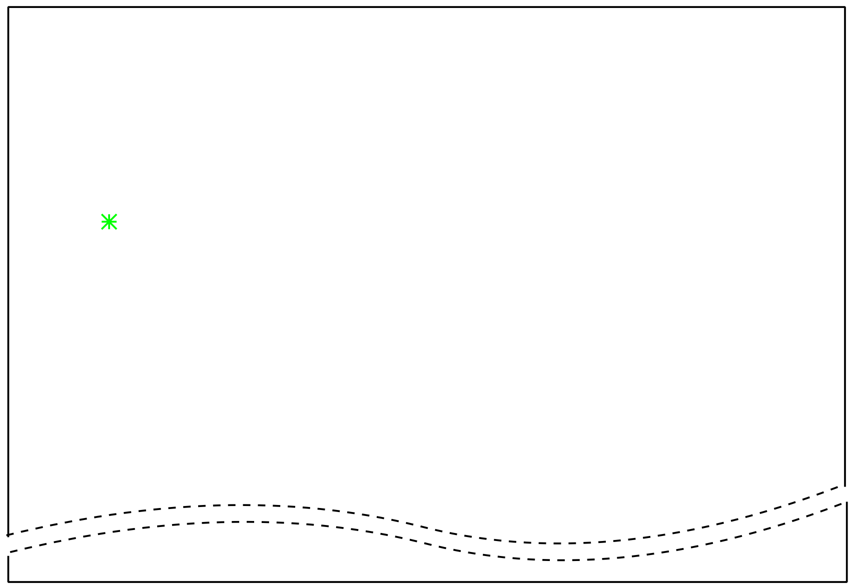


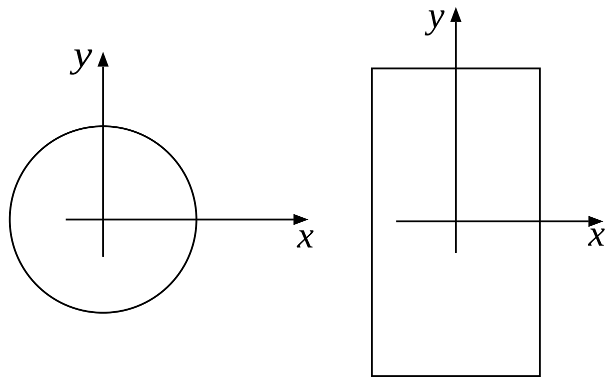
`right(cupboard, lamp)`

Both objects unknown.

Machine Learning. Depiction Generation: 1.

(1) (2) (3)

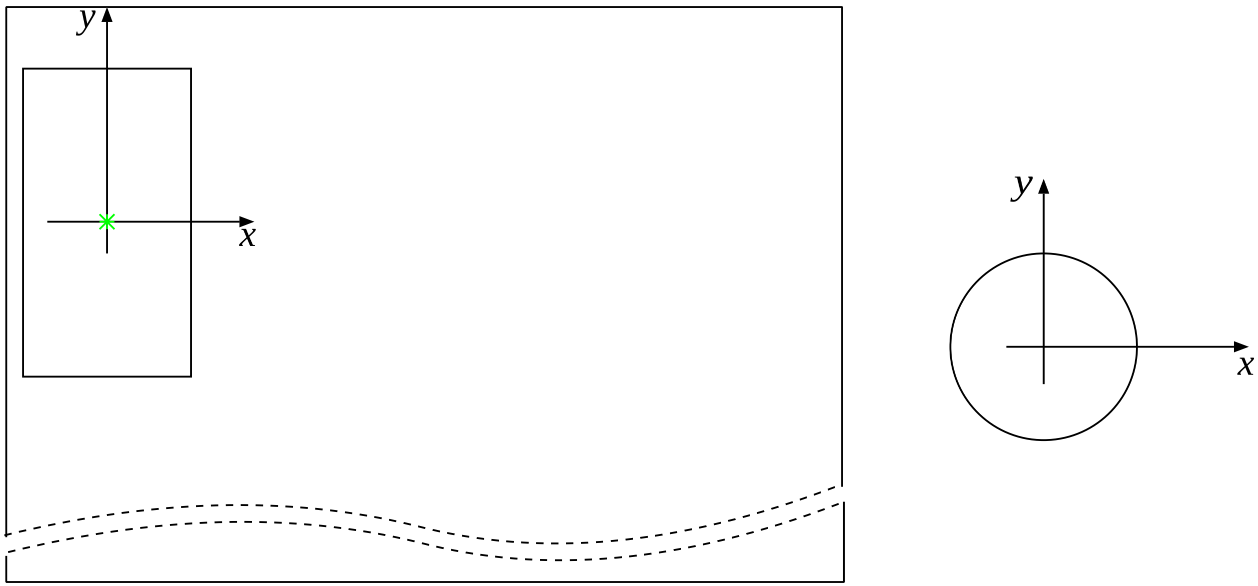




`right(cupboard, lamp)`

Select point randomly in room.

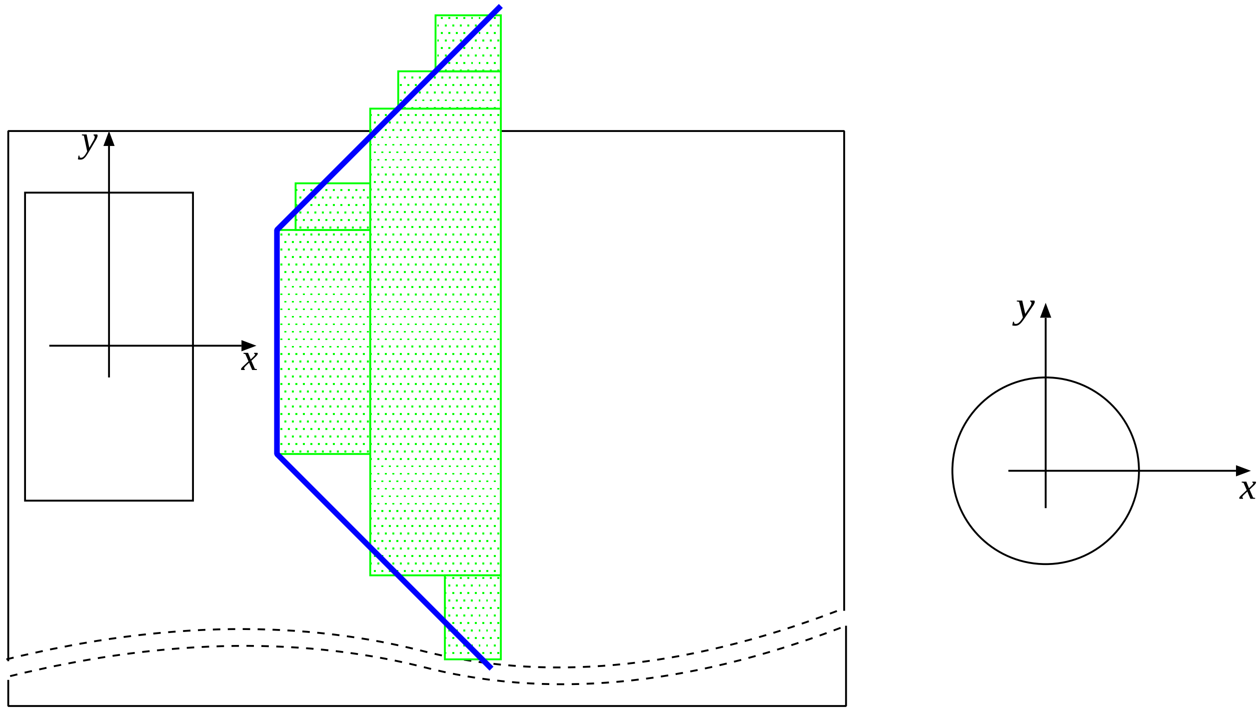
Machine Learning. Depiction Generation: 1.



`right(cupboard, lamp)`

Place cupboard.

Machine Learning. Depiction Generation: 1+2.

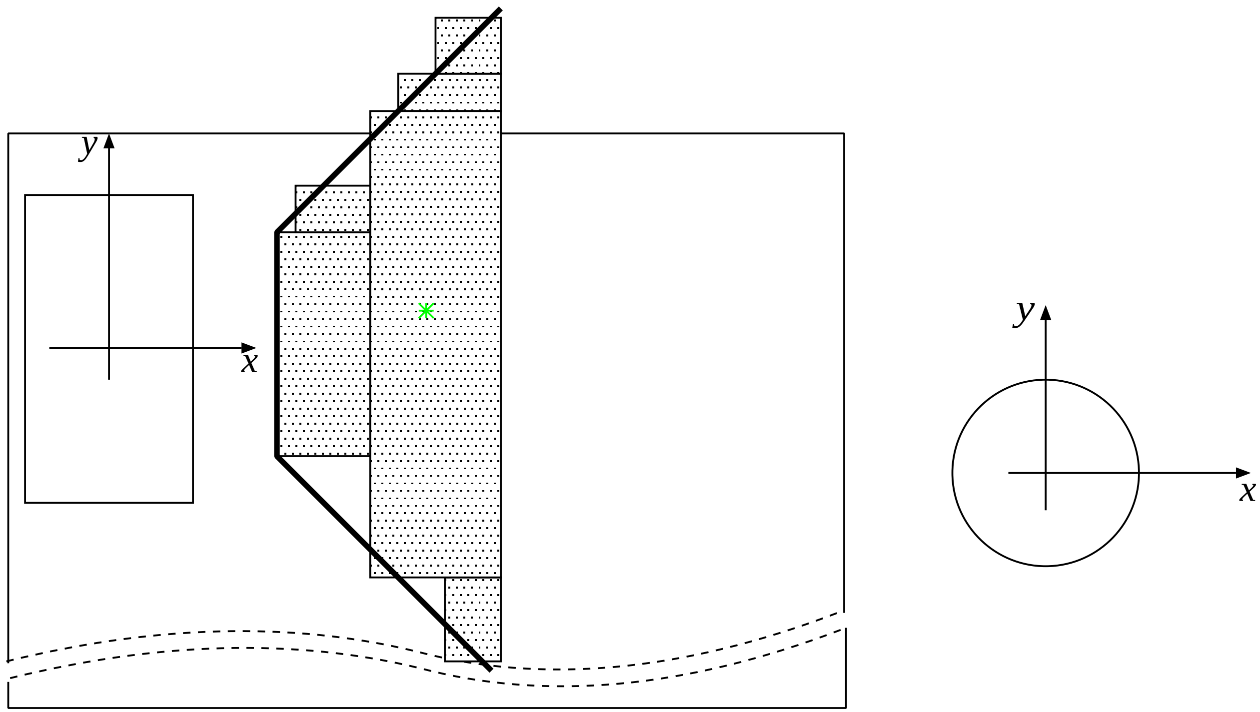


`right(cupboard, lamp)`

Use CAL5 decision tree.

Machine Learning. Depiction Generation: 1+2.

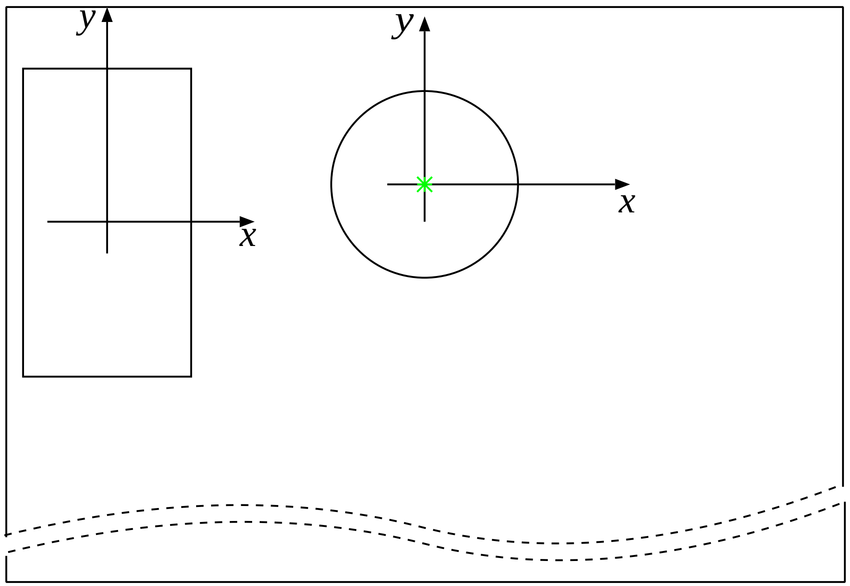
(1) (2) (3)



`right(cupboard, lamp)`

Select point in the class A region of the tree.

Machine Learning. Depiction Generation: 1+2.



`right(cupboard, lamp)`

Place lamp.

Spatial Inference – Constraint Solving and Learning

- Quantitative description of spatial relations.

Spatial Inference – Constraint Solving and Learning

- Quantitative description of spatial relations.
- Deciding spatial scenes using cooperative constraint solvers.

Spatial Inference – Constraint Solving and Learning

- Quantitative description of spatial relations.
- Deciding spatial scenes using cooperative constraint solvers.
- Machine learning: learning decision boundaries of spatial relations and generating depictions using the decision trees.

Spatial Inference – Constraint Solving and Learning

- Quantitative description of spatial relations.
- Deciding spatial scenes using cooperative constraint solvers.
- Machine learning: learning decision boundaries of spatial relations and generating depictions using the decision trees.
- Cooperation of machine learning and constraint solving:

Spatial Inference – Constraint Solving and Learning

- Quantitative description of spatial relations.
- Deciding spatial scenes using cooperative constraint solvers.
- Machine learning: learning decision boundaries of spatial relations and generating depictions using the decision trees.
- Cooperation of machine learning and constraint solving:
  - Detection of unsatisfiable descriptions using constraint solvers, fine-grain elaboration by learning approach.

Spatial Inference – Constraint Solving and Learning

- Quantitative description of spatial relations.
- Deciding spatial scenes using cooperative constraint solvers.
- Machine learning: learning decision boundaries of spatial relations and generating depictions using the decision trees.
- Cooperation of machine learning and constraint solving:
  - Detection of unsatisfiable descriptions using constraint solvers, fine-grain elaboration by learning approach.
  - Using constraint solvers for generating training sets.

References

[Bra02] The Brandeis Interval Arithmetic Constraint Solver, 2002. Available from `http://www.cs.brandeis.edu/~tim/`.

[Krz97] O. Krzikalla. Constraint Solver für lineare Constraints über reellen Zahlen. Großer Beleg. Technische Universität Dresden. (in German), 1997.

[CSP01] A 'C' Library of Routines for Solving Binary CSP, 2001. Available from `http://www.ai.uwaterloo.ca/~vanbeek/software/csplib.tar.gz`.

[HvEW98] T.J. Hickey, M.H. van Emden, and H. Wu. A Unified Framework for Interval Constraints and Interval Arithmetic. *CP'98*, LNCS 1520. Springer, 1998.

[GLM+96] P. Griebel, G. Lehrenfeld, W. Mueller, C. Tahedl, and H. Uhr. Integrating a Constraint Solver into a Real-Time Animation Environment. In *Proc. of the 1996 IEEE Symposium on Visual Languages*, September 1996.

[HGS01] P. Hofstedt, E. Godehardt, and D. Seifert. A Framework for Cooperating Constraint Solvers - A Prototypic Implementation. In E. Monfroy and L. Granvilliers, editors, *Workshop on Cooperative Solvers in Constraint Programming - CoSolv*, 2001.

[Hof00] P. Hofstedt. Better Communication for Tighter Cooperation. In *First International Conference on Computational Logic*, LNCS 1861. Springer, 2000.

References

[Gue89] H. W. Guesgen. Spatial Reasoning Based on Allen's Temporal Logic. Technical Report TR-89-049, ICSI, Berkeley, Cal., 1989.

[Her94] Daniel Hernández. *Qualitative Representation of Spatial Knowledge*. LNAI 804. Springer, 1994.

[AP75] A. P. Ambler and R. J. Popplestone. Inferring the Positions of Bodies from Specified Spatial Relationships. *Artificial Intelligence*, 6:157–174, 1975.

[JL83] P. N. Johnson-Laird. *Mental Models: Towards a Cognitive Science of Language, Inference and Consciousness*. Cambridge University Press, Cambridge, 1983.

[JW96] T. Jörding and I. Wachsmuth. An Antropomorphic Agent for the Use of Spatial Language. In *Proceedings of ECAI'96-Workshop on Representation and Processing of Spatial Expressions*, pages 41–53, 1996.

[NT97] G. Nakhaeizadeh and C. C. Taylor, editors. *Machine Learning and Statistics - The Interface*. Wiley, 1997.

[Röf97] T. Röfer. Routemark-based Navigation of a Wheelchair. In *Third ECPD International Conference on Advanced Robotics, Intelligent Automation and Active Systems*, Bremen, 1997.

References

(1) (2) (3)

[CEG+98] B. Claus, K. Eyferth, C. Gips, R. Hörnig, U. Schmid, S. Wiebrock, and F. Wysotzki. Reference Frames for Spatial Inferences in Text Comprehension. In C. Freksa, C. Habel, and K. F. Wender, editors, *Spatial Cognition*, LNAI 1404. Springer, 1998.

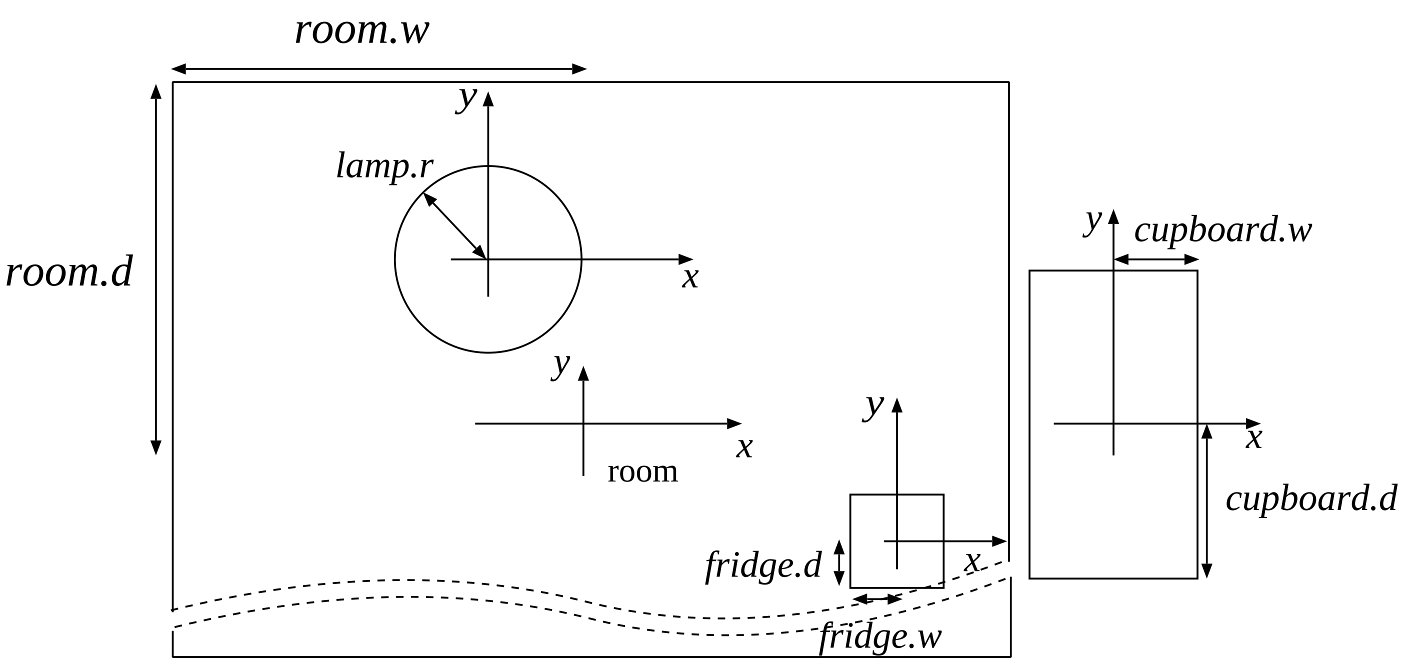
[GGWW98] P. Geibel, C. Gips, S. Wiebrock, and F. Wysotzki. Learning Spatial Relations with CAL5 and TRITOP. Technical Report 98-7, TU Berlin, 1998.

[WW99] S. Wiebrock and F. Wysotzki. Lernen von räumlichen Relationen mit CAL5 und DIPOL. Technical Report 99-17, TU Berlin, 1999.

[WWSW00] S. Wiebrock, L. Wittenburg, U. Schmid, and F. Wysotzki. Inference and Visualization of Spatial Relations. In C. Freksa, W. Brauer, C. Habel, and K. Wender, editors, *Spatial Cognition II*, LNAI 1849. Springer, 2000.

Machine Learning. Depiction Generation: 3.

(1) (2) (3)

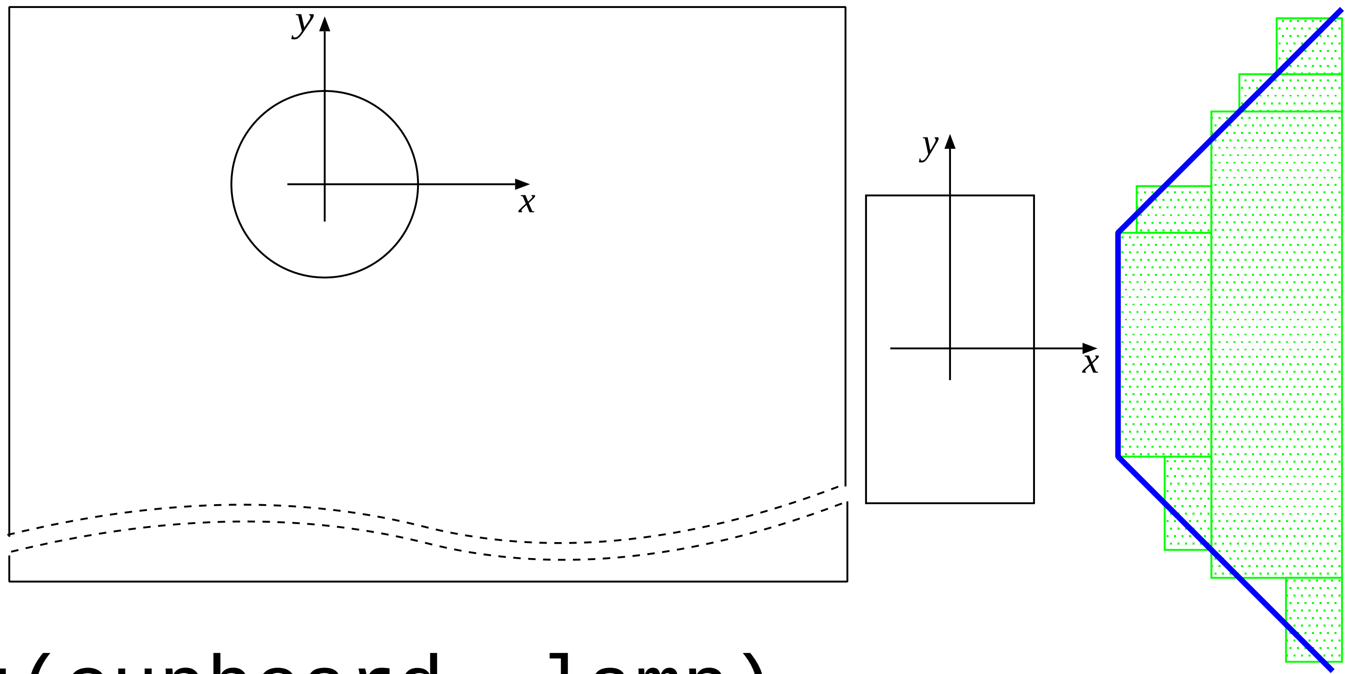


`right(cupboard, lamp)`

Lamp already placed.

Machine Learning. Depiction Generation: 3.

(1) (2) (3)

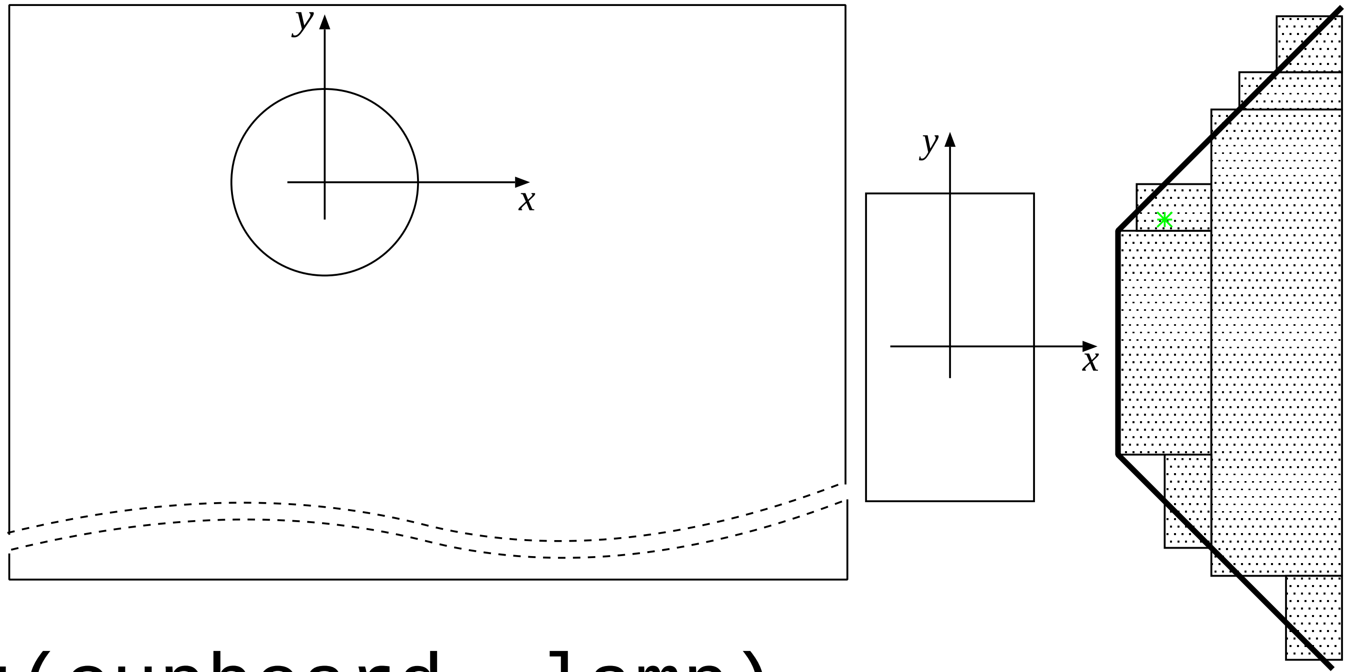


`right(cupboard, lamp)`

Use CAL5 decision tree.

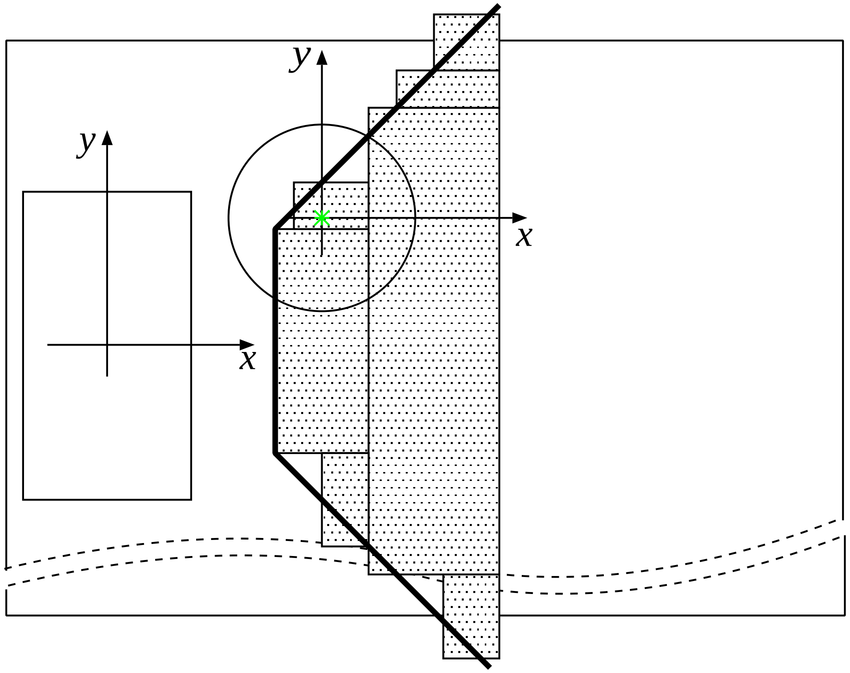
Machine Learning. Depiction Generation: 3.

(1) (2) (3)




```
right(cupboard, lamp)
```

Chose point in the class A region of the tree.

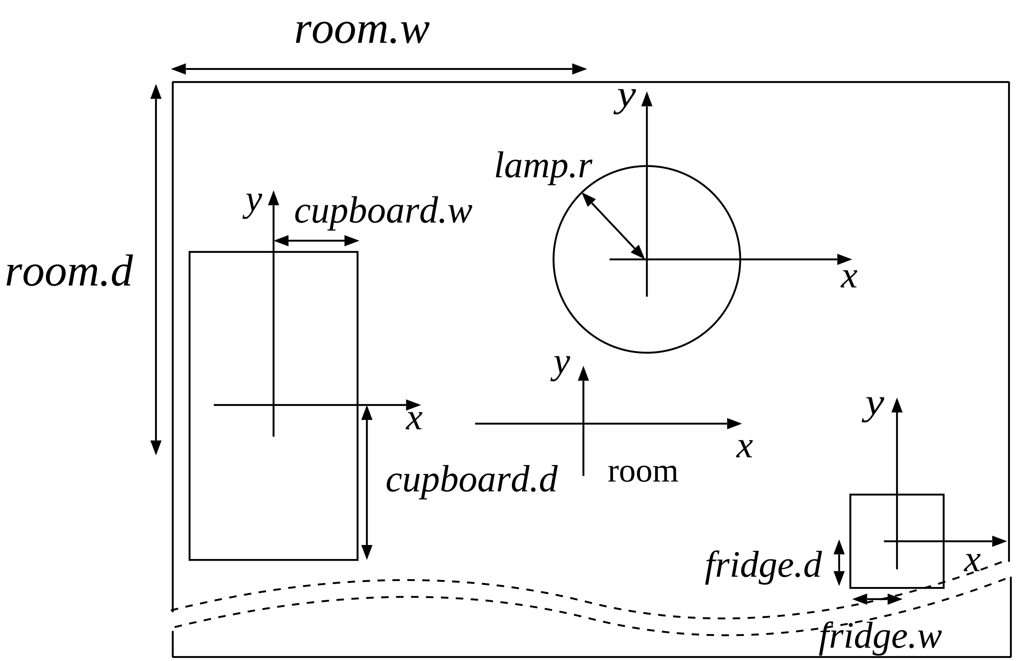


`right(cupboard, lamp)`

Map point to origin of lamp, backtransformation:
Cupboard fi xed.

Machine Learning. Depiction Generation: 4.

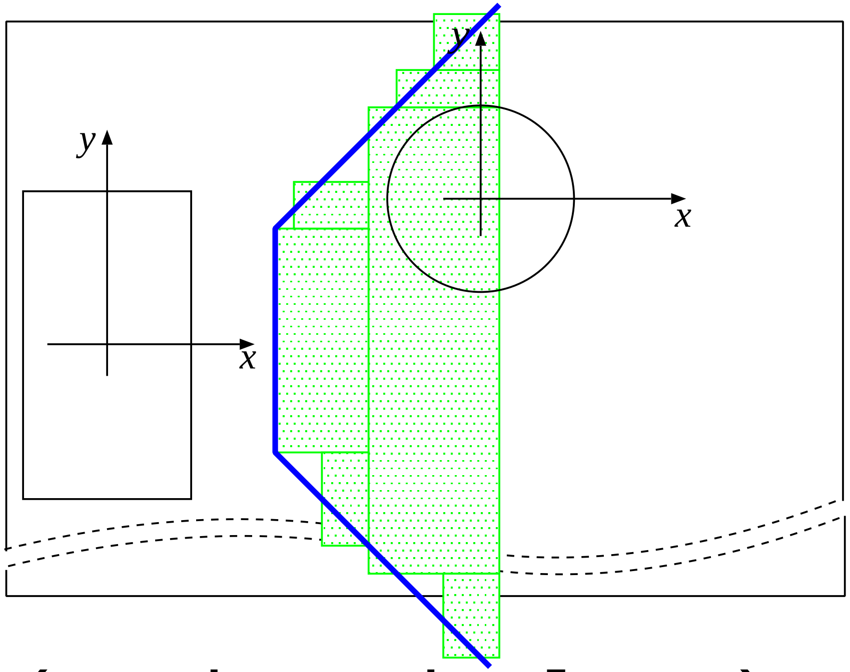
(1) (2) (3)



`right(cupboard, lamp)`

Both objects already placed.

Machine Learning. Depiction Generation: 4.

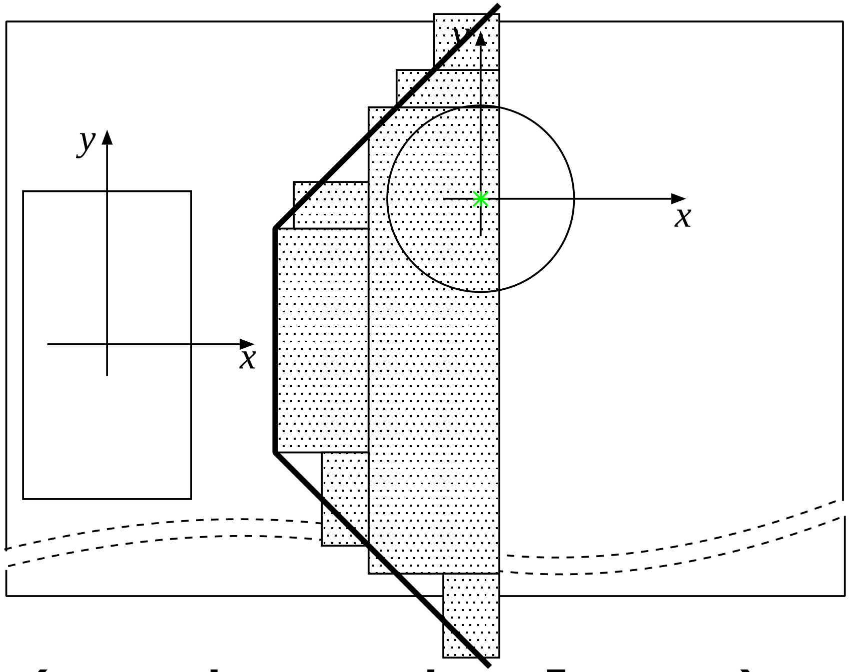


`right(cupboard, lamp)`

Use CAL5 decision tree.

Machine Learning. Depiction Generation: 4.

(1) (2) (3)



`right(cupboard, lamp)`

Origin of lamp is in the class A region of the tree: `right(cupboard, lamp)` holds.

Change of the basis:

$$\mathbf{p}_k{}^i = \mathbf{P}_j{}^i \mathbf{p}_k{}^j$$

$\mathbf{P}_j{}^i$ Homogeneous transformation matrix:

$$\mathbf{P}_j{}^i = \begin{pmatrix} \cos\Theta_j{}^i & -\sin\Theta_j{}^i & 0 & \Delta x_j{}^i \\ \sin\Theta_j{}^i & \cos\Theta_j{}^i & 0 & \Delta y_j{}^i \\ 0 & 0 & 1 & \Delta z_j{}^i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$