Machine learning using TIL

Adam Albert

Vysoká škola báňská – Technická univerzita Ostrava

Outline

- Supervised machine learning
- Refining the hypothesis space
- Algorithm framework
- Semantic network
- Algorithm specification
- Learning example

Supervised machine learning

- Examples described by attributes(input/output) provided by a teacher
- Hypothesis
- Prediction of output attributes values from input attributes
- Training/test examples
- Classification/regression

Refining the hypothesis space

- Learning is finding hypotheses that are consistent with the training data[Poole, 2010]
  - There is only one output(Boolean) attribute Y
  - Hypotheses determine output attribute value
  - There is no noise in data
- Hypothesis is written as a proposition
- Refining of hypothesis(in form of proposition) by induction learning

Algorithm framework

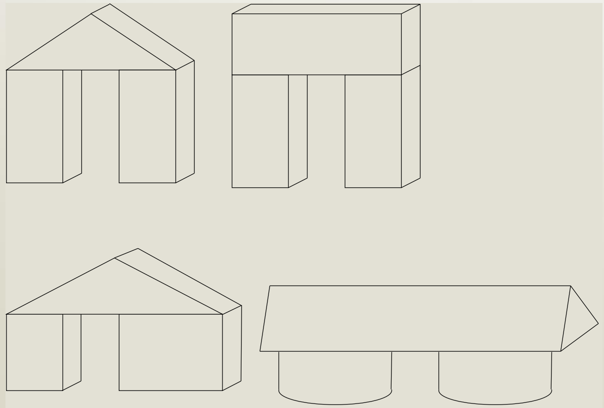
- Machine learning algorithm can be described by[Luger, 2009]:
  - Task goal
  - Training data
  - Data representation
  - Knowledge modifying operations

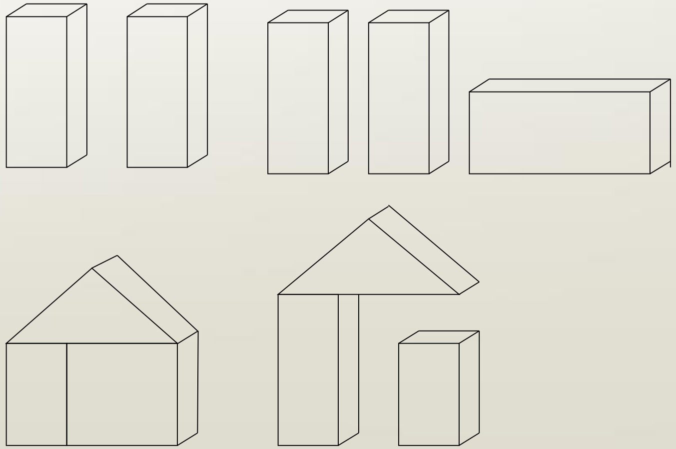
Algorithm framework – task goal

- The goal is to find a general concept describing the class of arches.

Algorithm framework – training data

- The learner is working with a set of (positive) examples of arches and a set of "almost" arches
- Teacher's responsibilities





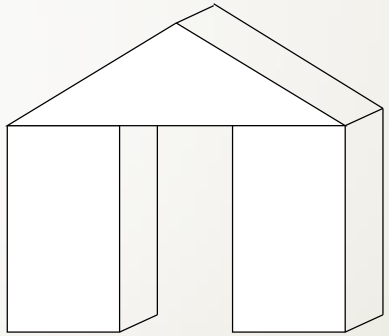
Algorithm framework – data representation

- The representation must be so fine that the agent is able to find the hypothesis
- Arches in our example are defined by means of TIL constructions

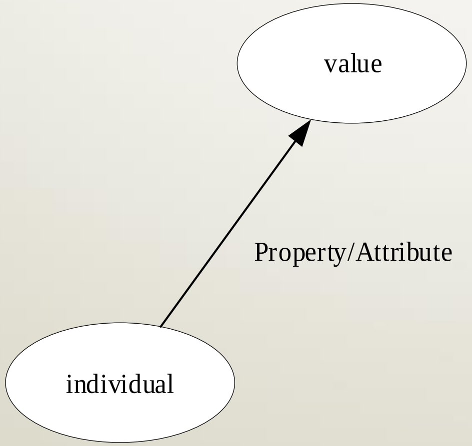
Algorithm framework - operations

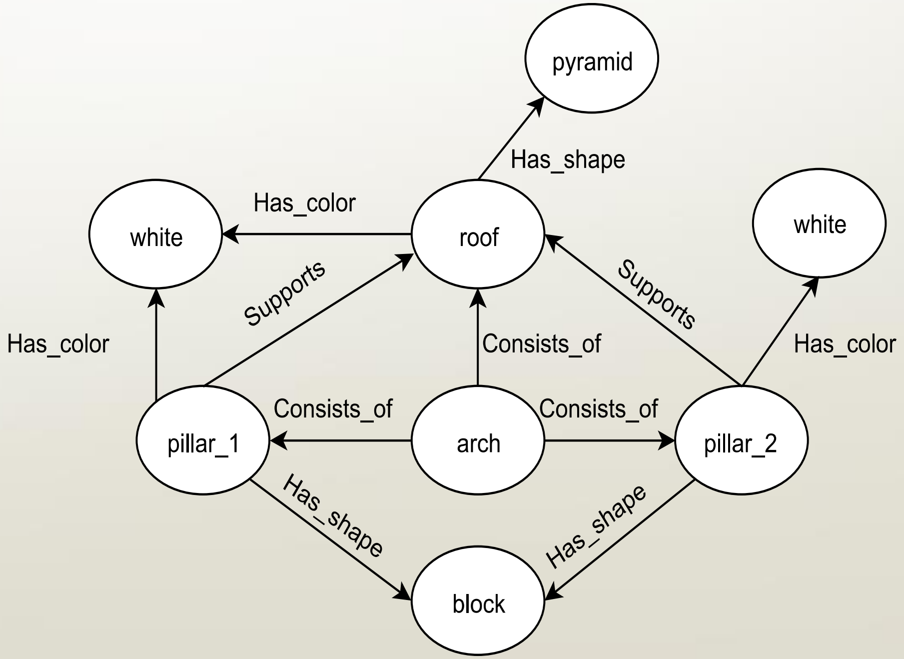
- Patrick Winston algorithm [Winston, 1992]
- Generalization – makes hypothesis more permissive
- Specialization – makes hypothesis more restrictive

Semantic network



- Relation link (Individual, Property/Attribute, Value)





Algorithm

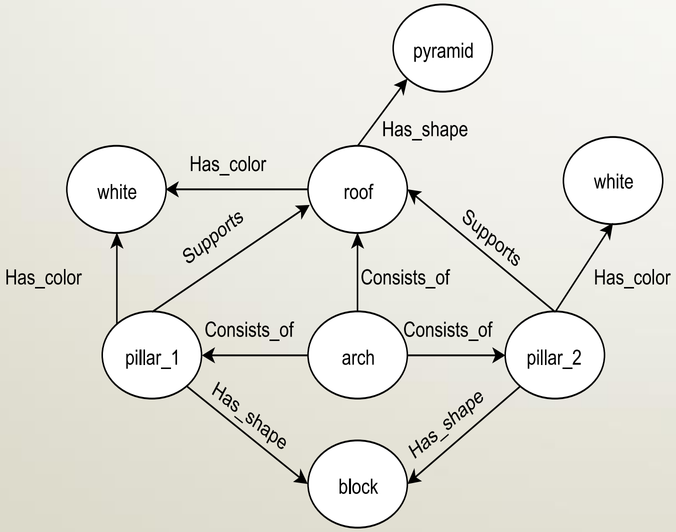
Specialization

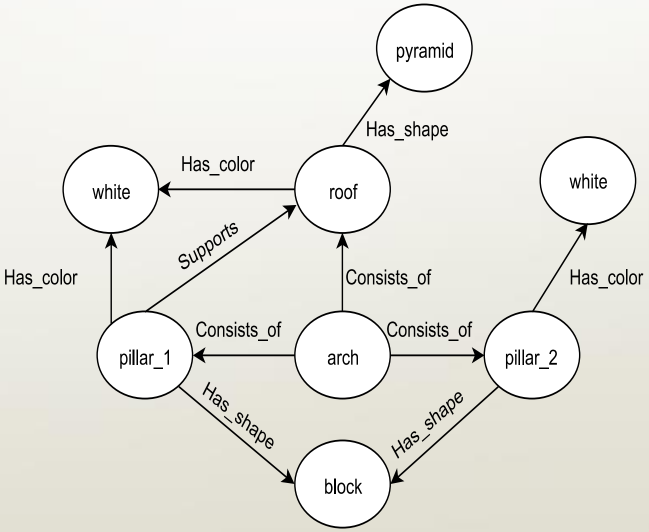
- Specialization is used to refine the hypothesis by a near-miss example.

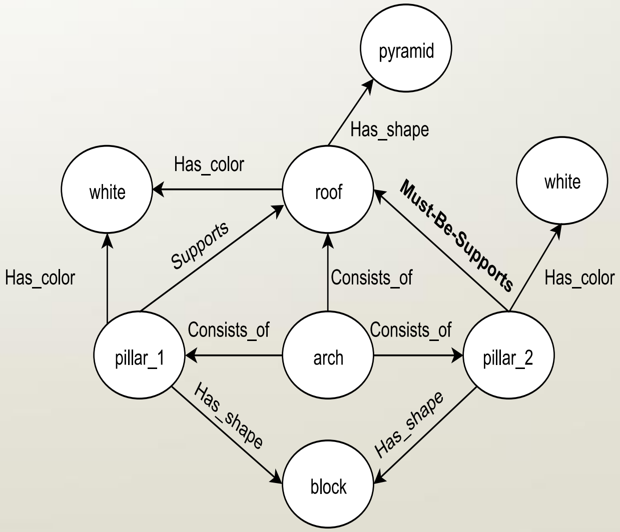
1. Compare the model hypothesis (to be refined) and the near-miss example to find a significant difference
2. If there is a significant difference between model and near-miss example, then
   a) if the model has a link relation while the near-miss example does not, use **require-link**

Require-link

- Heuristic is applied in case that the model has while the near-miss example does not have a *link relation*.
- In the model the *link relation* is marked as MUST-BE.







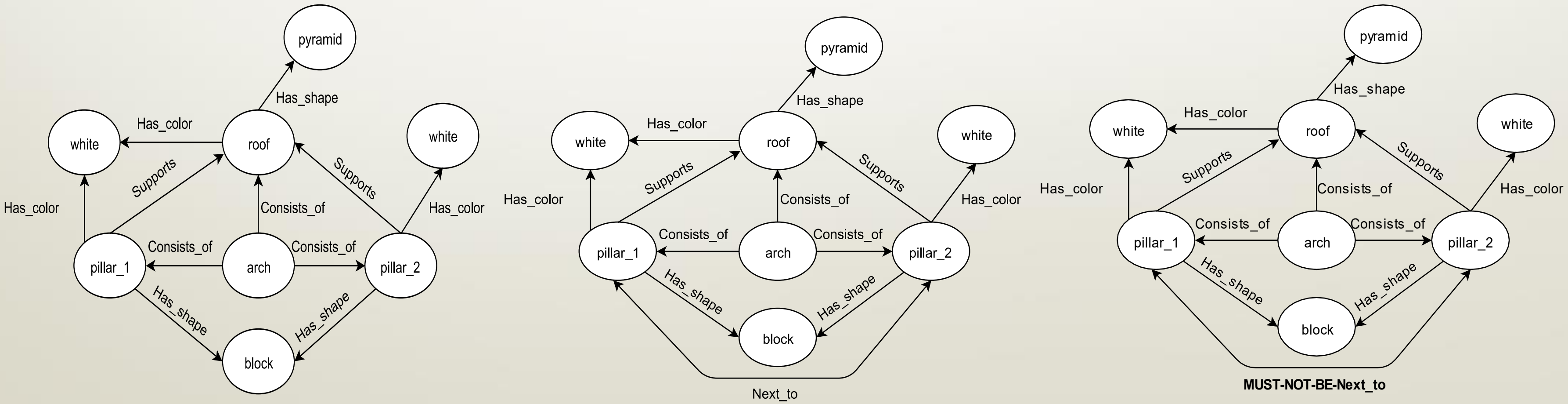
Specialization

- Specialization is used to refine the hypothesis by a near-miss example.

1. Compare the model hypothesis (to be refined) and the near-miss example to find a significant difference
2. If there is a significant difference between model and near-miss example, then
   a) if the model has a link relation while the near-miss example does not, use **require-link**
   b) if the near-miss example has a link relation that the model does not have, use **forbid-link**

Forbid-link

- There is a link relation in the near-miss example which is missing in the model.
- Model is enriched with this *link relation* marked by ‘MUST-NOT-BE’.



Specialization

- Specialization is used to refine the hypothesis by a near-miss example.

1. Compare the model hypothesis (to be refined) and the near-miss example to find a significant difference
2. If there is a significant difference between model and near-miss example, then
   a) if the model has a link relation while the near-miss example does not, use **require-link**
   b) if the near-miss example has a link relation that the model does not have, use **forbid-link**
   c) else ignore example

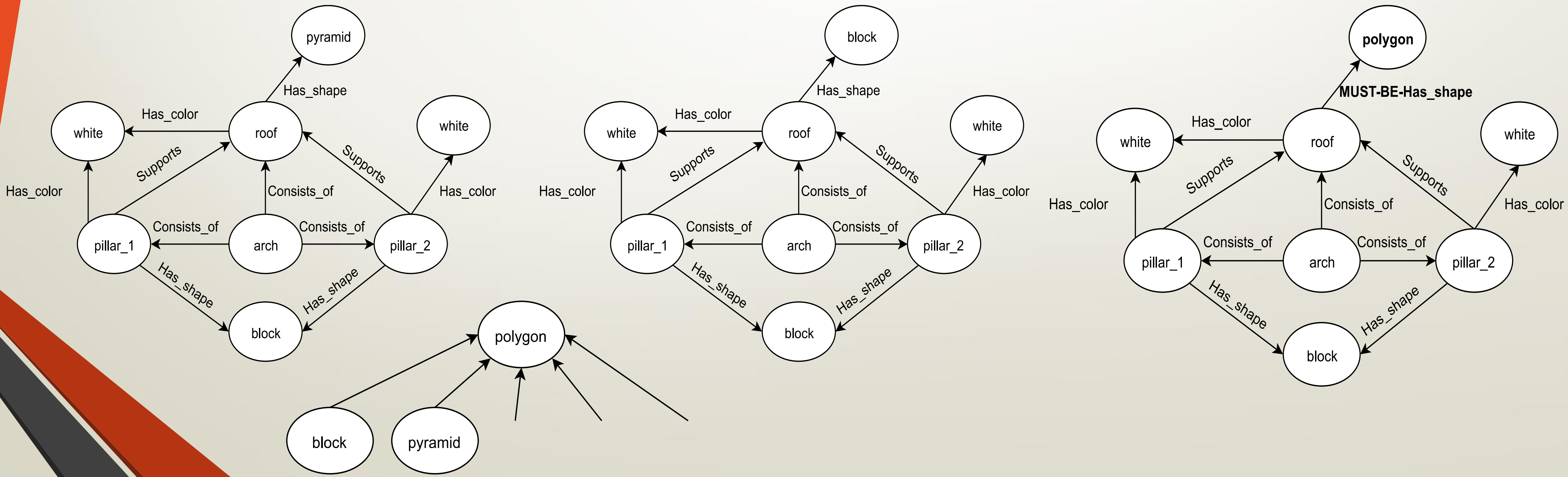
Generalization

- Generalization is used to refine hypothesis by a positive example

1. Compare the model hypothesis and the positive example to determine a difference
2. For each difference do
   a) if a link relation in the model points at a value that differs from the value in the example, then
      i. if the values in which the model and example differ have the most specific general class, use the **climb-tree**

Climb-tree

- This heuristic is applied in case we need to generalize the concepts to avoid problems with too specialized models.



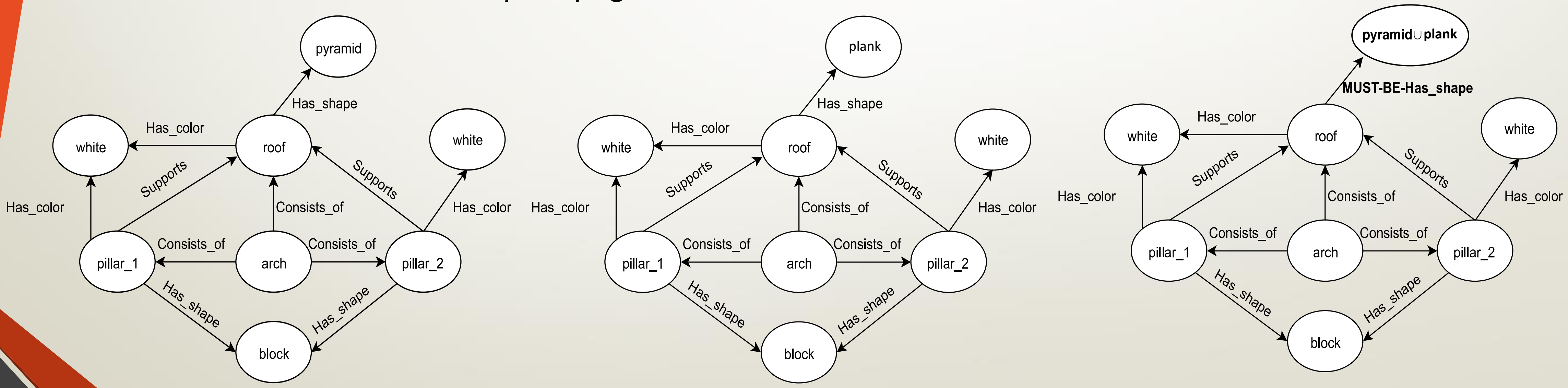
Generalization

- Generalization is used to refine hypothesis by a positive example

1. Compare the model hypothesis and the positive example to determine a difference
2. For each difference do
   a) if a link relation in the model points at a value that differs from the value in the example, then
      i. if the values in which the model and example differ have the most specific general class, use the **climb-tree**
      ii. else if the values in which the model and example differ don't have the most specific general class, use the **enlarge-set**

Enlarge-set

- This heuristic is applied in case we need to generalize concepts of classes but there is no common most specific general class at our disposal.
- Generalization is achieved by unifying these classes.



Generalization

- Generalization is used to refine hypothesis by a positive example

1. Compare the model hypothesis and the positive example to determine a difference
2. For each difference do
   a) if a link relation in the model points at a value that differs from the value in the example, then
      i. if the values in which the model and example differ have the most specific general class, use the **climb-tree**
      ii. else if the values in which the model and example differ don't have the most specific general class, use the **enlarge-set**
      iii. else if these classes are excluding each other use **drop-link**

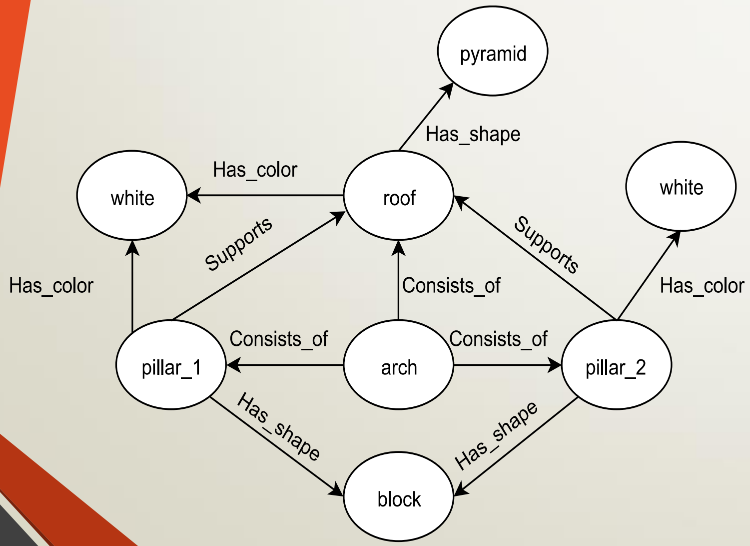
Generalization

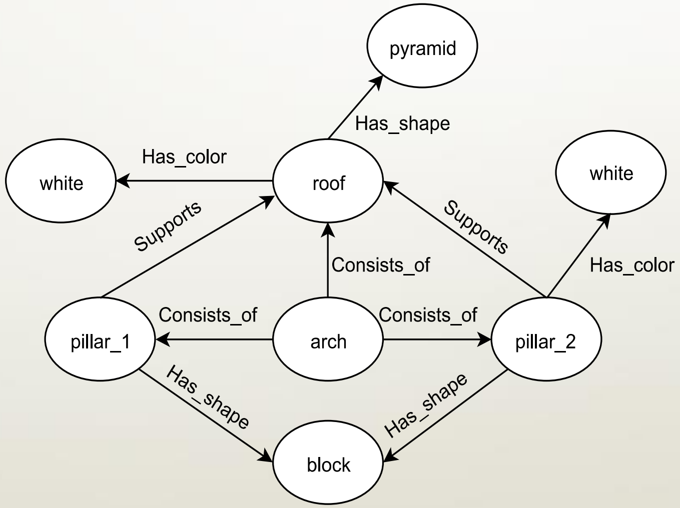
- Generalization is used to refine hypothesis by a positive example

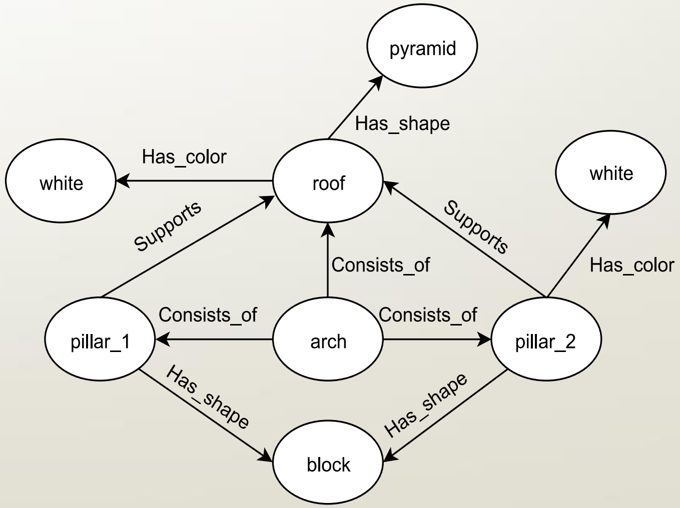
1. Compare the model hypothesis and the positive example to determine a difference
2. For each difference do
   a) if a link relation in the model points at a value that differs from the value in the example, then
      i. if the values in which the model and example differ have the most specific general class, use the **climb-tree**
      ii. else use **enlarge-set**
      iii. if these classes are excluding each other use **drop-link**
   b) if there is a link in the model that is missing in the example, use **drop-link**

Drop-link

- If the model contains link that is missing in the positive example or the values related to the link excluding each other







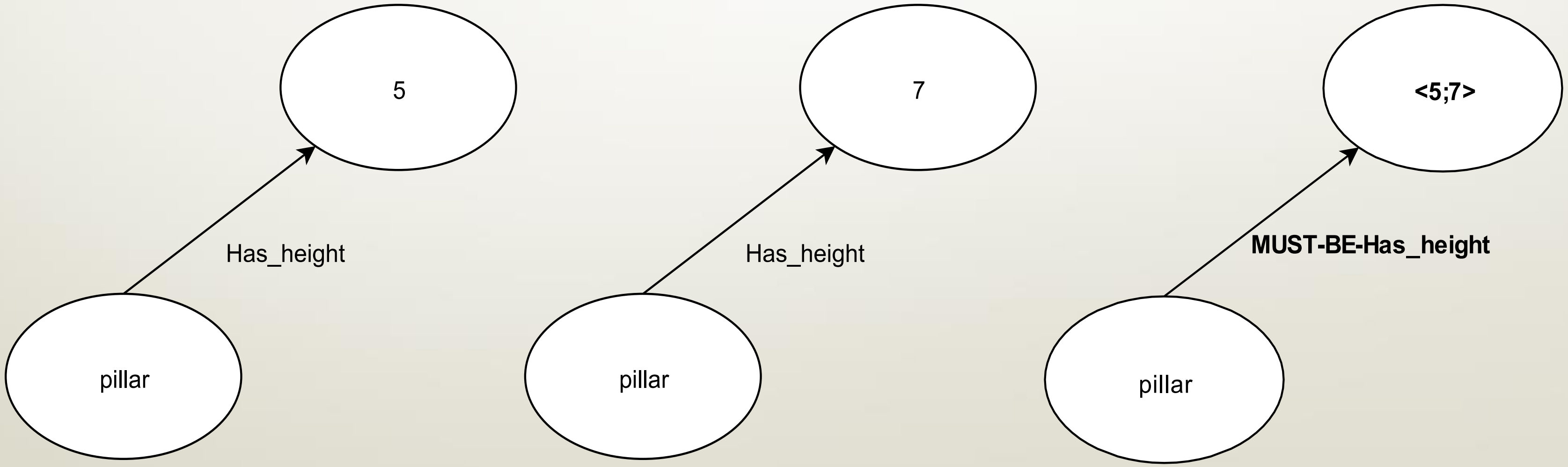
Generalization

- Generalization is used to refine hypothesis by a positive example

1. Compare the model hypothesis and the positive example to determine a difference
2. For each difference do
   a) if a link relation in the model points at a value that differs from the value in the example, then
      i. if the values in which the model and example differ have the most specific general class, use the **climb-tree**
      ii. else use **enlarge-set**
      iii. if these classes are excluding each other use **drop-link**
   b) if there is a link in the model that is missing in the example, use **drop-link**
   c) if the model and example differ at an numerical attribute value , use **close-interval**

Close-interval

- This heuristic is used if there are links in model and example with numerical value or interval



Generalization

- Generalization is used to refine hypothesis by a positive example

1. Compare the model hypothesis and the positive example to determine a difference
2. For each difference do
   a) if a link relation in the model points at a value that differs from the value in the example, then
      i. if the values in which the model and example differ have the most specific general class, use the **climb-tree**
      ii. if these classes are excluding each other use **drop-link**
      iii. else use **enlarge-set**
   b) if there is a link in the model that is missing in the example, use **drop-link**
   c) if the model and example differ at an attribute value , use **close-interval**
   d) else ignore example

Example of learning - types

- $x \rightarrow \iota$;
- $Pillar, White, Standing, Block, Tall, Short/(o\iota)_{\tau\omega}$;
- $True/(oo_{\tau\omega})_{\tau\omega}$
- $Colour, Shape, Size, Position/((o\iota)_{\tau\omega}\iota)_{\tau\omega}$: attributes, i.e. empirical functions that associate an individual with a property the individual has;
- $=/(o(o\iota)_{\tau\omega}\ (o\iota)_{\tau\omega})$: identity of properties.

Example of learning – positive examples

- Tall white standing object with a shape of block

$$\lambda w \lambda t \left[{}^0= \ {}^0Pillar\ \lambda w \lambda t \lambda x \left[\left[{}^0= \left[{}^0Colour_{wt}\ x \right] {}^0White \right] \wedge \left[{}^0= \left[{}^0Position_{wt}\ x \right] {}^0Standing \right] \wedge \left[{}^0= \left[{}^0Size_{wt}\ x \right] {}^0Tall \right] \wedge \left[{}^0= \left[{}^0Shape_{wt}\ x \right] {}^0Block \right] \right] \right]$$

- Tall standing object with a shape of block

$$\lambda w \lambda t \left[{}^0= \ {}^0Pillar\ \lambda w \lambda t \lambda x \left[\left[{}^0= \left[{}^0Size_{wt}\ x \right] {}^0Tall \right] \wedge \left[{}^0= \left[{}^0Position_{wt}\ x \right] {}^0Standing \right] \wedge \left[{}^0= \left[{}^0Shape_{wt}\ x \right] {}^0Block \right] \right] \right]$$

Example of learning – near-miss examples

- Tall white object with a shape of a block.

$$\lambda w \lambda t \left[{}^0{=}\ \ {}^0Pillar\ \lambda w \lambda t \lambda x\ [[{}^0{=}\ [{}^0Colour_{wt}\ x]\ {}^0White] \wedge [{}^0{=}\ [{}^0Size_{wt}\ x]\ {}^0Tall] \wedge [{}^0{=}\ [{}^0Shape_{wt}\ x]\ {}^0Block]] \right]$$

- Short standing object with a shape of a block.

$$\lambda w \lambda t \left[{}^0{=}\ \ {}^0Pillar\ \lambda w \lambda t \lambda x\ [[{}^0{=}\ [{}^0Size_{wt}\ x]\ {}^0Short] \wedge [{}^0{=}\ [{}^0Position_{wt}\ x]\ {}^0Standing] \wedge [{}^0{=}\ [{}^0Shape_{wt}\ x]\ {}^0Block]] \right]$$

Example of learning

- Initial hypothesis:

$$\lambda w \lambda t \left[{}^0{=}\ {}^0Pillar\ \lambda w \lambda t \lambda x\ \left[\left[{}^0{=}\ \left[{}^0Colour_{wt}\ x \right]\ {}^0White \right] \wedge \left[{}^0{=}\ \left[{}^0Position_{wt}\ x \right]\ {}^0Standing \right] \wedge \left[{}^0{=}\ \left[{}^0Size_{wt}\ x \right]\ {}^0Tall \right] \wedge \left[{}^0{=}\ \left[{}^0Shape_{wt}\ x \right]\ {}^0Block \right] \right] \right]$$

Example of learning - Specialization

- Near miss example: difference is in position -

$$\lambda w \lambda t \left[{}^0{=}\ {}^0Pillar\ \lambda w \lambda t \lambda x\ \left[\left[{}^0{=}\ \left[{}^0Colour_{wt}\ x \right]\ {}^0White \right] \wedge \left[{}^0{=}\ \left[{}^0Size_{wt}\ x \right]\ {}^0Tall \right] \wedge \left[{}^0{=}\ \left[{}^0Shape_{wt}\ x \right]\ {}^0Block \right] \right] \right]$$

- Require-link -> new hypothesis:

$$\lambda w \lambda t \left[{}^0{=}\ {}^0Pillar\ \lambda w \lambda t \lambda x\ \left[\left[{}^0{=}\ \left[{}^0Colour_{wt}\ x \right]\ {}^0White \right] \wedge \left[{}^0{=}\ \left[{}^0Size_{wt}\ x \right]\ {}^0Tall \right] \wedge \left[{}^0\boldsymbol{True}_{\boldsymbol{wt}}\ \lambda w \lambda t \left[{}^0{=}\ \left[{}^0Position_{wt}\ x \right]\ {}^0Standing \right] \right] \wedge \left[{}^0{=}\ \left[{}^0Shape_{wt}\ x \right]\ {}^0Block \right] \right] \right]$$

Example of learning - Generalization

- Positive example: difference is in color-

$$\lambda w \lambda t \left[{}^0{=}\ \ {}^0Pillar\ \lambda w \lambda t \lambda x\ \left[[\ {}^0{=}\ [\ {}^0Size_{wt}\ x]\ \ {}^0Tall] \wedge [\ {}^0{=}\ [\ {}^0Position_{wt}\ x]\ \ {}^0Standing] \wedge [\ {}^0{=}\ [\ {}^0Shape_{wt}\ x]\ \ {}^0Block] \right] \right]$$

- Drop-link -> new hypothesis:

$$\lambda w \lambda t \left[{}^0{=}\ \ {}^0Pillar\ \lambda w \lambda t \lambda x\ \left[[\ {}^0{=}\ [\ {}^0Size_{wt}\ x]\ \ {}^0Tall] \wedge \left[{}^0\boldsymbol{True}_{\ \boldsymbol{wt}}\ \lambda w \lambda t [\ {}^0{=}\ [\ {}^0Position_{wt}\ x]\ \ {}^0Standing] \right] \wedge [\ {}^0{=}\ [\ {}^0Shape_{wt}\ x]\ \ {}^0Block] \right] \right]$$

Example of learning - specialization

- Near-miss example:

$$\lambda w \lambda t \left[{}^0{=}\ {}^0Pillar\ \lambda w \lambda t \lambda x\ [[{}^0{=}\ [{}^0Size_{wt}\ x]\ {}^0Short] \wedge [{}^0{=}\ [{}^0Position_{wt}\ x]\ {}^0Standing] \wedge [{}^0{=}\ [{}^0Shape_{wt}\ x]\ {}^0Block]] \right]$$

- Conditions are not satisfied -> model is not modified:

$$\lambda w \lambda t \left[{}^0{=}\ {}^0Pillar\ \lambda w \lambda t \lambda x\ [[{}^0{=}\ [{}^0Size_{wt}\ x]\ {}^0Tall] \wedge [{}^0\mathbf{True}_{wt}\ \lambda w \lambda t [{}^0{=}\ [{}^0Position_{wt}\ x]\ {}^0Standing]] \wedge [{}^0{=}\ [{}^0Shape_{wt}\ x]\ {}^0Block]] \right]$$

Definition of an arch

$o, x, y, z \rightarrow \iota$;
$Arch, Pillar, Roof / (o\iota)_{\tau\omega}$;
$Composed\_of / (o\iota\iota\iota\iota)_{\tau\omega}$;
$Supports / (o\iota\iota)_{\tau\omega}$;
$Block, Polygon / (o\iota)_{\tau\omega}$;
$Shape, Position, Size /$
$((o\iota)_{\tau\omega}\iota)_{\tau\omega}$;
$Standing, Tall / (o\iota)_{\tau\omega}$;
$=_\iota / (o\iota\iota)$;
$= \ / (o(o\iota)_{\tau\omega} (o\iota)_{\tau\omega})$;
$\neg / (oo)$;

$$\lambda w \lambda t \Big[{}^0{=} \ {}^0Arch \ \lambda o \exists x \exists y \exists z \Big[[{}^0Composed\_of_{wt} \ o \ x \ y \ z]$$
$$\wedge [{}^0Pillar_{wt} \ x] \wedge [{}^0Pillar_{wt} \ y] \wedge [{}^0Roof_{wt} \ z] \wedge \big[{}^0\neg [{}^0{=}_\iota \ x \ y] \big]$$
$$\wedge [{}^0Supports_{wt} \ x \ z] \wedge [{}^0Supports_{wt} \ y \ z]$$
$$\wedge \big[{}^0True_{wt} \ \lambda w \lambda t [{}^0{=} \ [{}^0Position_{wt} \ x] \ {}^0Standing] \big]$$
$$\wedge \big[{}^0True_{wt} \ \lambda w \lambda t [{}^0{=} \ [{}^0Position_{wt} \ y] \ {}^0Standing] \big]$$
$$\wedge [{}^0{=} [{}^0Shape_{wt} \ x] \ {}^0Block] \wedge [{}^0{=} [{}^0Shape_{wt} \ y] \ {}^0Block]$$
$$\wedge [{}^0{=} [{}^0Shape_{wt} \ z] \ {}^0Polygon] \Big] \Big]$$

Sources

- Luger G. F.(2009): *Artificial intelligence: structures and strategies for complex problem solving*. 6th ed. Boston: Pearson Addison-Wesley, 2009. ISBN 978-0-321-54589-3.
- Poole D. L., Mackworth A. K.(2010). *Artificial intelligence: foundations of computational agents*. 2nd pub. Cambridge: Cambridge University Press, 2010. ISBN 978-0-521-51900-7.
- Winston P. H.(1992): *Artificial intelligence*. 3rd ed. Reading, Mass.: Addison-Wesley Pub. Co., 1992. ISBN 02-015-3377-4.

Thank you for your attention