Modular design patterns for neural-symbolic integration

Till Mossakowski

OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

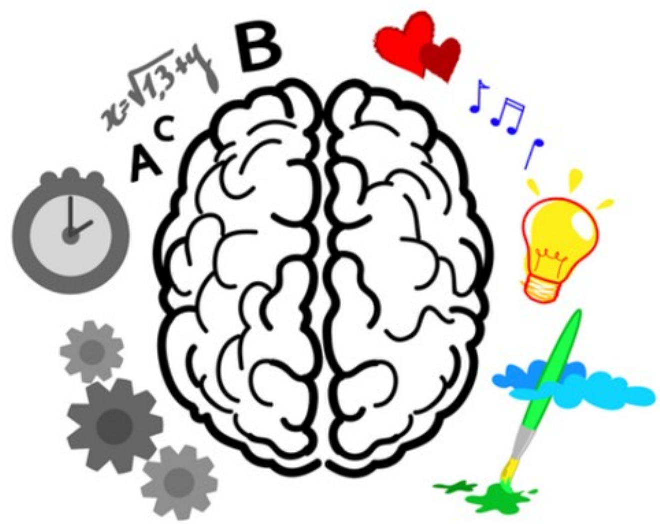
INF FAKULTÄT FÜR INFORMATIK

IFIP WG 1.3 meeting
Jan 18, 2022

The two brain hemispheres

Left Hemisphere

- Logic, ratio
- Language, numbers
- Analysis
- Time-bound
- Controlled
- Details



Right Hemisphere

- Emotions, intuition
- Images
- Synthesis
- Timeless
- Spontaneous
- Relations

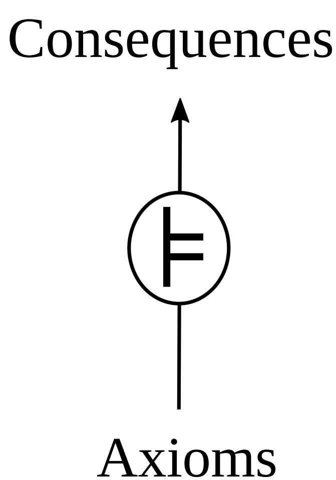
Neural networks versus symbolic reasoning

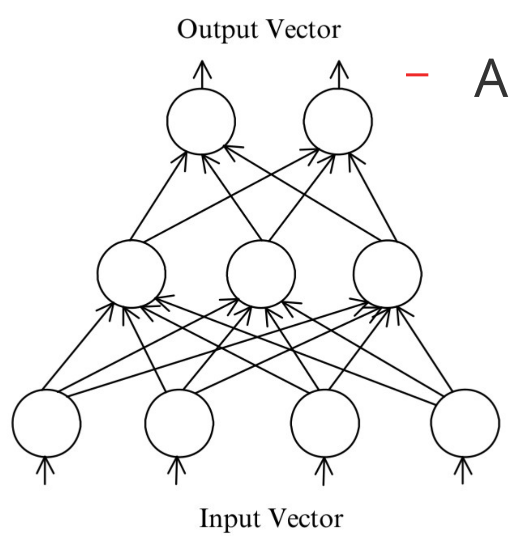
Symbolic reasoning

- "System 1"
- Slow
- Knowledge driven
- Abstraction
- Reliable
- Explainable

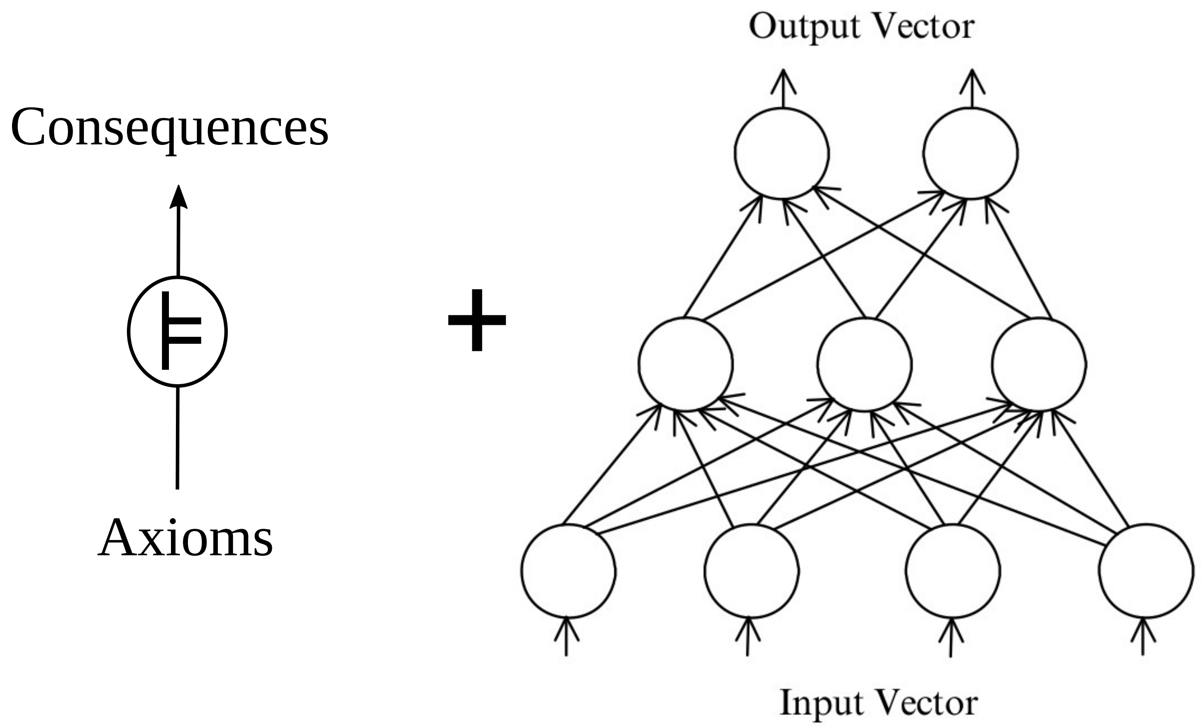
Neural networks

- "System 2"
- Fast
- Data driven
- Adapt to noisy data
- Flexible
- Automatic





Neural-symbolic systems



logical Reasoning

neural Networks

- combining the strengths of both worlds
- path to general AI (says IBM...)
- NeSy meetings, also at general AI conferences

Specification of neural-symbolic systems: why?

- high-level descriptions of the architectures of neural-symbolic systems
  - understand commonalities and differences between different systems
- bridge the gap between communities
  - knowledge-based systems
  - machine learning
  - cognitive science
- engineering neural-symbolic systems out of reusable components

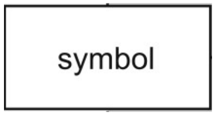
Specification of neural-symbolic systems: how?

- design patterns covering
  - symbols and data
  - neural and symbolic models

The subsequent presentation is based on the paper:
Michael van Bekkum, Maaike de Boer, Frank van Harmelen, André Meyer-Vitali, Annette ten Teije: Modular design patterns for hybrid learning and reasoning systems
Applied Intelligence, June 2021
doi.org/10.1007/s10489-021-02394-3

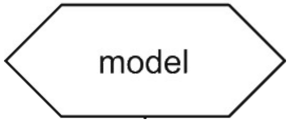
… and followed by some preliminary own work

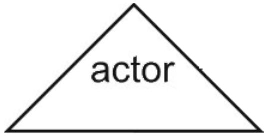
Elements of design patterns

- Instances
  - Symbols 

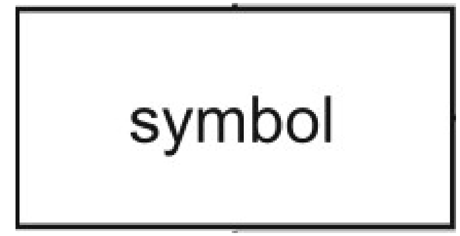
  - Data 

- Models
  - Semantic
  - Statistical



- Processes 
- Actors 

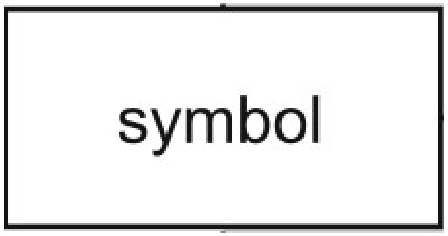
Symbols



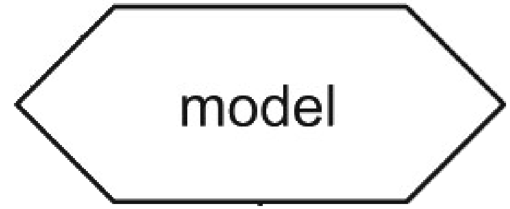
- 1) a symbol must designate an object, a class or a relation in the world, called the *interpretation* of the symbol;
- (2) symbols can be either *atomic* or *complex*, in which case they are composed of other symbols according to a formal set of compositional rules; and
- (3) there must be system of *operations* that, when applied to a symbol, generates new symbols, that again must have a designation.
- → This means that symbols are terms!

Instances

- Symbols (in the sense of the previous slide),
  - Logical formulas and relations: $p \land q \models p$
  - Labels
  - State machines
  - Relations between data items
- Data (everything else)
  - Numbers, text
  - Images, sound, other sensor input
  - Streams of data



Models

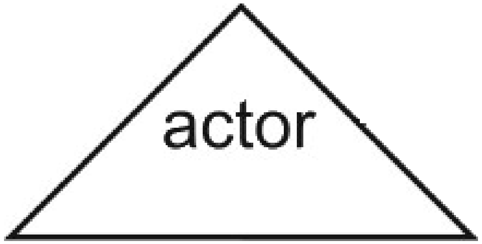


- Statistical models:
  dependencies between statistical variables
  - (Deep) Neural Networks
  - Bayesian Networks
  - Markov Models
- Semantic models / knowledge graphs
  - Taxonomies
  - Ontologies
  - Rule bases
  - Differential Equations

Processes

- Processes =
  steps that lead from inputs to results
  - generation of instances and models
  - transformation of instances and models
  - inferencing thereupon
    - Induction (from instances to model)
    - Deduction (from model to instances)

Actors



- autonomous (to different degrees)
- following intentions and goals
- Interaction of actors
  - collaboration, negotiation or competition
- Only rarely used in patterns so far

Generating a model

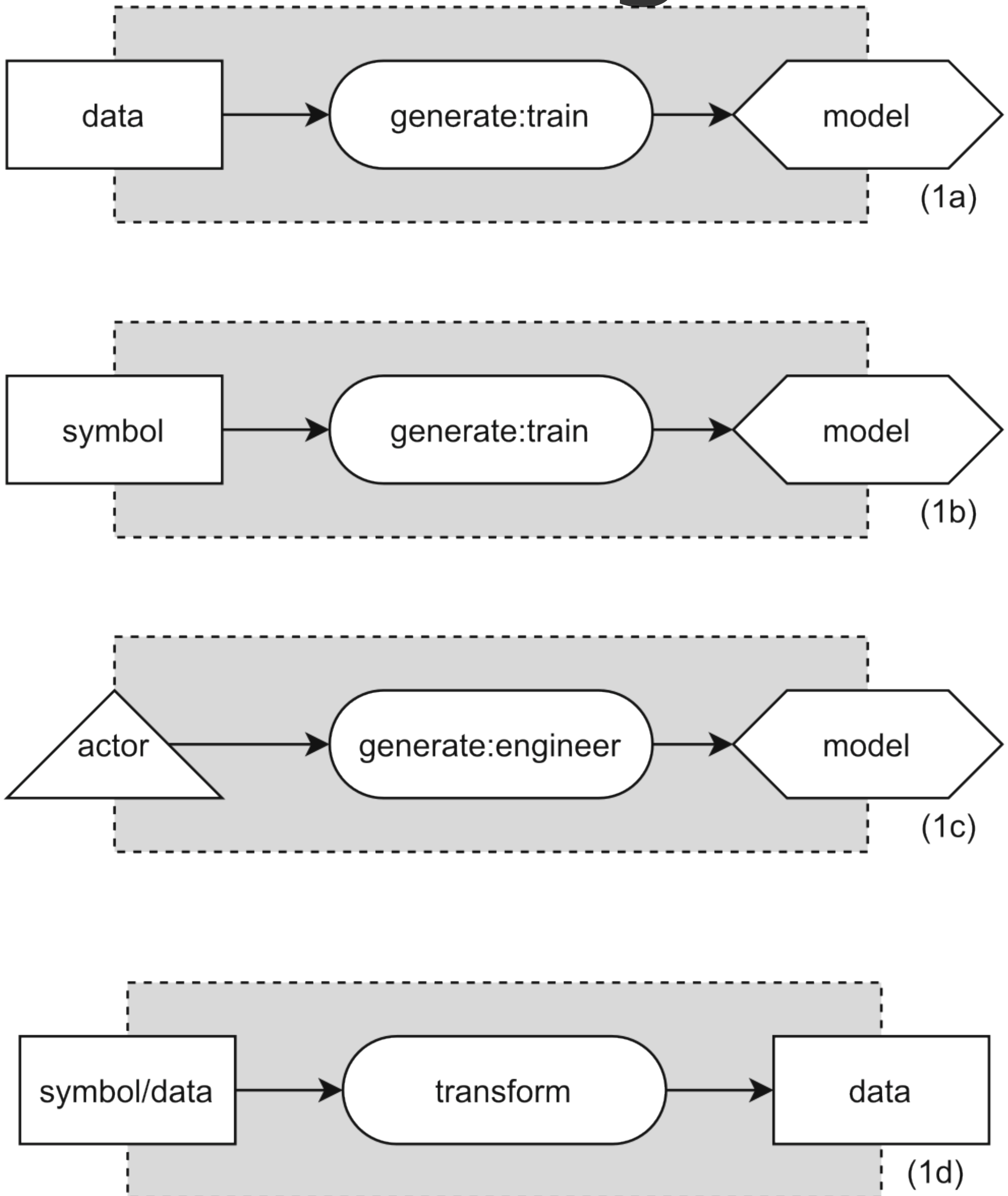


Fig. 1 Elementary patterns to generate a model

- Neural networks
- Bayesian learning
- Inductive logic programming
- Automata learning
- Manually designed logical theories
- Data preprocessing

Using a model (deduction)

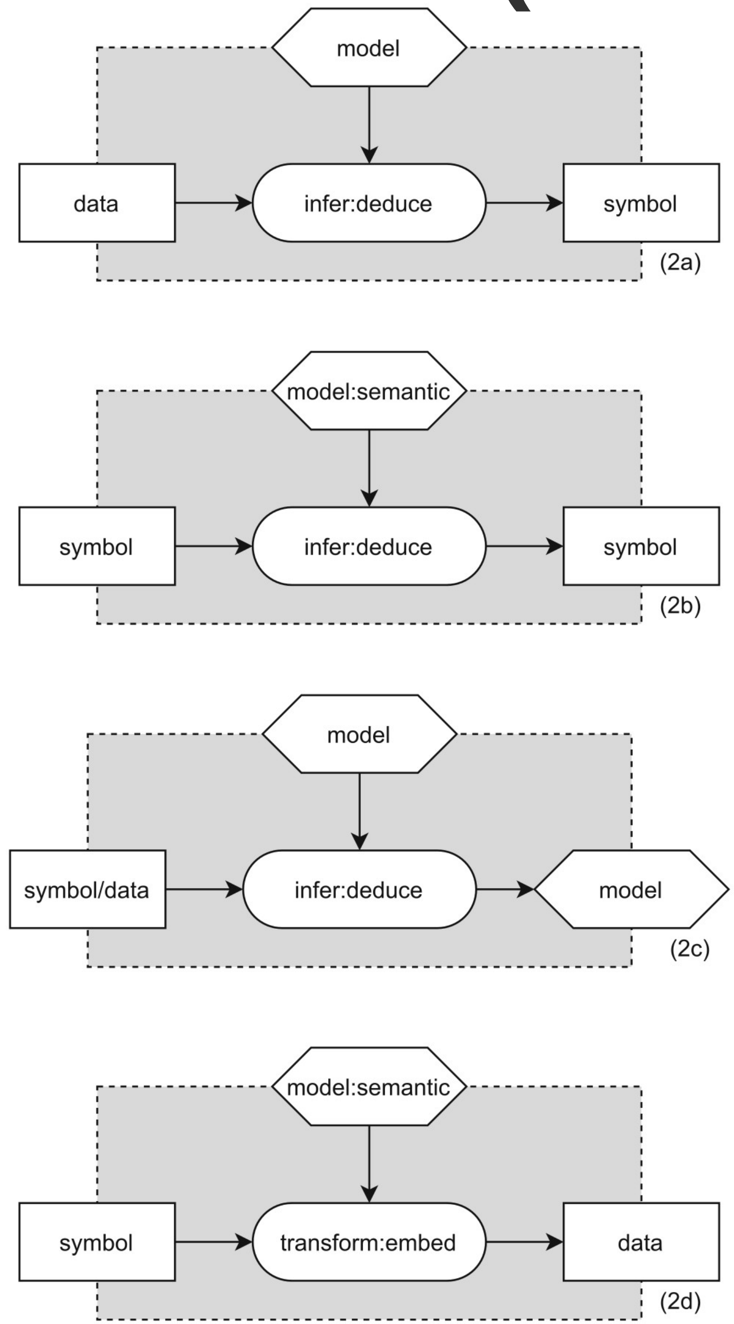
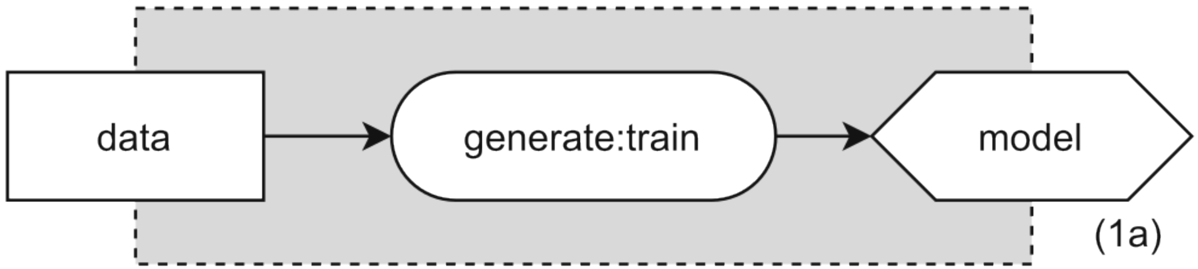


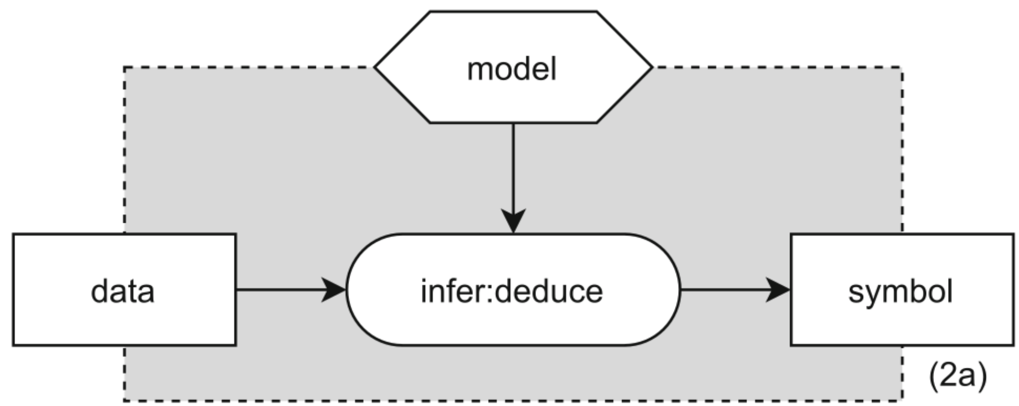
Fig. 2 Elementary patterns to use a model

Machine Learning Systems

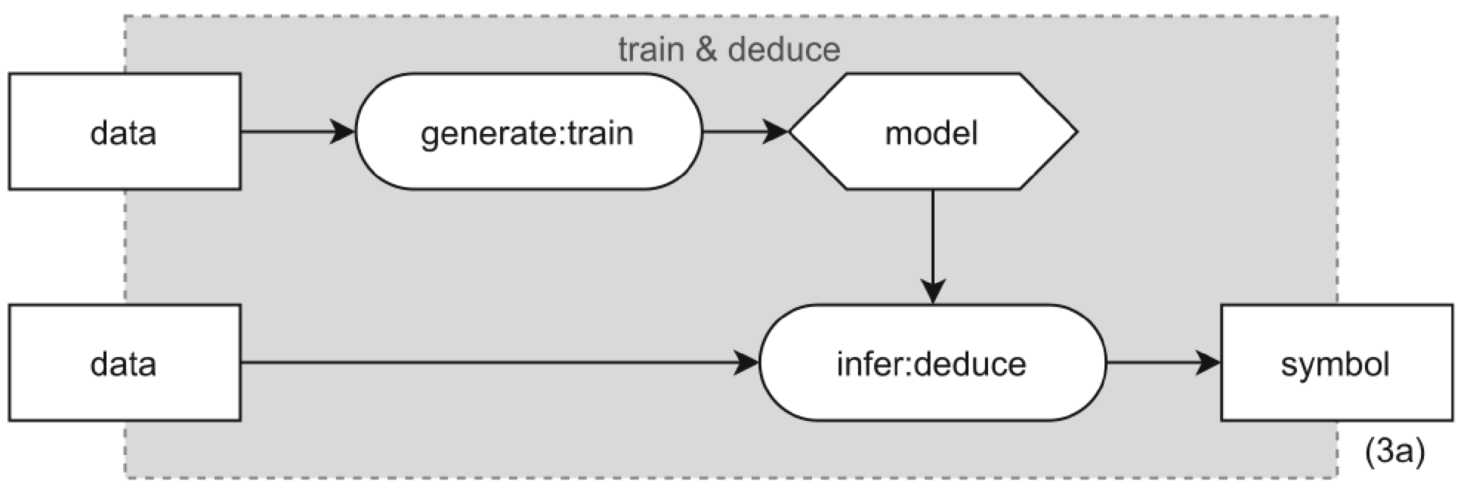
- e.g. neural networks



+

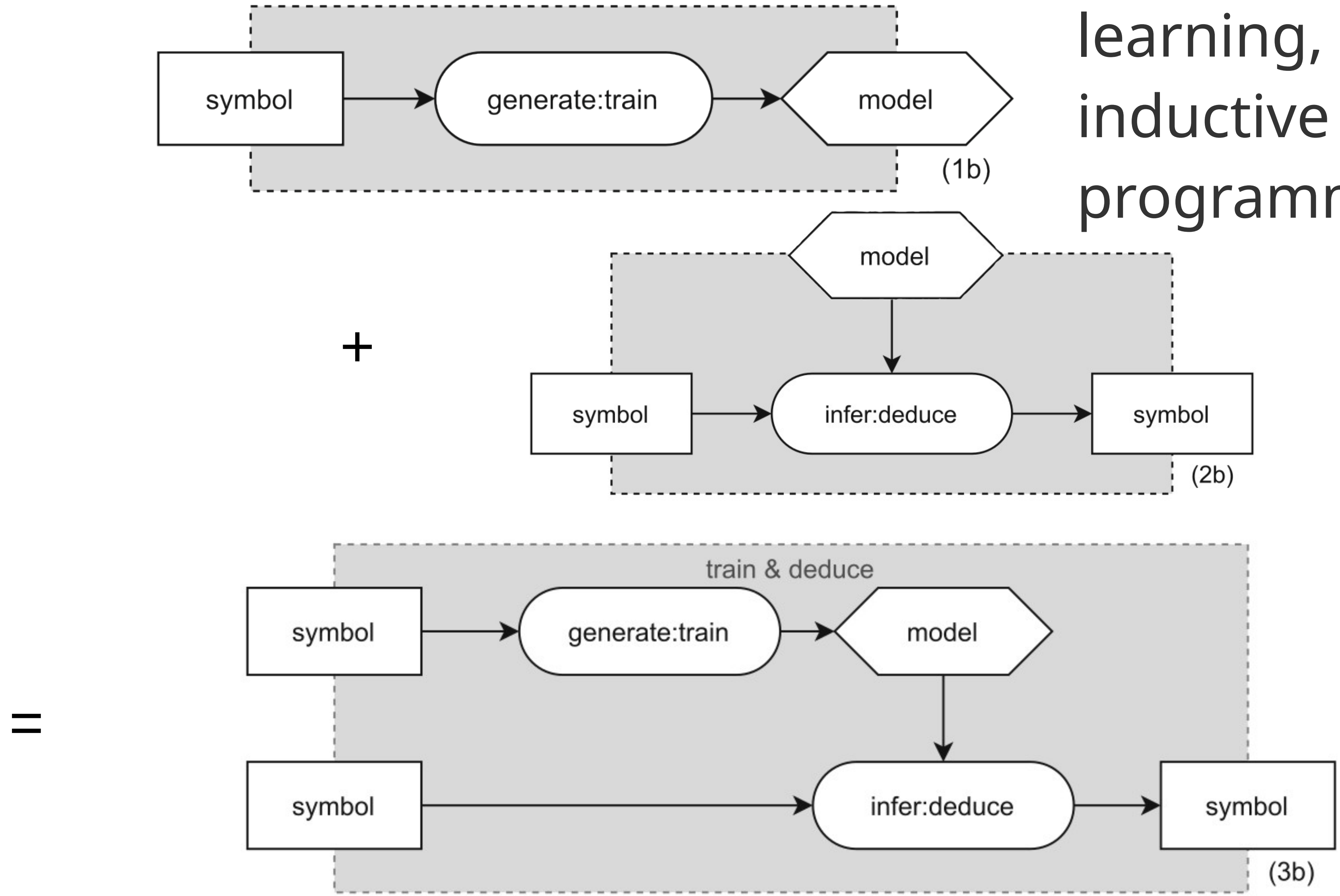


=



Knowledge Representation systems

- e.g. automata learning, inductive logic programming



Hybrid systems

- Hybrid system = everything else defined using the patterns
- Systems can be classified
- Systems can be developed
  - Starting from abstract patterns
  - Stepwise refinement

Explainability through an extra semantic model

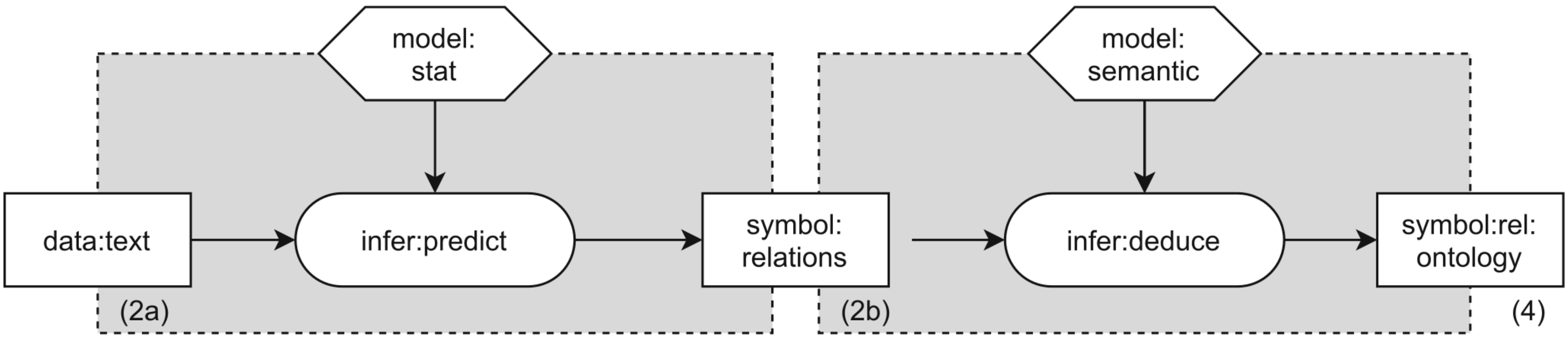
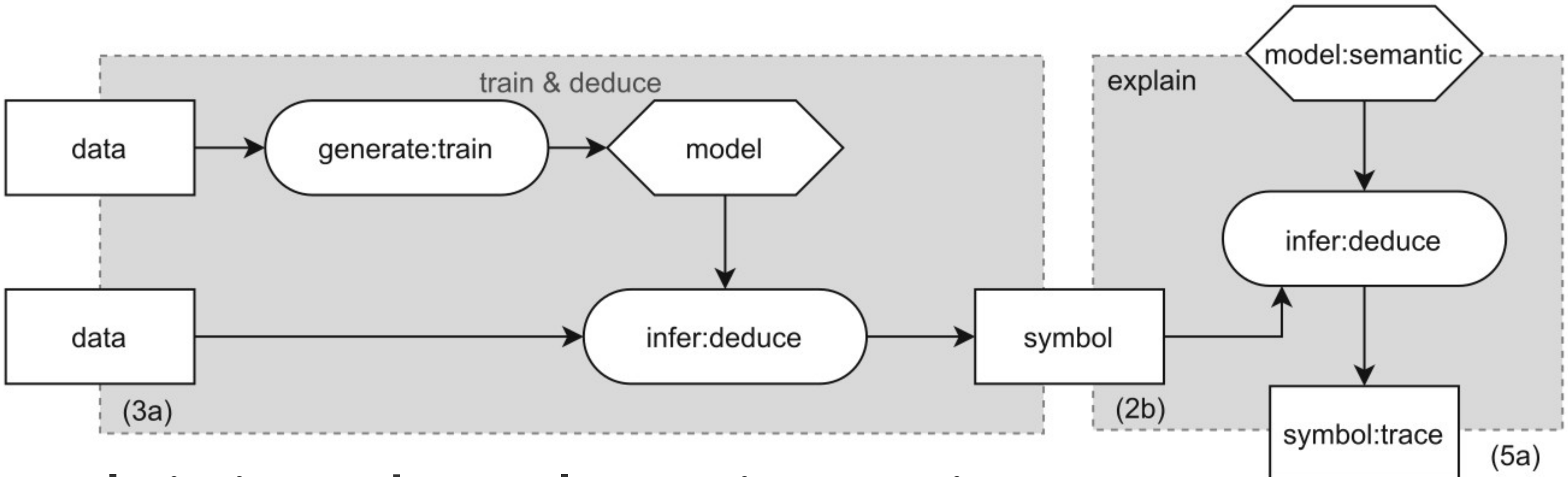


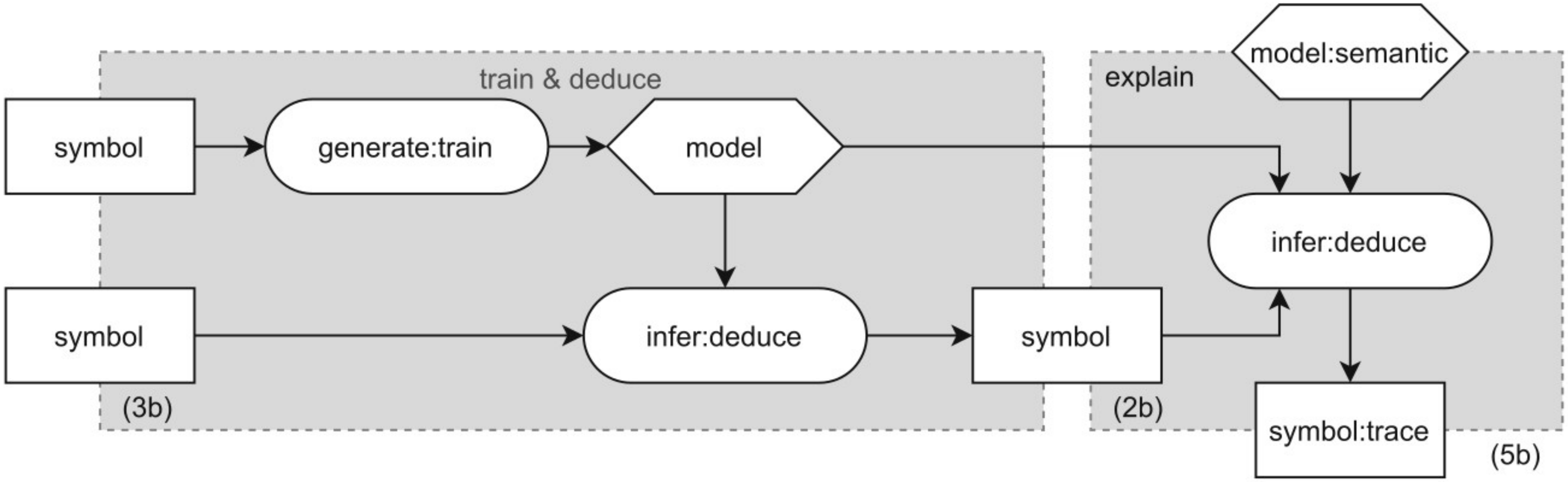
Fig. 4 Pattern for learning with symbolic output

- e.g. ontology learning from text

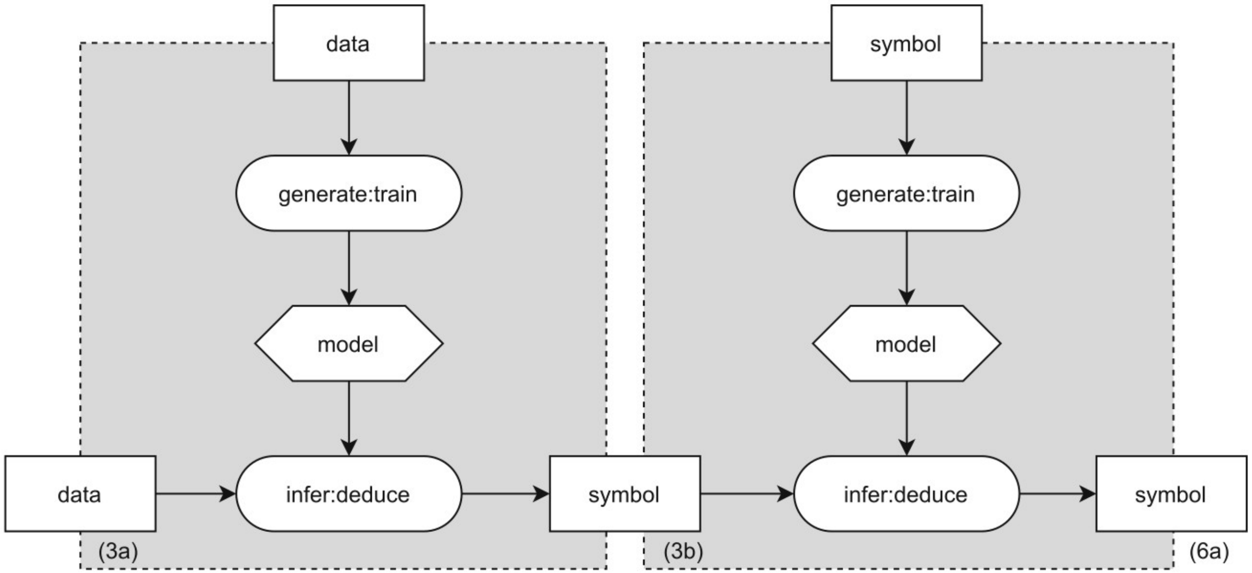
Explainable learning systems through rational reconstruction

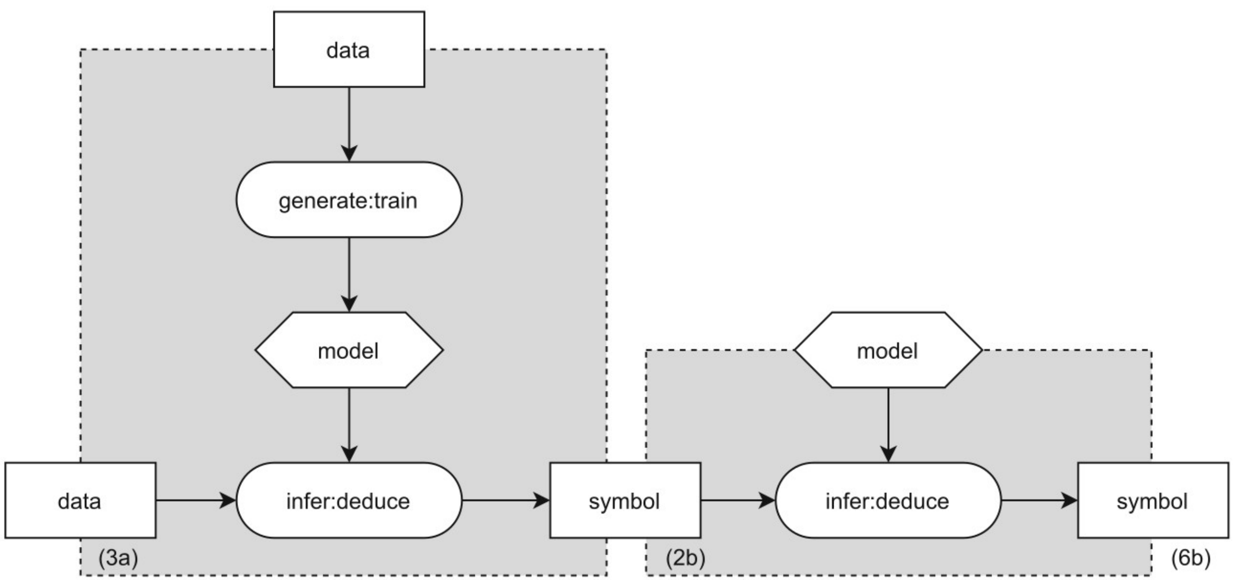


- e.g. explaining deep learning using DL reasoner



Learning intermediate abstractions

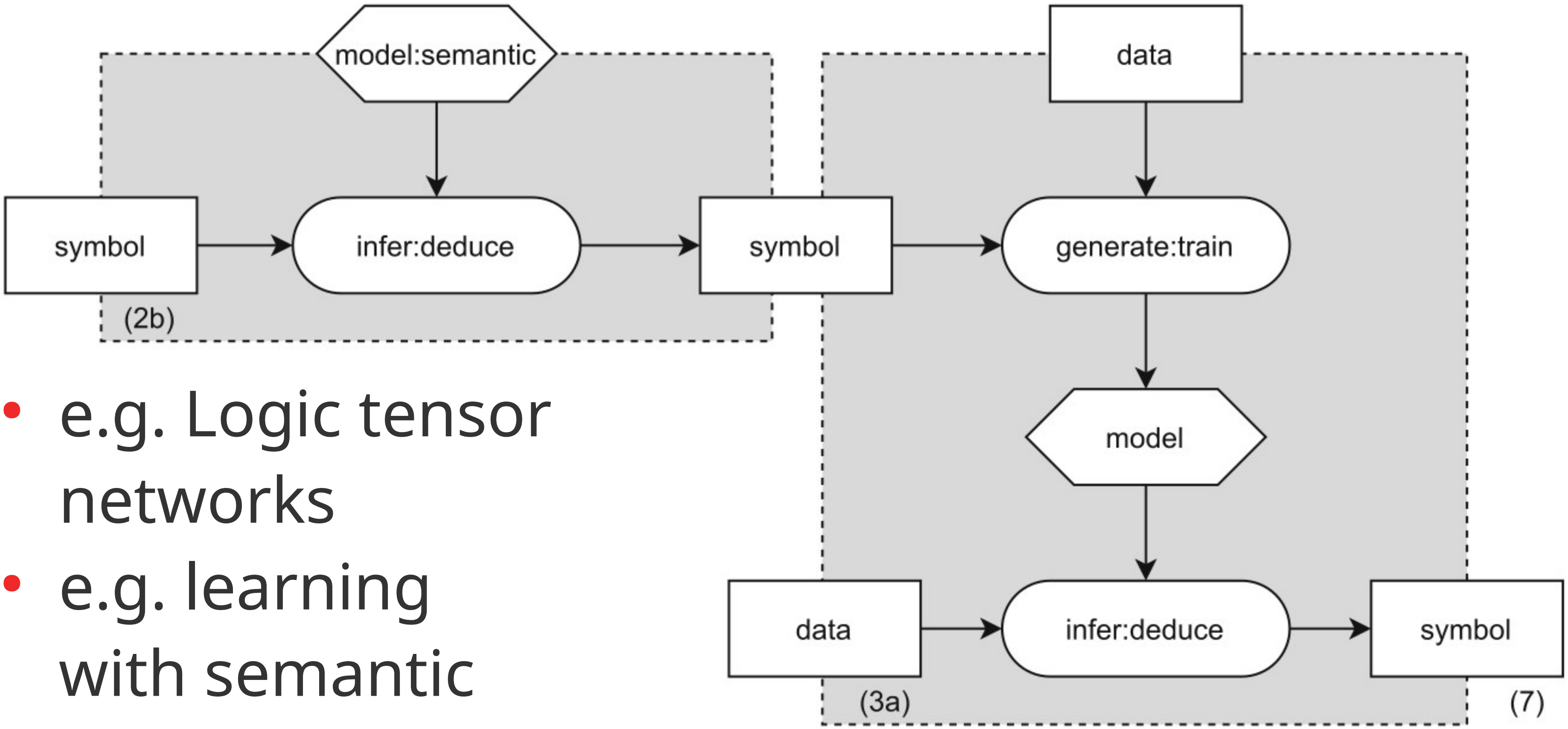




- e.g. DeepProbLog: addition of handwritten numbers

- e.g. AlphaGo

Informed learning with prior knowledge



- e.g. Logic tensor networks
- e.g. learning with semantic loss function
- More than 100 systems discussed in:
  Von Rueden L, et al. (2019) Informed machine learning–towards a taxonomy of explicit integration of knowledge into machine learning. Learning 18:19–20

Link prediction / graph completion

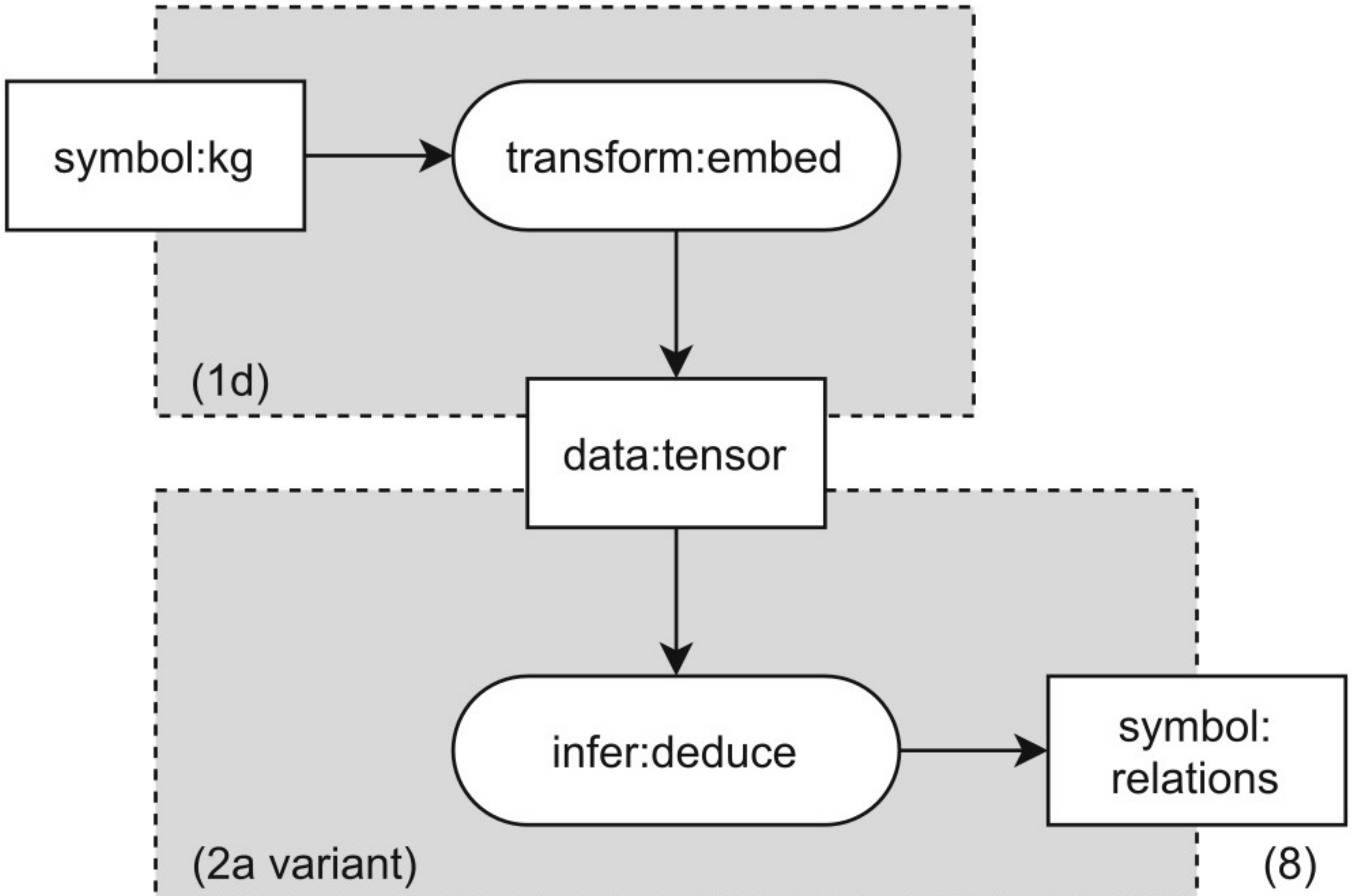


Fig. 8 From symbols to data and back again

Tensorising symbols

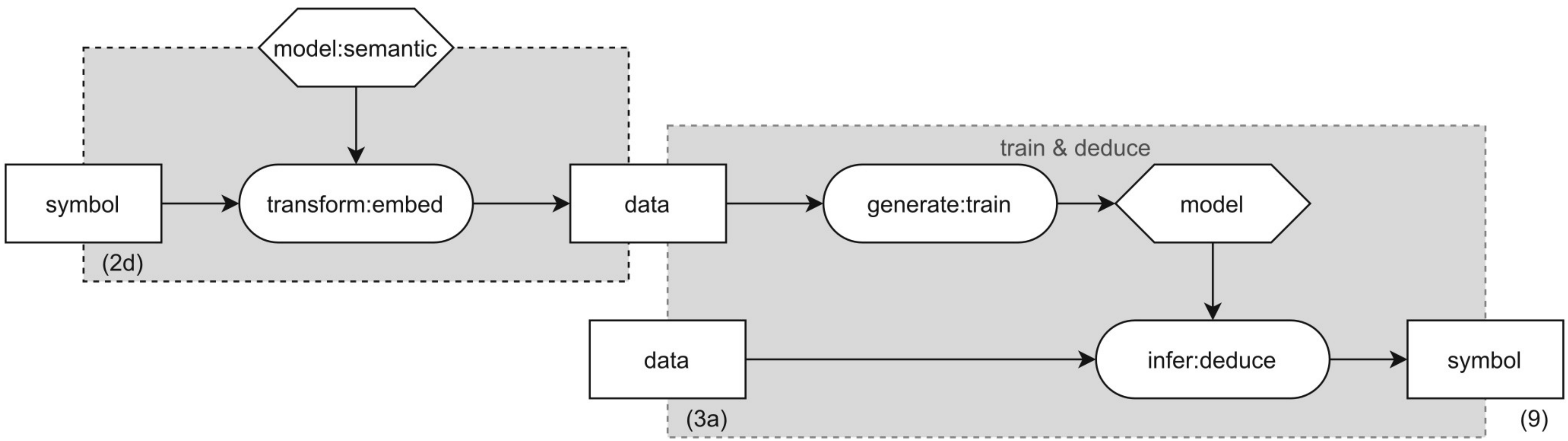
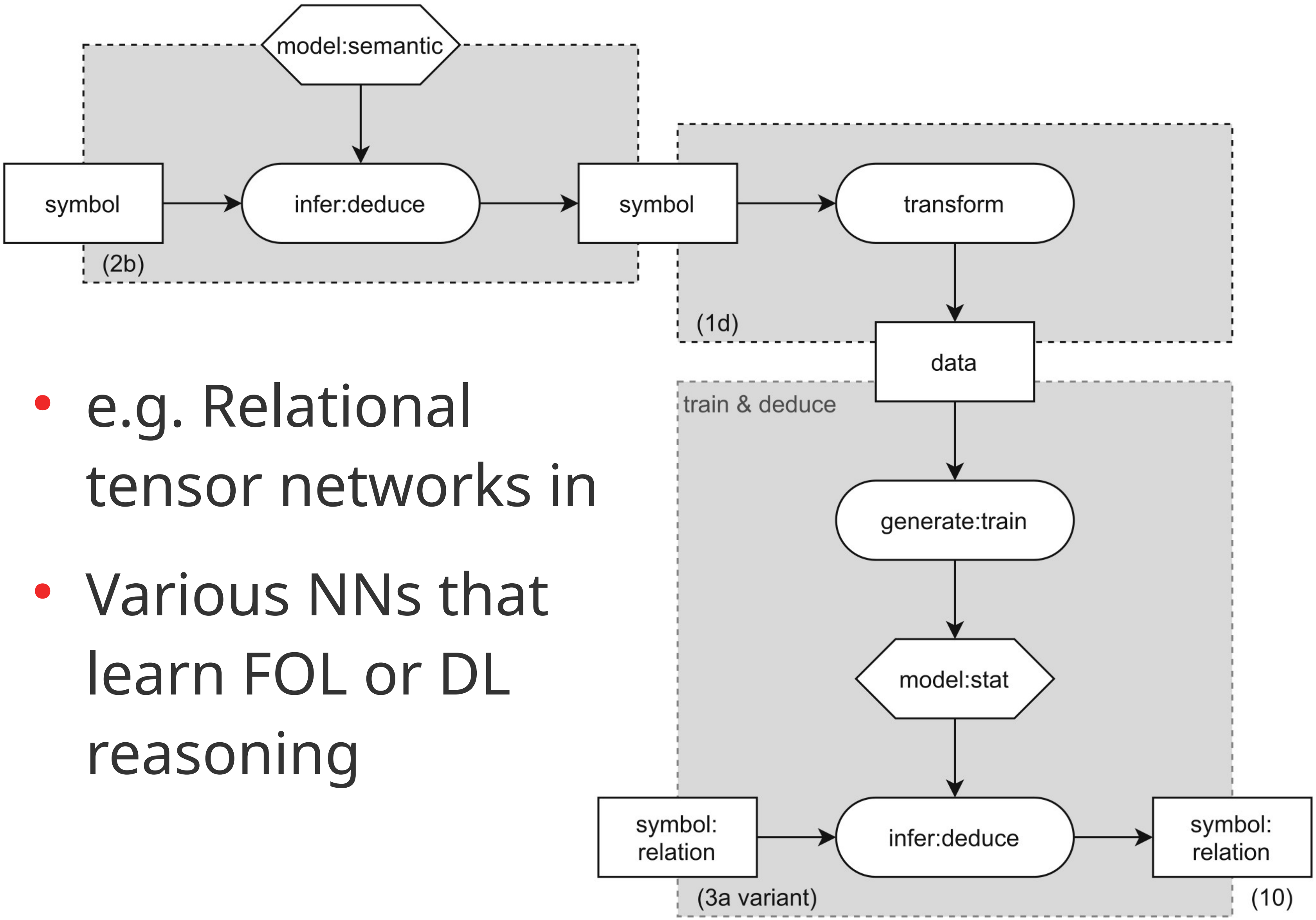


Fig. 9 Pattern for learning logical representations for statistical inferencing

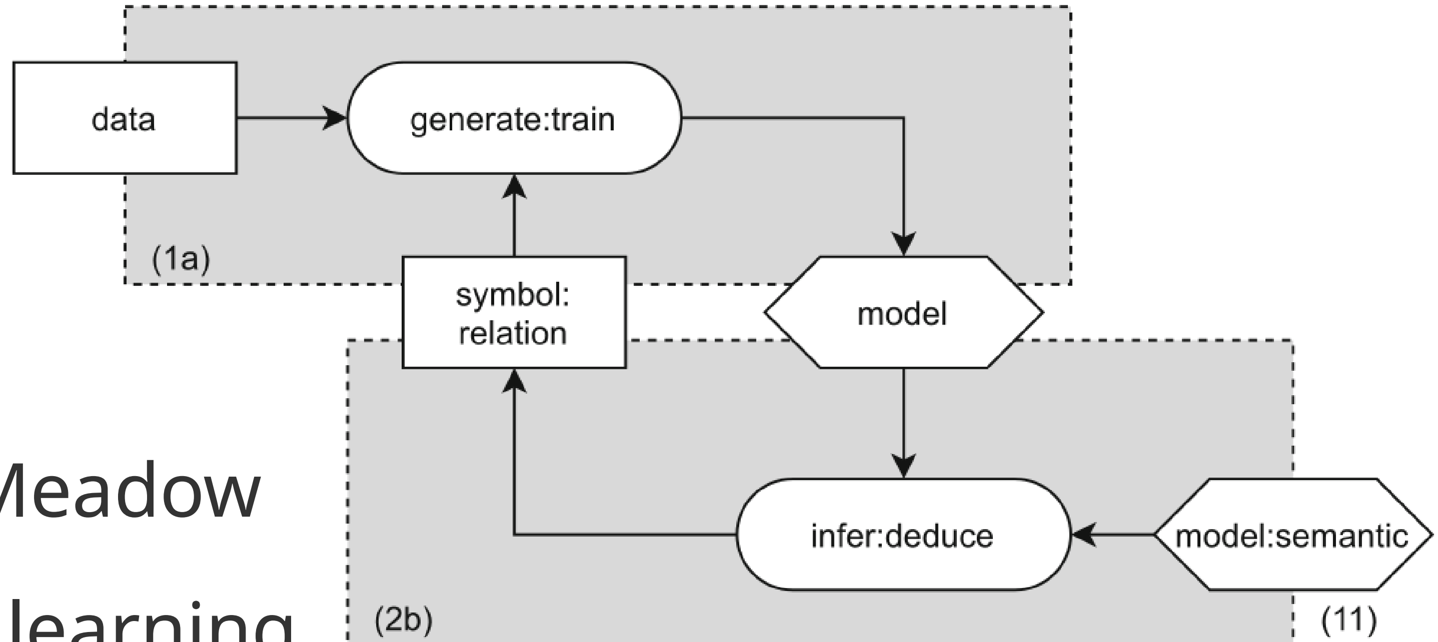
- e.g. Logic tensor networks in combination with convolutional neural networks
- e.g. graph neural networks

Learning to reason



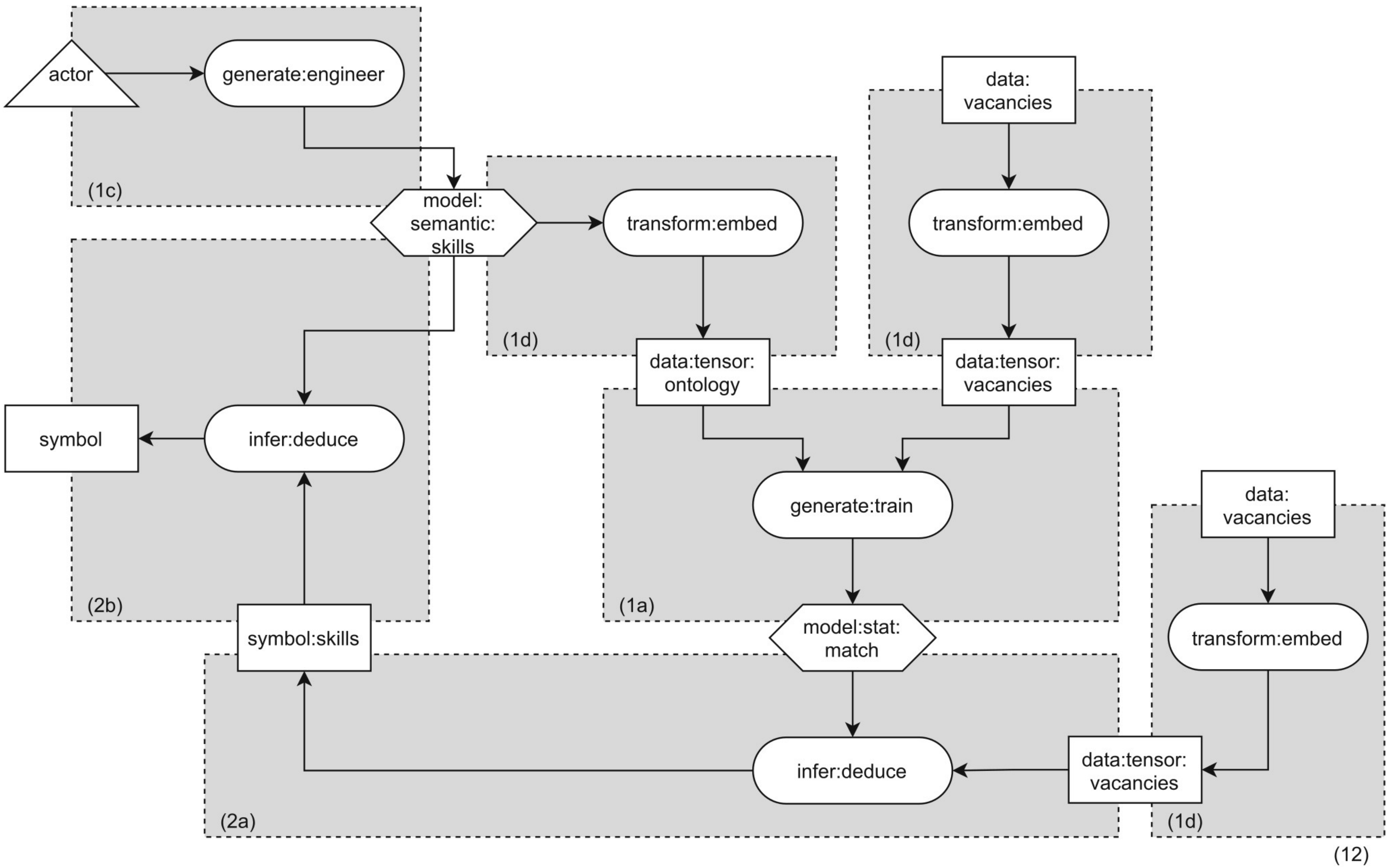
- e.g. Relational tensor networks in
- Various NNs that learn FOL or DL reasoning

Meta-reasoning for control

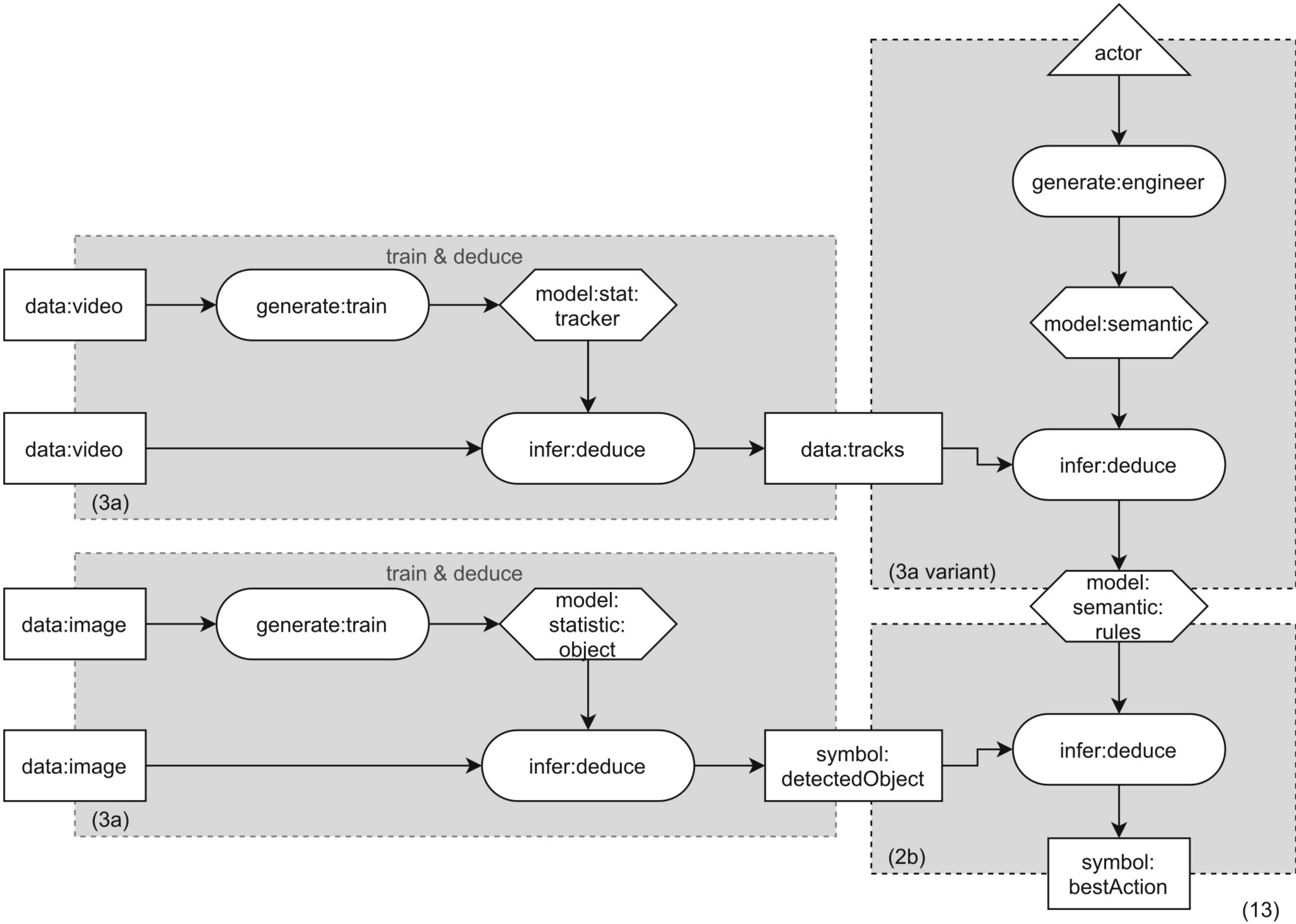


- e.g Alpine Meadow
- Curriculum learning

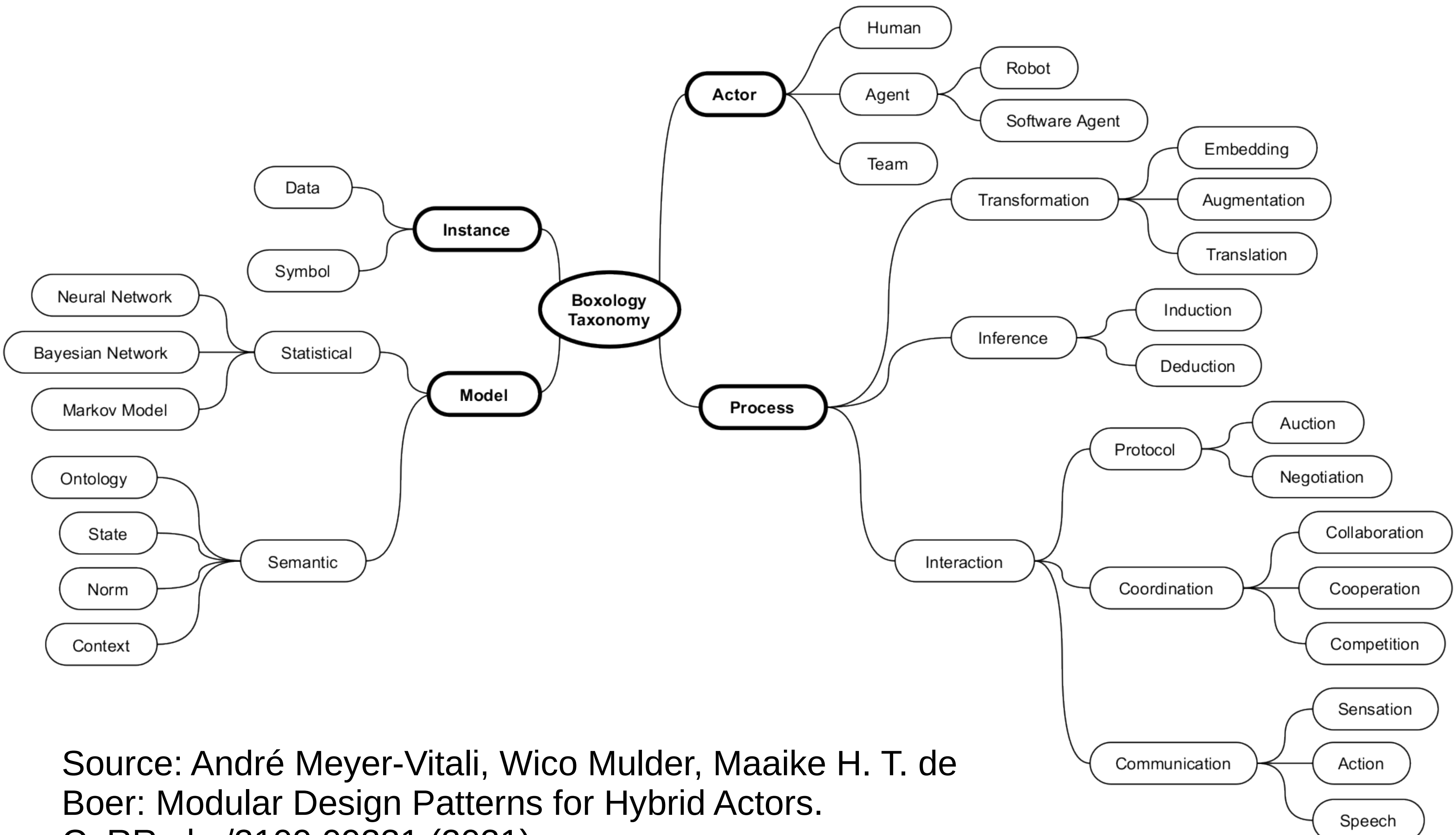
Use-case: skills matching



Use-case: robot in action



Taxonomy of elements



Source: André Meyer-Vitali, Wico Mulder, Maaike H. T. de Boer: Modular Design Patterns for Hybrid Actors. CoRR abs/2109.09331 (2021)

Beyond the pattern paper

- The pattern paper lists as future work: Coherent methodology for AI systems
  - **Grammar and logic for composing and verifying patterns**
  - Trustworthy by design
  - Guidelines for implementation of concepts

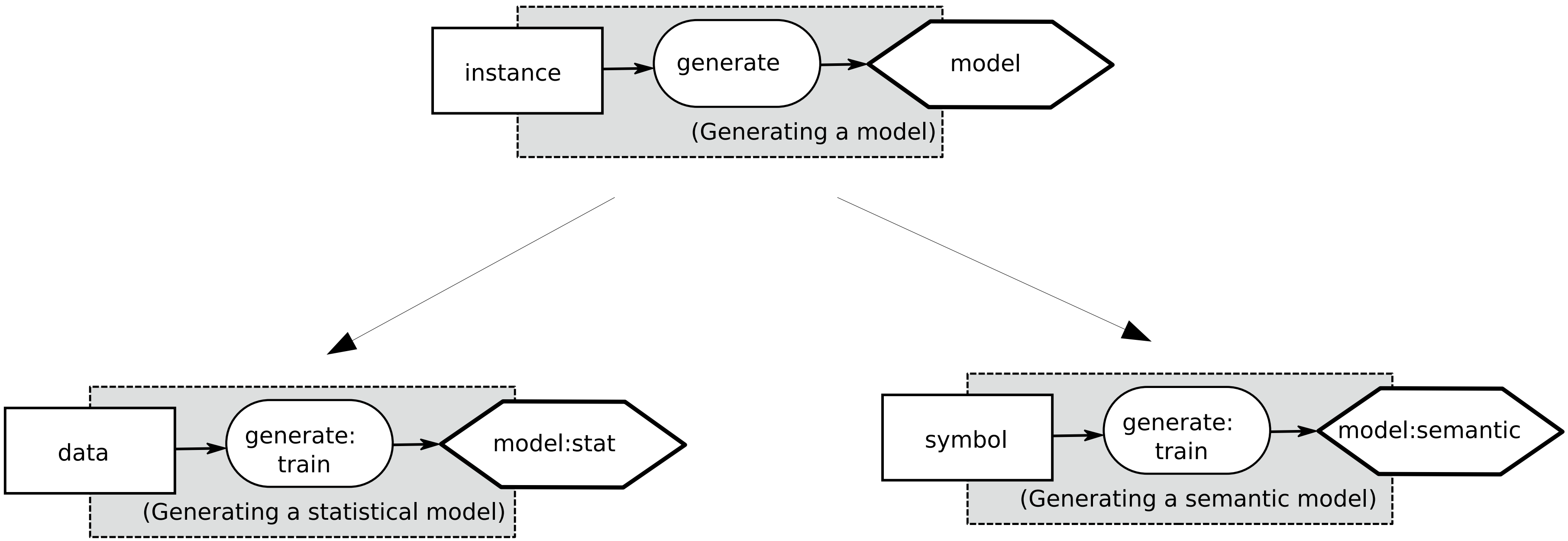
Neural-symbolic design patterns, formally

- Definition. A **neural-symbolic design pattern (NeSy pattern)** is a directed graph, with node labels = non-top classes from the taxonomy
- Given a NeSy pattern,
  - **input nodes** = nodes without ingoing edges
  - **output nodes** = nodes without outgoing edges

Refinement of neural-symbolic design patterns

- Definition. Given NeSy patterns P_1 and P_2, a **refinement** $\varphi : P_1 \to P_2$ is
  - a morphism of unlabled graphs $\varphi : P_1 \to P_2$, such that for each node $n \in P_1$,
    - $lab(\varphi(n)) \leq lab(n)$ in the taxonomy.

Refinement of neural-symbolic design patterns: example



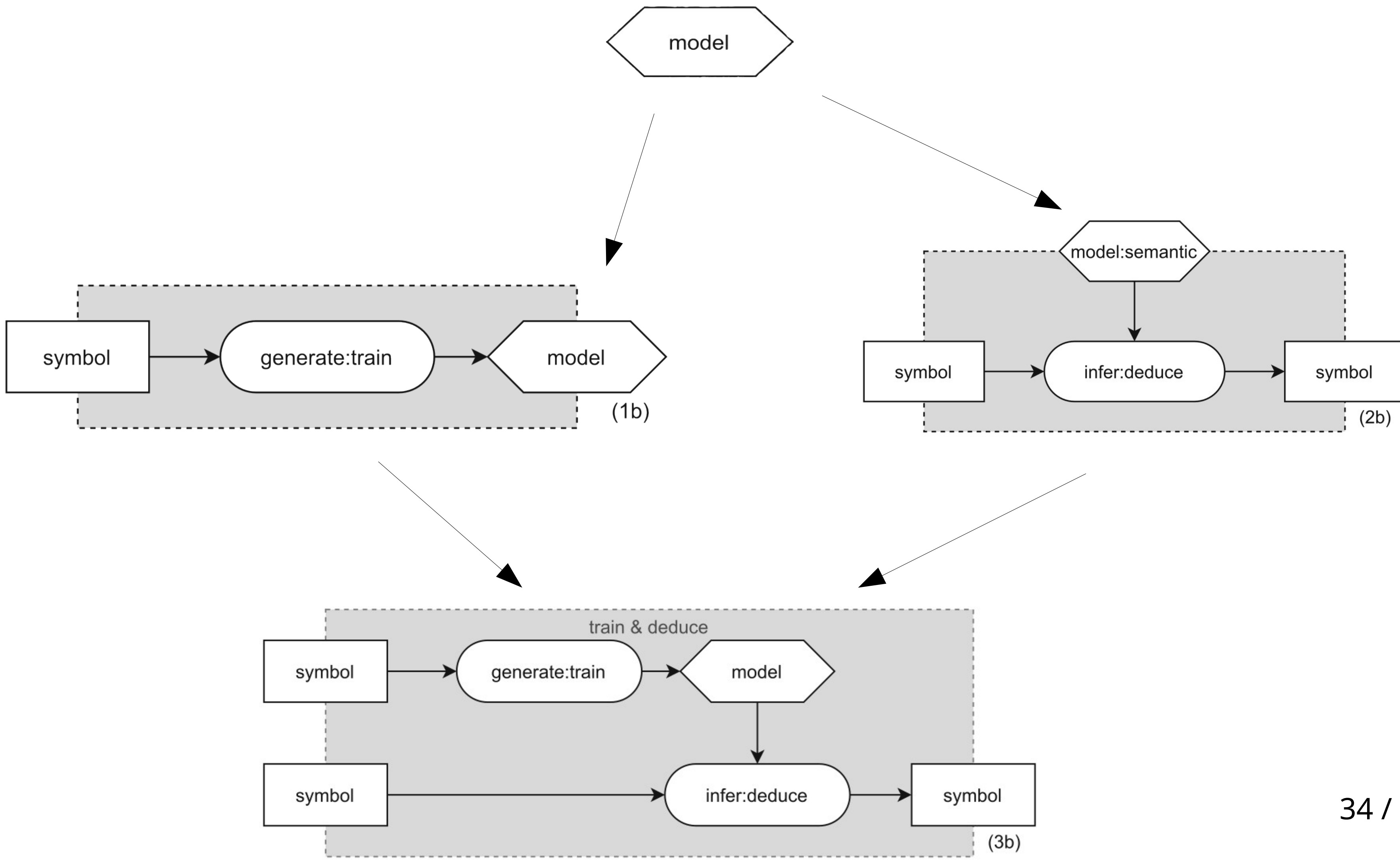
Combinations of NeSy patterns

- Definition. Given a diagram of NeSy patterns and refinements, its **combination** is the colimit in the category of NeSy patterns and refinement.
- It is built as follows:
  - The (unlabeled) directed graph is the colimit C in the category of directed graphs.
  - For a node $n \in C$, let $N = \bigcup_{i \in I} \mu_i^{-1}(n)$,

    where $(\mu_i)_{i \in I}$ are colimit injections. Then

    - $lab(n) = \inf_{m \in N} lab(m)$.

Combinations of NeSy patterns, example (ILP, automata training)



Combinations of NeSy patterns

- Usually, we are interested in combinations of diagrams where
  - only input and output nodes are identified,
  - only input nodes are refined.

Towards a model theory

- How can we interpret the different elements?
  - Symbol: term of some signature Σ
  - Data: vector in $\mathbb{R}^n$
  - Statistical model: transformation $\mathbb{R}^m \rightarrow \mathbb{R}^n$
  - Semantic model: function $T_{\Sigma 1} \rightarrow T_{\Sigma 2}$
  - Inference process: application of model
- $\rightarrow$ NeSy patterns need to be enriched with Σ, n

Conclusion

- We have recalled a graphical language for neural-symbolic design patterns
- We have started formalising these patterns
- So far covered:
  - Signatures, signature morphisms
  - First ideas for model theory

Future work

- A logic for specification and verifcation
  - Formalised as an institution?
  - DOL/Hets can be used to compute colimits
  - DOL alignments as friendly input syntax for diagrams
- Different types of refinement
  - For development
  - For composition
- Use string diagrams and lenses?
- Cooperations are welcome!