Production System

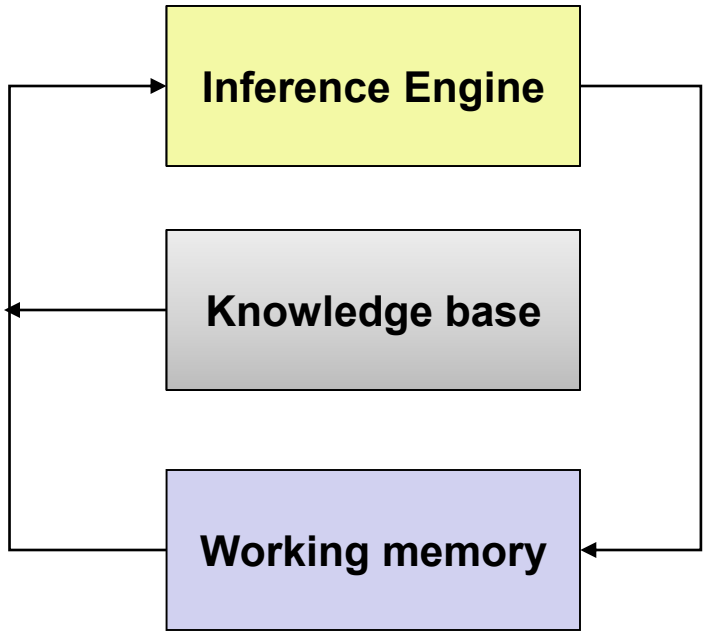
Topics of this lecture

- Production system
- Inference engine
- Working memory
- Knowledge base
- Pattern matching
- Conflict resolution
- Forward inference
- Back inference

- プロダクションシステム
- 推論エンジン
- 作業メモリ（短期メモリ）
- 知識ベース（長期メモリ）
- パターン照合
- 競合解消
- 前向き推論
- 後ろ向き推論

Production system

- Production system (or production rule system) was proposed by Allen Newell in 1973.
- A production system typically contains three components
  - Inference engine.
  - Knowledge base (Long-term memory), and
  - Working memory (Short-term memory).



The idea of separating the knowledge and the inference engine is useful for designing different expert systems.

Working memory

- The working memory is also called ***short-term memory.***
- It contains the observed data for making decisions, and the intermediate results (derived data) ***produced*** by the inference engine.

Example:

- f1: x has black strings
- f2: x has white body color
- f3: x has hoofs
- f4: x has odd number of toes
- f5: x drink milk

A child is often considered as "clever" if he/she has a good short memory capability.

Knowledge base (1)

(part of Table 4.1 in p. 66)

Rule name	Condition	Action
M1	Has body hair	Add “x is a mammal”
M2	Drink milk	Add “x is a mammal”
B1	Has feather	Add “x is a bird”
B2	Can fly AND lay eggs	Add “x is a bird”
B3	Is bird AND body size is large AND outlook is white	Add “x is a swan”
B4	is bird AND body size is small AND outlook is black	Add “x is a swallow”
B5	Is bird AND body size is moderate AND outlook is black	Add “x is a crow”

Knowledge base (2)

(part of Table 4.1 in p. 66)

Rule name	Condition	Action
C1	x is a mammal AND x eats meat	Add “x is a carnivore”
C2	x is a mammal AND x has sharp teeth AND x has sharp claws	Add “x is a carnivore”
C3	x is a carnivore AND x has brown body color AND x has big body	Add “x is a lion”
C4	x is a carnivore AND x has brown body color AND x has medium sized body	Add “x is a fox”

Knowledge base (3)

(part of Table 4.1 in p. 66)

U1	X is a mammal AND x has hoofs	Add “x is an ungulate”
U2	X is an ungulate AND x has an even number of toes	Add “x is an even-toed ungulate”
U3	X is an ungulate AND x ruminates	Add “x is an even-toed ungulate”
U4	X is an ungulate AND x has an odd number of toes	Add “x is an odd-toed ungulate”
U5	X is an even-toed ungulate AND x has brown body color AND x has black spots	Add “x is a deer”
U6	X is an odd-toed ungulate AND x has white body color AND x has black strings	Add “x is a zebra”

Definition of knowledge

- In a production system, a rule is usually defined as follows:

  - **Rule name**
  - **If (condition)**
  - **Then (Action)**

- Examples of actions: add a new datum, delete an old datum, replace an existing datum, etc.

Inference engine

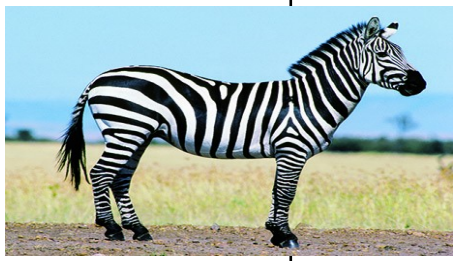
- The inference engine derives a result based on the knowledge in the knowledge base and the data in the working memory.
- The process for deriving a result is called inference or reasoning.
- **Forward reasoning**: Derive intermediate results using the observed data, and the last one is the final result.
- **Backward reasoning**: Make a hypothesis first, and verify or prove the hypothesis using the data.

Forward reasoning

- Step 1: Put the observed data into the working memory.
- Step 2: Pattern matching
  - Find a set C of rules that satisfy the observed data. This set C is called the conflict set.
- Step 3: Conflict resolution
  - Select a rule r from C based on some criteria, and
  - Do the action specified by the selected rule r.
  - If the result satisfies a given criterion, stop; otherwise, return to Step 2.

Step 2 and Step 3 together are called Recognition-Action Cycle (RAC)

Example 4.1:
Reasoning based on data given in Table 4.3

Cycles	Conflict Set	Selected rule	Status of the working memory
0			f1: x has black strings f2: x has white body color f3: x has hoofs f4: x has odd number of toes f5: x drink milk
1	M2	M2	f6：x is a mammal
2	M2,U1	U1	f7：x is an ungulate
3	M2,U1,U4	U4	f8：x is an odd-toed ungulate
4	M2,U1,U4,U6	U6	f9：x is a zebra
5	M2,U1,U4,U6	---	

Main problems in forward reasoning

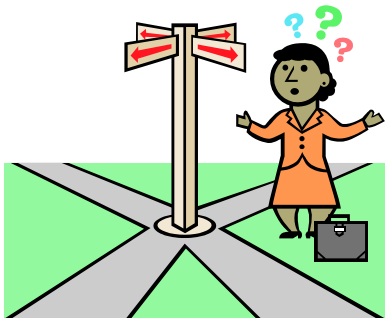
- Computational cost for pattern matching is high
  - All data and all conditions of all rules must be compared with each other in each cycle.
- Solution
  - Use **Rete algorithm** or its improved version.

- Rule selection effects the reasoning efficiency
  - Random selection or simple selection (e.g. depth first) may increase the redundancy of the reasoning process.
- Solution
  - Use **heuristics** (e.g. LEX)

LEX (Lexicographic sort)

1. Delete the “used” rules from the conflict set;
2. Assign higher priorities to rules that matches the newer data;
3. Assign higher priorities to rules with more detailed conditions;
4. Assign equal priorities otherwise.

Physical meaning?



Forward reasoning is best-first search

- Problem formulation
  - State (node): current status of the working memory.
  - State transition (edge): updating the working memory based on the selected rule.
  - Node expansion: finding the conflict set.
  - Heuristics = LEX
- If search is not successful, it is necessary to go back to some parent node, and search along a different path.

Backward reasoning

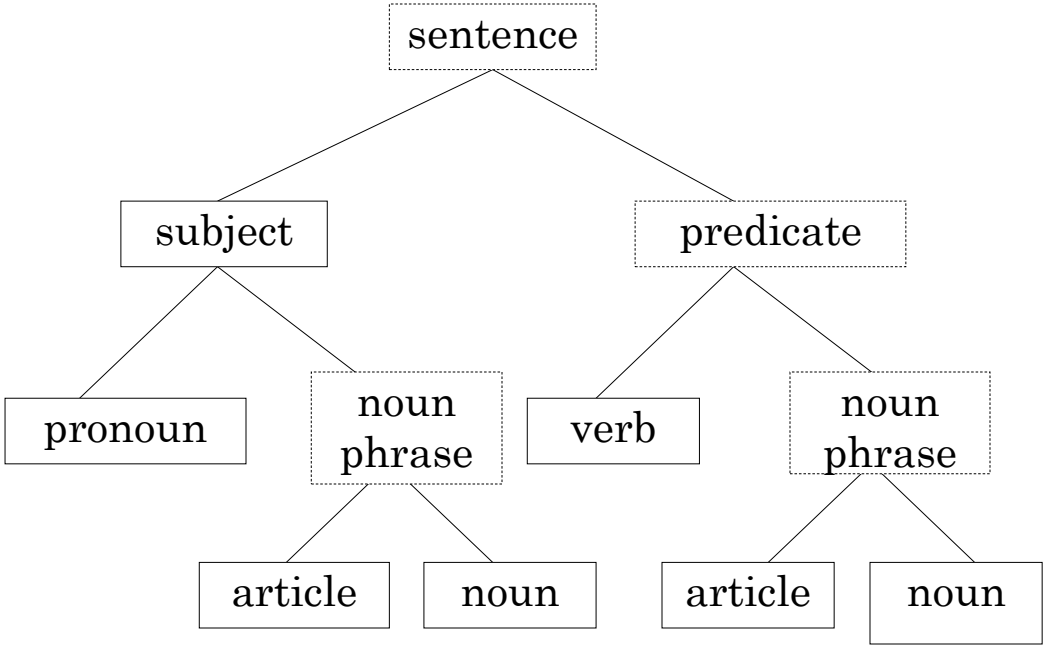
- Step 1: Find a set A of rules from the knowledge base. For any rule in A, its action part derives the given hypothesis h.
- Step 2: If A is empty, and return “False”.
- Step 3: Take out a rule from A, check its condition part, and find a set B of conditions that do not match the observed data. If B is empty, return “True”.
- Step 4: For each condition in B, verify its truth recursively.
- Step 5: If all returned values in Step 4 are “True”, return “True”; otherwise, return to Step 2.

Example 4.3 p. 74

cycle	Hypothesis to verify (conditions in B)	Rule in A	Data used
1	x is a zebra	U6	f1, f2
2	x is an odd-toed ungulate	U4	f4
3	x is an ungulate	U1	f3
4	x is a mammal	M2	f5

AND/OR tree

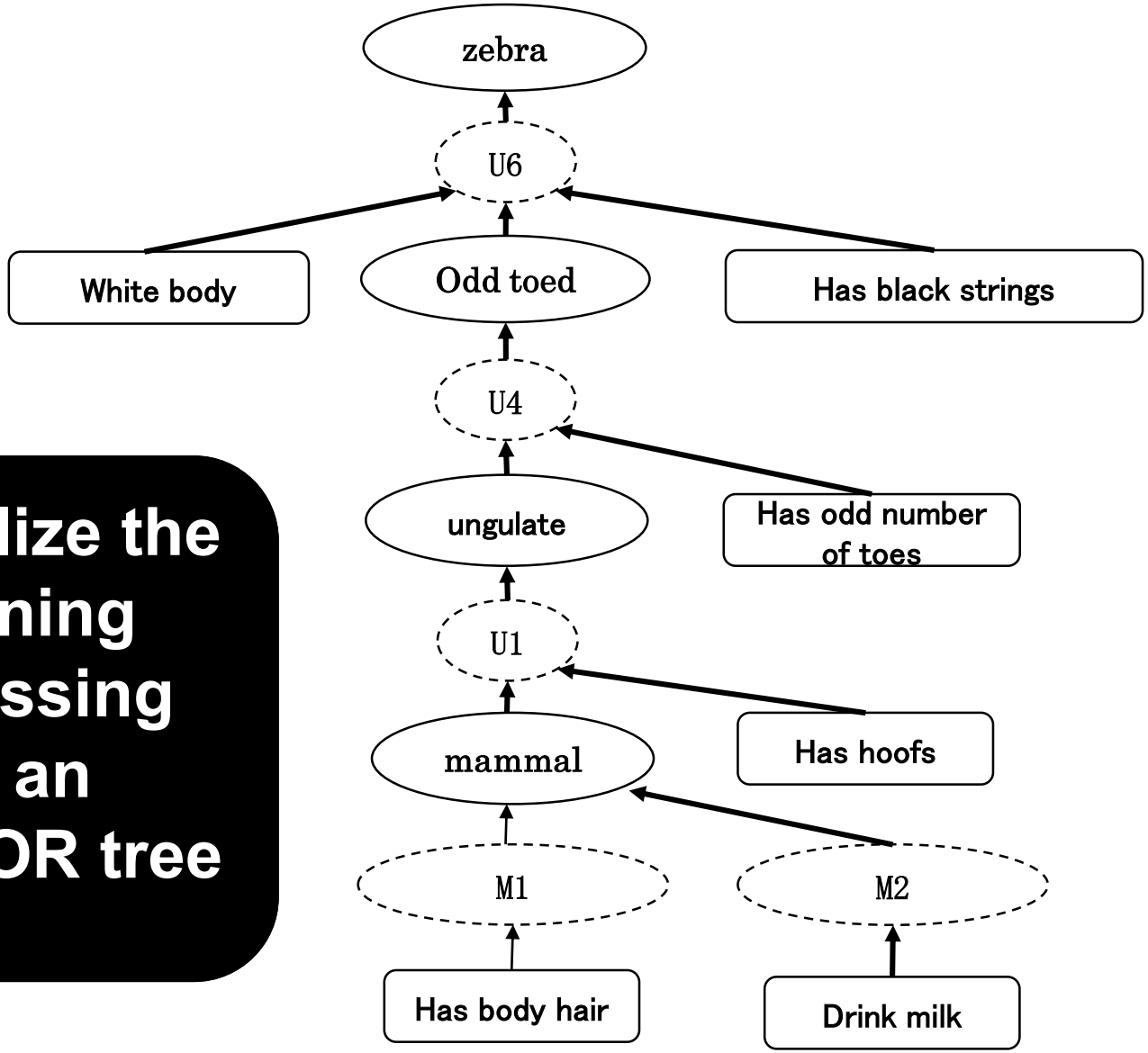
- To visualize backward reasoning, we can use AND/OR tree.
- There are two types of nodes in an AND/OR tree
  - AND node (dotted) an arc): true if all child nodes are true;
  - OR node (solid): true if any of the child nodes is true.



Ex1: The man made a desk

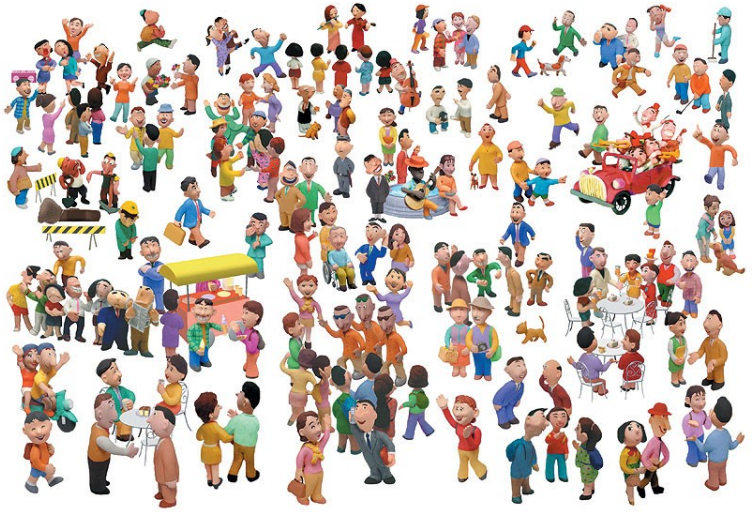
Ex2: She reads a book

Visualize the reasoning processing using an AND/OR tree



Features of production systems

- The knowledge can be used in free style.
  - It is not necessary to think about the order of the rules, nor the relations between the rules,
- It is easy to update the knowledge base
  - A rule can be added or deleted without effecting other rules.



Problems of production system

- Because the rules are not well organized,
  - it is difficult to understand the relations between rules, and
  - it is not efficient for reasoning.

Homework for lecture 5 (1)

- Solve Ex. 4.1 given in p. 71 of the textbook. The observed data are given as follows
  - X has body hair.
  - X has sharp teeth.
  - X has sharp claws.
  - X has a brown (ocher) body color.
  - X has a medium sized body
- Find the type of this animal based on the knowledge based given in Table 4.1 (p. 66), and summarize the results in the same form as Table 4.6.
- Submit the result (in hardcopy) to the TA within the exercise class.

Homework for lecture 5 (2)

- Complete a program for forward reasoning based on the skeleton.
- Confirm your program using the data given in Table 4.3 and the knowledge given in Table 4.1 in the textbook.
- You may also test the program using some other data sets, and see if the derived results are correct.

Quizzes of today

- What are the main components of a production system?
- How to define a “rule” in a production system?
- What is “forward reasoning” or “forward inference”?

- Try to explain the physical meaning of the LEX strategy for conflicting resolution.
- Draw an AND/OR tree for the sentence “The man made a desk”