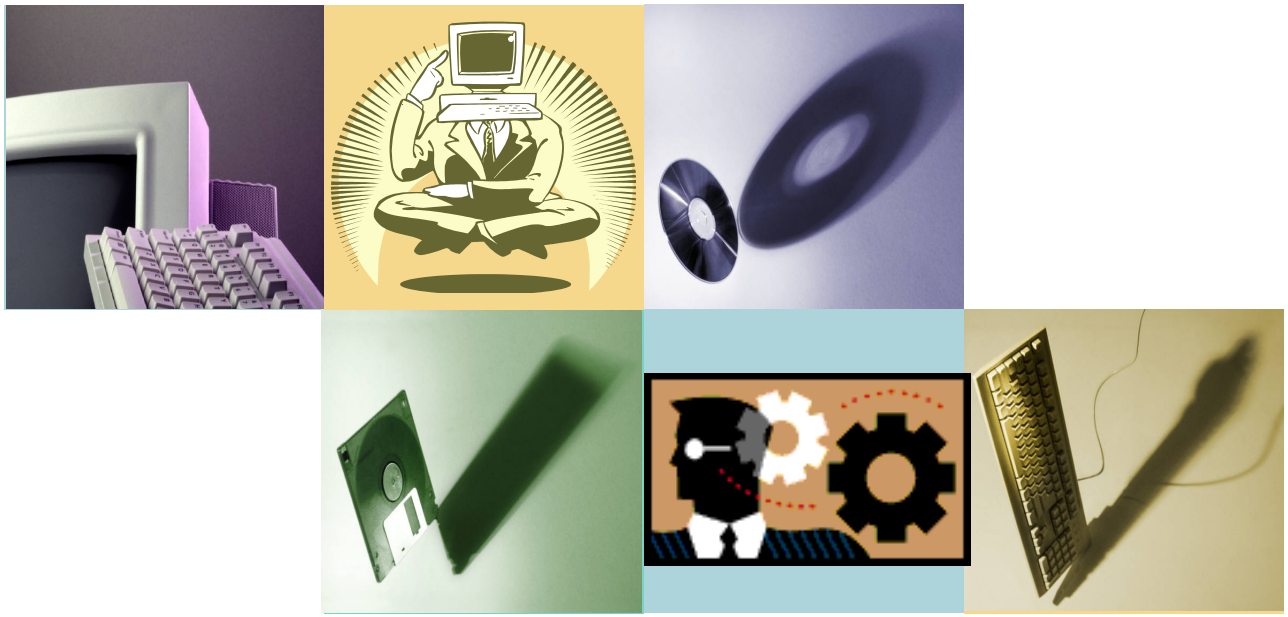
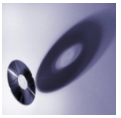
BMCS2003 Artificial Intelligence



Introduction to Expert System

Aims

- Introduction of expert system
- Expert system architecture (production system model)
- Expert system vs. conventional system
- Designing and developing an expert system


What is knowledge?

Knowledge

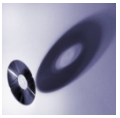
Subject/domain

theories

practices

understanding

Expert ?



Who is Expert?

Expertise
Knowledge

Subject/domain

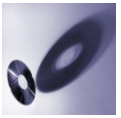
Theories

Experience

STRONG
practices

DEEP understanding

expert



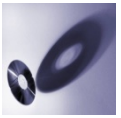
Complex mental processing and problem solving

Knowledge in RULES

Rules as Knowledge Representation

IF the 'traffic light' is green
THEN the action is go

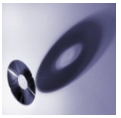
IF the 'traffic light' is red
THEN the action is stop


Rules as Knowledge Representation

- Any rule consists of two parts:
  - **IF** part, called the antecedent (premise or condition)
  - **THEN** part called the consequent (conclusion or action).

IF <antecedent>
THEN <consequent>

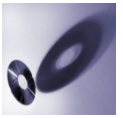




Using conjunction and disjunction

- A rule can have multiple antecedents joined by the keywords AND (conjunction), OR (disjunction) or a combination of both.

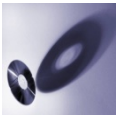
```
IF      <antecedent 1>
AND     <antecedent 2>
          • • •
AND       <antecedent n>
THEN      <consequent>
```

```
IF      <antecedent 1>
OR      <antecedent 2>
          • • •
OR      < antecedent n>
THEN    < consequent>
```

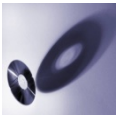

Production System Model

STRUCTURE OF A EXPERT SYSTEM

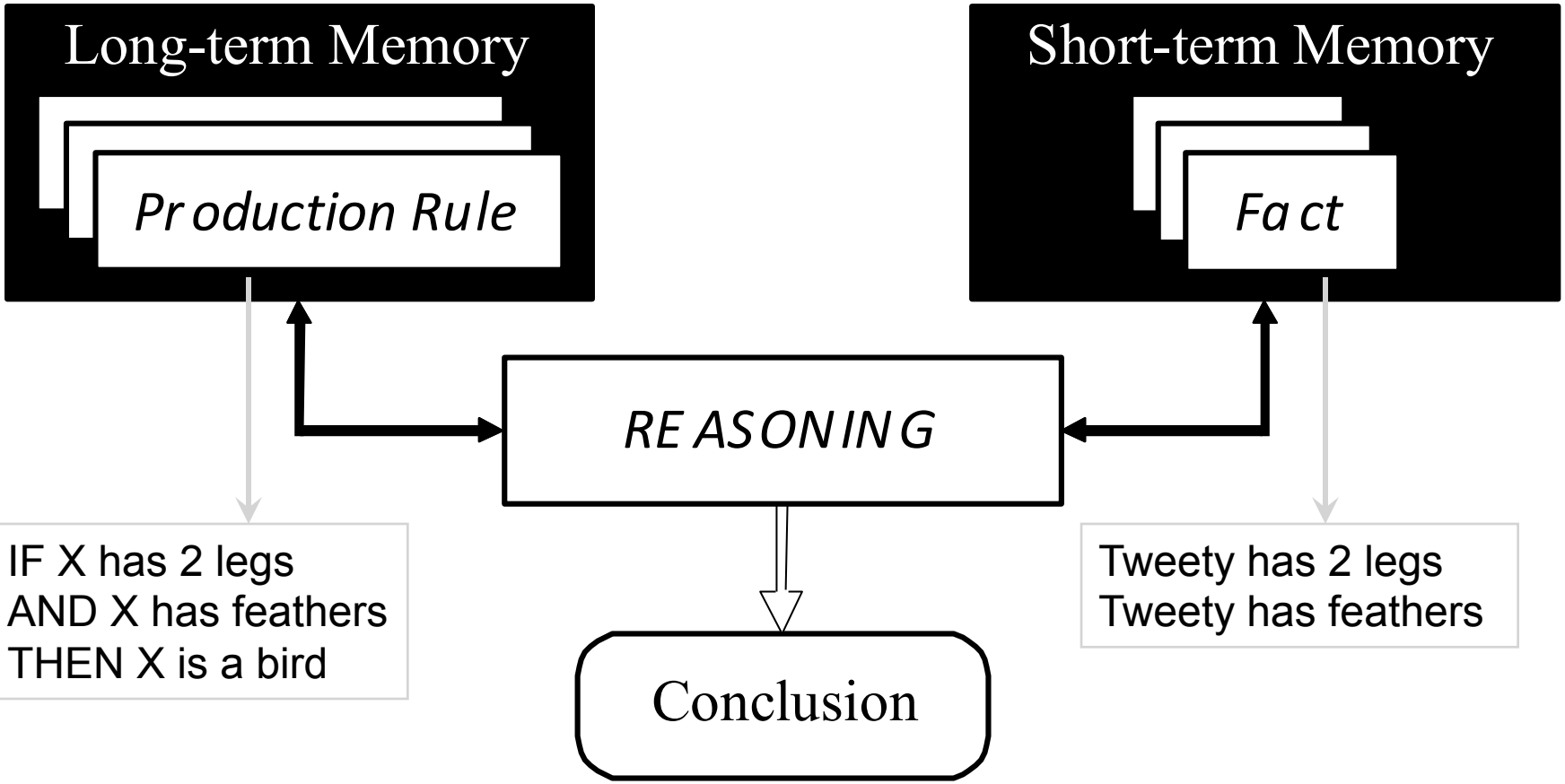



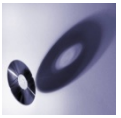

Production System Model

- It is the foundation of the modern rule-based expert systems.
- Proposed by Newell and Simon (1970s),Carnegie-Mellon University,
- Based on the idea that humans solve problems by applying their knowledge (production rules) to a given problem represented by problem-specific information

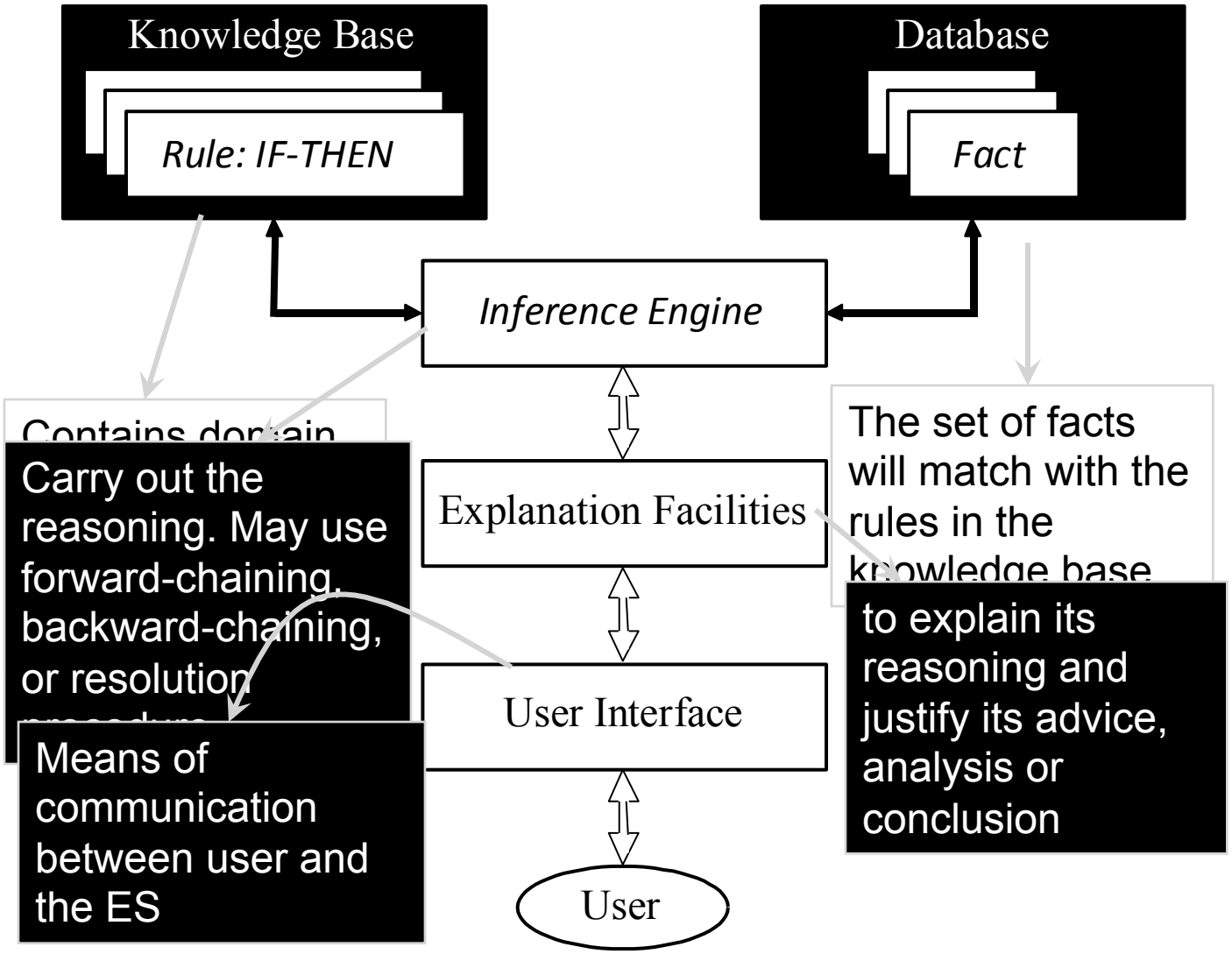




Production System Model

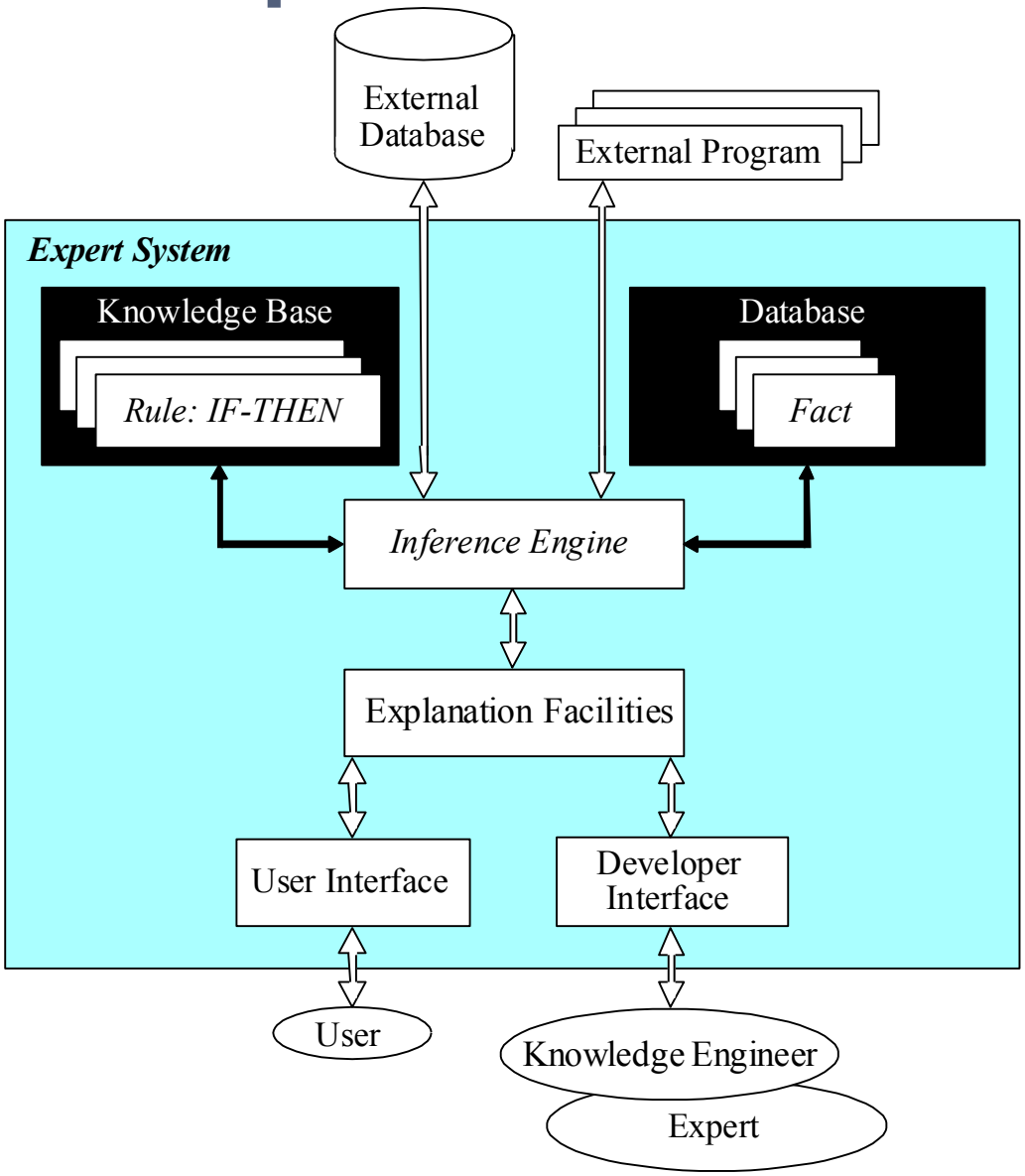



Basic Structure

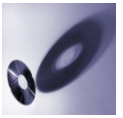


Complete Structure



Question

- What is the difference between knowledge-base and database?


Characteristics of an expert system

Narrow but specialized domain.

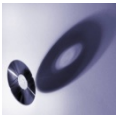
- e.g. Medical diagnosis

High-quality performance

- Correctness, accuracy, reliable

Speed

- - for instance, in an emergency, when a patient dies or a nuclear power plant explodes.


Characteristics of an expert system

Apply heuristics to guide reasoning and reduce search space

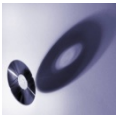
- Search performance 👍

Contains explanation capability

- Why (the input is needed)? How (the conclusion is yielded)?

employ symbolic reasoning when solving a problem.

- Natural language as input, e.g. “I am not sure”


ES vs Conventional Program

Expert System

- knowledge is separated from its processing (the knowledge base and the inference engine are split up).
- Easier to maintain, easy to understand

Conventional program

- mixture of knowledge and the control structure to process this knowledge.
- Difficult to understand and review the program code, as any change to the code affects both the knowledge and its processing.

Comparison of expert systems with conventional systems and human experts

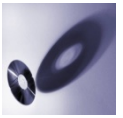
Human Experts	*Expert Systems*	*Conventional Programs*
Use knowledge in the form of rules of thumb or heuristics to solve problems in a narrow domain.	Process knowledge expressed in the form of rules and use symbolic reasoning to solve problems in a *narrow domain*.	Process data and use algorithms, a series of well-defined operations, to solve general numerical problems.
In a human brain, knowledge exists in a compiled form.	Provide a *clear separation of knowledge from its processing*.	Do not separate knowledge from the control structure to process this knowledge.
Capable of explaining a line of reasoning and providing the details.	*Trace the rules fired* during a problem-solving session and *explain how* a particular conclusion was reached and *why* specific data was needed.	Do not explain how a particular result was obtained and why input data was needed.

Comparison of expert systems with conventional systems and human experts (Continued)

Human Experts	*Expert Systems*	*Conventional Programs*
Use inexact reasoning and can deal with incomplete, uncertain and fuzzy information.	Permit *inexact reasoning* and can deal with incomplete, uncertain and fuzzy data.	Work only on problems where data is complete and exact.
Can make mistakes when information is incomplete or fuzzy.	*Can make mistakes* when data is incomplete or fuzzy.	Provide no solution at all, or a wrong one, when data is incomplete or fuzzy.
Enhance the quality of problem solving via years of learning and practical training. This process is slow, inefficient and expensive.	Enhance the quality of problem solving by adding new rules or adjusting old ones in the knowledge base. When new knowledge is acquired, *changes are easy* to accomplish.	Enhance the quality of problem solving by changing the program code, which affects both the knowledge and its processing, making changes difficult.

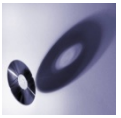
Question

- Discuss TWO (2) differences between an expert system of accounting and a conventional accounting software.

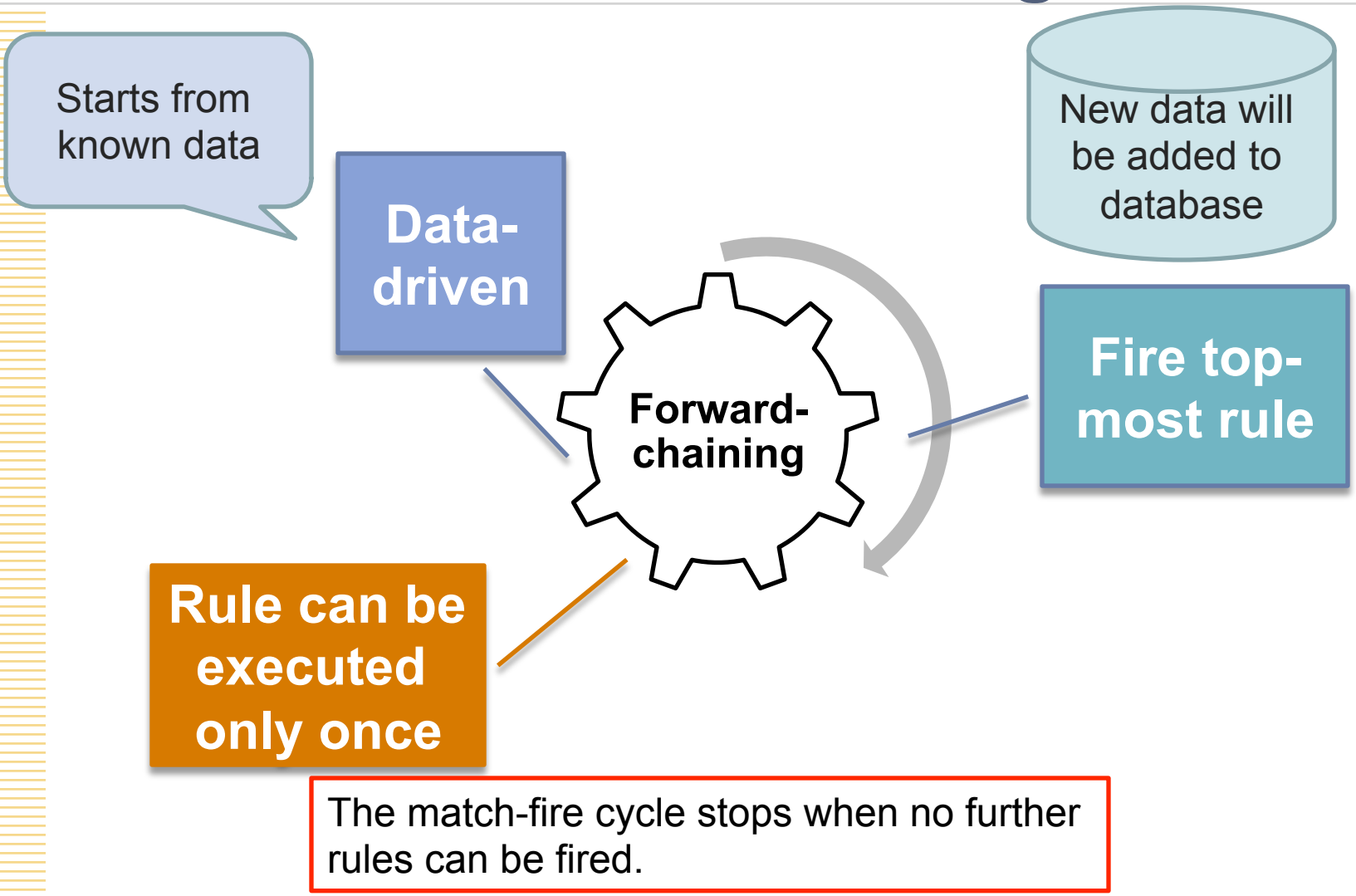




Inference Engine

- Forward-chaining
- Backward-chaining

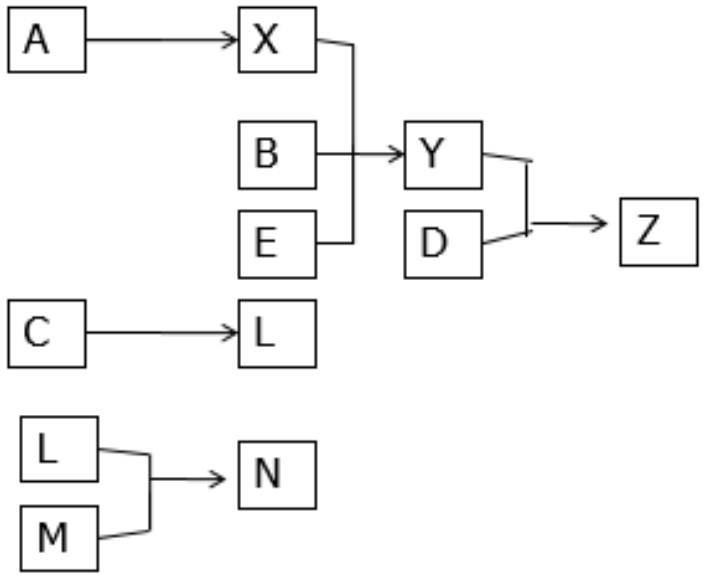




Forward chaining

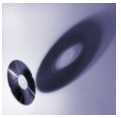


Forward chaining

Knowledge base	
Rule 1	IF Y is true AND D is true THEN Z is true
Rule 2	IF X is true AND B is true AND E is true THEN Y is true
Rule 3	IF A is true THEN X is true
Rule 4	IF C is true THEN L
Rule 5	IF L is true AND M is true THEN N
Working Memory/DB	
A, B, C, D, E	

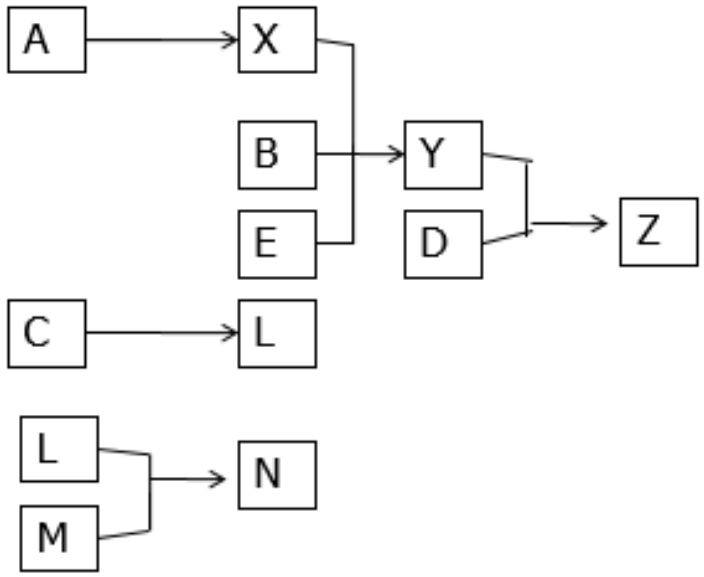


Cycle 1 : rule 3 and 4 match facts in kb

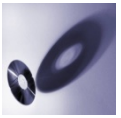




Forward chaining

Knowledge base	
Rule 1	IF Y is true AND D is true THEN Z is true
Rule 2	IF X is true AND B is true AND E is true THEN Y is true
Rule 3	IF A is true THEN X is true
Rule 4	IF C is true THEN L
Rule 5	IF L is true AND M is true THEN N
Working Memory/DB	
A, B, C, D, E, X	

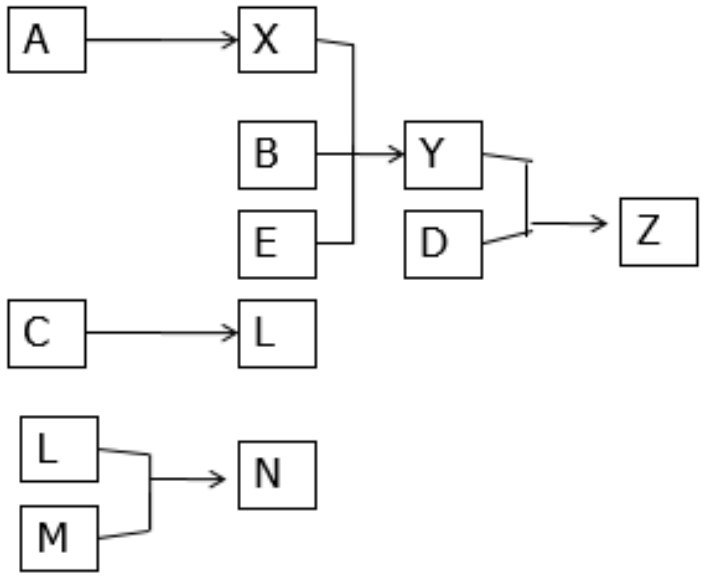


Cycle 1 : rule 3 and 4 match facts in kb
rule 3 fired first (topmost); new fact X added to db

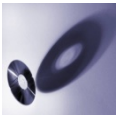




Forward chaining

Knowledge base	
Rule 1	IF Y is true AND D is true THEN Z is true
Rule 2	IF X is true AND B is true AND E is true THEN Y is true
Rule 3	IF A is true THEN X is true
Rule 4	IF C is true THEN L
Rule 5	IF L is true AND M is true THEN N
Working Memory/DB	
A, B, C, D, E, X, L	

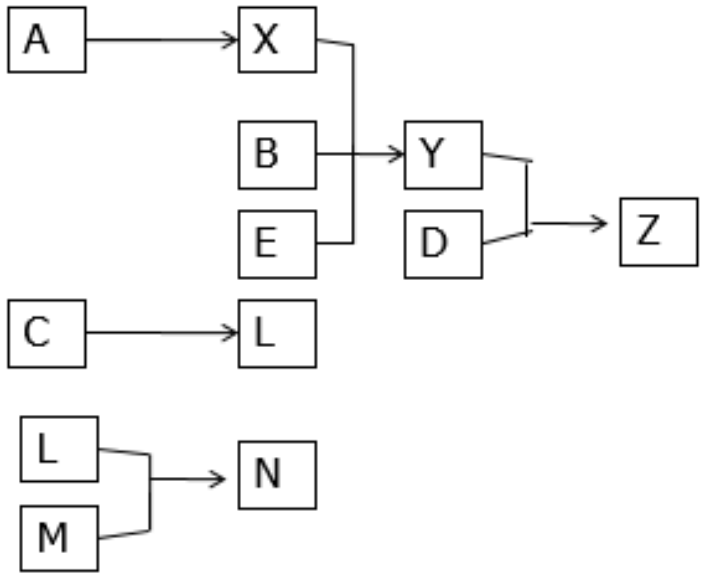


Cycle 1 : rule 3 and 4 match facts in kb
rule 3 fired first (topmost); new fact X added to db
rule 4 fired next; new fact L added to db

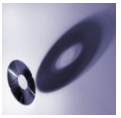




Forward chaining

Knowledge base	
Rule 1	IF Y is true AND D is true THEN Z is true
Rule 2	IF X is true AND B is true AND E is true THEN Y is true
Rule 3	IF A is true THEN X is true
Rule 4	IF C is true THEN L
Rule 5	IF L is true AND M is true THEN N
Working Memory/DB	
A, B, C, D, E, X, L, Y	

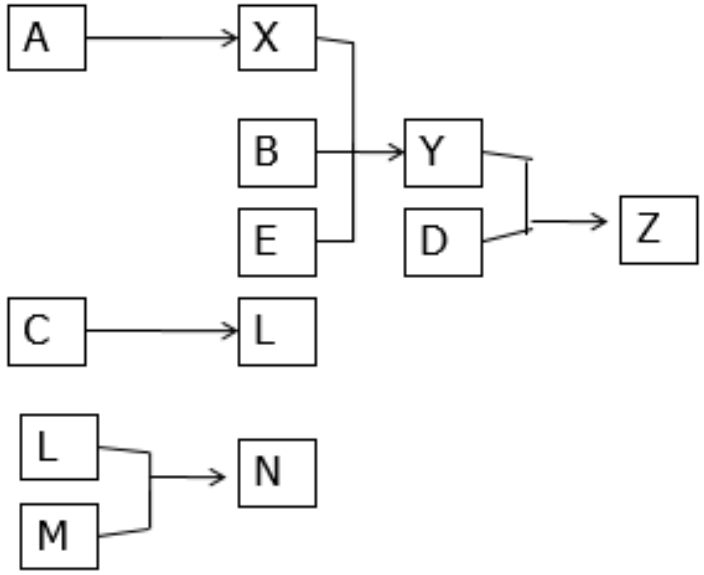


Cycle 1 : rule 3 and 4 match facts in kb
rule 3 fired first (topmost); new fact X added to db
rule 4 fired next; new fact L added to db
Cycle 2 : rule 2 is fired as B,E,X are in db; new fact Y added to db

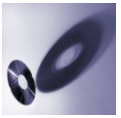

Forward chaining

Knowledge base	
Rule 1	IF Y is true AND D is true THEN Z is true
Rule 2	IF X is true AND B is true AND E is true THEN Y is true
Rule 3	IF A is true THEN X is true
Rule 4	IF C is true THEN L
Rule 5	IF L is true AND M is true THEN N
Working Memory/DB	
A, B, C, D, E, X, L, Y, Z	



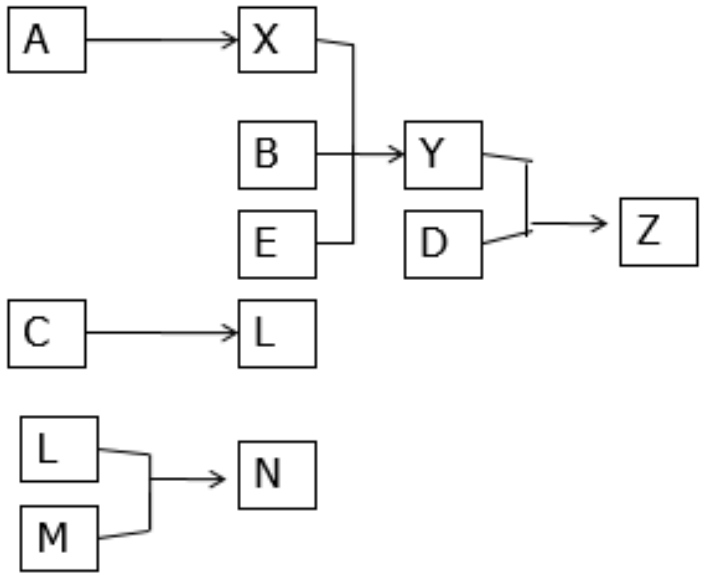
Cycle 1 : rule 3 and 4 match facts in kb
rule 3 fired first (topmost); new fact X added to db
rule 4 fired next; new fact L added to db
Cycle 2 : rule 2 is fired as B,E,X are in db; new fact Y added to db

Cycle 3 : rule 1 is fired; added Z in the db


Forward chaining

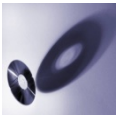
Knowledge base	
Rule 1	IF Y is true AND D is true THEN Z is true
Rule 2	IF X is true AND B is true AND E is true THEN Y is true
Rule 3	IF A is true THEN X is true
Rule 4	IF C is true THEN L
Rule 5	IF L is true AND M is true THEN N
Working Memory/DB	
A, B, C, D, E, X, L, Y, Z	



Cycle 1 : rule 3 and 4 match facts in kb
rule 3 fired first (topmost); new fact X added to db
rule 4 fired next; new fact L added to db
Cycle 2 : rule 2 is fired as B,E,X are in db; new fact Y added to db

Cycle 3 : rule 1 is fired; added Z in the db

Cycle 4 : stop because rule 5 does not match any facts


Forward Chaining

- Exp:

Rule 1

IF The patient has a sore throat
AND We suspect a bacterial infection
THEN We believe the patient has strep throat

Rule 2

IF The patient's temperature is > 100
THEN The patient has a fever

Rule 3

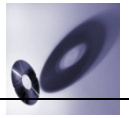
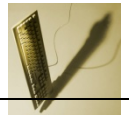
IF The patient has been sick over a month
AND The patient has a fever
THEN We suspect a bacterial infection

The following facts is assert into the working memory as supplied by the patient:

Patient's temperature = 102
Patient has been sick for two months
Patient has a sore throat

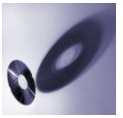
From the initial information entered into the WM, the system concludes three new pieces of information from the rules:

1. The patient has a fever
2. We suspect a bacterial infection
3. We believe the patient has strep throat

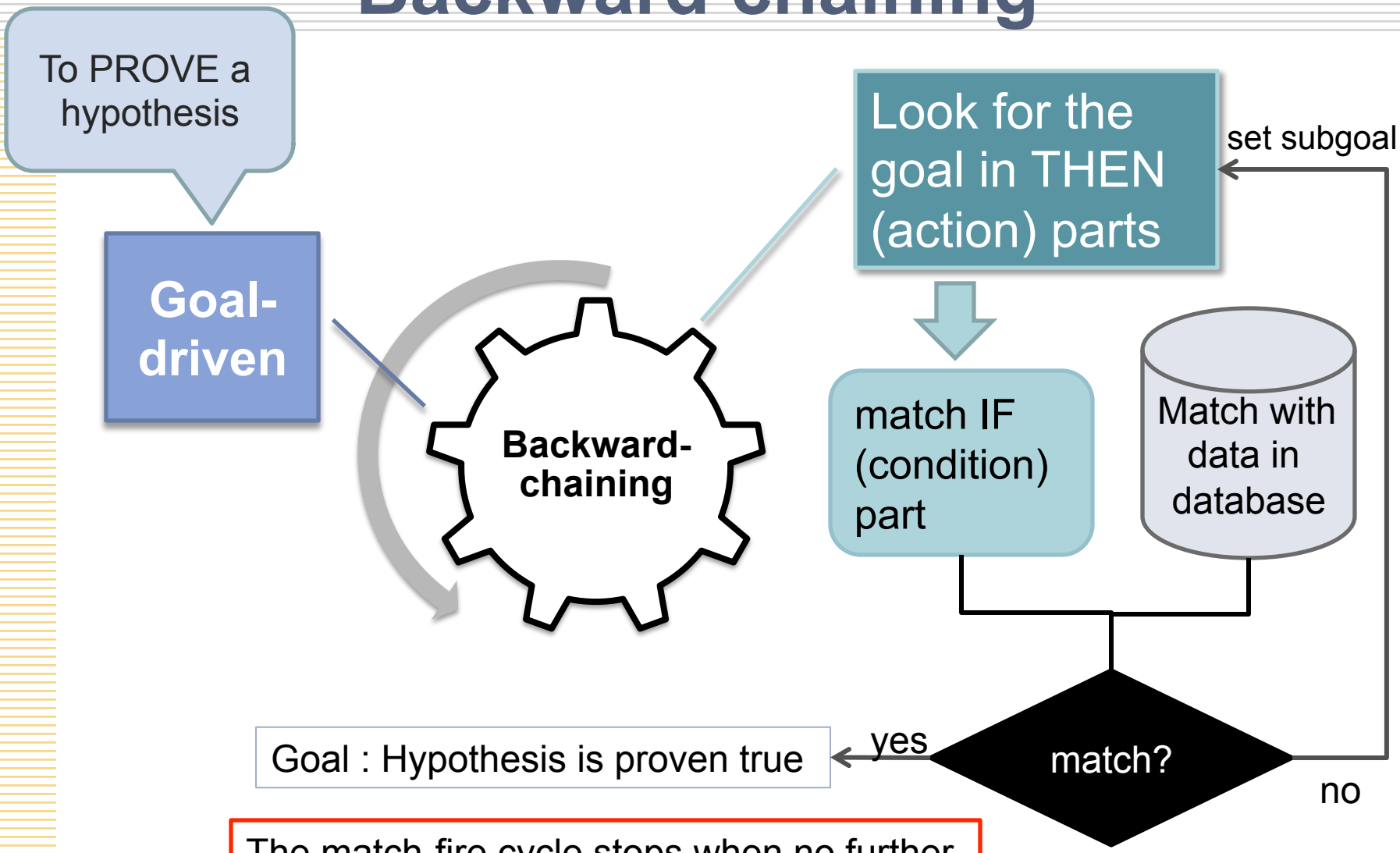




Forward Chaining - limitations

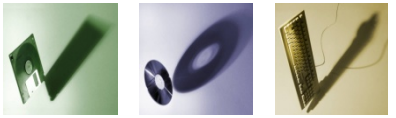
- We are required to gather all information which seems may be necessary
- However, many rules executed may have nothing to do with the established goal.


Backward chaining

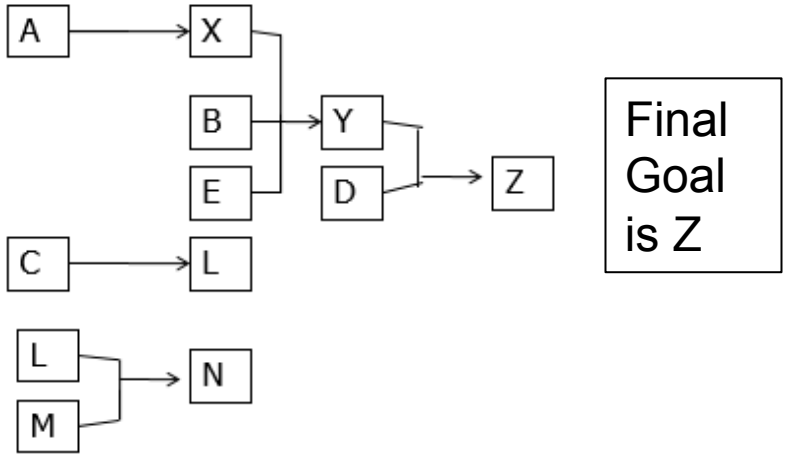


The match-fire cycle stops when no further rules can be fired to prove the (sub)goal(s).

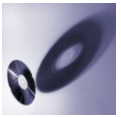


Backward chaining

Knowledge base	
Rule 1	IF Y is true AND D is true THEN Z is true
Rule 2	IF X is true AND B is true AND E is true THEN Y is true
Rule 3	IF A is true THEN X is true
Rule 4	IF C is true THEN L
Rule 5	IF L is true AND M is true THEN N
Working Memory/DB	
A, B, C, D, E	



Pass 1 : rule 1 stacked to confirm fact Y and D
Pass 2 : sets up subgoal Y, checks db, Y not there.
kb is searched for rule with Y in the THEN part
found rule 2, , stack rule 2.
Pass 3 : sets up subgoal X, checks db, X not there
kb searched for rule X in the THEN part,
found rule 3, stack rule 3
pass 4 : sets up subgoal A, checks db, A found,
rule 3 fired, new fact X inferred and add to db
Pass 5 : IE returns to subgoal Y, tries to execute rule 2,
X,B,E are in db, rule 2 fired, new fact Y inferred, add to db
Pass 6 : returns to goal Z, tries to execute rule 1, Y and D are in db,
rule 1 fired, thus goal Z is established


How do we choose?

Forward chaining

- needs to gather facts
- then only can try to infer from it
- e.g. identify a strange creature

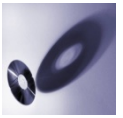
Backward chaining

- begins with a hypothetical solution
- then attempts to find facts to prove it
- e.g. Is it going to rain today?



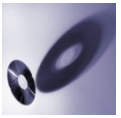
Expert System Shell

- When an expert system shell is used, a knowledge engineer or an expert simply enters rules in the knowledge base.
- Each new rule adds some new knowledge and makes the expert system smarter.
- E.g. Prolog, Drools, CLIPS, JESS, e2gLite, etc.

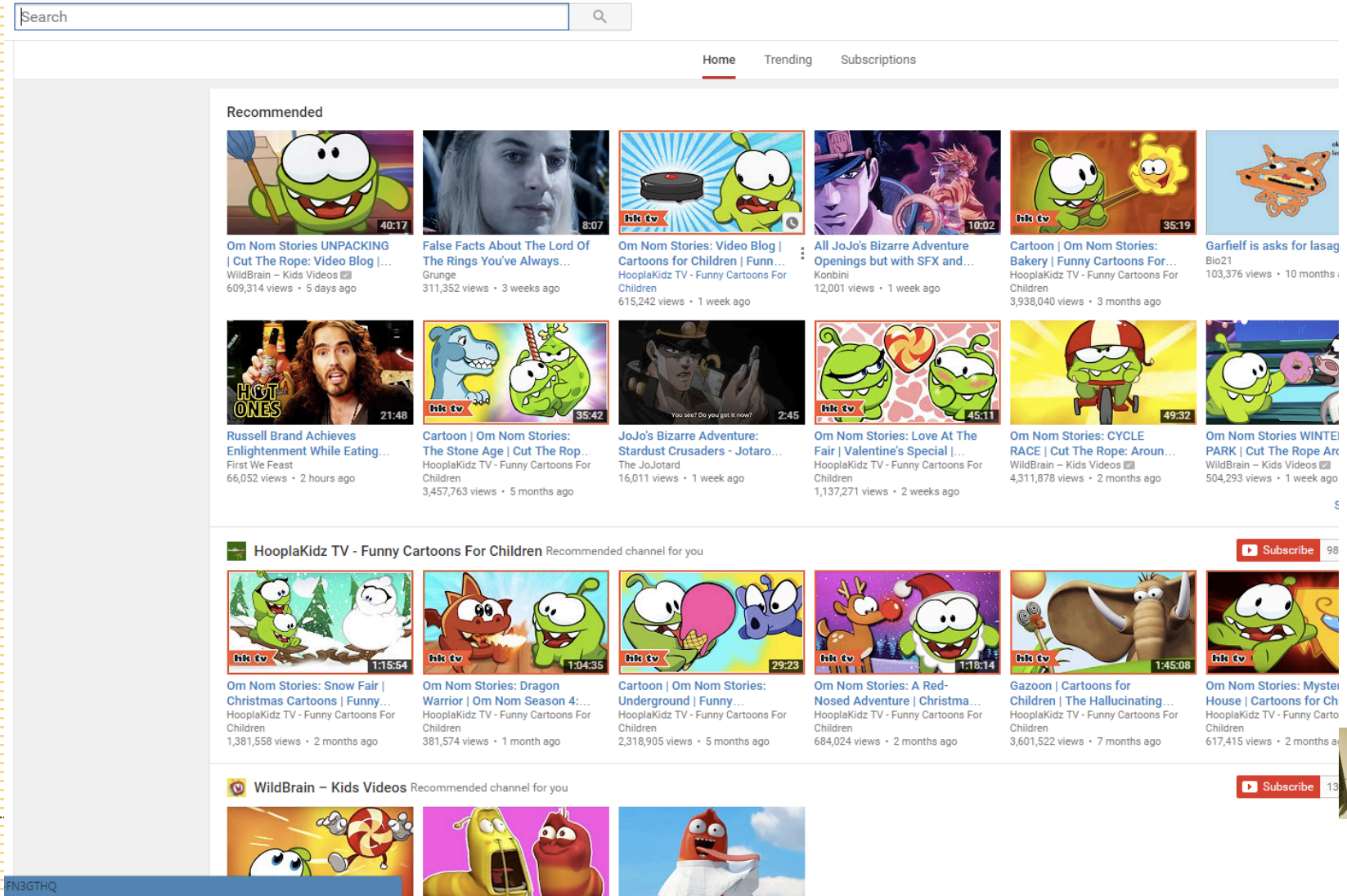

Next

RECOMMENDER SYSTEM/ ENGINE

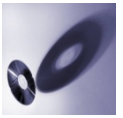

Recommender System

to provide the most relevant information to a user by discovering patterns in a dataset



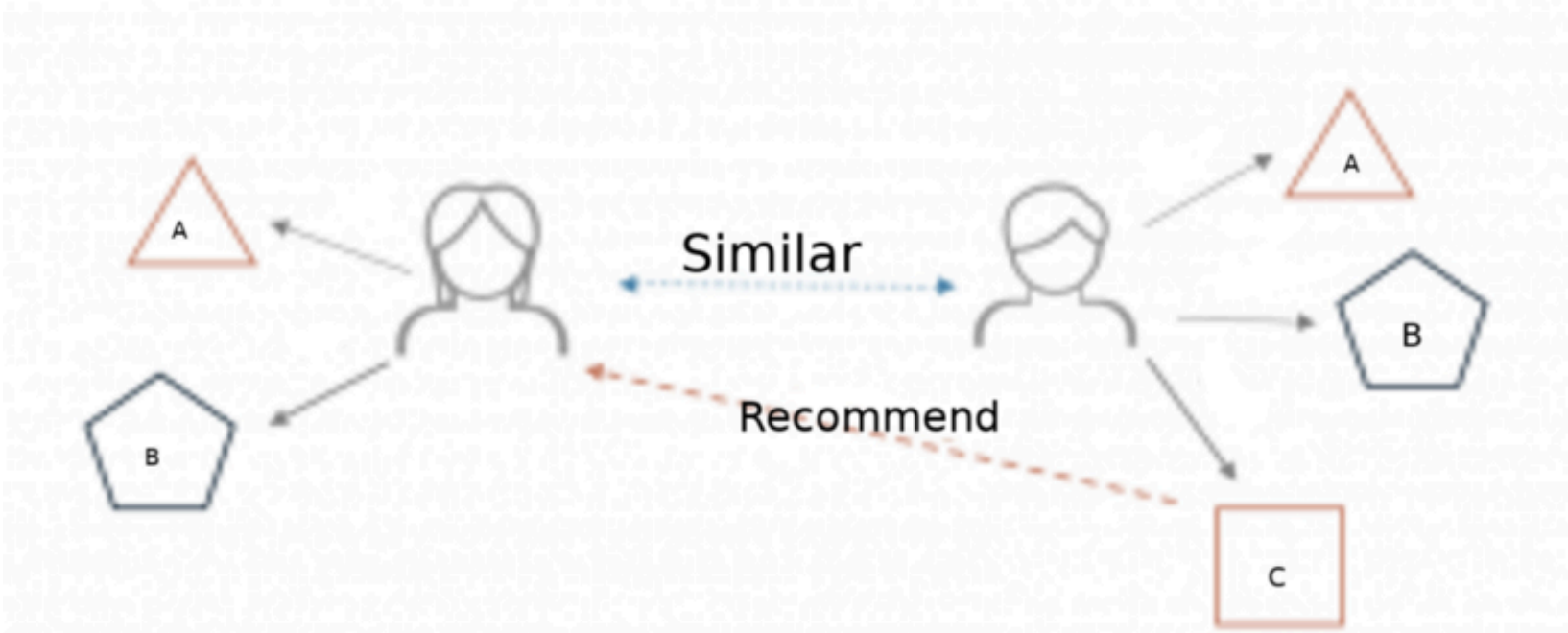
Types of Recommender Systems

- content based recommender systems
- and collaborative filtering recommender systems


Collaborative Filtering System

In this model products are recommended to a user based on the fact that the products have been liked by users similar to the user.

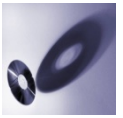


Depending on the score the system can recommend item C to the other user because it detects that those two users are similar in terms of the items they purchase.



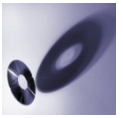
Content based Recommender System

- use meta data such as genre, producer, actor, musician to recommend items say movies or music.
- Such a recommendation would be for instance recommedation log that featured Vin Disiel because someone watched and liked The Fate of the Furious.


Item-based Recommender System

- Item-based collaborative filtering: These systems identify similar items based on users' previous ratings.
- For example if users A,B and C gave a 5 star rating to books X and Y then when a user D buys book Y they also get a recommendation to purchase book X because the system identifies book X and Y as similar based on the ratings of users A,B and C.


Next

UNCERTAINTY MANAGEMENT