



# 机器推理

# A historical note

- ❖ Logic was dominant paradigm in AI before 1990s

- ❖ **Strength:** provides expressiveness in a compact way

- ❖ **Problems:**

  - ❖ deterministic, didn't handle uncertainty

    - probability addresses this

  - ❖ rule-based, didn't allow fine tuning from data

    - machine learning addresses this

# Two goals of a logic language

- **Represent** knowledge about the world



- **Reason** with that knowledge





# ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to [InstructGPT](#), which is trained to follow an instruction in a prompt and provide a detailed response.

November 30, 2022  
13 minute read

## Samples

In the following sample, ChatGPT asks clarifying questions to debug code. (1/4)

USER this code is not working like i expect — how do i fix it?

```
resultWorkerErr := make(chan error)
defer close(resultWorkerErr)
go func() {
    defer cancel()
    resultWorkerErr <- b.resultWorker(ctx)
}()

err := b.worker(ctx)
cancel()
if err == nil {
    return <-resultWorkerErr
}
return multierror.Append(err, <-resultWorkerErr)
```

CHATGPT It's difficult to say what's wrong with the code without more context. Can you provide more information about what the code is supposed to do and what isn't working as expected? Also, is this the entire code or just a part of it?

USER just part of it — the error never surfaces. I think there is something wrong with the channel

CHATGPT It's difficult to say without more information about what the code is supposed to do and what's happening when it's executed. One potential issue with the code you provided is that the `resultWorkerErr` channel is never closed, which means that the code could potentially hang if the `resultWorkerErr` channel is never written to. This could happen if `b.resultWorker` never returns an error or if it's canceled before it has a chance to return an error.

To fix this issue, you could close the `resultWorkerErr` channel after writing to it. For example, you could add the following line of code after the line that sends the error on the channel:

# Can large language models do reasoning?

- Human performance on these benchmarks is closely tied to common sense and the capacity for logical reasoning.
- But it is not clear whether large language models are improving because they have acquired logical reasoning capabilities or simply because they have been exposed to very large amounts of data.

## Statistical features



## Logical reasoning

- Facts:
  - Alice is fast.
  - Alice is normal.
- Rules:
  - If Alice is fast and smart, then Alice is bad.
  - If Alice is normal, then Alice is smart.
  - If Alice is normal and unhappy, then Alice is sad.
- Query 1: Alice is bad. [Answer: True]
- Query 2: Alice is sad. [Answer: False]

# Reasoning = Knowledge + Inference

世界（知识）模型 + （语言）推理机器

**Yoshua Bengio**

Home Profile Research Publications Students Media Presentations News English ▼

The diagram illustrates the interaction between an Inference Machine, a World Model, and Data. The Inference Machine (top left) sends 'internal queries & samples from Bayesian posterior over world models & latents' to the World Model (right). The World Model provides a 'reward signal' back to the Inference Machine. Data (bottom left) is used by the World Model to 'Represent an explanation for' and 'Bounds description length' of the Inference Machine's output.

**Scaling in the service of reasoning & model-based ML**

Published 21 March 2023 by yoshuabengio

Recognized worldwide as one of the leading experts in artificial intelligence, Yoshua Bengio is most known for his pioneering work in deep learning, earning him the 2018 A.M. Turing Award, "the Nobel Prize of Computing," with Geoffrey Hinton and Yann LeCun.

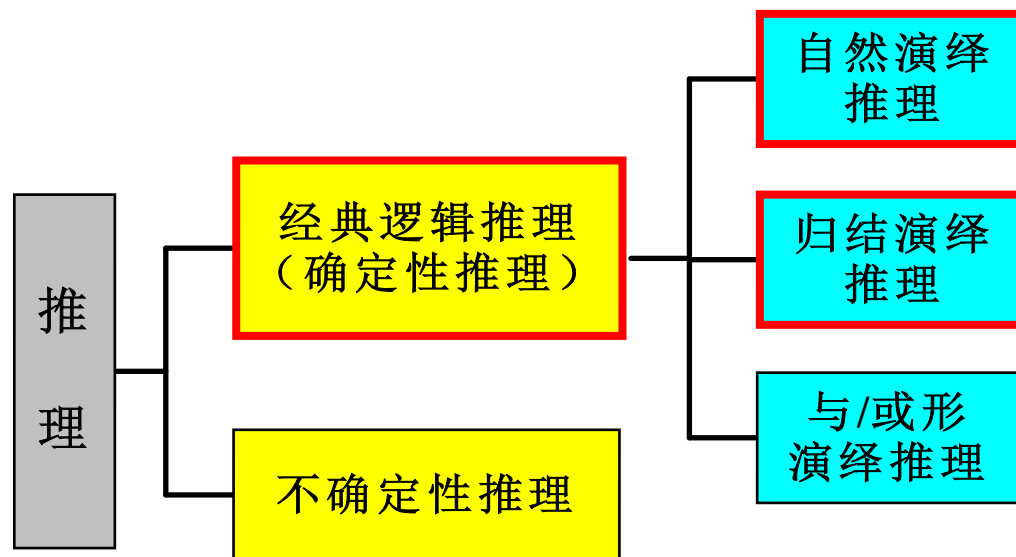
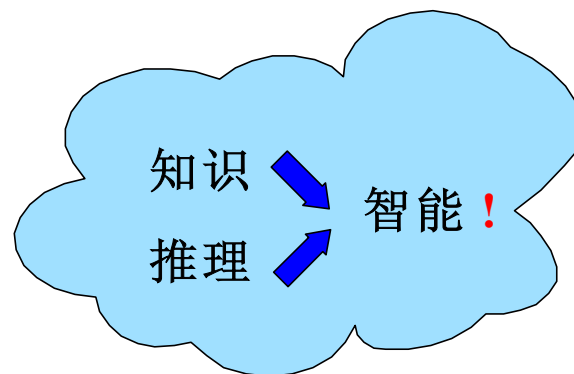
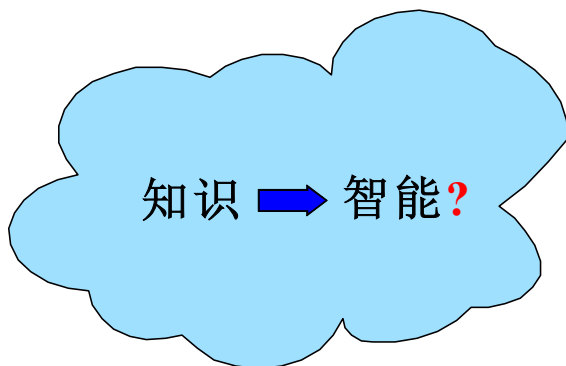
Current language model:

World Model + Inference Machine

?

In the future:

World Model + Inference Machine



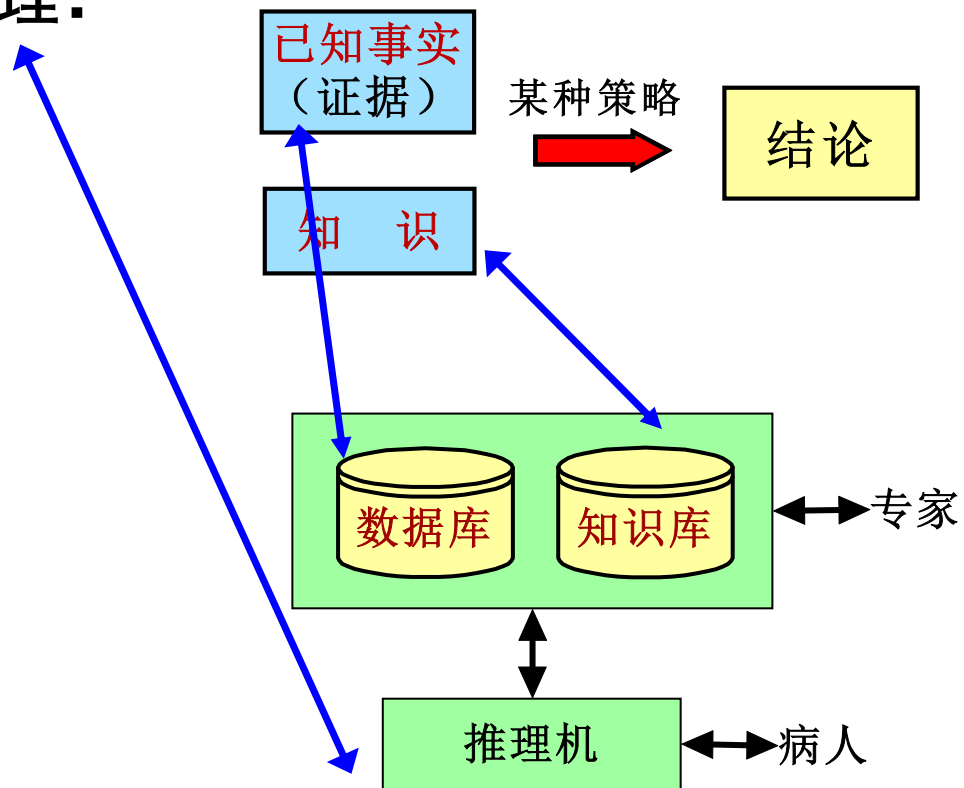
# 推理的基本概念

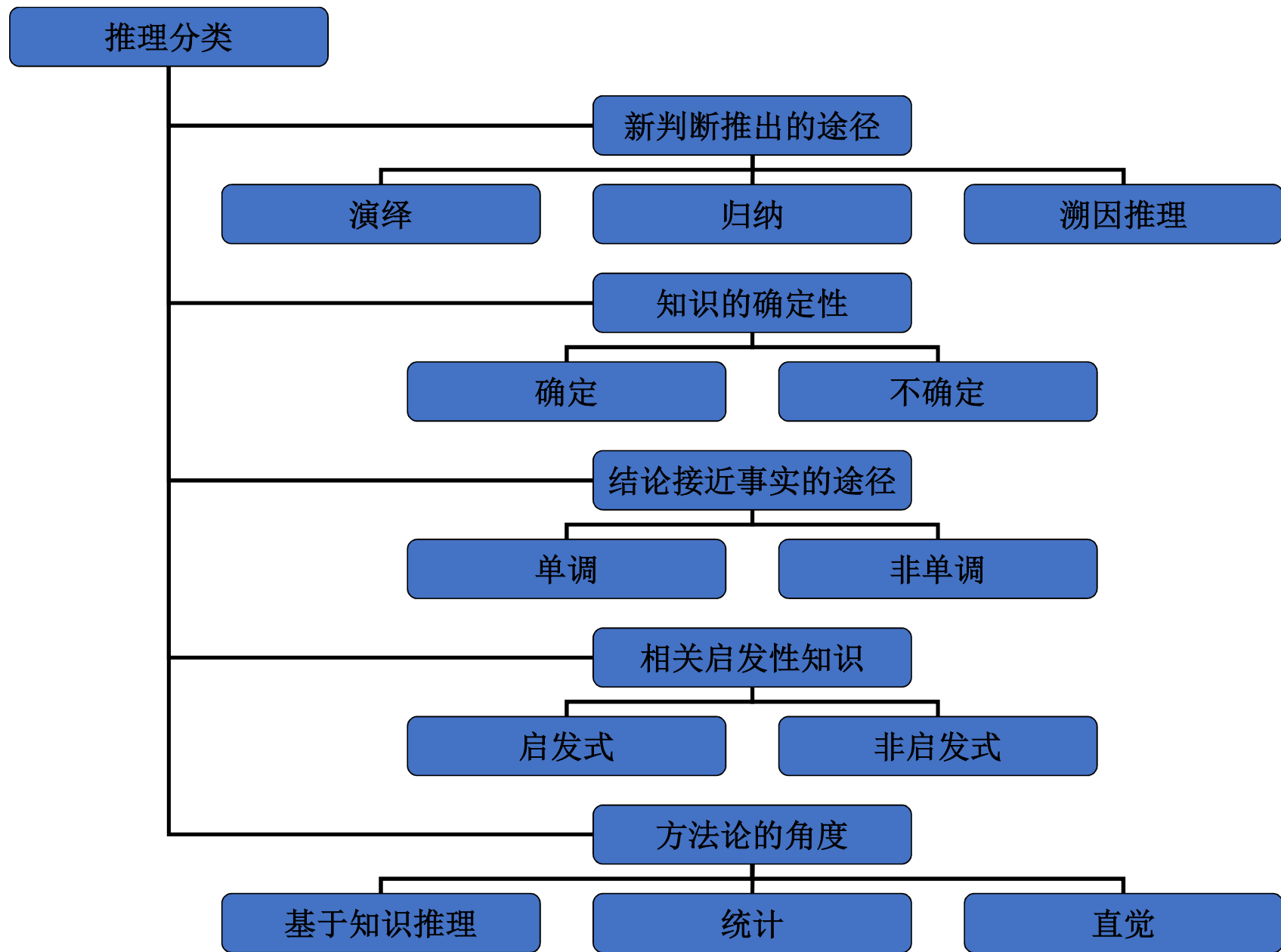
- 推理的定义
- 推理方式及其分类
- 推理的方向
- 冲突消解策略



# 一、推理的定义

- 推理：





# 推理方式及其分类

## 1. 演绎推理、归纳推理、溯因推理

(1) 演绎推理 (deductive reasoning) : 一般  $\rightarrow$  个别

### ■ 三段论式 (三段论法)

① 足球运动员的身体都是强壮的 ; ( 大前提 )

② Tom是一名足球运动员 ; ( 小前提 )

---

③ 所以, Tom的身体是强壮的。 ( 结 论 )

## (2) 归纳推理 (inductive reasoning): 个别 $\rightarrow$ 一般

完全归纳推理 (必然性推理)

不完全归纳推理 (非必然性推理)

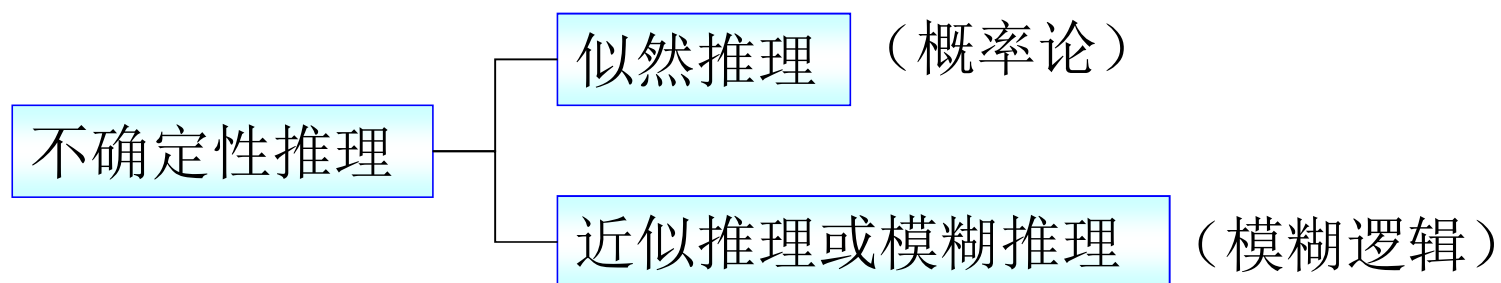
## (3) 溯因推理 (abductive reasoning)

- 溯因法或溯因推理：推理到最佳解释
  - 开始于事实的集合，并推导出其最佳解释的推理过程。
- 应用：
  - e.g. 用于自动检测系统中的故障：给出与有关故障和表现的理论和一组故障（故障的可见效果），可以使用溯因法来推导故障的某个集合好像是问题的原因。
  - e.g.
    - 观察：车子开不了，引擎下面有一滩液体
    - 结论：最可能的解释是车子的散热器漏水了

## 2. 确定性推理、不确定性推理

(1) **确定性推理**：推理时所用的知识与证据都是确定的，推出的结论也是确定的，其真值或者为真或者为假。

(2) **不确定性推理**：推理时所用的知识与证据不都是确定的，推出的结论也是不确定的。



### 3. 单调推理、非单调推理

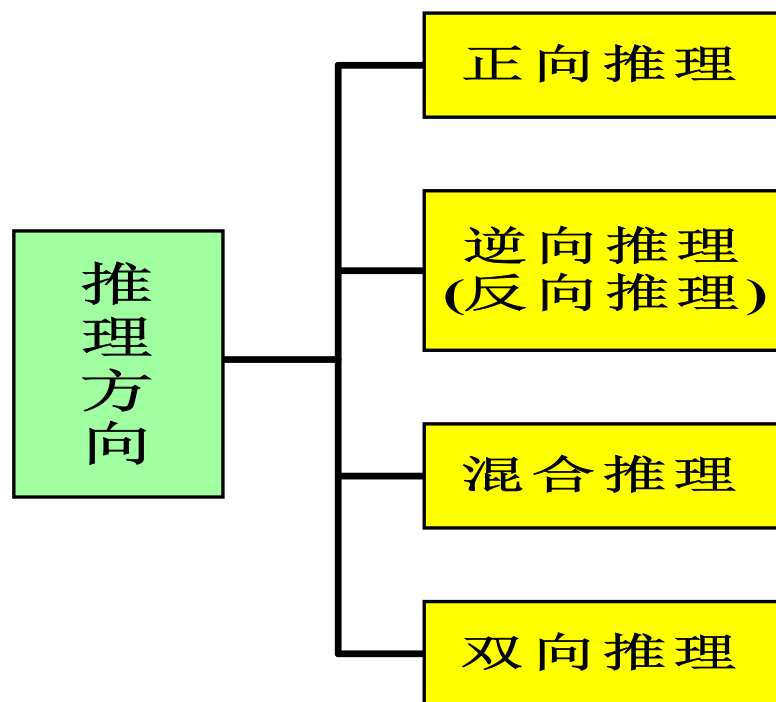
(1) **单调推理**：随着推理向前推进及新知识的加入，推出的结论越来越接近最终目标。

(2) **非单调推理**：由于新知识的加入，不仅没有加强已推出的结论，反而要否定它，使推理退回到前面的某一步，重新开始。

# 推理的控制策略

- 推理的控制策略主要包括：
  - 推理方向
  - 搜索策略
  - 冲突消解策略
  - 求解策略
  - 限制策略等

### 三、推理的方向





# 1. 正向推理

- 正向推理（事实驱动推理）：已知事实  $\rightarrow$  结论

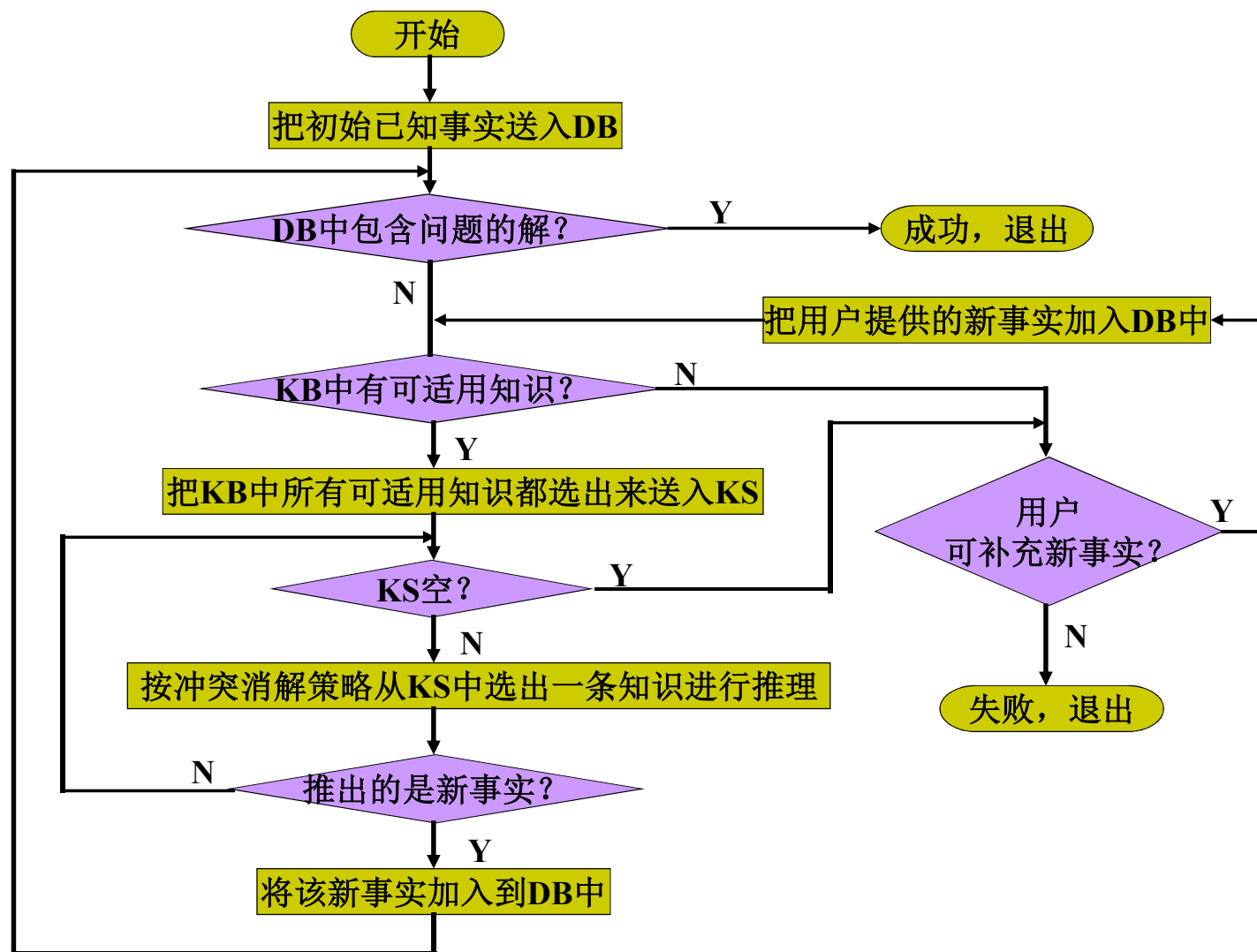
- 基本思想

- (1) 从初始已知事实出发，在知识库 $KB$ 中找出当前可适用的知识，构成可适用知识集 $KS$ 。
- (2) 按某种冲突消解策略从 $KS$ 中选出一条知识进行推理，并将推出的新事实加入到数据库 $DB$ 中作为下一步推理的已知事实，再在 $KB$ 中选取可适用知识构成 $KS$ 。
- (3) 重复 (2)，直到求得问题的解或 $KB$ 中再无可适用的知识。

- 实现正向推理需要解决的问题：

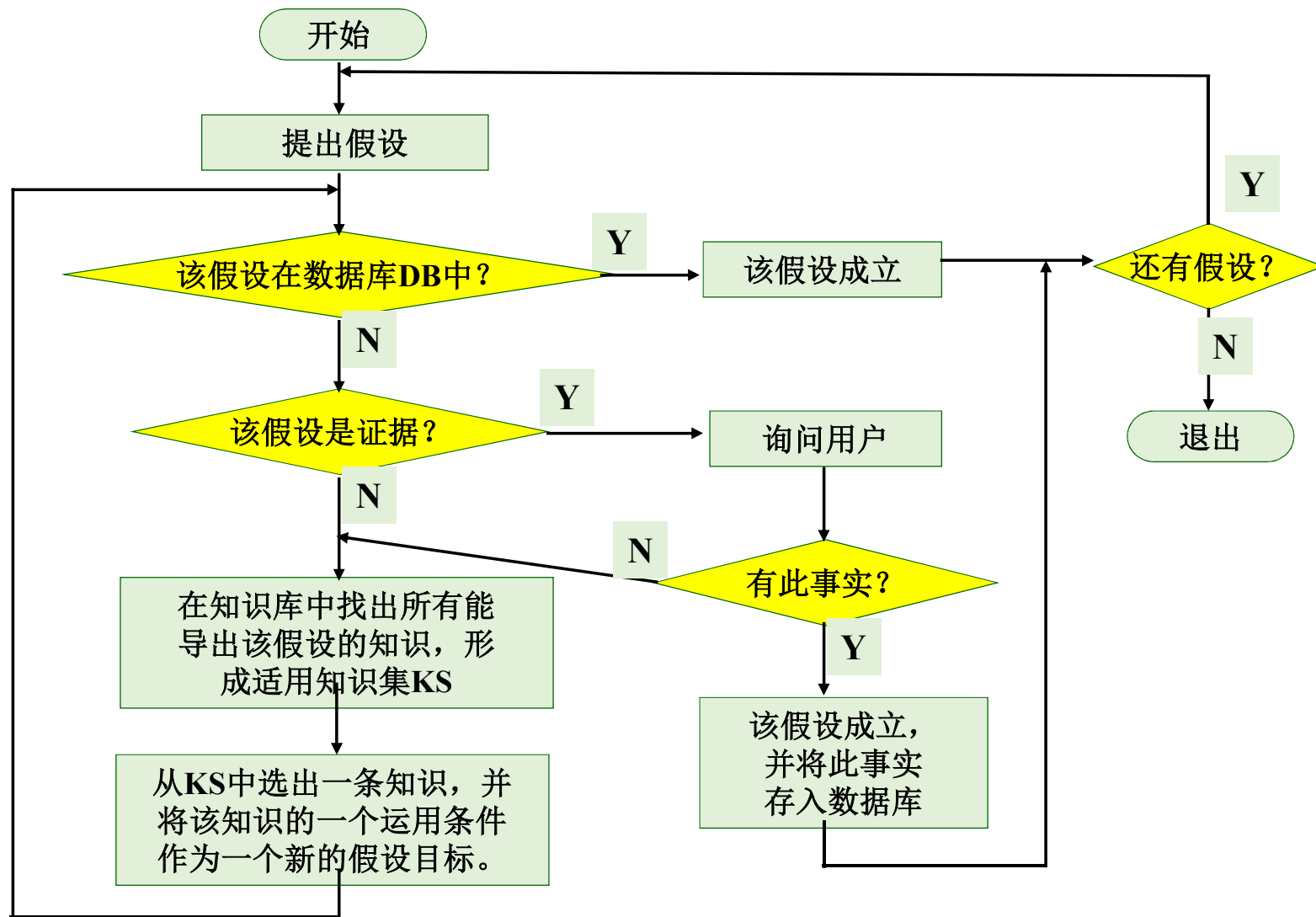
- 确定匹配（知识与已知事实）的方法
- 按什么策略搜索知识库
- 冲突消解策略

- 正向推理简单，易实现，但目的性不强，效率低。



## 2. 逆向推理

- **逆向推理（目标驱动推理）：**以某个假设目标作为出发点。
- **基本思想：**
  - 选定一个假设目标。
  - 寻找支持该假设的证据，若所需的证据都能找到，则原假设成立；若无论如何都找不到所需要的证据，说明原假设不成立的；为此需要另作新的假设。
- **主要优点：**不必使用与目标无关的知识，目的性强，同时它还有利于向用户提供解释。
- **主要缺点：**起始目标的选择有盲目性。



- 逆向推理需要解决的问题：

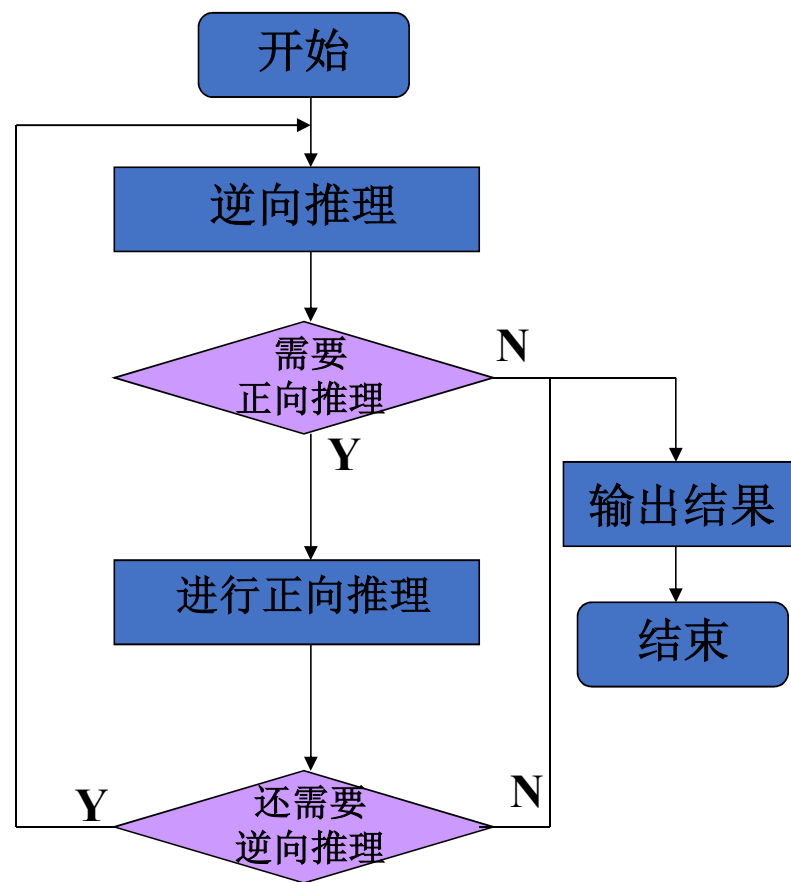
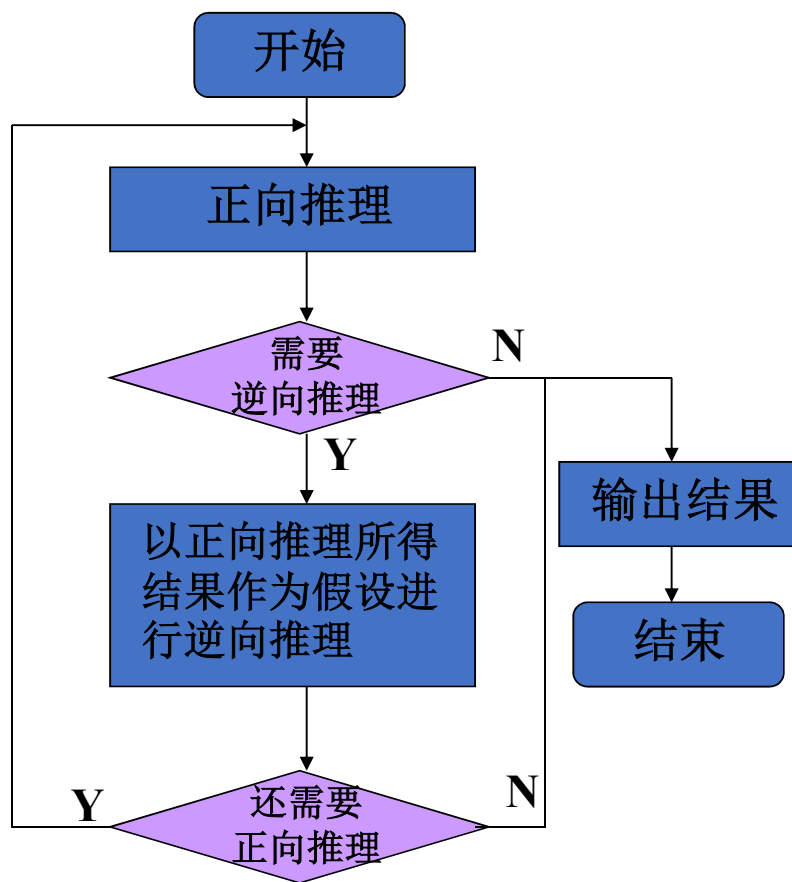
- ◆ 如何判断一个假设是否是证据？
- ◆ 当导出假设的知识有多条时，如何确定先选哪一条？
- ◆ 一条知识的运用条件一般都有多个，当其中的一个经验证成立后，如何自动地换为对另一个的验证？
- ◆ ..... .

❖ 逆向推理：目的性强，利于向用户提供解释，但选择初始目标时具有盲目性，比正向推理复杂。

### 3. 混合推理

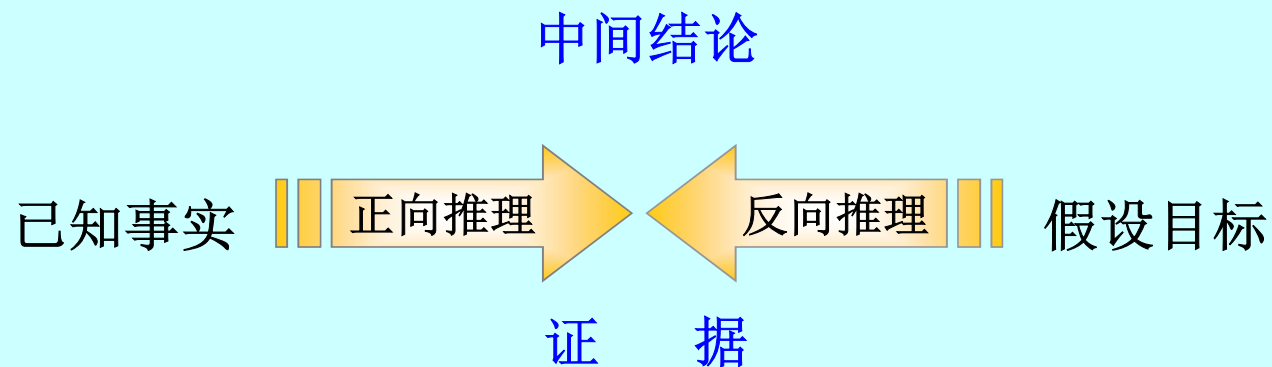
- **正向推理:** 盲目、效率低。
- **逆向推理:** 若提出的假设目标不符合实际, 会降低效率。
- **正反向混合推理:**
  - (1) **先正向后逆向:** 先进行正向推理, 帮助选择某个目标, 即从已知事实演绎出部分结果, 然后再用逆向推理证实该目标或提高其可信度;
  - (2) **先逆向后正向:** 先假设一个目标进行逆向推理, 然后再利用逆向推理中得到的信息进行正向推理, 以推出更多的结论。

# 两种混合推理的示意图



## 4. 双向推理

- **双向推理**：正向推理与逆向推理同时进行，且在推理过程中的某一步骤上“碰头”的一种推理。





## 四、冲突消解策略

■ 已知事实与知识的三种匹配情况：

(1) 不能匹配成功

(2) 恰好匹配成功（一对一）

(3) 多种匹配成功（一对多、多对一、多对多）



■ 多种冲突消解策略：

- (1) 按针对性排序
- (2) 按已知事实的新鲜性排序
- (3) 按匹配度排序
- (4) 按条件个数排序
- (5) 按上下文限制排序
- (6) 按冗余限制排序
- (7) 根据领域问题的特点排序

## 例：动物分类问题的产生式系统描述及其求解

设由下列动物识别规则组成一个规则库, 推理机采用上述正向推理算法, 建立一个产生式系统。该产生式系统就是一个小型动物分类知识库系统。

- 规则集:

r1: 若某动物有奶, 则它是哺乳动物。

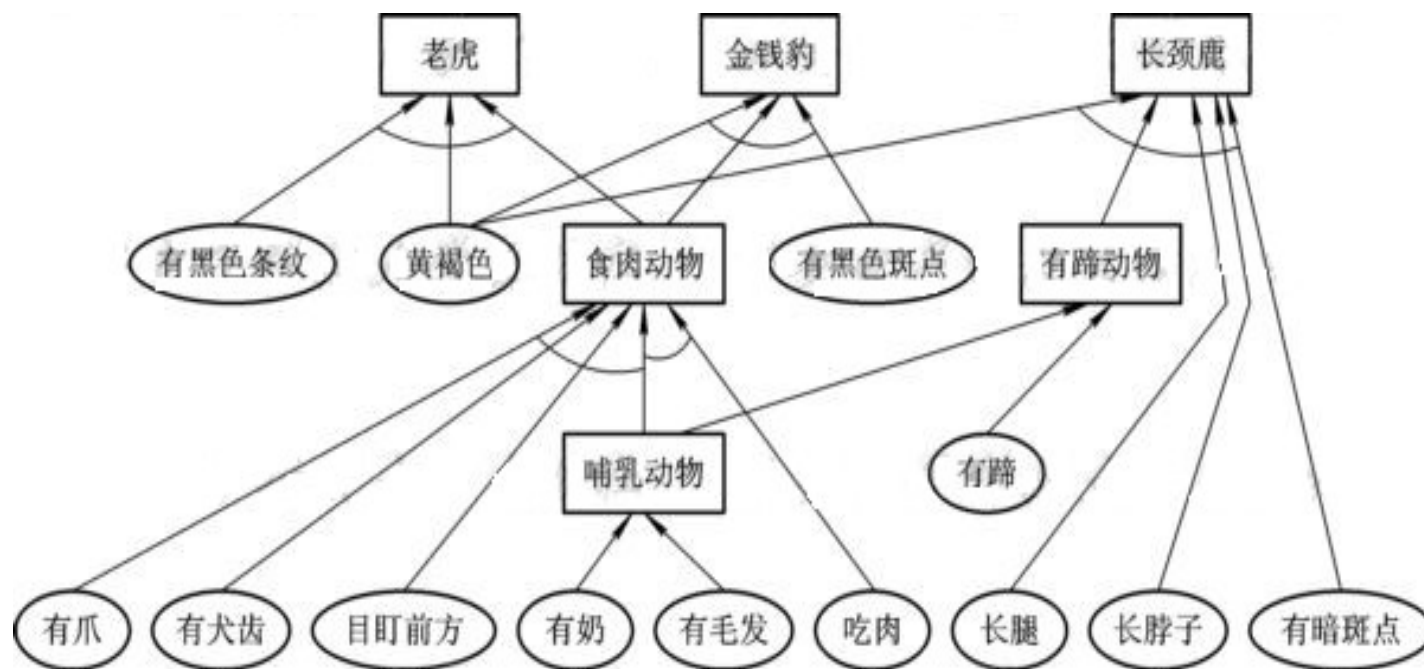
r2: 若某动物有毛发, 则它是哺乳动物。

r3: 若某动物有羽毛, 则它是鸟。

r4: 若某动物会飞且生蛋, 则它是鸟。

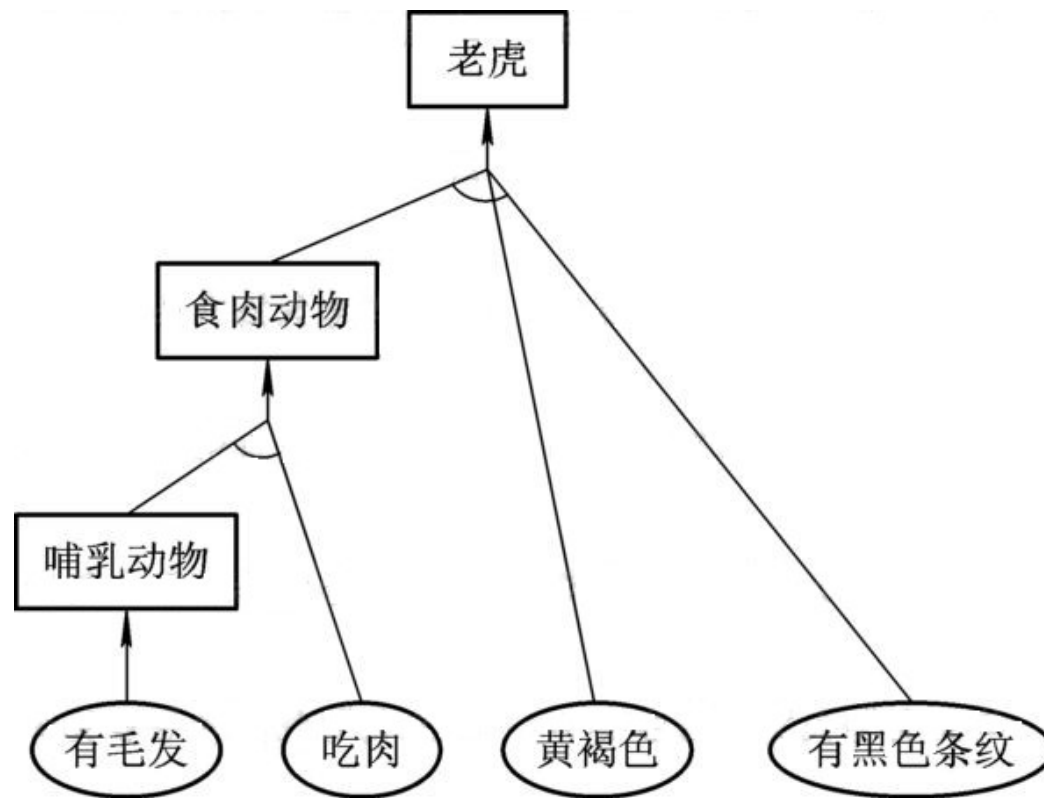
r5: 若某动物是哺乳动物且有爪且有犬齿且目盯前方, 则它是食肉动物。

- r6: 若某动物是哺乳动物且吃肉, 则它是食肉动物。
- r7: 若某动物是哺乳动物且有蹄, 则它是有蹄动物。
- r8: 若某动物是有蹄动物且反刍食物, 则它是偶蹄动物。
- r9: 若某动物是食肉动物且黄褐色且有黑色条纹, 则它是老虎。
- r10: 若某动物是食肉动物且黄褐色且有黑色斑点, 则它是金钱豹。
- r11: 若某动物是有蹄动物且长腿且长脖子且黄褐色且有暗斑点, 则它是长颈鹿。
- r12: 若某动物是有蹄动物且白色且有黑色条纹, 则它是斑马。
- r13: 若某动物是鸟且不会飞且长腿且长脖子且黑白色, 则它是鸵鸟。
- r14: 若某动物是鸟且不会飞且会游泳且黑白色, 则它是企鹅。
- r15: 若某动物是鸟且善飞且不怕风浪, 则它是海燕。



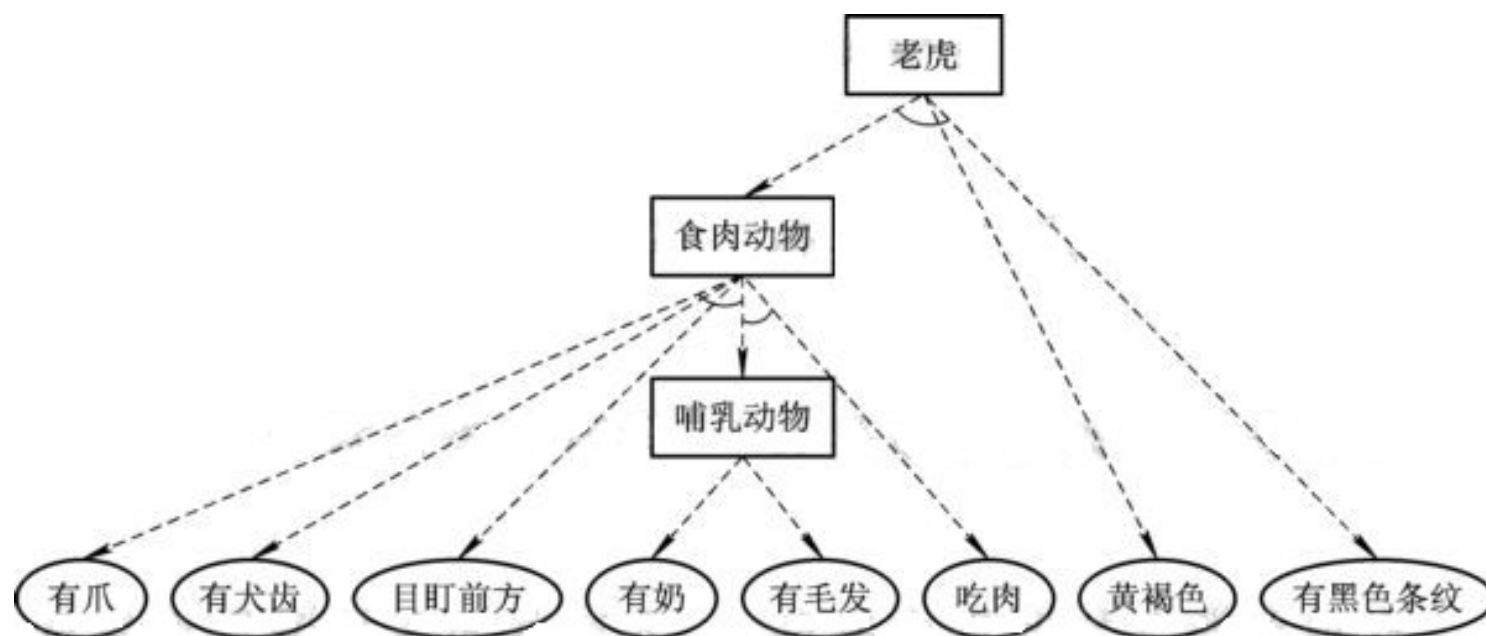
- 初始事实:
  - f1: 某动物有毛发。
  - f2: 吃肉。
  - f3: 黄褐色。
  - f4: 有黑色条纹。
- 目标: 该动物是什么?

规则集形成的部分推理网络



关于“老虎”的正向推理树

对于上例中的产生式系统, 改为反向推理算法, 则得到如图所示的推理树。



关于“老虎”的逆向推理树

Q & A