情報とコンピュータ

静岡大学工学部
安藤和敏

2004.11.15

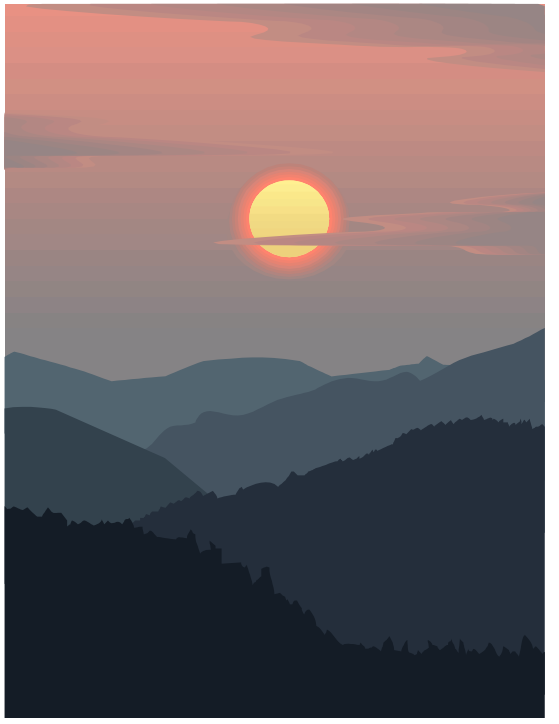
4章
トップダウンプログラミング，サブルーチン，データベースの応用

- 謎解きへの挑戦
- トップダウンプログラミングとデータベースプログラム
- サブルーチン

村の夕暮れ

ブラウン警部

一服する
だら.



シークレスト嬢からの電話

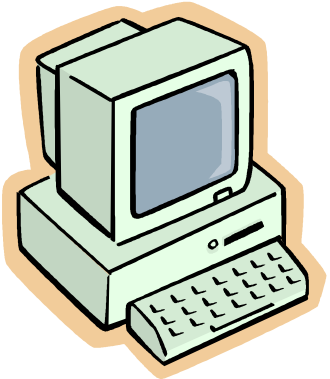
シークレスト嬢

お父ちゃんが，応接間で倒れてるの．死んでるみたいなの！

わかった．すぐ行く．

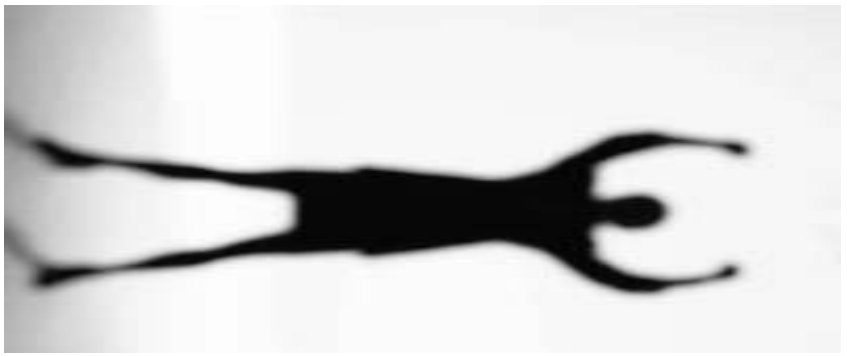
ブラウン警部

ブラウン警部が到着



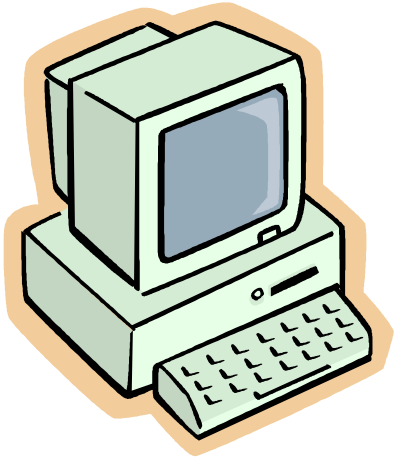
死因は，数時間前に毒薬を投与されたためです．

ブラウン警部



ピーターダンスモア卿の死体

コンピュータ部屋の片隅のコンピュータ



家族や使用人は取り乱していてとても尋問できる状態ではないな．．．

このコンピュータには，家族のデータが入ってるみたいだ．

警部はコンピュータの前に座った

警部

今日，お屋敷を訪ねたのは誰か？

コンピュータの答え

メイソン氏は午後3時に来た．
教授は午後3時に来た．
シークレスト嬢は午後5時に来た．

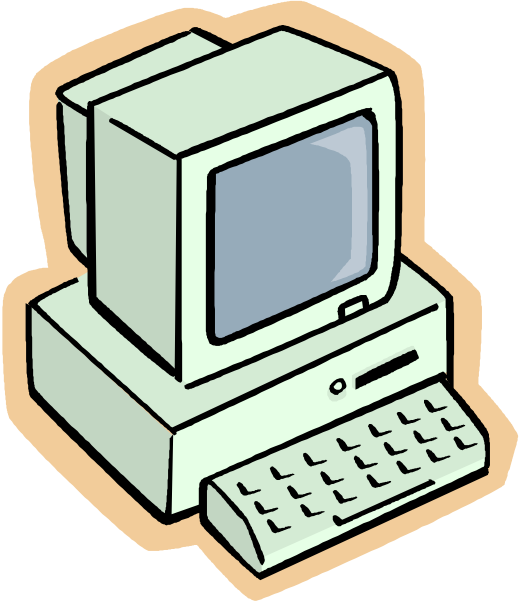
警部

メイソン氏について教えて欲しい．

コンピュータの答え

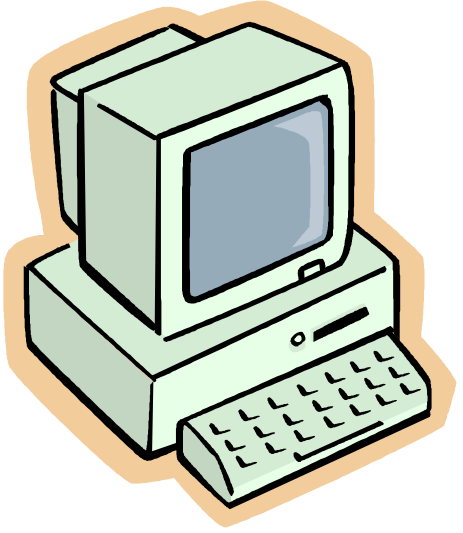
メイソン氏はテニスが趣味である．
メイソン氏は午後3時に来た．
メイソン氏は薬剤師である．

ブラウン警部はコンピュータを使いながら推理する



犯人には殺人の動機があって，毒薬が入手できて，それを投与する方法を知っている人物だから，えーっと．．．

犯人は誰かという話よりも…



このコンピュータのプログラムはどのように作られてるのかについての勉強を始めようじゃないか.

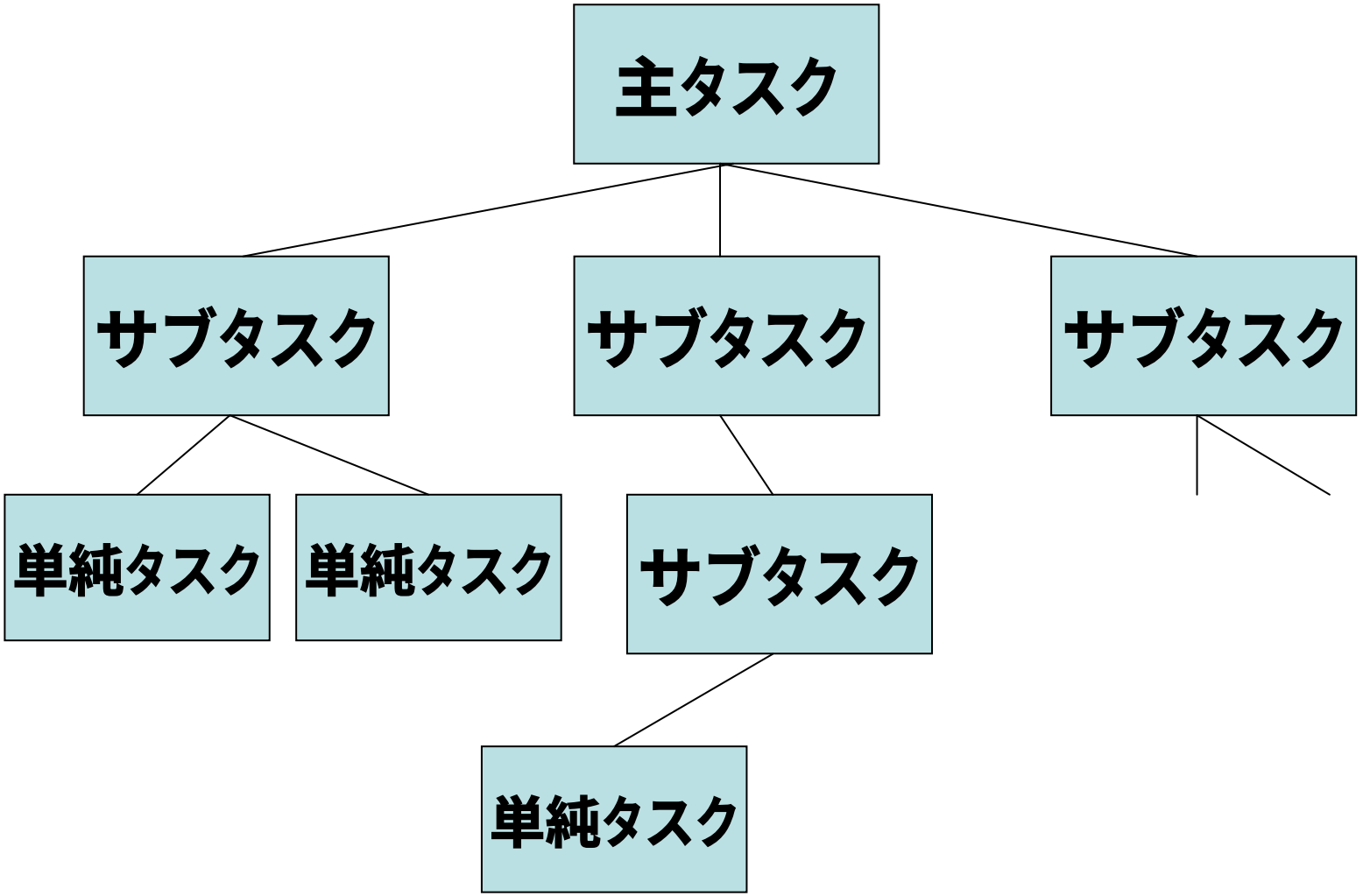
高度に複雑な問題を処理する方法

1. **問題を扱いやすく表現する.**
2. **問題をより簡単なサブタスクに分解し, さらに簡単に理解できるレベルまで各サブタスクを繰返し分解する. この分解によって得られた各サブタスクの解答を集めると, 問題全体に対する解答になる.**

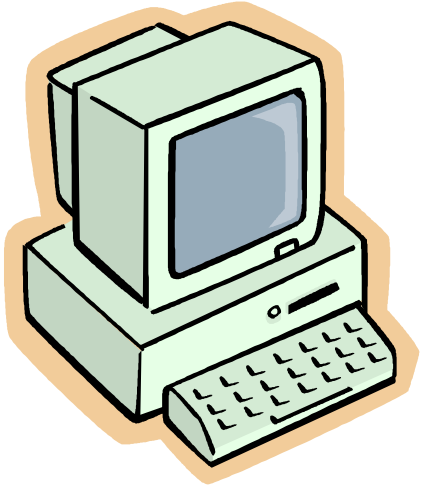
task: [**名詞**] (**つらくて骨の折れる**) **仕事**; (**課された**) **務め.**

— **研究社新英和中辞典第7版**

タスクの分解



データベースプログラム



ダンスモア家のコンピュータの中にあったような情報の蓄積は，**データベース**と呼ばれる．

データベースに情報を記憶させ，データベースを検索して，質問に答えるプログラムは，**データベースプログラム**と呼ばれる．

我々は，データベースプログラムを作ろうとしている．

問題を扱いやすくできるように表現する(1)

家族の情報は，一連の事実として平叙文で記憶されているとする．

メイソン氏は3時に来た．
(Mr. Maison visited at 3:00 P.M.)

メイソン氏は薬剤師である．
(Mr. Maison is a chemist.)

問題を扱いやすくできるように表現する (2)

記憶されている文を検索･出力するだけで，ユーザの質問に対する答えが全て得られると仮定する．

ユーザが質問したら，答えとなる事実がデータベースに存在するかどうかを決定し，
もし存在するならば，その事実を出力する．

コンピュータに推論させるようなことはしない．

コマンド制御ループ

以下の5つのコマンド(命令)を実行するデータベースプログラムを考える.

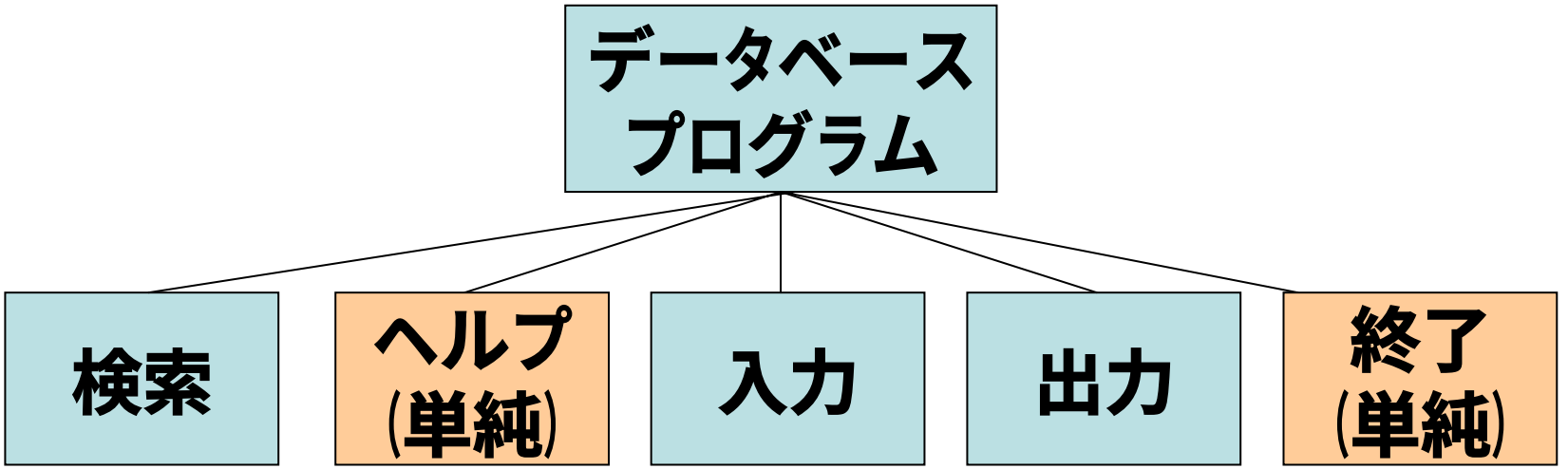
f	find	(検索　質問を受け取り, それに関する事実を全て出力する)
help	help	(ヘルプ　5つのコマンドリストを表示する)
i	input	(入力　事実を読み込む)
p	print	(出力　記憶されている全ての事実を出力する)
q	終了	

これらのコマンドは, whileループで読み込まれる.

プログラム Database0

```
program Database0(input, output);
var
  command: string;
begin
command := 'start';
while command <> 'q' do
  begin
  writeLn('Command:');
  readLn(command);
  if command = 'f' then
  質問の答えとなる事実を検索する;
  if command = 'help' then
  ヘルプメッセージを出力する;
  if command = 'i' then
  1つの事実を入力する;
  if command = 'p' then
  全ての事実を出力する;
  end;
end.
```

データベースプログラムのサブタスク



単純でない3つのサブタスクの検討

入出力ルーチン (入出力サブタスク)

入力

入力ルーチン:
メモリ内に新しい記憶場所を確保する.
新しい事実を読み込んで, そこに記憶する.

出力

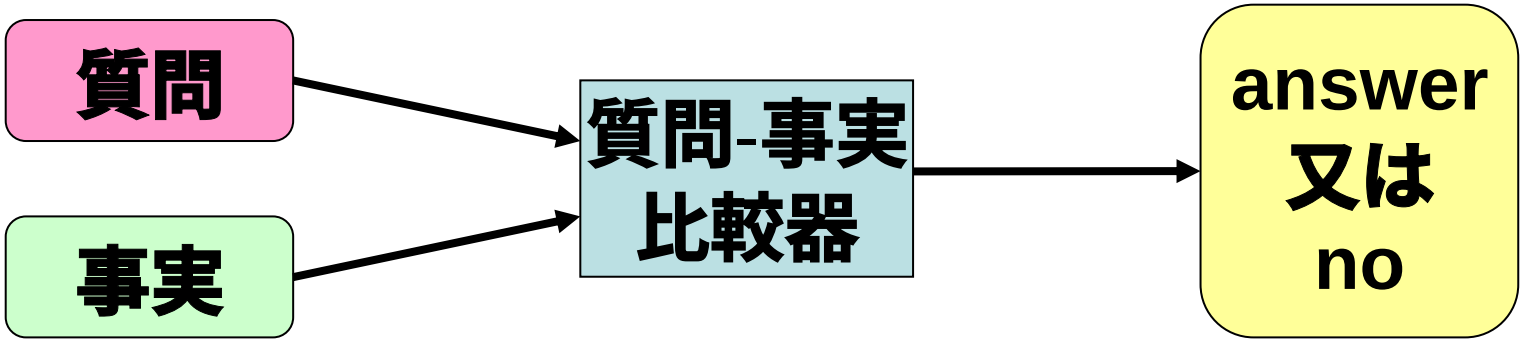
出力ルーチン:
事実が入っている各記憶場所ごとに,
その内容を出力する.

検索ルーチン (検索サブタスク)

検索

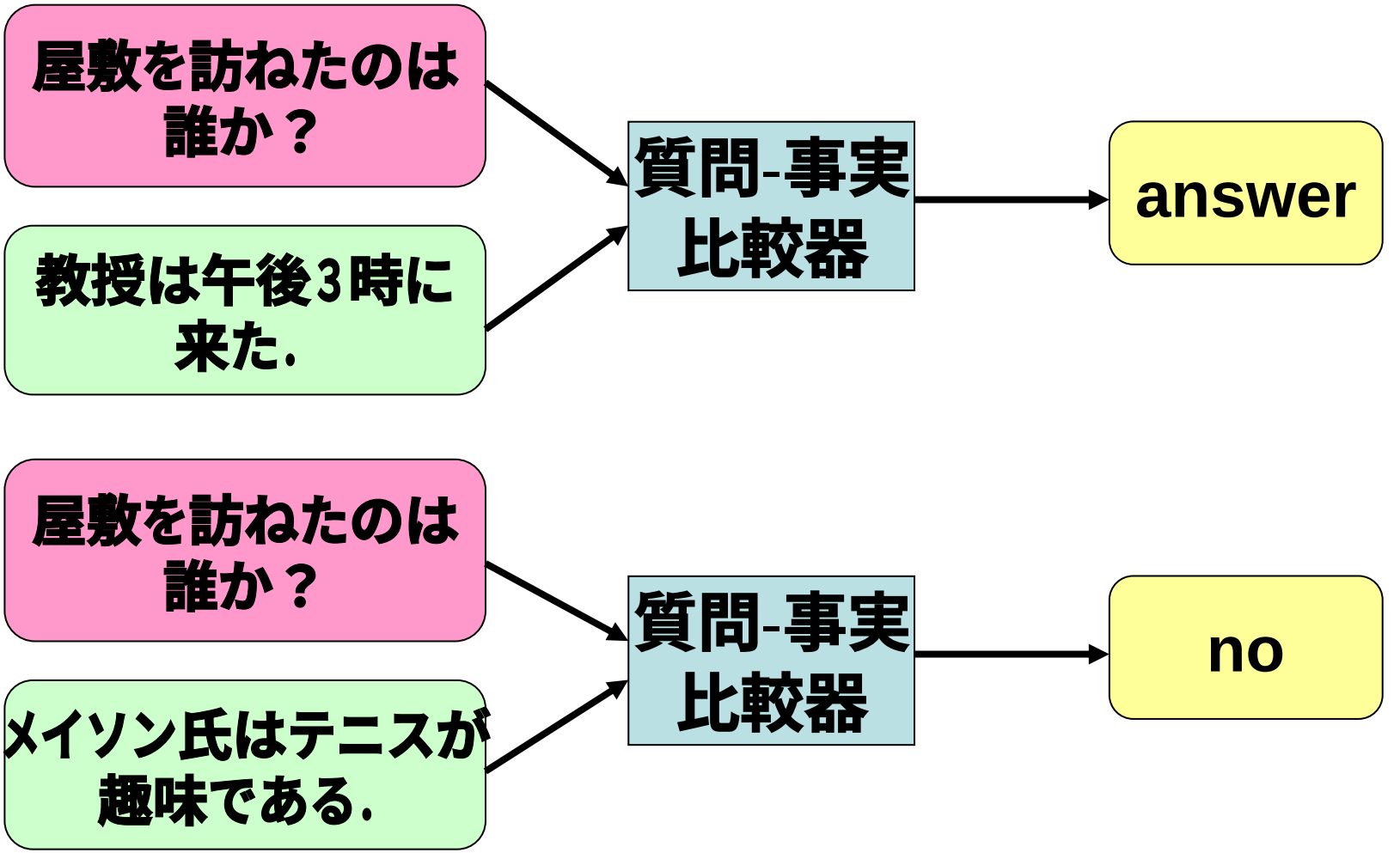
検索ルーチン:
ユーザの質問を読む.
事実が入っている各記憶場所調べて,
質問の答えに役立つ事実であれば,
それを出力する.

「質問-事実」比較器

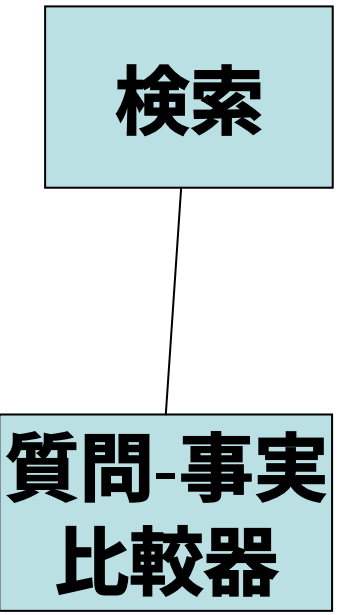


与えられた事実が, 与えられた質問に
部分的にでも答えていれば answer,
そうでなければ, no を返す関数.

「質問-事実」比較器 — 例

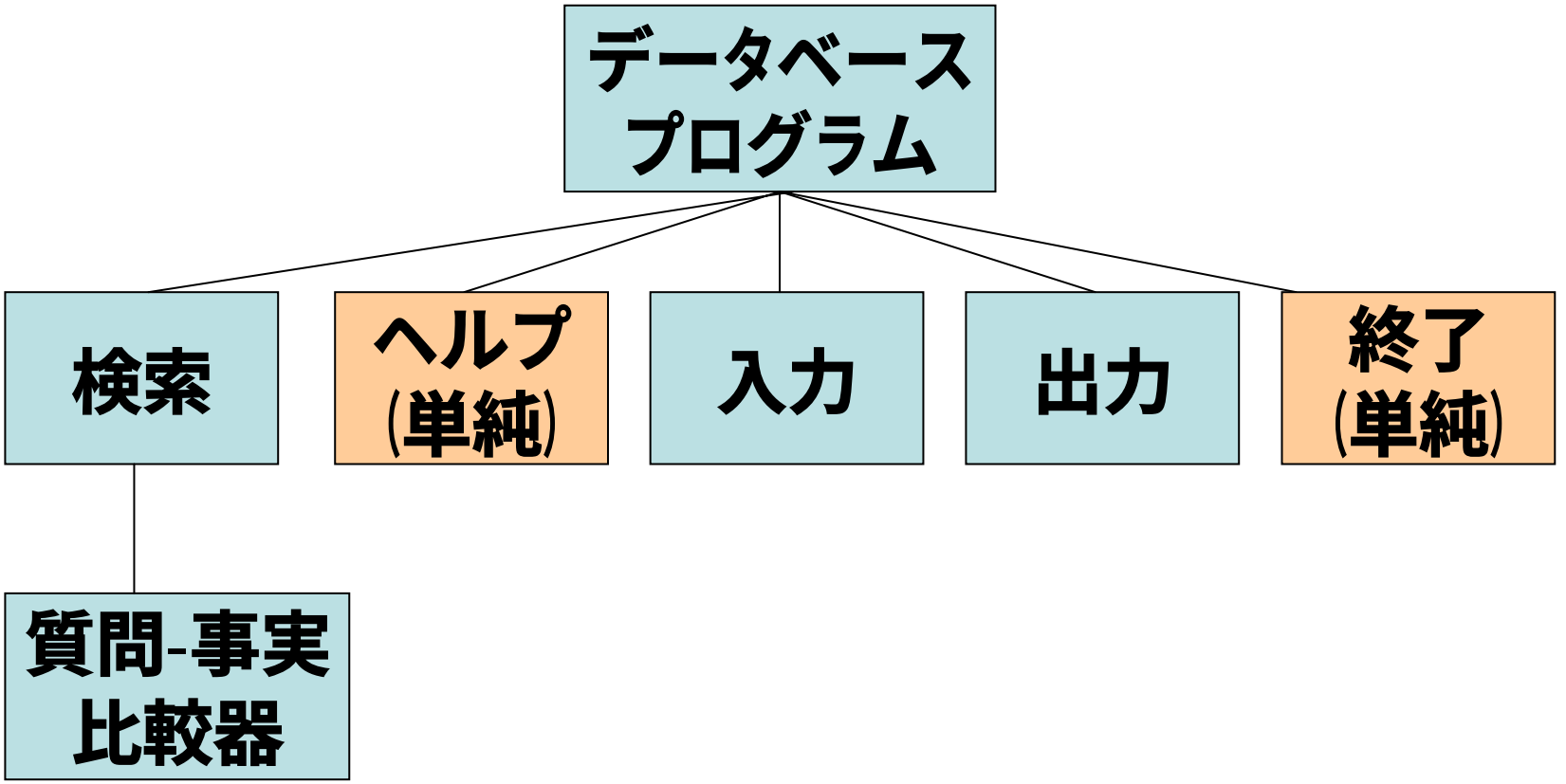

屋敷を訪ねたのは誰か？
教授は午後3時に来た.
質問-事実比較器
answer
屋敷を訪ねたのは誰か？
メイソン氏はテニスが趣味である.
質問-事実比較器
no

検索ルーチン (検索サブタスク)



検索ルーチン:
　ユーザの質問を読む.
　事実が入っている各記憶場所調べて,
　　質問-事実比較器を呼び出す.
　　比較器が「answer」と報告すれば,
　　　その事実を出力する.

データベースプログラムのサブタスク



サブルーチン

サブルーチンとは，サブタスクを実行するための文の並びである．
Pascal では，サブルーチンのことを**手続き**（procedure，プロシージャ）と呼ぶ．

Pascalにおけるサブルーチンの定義

```
procedure Byline;
  begin
  writeLn('-----------------------------');
  writeLn(' This program is written by  ');
  writeLn('     Lady Emily Dunsmore     ');
  writeLn('-----------------------------');
  end;
```

プログラムの中に
`Byline;`
という一文を入れる
だけで右の4行が
画面に表示される.

```
-----------------------------
 This program is written by
     Lady Emily Dunsmore
-----------------------------
```

Pascalにおけるサブルーチンの定義

```
program FirstSubrutine(input, output);
var ;{変数の宣言}
procedure Byline;
   begin
   writeLn('-----------------------------');
   writeLn(' This program is written by  ');
   writeLn('     Lady Emily Dunsmore     ');
   writeLn('-----------------------------');
   end;
begin {主プログラムの始まり}
……
end. {主プログラムの終わり}
```

サブルーチンの定義は，変数宣言の後に書く．

サブルーチン呼び出し

```
program FirstSubrutine(input, output);
var ;{変数宣言}
procedure Byline;
  begin
  writeLn('-----------------------------');
  writeLn(' This program is written by  ');
  writeLn('     Lady Emily Dunsmore     ');
  writeLn('-----------------------------');
  end;
begin
Byline;
…
Byline;
end.
```

主プログラムの中に入れられた
Byline;
という文は，サブルーチン呼び出しと呼ばれる．

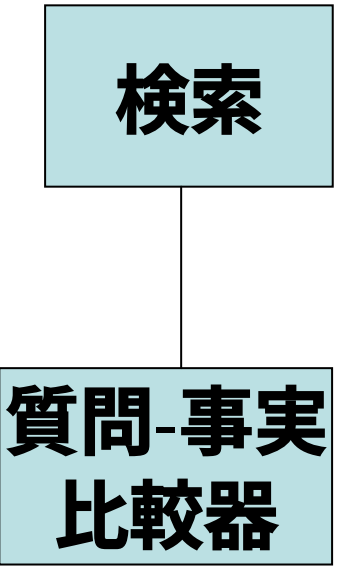
プログラムの実行の状況

```
program FirstSubrutine(input, output);
var ;{変数宣言}
procedure Byline;
  begin
  writeLn('-----------------------------');
  writeLn(' This program is written by  ');
  writeLn('     Lady Emily Dunsmore     ');
  writeLn('-----------------------------');
  end;
begin
Byline;
…
Byline;
end.
```

サブルーチンを用いることの効用

1. プログラムの中で，同じ行を何度も繰り返して書く必要がなくなる．
2. プログラマはある仕事を他から完全に隔離でき，問題の単純化に役立つ．

2．問題の単純化



検索ルーチン：
　ユーザの質問を読む．
　事実が入っている各記憶場所調べて，
　　質問-事実比較器を呼び出す．
　　比較器が「answer」と報告すれば，
　　　その事実を出力する．

「質問-事実比較器」という部分をサブルーチンにしてしまえば，検索ルーチンは簡単なものになる．

「質問-事実比較器」サブルーチンは，検索ルーチンからは切り離して後で考えることができる．

今日はおしまい

大学祭で疲れてしまったよ．この続きはまた来週にしよう．
みんなは，この前のレポートの決定木のプログラムをサブルーチンを使って簡単にすることを考えておいてくれ．
もちろん，前回のレポートについて「決定木の内容について問題がある」というコメントをもらった人は，決定木の内容自体を意味のあるものにすることを期待している．