

情報とコンピュータ

静岡大学工学部

安藤和敏

2004.10.04

この講義でなにを教わるのか？

コンピュータ科学 (= コンピュータ・サイエンス
= 情報科学)

コンピュータ・リテラシ(エクセル, ワードの使い方等)については, **教えない**. (そういうことは, 専門学校のパソコン教室や静岡大学の別の講義で教えられているかも知れない.)

コンピュータ科学とは何か？

コンピュータ科学とはなにか？

1. アルゴリズムとデータ構造
2. プログラミング言語
3. コンピュータ・アーキテクチャ
4. 数値および記号計算
5. オペレーティング・システム
6. ソフトウェアの方法論と工学
7. データベースおよび情報検索システム
8. 人工知能とロボティクス
9. 人間とコンピュータの関係

アルゴリズム コンピュータ科学に おける最も重要な概念

アルゴリズム = コンピュータ・プログラムに書き直すことに適した、問題を解くための方法を記述したもの。

料理のレシピにも似ているが、レシピはコンピュータ・プログラムに書き直すことには適していない。

NHK教育「ピタゴラスイッチ」と言う番組で「アルゴリズムたいそう」というのがあった。

カレーのレシピ

- 1 . なべにサラダ油を大さじ1杯そそいで熱する .
- 2 . みじん切りにしたたまねぎを炒める .
- 3 . 一口サイズに切ったジャガイモとニンジンと肉を炒める .
- 4 . 中火で煮て沸騰したら , アクを取る .
- 5 . 中火で材料がやわらかくなるまで煮る .
- 6 . 一旦火を止めて , カレーのルーを割り入れる .
- 7 . さらに , 10分くらい弱火で煮込む .

アルゴリズムの例

二つの整数の最大公約数を求めるアルゴリズム

1. 二つの整数のうち, 小さい方を y とし, 大きいほうを x とする.
2. y が 0 ならば, 終了. 答えは, x である. そうでなければ, 次の3に進む.
3. x に y を代入して, y に x を y で割った余りを代入する.
4. 2へ戻る.

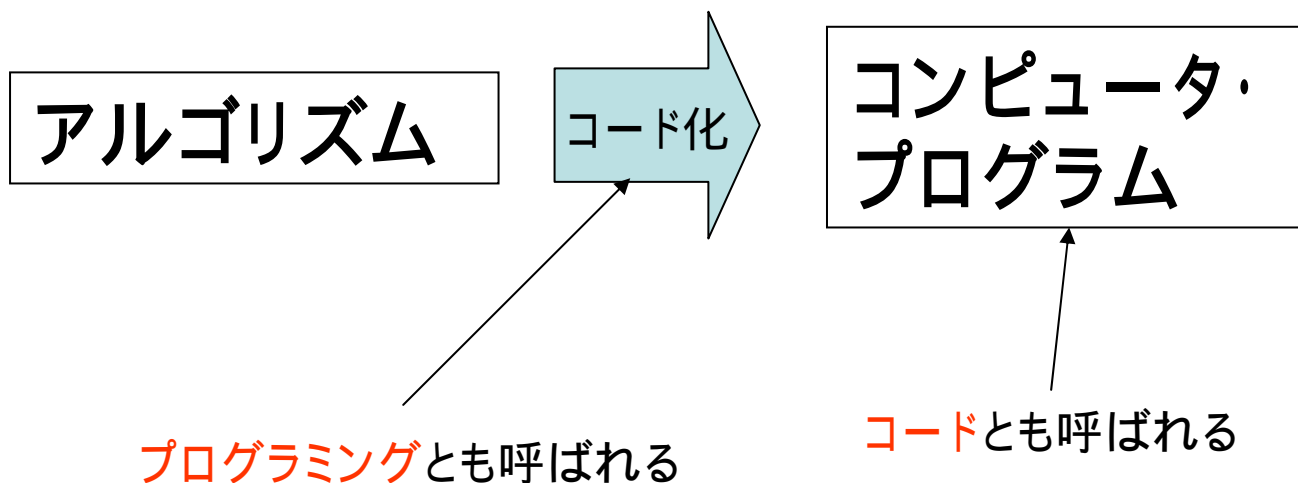
ユークリッドの互除法のPascalプログラム

```
program prog1(input, output);
var a,b,x,y,amari : integer;
begin
  a :=51;
  b :=30;

  x := a;
  y := b;
  while(y <> 0) do
  begin
    amari := x mod y;
    x := y;
    y := amari;
  end;
  writeLn(x);
end.
```


アルゴリズムのコード化

アルゴリズムをC言語, Pascal, Java 等の言語に書き直すことを**コード化**という。



テキスト

A . W . Biermann 著「やさしいコンピュータ科学」アスキー出版社, 1993年

(A . W . Biermann: Great Ideas in Computer Science. MIT Press, 1990. の翻訳)

本の帯「MIT(マサチューセッツ工科大学)で使われている教科書「Great Ideas in Computer Science」の日本語版です。専門家のみならず、コンピュータ科学に興味を持つすべてのの方々にコンピュータの深遠な概念をやさしく解説します。」

このテキストの特徴

- 数学的なアプローチをとらない。
- プログラミング中心ではなくて、コンピュータ科学の概論
- しかし、プログラミングが全くないわけではなくて、実際はある程度プログラミングについて字数を割いている。(14章のうち4章くらい。)
- プログラミング言語はPascalを用いている。

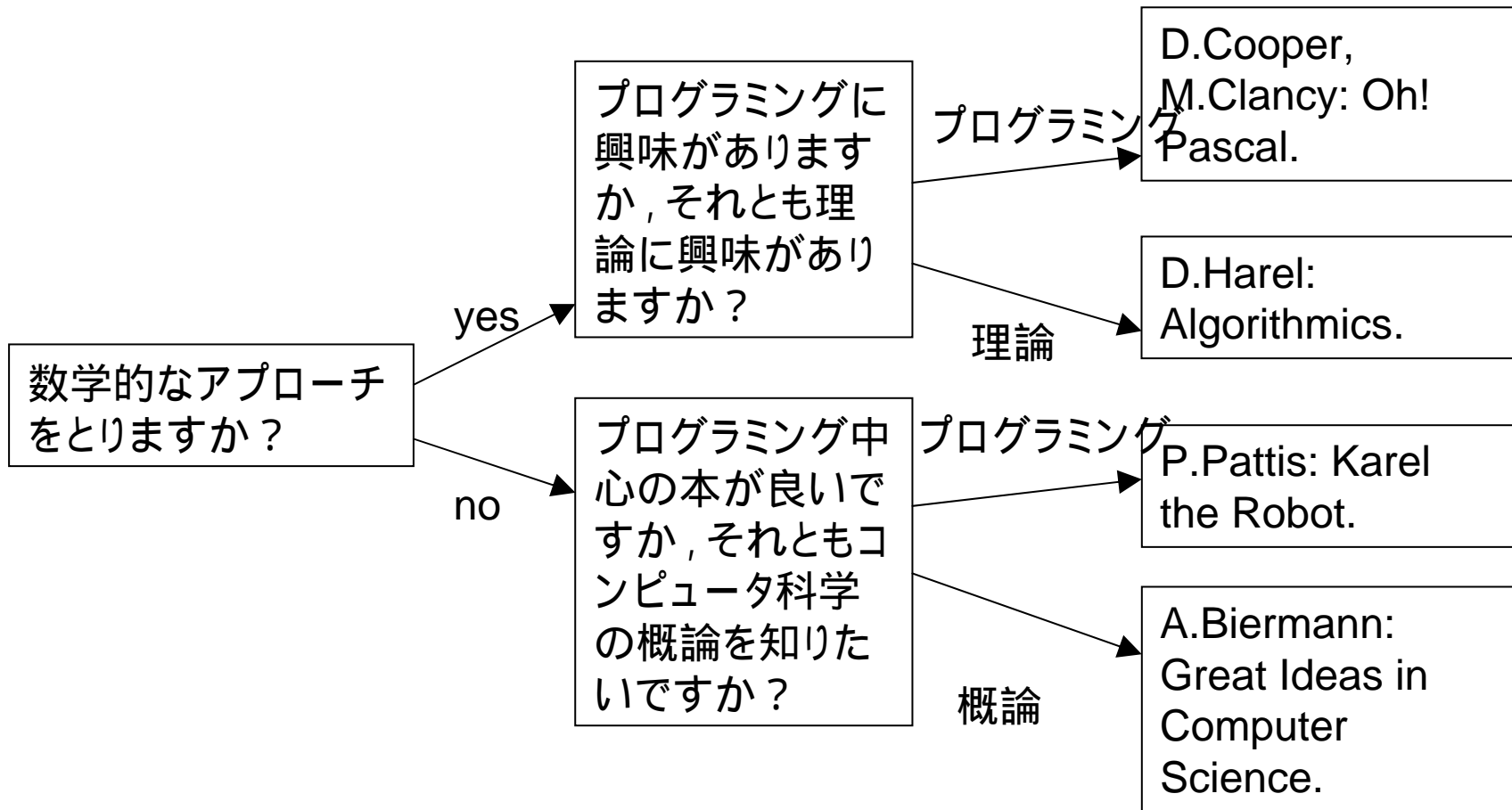
この講義を履修するために必要なもの

- パソコン(プログラミングのため)
- Pascal の処理系 : HelloPascal
<http://coconut.sys.eng.shizuoka.ac.jp/ic/> に置いてあるので各自ダウンロードしてください.

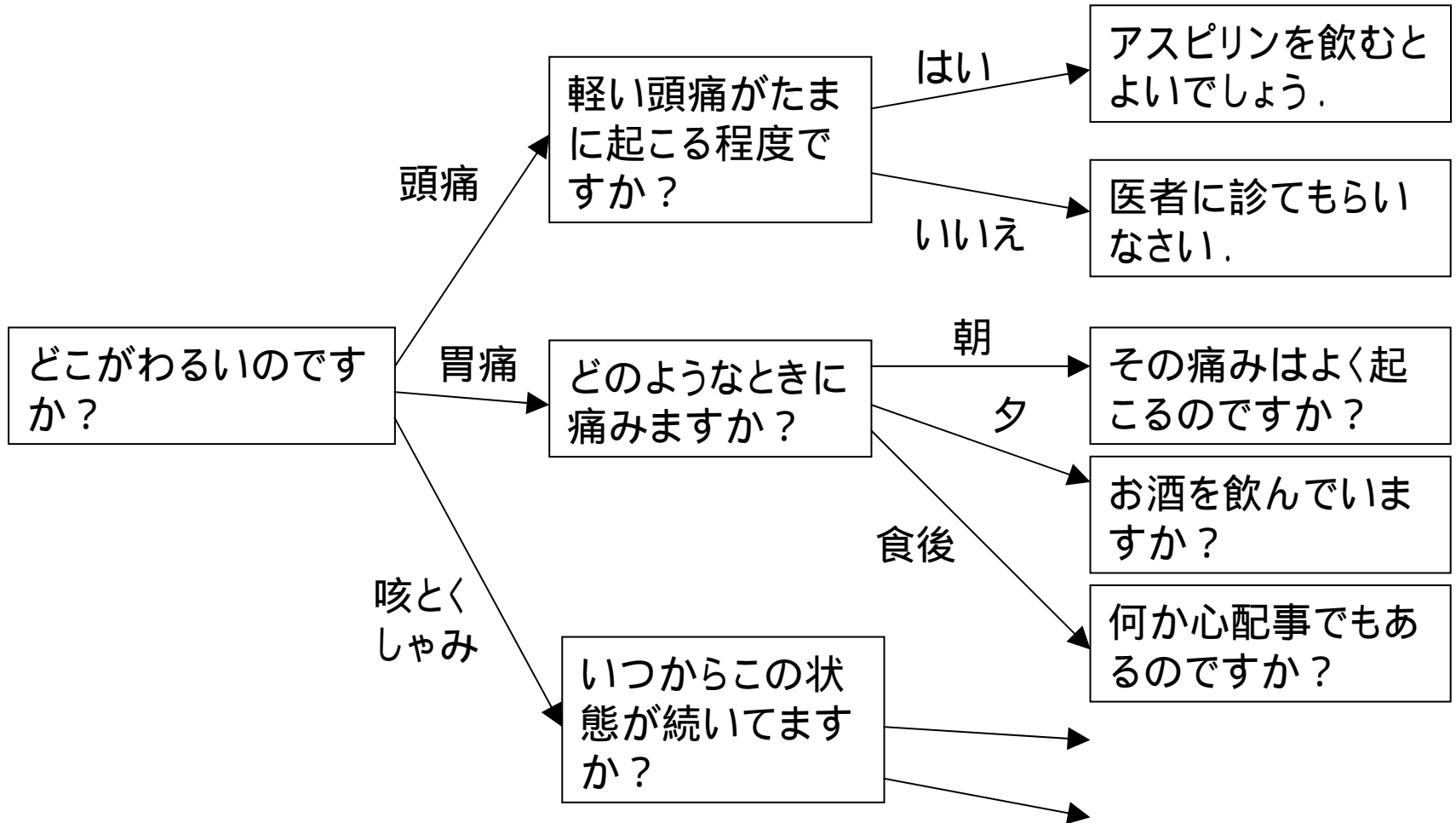
1章 プログラミング入門

決定木のコーディング

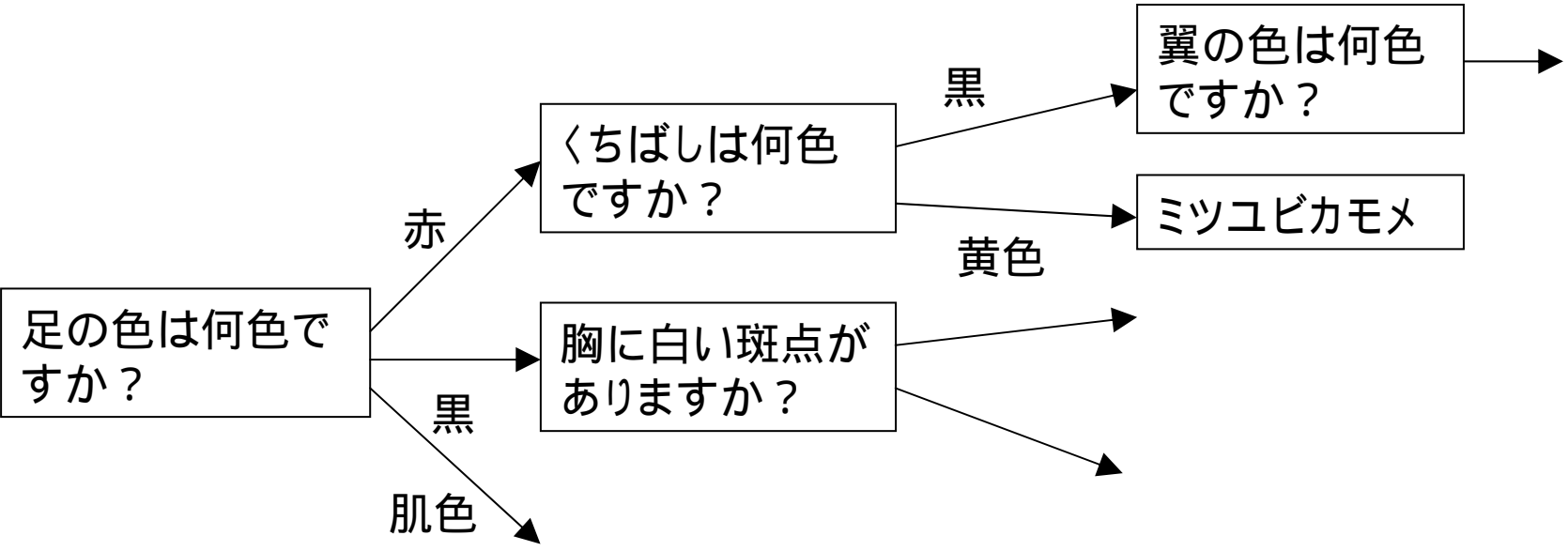
例1：図書推薦の決定木



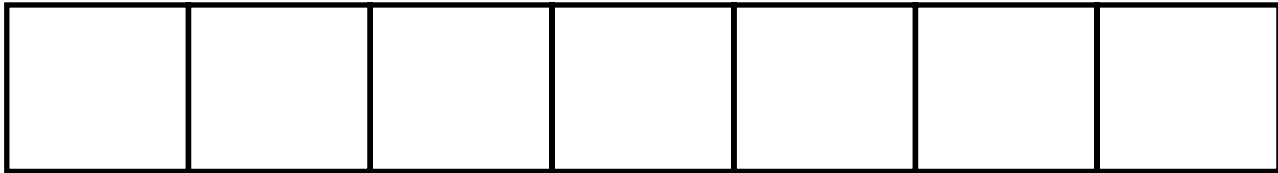
例2：医療アドバイスの決定木



例3：カモメの分類の決定木

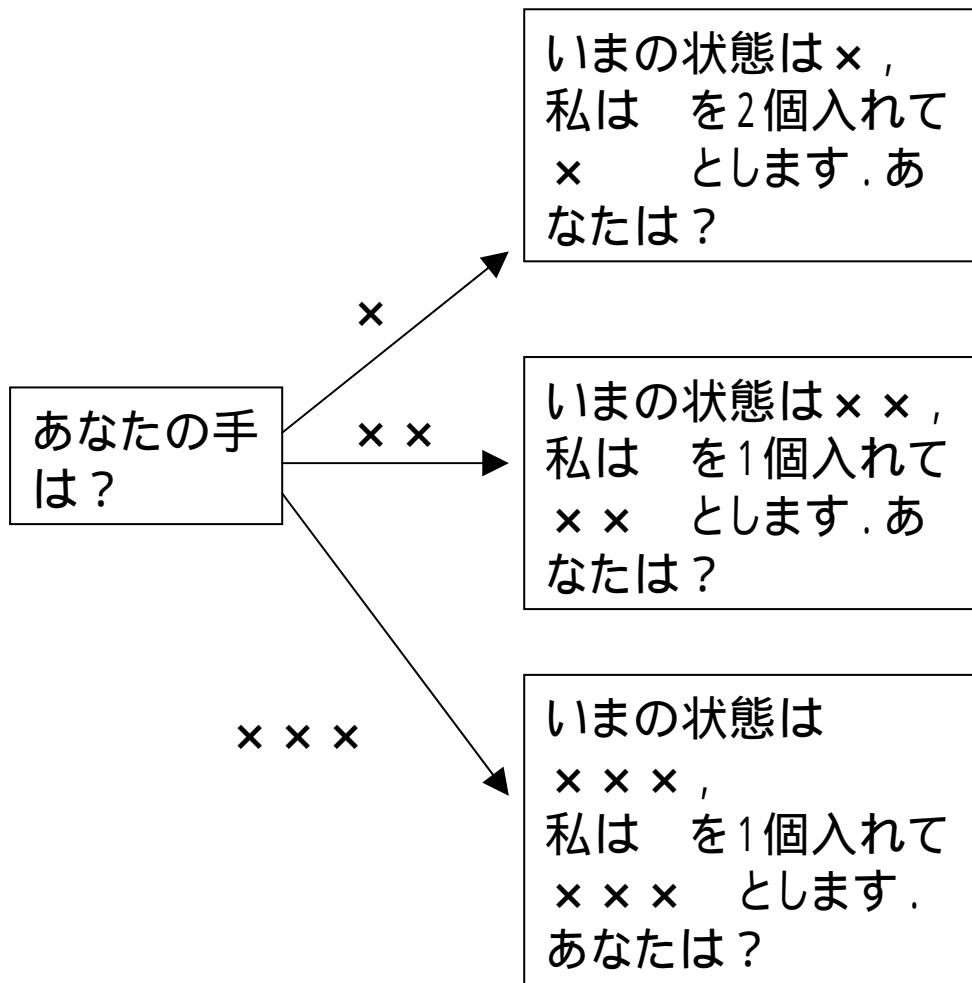


例4：ニム

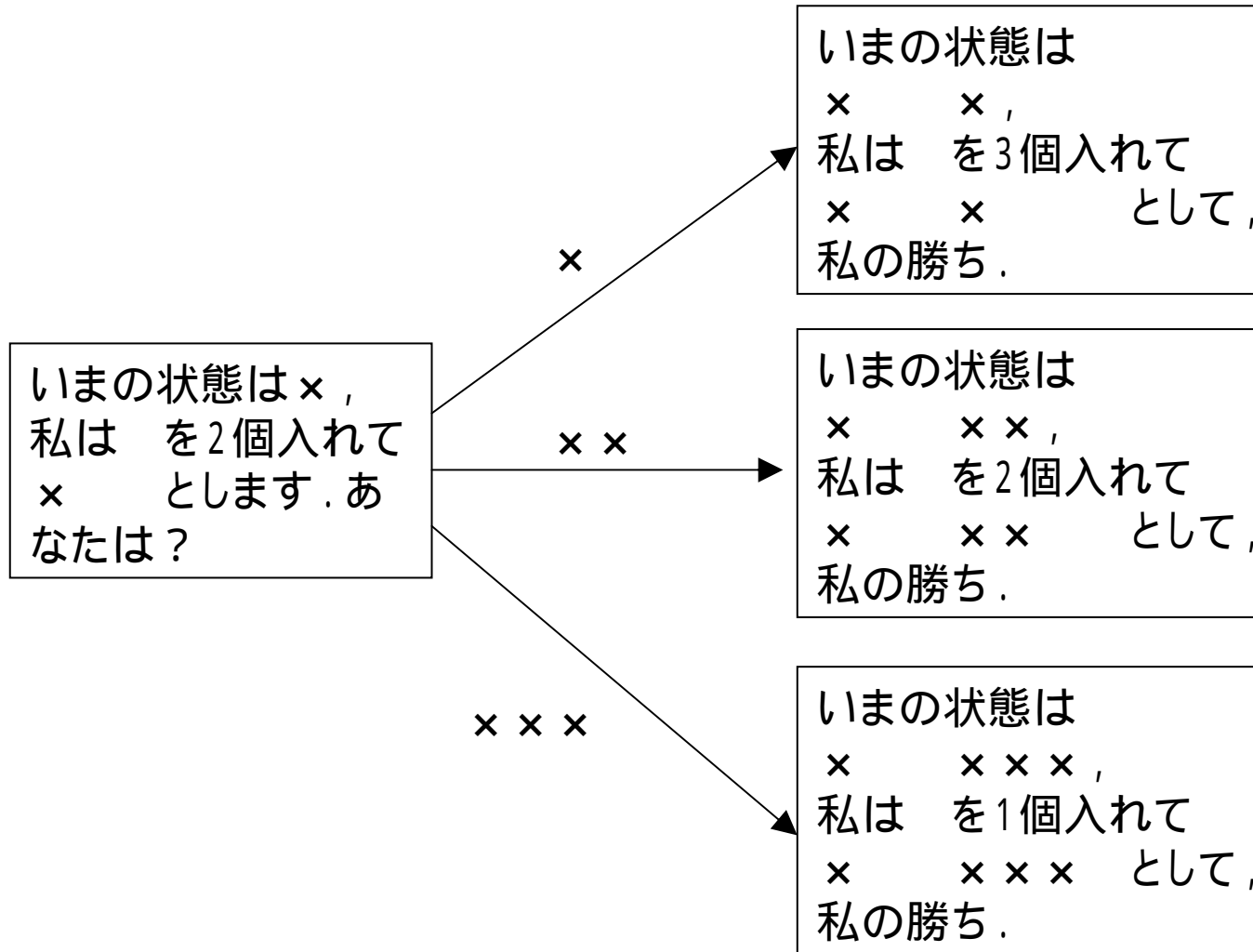


- ・ 2人で遊ぶゲーム。
- ・ 上の図のような7つのマス目を描き、
- ・ 最初の人には×を左端から1～3個書き込める。
- ・ 相手はそれに続けて を同じく1～3個書き込める。
- ・ これを繰り返していったら、一番右端のマスに書き込んだ人の勝ち。

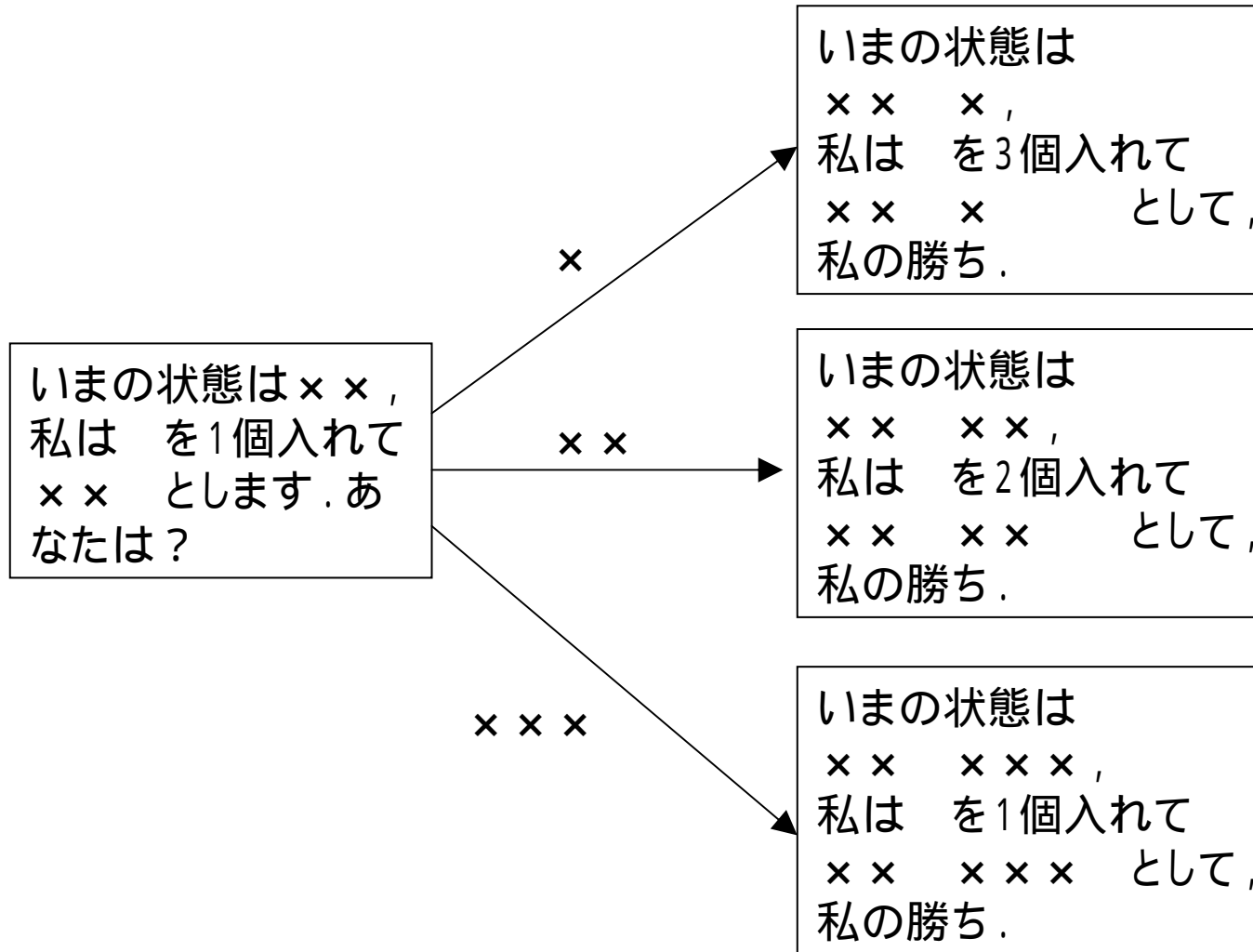
例4：二ムの決定木



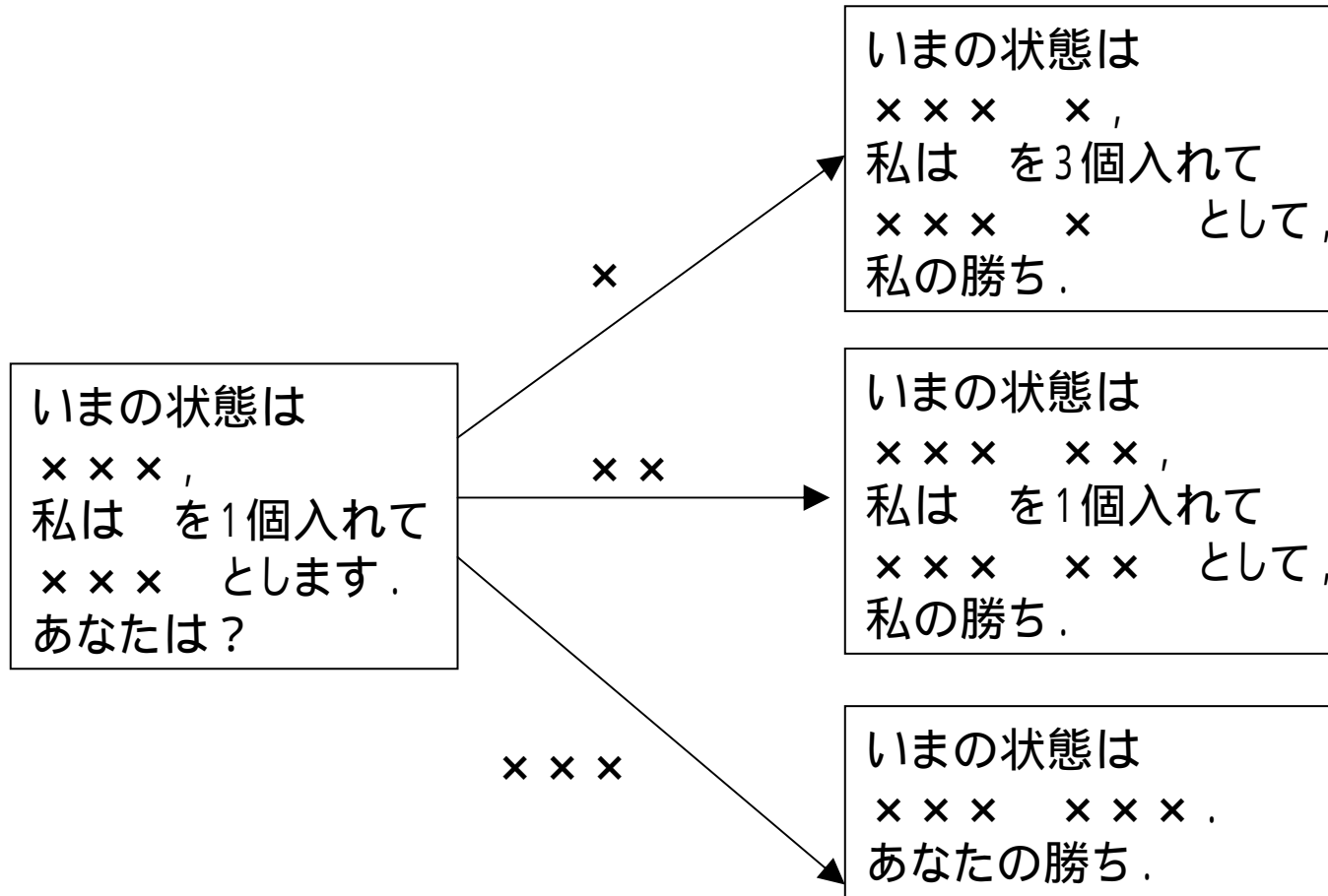
例4: ニムの決定木(続き)



例4: ニムの決定木(続き)



例4: ニムの決定木(続き)



プログラミングをはじめするには

コンピュータ・プログラムとは、コンピュータに実行させるコマンド(命令)を並べたもの。

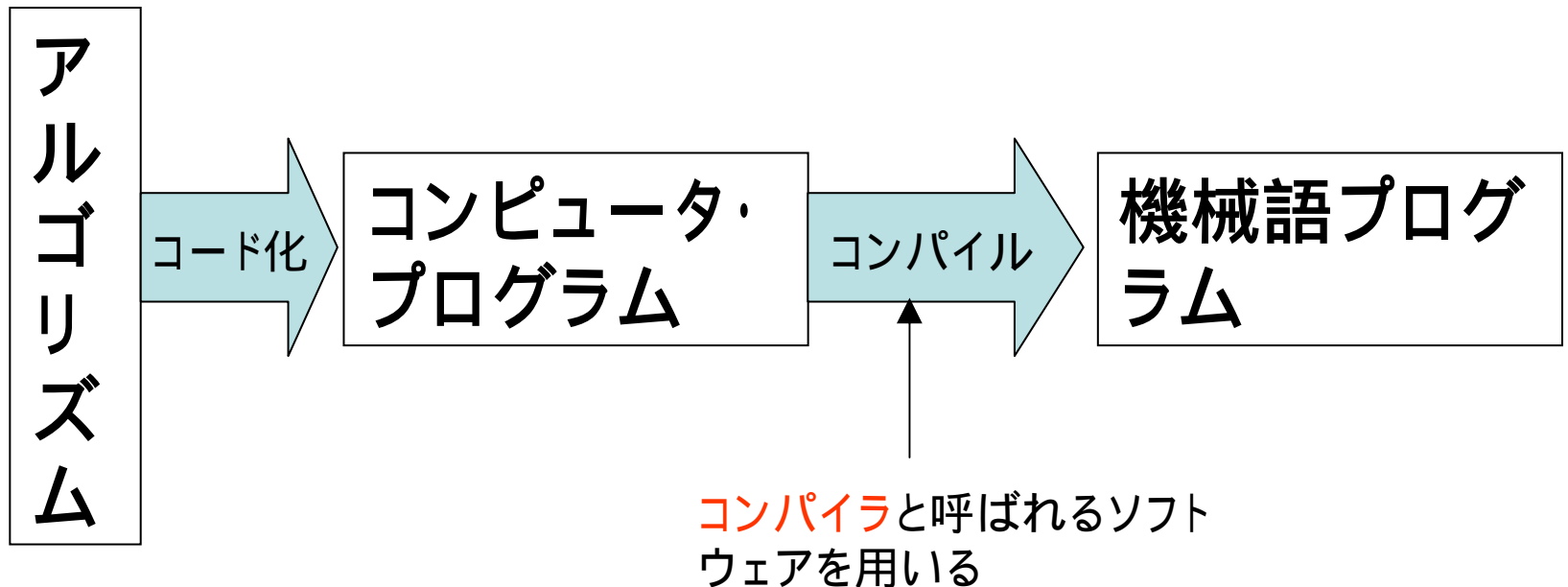
コンピュータ・プログラムは、コンピュータ上で**走らせ**(run)たり、**実行する**(execute)ことができる。

プログラムを実行するために必要なもの

- コンピュータ (MacよりもWindowsが望ましい)
- Pascalを処理させるためのソフトウェア・システム (Hello Pascal)
- コンピュータを正しく起動・動作させるためのマニュアル
- コンピュータをよく知っている人

機械語

実は、コンピュータ・プログラムはそのままでは、実行できない。



Pascalのコンパイラ

- テキストにはTurboPascalというコンパイラが紹介されているが、この講義ではHelloPascalというコンパイラを用いる。

はじめてのPascalプログラム

```
program FirstCode(input, output);  
begin  
    writeln('  Great Ideas  ');  
    writeln('    in    ');  
    writeln(' Computer Science ');  
end.
```

プログラムの書式

Pascalプログラムは、ヘッダ、キーワード `begin`、
セミコロン(`;`)で終わる一連の文、キーワード `end`
からなる。

```
program FirstCode(input, output);  
begin  
    writeln('    Great Ideas    ');  
    writeln('        in        ');  
    writeln(' Computer Science ');  
end.
```

ヘッダ
begin
文
文
文
end.

文

プログラムにおける**文**とは、コンピュータに指示する個々の**コマンド**(命令)のことで、英語の命令文に相当する。

例えば、

```
writeLn(' Great Ideas ');
```

という文は、

「画面に“ Great Ideas “という文字を書き出せ」
というコンピュータに対する命令である。

プログラムの実行

プログラムは、特別な文で指示しない限りは、コンピュータによって上から下に向かって、1文ずつ実行される。

```
program FirstCode(input, output);  
begin  
    writeln('    Great Ideas    ');  
    writeln('        in        ');  
    writeln(' Computer Science ');  
end.
```

文の意味と構造

文

```
writeln(' Great Ideas ');
```

=

```
writeln("");
```

+

```
Great Ideas
```

構文

データ

構文は正しくないといけない(1)

```
{ FirstCode }  
program FirstCode(input, output);  
begin  
    writein('    Great Ideas    ');  
    writeln('        in        ');  
    writeln(' Computer Science ');  
end.
```

構文は正しくないといけない(2)

```
{ FirstCode }  
program FirstCode(input, output);  
begin  
    writeln('    Great Ideas    ');  
    writeln('        in        ');  
    writeln(' Computer Science ');  
end.
```


構文は正しくないといけない(3)

```
{ FirstCode }  
program FirstCode(input, output);  
begin  
    please writeln('    Great Ideas    ');  
    writeln('        in        ');  
    writeln(' Computer Science ');  
end.
```

データは間違っているいてもプログラムは 実行される(1)

```
{ FirstCode }  
program FirstCode(input, output);  
begin  
    writeln('    Grit Iders    ');  
    writeln('        on        ');  
    writeln(' askdjfak%%768df');  
end.
```

こういう書き方をしても実行される
(けど、読みにくいのでやめましょう。)

```
{ FirstCode }
```

```
program
```

```
    FirstCode(input, output);
```

```
begin
```

```
    writeln
```

```
        (' Great Ideas ');
```

```
    writeln('    in ');
```

```
        writeln(' Computer Science ');
```

```
end.
```

Secod Code

```
{ SecondCode }  
program SecondCode(input, output);  
begin  
    writeln('*****');  
    writeln('*                *');  
    writeln('*   Decision Trees   *');  
    writeln('*       決定木       *');  
    writeln('*                *');  
    writeln('*****');  
end.
```