知識処理
第四回

知識の表現
その3
データベース（前編）

今日の内容

1. 大量の宣言的知識（特に、事実）
   を扱うためのデータベース
2. データベースの基本
   関係＝テーブル（レコードの集合）
3. SQL：データベース問い合わせ言語
4. テーブルの正規化

前回の問題

- 「XはYの孫である」という関係を、grandchild_of(X, Y)と書くとします。
- parent_of(A, B)という関係を使ってgrandchild_of(X, Y)を定義してください。
  - ここで「定義する」とは、grandchild_of(X, Y) :- ○○.という「ルール」を書くこと。
- 次に、「XはYの祖父母である」という関係を、grandparent_of(X, Y)と書くとします。
- grandchild_of(A, B)を使ってgrandparent_of(X, Y) を定義してください。

解答

- grandchild_of(X, Y)の定義（ルール）

  grandchild_of(X, Y) :- parent_of(Y, Z), parent_of(Z, X).

  YがZの親で、ZがXの親ならば、XはYの孫

- grandparent_of(X, Y)の定義（ルール）

  grandparent_of(X, Y) :- grandchild_of(Y, X).

  YがXの孫ならば、XはYの祖父母

知識（特に、事実）が膨大な量になったときにどうするか？

答え：データベースを使う

- データベース
  - 事実を表すデータを集めて、簡単に検索できるようにしたもの
- 関係データベース（RDB）
  - すべてのデータがテーブル（表形式）上に配置されるデータベース
  - 一つの行をレコードと呼び
    レコードには固有の識別子(ID)あるいは名前を付ける

名前	属性1	属性2	属性3
名前1	データ1-1	データ1-2	データ1-3
名前2	データ2-1	データ2-2	データ2-3
名前3	データ3-1	データ3-2	データ3-3
名前4	データ4-1	データ4-2	データ4-3

- 知識との対応関係
  - データは事実、属性（列）は関係
  - ルール（変数を含む知識）は扱えない

テーブルと事実

- テーブルの方が記述の制約が強い
  - 識別子あるいは名前の列には同じデータを複数入れられない
  - 一つのセルには複数のデータを入れられない

テーブル

名前	性別	母親	父親
家康	男	お大	松平広忠
秀忠	男	お愛	家康
忠吉	男	お愛	家康
家光	男	お江	秀忠
お愛	女	不明	戸塚五郎大夫忠春
お江	女	お市	浅井長政
千姫	女	お江	秀忠

事実(Prolog形式)

```
male(家康).      female(お愛).
male(秀忠).      female(お江).
male(忠吉).      female(千姫).
male(家光).

mother_of(お愛, 秀忠).
mother_of(お愛, 忠吉).
mother_of(お江, 千姫).
mother_of(お江, 家光).

father_of(家康, 秀忠).
father_of(家康, 忠吉).
father_of(秀忠, 千姫).
father_of(秀忠, 家光).
```

データベースの利点

- 大量のデータを格納・管理できる
  - フォルダは、大量のファイルを格納すると非常に処理（一覧表示など）が重くなる
  - データベースは、大量のデータを格納しても重くならない（ファイルとしては一つのものとして認識される）
- データを高速に検索できる
  - インデキシング（索引付け。後述）という仕組みで高速に必要なデータを探し出すことができる
- 高機能なクエリ（問い合わせ）言語がある
  - SQL (Structured Query Language)と呼ばれる

SQL（1/2）

- 関係データベースへのクエリ（問い合わせ）用言語
- テーブルを作る
  - CREATE TABLE テーブル名 (属性名1　データ型,
    属性名2　データ型, ・・・・・・);
  - データ型には、整数(integer)、小数(float)、文字列(text)、日付(date)などがある
- データを入力する
  - INSERT INTO テーブル名 (属性名1, 属性名2, ・・・・・・)
    VALUES (属性1のデータ, 属性2のデータ, ・・・・・・);
- データを表示する
  - SELECT 属性名（すべての属性の場合は*） FROM
    テーブル名;

SQL（2/2）

- 条件を付けてデータを探す
  - SELECT 属性名 FROM テーブル名 WHERE 条件;
  - 条件は、属性名=ある値、属性名<ある値、など
  - ANDかORで区切って複数の条件が書ける
- データを更新する
  - UPDATE テーブル名 SET 属性名=データ WHERE 条件;
  - 条件を省略すると、テーブル内のすべてのレコードが更新される
- レコードを削除する
  - DELETE FROM テーブル名 WHERE 条件;

家系図プログラムみたび

- 家系図のテーブルを作る
  - CREATE TABLE 徳川家系図 (名前 text, 性別 text, 母親 text, 父親 text);
- テーブルにデータ(レコード)を入れる
  - INSERT INTO 徳川家系図 VALUES (‘家康’, ‘男’, ‘お大’, ‘松平広忠’);
  - INSERT INTO 徳川家系図 VALUES (‘秀忠’, ‘男’, ‘お愛’, ‘家康’);
  - INSERT INTO 徳川家系図 VALUES (‘家光’, ‘男’, ‘お江’, ‘秀忠’);
  - ……

SQLによる家系図プログラム（1/3）

- 家康の「息子」の名前は？

  SELECT 名前 FROM 徳川家系図 WHERE 父親='家康' AND
  　性別='男';

- 家康の「孫息子」の名前は？

  まず、家康の子供のテーブルを作る

  CREATE TABLE 家康の子供 (名前, 性別, 母親, 父親) AS SELECT *
  　FROM 徳川家系図 WHERE 父親='家康';

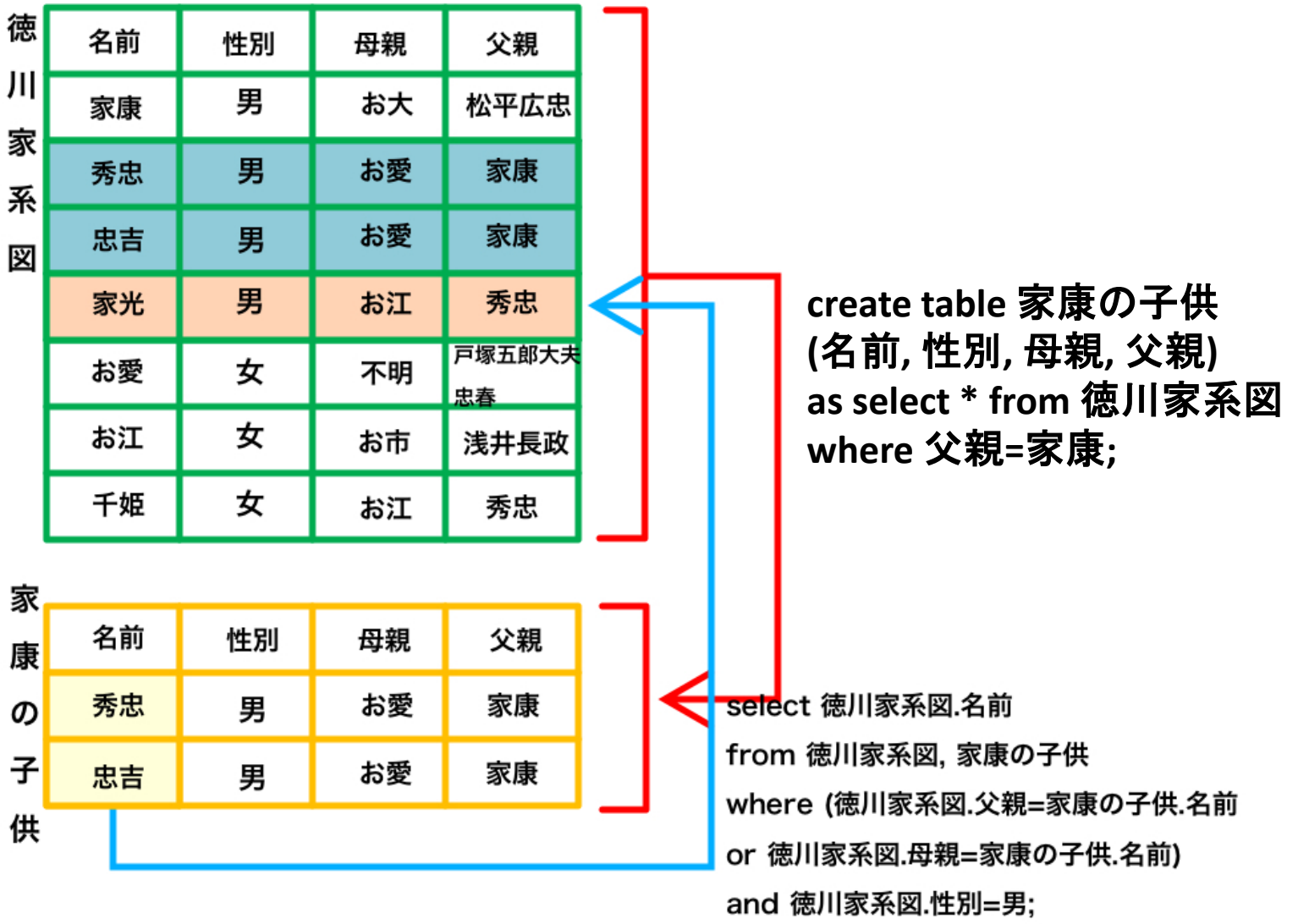
  次に、「家康の子供」テーブルの名前属性が「徳川家系図」の
  　父親か母親と一致するもので、性別が男のものを探す

  SELECT t.名前 FROM 徳川家系図 AS t, 家康の子供 AS c
  　WHERE (t.父親=c.名前 OR t.母親=c.名前) AND t.性別='男';

  - テーブル名（ASでエイリアス(別名)を付けられる）.属性名で、それぞれのテーブルを参照できる

SQLによる家系図プログラム（2/3）

- 家康の孫息子の名前は？



SQLによる家系図プログラム（3/3）

- 「孫息子」や「孫娘」に関する質問は、実は、CREATE TABLEを実行せず、1つのSELECT文で実行できる
- サブクエリを使う
  - SELECT文の中に別のSELECT文を書ける
    - SELECT t.名前 FROM 徳川家系図 AS t WHERE (t.父親 IN (SELECT t.名前 FROM t WHERE t.父親='家康') OR t.母親 IN (SELECT t.名前 FROM t WHERE t.父親='家康'));
  - サブクエリに関しては次回に詳しく説明します

インデキシング（索引付け）

- 検索速度を向上させるためにインデックス（索引）を作る
- 検索要求に頻繁に現れる属性に対して行う
- SQLでは、
  CREATE INDEX 索引名 ON テーブル名(属性名);
- 索引はデータベース内の任意の内容（文字や数字）とそれに関連するレコードの記憶場所（ディスク上の空間）を対応付ける
  - この仕組みのおかげで、通常のファイルより高速にデータを見つけることができる

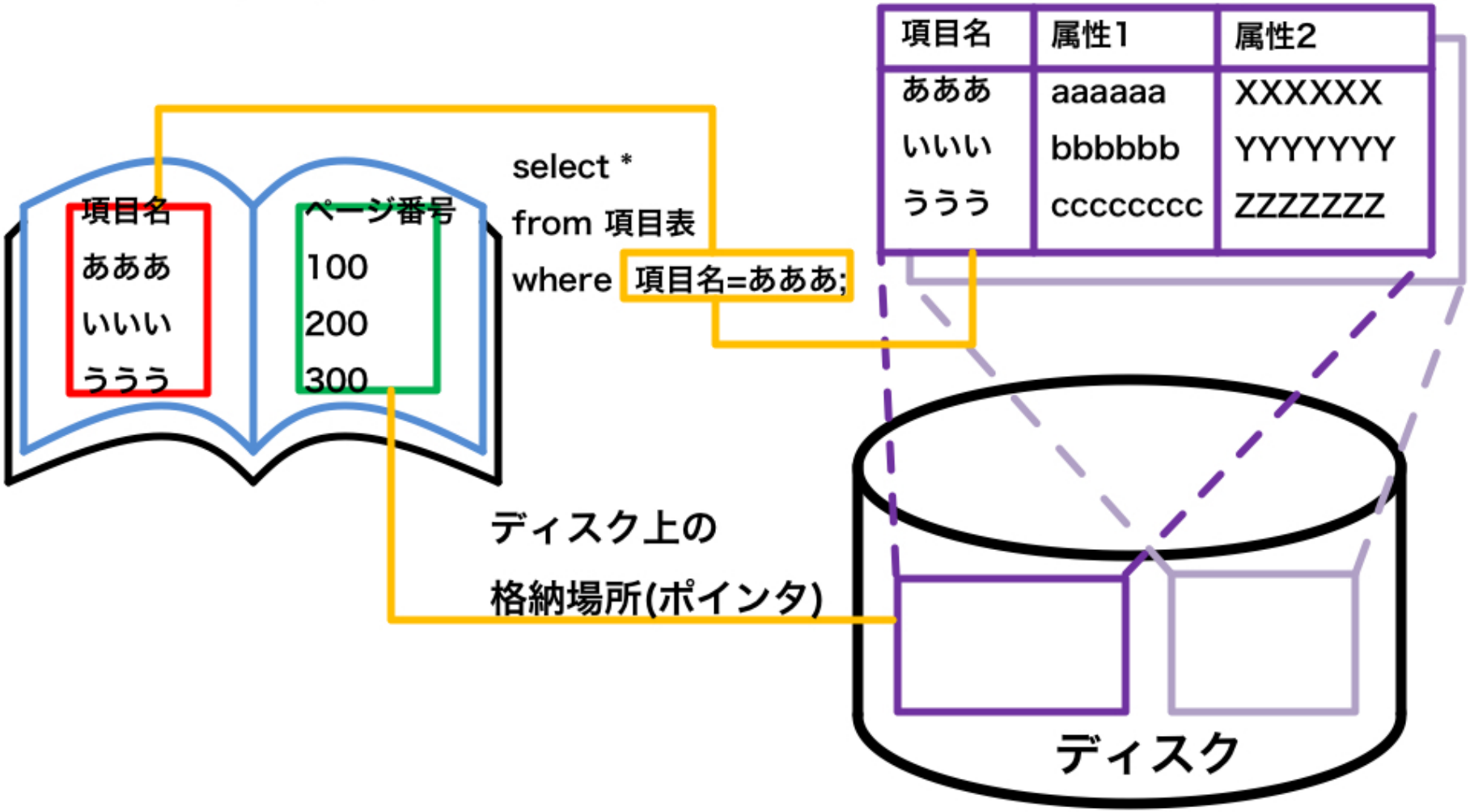
インデックス

インデックスとは

例えば、本の索引のようなもの

データベースは

ディスク上に定義される論理的空間



ここで、ブレイク

講義に関するコメント(1/3)

- prologで質問するのはデータベースで検索をかけるのとは何か違うことができるのでしょうか。データの集合=事実でルールのようなものも付加すれば、少なくとも家系図レベルのことはできる気はしました。
- 講義中にルールを具体化することで新しい事実が導かれるとありました。つまり、人間はルールに基づく相対評価を行わなければ、新しい事実を導き出すことはできないということになると思います。そもそも人間は絶対的な評価を行うことは殆どなく、大抵は、「■は▲だったから、●は★である」といったようなこれまでの経験や知識を基盤に相対的に物事を評価します。この評価方法は脳に保存された知識を組み合わせるといった、人間特有の柔軟性あってこその方法であると思います。人間とAIが搭載するロボットがよりナチュラルに会話できるまでの関係を築こうとしているのであれば、人間レベルの相対的な評価を行わなければならないと考えたのですが、どうでしょうか？

- データベースの検索と基本的にできることは同じですが、Prologの利点は知識が簡潔に表現できることです。また、パターンマッチングに関しても、非常に良くできています。ただし、その内部処理に関しては改良の余地がとても多いです。例えば、そのままではビッグデータには対応できません。例えば、今日の講義のように内部的にデータベースを使うのが一つの方法だと思います。
- その考えでだいたい合っていると思います。人間の脳は抽象化が得意で、ルールを自然に構築していると思われます。まだ、その仕組みは人工的には実現できません。それゆえに、人間と機械が自然な会話を行うことはまだできないのです。

講義に関するコメント(2/3)

- 今日の問題にあるgrandchild_of(A,B)の関係を人工知能に学習させる際に、あらかじめその関係が成立する組を列挙しておくのと、毎回parent_of(A,B)を2回適用して求めさせるのとどちらの方が一般的な手法なのでしょうか？前者はデータが大きくなり、後者の方は時間がかかるというデメリットがあるように思えるのですが、やはりケースバイケースなのでしょうか。
- この言語(prolog)はコンピュータにうちこめば動作してくれるソフトはあるんですか。それとも学問用の言語で、実装時は他の汎用言語で自分で定義していくものなのでしょうか。
- prologがなぜ人工知能のプログラミングに適しているのかもう少し詳しく知りたいと思いました。このプログラミングは推論規則のように思えたのですが、これが人工知能のどのような面に生かされているのか興味を持ちました。
- ルールを使って一度生成したものをデータベースに保存して、以後は、データを検索するだけにするのがよいでしょう。人間の記憶もおそらく似たようなことをやっていると思います。データベースには一般に忘却の機能がありませんが、もしかすると、効率化のために必要になるかも知れません。
- もちろんPrologにも処理系はあります。C#で実装されたものがあるので、この講義資料の最後にリンクを書いておきます。ちなみに、C#は初心者が勉強するにはとてもよい言語です。
- Prologは知識の処理系というよりも知識をわかりやすく表現する形式だと思った方がよいでしょう。処理系としては現在は、もっと適切なもの(Pythonなど)がありますが、ルールを表現するのにPrologほど適しているものは他にはないでしょう。

講義に関するコメント(3/3)

- 事実とルールの分類は考慮する範囲によって変わるものですか？例えば「人間は哺乳類である」は現実世界ではルールですが、生物学上の生物の分類の中で考えれば事実にもなると思います。
- 推論の定義が、「事実とルールから事実を導く」とありましたが、「ルールとルールからルールを導く」のは推論にはならないってことですか。(人間は哺乳類、哺乳類は動物から、人間は動物みたいな)
- 「事実はルールの具体例」のようなルールを人工知能が知っているとすれば、人工知能はヒトと違って自身の知識の構造を正解に伝えることができるのだろうと感じます。人間が自身の知識構造をうまく説明できないのは人工知能との大きな違いだと思いました。

- それは「事実」という言葉の使い方の問題です。知識処理では、事実とそれを抽象化したルールでは扱いが異なりますので、明確に区別する必要がありますが、分野によっては、それらを区別する必要がない場合があります。しかし、具体化できる余地が残っている記述はその具体例をすべて確認した場合(例外がないことがわかっている場合)でなければ、事実と呼ぶことはできないでしょう。
- それも推論の一種です。
そもそも3段論法という手法は、厳密には、ルール同士から新しいルールを導く推論を含んでいます。
- よい視点ですね。その通り、理想的な人工知能は自分の知識の構造を理解していて、推論の過程を論理的に説明できるでしょう。ただし、機械学習を用いているほとんどのAIにはこれができないのです。

その他のコメント(1/5)

- 「機械が人間の言葉を理解するのは難しい」とおっしゃっていましたが、C言語などを機械語に翻訳できるのとは全く違うのですか。
- 多くの一般の人が、「コンピュータは自ら正しい論理を導ける」(≒思考できる)と考えている気がします。コンピュータとはここまで融通の効かないものなのか、と、学ぶまで知りませんでした。多くの人が信じる「コンピュータ万能説」みたいなものが神話だと気付く日がこないと、誰かの思惑に沿って作られた人工知能に世間がだまされて世の中が悪い方向に向かうのではないかと心配です。
- 最近は機械学習をするならpython、データを処理するならRとかいう事をよく聞くのですが、先生の場合、どんな言語やツールを使っているのでしょうか。

- プログラミング言語は曖昧さがないように設計されていますから、翻訳するのはそれほど困難ではないでしょう。しかし、人間の言語(自然言語)には一般に多くの曖昧さが含まれます。例えば、「黒い瞳のきれいな女の子を見た」には様々な解釈があり得ます。そのため、言葉から正確な意味を理解するのが困難になるのです。
- まったくその通りですね。まだまだコンピュータは使う人次第で、賢くなったりならなかったりします。コンピュータの答えを鵜呑みにする前に、もっと原理的なことを学ぶ必要があります。この講義では、みなさんが、AIの原理をよく学んで、マスコミなどに騙されないようになって欲しいです。
- 機械学習にはR、一般のプログラミングには、C#, Java, Swiftなどを使っています。最もよく使うのはJavaですが、iOS用にはSwiftを使います。Pythonも興味はありますが、まだ使ってはいません。
あと、VRやARの開発環境として、Unityを使っています。

その他のコメント（2/5）

- VRの研究をされているそうですが、具体的にどういった研究を行っているんでしょうか。ゲーム関連ですか？
- 研究室で作っているVRは、装着する装置そのものを作っているのですか？一時期スマホを使った簡易的なものも話題になっていましたが、やってみたら酔ってしまったので、もっとしっかりしたものの方がより楽しめると思いました。
- うちのPS4のVRが2台あるのですが、それを使って何か面白いことはできますか。ある一定期間をすぎると目新しさがなくなりました。
- VRが色々な場面で使われているようになり、よりリアルな映像を追究するにつれて、現実世界との区別ができなくなるような気がしました。

- 実世界の環境を丸ごと仮想化して、現実には再現が困難な実験をシミュレーションによって行う、という研究をやっています。具体的には、災害状況をシミュレーションして、仮想的に体験するというものです。もちろん、これはゲームにも使えます。
- ヘッドマウントディスプレイなどは既存のもの（HTC Vive）を使っています。今後は、ハードウェアも自作しようと思っています。HTC Viveは動き回っても正確に頭の位置をトラッキングできるのでVR酔いは少ないようです。できれば研究室見学のときにでも試してみてください。
- それは贅沢な悩みですね。VRをマルチユーザー化するのは最近のトレンドですので、ヘッドセットが2台あるともっと面白いゲームができるようになると思います。ちなみに、私たちもVRのマルチユーザー化を実現しようとしています。
- そうですね。TVがハイビジョンから4Kに進化したように、どんどん高画質化して、より高い臨場感が得られるようになるでしょう。遠くから見るだけの観光名所などは相対的に価値が減っていくかも知れません。しかし、それでもリアルな体験には高い価値があると思っています。

その他のコメント（3/5）

- 先生が使われているWiiリモコンですが、他にも色々と使い道があるのではないかと思いました。例えば、駅や空港の大きなマップで、ポインタで示した場所を様々な言語で表示するといったことも可能になるのではないでしょうか？
- Wiiリモコンの技術について、昔ソニーで似たものを発明したとのことですが、研究所での新技術が実用化されるまでにどれほど時間がかかるものなのですか。また、他に当時開発された技術で今実用化されているものはありますか。
- 先生が使っているパワーポイントのシステムのように、便利なシステムやものを作ってみたいのですが、今はうまくできそうに思えないです。勉強をすることも大切だと思いますが、新しい発想を思い付くにはどんなことをすればよいですか。

- おそらく可能になるでしょう。
例えば、プロジェクションマッピングとこのようなポインタを組み合わせて、身の回りにあるものをポイントするとそれに関する情報がポップアップ見れる、という仕組みは、すぐに実現できると思います。そういう依頼があればやってみたいと思います。
- 発明の実用化（製品化）というのは、発明者のやりたいことと、経営者の売りたいものが一致したタイミングで可能になります。私は、ソニーとIBMの研究所にいましたが、自分の作ったものを直接製品にする機会は残念ながらありませんでした。ちなみに、昔作ったもので、今、一般になっているものは、例えば、歩行者用のナビゲーションシステムやニコ動のようにネット動画にコメントを付ける仕組みや、Pepperのような音声で対話するロボットなどがあります。
- まずいろいろなものに興味を持つことです。AIでもVRでも何でもよいです。次に、作りたいものを想像してみてください。それができたら、ぜひ私に相談してください。どうやったら実際に作れるのか一緒に考えてみましょう。

その他のコメント（4/5）

- 「こいつは俺の嫁」発言がありましたが、現代でも仮想的人格に疑似恋愛するというのは、わりとあることなのかなと思いました。2次元の世界に依存して、それが中心の生活になってしまっている人は少なからず存在します。そういう人達に需要がある限り、GateBoxなどの製品の開発も進んでいくと思います。
- 「メーカーがユーザーをコントロールするのは怖い」という発言がありました。確かに、現在のスマホゲームはほとんどが時間制イベントなどにより、ユーザーのプレイ時間をコントロールしているように感じます。今後別のコントロールが発生するとしたら、どのようなコントロールが発生すると予想しますか？
- GateBoxに対するコメント返し等から、長尾先生には確固たる恋愛観があるように感じました。長尾先生が学生の頃に恋愛に対して積極的にやったことはありますか？

- 疑似恋愛自体は別に悪くないと思います。ただ、疑似恋愛だけで満足してしまうことは危険だと思っています。GateBoxを作った会社は、Lineの会社に買収され、製品の販売を始めるみたいです。Lineでコミュニケーションがほぼ完結しているような人たちは、がっつりはまってしまう可能性がありますね。
- 今後の、メーカーやサービス提供者からのコントロールはほとんどすべて消費行動に直結するものになるでしょう。つまり、広告の次のステップです。恋人にねだられて、つい買ってしまう状況を考えるとよいでしょう。それ以上に進むと、洗脳に近くなってしまい、犯罪まがいのことになってしまうので、自制が必要でしょう。
- 恋愛観というほどのことではありませんが、恥ずかしい思いをしても、傷ついたりしたとしても常に相手を気づかってコミュニケーションできるようにすべきと思っていました。
ちなみに、積極的にやったのは、バイトして車を買って、江の島とかによくドライブに行ったりしたことです。あとはギターを弾けるように練習したことですね。今思うと、かなりベタなことをしていましたね。

その他のコメント(5/5)

- 私事で申し訳ないのですが、私は最近、自分のやりたいことがはっきりしていなくて困っています。どうしたら長尾先生の様に、興味深いことや、やってみたいことが次々に浮かぶクリエイティブな人間になれるのでしょうか。
- 四年生から研究室に配属されますが、先生のところの研究室は人気があるところですか。ブラック研究室といった言葉を聞きますが、そういった研究室は情報コースにあるのでしょうか。
- 人工知能が「感情」を持つことができるのかが気になります。「感情」を持つ必要はなくても、「感情」を理解できる必要はでてくると思っています。やはり、「感情」を「知識」として表現できるかが鍵になるのでしょうか？

- 自分が面白いと思うことを中心に考えるとよいと思います。面白いと思うことが増えてくると、興味の範囲が広がります。そうすると、自分がやれそうなことにつながってくるでしょう。やりたいこととやれそうなことの接点に、自分がこれからやるべきことが見つかれば、やる気が出てくるでしょう。クリエイティブというのは、やるべきことが見つかっていて、やる気があって、いろいろ試している状態のことです。
- 研究室を運営している側の人間としては、こちらの研究室はブラックじゃないし、充実した体験ができて、スキルが身につく場所です、といいたいところです。しかし、どの教授も似たようなことを言うと思いますので、比較するのはむずかしいでしょう。後は、自分のやりたいことでやれそうなことができるような場所かどうかを判断して欲しいと思います。
- はっきりしたことはわかりませんが、人間の感情を、機械が知識として理解することは可能になると思います。ただし、人間の感情を正確に読み取れるようになるかは、よくわかりません。人間が他の人間の内面を正しく理解することは永遠にできないのと同様のことではないかと思います。ちなみに、人間の健康状態を推定することは可能になると思います。

データベースの正規化（インデキシングを有効にする手法）

データベースの正規化（1/5）

- データベースのテーブルは一般に、
同じ名前を持つ異なるレコードを扱えない、
また、一つの属性には一つの値しか入れられない
- しかし、それでは困る場合がある
  - 例えば、同姓同名の人に関するレコード
  - 例えば、「趣味」のように複数の値がある属性
- 重複した名前を持ったり、一つの属性に複数の値を持つテーブルを、非正規のテーブルという
- 非正規のテーブルを正規のテーブルに書き換えることを正規化という
- 正規化されたテーブルは、適切なインデックスが作成され、効率よく検索できる

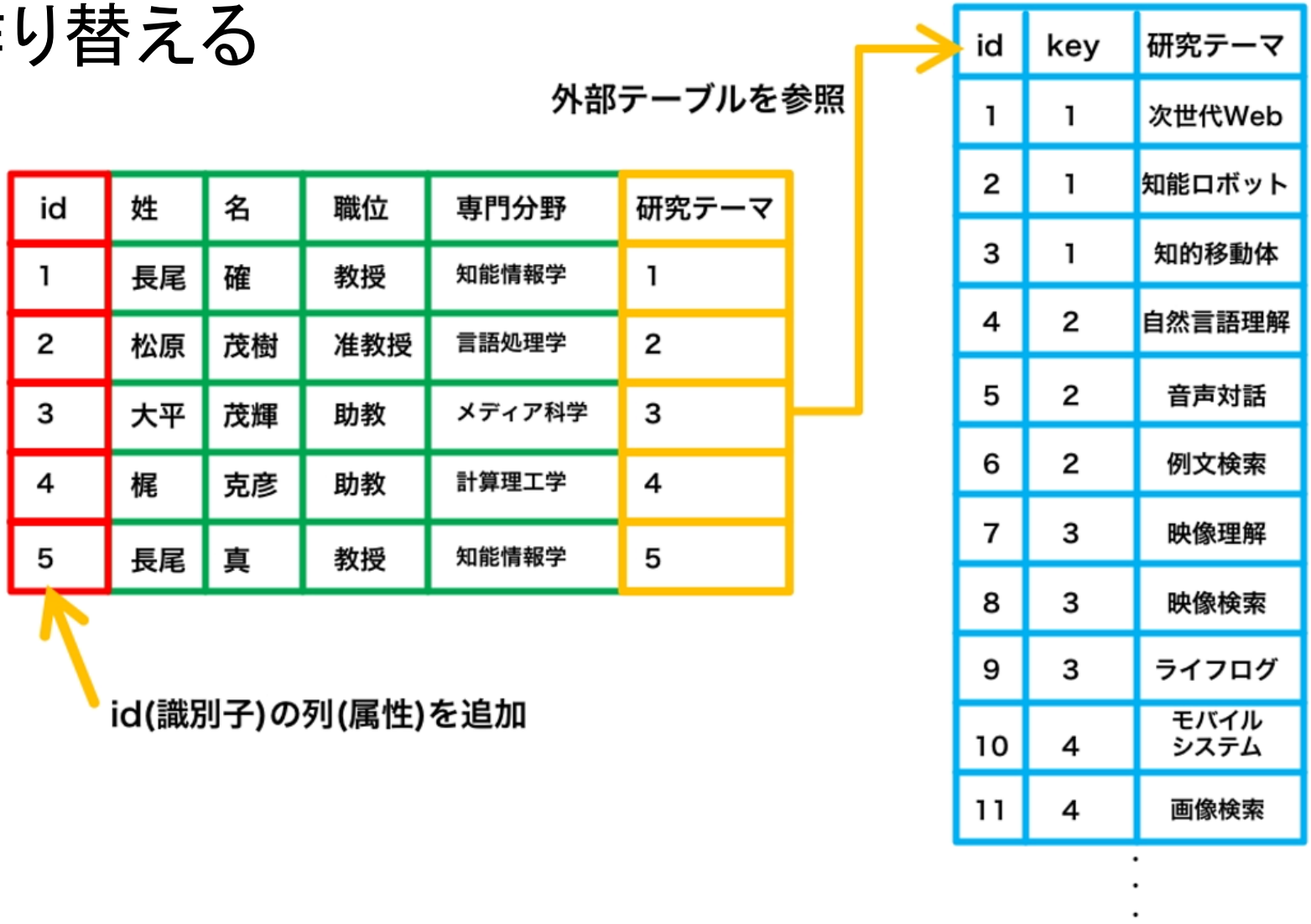
データベースの正規化（2/5）

- 非正規のテーブルの例
  - 最左の列（「姓」属性）に重複がある
    - レコードを一意に識別するための属性（主キーと呼ぶ）が設定されていない
  - 「研究テーマ」属性が複数の値を持っている
    - 研究テーマの取る値が文字列(text)ならばエラーにはならないが効率的な検索ができない

姓	名	職位	専門	研究テーマ
長尾	確	教授	知能情報学	次世代Web、知能ロボット、知的移動体
松原	茂樹	准教授	言語処理学	自然言語理解、音声対話、例文検索
大平	茂輝	助教	メディア科学	映像理解、映像検索、ライフログ
梶	克彦	助教	計算理工学	モバイルシステム、画像検索、知的UI
長尾	真	教授	知能情報学	次世代Web、機械翻訳、画像理解

データベースの正規化(3/5)

- 非正規のテーブルを変形して正規のテーブルに作り替える



データベースの正規化（4/5）

- レコードをユニーク（一意）にするSQL文
  - CREATE TABLE 研究者表 (
    id serial NOT NULL PRIMARY KEY,
    姓 text, 名 text, 専門 text,
    研究テーマ integer
    );
  - serialはレコードが追加されるたびに+1する整数（integer AUTO_INCREMENTと書く場合もある）
  - NOT NULLはその値が必ず入ることを意味する
  - PRIMARY KEYはidが主キーであることを示す

データベースの正規化（5/5）

- 他のテーブルを参照可能にするSQL文
  - 「研究テーマ表」というテーブルが別にある（ただし同じデータベース内）とする
  - CREATE TABLE 研究者表 (
    id serial NOT NULL PRIMARY KEY,
    姓 text, 名 text, 専門 text,
    研究テーマ integer,
    CONSTRAINT research_theme_fkey
    FOREIGN KEY (研究テーマ)
    REFERENCES 研究テーマ表 (key)
    );
  - CONSTRAINTはそのテーブルが持つ制約を表す
    - research_theme_fkeyは制約に付けられた名前
  - FOREIGN KEY (研究テーマ)は「研究テーマ」属性が外部テーブルを参照することを示す
    - REFERENCES 研究テーマ表 (key)は「研究テーマ表」テーブルのkeyという属性を参照することを示す
    - つまり、「研究者表」の「研究テーマ」属性と「研究テーマ表」の「key」属性は同じ値なら同一のものを指し示すことになる

その他のSQL文

数値を含むテーブルの扱い(1/5)

- エクセルのような表計算ソフトと同様に
  データベースでは、数値を含むテーブルを
  使って、合計、平均、最大値と最小値などが
  計算できる
  - データベースの方が巨大な表を管理できる
- また、数値の大小(または文字列の辞書順)
  に従ってレコードをソート(並び替え)できる

数値を含むテーブルの扱い(2/5)

ID	製品名	SSD	価格
A1	iPod touch	8	20,900
A2	iPod touch	32	27,800
A3	iPod touch	64	36,800
A4	iPad Wi-Fi	16	44,800
A5	iPad Wi-Fi	32	52,800
A6	iPad Wi-Fi	64	60,800
A7	MacBook Air	64	88,800
A8	MacBook Air	128	108,800

- 製品名でグループに分けて価格の合計を出す
  - SELECT 製品名, SUM(価格) FROM 製品表 GROUP BY 製品名;
- 製品ごとの価格の平均を出す
  - SELECT 製品名, AVG(価格) FROM 製品表 GROUP BY 製品名;

数値を含むテーブルの扱い(3/5)

- 同じ製品名の価格の合計
- SELECT 製品名, SUM(価格)
  FROM 製品表
  GROUP BY 製品名;

製品名	sum(価格)
iPod touch	85,500
iPad Wi-Fi	158,400
MacBook Air	197,600

- 同じ製品名の価格の平均
- SELECT 製品名, AVG(価格)
  FROM 製品表
  GROUP BY 製品名;

製品名	avg(価格)
iPod touch	28,500
iPad Wi-Fi	52,800
MacBook Air	98,800

数値を含むテーブルの扱い(4/5)

ID	製品名	SSD	価格
A1	iPod touch	8	20,900
A2	iPod touch	32	27,800
A3	iPod touch	64	36,800
A4	iPad Wi-Fi	16	44,800
A5	iPad Wi-Fi	32	52,800
A6	iPad Wi-Fi	64	60,800
A7	MacBook Air	64	88,800
A8	MacBook Air	128	108,800

- 製品ごとの価格の最小値と最大値を出す
  - SELECT 製品名, MIN(価格), MAX(価格) FROM 製品表 GROUP BY 製品名;
- 製品名をアルファベット順に並べて、製品ごとに価格の高い順に並べる
  - SELECT * FROM 製品表 ORDER BY 製品名, 価格 DESC;

数値を含むテーブルの扱い(5/5)

- 同じ製品名の価格の最小値と最大値
- SELECT 製品名, MIN(価格), MAX(価格) FROM 製品表 GROUP BY 製品名;

製品名	min(価格)	max(価格)
iPod touch	20,900	36,800
iPad Wi-Fi	44,800	60,800
MacBook Air	88,800	108,800

- 製品名(昇順)と価格(降順)で並べて表示
- SELECT * FROM 製品表 ORDER BY 製品名, 価格 DESC;

ID	製品名	SSD	価格
A6	iPad Wi-Fi	64	60,800
A5	iPad Wi-Fi	32	52,800
A4	iPad Wi-Fi	16	44,800
A3	iPod touch	64	36,800
A2	iPod touch	32	27,800
A1	iPod touch	8	20,900
A8	MacBook Air	128	108,800
A7	MacBook Air	64	88,800

まとめ

- データベース
  - 大量の知識（特に事実）を効率良く格納・管理するための仕組み
  - Prologの事実の記述形式に比べると制約が大きい
  - 正規化されたテーブル形式で表す
  - テーブルは、レコード（行）と属性（列）から成る
- SQL
  - テーブル形式のデータベース（関係データベース）から必要なものを見つけるための問い合わせ（クエリ）言語
  - 属性の値に関する条件を書くことができる
    - 条件には、文字列の完全一致あるいは部分一致、数値の比較などが書ける
  - 数値データの合計、平均、最小値・最大値の計算および値の大小による並べ替えができる
  - 複数のテーブルの利用や結合については次回

講義資料はダウンロードできます

- URLは、以下の通り。
  - http://www.nagao.nuie.nagoya-u.ac.jp/syllabus/knowledge_proc.html
- メールでの質問も受け付けます。
  研究室に遊びに来たい人は連絡ください。
  今月末に見学会を企画しています。
  現在のところ3人からメールをもらっています。
  - nagao@nuie.nagoya-u.ac.jp

今日の問題

- 下のテーブル「友人帳」を正規化したテーブルを書いてください。
- 必要に応じて外部テーブルを作成してください。

姓	名	性別	職業	趣味
A	B	男性	教師	読書、音楽、映画
C	D	女性	医師	読書、スポーツ
E	F	男性	弁護士	映画、スポーツ
A	F	女性	エンジニア	音楽、ダンス

参考

- C#で実装されたProlog処理系
  - https://sourceforge.net/projects/cs-prolog/
- C#のプログラミング環境（Visual Studio Community）
  - https://www.microsoft.com/ja-jp/dev/products/community.aspx