

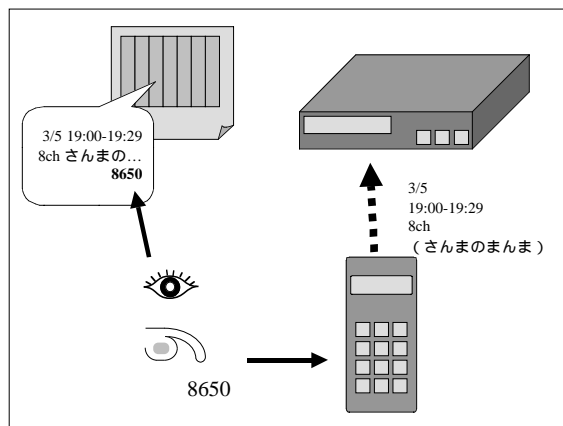
情報処理の概念

#1 一つのシステムを理解する - Gコード を例に / 2002 (春)

一般教育研究センター 安田豊

Gコード

- ビデオ予約のための数字列
 - ビデオ予約に必要な情報の入力を単純にするために開発された
 - 開発元は GEMSTAR 社 (<http://www.gemstar.co.jp/>)
 - 1988年開発、1990製品化、1992年朝日新聞テレビ欄に掲載
- 短い桁数(1-8桁) で全ての番組が予約可能
 - 8650 : 3/5 19:00-19:29 8ch さんまのまんま
 - 582 : 3/5 19:00-19:30 2ch ニュース7
 - 2060766 : 3/5 21:54-23:10 6ch ニュースステーション



どこに技術(工夫)があるか？

- 入力数字列を短くする
 - 4月10日18:00-18:30 8ch というような情報を入力するには、例えば 04101800183008 の14桁数字を入力すれば良いはず
 - 利用者を録画予約間違いから解放するという目的にはほど遠い (8650で「明日のさんまのまんま」は実に簡単)
- 考え方
 - ストレートにやったのでは決して短くならない
 - そこにインテリジェント(知的)な処理が入る
 - ここがコンピュータの「使いどころ」となる

短いコードを作る

- 全ての番組を短く表現することはできない
 - vs. 全ての番組について表現できるように
 - 情報量を減らしたくない
 - 情報量: 対象がとり得る状態(など)の数と考えればよい
 - そもそも全ての番組を表現するのに必要な情報量とは?
 - あり得る番組情報の数(種類)は幾らか?
- 情報量は失わず、データ量を減らしたい
 - 一般にデータの圧縮と呼ばれる作業

あり得る番組情報の数(種類)

- 条件設定
 - 何日先まで? (30日と仮定)
 - 何時何分開始? (24時間で分単位を前提)
 - 何ch? (10chと仮定)
 - 何分の番組か? (最大180分と仮定)
- $30日 * 24時間 * 60分 * 10ch * 180分 = 77760000$
 - 全ての番組を表現するのに必要な情報量とは(10進数表現で)8桁の単位である
 - Gコードの最大桁数(8)と同じ範囲で収まった
 - つまりGコードは情報量を落としてはいけない(落とす必要はない)
- 04101800183008 の14桁よりは少なくなったが、まだまだ長い
 - どうすれば短く(圧縮)できるか?

短いコードの作り方

- 「よくある番組に短いコードを振る」方法はないか？
 - 短く書ける数字をありがちな番組に割り振ればよい
 - 0から77760000までの数字の前の方は充分短く書ける
 - 多くの人が予約する番組について簡単に(桁数が短く)なれば、滅多に予約されない番組は多少長くなっても良い
 - 発想の逆転
 - では典型的でメジャーな番組時間帯ほど上に、マイナーでイレギュラーな時間帯のものほど下へと並べ直して、上から順番に番号を「割り当てて」いけば？
 - 0を4/1の19:00-20:00 2chに、1を同4chに、2を同6chに、、、
 - 30日、10ch分で19:00,20:00,21:00発の1時間番組を「押さえる」には $30 \times 10 \times 3 = 900$ つまり3桁で足りる。
 - 4桁なら楽々と「2時間枠」「30分枠」を押さえられる。
 - この方式をYコードと呼ぼう

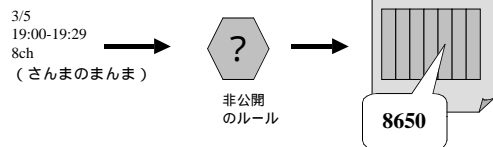
Gコードの実際

- 企業秘密になっているので詳細は不明
 - 「フシギ」から「なるほど」へのヒント
 - 8桁とはストレートに全番組を表現するに足りる幅だ。
 - これを前提に、よくある番組に短い(前方がゼロの)数列を与え、滅多に録画しない番組に省略できない数列を与えればよい。
 - 秘密を保持するために暗号化されている
- 各自で効率の良い方式を考えてみよう
 - いずれにしても有限の状態しかもたないものをコード化するのだから、様々な工夫の余地があるということ

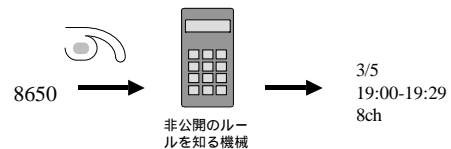
情報処理としてのGコード

- コード化：エンコードと呼ばれる
 - コード：多数ある「もの」を区別するために振った記号
 - 古典的な情報処理のひとつ
 - 「もの」に対応する情報を設定し、情報を相手に何らかの処理をすることで、ものに対する作業を助ける
 - 例：銀行の預金管理、書籍の発注処理
- コードの復元：デコードと呼ばれる
 - コードからもとの情報を取り出す
- Gコード：番組をコードで示したもの
 - 番組の情報をコード化する
 - つまり「いつから、どのチャンネルで、いつまで」を示すコードを作る

番組情報のGコード化（エンコード）



Gコードの番組情報化（復元、デコード）



ビジネスとしてのGコード

- 便利さを売る
 - エンコードするシステムもしくはサービスをテレビ局や出版社に提供
 - デコードするシステム（暗号処理チップ）をビデオのリモコンに内蔵させて売る
- 利益の源泉はコード化技術の提供
 - 機械の販売による利益は本質的ではない
 - 独占のためにコード化技術を非公開に
 - ブラックボックス（チップ）を作り、機器に含めてもらう
 - 解析されないよう暗号化も行う
- 企業秘密で利益の源泉となる技術を守る
 - 特許（公開が前提）で守るという方法もある

ビジネスとしてのGコード

- 非公開であることの問題
 - 価格が下がらない（かもしれない）
 - 技術革新が遅れる（より便利な何かが提案されにくくなる、かもしれない）
 - しかし技術非公開や暗号化によって利益の源泉を保護しているものは多い
 - DVDのCSS (Contents Scrambling System)
 - Adobe Acrobat の保護暗号解読
- Gコード暗号解析を行う人たちが現れる
 - それほど難しくないとしても、許されるのか？

知的財産・著作権

- DVDのCSS (Contents Scrambling System)
 - 暗号鍵を持つ機器でしか再生(デコード)できない
 - 1999.10 : DVD暗号をデコードするプログラムが公開
 - 1999.12 : 米国で提訴
 - 2000.1 : ノルウェーの15歳の少年が自宅捜索
 - 2000.8 : 連邦裁判所から有罪判決
 - 2001.2 : 連邦控訴裁で再開
- Adobe Acrobat の保護暗号解読
 - 電子書籍向けの暗号を解読するソフトを開発
 - 2001.7 : ロシア人プログラマーがラスベガスで逮捕
 - 2001.7 : Adobe 社が釈放要求 (合衆国の刑事訴訟であるためAdobeは直接関係ない)

犯罪か、権利か

- 表現の自由
 - プログラマは自由なコードを書けるべきだ
- DMCA (デジタルミレニアム著作権法)
 - 設けられた制約を迂回することを禁じる
- デジタル化の波
 - 数多くの知的財産ビジネスがデジタル化による影響を受けている
- 過去に学ぶ
 - ベータマックス訴訟 (デジタル化とは関係ないが、新しい技術が知的財産ビジネスに与える影響に直結した事件)

犯罪か、権利か

- USにおけるSONYのベータマックス訴訟
 - <http://www.sony.co.jp/Fun/SH/2-20/h5.html>
 - 裁判 (1976年)
 - 映画業界の売り上げが下がる
 - 原告 : ユニバーサルスタジオ、ディズニー
 - 被告 : ソニー本社、ソニー・アメリカ
 - 経過
 - 1979年10月 ソニー 全面勝訴
 - 1981年 米国連邦高等裁判所では敗訴、連邦最高裁判所へ
 - 1984年1月 ソニー 勝訴

ベータマックス訴訟から学ぶもの

- 視点
 - 法律は作りながら遵守するもの
 - 新しい技術に社会が対応する必要性
 - 目的(文化の振興->社会利益、幸福)を忘れずに
- デジタル化による影響
 - DVD暗号解読、Acrobat暗号解読や、Napstarなどによる複製データ提供サービスが、既存のビジネスや法律などの社会システム全体に大きな影響を与える

あるシステムを理解する

- Gコードを理解するということ
 - コード化作業 (狭義の情報処理)
 - データ圧縮 (数学的概念)
 - 暗号処理チップの埋め込み (技術的知識)
 - 技術非公開と公開、特許という選択肢 (ビジネス)
 - 著作権、知的財産ビジネスと表現の自由 (法律)
 - デジタル化による社会システムの変化
- (小さなシステムにすら) 現実世界と極めて多くの接点がある
 - 全体像を理解する必要性
 - 周囲の多くの事物との相対的な位置関係を把握する

情報処理の概念

- 現代の情報処理システムを理解する
 - 要素技術の理解では不十分
 - 全体を把握する
 - 技術だけではなく、必要性や社会的問題も含めて
 - しかし細部の技術的理解を抜きにしては見誤る
 - 高い視点とバランス感覚を求められる
- 全体を把握する意味を考える
 - そこから浮かび上がる情報処理のすがたを捉える
 - Gコードにおいては
 - エンコード・デコードに関わる暗号化と圧縮が内部で行われている情報処理
 - しかしそれだけを理解しても足りない
- 講義を通じてさまざまな事例を見ていきましょう

