アルゴリズムとデータ構造

第25週目

担当　情報システム部門　徳光政弘
2024年12月16日

参考資料

今日の内容

- 文字列の照合アルゴリズム　ちからまかせ法
- 文字列の照合アルゴリズム　KMP法
- 文字列の照合アルゴリズム　ボイヤー・ムーア法（次回）

用語の定義

- テキスト　検索対象の文字列
- パターン　テキストから検索したい文字列
- 文字列の照合　テキストからパターンに一致する文字列を探し出し、照合する部分の先頭の添字を返す

パターンはテキストのどこに含まれているか？

ちからまかせ法

- 先頭から順番に文字列を照合する

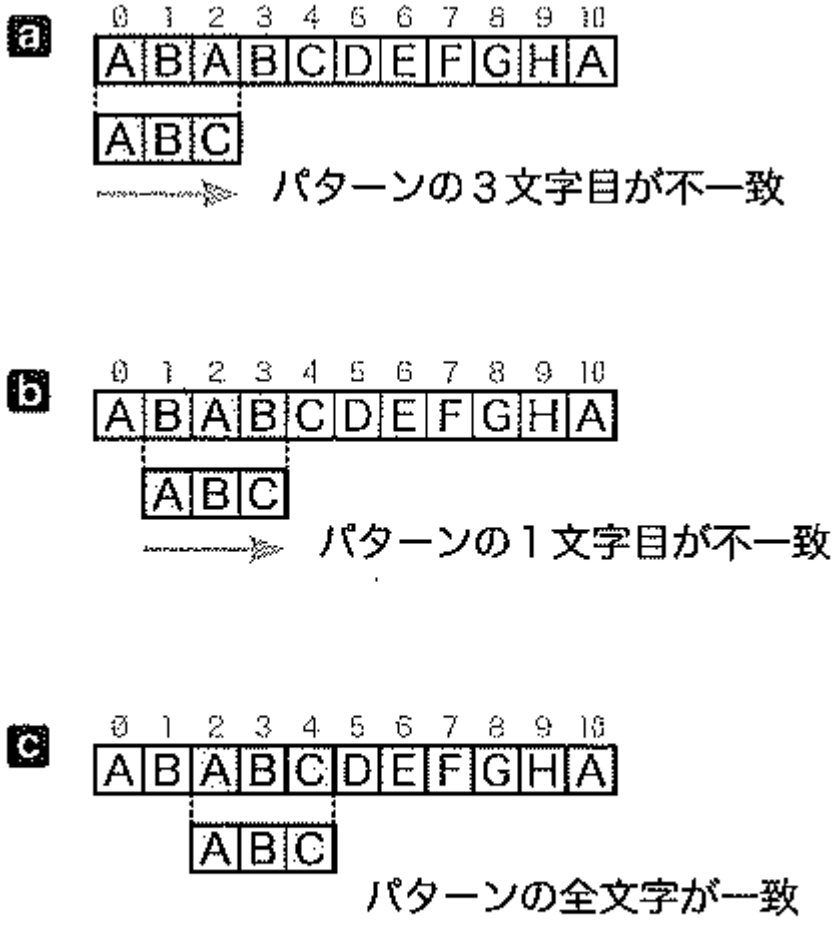


Fig.7-10　力まかせ法の概略

ちからまかせ法の問題点

- 照合済みの部分も再度少作業をするため、無駄が発生する

途中で不一致

AAACABCABED

AAABABCABED

再度照合する

AAACABCABED

AAABABCABED

KMP法

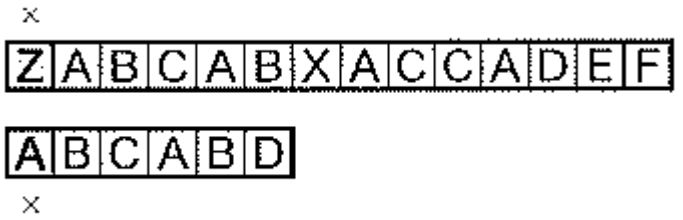
- Knuth–Morris–Prattアルゴリズム（同時期に3名が考案した）
- すでに照合済みの部分のデータを活用する
- パターン文字列に同じパターンが含まれていると有利
  - 例　ABCABDE

KMP法　有利な例

不一致

テキスト

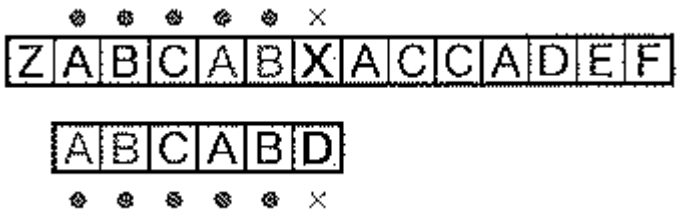
パターン



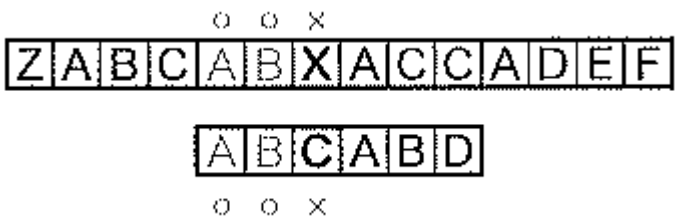
一致

テキスト

パターン

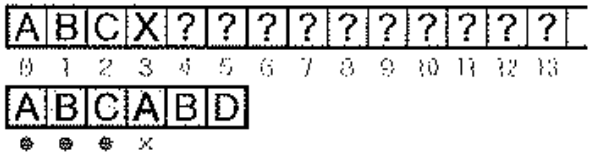


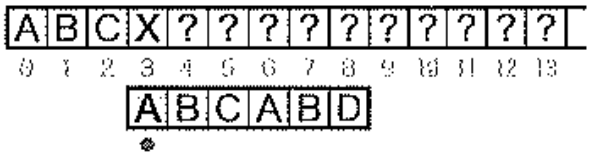
次のABまで開始位置をスキップできる



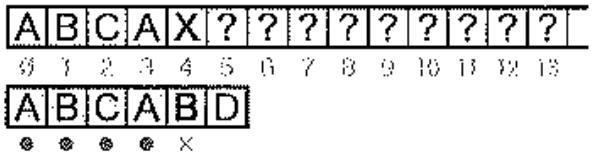
KMP法　読み飛ばし

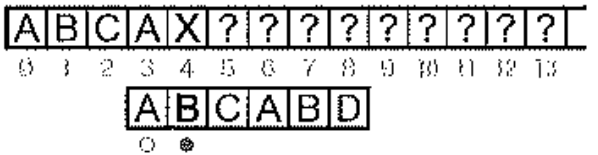
d ４文字目で不一致



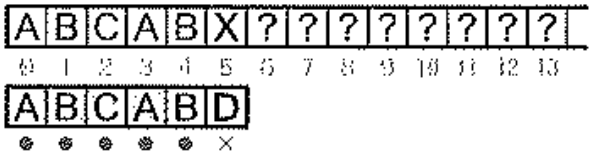


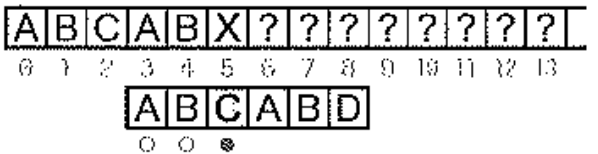
e ５文字目で不一致





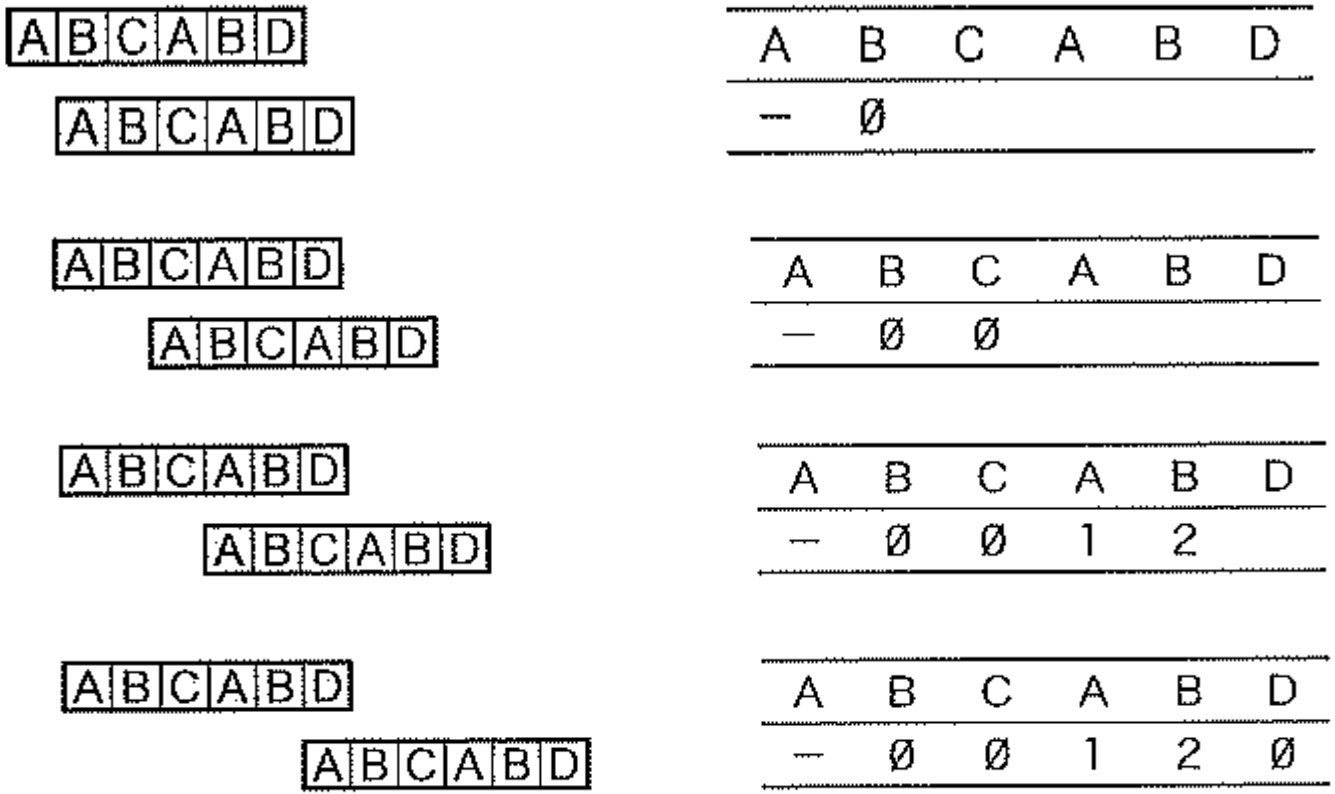
f ６文字目で不一致





部分マッチテーブル

事前に読み飛ばし用の表を作成する、照合済みの部分はスキップできる。



A	B	C	A	B	D
—	Ø				

A	B	C	A	B	D
—	Ø	Ø			

A	B	C	A	B	D
—	Ø	Ø	1	2	

A	B	C	A	B	D
—	Ø	Ø	1	2	Ø

ABCAまで合致していたら、Aを飛ばしてBから照合できる

実装の例

List 7-13 chap07/kmp_match.c

```
/*--- KMP法による文字列探索 ---*/
int kmp_match(const char txt[], const char pat[])
{
    int pt = 1;          // txtをなぞるカーソル
    int pp = 0;          // patをなぞるカーソル
    int skip[1024];      // スキップテーブル

    skip[pt] = 0;
    while (pat[pt] != '\0') {
        if (pat[pt] == pat[pp])
            skip[++pt] = ++pp;
        else if (pp == 0)
            skip[++pt] = pp;
        else
            pp = skip[pp];
    }

    pt = pp = 0;
    while (txt[pt] != '\0' && pat[pp] != '\0') {
        if (txt[pt] == pat[pp]) {
            pt++; pp++;
        } else if (pp == 0)
            pt++;
        else
            pp = skip[pp];
    }

    if (pat[pp] == '\0')
        return pt - pp;

    return -1;
}
```

1 表の作成

2 探索

patの ‘\0’の比較まで到達したら、パターンにマッチしている。
前置増分の評価順番に注意する。