九州産業大学大学院

KYUSHU SANGYO UNIVERSITY GRADUATE SCHOOL

令和2年度　研究成果発表会

正規表現同一判定の効率化及びその応用

博士前期課程

情報科学研究科　情報科学専攻　社会情報システム分野

吴 晟董

主査　成凱
副査　朝廣雄一
　　稲永健太郎

研究背景

- 正規表現は、小さなプログラミング言語と呼んでもよいくらいの汎用的なパターン記法によって、テキストの記述と解析を実現できる
- 正規表現は、プログラミングにおけるパターンチェックや文字列検索などで必要不可欠
- 正規表現は様々な分野で広く応用されている
  - 機械学習におけるデータの前処理
  - テキスト編集、システム管理における文字列検索・置き換え

正規表現の学習

- 正規表現は、自由度が高く、一定の訓練を受けないと使いこなすのが難しい
- 正規表現学習のためには、特定のパターンが正しく表現できたかをチェックする機能が必要
- しかし、同じ文字列のパターンは複数の正規表現で表すことができるので、正規表現の同一判定が必要

先行研究

- 有限オートマトンによる同一判定[1][2]
  - 正規表現、有限オートマトン、最小DFAの同等性及び最小DFAの唯一性に基づき、正規表現の同一判定を行う

- 正規表現の代数的性質を利用した同一判定[3][4]
  - Rewrite公理に基づき、正規表現の同一判定を行う
  - 符号列Σの大きさに依存するため、計算コスト上では上記の有限オートマトンによる同一判定法とは変わらない

問題提起

- 既存手法は効率が悪いと指摘されている
  - 一番良いアルゴリズムの最悪の計算量が$O(kn\log n)$
    $k = |\Sigma|$:記号列（アルファベット）の大きさ
    n：有限オートマトンの状態数

- 状態数が多い有限オートマトンの場合は非効率的
  - Rewrite公理に基づき、正規表現の同一判定を行う
    符号列Σの大きさに依存するため、計算コスト上では上記の有限オートマトンによる同一判定 法とは変わらない

本研究の提案

1. 正規表現の特徴量を用いた同一判定を提案
   特徴量は正規表現の文字通りの長さに関係し、オートマトンと関係しないため、実質定数時間$O(1)$

2. マッチングテストによる同一判定
   正規表現と正規言語の関係から、集合の同一性を利用して、判定を行う

3. 正規表現学習支援ンシステムを開発し、同一判定を応用

正規表現の特徴量

記号	説明
$\|\alpha\|_{min}$	正規表現αにマッチする文字列の長さの最小値
$\|\alpha\|_{max}$	正規表現αにマッチする文字列の長さの最大値
$\#(\alpha)$	正規表現αの言語の要素数, $\#(\alpha)=\|L(\alpha)\|$

正規表現の同一性に関する定理

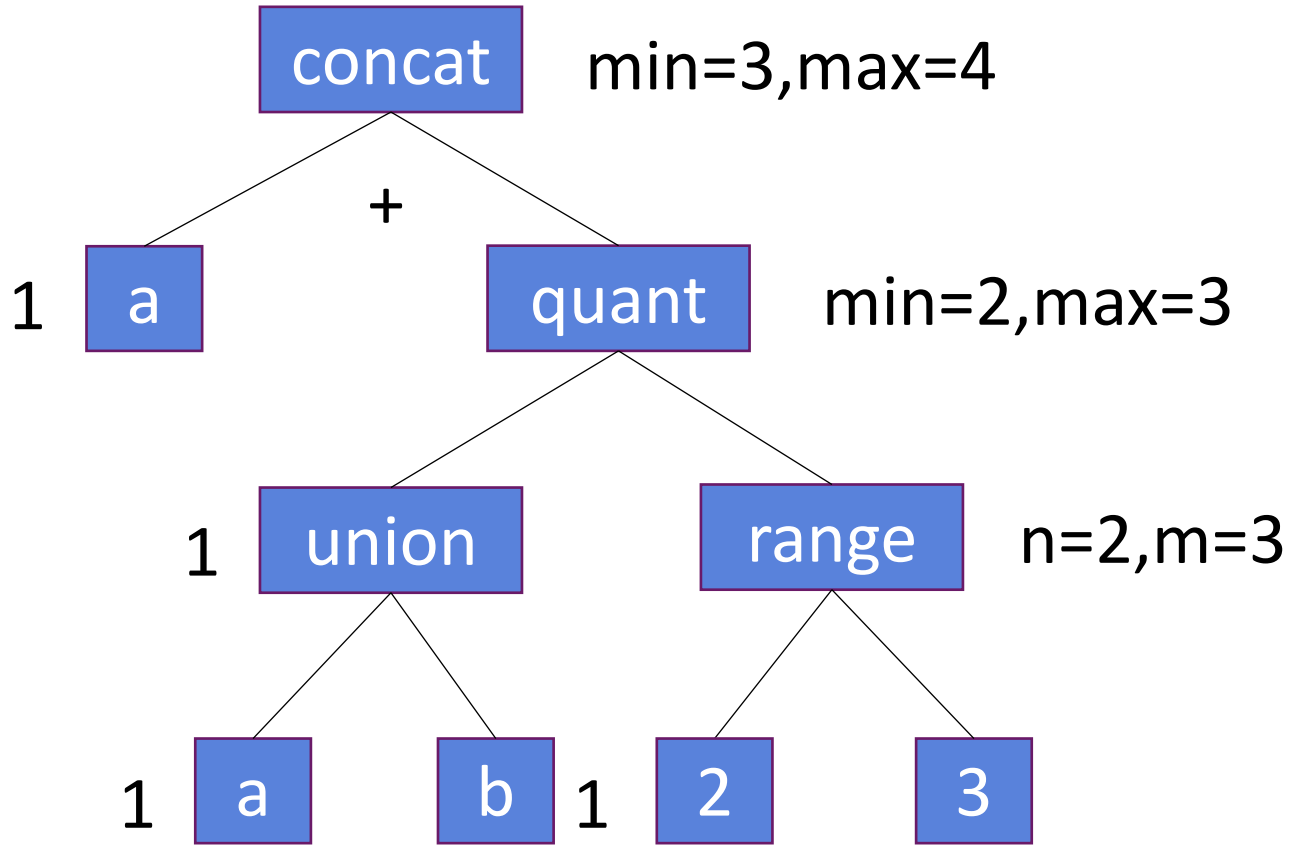
- 定理：同じΣ上の二つの正規表現αとβが同一であれば、下記の等式が成立する。

① $|\alpha|_{min} = |\beta|_{min}$
② $|\alpha|_{max} = |\beta|_{max}$
③ $\#(\alpha) = \#(\beta)$
④ 任意の$x \in L(\alpha)$であるとき、$x \in L(\beta)$ 、
逆に任意の$x \in L(\beta)$であるとき、$x \in L(\alpha)$

文字列の長さの計算方法

- anyof([…])の計算
  $|[\ldots]|_{min} = |[\ldots]|_{max} = 1$
- union(|)の計算
  $|\alpha \mid \beta\,|_{min} = min(|\alpha|_{min}, |\beta|_{min})$
  $|\alpha \mid \beta|_{max} = max(|\alpha|_{max}, |\beta|_{max})$
- concat(連結)の計算
  $|\alpha\beta|_{min} = |\alpha|_{min} + |\beta|_{min}$
  $|\alpha\beta|_{max} = |\alpha|_{max} + |\beta|_{max}$
- quant(桁数)の計算
  $|\alpha\{n,m\}|_{min} = n * |\alpha|_{min}$
  $|\alpha\{n,m\}|_{max} = m * |\alpha|_{max}$
- star(*)の計算
  $|(\alpha) * |_{min} = 0$
  $|(\alpha) * |_{max} = \infty$
- plus(+)の計算
  $|(\alpha) + |_{min} = |\alpha|_{min}$
  $|(\alpha) + |_{max} = \infty$

構文木における長さの計算例



$|a(a|b)\{2,3\}|_{min} = 3$

$|a(a|b)\{2,3\}|_{max} = 4$

言語の要素数の計算方法

- anyof([…])の計算

  $\#([\ldots]) = |[\ldots]|$
- union(|)の計算

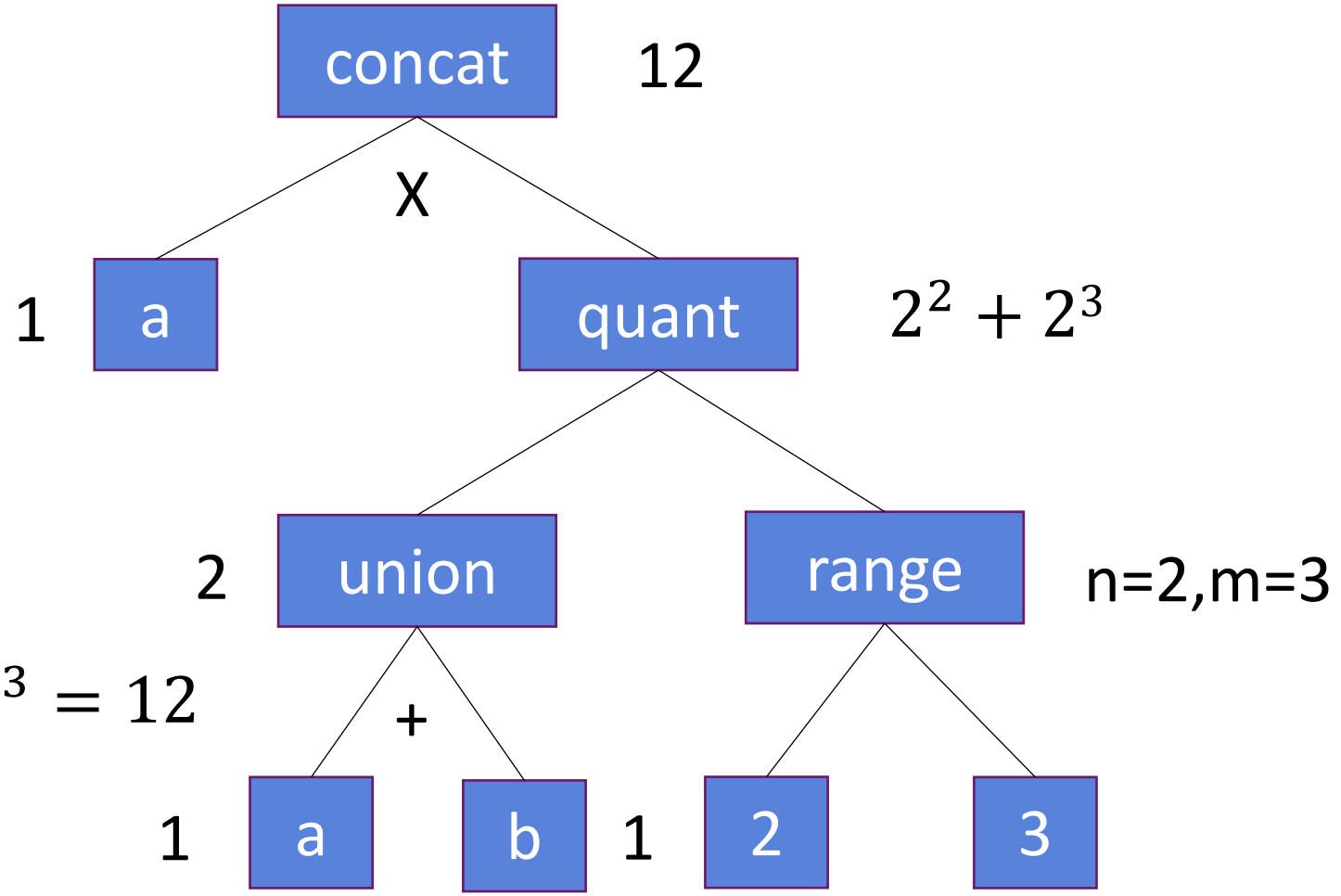
  $\#(\alpha|\beta) = \#(\alpha) + \#(\beta)$
- concat(連結)の計算

  $\#(\alpha\beta) = \#(\alpha)\times\#(\beta)$
- quant(桁数)の計算

  $\#(\alpha\{n\}) = \#(\alpha)^n$

  $\#(\alpha\{n,m\}) = \#(\alpha)^n + \#(\alpha)^{n+1} + \cdots + \#(\alpha)^m$

構文木における要素数の計算例



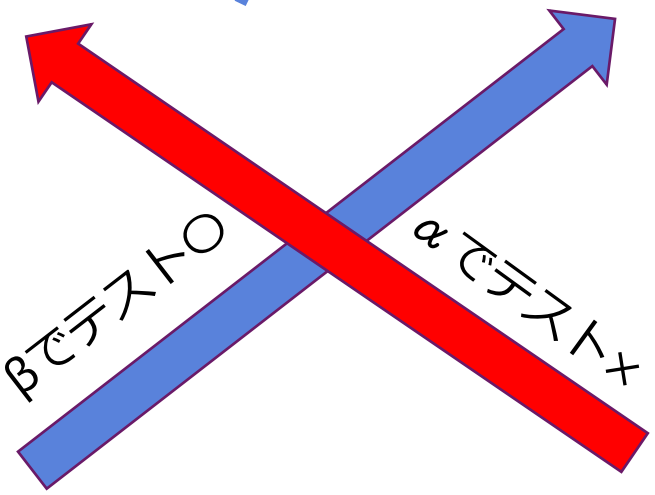
$$\#(a(a|b)\{2,3\}) = 2^2 + 2^3 = 12$$

マッチング・テストによる同一判定の例

α

β

0(8|9)0-¥d{4}-¥d{4} ≠ 0[8,9][0,2]-[0-9]{4}-[0-9]{4}



αの文字列

090-4534-3453

080-4563-5847

βの文字列

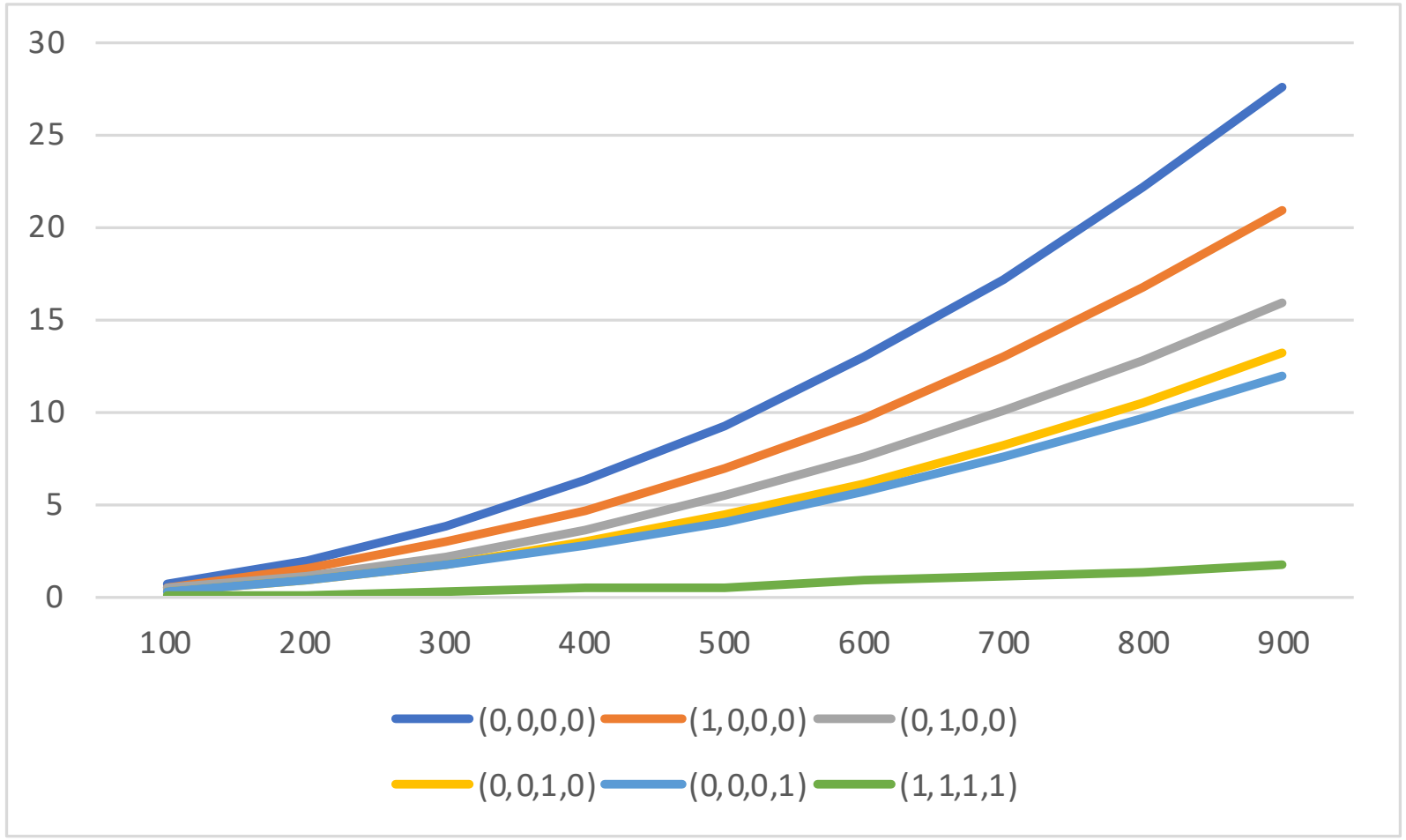
092-4587-3487

090 2387 4587

実験方法

- 提案方法は二つの正規表現が同一である場合は判定できないので、最後に必ず判定できるオートマトンによる同一判定を実行
- 最初はオートマトンによる同一判定のみ実行し、時間を測る
- 次はオートマトンによる同一判定を実行する前に提案の方法を入れて、時間を測り、時間が節約できて効率化が実現できるかどうかを評価

実験結果


30
25
20
15
10
5
0
100
200
300
400
500
600
700
800
900
(0,0,0,0)
(1,0,0,0)
(0,1,0,0)
(0,0,1,0)
(0,0,0,1)
(1,1,1,1)

まとめと今後の課題

- 正規表現の特徴量、マッチングテストによる同一判定を提案した
- 評価実験を行い、提案の効率化手法は有効であると確認できた
- 今後の課題として
  - 正規表現の種類、正規表現と正規表現セットの数を増やして実験を行う必要がある
  - 変形による同一判定を取り入れ、さらに効率をあげることが期待できる

指導教員コメント

本研究は、計算理論の基礎となる正規表現・オートマトンに関する課題を取り上げ、同一判定の効率化を図るものである。研究の遂行に当たって、言語理論、計算モデル論等の情報科学の基礎が必須となり、また、論理的思考力、高いプログラミング能力及びシステム開発力が求めらる。よく頑張った。

成凱