Regular Expressions
- **Example:** 1-\d+ (US Phones)
- **Wide range of applications:** text and document stores, bioinformatics, data mining, etc.

Existing Techniques
- Full Data Scans (NFA, DFA)
  - Do not scale with data size
  - E.g., MongoDB, Oracle, MySQL, etc
- m-gram Indexes with Partial Data Scans
  - Index token of length m for multiple or all values of m
  - Avoid scans for indexed tokens.
  - Partial scans for tokens not indexed.
  - Suffer from large memory footprint
    - E.g., ElasticSearch
- Succinct (Search on compressed data)
  - Memory-efficient search for arbitrary length tokens
  - Asymptotic search complexity similar to m-gram indexes
  - Can be used as a black box

Black-box RegEx
- Represent RegEx as an RTree
  - Leaves are tokens; Interior nodes are operators
- Search for leaves
- Traverse the RTree bottom-up, combining intermediate results at each node

Output cardinality for Concat operator can be arbitrarily smaller than cardinality of intermediate results
- Unnecessary operations if the gap is large

Complexity of Union, Repeat, and Wildcard operators depend on output cardinality
- Output cardinality smaller up the RTree

**Main Idea:** Transform the RTree for the RegEx such that:
- the black-box approach can be avoided for the Concat operator altogether
- Union, Repeat and Wildcard operators are pushed up the RTree

### Pull-Up Union Transformation:

\[
\text{Pull-Up Union Transformation:} \\
\text{Original: } (R_1|R_2)|R_3 \\
\text{Transformed: } (R_1) \cup (R_2) | R_3
\]

### Pull-Up Wildcard Transformation:

\[
\text{Pull-Up Wildcard Transformation:} \\
\text{Original: } (R_1(\text{RE}_1 \cdot \text{RE}_2)) \\
\text{Transformed: } (R_1(\text{RE}_1) \cdot \text{RE}_2)
\]

### Pull-Out Repeat Transformation:

Replace Repeat operator by Unions of Concat:

\[
\text{RE}^+ = (\text{RE}^1|\text{RE}^2|\text{RE}^3|...|\text{RE}^k)
\]

- k: smallest integer for which \( \text{RE}^{k+1} \) has 0 occurrences.
- Use heuristic to upper bound value of k
- k can be large when RegEx contains character classes
- Use partial scans beyond threshold

### Pull-Out Concat Transformation:

- Find Concat nodes whose children are tokens (T_1, T_2)
- Replace with new token T_1T_2

Transformations incorporated within Succinct data structures.

**Results**

<table>
<thead>
<tr>
<th>Query ID</th>
<th>Query</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>W1</td>
<td>&lt;script&gt; &lt;/script&gt;</td>
<td>HTML Script</td>
</tr>
<tr>
<td>W2</td>
<td>Motorola.<em>([MPC][0-9]+)+([0-9]+)</em></td>
<td>Motorola PowerPC Chip#</td>
</tr>
<tr>
<td>W3</td>
<td>[A-Z]+Clinton</td>
<td>William A-Z+ Clinton</td>
</tr>
<tr>
<td>W4</td>
<td>(a-zA-Z_ _ _)<em>(a-zA-Z_+ _)</em></td>
<td>Stanford Domain URLs</td>
</tr>
<tr>
<td>W5</td>
<td>(DE</td>
<td>RSW</td>
</tr>
<tr>
<td>P1</td>
<td>DE</td>
<td>RSW</td>
</tr>
<tr>
<td>P2</td>
<td>[AC]GLPV</td>
<td>CLUSTERED_1</td>
</tr>
<tr>
<td>P3</td>
<td>CRXCLXTC</td>
<td>MYELIN_PIP_1</td>
</tr>
<tr>
<td>P4</td>
<td>GMV</td>
<td>ALFGCGGH</td>
</tr>
<tr>
<td>P5</td>
<td>[PY][FS][RNA][LV][R][EQ][L][NHAT]</td>
<td>Succinct2</td>
</tr>
</tbody>
</table>

Succinct has 8x smaller storage footprint than uncompressed data structures.