An Adaptive Hybrid Pattern-Matching Algorithm on Indeterminate Strings

W. F. Smyth\(^1\), Shu Wang\(^1\) and Mao Yu\(^1\)

\(^1\)Algorithms Research Group
Department of Computing & Software
McMaster University, Canada
email: smyth,shuw@mcmaster.ca

The Prague Stringology Conference 2008
Outline

1 Introduction
2 Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3 Special Properties of Indeterminate Borders
4 The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5 Experimental Results
6 Conclusion
Outline

1 Introduction
2 Fundamental Algorithms
   • The Knuth-Morris-Pratt Algorithm
   • The Sunday Adaption of Boyer-Moore Algorithm
   • The Shift-And Algorithm
   • The Franek-Jennings-Smyth Algorithm
3 Special Properties of Indeterminate Borders
4 The New Hybrid Algorithm
   • Outline of the New Algorithm
   • Shift-And Matching
   • Sunday-Shift
   • Examples
5 Experimental Results
6 Conclusion

Smyth, Wang, Yu
Adaptive Indeterminate Pattern-Matching Algorithm
Outline

1. Introduction
2. Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3. Special Properties of Indeterminate Borders
4. The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5. Experimental Results
6. Conclusion

Smyth, Wang, Yu
Adaptive Indeterminate Pattern-Matching Algorithm
Introduction

Fundamental Algorithms
- The Knuth-Morris-Pratt Algorithm
- The Sunday Adaption of Boyer-Moore Algorithm
- The Shift-And Algorithm
- The Franek-Jennings-Smyth Algorithm

Special Properties of Indeterminate Borders

The New Hybrid Algorithm
- Outline of the New Algorithm
- Shift-And Matching
- Sunday-Shift
- Examples

Experimental Results

Conclusion

Smyth, Wang, Yu
Adaptive Indeterminate Pattern-Matching Algorithm
Outline

1. Introduction

2. Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaptation of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm

3. Special Properties of Indeterminate Borders

4. The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples

5. Experimental Results

6. Conclusion
Outline

1. Introduction
2. Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3. Special Properties of Indeterminate Borders
4. The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5. Experimental Results
6. Conclusion

Smyth, Wang, Yu

Adaptive Indeterminate Pattern-Matching Algorithm
Regular Pattern Matching Algorithms

Over the last several decades, dozens of regular pattern-matching algorithms have been proposed.

- **Window shifting**: KMP [KMP77], BM [BM77], FJS [FJS06], etc.
- **Bit-parallel**: Shift-Or [Döm68, WM92, BYG92], BNDM [NR98], etc.
Over the last several decades, dozens of regular pattern-matching algorithms have been proposed.

- **Window shifting**: KMP [KMP77], BM [BM77], FJS [FJS06], etc.
- **Bit-parallel**: Shift-Or [Döm68, WM92, BYG92], BNDM [NR98], etc.
Indeterminate Pattern-Matching Algorithms

Intuitive approach: Modify existing regular pattern-matching algorithms to do indeterminate pattern-matching.

- **Shift-Or**: Can be modified to indeterminate pattern-matching easily, with the same speed of regular pattern-matching.
- **iBMS**: A very fast indeterminate pattern-matching algorithm based on BMS has been proposed in [HSW06b].
- **iFJS**: An indeterminate pattern-matching algorithm based on modified FJS (cut-off border array) has been proposed in [HSW06a].
Indeterminate Pattern-Matching Algorithms

Intuitive approach: Modify existing regular pattern-matching algorithms to do indeterminate pattern-matching.

- **Shift-Or**: Can be modified to indeterminate pattern-matching easily, with the same speed of regular pattern-matching.
- **iBMS**: A very fast indeterminate pattern-matching algorithm based on BMS has been proposed in [HSW06b].
- **iFJS**: An indeterminate pattern-matching algorithm based on modified FJS (cut-off border array) has been proposed in [HSW06a].
Indeterminate Pattern-Matching Algorithms

Intuitive approach: Modify existing regular pattern-matching algorithms to do indeterminate pattern-matching.

- **Shift-Or**: Can be modified to indeterminate pattern-matching easily, with the same speed of regular pattern-matching.
- **iBMS**: A very fast indeterminate pattern-matching algorithm based on BMS has been proposed in [HSW06b].
- **iFJS**: An indeterminate pattern-matching algorithm based on modified FJS (cut-off border array) has been proposed in [HSW06a].
Indeterminate String

A (regular) string \( x \) on \( \Sigma \) is a finite sequence of letters drawn from \( \Sigma \). Two letters \( \lambda, \mu \in \Sigma \) are said to match \((\lambda \approx \mu)\) iff \( \lambda = \mu \).

Consider any specified subset \( S = \{\lambda_1, \lambda_2, \ldots, \lambda_j\} \) of \( \Sigma \), \( j \geq 2 \).

We introduce the idea of an indeterminate letter \( \lambda = \lambda_S \) with the property that it matches every element of \( S \) (but no other letter); we write

\[
\lambda \approx \lambda_1, \lambda \approx \lambda_2, \ldots, \lambda \approx \lambda_j.
\]

Given two subsets \( S, T \) of \( \Sigma \), \( |S| \geq 2 \), \( |T| \geq 2 \), and indeterminate letters \( \lambda, \mu \) associated with \( S, T \) respectively, \( \lambda \approx \mu \iff S \cap T \neq \emptyset \).

Given two indeterminate strings \( x \) and \( y \), \( x \approx y \iff (|x| = |y|) \land (\forall i \in 1..|x|, x[i] \approx y[i]) \).
Outline

1 Introduction
2 Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3 Special Properties of Indeterminate Borders
4 The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5 Experimental Results
6 Conclusion
The Knuth-Morris-Pratt Algorithm

- A well-known linear time pattern-matching algorithm.
- Based on border array calculation.
- However, not very fast in practice.
The KMP Algorithm - 1

Smyth, Wang, Yu

Adaptive Indeterminate Pattern-Matching Algorithm
The KMP Algorithm - 2

The longest border of $p[1..j-1]$ is equal to the pattern $b$. 

Smyth, Wang, Yu  Adaptive Indeterminate Pattern-Matching Algorithm
The KMP Algorithm - 3

The Knuth-Morris-Pratt Algorithm
The Sunday Adaption of Boyer-Moore Algorithm
The Shift-And Algorithm
The Franek-Jennings-Smyth Algorithm
The Sunday Adaption of Boyer-Moore Algorithm

- A simplified version of the Boyer-Moore algorithms.
- Time complexity $O(mn)$.
- However, very fast in practice.
The BMS Algorithm - 1

\[ x \ldots \Box_{a} \ldots \]

\[ p \Box_{b} \]

\[ 1 \quad i-m+1 \quad i \quad i \quad n \]

\[ 1 \quad m \]
The BMS Algorithm - 2

Smyth, Wang, Yu

Adaptive Indeterminate Pattern-Matching Algorithm
The BMS Algorithm - 3

Smyth, Wang, Yu

Adaptive Indeterminate Pattern-Matching Algorithm
The BMS Algorithm - 4

The Knuth-Morris-Pratt Algorithm
The Sunday Adaption of Boyer-Moore Algorithm
The New Hybrid Algorithm
The Shift-And Algorithm
The Franek-Jennings-Smyth Algorithm

Smyth, Wang, Yu
Adaptive Indeterminate Pattern-Matching Algorithm
The BMS Algorithm - 5

Smyth, Wang, Yu

Adaptive Indeterminate Pattern-Matching Algorithm
The BMS Algorithm - 6

\[ x \quad 1 \quad \mathbf{c} \quad n \]

\[ p \quad 1 \quad \mathbf{c} \quad m \]

compare
The Shift-And Algorithm

- Makes use of the bit-parallel nature of computer.
- Time complexity $O(mn/w)$.
- Can be easily modified for indeterminate pattern-matching.
The Shift-And Algorithm

<table>
<thead>
<tr>
<th>m\Sigma</th>
<th>A</th>
<th>C</th>
<th>G</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>T</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Preprocessing();

BitArray: $D[1..n]$

for $i = 2$ to $n$ do

$D[i] \leftarrow (\text{Shift}(D[i - 1]) \& S_x[i])$;

if $D_m \& 10^{m-1}$ then output $i - m + 1$;
The Franek-Jennings-Smyth Algorithm

A hybrid algorithm that combines the KMP and BMS algorithm.

Inherits the merits of both algorithms: very fast both asymptotically \( O(n) \) and in practice.
Outline of the FJS Algorithm

1. Perform **Sunday** shift along text.
2. When a match of letters is found at the end of the pattern, switch to **KMP** matching.
3. Continue **KMP** matching until no border can be used, then switch back to **Sunday** shift.
Outline

1. Introduction
2. Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3. Special Properties of Indeterminate Borders
4. The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5. Experimental Results
6. Conclusion
Example of Non-transitivity Effect

Suppose we are performing KMP matching along the text.

<table>
<thead>
<tr>
<th>Index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>......</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>$p$</td>
<td>a</td>
<td>*</td>
<td>*</td>
<td>b</td>
<td>a</td>
<td>*</td>
<td>a</td>
</tr>
<tr>
<td>1st Shift</td>
<td>a</td>
<td>*</td>
<td>*</td>
<td>b</td>
<td>a</td>
<td>......</td>
<td></td>
</tr>
<tr>
<td>2nd Shift</td>
<td>a</td>
<td>*</td>
<td>*</td>
<td>......</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3rd Shift</td>
<td>a</td>
<td>*</td>
<td>......</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table: First example of the non-transitivity effect
Proposition

Shifting the pattern to the right according to the longest border cannot guarantee a prefix match.

Proposition

A border of a border of $x$ is not necessarily a border of $x$. 
Index: 1 2 3 4 5 6 7
\( x \) ...... \( a \ b \ a \ * \ a \ * \ a \) ...... \
\( p \) \( a \ b \ a \ a \ a \ b \ b \)
Wrong Shift: \( a \ b \ a \) ...... 
Correct Shift: \( a \ b \ a \ a \ a \) ...... 

**Table:** Second example of the non-transitivity effect

**Proposition**

*Shifting the pattern to the right according to the longest border can miss some occurrences in between.*
Because of these properties, transforming regular pattern-matching algorithms that use border arrays into indeterminate pattern-matching algorithms is non-trivial (KMP, FJS, etc.)

However, since some of these regular algorithms are very fast in practice and have nice properties, we are motivated to invent indeterminate versions of them that avoid using border arrays.
Outline

1. Introduction
2. Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3. Special Properties of Indeterminate Borders
4. The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5. Experimental Results
6. Conclusion

Adaptive Indeterminate Pattern-Matching Algorithm
A New Hybrid Algorithm

We propose a new hybrid algorithm that uses Shift-And and BMS as complementary shift engines.

1. Perform **Sunday shift** along text.
2. When a match of letters is found at the end of the pattern, switch to **Shift-And matching**.
3. Continue **Shift-And matching** until no match can be found at the current position \((D = 0)\), then skip to next possible position and switch back to **Sunday shift**.
The usual Shift-And preprocessing is modified as follows:

\[
\text{for } i = 1 \text{ to } m \\
\quad \text{for } j = 1 \text{ to } |\Sigma| \\
\quad \quad \text{if } \text{MATCH}(p[i], \Sigma[j]) \text{ then } S[i, j] = 1 \\
\quad \quad \text{else } S[i, j] = 0
\]
Properties of Shift-Or

Notice some of the important properties of Shift-Or.

**Proposition**

\[ D[j] = 1 \iff p[1..j] \approx x[i - j + 1..i] \]

**Proposition**

\( D = 0 \) if and only if there doesn't exist any \( j \in 1..m \) such that \( p[1..j] \approx x[i - j + 1..i] \)

These properties enables us to move the pattern beyond \( x[i] \) when we finish \texttt{ShiftAnd-Match}.
ShiftAnd-MATCH

\[ D \leftarrow 0 \]
\[ \text{do} \]
\[ \quad D \leftarrow (D \ll 1) \& S_x[i] \]
\[ \quad \text{if } D \& 10^m \neq 0 \text{ then output } i \]
\[ \quad i \leftarrow i + 1 \]
\[ \text{//If } D = 0, \text{ terminate loop according to previous proposition} \]
\[ \text{while } (i \leq n \text{ and } D \neq 0) \]
BMS Preprocessing

The usual BMS preprocessing is modified as follows.

\[
\begin{align*}
\text{for } i = 1 & \text{ to } |\Delta| \\
\Delta[i] & = m + 1 \\
\text{for } i = 1 & \text{ to } m \\
& \quad \text{for } j = 1 \text{ to } |\Sigma| \\
& \quad \text{if } \text{MATCH}(p[i], \Sigma[j]) \text{ then } \Delta[p[i]] = i
\end{align*}
\]
Sunday-Shift

while not MATCH\(p[m], x[i']\) do
    \(i' \leftarrow i' + \Delta x[i' + 1]\)
    if \(i' > n\) then return
Algorithm Shift-And/Sunday

\[ i' \leftarrow m; \quad m' \leftarrow m - 1; \]
\[ \textbf{while } i' \leq n \textbf{ do} \]
\[ \quad \text{Sunday-Shift();} \]
\[ \quad i \leftarrow i' - m'; \]
\[ \quad //\text{After Sunday-Shift stops, perform ShiftAnd-MATCH} \]
\[ \quad \text{ShiftAnd-Match();} \]
\[ \quad //\text{After ShiftAnd-Match stops, shift pattern to the right} \]
\[ \quad i' \leftarrow i + m'; \]
Example of The New Hybrid Algorithm

\[
\begin{array}{cccccc}
  & x & \ldots & a & a & b & b & c & b & b & \ldots \\
p & a & \ast & b & b \\
D & 0 & 0 & 0 & 0 \\
1 \gg D & 1 & 0 & 0 & 0 \\
Sa & 1 & 1 & 0 & 0 \\
D' & 1 & 0 & 0 & 0
\end{array}
\]

\text{Shift-Or Match}
Example of The New Hybrid Algorithm

\[
\begin{align*}
  x & \quad \ldots \quad a \ a \ b \ b \ c \ b \ b \ b \ b \ \ldots \\
  p & \quad \quad \overset{a \ \ast \ b \ b}{\quad} \\
  D  & \quad \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \\
  1 \gg D & \quad \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \\
  S_a & \quad \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \\
  D' & \quad \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}
\end{align*}
\]
Example of The New Hybrid Algorithm

\[
\begin{align*}
x & \quad \ldots \quad a \quad a \quad b \quad b \quad c \quad b \quad b \quad \ldots \\
p & \quad \quad \quad \quad \quad a \quad * \quad b \quad b \\
D & \quad \quad \quad \quad \quad 1 \quad 1 \quad 0 \quad 0 \\
1 \gg D & \quad \quad \quad \quad \quad 1 \quad 1 \quad 1 \quad 0 \\
S_b & \quad \quad \quad \quad \quad 0 \quad 1 \quad 1 \quad 1 \\
D' & \quad \quad \quad \quad \quad 0 \quad 1 \quad 1 \quad 0
\end{align*}
\]
Example of The New Hybrid Algorithm

\[
x \quad \ldots \quad a \quad a \quad b \quad b \quad c \quad b \quad b \quad \ldots
\]

\[
p \quad \begin{array}{c}
a \\
\ast \\
b \\
b
\end{array}
\]

\[
D \quad \begin{array}{c}
0 \\
1 \\
1 \\
0
\end{array}
\]

\[
1 \gg D \quad \begin{array}{c}
1 \\
0 \\
1 \\
1
\end{array}
\]

\[
Sb \quad \begin{array}{c}
0 \\
1 \\
1 \\
1
\end{array}
\]

\[
D' \quad \begin{array}{c}
0 \\
0 \\
1 \\
1
\end{array}
\]
Example of The New Hybrid Algorithm

\[ x \ldots \begin{array}{cccccc}
  a & a & b & b & c & b & b \\
\end{array} \ldots \]

\[ p \quad \begin{array}{ccc}
  a & * & b & b \\
\end{array} \]

\[ D \quad \begin{array}{ccc}
  0 & 0 & 1 & 1 \\
\end{array} \]

\[ 1 \gg\gg D \quad \begin{array}{ccc}
  1 & 0 & 0 & 1 \\
\end{array} \]

\[ S_c \quad \begin{array}{ccc}
  0 & 1 & 0 & 0 \\
\end{array} \]

\[ D' \quad \begin{array}{ccc}
  0 & 0 & 0 & 0 \\
\end{array} \]
Example of The New Hybrid Algorithm

\begin{itemize}
  \item $x \ldots \text{a a b b c b b b b a} \ldots$
  \item $p \quad \text{a * b b}$
\end{itemize}

Begin \textit{Sunday-Shift}
Example of The New Hybrid Algorithm

\[ x \ldots a \_a b b c b b b b a \ldots \]
\[ p \quad a \_b b \]

Smyth, Wang, Yu
Adaptive Indeterminate Pattern-Matching Algorithm
Outline

1. Introduction
2. Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3. Special Properties of Indeterminate Borders
4. The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5. Experimental Results
6. Conclusion
Adaptive Indeterminate Pattern-Matching Algorithm
Smyth, Wang, Yu  Adaptive Indeterminate Pattern-Matching Algorithm
Smyth, Wang, Yu
Adaptive Indeterminate Pattern-Matching Algorithm
In all of these tests, the hybrid algorithm’s behaviour is very close to that of the better of BMS and Shift-And.

The new algorithms’s total running time is very competitive among these three algorithms being tested.
Outline

1. Introduction
2. Fundamental Algorithms
   - The Knuth-Morris-Pratt Algorithm
   - The Sunday Adaption of Boyer-Moore Algorithm
   - The Shift-And Algorithm
   - The Franek-Jennings-Smyth Algorithm
3. Special Properties of Indeterminate Borders
4. The New Hybrid Algorithm
   - Outline of the New Algorithm
   - Shift-And Matching
   - Sunday-Shift
   - Examples
5. Experimental Results
6. Conclusion

Smyth, Wang, Yu
Adaptive Indeterminate Pattern-Matching Algorithm
A new algorithm that performs fast pattern-matching on both regular and indeterminate strings.

Strong ability to adapt to the nature of text/pattern and to achieve faster performance over cases that arise in practice. This dynamic adaptivity is useful when we do not know the type of text or pattern: we don’t need to make a decision ahead of time about which algorithm to use.

Future work: Indeterminate pattern-matching algorithms based on variants of Shift-And such as BNDM and [Fre07], as well as on new convolution techniques [AAR07].


Frantisek Franek, Christopher G. Jennings, and W. F. Smyth.
A simple fast hybrid pattern-matching algorithm.

Kimmo Fredriksson.
Linear worst case time bndm.

Hybrid pattern-matching algorithms on indeterminate strings.


S. Wu and U. Manber.
Fast text searching with errors.