 imágenes

 kilograms
The regular expression language employed by XFS is an extended version of standard regular expressions.

* (down)* (down) and a generation operation. (down)

The interface of XFS includes a lookup operation.

XFS includes a regular expression compiler.

.intersection, composition etc.)

.algorithms such as union, concatenation, iteration,

XFS is an interface giving access to finite-state operations.

Introduction to XFS
equivalent to [c a t]  

a single multicharacter symbol with multicharacter print name  

denotes any single symbol  

Grouping brackets  

a concatenation of three symbols  

a simple symbol

Introduction to XES
set difference \( A - B \)

concatenation \( A \circ B \)

intersection \( A \cap B \)

optional, equivalent to \([A\mid0]\)

union \( A \cup B \)

backetning, equivalent to \( A \)

the empty string \( \varepsilon \)

the empty string \( \emptyset \)

Introduction to XFSY
the literal plus-sign symbol

the empty language

the complement of $A$, equivalent to $\{ \ast \} - A$

the universal language

one or more iterations

Kleene-star

Introdution to XFSF
Introduction to XFS – denoting relations
the complement of \( A \), i.e., \([\emptyset \setminus A]\)

any single symbol, except \( b \), compare to:

to \([\emptyset \setminus A]\), e.g.,

any single symbol, minus strings in \( A \), equivalent

aba etc.

includes strings such as \( a, aa, aaa, ba, ab, \)

strings from \( B \), e.g.,

the language of all the strings in \( A \), ignoring any

equivalent to \([\emptyset \setminus A]\), e.g.,

the language of all the strings that contain \( A \),

Introduction to XFS - useful abbreviations
Introduction to XFS — example
Introduction to XFS7 - user interface
and generation

Introduction to XFS – example of lookup
Introduction to XFS – Variables

```bash
xtst define var1 var2
xtst define var3 var2
xtst define var4 var3 var2
xts
```
Introduction to XFS – Variables
is replaced in the output by lower, string, preceded by leftcontext and followed by rightcontext, with the exception that an occurrence of upper in the input itself denotation is the relation which maps string to themselves,

\[ \text{upper} \rightarrow \text{lower} \mid \text{leftcontext} \rightarrow \text{rightcontext} \]

The simplest replace rule is of the form

Regular expression metajanguage:

Replace rules are an extremely powerful extension of the introduction to XES etc – replace rules
Introduction to XFSR – Replace rules

elsewhere.

The language also has an assimilation rule which changes $p$ to $m$ when the $p$ is followed by $m$.

\[ \text{xtst \ clear \ stack;} \]
\[ \text{xtst \ define \ Rule2 \ N -> u;} \]
\[ \text{xtst \ define \ Rule2 \ m -> p;} \]
\[ \text{xtst \ define \ Rule2 \ p -> m - u;} \]
\[ \text{xtst \ define \ Rule2 \ m -> p - u;} \]
\[ \text{xtst \ define \ Rule2 \ p -> m - u;} \]
\[ \text{xtst \ define \ Rule2 \ m -> p - u;} \]

The language as its environment: $n$ is realized as $m$ before $p$ and as $n$ on its environment: $N$ is realized as a labial $m$ or as a dental $n$ depending $N$ that is realized as a nasal $m$ or as a nasal $n$. The language Bambara has an underspecified nasal morpheme.
Word boundaries can be explicitly referred to:
Introduction to \textsc{xfst} – Replace rules

This can be used to clear unnecessary symbols introduced for

\texttt{xfst} define \texttt{Rule1} \% Morphemeboundary -> 0;

\texttt{bookkeeping}:

\texttt{xfst} define \texttt{Rule2} N -> n ;
\texttt{xfst} define \texttt{Rule2} P -> p ;
\texttt{xfst} define \texttt{Rule3} p -> \texttt{m} - ;
\texttt{xfst} define \texttt{Rule3} p -> \texttt{m} - ;
\texttt{xfst} define \texttt{Rule3} p -> \texttt{m} - ;
\texttt{xfst} define \texttt{Rule3} p -> \texttt{m} - ;
\texttt{xfst} define \texttt{Rule3} p -> \texttt{m} - ;

Contexts can be omitted:
or multiple replacements and multiple contexts:

\[
\text{A} \rightarrow \text{B}, \ \text{B} \rightarrow \text{A} \ | \ | \ \text{I} \rightarrow \text{R}, \ \text{I} \rightarrow \text{R}
\]

or multiple contexts:

\[
\text{A} \rightarrow \text{B}, \ \text{B} \rightarrow \text{A} \ | \ | \ \text{I} \rightarrow \text{R}
\]

or multiple replacements that share the same context:

\[
\text{A} \rightarrow \text{B}, \ \text{B} \rightarrow \text{A}
\]

Rules can define multiple replacements:

Introduction to XEST – Replace rules
Introduction to XFS - Replace rules

Rules can apply in parallel:
Introducing to XEST – Replace rules

\[ q \rightarrow a \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mid \# \mi
Introduction to XFST – marking

The special symbol “...” in the right-hand side of a replace rule stands for whatever was matched in the left-hand side of the rule.

xfst> clear stack;
xfst> read regex [a|e|i|o|u] -> %[ ... %];
xfst> apply down unnecessarily
[u]nn[e]c[e]ss[a]r[i]ly
Introduction to XEST – Marking
et al.

This expression denotes an infinite set of strings, such as

\[ [p \ast n^+] \]

any number of adjectives, and one or more nouns.

Assume that a noun phrase consists of an optional determiner, a noun and one or more adjectives.

letters in a text.

expression symbols represent whole words rather than single

and, \textit{y}, etc. In other words, in this example the regular

tag, using \textit{d} for determiner, \textit{a} for adjective, \textit{n} for noun,

Assume that text is represented as strings of part-of-speech

Introduction to \textit{XFS} — shallow parsing
transducer yields three alternative bracketings:

Applied to the input "danny" (many small cats like milk) this

\[
\{\text{\{\text{\{danny}}\}}\%
\]  

Glance: Consider the expression

A simple noun phrase parser can be thought of as a transducer.

Introduction to XFST – shallow parsing
\{\text{Dan}\} \\\ XST \text{ apply down dan}\n
[\{\% \cdot \cdot \} \% ~ \text{a} \text{n} \text{p} \text{o} \text{g} \text{(} \text{d})]

the desired result:
operator \text{op} \rightarrow \text{instead of the simple replace operator} \rightarrow \text{yields}

\text{using the left-to-right, longest-match replace}
unique parse, marking the maximal expansion of each NPs:

For certain applications it may be desirable to produce a

Introduction to XFS – Longest match
amounts to 65 cents. A vending machine disperses 5 cents for 65 cents a can.

Introduction to XFS – The Coke Machine

Construct a regular expression that compiles into a finite-state automaton that implements the behavior of the soft-drink machine, pairing "PLONK" with a legal sequence that

represents as "n"), 10 cents ("d") or 25 cents ("b")

It accepts any sequence of the following coins: 5 cents, 10 cents, or 25 cents.

A vending machine dispenses 5 cents for 65 cents a can.
The solution:

* [n \cdot x \cdot c_5] | [d \cdot x \cdot c_{10}] | [g \cdot x \cdot c_{25}] 

The corresponding value:

A mapping from all possible sequences of the three symbols

A nickel is worth 5 cents.

Thus the expression \( n \cdot x \cdot c_5 \) expresses the fact that a

Itself \( n \) times.

The construction \( A_n \) denotes the concatenation of \( A \) with

Introduction to XFS – the coke machine
clear stack

Introduction to XFST – the coke machine

determine the coke

Sixty minutes. 0. [c·65 x. Plonk]!
clear stack

dimes before nickels.

To ensure that the extra change is paid out only once, we

modify the lower language of BuyCoke to make it a subset of

Introduction to XFS – the coke machine
Introduction to XFS – the Coke machine
to the client. This goal can be achieved easily by wrapping objects that should not be accepted; they should pass right back into the machine (subway tokens, foreign coins, etc.). These final improvements. Some clients may insert unwanted items into the machine completely foolproof, we need one

Introduction to XFS – the coke machine