# How Fragments Become an NFA: Or, How Sausage is Made

March 22, 2011

## 1 Fragments

Figure 1 shows an arbitrary fragment $A$. Along the left edge of the fragment is its in list $i_0, \ldots, i_{n-1}$, a list of $n$ vertices by which the fragment may be entered; along the right edge is the fragment's out list $\langle o_0, s_0 \rangle, \ldots, \langle o_{m-1}, s_{m-1} \rangle$, a list of $m$ pairs where the $o_i$ is a vertex from which the fragment may be exited and $s_i$ is the position in $o_i$'s outgoing edge list where new edges should be inserted. $k$ is the position in the fragment's in list where edges skipping the fragment should be inserted. For nonskippable fragments, $k = \emptyset$. (Note that $\emptyset \neq 0$; rather, it is intended to mean "none". Zero is a valid insertion point in a list, $\emptyset$ is not.) For skippable fragments, $0 \leq k \leq n$. The order of outgoing edges for any vertex is clockwise, starting from the top.

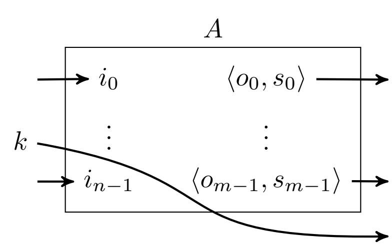



Figure 1: An arbitrary fragment

## 2 Atoms

Figure 2 shows an atomic fragment, i.e., a fragment consisting of a single vertex $v$. (Such a fragment may be produced by a literal, a character class, or the dot.) The in and out lists consist of $v$ only, and the new edge insertion point for $v$ is 0, the head of $v$'s out edge list, because $v$'s out edge list is empty. Atoms are not skippable, so $k = \emptyset$.

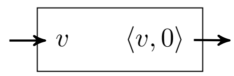
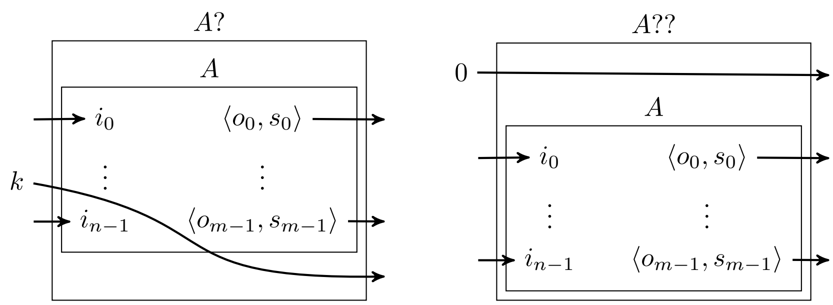



Figure 2: An atom

# 3 Repetition

Figure 3 shows how $A$ is converted to $A?$ or $A??$. In the greedy case, $A?.k = \min(A.k, |A.\mathrm{In}|)$ (where $\emptyset$ is treated like $+\infty$), while in the nongreedy case $A??.k = 0$. In both cases $A?.\mathrm{In} = A??.\mathrm{In} = A.\mathrm{In}$, $A?.\mathrm{Out} = A??.\mathrm{Out} = A.\mathrm{Out}$.

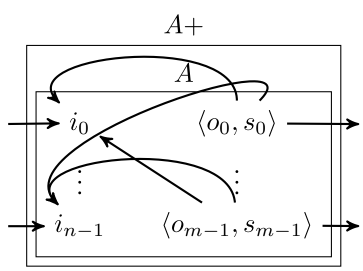



(a) Greedy

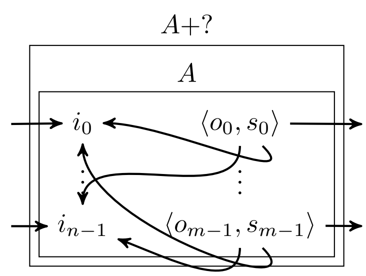
(b) Nongreedy

Figure 3: Single repetition

Figure 4 shows how $A$ is converted to $A+$ or $A+?$. Out edges are added from each $o_i$ to each $i_j$ to create the necessary loops. Adding a plus does not affect the skippability of $A$, due to the fact that matching the empty string once is the same as matching the empty string any greater number of times; hence $A?.k = A??.k = A.k$.




(a) Greedy




(b) Nongreedy

Figure 4: Unbounded repetition

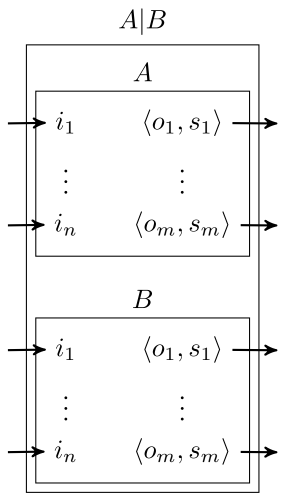
No diagrams are given for the conversion of $A$ to $A*$ or $A*?$, as these are equivalent to $(A+)?$ and $(A+?)??$, respectively, so can be constructed from the above.

## 4 Alternation

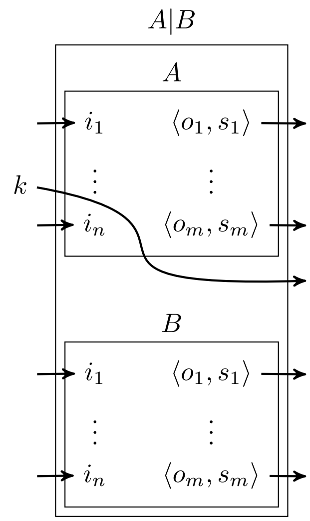
Figure 5 shows how $A|B$ is formed from $A$ and $B$. In all cases, $A|B.\mathrm{In} = A.\mathrm{In} + B.\mathrm{In}$, $A|B.\mathrm{Out} = A.\mathrm{Out} + B.\mathrm{Out}$. Finally,

$$A|B.k = \begin{cases} \emptyset & \text{if } A.k = B.k = \emptyset, \\ A.k & \text{if } A.k \neq \emptyset, \\ |A.\mathrm{In}| + B.k & \text{if } B.k \neq \emptyset. \end{cases}$$

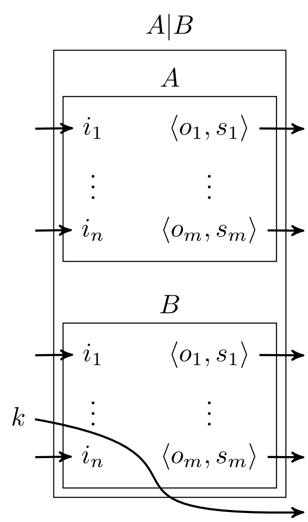
The intuition behind the skippability for $A|B$ is as follows: If a fragment is skippable, that means it matches the empty string. If $A$ matches the empty string, since $A$ matches for $A$ have priority over matches for $B$, the empty string should be matched by $A|B$ with the priority $A$ gives it. Otherwise, if $A$ is not skippable, but $B$ is, since $A|B.\mathrm{In}$ is just $B.\mathrm{In}$ with $A.\mathrm{In}$ prepended to it, and $B.k$ is an insertion position, $B.k$ needs to be shifted by the size of $A.\mathrm{In}$ to give us $A|B.k$.




(a) $A.k = B.k = \emptyset$




(b) $A.k \neq \emptyset$




(c) $A.k = \emptyset, B.k \neq \emptyset$

Figure 5: Alternation

# 5 Concatenation

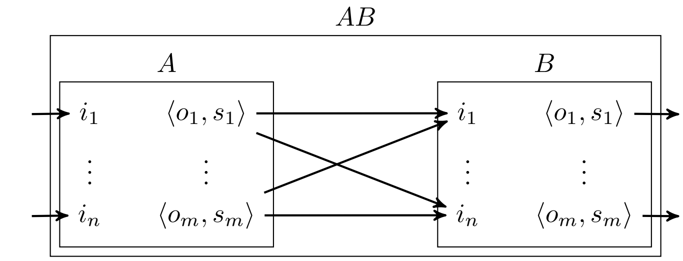
Figure 6 shows how $AB$ is formed from $A$ and $B$. There are four cases, depending on whether either $A$ or $B$ is skippable. In what follows, the bracket notation indicates array slices.

$$AB.k = \begin{cases} A.k + B.k & \text{if } A.k \neq \emptyset \text{ and } B.k \neq \emptyset, \\ \emptyset & \text{otherwise.} \end{cases}$$

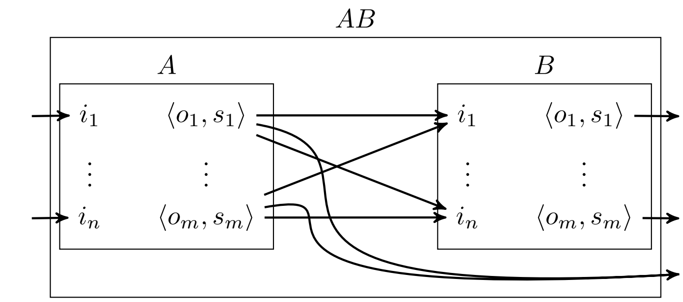
$$AB.\mathrm{In} = \begin{cases} A.\mathrm{In}[0 : A.k - 1] + B.\mathrm{In} + A.\mathrm{In}[A.k : |A.\mathrm{In}|] & \text{if } A.k \neq \emptyset, \\ A.\mathrm{In} & \text{otherwise.} \end{cases}$$

$$AB.\mathrm{Out} = \begin{cases} B.\mathrm{Out} + \{\langle v, s\rangle \mid \langle v, s'\rangle \in A.\mathrm{Out} \wedge s = |v.\mathrm{Out}| + B.k\} & \text{if } B.k \neq \emptyset, \\ B.\mathrm{Out} & \text{otherwise.} \end{cases}$$

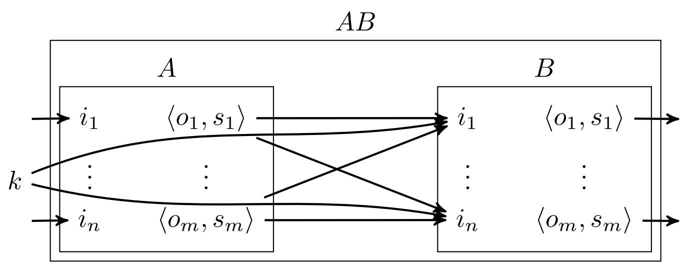
The skippability of $A$ determines $AB.\mathrm{In}$; the skippability of $B$ determines $AB.\mathrm{Out}$; the skippability of $A$ and $B$ jointly determine the skippability of $AB.k$.



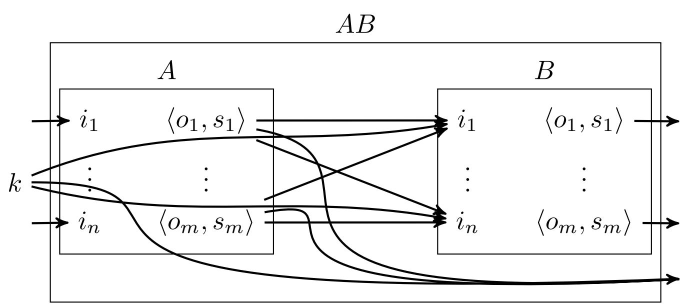

(a) $A.k = B.k = \emptyset$




(b) $A.k = \emptyset, B.k \neq \emptyset$




(c) $A.k \neq \emptyset, B.k = \emptyset$




(d) $A.k \neq \emptyset, B.k \neq \emptyset$

Figure 6: Concatenation