String Prefixes Abstract Domain

Ben Hardekopf

1 Prefix Lattice

Let Σ be an alphabet, $\Sigma*$ be the set of all strings using that alphabet, and $[\Sigma*]$ contain the same strings as $\Sigma*$ except the strings are recognizably from $[\Sigma*]$ (i.e., strings contained in brackets signify prefixes rather than exact strings). We will interpret a string $[prefix] \in [\Sigma]$ as representing the set of all strings that have *prefix* as a prefix. Let $P = \Sigma * \cup [\Sigma*]$.

For $str_1, str_2 \in P$, define $lcp(str_1, str_2)$ as the longest common prefix of str_1 and str_2. Define $S = \{\bot\} \cup P$. The prefix lattice is $(S, \sqsubseteq, \sqcup, \sqcap)$ where for $s_1, s_2 \in S$:

- $\top$ is defined as $[\epsilon]$.
- $\sqcup$ is defined as
  - $\bot \sqcup s_1 = s_1 \sqcup \bot = s_1$
  - $s_1 \sqcup s_1 = s_1$
  - $s_1 \sqcup s_2 = [lcp(s_1, s_2)]$, for $s_1 \neq s_2$ and $s_1, s_2 \neq \bot$
- $\sqcap$ is defined as
  - $s_1 \sqcap s_2 = s_1$ if $s_1 \sqcup s_2 = s_2$
  - $s_1 \sqcap s_2 = \bot$ otherwise.
- $\sqsubseteq$ is defined as $s_1 \sqsubseteq s_2$ iff $s_1 \sqcup s_2 = s_2$.

1.1 Lattice Properties

Chains in this lattice consist of taking a string and successively removing the last letter until we reach $\top$. The lattice is infinitely high because there is no bound on how long a string can be. However, any given string has a finite length, and thus we are guaranteed to reach $\top$ in a finite number of steps. Therefore the lattice is noetherian.

2 Abstraction and Concretization Functions

$$\alpha : \mathcal{P}(\Sigma*) \to S$$

$$\alpha(x) = \begin{cases} \bot & \text{if } x = \{\} \\ \mathit{str} & \text{if } x = \{\mathit{str}\} \\ [\mathit{lcp}(x)] & \text{otherwise} \end{cases}$$

$$\gamma : S \to \mathcal{P}(\Sigma*)$$

$$\gamma(x) = \begin{cases} \{\} & \text{if } x = \bot \\ \{\mathit{str}\} & \text{if } x = \mathit{str} \\ \{\mathit{str} \mid \mathit{lcp}(\mathit{str}, [\mathit{prefix}]) = \mathit{prefix}\} & \text{if } x = [\mathit{prefix}] \end{cases}$$

3 Abstract + Operator

Define $\widehat{+}$ as:

$\widehat{+}$	$\bot$	str_2	$[\mathit{prefix}_2]$
$\bot$	$\bot$	$\bot$	$\bot$
str_1	$\bot$	$\mathit{str}_1 + \mathit{str}_2$	$[\mathit{str}_1 + \mathit{prefix}_2]$
$[\mathit{prefix}_1]$	$\bot$	$[\mathit{prefix}_1]$	$[\mathit{prefix}_1]$

$\widehat{+}$ is monotone if $a \sqsubseteq a'$ and $b \sqsubseteq b'$ implies $a \mathbin{\widehat{+}} b \sqsubseteq a' \mathbin{\widehat{+}} b'$. We use the fact that for $\mathit{str}_1, \mathit{str}_2 \in P$, by definition of the $\sqsubseteq$ relation, $\mathit{str}_1 \sqsubseteq \mathit{str}_2$ iff $\mathit{lcp}(\mathit{str}_1, \mathit{str}_2) = \mathit{str}_2$. Consider the following cases in order (i.e., if multiple cases apply use the first case listed below):

Case 1. One of a or b is $\bot$. Then $a \mathbin{\widehat{+}} b = \bot$, which is $\sqsubseteq$ everything. QED.

Case 2. $a' = \mathit{str}_1$, $b' = \mathit{str}_2$, and $a, b \neq \bot$. Then $a = a'$ and $b = b'$, therefore $a \mathbin{\widehat{+}} b = a' \mathbin{\widehat{+}} b'$. QED.

Case 3. $a' = \mathit{str}_1$, $b' = [\mathit{prefix}_2]$, and $a, b \neq \bot$. Then $a = a'$ and $\mathit{lcp}(b, b') = b'$. By definition of the $\sqsubseteq$ relation, $[\mathit{str}_1 + \mathit{str}] \sqsubseteq [\mathit{str}_1 + \mathit{prefix}_2]$ for any $\mathit{str} \in P$ s.t. $\mathit{lcp}(\mathit{str}, [\mathit{prefix}_2]) = \mathit{prefix}_2$. QED.

Case 4. $a' = [\mathit{prefix}_1]$ and $a, b, b' \neq \bot$. Then $a' \mathbin{\widehat{+}} b' = a'$. Because $a \sqsubseteq a'$, $\mathit{lcp}(a + \mathit{str}, a') = a'$ for any $\mathit{str} \in P$. Therefore, by definition of the $\sqsubseteq$ relation, $a \mathbin{\widehat{+}} b \sqsubseteq a'$. QED.

4 Abstract $\leq$ Operator

Let $str_1, str_2 \in \Sigma*$ and $[prefix_1], [prefix_2] \in [\Sigma*]$ and $s \in S$.

$$
\begin{aligned}
s \mathrel{\widehat{\leq}} \bot &= \bot \\
\bot \mathrel{\widehat{\leq}} s &= \bot \\
str_1 \mathrel{\widehat{\leq}} str_2 &= str_1 \leq str_2 \\
str_1 \mathrel{\widehat{\leq}} [prefix_1] &= \begin{cases} \top & \text{if } str_1 \sqsubseteq [prefix_1] \text{ and } str_1 \neq prefix_1 \\ str_1 \leq prefix_1 & \text{otherwise} \end{cases} \\
[prefix_1] \mathrel{\widehat{\leq}} str_1 &= \begin{cases} \top & \text{if } str_1 \sqsubseteq [prefix_1] \\ prefix_1 \leq str_1 & \text{otherwise} \end{cases} \\
[prefix_1] \mathrel{\widehat{\leq}} [prefix_2] &= \begin{cases} \top & \text{if } [prefix_1] \sqsubseteq [prefix_2] \text{ or } [prefix_2] \sqsubseteq [prefix_1] \\ prefix_1 \leq prefix_2 & \text{otherwise} \end{cases}
\end{aligned}
$$

$\widehat{\leq}$ is monotone if $a \sqsubseteq a'$ and $b \sqsubseteq b'$ implies $a \mathrel{\widehat{\leq}} b \sqsubseteq a' \mathrel{\widehat{\leq}} b'$.

Case 1. $a \mathrel{\widehat{\leq}} b = \bot$ or $a' \mathrel{\widehat{\leq}} b' = \top$. Trivially true. QED.

Case 2. $a' = str_1$, $b' = str_2$, $a, b \neq \bot$. Then $a = a'$ and $b = b'$. Therefore $a \mathrel{\widehat{\leq}} b = a' \leq b'$. QED.

Case 3. $a' = str_1$, $b' = [prefix_1]$, $a, b \neq \bot$, and $a' \mathrel{\widehat{\leq}} b'$ yields an exact answer. Then $a = a'$ and b' is a prefix of b. If b is an exact string then $\widehat{\leq}$ will necessarily do an exact comparison and return the same answer as $a' \mathrel{\widehat{\leq}} b'$. Otherwise, we are comparing an exact string a against a prefix string b. Because $a = a'$ we know that $a \not\sqsubseteq b'$ or $a = b'$. In either case, becase b' is a prefix of b it must be that $a \not\sqsubseteq b$. Thus we go to the exact case and give the same answer as $a' \mathrel{\widehat{\leq}} b'$. QED.

Case 4. $a' = [prefix_1]$, $b' = str_1$, $a, b \neq \bot$, and $a' \mathrel{\widehat{\leq}} b'$ yields an exact answer. Using the same reasoning as Case 3 except switching the roles of a, a' and b, b', we must give the same answer for $a \mathrel{\widehat{\leq}} b$ as for $a' \mathrel{\widehat{\leq}} b'$. QED.

Case 5. $a' = [prefix_1]$, $b' = [prefix_2]$, $a, b \neq \bot$, and $a' \mathrel{\widehat{\leq}} b'$ yields an exact answer. a' must be a prefix of a and b' must be a prefix of b. Because $a' \mathrel{\widehat{\leq}} b'$ yields an exact answer we know that $a' \not\sqsubseteq b'$ and $b' \not\sqsubseteq a'$. Neither a' nor b' is a prefix of the other, and so neither a nor b can be a prefix of the other, i.e., $a \not\sqsubseteq b$ and $b \not\sqsubseteq a$. Thus regardless of what combination of exact or prefix strings a and b are, we know $a \mathrel{\widehat{\leq}} b$ must give an exact answer that is the same as $a' \mathrel{\widehat{\leq}} b'$. QED.