ZIML Translation

Jakub Kozakoszczak

November 4, 2024

Contents

1 Given:

1.1 Annotated text

Σ - alphabet
L - labels
$\Sigma \cap L = \emptyset$
$\underset{n \in \mathbb{N}}{(t_n)}$ - plain text, $t : \mathbb{N} \to \Sigma$
$\underset{n \in \mathbb{N}}{(a_n)}$ - annotation, $a : \mathbb{N} \to 2^L$

1.2 ZIML text

$\underset{n \in \mathbb{N}}{(c_n)}$ - cipher text, $c : \mathbb{N} \to \Sigma$
$\forall i, j \in \mathbb{N} : \ \langle t_i, a_i \rangle \neq \langle t_j, a_j \rangle \iff c_i \neq c_j$

Observation:
Relation $\{\langle c_i, t_i \rangle : i \in \mathbb{N}\}$ is a function.
Relation $\{\langle c_i, a_i \rangle : i \in \mathbb{N}\}$ is a function.

Given the observation, fix functions:
For $i \in \mathbb{N}$:

$decrypt\_char : c[\mathbb{N}] \to t[\mathbb{N}]$, $decrypt\_char(c_i) = t_i$

$decrypt\_anno : c[\mathbb{N}] \to a[\mathbb{N}]$, $decrypt\_anno(c_i) = a_i$

1.3 Regex language

1.3.1 Regex syntax

E^Σ - language of regular expressions over Σ

$$\emptyset \in E;\ \varepsilon \in E;\ \Sigma \subset E$$

for $e, f \in E$:
$e|f \in E;\ e \cdot f \in E;\ e^* \in E$

1.3.2 Regex semantics in plain text

$[\![\cdot]\!]_T$ - relational interpretation of E^Σ in family $2^{\mathbb{N}\times\mathbb{N}}$ of sets of spans of text (t_n) with operations of $\cup$ for alternative $e|f$, relational composition for concatenation $e \cdot f$, empty relation for $\emptyset$, identity for ε, reflexive transitive closure for *, and classes T_a, T_b... of one character long occurences of characters $a, b... \in \Sigma$ as individua designated by them. This interpretation models the search for regex matches in plain text with a standard regex engine.

for $\alpha \in \Sigma : T_\alpha = \{\langle i, i+1\rangle : i \in \mathbb{N} \wedge t_i = \alpha\}$

$[\![\emptyset]\!]_T = \{\}$
$[\![\varepsilon]\!]_T = \{\langle i, i\rangle : i \in \mathbb{N}\}$

for $\alpha \in \Sigma : [\![\alpha]\!]_T = T_\alpha$

for $e, f \in E$:
$[\![e|f]\!]_T = [\![e]\!]_T \cup [\![f]\!]_T$
$[\![e \cdot f]\!]_T = \{\langle i, k\rangle : \exists j(\langle i, j\rangle \in [\![e]\!]_T \wedge \langle j, k\rangle \in [\![f]\!]_T)\}$
$[\![e^*]\!]_T = \bigcup_{n\in\mathbb{N}} [\![e^n]\!]_T$

$$e^n = \begin{cases} \varepsilon, & \text{if } n = 0, \\ e \cdot e^{n-1}, & \text{otherwise.} \end{cases}$$

1.3.3 Regex semantics in cipher text

$[\![\cdot]\!]_C$ - the same relational interpretation as $[\![\cdot]\!]_T$ but in spans of cipher text (c_n) instead of plain text (t_n) and with occurences C_a, C_b... of characters in cipher not plain text $(T_a, T_b...)$.

for $\alpha \in \Sigma : C_\alpha = \{\langle i, i+1\rangle : i \in \mathbb{N} \wedge c_i = \alpha\}$
for $\alpha \in \Sigma : [\![\alpha]\!]_C = C_\alpha$

1.4 Extended regex

1.4.1 Extended regex syntax

$E^{\Sigma\cup L}$ - extended language of regular expressions over $\Sigma \cup L$, language of z-regexes

$\varnothing \in E^{\Sigma\cup L},\ \varepsilon \in E^{\Sigma\cup L},\ \Sigma \subset E^{\Sigma\cup L},\ L \subset E^{\Sigma\cup L}$

for $e, f \in E^{\Sigma\cup L}$:
$e|f \in E^{\Sigma\cup L};\ e \cdot f \in E^{\Sigma\cup L};\ e^* \in E^{\Sigma\cup L}$

1.4.2 Extended regex semantics

$[\![\cdot]\!]^+_{TA}$ - extended interpretation of z-regexes in the family of sets of spans of both annotation (a_n) and text (t_n). Regex engines fail to implement this desired behaviour.

for $\alpha \in \Sigma : T_\alpha = \{\langle i, i+1\rangle : i \in \mathbb{N} \wedge t_i = \alpha\}$
for $l \in L : A_l = \{\langle i, i+1\rangle : i \in \mathbb{N} \wedge l \in a_i\}$

$[\![\varnothing]\!]^+_{TA} = \{\}$
$[\![\varepsilon]\!]^+_{TA} = \{\langle i, i\rangle : i \in \mathbb{N}\}$

for $\alpha \in \Sigma : [\![\alpha]\!]^+_{TA} = T_\alpha$
for $l \in L : [\![l]\!]^+_{TA} = A_l$

for $e, f \in E^{\Sigma\cup L}$:

$$[\![e|f]\!]^+_{TA} = [\![e]\!]^+_{TA} \cup [\![f]\!]^+_{TA}$$

$$[\![e \cdot f]\!]^+_{TA} = \{\langle i, k\rangle : \exists j(\langle i, j\rangle \in [\![e]\!]^+_{TA} \wedge \langle j, k\rangle \in [\![f]\!]^+_{TA})\}$$

$$[\![e^*]\!]^+_{TA} = \bigcup_{n\in\mathbb{N}} [\![e^n]\!]^+_{TA}$$

$$e^n = \begin{cases} \varepsilon, & \text{if } n = 0, \\ e \cdot e^{n-1}, & \text{otherwise.} \end{cases}$$

2 Definitions

We define paradigms of plain characters, markup of labels and the regex encrypting function z from z-regexes over $\Sigma \cup L$ to standard regexes over Σ.

$$paradigm : \Sigma \to 2^{c[\mathbb{N}]}$$
$$paradigm(\alpha) = decrypt\_char[\{\alpha\}] = \{\beta \in c[\mathbb{N}] : decrypt\_char(\beta) = \alpha\}$$

$$markup : L \to 2^{c[\mathbb{N}]}$$
$$markup(l) = decrypt\_anno[\{l\}] = \{\beta \in c[\mathbb{N}] : l \in decrypt\_anno(\beta)\}$$

z - regex encrypting function : $E^{\Sigma \cup L} \to E$

$$z(\emptyset) = \emptyset$$
$$z(\varepsilon) = \varepsilon$$
$$\text{for } \alpha \in \Sigma : z(\alpha) = \bigvee_{\beta \in paradigm(\alpha)} \beta$$
$$\text{for } l \in L : z(l) = \bigvee_{\beta \in markup(l)} \beta$$
$$\text{for } e, f \in E :$$
$$z(e|f) = z(e)|z(f)$$
$$z(e \cdot f) = z(e) \cdot z(f)$$
$$z(e^*) = z(e)^*$$

3 Claim

Searching for a z-regex pattern e in the annotated text with a hypothetical engine implementing the extended semantics results in the same matches as searching for its encrypted counterpart $z(e)$ in the cipher text with a standard engine.

$$\forall e \in E^{\Sigma \cup L} : [\![e]\!]^+_{TA} = [\![z(e)]\!]_C$$

4 Proof

4.1 Domain independent meaning

To $\emptyset$, ε, $e|f$, $e \cdot f$, e^* both interpretations assign identical relational operations independent of structures.

4.2 Meaning of characters

For $e = \alpha \in \Sigma$:

$$[\![\bigvee_{\beta \in paradigm(\alpha)} \beta]\!]_C \stackrel{?}{=} [\![\alpha]\!]^+_{TA}$$

$$\bigcup_{\beta \in paradigm(\alpha)} [\![\beta]\!]_C$$

$$\bigcup_{\beta \in paradigm(\alpha)} \{\langle i, i+1 \rangle : c_i = \beta\}$$

$$\{\langle i, i+1 \rangle : c_i \in paradigm(\alpha)\}$$

$$\{\langle i, i+1 \rangle : c_i \in \{\beta \in c[\mathbb{N}] : decrypt\_char(\beta) = \alpha\}\}$$

$$\{\langle i, i+1 \rangle : decrypt\_char(c_i) = \alpha\}$$

$$\{\langle i, i+1 \rangle : t_i = \alpha\}$$

$$[\![\alpha]\!]^+_{TA}$$

4.3 Meaning of labels

For $e = l \in L$:

$$[\![\bigvee_{\beta \in markup(l)} \beta]\!]_C \stackrel{?}{=} [\![l]\!]^+_{TA}$$

$$\bigcup_{\beta \in markup(l)} [\![\beta]\!]_C$$

$$\bigcup_{\beta \in markup(l)} \{\langle i, i+1 \rangle : c_i = \beta\}$$

$$\{\langle i, i+1 \rangle : c_i \in markup(l)\}$$

$$\{\langle i, i+1 \rangle : c_i \in \{\beta \in c[\mathbb{N}] : l \in decrypt\_anno(\beta)\}\}$$

$$\{\langle i, i+1 \rangle : l \in decrypt\_anno(c_i)\}$$

$$\{\langle i, i+1 \rangle : l \in a_i\}$$

$$[\![l]\!]^+_{TA}$$