

Claims:

1. A computer-implemented method of refining rules for extracting information comprising:

5 receiving data;

retrieving a regular expression from an extraction pattern database;

annotating said regular expression;

applying a first phase of error detection extension having a plurality of first set of rules by subjecting said data to said annotated regular expression and said plurality of first set of rules based on partial derivative operations;

10 generating a match result from said application of said first ~~phase of error detection~~ extension;

applying a second phase of error detection extension having a plurality of second set of rules by subjecting said data to said annotated regular expression and said plurality of second set of rules based on partial derivative operations;

15 generating a mismatch result from said application of said second ~~phase of error detection~~ extension in response to a mismatch in said data;

applying a refinement process to said data in response to generating said mismatch report from said second phase of extraction error detection extension of information from said data, wherein said plurality of first set of rules is different from said plurality of second set of rules.

2. The computer-implemented method of claim 1 wherein said step of annotating said regular expression comprises:

25 determining functions in said regular expression starting from the outermost function to the innermost function; and

assigning annotation from left to right.

3. The computer-implemented method of claim 2 wherein said step of determining functions in said regular expression comprises:

30 adding parenthesis to each determine function.

4. The computer-implemented method of claim 3 wherein said step of assigning annotation from left to right comprises:

35 assigning in ascending order from the left to the right of the determined functions.

5. The computer-implemented method of claim 1 wherein said step of applying asaid first phase of error detection extension comprises:

- i) determining number of characters in said data;
- ii) determining number of annotations in said annotated regular expression;
- iii) selecting a first character;
- iv) determining a-rules among said plurality of first set rules based on annotated regular expression and said selected character;
- v) executing determined rules;
- vi) generating a regular expression partial derivative and append trace;
- vii) repeating steps (iv)-(vi) for each character in said data; and
- viii) generating a match result.

6. The computer-implemented method of claim 5 wherein said plurality of first set of rules comprises:

$$\boxed{\bar{i} \vdash r/l \Rightarrow (\overline{r'}, \bar{i}')} \quad$$

$$(\text{Eps}) \quad \bar{i} \vdash \epsilon/l \Rightarrow \{\} \quad (\text{Lab1}) \quad \bar{i} \vdash \epsilon/l \Rightarrow \{(\epsilon, \bar{i} + i)\} \quad (\text{Lab2}) \quad \bar{i} \vdash \epsilon/l' \Rightarrow \{\}$$

$$(\text{Seq1}) \quad \frac{\epsilon \in r_1 \quad \bar{i} \vdash r_1/l \Rightarrow (\overline{r'_1, \bar{j}}) \quad \bar{i} \vdash r_2/l \Rightarrow (\overline{r'_2, \bar{j}'})}{\bar{i} \vdash \epsilon.r_1r_2/l \Rightarrow \{(\overline{r'_1r_2, \bar{j}}) \mid (\overline{r'_1, \bar{j}}) \in (\overline{r'_1, \bar{j}}) \cup (\overline{r'_2, \bar{j}'})\}}$$

$$(\text{Seq2}) \quad \frac{\epsilon \notin r_1 \quad \bar{i} \vdash r_1/l \Rightarrow (\overline{r'_1, \bar{j}})}{\bar{i} \vdash \epsilon.r_1r_2/l \Rightarrow (\overline{r'_1, \bar{j}})}$$

$$(\text{Choice}) \quad \frac{\bar{i} \vdash r_1/l \Rightarrow (\overline{r'_1, \bar{j}}) \quad \bar{i} \vdash r_2/l \Rightarrow (\overline{r'_2, \bar{j}'})}{\bar{i} \vdash \epsilon.r_1 + r_2/l \Rightarrow (\overline{r'_1, \bar{j}}) \cup (\overline{r'_2, \bar{j}'})}$$

$$(\text{Star}) \quad \frac{\bar{i} \vdash r/l \Rightarrow (\overline{r'}, \bar{i}')}{\bar{i} \vdash \epsilon.r^*/l \Rightarrow \{(\epsilon.r^*, \bar{i}') \mid (\overline{r'}, \bar{i}') \in (\overline{r'}, \bar{i}')\}}$$

$$\boxed{\bar{i} \vdash r/w \Rightarrow (\overline{r'}, \bar{i}')} \quad$$

$$(\text{WordEps}) \quad \bar{i} \vdash r/\epsilon \Rightarrow \{(r, \bar{i})\}$$

$$(\text{WordCon}) \quad \frac{\bar{i} \vdash r/l \Rightarrow (\overline{r'}, \bar{i}') \quad \text{for } j \in \{1, \dots, n\} \quad \bar{i}'_j \vdash r'_j/w \Rightarrow (\overline{r''_j, \bar{i}''_j})}{\bar{i} \vdash r/lw \Rightarrow \bigcup_j (\overline{r''_j, \bar{i}''_j})}$$

takes an appended trace from the first part $\bar{l}_j$, an intermediate partial derivative regular expression from the first part r'_j , the rest of the string of literal w , and returns with $(\bar{r}_j'', \bar{l}_j'')$, which is the final result for this rule by aggregating across all the j in $\{1...n\}$.

7. The computer-implemented method of claim 6 wherein said step of applying said second phase of error detection extension comprises:

- i) determining a-rules among said plurality of second set rules based on annotated regular expression and said selected character;
- ii) executing determined rules;
- iii) generating a regular expression partial derivative and append matching error;
- iv) repeating steps (i)-(iii) for each character in said data; and
- v) generating a mismatch result.

8. The computer-implemented method of claim 7 wherein said a-plurality of second set of rules comprises:

$$(\text{Match Error}) \ E ::= E_l^{\bar{}} \mid E_c^{\bar{}}$$

$$\boxed{\bar{l} \vdash r/l \Rightarrow (\bar{r}', \bar{l}'), \bar{E}}$$

$$(\text{Eps}) \ \bar{l} \vdash \epsilon/l \Rightarrow \{\}, \{E_c^{\bar{}}\}$$

$$(\text{Lab1}) \ \bar{l} \vdash \epsilon/l \Rightarrow \{(\epsilon, \bar{l} + i)\}, \{\} \quad (\text{Lab2}) \ \bar{l} \vdash \epsilon/l' \Rightarrow \{\}, \{E_l^{\bar{}} + i\}$$

$$(\text{Seq1}) \ \frac{\epsilon \in r_1 \quad \bar{l} \vdash r_1/l \Rightarrow (\bar{r}'_1, \bar{j}), \bar{E}_1 \quad \bar{l} \vdash r_2/l \Rightarrow (\bar{r}'_2, \bar{j}'), \bar{E}_2}{\bar{l} \vdash \epsilon, r_1 r_2/l \Rightarrow \{(\bar{r}'_1 r_2, \bar{j}) \mid (\bar{r}'_1, \bar{j}) \in (\bar{r}'_1, \bar{j})\} \cup (\bar{r}'_2, \bar{j}'), \bar{E}_1 \cup \bar{E}_2}$$

$$(\text{Seq2}) \ \frac{\epsilon \notin r_1 \quad \bar{l} \vdash r_1/l \Rightarrow (\bar{r}'_1, \bar{j}), \bar{E}'}{\bar{l} \vdash \epsilon, r_1 r_2/l \Rightarrow (\bar{r}'_1, \bar{j}), \bar{E}'}$$

$$(\text{Choice}) \ \frac{\bar{l} \vdash r_1/l \Rightarrow (\bar{r}'_1, \bar{j}), \bar{E}_1 \quad \bar{l} \vdash r_2/l \Rightarrow (\bar{r}'_2, \bar{j}'), \bar{E}_2}{\bar{l} \vdash \epsilon, r_1 + r_2/l \Rightarrow (\bar{r}'_1, \bar{j}) \cup (\bar{r}'_2, \bar{j}'), \bar{E}_1 \cup \bar{E}_2}$$

$$(\text{Star}) \ \frac{\bar{l} \vdash r/l \Rightarrow (\bar{r}', \bar{l}'), \bar{E}'}{\bar{l} \vdash \epsilon, r^*/l \Rightarrow \{(\epsilon, r', \bar{l}') \mid (r', \bar{l}') \in (\bar{r}', \bar{l}')\}, \bar{E}'}$$

$$\boxed{\bar{l}, \bar{E} \vdash r/w \Rightarrow (\bar{r}', \bar{l}'), \bar{E}'}$$

$$(\text{WordEps}) \ \bar{l}, \bar{E} \vdash r/\epsilon \Rightarrow \{(r, \bar{l})\}, \bar{E}$$

$$(\text{WordCon}) \ \frac{\bar{l} \vdash r/l \Rightarrow (\bar{r}', \bar{l}'), \bar{E}' \quad \text{for } j \in \{1, \dots, n\} \quad \bar{l}'_j, \bar{E}'_j \vdash r'_j/w \Rightarrow (\bar{r}''_j, \bar{l}''_j), \bar{E}''_j}{\bar{l}, \bar{E} \vdash r/lw \Rightarrow \bigcup_j (\bar{r}''_j, \bar{l}''_j), \bar{E}' \cup \bigcup_j \bar{E}''_j}$$

wherein said plurality of second set of rules is an extension of said plurality of first set of rules to track literal mismatch error $E_l^{\bar{t}}$ and the epsilon mismatch error $E_\epsilon^{\bar{t}}$;

rule (Eps) raises an epsilon mismatch error;

rule (Lab2) raises a literal mismatch error;

5 rule (Lab1) consumes an input literal with the matching literal expression and append the matching source location i to the trace $\bar{t}$;

rules (Seq1) and (Seq2) handle sequence expression r_1r_2 , rule (Seq1) is applicable where $\epsilon \in r_1$, otherwise rule (Seq2) is applied;

rule (Choice) handles choice expression; and

10 rule (Star) handles the Kleene's star expression.

9. The computer-implemented method of claim 8 wherein said step of applying said refinement process to said data comprises:

15 i) determining a rules among said plurality of third set rules based on annotated regular expression and said selected character;

ii) executing determined rules;

iii) generating a regular expression partial derivative and refinement operation;

iv) repeating steps (i)-(iii) for each character in said data; and

v) applying refinement operations to said regular expression

20 vi) updating said refined regular expression in said extraction pattern database.

10. The computer-implemented method of claim 9 wherein said refinement operation comprises:

25 state-extension where a literal is added to the regular expression and the last two literals in the regular expression are finalised;

transition-extension where a new transition is added to the regular expression;

state-finalisation where the last second literal of the regular expression is finalised;

and

no-change where no changes are made.

30

11. The computer-implemented method of claim 10 wherein said plurality of third set rules comprises:

state-extension, $i:l + \vec{l}$ refers to transition-extension, $+^\epsilon$ refers to state-finalisation, and id_L refers to no-change,

rule (EpsW) proposes a fix to epsilon-literal mismatch by adding a state-extension, in case where the current source location i is not in the domain of the refinement requirement,

rule (EpsS) ensures that $(i:r) \in R$ and r can consume the input literal l ,

rules (LitS1) and (LitW1) handle cases where the input literal matches with the regular expression l where a no-change refinement operation is added,

rules (LitS2) and (LitS1) update the user requirement environment R with $(i:r')$

where r' is the partial derivative of r and $(i:r) \in R$,

rules (SeqS1), (SeqS2), (SeqW1) and (SeqW2) handle cases where the regular expression to be refined is a concatenation,

rules (ChoiceS) and (ChoiceW) handle cases where the regular expression to be refined is a choice,

rules (StarS) and (StarW) handle cases where the regular expression to be refined is a kleen's star, and

rules (WordCon) and (WordEps) extend the refinement operation to support a sequence of literal input.

123. The computer-implemented method of claim 112 wherein said step of applying refinement operations to said regular expression comprises:

applying the refinement operation obtained to the input word w ;

dividing result from the application of the refinement operation obtained to the input word into two categories, namely $\bar{\theta}$ and $\bar{\theta}'$ where $\bar{\theta}$ contains refinement environments that makes resulting partial derivative r' final, where $\bar{\theta}'$ contains refinement environments that makes the resulting partial derivative r' non-final;

determine the refinement environment from $\bar{\theta} \cup \bar{\theta}'$ with the fewest changes to the original regular expression; and

applying the refinement environment with the fewest changes to the original regular expression.