Learning and Using Formal Language

Anthony Cox[1], Maryanne Fisher[2,a], Diana Smith[2], and Josipa Granic[2]

[1] *Faculty of Computer Science*
Dalhousie University
amcox@cs.dal.ca

[2] *Department of Psychology*
York University
mlfisher,dsmith,jgranic@yorku.ca

Keywords: POP-I.B. cognitive processes, POP-II.A learning styles, POP-VI.E computer science education

Abstract

Regular expressions are a convenient and simple notation for expressing the members of a regular language. This simplicity enables regular expressions to serve as models of more complex context-free programming languages. To examine techniques in teaching programming, we exposed first year students with no knowledge of regular expressions to two tasks. The tasks asked participants to identify occurrences of a given expression (matching) or to create an expression to describe a set of occurrences (creation). Matching and creation can be viewed as parallel to reading and writing, suggesting that as reading skills help to develop writing skills, then similarly, matching skills help to develop creation skills. We hypothesised that a practice effect exists and that exposure to the matching task before the creation task improves performance on the latter. However, in an experimental setting, this hypothesis is not supported. Practice on recognition tasks provided no measurable benefits and suggests that the learning of formal language differs significantly from that of natural language. To verify this finding in an alternative setting, we performed a second experiment to test the hypothesis that performance will be similar when locating errors in HTML and when creating HTML. This experiment revealed no significant performance differences. One explanation for these results is that novices lack a lexicon of concepts to which language constructs can be mapped, and therefore view formal languages as rule-based systems. Consequently, the skills used for learning natural language can not be applied. These findings also suggest that for formal languages there is a greater need for instructional interventions such as performance feedback.

Introduction

Regular expressions are a concise and elegant notation for the expression of regular languages. However, one can also view regular expressions from several other perspectives. Having both an alternation operator, |, and an iteration operator, *, regular expressions can be considered as highly simplified programming languages. Language theory supports this view, as regular languages are a proper subset of the context-free languages in the Chomsky hierarchy (Chomsky, 1959). Thus, regular expressions provide a means for studying programming, and programmer training, using a language that is less complex than traditional context-free programming languages.

In computer software such as `grep`, `vi`, and Perl, regular expressions are used to describe the targets of search operations. For this role, regular expressions provide a pattern description mechanism with pattern matches providing search solutions. Search is a form of information retrieval, where the search targets are the information that is to be retrieved. Hence, it is possible to measure the effectiveness of a search using the traditional information retrieval measures of *precision* and r*ecall*. This perspective permits the evaluation of regular expression use and formation using well validated measures and techniques.

[a] Moving to St Mary's University, Halifax, Nova Scotia, Canada as of 1 July 2004.

In this paper, we combine these perspectives to explore and measure pattern matching and pattern creation. Our experiments are based on the premise that expression matching and creation can be viewed similarly to reading and writing. This equivalence stems from the fact that matching, like reading, is the identification of well-formed language constructs and creation, like writing, is the formation of well-formed constructs. As reading skills are considered necessary for the development of writing skills (Salvatori, 1983), we explored the hypothesis that performing a matching task before a creation task will cause a practice effect, as evidenced by improved performance on the creation task.

This hypothesis stems from the belief that exposure to well-formed expressions in the matching task increases participant's familiarity with the pattern system and consequently improves their ability to create well-formed expressions. Just as reading provides exposure to well-formed sentences and allows learning of grammatical rules, pattern matching provides exposure to well-formed expressions and allows the learning of expression formation rules. In the prototype theory of Posner and Keele (1968) , reading matches the text being viewed (i.e. an exemplar) against stored rules and concepts (i.e. prototypes) and in doing so strengthens the ability to access the prototype. To test our hypothesis, we experimentally compared the regular expression pattern matching and creation skills of novices.

Although we are examining and measuring pattern matching and creation, it must be remembered that this research considers regular expressions as an instance of a formal language. Hence, the tasks can also be viewed as programming tasks in a highly simplified and very terse programming language. Additionally, regular expressions are an important tool in Unix environments, thus making them worthy of examination independent of their relationship to other languages.

It should be noted that we consider pattern recognition to be different from pattern matching. Recognition is the observation that a pattern exists while matching is the identification of the specific pattern in existence. Recognition precedes matching since it is not possible to match a pattern that one can not see.

The remainder of this paper is organized as follows. A brief overview of regular expressions in the context of formal language theory is first provided. Then, we present and discuss our survey instrument, performance measures and the pilot study used to validate the survey and measures. The following section details experimental application of the survey and our findings. A supplementary HTML-based experiment is then described to demonstrate and verify that our findings can be applied in an alternative setting. Finally, the paper concludes with the presentation of some future research directions.

Regular Expressions

The Chomsky hierarchy of languages (Chomsky, 1959) orders languages into four classes identified by number. Each class is properly included in all higher numbered classes giving Class 0 the most languages and Class 3 the least. These language classes can also be identified using alternative names: Recursively Enumerable or Phrase Structured (Class 0), Context Sensitive (Class 1), Context Free (Class 2) and Regular (Class 3). Every language can be defined by a grammar or set of rules that describe valid constructions in the language. Formally, a grammar, G, is a four-tuple (N, T, P, S) where N and T are finite sets of non-terminal and terminal symbols, respectively. The set P is a set of productions (rules) describing the combination of the terminals and non-terminals, and S is a distinguished non-terminal called the *start symbol*. It is common to consider T as the alphabet, Σ, of the language. The language defined by G, $L(G)$, is the set of well-formed sentences (or expressions) that can be formed from Σ using a derivation rooted at S.

There are alternative representations of a language in addition to its grammar. It has been shown that every regular language can be described by a regular expression (Hopcroft and Ullman, 1979). Regular expressions are formed by combining elements of Σ using three operations: concatenation, alternation and repetition. Figure 1 provides a definition for well-formed regular expressions.

If *A* and *B* are well-formed regular expressions over an alphabet Σ :

1. *a* is a regular expression if $a \in \Sigma$.
2. *AB* is a regular expression. (Concatenation)
3. *A*|*B* is a regular expression. (Alternation)
4. *A** is a regular expression. (Repetition)
5. (*A*) is a regular expression. (Parenthesis)

Figure 1: Well-Formed Regular Expressions

```
1: ab          =  { ab }
2: a * b       =  { b, ab, aab, aaab, ...}
3: ab*         =  { a, ab, abb, abbb, ...}
4: (ab) *      =  { λ , ab, abab, ababab, ...}
5: a|b         =  { a, b }
6: a(b|c)d     =  { abd, acd }
7: (ab)|(ac)   =  { ab, ac }
8: (a|b)*      =  { λ, a, b, aa, ba, ab, bb, aaa, baa, aba, bba, aab, bab,
                       abb, bbb, ...}
```

Figure 2: Example Regular Expressions and Their Languages

Concatenation appends two regular expressions and is the mechanism by which longer expressions are built from shorter ones. Alternation is a selection mechanism with the expression *A*|*B* indicating a choice in selecting either the expression *A* or the expression *B*. Repetition (the Kleene closure) describes the set of zero or more successive occurrences of an expression. Parentheses allow expressions to be grouped and the group used as an operand. Figure 2 provides examples of well-formed regular expressions and their associated regular languages over the alphabet $\Sigma = \{a, b\}$.

In Figure 2, it can be seen that λ , the empty string, is a valid member of some languages. To avoid issues with coding results containing empty strings, we utilized a modified version of regular expressions that replaces the *, zero or more, operator with the +, one or more, operator. This change, apart from excluding λ as an element of any regular language, can be proven to have no effects on the expressivity of regular expressions (Clarke and Cormack, 1997).

Measurement of Regular Expression Usage

Regular languages are simple enough to be easily defined but provide sufficient expressibility for describing the targets of searches. It is for this role that regular expressions are best known in the field of computer science. Another mechanism for specifying search targets is Boolean algebra, as used in many information retrieval and WWW search tools. Boolean algebra also provides an alternation (or) operator, but replaces concatenation with a conjunction (and) operator. The repetition operator does not exist in Boolean algebra, but a negation (not) operator is available. Human performance when using Boolean algebra to specify search targets has been studied by Green *et al.* (1990) . A brief examination of the similarities between Boolean algebra and regular expressions is provided in the final section.

Performance Measures

To evaluate performance, we adopted the measures of precision and recall used in the discipline of information retrieval to quantify the accuracy and completeness of retrieval tasks. Precision and recall have been previously used to measure performance of Boolean search specifications (Turtle, 1994). Using precision and recall, it is possible to evaluate the performance of participants on both matching and creation tasks thereby permitting the tasks to be experimentally compared.

For a search that returns a set of solutions, S , where C is the complete set of possible solutions, the precision of the search, and hence of the search specification, is defined as:

$$Precision = \frac{|S \cap C|}{|S|}$$

The notation $|S|$ is used to identify the cardinality or size of the set S. Precision measures the fraction of the search results that are accurate or correct.

Recall measures the completeness of the search result and is the fraction of the correct results with respect to the total possible results. Recall is defined as:

$$Recall = \frac{|S \cap C|}{|C|}$$

To accurately employ these measures, we explored the effects of using different granularities for recording performance. It is possible that evaluating match solutions at the character level is under-sensitive since single character errors may not significantly affect results. Conversely, evaluating solutions as a whole may be overly-sensitive with respect to single character errors. Thus, as part of the experiments, we statistically compared measurements of precision and recall at both the character-level and substring-level. We hypothesised that character-level and substring-level measurements will significantly correlate indicating that the granularity of a correct solution does not adversely impact the measurement of a participant's performance.

The Survey

In order to explore expression matching and creation, we developed a four part, paper-based survey. *Part 1* is an instructional sheet provided to explain the formation of regular expressions. The instructional sheet was not taken from the participants and the experimenter suggested that participants consult the sheet for reference when completing the survey.

Part 2 of the survey was the pattern matching task. Participants were instructed to underline all occurrences of a pattern in a given string of characters. There were 10 items in the task, each having a different pattern and string. Figure 3 provides an example of an item. Participants were instructed to attempt each item in sequence and to not return to previously attempted items.

1. Pattern: `bg`
   String: `acdbggbcgbgbedccdfabagabadefbgcccfeedbbbbbgcbabcdgcef`

Figure 3: Survey Matching Task Item

Part 3 of the survey was the pattern creation task. Participants were given a written description of a search solution and instructed to generate a regular expression for which its regular language matched the search solution. For the last 7 of the 10 items, examples of some possible matches were provided to supplement the written description. An example of a creation task item can be found in Figure 4.

1. A sequence of `c`'s containing one `f` and that begins and ends with a `c`.
   e.g. `cfc, ccfc, cfcc, cccfc, ccfcc, cfccc,...`

Figure 4: Survey Creation Task Item

Again, participants were encouraged to work on the items in the order they were presented. To explore the primary hypothesis and determine whether an order effect was present, the survey was counter-balanced with half the participants performing the matching task first and the other half performing the creation task first.

In *part 4*, all participants were asked to answer a few demographic, exclusion identification and follow-up questions. The demographic data collected included the age and sex of the participants. The

exclusion questions identified the participants' field of study and their familiarity with regular expressions. The follow-up questions were used to examine participants' satisfaction with the instructional sheet and their opinions of task difficulty.

When generating precision and recall scores, the granularity used to determine the size of the solution has a direct affect on the generated score. For example, given the string:

`xxxyzzzxxxyzzxx`

and the pattern `xyz,` the response:

`xxxyzzzxxxyzzxx`

has a precision of `0.857`at the character level (6 of 7 characters correct) and a precision of `0.5` at the substring level (1 of 2 solution substrings correct). The term substring is used to indicate that match elements are substrings of the data string. To explore the relationship between these granularities, precision and recall values were calculated at both the character and substring level.

The generation of precision and recall values for the matching task is accomplished by counting the number of underlined and correctly underlined solutions and forming the appropriate ratios. For the creation task, the created strings were applied to a set of arbitrarily constructed "representative strings" and the precision and recall values calculated. The representative strings were generated by the same experimenter as the data strings of the matching task with the intent that both sets of strings contain similar character orderings and constructions. It is assumed that the application of the created expressions is error-free, as it was performed by an experimenter highly experienced in the use of regular expressions.

Pilot Study

To test the survey and validate the experimental measures, we conducted a pilot study using 36 volunteers from various classes in the Department of Psychology at York University, Canada. This sample included 9 males (age in years,*M* = 21.11) and 27 women (*M* = 20.15, *SD* = 2.11) excluding five surveys we omitted due to incompleteness or a clearly indicated lack of task comprehension. All participants reported that they had no previous experience using regular expressions, and debriefing revealed all were naïve of the experimental hypotheses.

Before beginning the survey, participants completed a consent form and, upon finishing the survey, participants were debriefed. All participants were tested individually, under conditions of anonymity. The pilot study was timed, with participants having three minutes to study the instructional sheet, five minutes to complete the matching task and five minutes to complete the creation task. Participants were encouraged to utilize all allotted time to the best of their ability.

Results of the Pilot Study

Due to the number of comparisons, we adopted a conservative significance level of p=0.1 reduce the possibility of creating a Type I error. As the direction of some of the hypotheses is unspecified, all reported analyses are two-tailed.

To validate the measures we utilized two scoring procedures to determine the effects of solution granularity on performance measurement. Paired-samples *t*-tests were conducted for precision and recall scores at the character and substring levels. Individual mean performance on character precision was significantly higher than substring precision, *t*(35)=12.82, p<0.001. Character precision yielded *M* = 0.88 (*SD* = 0.08) whereas substring precision yielded *M* = 0.69 (*SD* = 0.12) Individual mean character recall was also significantly higher than mean substring recall, *t*(35)=13.67, p<0.001; *M* = 0.74, *SD* = 0.12, and *M* = 0.59, *SD* = 0.14, respectively. In addition, we conducted paired-samples correlations to ensure that performance at the character level was related to performance at the substring level. For precision, character and substring performances were significantly correlated, *r* = 0.67, p<0.001. There was a corresponding finding for recall, as character and substring performances were significantly correlated, , *r* = 0.89, p<0.001.

The strong correlation between precision or recall values at the character and substring level indicates that either granularity can be used to measure performance. As expected, the values at the substring level are lower than those for the character level due to the fewer number of solutions at this level and the sensitivity of the solutions to single character errors.

The relationship between pattern matching and creation tasks was examined by collapsing the data across granularity and performance measures to generate an overall mean matching and creation value for each participant. The paired-samples *t*-tests yielded a significant difference, *t*(35)=-3.71, p<0.001. Creation scores were significantly lower than matching scores; *M* = 0.67 (*SD* = 0.16) and *M* = 0.78 (*SD* = 0.11), respectively. Furthermore, the scores were not significantly correlated, paired-samples correlation *r*=0.17, (not significant).

It is important that the matching and creation tasks are similar in difficulty. Significant differences in scores, as we found, suggest that the two tasks are not equal, thus confounding the comparison of skills between the two tasks. We had also expected to find that matching and creation scores would correlate, as both tasks tested participants' ability to apply expression formation rules. The lack of a correlation was surprising and caused us to re-evaluate our survey.

Examination of the completed surveys revealed that participants had considerable difficulty in creating expressions. Further examination and consultation with experienced regular expression users indicated a belief that the creation task was much more difficult than the matching task. The number of operators used in expressions for the matching task (26) is lower than for ideal solutions in the creation task (33). The number of alphabet symbols used in matching task expressions (27) is also lower than for creation task expressions (44). As a consequence, we modified the survey to decrease the difference in difficulty between the tasks.

Revised Survey

In the revised survey, the timing restrictions were removed and participants were permitted as much time as they desired for each section. As with the pilot survey, the tasks were counter-balanced and administered in the reverse order to one half of the participants. The instructional sheet was improved in accordance with the anecdotal reports obtained from participants during debriefing. The primary change was the inclusion of the example suite of Figure 2, modified to support + for * . Other changes included minor improvements in wording, additional instruction on the use of parentheses and deletion of the task alphabet definition.

The matching task was structured similarly, but the creation task was modified to be more like the matching task. Figure 5 provides an example of a revised creation task item. The modified creation task presents participants with an underlined string, where the underlined portions represent the solutions to the regular expression that the participants must generate.

For both matching and creation, the first six items are all structurally identical to an element of the example suite on the instructional sheet. Both tasks use the same six items but with the order varying. The remaining four items do not appear on the instructional sheet and can be considered as slightly more complex. The number of operators on both tasks is identical, although the creation task expressions have three more alphabet symbols. The revised and improved survey allowed us to conduct our primary experiment.

Main Experiment

The primary role of this experiment was to determine the effects of task ordering. As previously stated, it was hypothesised that performing matching first will have a positive influence on the creation task, and correspondingly, performing creation first will have little or no influence on the matching task. This hypothesis is based on the premise that exposure to well-formed expressions during matching affords participants an opportunity to learn the rules of expression formation.

To supplement our primary hypothesis on task ordering, we also investigated three additional hypotheses. First, we explored the relationship between precision and recall. Do participants use a conservative strategy and improve precision at the expense of recall, or an aggressive strategy that improves recall at the expense of precision? For example, the omission of a suspect, but correct solution, will have no effect upon precision, but will lower recall.

Second, we predicted a relationship between performance on pattern matching and pattern creation tasks. We believe that the two tasks utilize the same cognitive skill set. As suggested by Lunsford (1978) , all language skills are related such that one's level of reading comprehension is directly related to one's ability to form and manipulate syntactic structures. However, as suggested by Salvatori (1983), although the two skills are related, writing skill comes as a consequence of reading, and therefore builds on reading skills. Thus, we hypothesised that matching and creation scores will positively correlate but that creation scores will be significantly lower than matching scores.

Third, we examined the difference between the + and | operators. It is known that in Boolean algebra the alternation operator is more difficult to use than the conjunction operator (Greene et al., 1990). We believed that this effect will also appear in the context of regular expressions and will be evidenced by lower scores on items containing alternation, | , than on items containing repetition, * . This finding would imply that working with conditional constructs is more difficult than working with looping constructs.

Participants

Participants were solicited from various community locations in Toronto, including a manufacturing company, business office, retail outlet, athletic facility, restaurant, and hospital. There were a total of 64 participants in the final sample; 30 men (age, in years, *M* = 25.90, *SD* = 9.28) and 34 women *(M* = 24.70, *SD* = 8.56). One participant was excluded as he had previous experience with regular expressions, and three were excluded due to survey incompleteness or misunderstanding of the tasks. Participants' educational history, ethnicity, and socioeconomic status were diverse. All participants had no previous experience using regular expressions and debriefing revealed all were naïve of the experimental hypotheses.

Method

As the participants are from a community-based sample, the recruiting procedure was different to that in the pilot study. Participants were approached by female experimenter and asked if they would mind participating in a study on pattern and language formation. The remainder of the procedure was similar, apart from the use of the modified survey and the removal of any timing restrictions. Conditions of anonymity were maintained using the protocol of the pilot study.

Results

Due to the number of comparisons, we employed a conservative significance level of p=0.1 to reduce the possibility of creating a Type I error, and all reported analyses are two-tailed. There were four hypotheses for the experiment. First, we hypothesised an order effect and an increase in performance when matching is done before creation instead of after. Second, we predicted that there is a relationship between precision and recall and that this relationship identifies a search strategy. Third, we predicted the existence of a relationship between pattern matching and creation abilities. Lastly, we hypothesized that there is a difference in performance between items containing alternation and repetition. For additional validation, we examined the effect of scoring granularity on precision and recall values.

To verify our primary hypothesis and explore an order effect between matching and creation, we performed a repeated measures Analysis of Variance, with task (matching vs. creation) as the within-subjects variable and survey version (matching first or creation first) as the between-subjects variable. There was no main effect for task, *F*(1,62)= 0.719, NS, or version, *F*(1,62)= 1.76, NS. The interaction of task and version was also insignificant, *F*(1,62)= 0.155, NS.

The relationship between precision and recall was examined by collapsing the data across task and granularity to generate an overall mean for each measure. A paired-samples *t*-test resulted in significant differences, *t*(63)=5.83, p<0.001. We found participants' precision values to be significantly higher than their recall values; *M* = 0.78 (*SD* = 0.14) and *M* = 0.75 (*SD* = 0.14), respectively. Precision and recall are significantly correlated, paired-samples correlation *r* = 0.94, p<0.001.

The possibility of a relationship between pattern matching and creation was investigated by collapsing the data across granularity and performance measures which generated an overall mean for each task. Contrary to the pilot study, a paired-samples *t*-test did not yield significant results, *t*(63)=5.83, NS. Also in contrast with the pilot study, there was a significant relationship between matching and creation as revealed by a paired-samples correlation, *r* = 0.64, p<0.001.

Finally, we examined the differences in performance on items containing alternation or repetition in the creation and matching tasks. For the creation task, paired-samples *t*-test indicated significant differences between alternation and repetition items *t*(63)=-3.09, p<0.01. Alternation items resulted in lower values than repetition items, *M* = 0.71 (*SD* = 0.33) ND *M* = 0.83 (*SD* = 0.21) respectively. We also compared alternation items with items containing both alternation and repetition, *t*(61)=0.06, NS. A final comparison of repetition items with items containing both alternation and repetition revealed a significant difference, *t*(61)=-3.35, p<0.01. Items with both operators resulted in significantly lower values than alternation items, *M* = 0.70 (*SD* = 0.28) and *M* = 0.84 (*SD* = 0.20), respectively.

The same pattern emerged for the matching task. A paired-samples *t*-test yielded significant differences between alternation and repetition items, *t*(62)=-3.75, p<0.001. Alternation resulted in significantly lower scores (*M* = 0.72, *SD* = 0.19) than repetition (*M* = 0.82, *SD* = 0.18). Comparison of alternation with items containing both repetition and alternation revealed no significant difference, paired-samples *t*(63)=5.83, NS. Comparison of repetition with items containing both repetition and alternation resulted in significant differences, paired-samples *t*(39)=5.29, p<0.001. Repetition resulted in higher values than items containing both operators, *M* = 0.82 (*SD* = 0.17) and *M* = 0.68 (*SD* = 0.19), respectively.

A paired-samples *t*-test was used to examine the possibility of differences in performance due to granularity for both precision and recall measures. Individuals' character precision was significantly higher than their performance on substring precision, *t*(63)=13.87, p<0.001. Similar to the pilot study, character precision resulted in *M* = 0.86 (*SD* = 0.10) in contrast to substring precision that resulted in *M* = 0.70 (*SD* = 0.17) . Likewise, mean character recall was significantly higher than substring recall, *t*(63)=14.01, p<0.001. Character recall yielded *M* = 0.81 (*SD* = 0.12) whereas substring recall yielded *M* = 0.68 (*SD* = 0.17). Paired-sample correlations revealed significant relationships between character and substring precision, *r*=0.88, p<0.001, and between character and substring recall, *r*=0.94, p<0.001

Discussion

Surprisingly, our primary hypothesis of improved creation ability after the matching task was not confirmed. There is no significant order effect for the matching and creation tasks. This finding indicates that, without additional intervention or feedback, performance of either task does not provide sufficient opportunity for measurable learning to occur. In other words, exposure alone does not assist programmers in learning to manipulate a formal language.

The lack of an order effect between matching and creation provides evidence that measurable learning is not occurring during either task. This lack does not stem from difficulties with the instruction sheet. The modifications to the instruction sheet changed the participants' reported satisfaction 44.4% (pilot study) to 70.4% (main experiment). Anecdotal reports during debriefing indicate that the addition of an example suite was the primary cause of this increase. As well, some participants scored very highly on both tasks, indicating that they are strong learners that can readily master the formation of regular expressions from a one page instructional sheet. Hence, the non-significant difference is not a result of the tasks being too difficult to learn or perform. That is, it is not that the participants are weak learners, but that the tasks do not provide opportunity for learning to occur.

Pinker (1999) observes that children's sentences are often ungrammatical in some way and that in response, parents focus on content, not form, when providing correction. This observation identifies an important phenomenon – as humans we think, and hence focus, on concepts and not on the syntax in which these concepts are expressed. Thinking in concepts permits us to map multiple languages to a single set of mentally maintained concepts, and hence, to speak multiple languages or to think about concepts for which we have no words, and thus, invent new concepts.

Similarly to natural languages, formal languages must also be mapped to a set of mentally maintained concepts.

The text matched to the regular expressions in these surveys is effectively ‘contentless’ as each pattern identified letter combinations that had no meaning to English speakers. Thus, the only available content on which to build a mapping are the formation and matching rules associated with regular expressions. These rules are highly abstract and are not likely to map to most participants’ existing set of mental concepts. Consequently, it is likely that a simple one-to-one mapping of rules to concepts was used, where each rule was viewed as a new and independent concept. This mapping reduces pattern matching and creation to simple ‘mechanical’ rule application. Thus, the tasks measure participants’ ability to learn and apply rules, not their ability to read and write a language.

Support for this perspective can be obtained by examining the experiments of Ledgard et al. (1980) . They report that participants had significantly better performance when expressing search targets using English than when using a “notational query language”. Even when participants had experience using three distinct editors, they still had difficulty formulating queries in a new syntax. This effect can be interpreted as resulting from participants’ using newly generated syntax to concept mappings, where the mappings do not exploit concepts associated with previous editor use. That is, when learning formal language, mappings are syntax-specific and independent of more abstract concepts.

In the aforementioned study of editing (Ledgard et al., 1980), Whiteside, one of the paper’s co-authors, comments that “users made no distinction between syntax and semantics.” Participants equated function with syntax, providing evidence that a simple rule to concept mapping is being used. It is also noted that participants reacted with surprise when told that the two editors they used were functionally equivalent and only differed in command syntax. From our findings, we suggest that programmers learn a formal language by initially generating a set of syntax to concept mappings.

We believe that an order effect does not occur since participants are simply applying the formation and matching rules without a deep understanding of what the rules mean. The focus on syntax prevents participants from seeing the patterns that the rules describe, and hence from learning to recognize the patterns. The absence of measurable learning establishes an important baseline for the performance of future experiments in which the framework established here can be used to measure the effects of various pedagogical methodologies. As well, the findings indicate that mere exposure to well-formed expressions provides little help in understanding their use and formation. Instruction, such as tutorials and laboratory exercises, along with performance feedback is likely required before learning can occur.

For the first supplementary hypothesis, it was found that the recall scores of each participant are lower than their precision scores. We believe that this effect can be partially attributed to the test instrument. In the survey, it was observed that many participants successfully identified all but one of the possible solutions for a particular item. It is likely that these errors are the result of simple oversight and not caused by an inability to identify a correct solution. One explanation could be that the participants experienced a form of repetition blindness (Kanwisher, 1987) when multiple identical solutions appeared close together.

As precision and recall positively correlate, there is no indication of any individual strategy being used. No evidence exists for the use of an aggressive strategy favouring recall over precision. For example, an aggressive participant could have raised all her matching task, character level, recall scores to 1.0, by simply underlining the entire data string. This strategy would significantly lower her precision scores as a result of generating many invalid solutions. The fact that recall is significantly lower than precision indicates that a conservative strategy is consistently used and that accuracy is favoured over completeness. While strategy differences may exist, they are displayed with respect to the amount of conservatism a specific participant employed. This finding is consistent with the theory that participants are mechanically applying expression formation rules and consequently do not have the skills to develop and employ a strategy. It may also be possible that participants are conscientious in their completion of the surveys and tend to err on the side of caution. As an aside, it should be noted that the high means reported for precision and recall are a result of the survey design. The initial task items are intentionally easy and are intended to build participant's confidence and improve compliance.

There was no significant difference in the means for the matching and creation tasks. This finding indicates that the tasks are much more equivalent in difficulty than for the pilot study. Once the difference in task difficulty was removed, performance on matching was found to correlate with that of creation. This correlation is indicative of a common skill set being utilized for both tasks. The lack of an order effect also indicates the lack of a practice effect where the first task provides practice for the second. We believe that the lack of feedback given after the first task prevented individuals from improving their skill in expression manipulation. Future research is needed to explore the consequences of feedback; specifically, the type and duration of feedback that is most beneficial for learning. Investigation is also required to determine other aspects of this cognitive ability. For example, what is the relationship of pattern recognition to pattern matching and creation? Does pattern (or program) comprehension require the ability to form patterns, or is it based on matching and recognition skills?

Our results confirm the hypothesis that alternation is more difficult than repetition or concatenation. Participants had significantly lower performance on items containing the 'or' operator. This effect is observed in both the matching and the creation task. To ensure that the alternation operator is the cause of the effect we divided items into three groups: those containing only the alternation operator, those containing only the repetition operator and those containing both. The results indicate that there was a difference between the repetition and alternation group and between the repetition and both operator group. However, there was no difference between the alternation and both operator group. This finding indicates that it is the presence of the alternation operator that is responsible for the difference and not some form of operator interaction.

The replication of correlation between character level and substring level measures provides additional evidence of the interchangeability of the two scores. Future researchers may utilize either recording technique depending upon the granularity their studies require. However, it should be noted that the high sensitivity of substring level scores and the associated lower mean may obscure small variations in performance.

```
<htm><head>
   Sample Web Page
<body></head>
   <h1> Useful Links </h3>
   <ol>
       <li><a href="www.google.com">Internet Search Tool</a>
       <li><a href="canada.gc.ca">Government of Canada</a>
       <li><a href="www.theweathernetwork.com">The Weather Network</a>
</body></htm>
```

Figure 6: Erroneous HTML File

Supplementary Study

We performed a second, supplementary, experiment to verify the findings of our primary experiment in an alternative setting. Novice programmers, with minimal experience, were asked to locate errors in an HTML file and to generate an HTML file. It is hypothesized that, just as expression matching and

creation are applications of the same skill, so is the reading and writing of HTML. We selected HTML since the natural language content of HTML files should be readily understood by all participants. Hence, task performance should be mediated by the participants' skill with HTML and not their skill in manipulating the natural language content.

Participants

Participants were students in a first-year computing course for the life-sciences. A total of 22 participants was included in the final sample; 10 men (age, in years, *M* = 20.46, *SD* = 2.78) and 12 women (*M* = 19.22, *SD* = 2.08). No participant had undergone previous instruction in the use of HTML and debriefing revealed all were na¨ıve of the experimental hypotheses. Prior to recruitment, students received equivalent levels of classroom instruction on HTML and had completed an assignment requiring creation of a web page.

Method

The paper-based survey consisted of three parts: a demographic questionnaire, an error identification task and a web-page creation task. The demographic questionnaire was structured similarly to that of the previous experiment.

In the error identification task, participants were instructed to locate and correct errors in the HTML file of Figure 6. The file contained 5 errors, all of which were syntactic in nature. In the web-page creation task, participants were asked to create, on paper, a small web-page. The description of the page is as follows:

> Because of a hardware failure, your personal web page was deleted. You are to write a small web page to replace the one that was lost. Your web page should have your name as the title at the top and display the image `mypicture.gif`. In the page there should be an unnumbered list of 3 helpful hyperlinks. Invent fictitious, but correctly-formed URLs for the hyperlinks.

The differences between the tasks prevent precision and recall from being used as performance measures. Instead, an experimenter blind to the experimental hypotheses was asked to treat each task like an exam question and assign it a 'mark' out of 5. The marking scheme for the error location task gave one mark for each error found and corrected, less one mark for each non-existent error. The marking scheme for the creation task gave one mark for each of the following items: name in title, image displayed, unnumbered list present, URLs well-formed, and `html/head/bodytags` present and correct. A mark was subtracted for any other unspecified error (e.g. missing `<li>` tags). No mark below zero was assigned.

It was hypothesized that error correction and page creation are equivalent to expression matching and creation. Error correction and expression matching are similar, as both require application of syntax rules to identify elements of a target text. Page creation and expression creation are similar in that both require participants to apply formation rules to generate well-formed syntactic entities. Hence, just as expression creation and matching are not significantly different and are correlated, we predicted equivalent results for error correction and page creation.

Results

A paired-samples *t*-test was used to test the hypothesis that participants would have similar performance in the error location and page creation tasks. The was no significant difference between locating errors and page creation, *t*(63)=0.00, NS. In fact, the scores were, on average across all participants, equivalent between the two conditions, *M* = 4.23 (*SD* = 0.75 for error location, 0.81 for page creation).

Discussion

The follow-up experiment demonstrates that novice programmers have no difference in performance on tasks that manipulate existing code and tasks that require the writing of code. It is likely that, due to the syntactic nature of HTML, participants apply the formation rules of the language in a mechanical fashion, independent of the content of the web-page. This finding supports the hypothesis that programmers have difficulty mapping abstract language syntax rules to existing mental concepts and consequently use each rule to generate a new concept. Tasks involving the language then become mechanical in nature, as the new concepts are simply a mental representation of the rule. In general, these results support the idea of 'syntax to concept' mapping and demonstrate its generality for languages other than regular expressions.

Conclusions and Future Work

In these studies, we found no order effect for matching and creation tasks. This finding suggests that novice programmers use a simple syntax to concept mapping that reduces tasks to mechanical rule application. As well, participants uniformly used a conservative strategy that favoured precision over recall. Matching and creation scores significantly correlated, implying the use of the same cognitive skill set. Finally, it was found that, as is the case in Boolean algebra, participants were least effective when using the 'or' operator.

The lower recall performance on matching tasks, as a result of missed solutions, is of interest for further research. We believe that some form of repetition blindness is occurring and that performance is affected by this phenomenon in addition to participants' skill level. Future research will explore this possibility by examining the locations of missed solutions relative to similar character sequences.

We find it curious that there is no evidence of strategy use. One potential explanation is that different tasks would create situations involving the necessity to make tradeoffs, but since our study did not impose the need for a strategy participants did not use one. For example, as there was no need to generate a small and accurate solution set, as is desired when using a WWW search tool, users did not attempt to do so.

Pane and Myers (2000) explored the issue of pattern creation and matching in the context of Boolean algebra. They reported no difference in matching performance as a result of the format of a test item. While the use of a textual and a diagrammatic expression format had no effect on matching performance, it did significantly affect creation performance. Using a correct vs. incorrect scoring system, they found that creation is an easier task than matching. Participants in their study answered 72.5% of the matching tasks correctly and 89.5% of the creation tasks correctly, when averaged over both expression formats. No explanation was offered for their finding. In contrast, we obtained equivalent creation and matching scores. It is possible that they utilized a creation task that was easier than the matching task, or that the scoring system lacked sufficient granularity to accurately measure performance. This apparent discrepancy in reported findings requires further investigation.

It is not surprising that participants had more difficulty manipulating expressions with alternation than those without the operator since it is documented that a similar phenomenon occurs in Boolean query systems (Greene et al., 1990). While Vakkari (2000) reports that this effect decreases with improved conceptual representation of the search task domain, it is also possible that the reported improvement is due to improved skill in the use of a Boolean system. In addition, Vakkari describes the use of alternation as a "parallel search tactic" due to the need to simultaneously identify solutions for both elements of the construct. The data of Green *et al.* (1990) supports this concept of parallelism. Participants in their experiment took twice as long, 44.8 vs. 24.4 seconds, on queries with disjunction alone as compared to conjunction alone. Chui and Dillon (1999) suggest that this effect is the result of a greater level in difficulty for processing disjunctive information. This explanation is supported by Johnson-Laird (1983) who postulates that human processing of logical syllogisms is limited in the number of alternative models that can be simultaneously maintained in working memory. When working memory is depleted, processing will have to be performed sequentially, increasing the time needed to solve a task. Perhaps this effect is stronger in novices, as they may utilize working memory less efficiently while developing their cognitive skills.

In future work, we intend to explore the need for sequential solution of expressions containing alteration by using a timed study. If parallel processing of alternation expressions is occurring, matching of these expressions should be similar in time to that of repetition expressions. However, if sequential processing is used to match alternation expressions, the time to perform a matching task will increase with the number of alternation operations. Therefore, we intend to perform a timed task to verify the hypothesis that alternation tasks are solved sequentially.

The follow-up experiment provides support for the hypothesis that programmers use a simple rule-to-concept mapping when learning the syntax of a new programming language. As Ledgard *et al.* observed, even experienced programmers have difficulty using an unfamiliar system. Thus, it seems likely that programming languages, unlike natural languages, are learned as a set of rules, not as a rich set of semantic concepts. When learned as rules, additional learning is slow to occur and is not a product of exposure to well-formed examples of rule application. The lack of a practice effect, manifested as an order effect, verifies the minimal, and in this framework immeasurable, learning that is occurring.

While the research presented here has begun the exploration of the cognitive skills needed to manipulate regular expressions and other formal languages, there is still much to be done. In future research, we will continue the line of experimentation started here and to answer some of the questions that have been raised.

References

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, 2(2):137–167.

Chui, M. and Dillon, A. (1999). Speed and accuracy using four boolean query systems. In *10th AAAI Midwest Artificial Intelligence and Cognitive Science Conference*, pages 36–42, Bloomington, Indiana.

Clarke, C. and Cormack, G. (1997). On the use of regular expressions for searching text. *ACM Transactions on Programming Languages and Systems*, 19(3):413–426.

Greene, S., Devlin, S., Cannata, P., and Gomez, L. (1990). No IFs, ANDs, or ORs: A study of database querying. *International Journal of Man–Machine Studies*, 32(3):303–326.

Hopcroft, J. and Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, Reading, Massachusetts.

Johnson-Laird, P. (1983). *Mental Models*. Harvard University Press, Cambridge, Massachusetts. Kanwisher, N. (1987). Repetition blindness: type recognition without token individuation. *Cognition*,

27(2):117–143.

Ledgard, H., Whiteside, J., Singer, A., and Seymour, W. (1980). The natural language of interactive systems. *Communications of the ACM*, 23(10):556–563.

Lunsford, A. (1978). What we know—and don't know—about remedial writing. *College Composition and Communication*, 29:49–51.

Pane, J. and Myers, B. (2000). Improving user performance on boolean queries. In *ACM Conference on Human Factors in Computing Systems*, pages 269–270, The Hague, Netherlands.

Pinker, S. (1999). *Words and Rules*. Basic Books, New York, NY.

Posner, M. and Keele, S. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology*, 77:353–363.

Salvatori, M. (1983). Reading and writing a text: correlations between reading and writing. *College English*, 45(7):657–666.

Turtle, H. (1994). Natural language vs. boolean query evaluation: a comparison of retrieval performance. In *17th Annual International Conference on Research and Development in Information Retrieval*, pages 212–220, Dublin, Ireland. ACM SIGIR.

Vakkari, P. (2000). Cognition and changes of search terms and tactics during task performance. In *RIAO International Conference*, pages 894–907, Paris, France.