5

How Do Programmers Think?

Anthony Cox and Maryanne Fisher
Centre for Psychology and Computing, Saint Mary's University
Canada

1. Introduction

Regular expressions are an example of a regular language and are a subset of both the context free and the context sensitive languages. As the syntactic structure of many computer programming languages can be described using a context-free grammar, regular expressions can be viewed as a simplified and restricted programming language. While lacking many of the more sophisticated programming language features (e.g., types, functions), regular expressions still provide users with sequencing, alternation, and iteration constructs, and thus provide an abstract view of basic control-flow features. Concatenation can be regarded as a form of sequencing where the elements of an expression containing concatenation must occur in a specific, linear sequence. The "or" operator, represented with a vertical bar **|**, provides alternation and permits one to choose between two options, just as an "if-then-else" statement permits the choice of two alternatives. Finally, the Kleene closure, represented by a superscript asterisk ***** functions as an iteration operator and performs a role similar to looping constructs (e.g., "do", "while", or "for" in the C programming language). Thus, regular expressions can be viewed as simple programs with basic control-flow constructs, but with no explicit data management.

Alternatively, regular expressions are used in computer software, such as grep, vi, and Perl, as a mechanism to describe the targets of search operations. For this role, regular expressions provide a pattern description mechanism with pattern matches identifying desired search solutions. For example, the expression **[eE][nN][dD]** identifies the term "end" where the letters can independently be in upper or lower case (e.g., "eNd", "ENd", "END").

Regular languages, while being highly formalized and restrictive with regard to their expressiveness, are never-the-less a form of language. It has been documented that humans develop the ability to read before they develop the ability to write (Salvatori, 1983). Consequently, by comparing novice's skills as they learn to read (i.e., applying) and write (i.e., creating) regular expressions, one can examine the relationship of formal languages to natural languages (e.g., English) and potentially permit research on the use of natural language to be applied to computer programming. Increased understanding of regular expression use can therefore provide understanding of how we, as humans, interact with computers when using formal languages. That is, understanding the cognitive skills associated with lower level formal languages provides insight on the use of higher level imperative style languages such as Fortran, C++, or Pascal.

While there has been significant research on algorithms for automated matching and manipulation of regular expressions (Hopcraft & Ullman, 1979), there has been little research on the human element of these systems. Insight into the manipulation of regular expressions provides insight into the manipulation of formal languages, and hence on computer programming—a foundational task in human-computer interaction. In this chapter, we address this deficiency and explore the cognition underlying programming by examining performance on the manipulation of regular expressions.

The remainder of this chapter is organised as follows. A brief overview of regular expressions in the context of formal language theory is first provided. Then, we present the first of two studies that we conducted to investigate performance on expression application (i.e., matching) and creation tasks. The results of the second, revised, study are then presented, after which we describe a third study exploring the similarity between regular and Boolean expressions. Finally, the chapter concludes with an examination of some future research directions.

2. Regular Languages and Expressions

The Chomsky hierarchy of languages (Chomsky, 1959) orders languages into four classes identified by number. Each class is properly included in all lower numbered classes giving Class 0 the largest number of languages and Class 3 the smallest. In most of the literature, the language classes are identified by 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. The symbols that are combined to form valid constructions are known as the alphabet, Σ, of the language. Thus, given an alphabet and a grammar it is possible to decide whether a specified sequence of symbols is a member of the language described by the grammar. Furthermore, every regular language can be described by a regular expression (Hopcraft & Ullman, 1979). Regular expressions are formed by combining the elements of Σ using three operations: concatenation, alternation and repetition (i.e., the Kleene closure). Figure 1 provides a recursive definition for well-formed regular expressions.

Given an alphabet Σ where **a** ∈ Σ, **b** ∈ Σ:

(1)	**a**	is a regular expression	
(2)	**ab**	is a regular expression	(Concatenation)
(3)	**a|b**	is a regular expression	(Alternation)
(4)	**a***	is a regular expression	(Repetition or Kleene Closure)
(5)	**(a)**	is a regular expression	(Parenthesis)

Figure 1. Well-Formed Regular Expressions

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** but not both (i.e., exclusive or). Repetition describes the set of zero or more successive occurrences of an expression. Parentheses may be used to modify the order that operations are performed (i.e., precedence) or can be used to modify the scope of an operator's application. Every regular expression is equivalent to a grammar for the corresponding regular language and provides a mechanism for defining the language. Languages that are defined using only concatenation and alternation have a finite number of members while languages defined using repetition have an infinite number of members. Hence, for the alphabet, Σ = {**a**, **b**, **c**, **d**}, Figure 2 provides some examples of well-formed regular expressions and their associated regular languages.

(1)	**ab**	defines	{**ab**}
(2)	**a*b**	defines	{**b**, **ab**, **aab**, **aaab**, …}
(3)	**ab***	defines	{**a**, **ab**, **abb**, **abbb**, …}
(4)	**(ab)***	defines	{λ, **ab**, **abab**, **ababab**, …}
(5)	**a | b**	defines	{**a**, **b**}
(6)	**ab | cd**	defines	{**abd**, **acd**}
(7)	**(ab) | (cd)**	defines	{**ab**, **cd**}
(8)	**(a | b)***	defines	{λ, **a**, **b**, **aa**, **ab**, **ba**, **bb**, **aaa**, **aab**, **aba**, **abb**, **baa**, **bab**, **bba**, **bbb**, …}

Figure 2. Example Regular Expressions and Their Languages

In Figure 2, it can be seen that λ, the empty string, is a valid member of some languages. In our studies, we wished to avoid issues in coding results that contain λ. Consequently, we use 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 defined language, has no other effect on the expressivity of regular expressions.

Regular languages are simple enough to be easily defined but provide sufficient flexibility for describing the results of searches. It is for this role that regular expressions are best known in the field of computer science. Another mechanism for specifying search results is Boolean algebra, as used in many information retrieval and world wide web search tools. Boolean algebra also provides an alternation (i.e., or) operator, but replaces concatenation with a conjunction (i.e., and) operator. The repetition operator does not exist in Boolean algebra, but a negation (i.e., not) operator is available. The use of Boolean algebra to specify search results has been previously studied by Green et al. (1990).

Although Boolean algebra, a logical calculus for two valued systems, and regular expressions, a restricted class of formal language, are significantly different, the common use of an alternation operator and their application for similar roles provides a link between them. The studies presented here can be seen as a first attempt at examining the relationship between the skills used in the manipulation of each system. Study 3 uses performance times to explore this similarity.

To measure performance when manipulating (e.g., creating and applying) regular expressions, we adopted the information retrieval measures of *precision*, otherwise known as accuracy, and *recall,* which is also called completeness. Precision and recall have been previously used to measure performance of Boolean search specifications (Turtle, 1994). For a search that returns a set of solutions S, where C is the complete set of possible solutions, P the precision of the search, and hence of the search specification, is defined as:

$$P = \frac{|S \bigcap C|}{|S|}. \tag{1}$$

In the above equation, the notation |S| is used to identify the cardinality or size of the set S (i.e., number of members in 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. For the same set of solutions, R the recall of the search, is defined as:

$$R = \frac{|S \bigcap C|}{|C|}. \tag{2}$$

Specifically, our research addresses four distinct issues. First, we explore the effects of using different granularities for measuring precision and recall. It is possible that evaluating results at the character level is under-sensitive since small or single character errors may not significantly affect results. Conversely, evaluating solutions as a whole and thus at a higher level of granularity may be overly sensitive with respect to small errors. The exploration of multiple levels of granularity is intended to identify experimental results that can be attributed to overly sensitive, or conversely, insensitive measures. For example, given the string:

xxxyzzzxxxyzzxx

and the pattern **xyz**, the participant response, where the participant has underlined the matches to the pattern:

xx$\underline{\text{xyz}}$zzxx$\underline{\text{xyzz}}$xx

has a precision of .857 at the character level (6 of 7 characters correct) and a precision of .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 and then compared.

Second, we investigated the relationship between precision and recall to identify the use of specific strategies from an information retrieval perspective. We explored whether participants used a conservative strategy to improve precision at the expense of recall, or an aggressive strategy that improved recall at the expense of precision. For example, the conservative omission of a suspect, but correct solution, will have no effect upon precision, but will lower recall. We believe that regular expression use is more like natural language use than like information retrieval and will therefore demonstrate a consistent relationship between precision and recall that is not found when performing an information retrieval task.

As indicated by Salvatori (1983), writing skill can be increased by improving reading skill, but the two skills are not completely related. That is, writing and reading abilities may increase independently, which indicates their basis in different, but related, cognitive skills. Rouet (2006) classes reading comprehension as a restricted form of literacy that precedes the functional literacy needed to synthesise and express (i.e., write) ideas. Thus, it is likely that there is a relationship between pattern matching, which resembles reading in that an existing "sentence" must be understood, and pattern creation, which, like writing, requires the development of new sentences. We expect novice's matching ability to be better than their pattern creation ability in the same way that one's reading skills develop before one's writing skills.

Third, we therefore hypothesise that the incidental learning that occurs during each study will improve reading ability for regular expressions, but not writing ability, as writing skill develops more slowly than reading skill and requires more developed cognitive abilities (Salvatori, 1983). That is, participants may advance to the *interpreting* level of the Wilkinson Cognition Measure (Wilkinson, 1979) for expression matching, but not to the *generalising* level needed for accurate pattern formation.

In previous research, Ledgard et al. (1980) explored the hypothesis that making computer languages more like natural languages improves their ease of use, thus suggesting that the two types of language have some similarity that permits natural language skills to be employed when working with formal languages. However, Blackwell (2000) found that graphical notations for regular expressions exhibited improved usability over a conventional textual notation. Blackwell's finding indicates that, although there is a relationship between formal and natural language, the “content” of some formal languages (e.g., regular expressions) is better represented using other notations. On consideration of these findings, we believe that there is significant difference in the content of formal and natural language.

Fourth, it is known that in Boolean algebra the alternation operator is more difficult to use than the conjunction operator (Greene et al., 1990; Vakkari, 2000). We hypothesise that this effect will also appear in the context of regular expressions. Finding this effect would provide evidence of similar skills being applied when using the alternation operator, regardless of the context of use.

3. Study 1

Study 1 was our initial attempt at exploring the relationship between pattern application and pattern creation. The study provides some preliminary evidence that the manipulation of regular expressions is dissimilar to information retrieval.

3.1 Participants

Participants, from a diverse set of ethnic and socioeconomic backgrounds, were recruited as volunteers from various psychology classes at a major Canadian university located in a large metropolitan city. The final sample included 36 participants (age in years, M = 20.65, SD = 2.23), excluding 5 surveys we omitted due to a clearly indicated lack of task comprehension. We considered a participant as not understanding the task if he or she had less than 3 correct responses for 20 items. All participants reported that they had no previous programming experience, therefore mitigating any confounds introduced by prior experience or training in formal or programming language use.

3.2 Stimuli and Procedure

Participants were given a four-part survey. In part one, participants were given 3 minutes to study an instruction sheet that explained the formation of regular expressions. The instruction sheet was not taken from the participants and the experimenter suggested that it could be consulted for reference when completing the remainder of the survey.

In part two, participants were given 5 minutes to complete a 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 a matching item.

Matching	Pattern: **bg** String: **acdbggbcgbgbedccdfabagabadefbgcccfeedbbbbbgcbabcdgcef**
Creation	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 3. Sample Task Items for Study One

In part three, participants were given 5 minutes to complete a pattern creation task. Participants were presented with a written description of a search solution and asked to create a regular expression that matched the solution. For the last 7 (i.e., more complex) items, examples of possible matches were provided to supplement the written description. An example of a creation task item is shown in Figure 3. The order of presentation for the matching and creation tasks was counter-balanced, such that half of the participants received the matching task first, and the rest received the creation task first.

In part four, participants answered a few demographic and follow-up questions. The demographic items addressed the age and sex of the participants while the follow-up questions examined their satisfaction with the instruction sheet and opinions on the relative difficulty of the tasks.

The generation of precision and recall values for the matching task is accomplished by counting the number of attempted, and the subset of correct 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.

3.3 Results

There were three hypotheses for Study 1. First, we predicted a difference in performance based on the granularity of recording pattern matches. Second, we hypothesised the existence of a relationship between precision and recall measures. Third, we predicted a relationship between pattern matching and creation abilities. Due to the number of comparisons, we adopted a conservative significance level of $\alpha = .01$ to reduce the possibility of creating a Type I error. As well, because of the unspecified direction of some hypotheses, all reported analyses are two-tailed.

To test the first hypothesis, the possibility of differences in performance due to granularity, we conducted paired-samples *t*-tests for precision and recall scores at the character and substring levels. Individual mean performance on character precision was significantly higher than substring precision, $t(35) = 12.82$, $p < .000$. 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 substring recall, $t(35) = 13.67$, $p <.000$; $M = 0.74$, $SD = 0.12$, and $M = 0.59$, $SD = 0.14$, respectively. We additionally conducted paired-samples correlations to examine the possibility that performance at the character level is related to performance at the substring level. For precision, character and substring performances were significantly related, *r*(35) = 0.67, *p* < .000. There was a corresponding finding for recall, as character and substring performances were significantly related, *r*(35) = 0.89, *p* < .000.

The relationship between precision and recall was analysed by collapsing the data across task and granularity, thus generating an overall mean precision and recall value for each participant, which were significantly different; *t*(35) = 8.20, *p* < .000. Individuals' recall values were significantly lower than their precision scores; *M* = 0.67 (*SD* = 0.13) and *M* = 0.79 (SD = 0.09), respectively. In addition, there was a significant positive relationship between precision and recall; *r*(35) = .75, *p* < .000.

To examine the relationship between pattern matching and creation, we collapsed the data across granularity and performance measures to generate an overall mean matching and creation value for each participant. This comparison yielded a significant difference, *t*(35) = 3.71, *p* < .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 unrelated, *r*(35) = 0.17, *p* > .01.

3.4 Discussion

The correlations between the scores at the character and substring levels, for both precision and recall, indicate 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 as a result of the fewer number of solutions and the sensitivity of the solutions to small, single character errors.

It was found that the recall scores of each participant are significantly lower than their precision scores. This effect can be partially attributed to the testing instrument, as we observed that many participants successfully identified all but one of the possible solutions for a particular item. The effect is likely the result of simple oversight and not due to an inability to identify a correct solution. One explanation could be that the participants experienced a form of repetition blindness (Kanwisher, 1987) for multiple, adjacent solutions.

As precision and recall positively correlate, there is no evidence of significant variation in individual strategy. For example, an aggressive participant could have raised all their character level matching task recall scores to 1.0 by simply underlining the entire data string. This strategy would significantly lower their precision score as a result of generating many invalid solutions. The significantly lower score for recall than for precision indicates that a conservative strategy is consistently used by participants. It is likely that participants were conscientious in their completion of the surveys and tended to err on the side of caution. Alternatively, it is also possible that since the participants were students in an educational system where performance is measured using precision, they tended to focus more on precision than on recall. The second study uses a community sample to examine this possibility. It should be noted that the high means reported for precision and recall are a result of the survey design. The initial task items were intentionally easy and designed to build confidence for the purpose of improving compliance.

The consistent application of the same strategy does not match typical information retrieval behaviour. Individuals tend to show more variation in the trade-off between precision and recall than they did in this experiment. Thus, there is evidence that when using regular expressions, participants are thinking about the expressions and not the information that is being retrieved (i.e., search targets).

Although predicted, there was no correlation between the scores for matching and creation. Examination of the completed surveys reveals that participants had considerable difficulty in creating expressions. Furthermore, 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) was lower than for optimal solutions in the creation task (33). The number of alphabet symbols used in matching task expressions (27) was also lower than for creation task expressions (44). The significantly lower mean on the creation task than on the matching task provides support in the belief that creation is more difficult than matching.

4. Study 2

In Study 2 we aimed to replicate the findings from Study 1, as well exploring the differences between alternation and repetition. The survey used was a revised version of that used in Study 1, with modifications to increase the similarity of presentation between the matching and creation tasks.

4.1 Participants

Participants were solicited from various community locations in the same city as Study 1, and included a manufacturing company, business office, retail outlet, athletic facilities, restaurant, and hospital. There were a total of 64 participants in the final sample (age in years, *M* = 25.51, *SD* = 8.84) excluding 1 participant who had previous programming experience and 3 who demonstrated a clear misunderstanding of the tasks (i.e., matching or creation scores less than 3). Participants' educational history, ethnicity, and socioeconomic status were diverse.

4.2 Stimuli and Procedure

In Study 2, the timing restrictions were removed and participants were given as much time as they desired for each section. As in Study 1, the tasks were counter-balanced and administered in the reverse order to half of the participants. The instruction sheet of Study 2 was improved in accordance with the anecdotal reports obtained from participants during debriefing for Study 1. The primary change was the inclusion of an example suite similar to Figure 2. Other changes included minor improvements in wording, additional instruction on the use of parentheses and deletion of the task alphabet definition. The revisions were intended to permit participants to attain the *describing* level of the Wilkinson Cognition Model (Wilkinson, 1979).

The matching task was structured similarly to Study 1, but the creation task was modified to be more like the matching task. Figure 4 provides an example of a Study 2 creation task item. The modified creation task presents participants with an underlined string, where the underlined portions represent the solutions to an applied regular expression that the participants must generate.

Creation	String:	**zzuuxyxyzyzxzzxxyyzyxyxzzyxzyyyyyzyzzzyyzywyxxwuwu**
	Solution:	

Figure 4. Study Two Creation Task Item Sample

For both matching and creation, the first 6 items were structurally identical (i.e., the same operators arranged in the same order) to an element of the example suite on the instruction sheet. Moreover, both tasks used the same 6 items but with the order varying. The remaining 4 items did not appear on the instruction sheet and can be considered as slightly more complex. The number of operators in both tasks was identical, although the creation task expressions had 3 more alphabet symbols.

As the participants were from a community-based sample, the recruiting procedure was different than for Study 1. Participants were approached by a female experimenter and asked to complete a study on pattern and language formation. The remainder of the procedure was identical.

4.3 Results

There were 4 hypotheses for Study 2, with the first and second intended to replicate the findings of the first study. Therefore, we hypothesised a difference in performance due to the granularity of recording pattern matches, and a relationship between precision and recall. Due to the improved survey, we predicted a relationship between pattern matching and creation abilities that we did not find in Study 1. Lastly, we hypothesised a difference in performance on alternation items and repetition items. As in Study 1, we employed a conservative significance level of $\alpha = .01$ and all reported analyses were two-tailed.

A paired-samples *t*-test was used to examine the possibility of differences in performance due to granularity for both precision and recall measures. Similar to Study 1, individuals' character precision (*M* = 0.86, *SD* = 0.10) was significantly higher than their substring precision (*M* = 0.70, *SD* = 0.17), *t*(63) = 13.87, *p* < .000. Likewise, mean character recall (*M* = 0.81, *SD* = 0.12) was significantly higher than substring recall (*M* = .68, *SD* = 0.17), *t*(63) = 14.01, *p* < .000. Paired-sample correlations revealed significant relationships between character and substring precision, *r*(63) = 0.88, *p* < .000, and between character and substring recall, *r*(63) = 0.94, *p* < .000.

To examine the relationship between precision and recall, we collapsed the data across task and granularity to generate an overall mean for each measure. A *t*-test resulted in significant differences, *t*(63) = 5.83, *p* < .000. As we found in Study 1, participants' recall values were significantly less than their precision values; *M* = .75 (*SD* = .14) and *M* = 0.78 (*SD* = .14), respectively. The relationship between precision and recall was again significant, *r*(63) = .94, *p* < .000.

The possibility of a relationship between pattern matching and creation was investigated by collapsing the data across granularity and performance measures to generate an overall mean for each task. Contrary to Study 1, a *t*-test did not yield significant results, *t*(63) = 0.87, *p* > .01. Also in contrast with Study 1, there was a significant relationship between matching and creation, *r*(63) = 0.64, *p* < .000. To ensure that these findings were not due to an order effect, a repeated measures Analysis of Variance (ANOVA) was conducted. This analysis yielded non-significant results for the main effect of task, *F*(1,62) = 0.72, *p* > .01, and for the interaction of the task and version; *F*(1,62) = 0.16, *p* > .01.

To assess the relationships between the tasks of pattern creation and matching and the performance measures of recall and precision, we performed four paired-samples *t*-tests. First, we paired creation precision with matching precision to examine the influence of task on precision scores, yielding *t*(63) = 1.67, *p* > .01. Second, we paired creation recall with matching recall, again to investigate the influence of task on recall scores, yielding *t*(63) =

3.03, $p < .01$. Third, we paired creation precision with creation recall to examine the influence of performance within a task, resulting in $t(63) = 1.05$, $p > .01$. Fourth, we paired matching precision with matching recall, again to assess the influence of performance within a task, yielding $t(63) = 7.33$, $p < .000$. Paired-sample correlations resulted in significant relationships for all comparisons ($p < .000$); creation precision with matching precision $r = 0.63$, creation recall with matching recall, $r = 0.58$, creation precision and recall, $r = 0.97$, and matching precision and recall, $r = 0.81$.

Finally, we examined the differences in performance on items containing alternation or repetition in the creation and matching tasks. For the creation task, analysis indicated significant differences between alternation and repetition items, $t(63) = 3.09$, $p < .01$. Alternation items resulted in lower values than repetition items, $M = 0.71$ ($SD = 0.33$) and $M = 0.83$ ($SD = 0.21$), respectively. We also compared alternation items with items containing both alternation and repetition, $t(61) = 0.06$, $p > .01$. A final comparison of repetition items with items containing both alternation and repetition revealed a significant difference, $t(61) = 3.35$, $p < .01$. Items with both operators resulted in significantly lower values than alternation items, $M = 0.70$ ($SD = 0.28$) and $M = 0.84$ ($SD = 0.20$).

The same pattern emerged for the matching task. Analysis identified significant differences between alternation and repetition items, $t(62) = 3.75$, $p < .000$. Alternation resulted in significantly lower scores, $M = 0.72$ ($SD = 0.19$), than repetition, $M = 0.82$ ($SD = 0.18$). A comparison of alternation with items containing both repetition and alternation revealed no significant difference, $t(58) = 1.28$, $p > .01$. Finally a comparison of repetition with items containing both repetition and alternation resulted in significant differences, $t(39) = 5.29$, $p < .000$. Repetition resulted in higher scores, $M = 0.82$ ($SD = 0.17$), than items containing both operators $M = 0.68$ ($SD = 0.19$).

4.4 Discussion

The modifications to the instruction sheet changed the participants' reported satisfaction with the instruction sheet from 44.4% (Study 1) to 70.4% (Study 2). Anecdotal reports during debriefing indicated that the addition of an example suite was the primary cause of the participants' increased satisfaction.

The replication of the correlation between character level and substring level measures provides additional evidence of the exchangeability of the two scores. Future researchers may use either scoring technique without affecting results. However, it should be noted that the high sensitivity of substring level scores, and the associated lower mean for substring level than for character level scores, may obscure small effects in performance.

Replication of the correlation between precision and recall strongly suggests the absence of any significant individual strategy differences. As a community sample was used, it is unlikely that the correlation was due to any specific occupational factor (i.e., participants being students). Participants tend to use a conservative strategy and favour accuracy over completeness. While strategy differences may exist, they are displayed with respect to the amount of conservatism a specific participant employed. No evidence exists for the use of an aggressive strategy favouring recall over precision. During debriefing, participants indicated that they focused on the actual formation of patterns and not on the strings identified by the patterns. Thus, there was no evidence to indicate a relationship between regular expression use and information retrieval skills.

There was no significant difference in the means for the matching and creation tasks, unlike in Study 1. Consequently, when the difference in task difficulty was removed, performance on matching was found to correlate with that of creation. This correlation is suggestive of a common skill set being used 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.

As found by Salvatori (1983), humans develop the ability to read before they develop the ability to write and that the abilities to read and write are related but not correlated. Thus, it is unlikely that participants' skills when manipulating regular expressions, and hence formal language, are related to our skills for reading and writing natural language. The historic relationship of computer science to mathematics provides evidence of the similarity between the two fields. Debriefing of the participants determined that they viewed regular expressions according to their formation rules and considered them as a rule-based system, much like formal mathematics. Consequently, computer programming and formal language manipulation is more akin to a rule-based system than to a new and novel language. As it is not possible to speak or hear a formal language and thereby invoke the related cognitive abilities, there is considerable difference between programming and natural languages. This difference, along with how formal language is taught and presented, likely leads to the application of different cognitive skills when programming as opposed to when using natural language.

Our findings confirm the hypothesis that alternation is more difficult than concatenation or repetition. To ensure that the alternation operator was 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 unidentified form of operator interaction.

5. Study 3

Study 3 further explored the differences between regular and Boolean expressions, for which it has been found that using alternation is more time consuming, and hence difficult, than using conjunction and negation. We expected to find an analogous result and hypothesised that matching regular expressions containing alternation is slower than for expressions involving concatenation and repetition.

5.1 Participants

Participants were computer science students solicited at a Canadian university. A total of 32 participants (age, in years, *M* = 22.56, *SD* = 3.68) completed the study, but 2 participants were excluded as they were unable to correctly complete more than 3 of the 20 experimental items. The participants were predominantly male (*N* = 2 for females) due to the current under-representation of females in informatics disciplines.

5.2 Stimuli and Procedure

Participants were asked to locate solutions to a given expression in a target string. Customised software was used to present both the expression to match and the target string containing potential solutions. Beneath the string, a sequence of 7 buttons was provided for participants to select the number of matches in the target (None, 1, 2, 3, 4, 5, More than 5). Twenty expressions were presented in random order, with participants controlling when the exposure to each item was to begin and the software measuring the time needed to identify a solution.
During analysis, only correct solutions were used since solution times for incorrect results were believed to be inaccurate. The 20 items were divided into 4 subsets of 5 items: (C) concatenation only, (CR) concatenation and repetition, (CA) concatenation and alternation, and (CAR) concatenation, alternation, and repetition.

5.3 Results

We hypothesised that, similar to the reported results for Boolean expressions, users would take longer to correctly identify matches containing alternation operators. Table 1 shows the mean times in seconds for correctly identifying solutions to the 5 items in each subset.

Group	N	Mean	SD
C	147	5.31	1.30
CR	139	7.52	1.74
CA	141	8.86	2.45
CAR	126	11.96	2.96

Table 1. Descriptive Statistics for Study Three

Paired-samples *t*-tests were used to examine the effect that alternation had on performance. The mean time for items containing alternation (CA and CAR) was significantly longer than for items not containing alternation (C and CR) $t(29) = 14.18$, $p < .000$. To ensure that the difference was not due to expression complexity, the subsets CA and CR were compared and CA was found to have a significantly longer solution times than CR, $t(29) = 6.37$, $p<.000$.

5.4 Discussion

Our results confirm the hypothesis that alternation is more difficult than repetition or concatenation. Item groups containing the "or" operator had a significantly higher mean solution time than those not containing the operator. This result was also supported by the lower scores on items containing alternation for both the matching and the creation tasks of Study 2.
We did not compare the number of items correctly solved, as we attempted to recruit participants who were skilled at manipulating regular expressions and who we expected to obtain mostly correct results. Furthermore, the expressions were not overly complex so that the majority of participants would identify correct solutions, thus causing an expected ceiling effect.
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 increased skill in the use of a Boolean system. Vakkari also 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 versus 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. It is possible this effect is stronger in novices, as they may use working memory less efficiently while developing their cognitive skills.

We contend that the similarity of Boolean and regular expressions, with respect to increased solution times for expressions containing alternation, is due to the nature of the ideas that they are used to represent. Both can be viewed as examples of *rule-based* systems where the syntax and semantics are clearly defined by a set of rules. Unlike natural language, which can contain ambiguous, ideomatic, or metaphorical expressions, Boolean algebra and regular languages are clearly and concisely defined by a formalised set of rules. Thus, it is likely that we use similar cognitive skills for rule-based systems, and that these skills are more related to our faculties for reasoning than to those for language.

6. Future Work and Conclusions

The three experiments in this chapter consistently show that the lower recall performance on matching tasks, as a result of missed solutions for adjacent single-character substrings, is potentially due to some form of repetition blindness (Kanwisher, 1987). Performance may thus be affected by phenomena unrelated to participants' skill level. Future research will explore this hypothesis by examining the locations of missed solutions relative to similar solutions.

Pane and Myers (2000) explored the issue of pattern creation and matching in the context of Boolean algebra. They report 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. Their data suggests, on the basis of a correct versus incorrect scoring system, 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 lower creation than matching scores in Study 1 and equivalent scores in Study 2. This apparent discrepancy in reported findings requires further investigation.

In comparison to the findings of Greene et al. (1990), our solution times for the use of alternation do not show as great a difference to those for expressions without alternation. Whereas repetition requires the location of an arbitrary number of sequential solutions, conjunction requires the location of only two solutions and is potentially faster to solve. We intend to explore this issue in future studies.

Although the research presented here has begun a highly needed exploration of the cognitive skills needed to manipulate regular expressions, there is still much to be done. We have found evidence that, contrary to expectations, the manipulation of formal language has interesting differences to the manipulation of natural language. While the domain of language application, such as the use of regular expressions for describing search targets, may influence skilled users, there was no evidence of its influence on novice users. Thus, unlike natural language, which is based on mapping terms to real-world ideas, formal languages are likely mapped to the rules that describe and define their syntax and semantics. It might be considered that computer programming has more in common with other rule-based systems such as music, game-playing, and mathematics than it has to oral and written communication.

7. References

Blackwell, A. (2001). In: *Your Wish is My Command: Giving Users the Power to Instruct Their Software*, chapter 13 - SWYN: A Visual Representation for Regular Expressions, 245-270, Morgan Kaufmann, ISBN 1558606882, San Francisco, CA, USA.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, 2, 137–167, ISSN 00199958.

Chui, M. & Dillon, A. (1999). Speed and accuracy using four Boolean query systems. *Proceedings of 10th AAAI Midwest Artificial Intelligence and Cognitive Science Conference*, pp. 36–42, ISBN 1577350820, Bloomington, IN, USA.

Greene, S.; Devlin, S.; Cannata, P. & Gomez, L. (1990). No IFs, ANDs, or ORs: A study of database querying. *International Journal of Man–Machine Studies*, 32, 3, 303–326, ISSN 00207373.

Hopcroft J. & Ullman J. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, ISBN 020102988X, Reading, MA, USA.

Johnson-Laird, P. (1983). *Mental Models*. Harvard University Press, ISBN 0674568818, Cambridge, MA, USA.

Kanwisher, N. (1987). Repetition blindness: Type recognition without token individuation. *Cognition*, 27, 2, 117–143, ISSN 00100277.

Ledgard, H.; Whiteside, J.; Singer, A. & Seymour, W. (1980). The natural language of interactive systems. *Communications of the ACM*, 23, 10, 556-563, ISSN 00010782.

Pane, J. & Myers, B. (2000). Improving user performance on Boolean queries. *Proceedings of Conference on Human Factors in Computing Systems*, pp. 269–270, ISBN 9780201485639, The Hague, Netherlands.

Rouet, J.-F. (2006). *The Skills of Document Use: From Text Comprehension to Web-Based Learning*. Lawrence Erlbaum Associates, ISBN 0805846026 , Mahwah, NJ, USA.

Salvatori, M. (1983). Reading and writing a text: Correlations between reading and writing patterns. *College English*, 45,7, 657-666, ISSN 00100994.

Turtle, H. (1994). Natural language vs. Boolean query evaluation: A comparison of retrieval performance. *Proceedings of International Conference on Research and Development in Information Retrieval*, pp. 212–220, ISBN 354019889X, Dublin, Ireland.

Vakkari, P. (2000). Cognition and changes of search terms and tactics during task performance. *Proceedings of RIAO International Conference*, pp. 894–907, ISBN 290545007X, Paris, France.

Wilkinson, A.; Barnsley, G.; Hanna, P. & Swan, M. (1979). Assessing language development: The credition project. *Language for Learning*, 1, 2, 65, ISSN 00124576.