Exploring Regular Expression Usage and Context in Python

Carl Chapman, Kathryn T. Stolee*

Iowa State University, North Carolina State University

carlallenchapman@gmail.com, ktstolee@ncsu.edu

19 July, 2016

Why regular expressions?

Why regular expressions?

- Regexes are everywhere!

Why regular expressions?

- Regexes are everywhere!
- Everyone writes regexes!

Why regular expressions?

- Regexes are everywhere!
- Everyone writes regexes!
- Regexes are hard to read/write!

Why regular expressions?

- We wanted to write a tool to support regex creation.

Why regular expressions?

- We wanted to write a tool to support regex creation.
- But...

Why regular expressions?

- We wanted to write a tool to support regex creation.
- But...

We don't know how/when/why developers use regexes!

Why regular expressions?

- We wanted to write a tool to support regex creation.
- But...

We don't know how/when/why developers use regexes!

- and...

Why regular expressions?

- We wanted to write a tool to support regex creation.
- But...

We don't know how/when/why developers use regexes!

- and...

Regex feature usage references are missing!

Research goals

Explore regex

1. Context (developer survey)
2. Features (repository analysis)
3. Use cases (similarity analysis)

Regular expressions: The basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy

Regular expressions: The basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✗ abcccc
  - ✗ yxw

Regular expressions: The basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✗ abcccc
  - ✗ yxw
- `(ab*c|yz*)`

Regular expressions: The basics

- `(ab*c|yz*)$`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✗ abcccc
  - ✗ yxw
- `(ab*c|yz*)`
  - ✓ abbbbbbbc
  - ✓ y
  - ✓ abcy
  - ✓ abcccc
  - ✓ yxw

Part 1: Context

RQ1

In what contexts do professional developers use regular expressions?

Survey context

- 18 professional developers
- 9 years average development experience
- Small mobile payment management company
- 30 questions in a Google form

How often and where do developers use regexes?

- 50% – at least once per week

How often and where do developers use regexes?

- 50% – at least once per week

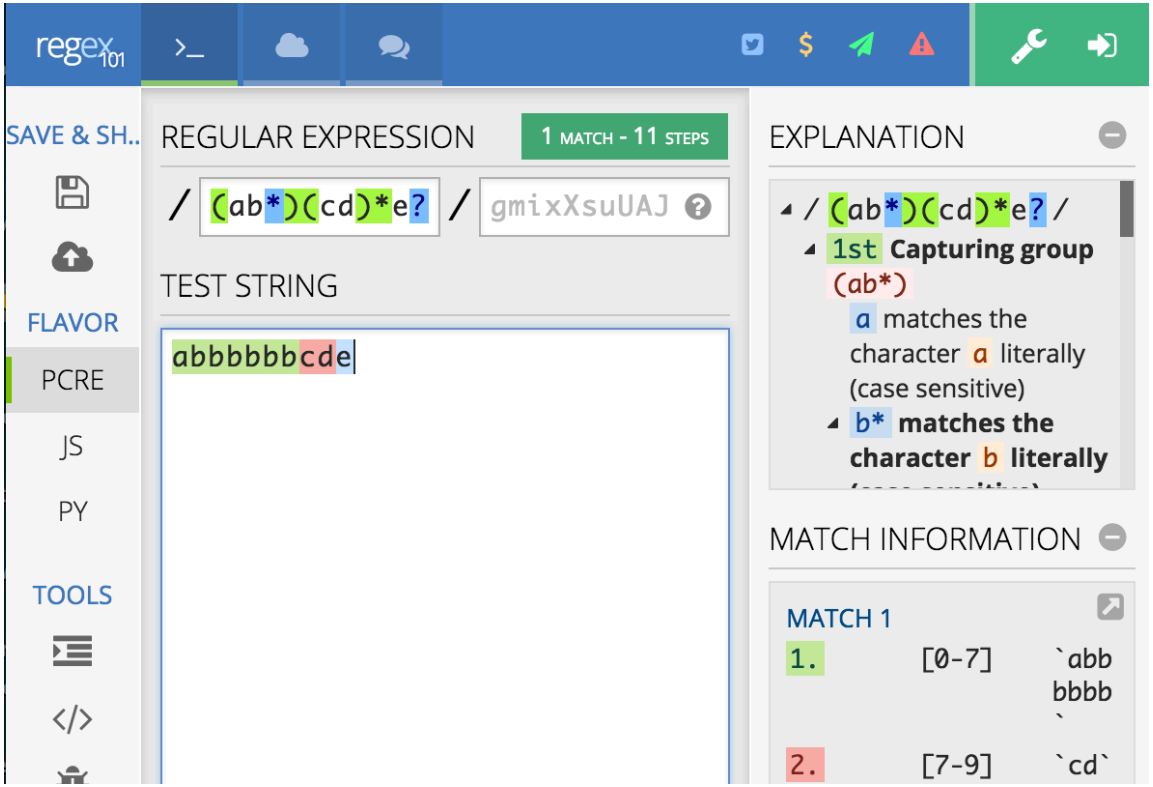
- **Most often**: command line and text editor tools
- **Often**: general purpose and scripting languages
- **Rare**: Database queries

Testing regular expressions

Developers test regular expressions less often than other code.

Testing regular expressions

Developers test regular expressions less often than other code.



50% say they use testing tools like www.regex101.com

Pain points

hard to compose (11 = 61%)

...very difficult to write them since I‘ve never read up on them.

Pain points

hard to compose (11 = 61%)

...very difficult to write them since I've never read up on them.

hard to read (7 = 39%)

It is terrible to read (especially later after initial development)

Pain points

hard to compose (11 = 61%)

...very difficult to write them since I‘ve never read up on them.

hard to read (7 = 39%)

It is terrible to read (especially later after initial development)

inconsistency across implementations (3 = 17%)

Some regexes work differently (or don‘t work) in some languages.

Notable observations: Context

- Everyone (sort of) writes regexes regularly
- Developers find regexes hard to read and write
- Most often written in text editors and IDEs
- Testing regexes is less common than testing other code

Notable observations: Context

- Everyone (sort of) writes regexes regularly
- Developers find regexes hard to read and write
- Most often written in text editors and IDEs
- Testing regexes is less common than testing other code

but....

Notable observations: Context

- Everyone (sort of) writes regexes regularly
- Developers find regexes hard to read and write
- Most often written in text editors and IDEs
- Testing regexes is less common than testing other code

but....

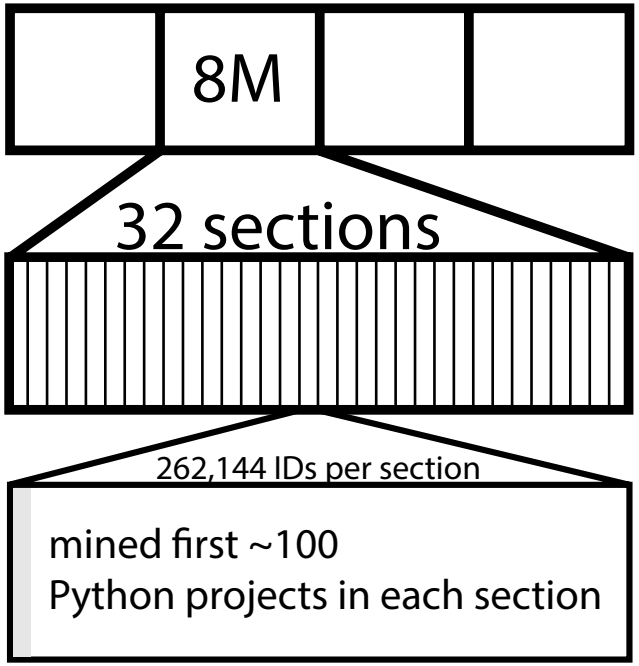
- Are regexes everywhere?
- Which features are everywhere?

Part 2: Features

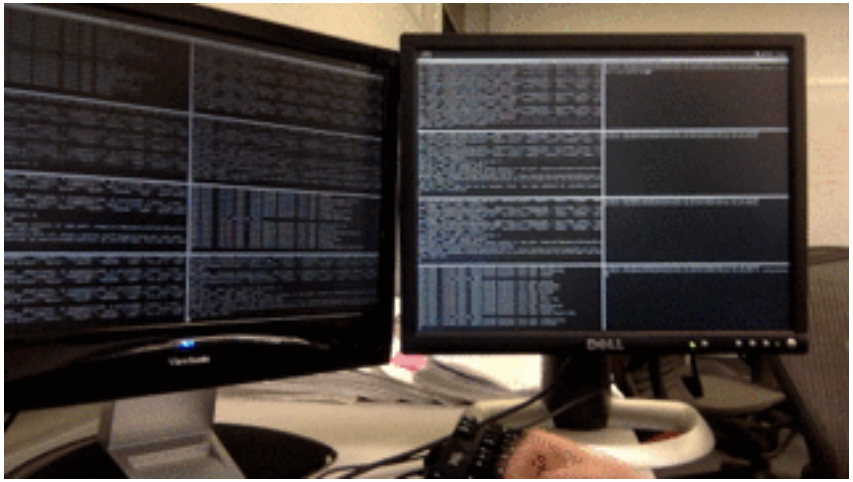
RQ3

Which regular expression language features are most commonly used in Python?

Project selection with the GitHub API



3,898 total Python projects found
(out of 42,000 inspected IDs)



Of 3,898 Python projects, 1,645 (42%) called the `re` module

In Python: Utilizations of the re module

function pattern flags

```
r1 = re.compile("(0|-?[1-9][0-9]*)$", re.MULTILINE)
```

function which function of the re module is called?

pattern string used to specify regex behavior

flags modifies the regex engine

Filtering utilizations and patterns

53,894 unique utilizations observed in 1,645 projects.

12.7% use behavioral flags

6.5% were non-static patterns

43,525 utilizations remain

Filtering utilizations and patterns

53,894 unique utilizations observed in 1,645 projects.

12.7% use behavioral flags

6.5% were non-static patterns

43,525 utilizations remain

13,711 distinct patterns

114 had various errors

Filtering utilizations and patterns

53,894 unique utilizations observed in 1,645 projects.

12.7% use behavioral flags

6.5% were non-static patterns

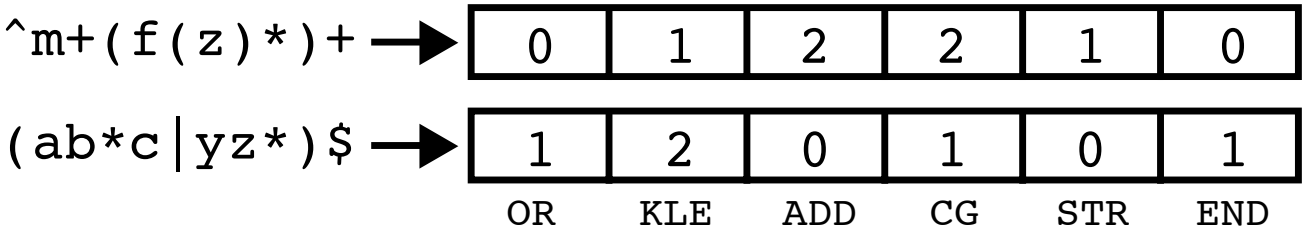
43,525 utilizations remain

13,711 distinct patterns

114 had various errors

13,597 patterns from 1,544 projects remain for analysis

PCRE parsing patterns



Feature statistics - Top 8

Rank	Code	Example	% Projects	% Patterns
1	ADD	`z+`	73.2	44.1
2	CG	`(caught)`	72.6	52.4
3	KLE	`.*`	66.8	44.3
4	CCC	`[aeiou]`	62.4	32.9
5	ANY	`.`	61.1	34.3
6	RNG	`[a-z]`	51.6	19.3
7	STR	`^`	51.4	26.2
8	END	`$`	50.3	23.3

Regex research tools

- Remember that we wanted to write a tool to support regex creation?

Regex research tools

- Remember that we wanted to write a tool to support regex creation?
- Regex feature usage references were missing (not anymore!).

Regex research tools

- Remember that we wanted to write a tool to support regex creation?
- Regex feature usage references were missing (not anymore!).
- So,

We analyzed your tools instead! (Hampi, Rex, RE2, brics, Automata.Z3)

Which features are supported by analysis tools?

Rank	Code	Example	Brics	Hampi	Rex	RE2	A.Z3
1	ADD	`z+`	●	●	●	●	●
2	CG	`(caught)`	●	●	●	●	●
3	KLE	`.*`	●	●	●	●	●
4	CCC	`[aeiou]`	●	●	●	●	●
5	ANY	`.`	●	●	●	●	●
6	RNG	`[a-z]`	●	●	●	●	●
7	STR	`^`	○	●	●	●	●
8	END	`$`	○	●	●	●	○
9	NCCC	`[^qwxf]`	●	●	●	●	○
10	WSP	`\s`	○	●	●	●	●
11	OR	`a|b`	●	●	●	●	●
12	DEC	`\d`	○	●	●	●	●
13	WRD	`\w`	○	●	●	●	●
14	QST	`z?`	●	●	●	●	●
15	LZY	`z+?`	○	●	○	●	○
16	NCG	`a(?:b)c`	○	●	○	●	○
17	PNG	`(?P<name>x)`	○	●	○	○	○

Rank	Code	Example	Brics	Hampi	Rex	RE2	A.Z3
18	SNG	`z{8}`	●	●	●	●	●
19	NWSP	`\S`	○	●	●	●	○
20	DBB	`z{3,8}`	●	●	●	●	●
21	NLKA	`a(?!yz)`	○	○	○	○	○
22	WNW	`\b`	○	○	○	●	○
23	NWRD	`\W`	○	●	●	●	○
24	LWB	`z{15,}`	●	●	●	●	○
25	LKA	`a(?=bc)`	○	○	○	○	○
26	OPT	`(?i)CasE`	○	●	○	●	○
27	NLKB	`(?<!x)yz`	○	○	○	○	○
28	LKB	`(?<=a)bc`	○	○	○	○	○
29	ENDZ	`\Z`	○	○	○	●	●
30	BKR	`\1`	○	○	○	○	○
31	NDEC	`\D`	○	●	●	●	○
32	BKRN	`\g<name>`	○	●	○	○	○
33	VWSP	`\v`	○	○	●	●	○
34	NWNW	`\B`	○	○	○	●	○

Notable observations: Features

- Regexes are (sort of) everywhere (42% of projects, 32 utilizations per project)
- Current regex research tools cover the most common features

Notable observations: Features

- Regexes are (sort of) everywhere (42% of projects, 32 utilizations per project)
- Current regex research tools cover the most common features

but....

Notable observations: Features

- Regexes are (sort of) everywhere (42% of projects, 32 utilizations per project)
- Current regex research tools cover the most common features

but....

What are the regexes doing?

Part 3: Use Cases

RQ4

How behaviorally similar are regexes across projects?

How to find common behaviors?

1. ~~thorough inspection of 53K utilizations~~

How to find common behaviors?

1. ~~thorough inspection of 53K utilizations~~
2. ~~cluster by syntactic similarity like Jaccard or longest substring~~

How to find common behaviors?

1. ~~thorough inspection of 53K utilizations~~
2. ~~cluster by syntactic similarity like Jaccard or longest substring~~
3. ~~formal analytical subsumption, no sufficient tools at the moment~~

How to find common behaviors?

1. ~~thorough inspection of 53K utilizations~~
2. ~~cluster by syntactic similarity like Jaccard or longest substring~~
3. ~~formal analytical subsumption, no sufficient tools at the moment~~
4. Chosen technique: cluster by behavioral similarity using Rex

Similarity metric example

A `(ab*c|yz*)$`

- abbbbbbbc
- y
- abcy
- pac
- abcyzzz

B `(ab*c|yz*)`

- y
- abc
- abcy
- abcccc
- yxw

Similarity metric example

A `(ab*c|yz*)$`

- abbbbbbbc
- y
- abcy
- pac
- abcyzzz

B `(ab*c|yz*)`

- y
- abc
- abcy
- abcccc
- yxw

A matches $3/5 =$ 60% of B's strings

Similarity metric example

A `(ab*c|yz*)$`

- abbbbbbbc
- y
- abcy
- pac
- abcyzzz

A matches $3/5 =$ 60% of B's strings

B `(ab*c|yz*)`

- y
- abc
- abcy
- abcccc
- yxw

B matches $5/5 =$ 100% of A's strings

Similarity metric example

A `(ab*c|yz*)$`

- abbbbbbbc
- y
- abcy
- pac
- abcyzzz

A matches $3/5 =$ 60% of B's strings

B `(ab*c|yz*)`

- y
- abc
- abcy
- abcccc
- yxw

B matches $5/5 =$ 100% of A's strings

A and B are 80% similar

Similarity matrix → MCL

	A	B	C	D
A	1.0	0.0	0.9	0.0
B	0.2	1.0	0.8	0.7
C	0.6	0.8	1.0	0.2
D	0.0	0.6	0.1	1.0

→

	A	B	C	D
A	1.0			
B	0.1	1.0		
C	0.75	0.8	1.0	
D	0.0	0.65	0.15	1.0

Rex generates
400 strings for each regex.
Average scores to
half-matrix for MCL

Scope

- 3,582 (26%) of patterns appeared in multiple projects
- 711 unsupported by Rex

Scope

- 3,582 (26%) of patterns appeared in multiple projects
- 711 unsupported by Rex

- 2,871 patterns analyzed from 722 (44%) of the projects
  - 186 clusters with size ≥ 2
  - 2,042 clusters with size $= 1$

Example cluster

Index	Pattern	NProjects	Index	Pattern	NProjects
1	`\s*([^: ]*)\s*:(.*)`	9	7	`[:]`	6
2	`:+`	8	8	`([^:]+):(.*)`	6
3	`(:)`	8	9	`\s*:\s*`	4
4	`(:+)`	8	10	`\:`	2
5	`(:)(:*)`	8	11	`^([^:]*):[^:]*$`	2
6	`^([^:]*): *(.*)`	8	12	`^[^:]*:([^:]*)$`	2

Six categories of clusters

Category	Clusters	Patterns	Projects	% Projects
Multi Matches	21	237	295	40%
Specific Char	17	103	184	25%
Anchored Patterns	20	85	141	19%
Two or More Chars	16	40	120	16%
Content of Parens	10	46	111	15%
Code Search	15	27	92	13%

Six categories of clusters

Category	Clusters	Patterns	Projects	% Projects
Specific Char	17	103	184	25%
Content of Parens	10	46	111	15%
Code Search	15	27	92	13%

Content of Parens `<(.+)>`, `<[^>]*?>`

Specific Char `:+`, `}`, `%`

Code Search `.*rlen=([0-9]+)`

Notable observations: Use cases

- Finding a specific character is quite common, 25% of projects (in contrast with survey)
- Regexes are often used to capture the contents of ( ), <>, and [ ] (aligning with survey)
- Regexes are often used to parse source code

Notable observations: Use cases

- Finding a specific character is quite common, 25% of projects (in contrast with survey)
- Regexes are often used to capture the contents of ( ), <>, and [ ] (aligning with survey)
- Regexes are often used to parse source code

but....

Notable observations: Use cases

- Finding a specific character is quite common, 25% of projects (in contrast with survey)
- Regexes are often used to capture the contents of ( ), <>, and [ ] (aligning with survey)
- Regexes are often used to parse source code

but....

- Similarity metric is approximate
- Metric is perhaps too sensitive to differences in literals

Opportunities for future work!

Opportunities for future work!

Better Similarity Metrics

Our similarity metrics are empirical, can we do it analytically?

Opportunities for future work!

Better Similarity Metrics

Our similarity metrics are empirical, can we do it analytically?

Migration Support for Developers

Supported regex features are different among languages.

Opportunities for future work!

Better Similarity Metrics

Our similarity metrics are empirical, can we do it analytically?

Migration Support for Developers

Supported regex features are different among languages.

Identifying Best Practices?

Could impact regex education and improve comprehension.

Opportunities for future work!

Better Similarity Metrics

Our similarity metrics are empirical, can we do it analytically?

Migration Support for Developers

Supported regex features are different among languages.

Identifying Best Practices?

Could impact regex education and improve comprehension.

Domain-Specific Support?

Does regex feature usage vary based on environment (IDE, code, text editor, etc.)?

Recap

- Regexes are (sort of) everywhere! (42% of Python projects)
- (almost) Everyone writes regexes! (50% of deves weekly)
- Regexes are hard to read/write! (this is a pain point)

Recap

- Regexes are (sort of) everywhere! (42% of Python projects)
- (almost) Everyone writes regexes! (50% of deves weekly)
- Regexes are hard to read/write! (this is a pain point)

also...

- Current tools support most of the most common features
- Regexes are often used for parsing/validating source code
- Many opportunities for future work!

Questions?

Katie Stolee – ktstolee@ncsu.edu

(psst! Graduate students! I'm hiring!)

Survey vs. Repository

How often do you use....

Group	Code	Survey	Repo Rank
endpoint anchors	(STR, END)	4.4	7, 8
capture groups	(CG)	4.2	2
word boundaries	(WNW)	3.5	22
lazy repetition	(LZY)	2.9	15
(neg) look-ahead/behind	(LKA, NLKA, LKB, NLKB)	2.5	25, 21 28, 27

Key: 6 = very frequently, 5 = frequently, 4 = occasionally, 3 = rarely, 2 = very rarely, 1 = never