

STATISTICAL PROGRAM ANALYSIS AND SYNTHESIS

Martin Vechev

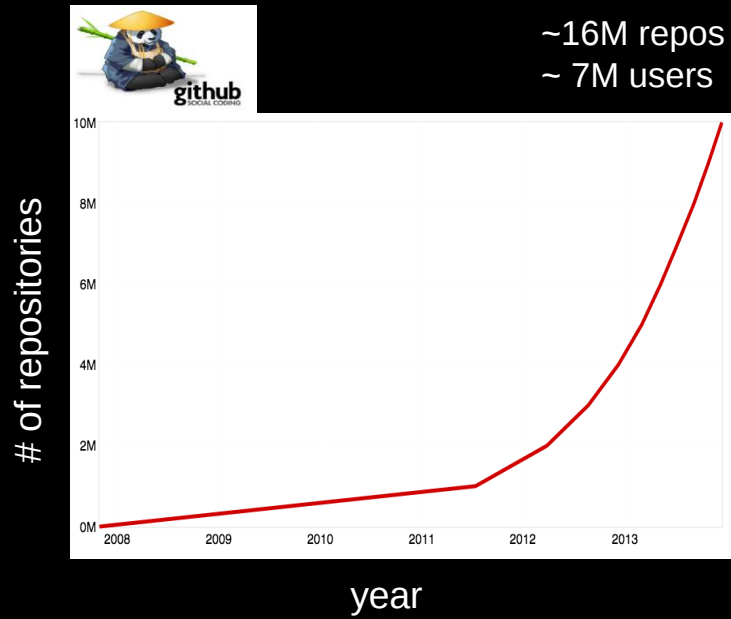
Department of Computer Science
ETH Zurich

Motivation

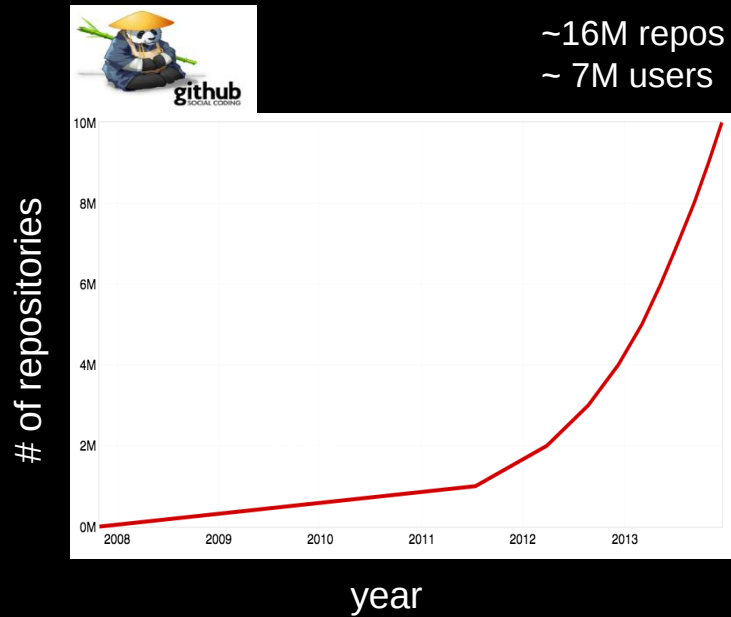
Unprecedented access to **massive codebases**
(termed “Big Code” by a recent DARPA initiative)



Motivation



Motivation

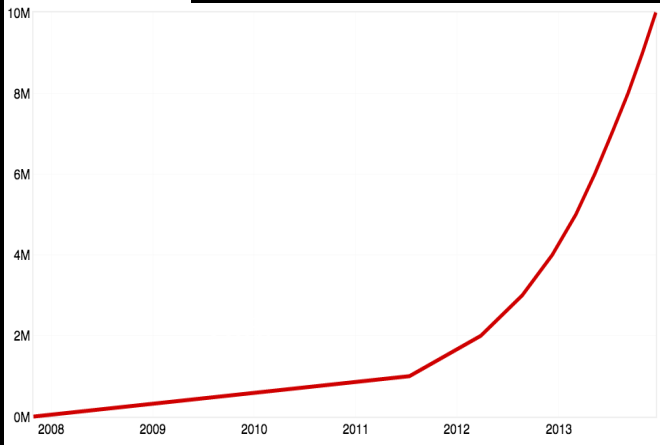


Motivation

of repositories



~16M repos
~ 7M users



year



Atlassian Bitbucket Passes 1 Million Users, Another Validation Of The Fast-Growing Developer Market

Posted Jun 4, 2013 by [Alex Williams \(@alexwilliams\)](#)
3 Share 128 in Share 45 Tweet 326



Number of Android applications

Current number of Android apps in the market:

1,375,385

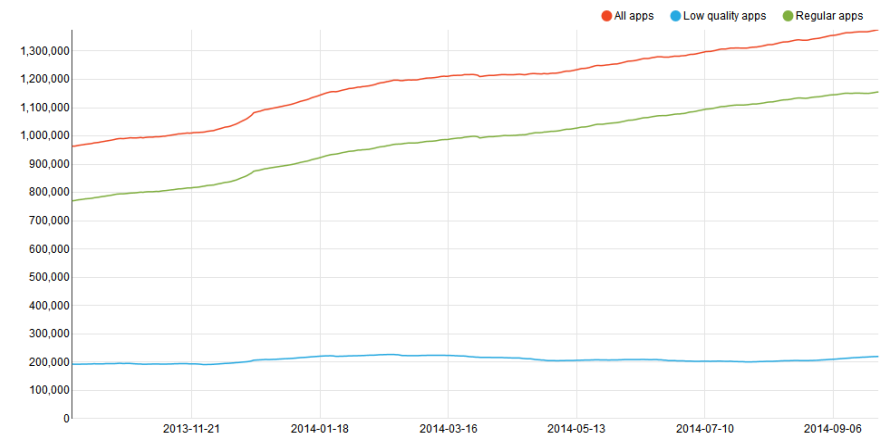
Percentage of low quality apps: **16%**

Development over time

This graph shows the number of available applications on Google Play. Last update: September 29, 2014.

The AppBrain low quality app detection filter detects automatically which apps are unlikely to be useful ("low quality apps"). Google seems to remove apps from the market roughly once a quarter, in which case the total number of available Android apps goes down. The removed apps are almost always classified by our system as low quality apps.

Android apps on Google Play



Vision

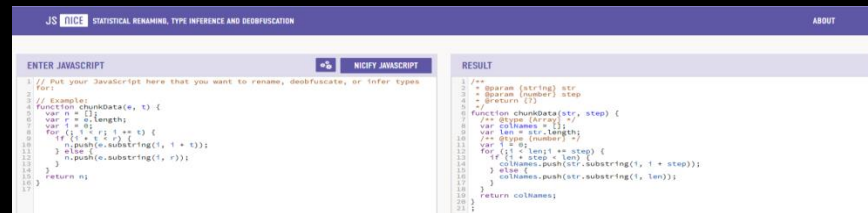
New techniques which **solve programming challenges** by leveraging availability of **massive codebases**

General Approach

Phrase the problem in a way which allows creative combination of advanced program analysis and powerful machine learning techniques

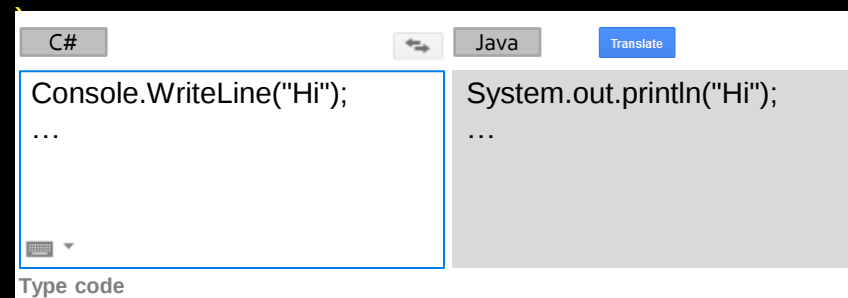
Statistical Program Analysis and Synthesis

Statistical Prediction of Program Properties, [POPL'15](#)
(V. Raychev, M. Vechev, A. Krause)



The screenshot shows the JS DICE web interface. The left pane, titled 'ENTER JAVASCRIPT', contains a JavaScript function `chunkData` that takes a string `s` and a number `n`, and returns an array of `n` chunks of `s`. The right pane, titled 'RESULT', shows the output of the function for `chunkData('123456789', 3)`, which is `["123","456","789"]`. The interface also includes a 'NIGHTY JAVASCRIPT' button and a 'STATISTICAL PREDICTION' button.

Statistical PL translation, [Onward'14](#)
(S. Karaivanov, V. Raychev, M. Vechev)



The screenshot shows the Onward'14 web interface. The left pane, titled 'C#', contains the code `Console.WriteLine("Hi");` followed by three dots. The right pane, titled 'Java', contains the code `System.out.println("Hi");` followed by three dots. A 'Translate' button is located between the two panes.

Statistical Code Synthesis, [PLDI'14](#)
(V. Raychev, M. Vechev, E. Yahav)

```
camera.setDisplayOrientation(90);
```

```
camera.unlock();
```

```
SurfaceHolder holder = getHolder();
```

Statistical Program Analysis and Synthesis

Statistical Code Synthesis, [PLDI'14](#)

(V. Raychev, M. Vechev, E. Yahav)

```
camera.setDisplayOrientation(90);
```

```
camera.unlock();
```

```
SurfaceHolder holder = getHolder();
```

Slang: Statistical Code Synthesis

```
void exampleMediaRecorder() throws IOException {  
    Camera camera = Camera.open();  
    camera.setDisplayOrientation(90);
```

```
    SurfaceHolder holder = getHolder();  
    holder.addCallback(this);  
    holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    MediaRecorder rec = new MediaRecorder();
```

```
    rec.setAudioSource(MediaRecorder.AudioSource.MIC);  
    rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);  
    rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
```

```
}
```

Slang: Statistical Code Synthesis

```
void exampleMediaRecorder() throws IOException {
```

```
    Camera camera = Camera.open();  
    camera.setDisplayOrientation(90);
```

```
    camera.unlock();
```

```
    SurfaceHolder holder = getHolder();  
    holder.addCallback(this);  
    holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    MediaRecorder rec = new MediaRecorder();
```

```
    rec.setCamera(camera);
```

```
    rec.setAudioSource(MediaRecorder.AudioSource.MIC);  
    rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);  
    rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
```

```
    rec.setAudioEncoder(1);  
    rec.setVideoEncoder(3);
```

```
}
```

Slang: Statistical Code Synthesis

```
void exampleMediaRecorder() throws IOException {  
    Camera camera = Camera.open();  
    camera.setDisplayOrientation(90);
```

```
    camera.unlock();
```

```
    SurfaceHolder holder = getHolder();  
    holder.addCallback(this);  
    holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    MediaRecorder rec = new MediaRecorder();
```

```
    rec.setCamera(camera);
```

```
    rec.setAudioSource(MediaRecorder.AudioSource.MIC);  
    rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);  
    rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
```

```
    rec.setAudioEncoder(1);  
    rec.setVideoEncoder(3);
```

```
}
```

sequence not in
training data

Slang: Statistical Code Synthesis

```
void exampleMediaRecorder() throws IOException {  
    Camera camera = Camera.open();  
    camera.setDisplayOrientation(90);
```

```
    camera.unlock();
```

```
    SurfaceHolder holder = getHolder();  
    holder.addCallback(this);  
    holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    MediaRecorder rec = new MediaRecorder();
```

```
    rec.setCamera(camera);
```

```
    rec.setAudioSource(MediaRecorder.AudioSource.MIC);  
    rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);  
    rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
```

```
    rec.setAudioEncoder(1);  
    rec.setVideoEncoder(3);
```

```
}
```

sequence not in
training data

multiple objects

Slang: Statistical Code Synthesis

```
void exampleMediaRecorder() throws IOException {  
    Camera camera = Camera.open();  
    camera.setDisplayOrientation(90);
```

```
    camera.unlock();
```

```
    SurfaceHolder holder = getHolder();  
    holder.addCallback(this);  
    holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    MediaRecorder rec = new MediaRecorder();
```

```
    rec.setCamera(camera);
```

```
    rec.setAudioSource(MediaRecorder.AudioSource.MIC);  
    rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);  
    rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
```

```
    rec.setAudioEncoder(1);  
    rec.setVideoEncoder(3);
```

```
}
```

sequence not in
training data

multiple objects

scalar
parameters

Slang: Statistical Code Synthesis

```
void exampleMediaRecorder() throws IOException {  
    Camera camera = Camera.open();  
    camera.setDisplayOrientation(90);
```

```
    camera.unlock();
```

```
    SurfaceHolder holder = getHolder();  
    holder.addCallback(this);  
    holder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);  
    MediaRecorder rec = new MediaRecorder();
```

```
    rec.setCamera(camera);
```

```
    rec.setAudioSource(MediaRecorder.AudioSource.MIC);  
    rec.setVideoSource(MediaRecorder.VideoSource.DEFAULT);  
    rec.setOutputFormat(MediaRecorder.OutputFormat.MPEG_4);
```

```
    rec.setAudioEncoder(1);  
    rec.setVideoEncoder(3);
```

```
}
```

sequence not in
training data

prediction is
instantaneous

multiple objects

scalar
parameters

Statistical Program Analysis and Synthesis

Statistical Code Synthesis, [PLDI'14](#)

(V. Raychev, M. Vechev, E. Yahav)

```
camera.setDisplayOrientation(90);
```

```
camera.unlock();
```

```
SurfaceHolder holder = getHolder();
```

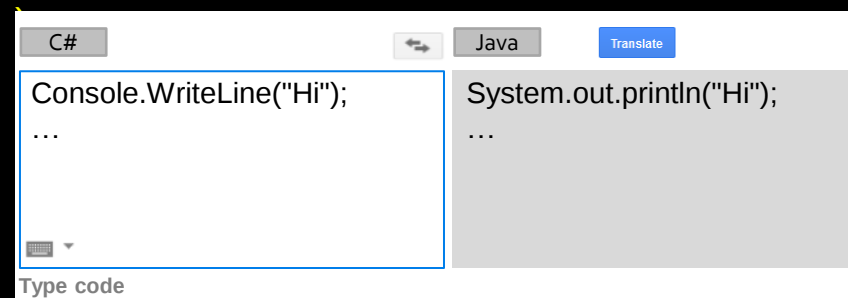
score programs with **generative models** such as
n-gram and recurrent neural networks (RNN)

+

sequence extraction with **alias and typestate**
static analysis

Statistical Program Analysis and Synthesis

Statistical PL translation, [Onward'14](#)
(S. Karaivanov, V. Raychev, M. Vechev)



Statistical Code Synthesis, [PLDI'14](#)
(V. Raychev, M. Vechev, E. Yahav)

```
camera.setDisplayOrientation(90);
```

```
camera.unlock();
```

```
SurfaceHolder holder = getHolder();
```

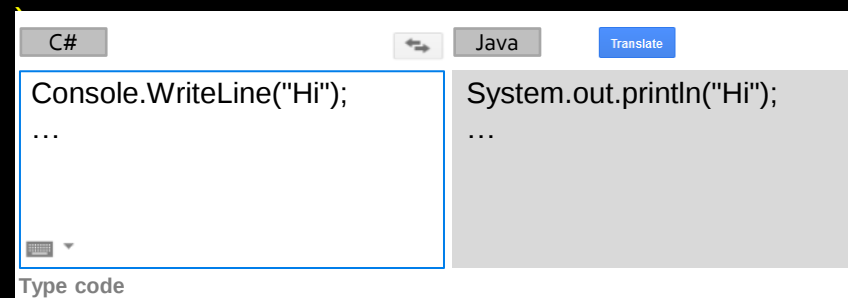
score programs with **generative models** such as
n-gram and recurrent neural networks (RNN)

+

sequence extraction with **alias and typestate**
static analysis

Statistical Program Analysis and Synthesis

Statistical PL translation, [Onward'14](#)
(S. Karaivanov, V. Raychev, M. Vechev)



phrase-based statistical machine translation

+

parse prefix grammars during translation

Statistical Code Synthesis, [PLDI'14](#)
(V. Raychev, M. Vechev, E. Yahav)

```
camera.setDisplayOrientation(90);
```

```
camera.unlock();
```

```
SurfaceHolder holder = getHolder();
```

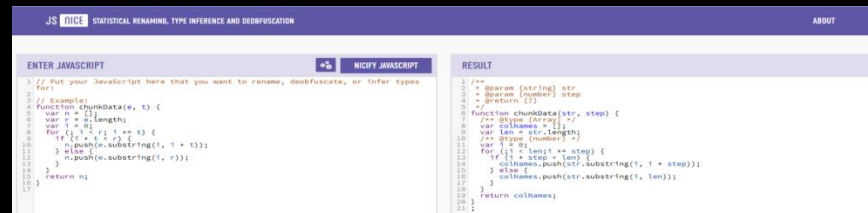
score programs with generative models such as
n-gram and recurrent neural networks (RNN)

+

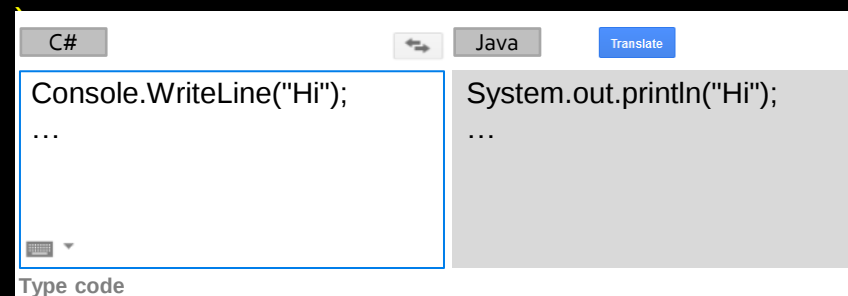
sequence extraction with alias and typestate
static analysis

Statistical Program Analysis and Synthesis

Statistical Prediction of Program Properties, [POPL'15](#)
(V. Raychev, M. Vechev, A. Krause)



Statistical PL translation, [Onward'14](#)
(S. Karaivanov, V. Raychev, M. Vechev)



phrase-based statistical machine translation

+

parse prefix grammars during translation

Statistical Code Synthesis, [PLDI'14](#)
(V. Raychev, M. Vechev, E. Yahav)

camera.setDisplayOrientation(90);

camera.unlock();

SurfaceHolder holder = getHolder();

score programs with generative models such as n-gram and recurrent neural networks (RNN)

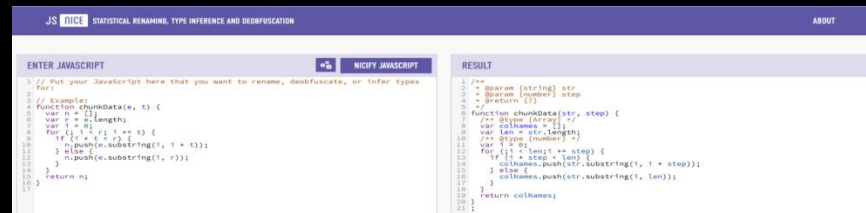
+

sequence extraction with alias and typestate static analysis

Statistical Program Analysis and Synthesis

Statistical Prediction of Program Properties, [POPL'15](#)
(V. Raychev, M. Vechev, A. Krause)

DEMO



The screenshot shows the JS-DICE web application interface. The top navigation bar includes 'JS DICE', 'STATISTICAL RENAMING, TYPE INFERENCE AND DEOBFUSCATION', and 'ABOUT'. Below the navigation bar, there are two tabs: 'ENTER JAVASCRIPT' and 'MODIFY JAVASCRIPT'. The 'ENTER JAVASCRIPT' tab is active, showing a text area with the following JavaScript code:

```
1 // Put your JavaScript here that you want to rename, deobfuscate, or infer types
2 for(
3
4 // Examples:
5 function chunkData(x, y) {
6   var n = x.length;
7   var t = 0;
8   for (i = 0; i < n; i++) {
9     if (x[i] > y) {
10       t++;
11     }
12     m.push(x.substr(i, y));
13   }
14   return m;
15 }
16 }
```

The 'MODIFY JAVASCRIPT' tab is also visible, showing the transformed code:

```
1 //
2 //
3 //
4 //
5 //
6 //
7 //
8 //
9 //
10 //
11 //
12 //
13 //
14 //
15 //
16 //
17 //
18 //
19 //
20 //
21 //
22 //
23 //
24 //
25 //
26 //
27 //
28 //
29 //
30 //
31 //
32 //
33 //
34 //
35 //
36 //
37 //
38 //
39 //
40 //
41 //
42 //
43 //
44 //
45 //
46 //
47 //
48 //
49 //
50 //
51 //
52 //
53 //
54 //
55 //
56 //
57 //
58 //
59 //
60 //
61 //
62 //
63 //
64 //
65 //
66 //
67 //
68 //
69 //
70 //
71 //
72 //
73 //
74 //
75 //
76 //
77 //
78 //
79 //
80 //
81 //
82 //
83 //
84 //
85 //
86 //
87 //
88 //
89 //
90 //
91 //
92 //
93 //
94 //
95 //
96 //
97 //
98 //
99 //
100 //
```

JSNice



800+ Tweets (sample below):



Kamil Tomšík @cztomsik · Jun 6

tell me [how this](#) works!

de-minify [#jquery](#) [#javascript](#) incl. args, vars & [#jsdoc](#)
impressive! jsnice.org

JSNice



800+ Tweets (sample below):



Alex Vanston @mrvidot · Jun 7

I've been looking for this for years: JS NICE buff.ly/1pQ5qfr #javascript
#unminify #deobfuscate #makeItReadable

Expand

↩ Reply ↻ Retweet ★ Favorite ⋮ More



Kamil Tomšík @cztomsik · Jun 6

tell me [how this](#) works!
de-minify #jquery #javascript incl. args, vars & #jsdoc
impressive! jsnice.org

JSNice



800+ Tweets (sample below):



Alvaro Sanchez @alvasavi · Jun 10

This is gold.

Statistical renaming, Type inference and Deobfuscation.

jsnice.org

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Alex Vanston @mrvidot · Jun 7

I've been looking for this for years: JS NICE buff.ly/1pQ5qfr #javascript
#unminify #deobfuscate #makeItReadable

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Kamil Tomšík @cztomsik · Jun 6

tell me how this works!

de-minify #jquery #javascript incl. args, vars & #jsdoc
impressive! jsnice.org

JSNice



800+ Tweets (sample below):



Brevity @seekbrevity · Jul 28

JSNice is an a [amazing](#) tool for de-minifying [#javascript](#) files. [JSNice.org](#), Its great for [#learning](#) and reverse engineering.

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Alvaro Sanchez @alvasavi · Jun 10

[This is gold.](#)

Statistical renaming, Type inference and Deobfuscation.

[jsnice.org](#)

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Alex Vanston @mrvidot · Jun 7

I've been looking for this for years: JS NICE [buff.ly/1pQ5qfr](#) [#javascript](#) [#unminify](#) [#deobfuscate](#) [#makeItReadable](#)

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Kamil Tomšik @cztomsik · Jun 6

tell me [how this](#) works!

de-minify [#jquery](#) [#javascript](#) incl. args, vars & [#jsdoc](#) impressive! [jsnice.org](#)

JSNice



800+ Tweets (sample below):



Ingvar Stepanyan @RRReverser · Aug 6

JSNice.org became my **must-have** tool for code deobfuscation.

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Brevity @seekbrevity · Jul 28

JSNice is an **amazing** tool for de-minifying **#javascript** files. JSNice.org, Its great for **#learning** and reverse engineering.

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Alvaro Sanchez @alvasavi · Jun 10

This is gold.

Statistical renaming, Type inference and Deobfuscation.

jsnice.org

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Alex Vanston @mrvdot · Jun 7

I've been looking for this for years: JS NICE buff.ly/1pQ5qfr **#javascript** **#unminify** **#deobfuscate** **#makeItReadable**

Expand

↩ Reply ↻ Retweet ★ Favorite ... More



Kamil Tomšík @cztomsik · Jun 6

tell me **how this** works!

de-minify **#jquery** **#javascript** incl. args, vars & **#jsdoc** impressive! jsnice.org

JSNice

- 30,000 users in 1st week
- used in > 170 countries



800+ Tweets (sample below):



Ingvar Stepanyan @RRReverser · Aug 6

JSNice.org beca

Expand



Brevity @seekbr

JSNice is an a
great for #learnir

Expand



Alvaro Sanchez

This is gold.

Statistical renaming, Type inference and Deobfuscation.

jsnice.org

Expand

Reply Retweet Favorite More



Alex Vanston @mrvidot · Jun 7

I've been looking for this for years: JS NICE buff.ly/1pQ5qfr #javascript
#unminify #deobfuscate #makeItReadable

Expand

Reply Retweet Favorite More



Kamil Tomšik @cztomsik · Jun 6

tell me how this works!

de-minify #jquery #javascript incl. args, vars & #jsdoc
impressive! jsnice.org

50 fantastic freebies for web designers, July 2014

By **Juan Pablo Sarmiento** | Resources | Jul 16, 2014

JSNice

- 30,000 users in 1st week
- used in > 170 countries



800+ Tweets (sample below):



Ingvar Stepanyan @RRReverser · Aug 6

JSNice.org beca

Expand



Brevity @seekbr

JSNice is an a
great for #learnir

Expand



Alvaro Sanchez

This is gold.

Statistical renaming, Type inference and Deobfuscation.

jsnice.org

Expand

Reply Retweet Favorite More



Alex Vanston @mrvidot · Jun 7

I've been looking for this for years: JS NICE buff.ly/1pQ5qfr #javascript
#unminify #deobfuscate #makeItReadable

Expand

Reply Retweet Favorite More



Kamil Tomšík @cztomsik · Jun 6

tell me how this works!

de-minify #jquery #javascript incl. args, vars & #jsdoc
impressive! jsnice.org

50 fantastic freebies for web designers, July 2014

By **Juan Pablo Sarmiento** | Resources | Jul 16, 2014

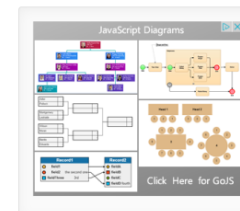


COMMENTS

HOME WORDPRESS WEB DESIGN TYPOGRAPHY RESOURCES ARCHIVES CONTACT ADVERTISE

20 Essential Tools for Coders

BY GAVIN IN DEVELOPMENT RESOURCES - 20 JUL, 2014



Over the last decade, third party tools and free development tools have become an increasingly popular solution for building web applications through a fast and cost-efficient development process. These tools are enriched with various features that turned out to be handy during the development process of an application. Probably this is the reason coders and developers keep looking for free and useful tools which make their work easier and quick.

If you are a coder and looking for some useful development tools then you are at

right place, I have gathered 20 Essential Tools for Coders that would help you out with your development tasks. Following development tools let you write your ideas into code, debug, beautify, store your snippet and help you to simplify your development tasks quickly. We hope you will find the list handy and according to your requirements. Enjoy!

1. JS Nice

JS NICE: STATISTICAL RENAMING, TYPE INFERENCE AND DEOBFUSCATION

ENTER JAVASCRIPT

JSNice

- 30,000 users in 1st week
- used in > 170 countries



800+ Tweets (sample below):



Ingvar Stepanyan @RRReverser · Aug 6

JSNice.org beca

Expand



Brevity @seekbr

JSNice is an a
great for #learnir

Expand



Alvaro Sanchez

This is gold.

Statistical renaming, Type inference

jsnice.org

Expand



Alex Vanston @mrvidot · Jun 7

I've been looking for this for years: JS NICE buff.ly/1pQ5qfr #javascript
#unminify #deobfuscate #makeItReadable

Expand

Reply Retweet Favorite More



Kamil Tomšik @cztomsik · Jun 6

tell me how this works!

de-minify #jquery #javascript incl. args, vars & #jsdoc
impressive! jsnice.org

50 fantastic freebies for July 2014

AUGUST 28, 2014

JavaScript迷宮救星 — JS nICE



我們網頁前端工程師常常要在網站上用瀏覽器的debug來替網站上的JavaScript檔案除錯，通常那些檔案都已經經過最小化、最佳化，還有變數混淆了，以增進檔案讀取的效率，並防止他人輕易盜用程式碼。不過一旦要除錯，就苦了前端工程師了。

就算現代的瀏覽器如Chrome, Safari與Firefox都支援美化程式碼的功能，讓最小化的程式還原成有排版的樣貌，但是遇到變數混淆過的程式，面對一堆變數名稱是a, b, c, d的程式，還是會除錯除得很辛苦。

不過網路上有些神奇的工具可以幫助我們不僅是把攪成一團的JS檔排的美美的，甚至它會替我們逆推算已經混淆的變數名稱！

那就是這篇要介紹的JS nICE網站。

這個網站的介面超級乾淨，就像下圖一樣，左邊是複製貼上的JS原始碼，只要按下NICIFY JAVASCRIPT按鈕，經過排版與替換變數的成果就會出現在右邊了！



COMMENTS

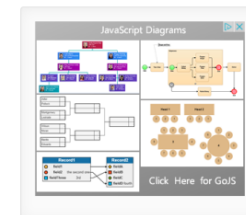
MOST RECENT

Frédéric Komez on:
20 Online Code Editors and Tools for Developers

HOME WORDPRESS WEB DESIGN TYPOGRAPHY RESOURCES ARCHIVES CONTACT ADVERTISE

20 Essential Tools for Coders

BY GAVIN IN DEVELOPMENT RESOURCES - 20 JUL, 2014

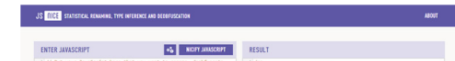


Over the last decade, third party tools and free development tools have become an increasingly popular solution for building web applications through a fast and cost-efficient development process. These tools are enriched with various features that turned out to be handy during the development process of an application. Probably this is the reason coders and developers keep looking for free and useful tools which make their work easier and quick.

If you are a coder and looking for some useful development tools then you are at

right place, I have gathered 20 Essential Tools for Coders that would help you out with your development tasks. Following development tools let you write your ideas into code, debug, beautify, store your snippet and help you to simplify your development tasks quickly. We hope you will find the list handy and according to your requirements. Enjoy!

1. JS Nice



JSNice

- 30,000 users in 1st week
- used in > 170 countries



800+ Tweets (sample below):



Ingvar Stepanyan @RRReverser · Aug 6

JSNice.org beca

Expand



Brevity @seekbr

JSNice is an a
great for #learnir

Expand



Alvaro Sanchez

This is gold.

Statistical renaming, Type inference
jsnice.org

Expand



Alex Vanston @mrvidot · Jun 7

I've been looking for this for years: JS NICE bu
#unminify #deobfuscate #makeItReadable

Expand



Kamil Tomšik @cztomsik · Jun 6

tell me how this works!

de-minify #jquery #javascript incl. args, vars
impressive! jsnice.org

50 fantastic freebies for July 2014

AUGUST 28, 2014

JavaScript迷宮救星 — JS nICE



我們網頁前端工程師常常要在網站上用瀏覽器的debug來替網站上的JavaScript檔案除錯，通常那些檔案都已經經過最小化、最佳化，還有變數混淆了，以增進檔案讀取的效率，並防止他人輕易盜用程式碼。不過一旦要除錯，就苦了前端工程師了。

就靠現代的瀏覽器如Chrome, Safari, Firefox, 還有像JSNice這樣的工具，面對一堆變數名稱是a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000

不過網路上有些神奇的工具可以幫助
那就是這篇要介紹的JS nICE網站。

這個網站的介面超級乾淨，就像下面一
變動的成果就會出現在右邊了！



Comment déminifier un code Javascript



COMMENTS

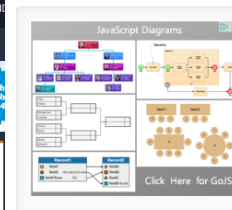
MOST RECENT

20 Online Code Editors and Tools for Developers

HOME WORDPRESS WEB DESIGN TYPOGRAPHY RESOURCES ARCHIVES CONTACT ADVERTISE

20 Essential Tools for Coders

BY GAVIN IN DEVELOPMENT RESOURCES - 20 JUL, 2014



Over the last decade, third party tools and free development tools have become an increasingly popular solution for building web applications through a fast and cost-efficient development process. These tools are enriched with various features that turned out to be handy during the development process of an application. Probably this is the reason coders and developers keep looking for free and useful tools which make their work easier and quick.

If you are a coder and looking for some useful development tools then you are at

right place, I have gathered 20 Essential Tools for Coders that would help you out with your development tasks. Following development tools let you write your ideas into code, debug, beautify, store your snippet and help you to simplify your development tasks quickly. We hope you will find the list handy and according to your requirements. Enjoy!

1. JS Nice



JSNice

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```



```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    if (i + step < len)
      colNames.push(str.substring(i, i + step));
    else
      colNames.push(str.substring(i, len));
  return colNames;
```



Kamil Tomšík @cztomsik · Jun 6

tell me **how this** works!

de-minify #jquery #javascript incl. args, vars & #jsdoc

impressive! jsnice.org

JSNice

$$y = \underset{y \in \Omega}{\operatorname{argmax}} \Pr(y \mid x)$$

JSNice

```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    if (i + step < len)
      colNames.push(str.substring(i, i+step));
    else
      colNames.push(str.substring(i, len));
  return colNames;
```

$$= \underset{y \in \Omega}{\operatorname{argmax}} \Pr(y |$$

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

JSNice

```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    if (i + step < len)
      colNames.push(str.substring(i, i+step));
    else
      colNames.push(str.substring(i, len));
  return colNames;
```

$$= \underset{y \in \Omega}{\operatorname{argmax}} \Pr(y |$$

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

learned from
“Big Code”



JSNice

```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    if (i + step < len)
      colNames.push(str.substring(i, i+step));
    else
      colNames.push(str.substring(i, len));
  return colNames;
```

$$= \underset{y \in \Omega}{\operatorname{argmax}} \Pr(y |$$

directly captured by a
discriminative model

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

learned from
“Big Code”



JSNice

captures **dependence**
(**structure**) between properties

directly captured by a
discriminative model

```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    if (i + step < len)
      colNames.push(str.substring(i, i+step));
    else
      colNames.push(str.substring(i, len));
  return colNames;
```

$$= \underset{y \in \Omega}{\operatorname{argmax}} \Pr(y |$$

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

learned from
“Big Code”



JSNice

captures **dependence**
(**structure**) between properties

directly captured by a
discriminative model

```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    if (i + step < len)
      colNames.push(str.substring(i, i+step));
    else
      colNames.push(str.substring(i, len));
  return colNames;
```

$$= \underset{y \in \Omega}{\operatorname{argmax}} \Pr(y \mid \dots)$$

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

learned from
“Big Code”

constraints
on properties



JSNice

captures **dependence**
(**structure**) between properties

directly captured by a
discriminative model

```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    if (i + step < len)
      colNames.push(str.substring(i, i+step));
    else
      colNames.push(str.substring(i, len));
  return colNames;
```

$$= \operatorname{argmax}_{y \in \Omega} \Pr(y \mid \dots)$$

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

candidate
properties

constraints
on properties

learned from
“Big Code”



JSNice

make a **joint** prediction for most likely properties

captures **dependence (structure)** between properties

directly captured by a **discriminative** model

```
function chunkData(str, step)
var colNames = [];
var len = str.length;
var i = 0;
for (; i < len; i += step)
  if (i + step < len)
    colNames.push(str.substring(i, i+step));
  else
    colNames.push(str.substring(i, len));
return colNames;
```

$$= \operatorname{argmax}_{y \in \Omega} \Pr(y \mid \dots)$$

```
function chunkData(e, t)
var n = [];
var r = e.length;
var i = 0;
for (; i < r; i += t)
  if (i + t < r)
    n.push(e.substring(i, i + t));
  else
    n.push(e.substring(i, r));
return n;
```

candidate properties

constraints on properties

learned from "Big Code"



JSNice: Inference

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

JSNice: Inference

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```



Unknown properties:

t r i ...

○ ○ ○ ...

Known properties:

o [] length

JSNice: Inference

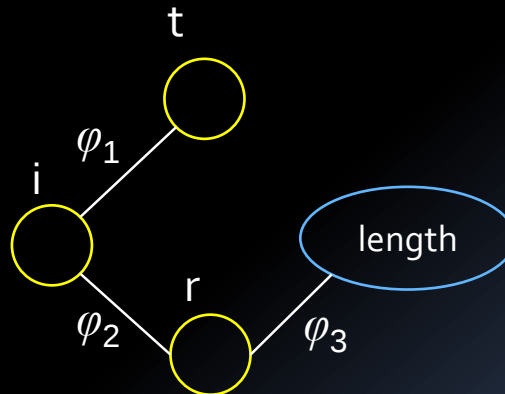
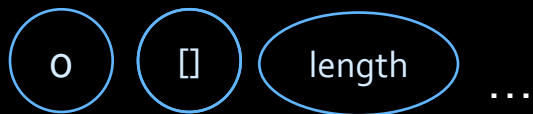
```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```



Unknown properties:



Known properties:



JSNice: Inference

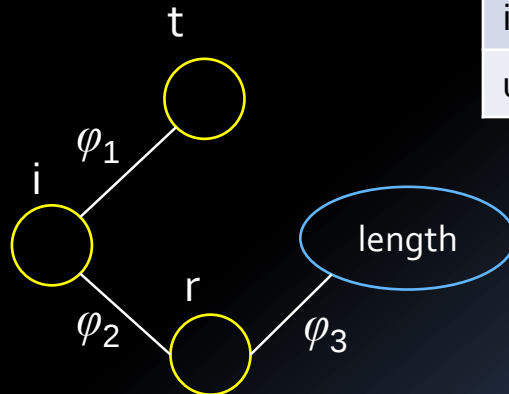
```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```



Unknown properties:



Known properties:



i	t	φ_1
i	step	0.5
j	j	0.4
i	j	0.2
u	q	0.05

JSNice: Inference

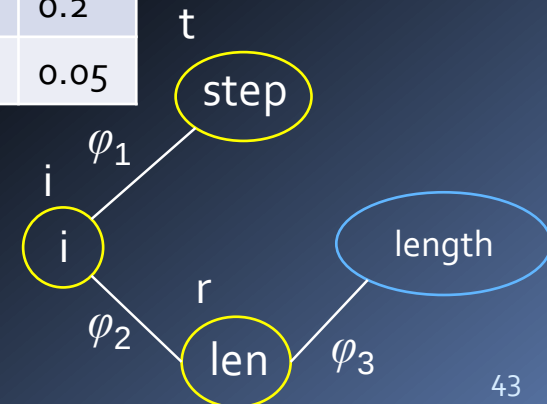
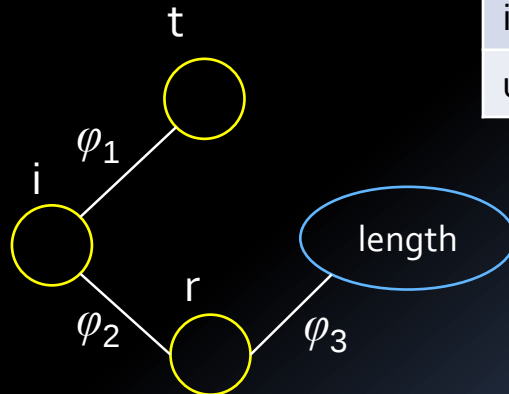
```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```



Unknown properties:



Known properties:



i	t	φ_1
i	step	0.5
j	j	0.4
i	j	0.2
u	q	0.05

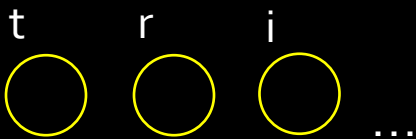
JSNice: Inference

```
function chunkData(e, t)
  var n = [];
  var r = e.length;
  var i = 0;
  for (; i < r; i += t)
    if (i + t < r)
      n.push(e.substring(i, i + t));
    else
      n.push(e.substring(i, r));
  return n;
```

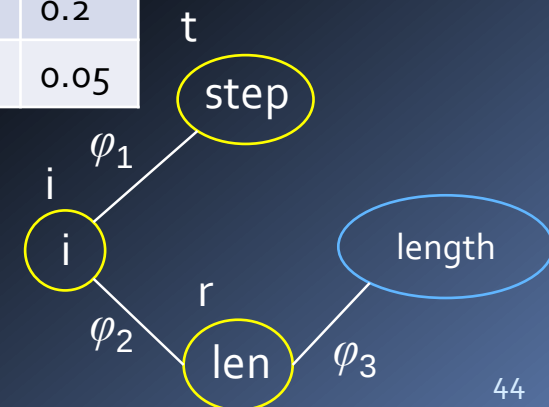
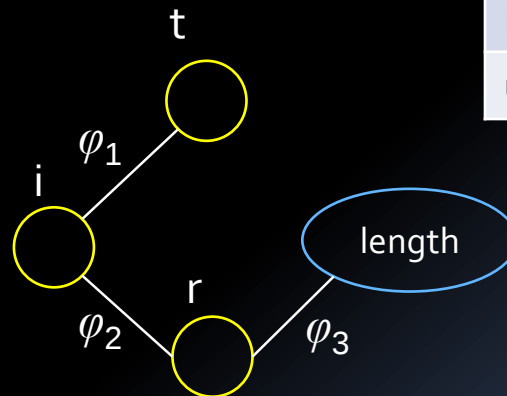
```
function chunkData(str, step)
  var colNames = [];
  var len = str.length;
  var i = 0;
  for (; i < len; i += step)
    colNames.push(str.substring(i, i + step));
  else
    colNames.push(str.substring(i, len));
  return colNames;
```



Unknown properties:



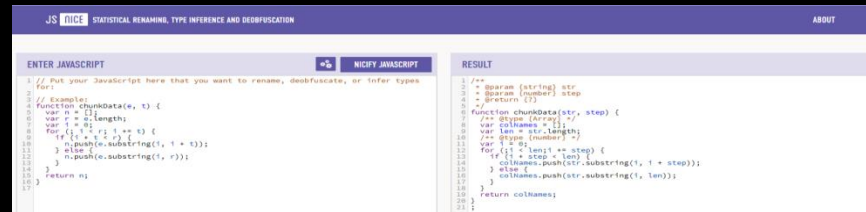
Known properties:



i	t	φ_1
i	step	0.5
j	j	0.4
i	j	0.2
u	q	0.05

Statistical Program Analysis and Synthesis

Statistical Prediction of Program Properties, [POPL'15](#)
(V. Raychev, M. Vechev, A. Krause)



```
// Put your Javascript here that you want to rename, deobfuscate, or infer types
// Example:
function chunkData(s, t) {
  var n = s.length;
  var i = 0;
  for (; i < n; i += t) {
    if (i + t > n) {
      s = s.substr(0, i + t);
    }
    n.push(s.substr(i, t));
  }
  return n;
}

// RESULT
1 // s: string [string] str
2 // n: number [number] step
3 // return: array [array]
4
5 // chunkData (string) str
6 // n: number [number] step
7 // return: array [array]
8
9 function chunkData(str, step) {
10   // s: string [string] str
11   // n: number [number] step
12   // return: array [array]
13   var n = [];
14   var len = str.length;
15   for (i = 0; i < len; i += step) {
16     if (i + step > len) {
17       str = str.substr(0, i + step);
18     }
19     n.push(str.substr(i, step));
20   }
21   return n;
22 }
```

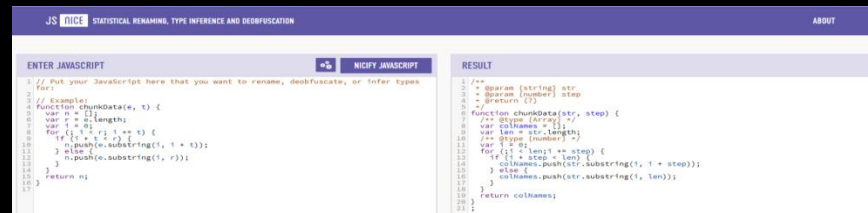
structured prediction with discriminative graphical models

+

features based on alias analysis important

Statistical Program Analysis and Synthesis

Statistical Prediction of Program Properties, [POPL'15](#)
(V. Raychev, M. Vechev, A. Krause)



The screenshot shows the JS DICE web interface. On the left, under 'ENTER JAVASCRIPT', there is a text area with the following code:

```
// Put your JavaScript here that you want to rename, deobfuscate, or infer types
// Examples:
function chunkData(x, y) {
  var n = x.length;
  var t = 0;
  for (i = 0; i < n; i++) {
    if (i % 2 === 0) {
      t = x.substr(i, 1);
    } else {
      t = y.substr(i, 1);
    }
    t.push(x.substr(i, 1));
  }
  return t;
}
```

 On the right, under 'RESULT', there is a text area with the following code:

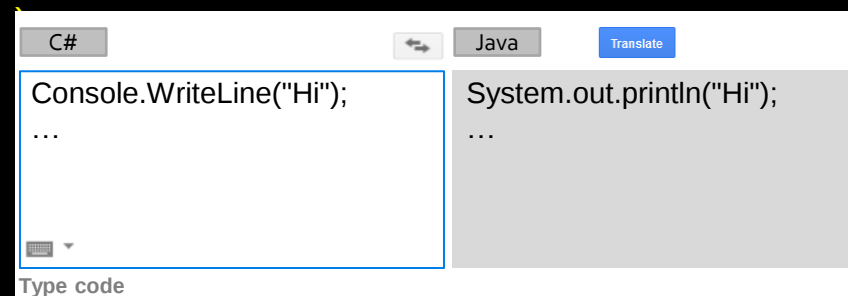
```
/*
 * Source [string] str
 * Source [number] step
 * Return [array]
 */
function chunkData(str, step) {
  var source = str;
  var columns = [];
  var len = str.length;
  var step = (step > 0 ? step : 1);
  for (i = 0; i < len; i += step) {
    if (i % 2 === 0) {
      columns.push(str.substr(i, step));
    } else {
      columns.push(str.substr(i, step));
    }
  }
  return columns;
}
```

structured prediction with discriminative graphical models

+

features based on alias analysis important

Statistical PL translation, [Onward'14](#)
(S. Karaivanov, V. Raychev, M. Vechev)



The screenshot shows a web interface for translating C# to Java. On the left, under 'C#', there is a text area with the following code:

```
Console.WriteLine("Hi");
...
```

 On the right, under 'Java', there is a text area with the following code:

```
System.out.println("Hi");
...
```

 A 'Translate' button is located between the two text areas.

phrase-based statistical machine translation

+

parse prefix grammars during translation

Statistical Code Synthesis, [PLDI'14](#)
(V. Raychev, M. Vechev, E. Yahav)

camera.setDisplayOrientation(90);

camera.unlock();

SurfaceHolder holder = getHolder();

score programs with generative models such as n-gram and recurrent neural networks (RNN)

+

sequence extraction with alias and typestate static analysis