

Data-Driven Evidence-Based Syntactic Sugar Design



David O'Brien, Robert Dyer, Tien Nguyen, and Hridesh Rajan

davidob@iastate.edu, rdyer@unl.edu, tien.n.nguyen@utdallas.edu, hridesh@iastate.edu



46th International Conference on Software Engineering



PEP 622 – Structural Pattern Matching

Author: Brandt Bucher <brandt at python.org>, Daniel F Moisset <dfmoisset at gmail.com>, Tobias Kohn <kohnt at tobiaskohn.ch>, Ivan Levkivskyi <levkivskyi at gmail.com>, Guido van Rossum <guido at python.org>, Talin <viridia at gmail.com>

BDFL-Delegate:

Discussions-To: Python-Dev list

Status: Superseded

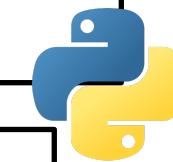
Type: Standards.Track

Created: 23-Jun-2020

Python-Version: 3.10

Post-History: 23-Jun-2020, 08-Jul-2020

Superseded-By: 634



JEP 395: Records

Owner: Gavin Bierman

Type: Feature

Scope: SE

Status: Closed / Delivered

Release: 16

Component: specification/language

Discussion: amber dash dev at openjdk dot java dot net

Relates to: JEP 359: Records (Preview)

JEP 384: Records (Second Preview)

Reviewed by: Alex Buckley, Brian Goetz

Endorsed by: Brian Goetz

Created: 2020/06/08 16:07

Updated: 2022/02/28 21:52

Issue: 8246771



Java™



Based off anecdotal experiences



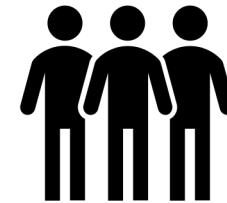
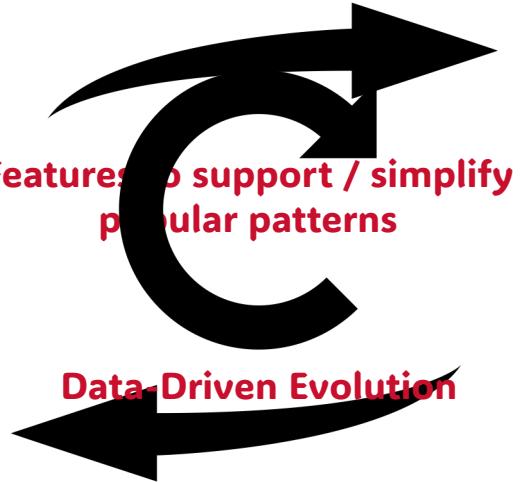
Navigated by designer's intuition



Low impact / irrelevant changes



Programming Language's
Designers



Programming Language's
Userbase

Contributions

1. *Technique* for recommending data-driven evidence-based programming language evolution.
2. *Evaluation* of our approach against a baseline.
3. *A catalog* of 7 data-driven Java syntactic sugars.
4. *Survey* evaluating user appreciation of catalog.

Data-Driven Evidence-Based Syntactic Sugars

Syntactic Sugar is a local transformation in programming languages, providing alternative syntax to express or condense code.

```
result = result ?: 0;
```

Elvis Operator (?:)

```
if (result == null) {  
    result = 0;  
}
```



Desugared Equivalent

Data-Driven Evidence-Based Syntactic Sugar
Syntactic sugar is a programming language construct that provides a more convenient or concise way to express common operations. It is often implemented as a syntactic sugar for more complex constructs, making the language easier to learn and use.

Frequent idioms can serve as data-driven evidence for programming language evolution!

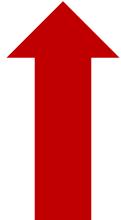
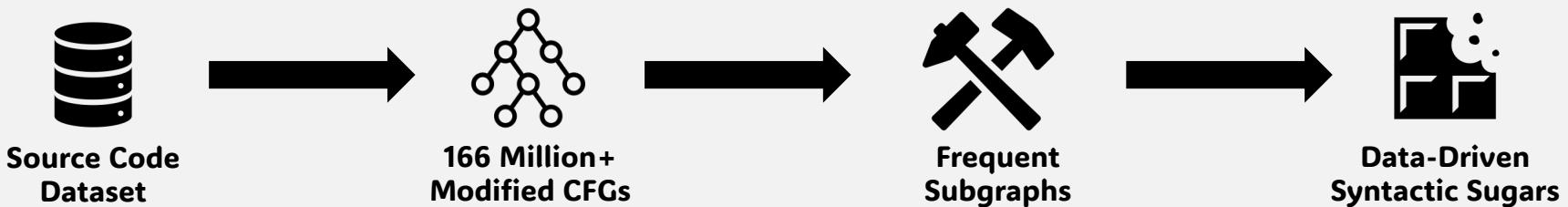
Elvis Operator (?:)

Desugared Equivalent

Overview of Approach



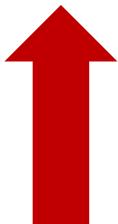
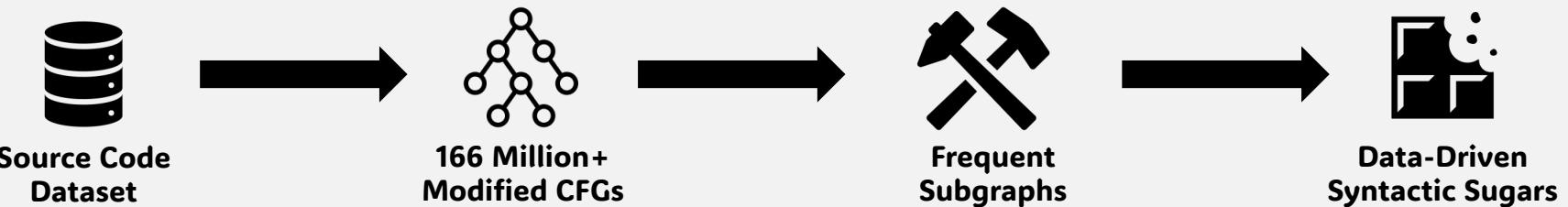
Overview of Approach



Boa Java Dataset

Granularity	Amount
Projects	380,125
Recent Snapshot Functions	166,827,154

Overview of Approach

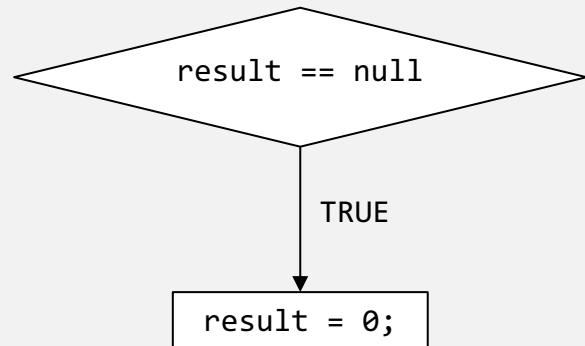


Control Flow Graph Overview

```
if (result == null) {  
    result = 0;  
}
```

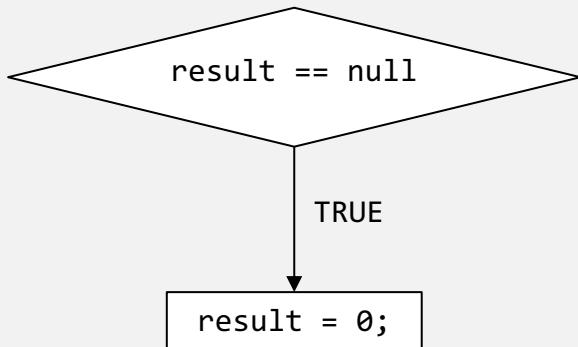


Example Java Code

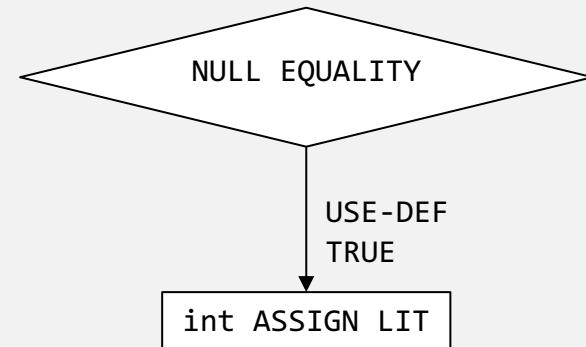


Control Flow Graph of Example Java Code

Control Flow Graph Overview

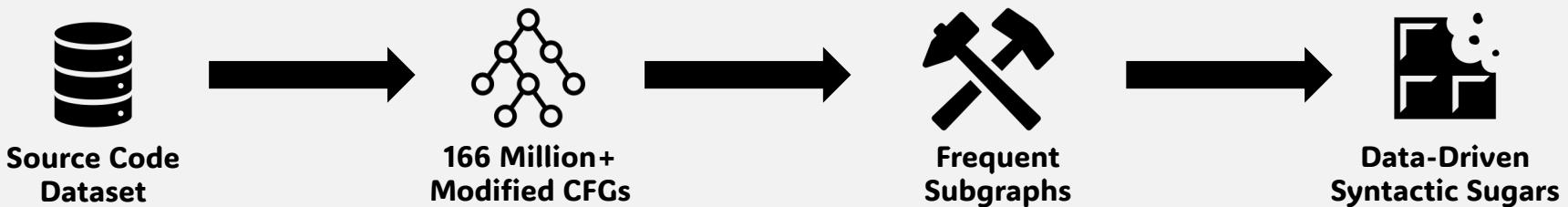


Control Flow Graph of Example Java Code

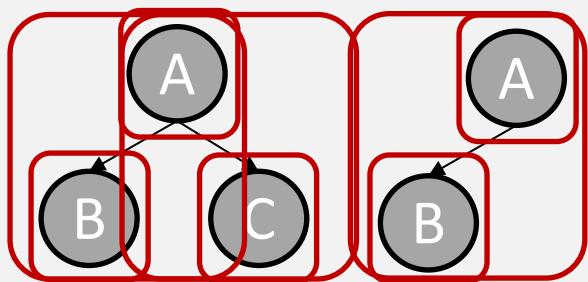


Generalized CFG of Example Java Code

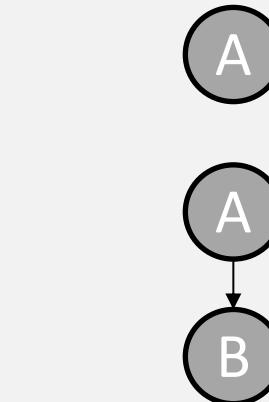
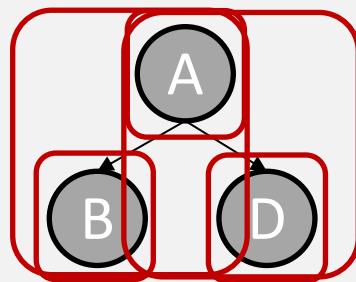
Overview of Approach



Frequent Subgraph Mining

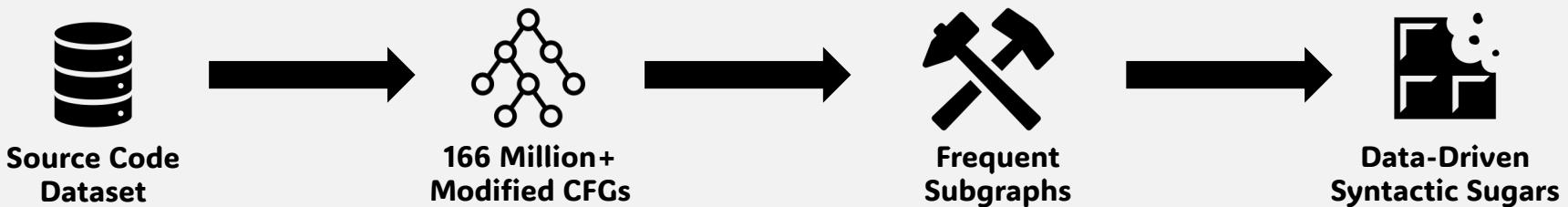


Graph Database

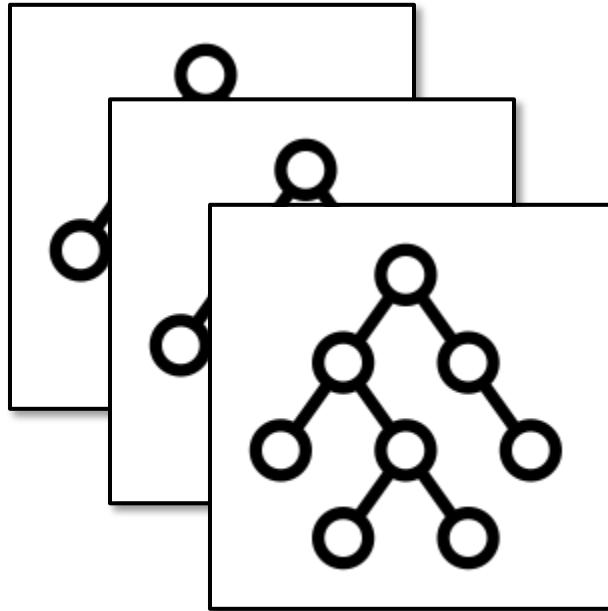


Found Frequent Subgraphs
(> 66%)

Overview of Approach



Data-Driven Evidence-Based Syntactic Sugars



166 Million+ Generalized CFG Database

1,865
Frequent Subgraphs

Data-Driven Evidence-Based Syntactic Sugars

Proposed Syntactic Sugar	Amount of Subgraphs
Multiple Assignment	5,543,853
Multiple ++	292,449
Unless	9,574,658
Any & All	10,267,105
Null if Null	236,567
requireType	116,742
Rethrow	1,994,698

Data-Driven Evidence-Based Syntactic Sugars

Multiple Assignment Proposal

```
id = 0;  
name = "Bob";  
age = 50;
```

Java Code

```
id, name, age = 0, "Bob", 50;
```

Proposed Syntactic Sugar

Data-Driven Evidence-Based Syntactic Sugars

Any & All Proposal

```
if (cond1 && cond2 && cond3) {  
    body  
}  
  
if (cond1 || cond2 || cond3) {  
    body  
}
```

Java Code

```
all(cond1, cond2, cond3) {  
    body  
}  
  
any(cond1, cond2, cond3) {  
    body  
}
```

Proposed Syntactic Sugar

Data-Driven Evidence-Based Syntactic Sugars

Unless Statement Proposal

```
if (!cond) {  
    body  
}
```

Java Code

```
unless (cond) {  
    body  
}
```

Proposed Syntactic Sugar

Data-Driven Evidence-Based Syntactic Sugars

JEP 430: String Templates (Preview)

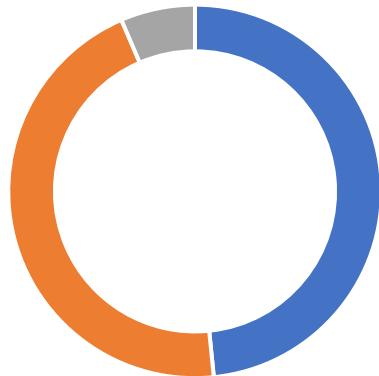
Owner	Jim Laskey
Type	Feature
Scope	SE
Status	Closed / Delivered
Release	21
Component	specification / language
Discussion	amber dash dev at openjdk dot org
Effort	M
Duration	M
Relates to	JEP 459: String Templates (Second Preview)
Reviewed by	Alex Buckley, Brian Goetz, Maurizio Cimadamore
Endorsed by	Brian Goetz
Created	2021/09/17 13:41
Updated	2023/10/13 13:52
Issue	8273943

```
String name = "Joan";
String info = STR."My name is \{name}";
assert info.equals("My name is Joan"); // true
```

Survey

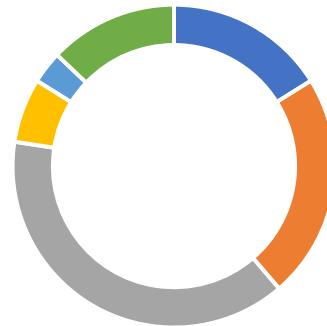
We survey a population consisting of:

Participants' Occupation



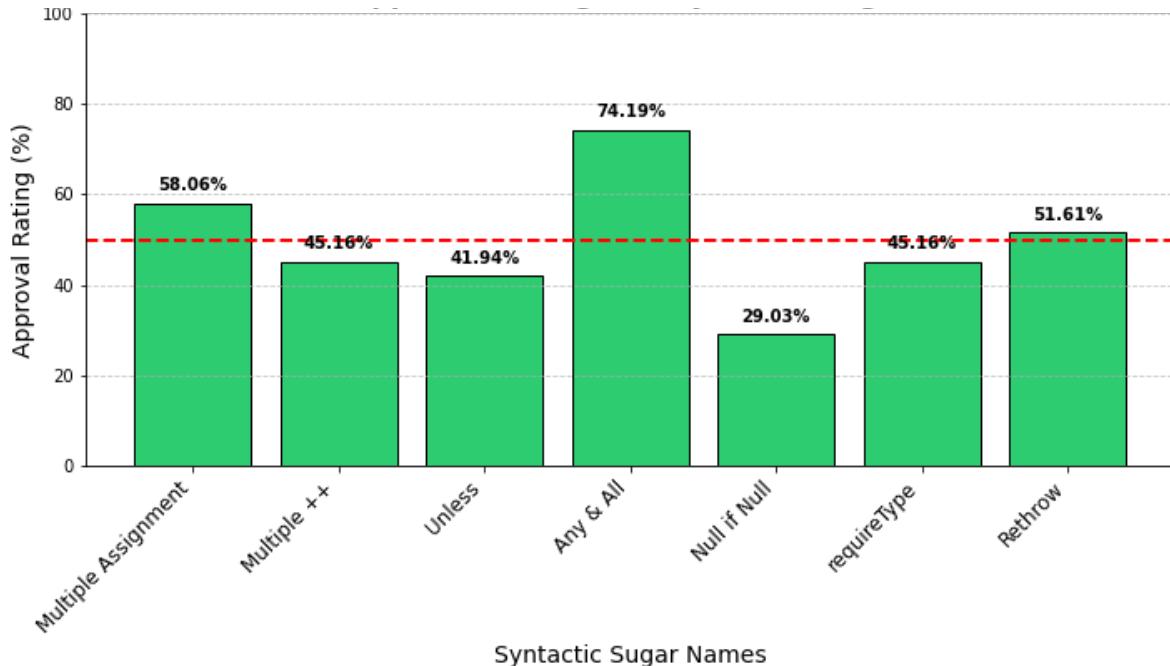
■ Graduate Student ■ Practitioner ■ Other

Participants' Java Experience



■ 0-2 Years ■ 2-5 Years ■ 5-10 Years
■ 10-15 Years ■ 15-20 Years ■ 20+ Years

Survey Results



Survey

Survey Results:

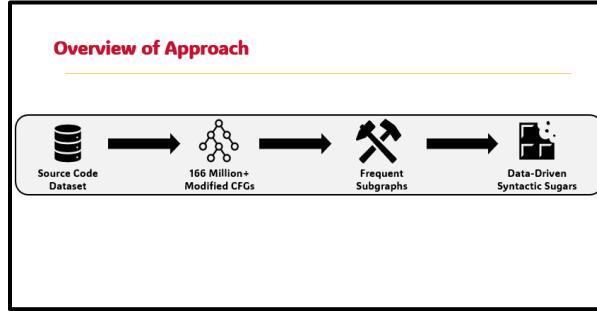
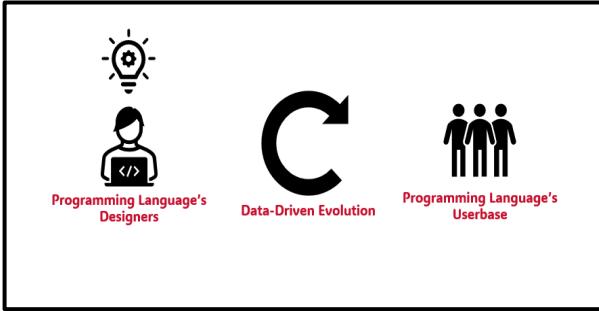
(Multiple Assignment 58.06%): Seems to read 'easier' (matches how we would state the condition in natural language) and unambiguously, so I'm a fan of the new feature.



(Unless 41.94%): I like the idea of unless, but the "else" case seems awkward. Is there a better word for unless?



Conclusion



Data-Driven Evidence-Based Syntactic Sugars

Proposed Syntactic Sugar	Amount of CFGs Present
Multiple Assignment	5,543,853
Multiple <code>++</code>	292,449
Unless	9,574,658
Any & All	10,267,105
Null if Null	236,567
<code>requireType</code>	116,742
Rethrow	1,994,698

