

Fig. 1. Statistics for the usage of the subtype rules when validating specification against GitHub packages

Julia Subtyping: A Rational Reconstruction

- FRANCESCO ZAPPA NARDELLI, Inria and Northeastern U.
- JULIA BELYAKOVA, Czech Technical U. in Prague 25
- ARTEM PELENITSYN, Czech Technical U. in Prague 26
- BENJAMIN CHUNG, Northeastern U. 27
- JEFF BEZANSON, Julia Computing 28

18

19 20

21 22 23

24

29

30 31

32

33

34

35

36

37

38

39

40

41

47

49

JAN VITEK, Northeastern U. and Czech Technical U. in Prague

STATISTICS OF USE OF SUBTYPE RULES

Figure 1 reports the number of uses (1a) and successes (1b) of each rule while building the derivations for the 6 million real-word tests we relied upon for validation. Scale on (1a) is logarithmic.

The first remark is that all our rules are actually considered by the algorithm at some point. The most frequent rule in derivations is, unsurprisingly, reflexivity (applied 37'298'823 times); we note that it was used, as expected, only against certain parts of the type grammar: kinds (Tuple, Union, UnionAll, DataType), class names, variable names and values. On the other hand, the two least frequent rules are TUPLE UNLIFT UNION and R L (occurred 27 and 1163 times); these are used in subtle corner cases of the algorithm. They have different success rates: the former succeeded every time it was invoked, while the latter never returned true.

Julia subtyping is a mixture of structural (tuples and unions) and nominal (user-defined types 42 with prescribed supertypes) features. Setting aside REFL, four most common rules, APP_INV, 43 UNION RIGHT, APP SUPER and TUPLE, show that the combination is vital in practice. Moreover, 44 it is clear that the ability to use parametric types is appreciated, although among all APP_SUPER 45 applications only somewhat 10% is about proper (non-nullary) type constructors. 46

From the multiple dispatching point of view it seems natural that the rules handling "interesting" right part of a judgment (UNION RIGHT and R INTRO) were tried more often than their 48

F. Zappa Nardelli, J. Belyakova, A. Pelenytsin, B. Chung, J. Bezanson, J. Vitek

left-counterparts (8'333'016 and 5'991'060 versus 1'278'605 and 2'241'454), because when resolving dispatch we have method signatures on the right and run-time type-tags of the values on the left. The same reasoning applies to R-variable being pervasive: R_LEFT applies 2'349'341 out of 2'356'449 and R_RIGHT applies 2'622'011 out of 2'533'770. Also vital for the dispatching mechanism, TUPLE is fifth most common rule, succeeding two thirds of times approximately (4'035'838 out of 6'550'185 tries to be precise) – this is a high success ratio, given that often tuples of different lengths appear in the logged tests (allegedly, those account for the cases when several overloadings of a function have different numbers of arguments and testing method applicability entails comparison of tuples of different sizes).

1:2

Proceedings of the ACM on Programming Languages, Vol. 1, No. CONF, Article 1. Publication date: January 2018.