The Duality of Subtyping (Artifact)

Bruno C. d. S. Oliveira
The University of Hong Kong, China
bruno@cs.hku.hk

Cui Shaobo
University of California San Diego, CA, USA
cuishaobo@gmail.com

Baber Rehman
The University of Hong Kong, China
brehman@cs.hku.hk

Abstract

This artifact contains the Coq formalization associated with the paper **The Duality of Subtyping** submitted in ECOOP 2020. This document explains how to run the Coq formalization. Artifact can either be compiled in the pre-built docker image with all the dependencies installed or it could be built from the scratch. Sections 1-7 explain the basic information about the artifact. Section A explains how to get the docker image for the artifact. Section B explains the prerequisites and the steps to run coq files from scratch. Section C explains coq files briefly. Section D shows the correspondence between important lemmas discussed in paper and their respective Coq formalization. The term MonoTyping used in artifact corresponds to the standard subtyping systems.

2012 ACM Subject Classification Software and its engineering → Object oriented languages
Keywords and phrases DuoTyping, OOP, Duality, Subtyping, Supertyping
Digital Object Identifier 10.4230/DARTS.6.2.8
Funding Funded by Hong Kong Research Grant Council projects number 17210617 and 17209519.
Acknowledgements We thank the anonymous reviewers for their helpful comments.
Related Article Bruno C. d. S. Oliveira, Cui Shaobo, and Baber Rehman, "The Duality of Subtyping", in 34th European Conference on Object-Oriented Programming (ECOOP 2020), LIPIcs, Vol. 166, pp. 29:1–29:29, 2020. https://doi.org/10.4230/LIPIcs.ECOOP.2020.29
Related Conference 34th European Conference on Object-Oriented Programming (ECOOP 2020), November 15–17, 2020, Berlin, Germany (Virtual Conference)

1 Scope

This artifact contains the Coq formalization associated with the paper **The Duality of Subtyping** submitted in ECOOP 2020. We provide the Coq formalization of various standard subtyping and our Duotyping systems.

2 Content

The artifact package includes:

- MonoTyping Coq Formalization
- DuoTyping Coq Formalization
- Copy of related paper (The Duality of Subtyping)
- README file with compilation instructions

3 Getting the artifact

The artifact endorsed by the Artifact Evaluation Committee is available free of charge on the Dagstuhl Research Online Publication Server (DROPS). In addition, the artifact is also available at: `https://github.com/baberrehman/coq-duotyping`.

4 Tested platforms

The artifact has been tested using Coq 8.7.0.

5 License

The artifact is available under license BSD.

6 MD5 sum of the artifact

5dcbcfceaba4c767a6dd8611b6320c86

7 Size of the artifact

797.0 KB

A Docker Image

This section explains how to pull the docker image of artifact from docker hub and use it. Run the following commands one by one in terminal:

1. $ docker pull baberrehman/duotyping
2. $ docker run -it baberrehman/duotyping
3. $ eval $(opam env)

The artifact is located in **/home/coq/coq-duotyping/coq/** directory.

There are two folders in the artifact, with **make** file in each:

1. **MonoTyping** → contains traditional subtyping formulation
2. **DuoTyping** → contains our duotyping formulation

Go to each folder and run make:

1. $ cd /home/coq/coq-duotyping/coq/DuoTyping
2. $ eval $(opam env)
3. $ make

```
coqc GStlc.v
coqc GStlcExtra.v
coqc GUnionInter.v
coqc GUnionInterExtra.v
coqc GFSubKernel.v
coqc GFSubKernelExtra.v
coqc GFSubKernelUnionInter.v
coqc GFSubKernelUnionInterExtra.v
coqc GFSubFull.v
coqc GFSubFullExtra.v
```

1. $ cd /home/coq/coq-duotyping/coq/MonoTyping
2. $ eval $(opam env)
3. $ make

```
coqc STLCSub.v
coqc STLCSubUnionInter.v
coqc FSubKernal.v
coqc FSubKernalUnionInter.v
coqc FSub.v
```

B Build from Scratch

This section explains how to build the artifact from scratch.

B.1 Prerequisites

We tested all the Coq files using Coq version 8.7.0. Please use same version for the sake of consistency. We recommend installing Coq using the opam package installer.

Run the following command to install Coq via opam:

$ opam install coq.8.7.0

Refer to this link for more information and installation steps:

`https://coq.inria.fr/opam-using.html`

B.2 Required Libraries

Coq TLC library release 20181116 is also required to compile the code. TLC library can also be installed using the opam package installer.

Run the following commands one by one to install TLC by opam package installer:

1. $ opam repo add coq-released http://coq.inria.fr/opam/released
2. $ opam install coq-tlc.20181116

Please refer to this link for detailed compilation and installation of Coq TLC:
`https://gitlab.inria.fr/charguer/tlc/-/blob/20181116/README.md`

User should be able to run each file individually using Coqide if these dependencies are installed.

B.3 Getting the artifact

Use the following commands to clone our git repo. Please note that $ symbol is not a part of command:

$ git clone https://github.com/baberrehman/coq-duotyping.git

Alternatively you can download the zip file from repo and you should be able to see all the Coq files after unzipping it.

B.4 Proof Structure

There are two folders in the artifact, **docs** and **coq**. Folder **coq** contains all the Coq files. Coq files are further divided into categories with separate folders:

1. **MonoTyping** → contains traditional subtyping formulation
2. **DuoTyping** → contains our duotyping formulation

B.5 Compilation

Please make sure to run the following command before running make if you installed the Coq via opam:

$ eval $(opam env)

Makefiles are available in both **MonoTyping** and **DuoTyping** folder. Run **make** command individually in each folder to compile.

B.6 Paper

You can also find a copy of our ECOOP'20 paper (The Duality of Subtyping) in **docs** folder.

C Overview of Coq Files

This section explains all the Coq files of our Duotyping systems and the traditional subtyping systems that we formalized. Table 1 shows the correspondence of Coq files and the respective systems. For example, one can find the Coq code for the system $\lambda_{<:}$in file Monotyping/STLCSub.v. We consider **coq** folder in unzipped artifact directory as reference point. We have both, algorithmic and the declarative formulation for all the Duotyping systems. The Coq formalizations for the traditional subtyping systems are based on existing Coq formalizations from the locally nameless representation with cofinite quantification tutorial and repository (`https://www.charguer aud.org/softs/ln/`) by Charguéraud [1]. The formalizations of $\lambda_{\Diamond}$, $\lambda_{\Diamond}^{\wedge\vee}$, $F_{k\Diamond}$, $F_{k\Diamond}^{\wedge\vee}$and $F_{F\Diamond}$are their respective Duotyping formulations, and modify the original ones with traditional subtyping.

D Overview of Important Lemmas

This section briefly explains the important lemmas discussed in paper and their correspondence with the coq formulation. Table 2 shows the correspondence between lemmas discussed in paper and their source coq code. For example, one can find the **Lemma 1** ($\lambda_{<:}$Transitivity) in file **Monotyping/STLCSub.v** and the lemma name in file is **sub_transitivity**.

References

1 Arthur Charguéraud. The locally nameless representation. *Journal of Automated Reasoning*, 2011.

Table 1 Description of Coq files of all systems.

Name	Description	Coq File
$\lambda_{<:}$	STLC with subtyping	Monotyping/STLCSub.v
$\lambda_{\Diamond}$	STLC with Duotyping (Algorithmic)	DuoTyping/GStlc.v
$\lambda_{\Diamond}$	STLC with Duotyping (Declarative)	DuoTyping/GStlcExtra.v
$\lambda_{<:}^{\wedge\vee}$	STLC with subtyping, union and intersection types	Monotyping/STLCSubUnionInter.v
$\lambda_{\Diamond}^{\wedge\vee}$	STLC with Duotyping, union and intersection types (Algorithmic)	DuoTyping/GUnionInter.v
$\lambda_{\Diamond}^{\wedge\vee}$	STLC with Duotyping, union and intersection types (Declarative)	DuoTyping/GUnionInterExtra.v
$F_{k<:}$	System $F_{<:}$ kernel	Monotyping/FSubKernal.v
$F_{k\Diamond}$	System $F_{<:}$ kernel with Duotyping (Algorithmic)	DuoTyping/GFSubKernel.v
$F_{k\Diamond}$	System $F_{<:}$ kernel with Duotyping (Declarative)	DuoTyping/GFSubKernelExtra.v
$F_{k<:}^{\wedge\vee}$	System $F_{<:}$ kernel with union and intersection types	Monotyping/FSubKernalUnionInter.v
$F_{k\Diamond}^{\wedge\vee}$	System $F_{<:}$ kernel with Duotyping, union and intersection types (Algorithmic)	DuoTyping/GFSubKernelUnionInter.v
$F_{k\Diamond}^{\wedge\vee}$	System $F_{<:}$ kernel with Duotyping, union and intersection types (Declarative)	DuoTyping/GFSubKernelUnionInterExtra.v
$F_{F<:}$	System full $F_{<:}$	Monotyping/FSub.v
$F_{F\Diamond}$	System full $F_{<:}$ with Duotyping (Algorithmic)	DuoTyping/GFSubFull.v
$F_{F\Diamond}$	System full $F_{<:}$ with Duotyping (Declarative)	DuoTyping/GFSubFullExtra.v

Table 2 Overview of important lemmas.

Lemma in Paper	**Coq File**	**Lemma in Coq File**
Lemma 1 ($\lambda_{<:}$Transitivity)	Monotyping/STLCSub.v	sub_transitivity
Lemma 2 ($\lambda_{\Diamond}$Transitivity)	DuoTyping/GStlc.v	trans
Theorem 1 (Reflexivity)	DuoTyping/GUnionInter.v	refl
Theorem 2 (Transitivity)	DuoTyping/GUnionInter.v	trans
Lemma 1 (Bound Selection)	DuoTyping/GUnionInter.v	invBound
Lemma 2 (Inversion for rule GDS-Both)	DuoTyping/GUnionInter.v	invChoose
Lemma 3 (Inversion for Union types)	Monotyping/STLCSubUnionInter.v	invOrS1
Lemma 4 (Inversion for Intersection types)	Monotyping/STLCSubUnionInter.v	invAndS1
Lemma 3 (Duality)	DuoTyping/GUnionInter.v	sym1
Lemma 5 (Type Preservation)	DuoTyping/GUnionInter.v	preservation_result
Lemma 6 (Progress)	DuoTyping/GUnionInter.v	progress_result
Theorem 3 (Reflexivity)	DuoTyping/GFSubKernelUnionInter.v	refl
Theorem 4 (Transitivity)	DuoTyping/GFSubKernelUnionInter.v	sub_transitivity
Lemma 7 (Bound Selection)	DuoTyping/GFSubKernelUnionInter.v	TopBtmMustEq1
Lemma 8 (Inversion for rule GDS-Both)	DuoTyping/GFSubKernelUnionInter.v	invChoose
Lemma 4 (Duality)	DuoTyping/GFSubKernelUnionInter.v	sym1
Lemma 9 ($F^{\wedge\vee}_{k<:}$Narrowing Lemma)	Monotyping/FSubKernalUnionInter.v	sub_narrowing
Lemma 10 ($F^{\wedge\vee}_{k\Diamond}$Narrowing Lemma)	DuoTyping/GFSubKernelUnionInter.v	sub_narrowing
Lemma 11 (Type Preservation)	DuoTyping/GFSubKernelUnionInter.v	preservation_result
Lemma 12 (Progress)	DuoTyping/GFSubKernelUnionInter.v	progress_result