

Steffen Breuer

# Integration and Evaluation of an ASP-Solver as an Alternative Reasoning Backend in the Rulewerk Toolkit

Dresden, December 17, 2020

# Introduction

- What is Rulewerk?
- What is the current backend?
- Why an alternative reasoning backend?
- What different backend was chosen?

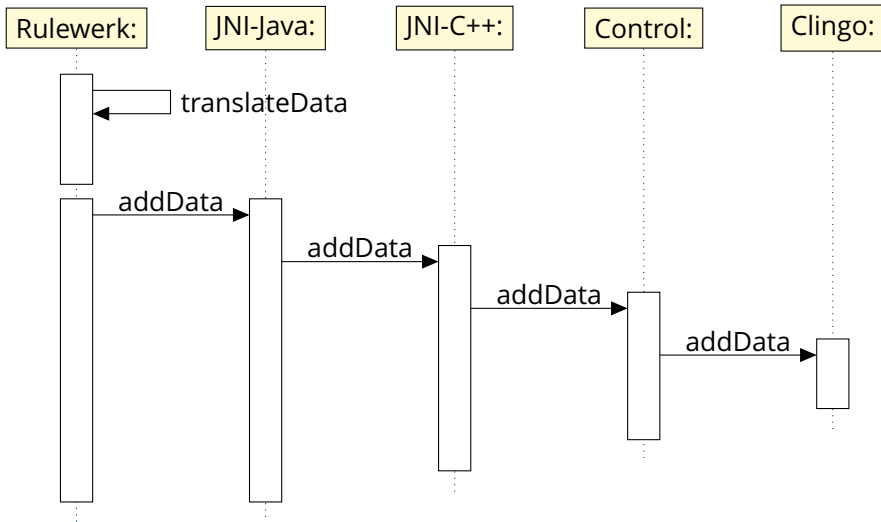
# Rulewerk

- Java API
- Knowledge modelling, data integration, and declarative computing
- **VLog**
  - Datalog with existential rules and stratified negation
  - C++
- CSV, DLGP format (Graal), define in Java, OWL ontologies, RDF, SPARQL query language

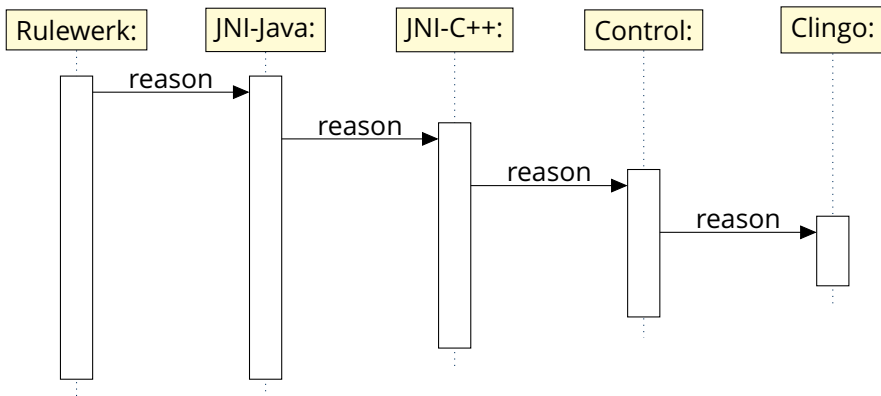
# Clingo

- Answer Set Programming (ASP) system
- gringo, clasp
- C++, C, Python

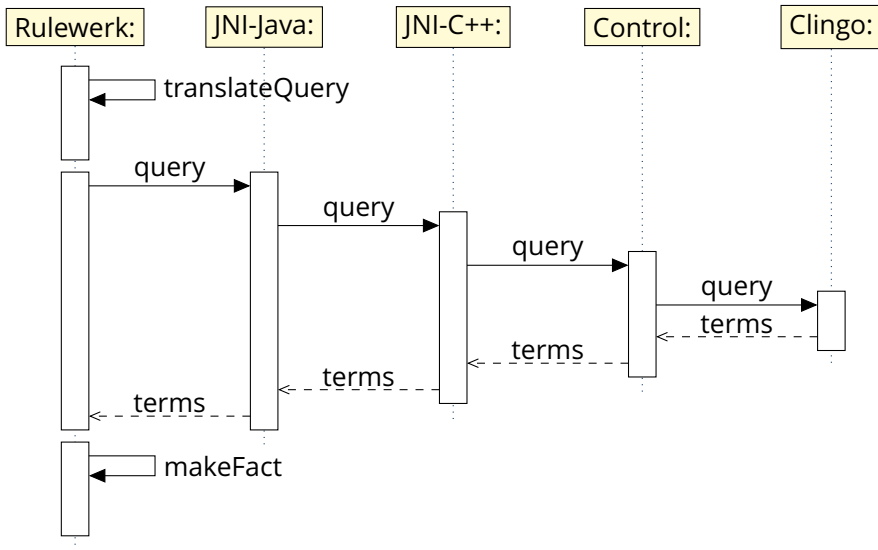
# General Workflow



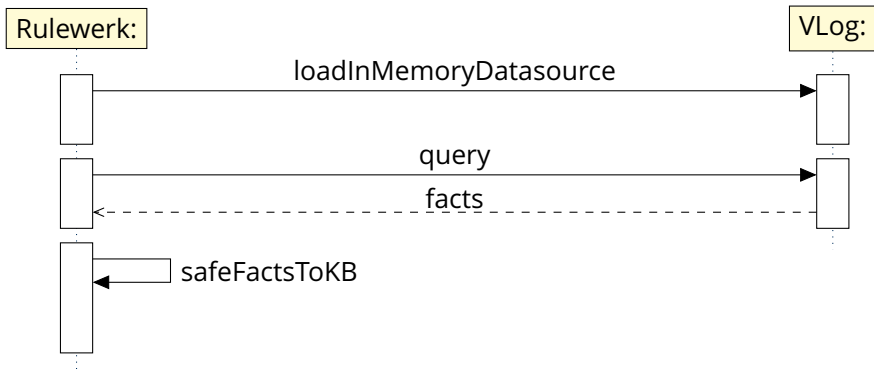
# General Workflow



# General Workflow



# General Workflow





# Interacting with Clingo

- How to add data?
- How to reason?
- How to get data out of Clingo?
- Why abstract the control of Clingo?

# Add Data

## ControlClingo

- Rules and facts (statements)
- Parse into Clingo syntax myself
  - only addable after grounding
- Add statements as strings
- Configurations
  - setting a part
  - setting parameters

# Reason over a Knowledge Base

## ControlClingo

- ground
  - set part
  - set params
- solve
  - add additional facts
  - save model as SymbolSpan (Vector)

# Get a model out of Clingo

## QueryTermIterator

- Returns the terms of a specific predicate/query
- First filtering SymbolSpan
- Filters SymbolSpan on the fly

# From Java to C++

## Java Native Interface

- Native methods
- Conversion of objects

# Usage within Rulewerk

- Translate Rulewerks syntax into Clingos syntax
- Make Clingos functions accessible
  - ClingoReasoner
- load CSV files
  - CSVloader
- Query a resulting model
  - ClingoQueryResultIterator

# Rulewerk vs Clingo syntax

- Both use rules and facts
- Both support negation
- Clingo does not support existential rules
- Rulewerk uses "?" / "!" in front of variables
- Rulewerk allows variables starting with a lower case letter
- Clingo does not support IRIs

# Translation: IRIs and Variables

- IRIs
  - Introduce aliases
  - Needs inner state
- Variables
  - make sure variables are in upper case
  - remove "?"/"!"



# Translation: Existential Rules

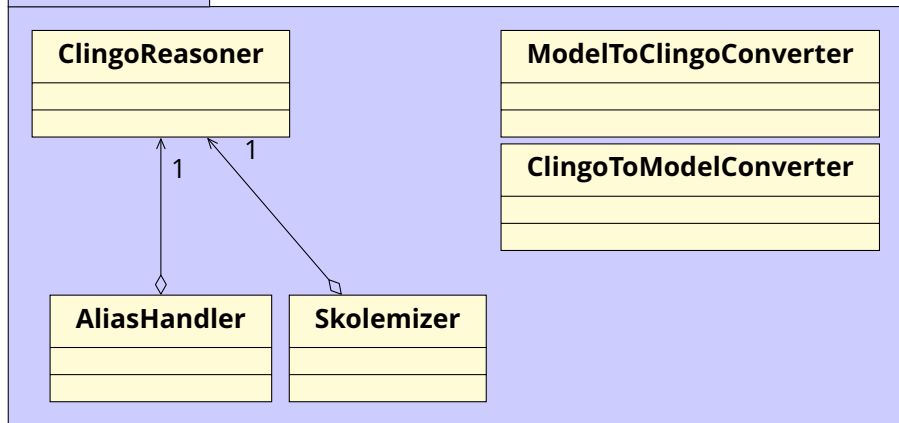
- Skolemization

$h(!X) \vdash b_1(?Y), b_2(?Z). \rightarrow h(f_1(Y, Z)) \vdash b_1(Y), b_2(Z).$

- Needs inner state

# Translation: Implementation

## Translation



# Setup

- Lenovo ideapad 520S, 8 GB of RAM, Intel Core i5-7200U
- Dataset: Lehigh University Benchmark (LUBM)
- OWL lite

# Rules	# Datalog rules	# Existential Rules
96	88	8

Table: TBox specs

ABox	Classes	Properties	facts
1	14	12	2283599
2	14	12	4549977

Table: ABox specs

# Results

Setup	whole run	loading from File	loading into backend	solving	querying
Clingo	87.892	16.201	25.708	23.490	22.469
VLog 1	30.942	15.324	8.388	1.301	5.516
VLog 2	12.810		6.737	1.063	4.801

Table: times in seconds for dataset 1

Setup	whole run	loading from File	loading into backend	solving	querying
Clingo	194.966	29.561	61.256	55.392	47.410
VLog 1	106.950	31.156	58.021	2.945	14.386
VLog 2	25.606		12.800	2.102	10.238

Table: times in seconds for dataset 2

# Conclusion

- It is possible to integrate Clingo
- Skolemization needed to express existential rules
- IRIS need to be replaced
- CSV files need to be passed through VLog
- Clingo is slow

# Improvements

- Improve skolemization by adding a function to access existential/datalog variables for predicates
- Load CSV files direktly into Rulewerk
- use the C API of Clingo