Hacking GF

Grégoire Détrez

August 29, 2013

Getting the code

With darcs

```
darcs get --lazy http://www.grammaticalframework.org/ GF
```

With git

```
git clone https://github.com/GrammaticalFramework/GF.git
```

As a zip archive

```
https://github.com/GrammaticalFramework/GF/archive/
master.zip
```

Preparing your computer

To be able to build gf, you will need the following:

- ghc & cabal (Recommended: Haskell Platform, 2012.4.0.0 or later)
- the haskeline

Debian (Ubuntu)

```
apt-get install haskell-platform libghc-haskeline-dev
```

Fedora

```
yum install haskell-platform ghc-haskeline-devel
```

Building gf

Install dependencies

```
cabal install --only-dependencies
```

Configure

```
cabal configure
```

Compile

```
cabal build rgl-none
```

Tips: use `cabal-dev` instead of `cabal` for a sandboxed developement environment

Source files

- Runtime (haskel, used by the compiler): `src/runtime/haskell`
- Shell & compiler: `src/compiler`

Compiler overview

Dependency analysis -> Parsing -> Renaming -> Grammar Checking -> Partial Evaluation -> PMCFG generation

(from https://code.google.com/p/grammatical-framework/wiki/CompilationPhases)

Dependency analysis

- traverses the dependencies of the top modules
- determines which modules need to be compiled
- . . . and in what order
- circular dependences are not allowed and are reported as error

location: `src/compiler/GF/Compile/ReadFiles.hs` `(getAllFiles)`

Parsing

- parse the `.gf` source files
- implemented using halex & happy

location: ‘src/compiler/GF/Grammar/{Parser.y,lexer/Lexer.x}

Renaming

- Ambiguous references are reported as warnings. The compiler picks the first choice and tries to compile with it.

location: `src/compiler/GF/Compile/Rename.hs`

Grammar Checking

- check that for every lin there is a corresponding fun or data definition
- check that for cat there is a corresponding lincat
- type checking

location: `src/compiler/GF/Compile/CheckGrammar.hs`

Partial Evaluation

- missing lindef declarations are automatically generated
- evaluation of the `oper` using call-by-name

location: ‘src/compiler/GF/Compile/Compute/Concrete.hs’

PMCFG generation

- convert GF rules to PMCFG rules
- this is where the gfo files are generated(?)

location `src/compiler/GF/Compile/GeneratePMCFG.hs`

Resources

Wiki

https://code.google.com/p/grammatical-framework/wiki/

Issue tracker

`https://code.google.com/p/grammatical-framework/issues/list`

GF Developer guide

`http://www.grammaticalframework.org/doc/gf-developers.html`

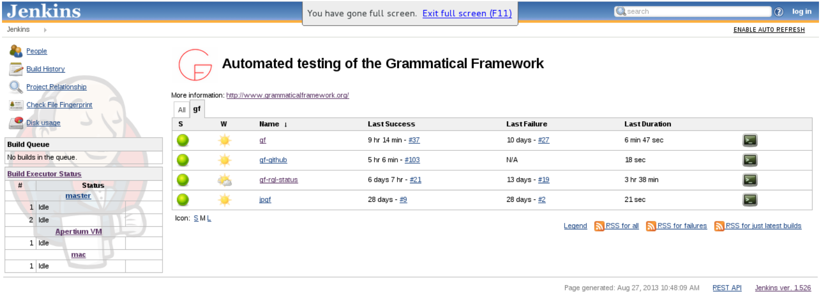
The future

C runtime

Location `src/runtime/c`

- python binding
- java binding (WIP)
- . . . other bindings?

Build server



Build Queue

No builds in the queue.

Build Executor Status

#	Status
master	
1	Idle
2	Idle
Apertium VM	
1	Idle
mac	
1	Idle

Automated testing of the Grammatical Framework

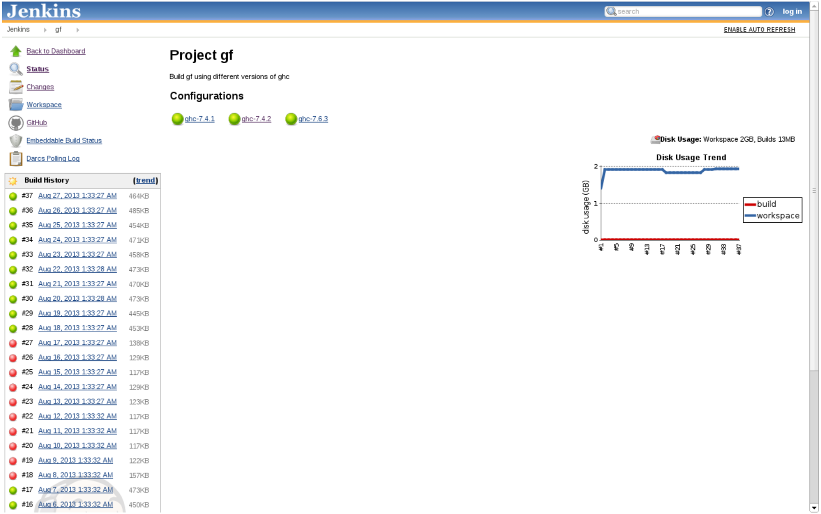
More information: http://www.grammaticalframework.org/

S	W	Name ↓	Last Success	Last Failure	Last Duration	
●	☀	gf	9 hr 14 min - #37	10 days - #27	6 min 47 sec	
●	☀	gf-github	5 hr 6 min - #103	N/A	18 sec	
●	⛅	gf-rgl-status	6 days 7 hr - #21	13 days - #19	3 hr 38 min	
●	☀	jpgf	28 days - #9	28 days - #2	21 sec	

Page generated: Aug 27, 2013 10:48:09 AM

Figure : Jenkins

Build server



Project gf

Build gf using different versions of ghc

Configurations

🟢 ghc-7.4.1 🟢 ghc-7.4.2 🟢 ghc-7.6.3

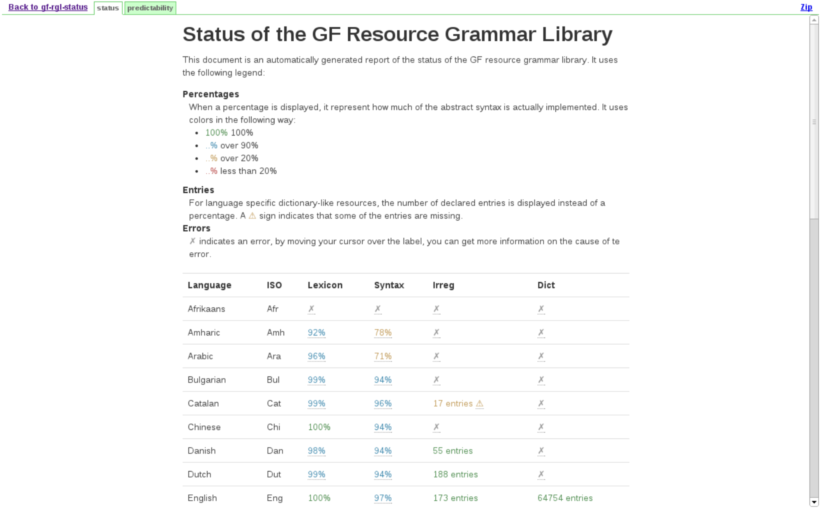
Build History (trend)

🟢	#37	Aug 27, 2013 1:33:27 AM	464KB
🟢	#36	Aug 26, 2013 1:33:27 AM	485KB
🟢	#35	Aug 25, 2013 1:33:27 AM	454KB
🟢	#34	Aug 24, 2013 1:33:27 AM	471KB
🟢	#33	Aug 23, 2013 1:33:27 AM	458KB
🟢	#32	Aug 22, 2013 1:33:28 AM	473KB
🟢	#31	Aug 21, 2013 1:33:27 AM	470KB
🟢	#30	Aug 20, 2013 1:33:28 AM	473KB
🟢	#29	Aug 19, 2013 1:33:27 AM	445KB
🟢	#28	Aug 18, 2013 1:33:27 AM	453KB
🔴	#27	Aug 17, 2013 1:33:27 AM	138KB
🔴	#26	Aug 16, 2013 1:33:27 AM	129KB
🔴	#25	Aug 15, 2013 1:33:27 AM	117KB
🔴	#24	Aug 14, 2013 1:33:27 AM	129KB
🔴	#23	Aug 13, 2013 1:33:27 AM	123KB
🔴	#22	Aug 12, 2013 1:33:32 AM	117KB
🔴	#21	Aug 11, 2013 1:33:32 AM	117KB
🔴	#20	Aug 10, 2013 1:33:32 AM	117KB
🔴	#19	Aug 9, 2013 1:33:32 AM	122KB
🔴	#18	Aug 8, 2013 1:33:32 AM	157KB
🟢	#17	Aug 7, 2013 1:33:32 AM	473KB
🟢	#16	Aug 6, 2013 1:33:32 AM	450KB

Disk Usage: Workspace 2GB, Builds 13MB

Figure : Jenkins gf page

Build server



Status of the GF Resource Grammar Library

This document is an automatically generated report of the status of the GF resource grammar library. It uses the following legend:

Percentages
When a percentage is displayed, it represent how much of the abstract syntax is actually implemented. It uses colors in the following way:

- 100% 100%
- ..% over 90%
- ..% over 20%
- ..% less than 20%

Entries
For language specific dictionary-like resources, the number of declared entries is displayed instead of a percentage. A ⚠ sign indicates that some of the entries are missing.

Errors
✗ indicates an error, by moving your cursor over the label, you can get more information on the cause of te error.

Language	ISO	Lexicon	Syntax	Irreg	Dict
Afrikaans	Afr	✗	✗	✗	✗
Amharic	Amh	92%	78%	✗	✗
Arabic	Ara	96%	71%	✗	✗
Bulgarian	Bul	99%	94%	✗	✗
Catalan	Cat	99%	96%	17 entries ⚠	✗
Chinese	Chi	100%	94%	✗	✗
Danish	Dan	98%	94%	55 entries	✗
Dutch	Dut	99%	94%	188 entries	✗
English	Eng	100%	97%	173 entries	64754 entries

Figure : RGL status

Git?

benefits

- more people know git than darcs
- more, easier to use (GUI) tools
- Github!
  - code & issues at the same place
  - good exposure
  - lots of young excited coders
  - facilitating casual contributions using github's pull requests

Git?

Separate repository for the RGL

- sugested on the ML
- more contributors to the RGL than the haskell code

Git?

Wishlist? Suggestions?