

The Why3 environment

<http://why3.lri.fr>

François Bobot, Jean-Christophe Filliâtre
Andrei Paskevich, Guillaume Melquiond
Claude Marché

INRIA-Saclay & LRI, CNRS & Université Paris-Sud
Orsay, France

Hi-Lite meeting — 29 nov 2011

Why3 core

- Why3 core: data
 - set of *theories*
 - a theory: logic declarations + set of *goals*
- Why3 core: operations
 - *transformations*: from a goal to some subgoals
 - *printers/drivers*: to call external provers on goals
- Distributed as an OCaml library and API

Why3 front-ends

- Why3ML:
 - *programs* in an ad-hoc annotated language
 - *VC generator*: produces theories
- C and Java front-ends: distributed with the former Why (version 2.30, October 2011)
 - produce Why3ML intermediate code
 - traceability from source (labels + source locs)
- Alfa/Ada frontend: GnatProve
 - also produce Why3ML intermediate code
 - also use traceability features

Example: scalar product

```

#define NMAX 10
#define B 0x1.1p-50

/*@ requires 0 <= n <= NMAX;
    @ requires \forall integer i; 0 <= i < n ==>
    @   \abs(x[i]) <= 1.0 && \abs(y[i]) <= 1.0 ;
    @ ensures
    @   \abs(\result - exact_scalar_product(x,y,n))
    @   <= n * B;
    @*/
double scalar_product(double x[], double y[],
                      int n) {
    double p = 0.0;
    for (int i=0; i < n; i++) p = p + x[i]*y[i];
    return p;
}

```

Why3 sessions and tools

- Session :
 - state for a given *project* (= set of files, where file = set of theories)
 - for each goal:
 - set of *proof attempts*:
prover, time limit given, result (Valid, Timeout, etc.), time
 - set of *transformations*:
which transformation, subgoals

→ tree structure, saved in XML file
- Tools operating on a session:
 - GUI (a.k.a. IDE)
 - batch replayer **DEMO**
 - regression testing
 - smoke detector
 - production of reports (LaTeX, HTML) and statistics

Example continued

	Alt-Ergo 0.93	Gamma 0.15.1
Proof obligations		
<i>Function scalar_product, default behavior</i>		
<i>loop invariant init</i>	0.22	
<i>assertion</i>	0.08	
<i>assertion</i>	0.09	
<i>assertion</i>	0.53	
<i>assertion</i>		0.02
<i>assertion</i>	0.07	
<i>assertion</i>	2.13	
<i>assertion</i>	1.56	
<i>loop invariant preservation</i>		
1	0.14	
2	0.20	
3	0.09	
4	0.09	
<i>normal postcondition</i>	0.05	
<i>Function scalar_product, Safety</i>		
<i>pointer dereference</i>	0.07	
<i>pointer dereference</i>	0.07	
<i>floating-point overflow</i>		0.06
<i>floating-point overflow</i>		0.06
<i>arithmetic overflow</i>	0.09	
<i>loop variant decreases</i>	0.10	

Some future works

- Sessions in Why3 core
 - available in Why3 library, with an API
 - any user can program his own kind of statistics or reports
- Why3ML library, with API
 - cloning of program modules
- Safer environment
 - Coq realizations of theories
- ...

Event: J.-C. Filliâtre Habilitation defense next Friday !