Tool session

NSV 3
FLOC 2010, Edinburgh

Organisation

- Quick presentation of an initial set of benchmarks
- 30 minutes break, except for tool owners
- Quick demo/synthesis of the benchmarks for each tool (5 minutes each)
- Discussion:
  - Tool competition as a stimulus for validation of numerical software?
  - How to constitute a widely accepted set of benchmarks? (on a wiki?)
  - How to classify tools?
  - What are the other initiatives in that direction?

Some comments on tools (I)

Huge variety of tools, able to prove/disprove a variety of properties...

- Model-checking/SMT: recent on numerical properties (e.g. SMV), full temporal logics. Checks properties, does not synthetise them.
- Abstract interpreters (e.g. Astrée, Fluctuat, Polyspace etc.): mostly invariant synthesis, but can check assertions. Generally used without functional specs, checking against implicit specs (no RTE, precision, etc.).
- Provers, based on theorem provers (e.g. Frama-C/Jessie...): can be used for inferring a wide class of (generally invariant) properties - human-assisted process.

Some comments on tools (II)

- Dynamic checkers, including testing environments, alternative arithmetics (stochastic, interval etc.). Checks some properties (temporal, precision etc.) on single executions. Can be used to formally disprove properties (counter-examples).

All these tools may consider idealized semantics of some level... (real number semantics, floating-point number semantics with some assumptions on the evaluation order etc.) hence might be difficult to compare.
They might also take rather different languages as input, or fragments of them...

Description of the benchmarks (from the FLUCTUAT distribution)

Name	Characteristic	Property of interest
absor.c	lin.	prec.
middl.c	lin.	prec./funct.
golde-a/b.c	(non-)lin., stat. loop	prec./stability
assoc.c	lin.	prec.
norma.c	non-lin.	comparison?
polyn.c	non-lin.	comparison?
inter-a/b.c	array/dyn. loop	comparison?
inver.c	non-lin.,pointer,bit,dyn. loop	convergence
filte.c	lin., unbounded loop	invariant
cav10.c	non-lin.	invariant

Description of the benchmarks (provided by Sylvie Boldo)

Name	Characteristic	Property of interest
Dekker.c	non-lin., stat. loop	exact+invariants/variants
Malcolm.c	lin.,dyn. loop	exact+invariants/variants
Sterbenz.c	lin.	exact
discri.c	non-lin.,interproc,stat. loop	precision
eps_line1.c	lin.	?
rec_lin2.c	lin.,stat. loop	?

Description of the benchmarks (provided by Nathalie Revol)

Name	Characteristic	Property of interest
count_to_6.c	non-lin.,stat. loop	precision
muller.c	non-lin.,stat. loop	precision/stability
sum.c	array,stat. loop	precision
integration.c	(non-)lin.,stat. loop	precision/stability
Rump.c	non-lin.	precision

Description of the benchmarks (provided by Stephen Siegel - FEVS)

(functional equivalence - C+MPI; careful: file IO)

Name	Characteristic	Property of interest
adder.c	lin.,stat. loop,array+para.	bad init
diffusion1d.c	lin.,stat. loop,array+para.	bad indices
diffusion2d.c	lin.,stat. loop,array+para.	wrong update
factorial.c	lin.,recursive	wrong init
fib.c	lin.,dyn. loop	bad scheme
gausselim.c	lin.,stat. loop,array	bad col. update
integrate.c	non-lin.,recursive+para.	bad use of MPI
laplace.c	lin.,dyn. loop,array+para.	bad topology

etc.

Description of the benchmarks (here, on the web, Gulwani's group)

Name	Characteristic	Property of interest
spmeter.c	lin.,unb. loop	invariant
hidden.c	lin.,dyn. loop	invariant
dloop.c	lin.,stat. loop	invariant

etc.

Starting a discussion

- How and where to set up an "unbiaised" set of benchmarks? Would extracts from the litterature be OK? (Gulavani etc.) NEC Small Static Analysis Benchmark... Relevance to "numerical software"?
- What initial format? C+annotations? what language of annotations (C assert? ACSL? other...)?
- What properties?
  - small, big? as an ultimate goal? checking just one specificity?
  - in real numbers? floating-point numbers?
  - value invariants (typical of automatic static analyzers)? functional properties (typical of proof based tools)?
  - oracles?
- What tool categories? Model-checkers, abstract interpreters, provers, dynamic analyzers etc.?