Some improvements to CodeContracts

CCVersions, Aggressive caching and a fresh new algorithm to infer object invariants

Mehdi Bouaziz's End-of-Internship talk

Joint work with Francesco Logozzo (mentor)

Code Contracts

- What:
  - methods preconditions
  - method postconditions
  - object invariants
- Verified:
  - dynamically at runtime
  - statically at compile time with Clousot
    - generates warnings and suggestions in Visual Studio

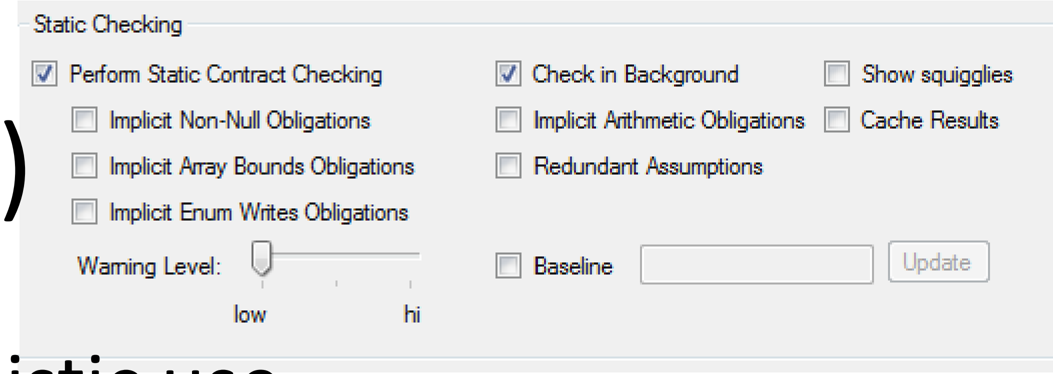
Contributions

- CCVersions
- Effective caching
- A new algorithm to infer object invariants

CCVersions

- Metrics on the verification over versions
- Why
  - Provide visual aid of use of CodeContracts
- How
  - Run Clousot on different repository versions
  - Draw nice graphs ;-)
- Demo

Caching (before)



Static Checking

☑ Perform Static Contract Checking
☑ Check in Background
□ Show squigglies

□ Implicit Non-Null Obligations
□ Implicit Arithmetic Obligations
□ Cache Results

□ Implicit Array Bounds Obligations
□ Redundant Assumptions

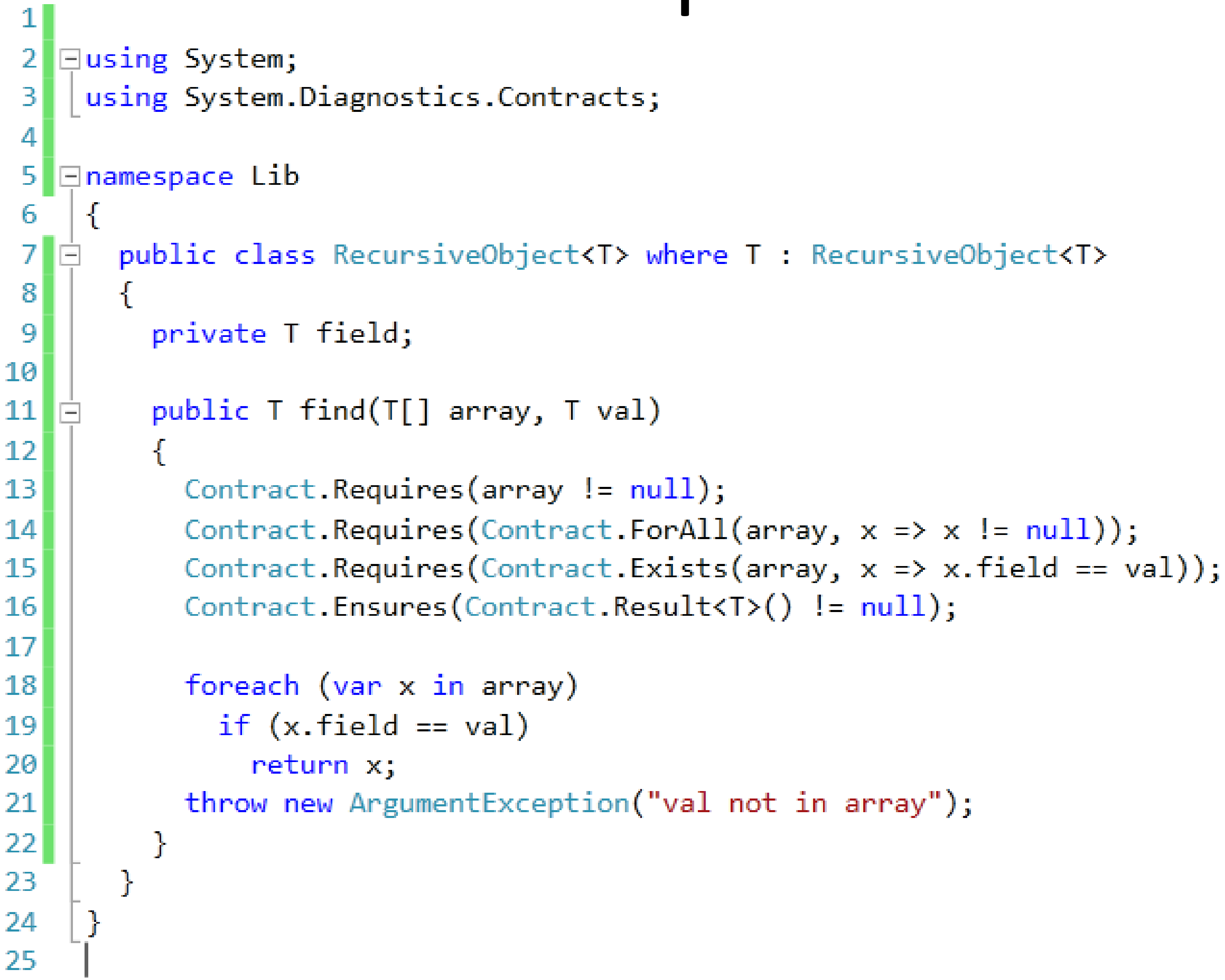
□ Implicit Enum Writes Obligations

Warning Level: low hi

□ Baseline

- Compulsory for realistic use
- Construct the CFG with contracts
- Hash the CFG
- If the same hash, read results from DB
- Otherwise, rerun the analysis
- Drawbacks:
  - Does not work with inferred contracts
  - Why? Very hard to serialize inferred contracts
    - Access paths, types, generic types …

Example



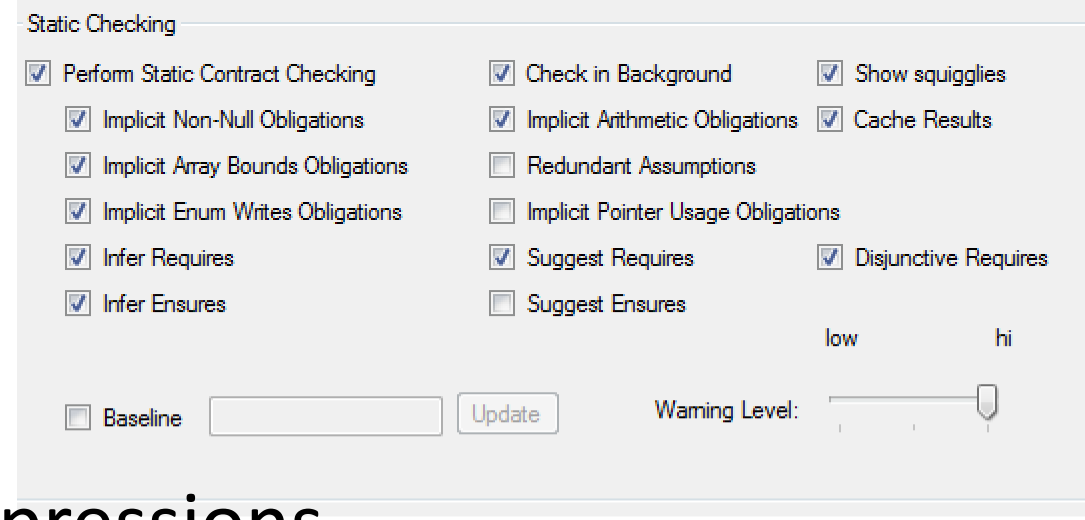
```
using System;
using System.Diagnostics.Contracts;

namespace Lib
{
    public class RecursiveObject<T> where T : RecursiveObject<T>
    {
        private T field;

        public T find(T[] array, T val)
        {
            Contract.Requires(array != null);
            Contract.Requires(Contract.ForAll(array, x => x != null));
            Contract.Requires(Contract.Exists(array, x => x.field == val));
            Contract.Ensures(Contract.Result<T>() != null);

            foreach (var x in array)
                if (x.field == val)
                    return x;
            throw new ArgumentException("val not in array");
        }
    }
}
```

Caching (now)



Static Checking

☑ Perform Static Contract Checking
☑ Implicit Non-Null Obligations
☑ Implicit Array Bounds Obligations
☑ Implicit Enum Writes Obligations
☑ Infer Requires
☑ Infer Ensures

☑ Check in Background
☑ Implicit Arithmetic Obligations
□ Redundant Assumptions
□ Implicit Pointer Usage Obligations
☑ Suggest Requires
□ Suggest Ensures

☑ Show squigglies
☑ Cache Results
☑ Disjunctive Requires

□ Baseline

Warning Level: hi

- Robust
- Serialize **all** the expressions
- Enable full pre/post/obj. inv. inference
  - Needed for Clousot integration with Roslyn
- Overhead on the first run: ~15%
  - To build the DB, to serialize the expressions
- Average time gain: >90% on big projects
  - Reached the limits of the DB
    - SQL Compact Edition vs. SQL Server

Assembly	Size (KB)	T1 (s)	T2 (s)	T2/T1	Size/T2 (KB/s)
System.Windows.Forms.dll	238309	4379	223.0	5.1%	1069
mscorlib.dll	201740	1988	199.0	10.0%	1014
System.Design.dll	141382	2186	134.0	6.1%	1055
System.Data.SqlXml.dll	69030	1168	65.0	5.6%	1062
Microsoft.JScript.dll	45502	985	50.3	5.1%	904
System.Web.Mobile.dll	44847	947	42.6	4.5%	1052
Microsoft.Build.Tasks.dll	41439	328	27.7	8.5%	1494
System.Drawing.dll	35148	159	27.5	17.3%	1279
System.Web.Services.dll	33509	1157	31.0	2.7%	1080
System.Deployment.dll	30429	291	18.7	6.4%	1623
System.DirectoryServices.dll	29643	218	21.8	10.0%	1358
System.Data.OracleClient.dll	23876	271	21.5	7.9%	1108
System.configuration.dll	23417	259	19.2	7.4%	1217
Microsoft.Build.Engine.dll	23089	445	17.0	3.8%	1362
Microsoft.VisualBasic.dll	22106	325	23.5	7.2%	943
System.Security.dll	19550	239	14.0	5.9%	1394
System.Runtime.Remoting.dll	18764	161	15.8	9.8%	1189
System.Messaging.dll	14897	102	12.6	12.4%	1181

Inference of object invariants

- #1 request by Clousot users:

  object invariant inference for readonly fields
- New algorithm: goal-oriented
- Better with an example!
- Integration in VS (stable) and in Roslyn (experimental)

Thank you!