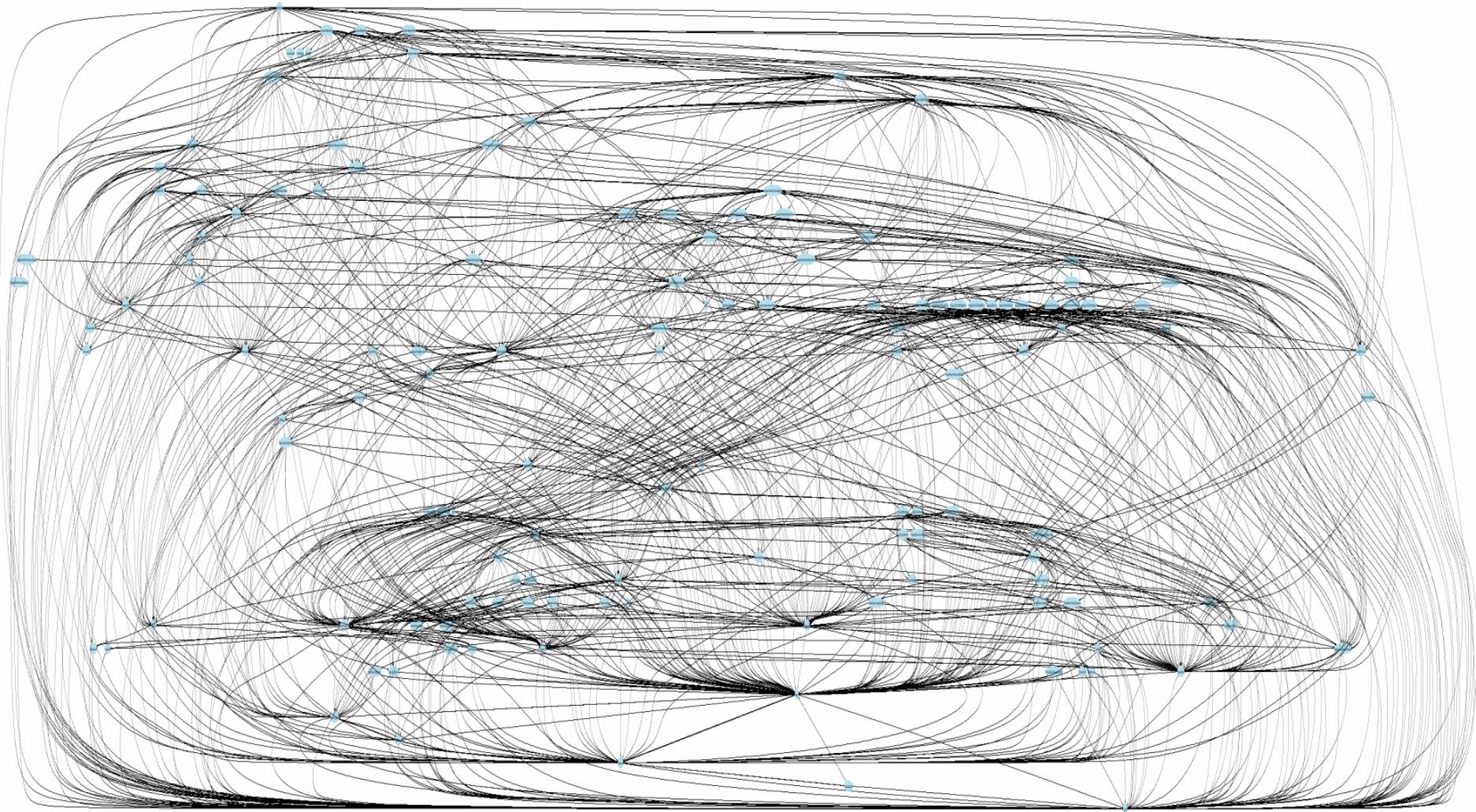


Detangling software dependency networks

Evelina Gabasova
@evelgab

Sometimes projects get ugly



Big ball of mud



Brian Foote and Joseph Yoder, 1997

Dependency networks

Effect of programming
language?

How do dependency networks
look in object-oriented and
functional code?

Scott Wlaschin

F# for fun and profit

Cyclic dependencies are e x

Cycles and modularity in t x

← → ↺

fsharpforfunandprofit.com/posts/cycles-and-modularity-in-the-wild/

☆ ☰

F# for fun and profit

Home Why use F#? Help with F# Site Contents Search

🐦

Part of the "Dependency cycles" series ([more](#))

Cycles and modularity in the wild

Comparing some real-world metrics of C# and F# projects

🐦 Tweet

84

(Updated 2013-06-15. See comments at the end of the post)

(Updated 2014-04-12. A [follow up post](#) that applies the same analysis to Roslyn)

This is a follow up post to two earlier posts on [module organization](#) and [cyclic dependencies](#).

I thought it would be interesting to look at some real projects written in C# and F#, and see how they compare in modularity and number of cyclic dependencies.

The plan

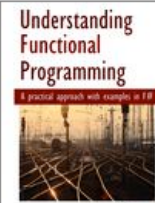
My plan was to take ten or so projects written in C# and ten or so projects written in F#, and somehow compare them.

I didn't want to spend too much time on this, and so rather than trying to analyze the source files, I thought I would cheat a little and analyze the compiled assemblies, using the [Mono.Cecil](#) library.

This also meant that I could get the binaries directly, using NuGet.

The projects I picked were:

GET THE BOOK



Understanding Functional Programming
A practical approach with examples in F#
by Scott Wlaschin

Understanding Functional Programming

A practical approach with examples in F#
by [Scott Wlaschin](#)

[Learn more](#)

Minimum price: \$19.99
Suggested price: \$29.99

Need more help?
Click here for F# training and consulting.

C# versus F#

The same execution runtime:

.NET framework

C#

Statically typed



Object oriented

Functional-first

Types see all types

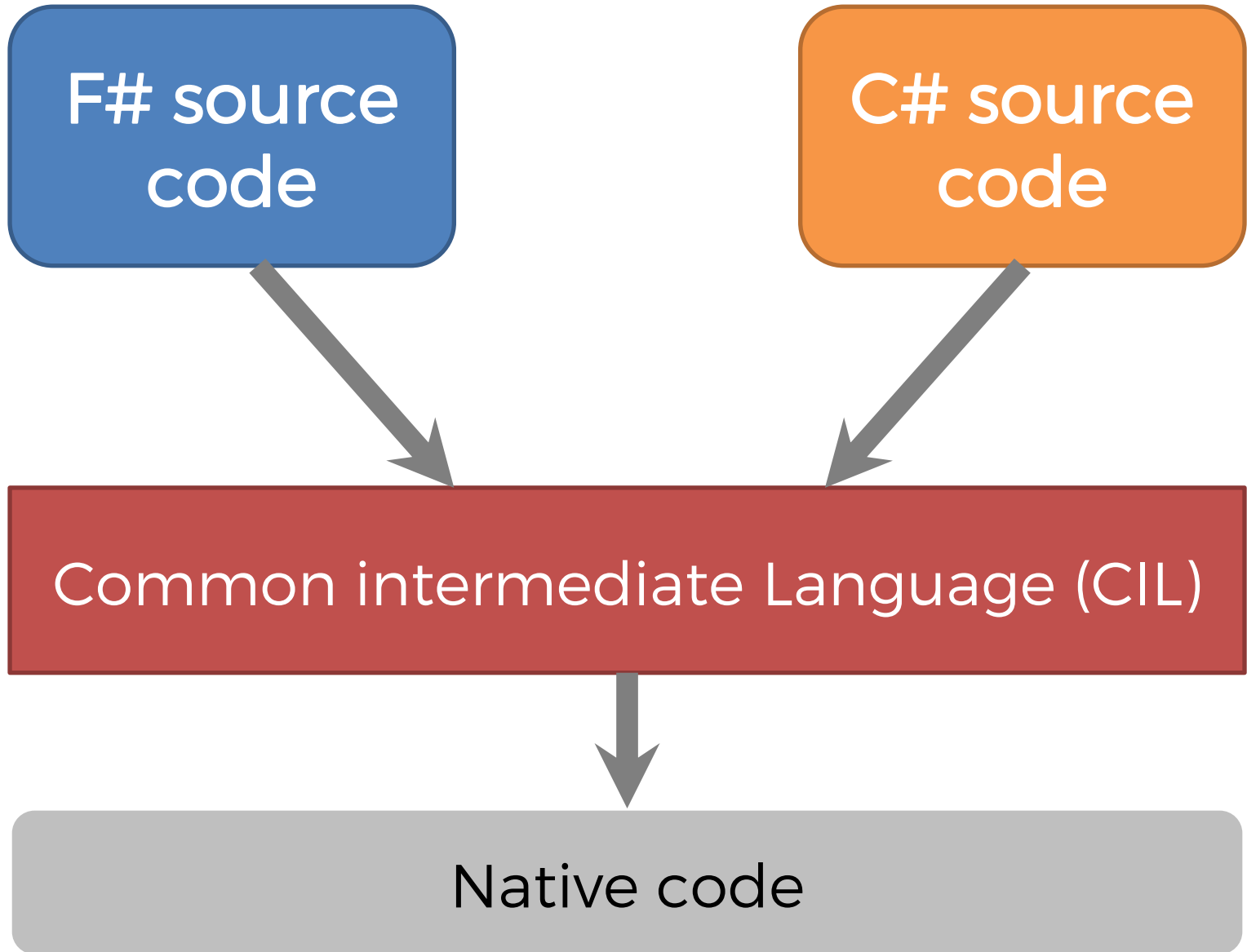
File order matters

F# source
code

C# source
code

Common intermediate Language (CIL)

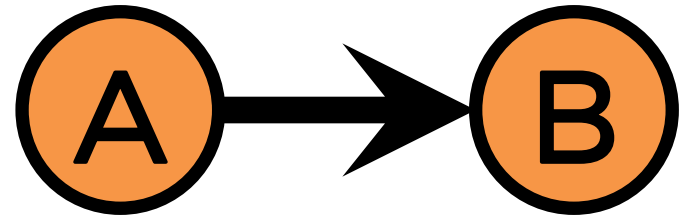
Native code



Structure of a network

Nodes

- Classes in C#
- Modules & types in F#



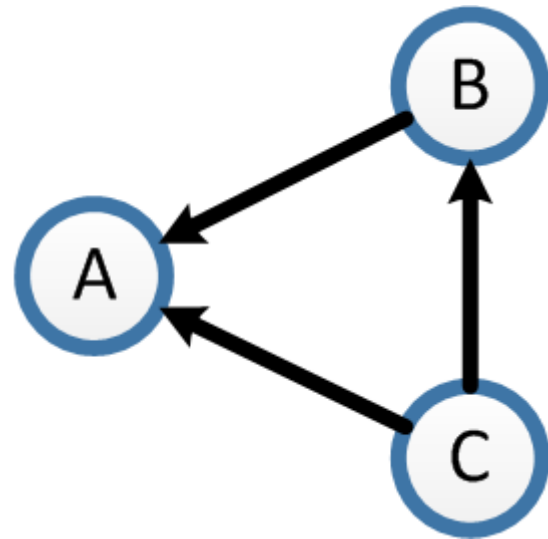
Links

- Class B inherits from class A or implements interface A
- Function in B calls a function or method from A
- Field, property, method or function in module B references A as a parameter or as a return type

Representing the network

```
type C = {Name : string}  
type B = {First: C; Second: C}  
module A =  
    let twice (x:C) = {First=x; Second=x}
```

	A	B	C
A	0	0	0
B	1	0	0
C	1	1	0



Comparing projects

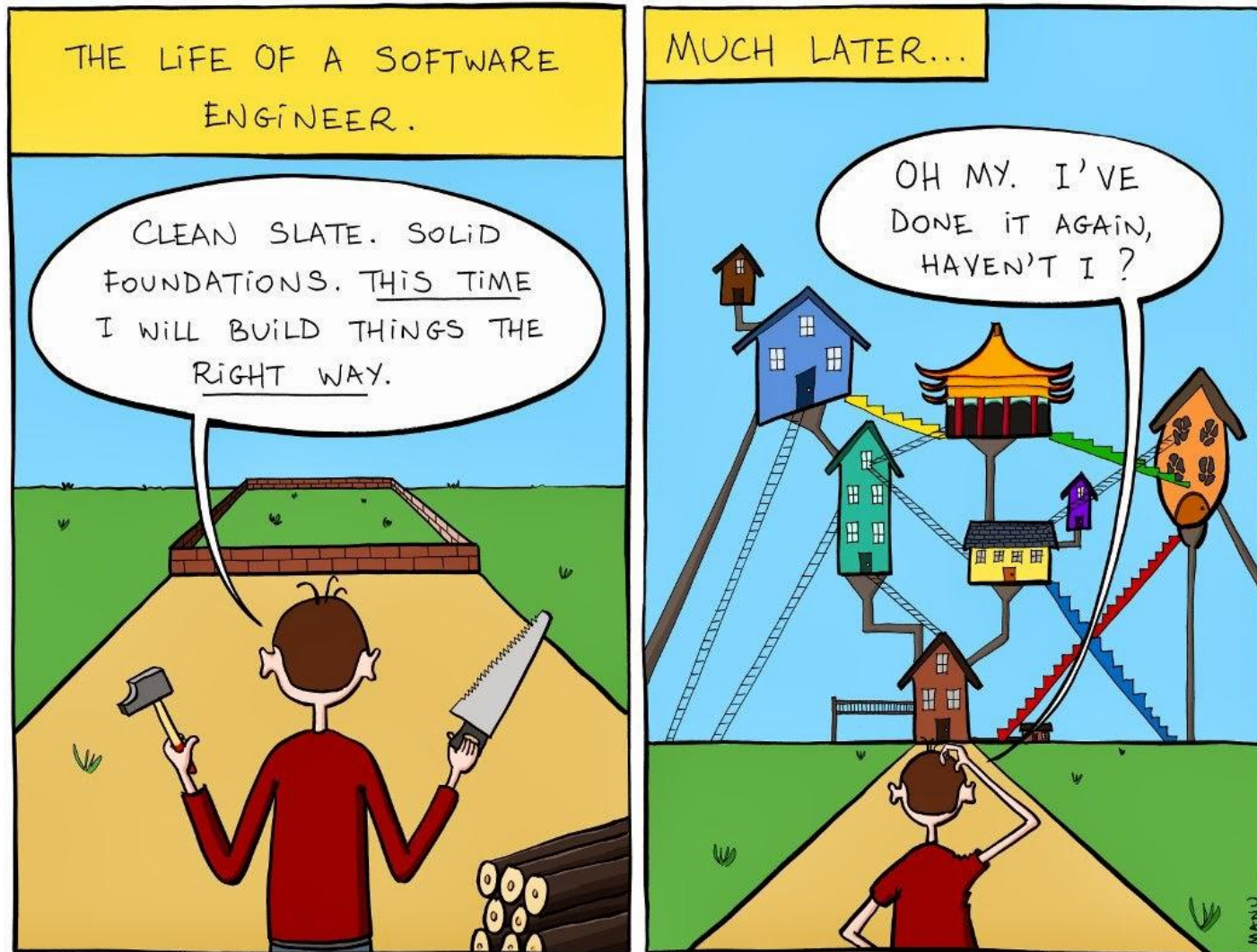
20 projects in each language

Hard to make an objective comparison

Antlr, AutoMapper, Castle, elmah, EntityFramework, FParsecCS, log4net, MathNet.Numerics, SignalR, Bcl.Runtime, Owin, Cecil, Moq, Nancy, Newtonsoft.Json, Nuget, NUnit, SpecFlow, xunit, YamlDotNet

canopy, Deedle, Fake, Foq, FParsecFS, FsCheck, FSharp.Compiler.Service, FSharp.Core, FSharp.Data, FSharp.Data.Twitter, FSharpx, FsPowerPack, FsSql, FsUnit, FsYaml, Storm, TickSpec, WebSharper, WebSharper.Core, WebSharper.Html

Good intentions

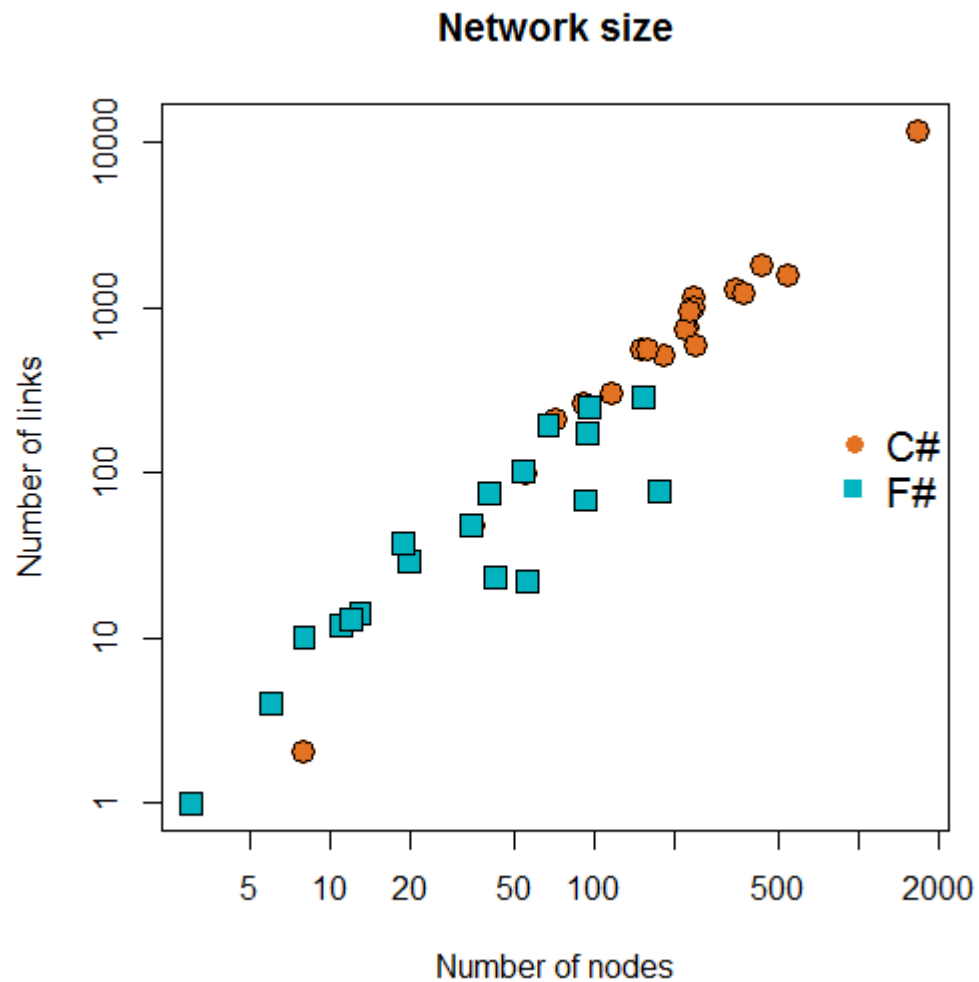


Focus

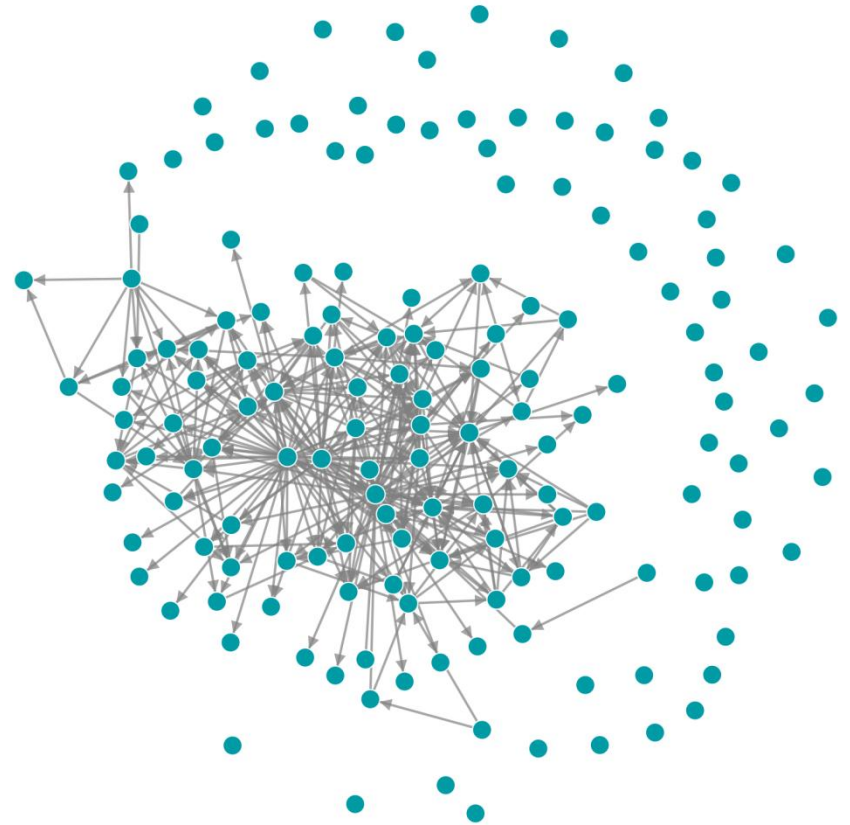
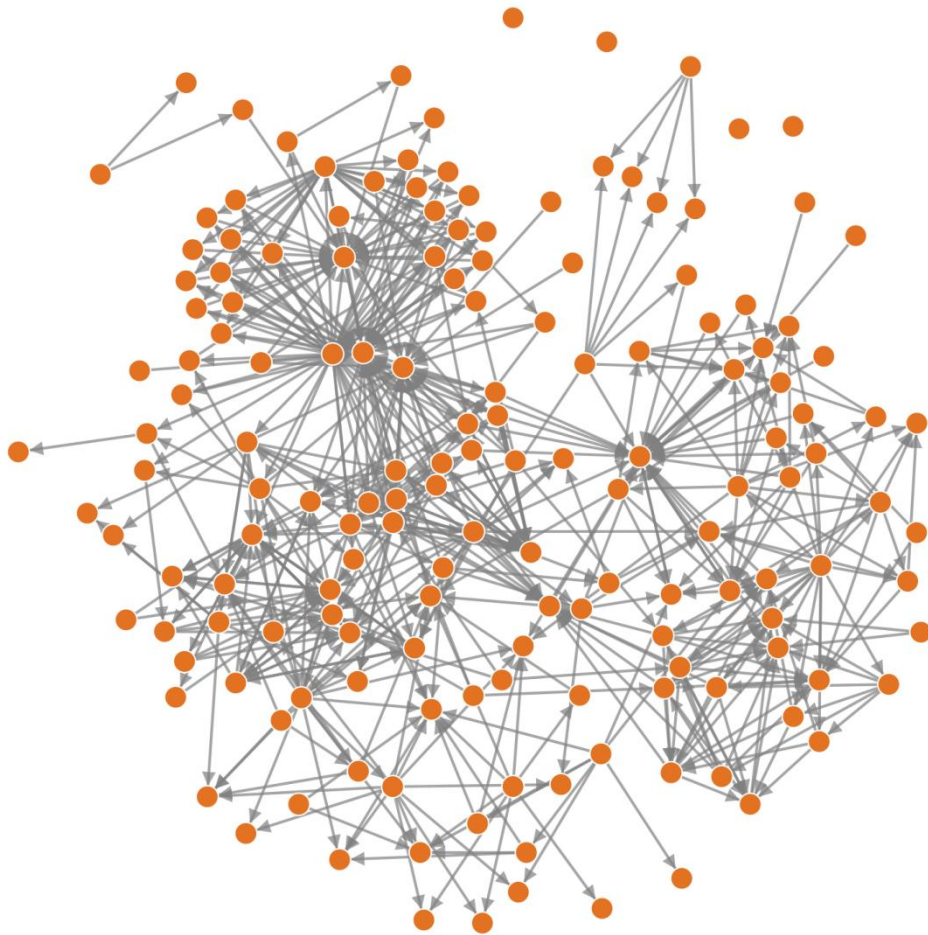
Json.NET and FSharp.Data



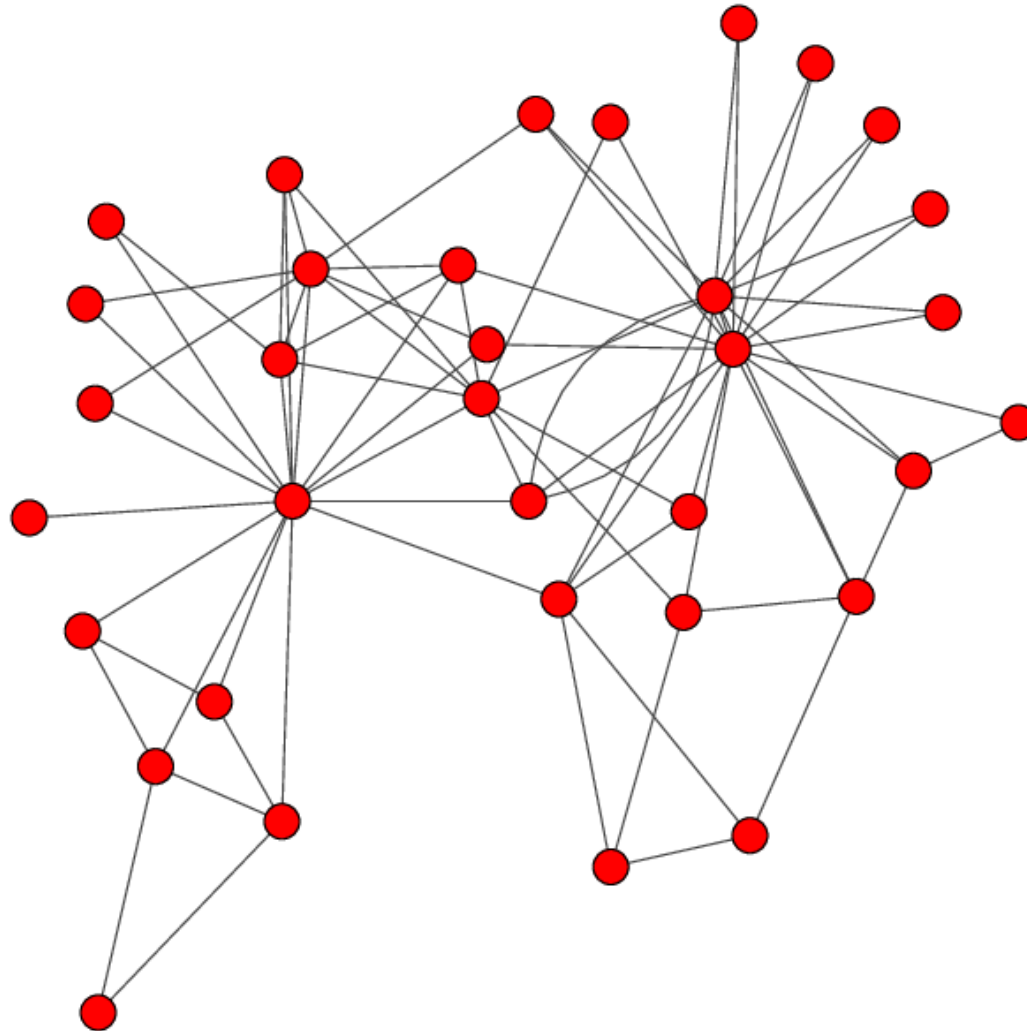
Network sizes



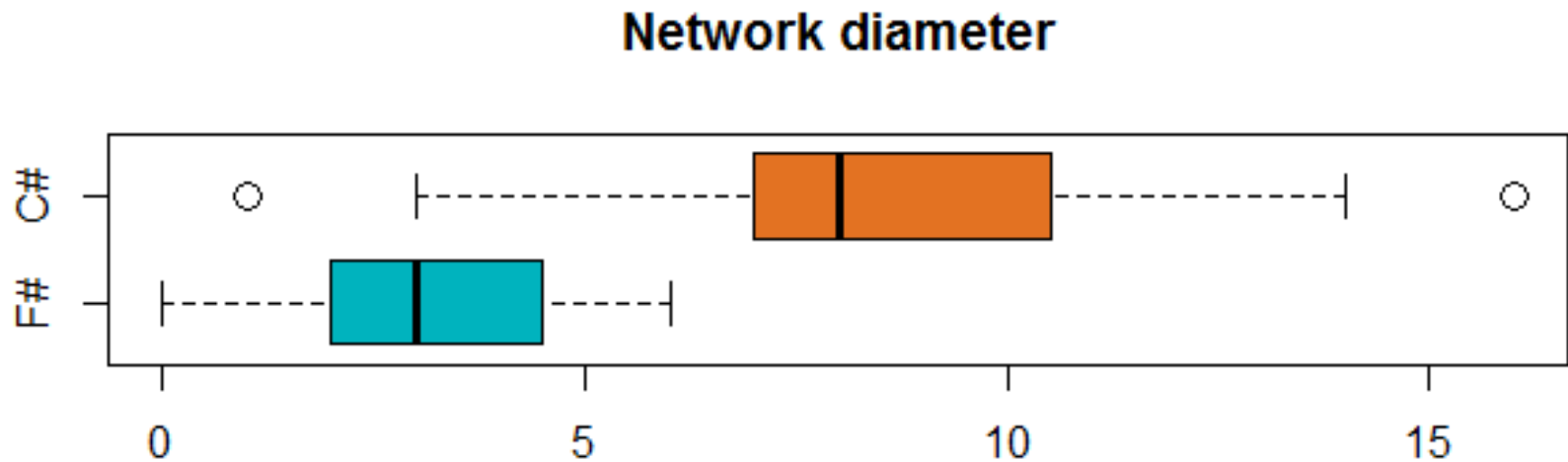
Network structures



Network diameter



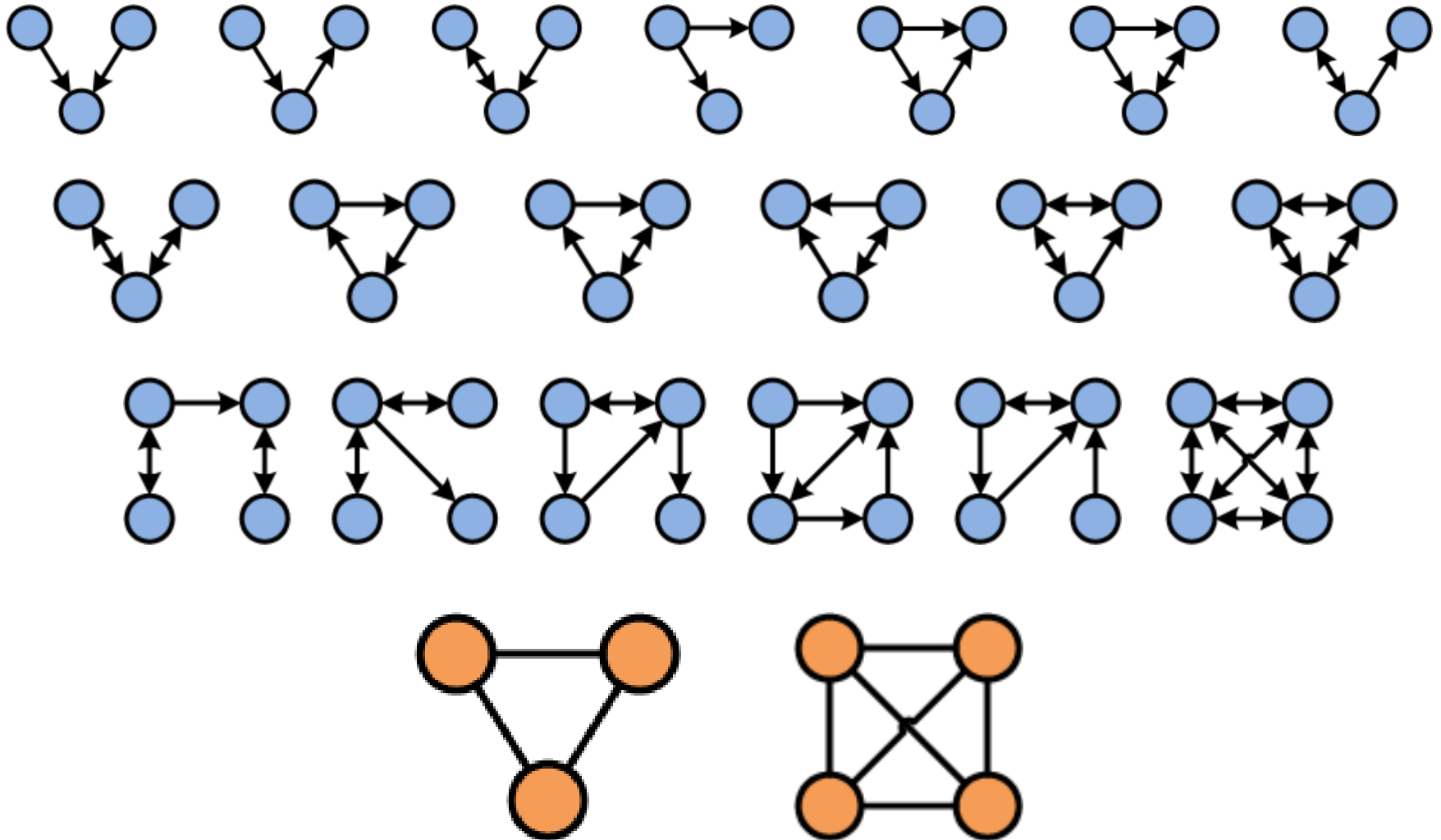
Diameters in C# and F#



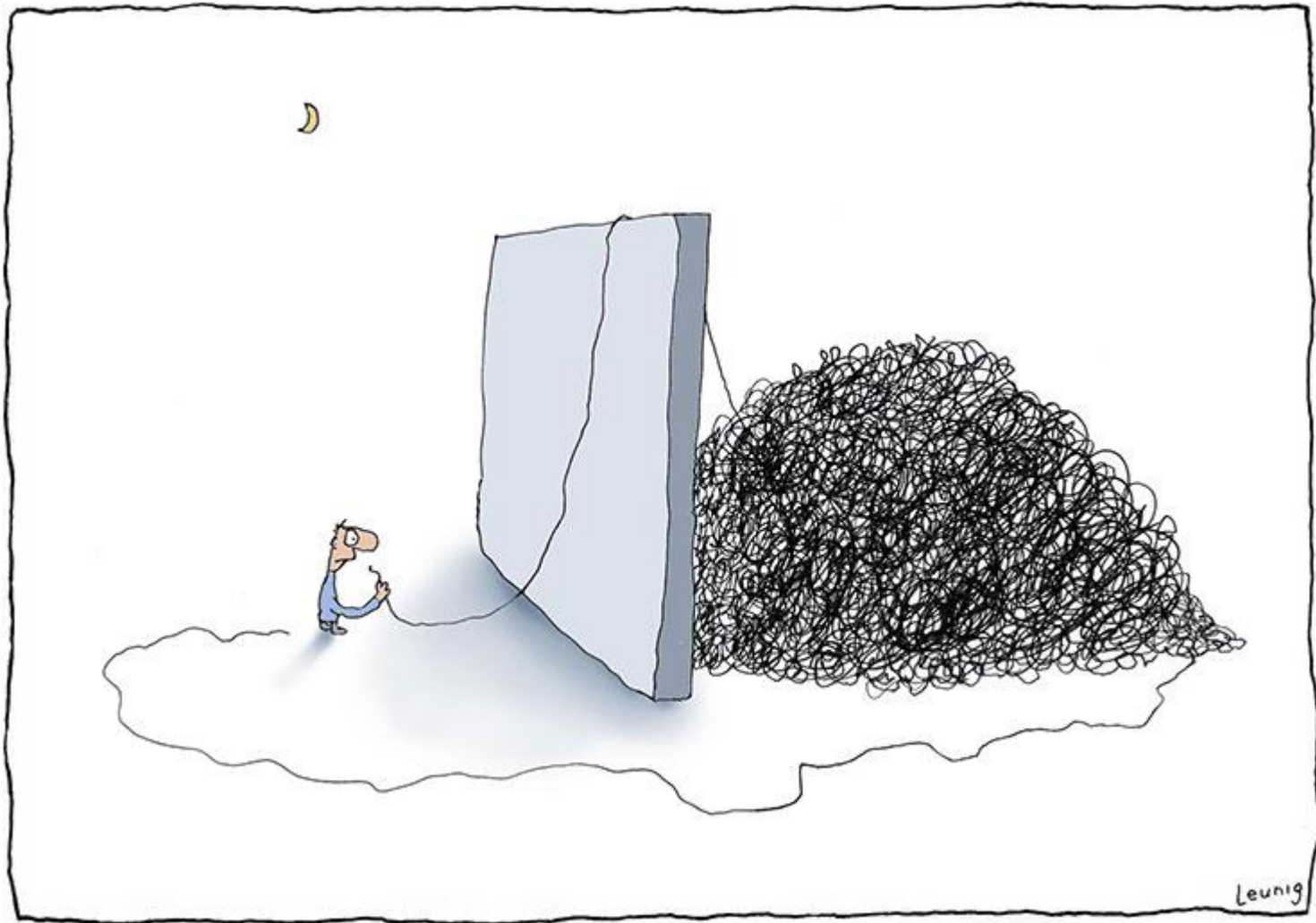
Spaghetti code



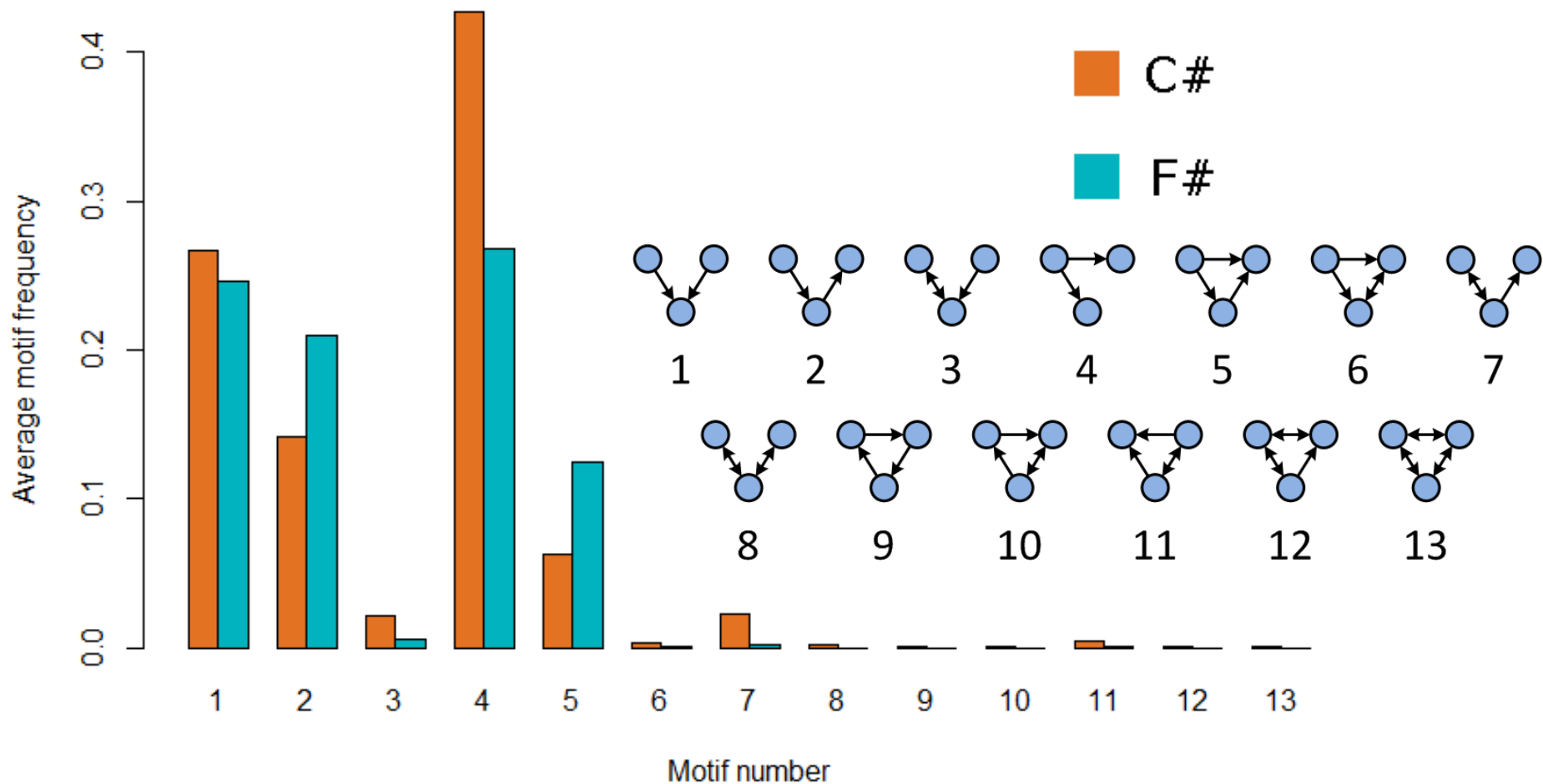
Motifs and cliques



I'll change this little thing...

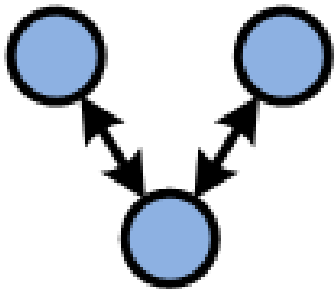


Frequent motifs

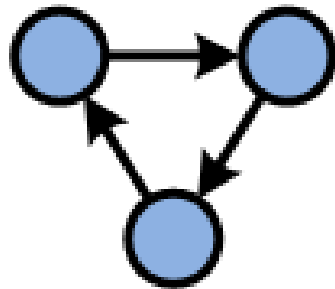


C#-only motifs

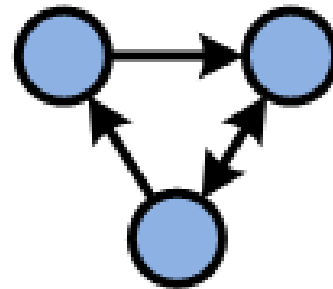
65%



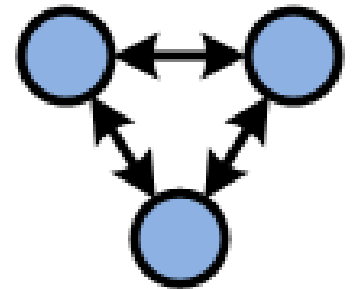
45%



70%

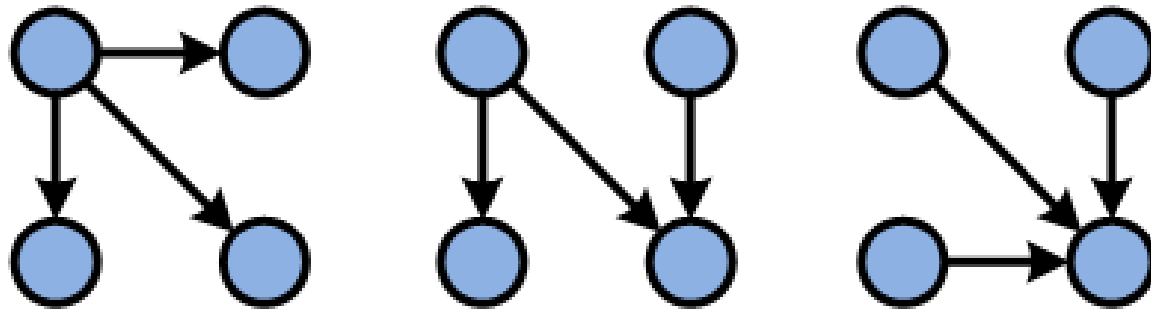


20%



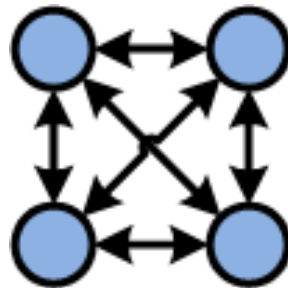
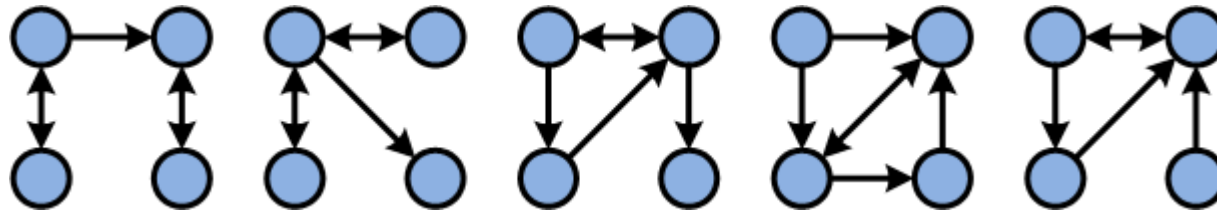
Motifs on 4 nodes

Most common
in F# and C#



C#-only motifs

129 C#-only motifs

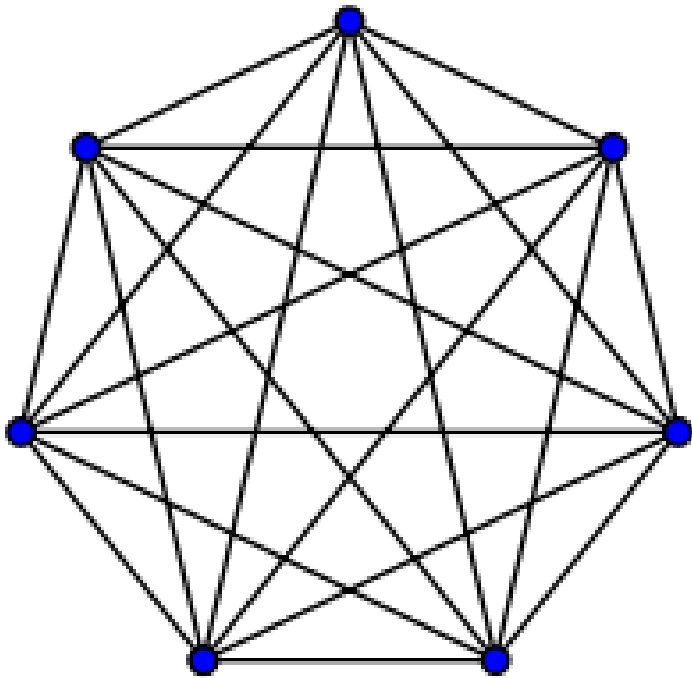


Entity.Framework, Json.NET, Mono.Cecil

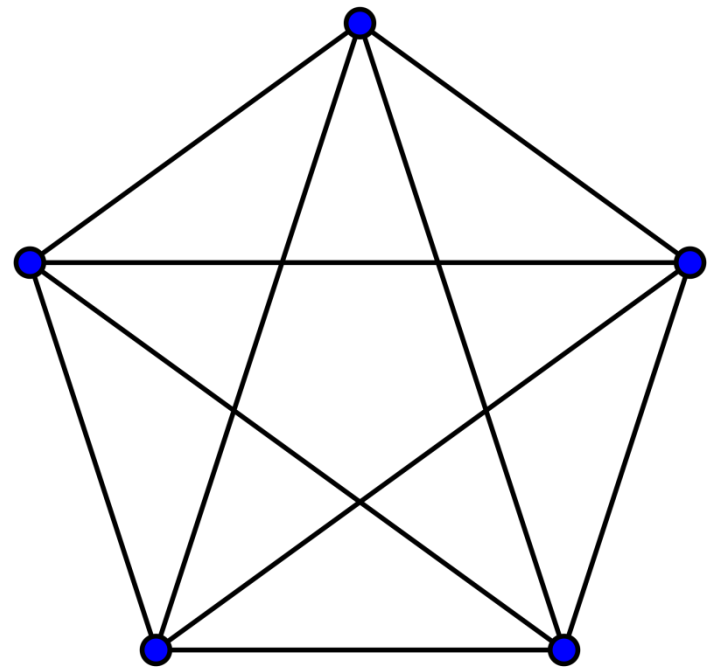


Largest cliques

Json.NET



FSharp.Data



Largest cliques

C#

Average: 5.6

Absolutely largest
clique: **11 nodes**

F#

Average: 3.9

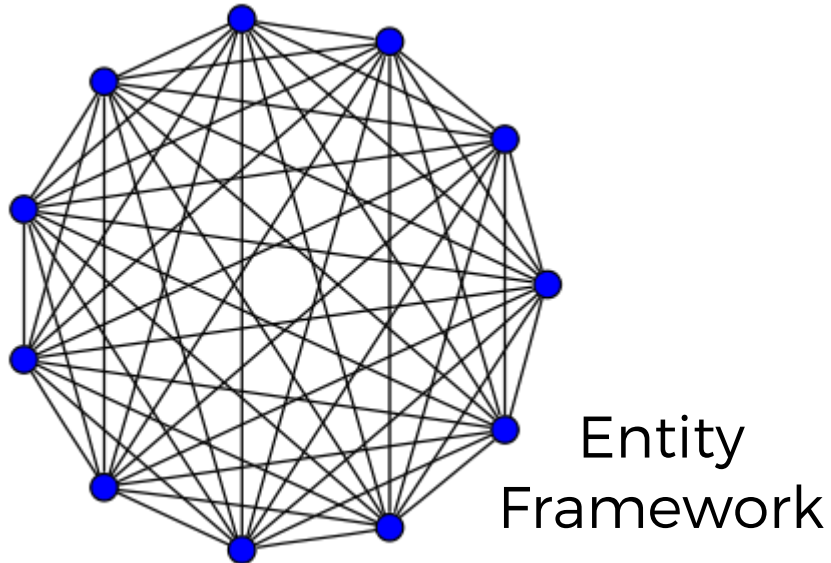
Absolutely largest
clique: **6 nodes**

Largest cliques

C#

Average: 5.6

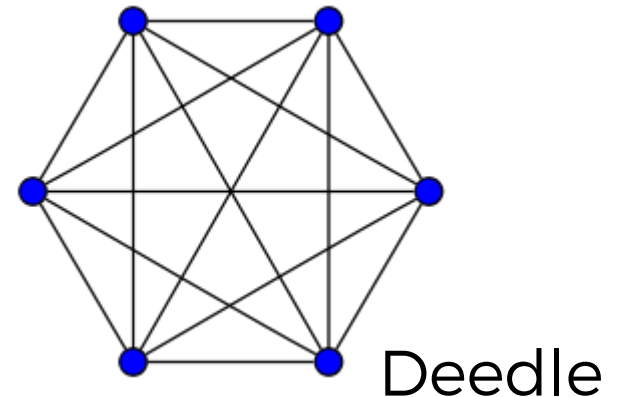
Absolutely largest
clique: **11 nodes**



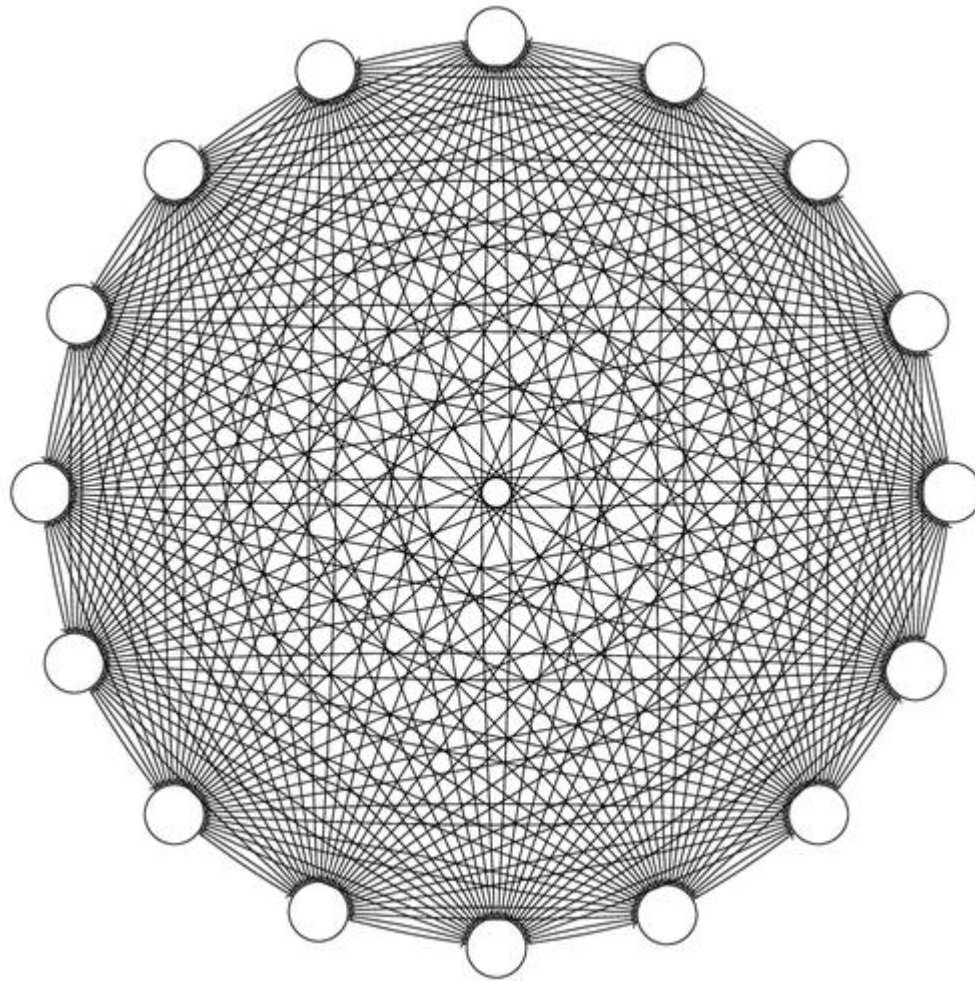
F#

Average: 3.9

Absolutely largest
clique: **6 nodes**



Roslyn





中水電海外投资有限公司上馬相迪A水电站项目经理部
SINOHYDRO RESOURCES LTD. UPPER MARSHANDI A HYDROELECTRIC PROJECT



Caution, Object
May Fall

सावधान केही वस्तु
खस्न सक्छ ।

Taming complexity

Beware of cyclic dependencies.

Language may help.

It's harder to create cycles in F#!

Thank you!

@evelgab

evelina@evelinag.com

fsharp.org

F# eXchange 2015

17 April, London

