

COMPSCI 734 –Introduction to Part I

Radu Nicolescu

Department of Computer Science
University of Auckland

5 Mar 2019

- ① Recommended Preparation: COMPSCI 335
- ② Parallel and Distributed Computing with FP
- ③ Monads in FP
- ④ Assignment #1
- ⑤ Miscellanea

Outline

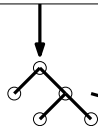
- 1 Recommended Preparation: COMPSCI 335
- 2 Parallel and Distributed Computing with FP
- 3 Monads in FP
- 4 Assignment #1
- 5 Miscellanea

Recommended Preparation: COMPSCI 335 in a nutshell

```
let young = { c ∈ Customers | c.Age < 60 }
```

```
var young = Customers.Where(c => c.Age < 60);
```

lazy function



```
http://.../Customers()?$filter=Age lt 60
```

```
DECLARE @p0 Int = 60
SELECT * FROM [Customers] AS [t0]
WHERE [t0].[Age] < @p0
```

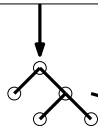
- Tutorials? One or two?

Recommended Preparation: COMPSCI 335 in a nutshell

```
let young = { c ∈ Customers | c.Age < 60 }
```

```
var young = Customers.Where(c => c.Age < 60);
```

lazy function



```
http://.../Customers()?$filter=Age lt 60
```

```
DECLARE @p0 Int = 60
SELECT * FROM [Customers] AS [t0]
WHERE [t0].[Age] < @p0
```

- **Tutorials?** One or two?

Outline

- 1 Recommended Preparation: COMPSCI 335
- 2 **Parallel and Distributed Computing with FP**
- 3 Monads in FP
- 4 Assignment #1
- 5 Miscellanea

What we may want more?

- All cores of your machine? Of a cluster? Available in the cloud?
- Automatic scaling, no explicit threads?
- Message-based communication?
- ...
- Nice models, to cover them all?
- Actor model: tasks with unbounded input queues, aka **mailboxes**
- CSP model: tasks with bounded input buffers, typically 0 size = **rendezvous, channels**

What we may want more?

- All cores of your machine? Of a cluster? Available in the cloud?
- Automatic scaling, no explicit threads?
- Message-based communication?
- ...
- Nice models, to cover them all?
- Actor model: tasks with unbounded input queues, aka **mailboxes**
- CSP model: tasks with bounded input buffers, typically 0 size = **rendezvous, channels**

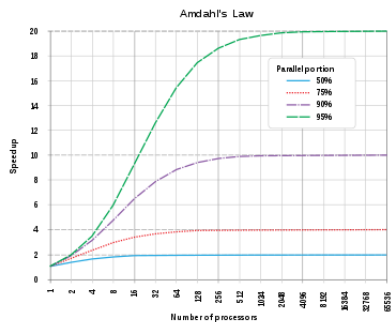
Parallel Computing

- Data Parallelism in FP

```
1 var young =  
2   Customers.AsParallel().Where(c => c.Age < 60);
```

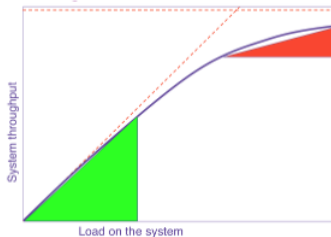
- Task Parallelism in FP
- Concurrency and communication in FP: **Actors**, **CSP Channels**
- Best Practices, Granularity / Partitioning, (False) Sharing

Parallel Limits – Amdahl



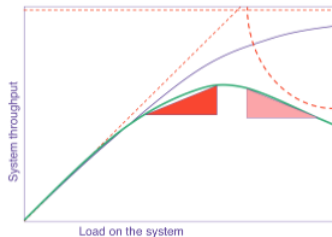
Parallel Limits – Gunther (Universal Scalability Law)

C. Diminishing returns from contention



$$\alpha \gg 0, \beta = 0$$

D. Negative returns from incoherency



$$\alpha \gg 0, \beta > 0$$

Outline

- 1 Recommended Preparation: COMPSCI 335
- 2 Parallel and Distributed Computing with FP
- 3 Monads in FP**
- 4 Assignment #1
- 5 Miscellanea

Monads – One ring to rule them all?

- Ubiquitous but hidden theoretical abstraction
- Covers: nullables, options, containers, arrays, lists, collections, virtual sequences, tasks, asyncs, actors, cloudlets, ...
- You Could Have Invented Monads!
And Maybe You Already Have!

Monads – One ring to rule them all?

- Ubiquitous but hidden theoretical abstraction
- Covers: nullables, options, containers, arrays, lists, collections, virtual sequences, tasks, asyncs, actors, cloudlets, ...
- **You Could Have Invented Monads!
And Maybe You Already Have!**

Basic discrete algebra – quiz

- How do you call these?

1	$(x \oplus y) \oplus z = x \oplus (y \oplus z)$	// \oplus is associative
2	$x \oplus y = y \oplus x$	// \oplus is commutative
3	$x \oplus e = e \oplus x = x$	// e is identity or neutral
4	$x \oplus x^{-1} = x^{-1} \oplus x = e$	// x^{-1} is the inverse

Outline

- 1 Recommended Preparation: COMPSCI 335
- 2 Parallel and Distributed Computing with FP
- 3 Monads in FP
- 4 Assignment #1**
- 5 Miscellanea

Assignment #1

- Parallel experiments and evaluations based on the classical LCS (Longest Common Subsequence) algorithm
- Languages: F#, C#
- Feedback: too much, too less, ambiguities, deadlines?
- Weight: 15%
- Bonuses? JS, Python, Golang?

Assignment #1

- Parallel experiments and evaluations based on the classical LCS (Longest Common Subsequence) algorithm
- Languages: F#, C#
- Feedback: too much, too less, ambiguities, deadlines?
- Weight: 15%
- Bonuses? JS, Python, Golang?

Outline

- 1 Recommended Preparation: COMPSCI 335
- 2 Parallel and Distributed Computing with FP
- 3 Monads in FP
- 4 Assignment #1
- 5 **Miscellanea**

Miscellanea

- Software: F#, C#, extra libraries, command-line
- F# Links
 - F# for fun and profit
<https://fsharpforfunandprofit.com/>
 - F# Foundation
<https://fsharp.org/>
 - MSDN F# Guide
<https://docs.microsoft.com/en-us/dotnet/fsharp/>
- Exam sample with model solution (part I)

Miscellanea

- Software: F#, C#, extra libraries, command-line
- F# Links
 - F# for fun and profit
<https://fsharpforfunandprofit.com/>
 - F# Foundation
<https://fsharp.org/>
 - MSDN F# Guide
<https://docs.microsoft.com/en-us/dotnet/fsharp/>
- Exam sample with model solution (part I)

Miscellanea

- Software: F#, C#, extra libraries, command-line
- F# Links
 - F# for fun and profit
<https://fsharpforfunandprofit.com/>
 - F# Foundation
<https://fsharp.org/>
 - MSDN F# Guide
<https://docs.microsoft.com/en-us/dotnet/fsharp/>
- Exam sample with model solution (part I)