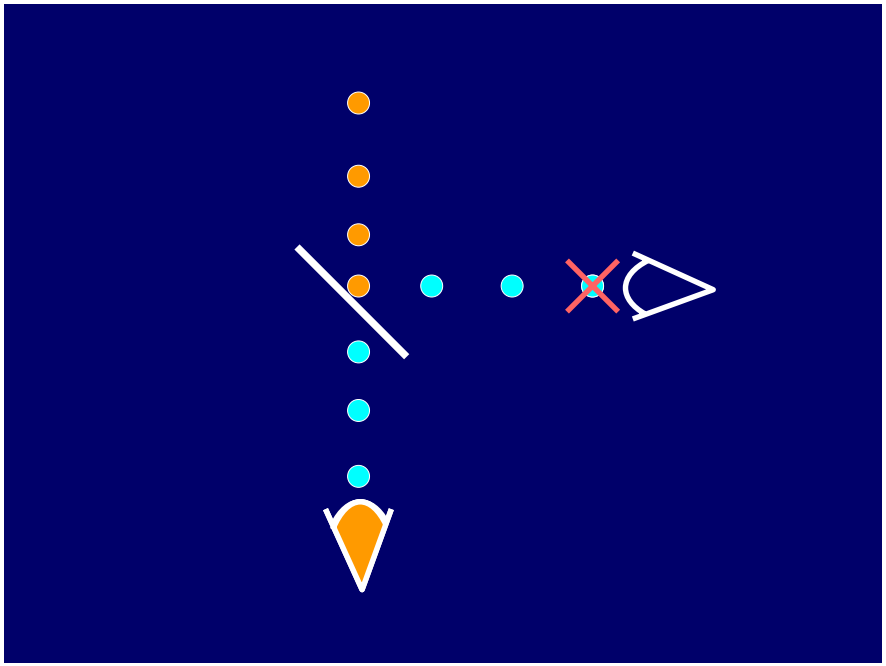
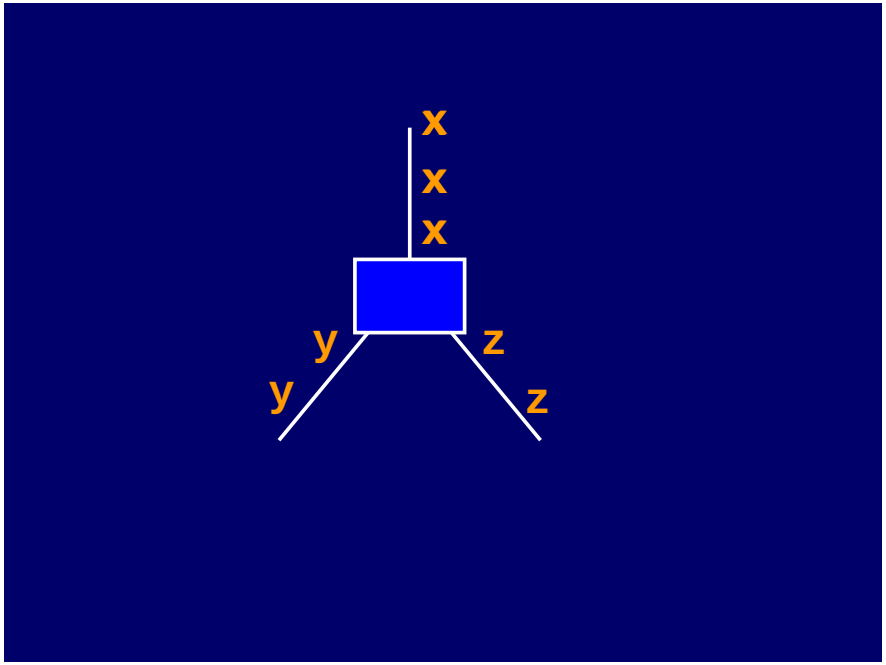
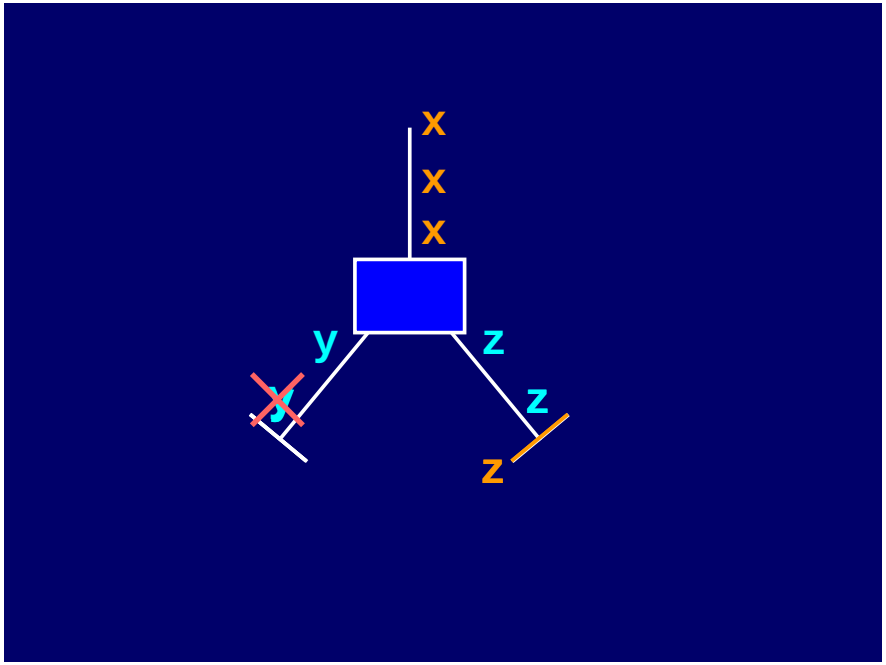
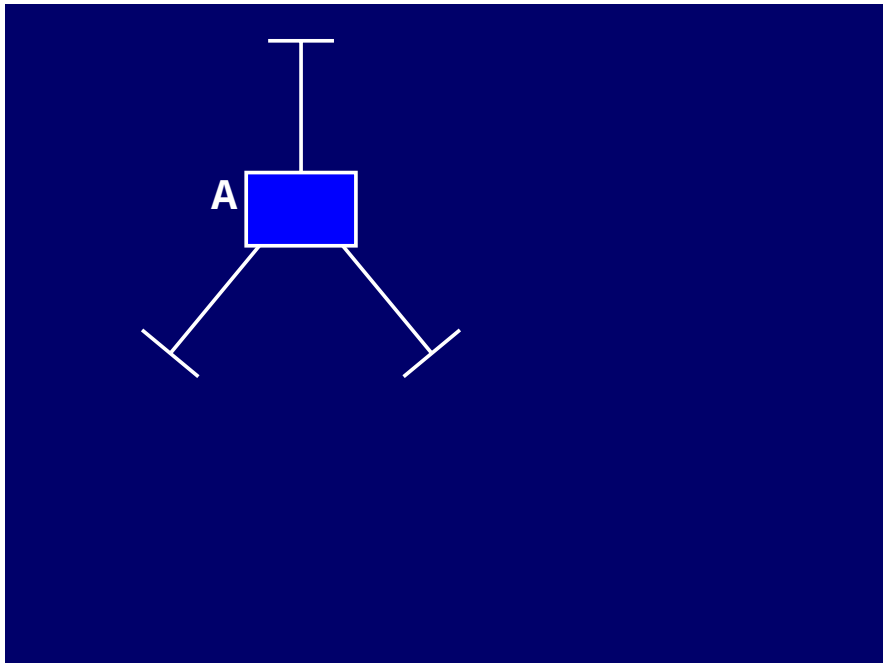
Wavenets

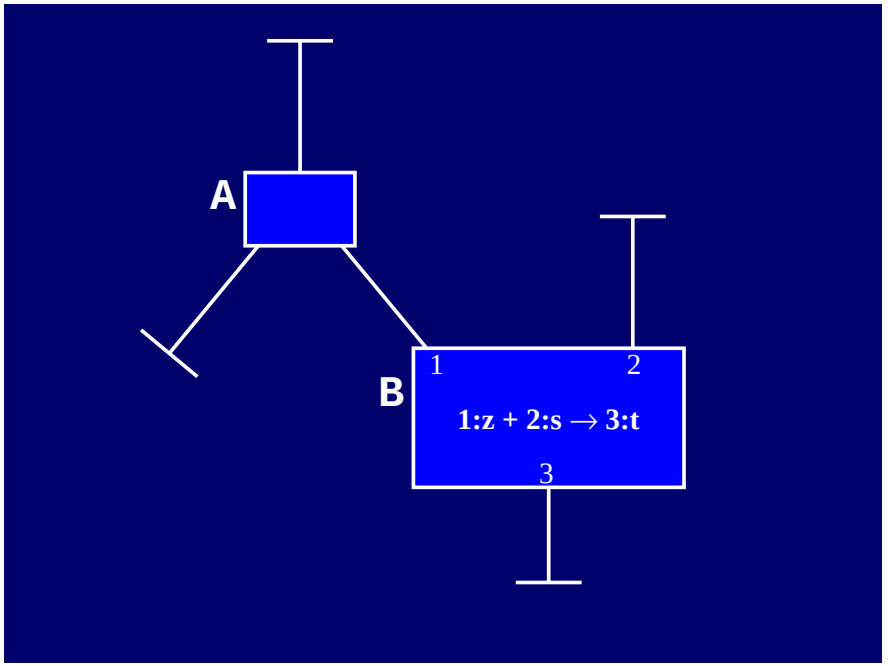
John Chapin
MIT

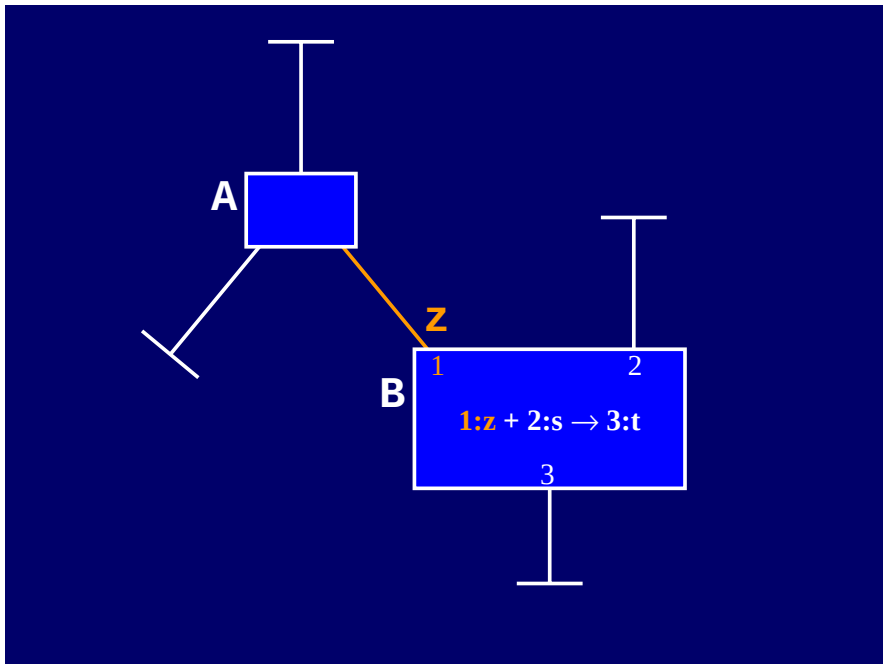


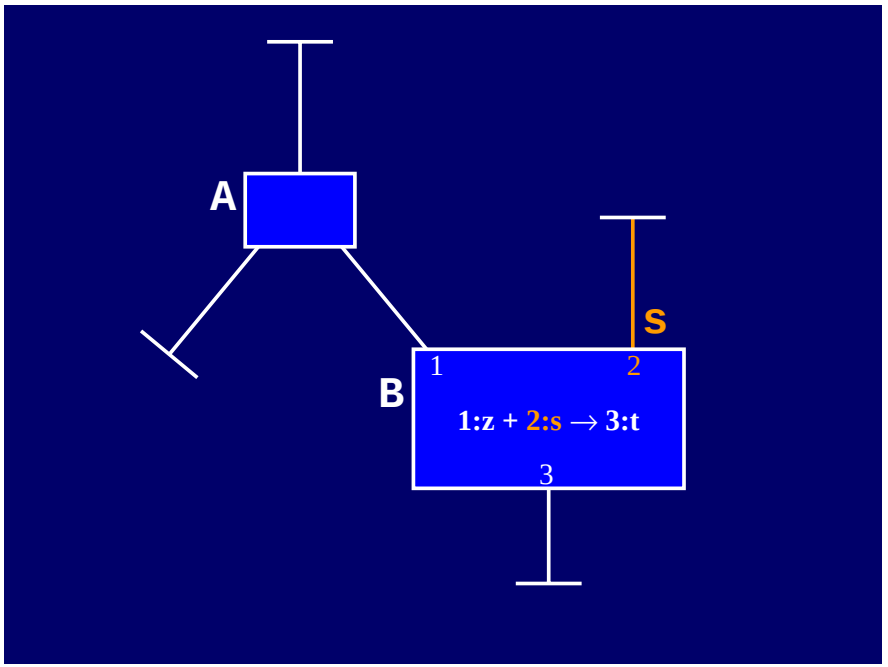


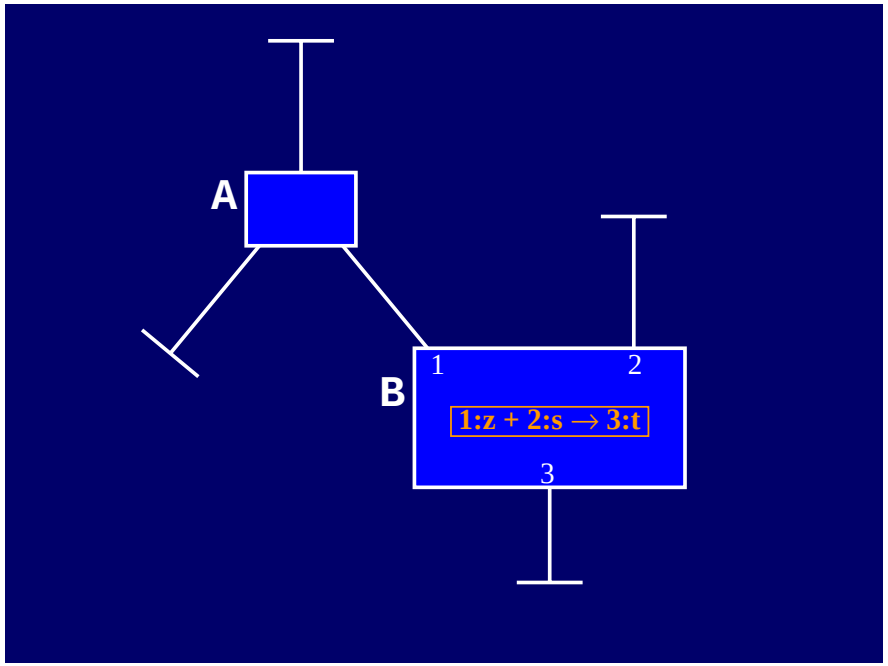


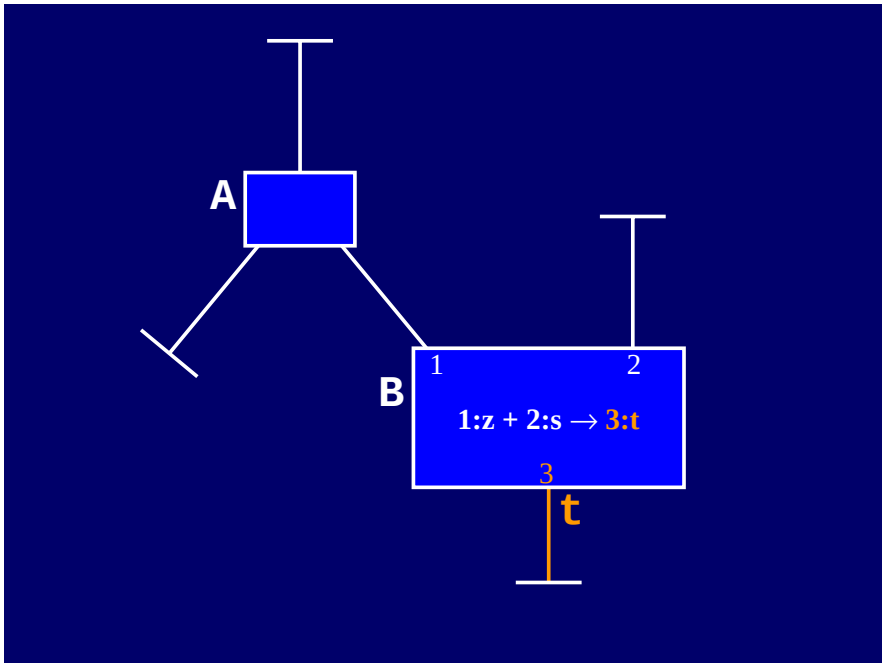

A

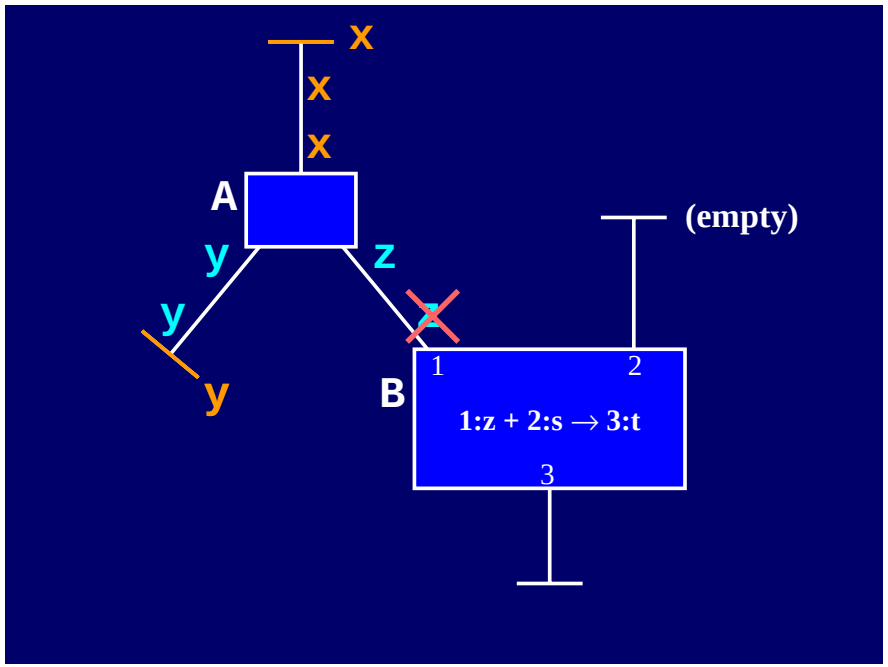

A
B
1
2
1:z + 2:s → 3:t
3

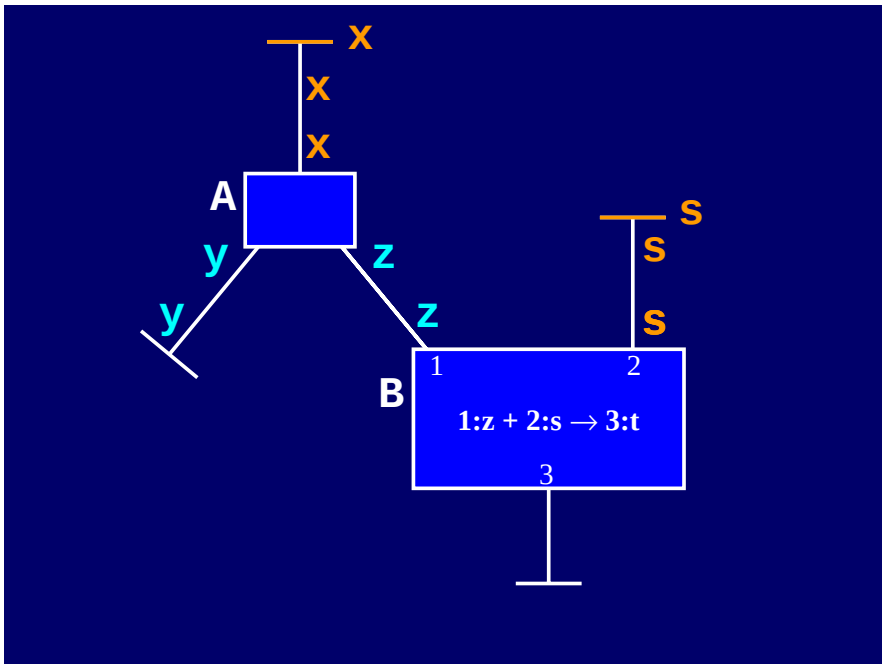

A
z
B
1
2
1:z + 2:s → 3:t
3

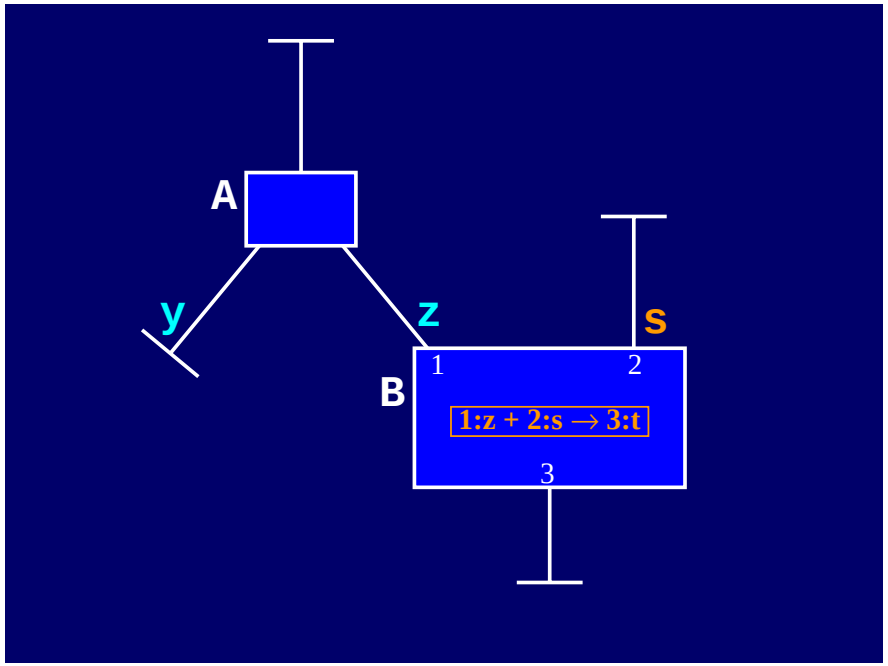

A
s
B
1
2
1:z + 2:s → 3:t
3

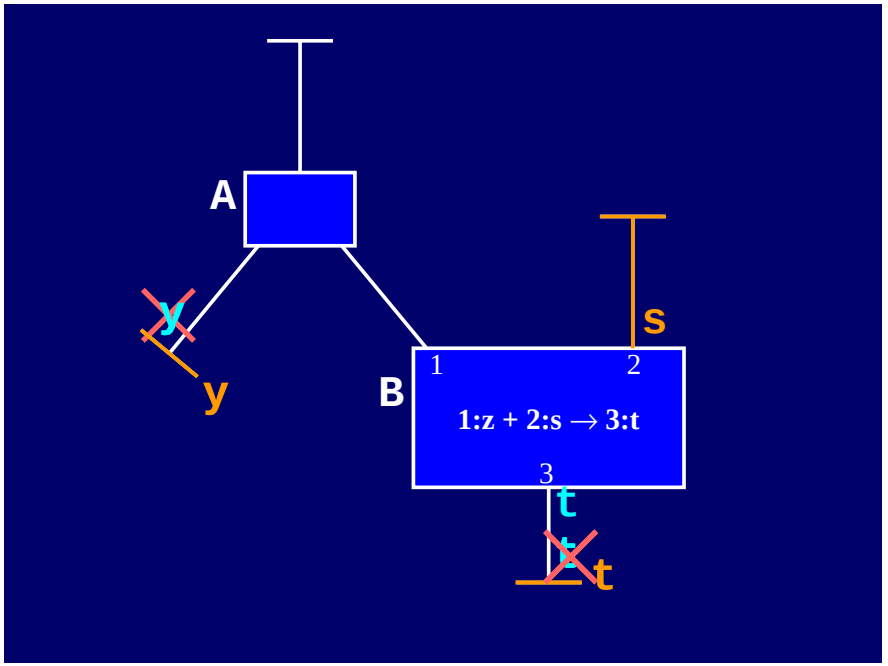

A
B
1
2
1:z + 2:s → 3:t
3

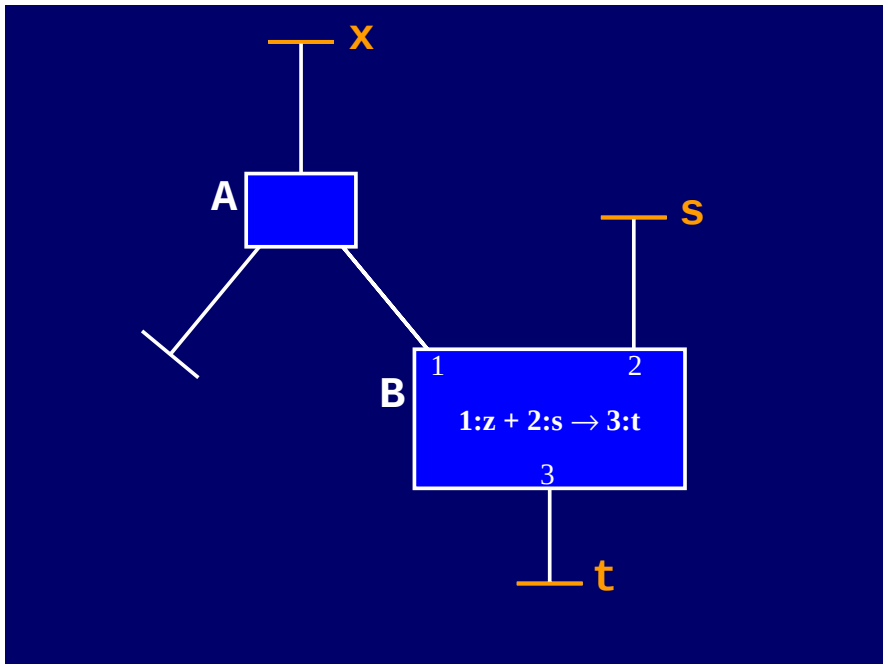

A
B
1
2
1:z + 2:s → 3:t
3
t

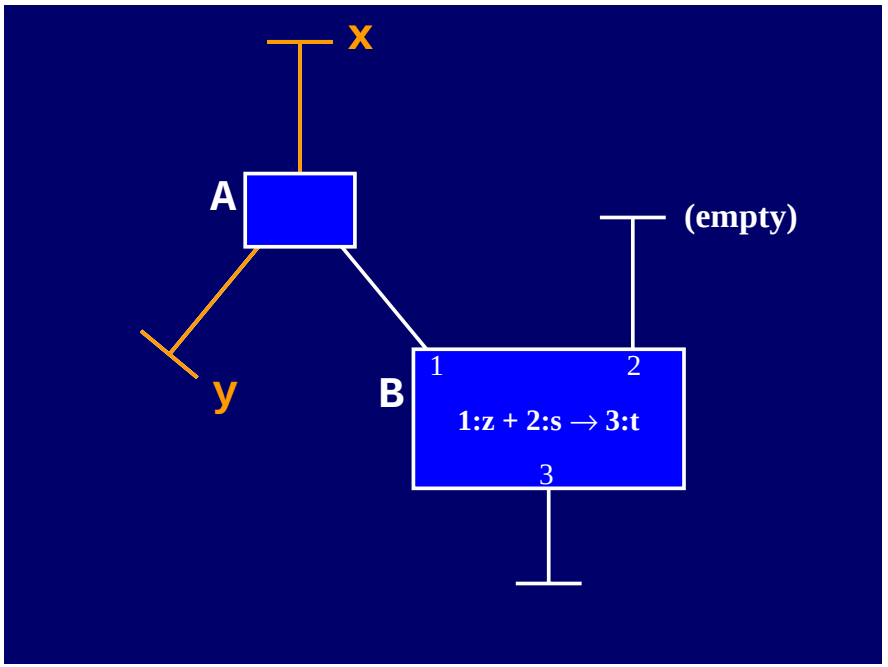

x
x
x
A
y
z
y
z
y
(empty)
B
1
2
1:z + 2:s → 3:t
3

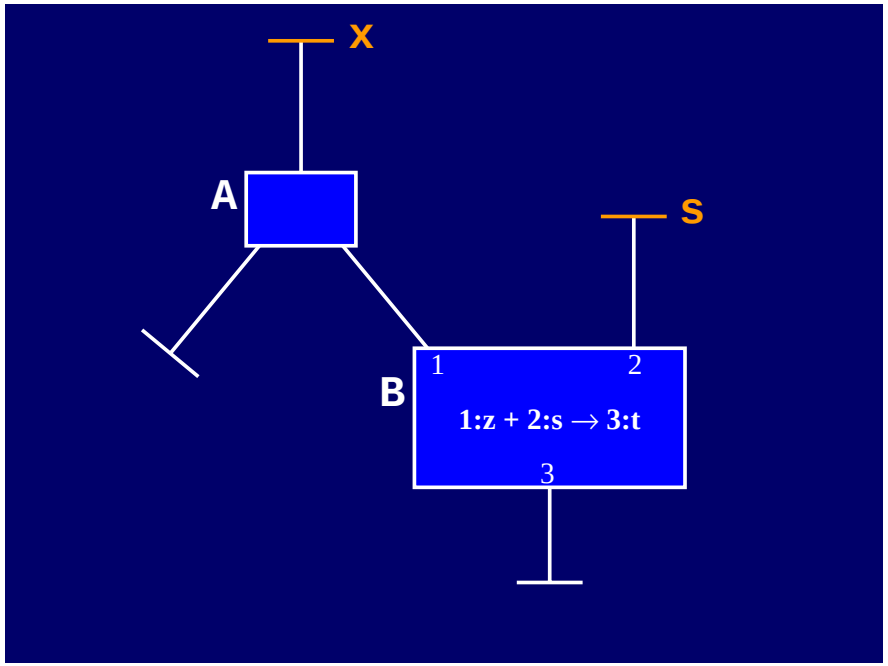

x
x
x
A
y
z
y
z
s
s
s
B
1
2
1:z + 2:s → 3:t
3

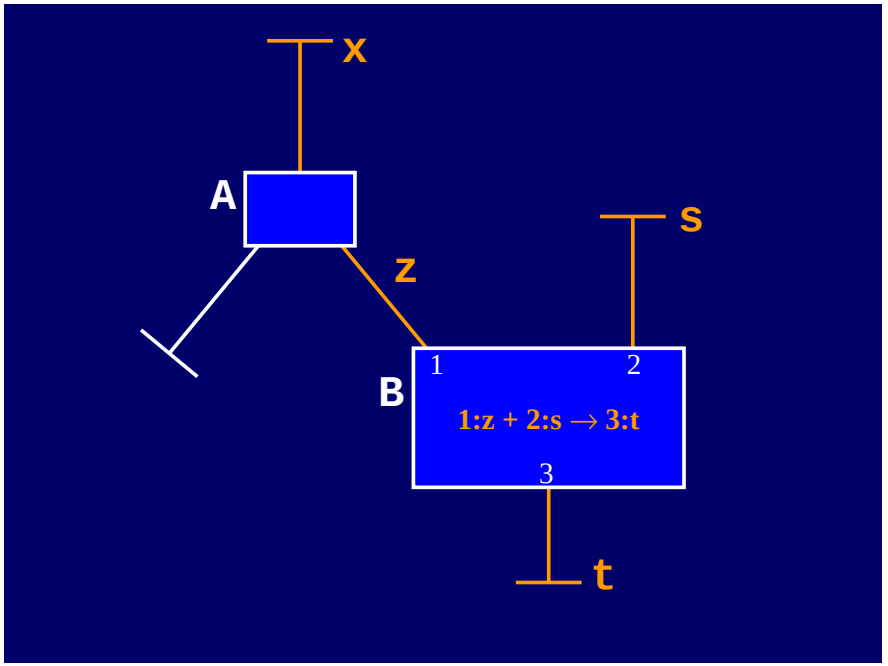

A
y
z
s
B
1
2
1:z + 2:s → 3:t
3

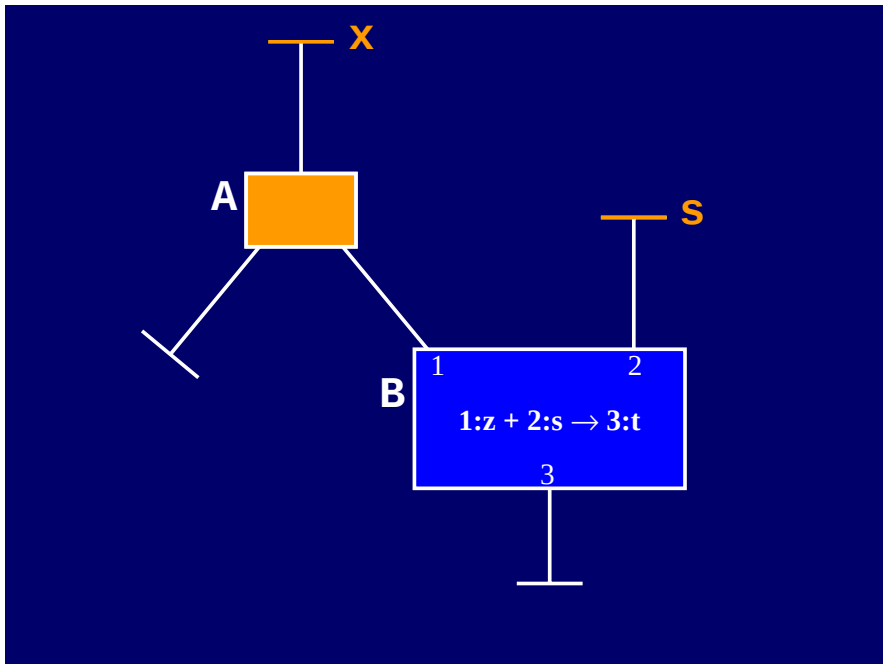

A
y
y
s
B
1
2
1:z + 2:s → 3:t
3
t
t


x
A
s
B
1
2
1:z + 2:s → 3:t
3
t


x
A
(empty)
y
B
1
2
1:z + 2:s → 3:t
3







Wavenet = Synchronous circuit

+ Superposition

+ Wave function collapse

+ Conservation of energy

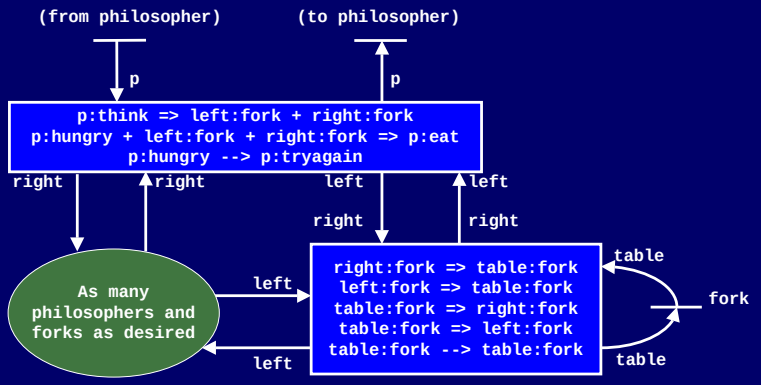
Wavenet =

A simple primitive that provides:

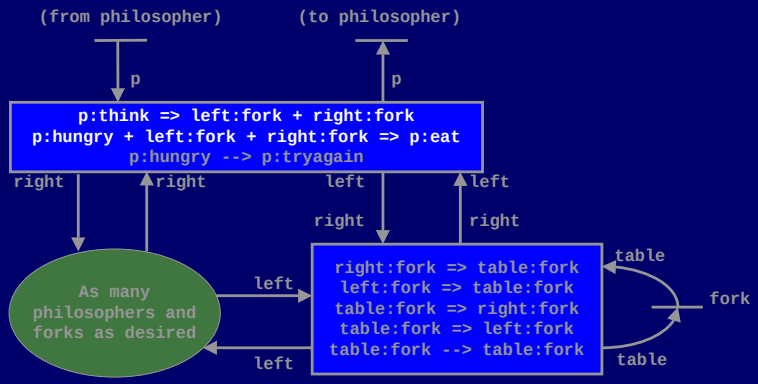
Global concurrency control

Automatic resource management

Dining philosophers solution



Dining philosophers solution



The challenge

Computing the latching is NP-complete

Conservation of energy =

- A powerful primitive that solves many different problems

Efficient implementation?

- Offline analysis
- Hardware support

Why?

Batch programs	Parallel servers
Input	Run forever
Processing	Maintain invariants
Output	Respond to multiple
Halt	inputs in parallel
Turing machines	Wavenets
Lambda calculus	Calculus?
Procedures	Prog. abstraction?

Take off the Turing blindfold!

`http://sdg.lcs.mit.edu/wavenets`