Lecture 4

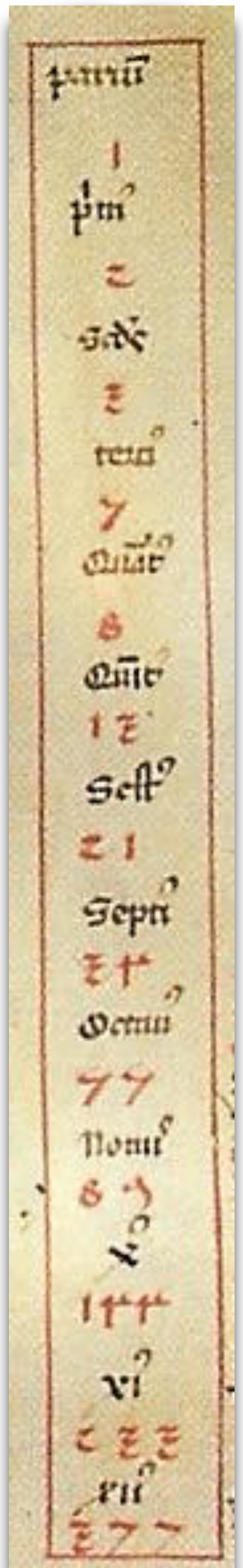
Signal Flow Graphs and recurrence relations

Plan

- **Fibonacci's rabbits and sustainable rabbit farming**
- Signal Flow Graphs
- Generating functions
- $\mathrm{IH}_{\mathbb{Q}[x]}$
- Operational semantics
- Solving sustainable rabbit farming

Fibonacci

(~1170 - ~1250)



1
2
3
5
8
13
21
34
55
89
144
233
377

A certain man had one pair of rabbits together in a certain enclosed place, and one wishes to know how many are created from the pair in one year when it is the nature of them in a single month to bear another pair, and in the second month those born to bear also.

You can indeed see in the margin how we operated, namely that we added the first number to the second, namely the 1 to the 2, and the second to the third, and the third to the fourth, and the fourth to the fifth, and thus one after another until we added the tenth to the eleventh, namely the 144 to the 233, and we had the above written sum of rabbits, namely 377, and thus you can in order find it for an unending number of months.

(extract from Liber Abaci, chapter 12,
translated from Latin by Lawrence Sigler)

The Fibonacci sequence

1, 2, 3, 5, 8…

in modern presentations often given as

1, 1, 2, 3, 5, … or 0, 1, 1, 2, 3, …

is an example of a **recurrence relation**.

All three satisfy

$$F_{n+2} = F_{n+1} + F_n$$

Coding Fibonacci

Natural to generalise Fibonacci's rabbit breeding

```
let ffib =
    let rec ffibaux r1 r2 xs =
        match xs with
            | [] -> []
            | xh :: xt ->
                xh + r2 + r1 ::
                    ffibaux (xh + r2) (xh + r2 + r1) xt
    in ffibaux 0 0;;
```

```
 val ffib : int list -> int list = <fun>
```

```
# ffib [1;0;0;0;0;0;0;0];;
- : int list = [1; 2; 3; 5; 8; 13; 21; 34]
```

```
# ffib [1;1;1;1;1;1;1;1];;
- : int list = [1; 3; 6; 11; 19; 32; 53; 87]
```

```
# ffib [1;1;-3;1;-2;-4;1];;
- : int list = [1; 3; 2; 3; 4; 1; 2]
```

Sustainable rabbit farming problem

- Suppose we want a sustainable rabbit farm, keeping four pairs of rabbits at all times
  - is it possible?
  - if so, how many pairs of rabbits must we add/remove and in which months?
- More generally, can we obtain a solution for any (possibly variable) number of rabbits in each month?

Achieving sustainable rabbit breeding

```
ffib: int list -> int list
```

To obtain solution, one could try to compute the **inverse**

```
bfib: int list -> int list
```

```
bfib [4;4;4;4;4;4;4;4;4;4;4;4];;
```

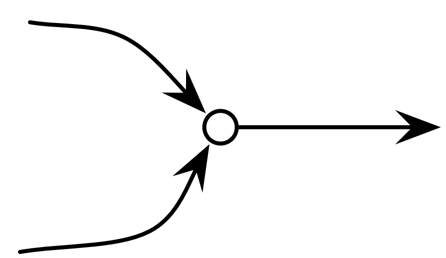
Plan

- Fibonacci's rabbits and sustainable rabbit farming
- **Signal Flow Graphs**
- Operational semantics
- Generating functions
- $IH_{Q[x]}$
- Solving sustainable rabbit farming

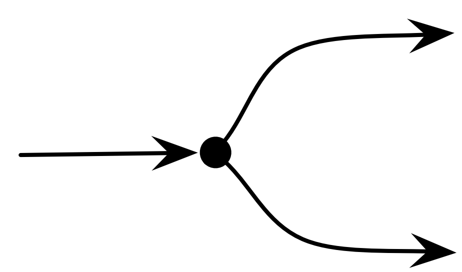
Signal Flow Graphs

(C. Shannon, 1942)

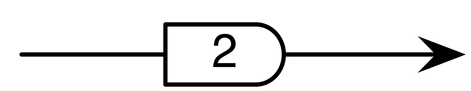
- Directed circuits wired by connecting
  - Adder gates



  - Copy Gates

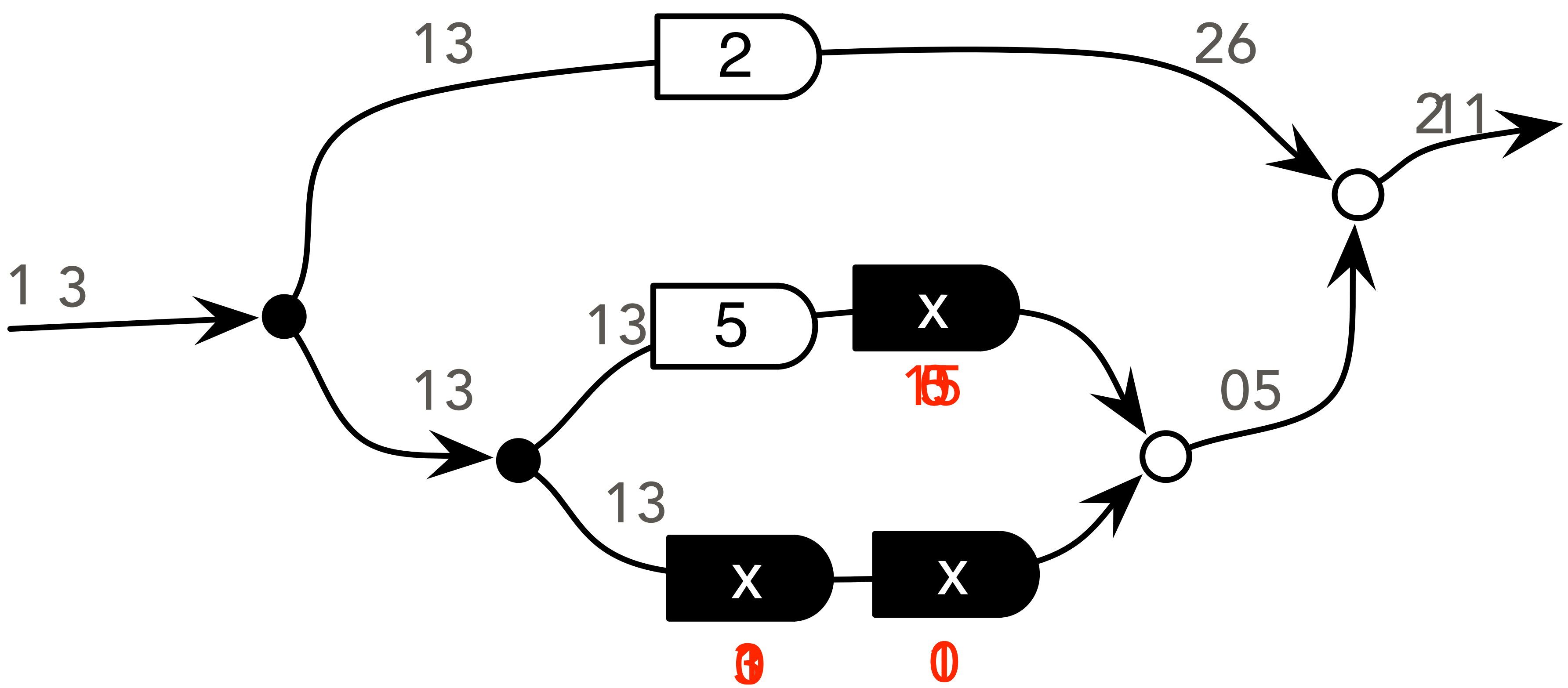


  - Amplifier gates



  - Register gates

Example execution

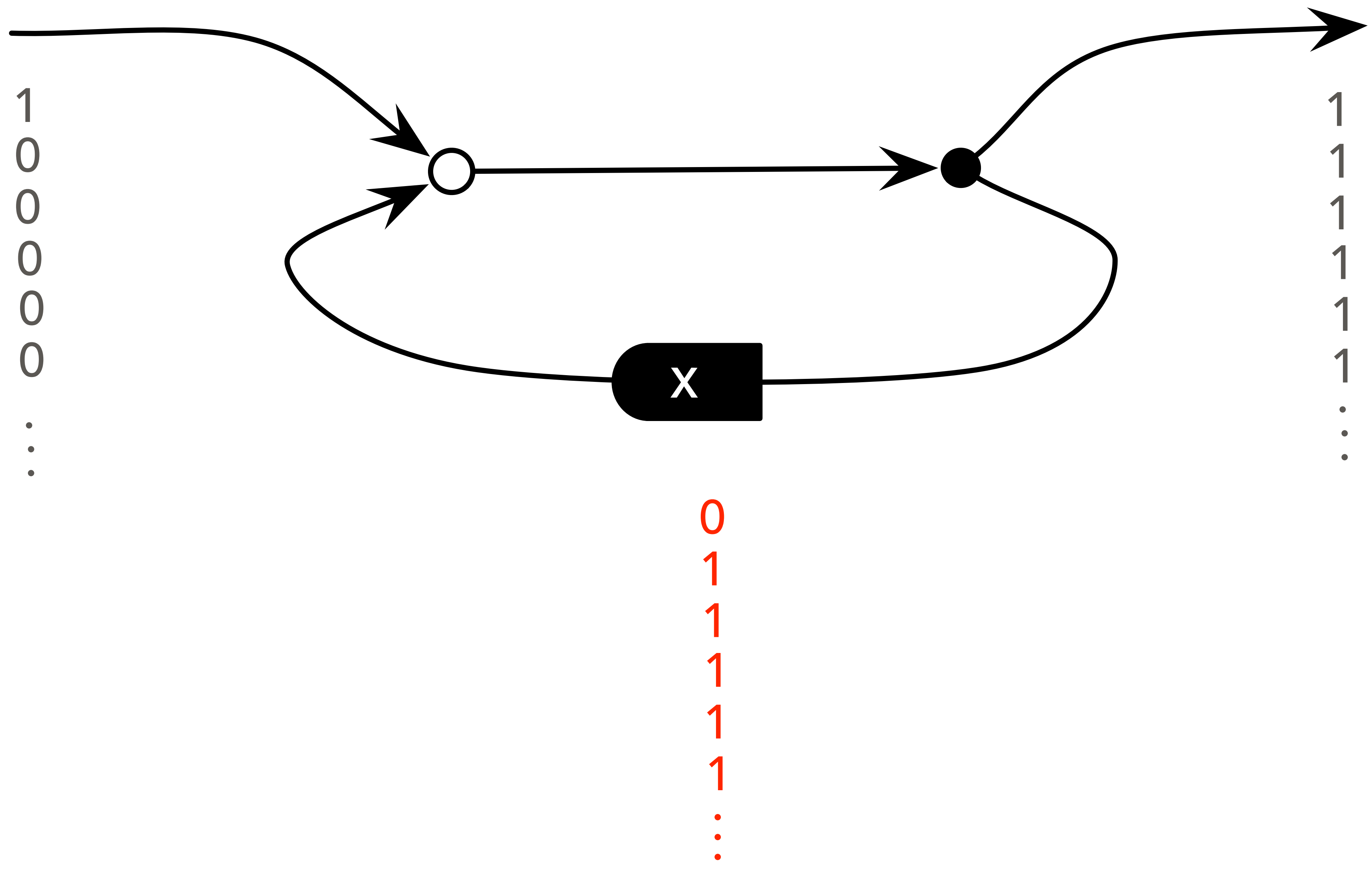


Input	Output
1	2
3	11

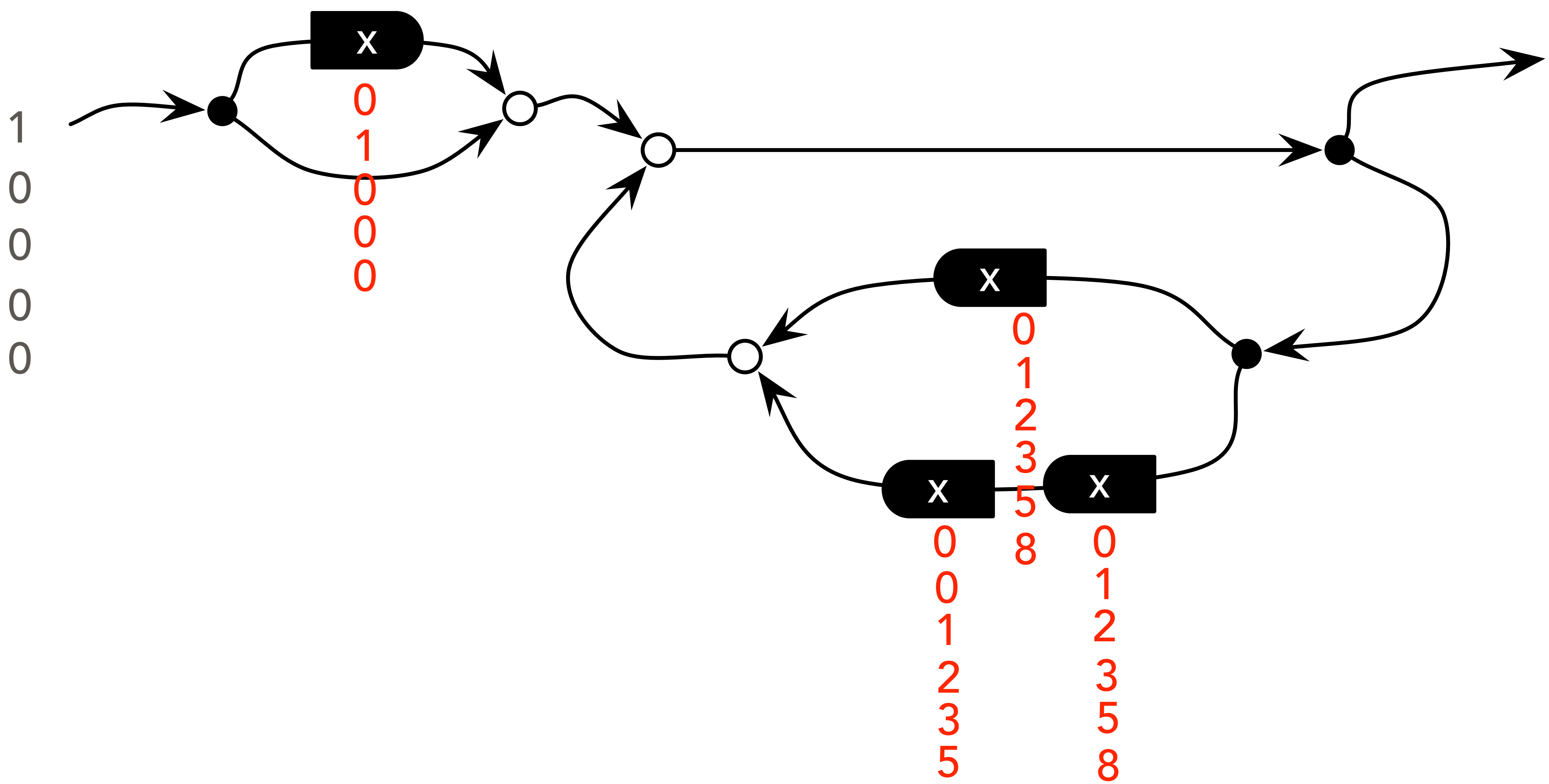
The circuit defines a function of type `int list -> int list`

note: if I keep pumping in zeros, then eventually all registers will get zeroed out and the output will stabilise at zero —- is this the case for every circuit?

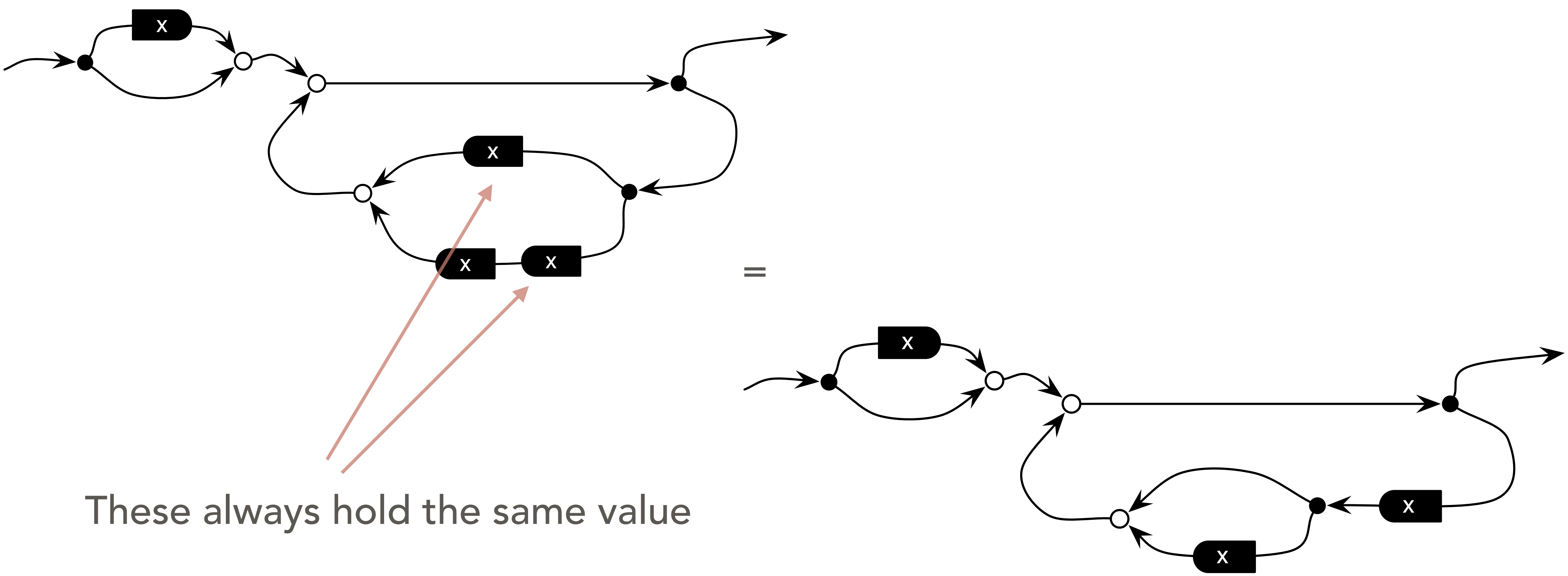
Feedback!



Example - Fibonacci

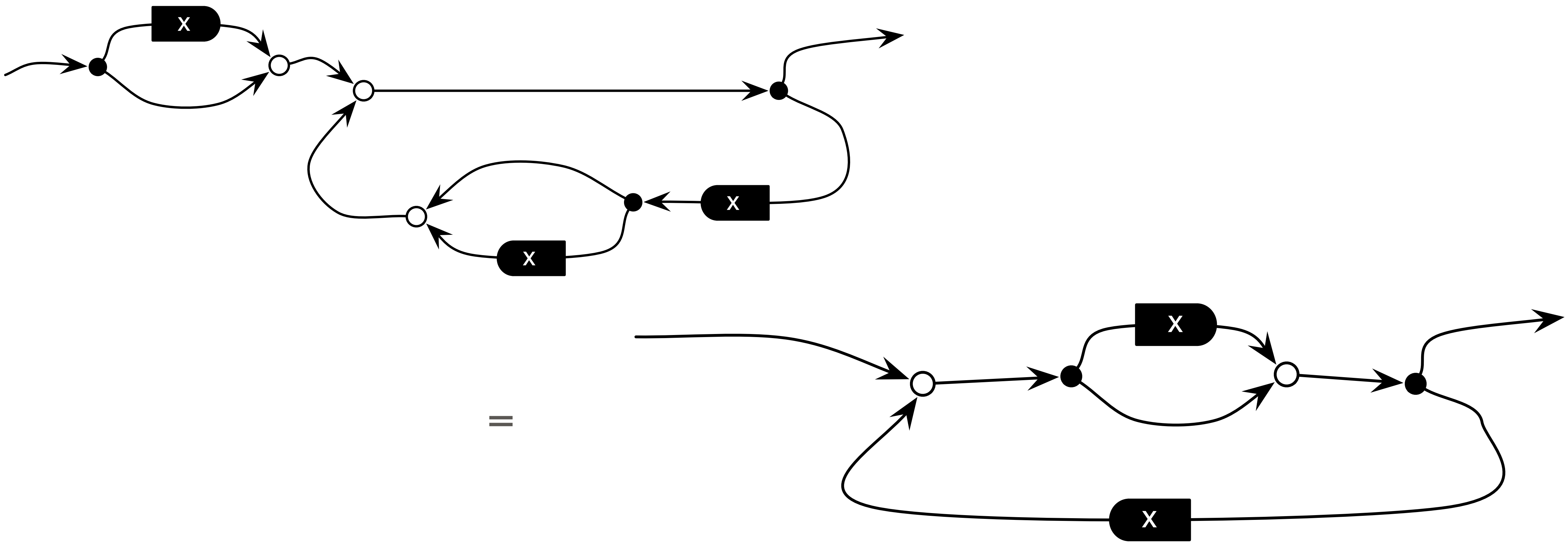


A little optimisation 1



ALGEBRAIC MANIPULATION, DIRECTLY ON THE CIRCUIT DIAGRAM!

A little optimisation 2



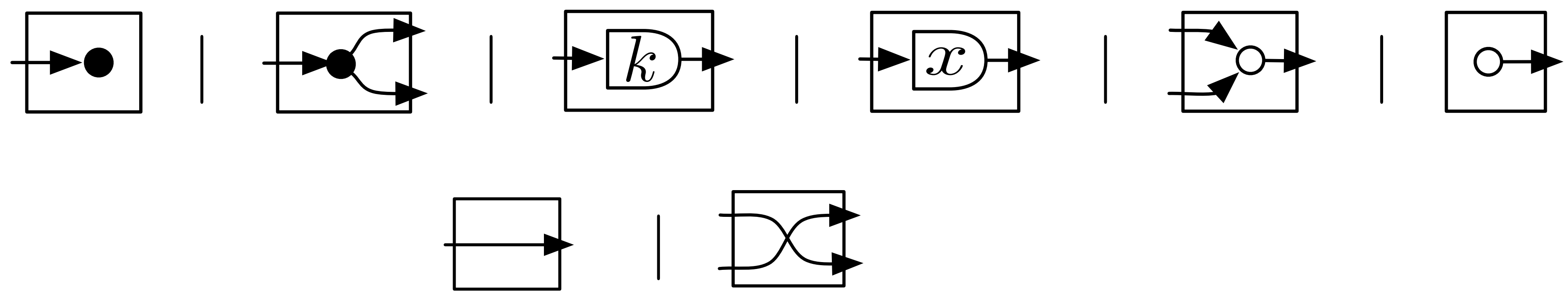

```
let ffib =
    let rec ffibaux r1 r2 xs =
        match xs with
            | [] -> []
            | xh :: xt ->
                xh + r2 + r1 ::
                    ffibaux (xh + r2) (xh + r2 + r1) xt
    in ffibaux 0 0;;
```

Plan

- Fibonacci's rabbits and sustainable rabbit farming
- Signal Flow Graphs
- **Operational semantics**
- Generating functions
- $IH_{Q[x]}$
- Solving sustainable rabbit farming

A directed calculus

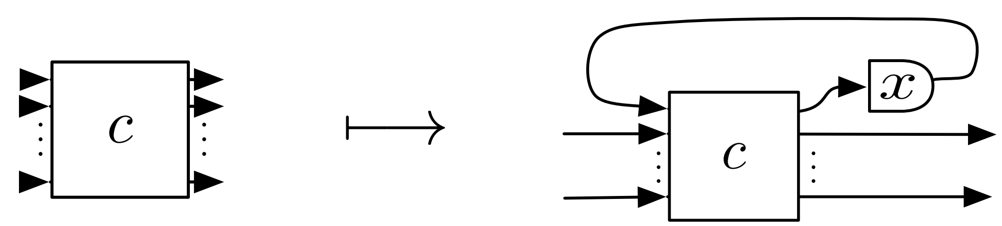
$e ::=$



T ::= T ; T | T⊕T | Tr(T)

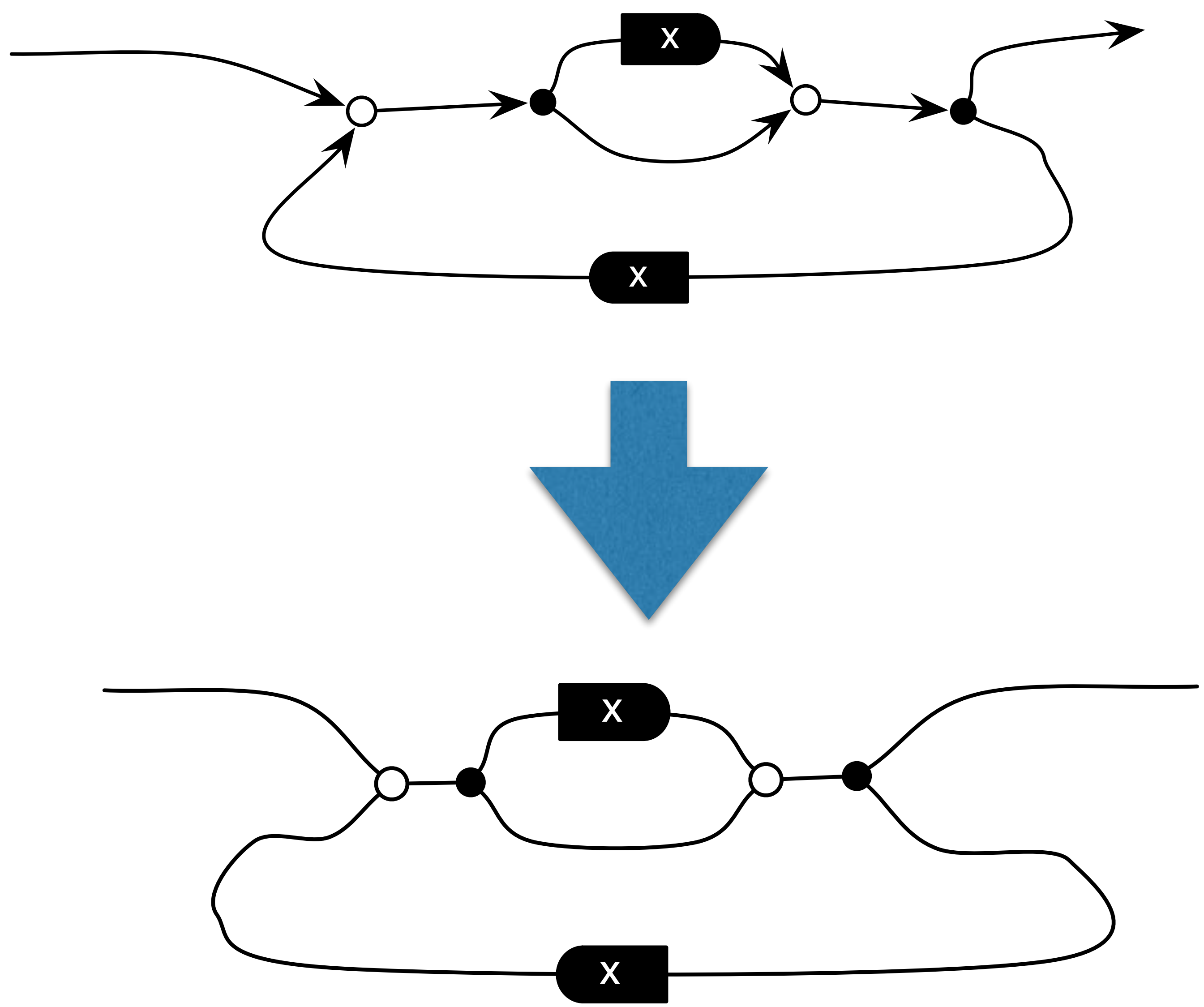
Trace operation

$$\frac{T : (m+1,\ n+1)}{Tr(T) : (m,\ n)}$$

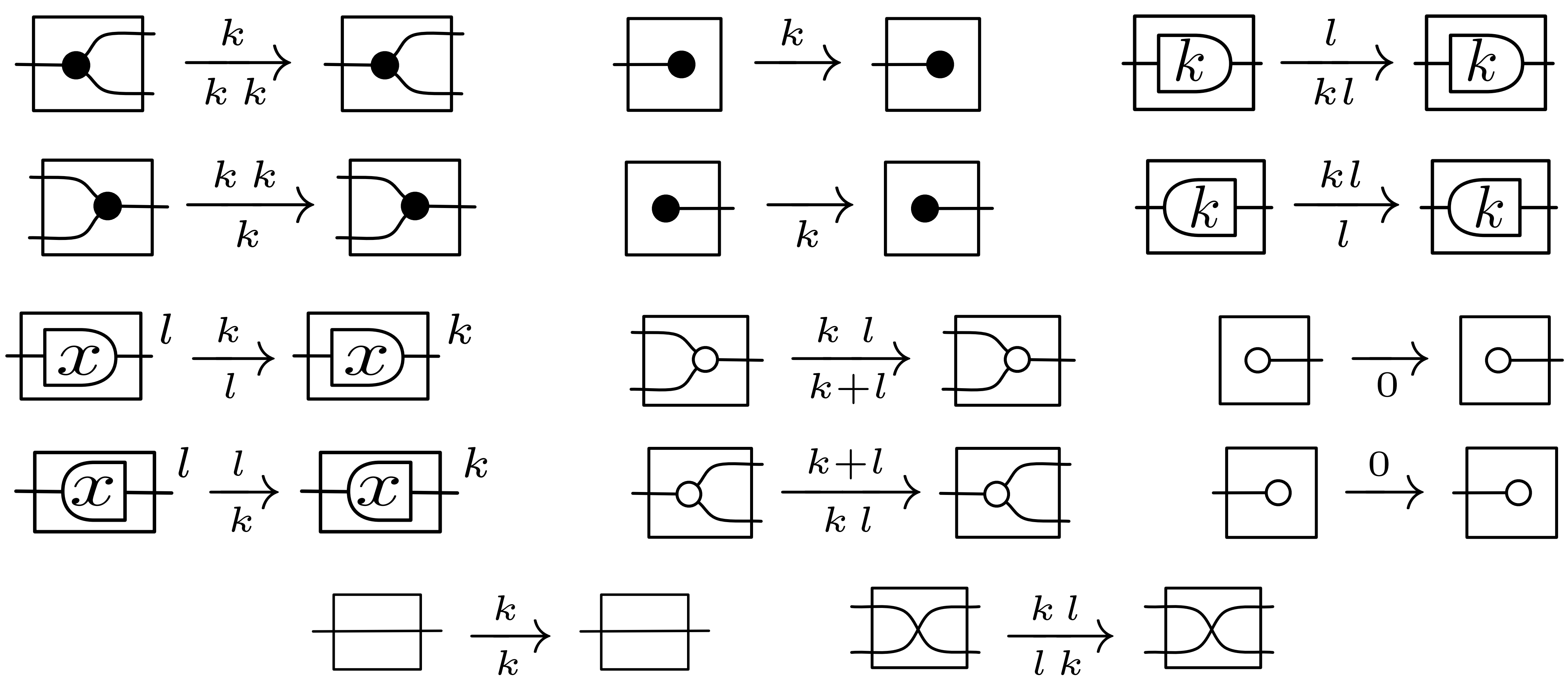


Signal flow graphs as string diagrams

- Easy - get rid of directions on the wires!



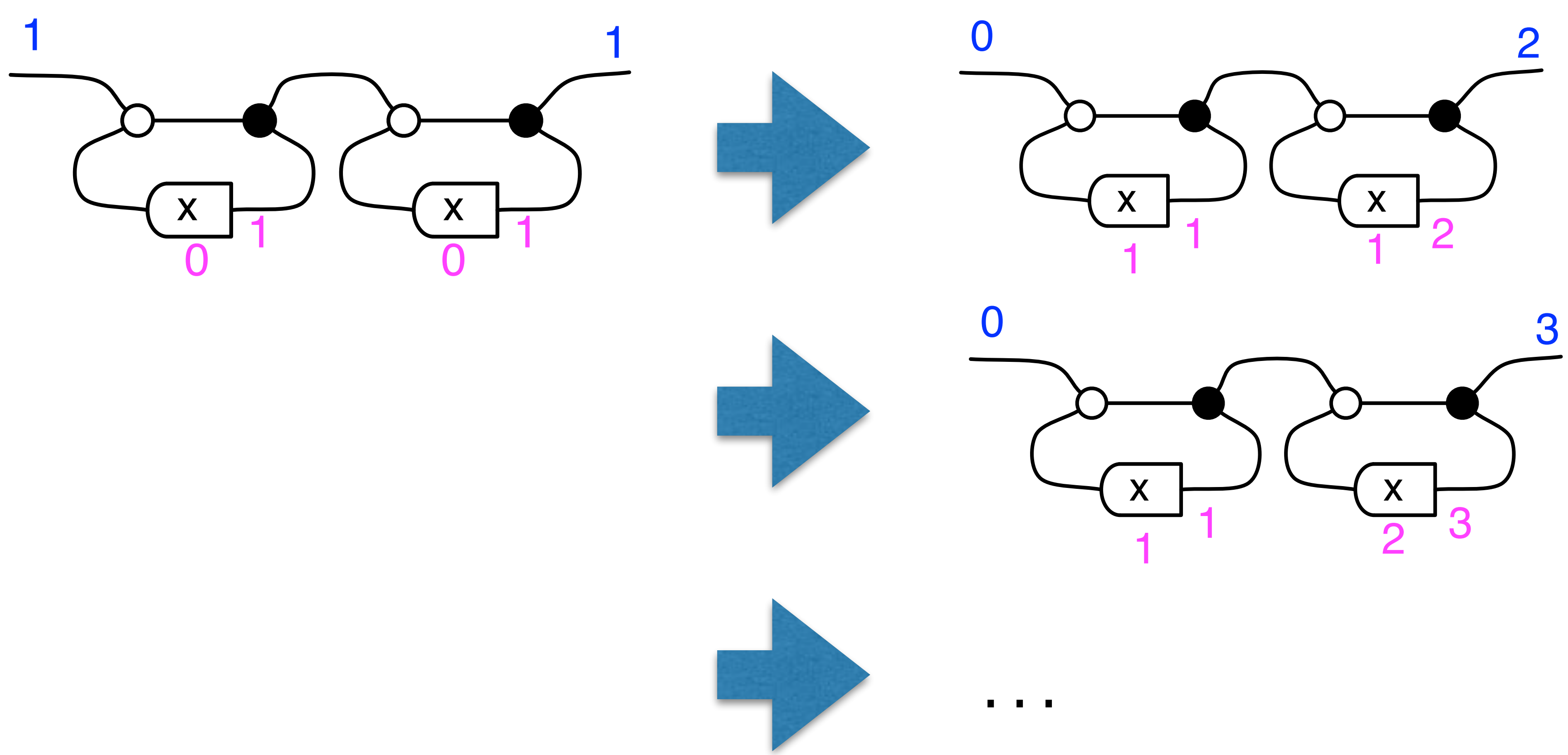
Forgetting directions



$$\frac{s \xrightarrow[v]{u} s' \qquad t \xrightarrow[w]{v} t'}{s\,;t \xrightarrow[w]{u} s'\,;t'}$$

$$\frac{s \xrightarrow[v_1]{u_1} s' \qquad t \xrightarrow[v_2]{u_2} t'}{s \oplus t \xrightarrow[v_1\,v_2]{u_1\,u_2} s' \oplus t'}$$

Example



Plan

- Fibonacci's rabbits and sustainable rabbit farming
- Signal Flow Graphs
- Operational semantics
- **Generating functions**
- $\mathbb{IH}_{\mathbb{Q}[x]}$
- Solving sustainable rabbit farming

Polynomial Zoo

- Ring of polynomials $\mathbf{Q}[x]$
  - → Field of polynomial fractions $\mathbf{Q}(x)$
- Ring of formal power series $\mathbf{Q}[[x]]$
  - → Field of Laurent power series $\mathbf{Q}((x))$
- injective ring hom. $\mathbf{Q}[x] \to \mathbf{Q}[[x]]$
  - → (injective) field hom. $\mathbf{Q}(x) \to \mathbf{Q}((x))$

Generatingfunctionology

(see famous book by H. Wilf)

Spec $F_0 = 1$ $F_1 = 2$ $F_{n+2} = F_{n+1} + F_n$

Define $$F(x) = \sum_{n=0}^{\infty} F_n x^n$$

$$\sum_{n=0}^{\infty} F_{n+2}x^n = \sum_{n=0}^{\infty} F_{n+1}x^n + \sum_{n=0}^{\infty} F_n x^n \quad \Rightarrow \quad F(x) = \sum_{n=0}^{\infty} F_{n+2}x^n - \sum_{n=0}^{\infty} F_{n+1}x^n$$

$$x^2\left(\sum_{n=0}^{\infty} F_{n+2}x^n\right) = \sum_{n=2}^{\infty} F_n x^n = F(x) - 2x - 1$$

$$x\left(\sum_{n=0}^{\infty} F_{n+1}x^n\right) = \sum_{n=1}^{\infty} F_n x^n = F(x) - 1$$

$$F(x) = \frac{F(x) - 2x - 1}{x^2} - \frac{F(x) - 1}{x}$$

$$\boxed{F(x) = \frac{1 + x}{1 - x - x^2}}$$

Obtaining the coefficients

k(x) → k((x))

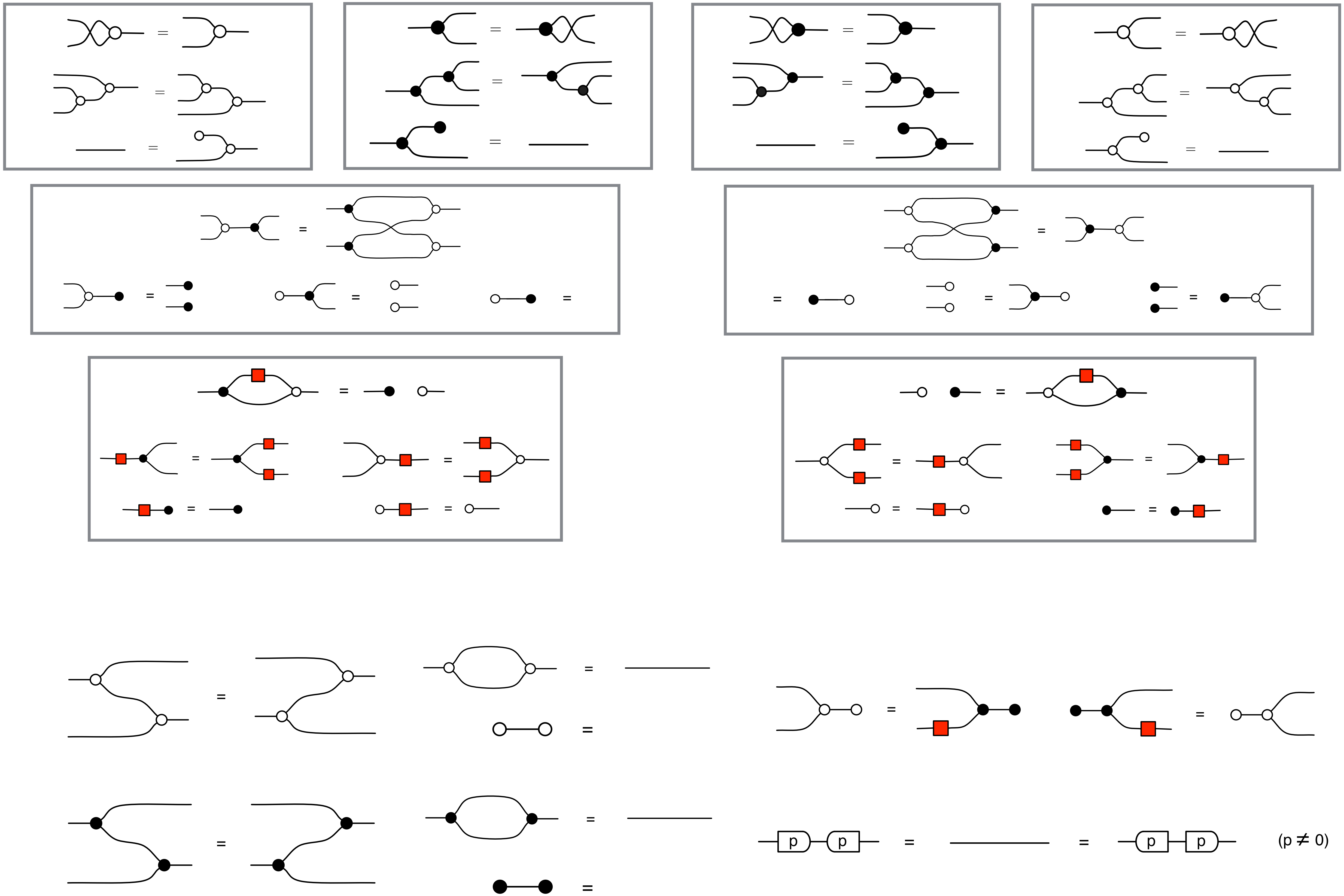
$$F(x) = \frac{1+x}{1-x-x^2} \quad \mapsto \quad 1 + 2x + 3x^2 + 5x^3 + 8x^4 + \ldots$$

(1, 2, 3, 5, 8, …)

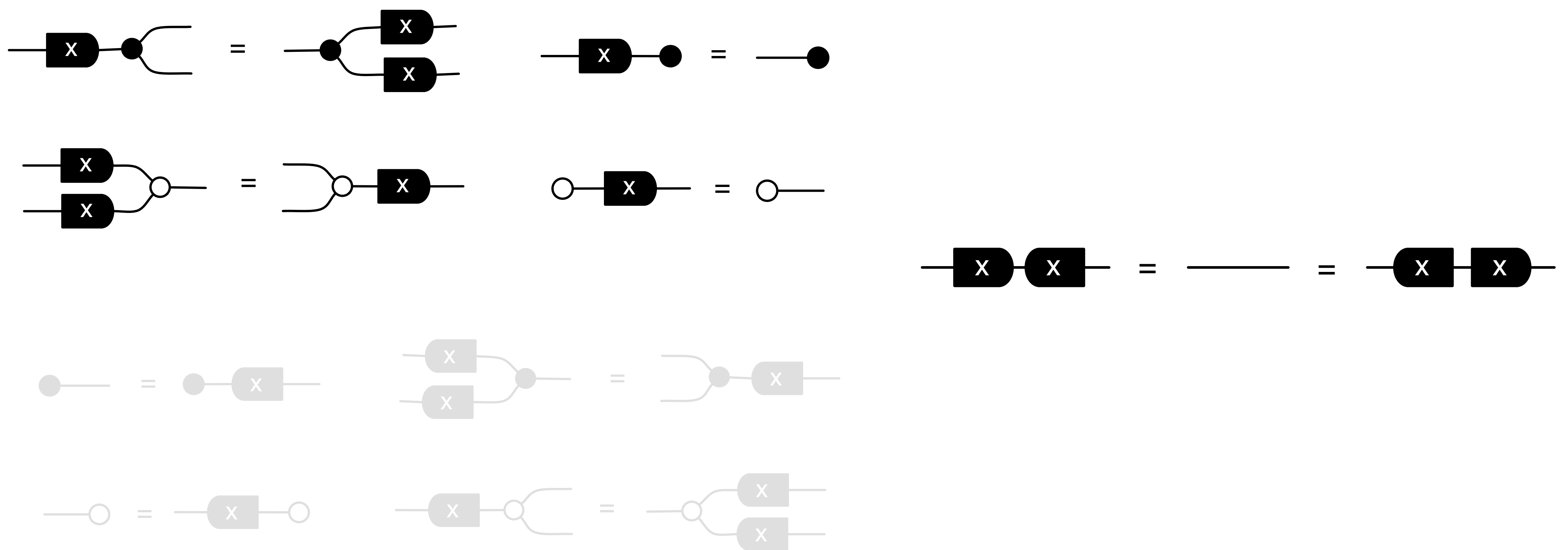
Moral of the story : polynomial fractions are useful for reasoning about recurrence relations

Plan

- Fibonacci's rabbits and sustainable rabbit farming
- Signal Flow Graphs
- Generating functions
- $\mathbf{IH_{Q[x]}}$
- Solving sustainable rabbit farming

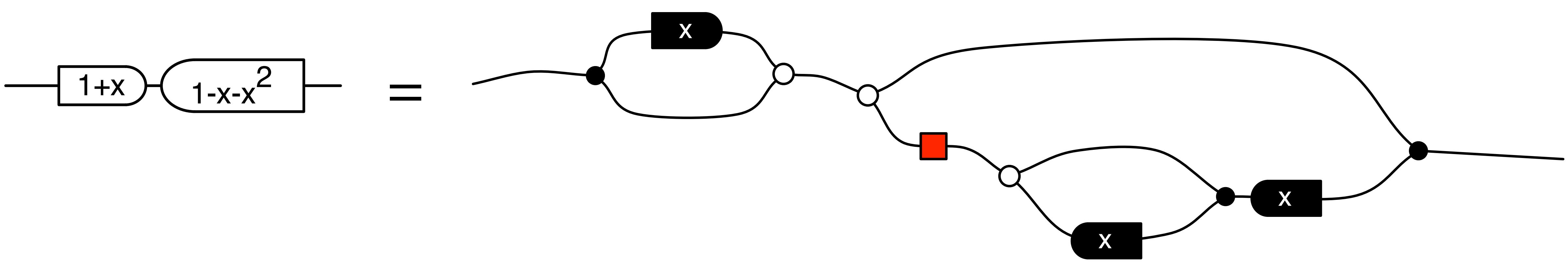

p
p
p
p
(p ≠ 0)

From linear relations over **Q** to linear relations over **Q**(x)

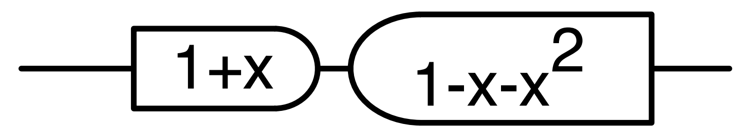


Polynomial fractions with diagrammatic syntax

$$F(x) = \frac{1+x}{1-x-x^2}$$



Example



As linear relation over **Q**(x) is the space generated by

$$(1,\ (1+x)/(1-x-x^2))$$

As linear relation over **Q**((x)) is the space generated by

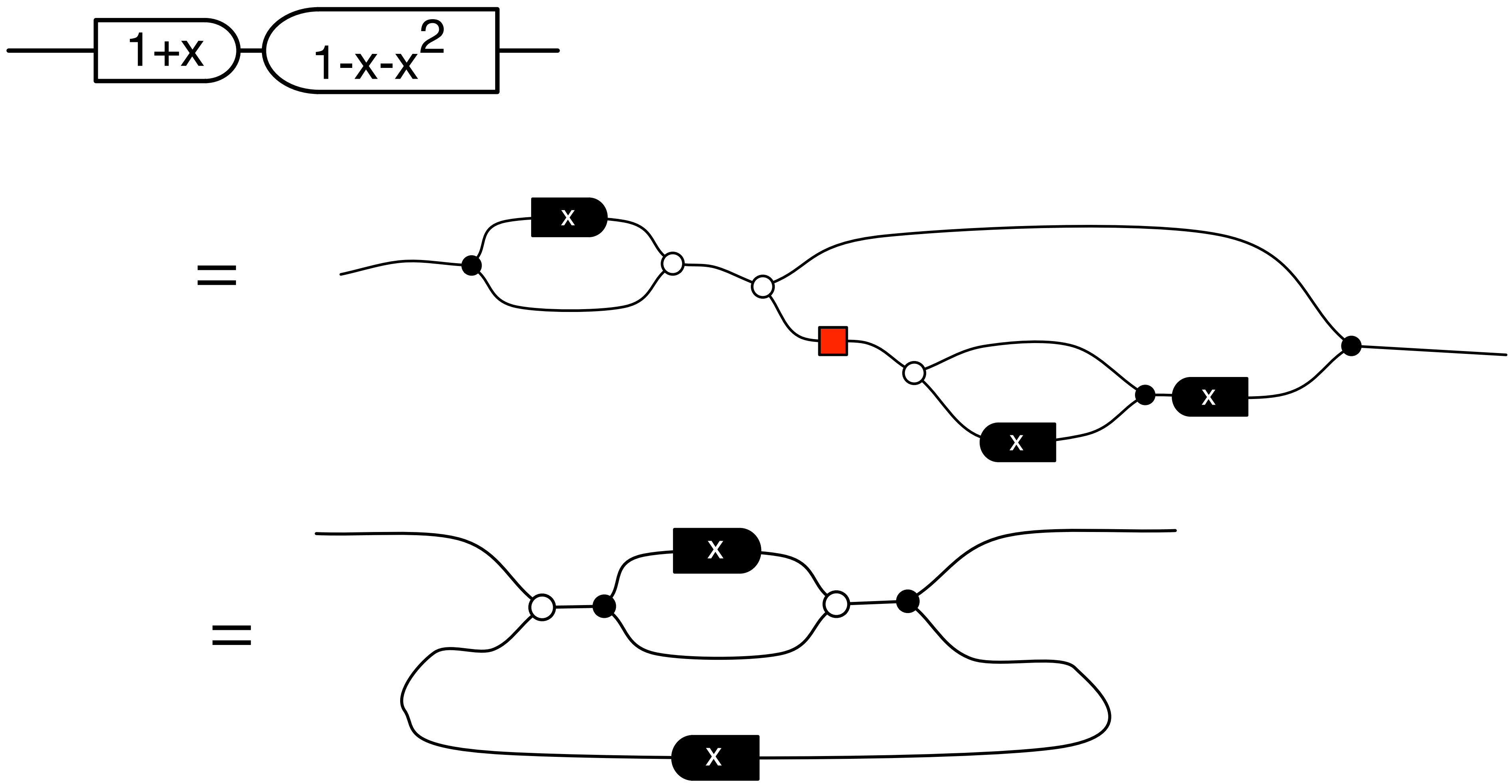
$$(\underline{1},0,0,\ldots \quad , \quad \underline{1},2,3,5,8,\ldots)$$

Realisability and Full Abstraction

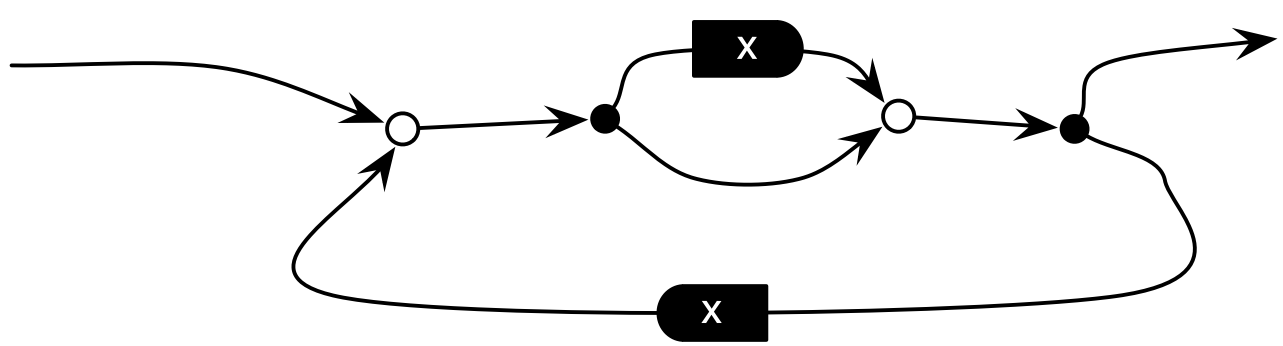
- **Realisability** Every diagram can be put in a form where the direction of signal flow is consistent
- **Full abstraction** *Operational equality* (in terms of behaviour, given by operational semantics) coincides with *denotational equality* (the denoted linear relation) on diagrams with consistent signal flow

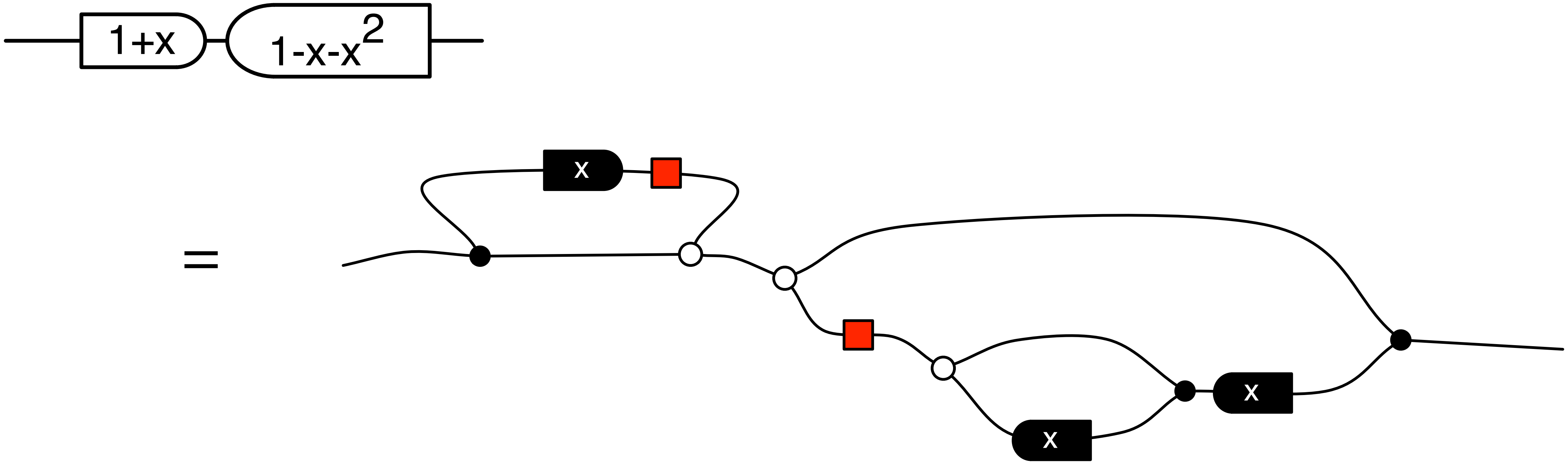
Plan

- Fibonacci's rabbits and sustainable rabbit farming
- Signal Flow Graphs
- Generating functions
- $IH_{Q[x]}$
- Operational semantics
- **Solving sustainable rabbit farming**

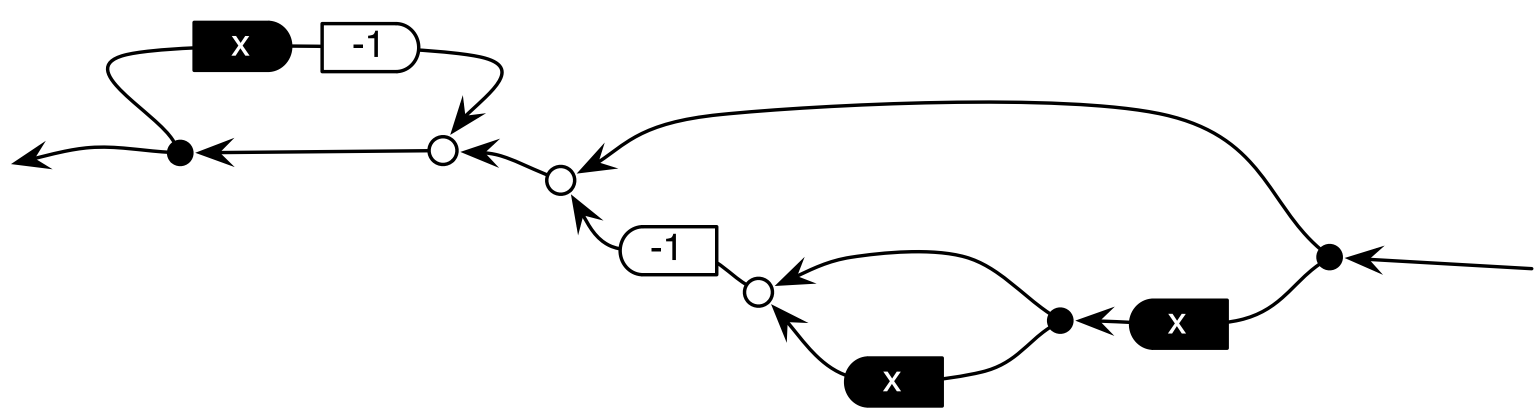


Which we can be directed from left to right





Which can be directed from right to left



Solving sustainable rabbit farming

```
# rfib [4;4;4;4;4;4;4;4;4;4;4;4;4];;
- : int list = [4; -4; 0; -4; 0; -4; 0; -4; 0; -4; 0; -4; 0]
```

i.e. buy four pairs of rabbits in the first month,
then sell four every two months…

Bibliography

- Bonchi, S., Zanasi - Interacting Bialgebras are Frobenius, FoSSaCS ’14
- Bonchi, S., Zanasi - Interacting Hopf Algebras, J Pure Applied Algebra 221:144–184, 2017
- Bonchi, S., Zanasi - The Calculus of Signal Flow Diagrams I: Linear Relations on Streams, Inf Comput 252:2–29, 2017
- Bonchi, S., Zanasi - A categorical semantics of signal flow graphs, CONCUR 2013
- Bonchi, S., Zanasi - Full abstraction for signal flow graphs, PoPL 2016
- Zanasi - Interacting Hopf Algebras: The theory of linear systems, PhD Thesis, ENS Lyon, 2015
- Bonchi, S., Zanasi - Lawvere Theories as composed PROPs, CMCS 2016
- Fong, Rapisarda, S. - A categorical approach to open and interconnected dynamical systems, LiCS 2016
- Bonchi, Gadducci, Kissinger, S. - Rewriting modulo symmetric monoidal structure, LiCS 2016
- Bonchi, Gadducci, Kissinger, S. - Confluence of Graph Rewriting with Interfaces, ESOP 2017

graphicallinearalgebra.net