Freezing Cellular Automata

Eric Goles (Univ. Adolfo Ibañez, Santiago, Chile)
Nicolas Ollinger (LIFO, Univ. Orléans, France)
Guillaume Theyssier (CNRS, Univ. Savoie, France)

AUTOMATA 2015, Turku — June 10th, 2015

Definition

Definition A **freezing CA** (FCA) is a cellular automaton F compatible with a partial order $\geqslant$ on states:

$$F(c)(z) \geqslant c(z) \qquad \forall z \in \mathbb{Z}^d \quad \forall c \in Q^{\mathbb{Z}^d}$$

Definition

Definition A **freezing CA** (FCA) is a cellular automaton F compatible with a partial order $\geqslant$ on states:

$$F(c)(z) \geqslant c(z) \qquad \forall z \in \mathbb{Z}^d \quad \forall c \in Q^{\mathbb{Z}^d}$$

Remark Freezing CA are essentially CA with a uniformly **bounded number of state change** per cell.

Definition

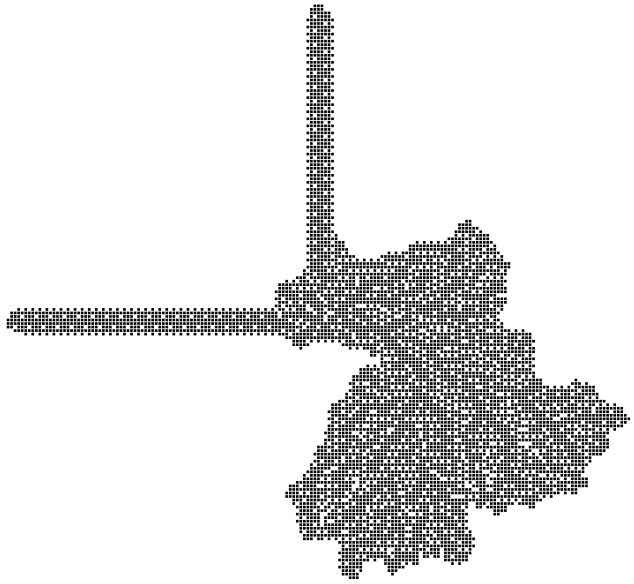
Definition A **freezing CA** (FCA) is a cellular automaton F compatible with a partial order $\geqslant$ on states:

$$F(c)(z) \geqslant c(z) \qquad \forall z \in \mathbb{Z}^d \quad \forall c \in Q^{\mathbb{Z}^d}$$

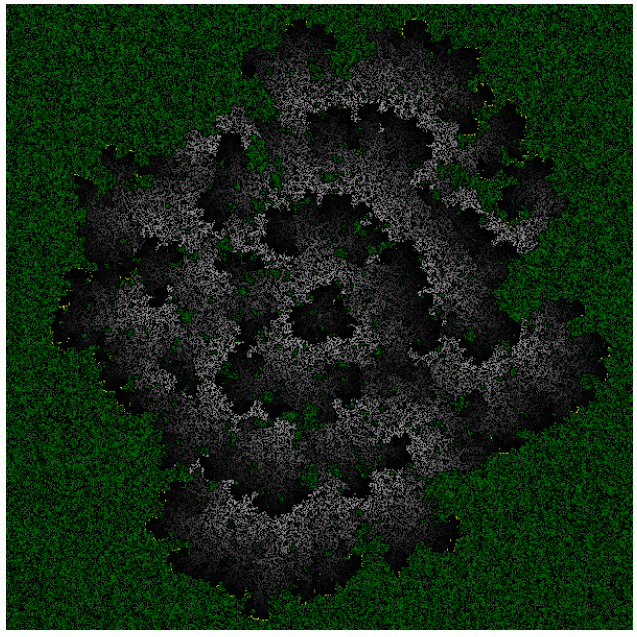
Remark Freezing CA are essentially CA with a uniformly **bounded number of state change** per cell.

We study **dynamics** and **complexity** of freezing CA.

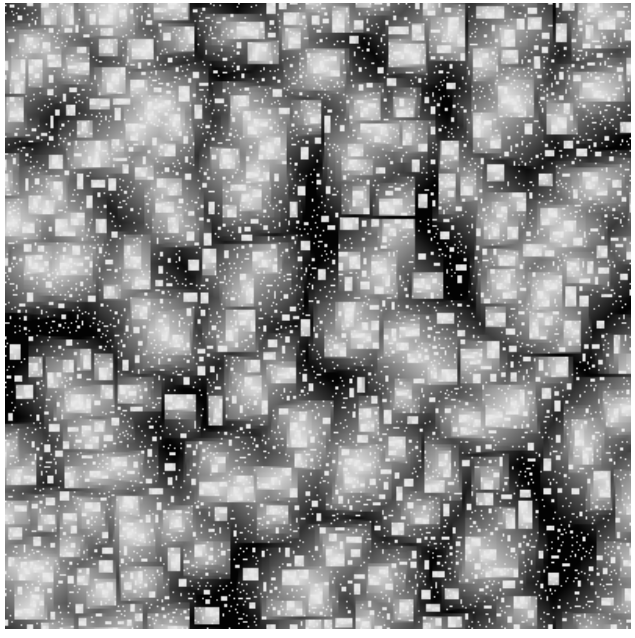
In this talk we focus on **deterministic updates**.



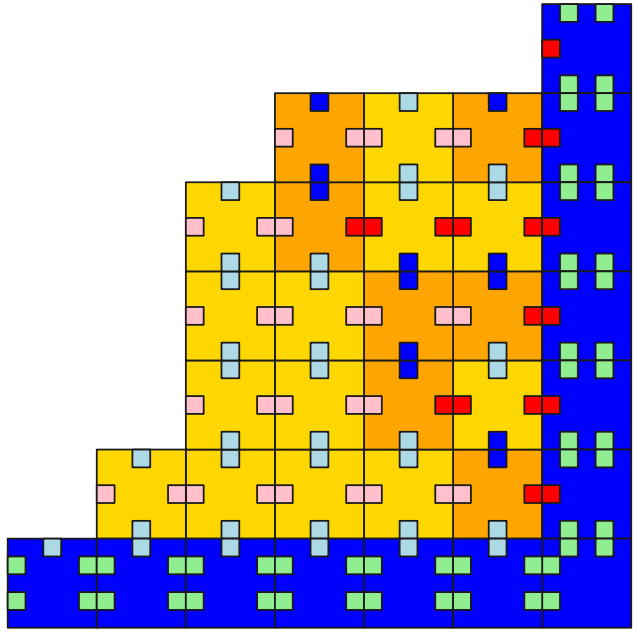
Life without death



Forest fire model



Bootstrap percolation



Self-assembly tiling

Basic dynamical properties

FCA are closed under iteration, Cartesian product, subautomaton, local factor and grouping **but not shift**.

Basic dynamical properties

FCA are closed under iteration, Cartesian product, subautomaton, local factor and grouping **but not shift**.

FCA do **freeze** any initial configuration: every configuration converges to a **fixpoint** in the Cantor metric.

Indeed, each cell changes its state a finite number of times ♦

Basic dynamical properties

FCA are closed under iteration, Cartesian product, subautomaton, local factor and grouping **but not shift**.

FCA do **freeze** any initial configuration: every configuration converges to a **fixpoint** in the Cantor metric.

Indeed, each cell changes its state a finite number of times ♦

The only **surjective** FCA is the **identity map**.

Let $f : Q^3 \to Q$ be a FCA over $\geqslant$ different from identity.
Let c be a minimal state such that $f(b, a, d) = c$ with $c > a$.
By minimality if $f(b', a', d') = a$ then $a' = a$.
Thus $|f^{-1}(a)| < |Q|^2$, the CA is unbalanced ♦

Nilpotency

When a FCA is **not nilpotent** then it has **two $\neq$ fixpoints**.

(1) Let F be a non nilpotent FCA with a maximal state a

First fixpoint: $F(\overline{a}) = \overline{a}$

Nilpotency

When a FCA is **not nilpotent** then it has **two $\neq$ fixpoints**.

(1) Let F be a non nilpotent FCA with a maximal state a

First fixpoint: $F(\overline{a}) = \overline{a}$

(2) Extract an infinite sequence of configurations (c_i) such that $F(c_{i+1}) = c_i$ and $c_i(0) < a$

Nilpotency

When a FCA is **not nilpotent** then it has **two $\neq$ fixpoints**.

(1) Let F be a non nilpotent FCA with a maximal state a

First fixpoint: $F(\overline{a}) = \overline{a}$

(2) Extract an infinite sequence of configurations (c_i) such that $F(c_{i+1}) = c_i$ and $c_i(0) < a$

(3) Each cell from (c_i) changes state a finite number of time: extract a fixpoint c with $c(0) < a$ ♦

Nilpotency

When a FCA is **not nilpotent** then it has **two $\neq$ fixpoints**.

(1) Let F be a non nilpotent FCA with a maximal state a

First fixpoint: $F(\overline{a}) = \overline{a}$

(2) Extract an infinite sequence of configurations (c_i) such that $F(c_{i+1}) = c_i$ and $c_i(0) < a$

(3) Each cell from (c_i) changes state a finite number of time: extract a fixpoint c with $c(0) < a$ ♦

Nilpotency for FCA is **decidable** in 1D and **undecidable** in higher dimensions.

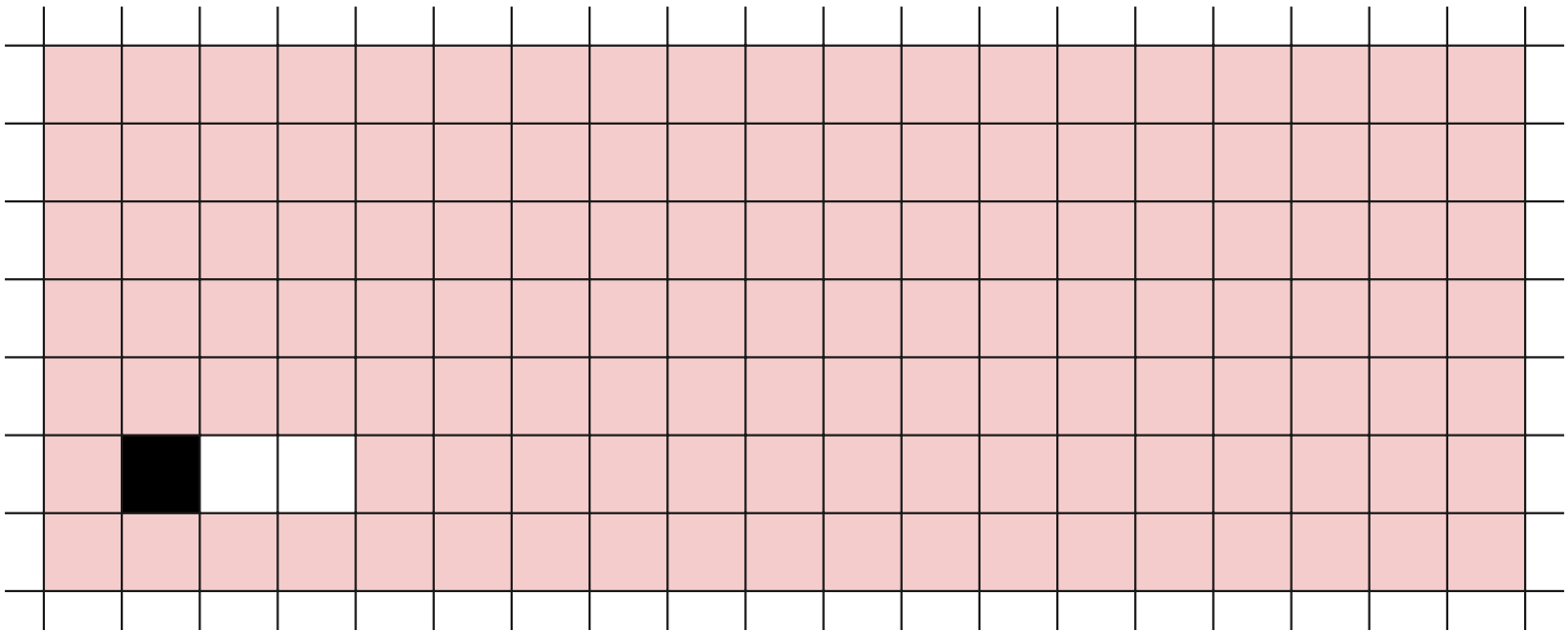
In 1D, detect two fixpoints. In higher dimension, the classical Domino Problem argument is freezing.

Computation complexity in 2D+

Prediction problem PRED$_F$ Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

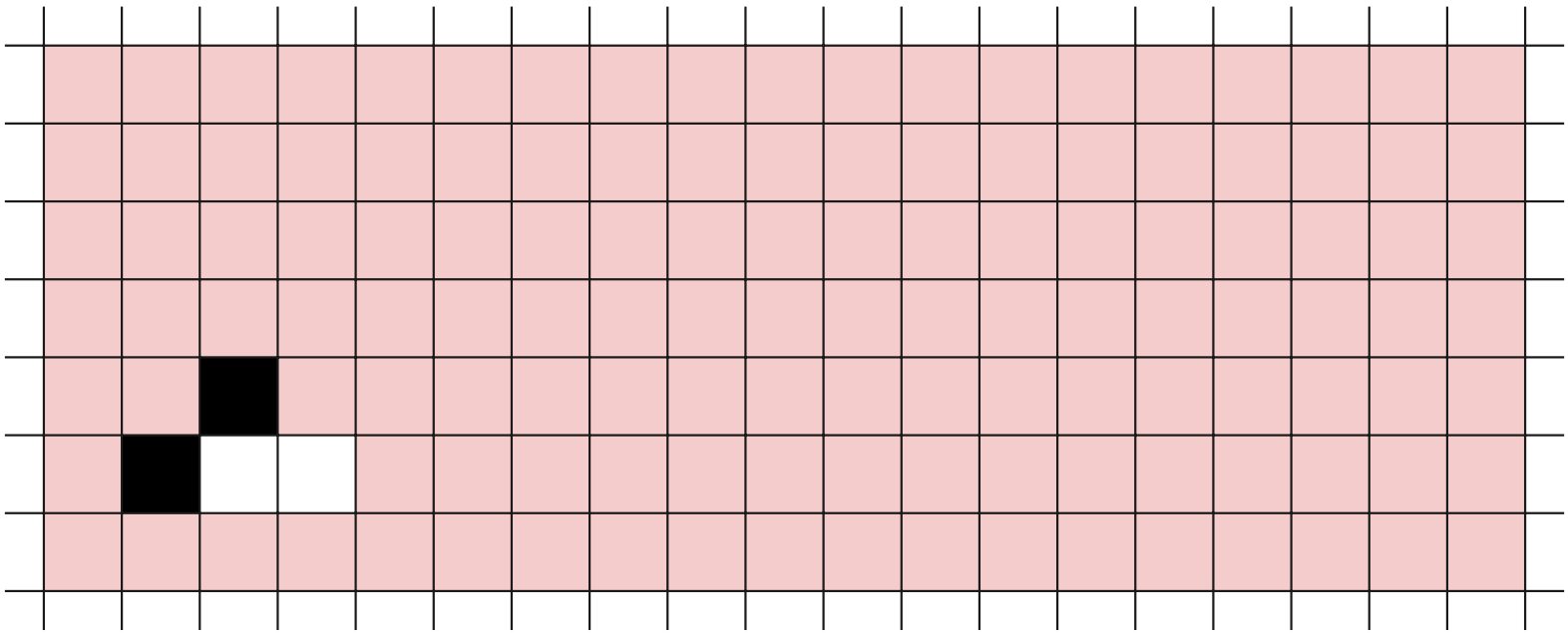


Computation complexity in 2D+

Prediction problem PRED$_F$ Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

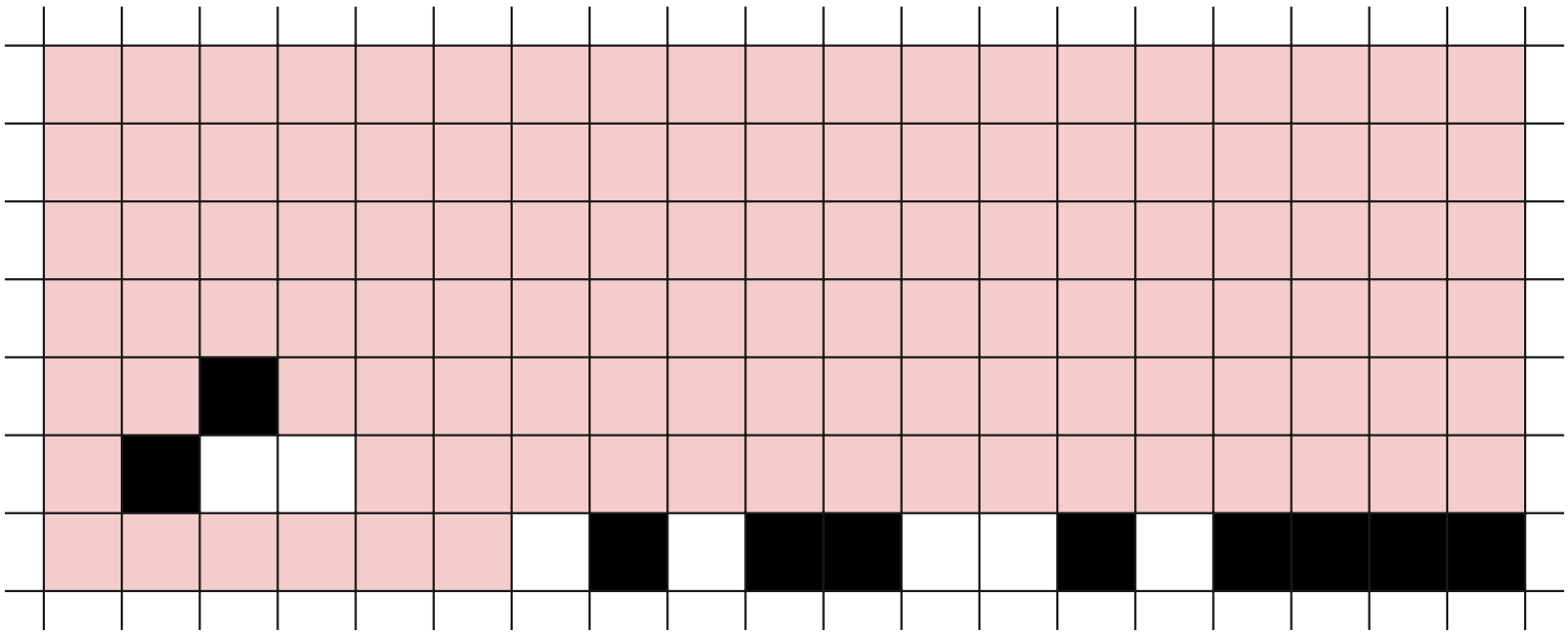


Computation complexity in 2D+

Prediction problem PRED_F Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

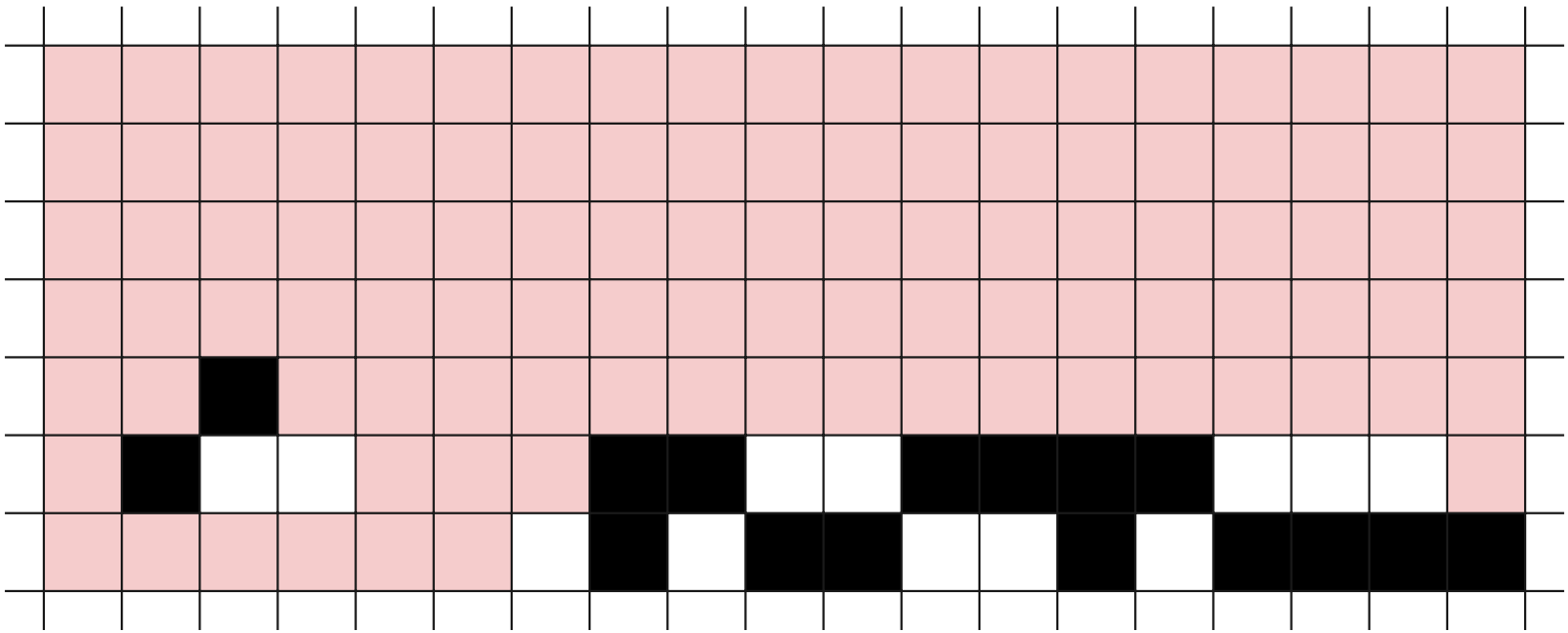


Computation complexity in 2D+

Prediction problem PRED_F Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

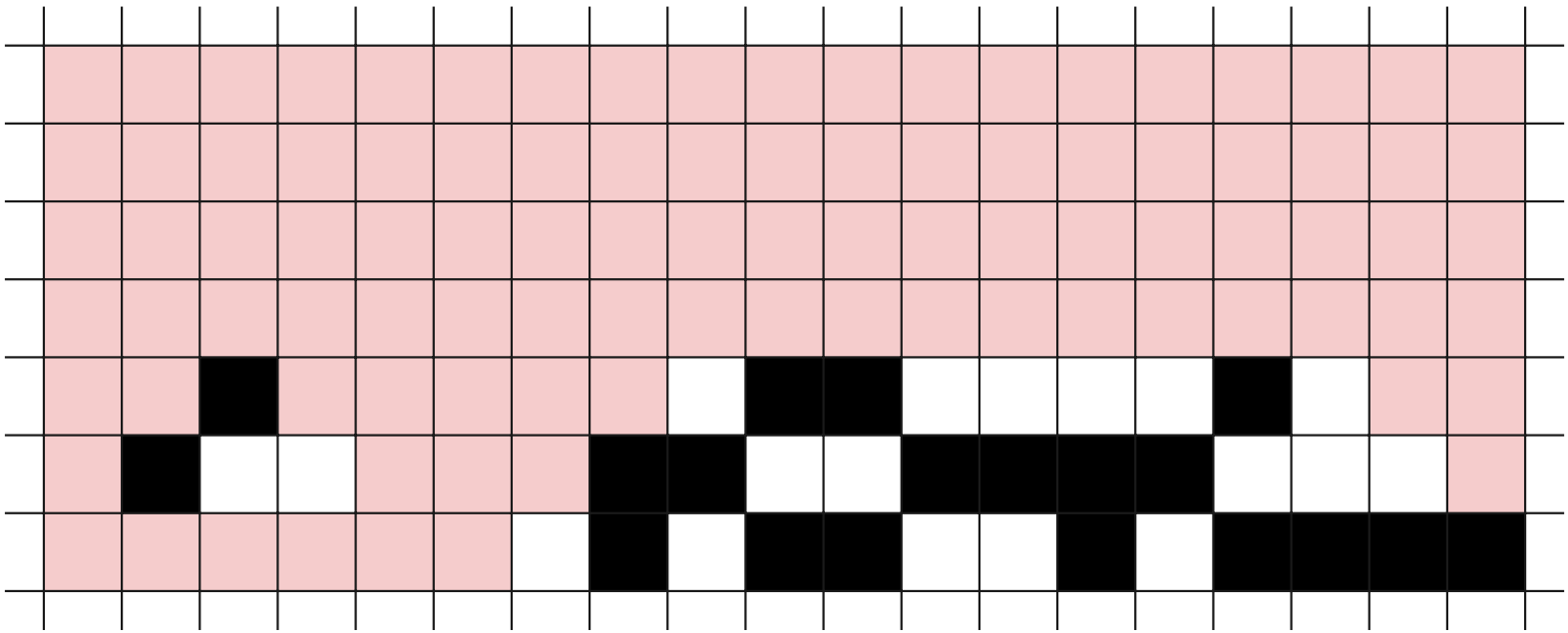


Computation complexity in 2D+

Prediction problem PRED_F Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

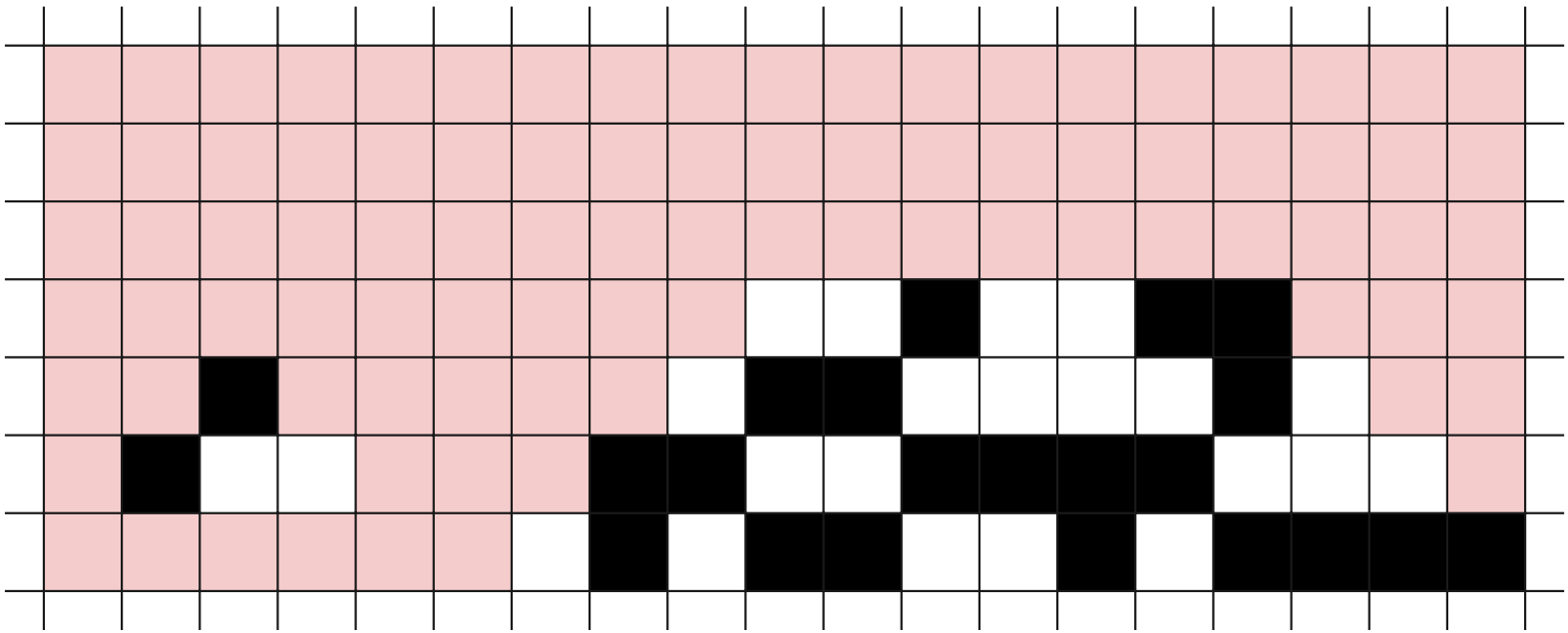


Computation complexity in 2D+

Prediction problem PRED$_F$ Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

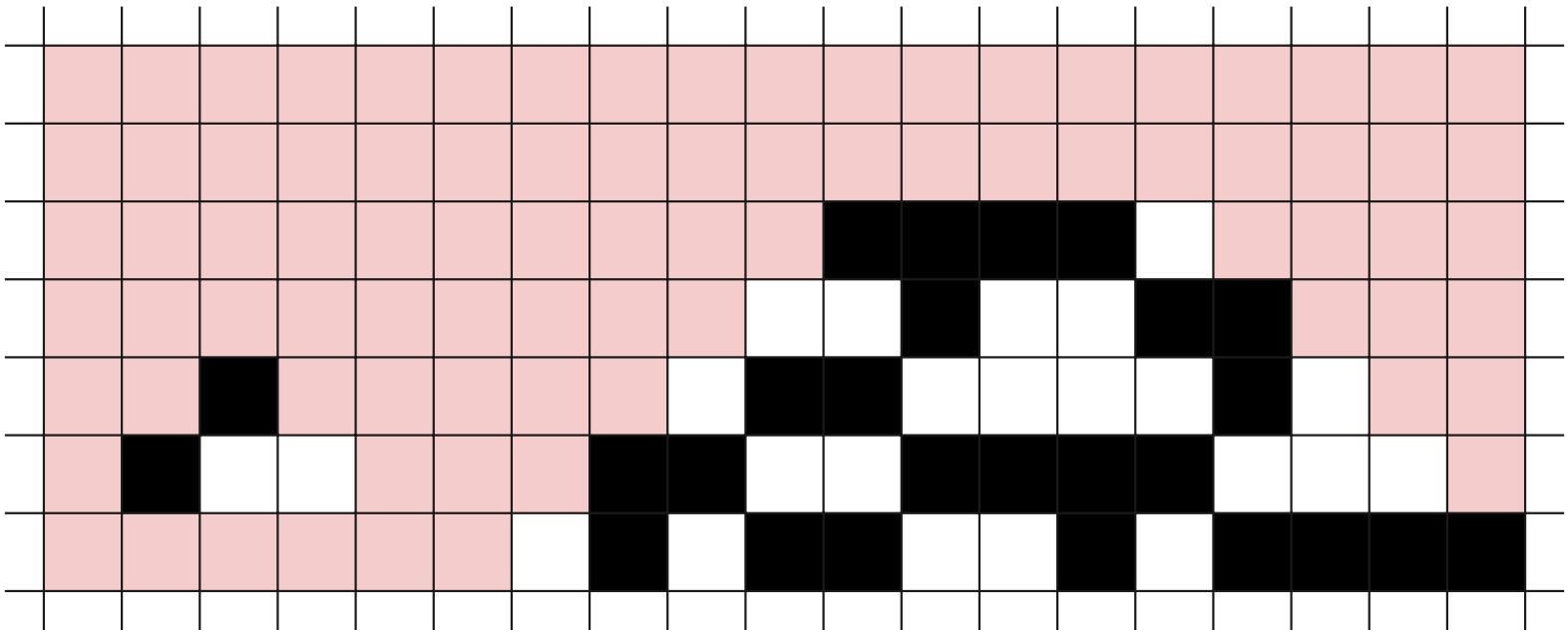


Computation complexity in 2D+

Prediction problem PRED$_F$ Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

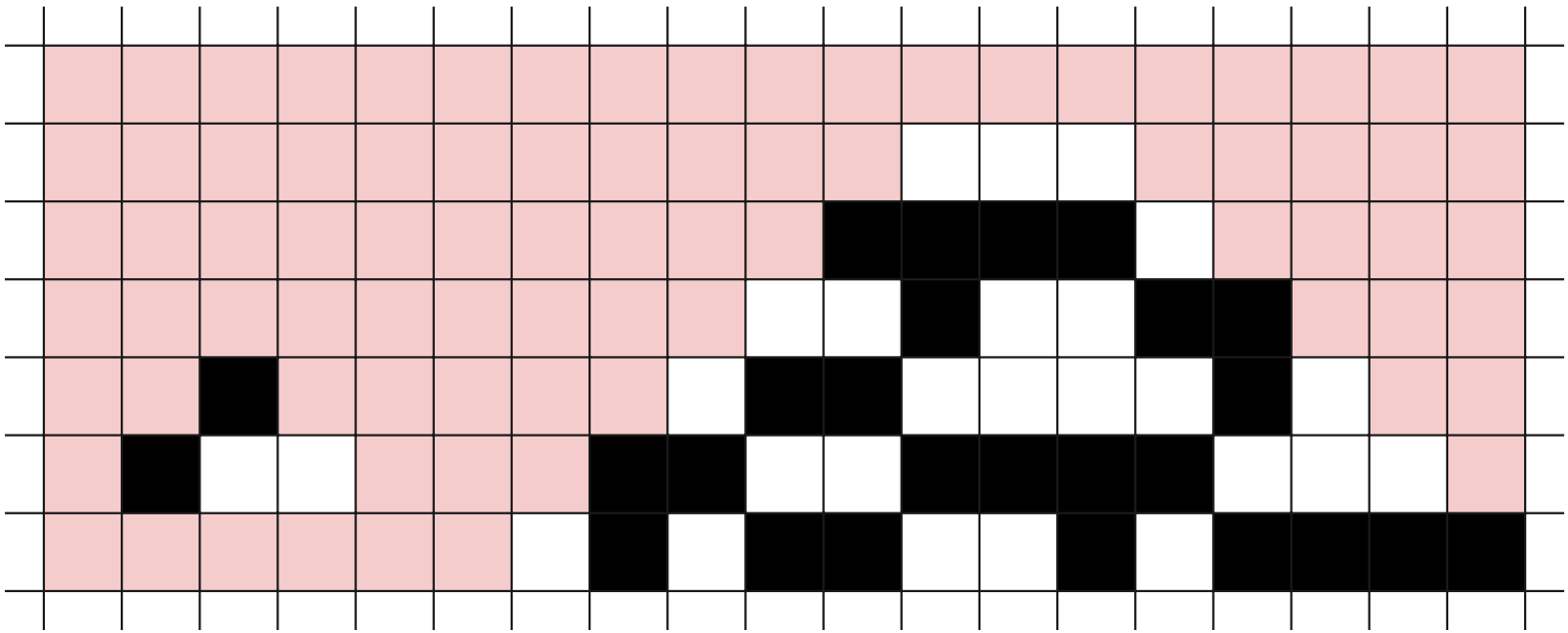


Computation complexity in 2D+

Prediction problem PRED$_F$ Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

Simulate any 1D CA with a 2D FCA with one more state.

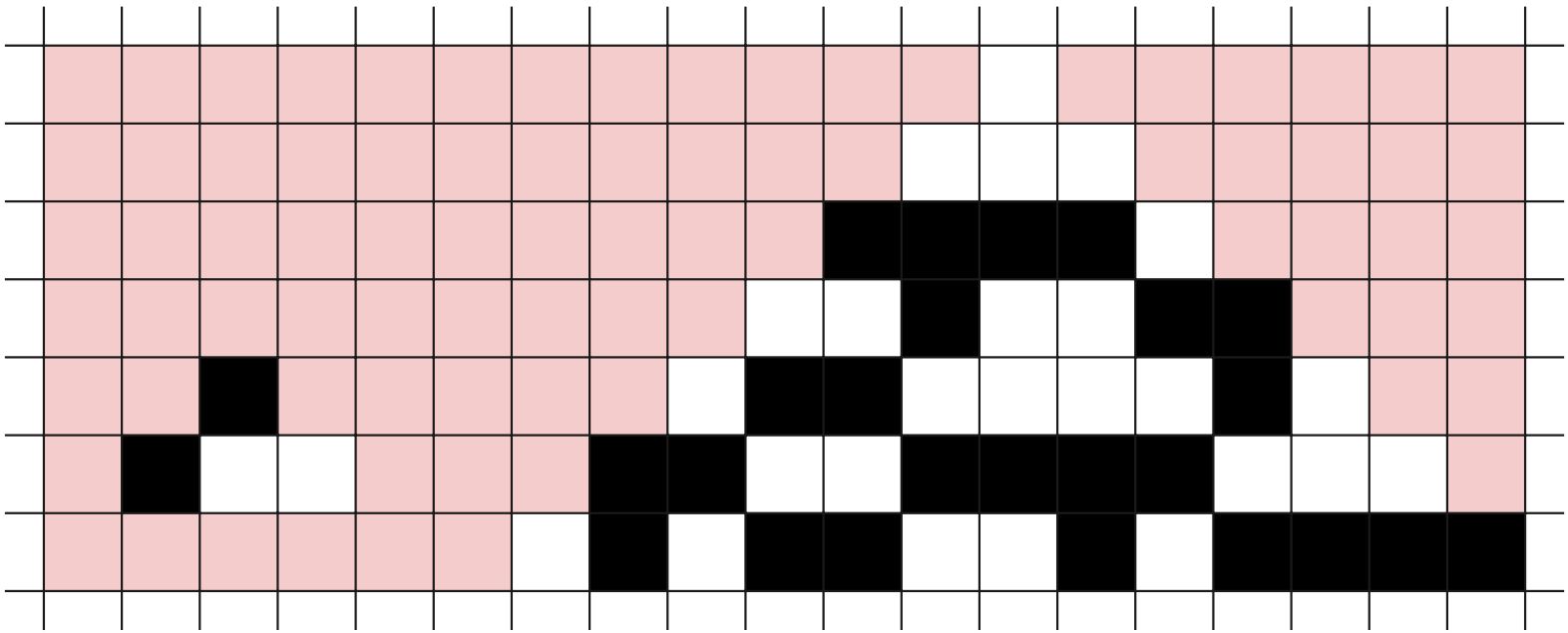


Computation complexity in 2D+

Prediction problem $\mathbf{PRED}_F$ Given a pattern of states u of size $(2rt+1)^d$ and a state a, determine if $F^t(u) = a$.

There exist a **Turing-universal** 2D FCA with a **P-complete** prediction problem.

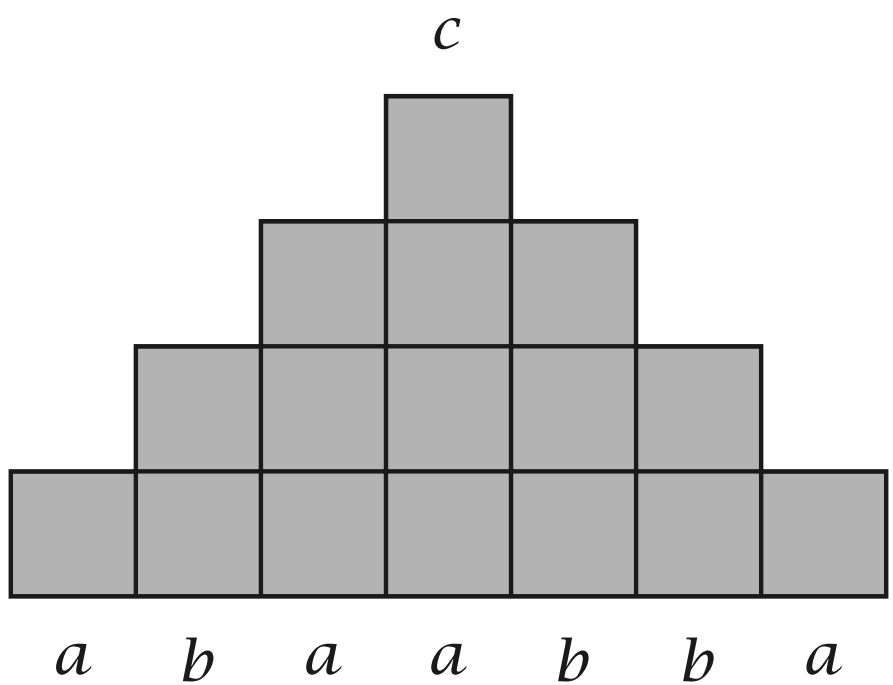
Simulate any 1D CA with a 2D FCA with one more state.



Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED$_F$ is **NLOGSPACE** for 1D FCA.

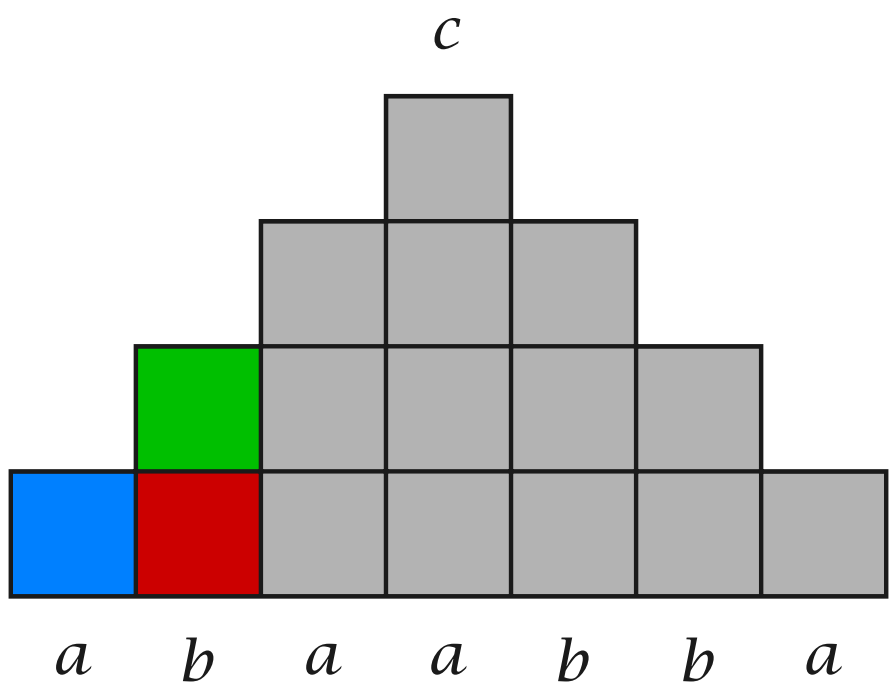


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED$_F$ is **NLOGSPACE** for 1D FCA.

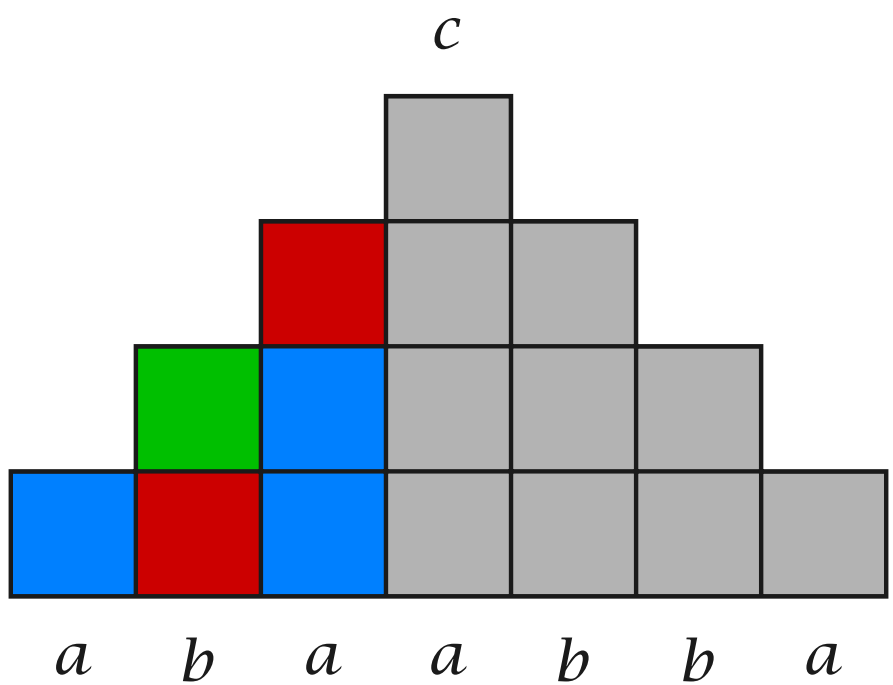


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED_F is **NLOGSPACE** for 1D FCA.

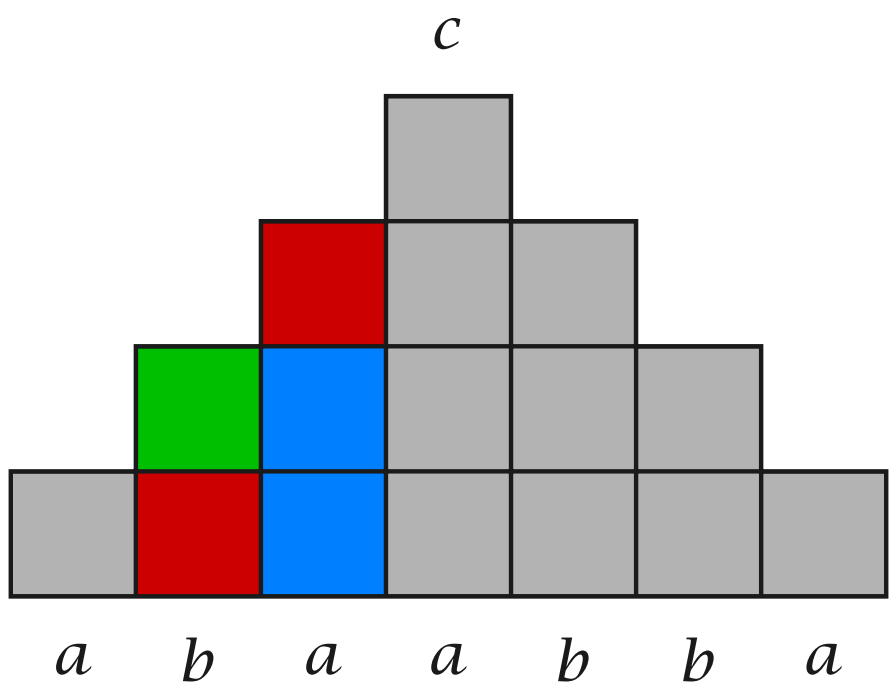


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED_F is **NLOGSPACE** for 1D FCA.

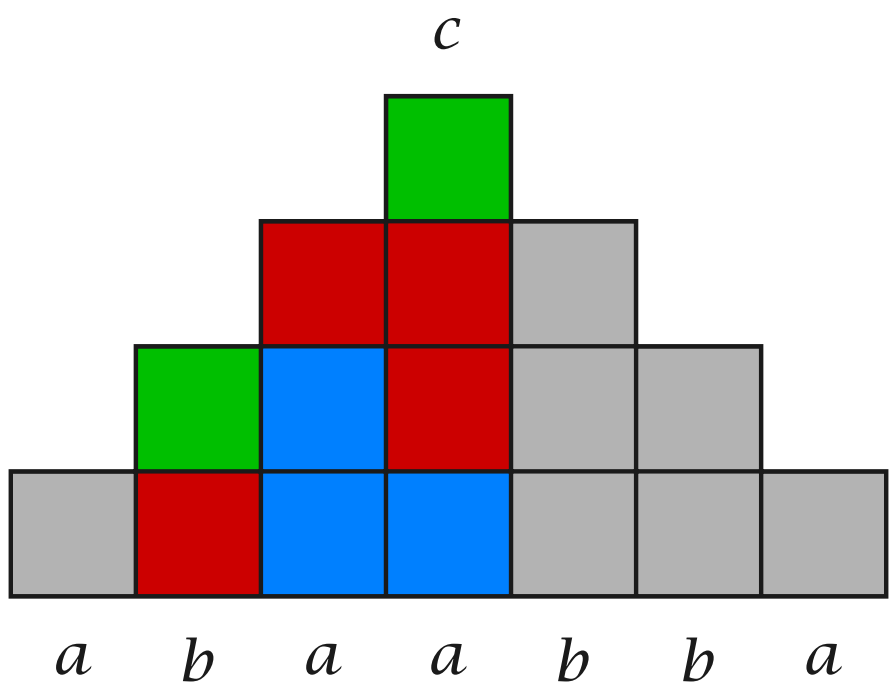


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED$_F$ is **NLOGSPACE** for 1D FCA.

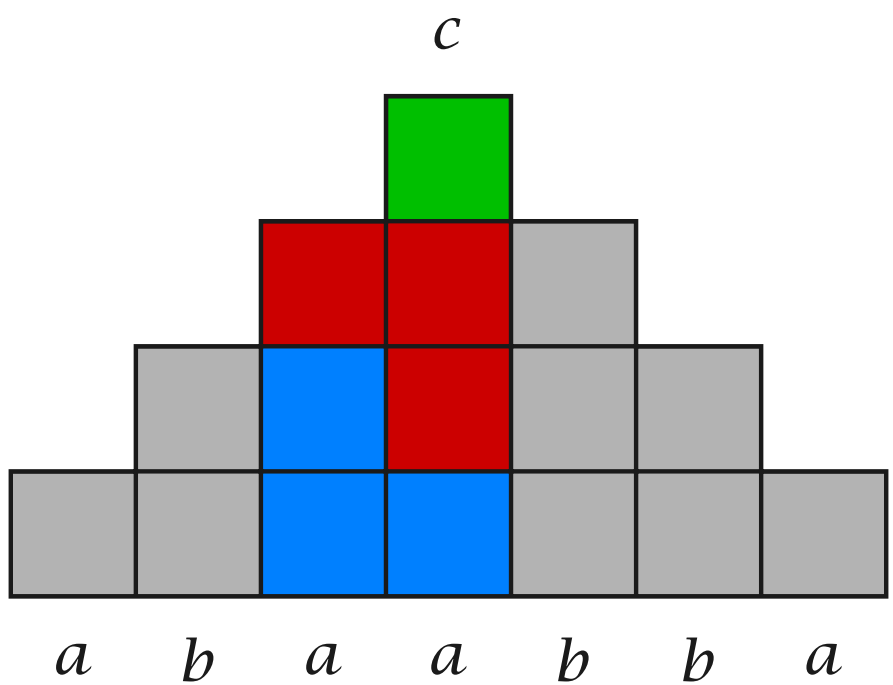


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED_F is **NLOGSPACE** for 1D FCA.

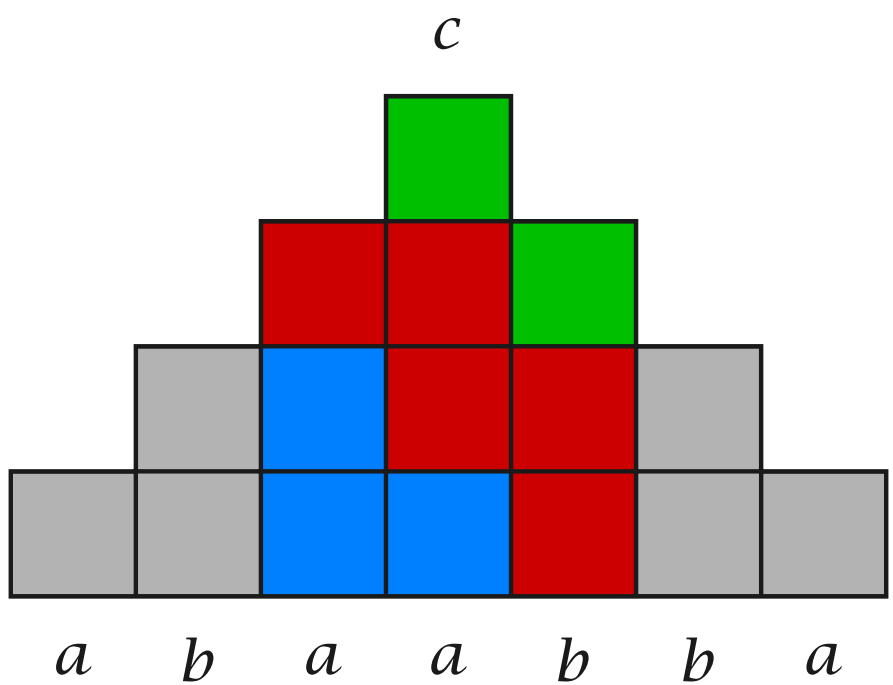


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED_F is **NLOGSPACE** for 1D FCA.

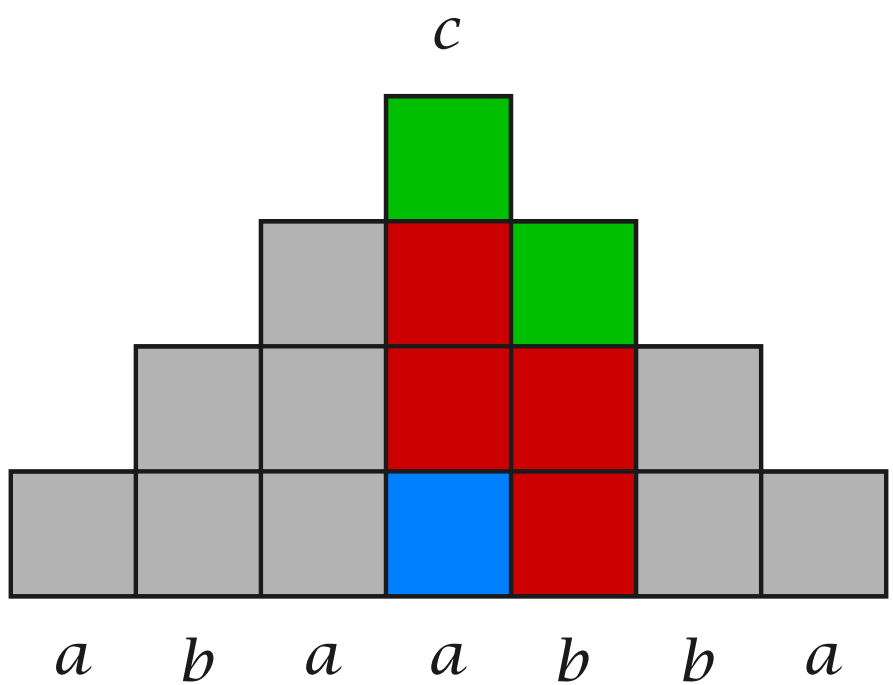


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED_F is **NLOGSPACE** for 1D FCA.

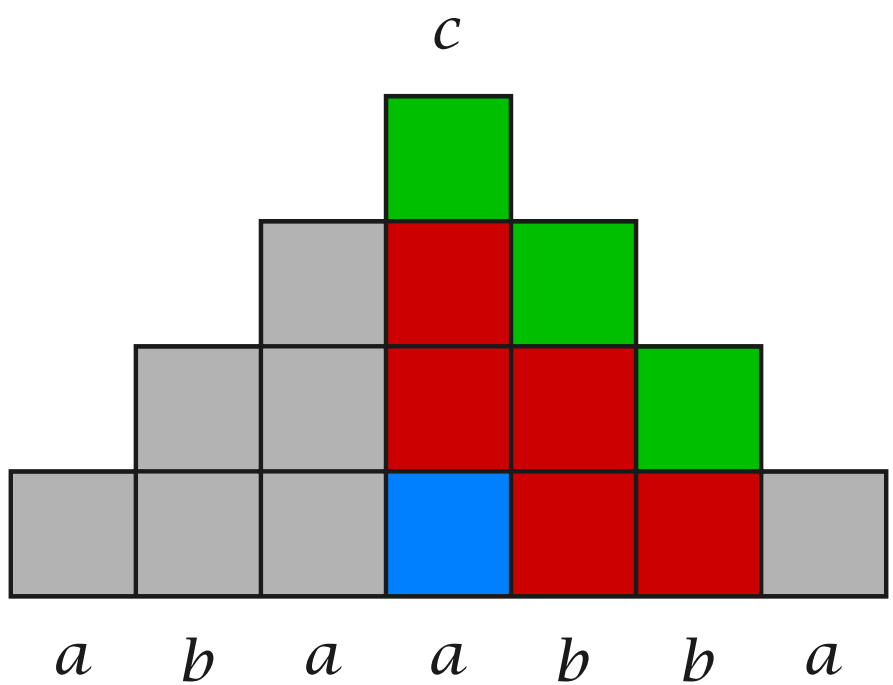


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED_F is **NLOGSPACE** for 1D FCA.

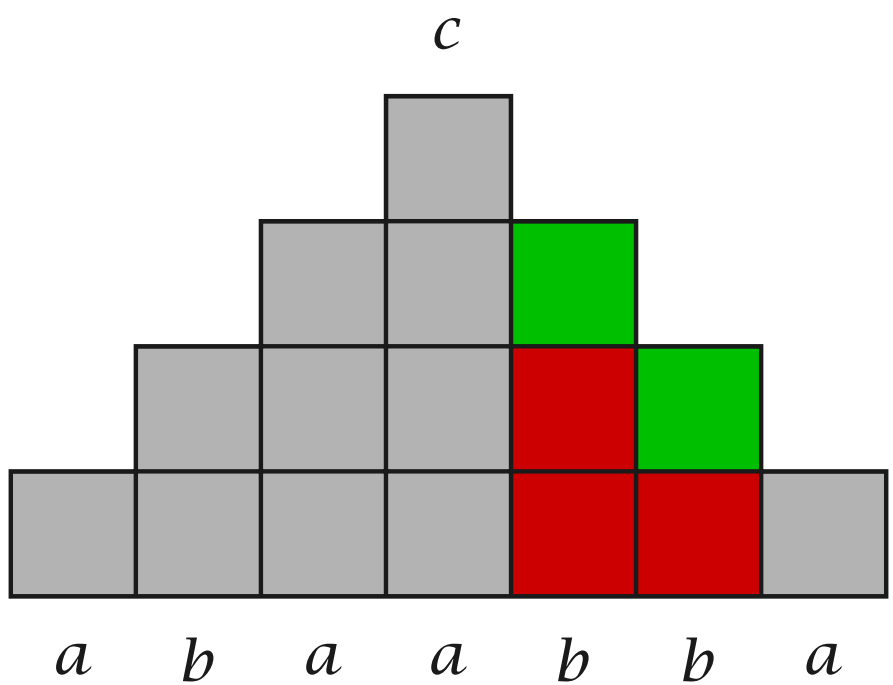


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED_F is **NLOGSPACE** for 1D FCA.

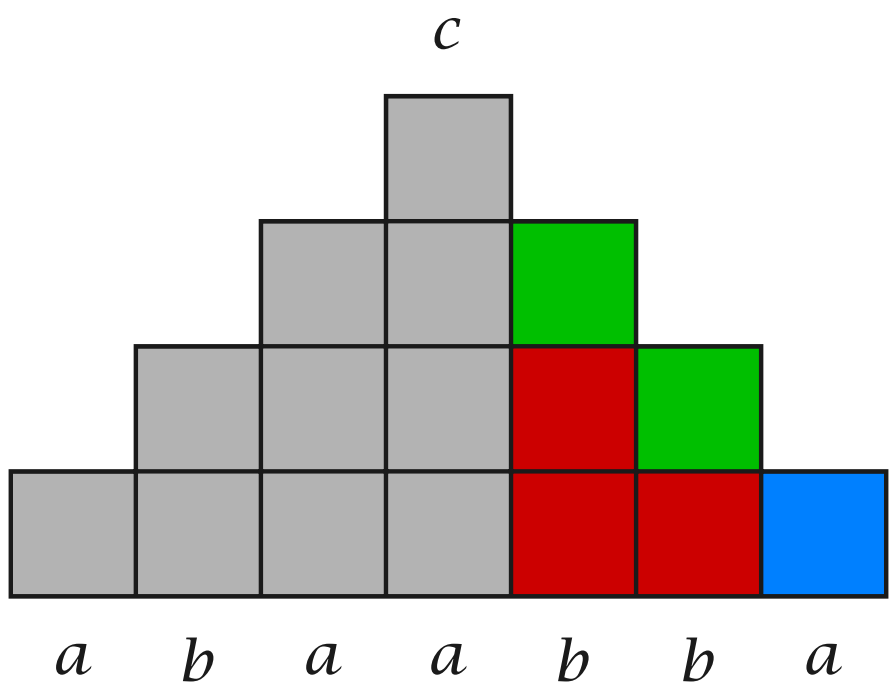


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

PRED$_F$ is **NLOGSPACE** for 1D FCA.

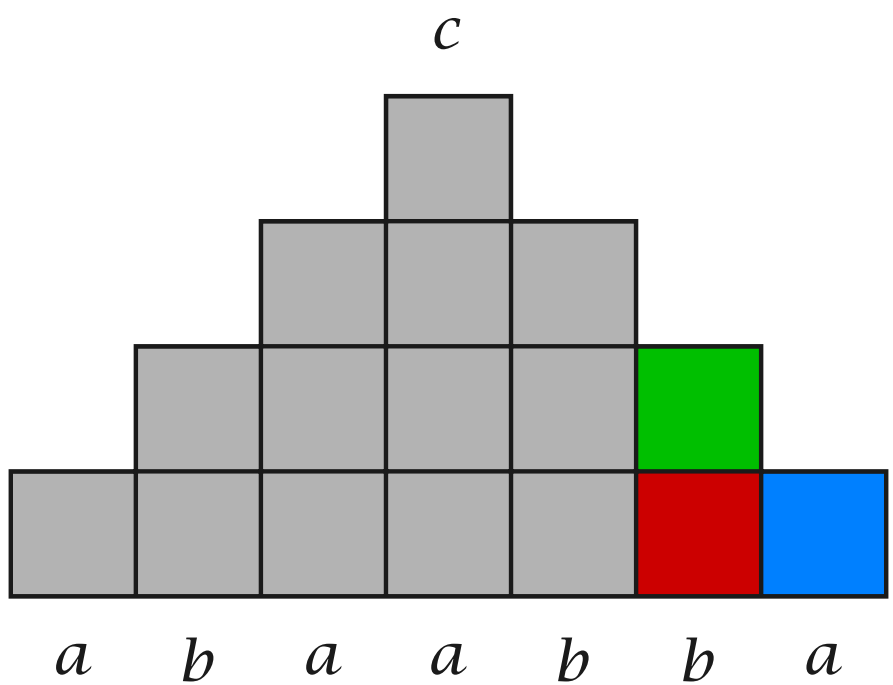


Each column can be described with $k \times \log(n)$ bits.

Computational complexity in 1D

The following result was already proved by **Vollmar in 1981**.

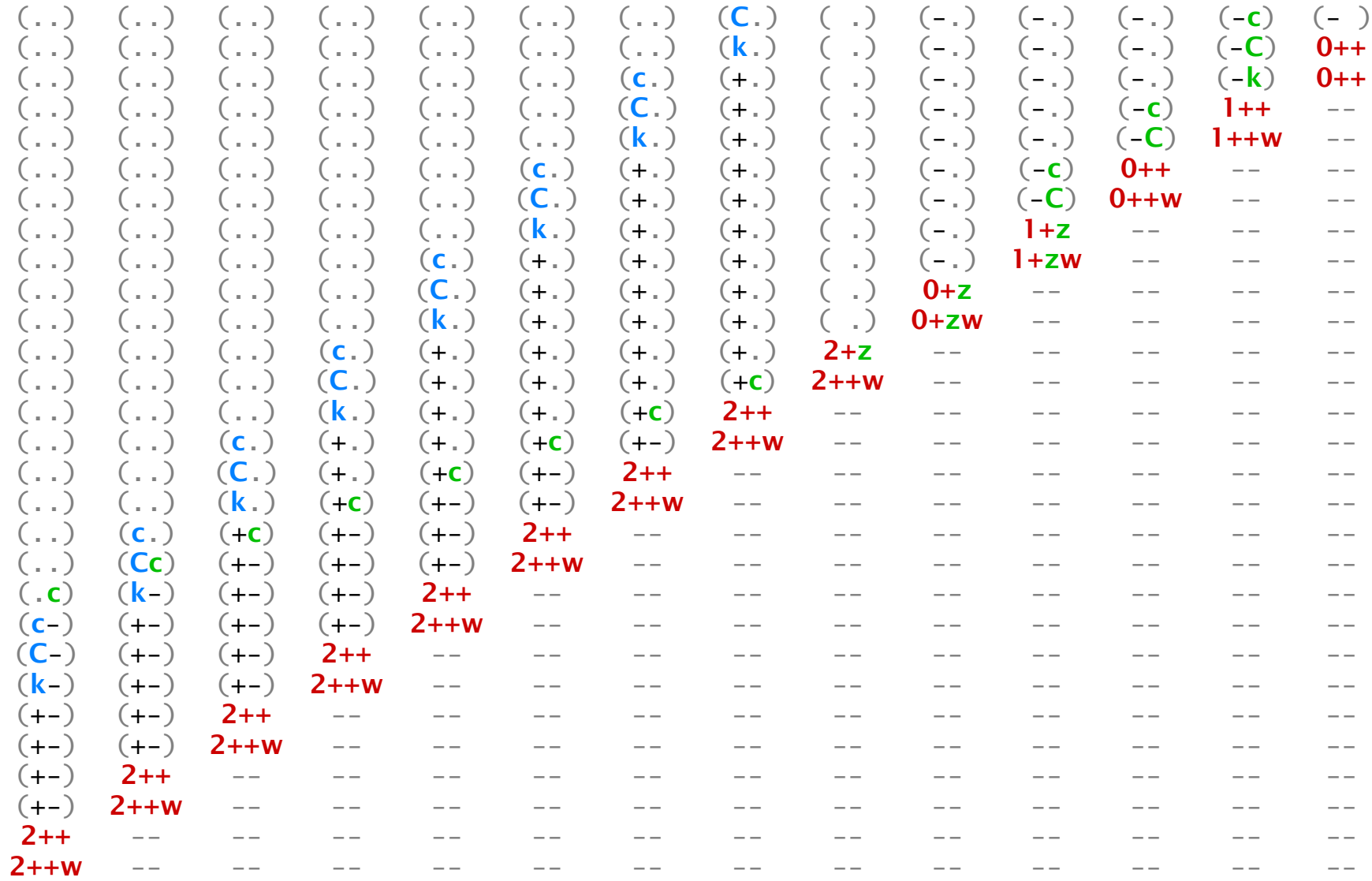
PRED_F is **NLOGSPACE** for 1D FCA.



Each column can be described with $k \times \log(n)$ bits.

Turing-universality in 1D

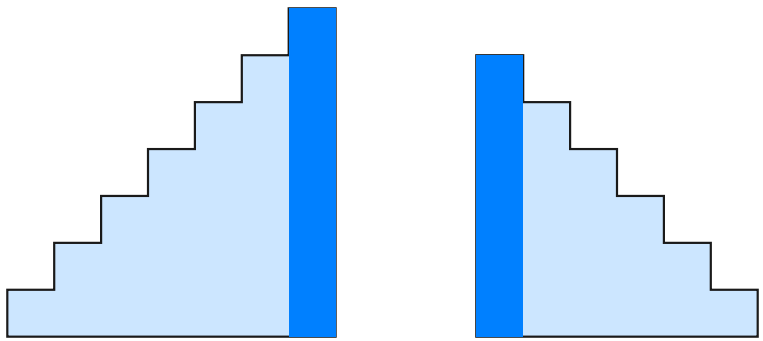
1D FCA are **Turing-universal**.



Communication complexity

Communication complexity of PRED_F is the cost of the best protocol to compute $F^t(u)$ when Alice and Bob both have a part of the pattern u.

Communication complexity of PRED_F is $O(n^{d-1}\log(n))$ on inputs of size n^d for any FCA.



The communicated boundary can be described with $k \times \log(n)$ bits.

What's next

Classify **limit sets** and **dynamics**

Study **intrinsic** freezing universality

Study **asynchronous** update modes

Study particular **order relations**