Computational analysis of the dynamics of logical regulatory graphs

Hierarchical compaction of state transition graphs

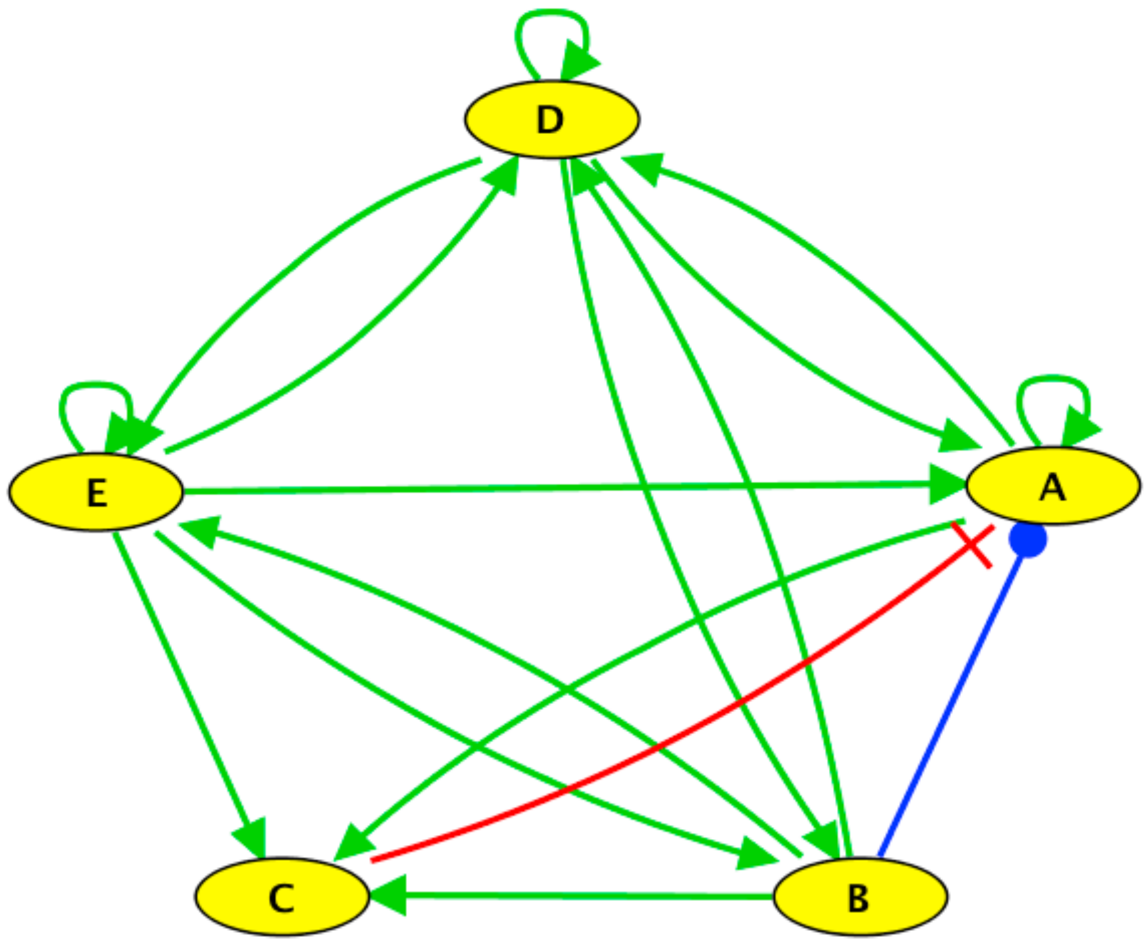
Duncan Berenguier

Claudine Chaouiya, Elisabeth Remy, Denis Thieffry

INSERM U928

Tuesday September 7 — JOBIM 2010

Model — Logical Regulatory Graph




```
A = (B & E & !C) | (A & !B & D & E)

B = D & E

C = A & B & E

D = D | (A & (B | E))

E = (D & B) | (E & D & !B)
```

A **node** is a **regulatory component** associated with a **discrete value** denoting its **level of activity**, *e.g.*

0 : Inactive

1 : Active

An **arc** is a **regulatory interaction**, whose effect (activation, inhibition, dual) is described by the logical function of its target

The **logical function** describe how the level of activity of a component evolve depending on its regulators

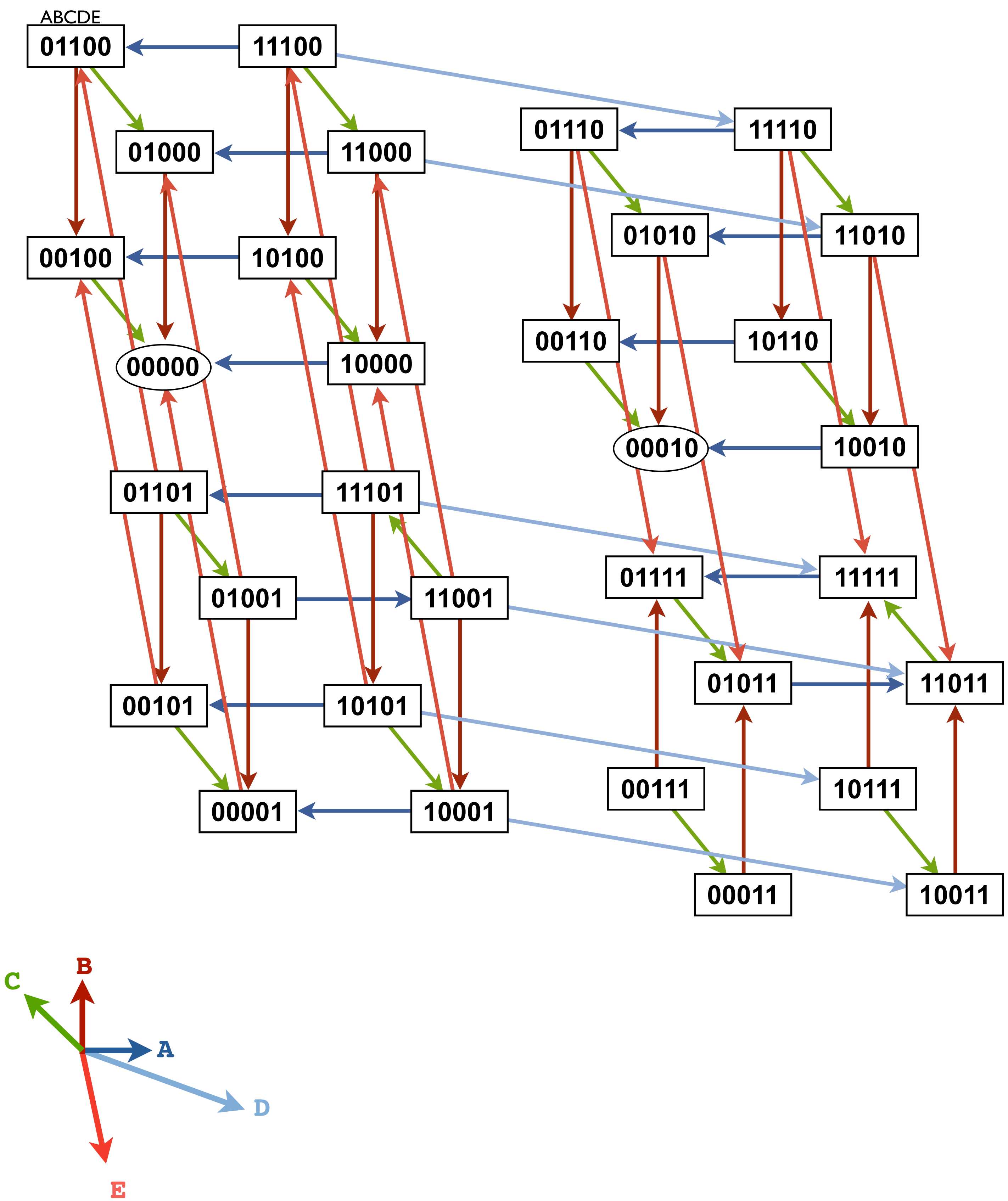
We consider several **updating strategies** :

synchronous

asynchronous

priorities classes

Dynamics — State Transition Graph



A node represents a **state** of the system

A **transition** between two states indicates a possible evolution

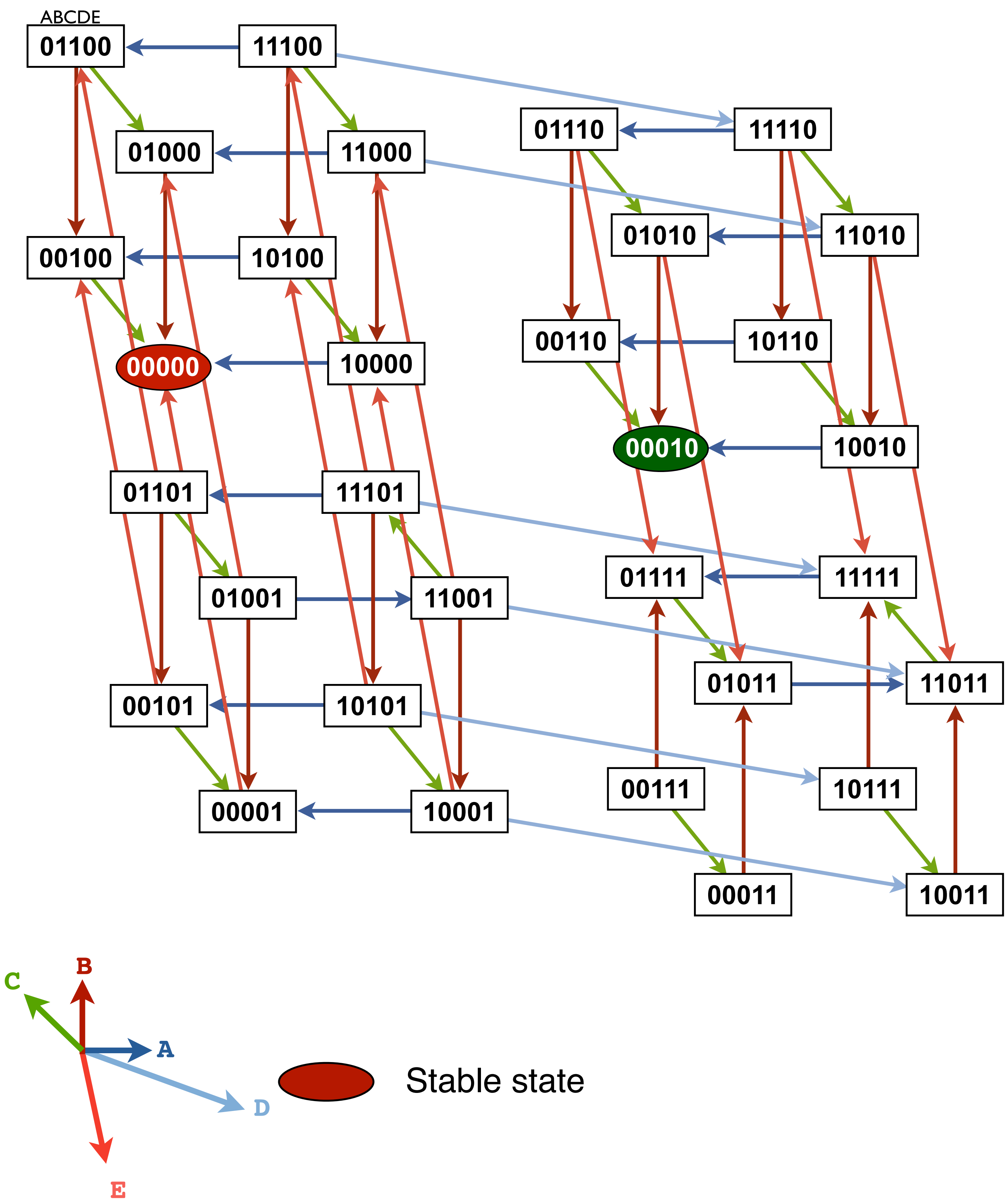
Problem : **Combinatory explosion**

5 components ⇒ 32 states

20 components ⇒ over 1 million states

N components ⇒ 2^N states

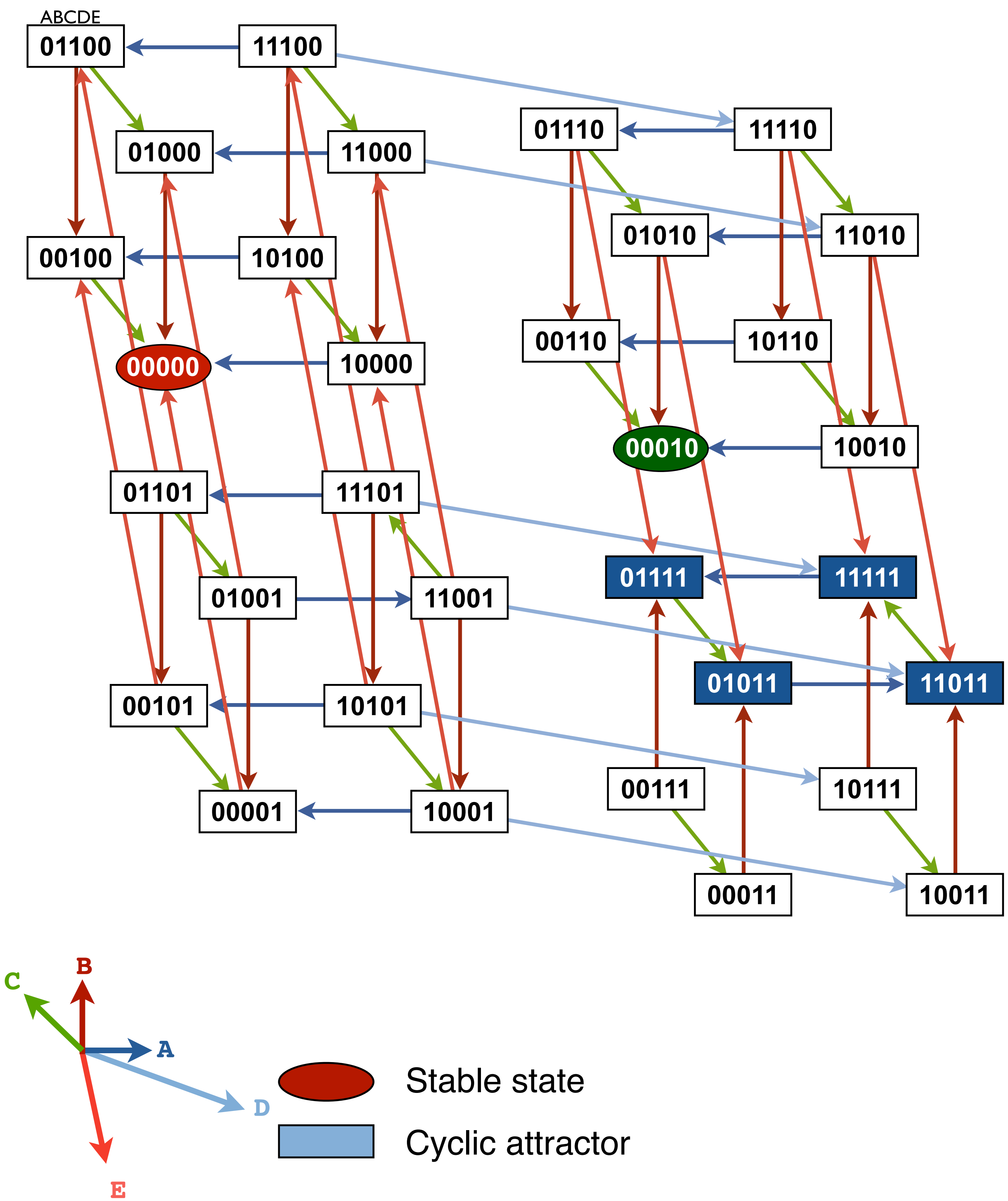
Dynamics — State Transition Graph



The asymptotic behaviour is described by **attractors**

Attractors correspond to **terminal strongly connected components**

Dynamics — State Transition Graph

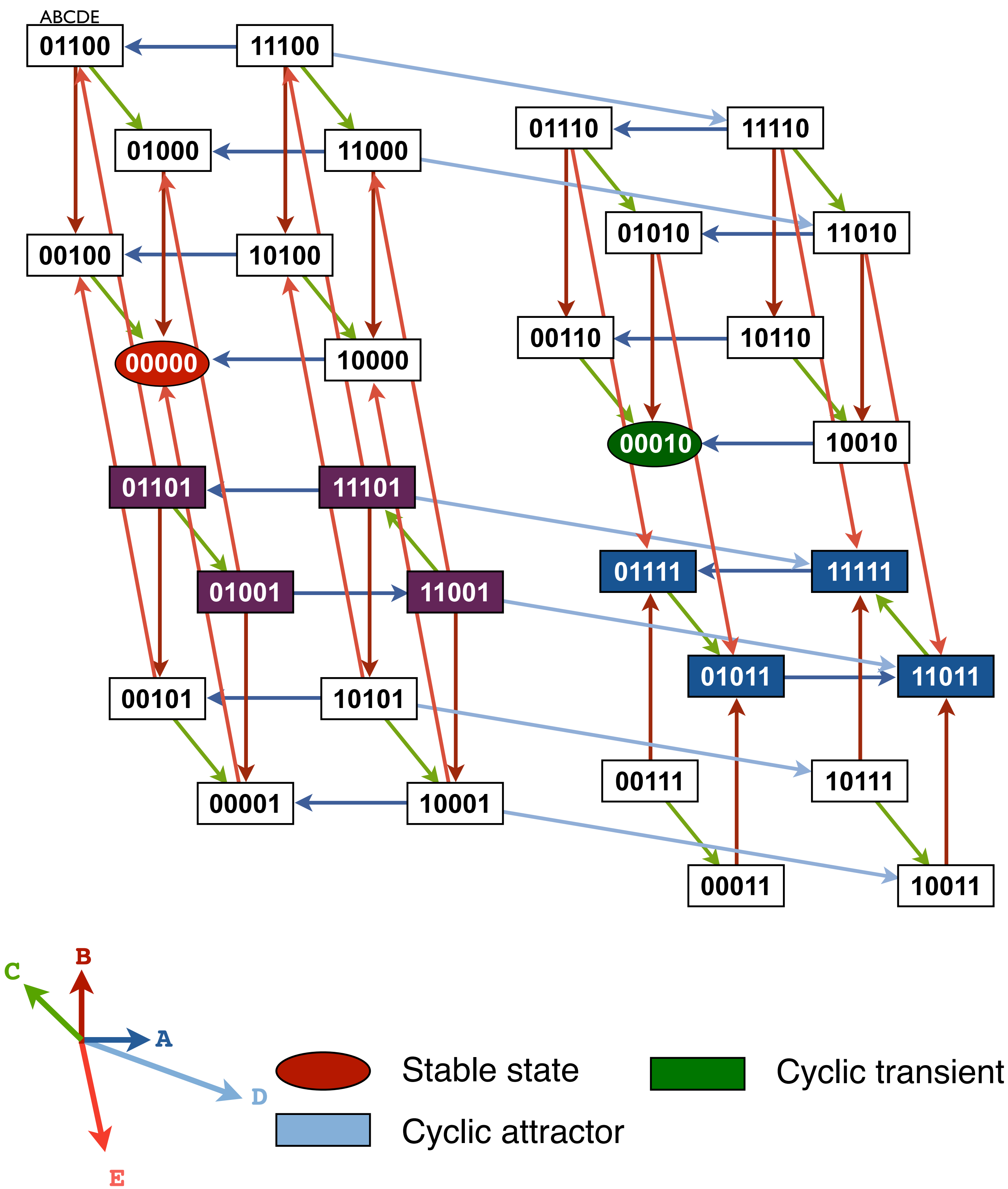


The asymptotic behaviour is described by **attractors**

Attractors correspond to **terminal strongly connected components**

The oscillatory behaviours are described by **cycles**

Dynamics — State Transition Graph



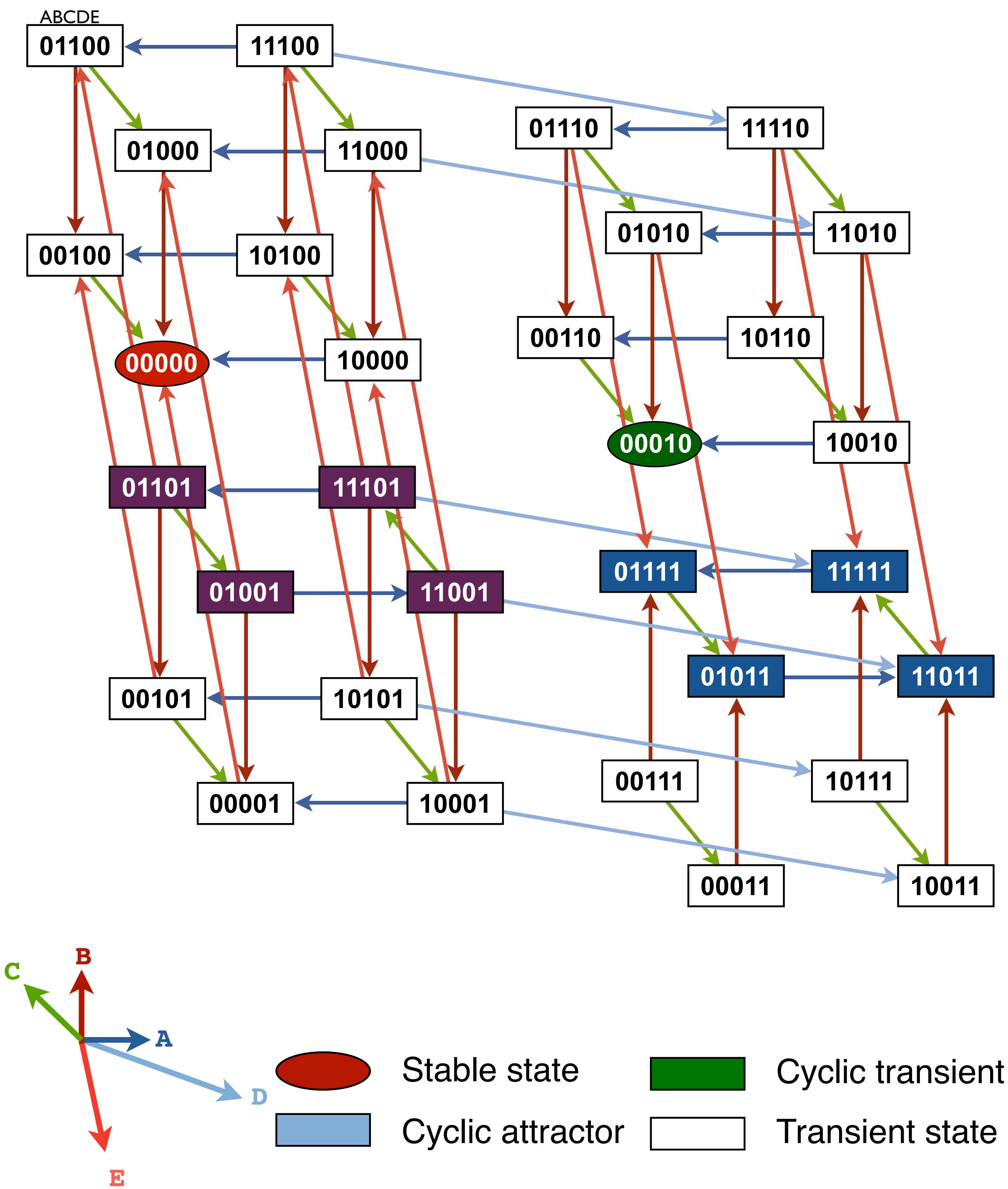
The asymptotic behaviour is described by **attractors**

Attractors correspond to **terminal strongly connected components**

The oscillatory behaviours are described by **cycles**

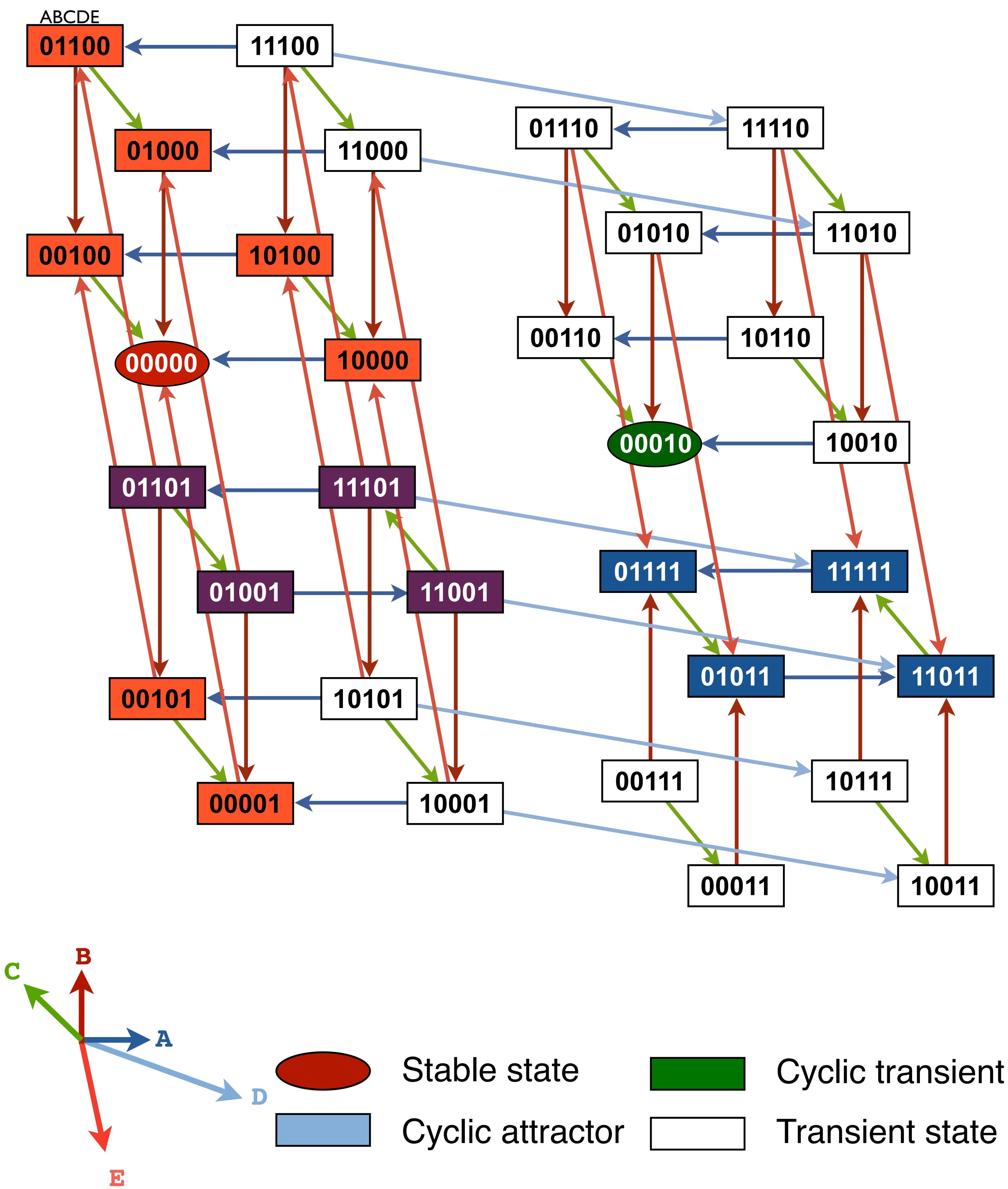
Transient oscillatory behaviours correspond to **transient** strongly connected components

Dynamics — State Transition Graph



What to do with the remaining **transient states** ?

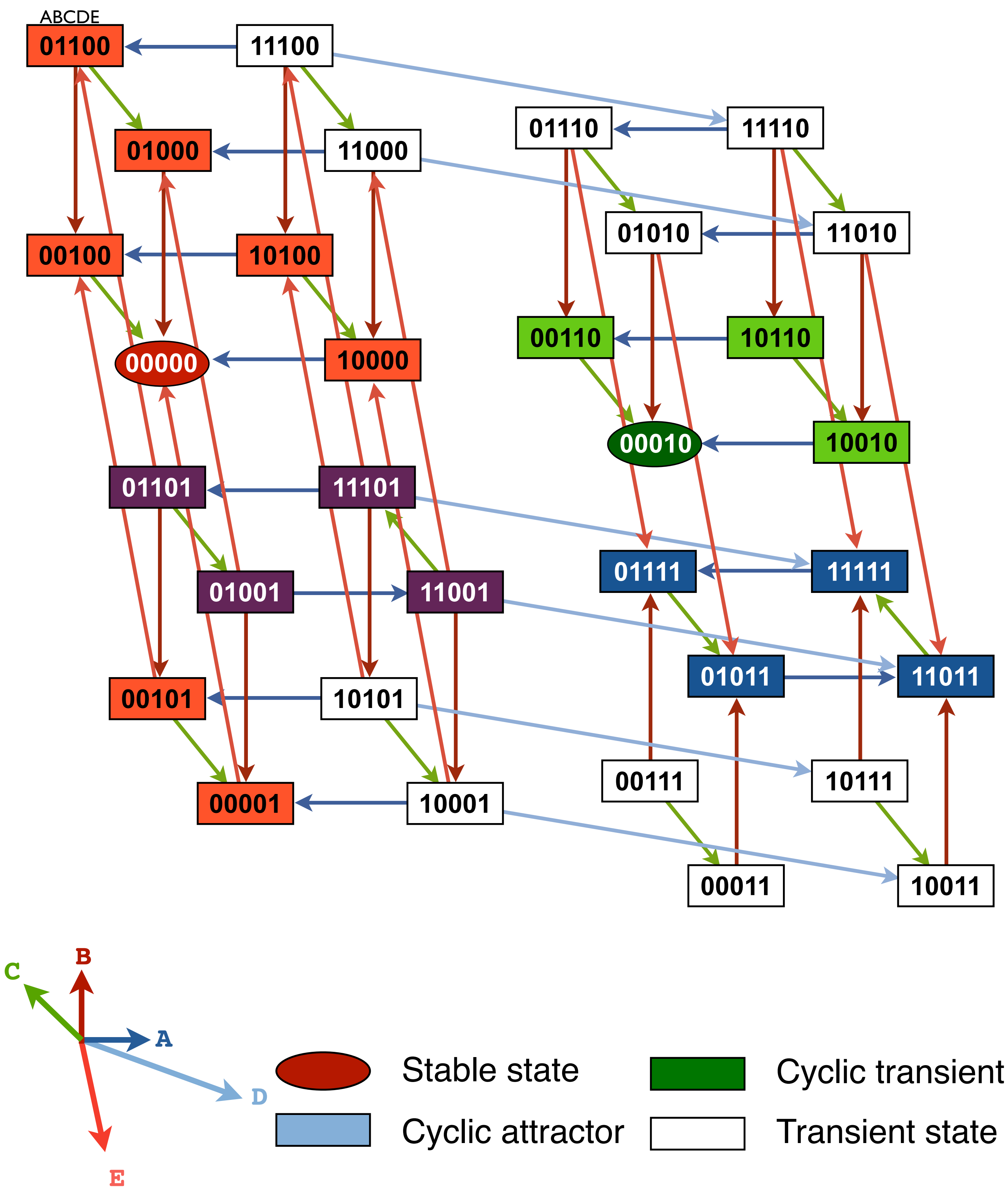
Dynamics — State Transition Graph



An (extended) **basin of attraction** is the set of node reaching a (set of) attractor(s)

We **classify** the transient states depending on they appartenance to a basin of attraction

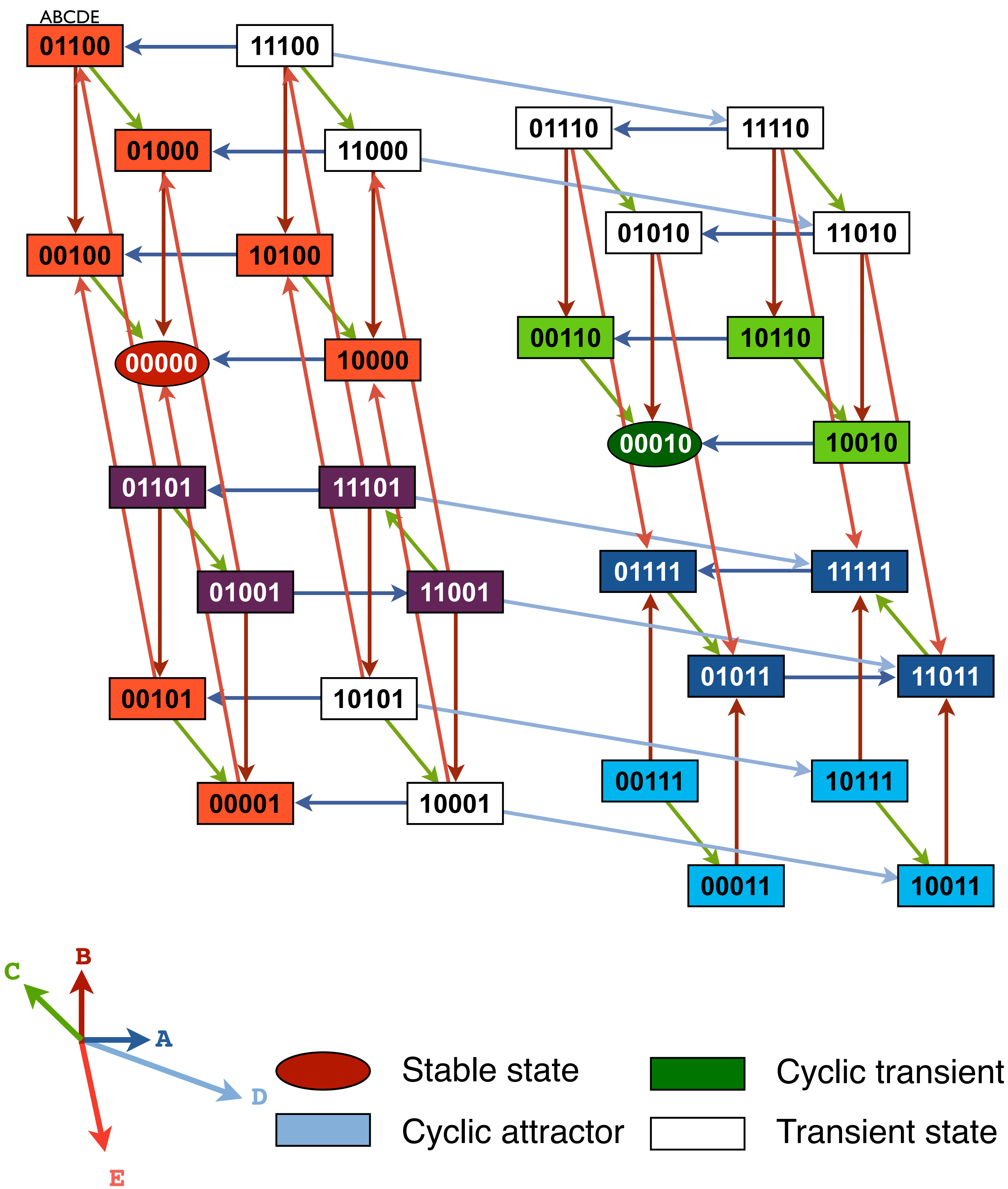
Dynamics — State Transition Graph



An (extended) **basin of attraction** is the set of node reaching a (set of) attractor(s)

We **classify** the transient states depending on they appartenance to a basin of attraction

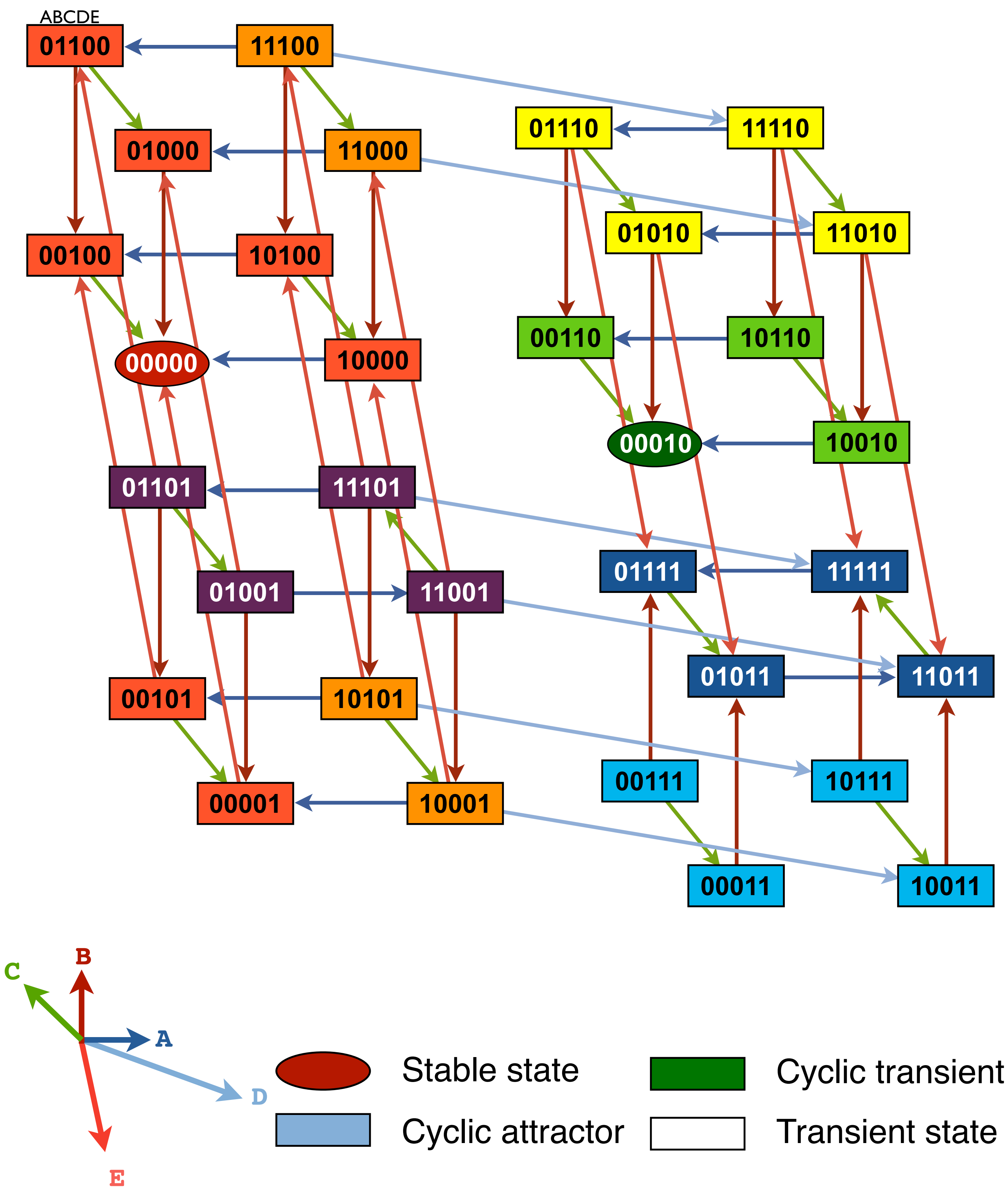
Dynamics — State Transition Graph



An (extended) **basin of attraction** is the set of node reaching a (set of) attractor(s)

We **classify** the transient states depending on they appartenance to a basin of attraction

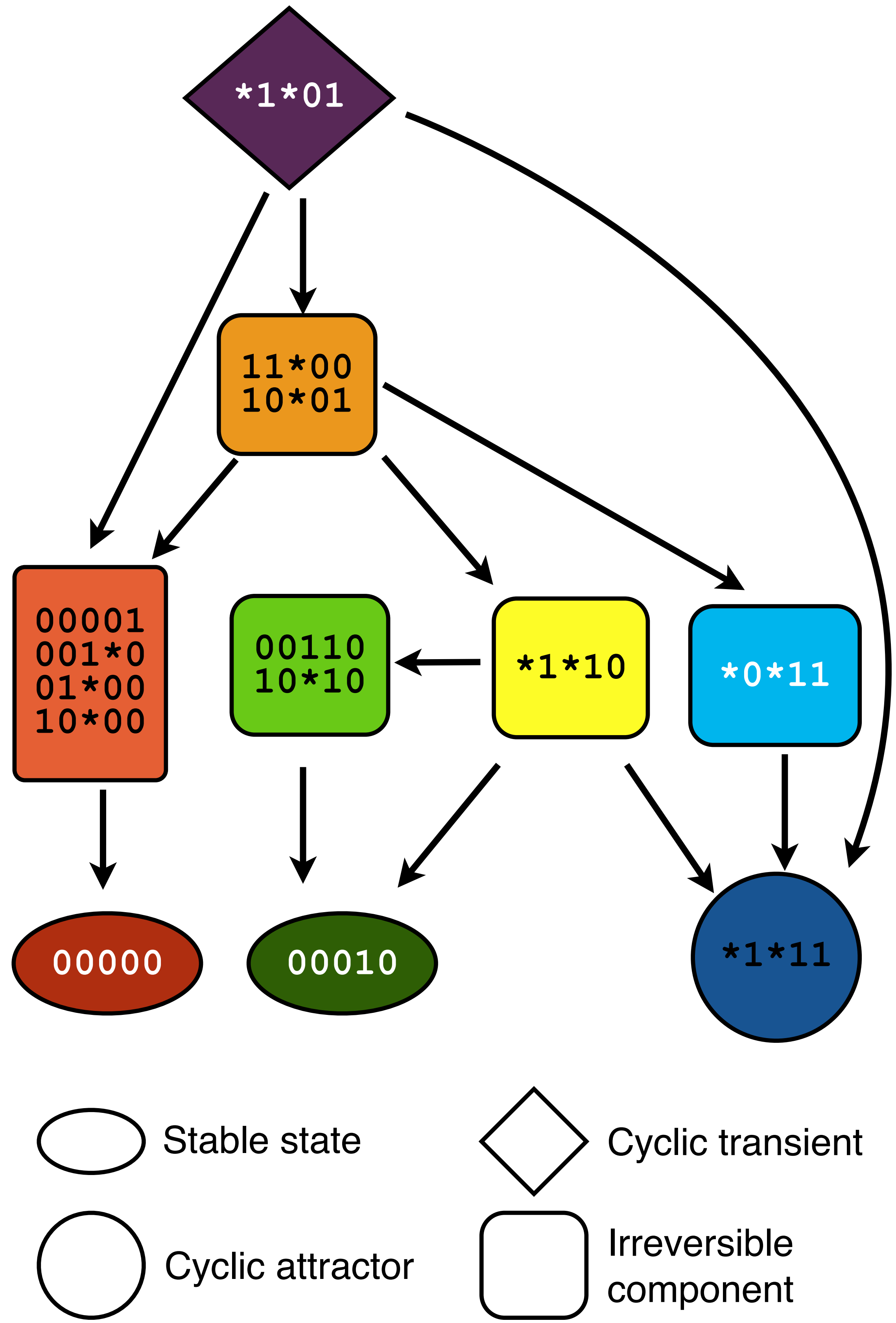
Dynamics — State Transition Graph



An (extended) **basin of attraction** is the set of node reaching a (set of) attractor(s)

We **classify** the transient states depending on they appartenance to a basin of attraction

Dynamics — Hierarchical Transition Graph



This classification of the states allows us to generate a **hierarchical transition graph**

A **node** represents a **set of states** (represented by schemata)

The graph is **smaller** and easier to analyse than the state transition graph

Take home message

The proposed compacted **hierarchical representation** of state transition graphs **eases the analysis** of the dynamics of large logical models

Prospects

What is the impact of **model reductions** on the organisation of hierarchical transition graphs ?

More generally, can we define a **class of models** producing the same hierarchical organisation ?

Thank you for your attention !

AGENCE NATIONALE DE LA RECHERCHE

APO-SYS

Belgian Inter-university Attraction Pole

Bioinformatics and Modelling : from Genomes to Networks