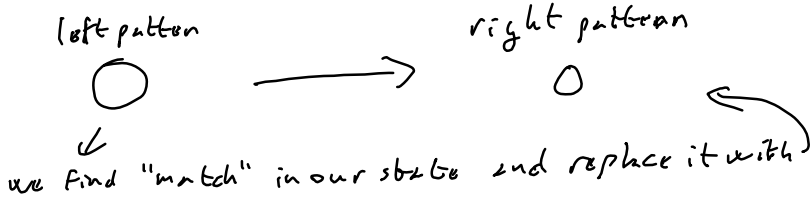
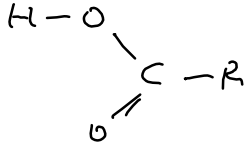
Kappa

Juri Kolčák

Formal Languages

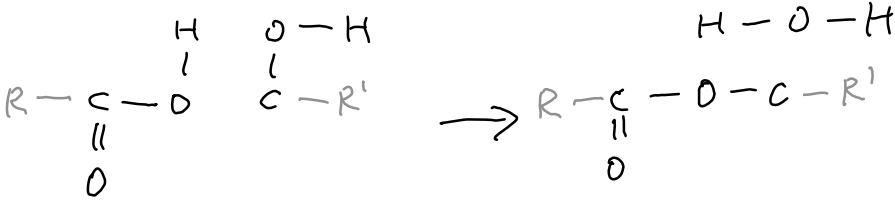
Formal languages are defined on strings, but the same **rule-based transformation** approach can be applied to different structures.





Carboxylic acid

Esterification: $RCOOH + R'COH \rightarrow RCOOCR' + H_2O$



Kappa

Kappa is a relatively young method, although graph transformation itself is much older.

~1973

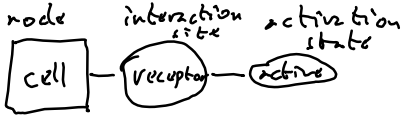
21st cen.

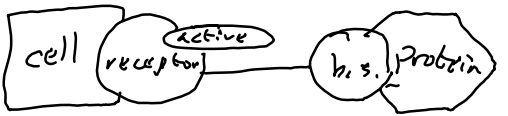
~2010

Applied to a special category of graphs, called **site graphs**.

site graph have nodes = agents / components of the system

each node has interaction sites





each interaction site can have an activation state

Each interaction has a binding state

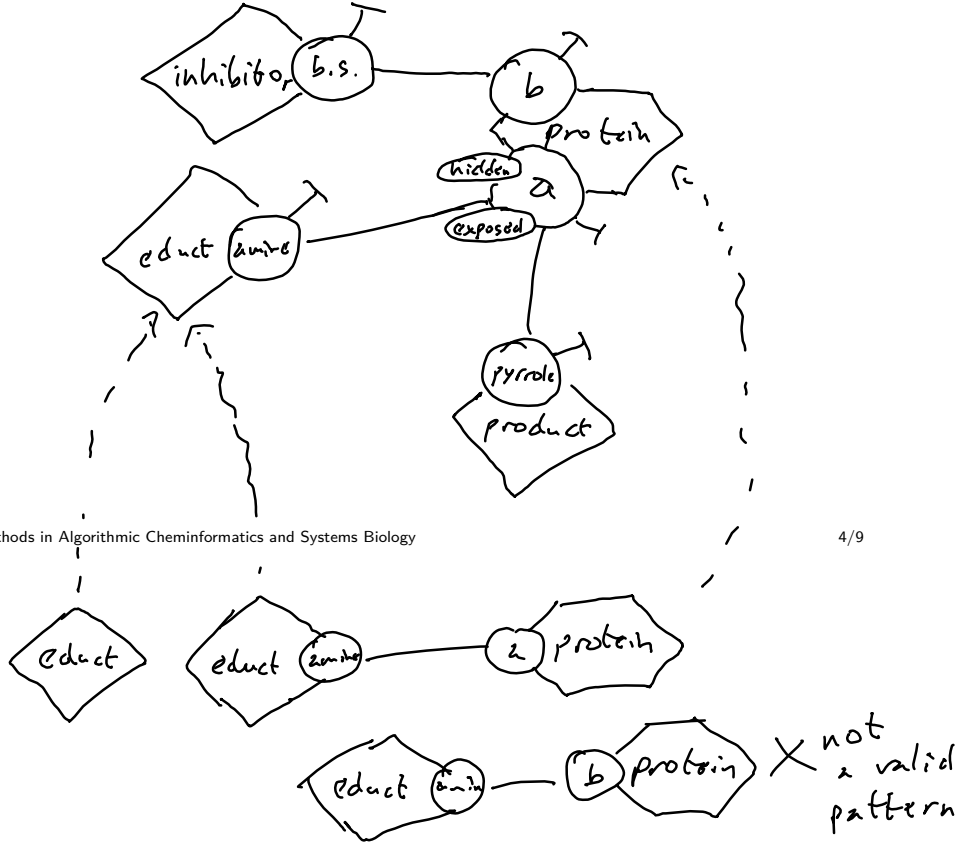
Each interaction site has to have a unique 'name'

site graphs are rigid

Type Graph

signature / contact map

Effectively the "alphabet" of the model, specifying all types of nodes (agents), their interaction sites, activation and binding states.



Patterns

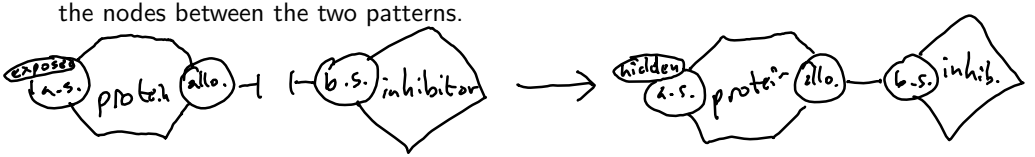
A pattern is a site graph for which the unique function mapping the pattern into the type graph is a homomorphism.

Patterns can also map into each other, these mappings are called **embeddings**.

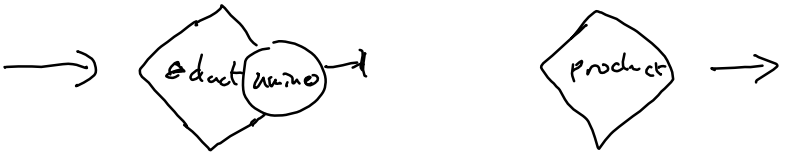
The maximal, fully specified, patterns in which all nodes specify all the interaction sites given by the type graph, and each interaction site specifies a binding state as well as an activation state if the type graph specifies any activation states for said site, as called **complexes** and make up the states of the model.

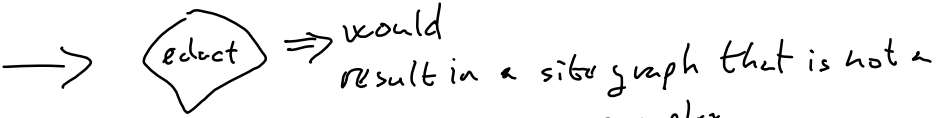
Rules

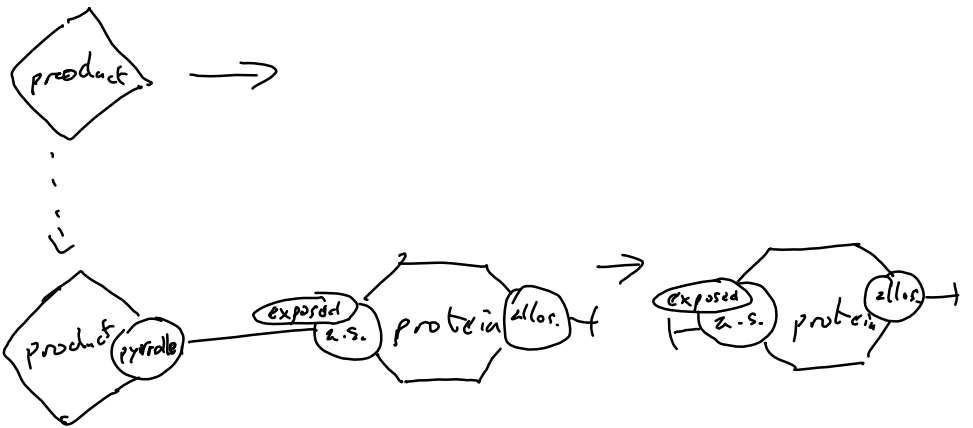
A rule is defined as a left and right pattern, with an implicit identity of the nodes between the two patterns.



A rule application consists of embedding the left pattern in a complex and then replacing it with the right pattern, thus creating a new complex.







Semantics

The underlying expanded network is the "explicit" representation of graph transformation model semantics.

In principle, there are multiple ways to interpret a Kappa model:

- Qualitative;
- Stochastic;
- Differential;

Simulation

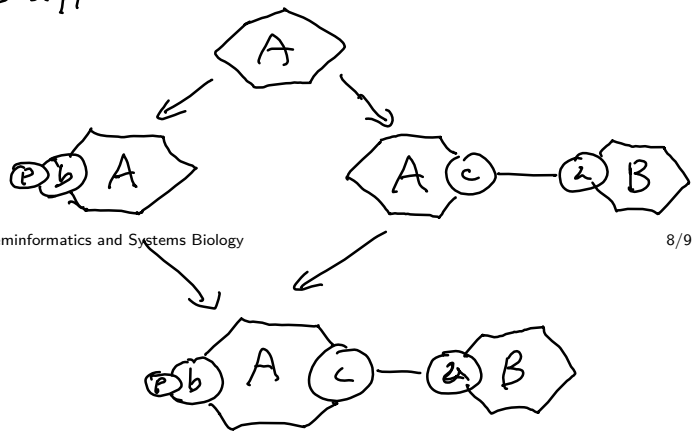
In a given state, all embeddings of all rules give the possible transitions. The rates are specified per rule.

$\leadsto \lambda$ for SSA

The states are generally much larger than left patterns of the rule. It is inefficient to recompute all the embeddings after each rule application.

"One can precompute which left patterns of a rules can be created or destroyed by rule applications."

gluing



category theory: pushout

Static Analysis

Identify (non-)reachability of a pattern without expanding the space.

Abstract interpretation-based method which computes an over-approximation of reachable patterns (false positives).

concrete $\{P \in \mathcal{P} \mid \exists M \in X, P \to M\}$ abstract

(α (abstraction), γ (concretization)) — $P \to M$: P embeds in M

$X \subseteq \mathcal{M}$ — subsets of states (mixtures)

$Y \subseteq \mathcal{P}$ — subsets of patterns

$\{M \in \mathcal{M} \mid \forall P \in \mathcal{P}, P \to M \Rightarrow P \to \gamma\}$

Choice of $\mathcal{P}$ heavily impacts both precision and efficiency.

Orthogal set of patterns for an agent (nodes) A (○)

set of pairs (P, f), $f: A \to P$

such that for any embedding $f'': A \to M \in \mathcal{M}$

$!\exists (P, f) \in ○ \wedge f': P \to M$ with $f'' = f' \circ f$

Set of patterns, such that any instance of the agent A can be explained with the embedding of a unique member of the set

Static Analysis provides a set of implications on patterns, such that whenever the left side is encountered, it can automatically be extended into the right side.