Project Presentation

24.07.2020

https://docs.google.com/presentation/d/1sbZMjxCUx9iFtrjptYZbGCkSuMbVUH0Yb_lUlHPoRZY/edit?usp=sharing

Introduction

In the previous presentation we looked at:

- a WA minimization algorithm by [Kiefer et al., 2013]
- a WA equivalence testing algorithm
- some implementation details

Since then:

- Parts of an ODE/CRN reduction algorithm have been implemented

Polynomial Ordinary Differential Equations

- System of equations of the form :

$$\frac{dx_h}{dt} = \sum_{i=1}^{n} a_i \prod_{j=1}^{k} x_j^{p_j}$$
$$= a_1 x_1^{p_1} x_2^{p_2} \cdots + a_2 x_1^{p_1} x_2^{p_2} \cdots + \ldots$$

- The superscript p may be power or derivative

Chemical Reaction Networks

Tuple (S, R) consisting of

- set of species $S = \{S_1, S_2, \ldots\}$
- set of reactions $R = \{lhs_1 \to rhs_1, \ldots\}$
- lhs & rhs consists of a multiset of species, i.e. $\sum_{i=1}^{k} a_i S_i$
- Each reaction has a rate $r_i = k_i \prod_j x_j^{a_i}$
- Rate follows mass action kinetics, x_j is the concentration of species j
- Other notions possible, see [Petrov, 2018]

Tribastone et al.: ODE to RN conversion

We encode each variable x_i with species S_i and each monomial $\alpha \prod_i x_i^{p_i}$ appearing in the ODE of x_k with the reaction

$$\sum_{i=1}^{n} p_i S_i \xrightarrow{\alpha} S_k + \sum_{i=1}^{n} p_i S_i, \qquad \textbf{[5]}$$

where the operator $+$ denotes multiset union and $p_i S_i$ is a multiset with p_i occurrences of S_i.

Tribastone et al.: Partition Refinement for ODEs

Basic schema:

1. actually uses RN structures => use previous method as first step
2. compute a quantity depending on current splitter per Species
3. split species with distinct value in new partition
4. goto 2 until there are no splitters left
5. apply partition to CRN
6. Convert back to ODEs

Quantities to split by

Change of species wrt. the set of reactions having rho as lhs

$$\phi(\rho, S_i) := \sum_{(\rho \xrightarrow{\alpha} \pi) \in R} (\pi_i - \rho_i) \cdot \alpha$$

Change of all species in a partition wrt. reactions having rho as lhs

$$\phi(\rho, G) := \sum_{S_i \in G} \phi(\rho, S_i)$$

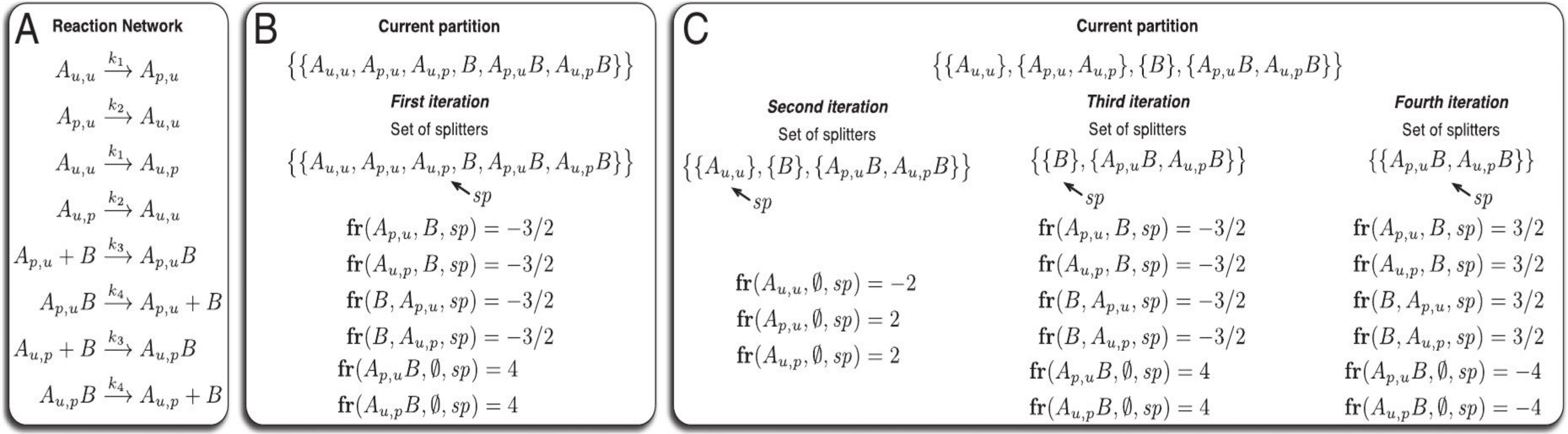
FDE

$$\mathbf{fr}(S_i, \rho, G) := \frac{\phi(S_i + \rho, G)}{[S_i + \rho]!}$$

BDE

$$\mathbf{br}(S_i, \mathcal{M}, H') := \sum_{S_k \in H'} \sum_{\rho \in \mathcal{M}} \frac{\phi(S_k + \rho, S_i)}{|S_k + \rho|_{H'}}$$

Example



	$\emptyset$	$A_{p,u}$	$A_{u,p}$	B
$A_{u,u}$	0	0	0	0
$A_{p,u}$	0	0	0	$\frac{-3}{2}$
$A_{u,p}$	0	0	0	$\frac{-3}{2}$
B	0	$\frac{-3}{2}$	$\frac{-3}{2}$	0
$A_{u,p}B$	4	0	0	0
$A_{p,u}B$	4	0	0	0

$$A^*_{u,u} \xrightarrow{2k_1} A^*_{p,u}$$

$$A^*_{p,u} \xrightarrow{k_2} A^*_{u,u}$$

$$A^*_{p,u} + B^* \xrightarrow{k_3} A_{p,u}B^*$$

$$A_{p,u}B^* \xrightarrow{k_4} A^*_{p,u} + B^*$$

Reminder: WA minimization

Find a basis F of the prefix space using random vectors r_i

- Add the vectors of all prefix words up to length n together and multiply this vector by n different factors yielding $\{v_1, \ldots, v_n\}$
- Factors are derived by random vectors and structure of prefixes
- Base is then the maximally linear independent subset of $\{\alpha, v_1, \ldots, v_n\}$

Use basis to do Schützenberger Construction [3]: $\overrightarrow{\mathcal{A}} = (\overrightarrow{n}, \Sigma, \overrightarrow{\alpha}, \overrightarrow{\mu}, \overrightarrow{\eta})$ With

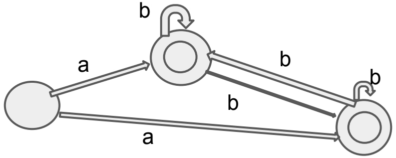
- $\overrightarrow{\mu} = \overrightarrow{F}\mu\overrightarrow{F}^{-1}$ or $\overrightarrow{F}\mu = \overrightarrow{\mu}\overrightarrow{F}$
- $\overrightarrow{\alpha} = e_1$
- $\overrightarrow{\eta} = \overrightarrow{F}\eta$
- $\overrightarrow{n} = \text{rank}(\overrightarrow{\mu})$

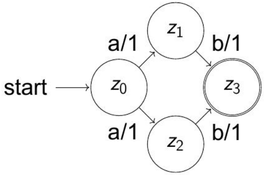
Example:

Weighted automaton $\mathcal{A} = (n, \Sigma, \mu, \alpha, \eta)$ with

- $n = 3,$
- $\Sigma = \{a, b\},$
- $\alpha = (1, 0, 0, 0),$
- $\eta = (0, 0, 0, 1),$

- $\mu(a) = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$
- $\mu(b) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$





Find basis of prefix space

$$r^{(1)} = \begin{pmatrix} 9 & 5 & 5 & 7 \\ 6 & 11 & 2 & 1 \end{pmatrix}; \quad r^{(2)} = \begin{pmatrix} 2 & 3 & 1 & 2 \\ 12 & 3 & 9 & 4 \end{pmatrix} r^{(3)} = \begin{pmatrix} 2 & 7 & 9 & 10 \\ 1 & 11 & 2 & 6 \end{pmatrix} r^{(4)} = \begin{pmatrix} 4 & 5 & 2 & 10 \\ 5 & 9 & 5 & 5 \end{pmatrix}$$

$$\alpha\mu(a)r_i + \alpha\mu(a)\mu(b)r_i = (0, 1, 1, 0)r_i + (0, 0, 0, 2)r_i$$

$$v_1 = 9 \cdot (0, 1, 1, 0) + 9 \cdot 11 \cdot (0, 0, 0, 2) = (0, 9, 9, 198)$$
$$v_2 = 2 \cdot (0, 1, 1, 0) + 2 \cdot 3 \cdot (0, 0, 0, 2) = (0, 2, 2, 12)$$
$$v_3 = 2 \cdot (0, 1, 1, 0) + 2 \cdot 11 \cdot (0, 0, 0, 2) = (0, 2, 2, 44)$$
$$v_4 = 4 \cdot (0, 1, 1, 0) + 4 \cdot 9 \cdot (0, 0, 0, 2) = (0, 4, 4, 72)$$

$$\overrightarrow{F} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 9 & 9 & 198 \\ 0 & 2 & 2 & 12 \end{pmatrix}$$

Do Schützenberger construction

$$\overrightarrow{F}\mu(\sigma) = \overrightarrow{\mu}(\sigma)\overrightarrow{F} \equiv \overrightarrow{\mu}(\sigma) = \overrightarrow{F}\mu(\sigma)\overrightarrow{F}_R^{-1}$$

$$\overrightarrow{\mu}(a) = \begin{pmatrix} 0 & \frac{-1}{24} & \frac{11}{16} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \overrightarrow{\mu}(b) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{8} & \frac{-9}{16} \\ 0 & \frac{1}{36} & \frac{-1}{8} \end{pmatrix}$$

$$\overrightarrow{\eta} = \overrightarrow{F}\eta = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 9 & 9 & 198 \\ 0 & 2 & 2 & 12 \end{pmatrix} \cdot (0, 0, 0, 1)^T = (0, 198, 12)^T$$

$$\Rightarrow \overrightarrow{\mathcal{A}} = (3, \{a, b\}, \overrightarrow{\mu}, (1, 0, 0), (0, 198, 12)^T)$$

Forward Bisimulation

Let surjective function $\mathcal{R}:\{1,\ldots,m\}\times\{1,\ldots,m\},\ n>m$ and $V\in\mathbb{K}^{n\times m}$ with $V(i,j)=1 \iff i\,R\,j$.

Let distributor matrix $W\in\mathbb{K}^{m\times n}$ with $\forall i\in\{1,\ldots m\}:\sum_{j=1}^{n}W(i,j)=1 \wedge W=(diag(w)V)^T,\ w\in\mathbb{K}^{1\times n}$.

Let weighted automata $A_1=(\alpha_1,\mu_1,\eta_1),\ A_2=(\alpha_2,\mu_2,\eta_2)$ with n and m states.

V is a forward bisimulation $\iff \alpha_1 V=\alpha_2,\ W\mu_1 V=\mu_2,\ \ W\eta_1=\eta_2,\ \ A_1\equiv A_2$

Backward Bisimulation

Similar, but is invariant wrt. weight vectors and uses V^T

V is a backward bisimulation $\iff \alpha_1=\alpha_2 V^T,\ \ V^T\mu_1=\mu_2 V^T,\ \ V^T\eta_1=\eta_2,\ \ A_1\equiv A_2$

$\Rightarrow$ ***induces equivalence relation*** $\sim$ ***with*** V ***as quotient map*** $\{1,\ldots,n\}\to\{1,\ldots,n\}\backslash\sim$

Bisimulation, partition & lumpability

- equivalence relation $\sim$ over space S induces quotient space $S\backslash\sim$
- $S\backslash\sim$ is set of equivalence classes, thus a *partition* of S

[Buchholz, 1994]

Definition 1 *Let $\underline{P}$ be the irreducible transition matrix of a finite Markov chain X on state space Z and $\Omega = \{\Omega(1)\ldots\Omega(N)\}$ a partition of the state space with collector matrix $\underline{V}$.*

- *Ω is ordinarily lumpable, iff for all $I \in \{1\ldots N\}$ and all $i, j \in \Omega(I)$: $(\underline{e}_i - \underline{e}_j)\underline{PV} = \underline{0}$*
- *Ω is exactly lumpable, iff for all $I \in \{1\ldots N\}$ and all $i, j \in \Omega(I)$: $(\underline{e}_i - \underline{e}_j)\underline{P}^T\underline{V} = \underline{0}$*
- *Ω is strictly lumpable, iff it is ordinarily and exactly lumpable.*

$\underline{e}_i$ is a row vector with 1.0 in position i and 0 elsewhere.

Kiefer vs. Tribastone

Bisimulations using change of basis wrt. prefix/postfix space for forward/backward

Depends on initial & final vector

Guaranteed to be minimal:

$$Rank(\mu) = Rank(forward) = Rank(backward)$$

Relies on randomness to find basis, sparsity?

$$\mathcal{O}\left(n^3\right) \in P$$

Bisimulation using partition refinement wrt. stoichiometry

Depends on BDE: initial value
FDE: transitions only

BDE: Species with same solutions at each step
not guaranteed to be minimal [Boreale, 2019]

FDE: Species with same dynamics wrt. all other
minimal?

$$\mathcal{O}(|R||S|\log(|S|))$$

CRN and WA semantics

Exist several CRN to WA encodings e.g.:

- [Petrov, 2018]: state=configuration, transition=reaction changing config
- [Feinberg, 1987]: state=complexes, transition=reaction changing complex

Found nothing on WA to CRN in literature

[Tribastone, 2017]: Minimizes *species*, not complexes or configurations

[Kiefer, 2013]: Requires initial and final vectors to work, CRN has none

=> No direct correspondence wrt. encoding that is required by reduction approaches

WA to CRN (wrt. partition refinement semantics)

For each state add one species

For each nz. entry in transition matrix add reaction with rate according to weight with lhs from state and rhs to state

Initial vector is not needed, for each entry in final add $S_i \to \emptyset$

Execution semantics not really preserved

$\alpha = (1,0,0,0)$

$$\mu = \left\{ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \right\}$$

Part	considered	split
$\{\{s_0, s_1, s_2, s_3\}\}$	$\{s_0, s_1, s_2, s_3\}$	$\{s_0\}$
$\{\{s_0\}, \{s_1, s_2, s_3\}\}$	$\{s_1, s_2, s_3\}$	$\{s_1, s_2\}$
$\{\{s_0\}, \{s_1, s_2\}, \{s_3\}\}$	$\{s_1, s_2\}$	—

$$\Longrightarrow \{\{s_0\}, \{s_1, s_2\}, \{s_3\}\}$$

2.

splitter

$\{\{s_0, s_1, s_2\}, \{s_3\}\}$

	$\emptyset$
s_0	0
s_1	-1
s_2	-1

$fr(s_1, \emptyset, \{s_0, s_1, s_2\}) = (-1 \cdot 1) \cdot 1$

$fr(s_2, \emptyset, \{s_0, s_1, s_2\}) = \overset{r_1 \cdot s_2}{(-1 \cdot 1) \cdot 1^{-1}}$

	$\emptyset$
s_1	
s_2	

$fr(s_1, \emptyset, \{s_1, s_2\}) =$ same as before

$fr(s_2, \emptyset, \{s_1, s_2\}) =$ — " —

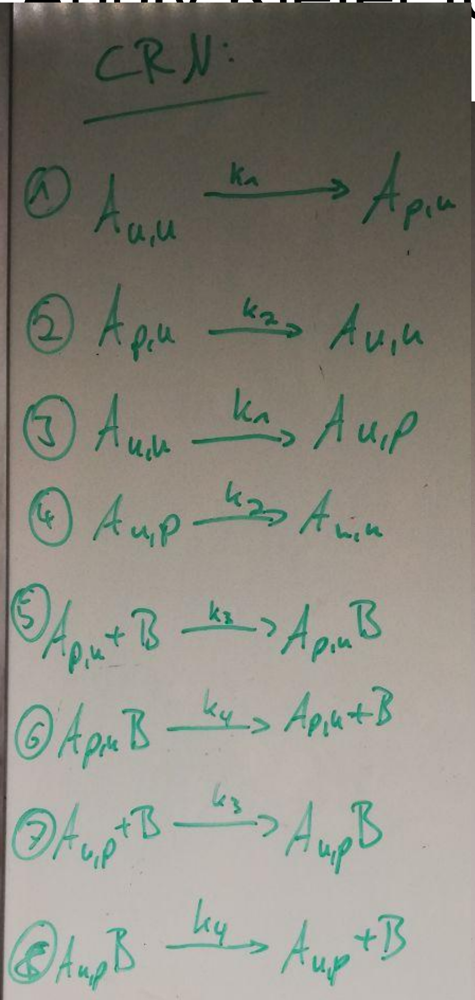
CRN to WA (semantics wrt. partition refinement)

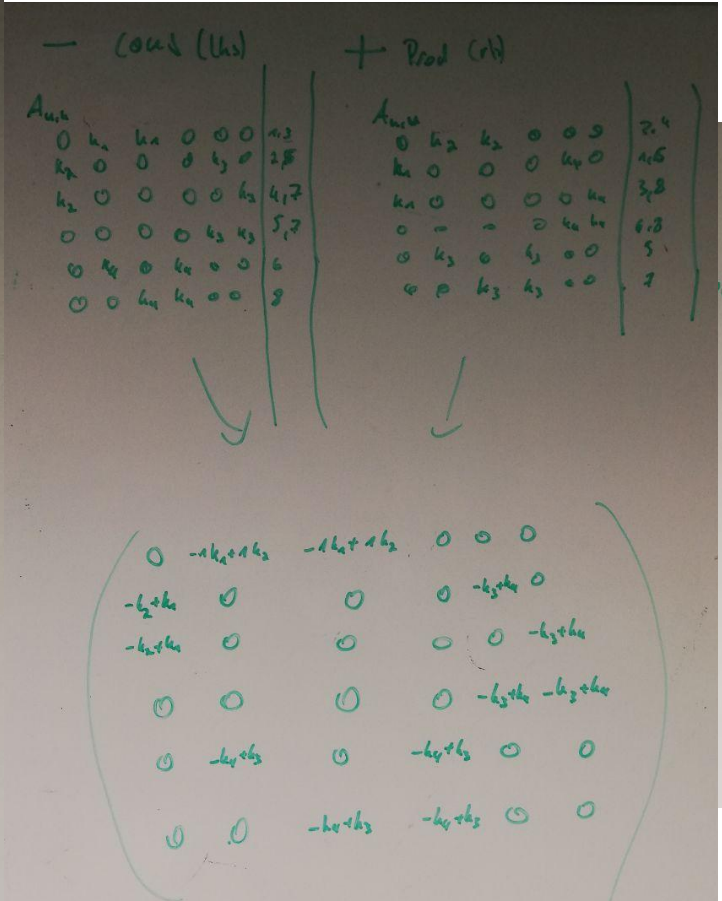
States = species

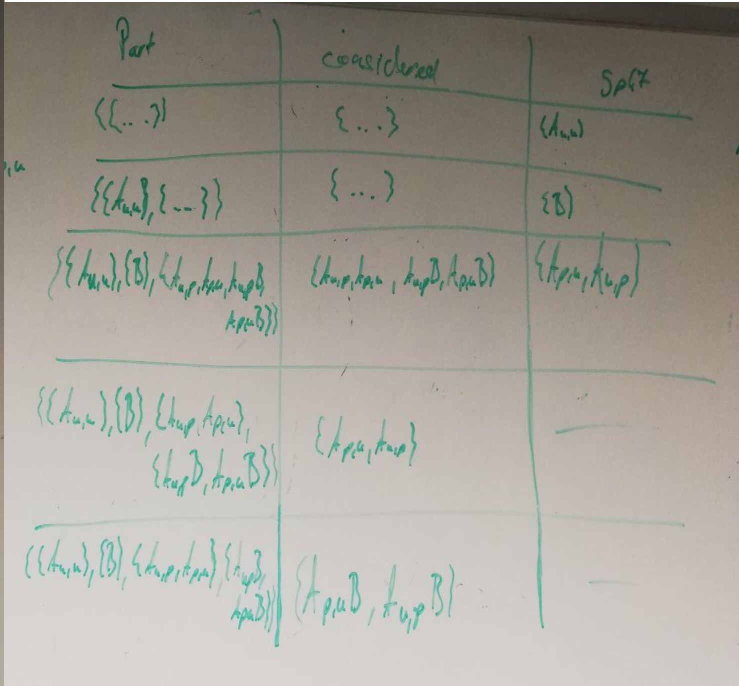
Transitions = $T \in \mathbb{R}^{|species| \times |species|}$

```
for s1 in species:
        for r in reactions:
            if (s1 in r.lhs)
                for s2 in r.rhs
                    T[s1, s2] -= s1.coeff * rate
            else if (s1 in r.rhs)
                for s2 in r.lhs
                    T[s1,s2] += s1.coeff * rate
```

Apply Kiefer to CRNs



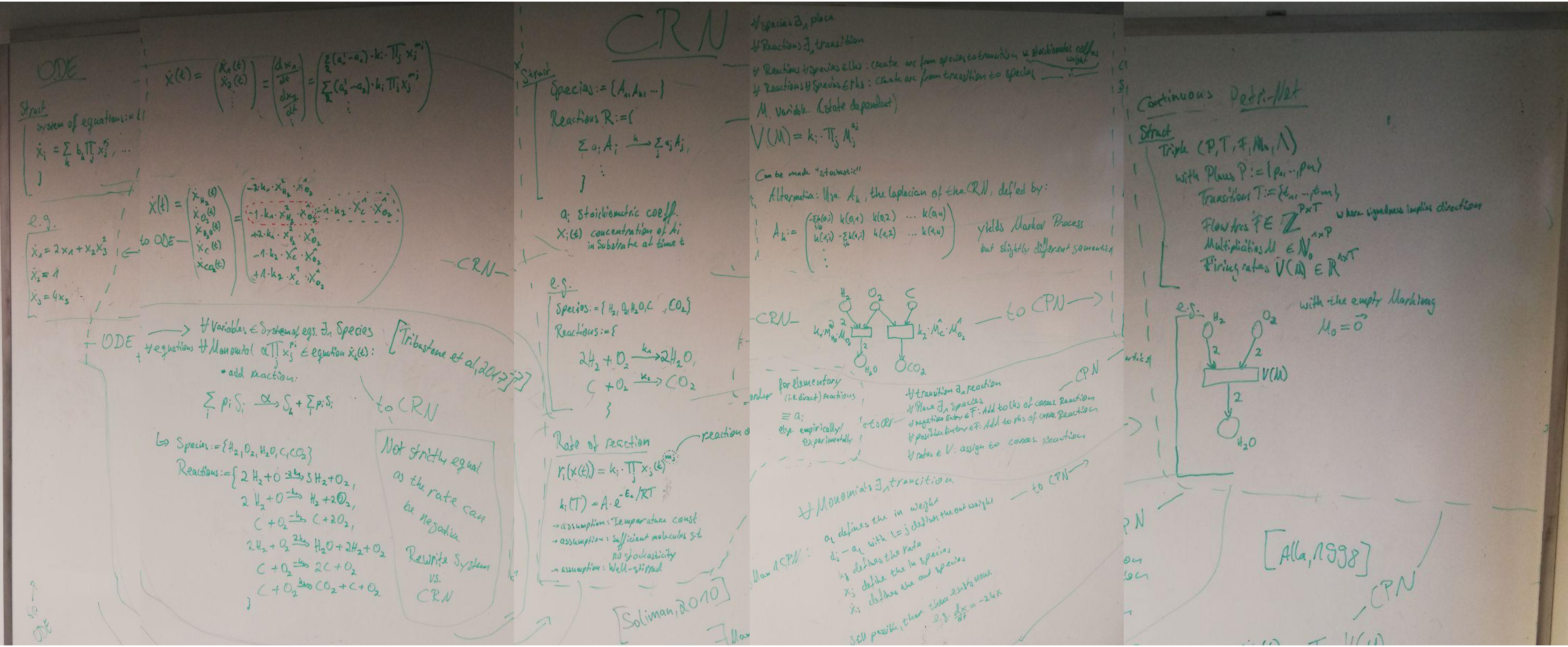




Conclusion

- At the core both algos find bisimulations, but for different conditions
- Partition refinement asymptotically faster
- Does Tribastone et al. hold for $\mathbb{Q}$-weighted automata?
- Does Hopcroft suffice and hold for the above?
- There exist connections/embeddings between frameworks
- With restrictions, semantic “problems”
- Hard to figure out encoding preserving semantics and satisfying requirements of the approaches

Extra: ODE, CRN and cont. Petri-Net



cont. Petri-Net, CTMC, Reaction Network Graph