in Symbolic Systems Biology: Theory and Methods, MS Iyengar (ed.),
Jones & Bartlett Learning, LCC, chapter 3, 61-97, 2011.

Petri Nets for Systems Biology

Monika Heiner, Robin Donaldson and David Gilbert

Abstract In this chapter we introduce qualitative, stochastic as well as continuous Petri nets and related analysis techniques in a rather informal way, and use as running example a model of the influence of the Raf Kinase Inhibitor Protein (RKIP) on the Extracellular signal Regulated Kinase (ERK) signalling pathway. We show how the qualitative and quantitative analyses complement each other, and how Petri nets can be used for step-wise modelling and analysis of biochemical networks as well as for structured design of systems of ordinary differential equations.

1 Motivation

Biochemical reaction systems have by their very nature three distinctive characteristics. (1) They are inherently bipartite, i.e. they consist of two types of game players, the species and their interactions. (2) They are inherently concurrent, i.e. several interactions can usually happen independently and in parallel. (3) They are inherently stochastic, i.e. the timing behaviour of the interactions is governed by stochastic laws. So it seems to be a natural choice to model and analyse them with a formal method, which shares exactly these distinctive characteristics: stochastic Petri nets.

Classical, i.e. time-free qualitative Petri nets combine an intuitive and executable modelling style with well-founded analysis techniques. It is for this reason that they

Monika Heiner
Department of Computer Science, Brandenburg University of Technology, Postbox 10 13 44, 03013 Cottbus, Germany, e-mail: monika.heiner@tu-cottbus.de

Robin Donaldson
Department of Computing Science, 18 Lilybank Gardens, University of Glasgow, Glasgow G12 8RZ, Scotland, UK, e-mail: radonald@brc.dcs.gla.ac.uk

David Gilbert
School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex UB8 3PH, UK, e-mail: david.gilbert@brunel.ac.uk

are widely used in various application areas, where they have been proven to be useful for a qualitative verification of technical as well as “natural” systems, i.e. biochemical networks like metabolic networks, signal transduction networks, or gene regulatory networks.

However, any real system behaviour happens in time. Thus the next step following on from qualitative analyses typically consists of quantitative analyses taking into account timing information. In the case of biochemical systems, all atomic actions, i.e. biochemical reactions, take place stochastically, which can be approximated continuously for larger quantities. Moreover, the rates of all reactions typically depend on the concentration of the involved substances. Hence systems of reaction rate equations (RREs) or ordinary differential equations (ODEs) appear well suited for quantitative modelling of biochemical networks.

In this chapter we bridge the gap between these two worlds, i.e. the (time-free) qualitative and the (timed) quantitative one, and demonstrate by means of one of the standard examples used in the systems biology community - the core model of the influence of the Raf-1 Kinase Inhibitor Protein (RKIP) on the ERK signalling pathway [CSK$^+$03] - how both sides can interact, providing different but complementary viewpoints on the same subject.

This chapter can be considered as a tutorial in step-wise modelling and analysis of biochemical networks as well as in structured design of ODEs. The qualitative model is introduced as a supplementary intermediate step, at least from the viewpoint of the biochemist accustomed to ODE modelling only, and serves mainly for model validation since this cannot be performed on the continuous level. Having successfully validated the qualitative model, the quantitative models are derived from the qualitative one by assigning rate equations to all reactions in the network. Thus the quantitative models preserve the structure of the qualitative one, and the stochastic Petri net is nothing else than a structured description of RREs, and the continuous Petri net of ODEs.

In the following we deliberately give an informal introduction avoiding formal notations as far as possible. See [HGD08] for formal definitions of the technical Petri net terms used in this chapter. All concepts are explained through the running example, introduced below.

2 Biochemical Context

There are many networks of interacting components known to exist as part of the machinery of living organisms. Biochemical networks can be regulatory, signal transduction, or metabolic networks. Regulatory networks are used to control the ways in which genes are expressed as RNAs or proteins, whereas signal transduction networks transmit biochemical signals between or within cells. The role of metabolic networks is to synthesize essential biochemical compounds from basic components, or to degrade compounds. In this chapter we focus on signal transduction, which is the mechanism which enables a cell to sense changes in its environment and to make appropriate responses. The basis of this mechanism is the conversion of one kind of signal into another.

The ERK pathway (also called Ras/Raf, or Raf-1/MEK/ERK pathway) is a ubiquitous pathway that conveys cell division and differentiation signals from the cell membrane to the nucleus. Ras is activated by an external stimulus, via one of many growth factor receptors; it then binds to and activates Raf-1 to become Raf-1* (activated Raf) which in turn activates MAPK/ERK Kinase (MEK) which in turn activates Extracellular signal Regulated Kinase (ERK). This cascade (Raf-1 → Raf-1* → MEK → ERK) of protein interaction controls cell differentiation, the effect being dependent upon the activity of ERK. An important area of experimental scientific investigation is the role that the Raf-1 Kinase Inhibitor Protein (RKIP) plays in the behaviour of this pathway. The hypothesis is that RKIP can inhibit activation of Raf-1 by binding to it, disrupting the interaction between Raf-1 and MEK, thus playing a part in regulating the activity of the ERK pathway. The subsystem we consider here, taken from [CSK$^+$03], comprises the eleven reactions summarized in Table 1.

Table 1 The reaction equations of the running example [a].

no.	precursors			products		
r_1	Raf-1*	+ RKIP	→	Raf-1*_RKIP		
r_2	Raf-1*_RKIP		→	Raf-1*	+ RKIP	
r_3	Raf-1*_RKIP	+ ERK-PP	→	Raf-1*_RKIP_ERK-PP		
r_4	Raf-1*_RKIP_ERK-PP		→	Raf-1*_RKIP	+ ERK-PP	
r_5	Raf-1*_RKIP_ERK-PP		→	Raf-1*	+ ERK	+ RKIP-P
r_6	MEK-PP	+ ERK	→	MEK-PP_ERK		
r_7	MEK-PP_ERK		→	MEK-PP	+ ERK	
r_8	MEK-PP_ERK		→	MEK-PP	+ ERK-PP	
r_9	RKIP-P	+ RP	→	RKIP-P_RP		
r_{10}	RKIP-P_RP		→	RKIP-P	+ RP	
r_{11}	RKIP-P_RP		→	RKIP	+ RP	

[a] Raf-1* is often written as Raf-1Star (if required by the tools used).

3 Qualitative Approach

In this section we introduce place/transition Petri nets, and interpret them in the standard way to model and analyse the pathway of interest.

3.1 Qualitative Modelling

Let us start with a bit of history. Petri nets, as we understand them today, have been initiated by concepts proposed by Carl Adam Petri in his Ph.D. thesis in 1962 [Pet62]. The first substantial results making up the still growing body of Petri net theory appeared around 1970. The initial textbooks devoted to Petri nets were issued in the beginning of the eighties. Since that time, Petri nets have been employed for technical and administrative systems in numerous application domains. The employment in systems biology has been first published in [RML93], [Hof94], [Red94]. In the meantime, the three biochemical network types as well as combinations of them have been investigated with different kind of Petri nets - qualitative as well as quantitative ones.

The idea to represent biochemical networks by Petri nets is rather intuitive and has been mentioned by Carl Adam Petri himself in one of his internal research reports on interpretation of net theory in the seventies. Place/transition Petri nets (or nets for short) are weighted, directed, bipartite graphs with the following four basic ingredients, compare Figure 1.

(1) There are two types of nodes, *places*, described as $S = \{s_1, \ldots, s_m\}$ and in the figures represented by circles, and *transitions*, described as $R = \{r_1, \ldots, r_n\}$ and in the figures represented by rectangles. Places usually model passive system components like conditions, species or any kind of chemical compounds, e.g. proteins or proteins complexes, playing the role of precursors or products. Transitions usually stand for active system components like atomic actions or any kind of chemical reactions, e.g. association, disassociation, phosphorylation, or dephosphorylation.

(2) The directed arcs, represented as arrows, (only) connect nodes of different type. They go from precursors to reactions (ingoing arcs), and from reactions to products (outgoing arcs). Enzymes establish side conditions and are connected in both directions with the reaction they catalyse. We define four types of sets:

- the preplaces of a transition, consisting of the reaction's precursors,
- the postplaces of a transition, consisting of the reaction's products,
- the pretransitions of a place, consisting of all reactions, producing this species,
- the posttransitions of a place, consisting of all reactions, consuming this species.

(3) Arcs are weighted by nonnegative integers (natural numbers), whereby the arc weight may be read as the multiplicity of the arc, reflecting known stoichiometries. The arc weight 1 is the default value and is usually not given explicitly.

(4) A place carries an arbitrary number of *tokens*, represented as black dots or a nonnegative integer. The number zero is the default value and usually not given explicitly. Tokens can be interpreted as the available amount of a given species in number of molecules or moles, or any abstract, i.e. discrete concentration level.

In the most abstract way, a concentration can be thought of as being "high" or "low" (present or absent); we get two levels. Generalizing this Boolean approach, any continuous concentration range can be divided into a finite number of equally sized subranges (equivalence classes), so that the concentrations within can be considered to be equivalent. The current number of tokens on a place will then specify the current level of the species' concentration, e.g. the absence of tokens specifies level 0. In the following, when speaking in terms of level semantics, we always give the highest level number.

A place is called marked, if it has at least one token, otherwise clean (unmarked). The tokens on all places constitute the *marking* of the net, which represents the current state of the system.

Having the net structure, the next step is to bring the net to life by moving the tokens through the net. We need to introduce the firing rule for the token game, which consists of two parts: the precondition and the firing itself.

- A transition is *enabled*, if all its preplaces carry at least as many tokens as required by the weights of the corresponding ingoing arcs.
- An enabled transition *may* fire, i.e. a transition is never forced to fire. The firing of a transition removes from all its preplaces as many tokens as specified by the ingoing arc weights, and adds to all its postplaces as many tokens as specified by the outgoing arc weights. The firing happens atomically and does not consume any time.

Figuratively, the firing of a transition moves tokens from its preplaces to its postplaces, while possibly changing the amount of tokens. Generally, the firing of a transition changes the formerly current marking to a new reachable one, where some transitions are not enabled anymore while others get enabled. The repeated firing of transitions establishes the behaviour of the net. The whole net behaviour consists of all possible partially ordered firing sequences (partial order semantics), or all possible totally ordered firing sequences (interleaving semantics), respectively, and the *state space* of the net of all reachable markings. In this introduction we confine ourselves to the interleaving semantics.

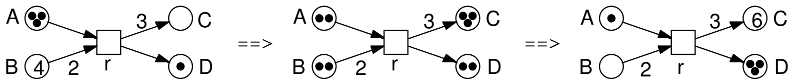
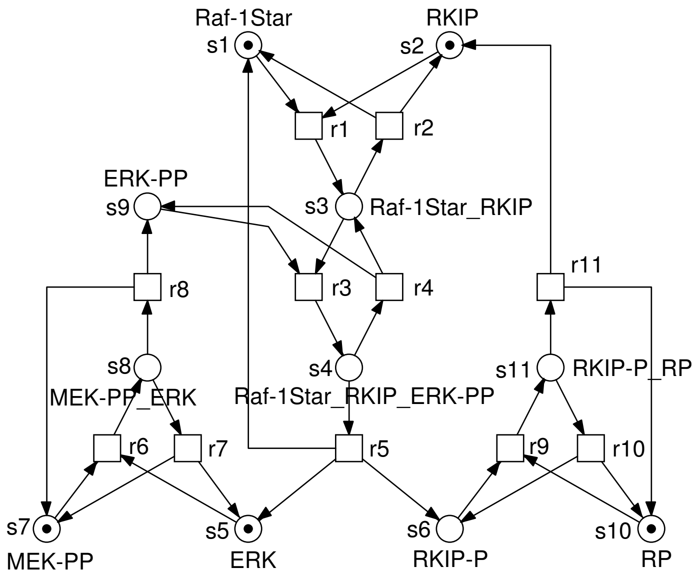


Fig. 1 The Petri net for the single reaction system $A + 2 \cdot B \rightarrow 3 \cdot C + D$ and three of its markings (states), each connected by a firing of the transition r. The transition is not enabled anymore in the marking reached after these two single firing steps.

In this modelling spirit we create a place/transition Petri net of our running example, see Figure 2, given in [CSK$^+$03] in the style of a bipartite graph. Circles (places) stand for the (local) states of a protein or protein complex and are labelled with the corresponding name; complexes are indicated by an underscore "_" between the protein names. For example, Raf-1Star and RKIP are proteins, and Raf-1Star_RKIP is a protein complex formed from Raf-1Star and RKIP. A suffix -P or -PP denotes a single or double phosphorylated protein, for example RKIP-P and ERK-PP. If appropriate, we will use the shortcuts s1, s2, ... instead of the full names. There are 11 species, and each is associated with a discrete concentration. In the most abstract way, these concentrations can be thought of as being "high" or "low" (present or absent). Later we will refine this Boolean approach to several discrete concentration levels.

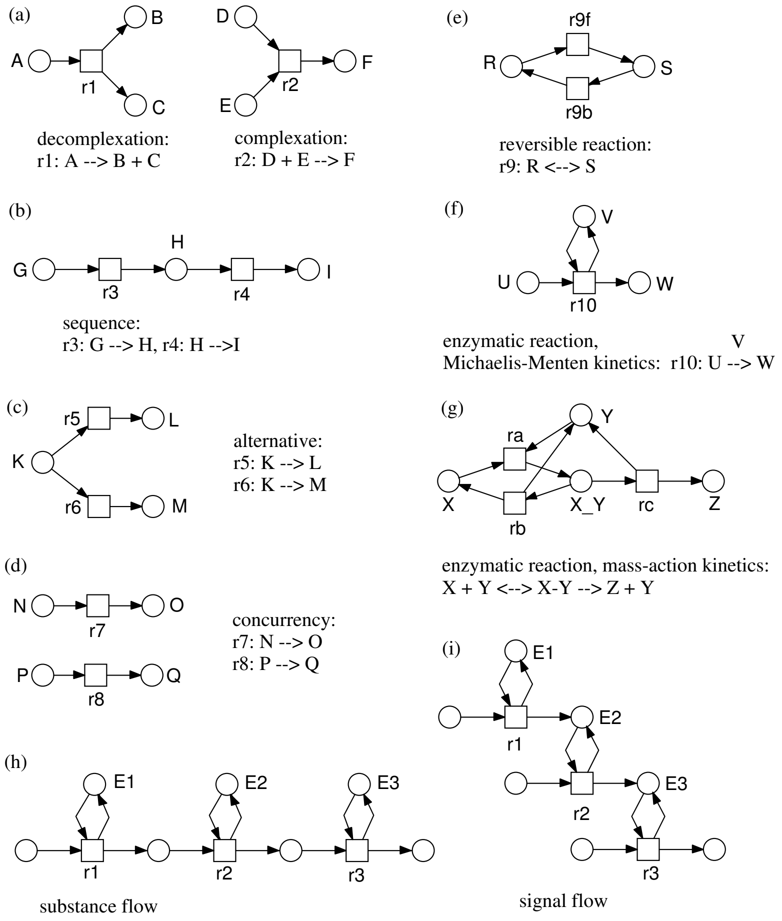


```
PUR ORD HOM NBM CSV SCF CON SC  FT0 TF0 FP0 PF0 NC
Y   Y   Y   Y   N   N   Y   Y   N   N   N   N   ES
DTP CPI CTI SCTI SB k-B 1-B DCF DSt DTr LIV REV
Y   Y   Y   N    Y  Y   Y   N   0   N   Y   Y
```

Fig. 2 The Petri net for the core model of the RKIP pathway, consisting of 11 places and 11 transitions (by chance). The places {s1, ..., s11} stand for proteins or protein complexes. Complexes are indicated by an underscore "_" between the protein names, phosphorylated forms by the suffix -P or -PP. The transitions {r1, ..., r11} model the reactions. The preplaces of a transition correspond to the reaction's precursors, and its postplaces to the reaction's products. The layout follows the suggestions by the graphical notation used in [CSK$^+$03]. The initial marking is constructed systematically using standard Petri net analysis techniques. At the bottom the two-line result vector as produced by Charlie [Cha08] is given. Properties of interest for this biochemical network in this vector are explained in the text.

Rectangles (transitions) stand for reactions, with reversible reactions being modelled by two opposite reactions. In this pathway, reactions comprise protein complexation and decomplexation events, often accompanied by phosphorylation or dephosphorylation. For example, Raf-1Star and RKIP combine in a forwards reaction to form Raf-1Star_RKIP which can disassociate in a backwards reaction into Raf-1Star and RKIP, or combine with ERK-PP to form the complex Raf1Star_RKIP_ERK-PP. Reactions are labelled with r1, r2, ... to ease referencing. In the case of reversible reactions, the labels are given by rn, rn+1 denoting that rn is the forward reaction and rn+1 the backward reaction. E.g., the forward reaction for the combination of Raf-1Star and RKIP is r1, and the disassociation of Raf-1Star_RKIP is r2.

Examples of basic structures in Petri net notation, which could help designing your own Petri net, are shown in Figure 3.



decomplexation:
r1: A --> B + C

complexation:
r2: D + E --> F

reversible reaction:
r9: R <--> S

sequence:
r3: G --> H, r4: H -->I

enzymatic reaction, V
Michaelis-Menten kinetics: r10: U --> W

alternative:
r5: K --> L
r6: K --> M

enzymatic reaction, mass-action kinetics:
X + Y <--> X-Y --> Z + Y

concurrency:
r7: N --> O
r8: P --> Q

substance flow

signal flow

Fig. 3 The Petri net components for some typical basic structures. **(a)** decomplexation and complexation; **(b)** - **(d)** the three basic structural principles: sequence, alternative, concurrency; **(e)** - **(g)** reversible reaction, enzymatic reaction in the Michaelis-Menten approach, and enzymatic reaction in the mass-action approach; **(h)**, **(i)** the essential distinctive principles of metabolic networks (substance flow) and signal transduction networks (signal flow).

3.2 Qualitative Analysis

A preliminary step will usually execute the net, which allows us to experience the model behaviour by following the token flow[1]. Having established initial confidence in the model by playing the token game, the system needs to be formally analysed. Formal analyses are exhaustive, opposite to the token game, which exemplifies the net behaviour.

The Petri net enjoys all the pleasant general properties a Petri net insider could dream of: boundedness, liveness, reversibility, which are three orthogonal basic behavioural net properties.

- *boundedness* For every place it holds that: Whatever happens, the maximal number of tokens on this place is bounded by a constant. This precludes overflow by unlimited increase of tokens.
- *liveness* For every transition it holds that: Whatever happens, it will always be possible to reach a state where this transition gets enabled. In a live net, all transitions are able to contribute to the net behaviour forever, which precludes dead states, i.e. states where none of the transitions is enabled.
- *reversibility* For every state it holds that: Whatever happens, the net will always be able to reach this state again. So the net has the capability of self-reinitialization.

In most cases these are requirable properties. The decision about the first two properties can be made for our example in a static way, i.e. without constructing the state space, while the last property requires dynamic analysis techniques, i.e. the construction of the state space. The essential steps of the systematic analysis procedure for our running example are given in more detail as follows. They represent a typical pattern how to proceed. So they may be taken as a recipe how to analyse your own system.

(1) Structural properties The following three structural properties reflect the modelling approach and can be taken as preliminary consistency checks to preclude production faults in drawing the net.

The net is *pure* (PUR), i.e. there are no place and transition connected in both directions. So, the net structure is fully represented by the incidence matrix, which is used for the calculation of the P- and T-invariants, see step (2).

The net is *ordinary* (ORD), i.e. all arc weights equal to 1. This includes homogeneity (HOM), i.e. the outgoing arcs of each place have the same multiplicity, which is a necessary prerequisite for the Deadlock Trap Property (DTP), see step (4) below.

The net is *strongly connected* (SC), i.e. there is a directed path between all pairs of nodes. This precludes boundary nodes, which exist in four types:

- input transition - a transition without preplace (FT0),
- output transition - a transition without postplace (TF0),

[1] If the reader would like to give it a try, just download our Petri net tool [Sno08].

- input place - a place without pretransition (FP0),
- output place - a place without posttransition (PF0).

Boundary nodes model interconnections of an open system with its environment. However, the given net is self-contained, i.e. a closed system. Therefore, in order to make the net live, we have to construct an initial marking, see step (3) below.

Finally, there are transitions sharing a preplace; e.g. the transitions r2 and r3 share the preplace Raf-1Star_RKIP. Consequently, these transitions compete for the tokens on the shared preplace. Such a situation is called *structural conflict*. If each transition is involved in one conflict at most, the net belongs to the structural class *extended simple* (ES), which is the case for our running example. Hence, we know that the net has the ability to be live independent of time, i.e. if it is live, then it remains live under any timing. Remarkably, neglecting the backward directions of all reversible reactions removes all structural conflicts in the net structure, which makes this reduced net to a *synchronisation graph* (SG), the structural net class, which is characterized by being free of structural conflicts (SCF).

(2) Static decision of marking-independent behavioural properties A beneficial technique in model validation is to check all invariants for their biological plausibility. We shortly recall the essential technical terms.

At the beginning, we need to represent the net structure by a matrix, which opens the door to analysis techniques based on linear algebra. The *incidence matrix* of a Petri net is an integer matrix $\mathbb{C}$ with a row for each place and a column for each transition. A matrix entry $\mathbb{C}(s,r)$ gives the token change on place s by the firing of transition r. Thus, a preplace of r, which is not a postplace of r, has a negative entry, while a postplace of r, which is not a preplace of r, has a positive entry, each corresponding to the arc multiplicities. The entry for a place, which is preplace as well as postplace of a transition, gives the difference of the multiplicities of the transition's outgoing arc minus the transition's ingoing arc. In this case we lose information; the non-ordinary net structure can not be reconstructed uniquely out of the incidence matrix.

The columns of $\mathbb{C}$ are place vectors, i.e. vectors with as many entries as there are places, describing the token change on a marking by the firing of the transition defining the column index. The rows of $\mathbb{C}$ are transition vectors, i.e. vectors with as many entries as there are transitions, describing the influence of all transitions on the tokens in the place, defining the row index. For stoichiometric reaction networks, e.g. metabolic networks, the incidence matrix coincides with the stoichiometric matrix.

A nonzero and nonnegative integer place vector x is called *P-invariant*, if $x \cdot \mathbb{C} = 0$; in words, for each transition it holds that: multiplying the P-invariant with the transition's column vector yields zero.

Thus, the total effect of each transition on the P-invariant is zero, which explains its interpretation as a token conservation component. A P-invariant x stands for a set of places over which the weighted sum of tokens is constant and independent of any firing, i.e. for any markings m_1, m_2, which are reachable by the firing of transitions, it holds that $x \cdot m_1 = x \cdot m_2$. In the context of metabolic networks, P-invariants reflect substrate conservations, while in signal transduction networks P-invariants often correspond to the several states of a given species (protein or protein complex). A place belonging to a P-invariant is obviously bounded.

Likewise, a nonzero and nonnegative integer transition vector y is called *T-invariant*, if $\mathbb{C} \cdot y = 0$; in words, for each place it holds that: multiplying the place's row with the T-invariant yields zero. Thus, the total effect of the T-invariant on a marking is zero. A T-invariant has two interpretations in the given biochemical context.

- The entries of a T-invariant represent a multiset of transitions which by their partially ordered firing reproduce a given marking, i.e. basically occurring one after the other. This partial order sequence of the T-invariant's transitions may contribute to a deeper understanding of the net behaviour. A T-invariant is called *feasible*, if such a behaviour is actually possible in the given marking situation.
- The entries of a T-invariant may also be read as the relative firing rates of transitions, all of them occurring permanently and concurrently. This activity level corresponds to the steady state behaviour.

The two transitions modelling the two directions of a reversible reaction make always a minimal T-invariant; thus they are called *trivial T-invariants*.

The computation of all P-invariants (T-invariants) requires solving the homogenous linear equation system $x \cdot \mathbb{C} = 0$ ($\mathbb{C} \cdot y = 0$) over the nonnegative integers. If there are solutions, then there are infinitely many. To characterize the solution space, we need a generating system. For this purpose we introduce the next terms, which hold equally for P- and T-invariants.

The set of nodes corresponding to an invariant's nonzero entries is called its *support*. An invariant x is called *minimal*, if its support does not contain the support of any other invariant, and the greatest common divisor of all nonzero entries of x is 1. The set of all minimal invariants of a given net is unique and is a generating system for all invariants. To compute all invariants out of the minimal ones, we use the following operations: multiplication with a nonnegative integer, addition of invariants, and division by a common divisor.

A minimal P-invariant (T-invariant) defines a connected subnet, consisting of its support, its pre- and posttransitions (pre- and postplaces), and all arcs in between. These minimal self-contained subnets may be read as a decomposition into token preserving or state repeating modules, which should have an enclosed biological meaning. However, minimal invariants generally overlap, and in the worst-case there are exponentially many of them.

A net is covered by P-invariants, shortly CPI, (covered by T-invariants, shortly CTI), if every place (transition) belongs to a P-invariant (T-invariant). CPI causes structural boundedness (SB), i.e. boundedness for any initial marking. CTI is a necessary condition for bounded nets to be live.

The net under consideration has five minimal P-invariants covering the net (CPI), consequently the net is structurally bounded (SB). All the P-invariants x_i contain only entries of 0 and 1, which allows a short-hand specification by just giving the names of the places involved.

$x_1 =$ (**Raf-1Star**, Raf-1Star_RKIP, Raf-1Star_RKIP_ERK-PP),

$x_2 =$ (**MEK-PP**, MEK-PP_ERK),
$x_3 =$ (**RP**, RKIP-P_RP),
$x_4 =$ (**ERK**, ERK-PP, MEK-PP_ERK, Raf-1Star_RKIP_ERK-PP),
$x_5 =$ (**RKIP**, Raf-1Star_RKIP, Raf-1Star_RKIP_ERK-PP, RKIP-P_RP, RKIP-P).

Each P-invariant x_i stands for a reasonable conservation rule, the species preserved being given in bold by the first name in the invariant. Due to the chosen naming convention, this particular name also appears in all the other place names of the same P-invariant.

The net under consideration is also covered by T-Invariants (CTI). Besides the expected four trivial T-invariants for the four reversible reactions, there is only one non-trivial minimal T-invariant $y =$ (r1, r3, r5, r6, r8, r9, r11), comprising the whole net, but without the backward reactions. The subnet defined by this T-invariant describes the essential partial order behavior of the modelled system.

The automatic identification of non-trivial minimal T-invariants is in general useful as a method to highlight important parts of a network, and hence aid its comprehension by biochemists, especially when the entire network is too complex to easily comprehend.

Notably, our running example collapses into a synchronisation graph, if we remove the backward directions of the reversible reactions. In a strongly connected ordinary synchronisation graph, each minimal cycle corresponds to a minimal P-invariant, and the set of all transitions constitutes a T-invariant. This confirms from a different perspective the invariants results, which we have just discussed.

All the properties above relate only to the structure, i.e. they are valid independently of the initial marking. In order to proceed we first need to generate an initial marking.

(3) Initial marking construction For a systematic construction of the initial marking, we consider the following criteria.

- Each P-invariant needs at least one token.
- All (non-trivial) T-invariants should be feasible, meaning, the transitions, making up the T-invariant's multiset can actually fire in an appropriate order.
- Additionally, it is common sense to look for a minimal marking (as few tokens as possible), which guarantees the required behaviour.
- Within a P-invariant, choose the species with the most *inactive* or the *monomeric* state.

Taking all these criteria together, the initial marking on hand is: Raf-1Star, RKIP, ERK, MEK-PP, RP get each one token, while all remaining places are clean. With this initial marking, the net is covered by 1-P-invariants (exactly one token in each P-invariant), therefore the net is 1-bounded (indicated as 1-B in the analysis result vector, compare Figure 2). That is in perfect accordance with the understanding that in signal transduction networks a P-invariant comprises all the different states of one species. Obviously, each species can be only in one state at any time.

This initial marking differs from the initial concentrations used in [CSK$^+$03] as part of their method to estimate rate parameters required for their ODE model of the

RKIP pathway. We will later see that their initial marking is equivalent to the initial marking which we have constructed, and that in fact both markings are members of a larger equivalence class of markings.

With the chosen marking we can check the non-trivial minimal T-invariant, see step (2), for realizability, which then involves the realizability of the trivial T-invariants. We obtain an infinite partial order run, the beginning of which is given as labelled condition/event net in Figure 4. We get this run by unfolding the behaviour of the subnet induced by the T-invariant, whereby any concurrency is preserved. Here, transitions represent events, labelled by the name of the reaction taking place, while places stand for binary conditions, labelled by the name of the species, set or reset by the event, respectively. This run can be characterized in a short-hand notation by the following set of partially ordered words out of the alphabet of all transition labels R:

{ (r1, (r6; r8)); [(r3; r5);((r9; r11; r1), (r6; r8))]* },

where ";" stands for "sequentiality" and "," for "concurrency". This partial order behaviour gives further insight into the dynamic behaviour of the network which may not be apparent from the standard net representation. In our running example, the partial order representation illustrates the central role of Raf-1Star and the fact that certain reactions take place sequentially, others can take place independently and that some reactions are constrained by the need for the presence of two or more precursors produced from prior reactions; e.g. the reaction r1 requires the participation of both proteins Raf-1* and RKIP, which are produced by reactions r5 and r11 respectively.

Having established and justified our initial marking we proceed to the next steps of the analysis.

(4) Static decision of marking-dependent behavioural properties The following advanced structural Petri net properties can be decided by combinatorial algorithms. First, we need to introduce two new notions.

A nonempty set of places $D \subseteq S$ is called *structural deadlock* if every transition, which fires tokens onto a place in this structural deadlock set also has a preplace in this set, i.e. the set of pretransitions of D is contained in the set of posttransitions of D. Pretransitions of a structural deadlock cannot fire, if the place set is clean. Therefore, a structural deadlock cannot get tokens again, as soon as it is clean, and then all its posttransitions are dead.

A nonempty set of places $Q \subseteq S$ is called *trap*, if every transition, which subtracts tokens from a place of the trap set, also has a postplace in this set, i.e. the set of posttransitions of Q is contained in the set of pretransitions of Q. Posttransitions of a trap always return tokens to the place set. Therefore, once a trap contains tokens, it cannot become clean again.

Structural deadlock and trap are closely related, but contrasting notions. When they come on their own, we get usually deficient behaviour. However, both notions have the power to complement each other perfectly. A Petri net has the *Deadlock-Trap Property (DTP)*, if every structural deadlock contains a marked trap. The DTP

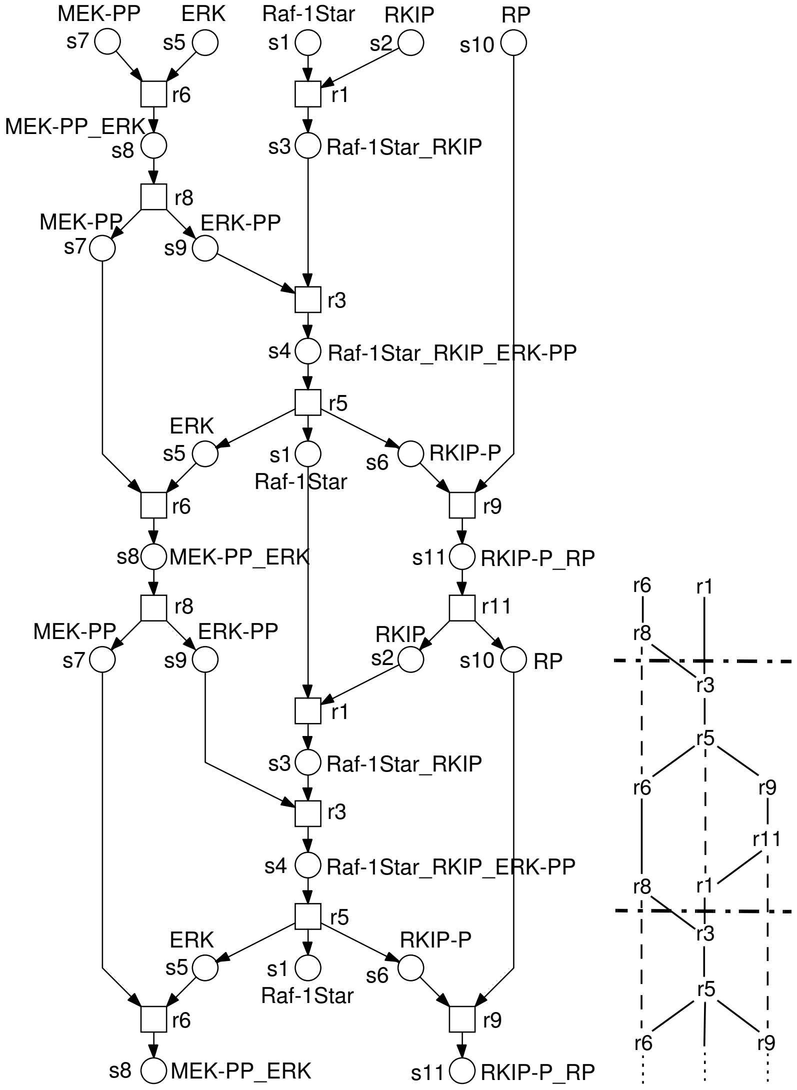


Fig. 4 The beginning of the infinite partial order run of the non-trivial minimal T-invariant $y =$ (r1, r3, r5, r6, r8, r9, r11) of the place/transition Petri net given in Figure 2. We get this run by unfolding the behaviour of the subnet induced by the T-invariant, whereby any concurrency is preserved. Here, transitions represent events, labelled by the name of the reaction taking place, while places stand for binary conditions, labelled by the name of the species, set or reset by the event, respectively. On the right, a shorthand notation is given, showing also the cuts between two "rounds".

can still be decided by structural reasoning only. The importance becomes clear by the following theorem: An ordinary extended simple net having the DTP is live.

Our running example is extended simple and the DTP holds, therefore the net is live (LIV). So, we are in the fortunate position to be able to decide liveness by structural reasoning only - the state space does not need to be constructed for this.

(5) Dynamic decision of behavioural properties In order to decide reversibility (REV) we need to construct the state space. This could be done according the partial order semantics or the interleaving semantics. To keep things simple we consider here the reachability graph only. The nodes of a reachability graph represent all possible states (markings) of the net. The arcs in between are labelled by single transitions, the firing of which causes the related state change. Altogether, the reachability graph gives us a finite automaton representation of all possible single step firing sequences. Consequently, concurrent behaviour is described by enumerating all interleaving firing sequences; so it reflects the behaviour of the net in the interleaving semantics.

Because we already know that the net under consideration is bounded, we also know that the reachability graph has to be finite. Here, the reachability graph has 13 states (out of 2048 = 2^{11} theoretically possible ones, due to the given 1-boundedness), see Figure 5 and Table 3. Generally, reachability graphs tend to be huge. Table 2 shows the size of the reachability graph with different number of concentration levels.

Table 2 The number of states in the state space as the number of levels in the model increases.

# Levels	# States	# Levels	# States	# Levels	# States
1	13	15	368,220	40	79,414,335
5	1,974	20	1,696,618	50	2.834 e+08
7	8,646	25	5,723,991	100	1.591 e+13
9	28,171	30	15,721,464	250	3.582 e+12
10	47,047	35	37,314,537	500	2.231 e+14

It is clear from the table that as the number of levels increase, the number of states in the reachability graph grows extremely fast. The term *state explosion* has been coined for this phenomenon. Independently of the size, the reachability graph we get is strongly connected. Therefore, the Petri net is reversible, i.e. the initial system state is always reachable again, and the system has the capability of self-reinitialization. Further, the liveness of the net has already been decided structurally, so we know that each transition (reaction) appears at least once as arc label in this reachability graph.

Moreover, from the viewpoint of the qualitative model, all these 13 states are equivalent, i.e. any of those 13 states could be taken as initial state resulting in exactly the same total (discrete) system behaviour (likewise for the larger state sets). That is in perfect accordance with the observations gained during quantitative analyses, see next section.

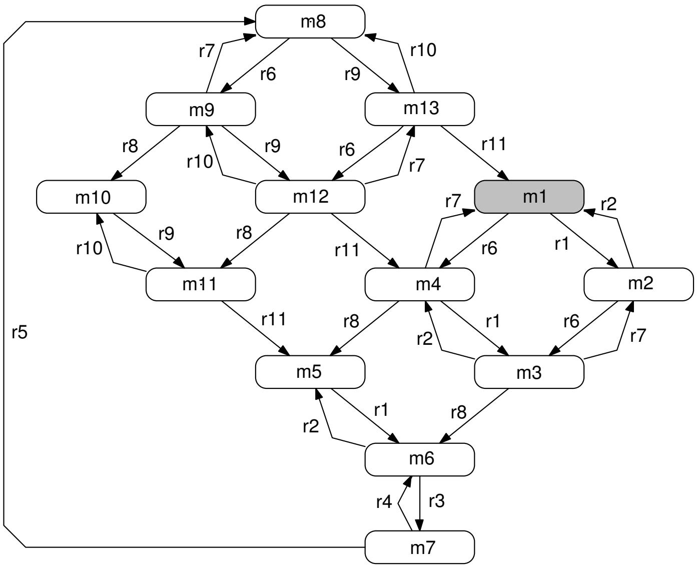


Fig. 5 The reachability graph of the place/transition Petri net given in Figure 2. Nodes represent states (markings), while the arcs are labelled with the transition responsible for a given change of states. For a list of the complete state descriptions see Table 3. The initial state is highlighted in gray. In the reachability graph, the two concurrent transition sequences (r6; r8) and (r9; r11; r1), compare the partial order run in Figure 4, going from m8 to m5, are represented by all possible totally ordered (interleaving) sequences. Thus, the reachability graph reflects the interleaving semantics.

This concludes the analysis of *general* behavioural net properties, i.e. of properties we can speak about in syntactic terms only, without any semantic knowledge. The next step consists in a closer look at *special* behavioural net properties, reflecting the expected special functionality of the network.

(6) Model checking of special behavioural properties To validate the model, it is often of interest to prove - besides the general properties - additional special properties which reflect the intended functionality of the network. We have to specify these special properties in an unambiguous language. Temporal logics, a mathematical mechanism, has been proven to be best suited for this purpose. It provides a flexible formalism which considers the validity of propositions in relation to the execution of the model.

One of the widely used temporal logics is the *Computational Tree Logic* (CTL). It is called after the data structure used - the computational tree, which we get by unwinding the reachability graph. Thus, CTL represents a branching time logic with interleaving semantics. The analysis technique, deciding whether a temporal-logic property holds in a model, is called *model checking,* and the tools implementing the algorithms are called *model checkers*. Model checking generally requires boundedness. If the net is 1-bounded, there exists a particularly rich choice of model check-

Table 3 States of the reachability graph given in Figure 5, which will be used as the initial 13 ‘good’ state configurations in the quantitative analyses.

#	species	m1	m2	m3	m4	m5	m6	m7	m8	m9	m10	m11	m12	m13
s1	Raf-1*	1	0	0	1	1	0	0	1	1	1	1	1	1
s2	RKIP	1	0	0	1	1	0	0	0	0	0	0	0	0
s3	Raf-1*_RKIP	0	1	1	0	0	1	0	0	0	0	0	0	0
s4	Raf-1*_RKIP_ERK-PP	0	0	0	0	0	0	1	0	0	0	0	0	0
s5	ERK	1	1	0	0	0	0	0	1	0	0	0	0	1
s6	RKIP-P	0	0	0	0	0	0	0	1	1	1	0	0	0
s7	MEK-PP	1	1	0	0	1	1	1	1	0	1	1	0	1
s8	MEK-PP_ERK	0	0	1	1	0	0	0	0	1	0	0	1	0
s9	ERK-PP	0	0	0	0	1	1	0	0	0	1	1	0	0
s10	RP	1	1	1	1	1	1	1	1	1	1	0	0	0
s11	RKIP-P_RP	0	0	0	0	0	0	0	0	0	0	1	1	1

ers, [SSE03], which get their efficiency by exploiting sophisticated data structures and algorithms. In the case of our rather simple example, the variety of model checkers is not important, and the properties could even be checked manually. However, as mentioned above, the state space of more complex networks is subject to the famous state explosion problem, and therefore calls for automatic evaluation.

The application of this analysis approach requires an understanding of temporal logics. CTL - as any temporal logic - is an extension of a classical (propositional) logic. The atomic propositions consist of statements on the current token situation in a given place. In the case of 1-bounded models, places can be read as Boolean variables, which allows propositions such as RKIP-P instead of $m(\text{RKIP-P}) = 1$, where $m(s)$ yields the number of tokens on s. Likewise, places are read as integer variables for bounded models.

Atomic propositions can be combined to composed propositions using the standard logical operators: $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), and $\rightarrow$ (implication), e.g. RKIP-P $\vee$ RKIP-P_RP.

The truth value of a proposition may change by the execution of the net; e.g. the proposition RKIP does hold in state m1, but not in state m2. Such temporal relations between propositions are expressed by the additionally available temporal operators. In CTL there are basically four of them (ne**X**t, **F**inally, **G**lobally, **U**ntil), which come in two versions (**E** for Existence, **A** for All), making together eight operators. Let $\phi_{[1,2]}$ be an arbitrary temporal-logic formulae. Then, the following formulae hold in state m,

- **EX** ϕ : if there is a state reachable by one step where ϕ holds.
- **EF** ϕ : if there is a path where ϕ holds finally, i.e., at some point.
- **EG** ϕ : if there is a path where ϕ holds globally, i.e., forever.
- **E** $(\phi_1$ **U** $\phi_2)$: if there is a path where ϕ_1 holds until ϕ_2 holds.

The other four operators, which we get by replacing the **E**xistence operator by the **A**ll operator, are defined likewise by extending the requirement *"there is a path"* to *"for all paths"*. A formula holds in a net, if it holds in its initial state.

We instantiate some of the generic property patterns provided in [CRCVDS04] and get the following samples of meaningful statements for our running example, whose truth can be determined via model checking. Please remember, places are interpreted as Boolean variables in the following formulae, in order to simplify notation.

- **property Q1:** There is a reachable state where ERK is phosphorylated and RKIP is not phosphorylated.
$$\begin{aligned}&\mathbf{EF}\ [\ (\text{ERK-PP} \vee \text{Raf-1Star\_RKIP\_ERK-PP}) \wedge \\ &\qquad (\text{RKIP} \vee \text{Raf-1Star\_RKIP} \vee \text{Raf-1Star\_RKIP\_ERK-PP}]\end{aligned}$$
The given state specification applies to state m5, and this state is of course reachable from m1; the shortest firing sequence is r6; r8, compare Figure 5.
- **property Q2:** The phosphorylation of ERK (to ERK-PP) is independent of the phosphorylated state of RKIP.
$$\begin{aligned}&\mathbf{AG}\ [\ (\ \text{ERK} \wedge \neg(\text{RKIP-P} \vee \text{RKIP-P\_RP})\ ) \rightarrow \\ &\qquad\quad \mathbf{E}\ [\ \neg(\ \text{RKIP-P} \vee \text{RKIP-P\_RP})\ \mathbf{U}\ \text{ERK-PP}\ ]\ \ ]\end{aligned}$$
In words: forever it holds (AG), if we are in a state where ERK is marked, but neither RKIP-P nor RKIP-P_RP, i.e. RKIP is not phosphorylated, then there is a path (E), along which in all states RKIP is not phosphorylated until (U) we reach a state were ERK-PP is marked.
- **property Q3:** A cyclic behaviour w.r.t. the presence/absence of RKIP is possible forever.
$$\mathbf{AG}\ [\ (\ \text{RKIP}\rightarrow \mathbf{EF}\ (\neg\text{RKIP})\ ) \wedge (\neg\text{RKIP}\rightarrow \mathbf{EF}\ (\text{RKIP})\ )\ ]$$
In words: forever it holds (AG), if we are in a state where RKIP is marked, then there is a path to a state (EF), where RKIP is not marked anymore, and vice versa. Taking into consideration that the backward reactions are comparatively slow, compare Table 4, which allows us to neglect them, we are able to expect a much stronger property: the cyclic behaviour w.r.t. the presence/absence of RKIP is not only possible forever, it is even guaranteed.
$$\mathbf{AG}\ [\ (\ \text{RKIP}\rightarrow \mathbf{AF}\ (\neg\text{RKIP})\ ) \wedge (\neg\text{RKIP}\rightarrow \mathbf{AF}\ (\text{RKIP})\ )\ ]$$
In words, forever it holds (AG), if we are in a state, where RKIP is marked, then on all paths we reach finally a state (AF), where RKIP is not marked anymore, and vice versa. This stronger property does not hold in the full net (with the backward reactions), because the reversible reactions establish infinite cycles, i.e. paths, which prevent the system from reaching a state as required.

Temporal logic is an extremely powerful and flexible language to describe special properties, however needs some experience to get accustomed to it. Applying this analysis technique requires seasoned understanding of the network under investigation combined with the skill to correctly express the expected behaviour in temporal logics. We will see in the next section how to employ the same technique in a quantitative setting.

3.3 Summary

To summarize the preceding validation steps, the model has passed the following validation criteria.

- **validation criterion 0** All expected structural properties hold, and all expected general behavioural properties hold.
- **validation criterion 1** The net is CPI, and there are no minimal P-invariant without biological interpretation.
- **validation criterion 2** The net is CTI, and there are no minimal T-invariant without biological interpretation. Most importantly, there is no known biological behaviour without a corresponding, not necessarily minimal, T-invariant.
- **validation criterion 3** All expected special behavioural properties expressed as temporal-logic formulae hold.

One of the benefits of using the qualitative approach is that systems can be modelled and analysed without any quantitative parameters. Moreover the qualitative analyses help in identifying suitable initial markings and potential quantitative analysis techniques, compare next section.

Now we are ready for a more sophisticated quantitative analysis of our model. Please note, up to now we employed a time-free model. However, the qualitative analysis techniques consider all behaviour possible under any timing. Adding timing constraints to our Petri net usually restricts the behaviour. So it should not come as a surprise if under the given timing assumptions certain behaviour disappears. However we know that what has not been possible in the qualitative model cannot become possible in the quantitative one; e.g. a bounded model cannot become unbounded.

4 Quantitative Approaches

In this section we transform our validated qualitative model, given as a place/transition Petri net, into quantitative ones, specified as stochastic or continuous Petri nets, and employ related analysis techniques.

4.1 Quantitative Modelling

To transform a qualitative Petri net into quantitative ones, we need to assign to all transitions (reactions) their rate functions, which generally depend on the current state of the reactions' substrates, or - in Petri net terms - the current marking of the transitions' preplaces. Technically, rate functions can be any mathematical functions. However, often they follow some kinetic patterns, with the mass-action kinetics and the Michaelis-Menten kinetics being the most famous ones. Table 4 gives for each reaction: the reaction equation, the rate function and the involved rate constants. The rate functions (employing the stochastic rate constants) are used in the stochastic model as the propensity (hazard) functions, determining the current stochastic firing rates, and in the continuous model as the deterministic rate functions (employing the deterministic rate constants), determining the current deterministic firing rates. The conversion of stochastic and deterministic rate constants into each other is well understood, see e.g. [Wil06]; especially it holds that they are equivalent for first-order reactions. For second-order reactions we get the stochastic rate constant by multiplying with a scaling factor (level semantics).

Table 4 The reaction equation, rate function, and rate constants for each reaction (transition), compare Figure 2. For better readability we use the abbreviations s1 ... s11 for the involved species. All reactions employ mass action kinetics. Please note that the backward reactions constants are by two orders of magnitude smaller than for the forward reactions.

#	reaction	equation		rate function v_i	rate constant [a]	
					stochastic	deterministic
r1	s1 + s2	$\rightarrow$	s3	$c_1 \cdot$ s1 $\cdot$ s2	$c'_1 = c_1 \cdot f_s$	$c_1 = 0.53$
r2	s3	$\rightarrow$	s1 + s2	$c_2 \cdot$ s3	$c'_2 = c_2$	$c_2 = 0.0072$
r3	s3 + s9	$\rightarrow$	s4	$c_3 \cdot$ s3 $\cdot$ s9	$c'_3 = c_3 \cdot f_s$	$c_3 = 0.625$
r4	s4	$\rightarrow$	s3 + s9	$c_4 \cdot$ s4	$c'_4 = c_4$	$c_4 = 0.00245$
r5	s4	$\rightarrow$	s1 + s5 + s6	$c_5 \cdot$ s4	$c'_5 = c_5$	$c_5 = 0.0315$
r6	s5 + s7	$\rightarrow$	s8	$c_6 \cdot$ s5 $\cdot$ s7	$c'_6 = c_6 \cdot f_s$	$c_6 = 0.8$
r7	s8	$\rightarrow$	s5 + s7	$c_7 \cdot$ s8	$c'_7 = c_7$	$c_7 = 0.0075$
r8	s8	$\rightarrow$	s7 + s9	$c_8 \cdot$ s8	$c'_8 = c_8$	$c_8 = 0.071$
r9	s6 + s10	$\rightarrow$	s11	$c_9 \cdot$ s6 $\cdot$ s10	$c'_9 = c_9 \cdot f_s$	$c_9 = 0.92$
r10	s11	$\rightarrow$	s6 + s10	$c_{10} \cdot$ s11	$c'_{10} = c_{10}$	$c_{10} = 0.00122$
r11	s11	$\rightarrow$	s2 + s10	$c_{11} \cdot$ s11	$c'_{11} = c_{11}$	$c_{11} = 0.87$

[a] The stochastic and deterministic rate constants are equivalent for first-order reactions. f_s is a scaling factor to map the given *mass* in the continuous concentration onto a finite number of levels (i.e tokens), with N being the highest level number, i.e. $f_s = mass/N$.

Finding the rate constants may turn out to be a difficult and time consuming process. It usually involves both searches for scientific papers and also discussions with the biologists knowledgeable in the pathway under investigation. We obtain the deterministic rate constants from [CSK$^+$03].

In the following we illustrate the strength of quantitative approaches by a few analysis examples only. We start with the stochastic approach to exclude eccentric system behaviour caused by stochastic noise before considering the averaged behaviour in the deterministic continuous approach.

4.2 Stochastic Analysis

As with a qualitative Petri net, a stochastic Petri net maintains a discrete number of tokens on its places. But contrary to the time-free case, an exponentially distributed firing rate (waiting time) is associated with each transition r, specified by a parameter λ. Generally, λ is state-dependent and defined by a propensity (hazard) function. The firing itself does not consume time and again follows the standard firing rule of qualitative Petri nets. Thus, all waiting times can theoretically still occur, but the likelihood depends on the probability distribution. Consequently, the system behaviour is described by the same discrete state space, and all the different execution runs of the underlying qualitative Petri net can still take place.

More formally, the semantics of a stochastic Petri net (with exponentially distributed waiting times for all transitions) is described by a continuous time Markov chain (CTMC). The CTMC of a stochastic Petri net without parallel transitions is isomorphic to the reachability graph of the underlying qualitative Petri net, while the arcs between the states are now labelled by the transition rates. This allows the use of the same powerful analysis techniques for stochastic Petri nets as already applied for qualitative Petri nets.

Table 4 provides the details which permit reading the net in Figure 2 as a stochastic Petri net, specifying at the same time reaction rate equations (RREs).

(1) Equivalence check by transient analysis Due to the isomorphy of reachability graph and CTMC, all qualitative analysis results obtained in the former section are still valid. The influence of time does not restrict the possible system behaviour. Specifically it holds that the CTMC in our case study is reversible, which ensures ergodicity; i.e. we could start the system in any of the reachable states, always resulting in the same CTMC with the same steady state probability distribution.

Additionally, probabilistic analyses of the transient and steady state behaviour are now available. Generally, this can be done in an analytic as well as in a simulative manner. The boundedness of the underlying qualitative model allows the application of standard Markov analysis techniques which are based on the state transition matrix. However, stochastic simulation algorithms (SSA), e.g. Gillespie's exact simulation algorithm [Gil77], are computationally less expensive and work also for huge or even infinite state spaces.

Applying SSA produces data describing the dynamic evolution of the biological system over time. Figure 6 shows the behaviour of a subset of the species in the model (phosphorylated proteins or complexes of them) over time, for various numbers of levels (tokens). The noise decreases as the number of levels increases, thus the behaviour of the stochastic Petri net approaches that of the deterministic net. Moreover it is obvious that the system reaches a steady state in all shown cases, determining the value at which the response saturates.

We proceed with transient analysis to prove the sufficient equivalence between the stochastic model in the level semantics and the corresponding continuous model, justifying the interpretation of the properties gained by the stochastic model also in terms of the continuous one. Probabilistic model checkers such as PRISM [PNK06]

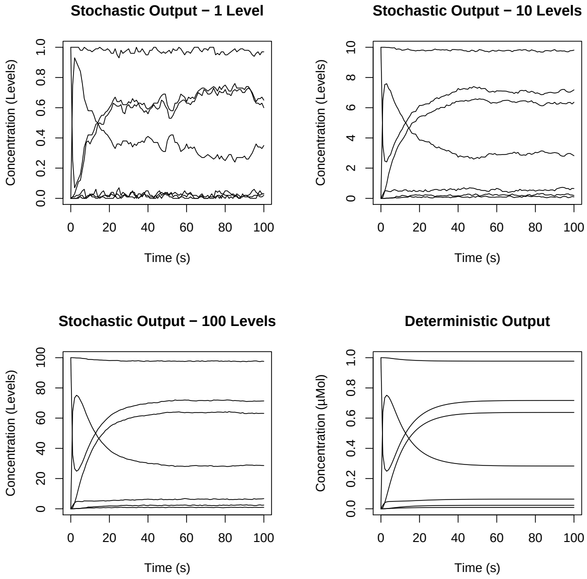


Fig. 6 Diagrams displaying the output of deterministic (ODE) and stochastic (SSA) simulation for a subset of the species in the system. This shows that the averaged stochastic behaviour (over 100 runs) tends towards the deterministic behaviour as the number of levels increases. The species shown are the phosphorylated species.

and IDD-CSL [SH09] permit the analysis of the transient behaviour of the stochastic model, e.g., the concentration of MEK-PP at time t is given by:

$$C_{MEK-PP}(t) \;=\; \tfrac{M}{N} \cdot \underbrace{\sum_{i=1}^{N} \big(i \cdot P(L_{MEK-PP}(t) = i)\big)}_{expected\ value\ of\ L_{MEK-PP}(t)} \;.$$

where N is the highest level number and M is the maximum concentration, $2.5\mu Mol$. The random variable $L_{MEK-PP}(t)$ stands for the level of MEK-PP at time t. Figure 7 shows the output of MEK-PP and Raf-1Star_RKIP in the time interval $[0..100]$ according to the continuous and the stochastic models respectively. These results confirm that 7 levels is sufficient to approximate the continuous model, and that there is little discernible difference in output between the number of levels used in the figure.

(2) Analytical stochastic model checking In the former section we employed CTL to express behavioural properties. Since we now have a stochastic model, we can apply the branching-time Continuous Stochastic Logic (CSL), which replaces the path quantifiers (**E**, **A**) in CTL by the probability operator $\mathbf{P}_{\bowtie p}$, whereby $\bowtie p$ specifies the probability of the given formula. First we introduce in CSL the abbreviations

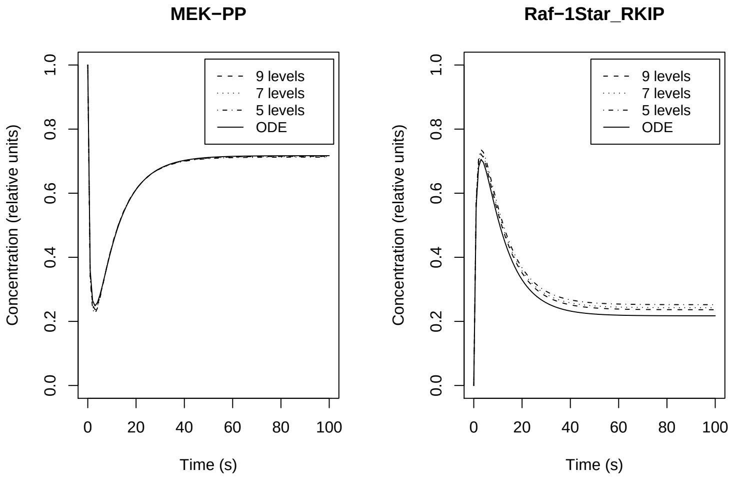


Fig. 7 Comparison of the concentration traces for 5, 7, and 9 levels in the stochastic model against the continuous behaviour.

$\mathbf{F}\,\phi$ for $true\,\mathbf{U}\,\phi$ and $\mathbf{G}\,\phi$ for $!\mathbf{F}\,!\phi$. The CTL formula $\mathbf{EF}\,\phi$ now becomes the CSL formula $\mathbf{P}_{>0}[\mathbf{F}\,\phi]$, and $\mathbf{AG}\,\phi$ becomes $\mathbf{P}_{\geq 1}[\mathbf{G}\,\phi]$. CSL also defines a steady state operator $\mathbf{S}_{\bowtie p}$ which specifies the probability of the given formula in the long run.

The probabilistic model checkers PRISM and IDD-CSL use CSL logic, *see Chapter 6 in this book*. Both tools take an analytical approach, where the probabilities in the probability operator are exact and the result of CTMC analysis. IDD-CSL reads a Petri net standard format. In order to use the PRISM model checker, we have to encode our running example in its modelling language. This translation requires knowledge of the boundedness degree of all species involved, which we acquire by the structural analysis technique of P-invariants. We consider here the level semantics, where we divide the concentration range $(0...2.5\mu Mol)$ into $N+1$ discrete levels. Thus with M, the maximum concentration, $2.5\mu Mol$, we define levels as such: $0,(0*M/N,1*M/N],(1*M/N,2*M/N],...((N-1)*M/N,N*M/N]$. The discrete tokens in the stochastic Petri net now relate to the levels. We use the result from transient analysis that 7 levels are sufficient to describe the behaviour of the system and continue with a 7-level model.

We can now check our stochastic Petri net with CSL properties equivalent to the qualitative CTL properties described in the previous section, whereby places are now interpreted as integer variables.

- **property S1a:** There are reachable states where ERK is phosphorylated and RKIP is not phosphorylated.

$$\mathbf{P}_{>0}\,[\,\mathbf{F}\,(\text{ERK-PP} + \text{Raf-1Star\_RKIP\_ERK-PP} \geq\ 1\ \wedge$$
$$\text{RKIP} + \text{Raf-1Star\_RKIP\_ERK-PP} + \text{Raf-1Star\_RKIP} =\ 7\,)]$$

- **property S2a:** The phosphorylation of ERK (to ERK-PP) is independent of the phosphorylated state of RKIP.
  $\mathbf{P}_{=?}\ [\ \mathbf{G}\ (\ (\text{ERK} =\ 7 \wedge \text{RKIP-P} =\ 0 \wedge \text{RKIP-P\_RP} =\ 0\ ) \rightarrow$
  $\mathbf{P}_{>0}\ [\ (\ \text{RKIP-P} =\ 0 \wedge \text{RKIP-P\_RP} =\ 0\ )\ \mathbf{U}\ \ (\ \text{ERK-PP} \geq\ 1\ )\ ]\ )\ ]$
  We take this opportunity to use the $\mathbf{P}_{=?}$ operator which returns (rather than compares) the probability of the property. As this property always holds true, the returned probability is 1.
- **property S3a:** A cyclic behaviour w.r.t. the presence/absence of RKIP is possible forever.
  $\mathbf{S}_{>0}\ [\ (\ \text{RKIP}= 0\ )\ ]$
  $\mathbf{S}_{>0}\ [\ (\ \text{RKIP}\geq 1\ )\ ]$
  This property is expressed using the steady state operator, which states that in the long run there are nonzero probabilities of RKIP being present or absent; hence it can cycle.

Analytical model checking constructs and analyses the entire state space which can become infeasible due to the state explosion problem. Thus we now move to a simulative approach.

(3) Simulative stochastic model checking A simulative approach handles large state spaces through approximating results by analysing only a subset of the state space – a finite set of finite paths through the CTMC produced by a simulation algorithm.

The type of logic now suitable for describing properties changes from branching-time (e.g., CSL operating over the CTMC) to linear-time. A linear-time logic operates in-turn over paths through the CTMC, equivalent to operating on simulation outputs. We use a probabilistic linear-time temporal logic (PLTL) which extends standard LTL [Pnu81] to a stochastic setting. PLTL is LTL extended with a $\mathbf{P}_{\bowtie p}$ operator and a filter construct, $\{\ \phi\ \}$, defining the initial state from which the property is checked. As PLTL is a linear-time logic, there can be no embedded probability operators.

We use the PLTL Monte Carlo model checker, MC2, with Gillespie's algorithm as featured in Snoopy. The input to the model checker was 100 simulation outputs from Gillespie's algorithm, with a simulation time of $10,000s$. We follow the result that 7 tokens are sufficient and use 7 molecules in the simulation. We now check the following PLTL properties.

- **property S1s:** There are reachable states where ERK is phosphorylated and RKIP is not phosphorylated.
  $\mathbf{P}_{>0}\ [\ \mathbf{F}\ (\ \text{ERK-PP} + \text{Raf-1Star\_RKIP\_ERK-PP} \geq\ 1\ \wedge$
  $\text{RKIP} + \text{Raf-1Star\_RKIP\_ERK-PP} + \text{Raf-1Star\_RKIP} =\ 7\ )]$
- **property S2s:** The phosphorylation of ERK (to ERK-PP) is independent of the phosphorylated state of RKIP.
  $\mathbf{P}_{\geq 1}\ [\ (\ \text{RKIP-P} + \text{RKIP-P\_RP} =\ 0\ )\ \mathbf{U}\ (\ \text{ERK-PP} \geq\ 1\ )\ ]$
  As PLTL cannot have embedded probability operators, we are restricted here to a much weaker property. This property says from the initial state of the system,

ERK can phosphorylate without RKIP being phosphorylated. In fact, from the initial state this is always true.

- **property S3s:** A cyclic behaviour w.r.t. the presence/absence of RKIP is possible forever.

  $\mathbf{P}_{>0}\ [\ \mathbf{F}\ (\text{RKIP} \geq\ 1)\{\text{time} \geq\ 5000 \wedge \text{RKIP} =\ 0\}\ ]$

  $\mathbf{P}_{>0}\ [\ \mathbf{F}\ (\text{RKIP} =\ 0)\{\text{time} \geq\ 5000 \wedge \text{RKIP} \geq\ 1\}\ ]$

  PLTL cannot directly perform steady state analysis. Steady state behaviour can be approximated through long SSA simulations with the property checked towards the end of the simulation. The filter, $\{\phi\}$, defines the initial state that the property is checked from. From late in the simulation where RKIP is absent, we check whether RKIP can become present, and vice versa. This property also holds with a higher time than $5,000s$ used.

Simulative stochastic model checking has its advantage when the range of behaviours that the model checker operates over becomes computationally infeasible. A simulative approach will allow model checking results in a feasible time through approximating. However, our stochastic Petri net with 7 tokens is feasible in both approaches and we would suggest that analytical approaches should be used where possible.

We continue with the computationally less expensive continuous approach considering the averaged behaviour.

4.3 Continuous Analysis

In a continuous Petri net the marking of a place is no longer an integer, but a nonnegative real number, which can be read as the concentration of the species modelled by the place, and transitions fire continuously according to the deterministic rate functions. Assigning these rate functions to all transitions, see Table 4, allows to read the net in Figure 2 as a continuous Petri net.

The semantics of a continuous Petri net is defined by a system of ordinary differential equations (ODEs), whereby one equation describes the continuous change over time on the token value of a given place by the continuous increase of its pretransitions' flow and the continuous decrease of its posttransitions' flow, i.e. each place *p* subject to changes gets its own equation. The complete system of non-linear ODEs generated from the continuous Petri net of our running example is as follows. Please note, the place names stand now for real number variables, and the rate function variables v_i are specified in Table 4. To make the model checking analysis more intuitive, we have mapped the concentrations of $2.5\mu Mol$ to $1\mu Mol$ such that all concentrations are relative.

$$\frac{ds_1}{dt} = v_2 + v_5 - v_1 \qquad \frac{ds_5}{dt} = v_5 + v_7 - v_6 \qquad \frac{ds_9}{dt} = v_4 + v_8 - v_3$$

$$\frac{ds_2}{dt} = v_2 + v_{11} - v_1 \qquad \frac{ds_6}{dt} = v_5 + v_{10} - v_9 \qquad \frac{ds_{10}}{dt} = v_{10} + v_{11} - v_9$$

$$\frac{ds_3}{dt} = v_1 + v_4 - v_2 - v_3 \qquad \frac{ds_7}{dt} = v_7 + v_8 - v_6 \qquad \frac{ds_{11}}{dt} = v_9 - v_{10} - v_{11}$$

$$\frac{ds_4}{dt} = v_3 - v_4 - v_5 \qquad \frac{ds_8}{dt} = v_6 - v_7 - v_8$$

(1) Steady state analysis Simulating the continuous Petri net, i.e. solving numerically the underlying system of ODEs, we get data as shown in Figure 9. Given the usual inexactitudes of the rate constants, biochemists will often interpret the results of ODE-based simulations as indicators of the behaviour of the components of the network, rather than being concerned with the exact value of the concentration of a particular species at a particular point in time.

We perform a quantitative analysis of the simulation results to determine whether the 13 “good” initial states (markings) suggested by the qualitative Petri net analysis, see Table 3, are indeed in some way equivalent (they all result in the same steady state), and that no other possible initial states can be used to give the same results. These are “sensible” initial states from the point of view of biochemistry, in that in all these 13 cases, and in none of the other 2035 states, each protein species is in a high initial concentration in only one of the following states: uncomplexed, complexed, unphosphorylated or phosphorylated. These conditions relate exactly to the 1-P-invariant interpretation given in our initial marking construction procedure in Section 3.2.

For the purposes of our computations we map any nonzero concentrations to 1. We then compute the steady state of the set of species for each possible initial state, using e.g. the MatLab ODE solver ode45, which is based on an explicit Runge-Kutta formula, the Dormand-Prince pair [DP80], with 100 time steps. We find that all of the 13 “good” initial states result in the same final state, within the bounds of computational error of the ODE solver, while none of the other 2035 “bad” initial states results in a steady state which is near that of the set of “good” states, compare Figure 8.

We reproduce two simulations of the model: State 1 (Figure 9, left) corresponding to the initial marking suggested by our approach described above in Section 3.2 with ERK-PP low and ERK high, and State 5 (Figure 9, right) corresponding to the initial marking suggested by Cho et al [CSK^{+}03] where the initial concentration of ERK-PP is high and ERK is low. State 1 has been confirmed by an expert signal transduction researcher as the most sensible starting state [Kol05]. The equivalence of the steady states, compared with the difference in some intermediate states is clearly illustrated in these figures. For example, the concentration of Raf-1Star_RKIP behaves overall in a similar manner in both State 1 and State 5, peaking before 10 seconds although the peak is greater when ERK is not phosphorylated at the start of the experiment. In Figure 10 we reproduce the computed behaviour of

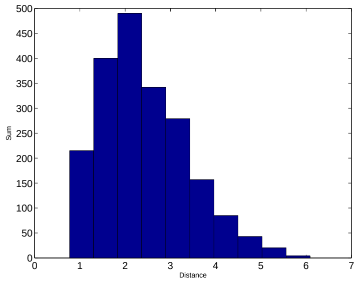


Fig. 8 Distribution of the Euclidean distances between the steady state vector of the “good” initial states and each of the steady state vectors for the “bad” initial states. The distances range between 0.7736 to 6.0889.

ERK-PP for all 13 good initial states, showing that despite differences in the concentrations at early time-points, the steady state concentration is the same in all 13 states.

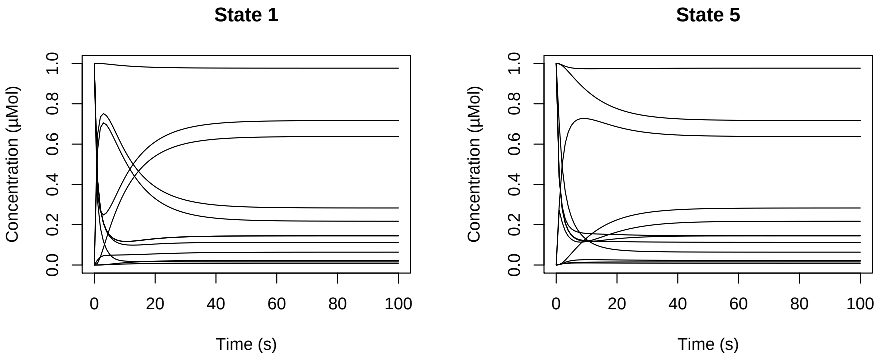


Fig. 9 Dynamic behaviour for m1 and m5.

(2) Continuous model checking of the transient behaviour The continuous model replaces the discrete values of species with continuous values, and hence is not able to describe the behaviour of species at the level of individual molecules, but only the overall deterministic behaviour via concentrations. The concentration of a particular species will have the same value at each point of time for repeated experiments. The state space of such models is continuous and linear. So, a linear-time logic is a natural choice to analyse such kind of behaviour.

We employ the PLTL-based MC2 model checker again as per the stochastic analysis. MC2 operates on a deterministic model in the same manner as on a stochastic model - analysing simulation output. However, in the deterministic model there is

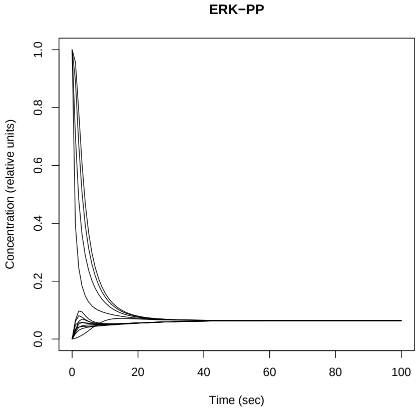


Fig. 10 Dynamic behaviour of ERK-PP for all 13 "good" states.

only one behaviour (the average behaviour) so the resulting probability is 1 or 0. To provide simulation output to MC2, we use the BioNessie simulator with a simulation time of $10,000s$.

The PLTL properties from the stochastic analysis are rewritten to concentration values, comparing to real numbers called *noise* and *significant*. The place names are now interpreted as real number variables.

- **property C1:** There are reachable states where ERK is phosphorylated and RKIP is not phosphorylated.
  $\mathbf{P}_{\geq 1}$ $[\,\mathbf{F}\,(\text{ERK-PP} + \text{Raf-1Star\_RKIP\_ERK-PP} > \textit{noise} \wedge$
  $\text{RKIP} + \text{Raf-1Star\_RKIP\_ERK-PP} + \text{Raf-1Star\_RKIP} \geq \textit{significant}\,)]$
- **property C2:** The phosphorylation of ERK (to ERK-PP) is independent of the phosphorylated state of RKIP.
  $\mathbf{P}_{\geq 1}$ $[\,(\text{RKIP-P} + \text{RKIP-P\_RP} \leq \textit{noise}\,)\ \mathbf{U}$
  $(\text{ERK-PP} + \text{Raf-1Star\_RKIP\_ERK-PP} > \textit{noise}\,)\,]$
  Due to the continuous change of token values and the large kinetic rate of ERK-PP's posttransition relative to its pretransitions, ERK-PP's concentration quickly becomes consumed in the complex Raf-1Star_RKIP_ERK-PP. Therefore, this complex has been added to the property.
- **property C3:** A cyclic behaviour w.r.t. the presence/absence of RKIP is possible forever.
  $\mathbf{P}_{\leq 0}\ [\,\mathbf{F}\,(\text{RKIP} > \textit{noise})\{\text{time} \geq\ 5000 \wedge \text{RKIP} \leq \textit{noise}\}\,]$
  $\mathbf{P}_{\leq 0}\ [\,\mathbf{F}\,(\text{RKIP} \leq \textit{noise})\{\text{time} \geq\ 5000 \wedge \text{RKIP} > \textit{noise}\}\,]$
  As this is the average behaviour of the system, the cyclic behaviour of RKIP is averaged and is no longer present. Hence the property above say that it is not true ($\mathbf{P}_{\leq 0}$) that RKIP can cycle between noise and not noise.

These properties hold for the intuitive values of $0.1\mu Mol$ and $0.9\mu Mol$ for *noise* and *significant*, respectively.

4.4 Summary

In summary, quantitative Petri nets can provide interesting analysis in two complementary worlds, stochastic and continuous. The key points below have been discussed in the text:

1. The qualitative Petri net can easily be converted to quantitative Petri nets by the addition of rate equations.
2. All of the 13 states identified by the reachability graph of the qualitative Petri net result in the same set of steady state values for the 11 species in the pathway.
3. None of the remaining 2035 possible initial states of the qualitative Petri net results in a final steady state close to that generated by the 13 markings in the reachability graph.
4. The transient behaviour - the crucial point of interest in signal transduction networks - of the quantitative models is sensible for the 13 states identified by our method.
5. There are significant differences between the three quantitative model checking approaches. Analytical model checking provides exact analysis but quickly becomes infeasible for larger number of levels. Simulative model checking provides analysis past that which analytical can through approximation, however there can be restrictions on the properties which can be checked. Continuous model checking analyses the average behaviour of the system only.
6. Model checking the stochastic and continuous Petri nets complement each other; it is possible that there are behaviours in the stochastic Petri net which are lost in the continuous net. One should model check the stochastic Petri net before considering the overall behaviour of the system in the continuous Petri anet.

5 Tools

The running example in its three interpretations has been done using Snoopy - a tool to design and animate or simulate hierarchical graphs, among them the qualitative, stochastic and continuous Petri nets as used in this chapter. Snoopy provides export to various analysis tools as well as SBML import and export [HRS08], [Sno08].

The qualitative analyses have been made with the Petri net analysis tool Charlie [Cha08] and the Model Checking Kit [SSE03], supported by IDD-CTL, a CTL model checker utilising interval decision diagrams for concise state space representations [Tov08], [HST09].

The quantitative analyses have been done using Snoopy's build-in simulation algorithms for stochastic and continuous Petri nets, and by Bionessie [Bio08], an SBML-based simulation and analysis tool for biochemical networks. Additionally, MATLAB [SR97] was used to produce the steady state analysis of all initial states in the continuous model. We employed PRISM [PNK06] and IDD-CSL [SH09] for analytical stochastic model checking, and MC2 [MC208] for simulative stochastic and continuous model checking. We use the Gillespie simulator in Snoopy as input to stochastic MC2 and the BioNessie simulator as input to continuous MC2.

More Petri nets tools and related material can be found on the Petri Nets World's web page: http://www.informatik.uni-hamburg.de/TGI/PetriNets/.

6 Further Reading

General introductions into Petri net theory can be found e.g. in [Mur89], [Rei82], [Sta90]; for stochastic Petri nets see specifically [BK02], [MBC$^+$95], and for continuous and hybrid Petri nets [DA05].

Biochemically interpreted stochastic Petri nets have been introduced in [GHL07]. An extension of this net class is used in [HLGM09] for the model-based design of wetlab experiments employing simulative stochastic model checking. Biochemically interpreted continuous Petri nets – as used in this chapter – have been introduced in [GH06], which are deployed in [BGHO08] for a structured approach for the engineering of biochemical network models.

Recent surveys on applying Petri nets for biochemical networks are [Cha07] and [MLM06], offering a rich choice of further reading pointers, among them numerous case studies applying Petri nets to biochemical networks. Besides the net classes introduced in this chapter, coloured Petri nets, duration and interval time Petri nets as well as hybrid Petri nets in various extensions have been employed.

P- and T-invariants have been established concepts in Petri net theory since the very beginning [Lau73]. There are corresponding notions in systems biology, called chemical moieties or conservation relations, and elementary modes or extreme pathways, which are elaborated in the setting of biochemical networks in [Pal06]. For biochemical systems without reversible reactions, the notions T-invariants, elementary modes and extreme pathways coincide.

A good starting point for qualitative model checking is [CGP01], and for numerical solution of Markov chains [Ste94]. An overview on stochastic issues for systems biology is given in [Wil06].

The systematic qualitative analysis of a metabolic network is demonstrated in [KH08], following basically the same outline as used in section 3.2. Further examples illustrating how to combine qualitative and quantitative analysis techniques are elaborated in [HGD08], [GHR$^+$08]. The first paper provides also the formal definitions of the technical Petri net terms used in this chapter as well as further references to related papers.

7 Summary

In this chapter we have introduced qualitative, stochastic and continuous Petri nets as well as related analysis techniques in a deliberately informal manner. Our motivation has been to make the concepts accessible for both modellers as well as biologists who are not familiar with this methodology. For the sake of biological coherence we have used a signalling pathway, namely the influence of Raf Kinase Inhibitor Protein (RKIP) on the Extracellular signal Regulated Kinase (ERK) signalling pathway [CSK$^+$03], as a running example. However, the techniques we have described can be equally applied to other types of biochemical networks in systems biology or synthetic biology, as e.g. metabolic or gene transcription networks, or even combinations of them.

The nature of biochemical reaction systems – bipartite (in terms of species and interactions), concurrent and stochastic – means that Petri nets are a natural choice for a formal modelling technique to describe their structure and behaviour. The qualitative aspects of these systems, e.g. the biochemical entities, and their relationships up to stoichiometry can be modelled with qualitative Petri nets, whereas quantitative behaviour can be described by stochastic Petri nets at the individual (e.g. molecular) level and continuous Petri nets on the larger scale. Petri nets are a well-established technique for modelling computational systems, and we hope that this chapter has convinced the reader of their suitability for biochemical reaction systems. There are a large range of robust Petri net modelling tools available, and we have chosen one, namely Snoopy [Sno08] which is freely obtainable and is available for the major computational platforms. As well as the ability to describe static information about biochemical reaction systems, Petri nets are able to simulate their behaviour using for example the token game for qualitative nets, stochastic simulation algorithms for stochastic nets, or ordinary differential equation solvers for continuous nets.

In addition to description and simulation, there are very powerful techniques available to analyse the behaviour of such systems. Qualitative Petri nets can be analysed based on their structural or behavioural properties using classical techniques. Model checking can be used to prove special properties which reflect the intended functionality of the network. This can be done for all three types of nets – qualitative, stochastic and continuous – using the appropriate temporal logics, and we have chosen to take model checking of properties of our running example as the unifying thread throughout this paper.

Altogether we advocate a three-step technology for the modelling and analysis of biochemical networks in a systematic manner: (1) qualitative, i.e. (time-free) qualitative modelling and analysis, esp. for the beneficial effect of confidence-increasing model validation, (2) stochastic modelling and analysis to exclude eccentric behaviour, and (3) continuous modelling and analysis, esp. with the hope of the reliable prediction of behaviour in the averaged case. For all steps we favour the deployment of both qualitative as well as quantitative Petri nets, sharing the same net structures for a given case. The quantitative models are derived from the qualitative one only by the addition of the quantitative parameters. Hence these models are likely to share some behavioural properties.

The running example in its three versions and all related analysis protocols are available at: www-dssz.informatik.tu-cottbus.de/examples/erk.

8 Exercises

You should be able to solve the first three tasks without tool support, while the remaining tasks suggest the employment of adequate tools.

1. The basic Petri Nets properties boundness, liveness, and reversibility are orthogonal, i.e. independent from each other. Prove this statement by providing a net for each possible combination (obviously, there are eight of them). Argue for every net why the properties hold as you claim.
   If you found this too easy, repeat this exercise with the additional restriction that the nets should be (a) ordinary, (b) strongly connected.
2. A well-defined matrix operation is the transposition, which exchanges rows and columns. If we apply the matrix transposition to the incidence matrix of a Petri net, what happens with the Petri net and its invariants?
3. While CTL model checking fits particularly for the decision of special properties, it can also be used to decide general properties. Express the following properties in CTL.

   - Check the liveness of reactions r1 and r9. What had to be done to decide the liveness of the net?
   - Check the reversibility of the net.
   - Check the token conservation of the P-invariants.
   - The CTL property 2, see Section 3.2, does not hold in its stronger version, where EU is replaced by AU, even not for the reduced network neglecting the backward directions of reversible reactions. Why?

4. Extend the running example of this chapter by the following reactions [CGH05]:

r12, r13:	MEK	+ Raf-1*	↔	MEK_RAF-1*	
r14:	MEK_RAF-1*		→	MEK-PP	+ Raf-1*
r15:	MEK-PP		→	MEK .	

   Adapt the given Petri net model and apply step-wise all analysis techniques presented in this chapter, following exactly the same outline. Use the following rate constants: $k_{12} = 0.05$, $k_{13} = 0.03$, $k_{14} = 0.06$, $\mathrm{k}_{15} = 0.02$.
   Discuss also the version, where the backward reactions of all reversible reactions are neglected; interesting behaviour, isn't it?

5. Use the techniques you have learned in this chapter to

   - derive qualitative and quantitative Petri net models for the 2-stage signal transduction cascade given in Figure 11,
   - construct a suitable initial marking,
   - animate and simulate their behaviours,
   - analyse their properties and behaviours.

   Use the following rate constants for the quantitative models:
   $k_1 = kr_1 = kk_1 = kkr_1 = 100$,
   $k_2 = kr_2 = kk_2 = kkr_2 = 4$,

$k_3 = kr_3 = kk_3 = kkr_3 = 1.$

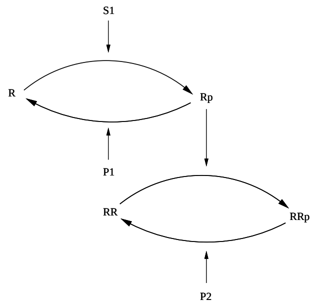


Fig. 11 The 2-stage signal transduction cascade.

$$R + S_1 \underset{k_2}{\overset{k_1}{\rightleftarrows}} R|S_1 \xrightarrow{k_3} R_p + S_1$$

$$R_p + P_1 \underset{kr_2}{\overset{kr_1}{\rightleftarrows}} R_p|P_1 \xrightarrow{kr_3} R + P_1$$

$$RR + R_p \underset{kk_2}{\overset{kk_1}{\rightleftarrows}} RR|R_p \xrightarrow{kk_3} RR_p + R_p$$

$$RR_p + P_2 \underset{kkr_2}{\overset{kkr_1}{\rightleftarrows}} RR_p|P_2 \xrightarrow{kkr_3} RR + P_2$$

References

[BGHO08] R. Breitling, D. Gilbert, M. Heiner, and R. Orton. A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. *Briefings in Bioinformatics*, (9):404 – 421, 2008.

[Bio08] BioNessie website. A biochemical pathway simulation and analysis tool. University of Glasgow, http://www.bionessie.org, 2008.

[BK02] F. Bause and P.S. Kritzinger. *Stochastic Petri Nets*. Vieweg, 2002.

[CGH05] M. Calder, S. Gilmore, and J. Hillston. Automatically deriving ODEs from process algebra models of signalling pathways. In *Proc. CSMB 2005*, pages 204–215. LFCS, Univ. of Edinburgh, 2005.

[CGP01] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model checking*. MIT Press 1999, third printing, 2001.

[Cha07] C. Chaouiya. Petri Net Modelling of Biological Networks. *Briefings in Bioinformatics*, 8(4):210 – 219, 2007.

[Cha08] Charlie Website. A Tool for the Analysis of Place/Transition Nets. BTU Cottbus, http://www-dssz.informatik.tu-cottbus.de/software/charlie/charlie.html, 2008.

[CRCVDS04] N. Chabrier-Rivier, M. Chiaverini, F. Fages V. Danos, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, 2004.

[CSK$^+$03] K.-H. Cho, S.-Y. Shin, H.-W. Kim, O. Wolkenhauer, B. McFerran, and W. Kolch. Mathematical modeling of the influence of RKIP on the ERK signaling pathway. *Lecture Notes in Computer Science*, 2602:127–141, 2003.

[DA05] R. David and H. Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, 2005.

[DP80] J. R. Dormand and P. J. Prince. A family of embedded runge-kutta formulae. *J. Comp. Appl. Math.*, 6:1–22, 1980.

[GH06] D. Gilbert and M. Heiner. From Petri nets to differential equations - an integrative approach for biochemical network analysis. In *Proc. ICATPN 2006*, pages 181–200. LNCS 4024, Springer, 2006.

[GHL07] D. Gilbert, M. Heiner, and S. Lehrack. A unifying framework for modelling and analysing biochemical pathways using Petri nets. In *Proc. CMSB 2007*, pages 200–216. LNCS/LNBI 4695, Springer, 2007.

[GHR+08] D. Gilbert, M. Heiner, S. Rosser, R. Fulton, Xu Gu, and M. Trybiło. A Case Study in Model-driven Synthetic Biology. In *2nd IFIP Conference on Biologically Inspired Collaborative Computing (BICC), IFIP WCC 2008, Milano*, pages 163–175, 2008.

[Gil77] D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[HGD08] M. Heiner, D. Gilbert, and R. Donaldson. Petri Nets in Systems and Synthetic Biology. In *Schools on Formal Methods (SFM)*, pages 215–264. LNCS 5016, Springer, 2008.

[HLGM09] M. Heiner, S. Lehrack, D. Gilbert, and W. Marwan. Extended Stochastic Petri Nets for Model-Based Design of Wetlab Experiments. pages 138–163. LNBI 5750, Springer, 2009.

[Hof94] R. Hofestädt. A Petri Net Application of Metabolic Processes. *Journal of System Analysis, Modeling and Simulation*, 16:113–122, 1994.

[HRS08] M. Heiner, R. Richter, and M. Schwarick. Snoopy - A Tool to Design and Animate/Simulate Graph-Based Formalisms. In *Proc. PNTAP 2008, associated to SIMUTools 2008*. ACM digital library, ISBN 978-963-9799-20-2, 2008.

[HST09] M. Heiner, M. Schwarick, and A. Tovchigrechko. DSSZ-MC-A Tool for Symbolic Analysis of Extended Petri Nets. In *Proc. Petri Nets 2009*, pages 323–332. LNCS 5606, Springer, 2009.

[KH08] I. Koch and M. Heiner. *Petri Nets, in Junker B.H. and Schreiber, F. (eds.), Biological Network Analysis*, chapter 7, pages 139 – 179. Wiley Book Series on Bioinformatics, 2008.

[Kol05] Walter Kolch. personal communication, 2005.

[Lau73] K. Lautenbach. Exact Liveness Conditions of a Petri Net Class (in German). Technical report, GMD Report 82, Bonn, 1973.

[MBC+95] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley Series in Parallel Computing, John Wiley and Sons, 1995. 2nd Edition.

[MC208] MC2 Website. MC2 - PLTL model checker. University of Glasgow, http://www.brc.dcs.gla.ac.uk/software/mc2/, 2008.

[MLM06] H. Matsuno, C. Li, and S. Miyano. Petri Net Based Descriptions for Systematic Understanding of Biological Pathways. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E89-A(11):3166 – 3174, 2006.

[Mur89] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proc.of the IEEE 77*, 4:541–580, 1989.

[Pal06] B.O. Palsson. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, 2006.

[Pet62] C.A. Petri. *Communication with Automata (in German)*. Schriften des Instituts für Instrumentelle Mathematik, Bonn, 1962.

[PNK06] D. Parker, G. Norman, and M. Kwiatkowska. *PRISM 3.0.beta1 Users' Guide*, 2006.

[Pnu81] A. Pnueli. The temporal semantics of concurrent programs. *Theor. Comput. Sci.*, 13:45–60, 1981.

[Red94] V. N. Reddy. Modeling Biological Pathways: A Discrete Event Systems Approch. Master thesis, University of Maryland, 1994.

[Rei82] W. Reisig. *Petri nets; An introduction*. Springer, 1982.

[RML93] V. N. Reddy, M. L. Mavrovouniotis, and M. L. Liebman. Petri Net Representations in Metabolic Pathways. In *Proc. of the Int. Conf. on Intelligent Systems for Molecular Biology*, 1993.

[SH09] M. Schwarick and M. Heiner. CSL model checking of biochemical networks with Interval Decision Diagrams. In *Proc. 7th CMSB*, pages 296–312. LNCS/LNBI 5688, Springer, 2009.

[Sno08] Snoopy Website. A Tool to Design and Animate/Simulate Graphs. BTU Cottbus, http://www-dssz.informatik.tu-cottbus.de/software/snoopy.html, 2008.

[SR97] L. F. Shampine and M. W. Reichelt. The MATLAB ODE Suite. *SIAM Journal on Scientific Computing*, 18:1–22, 1997.

[SSE03] C. Schröter, S. Schwoon, and J. Esparza. The Model Checking Kit. In *Proc. ICATPN, LNCS 2679*, pages 463–472. Springer, 2003.

[Sta90] P.H. Starke. *Analysis of Petri Net Models (in German)*. B. G. Teubner, Stuttgart, Stuttgart, 1990.

[Ste94] W.J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton Univ. Press, 1994.

[Tov08] Tovchigrechko, Alex. *Model Checking Using Interval Decision Diagrams*. PhD thesis, BTU Cottbus, Dep. of CS, 2008.

[Wil06] D.J. Wilkinson. *Stochastic Modelling for System Biology*. CRC Press, New York, 1st Edition, 2006.