Rule-based modeling of transcriptional attenuation at the tryptophan operon

Celine Kuttler, Cédric Lhoussaine, Mirabelle Nebut

Rule-based Modeling of Transcriptional Attenuation at the Tryptophan Operon

Céline Kuttler, Cédric Lhoussaine, Mirabelle Nebut[1,2]

[1] University of Lille
[2] BioComputing group, LIFL & IRI (CNRS UMR 8022 & USR 3078)

Abstract. Transcriptional attenuation at E.coli's tryptophan operon is a prime example of RNA-mediated gene regulation. In this paper, we present a discrete stochastic model of the fine-grained control of attenuation, based on chemical reactions. Stochastic simulation of our model confirms results that were previously obtained by master or differential equations. Our approach is easier to understand than master equations, although mathematically well founded. It is compact due to rule schemas that define finite sets of chemical reactions. Object-centered languages based on the π-calculus would yield less intelligible models. Such languages are confined to binary interactions, whereas our model heavily relies on reaction rules with more than two reactants, in order to concisely capture the control of attenuation.

1 Introduction

Transcriptional attenuation is a control mechanism deployed by bio-synthetic operons across bacterial species [14,15,38]. Operons are sequences of jointly transcribed genes, bio-synthetic operons encode enzymes for the synthesis of amino acids. Attenuation prematurely interrupts an ongoing round of the operon's transcription, in situations where the environment already contains a high concentration of the corresponding amino acid. Summarizing, it works as follows. First, the amino acid concentration determines the speed at which a ribosome translates the nascent messenger RNA (mRNA). Second, the ribosome's position controls how the mRNA folds into a two-dimensional structure. Finally, the mRNA structure sets the end point of transcription.

Although attenuation has been investigated within bacterial systems since the 1970s [19,36], it attracted significantly less interest than the control of transcription initiation, that is mediated by DNA binding proteins. This changed in the 2000s after the discovery of regulatory mechanisms in higher organisms that exploit RNA properties [4]. Quantitative investigations of RNA-mediated regulation gained momentum for therapeutic approaches and synthetic biology [3].

E.coli's tryptophan (*trp*) operon is the best understood bio-synthetic operon. It allows the bacterium to synthesize the amino acid tryptophan upon need. Tryptophan regulation in *E.coli* relies on two further mechanisms beyond transcriptional attenuation, that are not considered in this paper.

Santillan and Zeron (2004) [30] modeled all three levels of *trp* regulation in *E. coli* through delay differential equations (DDE), without investigating attenuation in detail. DDEs are usually directly derived from informal biochemical reactions. The main drawback of such deterministic models is that they only provide observations of average behavior. In particular, they do not account for possible stochastic noise from which multi-modal states may arise. In that case, the average behavior does not correspond to any of the actual states. Since regulatory systems involve few biological entities, a criterion known to increase stochastic effects, one may wonder if the deterministic assumption is appropriate regarding *E.coli*'s *trp* operon. This calls for stochastic modeling. The first stochastic treatment of attenuation at the *trp* operon indeed dates back to 1977 [34].

Elf and Ehrenberg (2005) [11] analyze the sensitivity of attenuation through probability functions and, more generally, discrete master equations. This approach benefits from a rich probability theory that gives valuable insights and measurement capabilities. However, apart from rare exceptions, master equations can only be evaluated numerically, and not solved symbolically. Each biological system requires an ad-hoc master equation or probability function that is usually hard to design from the mechanistic intuition of the system.

Discrete event models for stochastic simulation are commonly described by chemical reactions. These can be studied within formal rule-based modeling languages [6,7,8,21], where molecular systems are understood as multisets of molecules, that are rewritten by chemical reactions. Reaction speeds are derived from rate constants and cardinalities of sets. The stochastic semantics of chemical reactions is given in terms of continuous time Markov chains (CTMCs). The algorithm of Gillespie (1976) [12] allows direct stochastic simulation, starting from a given multiset of molecules and a set of chemical reactions. Rule-based models are intuitive in the sense that they describe molecular interactions and are simpler to modify and extend than models based on classical mathematical functions.

Certain authors [5,26] support the idea that *binary reactions* are sufficient to represent biochemical knowledge. They do so to advocate formal object-centric representations that are confined to binary interactions, namely recent languages based on the stochastic π-calculus [10,17,22,25,28]. However, rewriting n-ary to binary reactions is tedious and requires sufficient expressiveness of formal languages. Sequences of reactions need to be executed within atomic transactions, so that no other interactions intervene.

Contribution. In this work, we present the first formal rule-based stochastic model of transcriptional attenuation at *E. coli*'s tryptophan operon. We cover a similar extent of biological knowledge as Elf and Ehrenberg (2005) [11], but take a different methodological approach in the tradition of stochastic models of gene expression [1]. Our representation is based on chemical reactions. We use 71 reactions to faithfully cover the *trp* operon's control by attenuation, summarizing the rich narrative account in the biological literature [13,20,33]. We obtain a

concise description by two ingredients, 13 *rules schemas* (introduced in Section 3) from which we generate our 71 chemical reactions, and *n-ary chemical reactions*.

By means of *rule schemas*, we represent finite sets of chemical reactions in a compact manner, which differ only in the choice of certain molecule parameters that are quantified over e.g. folding or binding state, or location. This idea is well known from logic programming [23], unification grammars [31] or term rewriting [2].

N-ary reactions are indispensable to intelligible representations of the *trp* attenuator, as our work indicates. We hypothesize the same holds for many other biological cases. By n-ary reactions, we refer to rules with three or more inputs, as opposed to binary reactions. They allow to incorporate *global control* into models, of which our work distinguishes three categories.

First are *conditions for rule application*. Here, one among a rule's multiple inputs is neither consumed nor modified by the rule's application. However if this molecule wasn't available, the rule could not be applied. For instance, we use this mechanism to model that transcription only aborts if the nascent transcript has folded into the termination hairpin, i.e. the corresponding rule checks this later's presence.

In the second category, an actor undergoes a state change as a *side effect* of the reaction between two others. For instance, after translation has progressed beyond a certain threshold, a state change hinders the corresponding mRNA components to form hairpins.

The third category allows to *switch between abstraction levels*: upon application of a rule, one actor is replaced by the enumeration of its individual constituents, or vice versa. We use this mechanism for dedicated control segments of mRNA that can interact as a whole with other segments, or be processed stepwise. Depending on the circumstances our model opts for their representation either as a whole, or as the sequence of their constituents.

Paper outline. We review the biological background in Sect. 2, introduce our rule-based language in Sect. 3, and review related languages in Sect. 3.4. As a first example, we model the multi-step race between transcription and translation by four rule schemas in Sect. 4.1. We quantitatively investigate the hypersensitivity of this *basic model* by simulation with the Kappa Factory[3] [8]. The *concurrent elongation model* of Sect. 4.2 extends, it explicitly renders the simultaneity of transcription and translation of the same mRNA. We use it to investigate the quantitative impact on the multi-step race when only part of the mRNA is present at the beginning of the simulation. In Sect. 5, we present our model of the detailed attenuation mechanism at *E.coli*'s *trp* operon. We qualitatively reproduce and confirm results of Elf and Ehrenberg [11] regarding the probability of uninterrupted transcription into the full operon as a function of the rate of *trp*-codon translation, and discuss the quantitative differences between our results and theirs.

[3] This preliminary implementation of the κ-calculus was available to us as beta-testers, the web-based tool *Cellucidate* (`http://cellucidate.com`) is its successor.

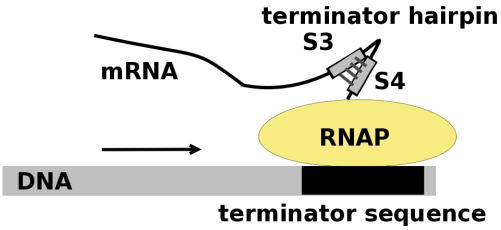

Fig. 1. Transcription terminates if the most recent portion of mRNA is folded into a hairpin, when RNAP reaches a terminator DNA sequence.

2 Transcriptional Attenuation

Transcriptional attenuation prematurely interrupts a gene's ongoing transcription, or that of an *operon*, when the cell does not actually need the corresponding proteins. *E.coli*'s *trp* operon encodes enzymes for the biosynthesis of the amino acid tryptophan (Trp). Their production is attenuated if the bacterium's environment provides sufficient amounts of tryptophan to feed on.

In this section, we first review the principles of gene expression in bacteria. Then, we introduce ribosome-mediated transcriptional attenuation, a regulatory mechanism used across bacterial species, and how specifically it functions at *E.coli*'s tryptophan (*trp*) operon [13,36,37]. We omit certain aspects documented in the biological literature that do not enter our formal model presented in this paper, such as the role of transfer RNA in translation, or the redundancy of the genetic code.

Transcription copies information content from a DNA sequence into an mRNA sequence. It is carried out by an enzyme called RNA polymerase. RNAP initiates its work by binding to a distinguished short DNA sequence which indicates the beginning of a gene (or operon), from where RNAP starts assembling an mRNA molecule. In the following *elongation phase* RNAP advances stepwise over DNA, extending the growing mRNA nucleotide by nucleotide, progressing at an average rate of 50 nucleotides per second. Transcription *terminates* when RNAP encounters a terminator sequence on DNA, if an additional condition is then fulfilled. Figure 1 illustrates this additional condition, that depends on a property of the mRNA being transcribed. The linear mRNA sequence can fold into stable secondary structures, which due to their shape are called *hairpins*. In order for transcription to terminate, while the RNAP encounters a terminator sequence on DNA, the most recent portion of the transcript must be folded into a hairpin.

Translation reads out an mRNA molecule into the corresponding sequence of amino acids. It initiates with the binding of a *ribosome* to the free end of an

Fig. 2. Leader region of the *trp* mRNA. Adjacent pairs of the four segments S_1 to S_4 fold into alternative hairpins. The *anti-terminator* hairpin $S_2 \cdot S_3$ promotes transcription of the operon, whereas the *terminator* $S_3 \cdot S_4$ aborts it.

mRNA, the other end of which is still being elongated by an RNAP. The ribosome advances along the mRNA towards the RNAP in steps of *codons*, which are words of three mRNA nucleotides. For each codon, the ribosome adds the corresponding amino acid to the growing sequence. These amino acid sequences later fold into three dimensional structures known as proteins. While the average rate of translation is 15 codons per second, each step of the ribosome is actually limited by the abundance of the currently required amino acid. The ribosome slows down on codons for which the corresponding amino acid is in short supply.

Transcriptional attenuation subtly couples the termination of an ongoing round of transcription to the translation efficiency of the first part of the nascent mRNA, where this latter is limited by a critical amino acid (tryptophan, for the *trp* operon). The so-called *leader* sequence consists of the operon's first few dozen nucleotides. Attenuation boils down to a *race* between the RNAP transcribing the leader DNA, and the ribosome translating the leader mRNA. In a nutshell, the attenuation race is as follows. If the amino acid of interest is *abundant*, the ribosome advances at its maximal speed, and the terminator hairpin forms. Transcription then aborts. Conversely if the critical amino acid is *rare*, the ribosome stalls early within the leader. The stalled ribosome inhibits the terminator hairpin, hence the RNAP wins the race, and transcription continues into the operon's protein coding regions.

Trp leader architecture. The leader's architecture is fundamental to attenuation at *E.coli*'s tryptophan operon. We distinguish four segments S_1 to S_4 within the leader mRNA, see Fig. 2. Each pair of adjacent segments folds into a hairpin, if neither of the required segments is masked by a ribosome. Three different secondary structures can occur within the leader mRNA of the *trp* operon. They are named by their respective roles in attenuation. The pairing of S_1 with S_2 represents the *pause* hairpin, that between S_2 and S_3 is the *anti-terminator* hairpin, and the *terminator* shown in Fig. 1 is the pairing of S_3 with S_4.

Hairpin co-occurrence and mutual exclusion. Each leader segment can only participate in one hairpin at the same time. Most importantly, the anti-terminator prohibits the terminator hairpin by sequestering S_3. Because both require S_2, the pause hairpin excludes the anti-terminator. On the other hand, the pause hairpin ($S_1 \cdot S_2$) and the terminator ($S_3 \cdot S_4$) can co-occur, since they do not

compete for a shared segment. With respect to our model, we need to mention that hairpin formation is fater than any other reaction in the system. It is also important to bear in mind that the segments become available one by one while RNAP transcribes the *trp* operon's leader, and that the leader transcript progressively forms hairpins whenever the ribosome's position allows. Hence, the leader mRNA never indeed remains unfolded, as simplifyingly shown in Fig. 2.

The role of hairpins. The impact of hairpins on transcription is significant, they determine *pausing* of the RNAP and *termination* of transcription. As opposed to this, mRNA hairpins do not impair translation. A translating ribosome disrupts hairpins along its way, without significantly slowing down. We now detail on the pause and terminator hairpins at *E.coli*'s *trp* operon.

Pause hairpin. After RNAP has transcribed the segments S_1 and S_2, it remains stalled on a strong DNA *pause site*, while the mRNA rapidly folds into the pause hairpin. This combination resembles the conditions for transcription termination, however, it is reversible: RNAP resumes transcriptions after a ribosome has arrived along the transcript and disrupted the pause hairpin. Let us consider the details of this *initial configuration for the attenuation race* in Fig. 3 (left). RNAP is stalled on the pause site on DNA, more precisely on the nucleotide that we refer to as DNA_0. It has so far transcribed the leader up to and including its segments S_1 and S_2, that have folded into the pause hairpin. The approaching ribosome disrupts the pause hairpin with its step onto the 7^{th} codon of the leader mRNA, which is the first step from the initial conformation in Fig. 3. The attenuation race now starts.

Terminator hairpin. The DNA leader of the *trp* operon contains a *terminator sequence*, just after the portion that encodes S_4, see Fig. 1. When RNAP arrives here, the terminator mRNA hairpin can form. The combination of terminator mRNA hairpin and terminator DNA sequence aborts transcription. However if the anti-terminator is already present when S_4 is completed - it can appear as soon as S_2 and S_3 have been transcribed - the terminator is prevented. In this case RNAP continues unhindered through the terminator sequence, and reaches the enzyme-coding region of the operon (the *structural genes*).

Trp codons within the leader mRNA. We have not yet mentioned the codons 10 and 11 of the leader mRNA (see Fig. 2), the translation of which each requires one tryptophan molecule. These two *control* codons determine the outcome of attenuation race. They act as sensors for the tryptophan concentration, and determine the speed of the ribosome's forward movement.

If tryptophan is in *rare*, the ribosome stalls on the control codons, hence its footprint does not advance far enough to mask the second segment. Soon later, the anti-terminator hairpin forms between S_2 and S_3, and transcription continues into the structural genes. This is depicted as *read-through configuration* in Fig. 3.

Conversely if tryptophan is *abundant*, the ribosome efficiently translates through the control codons. From the time point the ribosome has reached the

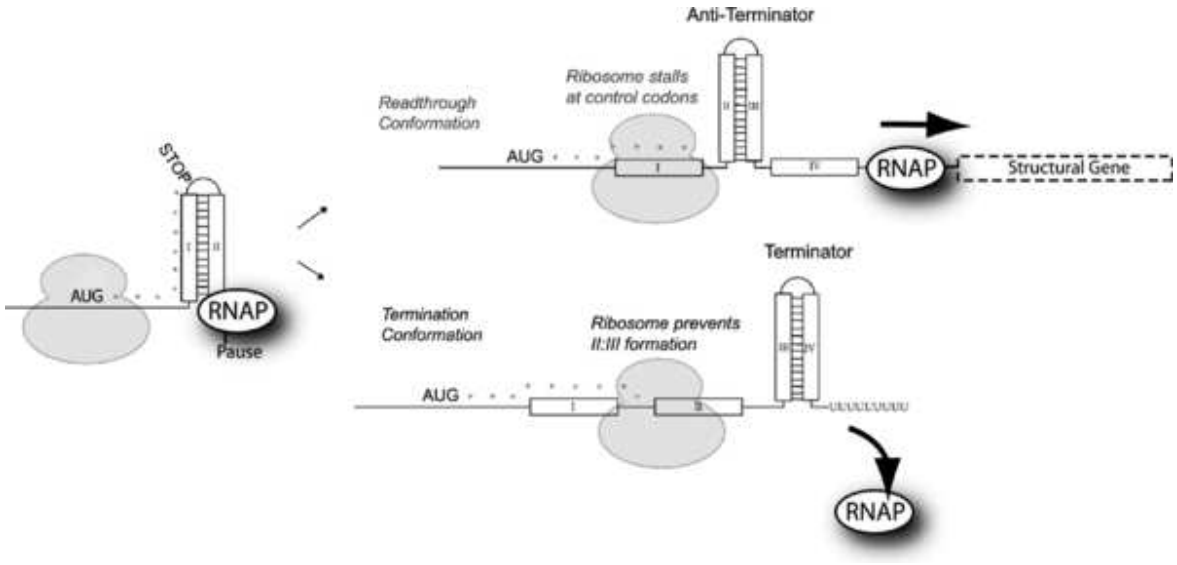

Fig. 3. Starting point and possible outcomes of the attenuation race at *E.coli*'s *trp* operon. *Initial conformation (left)*: RNAP is paused by the pause hairpin, awaiting to be released by the ribosome's next step. *Readthrough conformation*: when tryptophan supply is low, the ribosome stalls on the control codons, the anti-terminator hairpin forms and transcription continues into the operon. *Termination conformation*: when tryptophan supply is high, the ribosome rapidly translates over the control codons. Before it unbinds from the mRNA, the terminator hairpin forms and transcription aborts. Figure reproduced with permission from Elf and Ehrenberg (2005) [11].

13^{th} codon, and until it dissociates from the stop codon 15, the ribosome's footprint masks S_2, which prevents the anti-terminator [29]. The ribosome's unbinding delay from the stop codon is generally one second, which is a considerably long time scale, compared to all other reactions in the system. While S_2 remains blocked by the ribosome, RNAP continues transcription, it completes S_3 when reaching the 36^{th} DNA nucleotide, and S_4 at the 47^{th} DNA nucleotide. The terminator hairpin then forms and transcription aborts – this is the *termination configuration*.

Basal read-through due to premature ribosome release. A third possible outcome of the race is not covered by Fig. 3. When tryptophan supply is high, the ribosome occasionally dissociates from the stop codon sooner than expected. In that case S_3 can already have been transcribed, but S_4 not yet. Hence S_1, S_2 and S_3 are available at the same time. With equal probability, either the pause hairpin or the anti-terminator forms, and in case of the latter, transcription continues. This *basal read-through* of the operon has been experimentally observed for $10 - 15$ % of initiated transcripts when tryptophan is abundant [18].

3 Rule Schemas for Chemical Reactions

In this section we first provide a formal and minimal rule-based language tailored to our needs (Sect. 3.1). We define chemical reactions, that operate on multisets of complex molecules with attributes such as $\text{RNAP} \cdot \text{DNA}(23)$. Herein, the infix operator $\cdot$ indicates a complex between RNAP that is bound within a DNA sequence, more precisely at the position 23 stated by the attribute value of the DNA nucleotide. Other attributes of molecules could be the compartment of a molecule, or information on its states, for instance folding or binding state.

In Sect. 3.2, we present a language of *rules schemas*, that allows to define finite sets of chemical reactions in a compact manner. Rule schemas are like chemical reactions, except that attribute values are now extended to expressions with variables. All *variables are universally quantified over finite sets*, such that a rule schema defines a finite set of reactions. An example of a complex molecule is the term $\text{RNAP} \cdot \text{DNA}(x+1)$ where x is a variable with values in $\{0, \ldots, 50\}$. We introduce our language's stochastic semantics in Sect. 3.3.

As discussed in Sect. 3.4, more general ruled-based languages might have been used for our modeling study. The language in this paper is not intended as a contribution on its own, for the sake of its simplicity we however chose it for our presentation. Indeed, we relied on the software tool for another rule-based language to implement our models of Sect. 4 and 5.

3.1 Chemical Reactions

In order to define the syntax of attributed molecules, we fix a possibly infinite set of attribute values $\mathcal{C}$ and a finite set $\mathcal{N}$ of molecule names. We assume that each molecule name $N \in \mathcal{N}$ has a fixed arity $ar(N) \geq 0$, which specifies its number of attributes.

Molecules	$M \in Mol$	$::=$	$N(c_1, \ldots, c_n) \mid M_1 \cdot M_2$
Solutions	$S \in Sol$	$::=$	$M \mid S_1 \, , \, S_2$
Reactions	$S_1 \to_k S_2$		

Table 1. Chemical reactions where $N \in \mathcal{N}$, $c_1, \ldots, c_n \in \mathcal{C}$, $ar(N) = n$ and $k \in \mathbb{R}^+ \cup \{\infty\}$.

Expressions	$e \in Exp$	$::=$	$x \mid c \mid f(e_1, \ldots, e_n)$
Schematic molecules	$M \in SMol$	$::=$	$N(e_1, \ldots, e_n) \mid M_1 \cdot M_2$
Schematic solution	$S \in SSol$	$::=$	$M \mid S_1 \, , \, S_2$
Rule schema	$\forall x_1 \in D_1 \ldots \forall x_n \in D_n. \; S_1 \to_k S_2$		
	where $\mathcal{V}(S_1) \cup \mathcal{V}(S_2) \subseteq \{x_1, \ldots, x_n\}$		

Table 2. Rule schemas where $x \in \mathcal{V}$, $c \in \mathcal{C}$, $f \in \mathcal{F}$, $e_1, \ldots, e_n \in Exp$, $N \in \mathcal{N}$, $ar(N) = n$, D_1, …, $D_n \subseteq \mathcal{C}$ are finite sets, and $k \in \mathbb{R}^+ \cup \{\infty\}$.

A *molecule* M, defined in Table 1, is a complex of attributed molecules. We write $M_1 \cdot M_2$ for the complex of M_1 and M_2. For instance, if $\text{RNAP}, \text{DNA} \in \mathcal{N}$ and $47 \in \mathcal{C}$ then $\text{RNAP} \cdot \text{DNA}(47)$ is a molecule complex consisting of an RNAP that is bound to the DNA nucleotide at position 47. A *chemical solution* S is a multiset of molecules.

A *chemical reaction* is a rule that rewrites a solution S_1 into a solution S_2, it is assigned a possibly infinite stochastic rate constant $k \in \mathbb{R}^+ \cup \{\infty\}$. For instance, the following reaction states that an RNAP bound to the DNA nucleotide at position 23 may advance to the DNA nucleotide at position 24. The speed of this reaction is 50 sec^{-1}:

$$\text{RNAP} \cdot \text{DNA}(23) \, , \, \text{DNA}(24) \;\to_{50}\; \text{DNA}(23) \, , \, \text{RNAP} \cdot \text{DNA}(24)$$

In order to represent transcription, one would need many similar rules for the many other DNA nucleotides with different positions. This motivates the introduction of rule schemas, that allow to define such sets of chemical reactions in a compact manner.

3.2 Rule Schemas

In order to define rule schemas for chemical reactions, we introduce *variables* x for attribute values and *expressions* such as $x + 1$, in order to compute corresponding attribute values. By *universal quantification over a finite set*, we generalize the above chemical reaction to the following rule schema:

$$\begin{array}{l} \forall x \in \{1, \ldots, 49\}. \\ \text{RNAP} \cdot \text{DNA}(x) \, , \, \text{DNA}(x+1) \to_{50} \text{DNA}(x) \, , \, \text{RNAP} \cdot \text{DNA}(x+1) \end{array} \qquad (0)$$

We thus need a set $\mathcal{V}$ of variables that are ranged over by x, and a finite set $\mathcal{F}$ of function symbols $f \in \mathcal{F}$ with arities $ar(f) \geq 0$. Furthermore, we assume an interpretation $[\![f]\!] : \mathcal{C}^{ar(f)} \to \mathcal{C}$ for every $f \in \mathcal{F}$. An *expression* e with values in $\mathcal{C}$ is a term with the abstract syntax given in Table 2. In our modeling case studies, we will assume that symbol $+ \in \mathcal{F}$ of arity two is interpreted as addition on natural numbers. We freely use infix syntax as usual, i.e. we write $e_1 + e_2$ instead of $+(e_1, e_2)$. Given a variable assignment $\alpha : \mathcal{V} \to \mathcal{C}$, every expression $e \in \mathit{Exp}$ denotes an element $[\![e]\!]_\alpha \in \mathcal{C}$ that we define as follows:

$$[\![c]\!]_\alpha = c \qquad [\![x]\!]_\alpha = \alpha(x) \qquad [\![f(e_1, \ldots, e_n)]\!]_\alpha = [\![f]\!]([\![e_1]\!]_\alpha, \ldots, [\![e_n]\!]_\alpha)$$

A *schematic molecule* M is like a molecule, except we now allow for expressions in attribute positions rather than attribute values only. A *schematic solution* $S \in \mathit{SSol}$ is a multiset of schematic molecules. As usual, we write $\mathcal{V}(S)$ for the set of variables that occur in molecules of S. A *rule schema* specifies the domains of variables occurring in the schematic solutions of the rule by universal quantification over finite sets.

For every variable assignment $\alpha : \mathcal{V} \to \mathcal{C}$ that maps variables to values in their domain, we can instantiate the rule schema to finitely many reactions. A schematic molecule M is mapped to a molecule $[\![M]\!]_\alpha \in \mathit{Mol}$. Similarly, schematic solutions $S \in \mathit{SSol}$ get instantiated to solutions $[\![S]\!]_\alpha \in \mathit{Sol}$:

$$\begin{aligned} [\![N(e_1, \ldots, e_n)]\!]_\alpha &= N([\![e_1]\!]_\alpha, \ldots, [\![e_n]\!]_\alpha) \\ [\![M_1 \cdot M_2]\!]_\alpha &= [\![M_1]\!]_\alpha \cdot [\![M_2]\!]_\alpha \\ [\![S_1 \ , \ S_2]\!]_\alpha &= [\![S_1]\!]_\alpha \ , \ [\![S_2]\!]_\alpha \end{aligned}$$

A rule schema is instantiated to a set of chemical reactions, by enumerating the chemical reactions for all variable assignments licensed by the quantifiers:

$$\begin{aligned} &[\![\forall x_1 \in D_1 \ldots \forall x_n \in D_n.\ S_1 \to_k S_2]\!] = \\ &\qquad \{[\![S_1]\!]_\alpha \to_k [\![S_2]\!]_\alpha \mid \alpha : \mathcal{V} \to \mathcal{C},\ \alpha(x_1) \in D_1, \ldots, \alpha(x_n) \in D_n\} \end{aligned}$$

3.3 Stochastic Semantics and Simulation

For the sake of completeness, we recall the stochastic semantics of chemical reactions and how to use them for the stochastic semantics with Gillespie's algorithm. This underlines that our biological modeling case studies are indeed expressed in a formal modeling language.

The semantics of a set of chemical reactions is a continuous time Markov chain (CTMC). Note that, for modeling convenience, we allow infinite rate constant ∞. Chemical reactions with infinite rates always have the highest priority and are executed immediately, that is without time delay. Such *extended* CTMCs with infinite rate constants can actually be converted to regular CTMCs by elimination of immediate transitions, while preserving sojourn time (i.e. how long the Markov chain stays in a given state) and probability transitions (that is, given a current state, the probability to make a transition to another given state)[4].

[4] For such an elimination procedure, see [22] and references therein.

$$\frac{L \subseteq \{1,\ldots,n\} \qquad \oplus_{i\in L} M_i \equiv S \qquad S \rightarrow_k S'}{\oplus_{i=1}^{n} M_i \xrightarrow[L]{k} S' \ , \ \oplus_{i\notin L} M_i}$$

$$\frac{r = \sum_{\{(L,k) \mid S \xrightarrow[L]{k} S_1 \equiv S'\}} k \qquad \neg\exists L \exists S''.S \xrightarrow[L]{\infty} S''}{S \xrightarrow{r} S'}$$

$$\frac{n = \sharp\{L \mid S \rightarrow S_1 \equiv S'\} \qquad m = \sharp\{L \mid S \rightarrow S_2\}}{S \xrightarrow{\infty(n/m)} S'}$$

Table 3. Stochastic semantics of chemical reactions with finite and infinite rate constants.

The states of the extended CTMCs are congruence classes $[S]_{\equiv}$ of chemical solutions S with respect to the least congruence relation $\equiv$ that makes complexation and summation associative and commutative:

$$M_1 \cdot M_2 \equiv M_2 \cdot M_1 \qquad (M_1 \cdot M_2) \cdot M_3 \equiv M_1 \cdot (M_2 \cdot M_3)$$
$$S_1 \ , \ S_2 \equiv S_2 \ , \ S_1 \qquad (S_1 \ , \ S_2) \ , \ S_3 \equiv S_1 \ , \ (S_2 \ , \ S_3)$$

In Table 3, we introduce transitions $S \xrightarrow[L]{k} S'$ stating that S can be reduced to S' by applying a chemical reaction with rate constant $k \in \mathbb{R}^+ \cup \{\infty\}$ to the subset of molecules in S with positions in L. Positions are the indices in multisets such as $M_1 \ , \ \ldots \ , \ M_n$ that we also write as $\oplus_{i=1}^{n} M_i$. We next introduce two transitions

- $S \xrightarrow{r} S'$, where $r \in \mathbb{R}^+$ sums up all rate constants of chemical reactions reducing S to S', as many times as they apply for some index set L, provided that no immediate reaction can occur,
- $S \xrightarrow{\infty(r)} S'$ where the corresponding probability is $r = n/m$. The number of occurrences of immediate reactions leading from S to a solution congruent to S' is n, and the number of all occurrences of immediate reactions starting from S is m.

Such transitions are invariant under structural congruence, i.e. for all $S_1 \equiv S_1'$ and $S_2 \equiv S_2'$ it holds that $S_1 \xrightarrow{r} S_2$ (resp. $S_1 \xrightarrow{\infty(r)} S_2$) if and only if $S_1' \xrightarrow{r} S_2'$ (resp. $S_1' \xrightarrow{\infty(r)} S_2'$). We can thus define $[S]_{\equiv} \xrightarrow{r} [S']_{\equiv}$ by $S \xrightarrow{r} S'$ and $[S]_{\equiv} \xrightarrow{\infty(r)} [S']_{\equiv}$ by $S \xrightarrow{\infty(r)} S'$ as the transitions of the extended CTMC.

Gillespie's algorithm for stochastic simulation takes as input a finite set of chemical reactions and a chemical solution S. If reactions with infinite rate constants are applicable, it computes n and m as defined above for each immediately reachable solution S', and returns such an S' with probability n/m jointly with a

null time delay. Otherwise, it computes the overall rate of all possible transitions $R = \sum_{\{r|S \xrightarrow{r} S_1\}} r$, returns with probability r/R a solution S_1 with transition $S \xrightarrow{r} S_1$ jointly with a time delay drawn randomly from the exponential distribution with rate r.

3.4 Language Design Choices and Related Rule-based Languages

Models with rule schemas are more compact than if only simple reactions were used, thus easier to read. Attributed molecules and expressions that manipulate them were introduced, in the context of biological modeling languages, in [17]. Note that even if rule schemas could be defined solely by means of variables, function symbols allow a better control and precision of the collection of reactions that are generated. For example, without function symbols, we would need to resort to *name sharing* to represent DNA sequences[5]. Each DNA nucleotide would bear two parameters, one referring to its predecessor, the other to its successor. Given link names $\{\ell_0, \ldots, \ell_{50}\}$, our previous rule (0) on page 9 reads as

$$\forall x, y, z \in \{\ell_0, \ldots, \ell_{50}\}.\\ \text{RNAP} \cdot \text{DNA}(x, y)\ ,\ \text{DNA}(y, z) \rightarrow_{50} \text{DNA}(x, y)\ ,\ \text{RNAP} \cdot \text{DNA}(y, z)$$

Then starting from a DNA sequence $\text{DNA}(\ell_0, \ell_1), \text{DNA}(\ell_1, \ell_2), \ldots, \text{DNA}(\ell_{49}, \ell_{50})$, this rule schema instantiates into more ground rules than needed. For example, the rule $\text{RNAP} \cdot \text{DNA}(\ell_1, \ell_{10})\ ,\ \text{DNA}(\ell_{10}, \ell_{45}) \rightarrow_{50} \text{DNA}(\ell_1, \ell_{10})\ ,\ \text{RNAP} \cdot \text{DNA}(\ell_{10}, \ell_{45})$ is a meaningless instance of the above schema. Indeed, it is never applicable if the above DNA sequence is not modified as it is expected in a correct model.

All models written with our rule schemas can be compiled, by instantiation, to finite collections of simple and formally well-defined chemical reactions. Although reactions do not define a Turing-complete language [5], their expressiveness is sufficient for our purposes. Furthermore, such collections of reactions are supported by standard tools for stochastic simulation such as Dizzy [27] or the rule-based language BioCham [6].

Alternative rule-based languages with higher expressiveness are Turing complete, e.g. the graph rewriting language Kappa [9], BioNetGen [16], and bigraphs [21]. Their pattern based graph rewriting rules resemble schemas, but their semantics is not based on instantiations to ground rules. They rather directly apply to arbitrary subgraphs satisfying the pattern. In contrast to our approach, such patterns may describe infinitely many reactions. Furthermore, stochastic simulation is possible without inferring all those reactions on before hand. This generation process is uncritical in the present paper, since the overall number of reactions remains small, but is the bottleneck in other applications, where it grows exponentially [8,35]. Another promising language is LBS [24]. Its general purpose semantics allows for translations to different concrete semantics such as ODEs and CTMCs. LBS also features compact description of reactions with yet

[5] This is actually how we implemented our model of Sect. 4 in the Kappa Factory.

$\forall i \in \{0,\ldots,\text{n-1}\}.\textsc{Rnap} \cdot \textsc{Dna}_i \ , \ \textsc{Dna}_{i+1} \rightarrow_{e_1} \textsc{Dna}_i \ , \ \textsc{Rnap} \cdot \textsc{Dna}_{i+1}$	(1)
$\textsc{Rnap} \cdot \textsc{Dna}_n \rightarrow_{\infty} \textsc{Rnap} \ , \ \textsc{Dna}_n$	(2)
$\forall i \in \{0,\ldots,\text{m-1}\}.$ $\text{Ribosome} \cdot \textsc{mRna}_i \ , \ \textsc{mRna}_{i+1} \rightarrow_{e_2} \textsc{mRna}_i \ , \ \text{Ribosome} \cdot \textsc{mRna}_{i+1}$	(3)
$\text{Ribosome} \cdot \textsc{mRna}_m \rightarrow_{\infty} \text{Ribosome} \ , \ \textsc{mRna}_m$	(4)

Table 4. Rules for n transcription steps, in race with m translation steps.

another approach by means of parameterized modules, species expressions and "non-deterministic" species. These formal rule-based languages were designed and used so far to tackle protein-protein interactions that occur in cellular signaling such as metabolic pathways. In contrast to this, our rule-based model deals with a fine-grained mechanism of gene regulation.

4 Hyper-Sensitivity of Multi-Step Races

In this section we illustrate rule schemas for chemical reactions with a simple yet interesting example, borrowed from Elf and Ehrenberg (2005) [11]. Abstracting away from its detailed control by mRNA hairpins, transcriptional attenuation boils down to a plain race between the two competing multi-step processes of transcription and translation. As intuition easily confirms, the probability that transcription wins the race decreases as the ribosome speeds up, and vice versa. We present two rule-based models for this multi-step race.

The *basic model* of Sect. 4.1 investigates the hyper-sensitivity of attenuation depending on the respective number of transcription versus translation steps (n vs m). Using Elf and Ehrenberg's rate constants for transcription and translation, we reproduce the results of Fig. 2 in [11].

In Sect. 4.2 we enrich our basic model by what we call *concurrent elongation*. An additional parameter m_0 denotes the number of codons contained in the initial solution, the remaining codons are dynamically spawn by the RNAP at simulation time. We show the impact of this additional level of concurrency, with respect to the attenuation race, through simulation.

4.1 Basic Model of Transcription and Translation

Elf and Ehrenberg demonstrated that the relative change in the probability that transcription wins the race can be much sharper, than the relative change in the ribosome's speed. As our work confirms, this *hyper-sensitivity* of attenuation is determined by the number of transcription steps (n) versus translation steps (m). We give a basic rule-based model that allows to reproduce the results of Elf and Ehrenberg. As we believe, our framework is easier to understand and less prone

to error than the master equation approach, while compact and mathematically well-founded.

Model. The following initial solution describes the starting point of the multi-step race, where the RNAP and ribosome are bound to the first positions of DNA and mRNA respectively:

$$\text{RNAP} \cdot \text{DNA}_0 \ , \ \oplus_{i=1}^{n} \text{DNA}_i \ , \ \text{Ribosome} \cdot \text{mRNA}_0 \ , \ \oplus_{i=1}^{m} \text{mRNA}_i$$

We use the following notational conventions. Molecule names are $\mathcal{N} = \{\text{RNAP}, \text{mRNA}, \text{DNA}, \text{Ribosome}\}$, attribute values $\mathcal{C} = \mathbb{N}_0$, function symbols $\mathcal{F} = \{+\}$, variables $\mathcal{V} = \{i\}$, and value parameters $n, m \in \mathcal{C}$. Because our model's attributed molecules bear only few arguments, for the sake of presentation we slightly differ from the formal syntax introduced in Sect. 3. We write attributes as indices for molecule names, instead of parenthesizing them, e.g. DNA_i instead of $\text{DNA}(i)$. Moreover we write $\oplus_{i=1}^{n} \text{DNA}_i$ instead of $\text{DNA}(1), \ldots, \text{DNA}(n)$. Finally, we emphasize that each mRNA_i denotes one *codon*, which biologically speaking is a sequence of three individual mRNA nucleotides, that the ribosome reads out in one step.

Our model's rule schemas are listed in Table 4. Rule 1 for the n steps of the transcribing RNAP from one DNA nucleotide to the next remains as in Sect. 3.2, where it was the running example. The translation rule 3 is analogous and reflects the ribosome's m steps over codons. The remaining rules 2 and 4 model the dissociation from the respectively last positions of DNA and mRNA. Note that, bearing rate ∞, dissociation occurs without advance of the simulation clock. Hence it does not quantitatively affect our simulation results compared to the model of Elf and Ehrenberg, that does not include dissociation. In order to incorporate the control conditions at the *trp* operon, we will refine the dissociation rules in Sect. 5.

Simulation. The plot reporting our simulation results in Fig. 4 is organized as follows. The y-axis gives the probability that RNAP wins the race, on a scale between zero and one. It corresponds to the proportion of simulations in which RNAP dissociates before the ribosome does. The x-axis reports the translation rate on a logarithmic scale, that we vary from 0.01 to 100 codons per second in our simulations with the Kappa Factory [8].

The three models that each contribute one curve in the plot only differ in their numbers of translation (m) versus transcription (n) steps. We combined ($m = 1$ vs $n = 1$), ($m = 1$ vs $n = 50$), and ($m = 10$ vs $n = 50$).

Let us compare the sensitivity of these three models. When $m = 1$ and $n = 1$, the probability curve decreases gently, already showing some non-linearity. Increasing the number of transcription steps to $n = 50$ steepens the curve, i.e. increases the sensitivity. The transition becomes even sharper when the number of translation steps reaches higher values ($m = 10$, $n = 50$). Such values hold for systems where, unlike at *E.coli*'s *trp* operon, attenuation is the sole control mechanism [19]. It is worthwhile pointing out that in this model each of the m

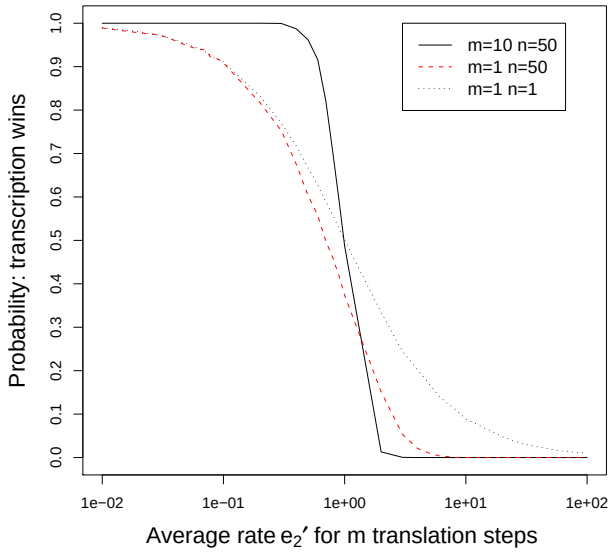

Fig. 4. Probability that transcription wins in the basic model, as a function of the average translation rate e_2', for different numbers of translation (m) versus transcription (n) steps.

translation steps potentially slows down the ribosome's advance, while at *E.coli*'s *trp* operon, only 2 in 9 steps do.

Our rate constants are calculated as in [11], to ensure the outcomes of the three races are comparable. We keep the total time to perform the series of n transcription steps constant, such that $1/e_1' = 1$ sec. Thus, the rate constant for one individual transcription step out of n is $e_1 = n \cdot e_1'$. For one translation step the rate constant is $e_2 = m \cdot e_2'$. Hereby e_2' is the average rate for m translation steps, which varies logarithmically between 0.01 and 100.

As the next section will show, our model can smoothly be extended by additional concurrent issues, that are more difficult to handle within the master equation approach.

4.2 Concurrent Elongation of mRNA

In the basic model, the multi-step race was represented by a ribosome and an RNAP advancing along two independent strands of mRNA and DNA. Here we add what we call *concurrent elongation* to the multi-step race. The idea is to reflect that RNAP still elongates a transcript when the ribosome starts translating its older end. Translation can now become limited by the slower transcription: the ribosome can only translate those codons that have previously been produced by the RNAP. Our simulation results demonstrate that the outcome of the race depends on the length of the initially available mRNA.

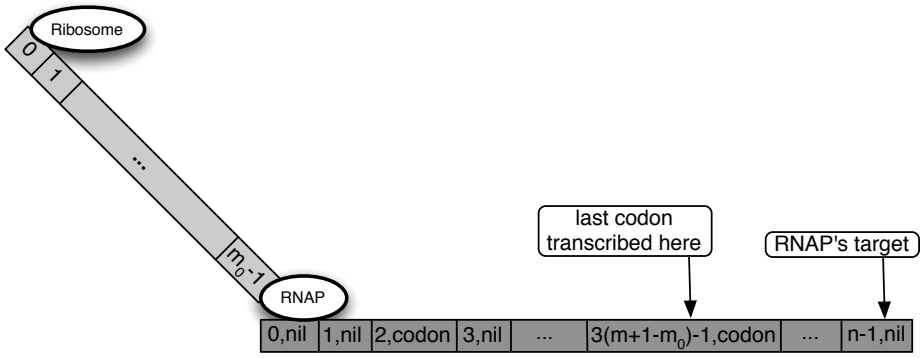

Fig. 5. General initial solution for the *concurrent elongation* model, containing an mRNA of length m_0 and a DNA of length n. The DNA is composed such that upon simulation, every three steps of the RNAP one new codon is spawn; the final solution contains m codons.

Model. Compared to the basic model, we now explicitly elongate the previously available mRNA in each transcription step. As Fig. 5 illustrates, a portion of the mRNA is available to the ribosome from the beginning of the race. In the basic model the parameters n and m denoted the respective lengths of the DNA and mRNA sequences for the attenuation race. Here the transcript dynamically grows from an initial length (for which we introduce the new parameter m_0) to its final length m.

We use two more function symbols for integer arithmetics than previously, $\mathcal{F} = \{+, -, /\}$, attribute values $\mathcal{C} = \mathbb{N}_0 \cup \{\text{codon}, \text{nil}\}$, the previous molecule names $\mathcal{N} = \{\text{RNAP}, \text{mRNA}, \text{DNA}, \text{Ribosome}\}$, and variables $\mathcal{V} = \{i, x\}$. DNA molecules now come with a second attribute with values in $\{\text{codon}, \text{nil}\}$, noted as an upper index and with the following meaning. When RNAP leaves the i^{th} nucleotide $\text{DNA}_i^{\text{codon}}$, it produces a new codon. As opposed to this, $\text{DNA}_i^{\text{nil}}$ indicates that no new codon is spawn when RNAP passes from the i^{th} nucleotide to the next.

The choice of an appropriate *initial solution* is crucial to the proper functioning of this model, because we want the polymerase to spawn one new codon every three DNA nucleotides. Assuming that RNAP is initially bound to DNA_0 the solution must be such that, for $i \bmod 3 = 2$, nucleotides are of the form $\text{DNA}_i^{\text{codon}}$, and otherwise $\text{DNA}_i^{\text{nil}}$. Correspondingly the first two nucleotides must be $\text{DNA}_0^{\text{nil}}$ and $\text{DNA}_1^{\text{nil}}$, followed by the nucleotide $\text{DNA}_2^{\text{codon}}$, and so forth respecting the pattern nil, nil, codon. If only one codon is part of the initial solution ($m_0 = 1$) we obtain:

$$\begin{aligned}&\text{Ribosome} \cdot \text{mRNA}_0 \ , \\ &\text{RNAP} \cdot \text{DNA}_0^{\text{nil}} \ , \ \text{DNA}_1^{\text{nil}} \ , \ \text{DNA}_2^{\text{codon}} \ , \\ &\text{DNA}_3^{\text{nil}} \ , \ \text{DNA}_4^{\text{nil}} \ , \ \text{DNA}_5^{\text{codon}} \ , \ \ldots \ , \ \text{DNA}_n^{\text{nil}}\end{aligned}$$

$$\forall i \in \{0,\ldots,n-1\}.\forall x \in \{\text{codon},\text{nil}\}.\quad \text{RNAP}\cdot\text{DNA}_i^{\text{nil}}\ ,\ \text{DNA}_{i+1}^{x} \rightarrow_{e_1} \text{DNA}_i^{\text{nil}}\ ,\ \text{RNAP}\cdot\text{DNA}_{i+1}^{x} \quad (1)$$

$$\forall i \in \{0,\ldots,n-1\}.\quad \text{RNAP}\cdot\text{DNA}_i^{\text{codon}},\text{DNA}_{i+1}^{\text{nil}} \rightarrow_{e_1} \text{DNA}_i^{\text{codon}},\text{RNAP}\cdot\text{DNA}_{i+1}^{\text{nil}},\text{mRNA}_{m_0-1+(i+1)/3} \quad (2)$$

$$\text{RNAP}\cdot\text{DNA}_n^{nil} \rightarrow_{\infty} \text{RNAP}\ ,\ \text{DNA}_n^{nil} \quad (3)$$

$$\forall i \in \{0,\ldots,m-1\}.\quad \text{Ribosome}\cdot\text{mRNA}_i\ ,\ \text{mRNA}_{i+1} \rightarrow_{e_2} \text{mRNA}_i\ ,\ \text{Ribosome}\cdot\text{mRNA}_{i+1} \quad (4)$$

$$\text{Ribosome}\cdot\text{mRNA}_m \rightarrow_{\infty} \text{Ribosome}\ ,\ \text{mRNA}_m \quad (5)$$

Table 5. Rules for *concurrent elongation* (n steps), where the transcribing RNAP adds one new codon to the solution every three DNA nucleotides.

For the sake of simplicity we do not show the DNA position corresponding to mRNA_m. When $m_0 = m+1$, the initial solution reduces to that of Sect. 4.1:

$$\begin{array}{l}\text{Ribosome}\cdot\text{mRNA}_0\ ,\ \oplus_{i=1}^{m}\text{mRNA}_i\ ,\\ \text{RNAP}\cdot\text{DNA}_0^{\text{nil}}\ ,\ \text{DNA}_1^{\text{nil}}\ ,\ \text{DNA}_2^{\text{codon}}\ ,\\ \text{DNA}_3^{\text{nil}}\ ,\ \text{DNA}_4^{\text{nil}}\ ,\ \text{DNA}_5^{\text{codon}}\ ,\ \ldots\ ,\ \text{DNA}_n^{\text{nil}}\end{array}$$

Figure 5 illustrates the general case. In addition to the constraint on DNA nucleotide alternation, we assume that the initial solution contains m_0 codons, that the rule set will lead to the dynamic supply of additional $m - m_0$ codons, such that the final solution shall contain $m+1$ codons (allowing for m translation steps), and that $1 \leq m_0 \leq m < \frac{1}{3}n$. The ribosome's target codon mRNA_m corresponds to the DNA position $3\cdot(m-m_0+1)-1$. Beyond this, we assume that RNAP eventually reaches its own target, the n^{th} position of DNA, without injecting additional codons to the solution.

Table 5 lists our rule schemas. The rule 1 for one step of the RNAP, in which no codon is produced, resembles rule 1 of Sect. 4.1. It applies when leaving nucleotides of the form $\text{DNA}_i^{\text{nil}}$, whether or not the step leaving the *next* nucleotide yields a codon. Hence the quantification over $x \in \{\text{codon},\text{nil}\}$ for $\text{DNA}_{i+1}^{\text{x}}$.

The complementary rule 2 injects a new codon into the solution when the RNAP leaves a nucleotide of the form $\text{DNA}_i^{\text{codon}}$. The new codon's index is calculated from the current DNA position i and the initially available number of codons by the arithmetic expression $m_0 - 1 + (i+1)/3$. By doing so, we ensure that DNA_2 yields mRNA_{m_0}, DNA_5 yields mRNA_{m_0+1}, etc. up to the ribosome's target mRNA_m.

The rule for the RNAP's dissociation from DNA (3) only marginally differs from that of the previous subsection (in that the nucleotide bears the second attribute nil), and the ribosome advance and release rules (4 and 5) remain just the same.

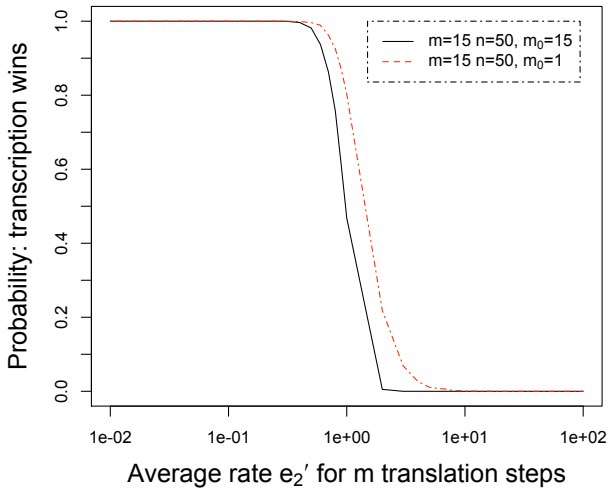

Fig. 6. Probability that transcription wins in the concurrent elongation model, as a function of the average translation rate e_2', for different numbers of initially available codons ($m_0 = 15$ vs $m_0 = 1$), but the same number of transcription steps ($n = 50$) and translation steps ($m = 15$).

Simulation. We simulated our *concurrent elongation* model within the Kappa factory with several combinations of m_0, m and n. Figure 6 shows the outcome of the race distinguishing whether only one codon is initially present ($m_0 = 1$), or all ($m_0 = m$), for the same number of translation ($m = 15$) and transcription steps ($n = 50$).

When all codons are contained in the initial solution ($m_0 = 15$), the simulation results reduce to those of the basic model, whereas for $m_0 = 1$, the simulation curve shifts to the right, meaning that the probability that transcription wins the race increases. Indeed for each translation step, the ribosome's advance is potentially limited by the polymerase, that needs to add a further codon to the mRNA. Hence, even if translation is efficient, the polymerase wins more often than for $m_0 = 15$. In our simulations we observed a lesser shift for $m_0 = 10$, not included in the plot.

Our simulations underline that the outcome of the multi-step race is parameterized not only by the n transcription steps and the m translation steps, but also by the number m_0 of initially available codons. This last parameter only appears when the model integrates concurrent elongation. We can now summarize our analysis of the multi-step race parameterized by m, n and m_0, in terms of the shape of the curve that represents the probability that the polymerase wins the race:

- As pointed out by Elf and Ehrenberg [11], the ratio of m to n determines the curve's slope. They are the key parameters of the hyper-sensitivity of ribosome-dependent transcriptional attenuation.
- Varying m_0 shifts the curve. The polymerase's chance to win increases with m_0, when m and n remain fixed, because m_0 constrains the ribosome's advance along mRNA. As we observed, the shift increases with the difference between m and m_0.

Incorporating concurrent elongation into our model was facilitated by our rule-based approach with arithmetic. It would have been more difficult with probability functions. In a model that includes concurrent elongation, the positions of the ribosome and the RNAP are not independent. The advance of the former is limited by that of the latter. This point was not considered in [11].

5 Modeling Transcriptional Attenuation

This section presents our rule-based model of ribosome-mediated transcriptional attenuation at *E.coli*'s tryptophan operon. It refines our basic model of Sect. 4.1 in several points. The messenger RNA's representation dynamically grows while we simulate RNAP's advance, similarly as in the concurrent elongation model. But whereas in Sect. 4.2, we only used individual codons as building blocks of the transcript, the attenuation model also features mRNA segments as a whole. Explicit representations of S_1, S_2, S_3, and S_4 allow us to smoothly cover the dynamics of secondary structure formation, and incorporate the regulatory impact of hairpins on transcription. We make one notable exception to our all-in-one representation of mRNA segments. Regarding S_1, we switch between two different abstraction levels depending on the context, either representing it as a whole, or enumerating its codon sequence ($\oplus_{i=10}^{14}\text{mRNA}_i$).

After introducing our attenuation model, we present simulation results in Sect. 5.2, and then explain the quantitative differences between our results and those of Elf and Ehrenberg [11] in Sect. 5.3.

5.1 Rule Schemas

Table 6 provides the rule schemas of our detailed attenuation model. The notational conventions are based on those of our basic model of Sect. 4.1. We use molecule names $\mathcal{N} = \{\text{RNAP}, \text{mRNA}, \text{DNA}, \text{Ribosome}, \text{S}\}$, where DNA nucleotides are *unary*, attribute values $\mathcal{C} = \mathbb{N}_0 \cup \{\text{fr}, \text{bl}, \text{hp}\}$, function symbols $\mathcal{F} = \{+\}$, and variables $\mathcal{V} = \{i, n, m, t, x\}$. Molecules with two attributes S_i^x represent segments of the mRNA leader. Their lower index $i \in \{1, 2, 3, 4\}$ denotes the segment's number, and the upper index x the segment's state which is among:

- *free* (fr): available for hairpin formation,
- *blocked* (bl): masked by the ribosome's footprint,
- *hairpin* (hp): complexed into a hairpin with a neighboring segment. For instance, $\text{S}_1^{\text{hp}} \cdot \text{S}_2^{\text{hp}}$ denotes the pause hairpin.

The initial solution for our simulations reflects the starting configuration for the attenuation race, depicted in Fig. 3 on page 7:

$$\begin{array}{l} \text{RNAP} \cdot \text{DNA}_0 \ , \ \oplus_{i=1}^{50} \text{DNA}_i \ , \\ \text{Ribosome} \cdot \text{mRNA}_6 \ , \ \text{mRNA}_7 \ , \ \text{mRNA}_8 \ , \ \text{mRNA}_9 \ , \\ \text{S}_1^{\text{hp}} \cdot \text{S}_2^{\text{hp}} \ , \ \text{mRNA}_{15} \end{array} \tag{0}$$

RNAP has transcribed the leader up to and including S_1 and S_2, that are paired into the pause hairpin, and is paused on the zero-th DNA nucleotide, that is followed by a sequence of 50. The ribosome has initiated translation and is located on the 6th codon of the transcript leader. We explicitly render the codons 6 to 9, which precede the segment S_1, and the stop codon 15, that is located between the segments S_1 and S_2. In contrast, we do not render the codons preceding 6, since they do not matter to the attenuation race, and for the same reason we will not provide rules for the initiation of transcription and translation.

Hairpin formation is covered by rule schema 1, be it for the pause hairpin, the anti-terminator or the terminator. Because hairpin formation occurs on a much faster time scale than any other reaction, we approximate it with an infinite rate constant.

Translation rules (schemas 2 to 7 in Table 6). Rule schema 2 covers the bulk of translation steps, that do not have side effects, nor depend on tryptophan availability or other side conditions. It bears the reaction rate constant $e_2 = 15s^{-1}$, i.e. the ribosome makes 15 steps over mRNA per second, in average. Rule schema 3 deals with the ribosome's step over the tryptophan codons within the leader, i.e. the control codons 10 and 11, where the distinct elongation rate constant e_3 holds. We will vary e_3 within $]0, 15]s^{-1}$ in our simulations, while e_2 remains fixed.

Starting from our initial solution (the above equation 0) the next important event is *melting the pause loop* $\text{S}_1^{\text{hp}} \cdot \text{S}_2^{\text{hp}}$, as the ribosome steps from mRNA_6 to mRNA_7. Two points are worthwhile noting in rule 4's right part. First, obviously since the pause loop is melt, S_2's state becomes *free* - and one could similarly expect a stage change at S_1. But second and more importantly, instead of switching S_1's state, we pass from the abstraction of the segment as a whole, to the enumeration of the codons $\oplus_{i=10}^{14}\text{mRNA}_i$ that make it up. The sequence enumeration remains part of the solution as long as the ribosome's footprint partially covers the first segment, i.e. until it dissociates from the stop codon. This implicitly sequesters S_1 from hairpin formation – which would instantaneously occur through schema 1 if both S_1 and S_2 were around and in their *free* state.

For the ribosome's step from mRNA_{12} to mRNA_{13}, we introduce two rules with distinct preconditions. The common result of both is to reflect that the second segment gets masked by the ribosome's footprint, i.e. both rules produce S_2^{bl}. When S_2 is initially free, rule 5 applies. Otherwise, S_2 is paired into the anti-terminator hairpin, and rule 6 handles its melting through the ribosome's advance. When the ribosome dissociates from the stop codon (rule 7), S_1 reassembles, and S_2 unblocks.

$\forall i \in \{1,2,3\}.\mathrm{S}_i^{\mathrm{fr}}\ ,\ \mathrm{S}_{i+1}^{\mathrm{fr}} \rightarrow_{\infty} \mathrm{S}_i^{\mathrm{hp}} \cdot \mathrm{S}_{i+1}^{\mathrm{hp}}$	(1)
$\forall m \in \{7,8,9,13,14\}.\mathrm{Ribosome} \cdot \mathrm{mRNA}_m\ ,\ \mathrm{mRNA}_{m+1} \rightarrow_{e_2} \mathrm{mRNA}_m\ ,\ \mathrm{Ribosome} \cdot \mathrm{mRNA}_{m+1}$	(2)
$\forall t \in \{10,11\}.\mathrm{Ribosome} \cdot \mathrm{mRNA}_t\ ,\ \mathrm{mRNA}_{t+1} \rightarrow_{e_3} \mathrm{mRNA}_t\ ,\ \mathrm{Ribosome} \cdot \mathrm{mRNA}_{t+1}$	(3)
$\mathrm{Ribosome} \cdot \mathrm{mRNA}_6\ ,\ \mathrm{mRNA}_7\ ,\ \mathrm{S}_1^{\mathrm{hp}} \cdot \mathrm{S}_2^{\mathrm{hp}} \rightarrow_{e_2} \mathrm{mRNA}_6\ ,\ \mathrm{Ribosome} \cdot \mathrm{mRNA}_7\ ,\ \oplus_{i=10}^{14}\mathrm{mRNA}_i\ ,\ \mathrm{S}_2^{\mathrm{fr}}$	(4)
$\mathrm{Ribosome} \cdot \mathrm{mRNA}_{12}\ ,\ \mathrm{mRNA}_{13}\ ,\ \mathrm{S}_2^{\mathrm{fr}} \rightarrow_{e_2} \mathrm{mRNA}_{12}\ ,\ \mathrm{Ribosome} \cdot \mathrm{mRNA}_{13}\ ,\ \mathrm{S}_2^{\mathrm{bl}}$	(5)
$\mathrm{Ribosome} \cdot \mathrm{mRNA}_{12}\ ,\ \mathrm{mRNA}_{13}\ ,\ \mathrm{S}_2^{\mathrm{hp}} \cdot \mathrm{S}_3^{\mathrm{hp}} \rightarrow_{e_2} \mathrm{mRNA}_{12}\ ,\ \mathrm{Ribosome} \cdot \mathrm{mRNA}_{13}\ ,\ \mathrm{S}_2^{\mathrm{bl}}\ ,\ \mathrm{S}_3^{\mathrm{fr}}$	(6)
$\mathrm{Ribosome} \cdot \mathrm{mRNA}_{15}\ ,\ \oplus_{i=10}^{14}\mathrm{mRNA}_i\ ,\ \mathrm{S}_2^{\mathrm{bl}} \rightarrow_{d} \mathrm{Ribosome}\ ,\ \mathrm{mRNA}_{15}\ ,\ \mathrm{S}_1^{\mathrm{fr}}\ ,\ \mathrm{S}_2^{\mathrm{fr}}$	(7)
$\forall n \in \{1,\ldots,49\} \setminus \{35,46,47\}.\mathrm{RNAP} \cdot \mathrm{DNA}_n\ ,\ \mathrm{DNA}_{n+1} \rightarrow_{e_1} \mathrm{DNA}_n\ ,\ \mathrm{RNAP} \cdot \mathrm{DNA}_{n+1}$	(8)
$\forall x \in \{\mathrm{fr},\mathrm{bl}\}.\mathrm{RNAP} \cdot \mathrm{DNA}_0\ ,\ \mathrm{DNA}_1\ ,\ \mathrm{S}_2^x \rightarrow_{e_1} \mathrm{DNA}_0\ ,\ \mathrm{RNAP} \cdot \mathrm{DNA}_1\ ,\ \mathrm{S}_2^x$	(9)
$\mathrm{RNAP} \cdot \mathrm{DNA}_{35}\ ,\ \mathrm{DNA}_{36} \rightarrow_{e_1} \mathrm{DNA}_{35}\ ,\ \mathrm{RNAP} \cdot \mathrm{DNA}_{36}\ ,\ \mathrm{S}_3^{\mathrm{fr}}$	(10)
$\mathrm{RNAP} \cdot \mathrm{DNA}_{46}\ ,\ \mathrm{DNA}_{47} \rightarrow_{e_1} \mathrm{DNA}_{46}\ ,\ \mathrm{RNAP} \cdot \mathrm{DNA}_{47}\ ,\ \mathrm{S}_4^{\mathrm{fr}}$	(11)
$\mathrm{RNAP} \cdot \mathrm{DNA}_{47}\ ,\ \mathrm{S}_3^{\mathrm{hp}} \cdot \mathrm{S}_4^{\mathrm{hp}} \rightarrow_{e_1} \mathrm{DNA}_{47}\ ,\ \mathrm{RNAP}\ ,\ \mathrm{S}_3^{\mathrm{hp}} \cdot \mathrm{S}_4^{\mathrm{hp}}$	(12)
$\mathrm{RNAP} \cdot \mathrm{DNA}_{47}\ ,\ \mathrm{DNA}_{48}\ ,\ \mathrm{S}_2^{\mathrm{hp}} \cdot \mathrm{S}_3^{\mathrm{hp}} \rightarrow_{e_1} \mathrm{DNA}_{47}\ ,\ \mathrm{RNAP} \cdot \mathrm{DNA}_{48}\ ,\ \mathrm{S}_2^{\mathrm{hp}} \cdot \mathrm{S}_3^{\mathrm{hp}}$	(13)

Table 6. Rule schemas for hairpin formation (1), translation (2-7), and transcription (8-13).

Transcription rules. Rules 8 to 13 in Table 6 represent transcription. Rule schema 8 represents one step of RNAP in the simplest possible fashion, that was already discussed in Sect. 4.1 with the rate constant $e_1 = 50s^{-1}$. It applies to all DNA positions from 0 to 50 with a few exceptions that we discuss in the order they are applied, starting from our initial solution.

Transcription *resumes* at position DNA_0 (rule schema 9) after the pause hairpin has been disrupted. This is witnessed by S_2 being either free or blocked. Note that it would have been simpler to check the *absence* of the pause hairpin, but such negative tests are neither supported by the language used in this paper, nor by most current rule-based frameworks and tools, a notable exception to a certain extent is offered by [16].

The remaining rules deal with the creation of new mRNA segments, and (anti)termination of transcription. When RNAP steps over to DNA_{36}, the RNA segment S_3 is injected into the solution (reaction 10). S_4 follows at DNA_{47} (reaction 11). Transcription terminates on DNA_{47} provided there is a terminator hairpin, see $S_3^{\text{hp}} \cdot S_4^{\text{hp}}$ in reaction 12. If conversely the anti-terminator is present transcription proceeds to DNA_{48} (rule 13) and continues transcription into the operon.

Summary: global control by n-ary rules. Finally we summarize our use of n-ary rules, that separates into three categories. Such n-ary rules can not be rendered in an intelligible fashion within object-centric approaches limited to binary interactions, namely π-calculus based modeling languages.

In the *first category*, we check whether the current solution fulfills a certain *prerequisite*, e.g. contains a certain molecule, or a certain molecule in a specific state. The rules for abortion versus continuation of transcription (rules 12 and 13) depend on which hairpin is around, terminator or anti-terminator. Other prerequisites we check are unfortunately less intuitive: sometimes one would prefer to impose negative conditions on rule application, which however is neither supported by our language, nor most other current rule-based frameworks. For instance, rule 9 resumes transcription if the pause hairpin is absent, which is the case if S_2's state is free or blocked.

The *second category* are reactions that actually occur between two molecules, but entail the *modification* of a third. Examples are the rules for blocking S_2 as the ribosome proceeds to mRNA_{13}: rule 5 blocks S_2, rule 6 disrupts the anti-terminator hairpin at the same time, such that the states of both S_2 and S_3 change. The *third category* is the abstraction level switching for mRNA segments, that is assembling the first segment from the codons 10 to 14 in rule 7, versus splitting it in rule 4.

5.2 Simulation

Figure 7 plots the relative transcription frequency (y-axis) against the rate of *trp*-codon translation e_3 (x-axis). For each value of e_3 from 0 to 15 s^{-1} in steps of one, we performed 5000 Gillespie simulations of our model. Recall that two

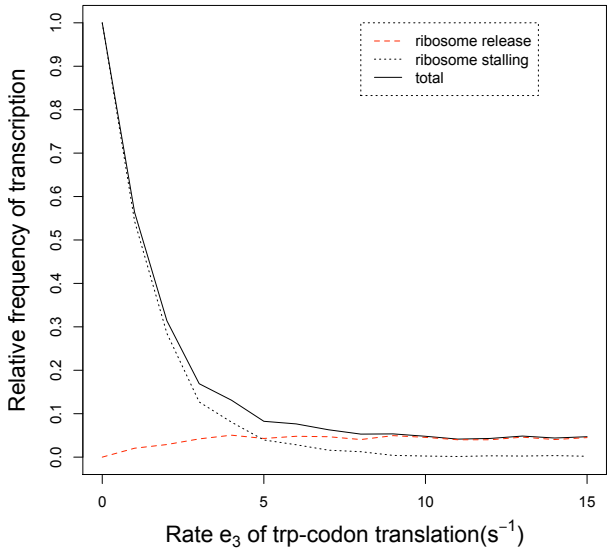

Fig. 7. Relative frequency of continued transcription as a function of the *trp*-codon translation rate. We distinguishing between anti-terminator formation during ribosome stalling, and after release of the ribosome from the stop codon.

different pathways lead to the anti-terminator, each of them corresponds to a distinct curve, and a third curve sums up.

The curve *ribosome stalling* corresponds to anti-terminator formation while the ribosome remains stalled on the control codons. This predominates when *trp*-codon translation is slow, becomes rarer as the translation efficiency increases, and drops below 1% when $e_3 \geq 9s^{-1}$.

In a second pathway, the anti-terminator hairpin forms after the ribosome has released from the stop codon 15. This represents the basal read-through level of the *trp* operon (see page 8). The corresponding curve *ribosome release* in our simulation plot starts from zero, steadily increases with the rate of *trp*-codon translation, reaches its maximal level around 4.5% at a rate of *trp*-codon translation of 4 s^{-1}, and remains stable henceforth.

Our results shown in Fig. 7 qualitatively confirm those of Elf and Ehrenberg [11]. However, even if the curves have the same shape, the asymptotic decrease of *ribosome stalling* toward 0 is less sharp in our case. While in their work the curves for ribosome release and ribosome stalling cross at a rate of $6s^{-1}$, in ours they already do so at $5s^{-1}$. Moreover our experiments predict a rate of basal transcription of slightly under 5% when *trp*-codon translation is efficient, where Elf and Ehrenberg predict 8%.

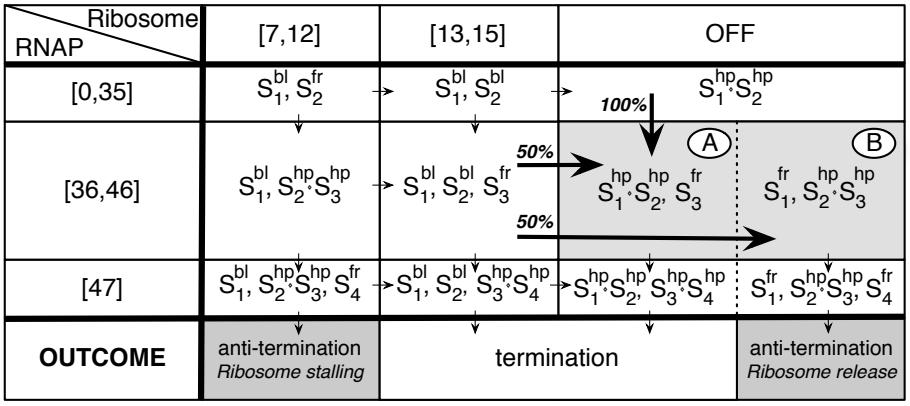

Table 7. Transitions between configurations of the *trp* leader that our model yields, depending on the relative positions of the ribosome versus RNAP. We list the first segment in its blocked state for the sake of presentation, when the solution indeed enumerates its sequence.

5.3 Discussion

We believe that the differences between our quantitative results and those of [11] are due to the greater level of detail rendered by our model. Elf and Ehrenberg reduce attenuation to a race between the ribosome and the polymerase, but their model does not make hairpins explicit. Instead, they infer transcription probabilities from the *relative positions* of both the RNAP and the ribosome. As opposed to this, our model explicitly renders hairpins.

Table 7 summarizes how the configuration of the system evolves, assuming simulations start from our initial solution (equation 0), and the ribosome has disrupted the pause hairpin. For each relative position of the ribosome on mRNA (columns) and the RNAP on DNA (rows), the table states which segments have been transcribed so far, and which hairpins have formed. Arrows between the table's cells corresponds to possible transitions of our solution. We distinguish the following positions of interest for the RNAP on DNA:

- **between DNA nucleotide 0 and 35:** only segments S_1 and S_2 are contained in the initial solution,
- **between 36 and 46:** RNAP has completed segment S_3,
- **on position 47:** RNAP has injected S_4 to the solution.

The *ribosome*'s positions of interest on the mRNA leader are:

- **between codon 7 and 12:** the ribosome's footprint has not yet reached S_2;
- **between codon 13 and 15:** the ribosome's footprint masks S_2 until dissociation from the mRNA.
- **off:** the ribosome has dissociated from the transcript, making segment S_2 newly available for hairpin formation.

Elf and Ehrenberg distinguish two cases of anti-termination (reproduced in our simulations in Fig. 7): anti-termination during *ribosome stalling* and anti-termination after *ribosome release*.

Stochastic trajectories leading to *ribosome stalling* descend our table's column $[7, 12]$. The ribosome then remains between the codons 7 and 12, most likely stalling on the control (*trp*) codons within the first segment. After the polymerase has transcribed segment S_3, the anti-terminator hairpin $S_2 \cdot S_3$ forms. This henceforth excludes the terminator hairpin, and transcription continues into the structural genes. We believe that our model and that of Elf and Ehrenberg show the same behavior for this first pathway.

We identified the following key difference between the models regarding the second pathway to anti-termination. Elf and Ehrenberg assume that the probability of anti-termination after *ribosome release* is half the probability of reaching the configuration (RNAP $\in [36, 46]$, ribosome OFF). The corresponding cell is highlighted in light gray in Table 7. It is important to note that we separate it into two sub-cells A and B. We refine Elf and Ehrenberg's assumption as follows: *stochastic trajectories leading to anti-termination after ribosome release pass through the sub-cell B, but* never *through the sub-cell A*.

Careful consideration of the table explains our refinement. The highlighted cell can be reached from either its top or left neighbor. Coming from the left neighbor (RNAP $\in [36, 46]$, ribosome $\in [13, 15]$), it is *equally likely* to reach sub-cell A that entails termination, as to reach B that entails anti-termination. However, coming from the top neighbor (RNAP $\in [0, 35]$, ribosome OFF) makes a difference. Because the ribosome has unbound the mRNA before segment S_3 was completed, the pause hairpin $S_1^{hp} \cdot S_2^{hp}$ is already part of the solution. Hence the anti-terminator does not appear once S_3 is completed. Thus, descending the column *ribosome off*, the system *always* reaches configuration A, and transcription always terminates.

Based on our detailed model, and contradicting Elf and Ehrenberg, we claim that the state (RNAP $\in [36, 46]$, ribosome off) does *not* lead with equal probabilities to anti-termination by ribosome release versus termination. The downward transitions into sub-cell A, that contains the pause hairpin $S_1^{hp} \cdot S_2^{hp}$ and hence excludes the anti-terminator, increase the termination probability[6]. This line of reasoning agrees with experimental knowledge on the impact of early ribosome release [29], and may explain why our experiments predict anti-termination less often than Elf and Ehrenberg's.

6 Conclusion

We have shown that rule-based modeling provides concise and elegant models for the fine-grained mechanism of transcriptional attenuation, a problem left open by previous work on discrete event modeling of the tryptophan operon [32]. The

[6] Another effect is due to the anti-terminator hairpin melting as the ribosome moves on to $m\text{RNA}_{13}$ that is included in our model, which lowers the *ribosome stalling* probability.

core ingredients for our model are rule schemas and n-ary chemical reactions. The importance of n-ary reactions renders representations of this case of genetic regulation in object-centered languages such as the stochastic pi-calculus [22,25,28] inappropriate, in practice. We used the Kappa factory for stochastic simulation [8], which provided us convenient analysis tools.

In our model we identified positions of individual nucleotides and codons within DNA and RNA sequences by numbers, abstracted over by variables, and addressed successors by simple arithmetic. This technique has its limitations when polymers become more complex than simple lists. Alternatively, we could assign names to molecular domains, and memorise those names in attribute values of adjacent molecules, similarly to Kappa. As shown in Sect. 3.4, the cost for such an alternative is that meaningless instances of rule schemas may be generated.

In future work, we plan to compute the exact probability that the ribosome dissociates before the segment S_3 is formed. This corresponds to the additional pathway that is not rendered by the model proposed by Elf and Ehrenberg (2005) [11]. This would formally prove what we conjectured to be the source of the quantitative difference between the two models. We can indeed compute such a probability because there are finitely many pathways and all pathways are terminating (i.e. the system always reaches a configuration in which no rule is applicable). In other words, one can exhaustively unfold the underlying CTMC and thus compute the probability associated to each pathway (i.e. each branch of the CTMC).

Acknowledgements. The Master's project with Valerio Passini at the Microsoft Research - University of Trento Center for Computational and Systems Biology sharpened our view of the system from a biological perspective and lead us to a rule-based approach. The previous Master's project of Gil Payet (co-supervised with Denys Duchier) had confronted us with the limitations of object-based approaches to the representation of the complex dependencies of transcriptional attenuation. We thank Maude Pupin, who was the first to point us at the intricate regulatory mechanisms at *E.coli*'s tryptophan operon, and Joachim Niehren for his valuable coaching. Plectix BioSystems kindly made the Kappa Factory available to us, and gave us useful support while we carried out the experiments reported in this paper. Finally, we thank the CNRS for a sabbatical to Cédric Lhoussaine, and the Agence Nationale de Recherche for funding this work through a *Jeunes Chercheurs* grant (ANR BioSpace, 2009-2011).

References

1. Adam Arkin, John Ross, and Harley H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells. *Genetics*, 149:1633–1648, 1998.
2. Franz Baader and Tobias Nipkow. *Term rewriting and all that.* Cambridge University Press, New York, NY, USA, 1998.

3. Matjaz Barboric and B. Matija Peterlin. A new paradigm in eukaryotic biology: HIV Tat and the control of transcriptional elongation. *PLoS Biology*, 3(2):0200–02003, 2005.
4. Chase L. Beisel and Christina D. Smolke. Design principles for riboswitch function. *PLoS Computational Biology*, 5(4):e1000363, 04 2009.
5. Luca Cardelli and Gianluigi Zavattaro. On the computational power of biochemistry. In *Proceedings of the 3rd international conference on Algebraic Biology*, pages 65–80, Berlin, Heidelberg, 2008. Springer-Verlag.
6. Nathalie Chabrier-Rivier, Francois Fages, and Sylvain Soliman. The biochemical abstract machine BioCham. In *Proceedings of CMSB 2004*, volume 3082 of *Lecture Notes in Bioinformatics*, pages 172–191, 2005.
7. Federica Ciocchetta and Jane Hillston. Bio-PEPA: a framework for modelling and analysis of biological systems. *Theoretical Computer Science*, 2008. To appear.
8. Vincent Danos, Jerome Feret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In *5th Asian Symposium on Programming Languages and Systems*, volume 4807 of *Lecture Notes in Computer Science*, pages 139–157, 2007.
9. Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Rule-based modelling of cellular signalling. In *18th International Conference on Concurrency Theory*, volume 4703 of *Lecture Notes in Computer Science*, pages 17–41, 2007.
10. L. Dematté, C. Priami, and A. Romanel. The beta workbench: A tool to study the dynamics of biological systems. *Briefings in Bioinformatics*, 9(5):437–449, 2008.
11. Johan Elf and Mans Ehrenberg. What makes ribosome-mediated trascriptional attenuation sensitive to amino acid limitation? *PLoS Computational Biology*, 1(1):14–23, 2005.
12. Daniel T. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22:403–434, 1976.
13. Paul Gollnick. Trp operon and attenuation. In William J. Lennarz and M. Daniel Lane, editors, *Encyclopedia of Biological Chemistry*, pages 267 – 271. Elsevier, New York, 2004.
14. Paul Gollnick, Paul Babitzke, Alfred Antson, and Charles Yanofsky. Complexity in regulation of tryptophan biosynthesis in Bacillus subtilis. *Annual Review of Genetics*, 39(1):47–68, 2005.
15. A Gutierrez-Preciado, R.A. Jensen, C. Yanofsky, and E. Merino. New insights into regulation of the tryptophan biosynthetic operon in Gram-positive bacteria. *Trends in Genetics*, 21(8):432–436, 2005.
16. Michael L. Blinov James R. Faeder and William S. Hlavacek. *Systems Biology*, volume 500 of *Methods in Molecular Biology*, chapter Rule-Based Modeling of Biochemical Systems with BioNetGen, pages 1–55. Humana Press, 2009.
17. Mathias John, Cédric Lhoussaine, Joachim Niehren, and Adelinde M. Uhrmacher. The attributed pi-calculus with priorities. *Transactions on Computational Systems Biology*, 2009. To appear.
18. Y Nakamura JR Roesser and C. Yanofsky. Regulation of basal level expression of the tryptophan operon of Escherichia coli. *J Biol Chem*, 264(21):12284–8, 1989.
19. T Kasai. Regulation of the expression of the histidine operon in Salmonella typhimurium. *Nature*, 249:523–527, 1974.
20. Kouacou Vincent Konan and Charles Yanofsky. Role of ribosome release in regulation of tna operon expression in Escherichia coli. *J. Bacteriol.*, 181:1530–1536, 1999.

21. Jean Krivine, Robin Milner, and Angelo Troina. Stochastic bigraphs. In *24th Conference on the Mathematical Foundations of Programming Semantics*, volume 218 of *Electronical notes in theoretical computer science*, pages 73–96. Elsevier, 2008.
22. Céline Kuttler, Cédric Lhoussaine, and Joachim Niehren. A stochastic pi calculus for concurrent objects. In *Second International Conference on Algebraic Biology*, volume 4545 of *Lecture Notes in Computer Science*, pages 232–246. Springer Verlag, July 2007.
23. John W. Lloyd. *Foundations of Logic Programming, 2nd Edition*. Springer Verlag, 1987.
24. Michael Pedersen and Gordon Plotkin. A language for biochemical systems. *Transactions on Computational Systems Biology*, 2009. To appear.
25. Andrew Phillips and Luca Cardelli. Efficient, correct simulation of biological processes in the stochastic pi-calculus. In *Computational Methods in Systems Biology, International Conference*, volume 4695 of *Lecture Notes in Computer Science*, pages 184–199. Springer Verlag, 2007.
26. S Pradalier, A Credi, M Garavelli, C Laneve, and G Zavattaro. Modelization and simulation of nano devices in the nano-kappa calculus. In *Computational Methods in Systems Biology, International Conference CMSB 2007*, 2007.
27. Stephen Ramsey, David Orrell, and Hamid Bolouri. Dizzy: stochastic simulation of large-scale genetic regulatory networks. *Journal of Bioinformatics and Computational Biology*, 3(2):415–436, 2005.
28. Aviv Regev. *Computational Systems Biology: A Calculus for Biomolecular Knowledge*. Tel Aviv University, 2002. PhD thesis.
29. James R. Roesser and Charles Yanofsky. Ribosome release modulates basal level expression of the trp operon of Escherichia coli. *Journal of Biological Chemistry*, 263(28):14251–14255, 1988.
30. Moises Santillan and Eduardo S. Zeron. Dynamic influence of feedback enzyme inhibition and transcription attenuation on the tryptophan operon response to nutritional shifts. *Journal of Theoretical Biology*, 231(2):287–298, 2004.
31. Stuart M. Shieber. *An Introduction to Unification-Based Approaches to Grammar*, volume 4. CLSI Publications, 1986.
32. E. Simão, Elisabeth Remy, Denis Thieffry, and Claudine Chaouiya. Qualitative modelling of regulated metabolic pathways: application to the tryptophan biosynthesis in E .coli. In *ECCB/JBI*, pages 190–196, 2005.
33. Nancy Trun and Janine Trempy. *Fundamental bacterial genetics*, chapter Gene expression and regulation, pages 191–212. Blackwell, 2003.
34. G von Heijne, L Nilsson, and C Blomberg. Translation and messenger RNA secondary structure. *Journal of Theoretical Biology*, 68:321–329, 1977.
35. Jin Yang, Michael I. Monine, James R. Faeder, and William S. Hlavacek. Kinetic monte carlo method for rule-based modeling of biochemical networks. *Physical Review E*, 78(3):7, 2008.
36. Charles Yanofsky. Attenuation in the control of expression of bacterial operons. *Nature*, 289:751–758, 1981.
37. Charles Yanofsky. Transcription attenuation: once viewed as a novel regulatory strategy. *J Bacteriology*, 182(1):1–8, 2000.
38. Charles Yanofsky. RNA-based regulation of genes of tryptophan synthesis and degradation, in bacteria. *RNA - A publication of the RNA Society*, 13(8):1141–1154, 2007.

A Kappa Rules for Section 5

In the following we list the Kappa code of our detailed attenuation model in Section 5, as we implemented it in Kappa Factory version 12.2.0. in order to run stochastic simulations. This encoding remains as close as possible to our rule-based model, notably it does not represent DNA or mRNA as interconnected chains, hence does not use *name sharing*.

Some comments on the syntax of Kappa seem appropriate. For instance consider the following reaction produced by rule schema 1:

```
'LoopS2S3 (anti-terminator)' S2(s~fr),S3(s~fr)
  -> S2(s~hp!1),S3(s~hp!1) @ $INF
```

This reaction is given the name `'LoopS2S3(anti-terminator)'` and an infinite rate `@ $INF`. The reactants `S2(s~fr)` and `S3(s~fr)` have an attribute `s` both with values `fr`. The reaction produces the molecules `S2(s~hp!1)` and `S3(s~hp!1)`, where the attribute `s` has the value `hp`. The modifier `!1` indicates that the molecules form a complex, which is linked by the edge `1`.

```
# rule schema 1
'LoopS2S3 (anti-terminator)' S2(s~fr),S3(s~fr)
  -> S2(s~hp!1),S3(s~hp!1) @ $INF
'LoopS3S4 (terminator)' S4(s~fr),S3(s~fr)
  -> S4(s~hp!1),S3(s~hp!1) @ $INF
'LoopS1S2 (pause)' S1(s~fr),S2(s~fr) -> S1(s~hp!1),S2(s~hp!1) @ $INF
# rule schema 2
'RiboTo8' Ribo(m!1),mRNA7(t!1),mRNA8(t)
  -> Ribo(m!2),mRNA7(t),mRNA8(t!2) @ 15.0
...
'RiboTo15' Ribo(m!1),mRNA14(t!1),mRNA15(t)
   -> Ribo(m!2),mRNA14(t),mRNA15(t!2) @ 15.0
# rule sche0ma 3
'RiboTo11_Trp' Ribo(m!1),mRNA10(t!1),mRNA11(t)
       -> Ribo(m!2),mRNA10(t),mRNA11(t!2) @ 1.0
'RiboTo12_Trp' Ribo(m!1),mRNA11(t!1),mRNA12(t)
       -> Ribo(m!2),mRNA11(t),mRNA12(t!2) @ 1.0
# rule schema 4
'RiboTo7_MeltS1S2' Ribo(m!1),mRNA6(t!1),mRNA7(t),
   S1(s~hp!2),S2(s~hp!2)
   -> Ribo(m!2),mRNA6(t),mRNA7(t!2),mRNA10(t),mRNA15(t),
   mRNA11(t), S2(s~fr),mRNA14(t),mRNA13(t),mRNA12(t) @ 15.0
# rule schema 5
'blockS2_ribo@13' Ribo(m!1),mRNA13(t!1),S2(s~fr)
  -> Ribo(m!1),mRNA13(t!1),S2(s~bl) @ $INF
# rule schema 6
'meltS2S3_ribo@13' Ribo(m!1),mRNA13(t!1),S2(s~hp!1),S3(s~hp!1)
  -> Ribo(m!1),mRNA13(t!1),S2(s~bl), S3(s~fr) @ $INF
```

```
# rule schema 7
'RiboRelease@15_joinS1' mRNA15(t!1),Ribo(m!1),mRNA10(t),
mRNA11(t),mRNA12(t),mRNA13(t),mRNA14(t),S2(s~bl)
-> Ribo(m),S1(s~fr),S2(s~fr) @ 1.0
# rule schema 8
'RNAPto2' RNAP(d!1),DNA1(t!1),DNA2(t)
  -> RNAP(d!2),DNA1(t),DNA2(t!2) @ 50.0
...
'RNAPto35' RNAP(d!1),DNA34(t!1),DNA35(t)
   -> RNAP(d!2),DNA34(t),DNA35(t!2) @ 50.0
'RNAPto37' RNAP(d!1),DNA36(t!1),DNA37(t)
   -> RNAP(d!2),DNA36(t),DNA37(t!2) @ 50.0
...
'RNAPto46' RNAP(d!1),DNA45(t!1),DNA46(t)
   -> RNAP(d!2),DNA45(t),DNA46(t!2) @ 50.0
'RNAPto49' RNAP(d!1),DNA48(t!1),DNA49(t)
   -> RNAP(d!2),DNA48(t),DNA49(t!2) @ 50.0
'RNAPto50' RNAP(d!1),DNA49(t!1),DNA50(t)
   -> RNAP(d!2),DNA49(t),DNA50(t!2) @ 50.0
# rule schema 9
'RNAPresumes_S1S2broken_a' RNAP(d!1),DNA0(t!1),DNA1(t),S2(s~bl) ->
 RNAP(d!2),DNA0(t),DNA1(t!2),S2(s~bl) @ 50.0
'RNAPresumes_S1S2broken_b' RNAP(d!1),DNA0(t!1),DNA1(t),S2(s~fr) ->
 RNAP(d!2),DNA0(t),DNA1(t!2),S2(s~fr) @ 50.0
# rule schema 10
'RNAPto36_spawnS3' RNAP(d!1),DNA35(t!1),DNA36(t)
   -> RNAP(d!2),DNA35(t),DNA36(t!2),S3(s~fr) @ 50.0
# rule schema 11
'RNAPto47_spawnS4' RNAP(d!1),DNA46(t!1),DNA47(t)
   -> RNAP(d!2),DNA46(t),DNA47(t!2),S4(s~fr) @ 50.0
# rule schema 12
'RNAP_dissociate' RNAP(d!1),DNA47(t!1),S3(s~hp!2),S4(s~hp,2!)
      -> RNAP(d),DNA47(t),DNA48(t),S3(s~hp,!2),S4(s~hp,!2) @ 50.0
# rule schema 13
'RNAP_antiTerm' RNAP(d!1),DNA47(t!1),DNA48(t),
   S2(s~hp,!2),S3(s~hp,!2) -> DNA47(t),RNAP(d!3),
   DNA48(t!3),S2(s~hp,!2),S3(s~hp,!2)  @ 50.0
```

All quantitative information reported in this article is indeed obtained from Kappa *stories*, which required additional control molecules that do not represent any actual biological actor. We refrain from showing this version of the code here, but it is available from the authors upon request.