Quick Direct-method Controlled (QDC): a simulator of metabolic experiments

Supplementary files

Davide Cangelosi, Salvatore Fabbiano, Claudio Felicioli, Luca Freschi and Roberto Marangoni

Department of Computer Science, University of Pisa, L.go B. Pontecorvo 3, 56127 Pisa, Italy.

Supplementary file 1: QDC described in the detail

QDC syntax and input file structure

QDC's core input is represented by an ASCII file, written in a home-defined format, where the syntax to express the reaction is very similar to that of biochemistry (see Fig. 1). The formal definition of the whole syntax is presented in the Supplementary file 2; here we give a brief summary. The input file is structured in subsequent blocks, separated by a blank line; each block contains a different category of information.

The first block contains the name declaration for each biochemical species that will take part in the system; each valid name is an alphanumeric string with possible underscore characters. To avoid confusion with stoichiometric coefficients, valid species names have to begin with a letter. One row can host several names separated by a comma. Some names are reserved for internal program usage or for SBML import/export; in particular: `NULL, null, parameter, event, reaction, addition, volume`. These terms cannot be used in the user-defined input code.

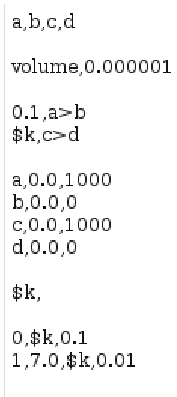
The second block declares the volume (measured in liters) taken by the system, as it is required for transforming reaction rates (deterministic) into stochastic propensities for chemical reactions whose order is greater than 1. The line statement is of the form: `volume,` $<$ *value* $>$.

The third block contains all the biochemical reactions hosted by the system. The general reaction syntax is:

$$r, \texttt{A + B} > \texttt{C + D + ...}$$

where `A, B, C, D` represent any allowed symbolic name of a chemical species.

The numerical value in input for the reaction rate is the deterministic one that QDC will transform into the stochastic propensity, accordingly with the reaction order. The usual arrow of the chemical equation is represented by the sign "$>$". QDC is able to manage and simulate reactions of different orders:



```
a,b,c,d

volume,0.000001

0.1,a>b
$k,c>d

a,0.0,1000
b,0.0,0
c,0.0,1000
d,0.0,0

$k,

0,$k,0.1
1,7.0,$k,0.01
```

Figure 1: An example of QDC input file: blocks of instructions are separated by blank lines.

- 0-order reactions. These reactions specify the “creation” of a given species: they are very useful for representing the effect of an uptake system that supply a certain species at a given rate. The syntax is of the type:

  $$r, \texttt{NULL} > \texttt{A} + \ldots$$

  As the term `NULL` (even lowercase `null`) is a reserved name, it must not be declared among the chemical species. For the same reason, it is not allowed to assign a number of molecules to it. The macroscopic rate of 0-order reactions has the dimensions of Mt^{-1}, but in our stochastic environment we are dealing with number of molecules, not with concentrations, then the reaction equation becomes: $N = N_0 - kt$, where the stochastic propensity k has the dimension of t^{-1} and therefore is independent of the system volume.

- 1st-order reactions. These reactions describe processes like isomerization, auto-splicing, decay, fission and similar, where the left term of the chemical equation involves a single molecule. The rate of 1st-order reactions depends only on the number of molecules present. The syntax of this kind of reaction is:

  $$r, \texttt{A} > \texttt{B + C + ...}$$

  A peculiar case of first-order reaction is when the right term of the chemical equation contains only the term “NULL”. This can be considered as an “excretion reaction”, where some chemical is released outside the chemical system. These reactions can be viewed as the opposite of 0-order reactions, at least as long as an uptaking systems can be thought as the opposite of an excreting system. In this case the syntax is:

$$r, A > \texttt{NULL} + \ldots$$

- 2nd-order reactions. These reactions feature two species on the left side of the chemical equation. Since this kind of reaction takes place when one molecule of the first species collides with one molecule of the second species, the rate of 2nd-order reactions depends on the number of molecules of the first species, the number of molecules of the second species, and on the volume of the system. 2nd-order reactions are the most common reactions in the cellular scenario, also because a higher-order reaction can be described as a combinatorial collection of several consecutive 2nd-order reactions.

- Immediate reactions. They take place immediately after the verification of the condition represented by the left side of the equation. In other words, the stoichiometry noted at the left side of the equation is a logical condition: once it is verified, the immediate reaction takes place and yields the products indicated at the right side. These reactions allow for complex stoichiometries, where both the reagents species number and the number of molecules per species can be greater than two. We want to remark that immediate reactions are not to be considered higher-order reactions since they do not have a kinetic law, and they happen immediately. The general syntax for an immediate reaction is:

$$-, \texttt{A + B +} \ldots > \texttt{C + D +} \ldots$$

The initial dash sign declares that this is an immediate reaction, with no associated reaction rate. Immediate reactions can also be used to simulate the simultaneous excretion of several chemical species, once they have reached a given threshold number of molecules: to do so, the right side of the equation must contain only the “NULL” term.

The fourth block contains the number of molecules assigned to all the declared chemical species that are supplied to the system, along with the time at which they are supplied. When such time is set to zero, the statement denotes the initial number of molecules for the declared species.

The last two blocks are optional and concern with the eventual presence of control variables, and in particular:

The fifth block contains all the control variables used during the experiment; their name begins with a dollar sign. As for the species names, multiple variables can be written on a line, separated by a comma; e.g.: $\$k$, $\$y$,...

The sixth block declares at what moment the control actions have to be executed and what is the final value of the controlled variable after the control action. To assign a value to a control variable at the beginning of the simulation, the statement is:

$$0,\ \$k,\ v_0$$

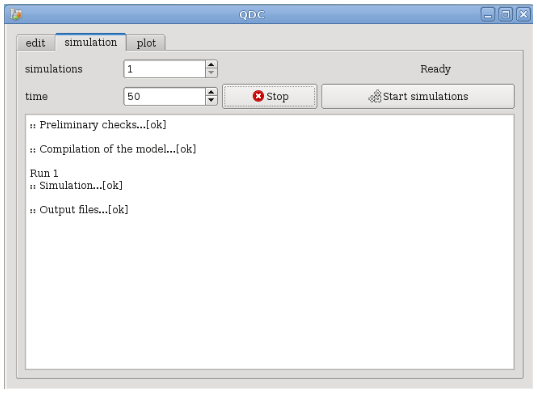
e.g., the statement: 0, $\$k$, 100 assigns the value 100 to the variable $\$k$ at the beginning of the simulation. If one wants to change such value during the simulation, the syntax is:

$$1,\ t,\ \$k,\ v_t$$

e.g.: 1, 10, $\$k$, 0.1 will indicates that at 10 sec of simulated time $\$k$ changes its value to 0.1.

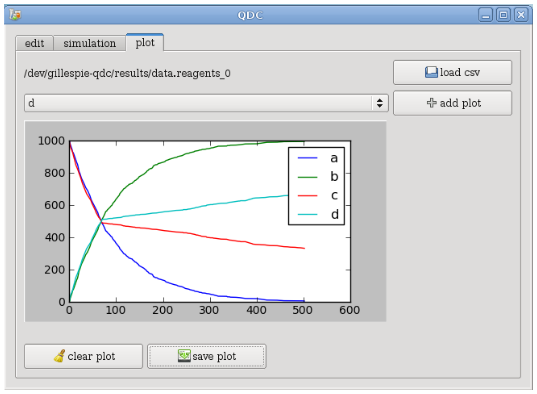
QDC’s GUI

QDC’s GUI has been developed to allow the user to have an immediate visualization of the simulated system behavior. The GUI is basic and easy to use: it has been developed in Python v.2.6 language and uses the PyQt libraries http://www.riverbankcomputing.co.uk/software/pyqt/ to manage interface’s elements. This choice confers QDC’s GUI to be of good portability, as it has been tested on different Linux distributions



```
simulations   1                                   Ready
time          50

:: Preliminary checks...[ok]

:: Compilation of the model...[ok]

Run 1
:: Simulation...[ok]

:: Output files...[ok]
```

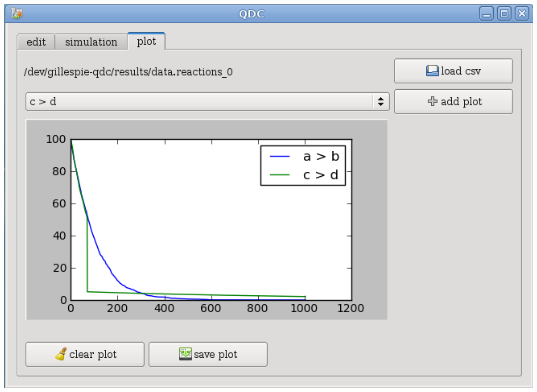
Figure 2: QDC GUI: monitoring the running status.



```
/dev/gillespie-qdc/results/data.reagents_0
d
```

Figure 3: QDC GUI: plotting the time-course of chemical species concentrations. In this figure, and in all the following ones, the abscissa reports the number of computed samples during the simulation. As the sampling frequency is set at 0.1 Hz, 10 samples take 1 s of simulation time. In this case, the change in the reaction rate after 7 s since the simulation start is clearly visible as the 70th sampling point.

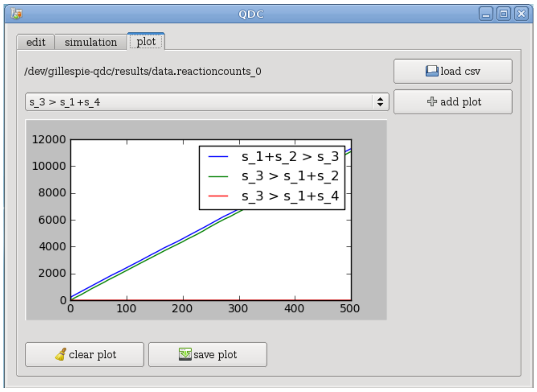
(Fedora, Ubuntu, etc.) and on Mac OS X. The Figure **??** shows the edit panel where one can directly write or paste the input file containing all the required blocks. Once completed the input, one can switch to the simulation tab where the simulation can be started by using the appropriate button; the next steps (compiling and running) are displayed once completed (see Fig. 2). Lastly, the plot tab provides a direct access to the plotting of the output files. These are released as .CSV files, and the user can choose which one to plot in the plot window, by using the “load CSV” button. Then, the user can choose a single species to plot, by selecting it out of the list and clicking on “add plot” (Fig. 3). The same goes for the reaction firing rates (Fig. 4) and the propensities (Fig. 5) files. That allows the user to see how the propensities change during the experiment.



/dev/gillespie-qdc/results/data.reactions_0

c > d

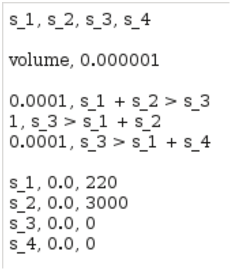
Figure 4: QDC GUI: plotting the time-course of reaction firing rates.



/dev/gillespie-qdc/results/data.reactioncounts_0

s_3 > s_1+s_4

Figure 5: QDC GUI: plotting the time-course of propensities.



```
s_1, s_2, s_3, s_4

volume, 0.000001

0.0001, s_1 + s_2 > s_3
1, s_3 > s_1 + s_2
0.0001, s_3 > s_1 + s_4

s_1, 0.0, 220
s_2, 0.0, 3000
s_3, 0.0, 0
s_4, 0.0, 0
```

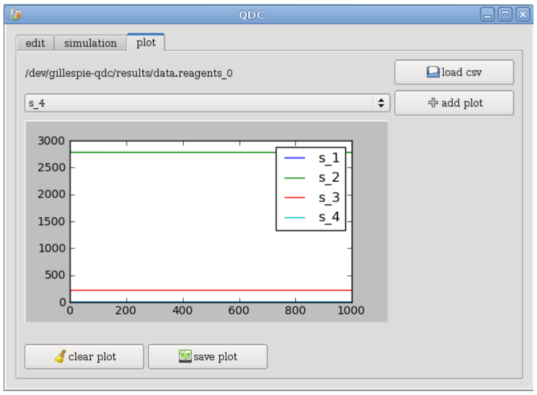
Figure 6: An example of a stiff system input file.

Using QDC: some examples

Figures 2-6 show an example of a QDC run of the biochemical system described in Fig. 1. It consists of four chemical species interacting in two first-order reactions. The rate of the second reaction is controlled by the variable $k, which is set to 0.1 at the beginning of the experiment and switches to 0.01 after 7 seconds, as indicated in the second line of the block that assigns numerical values to symbolic variables. As it is clearly shown in Fig. 2, the reaction rate change modifies abruptly both the curves of the reagent species ("c") and product species ("d") of the controlled reaction, while it does not affect the other two species involved in a fixed rate reaction. This is only one possible example out of many of a simple chemical system where a reaction is somehow controlled by an external factor.

A more sophisticated example, with no control actions, but with a very peculiar behavior by the system is shown in Figs. 6,7 and 8.

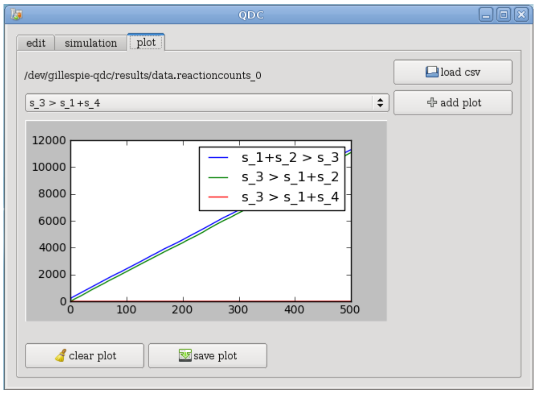
Here we used QDC to face a stiff system described by Cao *et al.* [2], when they considered the problem of a generic enzymatic reaction. It is well known, in fact, that in enzymatic reactions the enzyme-substrate complex is usually much more likely to decay into its original constituents than to generate a product molecule. This is because the decay reaction has a much higher rate than that of product formation reaction. In this simple model, an enzyme S1 binds a chemical species S2 and generates a enzyme-substrate complex S3. This can decay back into S1+S2 or continue with the reaction giving a product S4 and the enzyme free form S1. The decay reaction rate is three orders of magnitude higher than the two others. Figure 6 shows the complete QDC input file for this system.



/dev/gillespie-qdc/results/data.reagents_0

s_4

Figure 7: In the simulated stiff system, after a very short initial period (only one sampling period, 0.1 s), the species seem to have reached an equilibrium, as they possess a (quasi)-constant number of molecules. Two species are not visible here as their quantities exactly overlap those of the two displayed species.



/dev/gillespie-qdc/results/data.reactioncounts_0

s_3 > s_1+s_4

Figure 8: The effective reaction firing rate in the considered stiff system shows that there is one reaction (s3> s1 + s4) that has not yet taken place.

In a stiff system the slow part of the system (the system subset with very low rate reactions) is called after a relatively long time, if compared with the fast part of the system. If only the time course of the number of molecules of each species is checked (see Fig. 7), it could erroneously be concluded that the system has reached an equilibrium state, even if this is not true at all. To be aware of this it is also necessary to inspect the actual firing rate of each reaction (II output file of QDC): this makes evident that reaction 3 does not take place at all during all the experiment duration (see Fig. 8).

References

[1] Cao, Y., Gillespie, D.T., and Petzold, L.R.: ‘Accelerated stochastic simulation of the stiff enzyme-substrate reaction’, J. Chem. Phys., 2005, 123, p. 144917

QDC: Quick Direct-method Controlled

Supplementary file 2: QDC's syntax

1 QDC

Given a system with N molecular species $\{S_1, \cdots, S_N\}$ interacting through M different chemical reactions $\{R_1, \cdots, R_M\}$. Assumed that the system is well-stirred, that it is confined in a constant volume V and in thermal (but not chemical) equilibrium. The system state is represented by the multivariate variable $X(t) = \mathbf{x} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$. We write $\mathbf{x}_i(t)$, or more precisely $\mathbf{x}_i$, to indicate the number of molecules of species S_i in the system at time t. Given an initial value $X(t_0)$ for the state, the problem is to determine the value of $X(t)$ at any later time t. The evolution in time of a chemical system is consequence of chemical reactions. The reactions in the stochastic formulation of chemical kinetics approach can: unimolecular, written $S_1 \overset{c_j}{\to} Product(s)$, occurring as a result of processes internal to a single molecule with kinetic parameter c_j, and bimolecular, written $S_1 + S_2 \overset{c_j}{\to} Product(s)$ of two (distinct or not) molecular species occurring as a result of a collision. Any reaction R_j is characterized by two quantities. The first is the *state change vector* $\nu_j \equiv (\nu_{1j}, \cdots, \nu_{Nj})$, where ν_{ij} is the change in the population of S_i, caused by R_j. In other words, if the system is in the state $\mathbf{x}$ and one R_j reaction occurs, the system immediately jumps to state $\mathbf{x} + \nu_j$. The second quantity is the *propensity function* a_j, also written $a_j(\mathbf{x})$, which is defined so that $a_j(\mathbf{x})dt$ gives the probability that the reaction R_j will occur in the next infinitesimal time interval $[t, t + dt)$. Below we survey the propensity function formulas for each R_j.

If R_j is unimolecular, given $\mathbf{x}_1$ molecules of the species S_1 and the stochastic constant c_j, such that $c_j dt$ denotes the probability that any particular S_1 molecule will react in the next infinitesimal time dt, the propensity function is $a_j(\mathbf{x}) = c_j\mathbf{x}_1$. If R_j is bimolecular, given $\mathbf{x}_1$ molecules of the specie S_1, $\mathbf{x}_2$ of the specie S_2 and the constant c_j, such that $c_j dt$ gives the probability that a randomly chosen pair of S_1 and S_2 molecules will react according to R_j in the next infinitesimal time dt, the propensity function is $a_j(\mathbf{x}) = c_j\mathbf{x}_1\mathbf{x}_2$. Instead, if R_j is bimolecular but involving the same species S_1, the propensity function is $a_j(\mathbf{x}) = c_j\frac{1}{2}\mathbf{x}_1(\mathbf{x}_1 - 1)$. The stochastic reaction constant c_j is closely related to the more familiar reaction-rate constant k_j, which forms the basis for the deterministic approach to chemical kinetics. So, to make the conversion from k_j to c_j, QDC uses Table 1 for converting the reaction-rate constant k_j to the stochastic reaction constant c_j. In the next subsection, we will introduce our home-defined QDC language as a context free grammar and we will give the semantic of the language as a set of semantic rules.

R_j	Units of k_j	c_j
$NULL \xrightarrow{k} \dots$	$Msec^{-1}$	kN_AV
$S_1 \xrightarrow{k} \dots$	sec^{-1}	k
$S_1 + S_2 \xrightarrow{k} \dots$	$M^{-1}sec^{-1}$	$kN_A^{-1}V^{-1}$
$2S_1 \xrightarrow{k} \dots$	$M^{-1}sec^{-1}$	$2kN_A^{-1}V^{-1}$

Table 1: Conversion table between stochastic and deterministic kinetics constants. N_A is the Avogadro's constant.

1.1 QDC Language specification

A QDC program is made of mandatory and optional declarations. Mandatory are the declarations of the species, the volume, the reactions set, the initial amounts for the species. Whereas variables and events can be omitted. In this case the simulation have no perturbation set upped. Any QDC program is an ASCII file. The user can declare any *name* (sequence of Letters [a-zA-Z] and Digits [0-9]). A name is a generic identifier, here we will write speciesname or variablename in order to distinguish the names given to species by that given to the variables. We will write naturalnumber, realnumber, baserate, and time in order to distinguish the *numbers* given for the different cases in which a specific number type must be provided. However, because QDC uses the symbols $NULL, volume, >, \$,_{,} +, , , 0, 1$ as language statements, these symbols cannot be introduced in the declarations.

The syntax of the QDC language has the following syntax:

$$\begin{array}{lll}
QDCProgram & := & ProgramWithEvents \\
 & | & ProgramWithoutEvents \\
 & & \\
ProgramWithEvents & := & SpeciesList \\
 & & VolumeDec \\
 & & ReactionsList \\
 & & StateVector \\
 & & VariablesList \\
 & & EventList \\
 & & \\
ProgramWithoutEvents & := & SpeciesList \\
 & & VolumeDec \\
 & & ReactionsList
\end{array}$$

$$
\begin{array}{lcl}
 & & StateVector \\
 & & \\
SpeciesList & := & speciesname \\
 & | & speciesname, SpeciesList \\
 & & \\
VolumeDec & := & volume, realvalue \\
 & & \\
ReactionsList & := & Reaction \\
 & | & Reaction \\
 & & ReactionsList \\
 & & \\
Reaction & := & baserate, Reactants > Products \\
 & | & \$variablename, Reactants > Products \\
 & | & \_, Instantaneous > Instantaneous \\
 & & \\
Reactants & := & speciesname \\
 & | & speciesname + speciesname \\
 & | & 2speciesname \\
 & | & NULL \\
 & & \\
Products & := & speciesname \\
 & | & speciesname + speciesname \\
 & | & 2speciesname \\
 & | & NULL \\
 & & \\
Instantaneous & := & naturalnumberspeciesname \\
 & | & naturalnumberspeciesname + Instantaneous \\
 & & \\
StateVector & := & SpeciesAmount \\
 & | & SpeciesAmount \\
 & & StateVector \\
 & & \\
SpeciesAmount & := & speciesname, time, naturalnumber \\
 & & \\
VariablesList & := & Variable \\
 & | & Variable, VariablesList \\
 & & \\
Variable & := & \$variablename \\
 & & \\
EventsList & := & Event \\
 & | & Event \\
 & & EventsList \\
 & & \\
Event & := & 0, \$variablename, realnumber \\
 & | & 1, \$variablename, realnumber
\end{array}
$$

The syntax expressed before define in a very simple way any correct QDC program. Below we provide also how our Tool maps any syntactically well formed QDC file into the input parameters of the Gillespie's Direct Method. To do that we define the *state of a SSA simulation* as the tuple $Z = (\mathbf{X}(t), E, S, R, C, A, t)$, where $X(t) = \mathbf{x} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ is the state of the system at a given time t, $E = \{e_1, \cdots, e_h\}$ is the set of events, $S = \{S_1, \cdots, S_N\}$ is the set of all the species name, $R = \{R_1, \cdots, R_M\}$ is the set of all the system reactions, $C = \{c_1, \cdots, c_M\}$ is the set of all the stochastic coefficients, $A = \{a_1(\mathbf{x}), \cdots, a_M(\mathbf{x})\}$ is the set of the propensity values of the reactions and t is the simulation time.

Given $Z = (\mathbf{X}(t), E, S, R, C, A, t)$, an instance $S_1, S_2, \cdots, S_N$ obtained by the *SpeciesList* production is mapped into the set of species names $S = \{S_1, \cdots, S_N\}$. The volume declaration in the *VolumeDec* production becomes the volume of the compartment expressed in Liter. For example,
$volume, 0.0000000000015$ is interpreted as the declaration of the volume identifier $V = 0.0000000000015$. A sequence of $M = 8$ reaction declarations, as follows:

- $r1, S_1 > S_2$
- $r2, S_1 + S_2 > S_3$
- $r3, 2S_2 > S_4 + S_5$
- $r4, NULL > S_2$
- $\$var1, S_1 > S_2$
- $\$var2, S_1 + S_2 > S_3$
- $\$var3, 2S_2 > S_4 + S_5$
- $\$var4, NULL > S_2$

is interpreted as the reaction set:

$$R = \{S_1 \overset{r_1}{\to} S_2, S_1{+}S_2 \overset{r_2}{\to} S_3, 2S_2 \overset{r_3}{\to} S_3, NULL \overset{r4}{\to} S_2, S_1 \overset{\$var1}{\to}, S_2, S_1{+}S_2 \overset{\$var2}{\to} S_6, 2S_2 \overset{\$var3}{\to} S_3, NULL \overset{\$var4}{\to} S_2\}$$

The initial value of the variables $\$var1, \$var2, \$var3, \$var4$ must be assigned using the statement $0, \$s, r$ in the *EventList* production. Anyway, our tool interprets the values $r_1, r_2, r_3, r_4, \$var1, \$var2, \$var3, \$var4$ as deterministic reaction rates $k_1, \cdots, k_8$, then it converts them according by the conversion Table 1 and generates the stochastic coefficients set $C = \{c_1, \cdots, c_8\}$. Once the conversion is completed, our tool computes the propensity function set $A = \{a_1(\mathbf{x}), \cdots, a_8(\mathbf{x})\}$ using the classical propensity functions seen at the beginning of this Appendix.

Events declaration requires a different treatement. In fact, events are external statements to the SSA formulation. They reprensent perturbation actions with effects on the system evolution. The action fired by the occurrence of an event is subject to the verification of a trigger condition. The trigger condition can dependent both on the simualtion time and on stoichiometric conditions. In any case, the list of events declared using the *EventsList* production form the set E of the events and in the following we will give how our tool executes the events and what effects on the state of the simulation they induce. To do this we first define the *State Transition Function* $\mapsto$. *State Transition Function* $\mapsto$ is a function that maps a given tupla into another that is the result of a possible evolution of the first one. In the Gillespie's Stochastic Simulation Algorihtm we can see $\mapsto$ as the function that starting from a state of the simulation $Z = (\mathbf{X}(t), E, S, R, C, A, t)$ and given the pair (τ, j) passes to a new state $Z'' = (\mathbf{X}(t+\tau), E, S, R, C, A', t)$ where $\mathbf{X}(t+\tau) = \mathbf{X}(t) + \nu_j$ and A' is the new set of propensity function that changed their value for effect of the changing of the molecular amount. More formally

$$\frac{(\tau, j) \leftarrow GillespieTauandJselectionStep}{< Z > \mapsto < (\mathbf{X}[(\mathbf{X}(t) + \nu_j)/\mathbf{X}(t)], E, S, R, C, A', t) >} \tag{1}$$

We write $A[r/\$s]$ to specify that in a given set A we substitute the value of the variable $\$s$ with a new value r. We will also write $A/\$s$ to indicate the removal of the element with identifier $\$s$ from the set A. To complete the definition of the *State Transition Function* $\mapsto$ we give a set of Rules to define how our tool executes Events.

Let's suppose now to have an *instantaneous reaction* that we call *inst* for short, defined in the following way, $\_, l_1 s1 + l_2 s2 > l_3 s3$ where $l_1, l_2, l_3 > 0$ are stoichiometric values. When the system can satisfy the stoichiometry of the left side of the equation, QDC immediately puts the simulation in pause and it remains in pause since the *inst* trigger condition holds. To represent the pause of the simulation we write $(\mathbf{X}(t), \mathtt{E}, \mathtt{S}, \mathtt{R}, \mathtt{C}, \mathtt{A}, \mathtt{t})$ calling this new state Z'. Since the simulation is in pause all elements in Z' cannot be modified except $X(t)$. To deal with *inst* we identified four distinct cases. The first one happens when the simulation run and at a certain point the system has enough molecules to trigger this event. Formally, the Rule 2 expresses this case.

$$\frac{\mathbf{x}_{s1} \geq l_1 \; \mathbf{x}_{s2} \geq l_2 \; inst \in E}{< Z > \mapsto < Z'[(\mathbf{X}(t) + \nu_{inst})/\mathbf{X}(t)] >} \tag{2}$$

The second case Rule 3 takes place when the simulation has been already paused and the system has enough molecules to trigger again *inst*. Note that *inst* is not removed by any elements of Z'.

$$\frac{\mathbf{x}_{s1} \geq l_1 \; \mathbf{x}_{s2} \geq l_2 \; inst \in E}{< Z' > \mapsto < Z'[(\mathbf{X}(t) + \nu_{inst})/\mathbf{X}(t)] >} \tag{3}$$

The Rule 4 considers the third case, when *inst* cannot occur anymore. In this case, the state of the simulation is restored to a value in which all elements of Z' are no longer in pause and the simulation can continue normally.

$$\frac{\mathbf{x}_{s1} < l_1 \text{ or } \mathbf{x}_{s2} < l_2 \; inst \in E}{< Z' > \mapsto < Z >} \tag{4}$$

Last, the Rule 5 considers the fourth case in which the event cannot be triggered because there are not enough molecules and the simulation continues normally.

$$\frac{\mathbf{x}_{s1} < l_1 \text{ or } \mathbf{x}_{s2} < l_2 \; inst \in E}{< Z > \mapsto < Z >} \tag{5}$$

Supposing to have the statement s, t', n, called *aug* for short, the Rule 6 expresses the following behavior: if $t' \geq t$, only two elements of Z change due to *aug*: the state $\mathbf{x}_s$ of the species with name s, augments its current value of n molecules and the event set E, deleting *aug*, decreases its size.

$$\frac{t' \geq t \; aug \in E}{< Z > \mapsto < (\mathbf{X}[(\mathbf{x}_s + n)/\mathbf{x}_s], E/aug, S, R, C, A, t) >} \tag{6}$$

Supposing now to have the statement $1, t', \$s, r$, called *chan* for short, the Rule 7 expresses the following behavior: if $t' \geq t$, due to *chan*, Z the current value of the stochastic coefficient with name $\$s$ in C is substituted with the new value $f(r)$ and the event set E, deleting *chan*, decreases its size. Here, $f(r)$ represents the deterministic reaction rate r opportunely converted using Table 1. Consequently the reaction in which $\$s$ appears as kinetic parameter must be recomputed.

$$\frac{t' \geq t \; chan \in E}{< Z > \mapsto < (\mathbf{X}(t), E/chan, S, R, C[r/\$s], A[a_j[r/\$d](\mathbf{x})/a_j(\mathbf{x})], t) >} \tag{7}$$

QDC: Quick Direct-method Controlled
Supplementary file 3/A: dynamic change of a propensity

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version4" level="2" version="4">
  <model id="my_model">
    <listOfUnitDefinitions>
      <unitDefinition id="per_second">
        <listOfUnits>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="litre_per_mole_per_second">
        <listOfUnits>
          <unit kind="mole" exponent="-1"/>
          <unit kind="litre"/>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="mole_per_litre_per_second">
        <listOfUnits>
          <unit kind="mole"/>
          <unit kind="litre" exponent="-1"/>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
    </listOfUnitDefinitions>
    <listOfCompartments>
      <compartment id="comp" size="1"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id="a" name="a" compartment="comp" 
initialConcentration="1.66053878316273e-20"/>
      <species id="c" name="c" compartment="comp" initialConcentration="0"/>
      <species id="d" name="d" compartment="comp" initialConcentration="0"/>
    </listOfSpecies>
    <listOfParameters>
      <parameter id="parameter1" value="0.5" units="mole_per_litre_per_second"/>
      <parameter id="q" value="0.1" units="per_second" constant="false"/>
      <parameter id="parameter2" value="0.002" units="per_second"/>
    </listOfParameters>
    <listOfReactions>
      <reaction id="reaction1" reversible="false">
        <listOfProducts>
          <speciesReference species="a"/>
        </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <times/>
              <ci> parameter1 </ci>
              <ci> comp </ci>
            </apply>
          </math>
        </kineticLaw>
      </reaction>
      <reaction id="reaction2" reversible="false">
        <listOfReactants>
          <speciesReference species="a"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species="c"/>
          <speciesReference species="d"/>
        </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <times/>
              <ci> q </ci>
              <ci> comp </ci>
              <ci> a </ci>
            </apply>
          </math>
        </kineticLaw>
      </reaction>
    </listOfReactions>
    <listOfEvents>
      <event id="event1">
        <trigger>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <geq/>
              <csymbol encoding="text"
definitionURL="http://www.sbml.org/sbml/symbols/time"> t </csymbol>
              <cn> 10 </cn>
            </apply>
          </math>
        </trigger>
        <listOfEventAssignments>
          <eventAssignment variable="q">
            <math xmlns="http://www.w3.org/1998/Math/MathML">
              <ci> parameter2 </ci>
            </math>
          </eventAssignment>
        </listOfEventAssignments>
      </event>
    </listOfEvents>
  </model>
</sbml>
```

QDC: Quick Direct-method Controlled
Supplementary file 3/B: an *immediate* reaction

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version4" level="2" version="4">
  <model id="my_model">
    <listOfUnitDefinitions>
      <unitDefinition id="per_second">
        <listOfUnits>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="litre_per_mole_per_second">
        <listOfUnits>
          <unit kind="mole" exponent="-1"/>
          <unit kind="litre"/>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="mole_per_litre_per_second">
        <listOfUnits>
          <unit kind="mole"/>
          <unit kind="litre" exponent="-1"/>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
    </listOfUnitDefinitions>
    <listOfCompartments>
      <compartment id="comp" size="1"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id="a" name="a" compartment="comp"
initialConcentration="1.66053878316273e-24"/>
      <species id="b" name="b" compartment="comp" initialConcentration="0"/>
      <species id="c" name="c" compartment="comp" initialConcentration="0"/>
    </listOfSpecies>
    <listOfParameters>
      <parameter id="parameter1" value="0.9" units="mole_per_litre_per_second"/>
    </listOfParameters>
    <listOfReactions>
      <reaction id="reaction1" reversible="false">
        <listOfProducts>
          <speciesReference species="a"/>
        </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <times/>
              <ci> parameter1 </ci>
              <ci> comp </ci>
            </apply>
          </math>
        </kineticLaw>
      </reaction>
      <reaction id="reaction2" reversible="false" fast="true">
        <listOfReactants>
          <speciesReference species="a" stoichiometry="30"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species="c"/>
        </listOfProducts>
      </reaction>
    </listOfReactions>
  </model>
</sbml>
```

QDC: Quick Direct-method Controlled
Supplementary file 3/C: molecules addition at a given moment

```
<?xml version="1.0" encoding="UTF-8"?>
<sbml xmlns="http://www.sbml.org/sbml/level2/version4" level="2" version="4">
  <model id="my_model">
    <listOfUnitDefinitions>
      <unitDefinition id="per_second">
        <listOfUnits>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="litre_per_mole_per_second">
        <listOfUnits>
          <unit kind="mole" exponent="-1"/>
          <unit kind="litre"/>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
      <unitDefinition id="mole_per_litre_per_second">
        <listOfUnits>
          <unit kind="mole"/>
          <unit kind="litre" exponent="-1"/>
          <unit kind="second" exponent="-1"/>
        </listOfUnits>
      </unitDefinition>
    </listOfUnitDefinitions>
    <listOfCompartments>
      <compartment id="comp" size="1"/>
    </listOfCompartments>
    <listOfSpecies>
      <species id="a" name="a" compartment="comp"
initialConcentration="4.98161634948818e-22"/>
      <species id="e" name="e" compartment="comp" initialConcentration="0"/>
    </listOfSpecies>
    <listOfParameters>
      <parameter id="parameter1" value="0.001" units="per_second"/>
      <parameter id="a1" value="1.66053878316273e-22" units="mole"/>
    </listOfParameters>
    <listOfReactions>
      <reaction id="reaction1" reversible="false">
        <listOfReactants>
          <speciesReference species="a"/>
        </listOfReactants>
        <listOfProducts>
          <speciesReference species="e"/>
        </listOfProducts>
        <kineticLaw>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <times/>
              <ci> parameter1 </ci>
              <ci> comp </ci>
              <ci> a </ci>
            </apply>
          </math>
        </kineticLaw>
      </reaction>
    </listOfReactions>
    <listOfEvents>
      <event id="event1">
        <trigger>
          <math xmlns="http://www.w3.org/1998/Math/MathML">
            <apply>
              <eq/>
              <csymbol encoding="text"
definitionURL="http://www.sbml.org/sbml/symbols/time"> t </csymbol>
              <cn> 50 </cn>
            </apply>
          </math>
        </trigger>
        <listOfEventAssignments>
          <eventAssignment variable="a">
            <math xmlns="http://www.w3.org/1998/Math/MathML">
              <apply>
                <divide/>
                <apply>
                  <plus/>
                  <apply>
                    <times/>
                    <ci> a </ci>
                    <ci> comp </ci>
                  </apply>
                  <ci> a1 </ci>
                </apply>
                <ci> comp </ci>
              </apply>
            </math>
          </eventAssignment>
        </listOfEventAssignments>
      </event>
    </listOfEvents>
  </model>
</sbml>
```