7 Studying biomolecular systems with BioSpi 2.0: A compositional abstraction of glycogen biosynthesis

The incorporation of a stochastic semantics allows us to build more accurate abstractions of biomolecular systems. In this section we study glycogen biosynthesis as one example of these benefits. In this example, we use the distinction between channels with finite rates and instantaneous channels to build a detailed model of glycogen biosynthesis. We also extensively use an arithmetic extension of the calculus and its ability to abstract the *composition* of entities of growing complexity from simple building blocks.

7.1 Glycogen biosynthesis

Polymerization is a key event in the creation of various biomolecules including DNA, RNA, proteins, various prtoein filaments (*e.g.* actin, microtubuli) and polysaccharides. In contrast to proteins and nucleic acids, polysaccharides can form both branched and linear polymers.

Glycogen is a branched polysaccharide composed of glucose monomers [68]. Glycogen is biosynthesized in a combination of two alternative enzymatic reactions. In the first *elongation* reaction, catalyzed by the glycogen synthase enzyme, an $\alpha(1 \rightarrow 4)$ glycosidic bond is formed between the C_4 of the polymerized glucose at the end of a growing polymer and the C_1 of an "incoming" UDP-glucose monomer.[10] In the second *branching* reaction, catalyzed by the glycogen branching enzyme, an $\alpha(1 \rightarrow 6)$ glycosidic bond is formed between the C_6 of a polymerized glucose within the glycogen polymer and the C_1 of an "incoming" glycogen oligomer. The oligomer is branched "off" an existing polymer chain. In addition, the glycogen polymer is *initiated* by a third enzyme, called glycogenin. This enzyme forms an initial 7-residue primer, which we term a *seed*.

The biosynthesis of glycogen follows specific "rules". First, the specificity of glycogen synthase ensures that elongation is allowed only from growing 4' ends. After branching, there may be more than one such end, and elongation may proceed independently at each of these ends. The branching "rules" are more complex. A branch is specifically created by transferring a 7-residue segment from the end of one chain to the $C_6 - OH$ group of a polymerized glucose residue on the same or another glycogen chain. Each transferred segment must come from a chain of at least 11 residues, and the new branch point must be at least 4 residues away from any other branch point.

The balance between elongation and branching determines the architecture of the glycogen molecule: chain length and extent of branching. This architecture determines the compactness of glucose storage and its availability and is thus critical for the functional role of glycogen as an energy reserve.

[10] The glucose $\rightarrow$ UDP-glucose reaction is catalyzed by a separate enzyme which we will not discuss here. We will use the terms glucose and UDP-glucose interchangeably from here on.

7.2 Building the abstraction in the π-calculus

The first step in building an abstraction is identifying and defining the basic entities in the real world domain. To this end, we now break down the above description into several components.

First, each glycogen strand is directional (Figure 15). We term one end, with a C_1 carbon that cannot grow, as the *root side*. We term the other end, with a C_4 carbon that can grow, as the *leaf side*.

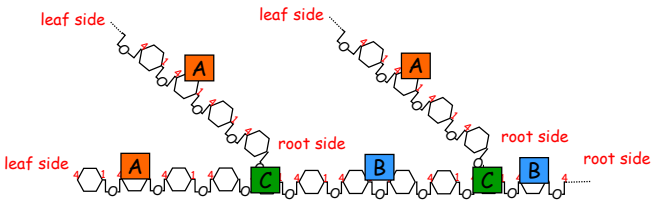

Fig. 15. The architecture of a glycogen molecule: definitions. The C_1 and C_4 end of a glycogen strand defined as "Root" and "Leaf" respectively. A polymerized residue may assume one of three potential positions: on a straight segment (A), on a branched segment (B) or at a branch point (C).

Second, we distinguish three potential positions for a polymerized residue (Figure 15). First, it may reside on a *straight segment*, where no branch points exist to the leaf side. Second, it may be part of a *branched segment*, that has a branch point to the leaf side. Third, the residue may be the *branch point* itself.

The exact position of the residue determines the types of reactions in which it may participate. A *polymerization enabled* residue may participate in an elongation reaction (as the C_4 donor). A *cleavage enabled* residue may be cleaved in a branching reaction and linked into another glycogen branch. A *branch enabled* residue may serve as a branch point onto which a new glycogen oligomer is added. The various types of residues and their reaction capabilities are listed in Table 3.

We are now ready to abstract the system in the stochastic π-calculus. First, each glucose residue is abstracted as a process. Different process states represent the different positions of a glucose residue (Table 3, rightmost column). Processes also represent the two types of enzyme molecules (*Glycogen_Synthase* and *Branching_Enzyme*) and the seed (*Seed_Glucose* for the "free" seed and *Root_Glucose* for the polymerized seed). The formation of bonds is abstracted as private channels, with one private channel shared between each pair of immediate neighbors. Thus, each process representing a fully polymerized residue has two private channels: one linking it with its

Table 3. Different types of residues defined by their reaction capabilities and abstracted as π-calculus processes.

General position	Residue capabilities	Detailed position	Process name
Non-polymerized	Polymerization enabled	Free UDP-glucose (1' end)	*UDP_Glucose*
Straight segment	Polymerization enabled	Leaf (4' end)	*Leaf_Glucose*
	Cleavage and branch enabled	At distance 7 exactly from leaf and at distance 4 at least from closest branch point or root	*BCE_Glucose*
	Branch but not cleavage enabled	At distance other than 7 from leaf and at distance 4 at least from closest branch point or root	*BNCE_ Glucose*
	Disabled	Not leaf and distance less than 4 from closest branch point or root	*Disabled_ Glucose*
Branched segment	Branch but not cleavage enabled	At distance at least 4 from both flanking branch points/root	*BNCE_ Glucose*
	Disabled	At distance less than 4 from at least one of its flanking branch points/root	*Disabled_ Branched_ Glucose*
Branch point	Disabled	At the branch point (C_6 donor)	*Branch_Point*

root-side neighbor and the other with its leaf-side neighbor. The processes representing the root and leaf processes share only a one private channel with a neighbor (Figure 16).

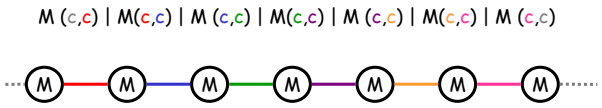

Fig. 16. A polymer abstracted as a chain of processes linked by private channels. Circles represent processes. Connecting colored lines represent individual private channels.

The exact position of a residue is critical to determine its behavior. Similarly, the exact "position" of the corresponding process is critical to determine its state. We abstract a residue's position in the glycogen polymer by the *values* of three *position variables* that are associated with each individual residue process (Figure 17): A

- **Leaf counter (LC)** measures the distance of the residue from the leaf. It is initialized to zero, incremented (+1) upon extension, and decreased (-7) upon cleavage in the remaining segment residues.
- **Root counter (RC)** measures the distance of the residue from root or from the flanking branch point on the root side. It is initialized to the value of the RC variable of the neighbor residue on the root side + 1, updated upon cleavage in the cleaved segment residues to the RC of the new root side neighbor + 1, and updated upon insertion in the segment on the leaf side of the new branch point to RC - (RC of new branch point).
- **Leaf-branch counter (LBC)** measures the distance of the residue from the flanking branch point on the leaf side. It is initialized to zero and updated upon insertion in the segment on the root-side of the new branch point to the LBC of the leaf side neighbor+1.

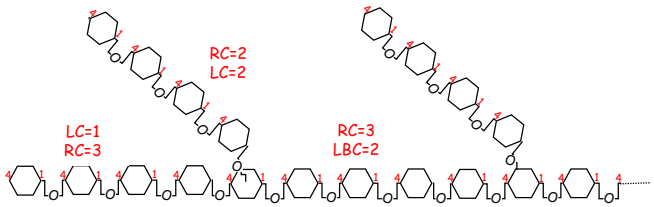

Fig. 17. The leaf (LC), root (RC), and leaf-branch (LBC) counters: An example. The values of the LC, RC, and LBC counters for several residues are shown.

As counters abstract the position of each residue in the polymer, and as the residue's position determines its state, we use the counters to define the corresponding process state. We use a *choice* construct, in which the counter values are checked according to the different rules:

$$
\begin{array}{ll}
\{LBC \ = \ 0\} \ , & \%\text{Straight segment} \\
 & (\ \{LC \ = \ 0\} \ , \ Leaf\_Glucose \ ; \\
 & \{LC = 7 \ , \ RC >= 4\} \ , \ BCE\_Glucose \ ; \\
 & \{LC > 0 \ , \ LC = / = 7 \ , \ RC >= 4\} \ , \ BNCE\_Glucose \ ; \\
 & \{LC > 0 \ , \ RC < 4\} \ , \ Disabled\_Glucose \ ) \ ; \\
\{LBC > 0\} \ , & \%\text{Branched segment} \\
 & (\ \{RC >= 4 \ , \ LBC >= 4\}, BNCE_Glucose; \\
 & \{RC < 4\} \ , \ Disabled\_Branched\_Glucose; \\
 & \{LBC < 4\} \ , \ Disabled\_Branched\_Glucose \ ) \ .
\end{array}
$$

The choice construct is examined each time that the counter value changes.

The position of a residue may change *directly*, when the residue participates in an interaction (polymerization or cleave/branch) or *indirectly*, when the residue is part of a segment which participates in those reactions (*e.g.* upon branching). In order for the abstract counters to correctly represent the position of the corresponding residues in the "real world" polymer, they must be constantly updated. Since only two processes participate in the immediate interaction, we incorporate an *update propagate mechanism* into our abstract model.

Counter updating will be performed by communication on the private channels linking the process residues. Importantly, all such private channels will be instantaneous, while all the communications representing *actual* reactions (elongation, cleavage and branching) will be carried out on channels with a real finite rate. As a result, counter updating will be done in zero simulation time, and will not interfere with the kinetics of biochemical reactions, while allowing us to maintain a detailed view of the glucose monomer's position within the glycogen polymer.

The full abstraction of the glycogen biosynthetic system in the π-calculus is given in Figure 18. The enzymatic elongation reaction is represented by the *glycogen* and *udp_glucose* channels, and the cleave and branch reaction by the *branch* and *cleave* channels. Each $UDP\_Glucose$ process is defined with two private channels (*to_root* and to_leaf) that will be subsequently used as specific links to residues in the resulting polymer. For simplicity, all the "reaction" channels are given an equal base rate of 1 (we will revisit this decision below), while all private channels are instantaneous with an infinite base rate.

Abstraction of initiation

In the initiation step (after a seed oligomer is already produced by glycogenin), glycogen synthase catalyzes the formation of a glycosidic bond between a UDP-glucose and the 4' leaf end of the seed oligomer. In the abstract π-calculus model we represent the initiation as two consecutive communication steps. In the first step, the *to_root* private channel of $UDP\_Glucose$ is sent to $Glycogen\_Synthase$ on the *udp_glucose* channel. In the second step, this private channel name is further relayed in a communication from $Glycogen\_Synthase$ to $Seed\_Glucose$ on *glycogen* (Figure 19A and B).

Following the two communications, $UDP\_Glucose$ and $Seed\_Glucose$ now share a private (instantaneous) channel. In the next two steps, the position counters of each of the processes are updated to reflect their relative position using this shared channel(Figure 19C and D). First, the RC counter is sent on the private channel from $Seed\_Glucose$ to $UDP\_Glucose$, and is used to update $UDP\_Glucose$'s RC value. This represents the change in the position of this now polymerized residue relative to its newly acquired root. Then, the LC and LBC counters are sent on the private channel from $UDP\_Glucose$

to the now *Root_Glucose*, and are used to update the respective counters in *Root_Glucose*. This represents the concomitant change in the position of the root oligomer relative to its newly acquired leaf. Finally ((Figure 19E), the new counter values are evaluated in $UDP\_Glucose$, changing it to its new $Leaf\_Glucose$ state.

Abstraction of elongation

Elongation is similar to initiation and involved the same processes and communications, with the exception of *Seed_Glucose* which is "replaced" by $Leaf\_Glucose$. Two consecutive communications (first on *udp_glucose* and then on *glycogen*) serve to relay a private channel from $UDP\_Glucose$ to $Leaf\_Glucose$ via *Glycogen_Synthase*. This private channel is then used to initiate an update of the positional counters of *all* the processes along the growing chain (each time using the relevant private channel). Finally, the updated counters determine the new state of each of the linked processes.

Abstraction of branching

Consider a relatively simple branching scenario in which a 12-residue chain is cleaved at the fifth residue. The 4-residue "rooted" part is then elongated by

```
global(glycogen(1), udp_glucose(1), dummy(1), branch(1), cleave(1)).
baserate(infinite).

Seed_Glucose(RC,LC,LBC)::= glycogen ? {to_leaf},to_leaf ! {RC,LBC},
                                   Root_Glucose(to_leaf,RC,LC,LBC).
Root_Glucose(to_leaf,RC,LC,LBC)::=
   to_leaf ? {LC,LBC} , {LC++} , Root_Glucose(to_leaf,RC,LC,LBC) .
UDP_Glucose(LC,LBC)+(to_root,to_leaf)::=
   udp_glucose ! {to_root} , to_root ? {RC,LBC} , {RC++} ,
         to_root ! {LC,LBC} , Glucose(to_root,to_leaf,RC,LC,LBC) .
Glucose(to_root, to_leaf, RC, LC, LBC)::=
{LC>=0},
<< {LBC = 0} , << {LC = 0} ,  Leaf_Glucose ;
            {LC = 7 , RC >= 4} , BCE_Glucose ;
  {LC > 0 , LC =\= 7 , RC >= 4} , BNCE_Glucose ;
  {LC > 0 , RC < 4} , Disabled_Glucose >> ;
   {LBC > 0} , << {RC >= 4 , LBC >=4} , BNCE_Glucose ;
          {RC < 4} , Disabled_Branched_Glucose ;
  {LBC < 4} , Disabled_Branched_Glucose >> .
```

Fig. 18. π-calculus code for the glycogen system. The entire system is shown with the exception of initialization of position counters. The code is continued on the next page.

```
Leaf_Glucose::=
   glycogen ? {to_leaf} , to_leaf ! {RC,LBC} , to_leaf ? {LC,LBC} , {LC++} ,
                            to_root ! {LC,LBC} , Glucose(to_root,to_leaf,RC,LC,LBC);
   to_root ? {RC,_} , <<  {RC >=0} , {RC++} , Glucose ;
  {RC < 0} , Disabled_Leaf_Glucose >> .
Disabled_Leaf_Glucose::=
   to_root ? {RC,_} , {RC++} , Glucose .
BNCE_Glucose::=
   to_leaf ? {LC,LBC} , {LC++} , << {LBC = 0} , to_root ! {LC,LBC} , Glucose ;
               {LBC > 0} , {LBC++} ,  to_root ! {LC,LBC} , Glucose >> ;
   to_root ? {RC,_} , << {RC >=0} , {RC++} , to_leaf ! {RC,LBC} , Glucose ;
 {RC < 0} , to_leaf ! {RC, LBC} , Disabled_Glucose >> ;
   branch ? {to_branch} , Branch_Synch1(to_branch,RC,LC,LBC) .
Branch_Synch1(to_branch,RC,LC,LBC)+(RC1,LBC1)::=
  {RC1=0} | {LBC1=1} |
    << to_branch ! {RC1,LBC} ,
        << to_leaf ! {RC1,LBC} , to_root ! {LC,LBC1} ,
             Branch_Point(to_root,to_branch,to_leaf) >> >>  .
Disabled_Glucose::=
   to_leaf ? {LC,LBC} , {LC++} ,
         << {LBC = 0} , to_root ! {LC,LBC} , Glucose ;
    {LBC > 0} , {LBC++} ,  to_root ! {LC,LBC} , Glucose >> ;
   to_root ? {RC,_} , {RC++} , to_leaf ! {RC,LBC} , Glucose .
BCE_Glucose+(new_to_root,RC1,LC1,LBC1)::=
   << to_leaf ? {LC,LBC} , {LC++} , << {LBC = 0} , to_root ! {LC,LBC} , Glucose ;
                {LBC > 0} , {LBC++} ,  to_root ! {LC,LBC} , Glucose >> ;
      to_root ? {RC,_} , << {RC >=0} , {RC++} , to_leaf ! {RC,LBC} , Glucose ;
    {RC < 0} , to_leaf ! {RC,LBC} , Disabled_Glucose >> ;
      branch ? {to_branch} , Branch_Synch(to_branch,RC,LC,LBC) ;
      cleave ! {new_to_root} , {LC1 = -1} | {RC1 = -1} | Cleave_Synch(to_leaf) .
Cleave_Synch(to_leaf)::=
   to_root ! {LC1,LBC} , to_leaf ! {RC1, LBC} , new_to_root ? {RC,_} ,
{RC++} , to_leaf ! {RC,LBC} , Glucose(new_to_root,to_leaf, RC, LC, LBC) >> .
Branch_Synch(to_branch,RC,LC,LBC)+(RC1,LBC1)::=
   {RC1=0} | {LBC1=1} |
                << to_branch ! {RC1,LBC} ,
             << to_leaf ! {RC1,LBC} ,
 to_root ! {LC,LBC1} , Branch_Point(to_root,to_branch,to_leaf) >> >> .
Disabled_Branched_Glucose::=
   to_leaf ? {LC,LBC} , {LC++} ,
        << {LBC = 0} , to_root ! {LC,LBC} , Glucose ;
           {LBC > 0} , {LBC++} ,  to_root ! {LC,LBC} , Glucose >> ;
   to_root ? {RC,_} , {RC++} , to_leaf ! {RC,LBC} , Glucose >> .
Branch_Point(to_root,to_branch,to_leaf)::=
   to_root ? {_,_} , self ;
   to_branch ? {_,_} , self ;
   to_leaf ? {_,_} , self .
Glycogen_Synthase::=
   udp_glucose ? {to_root} , glycogen ! {to_root} , Glycogen_Synthase  .
Branching_Enzyme::=
   cleave ? {to_branch} , branch ! {to_branch} , Branching_Enzyme .
```

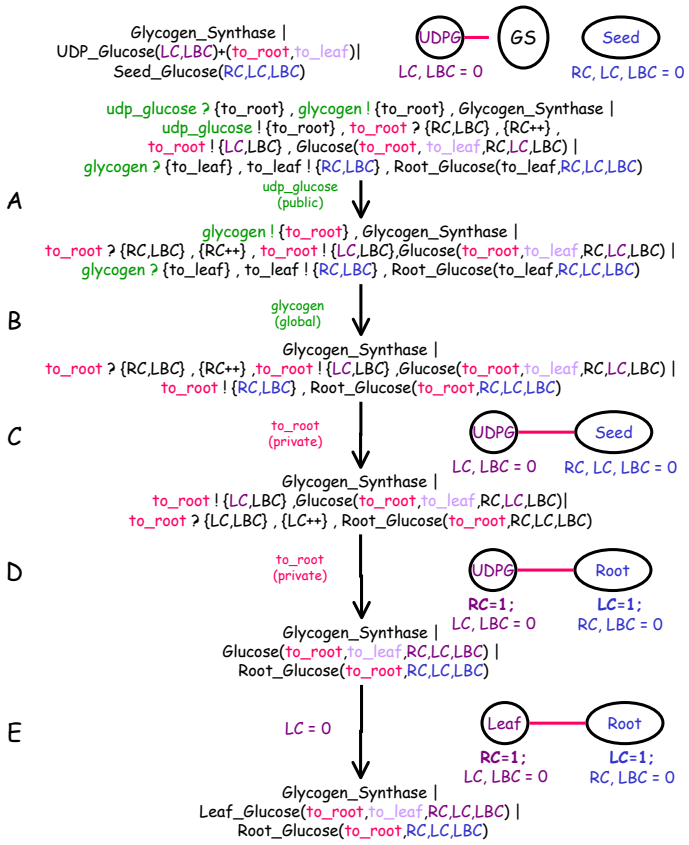

Fig. 19. Initiation as a multi-step communication between $UDP\_Glucose$**,** $Glycogen\_Synthase$ **and** $Seed\_Glucose$**.** A. Communication between $UDP\_Glucose$ and $Glycogen\_Synthase$. B. Communication between $Glycogen\_Synthase$ and $Seed\_Glucose$. C,D. Counter update using a shared private channel. E. $UDP\_Glucose$ changed to $Leaf\_Glucose$ based on counter values.

two more residues, followed by linking of the 8-residue "clipped" part on its fifth residue (Figure 20).

A process chain of size 12 is shown in Figure 20, where process state is determined according to its relative position. Each pair of processes is linked by a unique private channel, established during the elongation step, as described above. Cleavage is abstracted as a communication on *cleave* between a $Branching\_Enzyme$ and a $BCE\_Glucose$. Due to the short chain length in this example, there is only a single $BCE\_Glucose$. In the communication, a private channel (new_to_root) is sent from $BCE\_Glucose$ to

Branching_Enzyme. Following the communication, the *LC* and *RC* counters of *BCE_Glucose* are set to special values (-1), followed by a series of communications on the chains of private channels to update the relevant positional information. First, the "cleave event" is propagated to the root side of the cleave point. As a result, the immediate root-side neighbor of the cleaved residue process will become a *Leaf_Glucose*. The other root-side processes will be updated accordingly with the communication propagated up to the *Root_Glucose*.

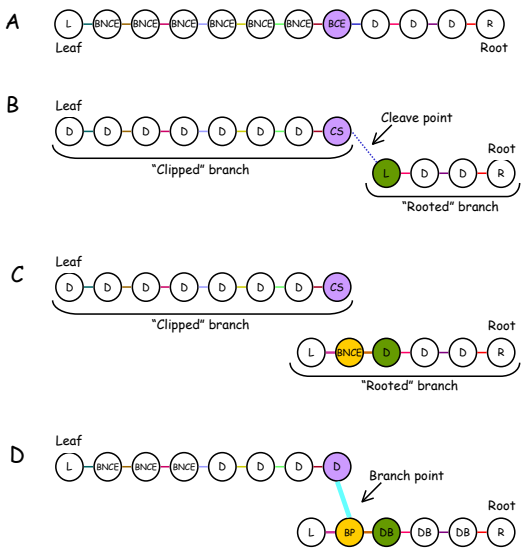

Fig. 20. A simple branch and cleave scenario. A. A chain of length 12. B. Following cleavage: a rooted chain (length 4) and a clipped chain (length 8). C. The rooted chain is elongated by two more residues. D. Linking the clipped chain to residue 5 of the rooted chain. Corresponding process names are shown (R - *Root_Glucose*, L - *Leaf_Glucose*, D - *Disabled_Glucose*, BP - *Branch_Point*, DB - *Disabled_Branched_Glucose*, CS - *Cleave_Synch*)

In parallel, the cleavage event is propagated to the leaf side of the cleavage point. Since cleavage and branching may be separated by additional steps (*e.g.* in the example scenario we have additional elongation of the "rooted branch"), the "clipped" branch is "frozen" until it is be linked to a new position. The "freeze" is based on propagating the special *RC* counter value (-1) throughout the clipped chain. The special value serves to set each process to a *Disabled_Glucose*, and will change only upon updating the positional counters (following branch linking, see below). On the other hand, the "rooted" chain

may participate like any other chain in elongation (*e.g.* the two elongation steps in our example scenario).

We now examine how to abstract the linking of a cleaved ("frozen") branch to form a branch point. Assuming two additional elongation steps, we start with a system with two process chains, one representing a "rooted" branch of size 6 and the other representing a cleaved branch of size 8. Branching is represented by a communication between *Branching_Enzyme* and *BNCE_Glucose* on *branch*, in which the private *new_to_root* channel, originating in the "cleaved" *BCE_Glucose* is relayed to *BNCE_Glucose*. This results in a private link between the two *Glucose* processes. A series of communications on private channels propagate the change as an update in positional counters from the cleave point (to the leaf direction, resulting in "unfreezing" of the cleaved branch) and from the branch point (to both the root and leaf direction).

7.3 Studying glycogen metabolism with BioSpi 2.0

The detailed glycogen biosynthesis model, based on a "monomer-as-process" abstraction, allows us to monitor the exact architecture of glycogen molecules through the architecture of the process "polymer". The tracing tools of BioSpi 2.0 provide us with the information (process names, counter values and private channel identity) necessary to reconstruct this architecture.

A simple example, based on a system initialized with a single *Seed_Glucose*, 15 *UDP_Glucose* processes, a single *Glycogen_Synthase* process and a single *Branching_Enzyme* is shown in Figure 21. The processes available in the system after it is run until suspension (no additional enabled communications) are shown in the so-called resolvent in Figure 21A, together with the channels they use and the values for the *RC*, *LC* and *LBC* counters. This information is sufficient to specify the molecular architecture of the corresponding glycogen molecule (Figure 21B).

BioSpi 2.0 also allows us to follow the time evolution of various quantitative properties in the abstracted population of glycogen molecules, such as the number of branch points, leaves and seeds, giving us an overall measure of molecular architecture. For example, Figure 22 shows the number of various processes representing different properties of the polymers (leaves, polymerized residues etc.) under different relative polymerization and branching rates.

8 Perspectives: Molecules as computation

The behavior of biomolecular systems is traditionally studied with *Dynamical Systems Theory* (described via differential equations) [16], which abstracts the cell and its molecular constituents to their quantifiable properties (e.g. concentration, position) and their couplings. While this approach is powerful,

A

```
...
.Root_Glucose.comm(.UDP_Glucose.to_root!)
Disabled_Branched_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_root!, 1, 4,
4, global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
Disabled_Branched_Glucose.comm(.UDP_Glucose.to_root!,.UDP_Glucose.to_root!, 2, 3,
3, global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
Disabled_Branched_Glucose.comm(.UDP_Glucose.to_root!,.UDP_Glucose.to_root!, 3, 2,
2, global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
.Branch_Point.comm(.UDP_Glucose.to_root!, BCE_Glucose.new_to_root!,
.UDP_Glucose.to_root!)
Disabled_Glucose.comm(BCE_Glucose.new_to_root!,.UDP_Glucose.to_root!, 1, 8, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
Disabled_Glucose.comm(.UDP_Glucose.to_root!,.UDP_Glucose.to_root!, 2, 7, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
Disabled_Glucose.comm(.UDP_Glucose.to_root!,.UDP_Glucose.to_root!, 3, 6, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
BNCE_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_root!, 4, 5, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
BNCE_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_root!, 5, 4, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
BNCE_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_root!, 6, 3, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
BNCE_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_root!, 7, 2, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
BNCE_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_root!, 8, 1, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
Disabled_Glucose.comm(.UDP_Glucose.to_root!,.UDP_Glucose.to_root!, 1, 1,
0, global.branch(1)!, global.cleave(1)!,global.glycogen(1)!)
Leaf_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_leaf, 2, 0, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
Leaf_Glucose.comm(.UDP_Glucose.to_root!, .UDP_Glucose.to_leaf, 9, 0, 0,
global.branch(1)!, global.cleave(1)!, global.glycogen(1)!)
.Glycogen_Synthase.comm(global.glycogen(1)!, global.udp_glucose(1)!)
.Branching_Enzyme.comm(global.branch(1)!, global.cleave(1)!)
```

B

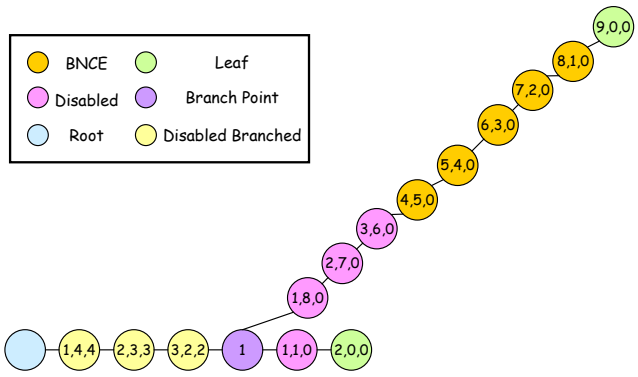

Fig. 21. A BioSpi 2.0 resolvent allows reconstruction of the molecular architecture of a glycogen molecule. A. A BioSpi resolvent following a complete run of a system initialized with a single *Seed_Glucose*, 15 *UDP_Glucose*, a single *Glycogen_Synthase* and a single *Branching_Enzyme*. B. The molecular architecture reconstructed from the resolvent. Counter values for *RC*, *LC* and *LBC* shown inside circles.

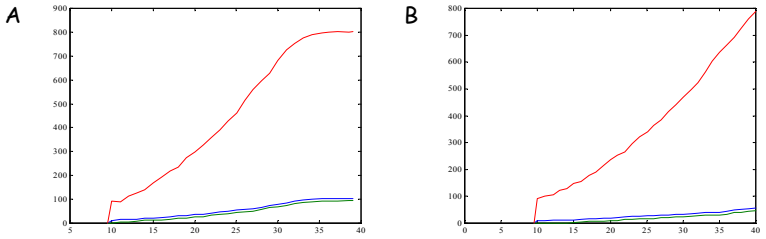

Fig. 22. BioSpi 2.0 simulation of glycogen synthesis under different conditions. The number of *Branch_point* (green), *Leaf_Glucoose* (blue) and "polymerized" (sum of *BCE_Glucose*, *BNCE_Glucose*, *Diabled_Glucose* and *Disabled_Branched_Glucose*) processes (red) is shown as a function of time either when the base rates of the polymerization and branching communications are equal (A) or when the polymerization rate is 10 times faster (B).

clear, and well understood, the abstraction it makes is problematic when describing the behavior of complex biomolecular systems [16]. It treats the cell as a monolithic unstructured entity, and does not identify molecular *objects* as they are modified. Biological systems are composed of dynamic molecular *objects*, but Dynamical Systems Theory is not compositional. Thus, it only indirectly captures the identity and fate of a single molecule, molecular complex or machine with several states (and/or sub-components).

8.1 The molecule-as-computation abstraction

Computer science offers an alternative starting point in the search for good abstractions of biological systems composed of dynamically changing objects. The view of computational systems as composed of interacting processes, renders them a tempting source of novel abstractions for the behavior of systems of interacting biological entities molecules, such as molecules and cells. In this work, we began the search for this abstraction with the π-calculus, a process algebra for the representation and study of *concurrent mobile systems*.

As a first step we identified the basic entities of biomolecular systems and the events in which they participate. We then developed general guidelines for the abstraction of these entities to the mathematical domain of the π-calculus, which we then employed for the modeling and study of real complex biomolecular systems, such as the receptor tyrosine kinase (RTK) - MAP kinase (MAPK) signaling pathway [52]. While the function of certain biomolecular systems may be abstracted in a qualitative or semi-quantitative way, the functionality of others, critically depends on quantitative aspects. We therefore further extended the original abstraction with a stochastic component [47], embedding the well-accepted Gillespie algorithm [18] within the