

Simplicial Models for Concurrency

Peter Bubenik^{1,2}

*Department of Mathematics
Cleveland State University
Cleveland, USA*

Abstract

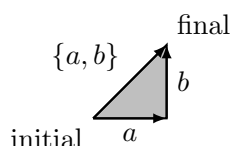
We model both concurrent programs and the possible executions from one state to another in a concurrent program using simplices. The latter are calculated using necklaces of simplices in the former.

Keywords: Concurrency; Simplicial set; Simplicial category; Necklace

1 Introduction

We develop a discrete model for processes that is useful in the concurrent setting. It is based on the traditional model given by graphs. In this model, vertices of the graph represent states and edges represent transitions. The edges may be directed or undirected.

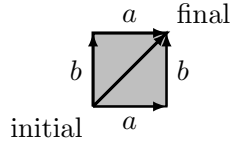
A more sophisticated model appropriate for concurrency is given by simplicial complexes. For example, the following triangle,



models two processes a and b where the boundary models two executions, either a followed by b , or a and b concurrently. The interior of the triangle models intermediate executions. Adding another triangle we obtain a simplicial complex that models all possible executions of a and b .

¹ The author would like to thank David Spivak for many useful conversations pertaining to this article. He would also like to thank the NSF for funding some of this research through the Midwest Topology Network (NSF grant DMS-0844249). In addition he would like to thank Martin Raussen and Lisbeth Fajstrup for organizing the GETCO 2010 conference that motivated this work.

² Email: p.bubenik@csuohio.edu



For three processes, the analogous model is a cube that is subdivided into six tetrahedra.

The combinatorial versions of simplicial complexes are simplicial sets. These consist of sets of abstract vertices, edges, triangles, tetrahedra, and higher dimensional simplices. For a simplicial set, the corresponding simplicial complex is its geometric realization. Our figures will consist of simplicial complexes, but we intend these to represent the corresponding simplicial set.

Using Dijkstra’s formalism for concurrent programs [4], we give an explicit construction of a simplicial model for a concurrent program using filtered simplicial sets. The vertices of this simplicial model give states of the concurrent program. The executions between states are modeled by paths between vertices in the simplicial model. There is a growing literature of related models, which are mostly continuous. For some examples, see [7,8,2,3,12,11,6].

In concurrency, understanding the possible execution paths is one of the main sources of difficulty. In a simplicial set, the equivalence classes of paths can be described by a category called the fundamental category [10] or the path category [9]. In this category the objects are the vertices of the simplicial set, and between objects there is a set of equivalence classes of paths.

We consider a related but more sophisticated construction in which the objects are vertices of the simplicial set, but the executions from one state to another are described by a simplicial set. That is, our model is a simplicial category (i.e., a category enriched over simplicial sets). This construction can detect higher order structure that is invisible to the path category. To calculate it, we consider necklaces of simplices in our simplicial set models. In a related paper [14], Raussen constructs simplicial models of execution spaces for continuous models.

Outline of the paper: In Section 2, we discuss simplicial sets, filtered simplicial sets, necklaces, simplicial categories, and some of the other necessary mathematical machinery. In Section 3, we apply these constructions to construct simplicial models of concurrent systems and also of their execution spaces. Our examples include an example in which the structure of the executions is not detected by the path category, but is captured by our methods. In Section 4, we discuss a suitable model structure for this setting and we remark that one may try to apply these methods using cubical sets instead of simplicial sets.

2 Mathematical machinery

2.1 Simplicial sets

A *simplicial set*, X , consists of a sequence of sets, $\{X_0, X_1, X_2, \dots\}$ together with *face maps* $d_i : X_k \rightarrow X_{k-1}$ for $0 \leq i \leq k$ and *degeneracy maps* $s_i : X_k \rightarrow X_{k+1}$ for

$0 \leq i \leq k$ satisfying the following *simplicial identities*.

$$\begin{aligned} d_i d_j &= d_{j-1} d_i \text{ if } i < j \\ s_i s_j &= s_{j+1} s_i \text{ if } i \leq j \\ d_i s_j &= s_{j-1} d_i \text{ if } i < j \\ d_j s_j &= \text{Id} = d_{j+1} s_j \\ d_i s_j &= s_j d_{i-1} \text{ if } i > j + 1 \end{aligned}$$

The elements of X_k are called *k-simplices*. The 0-simplices are called *vertices* and the 1-simplices are called *edges*. A morphism of simplicial sets, $f : X \rightarrow Y$ consists of a sequence of functions $f_k : X_k \rightarrow Y_k$ that commute with the face and degeneracy maps. The geometric realization of a simplicial set is a simplicial complex.

For example, for each n there is a simplicial set Δ^n whose geometric realization is the standard geometric n -simplex. Δ^0 is the simplicial set with one element in each X_k and all of the face and degeneracy maps given by the identity. The k -simplices that are not in the image of a degeneracy map are called *nondegenerate*. Thus, Δ^0 has only one nondegenerate simplex, which is a 0-simplex. The only nondegenerate simplices in Δ^1 are $a, b \in \Delta_0^1$ and $e \in \Delta_1^1$ where $d_0(e) = b$ and $d_1(e) = a$. Note that the initial vertex is $d_1(e)$ and the final vertex is $d_0(e)$.

Simplicial sets and their morphisms form a category. Geometric realization is a functor from this category to the category of topological spaces. A more elegant but more sophisticated way of defining simplicial sets is as contravariant functors from the category of finite ordinals and order preserving maps to the category of sets.

2.2 Filtered simplicial sets

A sub-simplicial set $A \subseteq X$ is a sequence of subsets $A_i \subseteq X_i$ that are closed under the restrictions of the face and degeneracy maps. A *filtration* on a simplicial set X is an increasing sequence $X_{(0)} \subseteq X_{(1)} \subseteq X_{(2)} \subseteq \dots$ of sub-simplicial sets of X . The simplices in $X_{(d)}$ are said to be in degree d . A *m-filtration* on X is a collection of sub-simplicial sets $X_{(i_1, \dots, i_m)}$ of X such that if $i_j \leq i'_j$ for $1 \leq j \leq m$ then $X_{(i_1, \dots, i_m)}$ is a sub-simplicial set of $X_{(i'_1, \dots, i'_m)}$.

For simplicial sets X and Y , their product $X \times Y$ is given by $(X \times Y)_n = X_n \times Y_n$ with face and degeneracy maps $d_i = d_i \times d_i$ and $s_i = s_i \times s_i$. Note that the nondegenerate simplices of $X \times Y$ are not necessarily products of nondegenerate simplices of X and Y . A standard example is $\Delta^1 \times \Delta^1$ which is a triangulation of the square. For filtered simplicial sets X and Y , $X \times Y$ has an induced filtration where $(X \times Y)_{(k)}$ is the union of the sub-simplicial sets $X_{(i)} \times Y_{(j)}$ where $i + j = k$. Similarly, products of m -filtered simplicial sets have an induced m -filtration.

2.3 The path category of a simplicial set

Given a simplicial set X , we can define the path category, $P(X)$, as follows. For a thorough exposition see [9]. The path category is also called the fundamental

category.

Definition 2.1 Let the *path category* $P(X)$ of the simplicial set X be the category whose objects are the vertices of X and whose morphisms are concatenations of edges in X modulo the equivalence relation generated by the following relations

$$\begin{aligned}s_0(a) &\sim \text{Id}_a, \quad \text{for all } a \in X_0, \text{ and} \\ d_1(t) &\sim d_2(t)d_0(t), \quad \text{for all } t \in X_2.\end{aligned}$$

2.4 The simplicial category of an ordered simplicial set

Since the path category only depends on the k -simplices for $k \leq 2$, it does not detect higher order structure.

For this purpose, we consider the following construction of a category enriched over simplicial sets. In such a category, between two objects, instead of set of morphisms, we have a simplicial set of morphisms. For brevity, this is also called a simplicially enriched category, or a simplicial category. For a thorough exposition of the simplicial category of a simplicial set see [5].

A *necklace* is a simplicial set, T , of the form

$$\Delta^{n_1} \vee \Delta^{n_2} \vee \dots \vee \Delta^{n_k}$$

where the final vertex of Δ^{n_i} is glued to the initial vertex of $\Delta^{n_{i+1}}$. The simplicial set Δ^{n_i} is called a *bead* of T . Vertices of T that are initial or final vertices of any of the beads are called *joints* of T . The set of joints of T is denoted J_T . Call the initial vertex of Δ^{n_1} the initial vertex of T and the final vertex of Δ^{n_k} the final vertex of T .

A *flag* of T is an increasing sequence $\mathbf{T} = (T^0 \subseteq \dots \subseteq T^n)$ of subsets of vertices of T . The *length* of \mathbf{T} is n . A flag is *flanked* if $T^0 = J_T$ and $T^n = T_0$.

A simplicial set X is *ordered* if for each $a \in X_0$, $P(X)(a, a)$ consists of only the identity morphism, and no two simplices have the same set of vertices. The simplicial sets arising in our models in Section 3 will all be ordered. One can define the simplicial category for a simplicial set without this condition, but for ordered simplicial sets we have the following nice explicit construction from [5].

Definition 2.2 For an ordered simplicial set X , let $\mathfrak{C}(X)$ be the category enriched in simplicial sets whose objects are vertices in X , and such that for vertices a and b , $\mathfrak{C}(X)(a, b)$ is the simplicial set given as follows. Let $\mathfrak{C}(X)(a, b)_n$ be the set of triples (T, f, \mathbf{T}) where T is a necklace, $f : T \rightarrow X$ is a injective map of simplicial sets such that the images of the initial and final vertices of T are a and b respectively, and \mathbf{T} is a flanked flag of length n . The degeneracy maps, s_i , for $0 \leq i \leq n$ are given by

$$s_i((T, f, T^0 \subseteq \dots \subseteq T^n)) = (T, f, T^0 \subseteq \dots \subseteq T^i \subseteq T^i \subseteq \dots \subseteq T^n).$$

The face maps, d_i , for $0 < i < n$ are given by

$$d_i((T, f, T^0 \subseteq \dots \subseteq T^n)) = (T, f, T^0 \subseteq \dots \subseteq T^{i-1} \subseteq T^{i+1} \subseteq \dots \subseteq T^n).$$

For $i = 0$, let T' the unique subnecklace of T whose set of joints is T^1 and whose set of vertices is T^n . Similarly, for $i = n$, let T'' be the unique subnecklace of T whose set of joints is T^0 , and whose set of vertices is T^{n-1} . Then

$$d_0((T, f, T^0 \subseteq \cdots \subseteq T^n)) = (T', f, T^1 \subseteq \cdots \subseteq T^n), \text{ and} \\ d_n((T, f, T^0 \subseteq \cdots \subseteq T^n)) = (T'', f, T^0 \subseteq \cdots \subseteq T^{n-1}).$$

3 Concurrency

We use Dijkstra's abstraction of concurrent programming in which *semaphores* are used to control access by multiple processes to a common resource [4]. More precisely, a resource that may be used by only k processes at once is controlled by a k -semaphore. This is simply a nonnegative counter which starts at k . When a process wants to use the resource it tries to decrement the counter and when it is finished using the resource it increments the counter. 1-semaphores are also called *binary semaphores*.

A concurrent program can be abstractly written as a sequence of operations on semaphores. Following Dijkstra's original notation, for a semaphore a , let Pa denote decrementing a and let Va denote incrementing a . We consider programs that are finite sequences of these operations. Abstractly a program for one process is given by

$$P = O_1 a_{i_1}. O_2 a_{i_2}. \cdots . O_N a_{i_N}, \quad (1)$$

where O_j is either operator P or V and a_{i_j} is one of the semaphores. Let s_0 be the initial state of the program P and let s_j be state of the program after the operation $O_j a_{i_j}$.

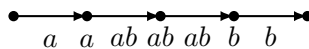
3.1 Simplicial models of concurrent programs

Definition 3.1 The *simplicial model* for the program P for a single process in (1) is modeled by the simplicial set X that is the necklace of N 1-simplices. The vertices of the necklace correspond to states of the program. If there are m shared resources a_1, \dots, a_m we define an m -filtration on this necklace. The initial vertex has degree $(0, \dots, 0)$. The operation Pa_i increases by one the a_i degree of the corresponding edge and the subsequent vertices and edges in the necklace. The operation Va_i decreases by one the a_i degree of the vertices and edges in the necklace following the corresponding edge.

Example 3.2 Let a and b be two binary semaphores. Consider the program

$$Pa.Pb.Va.Vb$$

This program can be modeled by the following bifiltered simplicial set.



Here the unlabeled vertices and edges are in degree $(0, 0)$. The vertices and edges labeled a , b and ab are in degree $(1, 0)$, $(0, 1)$ and $(1, 1)$ respectively.

Assume programs P_1, \dots, P_n with shared resources a_1, \dots, a_m have as simplicial models the m -filtered simplicial sets X_1, \dots, X_n . We want a model for the concurrent program $P = (P_1 | \dots | P_n)$. The simplicial set $X_1 \times \dots \times X_n$ has an induced m -filtration (see Section 2.2). A *state* of P is a vertex $x = (x_1, \dots, x_n) \in X_1 \times \dots \times X_n$.

Definition 3.3 We call a state $x = (x_1, \dots, x_n)$ of P *valid* if across $1 \leq i \leq n$ the uses of resources by P_i at s_{j_i} are compatible, where s_{j_i} is the state of P_i corresponding to the vertex x_i of the model X_i . More precisely, for $1 \leq i \leq n$, $1 \leq j \leq m$ and $x_i \in P_i$, let $D_{ij}(x_i)$ be the amount that a_j has decreased from k_j if the program for the process P_i runs from its initial state to the state x_i . A state $x = (x_1, \dots, x_n)$ is *valid* if for each $1 \leq j \leq m$

$$\sum_{i=1}^n D_{ij}(x_i) \leq k_j.$$

Lemma 3.4 *The set of valid states of P is given by the vertices in $(X_1 \times \dots \times X_n)_{(k_1, \dots, k_m)}$.*

Proof. Consider $x = (x_1, \dots, x_n) \in X_1 \times \dots \times X_n$. Let $1 \leq i \leq n$ and $1 \leq j \leq m$. From the definition of the grading on X_i , x_i has a_j degree $D_{ij}(X_i)$. So the a_j degree of x is $\sum_{i=1}^n D_{ij}(x_i)$. Thus the degree of x is

$$\left(\sum_{i=1}^n D_{i1}(x_i), \dots, \sum_{i=1}^n D_{im}(x_i) \right).$$

Therefore x is valid if and only if $x \in (X_1 \times \dots \times X_n)_{(k_1, \dots, k_m)}$. \square

An edge of $X_1 \times \dots \times X_n$ is of the form (e_1, \dots, e_n) where e_i is an edge of X_i . Note that e_i may be degenerate. That is $e_i = sx_i$ where x_i is a vertex of X_i .

Definition 3.5 Define an edge $e = (e_1, \dots, e_n)$ to be *valid* if across $1 \leq i \leq n$ the uses of resources by P_i from s_{j_i} to $s_{j'_i}$ are compatible, where s_{j_i} and $s_{j'_i}$ are the states corresponding to $d_1(e_i)$ and $d_0(e_i)$, respectively. More precisely, using the notation of Definition 3.3, e is valid if for each $1 \leq j \leq m$,

$$\sum_{i=1}^n \max\{D_{ij}(d_1(e_i)), D_{ij}(d_0(e_i))\} \leq k_j$$

Lemma 3.6 *The set of valid edges of P is given by the edges in $(X_1 \times \dots \times X_n)_{(k_1, \dots, k_m)}$.*

Proof. Consider $e = (e_1, \dots, e_n) \in X_1 \times \dots \times X_n$. Let $1 \leq i \leq n$ and $1 \leq j \leq m$. From the definition of the grading on X_i , e_i has a_j degree equal to the maximum of the a_j degree of its vertices $d_1(e_i)$ and $d_0(e_i)$. Thus the degree of e is

$$\left(\sum_{i=1}^n \max\{D_{i1}(d_1(e_i)), D_{i1}(d_0(e_i))\}, \dots, \sum_{i=1}^n \max\{D_{im}(d_1(e_i)), D_{im}(d_0(e_i))\} \right).$$

Therefore x is valid if and only if $x \in (X_1 \times \cdots \times X_n)_{(k_1, \dots, k_m)}$. \square

Definition 3.7 Assume the program $P = (P_1 | \dots | P_n)$ has shared resources a_1, \dots, a_m where a_i is a k_i -semaphore. The *simplicial model* for P is given by the simplicial set

$$(X_1 \times \cdots \times X_n)_{(k_1, \dots, k_m)}.$$

Example 3.8 Let a and b be two binary semaphores. Consider the two programs

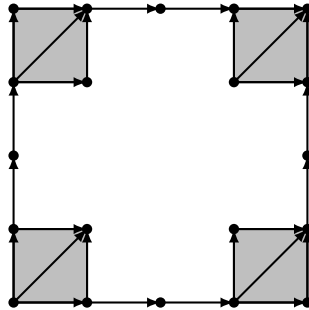
$$A = Pa.Pb.Vb.Va \quad \text{and} \quad B = Pb.Pa.Va.Vb$$

These can be modeled by the following bifiltered simplicial sets \bar{A} and \bar{B} .



Here the unlabeled vertices and edges are in filtration $(0, 0)$. The vertices and edges labeled a , b and ab are in filtration $(1, 0)$, $(0, 1)$ and $(1, 1)$ respectively.

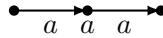
Now consider the concurrent execution of A and B , denoted $(A|B)$. It is modeled by the following simplicial set $(\bar{A} \times \bar{B})_{(1,1)}$.



Example 3.9 Let a be a 2-semaphore. Consider the three identical programs, A , B and C given by

$$Pa.Va$$

This program is modeled by the following filtered simplicial set X .



The concurrent program $(A|B|C)$ is modeled by the simplicial set $(X \times X \times X)_{(2)}$, which is a triangulation of the boundary of the cube.

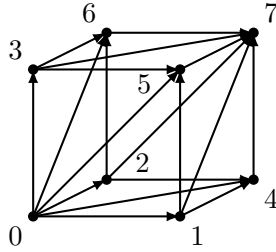
Remark 3.10 One can also define analogous cubical models for concurrent programs. The simplicial models presented here are the triangulations of those cubical models.

3.2 Simplicial models of execution spaces

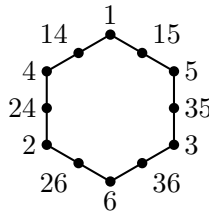
Next we construct simplicial models for the space of executions from one state to another in the simplicial models in Section 3.1.

Definition 3.11 For vertices a and b in a simplicial model X , we define the *simplicial model* of the execution space from a to b to be the simplicial set $\mathfrak{C}(X)(a, b)$.

Example 3.12 Let X be the boundary of the cube labeled as follows.



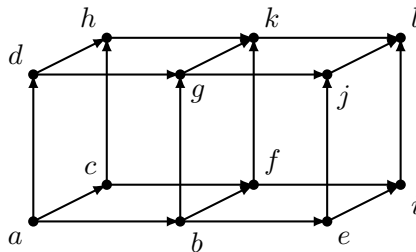
Then $\text{Map}_X(0, 7)$ is given by the following simplicial set.



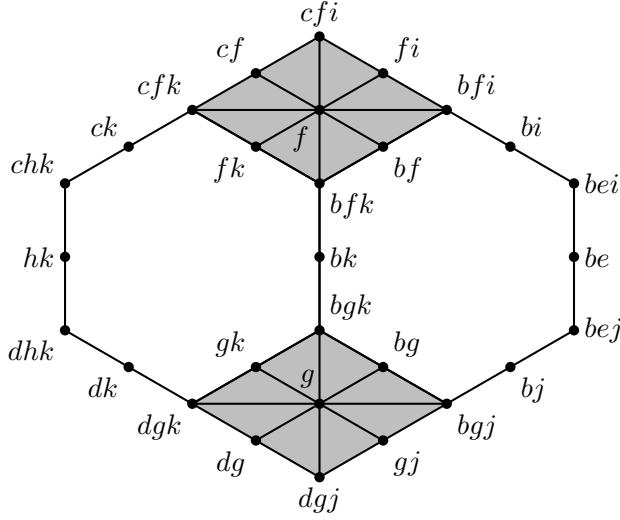
Here a vertex labeled i represents the necklace with flanked flag $(\Delta^1 \vee \Delta^1, \{0, i, 7\})$ and a vertex labeled ij represents the necklace with flanked flag $(\Delta^1 \vee \Delta^1 \vee \Delta^1, \{0, i, j, 7\})$. An edge between vertices i and ij represents the necklace with flanked flag $(\Delta^2 \vee \Delta^1, \{0, i, 7\} \subset \{0, i, j, 7\})$ or $(\Delta^1 \vee \Delta^2, \{0, i, 7\} \subset \{0, i, j, 7\})$.

Thus $\text{Map}_X(0, 7)$ is homotopy equivalent to the circle. We remark that this is an example of higher order structure that is not detected by the path category [9].

Example 3.13 Let X be the following simplicial set. For clarity, we omit the 1-simplices along the diagonals of each of the squares.



Then $\text{Map}_X(a, l)$ is given by the following simplicial set.



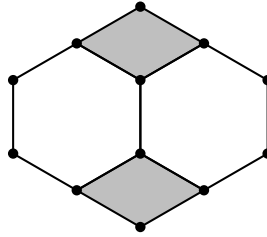
Here a vertex labeled x represents the necklace with flanked flag $(\Delta^1 \vee \Delta^1, \{a, x, l\})$, a vertex labeled xy represents the necklace with flanked flag $(\Delta^1 \vee \Delta^1 \vee \Delta^1, \{a, x, y, l\})$, and a vertex labeled xyz represents the necklace with flanked flag $(\Delta^1 \vee \Delta^1 \vee \Delta^1 \vee \Delta^1, \{a, x, y, z, l\})$. A triangle between vertices x , xy and xyz represents the necklace with flanked flag $(\Delta^2 \vee \Delta^2, \{a, x, l\} \subset \{a, x, y, l\} \subset \{a, x, y, z, l\})$.

Thus $\text{Map}_X(a, l)$ is homotopy equivalent to $S^1 \vee S^1$, the wedge of two circles.

4 Future directions

It would be very nice to have a Quillen model structure on simplicial categories appropriate to their use as models for concurrency. A natural candidate is J. Bergner's model structure on simplicial categories [1], which is closely related to A. Joyal's quasi-category model structure on simplicial sets [10]. However this model structure is too strong, since the weak equivalences induce equivalences of path categories. Thus a weaker notion of equivalence is needed. For some recent ideas in this direction, see [13].

The simplicial set models in this paper are in fact triangulations of cubical set models. When constructing models of the execution spaces from one state to another, we used (simplicial) necklaces on the simplicial models. Instead, one could use cubical necklaces on the cubical models, if one understood such things from a homotopy-theoretic point of view. These models would be more economical. For example, we should be able to redo Example 3.13 using cubical sets and cubical necklaces to obtain the following cubical set for $\text{Map}_X(a, l)$.



Acknowledgments

The author would like to thank David Spivak for many useful conversations pertaining to this article. He would also like to thank the NSF for funding some of this research through the Midwest Topology Network (NSF grant DMS-0844249). In addition he would like to thank Martin Raussen and Lisbeth Fajstrup for organizing the GETCO 2010 conference that motivated this work.

References

- [1] Julia E. Bergner. A model category structure on the category of simplicial categories. *Trans. Amer. Math. Soc.*, 359(5):2043–2058, 2007.
- [2] Peter Bubenik. Models and van Kampen theorems for directed homotopy theory. *Homology, Homotopy Appl.*, 11(1):185–202, 2009.
- [3] Peter Bubenik and Krzysztof Worytkiewicz. A model category for local po-spaces. *Homology, Homotopy Appl.*, 8(1):263–292, 2006.
- [4] E.W. Dijkstra. *Cooperating sequential processes*. Academic press, 1968.
- [5] Daniel Dugger and David Spivak. Rigidification of quasi-categories. arXiv:0910.0814 [math.CT], 2009.
- [6] L. Fajstrup, M. Raussen, E. Goubault, and E. Haucourt. Components of the fundamental category. *Appl. Categ. Structures*, 12(1):81–108, 2004. Homotopy theory.
- [7] Lisbeth Fajstrup, Martin Raussen, and Eric Goubault. Algebraic topology and concurrency. *Theoret. Comput. Sci.*, 357(1-3):241–278, 2006.
- [8] Marco Grandis. Directed homotopy theory. I. *Cah. Topol. Géom. Différ. Catég.*, 44(4):281–316, 2003.
- [9] J.F. Jardine. Path categories and resolutions. *Homology, Homotopy and Applications*, 12(2):231–244, 2010.
- [10] A. Joyal. Quasi-categories and Kan complexes. *J. Pure Appl. Algebra*, 175(1-3):207–222, 2002. Special volume celebrating the 70th birthday of Professor Max Kelly.
- [11] Thomas Kahl. Reducing cubical set models of concurrent systems. arXiv:1005.5443 [math.AT], 2010.
- [12] Sanjeevi Krishnan. A convenient category of locally preordered spaces. *Appl. Categ. Structures*, 17(5):445–466, 2009.
- [13] Sanjeevi Krishnan. Cubical approximation for directed topology. arXiv:1012.0509 [math.AT], 2010.
- [14] Martin Raussen. Simplicial models of trace spaces. *Algebr. Geom. Topol.*, 10(3):1683–1714, 2010.