Loops, Ditopology and Deadlocks

Lisbeth Fajstrup

December 15, 1999

1 Introduction

Geometric models in computer science are not a new phenomenon. Graph theory has been used for a long time in the modelling of both sequential and concurrent (non-interleaving) systems. The use of higher dimensional objects, progress graphs, to model concurrency dates back to Dijkstra [Dij68]. In [Pra91] and [vG91], R. van Glabbeek and V. Pratt introduce Higher Dimensional Automata as a model which allows both true concurrency and branching time in the same model. With this model, the heavy machinery from algebraic topology as well as geometric intuition enters concurrency theory. This is initiated by V. Pratt [Pra91], who suggests the study of monoidal homotopy theory, E. Goubault [Gou95] uses homological methods, especially the combinatorial versions of these, and J. Gunawardena [Gun94] applies methods from homotopy theory to scheduling in databases, and he gives a very nice description of these general ideas.

The point of view in this paper is as described in [FGR99], where the first notions of homotopy theory with a (time-) direction are formalized.

In [FGR98a], we used these methods to develop an algorithm for deadlock detection in concurrent systems. The algorithm allows the detection of deadlocks and unsafe states (from which no execution can finish) in a system of n processes sharing a set of objects which may be accessed by a limited number of processes at a time. We did not allow loops in the processes. For a discussion of the different algorithms for deadlock detection and comparison with our algorithm, see [FGR98b].

We know from [FGR98b], Ex. 2, p.9, which we give again here as Ex. 5.5, that even in the simplest case of just two processes with a loop in one of them, it may not suffice to unfold the loop once and study the corresponding non-looped system. This may introduce too many deadlocks and/or too many unsafe points.

The aim of this paper is twofold. We give the geometric model of a truly concurrent P-V system (in the sense of Dijkstra,[Dij68]) of n processes with finitely many (nested) loops in the framework of locally partially ordered semi-cubical complexes introduced in [FGR99] and we prove, that it suffices to study a finite number of deloopings of each loop to determine the unsafe area. Hence the deadlock algorithm can be applied to these finitely many deloopings and give the precise unsafe area in concurrent systems with loops. Many problems, for instance in verification, can be formulated as deadlock detection, and moreover, our algorithm can also be applied to determine unreachable states, since these are unsafe states when the direction of time is reversed.

It should also be noted, that the case of branchings is easier than the case of loops, and hence the deadlock algorithm can also be applied to systems with branches.

In most of this, we rely only on geometrical intuition. The development of the ditopology equivalent of coverings, homotopy lifting etc. is postponed till later, since we do not need it here, even if this is where the ideas used here came from.

Acknowledgements. It is a pleasure to thank two referees from Mathematical Structures in Computerscience, where this has been submitted, for many valuable suggestions to make this more readable. If this is still difficult to read, I am the one to blame.

2 Ditopology and deadlocks

We recall the basic definitions from [FGR99] and [FGR98b].

Definition 2.1 *1. A* partial order $\leq$ *on a set U is a reflexive, transitive and antisymmetric relation.*

2. A partial order $\leq$ on a topological space X is called closed *if $\leq$ is a closed subset of $X \times X$ in the product topology. In that case, $(X, \leq)$ is called a* pospace.

Definition 2.2 *Let X be a topological space.*

1. *A collection* $\mathcal{U}(X)$ *of pairs* $(U, \leq_U)$ *with partially ordered open subsets* U *covering* X *is a* local partial order *on* X *if*

   For every $x \in X$ *there is a non empty open neighbourhood* $W(x) \subset X$ *such that the restrictions of* $\leq_U$ *to* $W(x)$ *coincide for all* $U \in \mathcal{U}(X)$, *i.e.,*

   for all $U_1, U_2 \in \mathcal{U}(X)$ *and for all* $y, z \in W(x) \cap U_1 \cap U_2$:

   $$y \leq_{U_1} z \Leftrightarrow y \leq_{U_2} z.$$

2. *Two local partial orders* $\mathcal{U}(X)$ *and* $\mathcal{V}(X)$ *on* X *are equivalent if their* union $\mathcal{U}(X) \bigcup \mathcal{V}(X)$ *is a local partial order.*

3. *A topological space* X *together with an equivalence class of local partial orders is called a* locally partially ordered space. *If moreover there is a covering* $\mathcal{U}$ *in the equivalence class such that all* $(U, \leq_U) \in \mathcal{U}$ *are pospaces, then* X *is a local pospace.*

Remark 2.3 *Two local partial orders* $\mathcal{U}(X)$ *and* $\mathcal{V}(X)$ *are equivalent if and only if for every* $x \in X$ *there is a non empty open neighbourhood* $W(x) \subset X$ *such that the restrictions of* $\leq_U$ *and* $\leq_V$ *to* $W(x)$ *coincide for all* $U \in \mathcal{U}(X)$ *and* $V \in \mathcal{V}(X)$.

Example 2.4 *The circle* $S^1 = \{e^{i\theta} \in \mathbb{C}\}$ *has a local partial order: The open subsets* $U_1 = \{e^{i\theta} \in S^1 | 0 < \theta < 2\pi\}$ *and* $U_2 = \{e^{i\theta} \in S^1 | \pi < \theta < 3\pi\}$ *are (partially) ordered by the order on the* θ*'s. Notice that the relation on* S^1 *generated by these local partial orders by taking the transitive closure is* $x \leq y$ *for any pair* x, y. *Hence we do not take the transitive closure!*

Definition 2.5

1. *Let* $(X, \mathcal{U})$ *and* $(Y, \mathcal{V})$ *be* locally partially ordered spaces. *A continuous map* $f : X \to Y$ *is called a* dimap *(directed map) if there are refinements* $\mathcal{U}'$ *of* $\mathcal{U}$ *and* $\mathcal{V}'$ *of* $\mathcal{V}$ *such that*

   $$\forall U \in \mathcal{U}', V \in \mathcal{V}', x, y \in U \cap f^{-1}(V) : \quad x \leq_U y \Rightarrow f(x) \leq_V f(y).$$

2. *A* dipath *in* X *is a dimap* $f : I \to X$ *from a closed interval* $I \in \mathbb{R}$ *with the natural (global) order* $\leq$.

Definition 2.6 *For* X *a local po-space, and* $x \in X$ *we define*

1. *The future* $\uparrow x = \{y \in X | \text{there is a dipath from } x \text{ to } y\}$.

2. *The past* $\downarrow x = \{y \in X | \text{there is a dipath from } y \text{ to } x\}$

Definition 2.7 *Let X be a local po-space. Let $\mathcal{F} \subseteq X$. Then $x \in X$ is a deadlock with respect to $\mathcal{F}$ if $\uparrow x = \{x\}$ and $x \notin \mathcal{F}$. If $\mathcal{F} \cap \uparrow x = \emptyset$ then x is* unsafe *with respect to $\mathcal{F}$ and $\mathcal{F}$ is* unreachable *from x.*

Remark 2.8 *Our notion of safety is not the one used in program analysis, where being safe means, that you cannot get to a deadlock. Here it means that you can avoid deadlocks.*

Remark 2.9 *In the special case where X is the partially ordered space $I^n \setminus \bigcup_{j \in J} R^j$ and R^j are n-dimensional open hyperrectangles in $I^n \subset \mathbb{R}^n$ with the usual partial order Def. 2.10, we have an efficient algorithm calculating deadlocks and unsafe regions with respect to the maximal point in X,[FGR98a].*

Definition 2.10 *There is a partial order on (subsets of) $\mathbb{R}^n$*

$$(x_1, \dots, x_n) \leq (y_1, \dots, y_n) \text{ if } x_j \leq y_j, \text{ for all } j = 1, \dots, n$$

Definition 2.11 *A semi-cubical complex M is a family of sets $\{M_n | n \geq 0\}$ with face maps $\partial_i^k : M_n \to M_{n-1}$ $(1 \leq i \leq n,\ k = 0, 1)$ satisfying the semi-cubical relations:*

$$\partial_i^k \partial_j^l \quad = \quad \partial_{j-1}^l \partial_i^k \quad (i < j)$$

Let $\square_n$ be the standard cube in $\mathbb{R}^n$ $(n \geq 0)$,

$$\square_n = \{(t_1, \dots, t_n) | \forall i, 0 \leq t_i \leq 1\} \text{ and } \square_0 = \{0\}$$

and let $\delta_i^k : \square_{n-1} \to \square_n$, $1 \leq i \leq n$, $k = 1, 2$, be the continuous functions $(n \geq 1)$, defined by, $\delta_i^k(t_1, \dots, t_{n-1}) = (t_1, \dots, t_{i-1}, k, t_i, \dots, t_{n-1})$

Then, $\delta_i^k \delta_j^l = \delta_{j+1}^l \delta_i^k$, $(i \leq j)$

Definition 2.12 *For a semi-cubical complex M, we define the set $\mathbf{R}(M) = \coprod_n M_n \times \square_n$. The sets M_n have the discrete topology and $\square_n$ is topologized as a subset of $\mathbb{R}^n$ with the standard topology thus $\mathbf{R}(M)$ is a topological space with the disjoint sum topology.*
Let $\equiv$ be the equivalence relation induced by the identities:

$$\forall k, i, n, \forall x \in M_{n+1}, \forall t \in \square_n, n \geq 0, (\partial_i^k(x), t) \equiv (x, \delta_i^k(t))$$

Let $|M| = \mathbf{R}(M)/\equiv$ have the quotient topology. The topological space $|M|$ is called the geometric realization *of M.*

The underlying semi-cubical complex is a labelling of a subdivision of $|M|$ into cubes. An element $y_m \in M_m$ is the label of an m-cube in $|M|$. Let $p \in |M|$, then there is a minimal cube in the subdivision of $|M|$ containing p, namely the unique cube $x \times \Box_k$ which has p in the interior. We call x the *carrier* of p.

Definition 2.13 *The open star of a point $p \in |M|$ with respect to the subdivision M is*

$$St(p, M) = \{q \in |M| \big| \text{ carrier(p) is a face of carrier(q)}\}$$

Definition 2.14 *A semi-cubical complex is* non-selflinked *if for all its n-cubes y, $\partial_l^k(y) = \partial_{\tilde{l}}^{\tilde{k}}(y) \Rightarrow k = \tilde{k}$ and $l = \tilde{l}$*

In [FGR99] we defined a local partial order on the geometric realization of a non-selflinked cubical complex $|M|$. The covering is the open stars of the vertices, i.e., the elements of $M_0 \times \Box_0$, in $|M|$. The partial orders are induced by the partial order from $\mathbb{R}^n$ (Def. 2.10) in each cube together with the transitive closure of these *in each star* which induces relations from one cube to another via their common boundary.

Example 2.15 *The Möbius strip is the geometric realization of the cubical complex $M_2 = \{A, B, C\}$, $M_1 = \{a, b, c, d, e, f, g, h, i\}$, $M_0 = \{p, q, r, s, t, u\}$ with boundaries as on Fig. 1. The local partial order is as indicated on Fig. 1 and Fig. 2. Hence points in C are related to points in A via their common boundary b as one can see in the star of r.*

3 Notation and definitions

We give some notation and in particular we define an indexing of deloopings of a concurrent system. We focus on the interaction between processes, and this interaction is through shared resources, which are given by their semaphores. The execution of a process is then a sequence of locking and unlocking these resources. We have undeterminism in that we allow loops and nested loops.

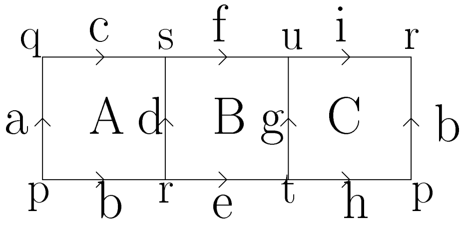

Figure 1: A “cubical” Möbius band.

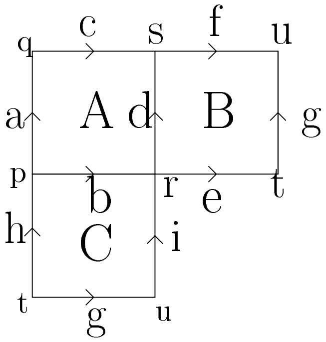

Figure 2: The closure of the star of the vertex r.

Definition 3.1 *A concurrent system is a system of n processes $T_1, \ldots, T_n$, where the i'th process has a finite number of loops:*

$$T_i = X_{i1}.L_{i1}^*.X_{i2}.L_{i2}^*.\ldots X_{im_i}.L_{im_i}^*.Y_i, \quad i = 1, \ldots, n$$

We allow a finite *number of nested loops. Hence for any finite multiindex $I = i_1 i_2 \cdots i_s$ we let*

$$L_I = X_{I1}.L_{I1}^*.X_{I2}.L_{I2}^*.\ldots X_{Im_I}.L_{Im_I}^*.Y_I$$

where $L_I \neq 1$ and $L_I \neq (L_{I1})^*$ to avoid redundance.

Each process T_i is a regular expression in the alphabet $\{P_A \big| \ A \in \mathcal{O}\} \bigcup \{V_A \big| \ A \in \mathcal{O}\}$, where $\mathcal{O}$ is a set of shared resources each of which is an $s(A)$ semafore, where $s : \mathcal{O} \to \mathbb{N}$, meaning that it will allow the access of $s(A)$ but not $s(A)+1$ processes at a time. The actions P_A, V_A are aquiring and relinquishing a lock on A in the sense of Dijkstra [Dij68]. A process can only access an object once before releasing it, and any object should be released eventually. Hence for any word $w_i = x_1 x_2 \ldots x_m$ in the language defined by T_i we require

- $x_i = P_A \Rightarrow \exists j > i : x_j = V_A$ *(objects are released eventually).*
- $x_i = V_A \Rightarrow \exists j < i : x_j = P_A$ *(an object cannot be released by T_i unless it is locked by T_i.)*
- $x_i = P_A$ *and* $x_k = P_A$ *and* $k > i \Rightarrow \exists j : i < j < k, x_j = V_A$ *(a process cannot access an object more than once before releasing it).*

- *$x_i = V_A$ and $x_k = V_A$ and $k > i \Rightarrow \exists j : i < j < k, x_j = P_A$ (an object cannot be released by T_i unless it is locked by T_i.)*

Remark 3.2 *Notice that in particular each loop L_I has to satisfy $\#P_A$ in $L_I = \#V_A$ in L_I. Otherwise, we have to restrict the deloopings of L_I allowed in a word and we will not do that. If $T = (P_A)^*.(V_A)^*$, by the restrictions (and the actual meaning of the locking and releasing), we would only allow words corresponding to $T = (P_A.V_A)^*$. Notice, however, that $T = P_A.(V_A.P_A)^*.V_A$ is allowed; hence the loops are not necessarily processes in the above sense, since they may violate the rule that an object cannot be released unless it is locked.*

Remark 3.3 *Let $\mathcal{I}$ be the set of indices of loops in T. By an abuse of notation, we will sometimes use the word index for a multiindex henceforth. Then*

- *$\mathcal{I}$ is finite.*
- *For all $I = i_1 i_2 \cdots i_s \in \mathcal{I}$, $1 \leq i_1 \leq n$ and $1 \leq i_k \leq m_{i_1 i_2 \ldots i_{k-1}}$ for $k = 2, \ldots, s$.*
- *We order the indices lexicographically and denote this $I \leq I'$.*
- *The index I is included in I' if $I' = I i_k i_{k+1} \ldots i_{s'}$. This is denoted $I \subset I'$.*

We consider the above system as representing all *finite deloopings*. A delooping of 3.1 is a system of n deterministic sequential processes $w_1, \ldots, w_n$ where w_i is a word in the language defined by T_i.

$$w_i = X_{i1}.L_{i1}^{k_{i1}}.X_{i2}.L_{i2}^{k_{i2}}.\ldots.X_{im_i}.L_{im_i}^{k_{im_i}}.Y_i, \quad i = 1, \ldots, n$$

where $L_I^k = L_I.L_I \ldots L_I$, i.e., L_I executed k times for $k \neq 0$ and $L_I^0 = 1$, the neutral element for concatenation ".". Without nested loops, we only need these integers $k_{i_1 i_2}$ for $i_1 = 1, \ldots, n$ and $i_2 = 1, \ldots, m_{i_1}$. When there are nested loops, it gets more complicated, since the nested loops in $L_{i_1 i_2}$ may be delooped differently in the different copies of $L_{i_1 i_2}$ in a given delooping.

To describe the number of deloopings of each (nested) loop, we introduce the following: A set of natural numbers $\mathbf{k} = \{k_{I,J}\}$, where I is the index of a loop and J is a new index, to describe how many times each loop is repeated (delooped) in the given delooping. Another index J (closely related to the first J) to point out a certain copy $L_{I,J}$ of the loop L_I in this delooping. We give an example first:

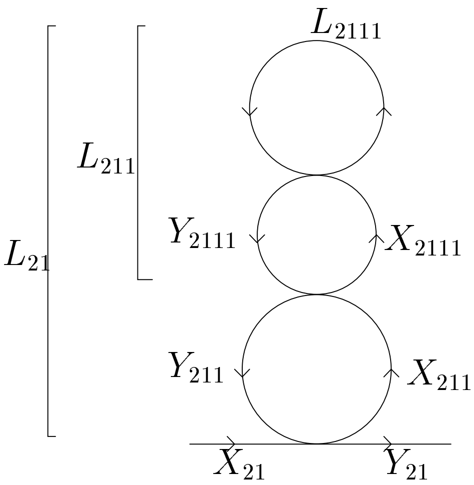

Figure 3: Nested loops and their indices.

Example 3.4 *Let*

$$T_2 = X_{21}.(X_{211}.L^*_{211}.X_{212}.L^*_{212}.Y_{21})^*.Y_2$$

The delooping

$$X_{21}.X_{211}.L_{211}.L_{211}.L_{211}.L_{211}.X_{212}.Y_{21}.X_{211}.L_{211}.X_{212}.L_{212}.L_{212}.Y_{21}.Y_2$$

has indices $k_{21} = 2$ meaning that L_{21} is repeated twice, $k_{211,1} = 4$ means that L_{211} is repeated 4 times in the first copy of L_{21}, $k_{212,1} = 0$ means that L_{212} is skipped in the first copy of L_{21}, $k_{211,2} = 1$ says that L_{211} is represented once in the second run of L_{21} and $k_{212,2} = 2$ indicates that L_{212} is repeated twice in the second run of L_{21}.

The two copies of L_{21} are numbered/indexed $L_{21,1}$ and $L_{21,2}$. Similarly the 4 copies $L_{211,J}$ in $L_{21,1}$ are denoted $L_{211,11}$, $L_{211,12}$, $L_{211,13}$ and $L_{211,14}$.

Finally we let $\mathcal{J}(I, \mathbf{k})$, where I is an index of a loop and $\mathbf{k}$ is a given delooping, be the set of indices J for which $L_{I,J}$ is in that delooping: $\mathcal{J}(21, \mathbf{k}) = \{1, 2\}$, $\mathcal{J}(211, \mathbf{k}) = \{11, 12, 13, 14, 21\}$ and $\mathcal{J}(212, \mathbf{k}) = \{21, 22\}$

Example 3.5 *With loops nested "deeper", the indices appearing will be quite complicated. Consider the process*

$$T_2 = X_{21}.(X_{211}.(X_{2111}.L^*_{2111}.Y_{2111})^*.Y_{21})^*Y_2$$

To describe a given delooping, we would have to tell for instance how many times L_{2111} is repeated in L_{211,j_1j_2}, i.e., in the j_2'th run of L_{211} in the j_1'th run of L_{21}; this number is k_{2111,j_1j_2}. See Fig. 3.

Not all choices of a set of $k_{I,J}$ will describe a delooping. If for instance L_{21} is skipped, i.e., $k_{21} = 0$, the nested loops are automatically skipped, hence $k_{21I,J}$ is undefined for any non empty I.

In general, to describe a delooping, if $k_{i_1i_2\ldots i_s,j_1j_2\ldots j_{s-2}} \neq 0$, we have to tell how many deloopings there are of every nested loop $L_{i_1i_2\ldots i_s}$ in each copy $L_{i_1i_2\ldots i_{s-1},j_1j_2\ldots j_{s-2}}$ of $L_{i_1i_2\ldots i_{s-1}}$, i.e., in the j_{s-2}'th copy of $L_{i_1i_2\ldots i_{s-1}}$ in the j_{s-3}'th copy of $L_{i_1i_2\ldots i_{s-2}}$ in the ... in the j_1'th copy of $L_{i_1i_2}$. That is, we have to give an integer $k_{i_1i_2\ldots i_s,j_1j_2\ldots j_{s-2}}$ for each $i_1i_2\ldots i_s \in \mathcal{I}$ and for $1 \leq j_{s-2} \leq k_{i_1i_2\ldots i_{s-1},j_1j_2\ldots j_{s-3}}$.

Definition 3.6 *Let w be a word in the language defined by T_{i_1} in the system 3.1. Then for $i_1i_2 \in \mathcal{I}$ we let $L^w_{i_1i_2,j_1}$ denote the j_1'th copy of $L_{i_1i_2}$ in w, and recursively we let $L^w_{i_1\ldots i_s,j_1\ldots j_{s-1}}$ denote the j_{s-1}'th copy of $L_{i_1\ldots i_s}$ in $L^w_{i_1\ldots i_{s-1},j_1\ldots j_{s-2}}$ whenever there is such a copy in w.*

Definition 3.7 *A finite set of non negative integers $\mathbf{k}_i = \{k_{I,J}\}$, where $I = i_1i_2\ldots i_s$ and $J = j_1j_2\ldots j_t$ are multiindices, is a delooping of the process T_i if*

1. *I is the index of a loop in T_i. In particular $i_1 = i$.*
2. *$t = s-2$ and for $t = 0$ $k_{I,J}$ is denoted k_I*
3. *$k_{i_1i_2}$ is in $\mathbf{k}_i$ for any $i_1i_2 \in \mathcal{I}$.*
4. *If $k_{I,J} = 0$, then $k_{Ii_{s+1},Jj_{s-1}}$ is undefined, i.e., it is not in $\mathbf{k}_i$.*
5. *If $k_{I,J} \neq 0$, then $k_{Ii_{s+1},Jj_{s-1}}$ is defined for all Ii_{s+1}, Jj_{s-1} such that $Ii_{s+1} \in \mathcal{I}(T_i)$ and $1 \leq j_{s-1} \leq k_{I,J}$.*

To determine a delooping of the system 3.1, let $\mathbf{k} = \mathbf{k}_1 \bigcup \ldots \bigcup \mathbf{k}_n$, where each $\mathbf{k}_i$ satisfies the above conditions.

We have to see, that a delooping is well-defined:

Lemma 3.8 *A set of integers satisfying Def. 3.7 defines an n-tuple of words in the language defined by $T_1, \ldots, T_n$. This map from sets of integers satisfying Def. 3.7 to n-tuples of words in the language defined by $T_1, \ldots, T_n$ is surjective.*

Proof. Given a set of integers $\mathbf{k}$ satisfying Def. 3.7, the corresponding delooping is constructed by delooping $L_{i_1i_2}$ $k_{i_1i_2}$ times, delooping $L_{i_1i_2i_3}$ $k_{i_1i_2i_3,j_1}$ times in the j_1'th copy, $L_{i_1i_2,j_1}$, of $L_{i_1i_2}$; and inductively L_{Ii_s} is delooped $k_{i_1i_2\ldots i_s,j_1j_2\cdots j_{s-2}}$ times in $L_{i_1\ldots i_{s-1},j_1j_2\cdots j_{s-2}}$.

Given a word w in the language defined by T_{i_1}, let $k_{i_1i_2}$ be the number of times $L_{i_1i_2}$ is repeated in w. Inductively, whenever $k_{i_1i_2\ldots i_{s-1},j_1j_2\cdots j_{s-3}} \neq 0$, let $k_{i_1i_2\ldots i_s,j_1j_2\cdots j_{s-2}}$ be the number of times L_{Ii_s} is repeated in $L_{i_1\ldots i_{s-1},j_1j_2\cdots j_{s-2}}$ for $Ii_s \in \mathcal{I}$ and for $0 < j_{s-2} \leq k_{i_1i_2\ldots i_{s-1},j_1j_2\cdots j_{s-3}}$.

We leave it to the reader to see, that this set of $k_{I,J}$ satisfies Def. 3.7, and that the above construction does give an n-tuple of words in the languages $T_1, \ldots, T_n$.

□

There is some ambiguity here if a loop consists only of loops, e.g., when $L_I = L^*_{I1}L^*_{I2}$ then the word $L_{I1}.L_{I2}$ can be considered as one run of L_I or as two runs, where L_{I2} is skipped in the first run, and L_{I1} is skipped in the second run. This is why the association of a word to $\mathbf{k}$ is not 1-1. But we only need it to be onto, and that is established by the above argument.

In view of this lemma, we shall refer to an n-tuple of words defined by $\mathbf{k}$ as *the delooping* $\mathbf{k}$ or just $\mathbf{k}$.

Remark 3.9 *Let $\mathcal{J}(I,\mathbf{k})$ be the set of J such that $L_{I,J}$ is in $\mathbf{k}$.*

- *In the delooping $\mathbf{k}$ there are $\Sigma_{J\in\mathcal{J}(i_1i_2\ldots i_{s-1},\mathbf{k})}k_{i_1i_2\ldots i_s,J}$ copies of the loop $L_{i_1i_2\ldots i_s}$.*
- *We order $\mathcal{J}(I,\mathbf{k})$ by the order of the copies of L_I in $\mathbf{k}$. This is numerical ordering of the $J = j_1j_2\ldots j_s$.*
- *$L^{\mathbf{k}}_{I,J}$ denotes the J'th copy of L_I in the delooping $\mathbf{k}_{i_1}$ of T_{i_1} with its nested loops delooped according to $\mathbf{k}$.*

We denote $I = 111\ldots1$ and $J = 111\ldots1$ by $I = \mathbf{1}$ and $J = \mathbf{1}$. Similarly the delooping of every loop once $k_{I,\mathbf{1}} = 1$ for any I is denoted $\mathbf{k} = \mathbf{1}$.

4 Geometric representation of concurrent systems with loops.

Here we specify what the geometry of a concurrent system of n processes with finitely many loops is, i.e., what the whole space is, what the forbidden area is etc.

Number the actions in the $\mathbf{k}_i$ delooped i'th process

$$T_i^{\mathbf{k}_i} = X_{i1}.L_{i1}^{k_{i1}}.X_{i2}.L_{i2}^{k_{i2}}.\ldots.X_{im_i}.L_{im_i}^{k_{im_i}}.Y_i$$

$$L_{I,J} = X_{I1}.L_{I1}^{k_{I1,J}}.X_{I2}.L_{I2}^{k_{I2,J}}.\ldots.X_{Im_I}.L_{Im_I}^{k_{Im_I,J}}.Y_I, \quad I \in \mathcal{I} \text{ and } i_1 = i$$

starting from 1. Action number m is denoted $T_i^{\mathbf{k}_i}(m)$. Notice that $T_i^{\mathbf{k}_i}$ is a process in the sense of 3.1 and it is deterministic.

Definition 4.1 *An action $P_A = T_i^{\mathbf{k}_i}(a_A)$ is represented by a halfopen directed interval $]a_A - 1, a_A]$. Similarly $V_B = T_i^{\mathbf{k}_i}(b_B)$ is represented by $]b_B - 1, b_B]$. Let $N_i^{\mathbf{k}_i}$ be the number of actions in $T_i^{\mathbf{k}_i}$. The geometric representation of $T_i^{\mathbf{k}_i}$ is the directed interval $I_i^{\mathbf{k}_i} = [0, N_i^{\mathbf{k}_i}]$, i.e., the union of the $N_i^{\mathbf{k}_i}$ halfopen intervals $]T_i^{\mathbf{k}_i}(j) - 1, T_i^{\mathbf{k}_i}(j)]$, $j = 1, \ldots, N_i^{\mathbf{k}_i}$ with an initial point 0.*
If $Y_{i_1 i_2} = 1$, we let $I_i^{\mathbf{k}_i} = [0, N_i^{\mathbf{k}_i} + 1]$ to ensure that there is a unique final point (Def. 4.9). The interval $]N_i^{\mathbf{k}_i}, N_i^{\mathbf{k}_i} + 1]$ can just be thought of as representing the trivial action.
The length $|X|$ of a (substring of a) delooping of a process is the number of actions in X.
We let X denote both the string of actions and the geometric representation of X. The endpoint X^+ and the beginning X^- are the maximal point and the infimum of the points in the geometric representation of X.
The interior of the geometric representation of X is denoted $\overset{o}{X}$.

The geometric representation of T_i is the quotient of $I_i^{\mathbf{1}}$ under identification of the endpoints L_I^+ with the beginning L_I^- for $I \in \mathcal{I}$. The topology is the quotient topology.
The concurrent system 3.1 is geometrically represented by $B = T_1 \times T_2 \times \ldots \times T_n$. The delooping $\mathbf{k}$ is represented by $B_{\mathbf{k}} = I_1^{\mathbf{k}_1} \times I_2^{\mathbf{k}_2} \times \ldots \times I_n^{\mathbf{k}_n}$.
A state *in the system 3.1 is a point in B.*

Remark 4.2 All *points are states in or model. The choice of the length of the intervals representing each (not necessarily atomic) action P_A or V_A is arbitrary, since we only want to specify the succesion of actions. Hence the model is well suited for subdivision of actions, since we do not single out any points as being special a priori.*
If $X_I \neq \emptyset$ then $L_I^- = X_I^+$. When there are nested loops or when some X_I are trivial, the identification of the endpoint of each loop with its beginning may identify more than two points. Hence the fiber $\pi_{\mathbf{1}}^{-1}(x)$ of $x \in B$, where $\pi_{\mathbf{1}} : B_{\mathbf{1}} \to B$ is the identification defining B, may have more than 2 points.

Lemma 4.3 *The spaces T_i and B are local pospaces. Moreover they are non-selflinked cubical complexes.*

Proof.

1. T_i is a local po-space. We give a covering $\mathcal{U}_i$ by pospaces: The interiors $\overset{o}{X}_{Ij}$ and $\overset{o}{Y}_I$ are open subsets with a canonical (partial) order by the local time. For $m_I = 0$, $\overset{o}{L}_I$ is also an open ordered interval. Let $p = L_I^+$ and let $s = \#\{\pi_{\mathbf{1}}^{-1}(p)\}$. Then there are $s-1$ (time-)directed circles glued together at p. An open partially ordered neighborhood $\mathcal{U}_p$ of p is constructed by removing a point from each circle. Then $\mathcal{U}_p$ is a union of 2(s-1) intervals (from the circles), an interval $\pi_{\mathbf{1}}(]\min\{\pi_{\mathbf{1}}^{-1}(p)\}-1, \min\{\pi_{\mathbf{1}}^{-1}(p)\}])$ representing coming into the loops at p, and an interval $\pi_{\mathbf{1}}([\max\{\pi_{\mathbf{1}}^{-1}(p)\}, \max\{\pi_{\mathbf{1}}^{-1}(p)\}+1[)$ representing leaving the loops at p.

   It is a local pospace, since the partial order on a $U_i \in \mathcal{U}_i$ is either the total ordering of an interval or it is $2(s-1)+2$ intervals glued at a point with partial order induced by the transitive closure of the total ordering of each interval as at the choicepoint in Fig. 4.

2. Since B is the product $T_1 \times T_2 \times \ldots \times T_n$, there is a local partial order on B given by $\mathcal{U} = \{U_1 \times U_2 \times \ldots \times U_n | U_i \in \mathcal{U}_i, \ \ i = 1, \ldots, n\}$ with partial order on $U_1 \times U_2 \times \ldots \times U_n$ defined as follows: $(u_1, \ldots, u_n) \leq (u_1', \ldots, u_n')$ if $u_i \leq u_i'$ for all $i = 1, \ldots, n$. This is a covering by pospaces.

3. A loop contains at least two actions by 3.2. So the intervals defined by the actions induce a semi-cubical complex structure on T_i and hence on B, which is non-selflinked. The local partial ordering defined via the semi-cubical structure is in fact the same as the one defined above, as the reader may easily check.

□

Definition 4.4 *The projection $\pi_{\mathbf{k}} : B_{\mathbf{k}} \to B$ is given by the identification of all copies of L_I for all $I \in \mathcal{I}$ and identification of the endpoint L_I^+ of L_I with the beginning L_I^-. This defines an equivalence of points. For $x \in B_{\mathbf{k}}$ and $y \in B_{\mathbf{k}'}$, $x \equiv y$ if $\pi_{\mathbf{k}}(x) = \pi_{\mathbf{k}'}(y)$. A* state *in the concurrent system 3.1 is a point in B, i.e., an equivalence class of points in the set of deloopings. We refer to a state $x \in B$ by the cartesian coordinates of one of its representatives $\pi_{\mathbf{1}}^{-1}(x)$ in $B_{\mathbf{1}} \subset \mathbb{R}^n$.*

Lemma 4.5 *The projections $\pi_{\mathbf{k}}$ are continuous maps which preserve the local partial order, and there is a* lifting property*: Let $\gamma : [0,T] \to B$ be a finite dipath in B then there is a delooping $\mathbf{k} = (\mathbf{k}_1, \dots \mathbf{k}_n)$ and a dipath $\tilde{\gamma} : [0,T] \to B_{\mathbf{k}}$ such that $\pi_{\mathbf{k}} \circ \tilde{\gamma}(t) = \gamma(t)$ for all $t \in [0,T]$*

Proof. The local partial order is preserved, since this is basically how it is defined. The projection is continuous, since B has the quotient topology.

For the lifting notice that $\pi_{(\mathbf{k}_1,\dots,\mathbf{k}_n)}(x_1,\dots,x_n) = (\pi_{\mathbf{k}_1}(x_1),\dots,\pi_{\mathbf{k}_n}(x_n))$ as the identifications defining $\pi_{(\mathbf{k}_1,\dots\mathbf{k}_n)}$ are all coordinatewise. Hence it suffices to lift the coordinate functions of $\gamma = (\gamma_1,\dots,\gamma_n)$. Now $\gamma_i([0,T])$ defines a substring $x_1x_2\dots x_l$ of a word in the language defined by T_i, namely the string of actions traversed by γ_i in the order of traversing. An action is traversed by γ_i at time $t \in]t_1,t_2]$ if a connected component of $\pi_{\mathbf{1}}^{-1}(\gamma_i(]t_1,t_2]))$ is the interval representing that action in $T^{\mathbf{1}}$. (The other components, if any, are points in $\pi_{\mathbf{1}}^{-1}(t_2)$.) To define this at the endpoints, if a connected component of $\pi_{\mathbf{1}}^{-1}(\gamma_i(]0,t_1]))$ is an open subset $]x_A - \varepsilon, x_A]$ of the interval defining the action x, then $x_1 = x$. We define x_l similarly.

Extend the substring $x_1x_2\dots x_l$ arbitrarily to a word w in the language defined by T_i. Let $\mathbf{k}_i$ be a delooping defining w as in lemma 3.8. Then $\gamma_i(t)$ lifts to this delooping.

□

There are also maps from $B_{\mathbf{k}}$ to $B_{\mathbf{1}}$, but these are not continuous at the endpoints of loops and they are not necessarily onto.

Some states in B are not allowed because more than $s(A)$ processes hold a lock on the resource A. These states constitute the forbidden region. (See Rem. 4.8.)

Definition 4.6 *The* forbidden region *of a concurrent system without loops is the union of the* forbidden rectangles*. We get a forbidden n-rectangle for each partition of $\{1,\dots,n\}$ into a disjoint union $U \bigsqcup V$ such that*

1. *$card(U) = s(A) + 1$ for some $A \in \mathcal{O}$*
2. *There is a set of integers $x_i, y_i \in [0, N_i^{\mathbf{k}_i}]$, $i \in U$ such that $T_i(x_i) = P_A, T_i(y_i) = V_A$. And $x_i < z < y_i \Rightarrow T_i(z) \neq P_A$ and equivalently $x_i < z < y_i \Rightarrow T_i(z) \neq V_A$.*

The forbidden rectangle is then $J_1 \times J_2 \times \ldots \times J_n$ *where* $J_i =]x_i - 1, y_i[$ *if* $i \in U$ *and* $I_i^{\mathbf{k}_i}$ *for* $i \in V$. *The forbidden region of the delooping* $B_{\mathbf{k}}$ *is denoted* $F_{\mathbf{k}}$.

The forbidden region F *of* B *is the image under projection of* $F_{\mathbf{1}}$.

Here an n-rectangle is an n-dimensional "box". We will denote this a rectangle for short.

Lemma 4.7 *The forbidden region of* B *contains the image of the forbidden region* $F_{\mathbf{k}}$ *under the projection* $\pi_{\mathbf{k}}$ *for any* $\mathbf{k}$. *If* $\pi_{\mathbf{k}}$ *is surjective, then* $F_{\mathbf{k}}$ *maps onto* F.

Proof. The projection is defined coordinatewise, $\pi_{\mathbf{k}} = (\pi_{\mathbf{k}_1}, \pi_{\mathbf{k}_2}, \ldots, \pi_{\mathbf{k}_n})$. Denote this $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ without the $\mathbf{k}$ if this is obvious from the context.

Let $R = J_1 \times J_2 \times \ldots \times J_n$ be a forbidden rectangle in $B_{\mathbf{k}}$. We have to prove that there are forbidden rectangles $R_1, \ldots, R_m$ in $B_{\mathbf{1}}$ such that $\pi(R) \subseteq \pi(R_1 \cup R_2 \cdots \cup R_m)$ and that if $\pi_{\mathbf{k}}$ is surjective, the projection of a forbidden rectangle in $B_{\mathbf{1}}$ is contained in the projection of a union of forbidden rectangles in $B_{\mathbf{k}}$. Again we do this coordinate-wise, i.e., for the intervals J_i.

If $J_i = [0, I_i^{\mathbf{k}_i}]$ then $\pi_i(J_i) = T_i = \pi_{\mathbf{1}}(I_i)$.

If $J_i =]x_i - 1, y_i[$ where $T_i^{\mathbf{k}_i}(x_i) = P_A$ and $T_i^{\mathbf{k}_i}(y_i) = V_A$, we have five cases. The first four are not difficult; the five cases are described in Fig. 4.

1. $x_i \in L^{\mathbf{k}}_{I,J}$ and $y_i \in L^{\mathbf{k}}_{I,J}$. Then there is a P_A equivalent to x_i and a V_A equivalent to y_i in $L^{\mathbf{k}}_{I,\mathbf{1}}$, hence there is a $J_{\mathbf{1},i} \in I_i$ with $\pi_{\mathbf{1},i}(J_{\mathbf{1},i}) = \pi_{(\mathbf{k}_1,\ldots \mathbf{k}_n),i}(J_i)$

2. $x_i \in X_{ij}$ and $y_i \in X_{i(j+l)}$. Then $L^{\mathbf{k}}_{I,J} \subset J_i$ for $i_1 \subset I$ and $j \leq i_2 < j + l$. Hence there are no P_A or V_A in the loops $L^{\mathbf{k}}_{I,J}$ and we get a corresponding interval $J_{\mathbf{1},i} = [(\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k}_i}(x_i), (\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k}_i}(y_i)[$ in I_i such that $\pi_{\mathbf{1}}(J_{\mathbf{1}}) = \pi_{\mathbf{k}_i}(J_i)$

3. If $x_i \leq (L^{\mathbf{k}}_{I,J})^-$ and $y_i \in L^{\mathbf{k}}_{I,J}$ then $J = \mathbf{1}$: Since there is a V_A in L_I, there is also a P_A, which is then in all $L^{\mathbf{k}}_{I,J}$ and this would contradict 4.6.2 if $J \neq \mathbf{1}$. The rest is as above.

4. $x_i \in L^{\mathbf{k}}_{I,J}$ and $y_i > (L^{\mathbf{k}}_{I,J'})^+$ for any $J' \in \mathcal{J}(I, \mathbf{k})$. By a symmetrical argument this implies that J is maximal and the rest follows.

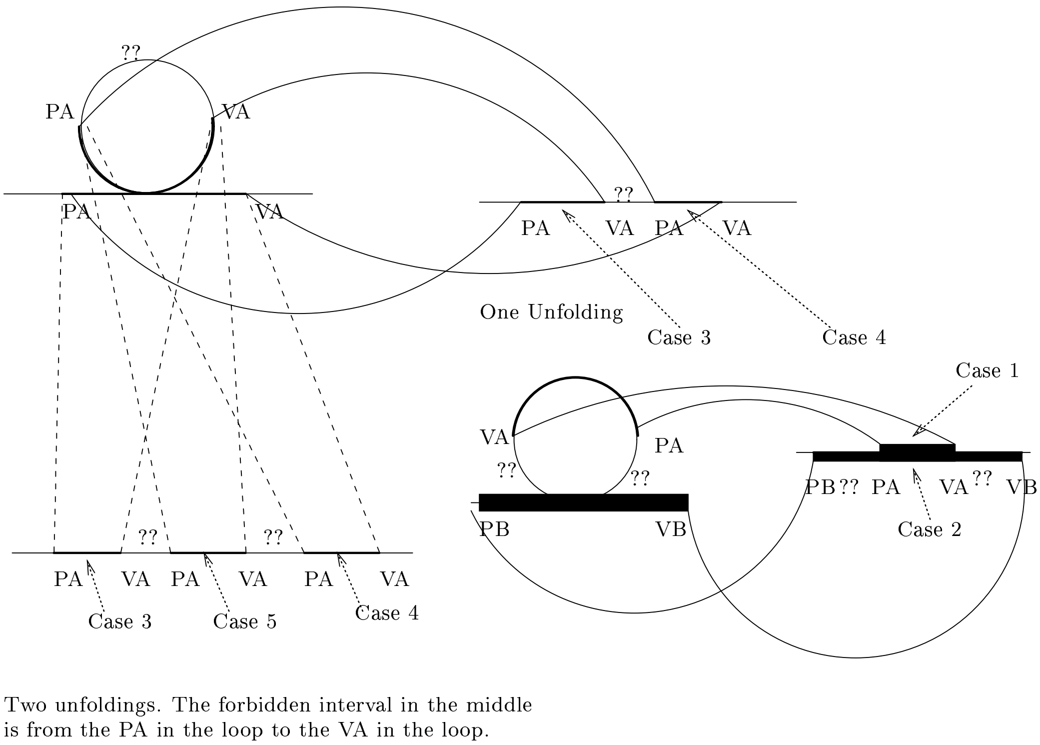

Figure 4: Forbidden intervals of different types.

5. $x_i \in L^{\mathbf{k}}_{I,J}$ and $y_i \in L^{\mathbf{k}}_{I,J'}$ where $J = j_1 j_2 \ldots j_s$ and $J' = j_1 j_2 \ldots (j_s + 1)$. By 4.6 there is no P_A or V_A in $]x_i, y_i[$, so $(\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k}_i}(x_i)) > (\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k}_i}(y_i))$ in L_I, i.e., $L_I = Z_1.V_A.Z_2.P_A.Z_3$, where neither Z_1 nor Z_3 contain any P_A or V_A. Hence there is *not* a corresponding interval J_1 in L_I. However, the syntax forces the existence of an $x \leq L_I^-$ such that $T^{\mathbf{1}}(x) = P_A$ and $]x-1, (\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k}_i}(y_i)[$ satisfies 4.6.2. Similarly there is a $y > L_I^+$ such that $](\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k},i}(x_i)) - 1, y[$ satisfies 4.6.2. Then

$$\pi_{\mathbf{1}}(]x-1, (\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k}_i}(y_i)[\cup](\pi_{\mathbf{1}})^{-1}(\pi_{\mathbf{k}_i}(x_i)) - 1, y[) \supset \pi_{\mathbf{k}_i}(J_i)$$

We also have to see that if $\pi_{\mathbf{k}}$ is surjective, then for any $J \in I_i$ there is a J' such that $\pi_{\mathbf{1}}(J) = \pi_{\mathbf{k}_i}(J')$, but this is similar and easier because case 5 is not present in that case.

□

Remark 4.8 *1. Topologically, T_i is a directed interval with a directed circle glued on for each loop.*

2. We have changed the geometry for the non-looped situation from our earlier approach [FGR98a]. In our original approch, the intervals defining the forbidden rectangles were $]a_A, b_A[$, and now they are $]a_A - 1, b_A[$. As long as there are no loops, this gives the same geometry up to inessential deformations, but we had *to change the geometry when we want loops. In Fig.4, the process $P_A.(P_B.V_B)^*.V_A$ definitely has a lock on A while leaving the loop, and hence there has to be a lock on A at the point $\pi(a_A) = \pi(b_B)$ where it enters the loop. On the other hand, there is not a lock on B, when it leaves the loop. This forces the forbidden intervals and hence the forbidden rectangles to include these points, i.e., the intervals defining the forbidden rectangles should contain $[a_A, b_B - 1]$, they should not contain $a_A - 1$ or b_B and we have chosen $]a_A - 1, b_B[$, since we want the allowed states to constitute a subcomplex of B.*

Without choices these changes are inessential and they are not necessary. The deadlock algorithm is of course discrete, and hence it cannot see the difference.

Definition 4.9 *The* allowed states *in 3.1 are the points in $X = B \setminus F$. A point p in B and X is referred to by the cartesian coordinates of one of the*

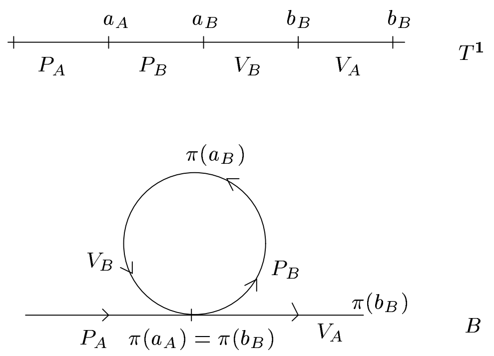

Figure 5: A process with a loop

points in $\pi_{\mathbf{1}}^{-1}(p) \in B_{\mathbf{1}}$. *The* allowed states *in a delooping* $\mathbf{k}$ *are* $X_{\mathbf{k}} = B_{\mathbf{k}} \setminus F_{\mathbf{k}}$. *The* final point *in* $X_{\mathbf{k}}$ *is the global maximum,* $(|I_1^{\mathbf{k}_1}|, \ldots, |I_n^{\mathbf{k}_n}|)$. *The final point in* X *is the projection of the final point in* $X_{\mathbf{k}}$. *By an abuse of notation, the final point is denoted* $\mathcal{F}$ *in* B, *in* X *and in all deloopings. An* execution *of 3.1 is a dipath in* X *from 0 to* $\mathcal{F}$. *And as in Def. 2.7, a* deadlock *is a point in* X *such that* $\uparrow x = x \neq \mathcal{F}$, *a point* $p \in X$ *is* safe *if there is a dipath in* X *from* p *to* $\mathcal{F}$, *i.e., if* $\mathcal{F} \in \uparrow x$. *A point in* X *is* unsafe, *if it is not safe.*

Lemma 4.10 *A point* $p \in X$ *is safe if and only if there is a delooping* $\mathbf{k}$ *and a point* p' *such that* $\pi_{\mathbf{k}}(p') = p$ *and there is a dipath in* $X_{\mathbf{k}}$ *from* p' *to* $\mathcal{F}$

Proof. A dipath in $X_{(\mathbf{k}_1, \ldots, \mathbf{k}_n)}$ from p' to $\mathcal{F}$ projects to a dipath in B from p to $\mathcal{F}$. By 4.7 this path will avoid the forbidden region. Conversely a dipath in X from p to $\mathcal{F}$ lifts to a dipath in some delooping $B_{\mathbf{k}}$ by the lifting property 4.5. By 4.7 this will be a path in $X_{\mathbf{k}}$, and a dipath to $\mathcal{F}$ will lift to a dipath to $\mathcal{F}$, since $\pi_{\mathbf{k}}$ is one to one at $\mathcal{F}$.

□

Remark 4.11 *1. Notice that we can apply the deadlock algorithm, 2.9, to the deloopings.*

2. *For a point* p *to be safe, we require, that there is a dipath from* p *to* $\mathcal{F}$. *Hence a state is unsafe, even if one could proceed indefinitely from this state by staying in a loop.*

3. *A point is safe, if there is a dipath from it to a safe point.*

4. *The union of all the finite deloopings is the universal dicover B. This is constructed as the usual universal cover, i.e., it is the space of all dipaths in B.*

Lemma 4.12 *A point $p \in X$ is a deadlock if and only if $x \in \uparrow \pi_{\mathbf{k}}^{-1}(p) \Rightarrow \uparrow x = x$ for all $\mathbf{k}$ such that $k_{I,J} \leq 1$ or undefined for all I, J.*

Proof. We have to see, that if $\uparrow p \neq p$ in X then there is a $\mathbf{k}$ with all loops delooped at most once such that there is an $x \in \pi_{\mathbf{k}}^{-1}(p)$ with $\uparrow x \neq x$. Let $\gamma : [0,T] \to X$ be a dipath with $\gamma(0) = p$ and $\gamma(t) \neq p$ for $t > 0$. Then for ε sufficiently small, $\gamma([0,\varepsilon[)$ may intersect some loops, but it will not contain L_I for any I. Hence $\gamma([0,\varepsilon[)$ lifts to a delooping $\mathbf{k}$ where $k_{I,\mathbf{1}} = 1$ if $\gamma([0,\varepsilon[) \cap L_I \neq \emptyset$, $k_{Ii_s,Jj_{s-2}} = 0$ if $k_{I,J} = 1$ and undefined if $k_{I,J} = 0$. $\square$

5 Examples

Here we give some examples of what to expect in the deloopings. We saw in Lemma 4.12 that it suffices to study deloopings where each loop is delooped at most once to find all deadlocks. Finding the unsafe area is not that simple, and this will become clear in the examples. All shared objects are mutual exclusion in this section. The pictures show various unfoldings with the forbidden area and the unsafe regions below.

Remark 5.1 *In the pictures, we have the following conventions: The forbidden area is black, the unsafe area is grey and the safe area is white. In order to be able to see, that there is a (one dimensional) gap between for instance the forbidden rectangles corresponding to A in Fig. 7, we have shrunk the forbidden rectangles a little. Or in other words, we have fattened up the 1-skeleton. Furthermore, we write PA and VA for P_A and V_A and we have not written a_A, b_A etc. in order not to crowd the pictures.*

Example 5.2 *Here, the state $(1,1)$ is a deadlock without the loop, Fig. 6, but becomes safe after inserting the loop once, Fig. 7.*

$$T1 = P_A.(V_A.P_A)^* P_B.V_B.V_A$$

$$T2 = P_B.P_A.V_A.V_B$$

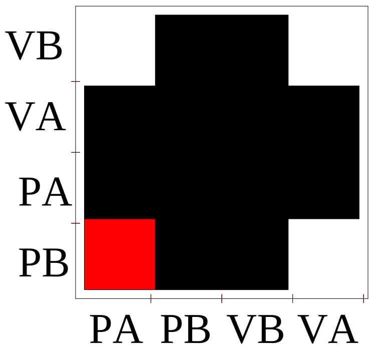

Figure 6:

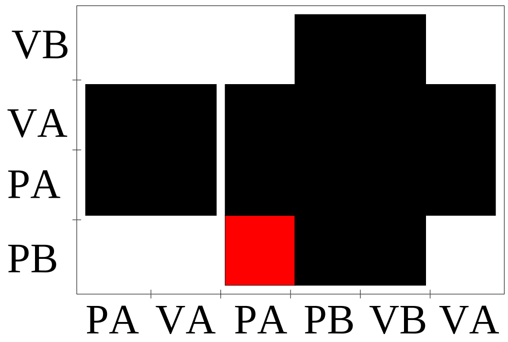

Figure 7:

Example 5.3 *In this example, the state* $(1/2, 2)$ *is safe, since there is a path to the final point in* $X_{\mathbf{0}}$*, the space without delooping of the loop Fig. 8. But* $(1/2, 2)$ *looks unsafe in* $X_{\mathbf{1}}$*, Fig. 9, where the loop is delooped once:*

$$T_1 = P_A.(P_C.V_A.P_B.V_B.P_D.V_D.V_C.P_A)^*.V_A$$

$$T_2 = P_B.P_D.P_A.V_B.V_A.P_C.V_C.V_D$$

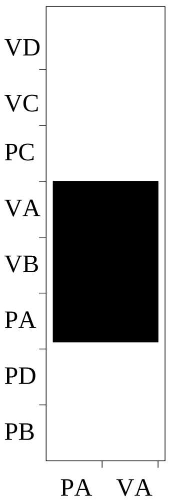

Figure 8:

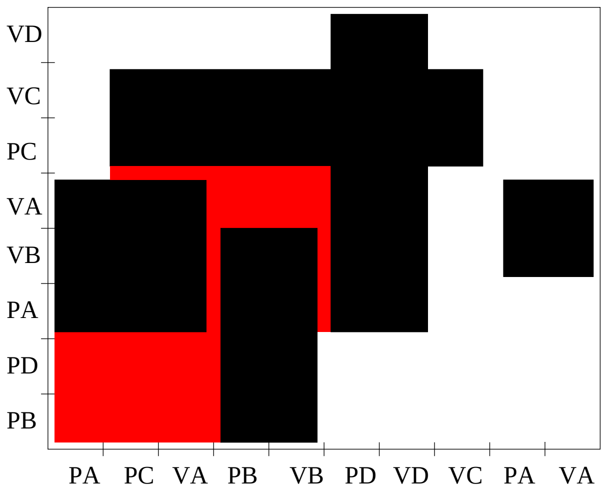

Figure 9:

Example 5.4

$$T1 = P_A.(P_B.V_B.P_C.V_C)^*.V_A$$

$$T2 = P_C.P_B.V_C.P_A.V_A.V_B$$

The state $(1/2, 1/2)$ *is safe, since there is a path to the final state with one or no unfoldings of the loop, Fig. 10, but with two unfoldings, it appears to be unsafe, Fig. 11.*

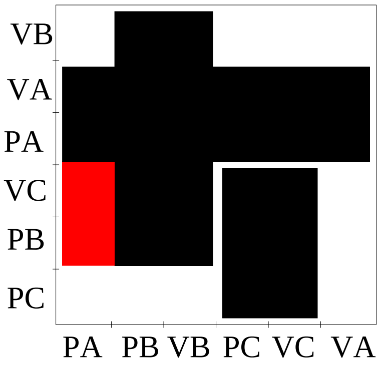

Figure 10:

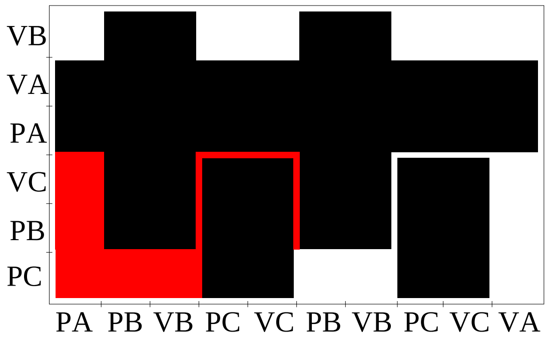

Figure 11:

Example 5.5

$$T1 = P_D.(P_A.V_D.P_B.V_A.P_D.V_B.P_E.V_E)^*.P_F.V_D.V_F$$

$$T2 = P_F.P_E.P_A.V_E.P_B.V_A.P_D.V_B.P_A.V_D.P_B.V_A.P_D.V_B.V_D.V_F$$

Unfolding the loop once gives Fig. 12 and unfolding twice gives Fig. 13. Notice how the unsafe region evolves. The two apparent deadlocks are not actually deadlocks. They just force reentrance of the loop. There are no deadlocks in this system and hence no unsafe region. Hence, one may have to unfold a loop several times to determine the unsafe area.

Example 5.6 *With loops in more than one process, the geometry is more complex. Even with just two looped processes, we get tori in B. If the allowed states in such a torus constitute a* (p, q)*-torusknot:* $\{(e^{\frac{2\pi i}{p}t}, e^{\frac{2\pi i}{q}t}) \in S^1 \times S^1 \big| \ t \in [0, pq]\}$*, then even it is only safe to leave the torus at one point (e.g.,*$(e^{2\pi i}, e^{\frac{2\pi i}{q}p})$*), an unfolding of the first loop* q *times and the second loop*

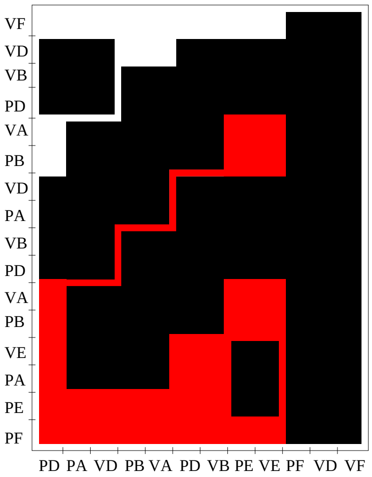

Figure 12: Unfold once.

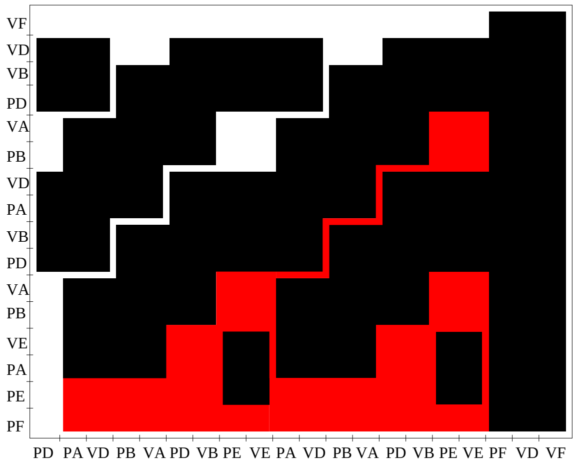

Figure 13: Unfold twice.

p times will reveal, that all points in the torusknot are safe. But if p and q are pairwise prime, this cannot be seen in any lower unfolding. Here is an example of a $(3,2)$ torusknot:
$T1 = P_D.P_A.(P_B.V_A.V_D.P_C.V_B.P_A.P_D.V_C.P_B.V_A.P_C.V_B.P_A.V_C.P_B.V_A.P_C.V_B.P_A.V_C)^*$
$.V_A.P_E.V_D.V_E$
$T2 = P_E.P_A.(P_B.V_A.P_C.V_B.P_A.V_C.P_B.V_A.P_C.V_B.P_A.V_C)^*.V_A.P_D.V_E.V_D$
See Fig.14. The area which projects to a torus in B is within the "dotted box". We leave it to the reader to see, that there are no unsafe points in the torus, i.e., in the product of the two loops.

In higher dimensions, there may be torusknots $\{(e^{\frac{2\pi i}{p_1}t}, e^{\frac{2\pi i}{p_2}t}, \ldots, e^{\frac{2\pi i}{p_n}t}) \in S^1 \times \ldots \times S^1 \big|\ t \in [0, p_1 \cdot \ldots \cdot p_n]\}$. In general, it would then take $\prod_{i \neq i_0} p_i$ deloopings of the loop in T_{i_0} for $i_0 = 1, \ldots, n$ to find the true safe/unsafe area. (If the p_i are not coprime, the number of deloopings of the loop in T_{i_0} needed is the least common multiple of all p_i divided by p_{i_0}.)

Hence to make sure not to mark a safe state unsafe, by Ex. 5.4 we cannot just take a huge delooping. This would give a superset of the unsafe states. On the other hand there is no way of knowing in general how many times each loop would have to be unfolded, as one can see from the torusknots. A priori it is not even clear, that the safe states will be found in a finite unfolding, or that there is any way of knowing when to stop unfolding to look for safe states. But this is what we will prove.

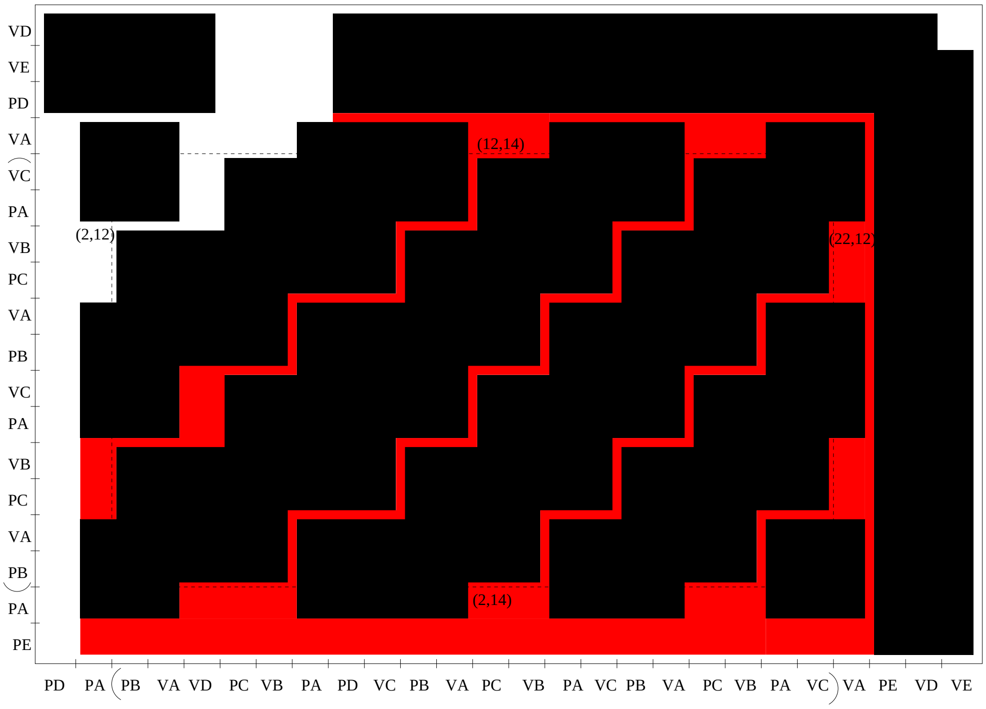

Figure 14: A $(2, 3)$ PV torusknot.

6 The unsafe region can be found in a finite number of unfoldings.

We prove, that it will always suffice to consider a finite number of unfoldings of the loops to give a precise determination of the unsafe region. We will not distinguish between deadlocks and unsafe regions. The deadlocks will just be considered as points in the unsafe region in the following. There are some basic facts about unfolding loops, which we will need:

Remark 6.1 1. *Further unfolding of a loop will* never add to the unsafe area*: If a state* $\mathbf{x}$ *is found safe in some unfolding, there is a dipath to the final point in that unfolding, hence by projection in the local pospace* X*, and this is of course not affected by the fact that there may be no dipath to the final point in some other unfolding, i.e., that the dipath we found in* X *does not lift to that other unfolding.*

2. *Unfolding a loop may reduce the unsafe area, e.g., in example 5.2, the state* $(1,1)$ *is declared unsafe in* X_0*, but safe in* X_1.

The strategy of the proof is to show, that if in some delooping, $\mathbf{k}$, there is a safe point in the equivalence class $\pi_{\mathbf{k}}^{-1}(x)$ and this is not the case in any lower delooping, see Def. 6.5, then there is a reason for this: In Ex. 5.5, the state with cartesian coordinates $(8,6)$ will be found safe in the third unfolding *because* there is a dipath in that unfolding to points which were found safe in the second unfolding (but not in the first unfolding). If no new points were found safe in the i'th unfolding, then there will be no new safe points in the (i+1)'th unfolding. With more loops, nested loops and especially loops in more that one proces, this is more intricate, as one can see in Ex. 5.6, but the idea is the same: If in Ex. 5.6, we unfold the loop in T_1 twice and the loop in T_2 is still only unfolded once, i.e., $k_{11} = 2$ and $k_{21} = 1$ the state $(22,12)$ and the "channel" below it becomes safe, because they get connected to (or rather: one can see, that it is equivalent to) $(2,12)$, which is safe in the delooping $\mathbf{1}$. Notice, however, that delooping the loop in T_2 giving $k_{11} = 1$ and $k_{21} = 2$ does not give more safe states than $\mathbf{k} = \mathbf{1}$. *But* in the delooping $k_{11} = k_{21} = 2$, the state $(12,14)$ and the "channel" below it becomes safe, because it gets connected to $(12,2)$ which became safe in $k_{11} = 2$, $k_{21} = 1$.

Remark 6.2 *We formulate these theoretical considerations in terms of safe states, whereas in our algorithm, we have efficient ways of determining unsafe state. Finding complements is not difficult, however, so we can still use the algorithm.*

Definition 6.3 *An unfolding* $\mathbf{k}$ *is a* further unfolding *of* $\mathbf{k}'$ *if there is a pair of indices* I^0, J^0 *such that* $k_{I^0,J^0} > 1$ *and* $\mathbf{k}'$ *is obtained from* $\mathbf{k}$ *by removing a loop* $L^{\mathbf{k}_l}_{I^0,J^0 j_{s-1}}$ *with all its nested loops, i.e., removing* $I_1^{\mathbf{k}_1} \times \cdots \times I_{i_1-1}^{\mathbf{k}_{i_1-1}} \times L^{\mathbf{k}}_{I^0,J^0 j_{s-1}} \times I_{i_1+1}^{\mathbf{k}_{i_1+1}} \times \cdots \times I_n^{\mathbf{k}_n}$.

Lemma 6.4 *Let* $\mathbf{k}$ *be a further unfolding of* $\mathbf{k}'$*, where* $\mathbf{k}'$ *is obtained from* $\mathbf{k}$ *by removing the loop* $L_{I^0,J^0 j^0_{s-1}} = L_{i^0_1 \dots i^0_s, j^0_1 \dots j^0_{s-1}}$. *Then*

1. $k'_{I^0, j^0_1 \dots j^0_{s-2}} = k_{I^0, j^0_1 \dots j^0_{s-2}} - 1$

2. $k'_{I^0 I, j^0_1 \dots j^0_{s-2}(j_{s-1}-1)J} = k_{I^0 I, j^0_{s-2} j_{s-1} J}$ *for all* I, J*, for which* $I^0 I \in \mathcal{I}$ *and* $j_s > j^0_s$ *and* $k_{I^0 I, j^0_{s-2} j_s J}$ *is defined. In particular* $k'_{I^0 I, j^0_1 \dots j^0_{s-2} m J}$ *is undefined when* $m = k_{I^0, j^0_1 \dots j^0_{s-2}}$.

3. For all other indices $k'_{I,J} = k_{I,J}$.

Proof. Removing L_{I^0,J^0} has the effects

1. The number of L_{I^0} in $L_{i_1^0 \ldots i_{s-1}^0, j_1^0 \ldots j_{s-2}^0}$ is reduced by 1.

2. The loops $L_{I^0, j_1^0 \ldots j_{s-2}^0 j_{s-1}}$ in $L_{i_1^0 \ldots i_{s-1}^0, j_1^0 \ldots j_{s-2}^0}$ which had $j_{s-1} > j_{s-1}^0$ "move one up", i.e. get indexed $I^0, j_1^0 \ldots j_{s-2}^0 j_{s-1} - 1$ and the indices of their nested loops are changed accordingly.

3. Nothing else is affected except $L_{I^0, j_1^0 \ldots j_{s-2}^0 m}$ where $m = k_{I^0, j_1^0 \ldots j_{s-2}^0}$ is not there anymore, so $k'_{I,J}$ are undefined for these indices.

□

Definition 6.5 *We define a partial order of unfoldings by taking the transitive closure of the relation $\mathbf{k} \leq \mathbf{k}'$ if $\mathbf{k}'$ is a further unfolding of $\mathbf{k}$.*

The following proposition says, that if a further unfolding decreases the safe area, then there is an unfolding right before, which also decreased the safe area. Hence, if some unfolding does not increase the safe area, we do not have to unfold that even more. The idea of the proof is as follows: Suppose in Fig. 6 that all points equivalent to $\mathbf{x}$ are unsafe in deloopings $\mathbf{k}'$ such that $\mathbf{k} > \mathbf{k}'$ and that there are no $L_{I,J}$ with $J > \mathbf{1}$ below $\mathbf{y}$ (except the one containing $\mathbf{y}$).
There is still a path from $\mathbf{y}$ to $\mathcal{F}$ if we remove $L_{15,1}^{\mathbf{k}} \times I_2^{\mathbf{k}}$, i.e., there is a path from a point equivalent to $\mathbf{y}$ to $\mathcal{F}$ in a delooping before $\mathbf{k}$. Suppose that we can remove yet another loop $L_{I',J'j'}$ with $k_{I',J'} > 1$ (which is then above $\mathbf{y}$; in Fig. 6 it is $L_{23,2}$) and still have a path γ' from a point equivalent to $\mathbf{y}$ to $\mathcal{F}$ in the resulting delooping, then if we reinsert $L_{15,1}$, the geometry between $\mathbf{x}$ and $\mathbf{y}$ is the same, so γ still defines a path from $\mathbf{x}$ to $\mathbf{y}$ and together with γ' this gives a path from $\mathbf{x}$ to $\mathcal{F}$ in this lower (since we removed $L_{23,2}$) delooping. This is a contradiction.

Lemma 6.6 *Suppose $\mathbf{x}$ is safe in $\mathbf{k}$ and that for all $\mathbf{k}' < \mathbf{k}$, the set $\pi_{\mathbf{k}'}^{-1} \pi_{\mathbf{k}}(\mathbf{x})$ contains no safe points. Then*

1. If $x_{i_1^0} \in X_{i_1^0 i_2^0}$. Then $k_{i_1^0 i_2^0} \leq 1$ when $i_2 \leq i_2^0$ and for all loops $I = i_1^0 i_2 I'$ where $i_2 \leq i_2^0$, $k_{I,J} \leq 1$ or it is undefined.

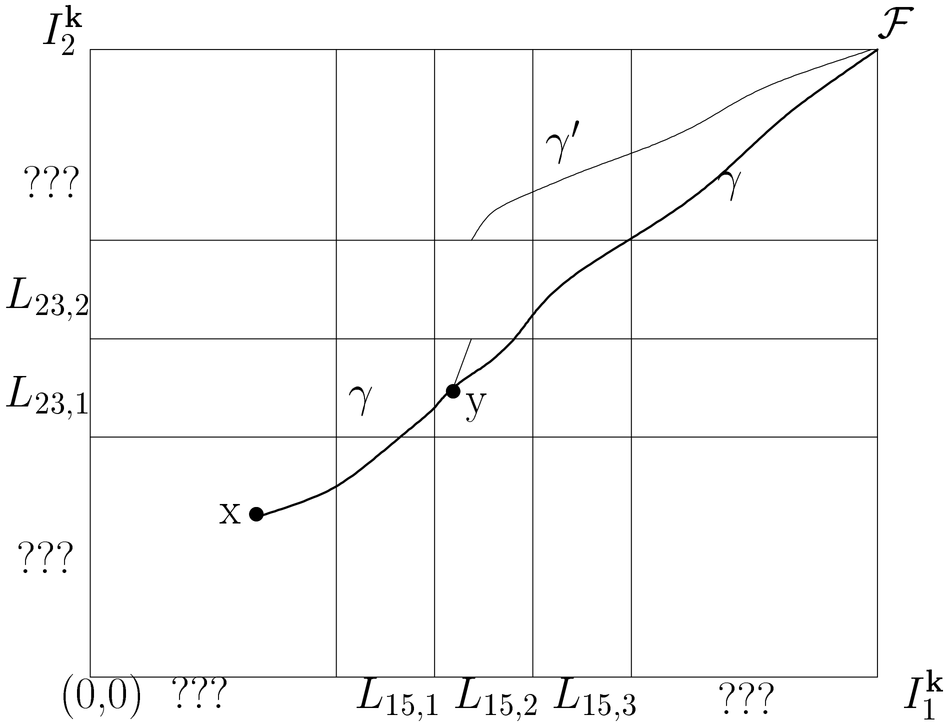

Figure 15: The basic idea of lemma 6.7

2. *If $x_{i_1^0} \in Y_{i_1^0}$, $k_{i_1^0 i_2} \leq 1$ for all i_2 and $k_{I,J} \leq 1$ or undefined for all $I = i_1^0 i_2 I'$.*

3. *If $x_i \in L_{I,J}$ and $x_i \notin L_{Ii_{s+1},Jj_s}$ for any pair i_{s+1}, j_s, then $J = \mathbf{1}$. Moreover, for any loop L'_I with $i'_1 = i_1$ and $I' < I$ in the lexicographical ordering, and for any J', $k_{I',J'}$ is 0, 1 or undefined.*

Proof. Let $\gamma : [0,T] \to X_{\mathbf{k}}$ be a dipath from $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ to $\mathcal{F}$.

Since γ does not traverse the loops before $\mathbf{x}$, those could be removed without affecting γ giving a $\mathbf{k}' < \mathbf{k}$, where a point in $\pi_{\mathbf{k}'}^{-1}\pi_{\mathbf{k}}(\mathbf{x})$ is safe. Hence, 1 and 2 follow immediately. For 3) suppose $J = j_1 j_2 \ldots j_s$ and $j_l > 1$. Since γ only traverses points (and thus loops) above $\mathbf{x}$, we could then lift $\pi \circ \gamma$ to a unfolding $\mathbf{k}'$ obtained from $\mathbf{k}$ by removing $L_{i_1 i_2 \ldots i_l, \mathbf{1}}$ in the sense of 6.3, which would lead to a contradiction. Moreover, for $i'_1 = i_1$ and $I' < I$ in the lexicographical ordering, we have $L^+_{I',J'} < x_{i_1}$ for any J'. Hence $k_{I',J'}$ is

0, 1 or undefined.

□

Proposition 6.7 *Suppose* $\mathbf{x}$ *is a safe point in* $\mathbf{k}$ *and there is a pair* I, J *such that* $k_{I,J} > 1$*. Suppose moreover that for all* $\mathbf{k}' < \mathbf{k}$*, the set* $\pi_{\mathbf{k}'}^{-1}\pi_{\mathbf{k}}(\mathbf{x})$ *contains no safe points.*

Then there is a state $\mathbf{y} \in X_{\mathbf{k}}$ *which was found safe in an unfolding just before* $\mathbf{k}$*, i.e., there is an* I^0 *and an unfolding* $\mathbf{k}''$ *such that*

1. *All nested loops in* $L^{\mathbf{k}}_{I^0,\mathbf{1}}$ *are delooped at most once and* $\mathbf{k}''$ *is obtained from* $\mathbf{k}$ *by removing* $L^{\mathbf{k}}_{I^0,\mathbf{1}}$*.*
2. *Some point in* $\pi_{\mathbf{k}''}^{-1}\pi_{\mathbf{k}}(\mathbf{y})$ *is safe in* $\mathbf{k}''$
3. *All points in* $\pi_{\tilde{\mathbf{k}}}^{-1}(\pi_{\mathbf{k}}(\mathbf{y}))$ *are unsafe whenever* $\tilde{\mathbf{k}} < \mathbf{k}''$*.*

Proof. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be the coordinates of $\mathbf{x}$ in $X_{\mathbf{k}}$. Let $\gamma : [0, T] \to X_{\mathbf{k}}$ be a dipath from $\mathbf{x}$ to $\mathcal{F}$.

1. Since the forbidden area is a union of rectangles parallel to the coordinate-axes, we can deform γ so that if for some set of indices $i_0, \ldots i_m$, $\gamma_{i_1}(t_0), \gamma_{i_2}(t_0), \ldots, \gamma_{i_m}(t_0)$ are all integers, then there is an $\varepsilon > 0$ such that for $t \in]t_0, t_0 + \varepsilon[$, $(\gamma_{i_1}(t), \gamma_{i_2}(t) \ldots, \gamma_{i_m}(t))$ has at most one non-integer coordinate. (In particular γ enters at most one loop in the interval $[t_0, t_0 + \varepsilon[$). Suppose in the following, that γ has this property.
2. Let $H = \{t \in [0, T] | \exists I \in \mathcal{I}$ and $J \in \mathcal{J}(I, \mathbf{k})$ such that $\gamma_i(t) \in L_{I,J}$ and $J > \mathbf{1}\}$ then H is non empty: Suppose H is empty. Then, since $\gamma([0, T])$ is connected, we conclude that every loop above or including $\mathbf{x}$ is delooped at most once, so $k_{I,J} \leq 1$ or undefined for any I, J such that $x_i \leq L^+_{I,\mathbf{1}}$, but since all loops below $\mathbf{x}$ are delooped at most once (by Lemma 6.6 1 and 2), this is a contradiction.

Because of 1, there is a $t_0 \in H$ which is sufficiently small to imply tha when $\mathbf{y} = \gamma(t_0)$, $y_{i^0} \in L_{I^0,J^0}$, the first $L_{I,J}$ with $J > \mathbf{1}$ intersected by γ, i.e.,

$$\forall t \in H, t \leq t_0 : \gamma(t)_{i_0} \in L_{I^0,J^0} \text{ and } \gamma(t_i) \in L_{I,J}, I \neq I^0 \Rightarrow J = \mathbf{1}.$$

Then

- $\mathbf{y}$ is safe, since γ intersects it.

- There is an l such that $j^0_l = 2$ and $j^0_u = 1$ otherwise. This follows from 1 above.

- Let $\mathbf{k}''$ be obtained from $\mathbf{k}$ by removing the loop $L^{\mathbf{k}}_{i^0_1 i^0_2 \ldots i^0_l, \mathbf{1}}$ and let $\mathbf{y}'' \in X_{\mathbf{k}''}$ have coordinates $y''_{i^0} = y_{i^0} - |L^{\mathbf{k}}_{i^0_1 i^0_2 \ldots i^0_l, \mathbf{1}}|$ and $y''_i = y_i$ else. Then $\mathbf{y}''$ is equivalent to $\mathbf{y}$ and safe, since $\pi_{\mathbf{k}} \circ \gamma|_{[t_0,T]}$ lifts to $\gamma'' : [t_0, T] \to X_{\mathbf{k}''}$, where $\gamma''_{i^0}(t) = \gamma_{i^0}(t) - |L^{\mathbf{k}}_{i^0_1 i^0_2 \ldots i^0_l, \mathbf{1}}|$ and $\gamma''_i(t) = \gamma_i(t)$ else.

Now suppose that there is a safe point $\tilde{\mathbf{y}} \in \pi^{-1}_{\tilde{\mathbf{k}}} \pi_{\mathbf{k}}(y)$ in some delooping $\tilde{\mathbf{k}}$ such that $\tilde{\mathbf{k}}$ is obtained from $\mathbf{k}''$ by removing one or more loops $L_{\tilde{I}, \tilde{J} \tilde{j}_{s-1}}$. Then $\tilde{I} \geq I^0$, since we require $k_{\tilde{I}, \tilde{J}} > 1$.

Suppose, without loss of generality, by Lemma 6.6, that $\tilde{y}_{i^0_1} \in L^{\tilde{\mathbf{k}}}_{I^0, \mathbf{1}}$.

Let $\mathbf{k}'$ be obtained from $\mathbf{k}''$ by reinserting $I^{\mathbf{k}''}_1 \times \cdots \times I^{\mathbf{k}''}_{i^0_1 - 1} \times L^{\mathbf{k}}_{i^0_1 i^0_2 \ldots i^0_l, \mathbf{1}} \times I^{\mathbf{k}''}_{i^0_1 + 1} \times \cdots \times I^{\mathbf{k}''}_n$. Then $\mathbf{k}' < \mathbf{k}$, and we claim that there is a point equivalent to $\mathbf{x}$ which is safe in $X_{\mathbf{k}'}$. Let $\mathbf{x}' \in \pi^{-1}_{\mathbf{k}'}(\pi_{\mathbf{k}}(x))$ be the point such that $x_i \in L_{I,J} \Rightarrow J = \mathbf{1}$ and let $\mathbf{y}' \in \pi^{-1}_{\mathbf{k}'} \pi_{\mathbf{k}}(\mathbf{y})$ be determined by $y'_{i_0} \in L_{I^0, J^0}$ and $y'_i \in L_{I,J} \Rightarrow J = \mathbf{1}$ for $i \neq i^0$. Then the coordinates of $\mathbf{x}'$ and $\mathbf{y}'$ in $X_{\mathbf{k}'}$ are the coordinates of $\mathbf{x}$ and $\mathbf{y}$ in $X_{\mathbf{k}}$ and $\gamma : [0, t_0] \to X_{\mathbf{k}'}$ is a dipath from $\mathbf{x}'$ to $\mathbf{y}'$. We reinserted $L^{\mathbf{k}}_{i^0_1 i^0_2 \ldots i^0_l, \mathbf{1}}$ below $\tilde{\mathbf{y}}$, and did not alter $X_{\tilde{\mathbf{k}}}$ above $\tilde{\mathbf{y}}$. Hence $\gamma'_{i^0}(t) = \tilde{\gamma}_{i^0}(t) + |L^{\mathbf{k}}_{i^0_1 i^0_2 \ldots i^0_l, \mathbf{1}}|$ and $\gamma'_i(t) = \tilde{\gamma}_i(t)$ else is a lift of $\pi_{\tilde{\mathbf{k}}} \circ \tilde{\gamma}$ to a path in $X_{\mathbf{k}'}$ from $\mathbf{y}'$ to $\mathcal{F}$

Hence $\mathbf{x}'$ is safe, which is a contradiction.

□

Definition 6.8 *A further delooping from* $\mathbf{k}$ *to* $\mathbf{k}'$ increases the safe region, *if there is a point* $x \in X$ *such that for all* $\mathbf{k}'' < \mathbf{k}$, *all points in* $\pi^{-1}_{\mathbf{k}''}(x)$ *are unsafe, and there is a safe point in* $\pi^{-1}_{\mathbf{k}'}(x)$. *If* $\pi^{-1}_{\mathbf{k}}(x)$ *is nonempty, the further delooping also* decreases the unsafe region.

Theorem 6.9 *To determine the unsafe region of a concurrent system with loops as in 3.1, only a finite number of unfoldings is needed.*

Proof. Define a digraph (a directed graph) G with vertices $\mathbf{k}$, all possible finite deloopings of the system. There is an edge from $\mathbf{k}$ to $\mathbf{k}'$ if $\mathbf{k} < \mathbf{k}'$ and $\mathbf{k}'$ is obtained from $\mathbf{k}$ by inserting one loop with all its nested loops delooped at most once and if the safe area increases from $\mathbf{k}$ to $\mathbf{k}'$.

By 6.7 and with the notation from there, and by 6.3, if the delooping $\mathbf{k}$ increases the safe region and $k_{I,J} > 1$ for some I, J, then there is a $\mathbf{k}''$, obtained from $\mathbf{k}$ by removing a loop $L^{\mathbf{k}}_{I^0,\mathbf{1}}$, where $k_{I^0,\mathbf{1}} > 1$; such that there is an edge in G from $\mathbf{k}''$ to $\mathbf{k}$. Repeating this, we see that there is a dipath in G to $\mathbf{k}$ from some $\tilde{\mathbf{k}}$ such that $\tilde{k}_{I,J} \leq 1$ for all I, J. Since the minimal volume (measured in $T^{\mathbf{1}}$) added to the safe area along an edge in G is 1 and the total volume V is finite, the length of this dipath is at most V, hence $k_{I,J} \leq V$ for all I, J.

□

We just prove that there is a finite number of unfoldings needed, but fortunately this maximum is never obtained. The true number of unfoldings needed is of course much lower in general:

1. Delooping a loop will only affect the states with a coordinate *below* or in the loop.
2. The length of the dipath to $\mathbf{k}$ is actually bounded by the volume of the states which are either unsafe or not represented in $\tilde{\mathbf{k}}$. This is of course less than the total volume.
3. We conjecture that only delooping loops of type 5 Fig. 4 will reduce the unsafe area.

It is important to notice 2. This means, that we only have to consider the forbidden region and then add or subtract unsafe points. Hence there is still the advantage from the original deadlock algorithm, that we do not consider the whole state space, but the "dual" part, the forbidden region and the unsafe region.

We have not implemented this yet, but building the graph G can be done using the deadlock algorithm on the deloopings. One way of doing this is to

- Calculate the unsafe area in all deloopings with loops delooped at most once. By Lemma 4.12, this is enough to determine the deadlocks as the intersection of the sets of deadlocks.
- For all further deloopings corresponding to inserting loops with nested loops delooped at most once, calculate the unsafe region as a subset of $T^{\mathbf{1}}$ and whether it contains the unsafe region reached before. Do this in order of magnitude, which is defined $|\mathbf{k}| = \Sigma_{I,J} k_{I,J}$. An edge in G will

increase the magnitude by at most $M = \max_{i_1 i_2 \in \mathcal{I}} \#\{I \in \mathcal{I} \big| \ i_1 i_2 < I\}$, the maximal number of loops embedded in a loop in B. If there is no increasing of the safe region (reduction of the unsafe region) in any deloopings from size N to size up to $N + M$, there will not be more edges found in G and hence no more safe states to find.

- Now the unsafe region is the intersection of the unsafe regions $U(\mathbf{k})$ where $\mathbf{k}$ is connected to some $\mathbf{k}' \leq \mathbf{1}$ in G.

A priori, there seems to be a huge number of deloopings to check to build the graph G, but with the reductions mentioned above, this can be simplified a lot.

7 Future prospects

The implementation of the deadlock algorithm [FGR98a] for systems with loops and branchings, which geometrically give rise to a product of digraphs, is an important project, which should be feasible now.

On the mathematical side, there are many open questions and things to be defined and developed. In looped systems, the local partial orders have to be taken seriously, and this raises interesting question about dicoverings, diliftings, difibrations, etc. as the ditopology analogues of wellknown concepts in algebraic topology, see for instance [Bre93].

With this machinery built, we may be able to give a much more slick argument for many of the results in this article, but that still has to be seen.

References

[Bre93] Glen E. Bredon. *Topology and Geometry*, volume 139 of *GTM*. Springer-Verlag, 1993.

[Dij68] E.W. Dijkstra. Co-operating sequential processes. In F. Genuys, editor, *Programming Languages*, pages 43–110. Academic Press, New York, 1968.

[FGR98a] L. Fajstrup, É. Goubault, and M. Raussen. Detecting Deadlocks in Concurrent Systems. In D. Sangiorgi and R. de Simone, editors, *CONCUR '98; Concurrency Theory*, volume 1466 of *Lect. Notes*

Comp. Science, pages 332 – 347, Nice, France, September 1998. Springer-Verlag. 9th Int. Conf., Proceedings.

[FGR98b] L. Fajstrup, É. Goubault, and M. Raussen. Detecting Deadlocks in Concurrent Systems. DTA/LETI/DEIN/SLA 98-61, LETI (CEA - Technologies Avancées), Saclay, France, August 1998. 25 pp.

[FGR99] L. Fajstrup, É. Goubault, and M. Raussen. Algebraic topology and concurrency. Technical Report R-99-2008, Department of Mathematical Sciences, Aalborg University, 9220 Aalborg Øst, 1999.

[Gou95] É. Goubault. *The Geometry of Concurrency.* PhD thesis, Ecole Normale Superieure, Paris, 1995.

[Gun94] J. Gunawardena. Homotopy and concurrency. *Bulletin of the EATCS*, 54:184–193, 1994.

[Pra91] V. Pratt. Modeling concurrency with geometry. *Proc. of the 18th ACM Symposium on Principles of Programming Languages.*, 1991.

[vG91] R. van Glabbeek. Bisimulation semantics for higher dimensional automata. Technical report, Stanford University, 1991.

Department of Mathematical Sciences, Aalborg University, Fredrik Bajers Vej 7E, DK 9220 Aalborg Ø
E-mail address: fajstrup@iesd.auc.dk