Adapting Biochemical Kripke Structures for Distributed Model Checking

Susmit Jha[1] and R.K. Shyamasundar[2,⋆]

[1] Department of Computer Science, Indian Institute of Technology, Kharagpur, India
susmit.kumar.jha@ieee.org

[2] School of Technology and Computer Science, Tata Institute of Fundamental Research, Bombay, India
shyam@tcs.tifr.res.in

Abstract. In this paper, we use some observations on the nature of biochemical reactions to derive interesting properties of qualitative biochemical Kripke structures. We show that these characteristics make Kripke structures of biochemical pathways suitable for assumption based distributed model checking. The number of chemical species participating in a biochemical reaction is usually bounded by a small constant. This observation is used to show that the Hamming distance between adjacent states of a qualitative biochemical Kripke structures is bounded. We call such structures as Bounded Hamming Distance Kripke structures (BHDKS). We, then, argue the suitability of assumption based distributed model checking for BHDKS by constructively deriving worst case upper bounds on the size of the fragments of the state space that need to be stored at each distributed node. We also show that the distributed state space can be mapped naturally to a hypercube based distributed architecture. We support our results by experimental evaluation over benchmarks and biochemical pathways from public databases.

1 Introduction

Recently, there has been a lot of work in the application of formal methods for the modeling and reasoning of biochemical pathways. A popular approach uses the formal model of Kripke structure derived from boolean abstractions of biochemical reactions [6,4]. Model checking of these Kripke structures is capable of deriving valuable information about the underlying biochemical pathways that cannot be understood from classical simulation techniques. However, model checking techniques suffer from state space explosion and there have been several investigations into the scalability of model checking techniques [1,3,7].One such method is the technique of assumption based distributed model checking as envisaged in [2].

However, little effort has been made in the direction of exploiting properties specific to biochemical Kripke structures for the design of scalable model checking approaches. We take the assumption based distributed model checking paradigm [2], where the state space of a system is partitioned into several distributed nodes, as the basis of our work.

[⋆] Current address. IBM India Research Lab Block1, IIT Delhi, Hauz Khas, New Delhi 110016, India, rshyamas@in.ibm.com

C. Priami et al. (Eds.): Trans. on Comput. Syst. Biol. VII, LNBI 4230, pp. 107–122, 2006.

Biochemical Kripke structures have been well studied in BIOCHAM [6,4]. We develop a framework for distributing the state space of a biochemical Kripke structure among several distributed nodes for model checking, by using structural properties of Kripke structures derived from biochemical systems. In this paper, we present the following results:

- Two states in a Kripke structure derived from biochemical pathways are connected by a transition only if the Hamming distance between their propositional labels is bounded by a small constant derived from the stoichiometry of the underlying biochemical reactions. We call such structures as k - Bounded Hamming Distance Kripke structures (BHDKS) where k is a small constant obtained from the stoichiometry of the reactions .
- Bounded Hamming Distance Kripke structures can be well partitioned into fragments each having a size that can be made small enough to be only polynomial in the number of propositions of the Kripke structure (N), and hence amenable to extensive fragmentation [1] for assumption based distributed model checking. The result shows that it is possible to split the exponential state space of the BHDKS $(O(2^N))$ into fragments each of which is only polynomial in the number of the propositions involved $(O(N^p)$, where p is a small constant).
- When the number of distributed nodes across which the state space is to be distributed is not too large (smaller than $2^{N/k}$ for a k - Bounded Hamming Distance Kripke structure with N atomic propositions), we present a hypercube based fragmentation approach which forms smaller fragments and ensures that the neighbours of all the states on a distributed node lie only on the adjacent distributed nodes in the hypercube.

We also note that a k - BHDKS with n states can be partitioned into $n^{1-1/k}$ size fragments along the nodes of a hypercube despite the fact that, in general, the corresponding class of graphs do not have "good" vertex separators i.e., $n^{1-\epsilon}$ separators for any $\epsilon > 0$.

We organize the rest of the paper as follows: Section 2 presents new insights into Kripke structures formed from biological systems by showing that the Hamming distance between any two successive states in the Kripke structure is bounded by a small constant. Such Kripke structure are referred to as bounded Hamming distance Kripke structures (BHDKS). We use these structures to derive a bound on the edge density in Section 3. Section 4 presents relevant background results and definitions related to distributed model checking. In section 5, we use the existence of a small bound on the Hamming distance between successive states in BHDKS to argue that biochemical pathways are more amenable to distributed model checking techniques by presenting the worst case bounds on the size of the fragments of the distributed Kripke structure. The proof presented is constructive and suggests methods of partitioning BHDKS. We discuss the results of our experimental evaluation on benchmarks and public databases in Section 6. The paper concludes with section 7 identifying scopes for further work.

[1] We will also illustrate that general Kripke structures need not have any reduction in size during fragmentation.

2 Bounded Hamming Distance of Biochemical Kripke Structures

In this section, we shall describe the modeling of biochemical pathways and demonstrate as to how the characteristics of biochemical pathways lead to their representation as BHDKS.

2.1 Background

In the abstract boolean Kripke structure model [4,5,6], a biochemical reaction takes the system from a state with biochemical entities matching the lefthand side of the reaction rule, into one of the other states in which the biochemical entities of the righthand side have been added. The biochemical entities which appear only in the lefthand side of the rule and not in the righthand side may be nondeterministically present or absent in the target state. By using this boolean abstraction, such models are capable of reasoning about all possible behaviors of the system with unknown concentration values and unknown kinetic parameters[4]. This modeling is particularly useful for complex chemical systems like biochemical pathways where even a boolean abstraction can generate valuable results. It is also now well appreciated that biological models, despite their hybrid nature, indeed have many digital (boolean) controls. In the model checking algorithm, each biochemical entity is associated with a proposition. If the biochemical entity is present in a state, the associated boolean proposition is *true*; other wise, it is *false*. Thus, the biochemical Kripke structure makes a transition from one state to another by "executing" a biochemical reaction and the truth values of the boolean propositions change to reflect the biochemical entities added or removed from the system.

The detailed methodology which takes a biochemical pathway as input and forms a Kripke structure is presented in [5]. In the following, we shall illustrate the derivation of Kripke structures for biochemical pathways through some examples.

Example 1. Simple modeling of a chemical reaction.
Here, the presence and absence of reactants is encoded in the state tuple of the Kripke structure. This is an implicitly assumed reasonable assumption in biochemical pathway representations. Let us try to capture a transition wherein A and B react to form C and D. A typical one is denoted:

$$A + B + \neg C + \neg D \rightarrow A + B + C + D$$

which is interpreted as follows: The transition is defined from all states where the propositions associated with A and B are true, and C and D are false to those states where propositions associated with C and D are true as well as A and B are true. The reasonable assumption is that the reaction does not consume all its reactants and hence, some quantity of reactants A and B are still present after the reaction.

Example 2. Abstract Modeling.
Consider the scenario of A and B reacting to form C and D,

$$A + B \rightarrow C + D$$

and we want to nondeterministically capture all possible scenarios. This is captured by

$$\begin{array}{l}
A + B + \neg C + \neg D \rightarrow \neg A + B + C + D\\
A + B + \neg C + \neg D \rightarrow A + \neg B + C + D\\
A + B + \neg C + \neg D \rightarrow \neg A + \neg B + C + D\\
A + B + \neg C + \neg D \rightarrow A + B + C + D\\
A + B + C + \neg D \rightarrow \neg A + B + C + D\\
A + B + C + \neg D \rightarrow A + \neg B + C + D\\
A + B + C + \neg D \rightarrow \neg A + \neg B + C + D\\
A + B + C + \neg D \rightarrow A + B + C + D\\
A + B + \neg C + D \rightarrow \neg A + B + C + D\\
A + B + \neg C + D \rightarrow A + \neg B + C + D\\
A + B + \neg C + D \rightarrow \neg A + \neg B + C + D\\
A + B + \neg C + D \rightarrow A + B + C + D\\
A + B + C + D \rightarrow \neg A + B + C + D\\
A + B + C + D \rightarrow A + \neg B + C + D\\
A + B + C + D \rightarrow \neg A + \neg B + C + D\\
A + B + C + D \rightarrow A + B + C + D
\end{array}$$

In an abstract model, each chemical reaction is interpreted as a set of chemical reactions where some of the reactants may be present even after the execution of the reaction and the products may be present even before the execution.

Example 3. The E. Coli K-12 Pathway: leucine biosynthesis [9].
Using the following abbreviations: K — 2-keto-isovalerate, AC — Acetyl-CoA, C— Coenzyme A, H— 3-carboxy-3-hydroxy-isocaproate, T — 2-D-threo-hydroxy-3-carboxy-isocaproate, CN — CO_2 NADH, N — NADH, M — 2-keto-4-methyl-pentanoate, L — L-leucine, AG — α-ketoglutarate, G — L-glutamate, the biochemical pathway is given by the following reactions:

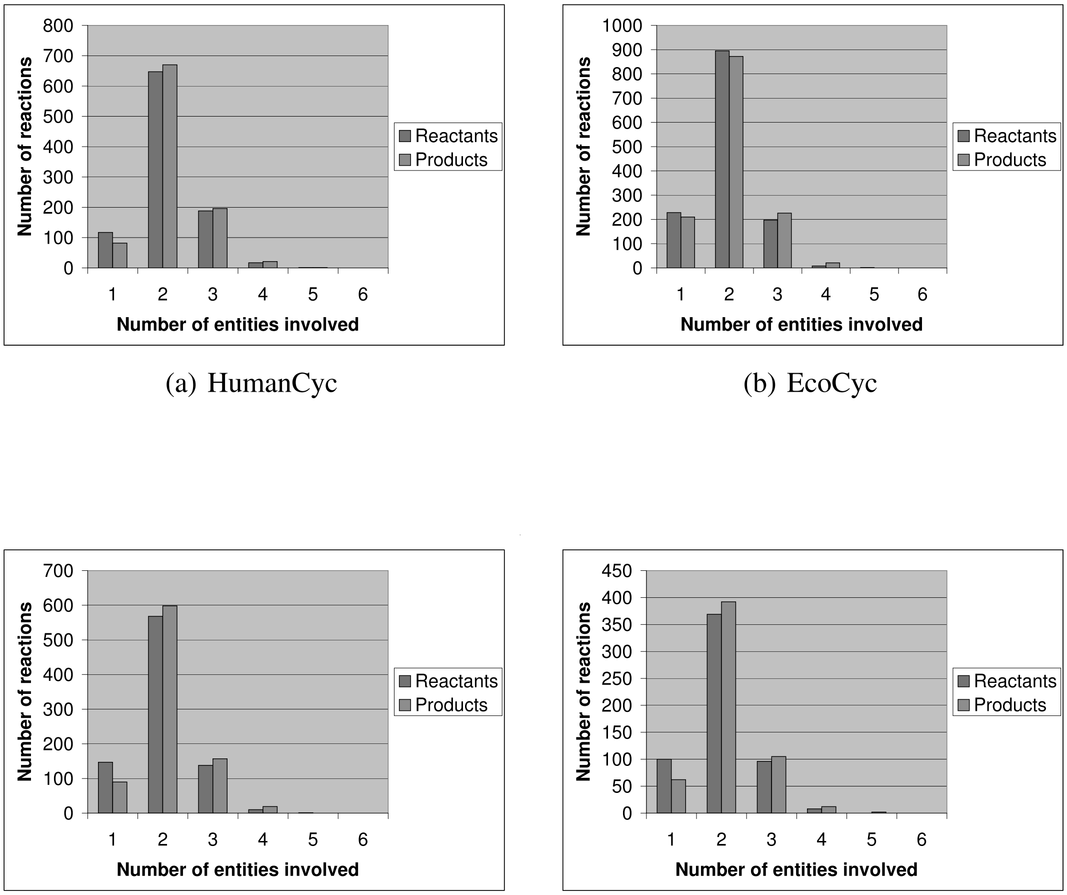
$K + AC \rightarrow C + H$
$H \rightarrow T$
$T + N \rightarrow M + CN$
$M + G \rightarrow L + AG$
$AG \rightarrow K$

The reactions can be easily extrapolated to their abstract interpretation.

It may be noted that a Kripke structure is an asynchronous formalism. In particular, two reactions occurring "simultaneously" can be modeled as one occurring after another because of the nondeterministic modeling with respect to the reactants and the asynchronous interleaving semantics of Kripke structures.

2.2 Bound on the Number of Chemical Entities Involved in a Reaction

A study of pathways [9,11] shows that for biochemical pathways, the number of biochemical entities reacting in a chemical reaction is fairly small. As illustrated in Fig. 1, almost 60% of the reactions in each of these databases have no more than two reactants or two products. Also, no reaction was found with more than six reactants or products in these databases. The statistics gathered from the databases of these widely differing organisms shows that there is a very low probability of the interaction of more than a few entities at the atomic level. Hence, all biochemical reactions indeed involve interaction of a fairly small number of chemical entities, and the number of chemical entities produced as a result of biochemical reactions are also small. We may contrast this with an arithmetic operation $a := a \times b$, a system wide reset in a VLSI chip or the setting of bits in a long flag register. Each of these can take the Kripke structure of these hardware or software systems from one state to another such that the Hamming distance between them is arbitrarily large.



(a) HumanCyc

(b) EcoCyc

(c) AnthraCyc

(d) YeastCyc

Fig. 1. The HumanCyc, EcoCyc, AnthraCyc and YeastCyc Databases Reactions Summary: The bar charts clearly show that most reactions have small number of reactants and products. There is no reaction having more than 6 reactants or products among some 3000 biochemical reactions in these databases.

2.3 Bounded Hamming Distance Kripke Structures

In order to separate the development of the partitioning algorithm from the details of the biochemical Kripke structure [6], we consider the earlier introduced BHDKS model. This abstract model is sufficient for the construction of our partitioning algorithm.

Definition 1. *Let K = (S,R,$\mathcal{AP}$,$\mathcal{L}$,F) be a Kripke structure, where S is the set of states, R is the transition relation, $\mathcal{AP}$ is the set of atomic propositions, $\mathcal{L}$ is the labeling of states with atomic propositions, F is the set of final states, and H(x,y) denotes the Hamming distance between x and y. Then, K is called a k - Bounded Hamming Distance Kripke structure iff*

$$\forall s, s' \in S, \quad R(s, s') \implies (H(\mathcal{L}(s), \mathcal{L}(s')) \leq k)$$

Intuitively, a k-BHDKS has a transition between two states in the Kripke structure only if the Hamming distance between the propositional labels of these states is at most k.

Theorem 1. *A biochemical Kripke structure is a k-BHDKS for some small k.*

Proof. Let K be a biochemical Kripke structure[6]. Consider two states s and s' in K. If there is no transition from s to s', we are done.

If there is a transition from s to s', then the system executes some reaction at state s. From our earlier observation, the reaction has at most r reactants and at most p products, where r and p are small. When the reaction is executed, the reactants can nondeterministically be removed from the system, while the products are added to the system. Thus, s' can differ from s in at most $k = r + p$ chemical entities, that is $H(s, s')^2 \leq k$. Hence, the biochemical Kripke structure is a k-BHDKS for some small k.

3 Density of Bounded Hamming Distance Kripke Structures

In this section, we shall establish certain properties of BHDKS and show that they are "reasonably sparse" in nature. We use the bound on the Hamming distance of neighbouring states in a BHDKS to derive a bound on the edge density of these Kripke structures. We show that the edge density is only polynomial in the number of propositions of the state space.

Theorem 2. *A state in the k - Bounded Hamming Distance Kripke structure with* $\log n$ *number of propositions (where* $n > 1$*) has a degree of at most* $(\log n)^k$.

Proof. Let s be any state such that $s \in S$, where S is the state space of the k - Bounded Hamming Distance Kripke structure. Now, consider all possible neighbours $N(s)$ of s. From the definition of BHDKS, we know that $s' \in N(s)$ only if $H(s, s') \leq k$. Now, we define a set of states P_i such that $p \in P_i$ if and only if $H(s, p) = i$. Further, let us define $P = \bigcup_{i=0...k} P_i$. Clearly,

- $|P_i| = \binom{log(n)}{i}$
- $P_i \cap P_j = \phi$

So, $|P| = |\bigcup_{i=0...k} P_i|$
$= \sum |P_i| \quad (\because P_i \cap P_j = \phi)$
$= \sum_{i=0}^{k} \binom{log(n)}{i}$
$\leq (log(n))^k$
Also, $N(s) \subset P$. Hence, $|N(s)| \leq |P| \leq (log(n))^k$
Thus, each state has no more than $(log(n))^k$ neighbours.

Thus, the number of transitions in a Bounded Hamming Distance Kripke structure are no more than polynomially (in the number of propositions in the Kripke structure) larger than the number of states.

4 Background on Assumption Based Distributed Model Checking

Distributed model checking as presented in [1,2] decomposes the Kripke structure into fragments. Each distributed node in the distributed computing cluster stores only one of

[2] Eventually, we will use the notation H(s,s') to mean $H(\mathcal{L}(s), \mathcal{L}(s'))$.

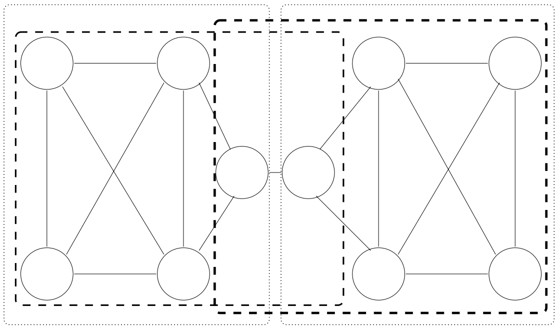


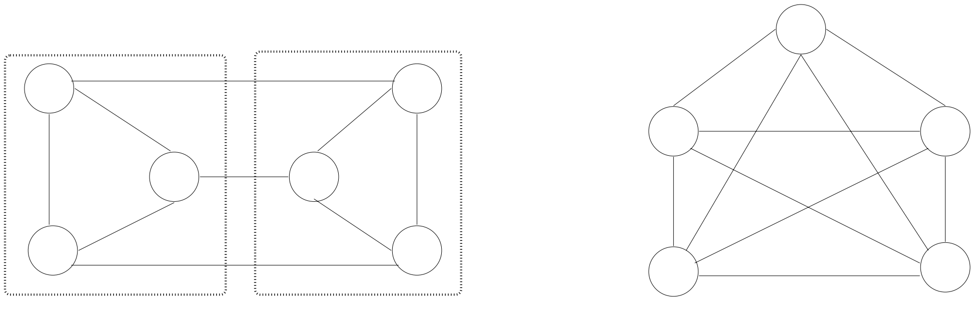
Fig. 2. An example of a Kripke structure and the fragments formed by dividing into two parts. The dotted boxes surround the subsets used for constructing the partition. The dashed lines show the actual partitions themselves. Observe that the partition was able to reduce the size of the Kripke structure rather well. Also, the undirected edges indicate transitions possible in both directions.

these fragments; hence, the size of the model checking problem which can be processed by the distributed model checking algorithm is bounded by the size of the smallest fragments we can construct.

Definition 2. *A Kripke structure $M' = (S', R')$ is a fragment of a Kripke structure $M = (S, R)$ iff*

- $S' \subseteq S$*,*
- $R' \subseteq R$ *and*
- $\forall (s, s') \in R$ *if* $s \in S'$*, then either* $(s, s') \in R'$ *or* $\nexists t \in S'$ *such that* $(s, t) \in R'$*.*

Given a Kripke structure M, it is now pertinent to generate these fragments. Any subset of the state space can be naturally extended to form a fragment by including those states which are immediate neighbours of the states in this subset and the rest of the Kripke structure, as shown in Fig 2. Formally,



(A bad instance) **(A 5-clique)**

Fig. 3. Bad instances for distributed model checking:In the left figure, the subsets are shown by dotted boxes. For these subsets, each of the fragment will be as large as the original Kripke structure and the purpose of the distributed algorithm will fail. In the right figure, a 5-clique is shown. Irrespective of the choice of our subsets, each fragment will be as large as the whole Kripke structure once again.

Definition 3. *Let $M = (S, R)$ be a Kripke structure and $T \subseteq S$. The distributed fragment of the Kripke structure $Fragment_M(T) = (S_T, R_T)$ is defined as*

- $S_T = \{s \in S | s \in T \vee \exists s' \in T$ *such that* $(s', s) \in R\}$
- $R_T = \{(s_1, s_2) \in R | s_1 \in T, s_2 \in S_T\}$

Thus, a distributed computation node i in the distributed model checking paradigm contains all states from T (called *core* states) and their immediate predecessors
$S_T \setminus T$(called *border* states).

The central idea of the distributed algorithm in [2] is presented in the following algorithm:

```
proc Distributed Algorithm( input: total Kripke Structure $M$, $\psi$, $f$;
output:$A_{f(s_0)}(s_0, \psi)$)
  Split $M$ into $K_i$;
  for all $i \in \{1, \ldots, n\}$ do in parallel { for all $K_i$ }
  Take the initial assumption function;
    repeat
      repeat
        Compute all you can;
        Send relevant information to other nodes;
        Receive relevant information from other nodes;
      until all processes reach fixpoint;
      Extrapolate additional information;
    until all is computed;
  Return result for the initial state $s_0$;
  od
end
```

In order to abstract the concerns of the assumption based distributed model checking problem and allow a mathematical formulation of the fragmentation problem, we define the notion of a separator of a set of states in a Kripke structure.

Definition 4. *Given a set of states $T \subset S$ of the Kripke structure K, the set V is said to be a separator of T w.r.t S iff*

- $V \subset S$
- *There is no path from a state in $S \setminus (T \cup V)$ to a state in T which does not pass through some state in V.*
  That is, in the graph formed by removing V from S, $K_V = (S \setminus V, R \setminus R_V)$,
  $\forall t \in T, \forall s \in S \setminus (V \cup T)$, *there is no path from s to t in K_V.*
  Clearly, $R_V = \{(x, y) \in R | x \in V$ or $y \in V\}$.

Intuitively, T is the *core* of the fragment and V is the set of *border* states. Thus, any set of states along with its separator with respect to the rest of the Kripke structure contains a fragment for assumption based distributed model checking.

5 Fragmentation of BHDKS

Several efforts have been made to solve the problem of state space explosion in model checking. The art and science of symbolic model checking [3] has made considerable progress in increasing the size of the state space that can be model checked. Distributed Model Checking is a technique which aims at exploiting the memory of a large number of systems in a distributed environment. In the past, there has been work on developing good distributed model checking algorithms for software by making use of the information in control flow graphs [8]. However, to the best of our knowledge, there has been no work on developing distributed algorithms for biochemical systems that establishes worst case bounds on the size of each fragment by the use of structural properties of biochemical Kripke structures. The background definitions related to assumption based distributed model checking are presented in Sec. 4. We just recall the definition of a fragment here.

Definition 5. *Let $M = (S, R)$ be a Kripke structure and $T \subseteq S$. The distributed fragment of the Kripke structure $Fragment_M(T) = (S_T, R_T)$ is defined as*

- $S_T = \{s \in S | s \in T \vee \exists s' \in T$ *such that* $(s, s') \in R\}$
- $R_T = \{(s_1, s_2) \in R | s_1 \in T, s_2 \in S_T\}$

Thus, a distributed computation node i in the distributed model checking paradigm contains all states from some subset T of S(called *core* states) and their immediate predecessors $S_T \setminus T$(called *border* states). Thus, any set of states, along with its vertex separator with respect to the rest of the Kripke structure, contains a fragment for assumption based distributed model checking. A set of vertices V is said to be a vertex separator of T with respect to S if all paths from $S \setminus T$ to T pass through some vertex in V. Now, we will present results on the size of separators for BHDKS.

5.1 Polynomial Separators for BHDKS

We will first show that the size of the separator of an arbitrary subset of the state space of a BHDKS is at most polynomially (in the number of propositions in the Kripke structure) larger than the subset itself.

Theorem 3. *Given any set $T \subset S$ of the state space of a k - Bounded Hamming Distance Kripke structure $K = (S, R)$ with log(n) propositions, the size of the smallest separator V of T with respect to S is no more than $|T|.(log(n))^k$.*

Proof. For each state $t \in T$, consider the neighbours of t. As shown earlier, $N(t) \leq (log(n))^k$. Clearly, $\bigcup_{t \in T} N(t)$ is a separator of T with respect to S. Hence, the size of the smallest separator of $T = |V|$

$$\leq |\bigcup_{t \in T} N(t)|$$
$$\leq \sum_{t \in T} |N(t)|$$
$$\leq |T|.(log(n))^k.$$

Corollary 1. *Given any set $T \subset S$ of the state space of a k - Bounded Hamming Distance Kripke structure $K = (S, R)$ with log(n) propositions, the size of the fragment associated with T is no more than $|T|.(1 + (log(n))^k)$.*

Proof. Any set of states with its separator with respect to the rest of the Kripke structure contains a fragment.

This shows that the size of the state space which needs to be put at one node of the distributed computation grows only polynomially in the number of propositions in the Bounded Hamming Distance Kripke structure. It is noted that this distribution can compute the separators for only the reachable set of states in T, which can be useful if the reachable set is significantly small.

5.2 Hypercube Based Fragmentation

Now, we present another approach to distribute the state space which shows that BHDKS are very suitable for distributed computation in a hypercube grid. We prove the following results on the hypercube based partition in this section:

- A k - BHDKS with $\log n$ atomic propositions can be embedded in a l -hypercube as long as $l < \log(n)/k$.
- When embedded in a l-dimensional hypercube of distributed nodes, the size of the separator for the *core* set of states, mapped to each node in the distributed system, is no more than $\frac{l}{2^l}.n$.
- The separator for the set of *core* states associated with any node then lie only on the adjacent nodes of the hypercube. Also, there exist several states in the *core* which do not have any transitions connecting them to states outside this node.
- Thus, the size of the state space of the fragment (*core* and *border*) associated with each distributed node is given by $\frac{l+1}{2^l}.n$. Thus, the ratio of the *border* states to the *core* states is only $l < \log n$ as opposed to a ratio of $(\log n)^k$ in the polynomial fragmentation case.
- The partition ensures that only neighboring nodes in the hypercube grid need to interchange any information during the operation of the distributed model checking algorithm.

Construction of the Partitioning. We select $d = 2^l$ centers which are symmetrically placed d points, $P_1, P_2 \ldots P_d$, using the Hamming distance as a metric. It is easy to verify that these d points exist whenever $d = 2^l$ for any $l < log(n)$, where log(n) is the number of propositions.

- $000 \ldots 000 : 0$
- $000 \ldots 001 : 1$
- $000 \ldots 010 : 2$
- $000 \ldots 011 : 3$
- $\ldots\ldots\ldots$
- $\ldots\ldots\ldots$
- $111 \ldots 111 : 2^l - 1$

Using this list of binary numbers of length l, we generate the points P_i by replacing each 0 by the string made of $(log(n)/l)$ zeroes and similarly each 1 is replaced by the string made of $(log(n)/l)$ ones. The case of $l = 2$ is shown in Fig. 4.

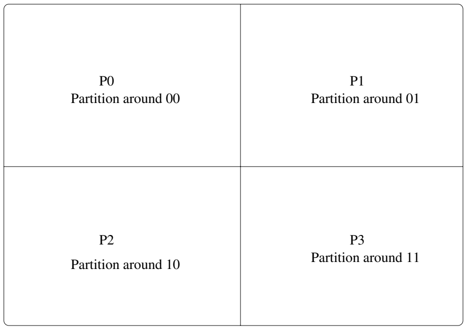


Fig. 4. The figure shows the distribution of states among 4 subsets - a 2-hypercube. The center of each subset is a P_i with the binary representation corresponding to 00,01,10 or 11 respectively in our list. For $log(n)$ = 6, these may be 000000, 000111, 111000 and 111111.

It can be observed that these 2^l centers satisfy the following:

- $\forall i \exists j$ such that $H(P_i, P_j) = log(n)/l$
- $\forall i \ \nexists j \neq i$ such that $H(P_i, P_j) < log(n)/l$.

Given a state s in the Kripke structure, $L(s)$ associates a binary label with s. We define the partition $P^{Hamming} = \{S_0^h, S_1^h, \ldots S_d^h\}$ such that $s \in S_i^h$ iff $\forall j \neq i, H(s, P_i)$ $< H(s, P_j)$, or $\exists j \neq i H(s, P_i) = H(s, P_j)$ and *generate_fair_partition*$(i, j) = i$. *generate_fair_partition* returns i or j with equal probability.These conditions ensure that the sets in the partition are disjoint as well as balanced. The *generate_fair_partition* ensures the points equidistant from more than one P_i to be distributed in a balanced manner among the nodes. Each S_i^h is associated with the i^{th} node of the distributed system as its *core* set of states. We will later add the separator of this *core* set of states with respect to the rest of the Kripke structure as the set of *border* states. We illustrate such a partition by a small example.

An Example of Hypercube Fragmentation. Consider Fig. 5 which corresponds to the case with l = 2. The sets $S1, S2, S3$ and $S4$ are formed as before by dividing the state space into 4 parts around 4 equidistant centers $0^{2p}, 0^p 1^p, 1^{2p}$ and $1^p 0^p$ respectively as

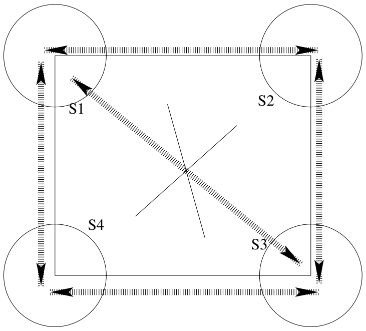


Fig. 5. For each subset around a point, it is connected only to two other sets and not to the diagonally opposite points

before. These form the core states of the fragment. We now motivate our next result by showing that for sufficiently large Kripke structures, if we take these P_is as the corners of a 2-D hypercube (square), then there can be no transitions between the distributed nodes along the diagonals.

Theorem 4. *For a BHDKS Kripke structure split uniformly around four centers $0^{2p}, 0^p1^p, 1^{2p}$ and 1^p0^p, there can be no transition along the diagonal as long as $p > k$.*

Proof. Suppose the contrary; without loss of generality, assume that there is a transition from the set around 0^{2p} to the set around 1^{2p} say from x to y. Then, $H(x, y) \leq k$. Also, by construction, $H(x, 0^{2p}) \leq p/2$ and $H(y, 1^{2p}) \leq p/2$. By triangle inequality, $H(y, 0^{2p}) + H(y, 1^{2p} \geq H(0^{2p}, 1^{2p}$ i.e., $H(y, 0^{2p}) \geq 2p - p/2$.

Again, by triangle inequality, $H(x, y) + H(x, 0^{2p}) \geq H(y, 0^{2p})$
i.e., $H(x, y) \geq H(y, 0^{2p}) - H(x, 0^{2p})$
i.e., $H(x, y) \geq 2p - p/2 - p/2$
i.e., $H(x, y) \geq p$

Thus, as long as $p > k$, there can be no transition along the diagonal. So the size of each fragment is at most 3 times the size of the core set at each node i.e., $(2+1)/2^2$ of the whole Kripke structure.

Bound on the Size of Fragment Associated with Each Distributed Node. Now consider a state space split into 2^l parts in a l-dimensional hypercube. Recall that we map each S_i^h to the node i of the hypercube, formed naturally by the binary encoding of i . We show that there can be no transition along any of the diagonals of this hypercube. The case of a 3-D cube is illustrated in Fig. 6

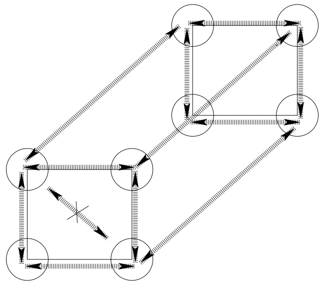


Fig. 6. A 3-D cube. There can be no transitions along any of the diagonals.

Theorem 5. *For a k-BHDKS Kripke structure with (log(n)) propositions split uniformly around 2^l centers $0^{lp}, 0^{(l-1)p}1^p, \ldots\ldots 0^p1^{(l-1)p}, 1^{lp}$ (where p = (log(n)/l)) and $p > k$, there can be no transition along any of the diagonals of this l-dimensional hypercube.*

Proof. Suppose the contrary; without loss of generality, assume that there is a transition from the set around θ to the set around δ say from x to y. Then, $H(x, y) \leq k$ Also, δ, θ are along some diagonal and not adjacent. So, $H(\theta, \delta) \geq 2p$. By construction, $H(x, \theta) \leq p/2$. and $H(y, \delta) \leq p/2$. By triangle inequality $H(y, \theta) + H(y, \delta) \geq$

$H(\theta, \delta)$ i.e., $H(y, \theta) \geq 2p - p/2$ (assuming the worst case that δ and θ are as close as possible without being neighbors in the l-dimensional hypercube) .

By triangle inequality, $H(x, y) + H(x, \theta) \geq H(y, \theta)$ i.e., $H(x, y) \geq H(y, \theta) - H(x, \theta)$

i.e., $H(x, y) \geq 2p - p/2 - p/2$

i.e., $H(x, y) \geq p$

Thus, as long as $p > k$, there can be no transition along the diagonal. Hence, there cannot be a transition along any diagonal of the l-dimensional cube.

Corollary 2. *The size of the separator of the set associated with each distributed node in the l-Dimensional hypercube is at most l times the size of the largest possible core set at each node i.e.,* $\frac{l}{2^l}.n$.

Proof. Each node in the l-dimensional hypercube has transitions only to the neighbouring nodes in the hypercube. In an l-dimensional hypercube, there are l neighbours. By construction, each neighbour has no more than $\frac{1}{2^l}.n$ *core* states. Hence, the size of the separator of a node $\leq$ sum of the size of the *core* sets associated with all the neighbouring nodes in the hypercube (since the *border* states of a node do not have any transitions to any other node)$\leq \frac{l}{2^l}.n$

Corollary 3. *The size of the fragment associated with each node in the l-Dimensional hypercube is at most* $(l + 1)$ *times the size of the largest possible core set at each node i.e.,* $\frac{(l+1)}{2^l}.n$.

Proof. A set and its separator form a fragment corresponding to that set.

Corollary 4. *The fragment associated with each node in the distributed system can be made as small as* $(\frac{2.log(n)}{k}).n^{1-1/k}$ *in the size of the k-BHDKS*

Proof. We know that the size of a fragment is bounded by $\frac{(l+1)}{2^l}.n$, as long as $l < log(n)/k$. Let us choose: $l = (log(n)/k) - 1$. Then the size of the fragment is bounded by $\frac{(l+1)}{2^l}.n$ = $(\frac{2.log(n)}{k}).n^{l-1/k}$.

At first sight one might feel that the hypercube based approach produces fragments larger than the simple subset construction presented earlier. However, the hypercube based approach trades off the size of the fragment for both a structure in the resulting partition and the greater ratio of core to fragment states in each node, which implies that less of the state space has to be copied across multiple nodes. Also, because of the small value of k and the large values of n the result is practically significant for model checking; e.g., for a 2^{20} state Kripke structure, one could partition it into 2^4 nodes each of size 2^{16} for k = 4. We remark here that the hypercube based partitioning not only provides a bound on the size of the fragment but also ensures that the communication among the nodes of the distributed computation having these fragments occurs only along the edges of the hypercube and not along its diagonals. As such, it also suggests the architecture of the distributed system and bounds the cost of the links required to connect these distributed nodes.

An important point to note is that the traditional way of recursively finding vertex separators [10] of the underlying graph to break it into smaller graphs is not feasible for the case of BHDKS. It is a well known result that the existence of $O(n^{1-\epsilon})$ vertex separator for a class of graphs implies that the class of graphs has no more than constant degree for each vertex. However, we know that the BHDKS vertex degree polylogarithmic in the number of vertices ($(\log n)^k$). As such, BHDKS do not have good vertex separators. Our hypercube based fragmentation approach avoids the construction of vertex separators and actually creates fragments $O(n^{1-\epsilon})$ in size, where $\epsilon = 1/k$, by exploiting the difference between "good" fragments and "good" vertex separators.

6 Experimental Results

We used the Cyc public database [9,11] and the CMBSLib benchmark [12] to study the performance of the hypercube based partitioning method. We took all of the thousand biochemical reactions for the Humancyc and the Ecocyc reaction pathway databases and computed an upper bound on the size of the fragment in the hypercube based fragmentation of the Kripke structure for these reaction pathways. We counted the number of edges into the core state (around the center 1111...11) using on-the-fly traversal of the state space and then used the number of edges as an upper bound on the number of *border* states. The upper bound on the size of the fragment clearly shows that the size of the fragment obtained using our worst case analysis is slightly larger than that obtained in experimental results (though of the same order).

We also took the boolean biochemical benchmark systems in CMBSLib benchmark [12] and calculated the exact size of the fragment using hypercube based partitioning method. These results indicate that the size of the fragment built using hypercube based partitioning method is of the same order as the size of the core around which it is built.

Table 1. HumanCyc 1120 atoms and EcoCyc 1313 atoms: the ratios are approximate

Sl No	Database	Radius of the Fragment	Number of States in the core	Maximum number of states in the fragment	Ratio of fragment to the core
1	HumanCyc	8	60321482688944611644	58218118459069712450424	965
2	HumanCyc	9	7459853563127158123804	7198881888172413564515156	965
3	HumanCyc	10	829547867699812679324780	800431570432559915098596984	964
4	HumanCyc	11	83785702021492624364150540	80835123199556021465682097364	964
5	HumanCyc	12	7750316948401178304236797860	7476468464640846435077137076096	964
6	EcoCyc	8	215766787047246662253	286658426283477266973032	1328
7	Ecocyc	9	31310453270114925645193	41591878653908316107275044	1328
8	Ecocyc	10	4086057570662140265020569	5427030699859074477210284960	1328
9	Ecocyc	11	484389284294462960011031017	643265834966726583668110535208	1327
10	Ecocyc	12	52597289383826851902453164625	69838841881773224220828914800104	1327

Table 2. Fragmentation results for the CMBSLib Benchmark: http://contraintes.inria.fr/CMBSlib/

Sl No	**Benchmark**	**Hamming Diameter**	Size of core	Size of border	**Fraction of core to fragment size**
1	Circadian oscillations	2	10	59	0.1449
2	Circadian oscillations	3	51	127	0.2865
3	Circadian Oscillations	4	140	149	0.48445
4	Circadian Oscillations	5	251	102	0.7110
5	Circadian Oscillations	6	333	41	0.8904
6	Cell Division Cycle	2	7	25	0.2187
7	Cell Division Cycle	3	29	48	0.3766
8	Cell Division Cycle	4	71	63	0.5299
9	Cell Division Cycle	5	126	59	0.6811
10	Cell Division Cycle	6	179	41	0.8136

It shows that the hypercube based approach performs better than our worst case bounds on real benchmarks. [3]

7 Conclusion and Future Work

In this paper, the focus has been on showing that the biochemical Kripke structures are BHDKS and are very amenable to fragmentation. In particular, it is shown that such Kripke structures can be divided into fragments as small as polynomial in the number of atomic propositions present in the Kripke structure. The hypercube algorithm tends to distribute the exponential state space in a uniform manner, and one may raise the question as to the benefit of this exercise when the reachable state space is small. A simple heuristic of merging those nodes, which can be merged into one without violating the bound on the size of the core set $(n/2^l)$, helps to handle this scenario when the distribution of the reachable state space in the hypercube is not uniform.

In particular, our explicit distributed construction of the state space partitioning assumes that there is a number close to $\log n$ which has factors that can be used as l – the dimension of the embedding hypercube. A naive recursive bi-partitioning approach which splits the entire state space around two maximally separated points in the Hamming distance space can overcome this difficulty. However, an explicit centralized construction of the state space for partitioning would defeat the purpose of the distributed model checker. Future directions of research include the development of distributed algorithms to distribute the reachable state space onto a hypercube. Also, the choice of the hypercube in which the system is embedded and the assignment of different embeddings onto the same hypercube (by changing the order of propositions in the state space) needs to studied. In short, BHDKS are very suitable for bounded model checking. Hamming Distance Kripke structures are also very suitable for Bounded Model Checking.

[3] The result of the benchmark differs from that of the public databases because we abstract all the reactions in the public databases for nondeterministic vanishing of reactants after the reactions to illustrate a worst case scenario.

Acknowledgement

The author acknowledges the support of the Indian Academy of Sciences Summer Fellowship 2004 and grateful to Prof. D. Sarkar for the encouragement and support for the work. The second author thanks ITPAR (Indo-Italian Promotion of Advanced Research) for his visit to University of Trento during which some initial exploration of such applications was done.

References

1. Luboš Brim, Karen Yorav, and Jitka Žídková. Assumption-based distribution of CTL model checking. *International Journal on Software Tools for Technology Transfer (STTT)*, 7(1):61–73, February 2005.
2. Luboš Brim, Jitka Žídková, and Karen Yorav. Using assumptions to distribute CTL model checking. *Electr. Notes Theor. Comput. Sci.*, 68(4), 2002.
3. J. R. Burch, E. M. Clark, K. L. McMillan, D. L. Dill, and L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. In John C. Mitchell, editor, *Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science*, pages 428–439, Philadelphia, PA, June 1990. IEEE Computer Society Press.
4. Nathalie Chabrier and François Fages. Symbolic model checking of biochemical networks. In Corrado Priami, editor, *CMSB*, volume 2602 of *Lecture Notes in Computer Science*, pages 149–162. Springer, 2003.
5. Nathalie Chabrier-Rivier, Marc Chiaverini, Vincent Danos, François Fages, and Vincent Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25–44, September 2004.
6. Nathalie Chabrier-Rivier, François Fages, and Sylvain Soliman. The biochemical abstract machine biocham. In Vincent Danos and Vincent Schachter, editors, *CMSB*, volume 3082 of *Lecture Notes in Computer Science*, pages 172–191. Springer, 2004.
7. Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. The MIT Press, Cambridge, Massachusetts, 1999.
8. Matthew B. Dwyer, editor. *Model Checking Software, 8th International SPIN Workshop, Toronto, Canada, May 19-20, 2001, Proceedings*, volume 2057 of *Lecture Notes in Computer Science*. Springer, 2001.
9. Peter D. Karp, Monica Riley, Suzanne M. Paley, and Alida Pellegrini-Toole. Ecocyc: an encyclopedia of escherichia coli genes and metabolism. *Nucleic Acids Research*, 24(1):32–39, 1996.
10. George Karypis and Navaratnasothie Selvakkumaran. Multi-objective hypergraph partitioning algorithms for cut and maximum subdomain degree minimization. In *ICCAD '03: Proceedings of the 2003 IEEE/ACM international conference on Computer-aided design*, page 726, Washington, DC, USA, August 18 2003. IEEE Computer Society.
11. Cynthia J. Krieger, Peifen Zhang, Lukas A. Mueller, Alfred Wang, Suzanne M. Paley, Martha Arnaud, John Pick, Seung Yon Rhee, and Peter D. Karp. Metacyc: a multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Research*, 32(Database-Issue):438–442, 2004.
12. Sylvain Soliman and François Fages. Cmbslib: A library for comparing formalisms and models of biological systems. In Vincent Danos and Vincent Schachter, editors, *CMSB*, volume 3082 of *Lecture Notes in Computer Science*, pages 231–235. Springer, 2004.