A Fast, Scalable Mutual Exclusion Algorithm*

Jae-Heon Yang
Department of Mathematics and Computer Science
Mills College
Oakland, California 94613

James H. Anderson
Department of Computer Science
The University of North Carolina at Chapel Hill
Chapel Hill, North Carolina 27599-3175

October 1993
Revised November 1994

Abstract

This paper is concerned with synchronization under read/write atomicity in shared memory multiprocessors. We present a new algorithm for N-process mutual exclusion that requires only read and write operations and that has $O(\log N)$ time complexity, where "time" is measured by counting remote memory references. The time complexity of this algorithm is better than that of all prior solutions to the mutual exclusion problem that are based upon atomic read and write instructions; in fact, the time complexity of most prior solutions is unbounded. Performance studies are presented that show that our mutual exclusion algorithm exhibits scalable performance under heavy contention. In fact, its performance rivals that of the fastest queue-based spin locks based on strong primitives such as compare-and-swap and fetch-and-add. We also present a modified version of our algorithm that generates only $O(1)$ memory references in the absence of contention.

Keywords: Fast mutual exclusion, local spinning, mutual exclusion, read/write atomicity, scalability, synchronization primitives, time complexity.

CR Categories: D.4.1, D.4.2, F.3.1

*Preliminary version was presented at the Twelfth Annual ACM Symposium on Principles of Distributed Computing, Ithaca, New York, August 1993. Work supported, in part, by NSF Contracts CCR-9109497 and CCR-9216421 and by the Center for Excellence in Space Data and Information Sciences (CESDIS).

1 Introduction

The mutual exclusion problem is a paradigm for resolving conflicting accesses to shared resources and has been studied for many years, dating back to the seminal paper of Dijkstra [6]. In this problem, each of a set of processes repeatedly executes a program fragment known as its "critical section". Before and after executing its critical section, a process executes two other program fragments, its "entry section" and "exit section", respectively. The entry and exit sections must be designed so that at most one process executes its critical section at any time, and so that each process in its entry section eventually executes its critical section. The former is known as the *mutual exclusion* property, and the latter as the *starvation-freedom* property. In some variants of the problem, starvation-freedom is replaced by the weaker requirement of *livelock-freedom*: if some process is in its entry section, then some process eventually executes its critical section.

Most early solutions to the mutual exclusion problem required only minimal hardware support, specifically atomic read and write instructions. Although of theoretical importance, most such algorithms were judged to be impractical from a performance standpoint, leading to the development of solutions requiring stronger hardware support such as read-modify-write operations. The poor performance of early read/write algorithms stems partially from two factors. First, such algorithms are not scalable, i.e., performance degrades dramatically as the number of contending processes increases. Second, even in the absence of contention, such algorithms require a process contending for its critical section to execute many operations.

The second of these two problems has been subsequently addressed, specifically by Lamport in [9], where a read/write algorithm is given that requires only a constant number of operations per critical section acquisition in the absence of contention. Following the title of Lamport's paper, such algorithms have come to be known simply as "fast mutual exclusion algorithms". This designation is somewhat of a misnomer, as such algorithms are not necessarily fast in the *presence* of contention. In fact, the problem of designing a scalable mutual exclusion algorithm that requires only read/write atomicity, and that exhibits good performance under contention, has remained open. In this paper, we present such an algorithm.

Many mutual exclusion algorithms that exhibit poor scalability do so, in part, because they require processes to busy-wait on shared variables that are remotely-accessible, i.e., that require a traversal of the global interconnect between processors and shared memory when accessed. Recently, several queue-based mutual exclusion algorithms based on primitives that are stronger than reads and writes have been proposed in which this type of busy-waiting is avoided [3, 7, 11]. These algorithms are based on the idea of "local spinning". In particular, processes busy-wait only on *locally accessible* shared variables that do not require a traversal of the global interconnect when accessed. Performance studies presented in [3, 7, 11] have established the importance of local spinning in ensuring good performance under heavy contention.

Although the notion of a *locally* accessible *shared* variable may at first seem counterintuitive, there are two architectural paradigms that support it. In particular, on distributed shared memory machines, a shared variable can be made locally accessible by storing it in a local portion of shared memory, and on cache-coherent machines, programs can be structured so that when a process busy-waits on a shared variable, that variable migrates to a local cache line. Note that, because distributed shared memory machines require static allocation of shared variables to processors, an algorithm that locally spins on such a machine will also locally spin on cache-coherent machines. The reverse is not necessarily true. For this reason, we adopt the stricter distributed shared memory model when distinguishing between local and remote shared memory accesses in the time complexity computations given in this paper. Alternative definitions of local and remote memory accesses based on cache-coherence are briefly considered in Section 6.

In a recent paper [2], Anderson presented a mutual exclusion algorithm that uses only local spins and that requires only atomic read and write operations. In his algorithm, each of N processes executes $O(N)$ remote operations to enter its critical section whether there is contention or not. All other previously published mutual exclusion algorithms that are based on atomic reads and writes employ global busy-waiting and hence induce an unbounded number of remote operations under heavy contention. Most such algorithms also require $O(N)$ remote operations in the absence of contention. Some exceptions to the latter include the algorithm given by Kessels in [8] and the previously mentioned one given by Lamport in [9]. Kessels' algorithm generates $O(\log N)$ remote operations in the absence of contention, while Lamport's generates $O(1)$. A variant of Lamport's algorithm was recently presented by Styer in [14]. Although Styer claims that his algorithm is more scalable than Lamport's, in terms of time complexity, they are actually very similar: both generate unbounded remote operations under heavy contention and $O(1)$ operations in the absence of contention. Styer's claims of scalability are predicated upon complexity calculations that ignore operations performed within busy-waiting constructs. Because the processes in his algorithm busy-wait on remote variables, such complexity calculations do not give a true indication of scalability. Another recent algorithm of interest is one given by Michael and Scott in [12]. Although this algorithm generates $O(N)$ remote memory references in the presence of contention and $O(1)$ in the absence of contention, it requires both full- and half-word reads and writes to memory, which is a level of hardware support more powerful than ordinary read/write atomicity.

In this paper, we present a new mutual exclusion algorithm that requires only atomic reads and writes and in which all spins are local. Our algorithm induces $O(\log N)$ remote operations under any amount of contention, and thus is an improvement over the algorithm given by Anderson in [2]. We also present a modified version of this algorithm, called the "fast-path" algorithm, that requires only $O(1)$ remote operations in the absence of contention. Unfortunately, in the fast-path algorithm, worst-case complexity rises to $O(N)$. However, we argue that this $O(N)$ behavior is rare, occurring only when transiting from a period of high contention to a period of low contention. Under high contention, the fast-path algorithm induces only $O(\log N)$ remote operations. It is worth noting that our algorithm and its variation are starvation-free, whereas some of the aforementioned algorithms are not.

The results of this paper suggest some interesting questions regarding the inherent complexity of synchronization. The queue-based algorithms in [3, 7, 11] require $O(1)$ remote operations, whether there is contention or not (actually the algorithms in [3, 7] have $O(1)$ complexity only if coherent caches are provided), whereas our algorithm requires $O(\log N)$ remote operations. It is natural to ask whether this gap is due to a fundamental weakness of atomic reads and writes, or whether there is a mutual exclusion algorithm based on such operations that has $O(1)$ complexity (which seems doubtful). A significant step towards answering this question was recently presented by us in [16]. This new result involves programs with limited "write-contention". The *write-contention* of a concurrent program is the number of processes that may be simultaneously enabled to write the same shared variable. It is shown in [16] that for any N-process mutual exclusion algorithm with write-contention K, there is an execution involving only one process in which that process executes at least $\Omega(\log_K N)$ remote memory references in its entry section. This lower bound holds assuming any of a variety of synchronization primitives, including read, write, test-and-set, load-and-store, compare-and-swap, and fetch-and-add. Our mutual exclusion algorithm is contained within the class of algorithms based on such primitives that have constant write-contention. Hence, our algorithm is asymptotically optimal for this class. The execution that establishes the $\Omega(\log_K N)$ lower bound in [16] involves only one process, implying that fast mutual exclusion requires high write-contention (i.e., K must approach N). All fast mutual exclusion algorithms for N processes presented to date (including the fast-path algorithm presented in this paper) have write-contention N.

Despite the significance of the lower bound proved in [16], we still do not know whether the apparent gap between mutual exclusion using reads and writes and mutual exclusion using stronger primitives is fundamental. If the lower bound for mutual exclusion under read/write atomicity turns out to be $\Omega(\log N)$, then an additional question arises: namely, is it possible to develop a mutual exclusion algorithm under read/write atomicity that requires $O(1)$ remote operations in the absence of contention and $O(\log N)$ remote operations in the presence of contention? As mentioned above, our fast-path algorithm *almost* achieves such bounds.

The rest of the paper is organized as follows. In Section 2, we present our model of concurrent programs. The above-mentioned mutual exclusion algorithm is then presented in Section 3. In Section 4, we consider the fast-path algorithm discussed above. In Section 5, we present results from performance studies conducted on the BBN TC2000 and Sequent Symmetry multiprocessors. These studies indicate that our mutual exclusion algorithm exhibits scalable performance under heavy contention. In fact, the performance of our algorithm rivals (and sometimes beats) that of the queue-based algorithms mentioned above. We end the paper with concluding remarks in Section 6. Correctness proofs for our algorithms are given in an appendix.

2 Definitions

In this section, we present our model of concurrent programs and define the relations used in reasoning about such programs. A *concurrent program* consists of a set of processes and a set of variables. A *process* is a sequential program consisting of labeled statements. Each *variable* of a concurrent program is either private or shared. A *private variable* is defined only within the scope of a single process, whereas a *shared variable* is defined globally and may be accessed by more than one process. Each process of a concurrent program has a special private variable called its *program counter*: the statement with label k in process p may be executed only when the value of the program counter of p equals k. For an example of the syntax we employ for programs, see Figure 1.

A program's semantics is defined by its set of "fair histories". The definition of a fair history, which is given below, formalizes the requirement that each statement of a program is subject to weak fairness. Before giving the definition of a fair history, we introduce a number of other concepts; all of these definitions apply to a given concurrent program.

A *state* is an assignment of values to the variables of the program. One or more states are designated as *initial states*. If state u can be reached from state t via the execution of statement s, then we say that s is *enabled* at state t and we write $t \xrightarrow{s} u$. If statement s is not enabled at state t, then we say that s is *disabled* at t. A *history* is a sequence $t_0 \xrightarrow{s_0} t_1 \xrightarrow{s_1} \cdots$, where t_0 is an initial state. A history may be either finite or infinite; in the former case, it is required that no statement be enabled at the last state of the history. A history is *fair* if it is finite or if it is infinite and each statement is either disabled at infinitely many states of the history or is infinitely often executed in the history. Note that this fairness requirement implies that each continuously enabled statement is eventually executed. Unless otherwise noted, we henceforth assume that all histories are fair.

With regard to complexity, we assume that each shared variable is *local* to at most one process and is *remote* to all other processes. This assumption is reflective of a distributed shared memory model. We refer to a statement execution as an *operation*. An operation is *remote* if it accesses remote variables, and is *local* otherwise.

Following [5], we define safety properties using *unless* assertions and progress properties using *leads-to* assertions. Consider two predicates P and Q over the variables of a program. The assertion P *unless* Q holds iff for any pair of consecutive states in any history of the program, if $P \wedge \neg Q$ holds in the first state, then $P \vee Q$ holds in the second. If predicate P is initially true and if P *unless false* holds, then predicate

P is said to be an *invariant*. We say that predicate P *leads-to* predicate Q, denoted $P \mapsto Q$, iff for each history $t_0 \xrightarrow{s_0} t_1 \xrightarrow{s_1} \cdots$ of the program, if P is true at some state t_i, then Q is true at some state t_j where $j \geq i$.

3 Mutual Exclusion Algorithm

In this section, we present our mutual exclusion algorithm. We begin by stating more precisely the conditions that must be satisfied by such an algorithm. In the mutual exclusion problem, there are N processes, each of which has the following structure.

while *true* **do**
    Noncritical Section;
    Entry Section;
    Critical Section;
    Exit Section
od

It is assumed that each process begins execution in its noncritical section. It is further assumed that each critical section execution terminates. By contrast, a process is allowed to halt in its noncritical section. No variable appearing in any entry or exit section may be referred to in any noncritical or critical section (except, of course, program counters). A program that solves this problem is required to satisfy the mutual exclusion and starvation-freedom properties, given earlier in Section 1. We also require that each process in its exit section eventually enters its noncritical section; this requirement is trivially satisfied by our solution (and most others), so we will not consider it further.

As in [8], we first solve the mutual exclusion problem for two processes, and then apply our two-process solution in a binary arbitration tree to get an N-process solution. Our algorithm is depicted in Figure 1. In order to obtain an N-process algorithm, we associate each link in a binary tree with an entry section and an exit section of the two-process solution. In other words, the entry and exit sections associated with the two links connecting a given node to its subtree constitute a two-process mutual exclusion algorithm. This is depicted in Figure 2. Initially, all processes start at the leaves of the tree. To enter its critical section, a process is required to traverse a path from its leaf up to the root, executing the entry section of each link on this path. Upon exiting its critical section, a process traverses this path in reverse, this time executing the exit section of each link.

The first mutual exclusion algorithm that used a binary arbitration tree is that given by Peterson and Fischer in [13]. However, in this algorithm, a process that reaches the top of a left subtree checks all leaves of the corresponding right subtree, resulting in $O(N)$ remote memory references outside of busy-waiting loops. The structure of our arbitration tree is inherited from Kessels's solution in [8], which induces only $O(\log N)$ memory references outside of busy-waiting loops.

In Figure 1, our two-process mutual exclusion algorithm (from statement 4 to statement 19) is placed "on top" of two $(N/2)$-process versions of our algorithm. In this figure, *ENTRY_LEFT* and *EXIT_LEFT*, denoting the entry and exit sections of the $(N/2)$-process version of our algorithm, enforce mutual exclusion in the left subtree. Similarly, *ENTRY_RIGHT* and *EXIT_RIGHT* are used in the right subtree. We depict the algorithm in this way in order to hide the details relating to how variables are named in the arbitration tree.

The **always** section [5] in Figure 1 is used to define the expression $side(i)$. Informally, $side(i) = 0$ iff process i is a process from the left subtree, and $side(i) = 1$ iff process i is a process from the right subtree.

```
shared var C : array[0, 1] of −1..N − 1;
           P : array[0..N − 1] of 0..2;
           T : 0..N − 1
initially  C[0] = −1 ∧ C[1] = −1 ∧ (∀i :: P[i] = 0)
always     (∀i : i < ⌊N/2⌋ :: side(i) = 0) ∧ (∀i : i ≥ ⌊N/2⌋ :: side(i) = 1)

process i
private var rival : −1..N − 1;

while true do
   0: Noncritical Section;
   1: if i < ⌊N/2⌋ then
   2:    ENTRY_LEFT
      else
   3:    ENTRY_RIGHT
      fi;
   4: C[side(i)] := i;     /* side(i) = 0 ∨ side(i) = 1 */
   5: T := i;
   6: P[i] := 0;
   7: rival := C[1 − side(i)];
   8: if rival ≠ −1 then
   9:    if T = i then
  10:       if P[rival] = 0 then
  11:          P[rival] := 1 fi;
  12:       while P[i] = 0 do /* null */ od;
  13:       if T = i then
  14:         while P[i] ≤ 1 do /* null */ od fi
         fi
      fi;
  15: Critical Section;
  16: C[side(i)] := −1;
  17: rival := T;
  18: if rival ≠ i then
  19:    P[rival] := 2 fi;
  20: if i < ⌊N/2⌋ then
  21:    EXIT_LEFT
      else
  22:    EXIT_RIGHT
      fi
od
```

Figure 1: N-process mutual exclusion algorithm.

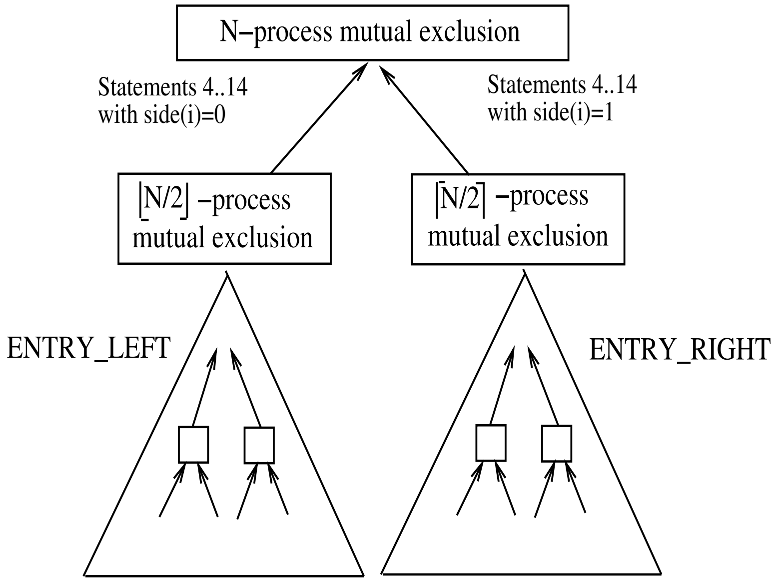

Figure 2: Entry section of our N-process mutual exclusion algorithm.

The expression $1 - side(i)$ is used to identify the C-variable of process i's competitor. We assume that process identifiers range over $\{0..N-1\}$.

The algorithm employs shared variables, $C[0]$, $C[1]$, and T, in addition to a spin location $P[i]$ for each process i. Variable $C[0]$ is used by a process from the left subtree to inform processes in the right subtree of its intent to enter its critical section. Observe that $C[0] = i \neq -1$ holds while a process from the left subtree i executes its statements 5 through 16, and $C[0] = -1$ holds when no process from the left subtree executes these statements. Variable $C[1]$ is used similarly. Variable T is used as a tie-breaker in the event that a process from the left subtree and a process from the right subtree attempt to enter their critical sections at the same time. The algorithm ensures that the two processes enter their critical sections according to the order in which they update T. Variable $P[i]$ ranges over $\{0, 1, 2\}$ and is used by process i whenever it needs to busy-wait. Note that $P[i]$ is waited on only by process i, and thus can be stored in a memory location that is locally accessible to process i (in which case all spins are local).

Loosely speaking, the algorithm works as follows. Let process i be a process from the left subtree. When process i wants to enter its critical section, it informs processes in the right subtree of its intention by establishing $C[0] = i$. Then, process i assigns its identifier i to the tie-breaker variable T, and initializes its spin location $P[i]$. If no process in the right subtree has shown interest in entering its critical section, in other words, if $C[1] = -1$ holds when i executes statement 7, then process i proceeds directly to its critical section. Otherwise, i reads the tie-breaker variable T. If $T = j \neq i$, (i.e., if process j from the right subtree is competing against process i,) then i can enter its critical section, as the algorithm prohibits process j from entering its critical section when $C[0] = i \ \wedge\ T = j$ holds (recall that ties are broken in favor of the first process to update T). If $T = i$ holds, then either process j executed statement 5 before process i, or process j has executed statement 4 but not statement 5. In the first case, i should wait until j exits its critical section, whereas, in the second case, i should be able to proceed to its critical section. This ambiguity is resolved by having process i execute statements 10 through 14. Statements 10 and 11 are executed by process i to release process j in the event that it is waiting for i to update the tie-breaker variable (i.e., j is busy-waiting at statement 12). Statements 12 through 14 are executed by i to determine which process updated the tie-breaker variable first. Note that $P[i] \geq 1$ implies that j has already updated the tie-breaker, and $P[i] = 2$ implies that j has finished its critical section. To handle these two cases, process i first waits until $P[i] \geq 1$ (i.e., until j has updated the tie-breaker), re-examines T to see which process updated T last, and finally, if necessary, waits until $P[i] = 2$ (i.e., until process j finishes its critical section).

After executing its critical section, process i informs process j that it is finished by establishing $C[i] = -1$. If $T = j$, in which case process j is waiting to enter its critical section, then process i updates $P[j]$ in order to terminate j's busy-waiting loop.

With regard to complexity, note that if variable $P[i]$ is local to process i in the two process algorithm, then process i executes a constant number of remote operations in its two-process entry and exit sections. It follows that, in an N-process algorithm that employs a balanced binary tree, each process executes $O(\log N)$ remote operations in its (N-process) entry and exit sections.

4 Fast Mutual Exclusion in the Absence of Contention

As discussed in Section 1, most early mutual exclusion algorithms based on read/write atomicity are neither fast in the absence of contention, nor able to cope with high contention. Because Lamport's fast mutual exclusion algorithm induces $O(1)$ remote operations in the absence of contention, and our mutual exclusion algorithm requires $O(\log N)$ remote operations given any level of contention, it seems reasonable to expect a solution to exist that induces $O(1)$ remote operations when contention is absent, and $O(\log N)$ remote operations when contention is high.

The fast-path algorithm given in Figure 3 almost achieves that goal. The basic idea, illustrated in Figure 4, is to combine Lamport's fast mutual exclusion algorithm and our algorithm, specifically by placing an extra two-process version of our algorithm "on top" of the arbitration tree. The "left" entry section of this extra two-process program is executed by a process if that process detects no contention. The "right" entry section of this extra program is executed by the winning process from the arbitration tree. A process will compete within the arbitration tree (as before) if it detects any contention. As seen in Figure 3, the scheme used to detect contention is similar to that used in Lamport's algorithm. In this figure, we use $ENTRY_k$ and $EXIT_k$ to denote the entry and exit sections of the k-process version of our algorithm. A correctness proof for the algorithm of Figure 3 is given in the Appendix.

It should be clear that, in the absence of contention, a process enters its critical section after executing $O(1)$ remote operations. Also, in the presence of contention, a process enters its critical section after executing $O(\log N)$ remote operations. However, when a period of contention ends, N remote operations might be required in order to re-open the fast entry section — see the **while** loop at line 22 in Figure 3. Nonetheless, performance studies we have done show that, under high contention, these statements are rarely executed. (Under low contention, they are obviously never executed.) For example, out of the 100,000 critical section executions in one experiment, these N statements were performed after only 55 critical section executions in the four-process case, and after only one in the eight- and sixteen-process cases.

In the absence of contention, our algorithm generates about twice as many remote memory operations as Lamport's. However, under high contention, our algorithm is clearly superior, as Lamport's induces an unbounded number of remote operations. Also, our fast-path algorithm ensures starvation-freedom, whereas Lamport's algorithm does not.

5 Performance Results

To compare the scalability of our mutual exclusion algorithm with that of other algorithms, we conducted a number of experiments on the BBN TC2000 and Sequent Symmetry multiprocessors. Results from some of

```
shared var B : array[0..N − 1] of boolean;
           X, Y : −1..N − 1;
           Z : boolean
initially  Y = −1 ∧ Z = false ∧ (∀i :: B[i] = false)

process i
private var flag : boolean;
            n : 1..N

while true do
  TOP:   Noncritical Section;
         X := i;
         if Y ≠ −1 then goto SLOW fi;
         Y := i;
         if X ≠ i then goto SLOW fi;
         B[i] := true;
         if Z then goto SLOW fi;
         if Y ≠ i then goto SLOW fi;
         ENTRY_2;                          /* Two-Process Entry Section */
           Critical Section;
         EXIT_2;                           /* Two-Process Exit Section */
         Y := −1;
         B[i] := false;
         goto TOP;

  SLOW:  ENTRY_N;                          /* Arbitration Tree */
           ENTRY_2;                        /* Two-Process Entry Section */
             Critical Section;
             B[i] := false;
             if X = i then
               Z := true;
               flag := true;
               n := 0;
               while (n < N) do
                 if B[n] then flag := false fi;
                 n := n + 1
               od;
               if flag then Y := −1 fi;
               Z := false
             fi;
           EXIT_2;                         /* Two-Process Exit Section */
         EXIT_N                            /* Arbitration Tree */
od
```

Figure 3: Fast, scalable mutual exclusion algorithm.

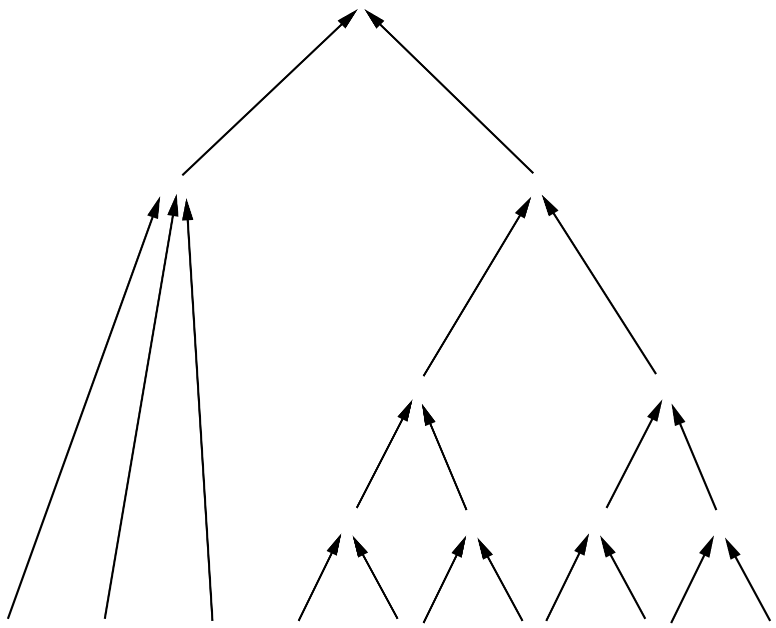

Figure 4: Providing a fast path.

these experiments are presented in this section.

BBN TC2000

The BBN TC2000 is a distributed shared memory multiprocessor, each node of which contains a processor and a memory unit. The nodes are connected via a multi-stage interconnection network, known as the Butterfly switch. Each access to a remote memory location (i.e., one that requires a traversal of the interconnection network) takes about 2 microseconds, whereas each local reference takes about 0.6 microseconds. Each node's processor, a Motorola 88100, provides an atomic fetch-and-store instruction called `xmem`. Other strong primitives such as compare-and-swap and fetch-and-add are provided using the TC2000 hardware locking protocol [4]. The TC2000 has cache memory, but does not provide a cache coherence mechanism.

We tested seven mutual exclusion algorithms on the TC2000: a simple test-and-set algorithm; the queue-based algorithm using compare-and-swap given by Mellor-Crummey and Scott in [11]; the queue-based algorithm using fetch-and-add given by T. Anderson in [3]; the fast mutual exclusion algorithm given by Lamport in [9]; the tree-based algorithm given by Styer in [14]; the tree-based algorithm given by Peterson and Fischer in [13]; and the mutual exclusion algorithm described in Section 3. Performance results obtained by running these seven algorithms on the TC2000 are summarized in Figure 5. Each point (x, y) in each graph represents the average time y for one critical section execution with x competing processors. The timing results summarized in the graph were obtained by averaging over 10^5 critical section executions. The critical section consists of a read and an increment of shared counter. Results obtained using larger critical sections, which for brevity are not presented here, show similar performance to that depicted in Figure 5. The timing results presented include the execution time of critical sections.

The performance of the test-and-set algorithm is given by the graph labeled T&S, Mellor-Crummey and Scott's algorithm by the graph labeled MCS, T. Anderson's algorithm by the graph labeled AND, Lamport's

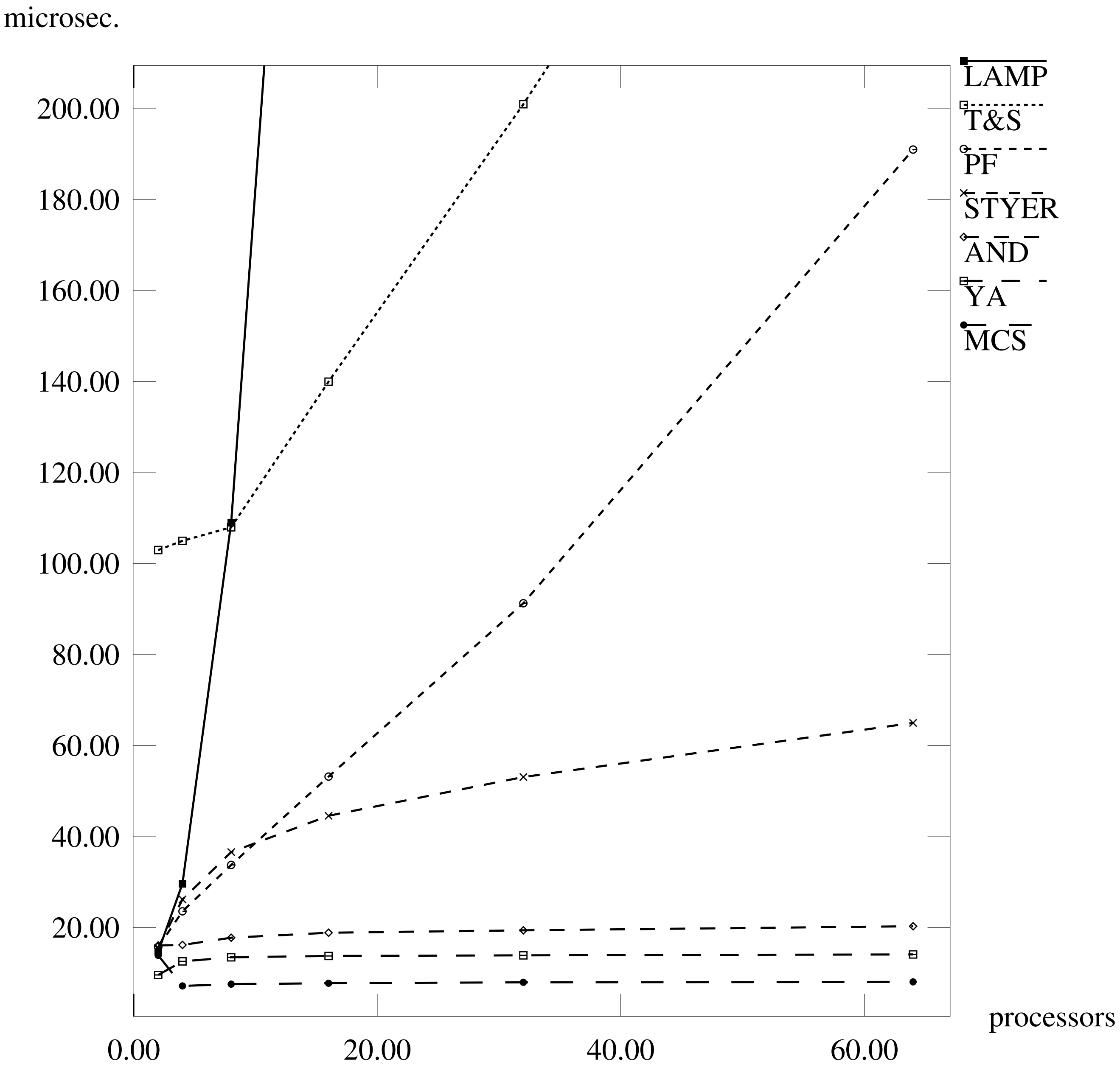

Figure 5: TC2000, small critical section.

algorithm by the graph labeled LAMP, Styer's algorithm by the graph labeled STYER, Peterson and Fischer's algorithm by the graph labeled PF, and our algorithm by the graph labeled YA. On the TC2000, the MCS algorithm was the best overall performer of the alternatives considered here. The graph depicted for the MCS algorithm is mostly flat, except at the point for two processors. This anomaly at two processors coincides with results reported by Mellor-Crummey and Scott on the Sequent Symmetry, and was attributed by them to the lack of a compare-and-swap instruction on the Symmetry [11]. As our implementation of their algorithm did employ compare-and-swap, we have not found a satisfying explanation for this behavior on the TC2000.

T. Anderson's algorithm requires only local spinning when implemented on a machine with coherent caches. On the Symmetry, where each process can spin on its own coherent cache, Anderson's algorithm outperforms the MCS algorithm. However, on the TC2000, which does not support coherent caching, Anderson's algorithm requires remote spinning, slowing its performance.

The simple T&S algorithm exhibited poor scalability. The average execution time for the 64 processor case, which is not depicted in Figure 5, is about 330 microseconds. Where there is a possibility of contention among a large number of processors, it should be avoided, or used with good backoff scheme [1].

Three algorithms based on atomic reads and writes — Lamport's, Peterson and Fischer's, and Styer's — also showed poor scalability. In particular, the performance of Lamport's algorithm degrades dramatically as the number of contenders increases. The average execution time for the 64 processor case, which is not depicted in Figure 5, is about 4000 microseconds. The performance of Styer's algorithm, which is better than that of Lamport's, is due to the tree structure employed. Styer's algorithm generates $O(\log N)$ remote operations outside of busy-waiting loops. Even though Peterson and Fischer's algorithm is also tree-based, it induces $O(N)$ remote operations outside of busy-waiting loops, which results in poorer scalability.

Our mutual exclusion algorithm shows performance that is comparable to that of T. Anderson's and Mellor-Crummey and Scott's algorithms. Its good scalability emphasizes the importance of local spinning. The difference seen between our mutual exclusion algorithm and the MCS algorithm is explained by the amount of global traffic generated by each algorithm. The MCS algorithm generates $O(1)$ remote operations per critical section execution, whereas ours generates $O(\log N)$. The global traffic of the other five algorithms is unbounded, as each employs global spinning. The performance of T. Anderson's algorithm is far better than that of the simple test-and-set algorithm. Because the processes in Anderson's algorithm spin globally on the TC2000, this might be interpreted as a counterexample to our belief that minimizing remote operations is important for good scalability. However, Mellor-Crummey and Scott reported in [11] that Anderson's algorithm produced far fewer remote operations than the test-and-set algorithm.

Sequent Symmetry

Performance results of experiments on the Sequent Symmetry are summarized in Figure 5. The Sequent Symmetry is a shared memory multiprocessor whose processor and memory nodes are interconnected via a shared bus. A processor node consists of an Intel 80386 and a 64 Kbyte, two-way set-associative cache. Cache coherence is maintained by a snoopy protocol. The Symmetry provides an atomic fetch-and-store instruction. Because other strong primitives are not provided, we used a version of Mellor-Crummey and Scott's algorithm that is implemented with fetch-and-store and that does not ensure starvation-freedom [11]. Fetch-and-add, which is used in T. Anderson's algorithm, was simulated by a test-and-set algorithm with randomized backoff, as Anderson did in [3].

The experiments on the Symmetry show similar results to that for the TC2000. However, on the Symmetry, T. Anderson's algorithm has the best overall performance, mainly because the availability of coherent

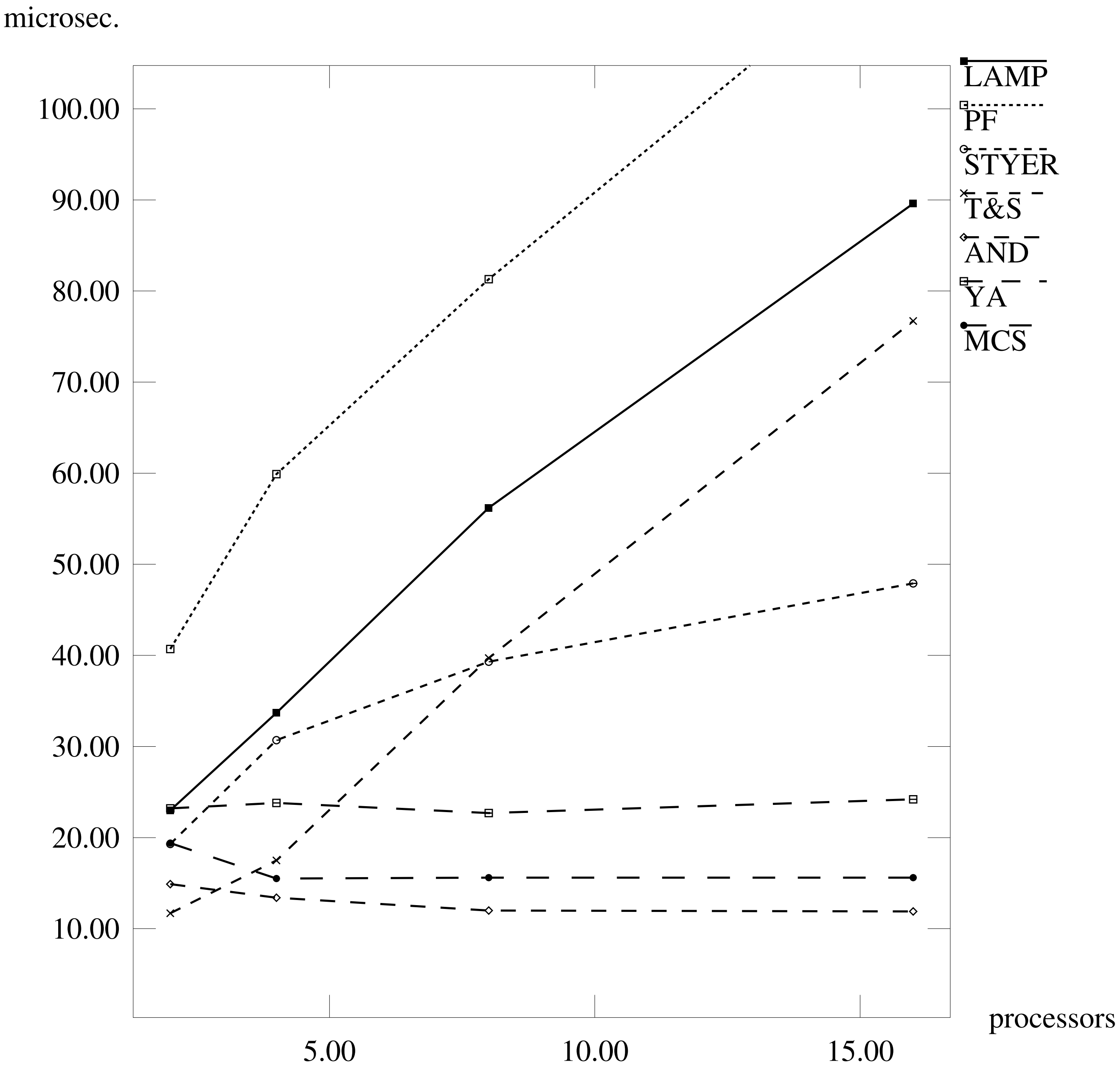

Figure 6: Symmetry, small critical section.

caches makes all spins in his algorithm local. The performance of Lamport's algorithm on the Symmetry is far better than that on the TC2000. This seems partly due to the fact that his algorithm is not starvation-free. Specifically, when a process enters its critical section, it can keep all needed variables in its own cache and repeatedly enter its critical section, without yielding to the other processes. In one of our tests for the two-process case, one process executed 50,000 critical sections during a period of time in which the other process executed only 120 critical sections.

Dependence on coherent caching for efficient synchronization [3, 7] is questionable, as many caching schemes do not cache shared writable data. Our solution neither requires a coherent cache for efficient implementation nor any strong primitives. An efficient implementation of our algorithm requires only that each processor has some part of shared memory that is locally accessible, and that read and write operations are atomic. We consider these to be minimal hardware requirements for efficient synchronization. It is worth noting that, without fetch-and-add and compare-and-swap primitives, T. Anderson's algorithm and Mellor-Crummey and Scott's algorithm are not starvation-free.

6 Concluding Remarks

We have presented a scalable mutual exclusion algorithm for shared memory multiprocessors that does not require any hardware support other than atomic read and write operations. Our algorithm has better worst-case time complexity than any previously published mutual exclusion algorithm based on read/write atomicity, requiring $O(\log N)$ remote operations under any amount of contention. We have also presented an extension of our algorithm for fast mutual exclusion in the absence of contention that generates $O(1)$ remote operations in the presence of contention and $O(N)$ in the absence of contention. In addition, we have suggested several open questions concerning the inherent complexity of multiprocessor synchronization.

In the complexity calculations given in this paper, the distinction between remote and local operations is based upon a static assignment of shared variables to processes. Other definitions, which incorporate specific architectural details of systems, are also possible. For example, for programs intended for machines with coherent caching, it might be appropriate to consider a read of a shared variable x by a process p to be local if x has not been written by another process since p's most recent access of x. However, because of the many parameters that go into defining a cache-coherence protocol, such definitions can be problematic. We therefore choose to leave this as a subject for further study.

A natural approach to measuring the time complexity of concurrent programs would be to simply count the total number of operations executed. However, a straightforward application of such an approach does not provide any insight into the behavior of mutual exclusion algorithms under heavy contention. In particular, in any algorithm in which processes busy-wait, the number of operations needed for one process to get to its critical section is unbounded. In order to serve as a measure of time complexity, a measure should be both intuitive and easy to compute. In sequential programming, the usual measure of time complexity, which is obtained by simply counting operations, satisfies these criteria. By contrast, there has been much disagreement on how time complexity should be measured in concurrent programs, and a complexity measure satisfying these criteria has yet to be adopted. We believe that an appropriate time complexity measure for concurrent algorithms is one based on the number of remote memory references. As seen in this paper, such a measure can be used to make meaningful distinctions concerning the performance of concurrent programs.

Because minimizing the number of remote memory references in shared-memory algorithms is analogous to minimizing the number of messages in distributed message-passing algorithms, one might be tempted to compare our algorithm with distributed mutual exclusion algorithms. However, in distributed message-passing algorithms, processes respond to messages even when they are in their noncritical sections. By contrast, in shared-memory algorithms (including ours), a process may halt while in its noncritical section, without affecting other processes.

Acknowledgement: We would like to thank Howard Gobioff for helping with the performance studies given in Section 5. We would also like to acknowledge Argonne National Laboratories and Lawrence Livermore National Laboratories for providing us with access to the machines used in these studies. We are particularly grateful to Terry Gaasterland at Argonne, and Tammy Welcome and Brent Gorda at Lawrence Livermore for their help. We also thank the referees for their helpful remarks, and Leslie Lamport for sharing his hierarchical proof macros with us.

Appendix: Correctness Proof

In this appendix, we present the correctness proofs for the algorithms presented in Sections 3 and 4. As depicted in Figure 1, our N-process solution is obtained by associating an instance of the two process solution (statements 4 to 19) with each internal node of a binary arbitration tree. By induction on the depth of the tree, it can be shown that the mutual exclusion and starvation-freedom properties hold for the N-process program provided they hold for the two-process program. In the rest of this section, we focus on the proof that the mutual exclusion and starvation-freedom properties hold for the two-process program.

Our assertions and proofs are stated in a hierarchical structure proposed by Lamport [10]. A sequence of conjunctions (disjunctions) is denoted by putting each conjunct (disjunct), preceded by a $\wedge$ ($\vee$), in a separate line. In proofs in Lamport's style, an assertion is proven by a series of high-level steps. Each of these steps may be proven by a list of steps at a lower level. Every step is preceded by $<$ *level* $>$ *number* for reference in subsequent steps. We begin by presenting definitions and notational conventions that will be used in the remainder of the paper.

Notational Conventions: Unless specified otherwise, we assume that i, p, and q range over $\{0..N-1\}$. We denote statement number k of process i as $k.i$. Let S be a subset of the statement labels in process i. Then, $i@\{S\}$ holds iff the program counter for process i equals some value in S. The expression $(\mathbf{N}i :: P(i))$ denotes the number of i's satisfying predicate $P(i)$. The following is a list of symbols we will use ordered by increasing binding power: $\equiv$, $\mapsto$, $\Rightarrow$, $\vee$, $\wedge$, $(=, \neq, >, <, \geq, \leq)$, $(+, -)$, $'$, $\neg$, $(., @)$. The symbols enclosed in parentheses have the same priority. We sometimes use parentheses to override this binding rule. $\Box$

Mutual Exclusion

To facilitate the presentation, we define the following predicate.

$$LME \equiv (\forall i :: (\mathbf{N}p : side(p) = side(i) :: p@\{4..19\}) \leq 1)$$

Informally, LME holds if the $(N/2)$-process solutions guarantee mutual exclusion in the subtrees. In other words, the predicate LME serves as the induction hypothesis in our proof of mutual exclusion. We henceforth assume that LME is an invariant.

We next prove that the mutual exclusion property holds. In particular, we prove that if at most one process from the left subtree and at most one process from the right subtree compete, then at most one process may enter its critical section at a time. This property can be stated as follows.

$$LME \;\Rightarrow\; (\mathbf{N}i :: i@\{15\}) \leq 1$$

Next, we prove several assertions that are needed to establish $((\mathbf{N}i :: i@\{15\}) \leq 1)$.

(I1) $\quad C[1 - side(i)] = p \;\wedge\; p \neq -1$
$\quad \Rightarrow side(p) = 1 - side(i)$

$\langle 1 \rangle 1.$ $Init \;\Rightarrow\; C[1 - side(i)] = -1$
$\quad \Rightarrow\;$ I1

$\langle 1 \rangle 2.$ I1 $\Rightarrow$ I1$'$

ASSUME: I1 $\wedge\; \neg$I1$'$

PROVE: $\perp$

$\langle 2 \rangle 1.$ $side(p) \neq 1 - side(i)$ — by $\neg$I1$'$

$\langle 2 \rangle 2.$ $4.p \;\wedge\; side(p) = 1 - side(i)$ — by I1 $\wedge\; \neg$I1$'$

$\langle 2 \rangle 3.$ $\perp$ — by $\langle 2 \rangle 1$ and $\langle 2 \rangle 2$

$\square$

(I2) $\quad C[side(i)] = i$
$\quad \Rightarrow i@\{5..16\}$

$\langle 1 \rangle 1.$ $Init \;\Rightarrow\; C[side(i)] = -1$
$\quad \Rightarrow\;$ I2

$\langle 1 \rangle 2.$ I2 $\Rightarrow$ I2$'$

ASSUME: I2 $\wedge\; \neg$I2$'$

PROVE: $\perp$

$\langle 2 \rangle 1.$ $C'[side(i)] = i$ — by $\neg$I2$'$

$\langle 2 \rangle 2.$ $\neg(i@\{5..16\})'$ — by $\neg$I2$'$

$\langle 2 \rangle 3.$ $\vee\; 4.i$
$\vee\; 16.i$ — by I2 $\wedge\; \neg$I2$'$

CASE: $4.i$

$\langle 3 \rangle 1.$ $(i@\{5\})'$ — by $4.i$

$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 2$

CASE: $16.i$

$\langle 3 \rangle 1.$ $C'[side(i)] = -1$ — by $16.i$

$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 1$

$\square$

(I3) $\quad i@\{5..16\}$
$\quad \Rightarrow C[side(i)] = i$

$\langle 1\rangle 1.\ \mathit{Init} \ \Rightarrow\ i@\{0\}$
$\qquad\qquad \Rightarrow\ \mathrm{I3}$
$\langle 1\rangle 2.\ \mathrm{I3} \ \Rightarrow\ \mathrm{I3}'$
ASSUME: $\mathrm{I3} \ \wedge\ \neg\mathrm{I3}'$
PROVE: $\perp$
$\langle 2\rangle 1.\ (i@\{5..16\})'$ — by $\neg\mathrm{I3}'$
$\langle 2\rangle 2.\ C'[side(i)] \neq i$ — by $\neg\mathrm{I3}'$
$\langle 2\rangle 3.\ \vee\ 4.i$ — by $\mathrm{I3} \ \wedge\ \neg\mathrm{I3}'$
$\qquad \vee\ (4.p \ \vee\ 16.p) \ \wedge\ side(p) = side(i)$
CASE: $4.i$
$\langle 3\rangle 1.\ C'[side(i)] = i$ — by $4.i$
$\langle 3\rangle 2.\ \perp$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 2$
CASE: $(4.p \ \vee\ 16.p) \ \wedge\ side(p) = side(i)$
CASE: $p = i$
$\langle 4\rangle 1.\ C'[side(i)] = i \ \vee\ (i@\{17\})'$ — by $4.i \ \vee\ 16.i$
$\langle 4\rangle 2.\ \perp$ — by $\langle 4\rangle 1$, $\langle 2\rangle 1$, and $\langle 2\rangle 2$
CASE: $p \neq i$
$\langle 4\rangle 1.\ (p@\{5, 17\})'$ — by $4.p \ \vee\ 16.p$
$\langle 4\rangle 2.\ \neg(i@\{4..19\})'$ — by LME, $\langle 4\rangle 1$, and $side(p) = side(i)$
$\langle 4\rangle 3.\ \perp$ — by $\langle 4\rangle 2$ and $\langle 2\rangle 1$

$\square$

$$(\mathrm{I4}) \quad \begin{array}{l} \wedge\ p@\{6..19\} \\ \wedge\ side(q) = 1 - side(p) \\ \wedge\ q@\{6..19\} \\ \Rightarrow\ \vee\ T = p \\ \quad\ \ \vee\ T = q \end{array}$$

$\langle 1\rangle 1.\ \mathit{Init} \ \Rightarrow\ p@\{0\}$
$\qquad\qquad \Rightarrow\ \mathrm{I4}$
$\langle 1\rangle 2.\ \mathrm{I4} \ \Rightarrow\ \mathrm{I4}'$
ASSUME: $\mathrm{I4} \ \wedge\ \neg\mathrm{I4}'$
PROVE: $\perp$
$\langle 2\rangle 1.\ (p@\{6..19\})'$ — by $\neg\mathrm{I4}'$
$\langle 2\rangle 2.\ side(q) = 1 - side(p)$ — by $\neg\mathrm{I4}'$
$\langle 2\rangle 3.\ (q@\{6..19\})'$ — by $\neg\mathrm{I4}'$
$\langle 2\rangle 4.\ T' \neq p$ — by $\neg\mathrm{I4}'$
$\langle 2\rangle 5.\ T' \neq q$ — by $\neg\mathrm{I4}'$
$\langle 2\rangle 6.\ 5.i$ — by $\mathrm{I4} \ \wedge\ \neg\mathrm{I4}'$
CASE: $i = p$
$\langle 3\rangle 1.\ T' = p$ — by $5.p$
$\langle 3\rangle 2.\ \perp$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 4$

CASE: $i \neq p \ \wedge \ side(i) = side(p)$

$\langle 3 \rangle 1.$ $(i@\{6\})'$ — by $5.i$

$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$, $\langle 2 \rangle 1$, $side(i) = side(p)$, and LME

CASE: $i = q$

$\langle 3 \rangle 1.$ $T' = q$ — by $5.q$

$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 5$

CASE: $i \neq q \ \wedge \ side(i) = side(q)$

$\langle 3 \rangle 1.$ $(i@\{6\})'$ — by $5.i$

$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$, $\langle 2 \rangle 3$, $side(i) = side(q)$, and LME

$\square$

$$
\begin{array}{ll}
\text{(I5)} & \wedge\ p@\{8..11\} \\
& \wedge\ C[1 - side(p)] = q \ \wedge \ q \neq -1 \\
& \Rightarrow \vee\ p.rival = q \\
& \qquad \vee\ T = q \\
& \qquad \vee\ q@\{5\}
\end{array}
$$

$\langle 1 \rangle 1.$ $Init \ \Rightarrow \ p@\{0\}$
$\qquad \Rightarrow \ \text{I5}$

$\langle 1 \rangle 2.$ $\text{I5} \ \Rightarrow \ \text{I5}'$

ASSUME: $\text{I5} \ \wedge \ \neg \text{I5}'$

PROVE: $\perp$

$\langle 2 \rangle 1.$ $(p@\{8..11\})'$ — by $\neg \text{I5}'$

$\langle 2 \rangle 2.$ $C'[1 - side(p)] = q \ \wedge \ q \neq -1$ — by $\neg \text{I5}'$

$\langle 2 \rangle 3.$ $p.rival' \neq q$ — by $\neg \text{I5}'$

$\langle 2 \rangle 4.$ $T' \neq q$ — by $\neg \text{I5}'$

$\langle 2 \rangle 5.$ $\neg (q@\{5\})'$ — by $\neg \text{I5}'$

$\langle 2 \rangle 6.$ $side(q) = 1 - side(p)$ — by I1 and $\langle 2 \rangle 2$

$\langle 2 \rangle 7.$ $\vee\ 4.q$ — by $\text{I5} \ \wedge \ \neg \text{I5}'$
$\vee\ 5.i$
$\vee\ 7.p$
$\vee\ 17.p$

CASE: $4.q$

$\langle 3 \rangle 1.$ $(q@\{5\})'$ — by $4.q$

$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 5$

CASE: $5.i$

CASE: $i = p$

$\langle 4 \rangle 1.$ $(p@\{6\})'$ — by $5.p$

$\langle 4 \rangle 2.$ $\perp$ — by $\langle 4 \rangle 1$ and $\langle 2 \rangle 1$

CASE: $i \neq p \ \wedge \ side(i) = side(p)$

$\langle 4 \rangle 1.$ $(i@\{6\})'$ — by $5.i$

$\langle 4 \rangle 2.$ $\perp$ — by $\langle 4 \rangle 1$, $\langle 2 \rangle 1$, $side(i) = side(p)$, and LME

CASE: $i = q$

$\langle 4\rangle 1.\ T' = q$ by $5.q$

$\langle 4\rangle 2.\ \bot$ by $\langle 4\rangle 1$ and $\langle 2\rangle 4$

CASE: $i \neq q \ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.\ (i@\{6\})'$ by $5.i$

$\langle 4\rangle 2.\ C'[side(i)] = i$ by I3 and $\langle 4\rangle 1$

$\langle 4\rangle 3.\ C'[1 - side(p)] = i$ by $\langle 4\rangle 2$, $\langle 2\rangle 6$, and $side(i) = side(q)$

$\langle 4\rangle 4.\ \bot$ by $\langle 4\rangle 3$, $\langle 2\rangle 2$, and $i \neq q$

CASE: $7.p$

$\langle 3\rangle 1.\ p.rival' = q$ by $7.p$ and $\langle 2\rangle 2$

$\langle 3\rangle 2.\ \bot$ by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $17.p$

$\langle 3\rangle 1.\ (p@\{18\})'$ by $17.p$

$\langle 3\rangle 2.\ \bot$ by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

□

(I6) $\wedge\ p@\{7..15\}$
$\wedge\ P[p] = 2$
$\wedge\ C[1 - side(p)] = q \ \wedge\ q \neq -1$
$\Rightarrow \vee\ T = q$
$\vee\ q@\{5\}$

$\langle 1\rangle 1.\ Init \ \Rightarrow\ p@\{0\}$
$\Rightarrow\ \text{I6}$

$\langle 1\rangle 2.\ \text{I6} \ \Rightarrow\ \text{I6}'$

ASSUME: $\text{I6} \ \wedge\ \neg\text{I6}'$

PROVE: $\bot$

$\langle 2\rangle 1.\ (p@\{7..15\})'$ by $\neg\text{I6}'$

$\langle 2\rangle 2.\ P'[p] = 2$ by $\neg\text{I6}'$

$\langle 2\rangle 3.\ C'[1 - side(p)] = q \ \wedge\ q \neq -1$ by $\neg\text{I6}'$

$\langle 2\rangle 4.\ T' \neq q$ by $\neg\text{I6}'$

$\langle 2\rangle 5.\ \neg(q@\{5\})'$ by $\neg\text{I6}'$

$\langle 2\rangle 6.\ side(q) = 1 - side(p)$ by I1 and $\langle 2\rangle 3$

$\langle 2\rangle 7.\ (q@\{5..16\})'$ by I2, $\langle 2\rangle 3$, and $\langle 2\rangle 6$

$\langle 2\rangle 8.\ \vee\ 4.q$
$\vee\ 5.i$
$\vee\ 6.p$
$\vee\ 19.i$

CASE: $4.q$

$\langle 3\rangle 1.\ (q@\{5\})'$ by $4.q$

$\langle 3\rangle 2.\ \bot$ by $\langle 3\rangle 1$ and $\langle 2\rangle 5$

CASE: $5.i$

CASE: $i = p$

$\langle 4\rangle 1.\ (p@\{6\})'$ by $5.p$

$\langle 4\rangle 2.\ \bot$ by $\langle 4\rangle 1$ and $\langle 2\rangle 1$

CASE: $i \neq p \ \wedge\ side(i) = side(p)$

$\langle 4\rangle 1.\ (i@\{6\})'$ — by $5.i$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$, $\langle 2\rangle 1$, $side(i) = side(p)$, and LME

Case: $i = q$

$\langle 4\rangle 1.\ T' = q$ — by $5.q$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$ and $\langle 2\rangle 4$

Case: $i \neq q \ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.\ (i@\{6\})'$ — by $5.i$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$, $\langle 2\rangle 7$, $side(i) = side(q)$, and LME

Case: $6.p$

$\langle 3\rangle 1.\ P'[p] = 0$ — by $6.p$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 2$

Case: $19.i$

Case: $i = p$

$\langle 4\rangle 1.\ (p@\{20\})'$ — by $19.p$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$ and $\langle 2\rangle 1$

Case: $i \neq p \ \wedge\ side(i) = side(p)$

$\langle 4\rangle 1.\ i@\{19\} \ \wedge\ p@\{7..15\}$ — by $19.i$ and $\langle 2\rangle 1$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$, $side(i) = side(p)$, and LME

Case: $i = q$

$\langle 4\rangle 1.\ (q@\{20\})'$ — by $19.q$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$ and $\langle 2\rangle 7$

Case: $i \neq q \ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.\ i@\{19\} \ \wedge\ q@\{5..16\}$ — by $19.i$ and $\langle 2\rangle 7$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$, $side(i) = side(p)$, and LME

□

The next invariant, (I7), is the crux of our mutual exclusion proof. The mutual exclusion property is then established in (I8).

$$
\text{(I7)}\quad \begin{array}{l} \wedge\ p@\{15..19\} \\ \wedge\ C[1 - side(p)] = q \ \wedge\ q \neq -1 \\ \Rightarrow \vee\ T = q \\ \quad\ \ \vee\ q@\{5\} \end{array}
$$

$\langle 1\rangle 1.\ Init \ \Rightarrow\ p@\{0\}$

$\qquad\quad \Rightarrow\ \text{I7}$

$\langle 1\rangle 2.\ \text{I7} \ \Rightarrow\ \text{I7}'$

Assume: $\text{I7} \ \wedge\ \neg\text{I7}'$

Prove: $\bot$

$\langle 2\rangle 1.\ (p@\{15..19\})'$ — by $\neg\text{I7}'$

$\langle 2\rangle 2.\ C'[1 - side(p)] = q \ \wedge\ q \neq -1$ — by $\neg\text{I7}'$

$\langle 2\rangle 3.\ T' \neq q$ — by $\neg\text{I7}'$

$\langle 2\rangle 4.$ $\neg(q@\{5\})'$ by $\neg$I7$'$

$\langle 2\rangle 5.$ $side(q) = 1 - side(p)$ by I1 and $\langle 2\rangle 2$

$\langle 2\rangle 6.$ $(q@\{5..16\})'$ by I2, $\langle 2\rangle 2$, and $\langle 2\rangle 5$

$\langle 2\rangle 7.$ $\vee\ 4.q$ by I7 $\wedge$ $\neg$I7$'$

$\vee\ 5.i$

$\vee\ 8.p \ \wedge\ p.rival = -1$

$\vee\ (9.p \ \vee\ 13.p) \ \wedge\ T \neq p$

$\vee\ 14.p \ \wedge\ P[p] = 2$

Case: $4.q$

$\langle 3\rangle 1.$ $(q@\{5\})'$ by $4.q$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 4$

Case: $5.i$

Case: $i = p$

$\langle 4\rangle 1.$ $(p@\{6\})'$ by $5.p$

$\langle 4\rangle 2.$ $\perp$ by $\langle 4\rangle 1$ and $\langle 2\rangle 1$

Case: $i \neq p \ \wedge\ side(i) = side(p)$

$\langle 4\rangle 1.$ $(i@\{6\})'$ by $5.i$

$\langle 4\rangle 2.$ $\perp$ by $\langle 4\rangle 1$, $\langle 2\rangle 1$, $side(i) = side(p)$, and LME

Case: $i = q$

$\langle 4\rangle 1.$ $T' = q$ by $5.q$

$\langle 4\rangle 2.$ $\perp$ by $\langle 4\rangle 1$ and $\langle 2\rangle 3$

Case: $i \neq q \ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.$ $(i@\{6\})'$ by $5.i$

$\langle 4\rangle 2.$ $\perp$ by $\langle 4\rangle 1$, $\langle 2\rangle 6$, $side(i) = side(q)$, and LME

Case: $8.p \ \wedge\ p.rival = -1$

$\langle 3\rangle 1.$ $p@\{8\} \ \wedge\ p.rival = -1$ by $8.p$

$\langle 3\rangle 2.$ $C[1 - side(p)] = q \ \wedge\ q \neq -1$ by $8.p$ and $\langle 2\rangle 2$

$\langle 3\rangle 3.$ $T \neq q$ by $8.p$ and $\langle 2\rangle 3$

$\langle 3\rangle 4.$ $\neg q@\{5\}$ by $8.p$ and $\langle 2\rangle 4$

$\langle 3\rangle 5.$ $\vee\ \neg(C[1 - side(p)] = q \ \wedge\ q \neq -1)$ by I5 and $\langle 3\rangle 1$

$\vee\ T = q$

$\vee\ q@\{5\}$

$\langle 3\rangle 6.$ $\perp$ by $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 3\rangle 4$, and $\langle 3\rangle 5$

Case: $(9.p \ \vee\ 13.p) \ \wedge\ T \neq p$

$\langle 3\rangle 1.$ $(p@\{15\})' \ \wedge\ T' \neq p$ by $9.p \ \vee\ 13.p$ and $T \neq p$

$\langle 3\rangle 2.$ $T' = q$ by I4, $\langle 3\rangle 1$, $\langle 2\rangle 5$, and $\langle 2\rangle 6$

$\langle 3\rangle 3.$ $\perp$ by $\langle 3\rangle 2$ and $\langle 2\rangle 3$

Case: $14.p \ \wedge\ P[p] = 2$

$\langle 3\rangle 1.$ $(p@\{15\})' \ \wedge\ P'[p] = 2$ by $14.p$

$\langle 3\rangle 2.$ $T' = q \ \vee\ (q@\{5\})'$ by I6, $\langle 3\rangle 1$, and $\langle 2\rangle 2$

$\langle 3\rangle 3.$ $\perp$ by $\langle 3\rangle 2$, $\langle 2\rangle 3$, and $\langle 2\rangle 4$

$\square$

(I8) $(\mathbf{N} i :: i@\{15\}) \ \leq\ 1$

ASSUME: $\neg$I8

PROVE: $\perp$

$\langle 1\rangle 1.\ \wedge\ p@\{15\}\ \wedge\ p \neq -1$ for some p, q, by $\neg$I8
$\quad \wedge\ q@\{15\}\ \wedge\ q \neq -1$
$\quad \wedge\ p \neq q$

$\langle 1\rangle 2.\ side(q) = 1 - side(p)$ by $\langle 1\rangle 1$ and LME

$\langle 1\rangle 3.\ C[side(p)] = p\ \wedge\ C[side(q)] = q$ by I3 and $\langle 1\rangle 1$

$\langle 1\rangle 4.\ C[1 - side(q)] = p\ \wedge\ C[1 - side(p)] = q$ by $\langle 1\rangle 2$ and $\langle 1\rangle 3$

$\langle 1\rangle 5.\ T = p\ \wedge\ T = q$ by I7, $\langle 1\rangle 1$, and $\langle 1\rangle 4$

$\langle 1\rangle 6.\ \perp$ by $\langle 1\rangle 1$ and $\langle 1\rangle 5$

$\square$

Progress

Next, we prove that our two-process algorithm is starvation-free. We first prove a number of invariants that are needed to establish starvation-freedom.

(I9) $\quad C[side(i)] = p\ \wedge\ p \neq -1$
$\quad \Rightarrow side(p) = side(i)$

$\langle 1\rangle 1.\ Init\ \Rightarrow\ C[side(i)] = -1$
$\qquad \Rightarrow\ \text{I9}$

$\langle 1\rangle 2.$ I9 $\Rightarrow$ I9$'$

ASSUME: I9 $\wedge$ $\neg$I9$'$

PROVE: $\perp$

$\langle 2\rangle 1.\ side(p) \neq side(i)$ by $\neg$I9$'$

$\langle 2\rangle 2.\ 4.p\ \wedge\ side(p) = side(i)$ by I9 $\wedge$ $\neg$I9$'$

$\langle 2\rangle 3.\ \perp$ by $\langle 2\rangle 1$ and $\langle 2\rangle 2$

$\square$

(I10) $\quad i@\{8..11\}$
$\quad \Rightarrow \vee\ i.rival = -1$
$\qquad \vee\ side(i.rival) = 1 - side(i)$

$\langle 1\rangle 1.\ Init\ \Rightarrow\ i@\{0\}$
$\qquad \Rightarrow\ \text{I10}$

$\langle 1\rangle 2.$ I10 $\Rightarrow$ I10$'$

ASSUME: I10 $\wedge$ $\neg$I10$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(i@\{8..11\})'$ by $\neg$I10$'$

$\langle 2\rangle 2.$ $i.rival' \neq -1$ by $\neg$I10$'$

$\langle 2\rangle 3.$ $side(i.rival') \neq 1 - side(i)$ by $\neg$I10$'$

$\langle 2\rangle 4.$ $\vee\ 7.i$ by I10 $\wedge$ $\neg$I10$'$

$\vee\ 17.i$

CASE: $7.i$

$\langle 3\rangle 1.$ $i.rival' = C'[1 - side(i)]$ by $7.i$

$\langle 3\rangle 2.$ $side(i.rival') = 1 - side(i)$ by I1, $\langle 3\rangle 1$, and $\langle 2\rangle 2$

$\langle 3\rangle 3.$ $\perp$ by $\langle 3\rangle 2$ and $\langle 2\rangle 3$

CASE: $17.i$

$\langle 3\rangle 1.$ $(i@\{18\})'$ by $17.i$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

$\square$

(I11) $(\forall j : side(j) = side(i) :: j@\{0..4, 17..22\})$
$\Rightarrow C[side(i)] = -1$

$\langle 1\rangle 1.$ $Init \ \Rightarrow\ C[side(i)] = -1$
$\Rightarrow$ I11

$\langle 1\rangle 2.$ I11 $\Rightarrow$ I11$'$

ASSUME: I11 $\wedge$ $\neg$I11$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(\forall j : side(j) = side(i) :: (j@\{0..4, 17..22\})')$ by $\neg$I11$'$

$\langle 2\rangle 2.$ LET: $C'[side(i)] = p \ \wedge\ p \neq -1$ by $\neg$I11$'$

$\langle 2\rangle 3.$ $side(p) = side(i)$ by I9 and $\langle 2\rangle 2$

$\langle 2\rangle 4.$ $(p@\{5..16\})'$ by I2 and $\langle 2\rangle 2$

$\langle 2\rangle 5.$ $\perp$ by $\langle 2\rangle 4$, $\langle 2\rangle 3$, and $\langle 2\rangle 1$

$\square$

(I12) $\wedge\ p@\{8\} \ \wedge\ p.rival \neq -1 \ \vee\ p@\{9..14\}$
$\wedge\ side(q) = 1 - side(p)$
$\wedge\ q@\{18, 19\}$
$\Rightarrow \vee\ q.rival = p$
$\vee\ q.rival = q$

$\langle 1\rangle 1.$ $Init \ \Rightarrow\ p@\{0\}$
$\Rightarrow$ I12

$\langle 1\rangle 2.$ I12 $\Rightarrow$ I12$'$

ASSUME: I12 $\wedge$ $\neg$I12$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(p@\{8\})' \ \wedge\ p.rival' \neq -1 \ \vee\ (p@\{9..14\})'$ — by $\neg$I12$'$

$\langle 2\rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg$I12$'$

$\langle 2\rangle 3.$ $(q@\{18, 19\})'$ — by $\neg$I12$'$

$\langle 2\rangle 4.$ $q.rival' \neq p$ — by $\neg$I12$'$

$\langle 2\rangle 5.$ $q.rival' \neq q$ — by $\neg$I12$'$

$\langle 2\rangle 6.$ $C'[side(q)] = -1$ — by I11, $\langle 2\rangle 3$, and *LME*

$\langle 2\rangle 7.$ $T' = p \ \vee\ T' = q$ — by I4, $\langle 2\rangle 1$, $\langle 2\rangle 2$, and $\langle 2\rangle 3$

$\langle 2\rangle 8.$ $\vee\ 7.p$ — by I12 $\wedge$ $\neg$I12$'$
$\vee\ 7.q$
$\vee\ 17.q$

CASE: $7.p$

$\langle 3\rangle 1.$ $C'[1 - side(p)] = -1$ — by $7.p$, $\langle 2\rangle 6$, and $\langle 2\rangle 2$

$\langle 3\rangle 2.$ $(p@\{8\})' \ \wedge\ p.rival' = -1$ — by $7.p$ and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\perp$ — by $\langle 3\rangle 2$ and $\langle 2\rangle 1$

CASE: $7.q$

$\langle 3\rangle 1.$ $(q@\{8\})'$ — by $7.q$

$\langle 3\rangle 2.$ $\perp$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $17.q$

$\langle 3\rangle 1.$ $q.rival' = p \ \vee\ q.rival' = q$ — by $17.q$ and $\langle 2\rangle 7$

$\langle 3\rangle 2.$ $\perp$ — by $\langle 3\rangle 1$, $\langle 2\rangle 4$, and $\langle 2\rangle 5$

$\square$

(I13) $\wedge\ p@\{8\} \ \wedge\ p.rival \neq -1 \ \vee\ p@\{9..14\}$
$\wedge\ side(q) = 1 - side(p)$
$\wedge\ q@\{19\}$
$\Rightarrow q.rival = p$

$\langle 1\rangle 1.$ $Init \ \Rightarrow\ p@\{0\}$
$\Rightarrow$ I13

$\langle 1\rangle 2.$ I13 $\Rightarrow$ I13$'$

ASSUME: I13 $\wedge$ $\neg$I13$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(p@\{8\})' \ \wedge\ p.rival' = -1 \ \vee\ (p@\{9..14\})'$ — by $\neg$I13$'$

$\langle 2\rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg$I13$'$

$\langle 2\rangle 3.$ $(q@\{19\})'$ — by $\neg$I13$'$

$\langle 2\rangle 4.$ $q.rival' \neq p$ — by $\neg$I13$'$

$\langle 2\rangle 5.$ $C'[side(q)] = -1$ — by I11, $\langle 2\rangle 3$, and *LME*

$\langle 2\rangle 6.$ $\vee\ 7.p$ — by I13 $\wedge$ $\neg$I13$'$
$\vee\ 7.q \ \vee\ 17.q$
$\vee\ 18.q \ \wedge\ q.rival \neq q$

CASE: $7.p$

$\langle 3\rangle 1.$ $C'[1 - side(p)] = -1$ — by $7.p$, $\langle 2\rangle 5$, and $\langle 2\rangle 2$

$\langle 3\rangle 2.$ $(p@\{8\})' \ \wedge\ p.rival' = -1$ — by $7.p$ and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\bot$ — by $\langle 3\rangle 2$ and $\langle 2\rangle 1$

CASE: $7.q \ \vee\ 17.q$

$\langle 3\rangle 1.$ $\neg(q@\{19\})'$ — by $7.q \ \vee\ 17.q$

$\langle 3\rangle 2.$ $\bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $18.q \ \wedge\ q.rival \neq q$

$\langle 3\rangle 1.$ $q.rival' = p$ — by I12, $\langle 2\rangle 1$, $\langle 2\rangle 2$, $\langle 2\rangle 3$, and $q.rival' \neq q$

$\langle 3\rangle 2.$ $\bot$ — by $\langle 3\rangle 1$, and $\langle 2\rangle 4$

$\square$

$$\text{(I14)}\quad \begin{array}{l} \wedge\ p@\{8\} \ \wedge\ p.rival \neq -1 \ \vee\ p@\{9..14\} \\ \wedge\ side(q) = 1 - side(p) \\ \wedge\ q@\{18\} \\ \wedge\ q.rival = q \\ \Rightarrow T = q \end{array}$$

$\langle 1\rangle 1.$ $Init \ \Rightarrow\ p@\{0\}$
$\quad\quad \Rightarrow\ \text{I14}$

$\langle 1\rangle 2.$ $\text{I14} \ \Rightarrow\ \text{I14}'$

ASSUME: $\text{I14} \ \wedge\ \neg\text{I14}'$

PROVE: $\bot$

$\langle 2\rangle 1.$ $(p@\{8\})' \ \wedge\ p.rival' \neq -1 \ \vee\ (p@\{9..14\})'$ — by $\neg\text{I14}'$

$\langle 2\rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg\text{I14}'$

$\langle 2\rangle 3.$ $(q@\{18\})'$ — by $\neg\text{I14}'$

$\langle 2\rangle 4.$ $q.rival' = q$ — by $\neg\text{I14}'$

$\langle 2\rangle 5.$ $T' \neq q$ — by $\neg\text{I14}'$

$\langle 2\rangle 6.$ $C'[side(q)] = -1$ — by I11, $\langle 2\rangle 3$, and LME

$\langle 2\rangle 7.$ $\vee\ 5.i$
$\quad \vee\ 7.p$
$\quad \vee\ 7.q$
$\quad \vee\ 17.q$ — by $\text{I14} \ \wedge\ \neg\text{I14}'$

CASE: $5.i$

CASE: $i = p$

$\langle 4\rangle 1.$ $(p@\{6\})'$ — by $5.p$

$\langle 4\rangle 2.$ $\bot$ — by $\langle 4\rangle 1$ and $\langle 2\rangle 1$

CASE: $i \neq p \ \wedge\ side(i) = side(p)$

$\langle 4\rangle 1.$ $(i@\{6\})'$ — by $5.i$

$\langle 4\rangle 2.$ $\bot$ — by $\langle 4\rangle 1$, $\langle 2\rangle 1$, $side(i) = side(p)$, and LME

CASE: $i = q$

$\langle 4\rangle 1.$ $(q@\{6\})'$ — by $5.q$

$\langle 4\rangle 2.$ $\bot$ — by $\langle 4\rangle 1$ and $\langle 2\rangle 3$

CASE: $i \neq q \ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.$ $(i@\{6\})'$ — by $5.i$

$\langle 4\rangle 2.$ $\bot$ — by $\langle 4\rangle 1$, $\langle 2\rangle 3$, $side(i) = side(q)$, and LME

CASE: $7.p$

$\langle 3\rangle 1.$ $C'[1 - side(p)] = -1$ by $7.p$, $\langle 2\rangle 6$, and $\langle 2\rangle 2$

$\langle 3\rangle 2.$ $(p@\{8\})' \wedge p.rival' = -1$ by $7.p$ and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\perp$ by $\langle 3\rangle 2$ and $\langle 2\rangle 1$

CASE: $7.q$

$\langle 3\rangle 1.$ $(q@\{8\})'$ by $7.q$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $17.q$

$\langle 3\rangle 1.$ $q.rival' = T'$ by $17.q$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$, $\langle 2\rangle 4$, and $\langle 2\rangle 5$

$\square$

(I15) $\wedge\, p@\{8\} \wedge p.rival \neq -1 \vee p@\{9..14\}$
$\wedge\, (\forall i : side(i) = 1 - side(p) :: i@\{0..4, 20..22\})$
$\Rightarrow P[p] = 2$

$\langle 1\rangle 1.$ $Init \Rightarrow p@\{0\}$
$\Rightarrow$ I15

$\langle 1\rangle 2.$ I15 $\Rightarrow$ I15$'$

ASSUME: I15 $\wedge \neg$I15$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(p@\{8\})' \wedge p.rival' \neq -1 \vee (p@\{9..14\})'$ by $\neg$I15$'$

$\langle 2\rangle 2.$ $(\forall i : side(i) = 1 - side(p) :: (i@\{0..4, 20..22\})')$ by $\neg$I15$'$

$\langle 2\rangle 3.$ $P'[p] \neq 2$ by $\neg$I15$'$

$\langle 2\rangle 4.$ $C'[side(p)] = p$ by I3 and $\langle 2\rangle 1$

$\langle 2\rangle 5.$ $C'[1 - side(p)] = -1$ by I11 and $\langle 2\rangle 2$

$\langle 2\rangle 6.$ $\vee\, 6.p$ by I15 $\wedge \neg$I15$'$
$\vee\, 7.p$
$\vee\, 11.i \wedge i.rival = p$
$\vee\, 18.i \wedge side(i) = 1 - side(p) \wedge i.rival = i$
$\vee\, 19.i \wedge side(i) = 1 - side(p)$

CASE: $6.p$

$\langle 3\rangle 1.$ $(p@\{7\})'$ by $6.p$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $7.p$

$\langle 3\rangle 1.$ $(p@\{8\})' \wedge p.rival' = -1$ by $7.p$ and $\langle 2\rangle 5$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $11.i \wedge i.rival = p$

$\langle 3\rangle 1.$ $(i@\{12\})' \wedge i.rival' = p$ by $11.i$

$\langle 3\rangle 2.$ $side(i) = 1 - side(p)$ by I10 and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\perp$ by $\langle 3\rangle 1$, $\langle 3\rangle 2$, and $\langle 2\rangle 2$

CASE: $18.i \wedge side(i) = 1 - side(p) \wedge i.rival = i$

$\langle 3\rangle 1.$ $i@\{18\}$ by $18.i$

$\langle 3\rangle 2.$ $p@\{8\} \;\wedge\; p.rival = -1 \;\vee\; p@\{9..14\}$ — by 18.i and $\langle 2\rangle 1$

$\langle 3\rangle 3.$ $T = i$ — by I14, $\langle 3\rangle 2$, $side(i) = 1 - side(p)$, $\langle 3\rangle 1$, and $i.rival = i$

$\langle 3\rangle 4.$ $T \neq p$ — by $\langle 3\rangle 3$, and $side(i) = 1 - side(p)$

$\langle 3\rangle 5.$ $C[1 - side(i)] = p$ — by 18.i, $\langle 2\rangle 4$, and $side(i) = 1 - side(p)$

$\langle 3\rangle 6.$ $p@\{5\}$ — by I7,$\langle 3\rangle 1$, $\langle 3\rangle 5$, and $\langle 3\rangle 4$

$\langle 3\rangle 7.$ $\perp$ — by $\langle 3\rangle 6$ and $\langle 3\rangle 2$

CASE: 19.i $\wedge$ $side(i) = 1 - side(p)$

$\langle 3\rangle 1.$ $i@\{19\}$ — by 19.i

$\langle 3\rangle 2.$ $p@\{8\} \;\wedge\; p.rival = -1 \;\vee\; p@\{9..14\}$ — by 19.i and $\langle 2\rangle 1$

$\langle 3\rangle 3.$ $i.rival = p$ — by I13, $\langle 3\rangle 2$, $side(i) = 1 - side(p)$, and $\langle 3\rangle 1$

$\langle 3\rangle 4.$ $P'[p] = 2$ — by $\langle 3\rangle 3$ and 19.i

$\langle 3\rangle 5.$ $\perp$ — by $\langle 3\rangle 4$ and $\langle 2\rangle 3$

$\square$

(I16)
$$\begin{aligned}
&\wedge\; p@\{10..12\}\\
&\wedge\; side(q) = 1 - side(p)\\
&\wedge\; q@\{8..12\}\\
&\wedge\; T = q\\
&\Rightarrow q.rival = p
\end{aligned}$$

$\langle 1\rangle 1.$ $Init \;\Rightarrow\; p@\{0\}$

$\quad\Rightarrow\;$ I16

$\langle 1\rangle 2.$ I16 $\Rightarrow$ I16$'$

ASSUME: I16 $\wedge$ $\neg$I16$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(p@\{10..12\})'$ — by $\neg$I16$'$

$\langle 2\rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg$I16$'$

$\langle 2\rangle 3.$ $(q@\{8..12\})'$ — by $\neg$I16$'$

$\langle 2\rangle 4.$ $T' = q$ — by $\neg$I16$'$

$\langle 2\rangle 5.$ $q.rival' \neq p$ — by $\neg$I16$'$

$\langle 2\rangle 6.$ $C'[side(p)] = p$ — by I3 and $\langle 2\rangle 1$

$\langle 2\rangle 7.$ — by I16 $\wedge$ $\neg$I16$'$
$$\begin{aligned}
&\vee\; 5.q\\
&\vee\; 7.q\\
&\vee\; 9.p \;\wedge\; T = p\\
&\vee\; 17.q
\end{aligned}$$

CASE: 5.q

$\langle 3\rangle 1.$ $(q@\{6\})'$ — by 5.q

$\langle 3\rangle 2.$ $\perp$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: 7.q

$\langle 3\rangle 1.$ $C[1 - side(q)] = p$ — by 7.q, $\langle 2\rangle 2$, and $\langle 2\rangle 6$

$\langle 3\rangle 2.$ $q.rival' = p$ — by 7.q and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\perp$ — by $\langle 3\rangle 2$ and $\langle 2\rangle 5$

CASE: 9.p $\wedge$ $T = p$

$\langle 3\rangle 1.\ T' = p$ — by $9.p$ and $T = p$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$, $\langle 2\rangle 4$, and $\langle 2\rangle 2$

CASE: $17.q$

$\langle 3\rangle 1.\ (q@\{18\})'$ — by $17.q$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

∎

$$(\mathrm{I17})\quad \begin{aligned} &\wedge\ p@\{10..12\}\\ &\wedge\ side(q) = 1 - side(p)\\ &\wedge\ q@\{12..14\}\\ &\wedge\ T = q\\ &\Rightarrow P[p] \geq 1 \end{aligned}$$

$\langle 1\rangle 1.\ Init \ \Rightarrow\ p@\{0\}$

$\quad\quad \Rightarrow\ \mathrm{I17}$

$\langle 1\rangle 2.\ \mathrm{I17}\ \Rightarrow\ \mathrm{I17}'$

ASSUME: $\mathrm{I17}\ \wedge\ \neg\mathrm{I17}'$

PROVE: $\bot$

$\langle 2\rangle 1.\ (p@\{10..12\})'$ — by $\neg\mathrm{I17}'$

$\langle 2\rangle 2.\ side(q) = 1 - side(p)$ — by $\neg\mathrm{I17}'$

$\langle 2\rangle 3.\ (q@\{12..14\})'$ — by $\neg\mathrm{I17}'$

$\langle 2\rangle 4.\ T' = q$ — by $\neg\mathrm{I17}'$

$\langle 2\rangle 5.\ P'[p] = 0$ — by $\neg\mathrm{I17}'$

$\langle 2\rangle 6.\ \vee\ 5.q$ — by $\mathrm{I17}\ \wedge\ \neg\mathrm{I17}'$

$\quad\vee\ 6.p$

$\quad\vee\ 9.p\ \wedge\ T = p$

$\quad\vee\ 10.q\ \wedge\ P[q.rival] \neq 0$

$\quad\vee\ 11.q$

CASE: $5.q$

$\langle 3\rangle 1.\ (q@\{6\})'$ — by $5.q$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $6.p$

$\langle 3\rangle 1.\ (p@\{7\})'$ — by $6.p$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $9.p\ \wedge\ T = p$

$\langle 3\rangle 1.\ T' = p$ — by $9.p$ and $T = p$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$, $\langle 2\rangle 4$, and $\langle 2\rangle 2$

CASE: $10.q\ \wedge\ P[q.rival] \neq 0$

$\langle 3\rangle 1.\ q@\{10\}\ \wedge\ p@\{10..12\}\ \wedge\ T = q$ — by $10.q$, $\langle 2\rangle 1$, and $\langle 2\rangle 4$

$\langle 3\rangle 2.\ q.rival = p$ — by I16, $\langle 3\rangle 1$, and $\langle 2\rangle 2$

$\langle 3\rangle 3.\ P[p] = 0$ — by $10.q$ and $\langle 2\rangle 5$

$\langle 3\rangle 4.\ (q@\{11\})'$ — by $10.q$, $\langle 3\rangle 2$, and $\langle 3\rangle 3$

$\langle 3\rangle 5.\ \bot$ — by $\langle 3\rangle 4$ and $\langle 2\rangle 3$

CASE: $11.q$

$\langle 3\rangle 1.$ $q@\{11\} \;\wedge\; p@\{10..12\} \;\wedge\; T = q$ — by $11.q$, $\langle 2\rangle 1$, and $\langle 2\rangle 4$

$\langle 3\rangle 2.$ $q.rival = p$ — by I16, $\langle 3\rangle 1$, and $\langle 2\rangle 2$

$\langle 3\rangle 3.$ $P'[p] = 1$ — by $11.q$ and $\langle 3\rangle 2$

$\langle 3\rangle 4.$ $\perp$ — by $\langle 3\rangle 3$ and $\langle 2\rangle 5$

$\square$

(I18) $\wedge\, p@\{12\}$
$\wedge\, side(q) = 1 - side(p)$
$\wedge\, q@\{12\}$
$\Rightarrow \vee\, P[p] \geq 1$
$\quad \vee\, P[q] \geq 1$

ASSUME: $\wedge\, p@\{12\}$
$\wedge\, side(q) = 1 - side(p)$
$\wedge\, q@\{12\}$

PROVE: $\vee\, P[p] \geq 1$
$\vee\, P[q] \geq 1$

$\langle 1\rangle 1.$ $\vee\, T = p$
$\vee\, T = q$ — by I4 and the assumptions

$\langle 1\rangle 2.$ $\vee\, P[p] \geq 1$
$\vee\, P[q] \geq 1$ — by I17, $\langle 1\rangle 1$, and the assumptions

$\square$

(I19) $i@\{13, 14\}$
$\Rightarrow P[i] \geq 1$

$\langle 1\rangle 1.$ $Init \;\Rightarrow\; i@\{0\}$
$\Rightarrow\;$ I19

$\langle 1\rangle 2.$ I19 $\;\Rightarrow\;$ I19$'$

ASSUME: I19 $\wedge\; \neg$I19$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(i@\{13, 14\})'$ — by $\neg$I19$'$

$\langle 2\rangle 2.$ $P'[i] = 0$ — by $\neg$I19$'$

$\langle 2\rangle 3.$ $\vee\, 6.i$
$\vee\, 12.i \;\wedge\; P[i] \geq 1$ — by I19 $\wedge\; \neg$I19$'$

CASE: $6.i$

$\langle 3\rangle 1.$ $(i@\{7\})'$ — by $6.i$

$\langle 3\rangle 2.$ $\perp$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $12.i \;\wedge\; P[i] \geq 1$

$\langle 3\rangle 1.\ P'[i] \geq 1$ — by $12.i$ and $P[i] \geq 1$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 2$

$\square$

$$
\text{(I20)} \quad \begin{array}{l} \wedge\ p@\{14\} \\ \wedge\ side(q) = 1 - side(p) \\ \wedge\ q@\{7..14\} \\ \wedge\ T = q \\ \Rightarrow\ P[q] = 0 \end{array}
$$

$\langle 1\rangle 1.\ Init \ \Rightarrow\ p@\{0\}$

$\qquad \Rightarrow\ \text{I20}$

$\langle 1\rangle 2.\ \text{I20} \ \Rightarrow\ \text{I20}'$

ASSUME: $\text{I20} \ \wedge\ \neg\text{I20}'$

PROVE: $\bot$

$\langle 2\rangle 1.\ (p@\{14\})'$ — by $\neg\text{I20}'$

$\langle 2\rangle 2.\ side(q) = 1 - side(p)$ — by $\neg\text{I20}'$

$\langle 2\rangle 3.\ (q@\{7..14\})'$ — by $\neg\text{I20}'$

$\langle 2\rangle 4.\ T' = q$ — by $\neg\text{I20}'$

$\langle 2\rangle 5.\ P'[q] \neq 0$ — by $\neg\text{I20}'$

$\langle 2\rangle 6.$ — by $\text{I20} \ \wedge\ \neg\text{I20}'$

$$
\begin{array}{l} \vee\ 5.q \\ \vee\ 6.q \\ \vee\ 11.i \ \wedge\ i.rival = q \\ \vee\ 13.p \ \wedge\ T = p \\ \vee\ 19.i \end{array}
$$

CASE: $5.q$

$\langle 3\rangle 1.\ (q@\{6\})'$ — by $5.q$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $6.q$

$\langle 3\rangle 1.\ P'[q] = 0$ — by $6.q$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 5$

CASE: $11.i \ \wedge\ i.rival = q$

$\langle 3\rangle 1.\ i@\{11\}$ — by $11.i$

$\langle 3\rangle 2.\ side(i) = 1 - side(q)$ — by I10, $\langle 3\rangle 1$, and $i.rival = q$

$\langle 3\rangle 3.\ p@\{14\}$ — by $11.i$ and $\langle 2\rangle 1$

$\langle 3\rangle 4.\ \bot$ — by $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, $\langle 2\rangle 2$, and LME

CASE: $13.p \ \wedge\ T = p$

$\langle 3\rangle 1.\ T' = p$ — by $13.p$ and $T = p$

$\langle 3\rangle 2.\ \bot$ — by $\langle 3\rangle 1$, $\langle 2\rangle 4$, and $\langle 2\rangle 2$

CASE: $19.i$

CASE: $i = p$

$\langle 4\rangle 1.\ (p@\{20\})'$ — by $19.p$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$ and $\langle 2\rangle 1$

CASE: $i \neq p \ \wedge\ side(i) = side(p)$

$\langle 4\rangle 1.\ i@\{19\} \ \wedge\ p@\{14\}$ — by $19.i$ and $\langle 2\rangle 1$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$, $side(i) = side(p)$, and LME

CASE: $i = q$

$\langle 4\rangle 1.\ (q@\{20\})'$ — by $19.q$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$ and $\langle 2\rangle 3$

CASE: $i \neq q \ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.\ i@\{19\} \ \wedge\ q@\{7..14\}$ — by $19.i$ and $\langle 2\rangle 3$

$\langle 4\rangle 2.\ \bot$ — by $\langle 4\rangle 1$, $side(i) = side(q)$, and LME

□

(I21) $\neg(p@\{14\} \ \wedge\ side(q) = 1 - side(p) \ \wedge\ q@\{14\})$

ASSUME: $\neg$I21

PROVE: $\bot$

$\langle 1\rangle 1.\ \wedge\, p@\{14\}$ — by $\neg$I21

$\quad \wedge\, side(q) = 1 - side(p)$

$\quad \wedge\, q@\{14\}$

$\langle 1\rangle 2.\ P[p] \geq 1 \ \wedge\ P[q] \geq 1$ — by I19 and $\langle 1\rangle 1$

$\langle 1\rangle 3.\ T = p \ \vee\ T = q$ — by I4 and $\langle 1\rangle 1$

$\langle 1\rangle 4.\ P[p] = 0 \ \vee\ P[q] = 0$ — by I20, $\langle 1\rangle 1$, and $\langle 1\rangle 3$

$\langle 1\rangle 5.\ \bot$ — by $\langle 1\rangle 2$ and $\langle 1\rangle 4$

□

(I22) $\wedge\, p@\{6..14\}$

$\quad \wedge\, side(q) = 1 - side(p)$

$\quad \wedge\, q@\{15..19\}$

$\quad \Rightarrow T = p$

$\langle 1\rangle 1.\ Init \ \Rightarrow\ p@\{0\}$

$\quad \Rightarrow\ $ I22

$\langle 1\rangle 2.$ I22 $\Rightarrow$ I22$'$

ASSUME: I22 $\wedge\ \neg$I22$'$

PROVE: $\bot$

$\langle 2\rangle 1.\ (p@\{6..14\})'$ — by $\neg$I22$'$

$\langle 2\rangle 2.\ side(q) = 1 - side(p)$ — by $\neg$I22$'$

$\langle 2\rangle 3.\ (q@\{15..19\})'$ — by $\neg$I22$'$

$\langle 2\rangle 4.\ T' \neq p$ — by $\neg$I22$'$

$\langle 2\rangle 5.\ C'[side(p)] = p$ — by I3 and $\langle 2\rangle 1$

$\langle 2\rangle 6.\ \vee\ 5.i$ by I22 $\wedge$ $\neg$I22$'$

$\vee\ 8.q\ \wedge\ q.rival = -1$

$\vee\ (9.q\ \vee\ 13.q)\ \wedge\ T \neq q$

$\vee\ 14.q\ \wedge\ P[q] = 2$

CASE: $5.i$

CASE: $i = p$

$\langle 4\rangle 1.\ T' = p$ by $5.p$

$\langle 4\rangle 2.\ \perp$ by $\langle 4\rangle 1$ and $\langle 2\rangle 4$

CASE: $i \neq p\ \wedge\ side(i) = side(p)$

$\langle 4\rangle 1.\ (i@\{6\})'$ by $5.i$

$\langle 4\rangle 2.\ \perp$ by $\langle 4\rangle 1$, $\langle 2\rangle 1$, $side(i) = side(p)$, and LME

CASE: $i = q$

$\langle 4\rangle 1.\ (q@\{6\})'$ by $5.q$

$\langle 4\rangle 2.\ \perp$ by $\langle 4\rangle 1$ and $\langle 2\rangle 3$

CASE: $i \neq q\ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.\ (i@\{6\})'$ by $5.i$

$\langle 4\rangle 2.\ \perp$ by $\langle 4\rangle 1$, $\langle 2\rangle 3$, $side(i) = side(q)$, and LME

CASE: $8.q\ \wedge\ q.rival = -1$

$\langle 3\rangle 1.\ q@\{8\}$ by $8.q$

$\langle 3\rangle 2.\ C[1 - side(q)] = p\ \wedge\ p \neq -1$ by $8.q$, $\langle 2\rangle 2$, and $\langle 2\rangle 5$

$\langle 3\rangle 3.\ p@\{6..14\}$ by $8.q$ and $\langle 2\rangle 1$

$\langle 3\rangle 4.\ T = p$ I5, $\langle 3\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 3$, and $q.rival = -1$

$\langle 3\rangle 5.\ T' = p$ $8.q$ and $\langle 3\rangle 4$

$\langle 3\rangle 6.\ \perp$ by $\langle 3\rangle 5$ and $\langle 2\rangle 4$

CASE: $(9.q\ \vee\ 13.q)\ \wedge\ T \neq q$

$\langle 3\rangle 1.\ (q@\{15\})'\ \wedge\ T' \neq q$ by $9.q\ \vee\ 13.q$ and $T \neq q$

$\langle 3\rangle 2.\ T' = p$ by I4, $\langle 2\rangle 1$, $\langle 2\rangle 2$, and $\langle 3\rangle 1$

$\langle 3\rangle 3.\ \perp$ by $\langle 3\rangle 2$ and $\langle 2\rangle 4$

CASE: $14.q\ \wedge\ P[q] = 2$

$\langle 3\rangle 1.\ (q@\{15\})'\ \wedge\ P'[q] = 2$ by $14.q$ and $P[q] = 2$

$\langle 3\rangle 2.\ C[1 - side(q)] = p\ \wedge\ p \neq -1$ by $14.q$, $\langle 2\rangle 2$, and $\langle 2\rangle 5$

$\langle 3\rangle 3.\ T' = p$ by I6, $\langle 3\rangle 1$, $\langle 3\rangle 2$, and $\langle 2\rangle 1$

$\langle 3\rangle 4.\ \perp$ by $\langle 3\rangle 3$ and $\langle 2\rangle 4$

$\square$

(I23) $\wedge\ p@\{7..14\}$

$\wedge\ side(q) = 1 - side(p)$

$\wedge\ q@\{18, 19\}$

$\Rightarrow\ \vee\ q.rival = p$

$\vee\ P[p] = 0$

$\vee\ P[p] = 2$

$\langle 1\rangle 1.\ Init\ \Rightarrow\ p@\{0\}$

$\Rightarrow$ I23

$\langle 1 \rangle 2.$ I23 $\Rightarrow$ I23$'$

ASSUME: I23 $\wedge$ $\neg$I23$'$

PROVE: $\bot$

$\langle 2 \rangle 1.$ $(p@\{7..14\})'$ — by $\neg$I23$'$

$\langle 2 \rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg$I23$'$

$\langle 2 \rangle 3.$ $(q@\{18, 19\})'$ — by $\neg$I23$'$

$\langle 2 \rangle 4.$ $q.rival' \neq p$ — by $\neg$I23$'$

$\langle 2 \rangle 5.$ $P'[p] \neq 0$ — by $\neg$I23$'$

$\langle 2 \rangle 6.$ $P'[p] \neq 2$ — by $\neg$I23$'$

$\langle 2 \rangle 7.$ $\vee$ $6.p$ — by I23 $\wedge$ $\neg$I23$'$

$\vee$ $7.q$

$\vee$ $11.i \wedge i.rival = p$

$\vee$ $17.q$

CASE: $6.p$

$\langle 3 \rangle 1.$ $P'[p] = 0$ — by $6.p$

$\langle 3 \rangle 2.$ $\bot$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 5$

CASE: $7.q$

$\langle 3 \rangle 1.$ $(q@\{8\})'$ — by $7.q$

$\langle 3 \rangle 2.$ $\bot$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 3$

CASE: $11.i \wedge i.rival = p$

$\langle 3 \rangle 1.$ $i@\{11\}$ — by $11.i$

$\langle 3 \rangle 2.$ $side(i) = 1 - side(p)$ — by I10, $\langle 3 \rangle 1$, and $i.rival = p$

$\langle 3 \rangle 3.$ $side(i) = side(q)$ — by $\langle 3 \rangle 2$, and $\langle 2 \rangle 2$

CASE: $i = q$

$\langle 4 \rangle 1.$ $(q@\{12\})'$ — by $11.q$

$\langle 4 \rangle 2.$ $\bot$ — by $\langle 4 \rangle 1$ and $\langle 2 \rangle 3$

CASE: $i \neq q$

$\langle 4 \rangle 1.$ $(i@\{12\})'$ — by $11.i$

$\langle 4 \rangle 2.$ $\bot$ — by $\langle 4 \rangle 1$, $\langle 2 \rangle 3$, $\langle 3 \rangle 3$, and LME

CASE: $17.q$

$\langle 3 \rangle 1.$ $p@\{7..14\} \wedge q@\{17\}$ — by $17.q$ and $\langle 2 \rangle 1$

$\langle 3 \rangle 2.$ $T = p$ — by I22, $\langle 3 \rangle 1$, and $\langle 2 \rangle 2$

$\langle 3 \rangle 3.$ $q.rival' = p$ — by $17.q$ and $\langle 3 \rangle 2$

$\langle 3 \rangle 4.$ $\bot$ — by $\langle 3 \rangle 3$ and $\langle 2 \rangle 4$

$\square$

(I24)

$$
\begin{aligned}
& \wedge\ p@\{7..14\} \\
& \wedge\ (\forall i : side(i) = 1 - side(p) :: i@\{0..5, 20..22\}) \\
\Rightarrow\ & \vee\ P[p] = 0 \\
& \vee\ P[p] = 2
\end{aligned}
$$

$\langle 1 \rangle 1.$ $Init \Rightarrow p@\{0\}$

$\Rightarrow$ I24

$\langle 1 \rangle 2.$ I24 $\Rightarrow$ I24$'$

ASSUME: I24 $\wedge$ $\neg$I24$'$

PROVE: $\perp$

$\langle 2 \rangle 1.$ $(p@\{7..14\})'$ — by $\neg$I24$'$

$\langle 2 \rangle 2.$ $(\forall i : side(i) = 1 - side(p) :: (i@\{0..5, 20..22\})')$ — by $\neg$I24$'$

$\langle 2 \rangle 3.$ $P'[p] \neq 0$ — by $\neg$I24$'$

$\langle 2 \rangle 4.$ $P'[p] \neq 2$ — by $\neg$I24$'$

$\langle 2 \rangle 5.$ $\vee\ 6.p$ — by I24 $\wedge$ $\neg$I24$'$

$\vee\ 11.i \wedge i.rival = p$

$\vee\ 18.i \wedge side(i) = 1 - side(p) \wedge i.rival = i$

$\vee\ 19.i \wedge side(i) = 1 - side(p)$

CASE: $6.p$

$\langle 3 \rangle 1.$ $P'[p] = 0$ — by $6.p$

$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 3$

CASE: $11.i \wedge i.rival = p$

$\langle 3 \rangle 1.$ $i@\{11\}$ — by $11.i$

$\langle 3 \rangle 2.$ $side(i) = 1 - side(p)$ — by I10, $\langle 3 \rangle 1$, and $i.rival = p$

$\langle 3 \rangle 3.$ $(i@\{12\})'$ — by $11.i$

$\langle 3 \rangle 4.$ $\perp$ — by $\langle 3 \rangle 2$, $\langle 3 \rangle 3$, and $\langle 2 \rangle 2$

CASE: $18.i \wedge side(i) = 1 - side(p) \wedge i.rival = i$

$\langle 3 \rangle 1.$ $i@\{18\}$ — by $18.i$

$\langle 3 \rangle 2.$ $p@\{7..14\}$ — by $18.i$ and $\langle 2 \rangle 1$

$\langle 3 \rangle 3.$ $P[p] = 0 \vee P[p] = 2$ — by I23, $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, $side(i) = 1 - side(p)$, and $i.rival = i$

$\langle 3 \rangle 4.$ $P'[p] = 0 \vee P'[p] = 2$ — by $18.i$ and $\langle 3 \rangle 3$

$\langle 3 \rangle 5.$ $\perp$ — by $\langle 3 \rangle 4$, $\langle 2 \rangle 3$, and $\langle 2 \rangle 4$

CASE: $19.i \wedge side(i) = 1 - side(p)$

$\langle 3 \rangle 1.$ $i@\{19\}$ — by $19.i$

$\langle 3 \rangle 2.$ $p@\{7..14\}$ — by $19.i$ and $\langle 2 \rangle 1$

$\langle 3 \rangle 3.$ $i.rival = p \vee P[p] = 0 \vee P[p] = 2$ — by I23, $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, and $side(i) = 1 - side(p)$

$\langle 3 \rangle 4.$ $P'[p] = 0 \vee P'[p] = 2$ — by $19.i$ and $\langle 3 \rangle 3$

$\langle 3 \rangle 5.$ $\perp$ — by $\langle 3 \rangle 4$, $\langle 2 \rangle 3$, and $\langle 2 \rangle 4$

$\square$

(I25) $\wedge\ p@\{8\} \wedge p.rival \neq -1 \vee p@\{9..14\}$

$\wedge\ side(q) = 1 - side(p)$

$\wedge\ q@\{18, 19\}$

$\Rightarrow q.rival = p$

$\langle 1 \rangle 1.$ $Init \Rightarrow p@\{0\}$

$\Rightarrow$ I25

$\langle 1 \rangle 2.$ I25 $\Rightarrow$ I25$'$

ASSUME: I25 $\wedge$ $\neg$I25$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(p@\{8\})' \ \wedge \ p.rival' \neq -1 \ \vee \ (p@\{9..14\})'$ — by $\neg$I25$'$

$\langle 2\rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg$I25$'$

$\langle 2\rangle 3.$ $(q@\{18, 19\})'$ — by $\neg$I25$'$

$\langle 2\rangle 4.$ $q.rival' \neq p$ — by $\neg$I25$'$

$\langle 2\rangle 5.$ $C'[side(q)] = -1$ — by I11, $\langle 2\rangle 3$, and LME

$\langle 2\rangle 6.$ $\vee\ 7.p$ — by I25 $\wedge$ $\neg$I25$'$
$\vee\ 7.q$
$\vee\ 17.q$

CASE: $7.p$

$\langle 3\rangle 1.$ $C'[1 - side(p)] = -1$ — by $7.p$, $\langle 2\rangle 2$, $\langle 2\rangle 5$

$\langle 3\rangle 2.$ $(p@\{8\})' \ \wedge \ p.rival' = -1$ — by $7.p$ and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\perp$ — by $\langle 3\rangle 2$ and $\langle 2\rangle 1$

CASE: $7.q$

$\langle 3\rangle 1.$ $(q@\{8\})'$ — by $7.q$

$\langle 3\rangle 2.$ $\perp$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $17.q$

$\langle 3\rangle 1.$ $q@\{17\} \ \wedge \ p@\{8..14\}$ — by $17.q$ and $\langle 2\rangle 1$

$\langle 3\rangle 2.$ $T = p$ — by I22, $\langle 3\rangle 1$, and $\langle 2\rangle 2$

$\langle 3\rangle 3.$ $q.rival' = p$ — by $17.q$ and $\langle 3\rangle 2$

$\langle 3\rangle 4.$ $\perp$ — by $\langle 3\rangle 3$ and $\langle 2\rangle 4$

$\square$

(I26) $\quad \wedge\ p@\{13, 14\}$
$\wedge\ (\forall i : side(i) = 1 - side(p) :: i@\{0..5, 20..22\})$
$\Rightarrow P[p] = 2$

$\langle 1\rangle 1.$ $Init \ \Rightarrow \ p@\{0\}$
$\Rightarrow$ I26

$\langle 1\rangle 2.$ I26 $\Rightarrow$ I26$'$

ASSUME: I26 $\wedge$ $\neg$I26$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(p@\{13, 14\})'$ — by $\neg$I26$'$

$\langle 2\rangle 2.$ $(\forall i : side(i) = 1 - side(p) :: (i@\{0..5, 20..22\})')$ — by $\neg$I26$'$

$\langle 2\rangle 3.$ $P'[p] \neq 2$ — by $\neg$I26$'$

$\langle 2\rangle 4.$ $\vee\ 6.p$ — by I26 $\wedge$ $\neg$I26$'$
$\vee\ 11.i \ \wedge \ i.rival = p$
$\vee\ 12.p \ \wedge \ P[p] \neq 0$
$\vee\ 18.i \ \wedge \ side(i) = 1 - side(p)$
$\vee\ 19.i \ \wedge \ side(i) = 1 - side(p)$

CASE: $6.p$

$\langle 3\rangle 1.$ $(p@\{7\})'$ — by $6.p$

$\langle 3\rangle 2.$ $\perp$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $11.i \ \wedge \ i.rival = p$

$\langle 3 \rangle 1.$ $i@\{11\}$ — by $11.i$

$\langle 3 \rangle 2.$ $side(i) = 1 - side(p)$ — by I10, $\langle 3 \rangle 1$, and $i.rival = p$

$\langle 3 \rangle 3.$ $(i@\{12\})'$ — by $11.i$

$\langle 3 \rangle 4.$ $\perp$ — by $\langle 3 \rangle 2$, $\langle 3 \rangle 3$, and $\langle 2 \rangle 2$

CASE: $12.p \ \wedge \ P[p] \neq 0$

$\langle 3 \rangle 1.$ $(p@\{13\})' \ \wedge \ P'[p] \neq 0$ — by $12.p$

$\langle 3 \rangle 2.$ $P'[p] = 2$ — by I24, $\langle 3 \rangle 1$, and $\langle 2 \rangle 2$

$\langle 3 \rangle 3.$ $\perp$ — by $\langle 3 \rangle 2$ and $\langle 2 \rangle 3$

CASE: $18.i \ \wedge \ side(i) = 1 - side(p)$

$\langle 3 \rangle 1.$ $i@\{18\}$ — by $18.i$

$\langle 3 \rangle 2.$ $p@\{13, 14\}$ — by $18.i$ and $\langle 2 \rangle 1$

$\langle 3 \rangle 3.$ $i.rival = p$ — by I25, $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, and $side(i) = 1 - side(p)$

$\langle 3 \rangle 4.$ $(i@\{19\})'$ — by $18.i$ and $\langle 3 \rangle 3$

$\langle 3 \rangle 5.$ $\perp$ — by $\langle 3 \rangle 4$, $\langle 2 \rangle 2$, and $side(i) = 1 - side(p)$

CASE: $19.i \ \wedge \ side(i) = 1 - side(p)$

$\langle 3 \rangle 1.$ $i@\{19\}$ — by $19.i$

$\langle 3 \rangle 2.$ $p@\{13, 14\}$ — by $19.i$ and $\langle 2 \rangle 1$

$\langle 3 \rangle 3.$ $i.rival = p$ — by I25, $\langle 3 \rangle 1$, $\langle 3 \rangle 2$, and $side(i) = 1 - side(p)$

$\langle 3 \rangle 4.$ $P'[p] = 2$ — by $19.i$ and $\langle 3 \rangle 3$

$\langle 3 \rangle 5.$ $\perp$ — by $\langle 3 \rangle 4$ and $\langle 2 \rangle 3$

$\square$

(I27)
$$\begin{aligned} & \wedge\ p@\{7..14\} \\ & \wedge\ side(q) = 1 - side(p) \\ & \wedge\ q@\{11\} \\ & \wedge\ q.rival = p \\ \Rightarrow\ & P[p] = 0 \end{aligned}$$

$\langle 1 \rangle 1.$ $Init \ \Rightarrow \ p@\{0\}$
$\qquad \Rightarrow$ I27

$\langle 1 \rangle 2.$ I27 $\Rightarrow$ I27$'$

ASSUME: I27 $\wedge$ $\neg$I27$'$

PROVE: $\perp$

$\langle 2 \rangle 1.$ $(p@\{7..14\})'$ — by $\neg$I27$'$

$\langle 2 \rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg$I27$'$

$\langle 2 \rangle 3.$ $(q@\{11\})'$ — by $\neg$I27$'$

$\langle 2 \rangle 4.$ $q.rival' = p$ — by $\neg$I27$'$

$\langle 2 \rangle 5.$ $P'[p] \neq 0$ — by $\neg$I27$'$

$\langle 2\rangle 6.$ $\vee\ 6.p$ $\qquad$ by I27 $\wedge$ $\neg$I27$'$

$\vee\ 7.q \ \vee\ 17.q$

$\vee\ 10.q \ \wedge\ P[q.rival] = 0$

$\vee\ 11.i \ \wedge\ i.rival = p$

$\vee\ 19.i$

CASE: $6.p$

$\langle 3\rangle 1.$ $P'[p] = 0$ $\qquad$ by $6.p$

$\langle 3\rangle 2.$ $\perp$ $\qquad$ by $\langle 3\rangle 1$ and $\langle 2\rangle 5$

CASE: $7.q \ \wedge\ 17.q$

$\langle 3\rangle 1.$ $(q@\{8, 18\})'$ $\qquad$ by $7.q \ \vee\ 17.q$

$\langle 3\rangle 2.$ $\perp$ $\qquad$ by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

CASE: $10.q \ \wedge\ P[q.rival] = 0$

$\langle 3\rangle 1.$ $q.rival = p$ $\qquad$ by $10.q$ and $\langle 2\rangle 4$

$\langle 3\rangle 2.$ $P[p] = 0$ $\qquad$ by $\langle 3\rangle 1$ and $P[q.rival] = 0$

$\langle 3\rangle 3.$ $P'[p] = 0$ $\qquad$ by $10.q$ and $\langle 3\rangle 2$

$\langle 3\rangle 4.$ $\perp$ $\qquad$ by $\langle 3\rangle 3$ and $\langle 2\rangle 5$

CASE: $11.i \ \wedge\ i.rival = p$

$\langle 3\rangle 1.$ $i@\{11\}$ $\qquad$ by $11.i$

$\langle 3\rangle 2.$ $side(i) = 1 - side(p)$ $\qquad$ by I10, $\langle 3\rangle 1$, and $i.rival = p$

$\langle 3\rangle 3.$ $side(i) = side(q)$ $\qquad$ by $\langle 3\rangle 2$, and $\langle 2\rangle 2$

CASE: $i = q$

$\langle 4\rangle 1.$ $(q@\{12\})'$ $\qquad$ by $11.q$

$\langle 4\rangle 2.$ $\perp$ $\qquad$ by $\langle 4\rangle 1$ and $\langle 2\rangle 3$

CASE: $i \neq q$

$\langle 4\rangle 1.$ $(i@\{12\})'$ $\qquad$ by $11.i$

$\langle 4\rangle 2.$ $\perp$ $\qquad$ by $\langle 4\rangle 1$, $\langle 2\rangle 3$, $side(i) = side(q)$, and LME

CASE: $19.i$

$\langle 3\rangle 1.$ $p@\{7..14\}$ $\qquad$ by $19.i$ and $\langle 2\rangle 1$

$\langle 3\rangle 2.$ $q@\{11\}$ $\qquad$ by $19.i$ and $\langle 2\rangle 3$

$\langle 3\rangle 3.$ $i@\{19\}$ $\qquad$ by $19.i$

CASE: $i = p$

$\langle 4\rangle 1.$ $\perp$ $\qquad$ by $\langle 3\rangle 1$ and $\langle 3\rangle 3$

CASE: $i \neq p \ \wedge\ side(i) = side(p)$

$\langle 4\rangle 1.$ $\perp$ $\qquad$ by $\langle 3\rangle 1$, $\langle 3\rangle 3$, $side(i) = side(p)$, and LME

CASE: $i = q$

$\langle 4\rangle 1.$ $\perp$ $\qquad$ by $\langle 3\rangle 2$ and $\langle 3\rangle 3$

CASE: $i \neq q \ \wedge\ side(i) = side(q)$

$\langle 4\rangle 1.$ $\perp$ $\qquad$ by $\langle 3\rangle 2$, $\langle 3\rangle 3$, $side(i) = side(q)$, and LME

$\square$

(I28) $\quad \wedge\ p@\{14\}$

$\wedge\ side(q) = 1 - side(p)$

$\wedge\ T = q$

$\Rightarrow P[p] = 2$

$\langle 1 \rangle 1.$ $Init \;\Rightarrow\; p@\{0\}$
$\quad\quad\quad\;\Rightarrow\;$ I28
$\langle 1 \rangle 2.$ I28 $\Rightarrow$ I28$'$
ASSUME: I28 $\wedge$ $\neg$I28$'$
PROVE: $\perp$
$\langle 2 \rangle 1.$ $(p@\{14\})'$ — by $\neg$I28$'$
$\langle 2 \rangle 2.$ $side(q) = 1 - side(p)$ — by $\neg$I28$'$
$\langle 2 \rangle 3.$ $T' = q$ — by $\neg$I28$'$
$\langle 2 \rangle 4.$ $P'[p] \neq 2$ — by $\neg$I28$'$
$\langle 2 \rangle 5.$ $\vee\; 5.q$ — by I28 $\wedge$ $\neg$I28$'$
$\vee\; 6.p$
$\vee\; 11.i \;\wedge\; i.rival = p \;\wedge\; P[p] = 2$
$\vee\; 13.p \;\wedge\; T = p$
CASE: $5.q$
$\langle 3 \rangle 1.$ $q@\{5\}$ — by $5.q$
$\langle 3 \rangle 2.$ $(\forall i : side(i) = side(q) :: i@\{0..5, 20..22\})$ — by $\langle 3 \rangle 1$ and LME
$\langle 3 \rangle 3.$ $p@\{14\}$ — by $5.q$ and $\langle 2 \rangle 1$
$\langle 3 \rangle 4.$ $P[p] = 2$ — by I26, $\langle 3 \rangle 3$, $\langle 3 \rangle 2$, and $\langle 2 \rangle 2$
$\langle 3 \rangle 5.$ $P'[p] = 2$ — by $5.q$ and $\langle 3 \rangle 4$
$\langle 3 \rangle 6.$ $\perp$ — by $\langle 3 \rangle 5$ and $\langle 2 \rangle 4$
CASE: $6.p$
$\langle 3 \rangle 1.$ $(p@\{7\})'$ — by $6.p$
$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$ and $\langle 2 \rangle 1$
CASE: $11.i \;\wedge\; i.rival = p \;\wedge\; P[p] = 2$
$\langle 3 \rangle 1.$ $i@\{11\}$ — by $11.i$
$\langle 3 \rangle 2.$ $side(i) = 1 - side(p)$ — by I10, $\langle 3 \rangle 1$, and $i.rival = p$
$\langle 3 \rangle 3.$ $p@\{14\}$ — by $11.i$ and $\langle 2 \rangle 1$
$\langle 3 \rangle 4.$ $P[p] = 0$ — by I27, $\langle 3 \rangle 3$, $\langle 3 \rangle 2$, $\langle 3 \rangle 1$, and $i.rival = p$
$\langle 3 \rangle 5.$ $\perp$ — by $\langle 3 \rangle 4$ and $P[p] = 2$
CASE: $13.p \;\wedge\; T = p$
$\langle 3 \rangle 1.$ $T' = p$ — by $13.p$ and $T = p$
$\langle 3 \rangle 2.$ $\perp$ — by $\langle 3 \rangle 1$, $\langle 2 \rangle 3$, and $\langle 2 \rangle 2$

$\square$

$$(\text{I29})\quad \begin{array}{l} \wedge\; i@\{12\} \\ \wedge\; side(q) = 1 - side(i) \\ \wedge\; q@\{14\} \\ \Rightarrow \vee\; P[i] \geq 1 \\ \quad\;\; \vee\; P[q] = 2 \end{array}$$

ASSUME: $\neg$I29
PROVE: $\perp$
$\langle 1 \rangle 1.$ $i@\{12\}$ — by $\neg$I29

$\langle 1\rangle 2.\ side(q) = 1 - side(i)$ by $\neg$I29
$\langle 1\rangle 3.\ q@\{14\}$ by $\neg$I29
$\langle 1\rangle 4.\ P[i] = 0$ by $\neg$I29
$\langle 1\rangle 5.\ P[q] \leq 1$ by $\neg$I29
$\langle 1\rangle 6.\ T = i \ \vee \ T = q$ by I4, $\langle 1\rangle 1$, $\langle 1\rangle 3$, and $\langle 1\rangle 2$
$\langle 1\rangle 7.\ T = i$ by I17, $\langle 1\rangle 1$, $\langle 1\rangle 3$, $\langle 1\rangle 2$, $\langle 1\rangle 4$, and $\langle 1\rangle 6$
$\langle 1\rangle 8.\ P[q] = 2$ by I28, $\langle 1\rangle 3$, $\langle 1\rangle 2$, and $\langle 1\rangle 7$
$\langle 1\rangle 9.\ \perp$ by $\langle 1\rangle 8$ and $\langle 1\rangle 5$

$\square$

(I15) implies that if only one process is in its entry section, then that process's busy-waiting loops will terminate. (I18), (I21), and (I29) imply that if two processes are in their entry sections then at least one of them can make progress. Note that at most two process may execute their entry sections at this level. Hence, we conclude that the two-process solution in Figure 1 is free from livelock.

Next, we prove that the algorithm is free from starvation. To facilitate the presentation, we define the following predicate, which represents the starvation-freedom property.

$$(\forall i :: i@\{1..14\} \ \mapsto \ i@\{15\}) \tag{SF}$$

The following two assertions imply that (SF) holds.

$$(\forall i :: i@\{1..3\} \ \mapsto \ i@\{4\}) \tag{LSF}$$

$$(\forall i :: i@\{4..14\} \ \mapsto \ i@\{15\}) \tag{2SF}$$

As we explained above, *ENTRY_LEFT(ENTRY_RIGHT)* and *EXIT_LEFT(EXIT_RIGHT)* are obtained by nesting the two process mutual exclusion algorithm in lines from 4 to 19. It is straightforward to show, by induction on the depth of the tree, that (LSF) follows from (2SF). In the rest of this section, we prove that (2SF) holds. First, we prove two *unless* assertions.

$$i@\{12\} \ \wedge \ P[i] \geq 1 \ \ unless \ \ i@\{13\} \tag{U1}$$

$\langle 1\rangle 1.\ i@\{12\} \ \wedge \ P[i] \geq 1 \ \Rightarrow \ (i@\{12\})' \ \wedge \ P'[i] \geq 1 \ \vee \ (i@\{13\})'$
 ASSUME: $i@\{12\} \ \wedge \ P[i] \geq 1 \ \wedge \ (\neg(i@\{12\})' \ \vee \ P'[i] = 0) \ \wedge \ \neg(i@\{13\})'$
 PROVE: $\perp$
  $\langle 2\rangle 1.\ i@\{12\}$ by the assumption
  $\langle 2\rangle 2.\ P[i] \geq 1$ by the assumption
  $\langle 2\rangle 3.\ \neg(i@\{13\})'$ by the assumption
  $\langle 2\rangle 4.\ \vee\ 6.i$ by $\neg$U1
   $\vee\ 12.i$
   CASE: $6.i$
    $\langle 3\rangle 1.\ i@\{6\}$ by $6.i$
    $\langle 3\rangle 2.\ \perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $12.i$

$\langle 3\rangle 1.$ $(i@\{13\})'$ — by $12.i$ and $\langle 2\rangle 2$

$\langle 3\rangle 2.$ $\bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

$\square$

$$i@\{14\} \ \wedge\ P[i] = 2 \ \ \textit{unless} \ \ i@\{15\} \tag{U2}$$

$\langle 1\rangle 1.$ $i@\{14\} \ \wedge\ P[i] = 2 \ \Rightarrow\ (i@\{14\})' \ \wedge\ P'[i] = 2 \ \vee\ (i@\{15\})'$

ASSUME: $i@\{14\} \ \wedge\ P[i] = 2 \ \wedge\ (\neg(i@\{14\})' \ \vee\ P'[i] \leq 1) \ \wedge\ \neg(i@\{15\})'$

PROVE: $\bot$

$\langle 2\rangle 1.$ $i@\{14\}$ — by the assumption

$\langle 2\rangle 2.$ $P[i] = 2$ — by the assumption

$\langle 2\rangle 3.$ $\neg(i@\{15\})'$ — by the assumption

$\langle 2\rangle 4.$ $\vee\ 6.i$ — by $\neg$U2

$\vee\ 11.q \ \wedge\ q.rival = i$

$\vee\ 14.i$

CASE: $6.i$

$\langle 3\rangle 1.$ $i@\{6\}$ — by $6.i$

$\langle 3\rangle 2.$ $\bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $11.q \ \wedge\ q.rival = i$

$\langle 3\rangle 1.$ $q@\{11\}$ — by $11.q$

$\langle 3\rangle 2.$ $side(q) = 1 - side(i)$ — by I1, $\langle 3\rangle 1$, and $q.rival = i$

$\langle 3\rangle 3.$ $P[i] = 0$ — by I27, $\langle 2\rangle 1$, $\langle 3\rangle 2$, $\langle 3\rangle 1$, and $q.rival = i$

$\langle 3\rangle 4.$ $\bot$ — by $\langle 3\rangle 3$ and $\langle 2\rangle 2$

CASE: $14.i$

$\langle 3\rangle 1.$ $(i@\{15\})'$ — by $14.i$ and $\langle 2\rangle 2$

$\langle 3\rangle 2.$ $\bot$ — by $\langle 3\rangle 1$ and $\langle 2\rangle 3$

$\square$

The next two assertions follow from these *unless* assertions, the definition of a fair history, and the program text; (U1) is used to prove (L1) and (U2) is used to prove (L2).

$$i@\{12\} \ \wedge\ P[i] \geq 1 \ \mapsto\ i@\{13\} \tag{L1}$$

$$i@\{14\} \ \wedge\ P[i] = 2 \ \mapsto\ i@\{15\} \tag{L2}$$

The following assertions, which are stated without proof, follow directly from the definition of a fair history and the program text.

$$\begin{aligned} & i@\{12\} \ \wedge\ side(q) = 1 - side(i) \ \wedge\ q@\{5..11\} \ \mapsto \\ & \quad i@\{13\} \ \vee \\ & \quad i@\{12\} \ \wedge\ side(q) = 1 - side(i) \ \wedge\ q@\{12, 15\}) \end{aligned} \tag{L3}$$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{15..18\} \ \mapsto \\
& \quad i@\{13\} \ \vee \\
& \quad i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{18\}
\end{aligned} \tag{L4}
$$

Assertions (L5) through (L10), given next, easily follow from the preceding assertions. In particular, (I25) and (L4) imply that (L5) holds; (I25) and (L5) imply that (L6) holds; (L6) implies that (L7) holds; (L1) and (L7) imply that (L8) holds; (I18) implies that (L9) holds; and (L1) implies that (L10) holds.

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{15..18\} \ \mapsto \\
& \quad i@\{13\} \ \vee \\
& \quad i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{18\} \ \wedge \ q.rival = i
\end{aligned} \tag{L5}
$$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{15..19\} \ \mapsto \\
& \quad i@\{13\} \ \vee \\
& \quad i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{19\} \ \wedge \ q.rival = i
\end{aligned} \tag{L6}
$$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{15..19\} \ \mapsto \\
& \quad i@\{13\} \ \vee \\
& \quad i@\{12\} \ \wedge \ P[i] = 2
\end{aligned} \tag{L7}
$$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{15..19\} \ \mapsto \\
& \quad i@\{13\}
\end{aligned} \tag{L8}
$$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{12\} \ \mapsto \\
& \quad i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{12\} \ \wedge \ (P[i] \geq 1 \ \vee \ P[q] \geq 1)
\end{aligned} \tag{L9}
$$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{12\} \ \wedge \ (P[i] \geq 1 \ \vee \ P[q] \geq 1) \ \mapsto \\
& \quad i@\{13\} \ \vee \\
& \quad i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{13\}
\end{aligned} \tag{L10}
$$

The next assertion, which is stated without proof, follows directly from the definition of a fair history and the program text.

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{13\} \ \mapsto \\
& \quad i@\{13\} \ \vee \\
& \quad i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{14, 15\}
\end{aligned} \tag{L11}
$$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{14\} \mapsto \\
& \quad i@\{13\} \ \vee \\
& \quad i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{15\}
\end{aligned} \tag{L12}
$$

By (L1) and (L2), (I29) implies that (L12) holds. $\square$

$$
\begin{aligned}
& i@\{12\} \ \wedge \ side(q) = 1 - side(i) \ \wedge \ q@\{5..15\} \ \mapsto \\
& \quad i@\{13\}
\end{aligned} \tag{L13}
$$

Assertions (L3), (L8), (L11), and (L12) imply that (L13) holds. $\square$

$$i@\{12\} \;\mapsto\; i@\{13\} \tag{L14}$$

(I15) implies that $i@\{12\} \;\Rightarrow\; P[i] = 2 \;\vee\; (\exists q :: side(q) = 1 - side(i) \;\wedge\; q@\{5..19\})$ holds. By (L1), (L8), and (L13), this implies that (L14) holds. $\square$

By proving assertions similar to (L3) through (L13), it is possible to establish (L15), given next, which is similar to (L14). For brevity, we omit the proof of (L15).

$$i@\{14\} \;\mapsto\; i@\{15\} \tag{L15}$$

Note that (L14) implies that the first busy-waiting loop of process i terminates, while (L15) implies that the second busy-waiting loop of i terminates. Thus, we conclude that the algorithm in Figure 1 is free from starvation.

Correctness Proof for Fast Algorithm

In this section, we prove that the mutual exclusion and starvation-freedom properties hold for mutual exclusion algorithm of Figure 3. We first prove five invariants that are needed to prove that mutual exclusion holds. The first three are quite simple: (I30) follows from the mutual exclusion property of the algorithm in Figure 1, (I31) follows directly from the program text, and (I32) follows from (I30).

(I30) $(\mathbf{N}i :: i@\{15..27\}) \;\leq\; 1$

(I31) $i@\{6..12\} \;\Rightarrow\; B[i]$

(I32) $i@\{20..26\} \;\Rightarrow\; Z$

(I33)
$$\begin{array}{l} \wedge\; i.flag \\ \wedge\; \vee\; (i@\{22,23\} \;\wedge\; i.n > p) \\ \quad\;\; \vee\; (i@\{24\} \;\wedge\; i.n \geq p) \\ \Rightarrow \neg p@\{7..12\} \end{array}$$

$\langle 1\rangle 1.$ $\mathit{Init} \;\Rightarrow\; p@\{0\}$
$\qquad\quad \Rightarrow\; \mathrm{I33}$

$\langle 1\rangle 2.$ $\mathrm{I33} \;\Rightarrow\; \mathrm{I33}'$

ASSUME: $\mathrm{I33} \;\wedge\; \neg\mathrm{I33}'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $i.flag'$ — by $\neg\mathrm{I33}'$

$\langle 2\rangle 2.$ $\vee\; ((i@\{22,23\})' \;\wedge\; i.n' > p)$
$\qquad \vee\; ((i@\{24\})' \;\wedge\; i.n' \geq p)$ — by $\neg\mathrm{I33}'$

$\langle 2\rangle 3.$ $(p@\{7..12\})'$ — by $\neg\mathrm{I33}'$

$\langle 2\rangle 4.$ $\vee\ 6.p\ \wedge\ \neg Z$ by I33 $\wedge\ \neg$I33$'$

$\vee\ 20.i$

$\vee\ 21.i$

$\vee\ 23.i\ \wedge\ i.n = p\ \wedge\ \neg B[i.n]$

CASE: $6.p\ \wedge\ \neg Z$

$\langle 3\rangle 1.$ $\neg i@\{20..26\}$ by I32 and $\neg Z$

$\langle 3\rangle 2.$ $\neg(i@\{20..26\})'$ by $6.p$ and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\perp$ by $\langle 3\rangle 2$ and $\langle 2\rangle 2$

CASE: $20.i$

$\langle 3\rangle 1.$ $(i@\{21\})'$ by $20.i$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 2$

CASE: $21.i$

$\langle 3\rangle 1.$ $(i@\{22\})'\ \wedge\ i.n' = 0$ by $21.i$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$, $\langle 2\rangle 2$, and $p \geq 0$

CASE: $23.i\ \wedge\ i.n = p\ \wedge\ \neg B[i.n]$

$\langle 3\rangle 1.$ $\neg B[p]$ by $i.n = p\ \wedge\ \neg B[i.n]$

$\langle 3\rangle 2.$ $\neg p@\{6..12\}'$ by I31 and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\neg(p@\{6..12\})'$ by $23.i$ and $\langle 3\rangle 2$

$\langle 3\rangle 4.$ $\perp$ by $\langle 3\rangle 3$ and $\langle 2\rangle 3$

$\square$

(I34) $\wedge\ i@\{25\}$

$\wedge\ i.flag$

$\Rightarrow (\forall p :: \neg p@\{7..12\})$

$\langle 1\rangle 1.$ $Init\ \Rightarrow\ i@\{0\}$

$\Rightarrow$ I34

$\langle 1\rangle 2.$ I34 $\Rightarrow$ I34$'$

ASSUME: I34 $\wedge\ \neg$I34$'$

PROVE: $\perp$

$\langle 2\rangle 1.$ $(i@\{25\})'$ by $\neg$I34$'$

$\langle 2\rangle 2.$ $i.flag'$ by $\neg$I34$'$

$\langle 2\rangle 3.$ $(\exists p :: (p@\{7..12\})')$ by $\neg$I34$'$

$\langle 2\rangle 4.$ $\vee\ 6.p\ \wedge\ \neg Z$ by I34 $\wedge\ \neg$I34$'$

$\vee\ 20.i$

$\vee\ 22.i\ \wedge\ i.n \geq N$

CASE: $6.p\ \wedge\ \neg Z$

$\langle 3\rangle 1.$ $\neg i@\{20..26\}$ by I32 and $\neg Z$

$\langle 3\rangle 2.$ $\neg(i@\{20..26\})'$ by $6.p$ and $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $\perp$ by $\langle 3\rangle 2$ and $\langle 2\rangle 1$

CASE: $20.i$

$\langle 3\rangle 1.$ $(i@\{21\})'$ by $20.i$

$\langle 3\rangle 2.$ $\perp$ by $\langle 3\rangle 1$ and $\langle 2\rangle 1$

CASE: $22.i \wedge i.n \geq N$

$\langle 3 \rangle 1.$ $i@\{22\}$ — by $22.i$

$\langle 3 \rangle 2.$ $(\forall p :: i.n > p)$ — by $i.n \geq N$

$\langle 3 \rangle 3.$ $i.flag$ — by $22.i$ and $\langle 2 \rangle 2$

$\langle 3 \rangle 4.$ $(\forall p :: \neg p@\{7..12\})$ — by I33, $\langle 3 \rangle 3$, $\langle 3 \rangle 1$, and $\langle 3 \rangle 2$

$\langle 3 \rangle 5.$ $(\forall p :: \neg(p@\{7..12\})')$ — by $22.i$ and $\langle 3 \rangle 4$

$\langle 3 \rangle 6.$ $\perp$ — by $\langle 3 \rangle 5$ and $\langle 2 \rangle 3$

□

The following assertion implies that the mutual exclusion property holds for the fast entry section.

$$
\begin{aligned}
\text{(I35)} \quad & \wedge \; ((\mathbf{N}i :: i@\{2\} \wedge X = i \wedge Y = -1) + \\
& \quad (\mathbf{N}i :: i@\{3\} \wedge X = i) + \\
& \quad (\mathbf{N}i :: i@\{4\} \wedge X = i \wedge Y = i) + \\
& \quad (\mathbf{N}i :: i@\{5..7\} \wedge Y = i) + \\
& \quad (\mathbf{N}i :: i@\{8..11\})) \qquad \leq 1 \\
& \wedge \; ((\exists p :: p@\{8..11\}) \Rightarrow Y \neq -1)
\end{aligned}
$$

$\langle 1 \rangle 1.$ $Init \Rightarrow (\forall i :: i@\{0\})$

$\quad \Rightarrow$ I35

$\langle 1 \rangle 2.$ I35 $\Rightarrow$ I35$'$

ASSUME: I35 $\wedge\ \neg$I35$'$

PROVE: $\perp$

$\langle 2 \rangle 1.$ — by $\neg$I35$'$

$$
\begin{aligned}
\vee \; & ((\mathbf{N}i :: (i@\{2\})' \wedge X' = i \wedge Y' = -1) + \\
& (\mathbf{N}i :: (i@\{3\})' \wedge X' = i) + \\
& (\mathbf{N}i :: (i@\{4\})' \wedge X' = i \wedge Y' = i) + \\
& (\mathbf{N}i :: (i@\{5..7\})' \wedge Y' = i) + \\
& (\mathbf{N}i :: (i@\{8..11\})')) \qquad > 1 \\
\vee \; & \wedge (\exists p :: (p@\{8..11\})') \\
& \wedge Y' = -1
\end{aligned}
$$

$\langle 2 \rangle 2.$ $(\mathbf{N}i :: X' = i) \leq 1$ — because process identifiers are unique.

$\langle 2 \rangle 3.$ — by $\langle 2 \rangle 2$

$$
\begin{aligned}
& ((\mathbf{N}i :: (i@\{2\})' \wedge X' = i \wedge Y' = -1) + \\
& (\mathbf{N}i :: (i@\{3\})' \wedge X' = i) + \\
& (\mathbf{N}i :: (i@\{4\})' \wedge X' = i \wedge Y' = i)) \qquad \leq 1
\end{aligned}
$$

$\langle 2 \rangle 4.$ — by I35 $\wedge\ \neg$I35$'$

$$
\begin{aligned}
& \vee\; 1.q \wedge Y = -1 \wedge ((\exists p :: p@\{8..11\}) \Rightarrow Y \neq -1) \\
& \vee\; 7.q \wedge Y = q \\
& \vee\; 11.q \\
& \vee\; 25.q \wedge q.flag
\end{aligned}
$$

CASE: $1.q \wedge Y = -1 \wedge ((\exists p :: p@\{8..11\}) \Rightarrow Y \neq -1)$

$\langle 3 \rangle 1.$ $(\forall p :: \neg p@\{8..11\})$ — by $Y = -1$ and $((\exists p :: p@\{8..11\}) \Rightarrow Y \neq -1)$

$\langle 3 \rangle 2.$ $(\forall p :: \neg(p@\{8..11\})')$ — by $1.q$ and $\langle 3 \rangle 1$

$\langle 3 \rangle 3.$ $Y' = -1$ — by $1.q$ and $Y = -1$

$\langle 3\rangle 4.$ $(\mathbf{N}i :: (i@\{5..7\})' \ \wedge \ Y' = i) = 0$ by $\langle 3\rangle 3$

$\langle 3\rangle 5.$ $\perp$ by $\langle 2\rangle 1$, $\langle 2\rangle 3$, $\langle 3\rangle 4$, and $\langle 3\rangle 2$

CASE: $7.q \ \wedge \ Y = q$

$\langle 3\rangle 1.$ $Y' = q$ by $7.q$ and $Y = q$

$\langle 3\rangle 2.$
$$((\mathbf{N}i :: (i@\{2\})' \ \wedge \ X' = i \ \wedge \ Y' = -1)+ \\ (\mathbf{N}i :: (i@\{3\})' \ \wedge \ X' = i)+ \\ (\mathbf{N}i :: (i@\{4\})' \ \wedge \ X' = i \ \wedge \ Y' = i)+ \\ (\mathbf{N}i :: (i@\{5..7\})' \ \wedge \ Y' = i)+ \\ (\mathbf{N}i :: (i@\{8..11\})')) \qquad \leq 1$$

by I35 and that $7.q$ decrements $(\mathbf{N}i :: i@\{5..7\} \ \wedge \ Y = i)$ by one when it increments $(\mathbf{N}i :: i@\{8..11\})$ by one.

$\langle 3\rangle 3.$ $\perp$ by $\langle 2\rangle 1$, $\langle 3\rangle 1$, and $\langle 3\rangle 2$

CASE: $11.q$

$\langle 3\rangle 1.$ $Y' = -1$ by $11.q$

$\langle 3\rangle 2.$ $(\mathbf{N}i :: (i@\{5..7\})' \ \wedge \ Y' = i) = 0$ by $\langle 3\rangle 1$

$\langle 3\rangle 3.$ $(\mathbf{N}i :: (i@\{8..11\})') = 0$ by I35 and that $11.q$ decrements $(\mathbf{N}i :: i@\{8..11\})$ by one.

$\langle 3\rangle 4.$ $\perp$ by $\langle 2\rangle 1$, $\langle 2\rangle 3$, $\langle 3\rangle 2$, and $\langle 3\rangle 3$

CASE: $25.q \ \wedge \ q.flag$

$\langle 3\rangle 1.$ $q@\{25\}$ by $25.q$

$\langle 3\rangle 2.$ $(\forall p :: \neg p@\{7..12\})$ by I34, $\langle 3\rangle 1$, and $q.flag$

$\langle 3\rangle 3.$ $Y' = -1$ by $25.q$ and $q.flag$

$\langle 3\rangle 4.$ $(\mathbf{N}i :: (i@\{5..7\})' \ \wedge \ Y' = i) = 0$ by $\langle 3\rangle 3$

$\langle 3\rangle 5.$ $\perp$ by $\langle 2\rangle 1$, $\langle 2\rangle 3$, $\langle 3\rangle 4$, and $\langle 3\rangle 2$

$\Box$

ENTRY_2 and EXIT_2 satisfy the following properties.

(I36) $((\mathbf{N}i :: i@\{8..10\}) \leq 1 \ \wedge \ (\mathbf{N}i :: i@\{15..27\}) \leq 1) \ \Rightarrow \ (\mathbf{N}i :: i@\{9, 16..26\}) \leq 1$

$$i@\{8\} \ \mapsto \ i@\{9\} \qquad \text{(L16)}$$

$$i@\{15\} \ \mapsto \ i@\{16\} \qquad \text{(L17)}$$

The proof of (I36) is similar to that of (I8), and is omitted for brevity. The proof of (L16) and (L17) are similar to that of (2SF), and are omitted for brevity. (Note that process identifiers and function *side* are used for convenience in the proof of (I8) and (2SF).)

The following two assertions imply that the mutual exclusion and starvation-freedom properties hold for the algorithm of Figure 3.

(I37) $(\mathbf{N}i :: i@\{9, 16\}) \leq 1$

(I30), (I35), and (I36) imply that (I37) holds. $\Box$

$$i@\{1..8, 14, 15\} \ \mapsto \ i@\{9, 16\} \qquad \text{(L18)}$$

By the program text, $i@\{1..7\} \mapsto i@\{8, 14\}$. (SF) implies that $i@\{14\} \mapsto i@\{15\}$. Hence, (L18) follows from (L16) and (L17). $\Box$

References

[1] A. Agarwal and M. Cherian, "Adaptive Backoff Synchronization Techniques", *Proceedings of the 16th International Symposium on Computer Architecture*, May, 1989, pp. 396-406.

[2] J. Anderson, "A Fine-Grained Solution to the Mutual Exclusion Problem", *Acta Informatica*, Vol. 30, No. 3, 1993, pp. 249-265.

[3] T. Anderson, "The Performance of Spin Lock Alternatives for Shared-Memory Multiprocessors", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 1, No. 1, January, 1990, pp. 6-16.

[4] BBN Advanced Computers, *Inside the TC2000 Computer*, February, 1990.

[5] K. Chandy and J. Misra, *Parallel Program Design: A Foundation*, Addison-Wesley, 1988.

[6] E. Dijkstra, "Solution of a Problem in Concurrent Programming Control", *Communications of the ACM*, Vol. 8, No. 9, 1965, pp. 569.

[7] G. Graunke and S. Thakkar, "Synchronization algorithms for shared-memory multiprocessors", *IEEE Computer*, Vol. 23, June, 1990, pp. 60-69.

[8] J. Kessels, "Arbitration Without Common Modifiable Variables", *Acta Informatica*, Vol. 17, 1982, pp. 135-141.

[9] L. Lamport, "A Fast Mutual Exclusion Algorithm", *ACM Transactions on Computer Systems*, Vol. 5, No. 1, February, 1987, pp. 1-11.

[10] L. Lamport, "How to Write a Proof", Research Report 94, Digital Equipment Corporation Systems Research Center, February, 1993.

[11] J. Mellor-Crummey and M. Scott, "Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors", *ACM Transactions on Computer Systems*, Vol. 9, No. 1, February, 1991, pp. 21-65.

[12] M. Michael and M. Scott, "Fast Mutual Exclusion, Even With Contention", Technical Report, University of Rochester, June, 1993.

[13] G. Peterson and M. Fischer, "Economical Solutions for the Critical Section Problem in a Distributed System", *Proceedings of the 9th ACM Symposium on Theory of Computing*, May, 1977, pp. 91-97.

[14] E. Styer, "Improving Fast Mutual Exclusion", *Proceedings of the Eleventh Annual ACM Symposium on Principles of Distributed Computing*, 1992, pp. 159-168.

[15] J. Yang and J. Anderson, "Fast, Scalable Synchronization with Minimal Hardware Support (Extended Abstract)", *Proceedings of the 12th Annual ACM Symposium on Principles of Distributed Computing*, August, 1993, pp. 171-182.

[16] J. Yang and J. Anderson, "Time Bounds for Mutual Exclusion and Related Problems", *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, May, 1994, pp. 224-233.