Complexity of Multiprocessor Blocking Analysis with Nested Critical Sections

Alexander Wieder

Björn Brandenburg

Max Planck Institute for Software Systems

Max Planck Institute for Software Systems

RTSS 2014

20141203

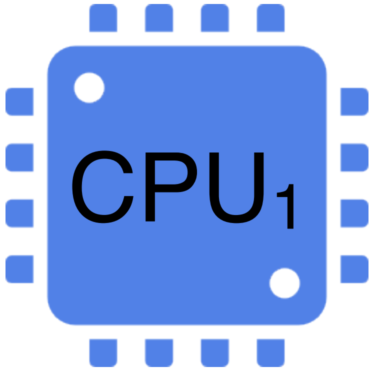
MAX-PLANCK-GESELLSCHAFT

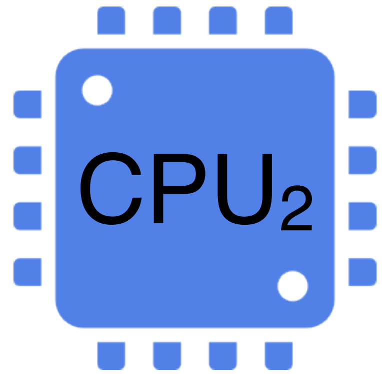
This talk

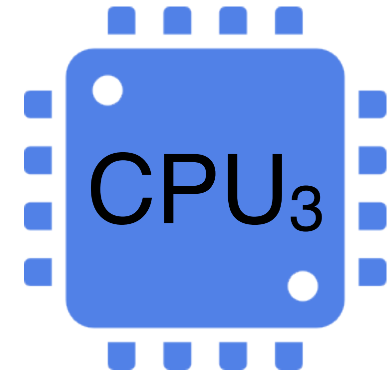
Complexity of tight blocking analysis for:

- **multiprocessor** systems
- locks with **strong ordering guarantees**
- **nested** critical sections

Nested Locks on Multiprocessors







T_1

```
lock(RED);
    .
unlock(RED);
```

T_2

```
lock(RED);
      .
      .
      .
      .
      .
      .
      .
      .
unlock(RED);
```

```
lock(RED);
   lock(GREEN);
        .
   unlock(GREEN);
unlock(RED);
```

T_3

```
lock(GREEN);
       .
       .
       .
       .
       .
unlock(GREEN);
```

Nested Locks on Multiprocessors

CPU_3

Critical section for **RED** contains nested critical section for **GREEN**.

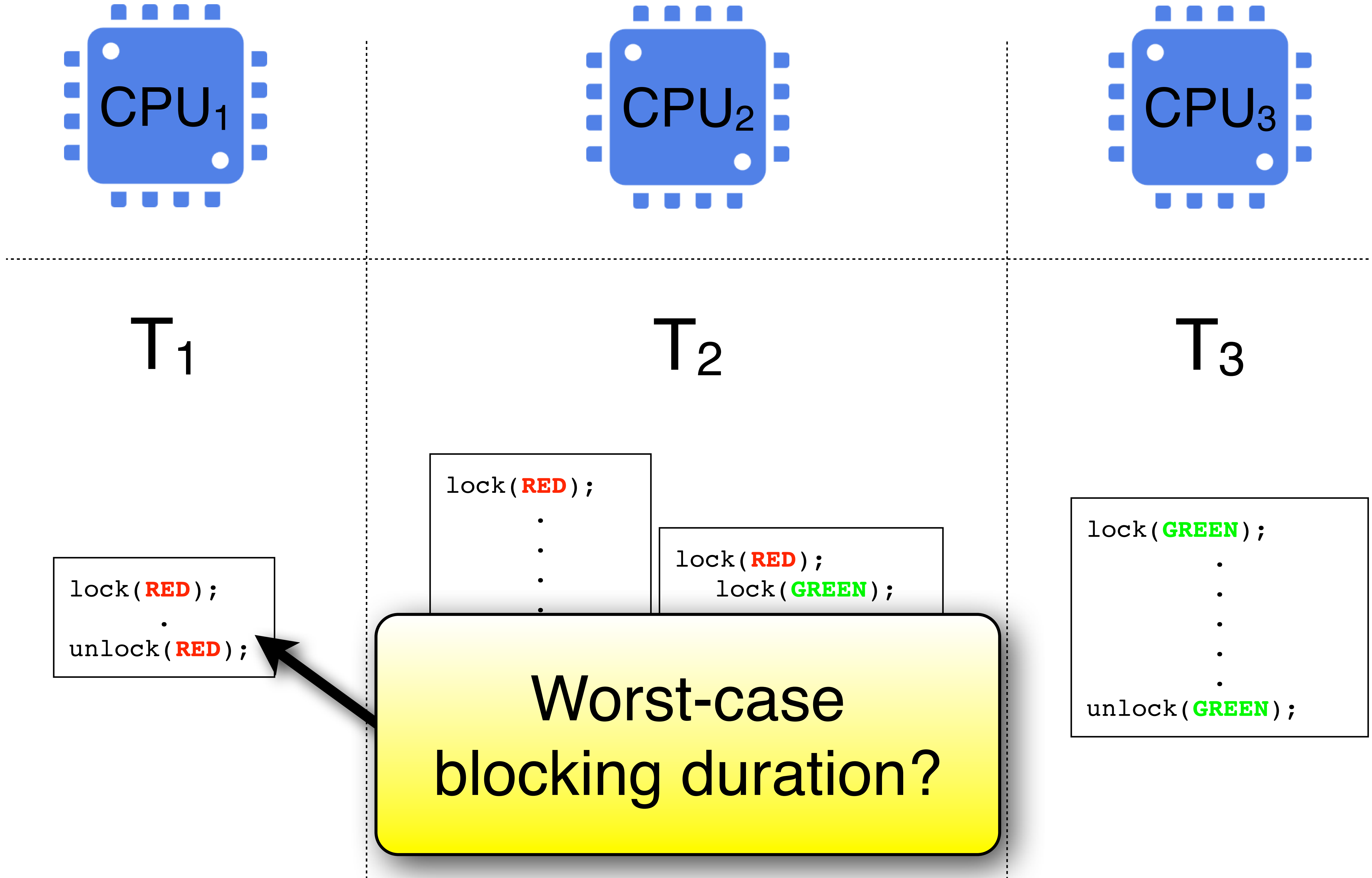
```
lock(RED);
    .
unlock(RED);
```

```
lock(RED);
      .
      .
      .
      .
      .
      .
      .
      .
unlock(RED);
```

```
lock(RED);
   lock(GREEN);
       .
   unlock(GREEN);
unlock(RED);
```

```
lock(GREEN);
      .
      .
      .
      .
      .
unlock(GREEN);
```

Nested Locks on Multiprocessors



The Blocking Analysis Problem

```
lock(RED);
    .
unlock(RED);
```

Worst-case blocking duration?

The Blocking Analysis Problem

```
lock(RED);
    .
unlock(RED);
```

Worst-case blocking duration?

Blocking Analysis Problem:

Bound the blocking duration that a task can incur in the worst case.

The Blocking Analysis Problem

```
lock(RED);
    .
unlock(RED);
```

Worst-case blocking duration?

Blocking Analysis Problem:

Bound the blocking duration that a task can incur in the worst case.

Tight Blocking Bounds:

There is a schedule in which the blocking bound is reached.

Main Result

multiprocessor system

locks with **strong ordering guarantees**

nested critical sections

Main Result

FIFO/priority ordering

multiprocessor system

locks with **strong ordering guarantees**

nested critical sections

Main Result

multiprocessor system

locks with **strong ordering guarantees**

nested critical sections

Blocking analysis
NP-hard!

Context: Mutex Locks

protect shared resources such as

- shared bus
- shared data structures
- peripheral devices

Context: Mutex Locks

ensure
mutual
exclusion

protect shared resources such as

- shared bus
- shared data structures
- peripheral devices

Context: Mutex Locks

ensure
mutual
exclusion

protect shared resources such as

- shared bus
- shared data structures
- peripheral devices

mandated by AUTOSAR, ...

multiprocessor system

locks with **strong ordering guarantees**

nested critical sections

Blocking analysis
NP-hard!

So what?
Formally, most interesting scheduling problems are hard!

Complexity of Scheduling-Related Problems

- Response-Time Analysis (Eisenbrand and Rothvoß, 2008)
- Deciding Periodic Task Set Feasibility (Leung and Whitehead, 1982)
- Scheduling Task Sets with Self-Suspensions (Ridouard et al., 2006)
- [...]

Complexity of Scheduling-Related Problems

- Response-Time Analysis (Eisenbrand and Rothvoß, 2008)
- Deciding Periodic Task Set Feasibility (Leung and Whitehead, 1982)
- Scheduling Task Sets with Self-Suspensions (Ridouard et al., 2006)
- [...]

NP-HARD

Complexity of Scheduling-Related Problems

- Response-Time Analysis (Eisenbrand and Rothvoß, 2008)
- Deciding Periodic Task Set Feasibility (Leung and Whitehead, 1982)
- Scheduling Task Sets with Self-Suspensions (Ridouard et al., 2006)
- [...]

NP-HARD

Blocking analysis
is a **seemingly**
much easier problem!

Blocking Analysis Complexity

for commonly used protocols

		Architecture	
		Uniprocessor	Multiprocessor
Nested Critical Sections	no		
	yes		

Blocking Analysis Complexity

for commonly used protocols

Architecture

Nested Critical Sections		Uniprocessor	Multiprocessor
	no	polynomial	
	yes	polynomial	

PCP: find longest critical section

Blocking Analysis Complexity

for commonly used protocols

LP-Based Analysis takes polynomial time

Architecture

		Uniprocessor	Multiprocessor
Nested Critical Sections	no	polynomial	polynomial
	yes	polynomial	

Blocking Analysis Complexity

for commonly used protocols

		Architecture	
		Uniprocessor	Multiprocessor
Nested Critical Sections	no	polynomial	polynomial
	yes	polynomial	

Blocking Analysis Complexity

for commonly used protocols

		Architecture	
		Uniprocessor	Multiprocessor
Nested Critical Sections	no	polynomial	polynomial
	yes	polynomial	**NP-hard**

Blocking Analysis Complexity

for commonly used protocols

Architecture

Nested Critical Sections		Uniprocessor	Multiprocessor
	no	polynomial	polynomial
	yes	polynomial	**NP-hard**

What makes it difficult on multiprocessors with nesting?

Outline

- Introduction ✓
- Intuition: Why does nesting make the analysis difficult?
- Reduction: From Multiple Choice Matching to Blocking Analysis
- Summary and Conclusion

What makes the blocking analysis difficult?

Nested Critical Sections		Architecture	
		Uniprocessor	Multiprocessor
	no	polynomial	polynomial
	yes	polynomial	**NP-hard**

FIFO-ordered locks:

processor-local problems:
greedy approach to determine worst case

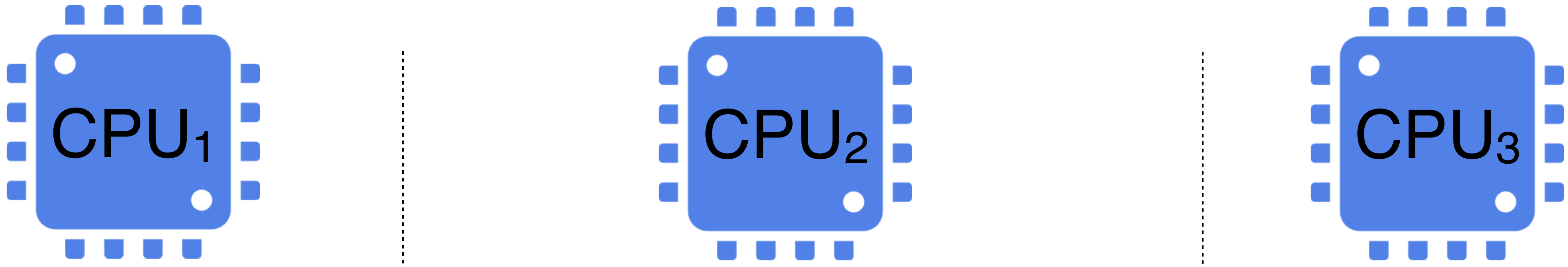
What makes the blocking analysis difficult?

		Architecture	
		Uniprocessor	Multiprocessor
Nested Critical Sections	no	polynomial	polynomial
	yes	polynomial	**NP-hard**

FIFO-ordered locks:

not a processor-local problem:
Consider **all critical sections from all processors** at once!

Non-locality in analysis of nested locks



T_1

```
lock(RED);
      .
unlock(RED);
```

T_2

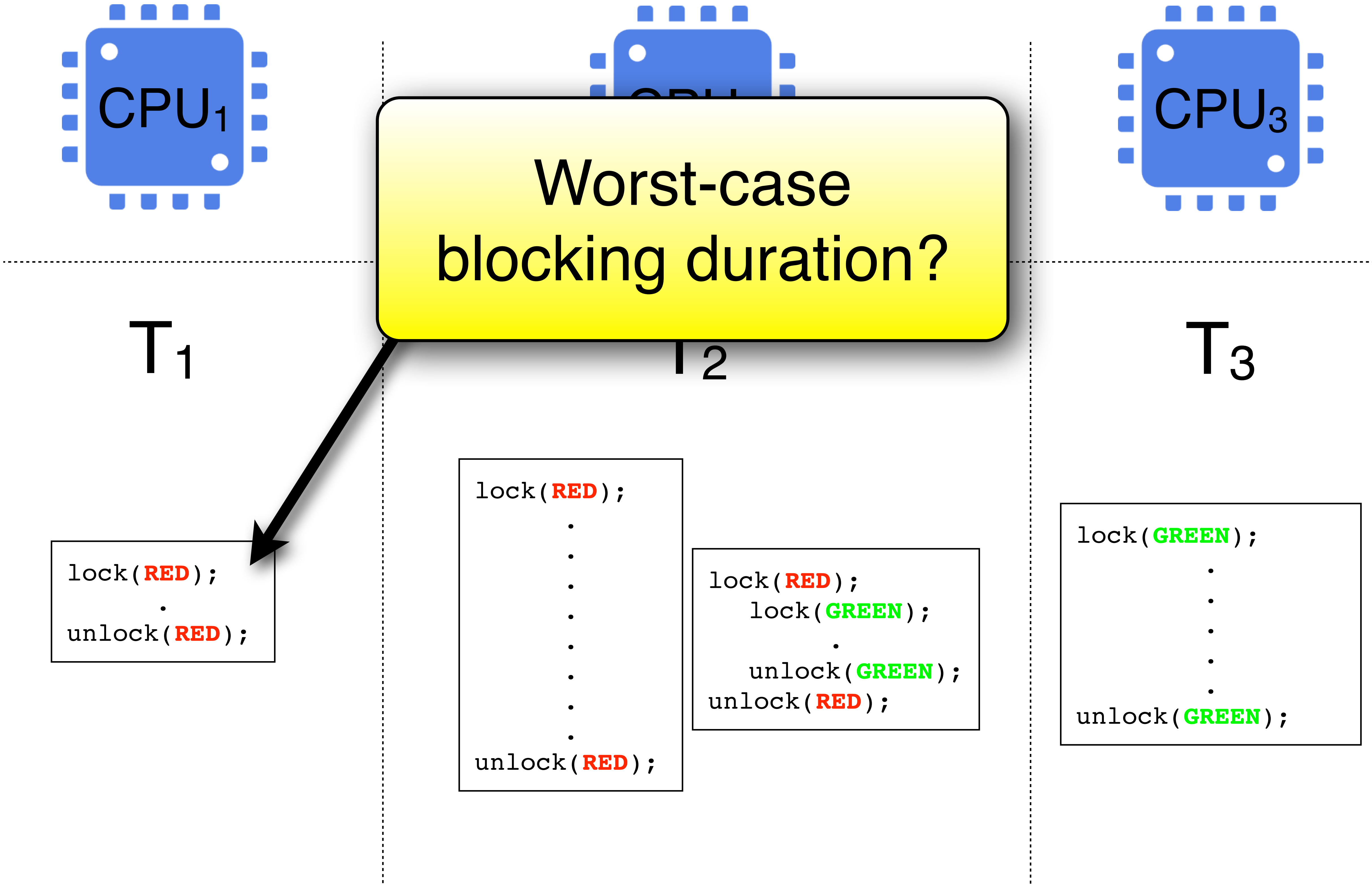
```
lock(RED);
        .
        .
        .
        .
        .
        .
        .
        .
unlock(RED);
```

```
lock(RED);
   lock(GREEN);
         .
   unlock(GREEN);
unlock(RED);
```

T_3

```
lock(GREEN);
         .
         .
         .
         .
         .
unlock(GREEN);
```

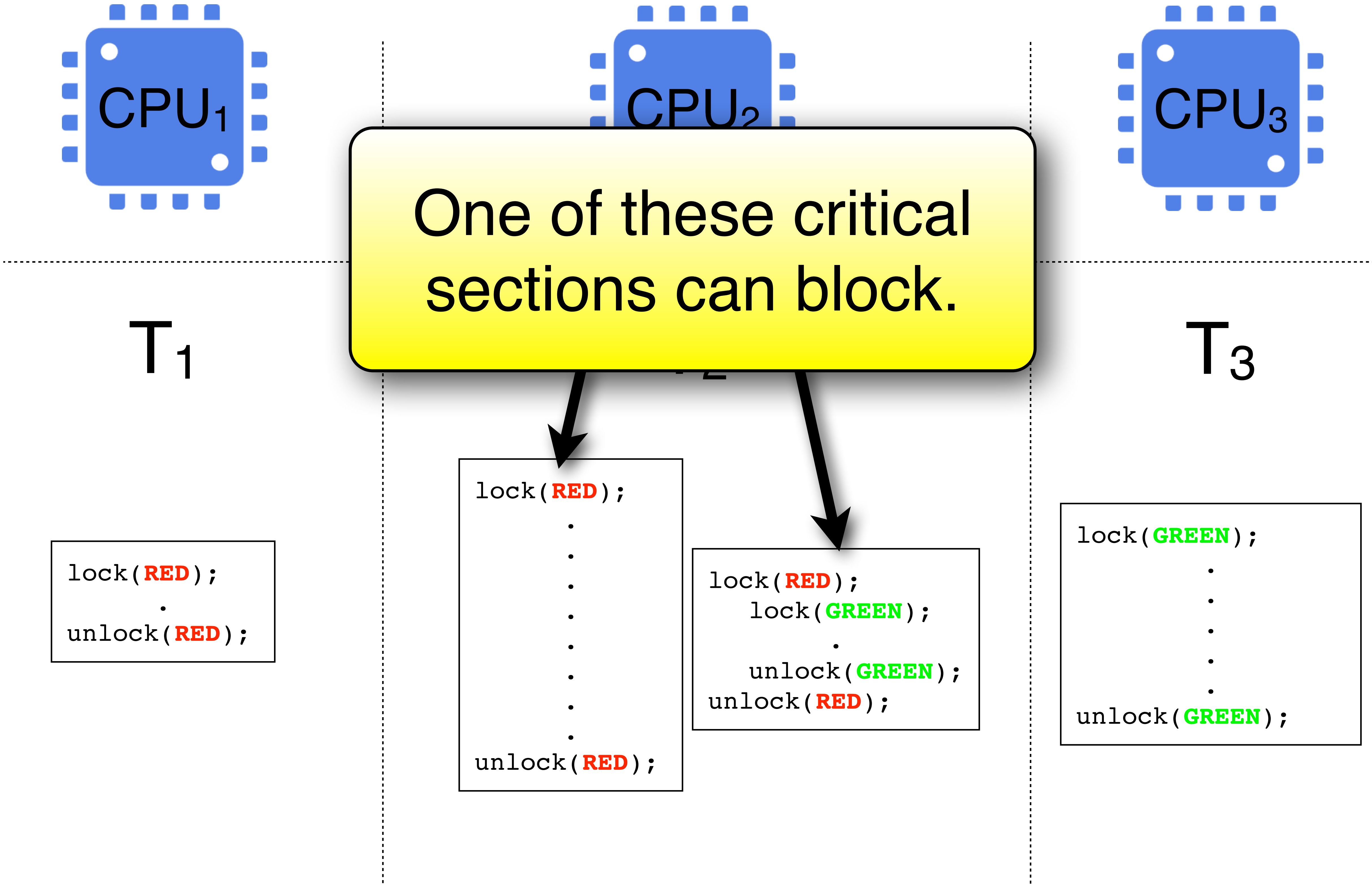
Non-locality in analysis of nested locks



FIFO-ordered spin locks

Each critical section can be blocked
by **at most one** critical section
for the same resource
from each remote processor.

Non-locality in analysis of nested locks



Non-locality in analysis of nested locks

CPU_1	CPU_2	CPU_3
T_1	T_2	T_3

T_1:

```
lock(RED);
      .
unlock(RED);
```

T_2:

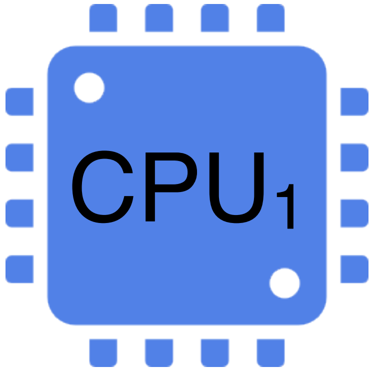
```
lock(RED);
        .
        .
        .
        .
        .
        .
        .
        .
unlock(RED);
```

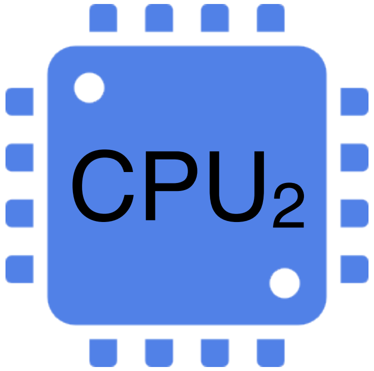
```
lock(RED);
   lock(GREEN);
          .
   unlock(GREEN);
unlock(RED);
```

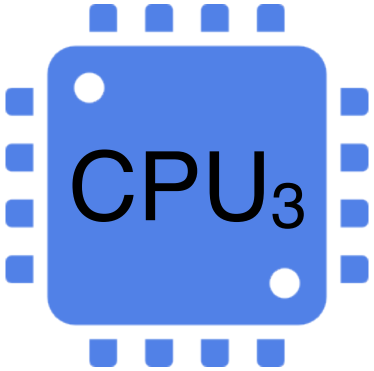
T_3:

```
lock(GREEN);
         .
         .
         .
         .
         .
unlock(GREEN);
```

Non-locality in analysis of nested locks







T_1

```
lock(RED);
      .
unlock(RED);
```

T_2

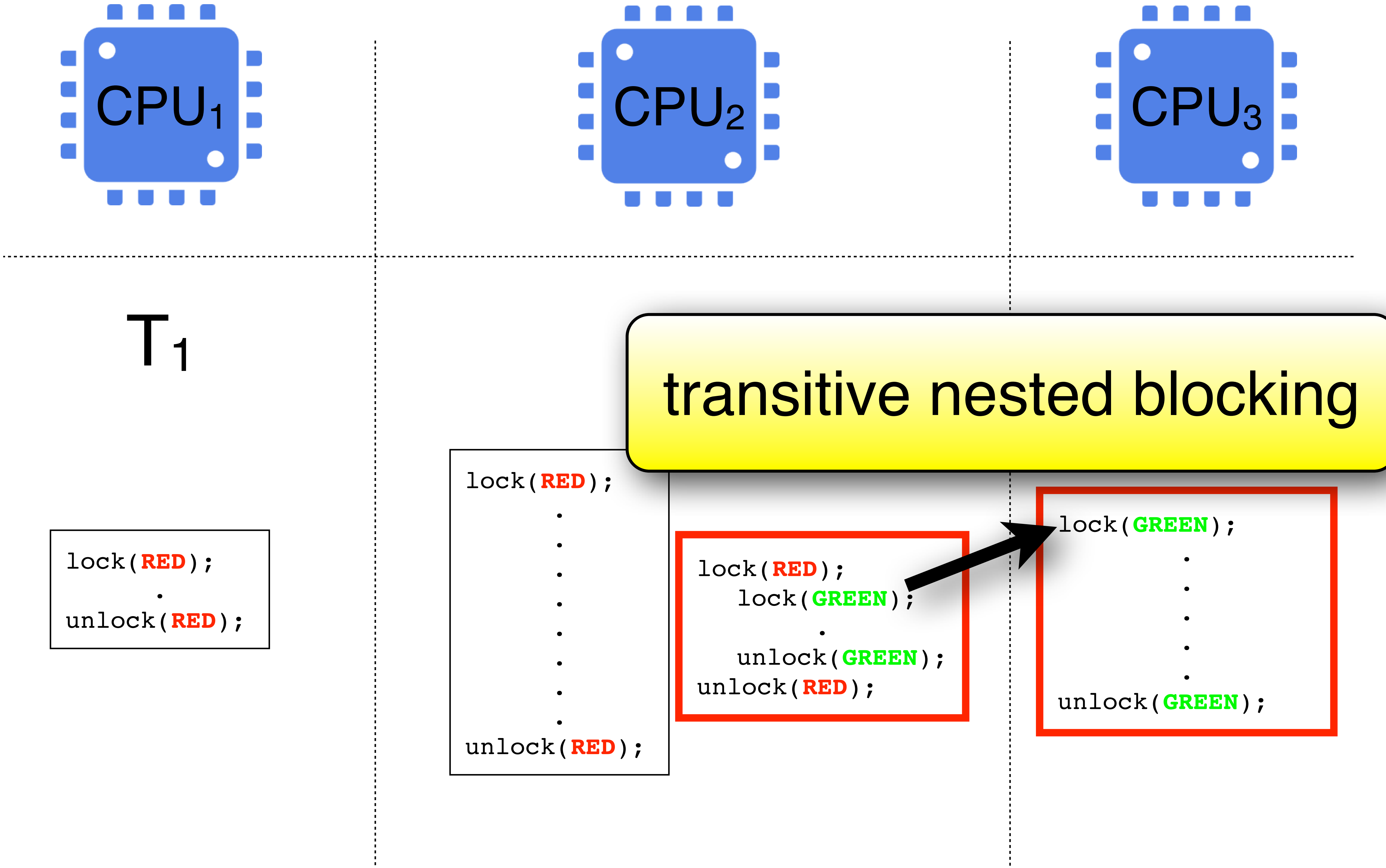
```
lock(RED);
       .
       .
       .
       .
       .
       .
       .
       .
unlock(RED);
```

```
lock(RED);
   lock(GREEN);
        .
   unlock(GREEN);
unlock(RED);
```

T_3

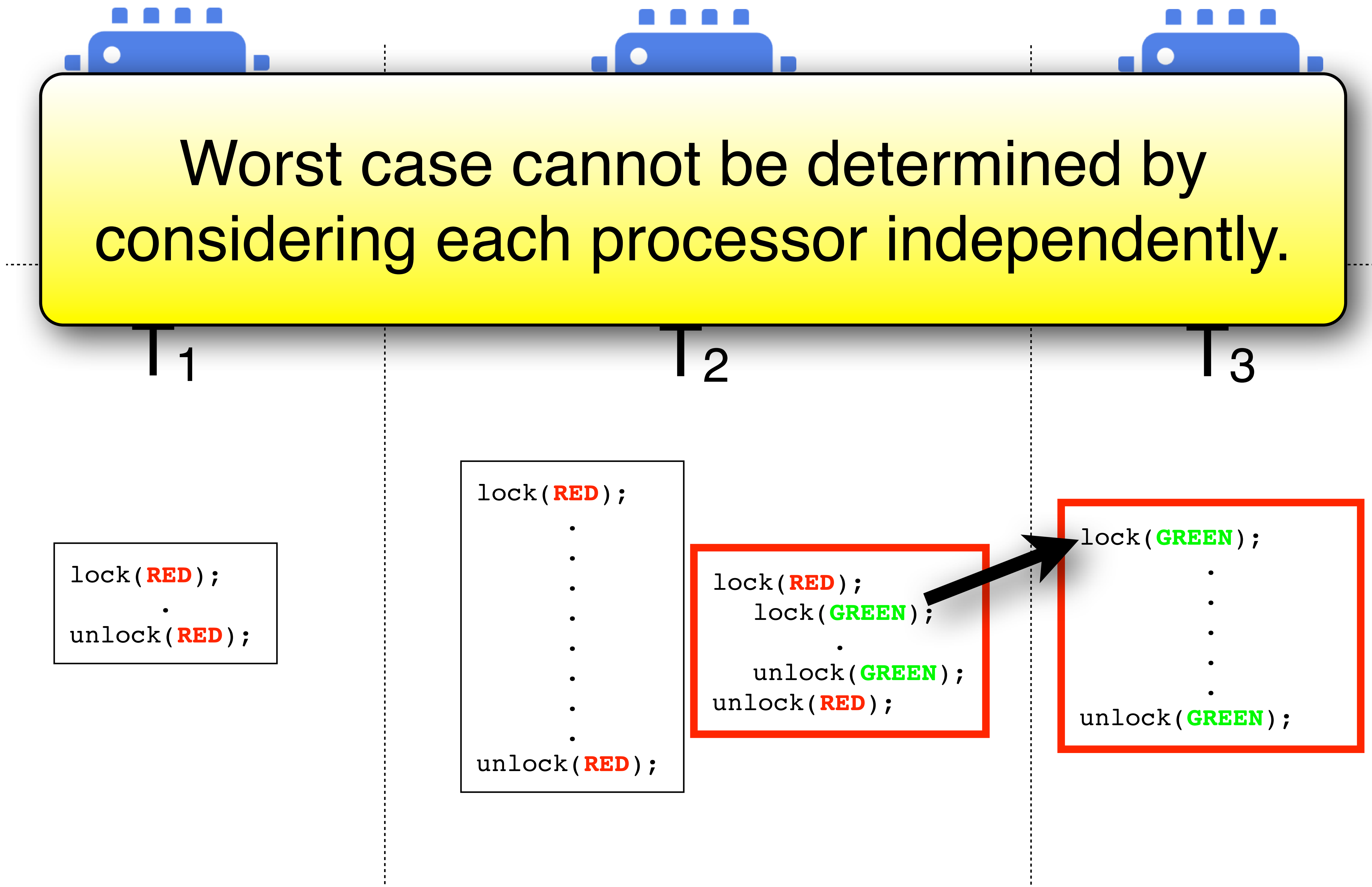
```
lock(GREEN);
       .
       .
       .
       .
       .
unlock(GREEN);
```

Non-locality in analysis of nested locks

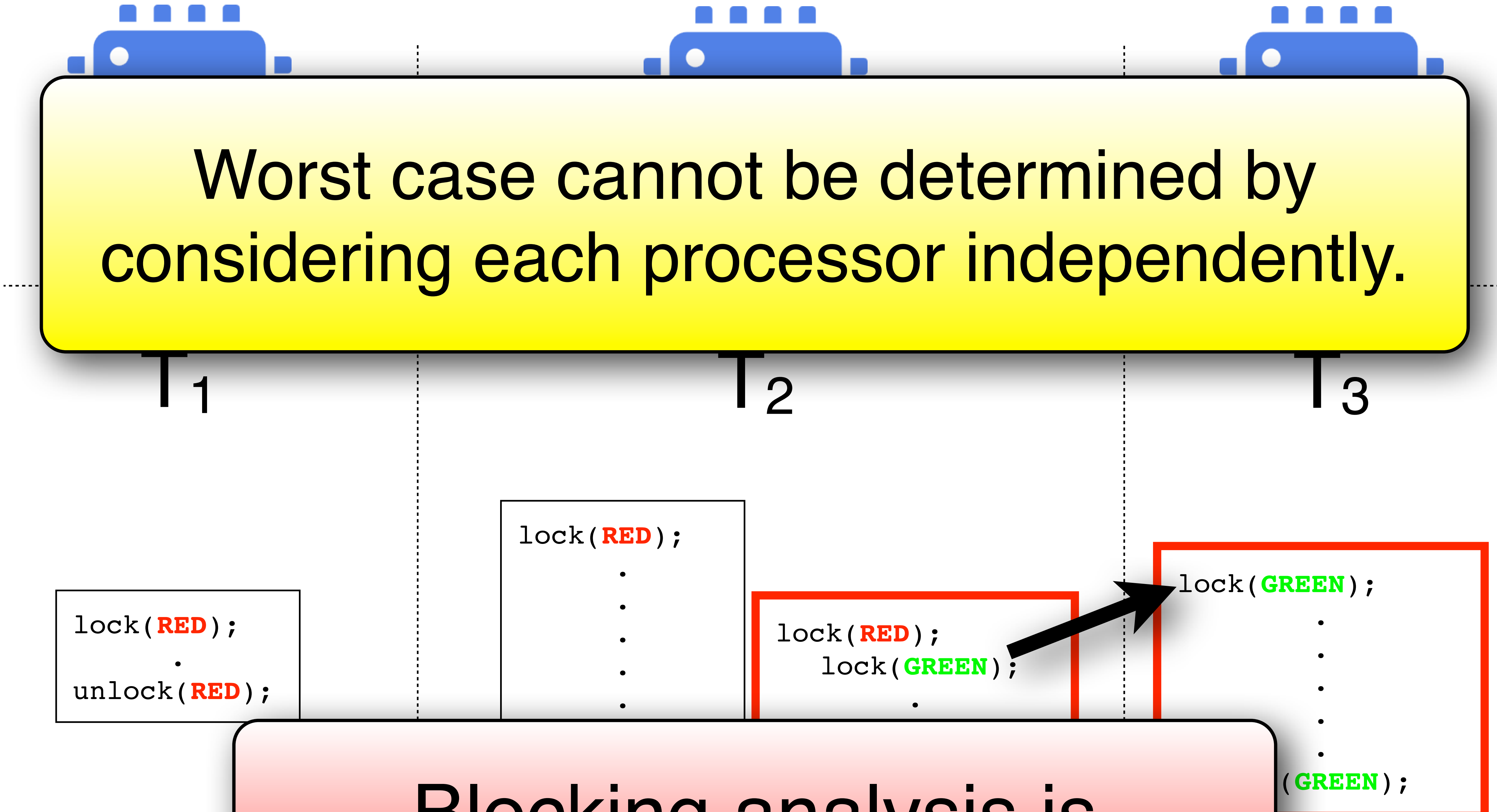


Non-locality in analysis of nested locks

Worst case cannot be determined by considering each processor independently.



Non-locality in analysis of nested locks



Worst case cannot be determined by considering each processor independently.

Blocking analysis is non-local problem!

Similarities to Matching

Blocking analysis is non-local problem!

Similarities to Matching

Blocking analysis is
non-local problem!

Blocking Analysis at least as hard as
Multiple Choice Matching

The Multiple-Choice Matching Problem

Input:

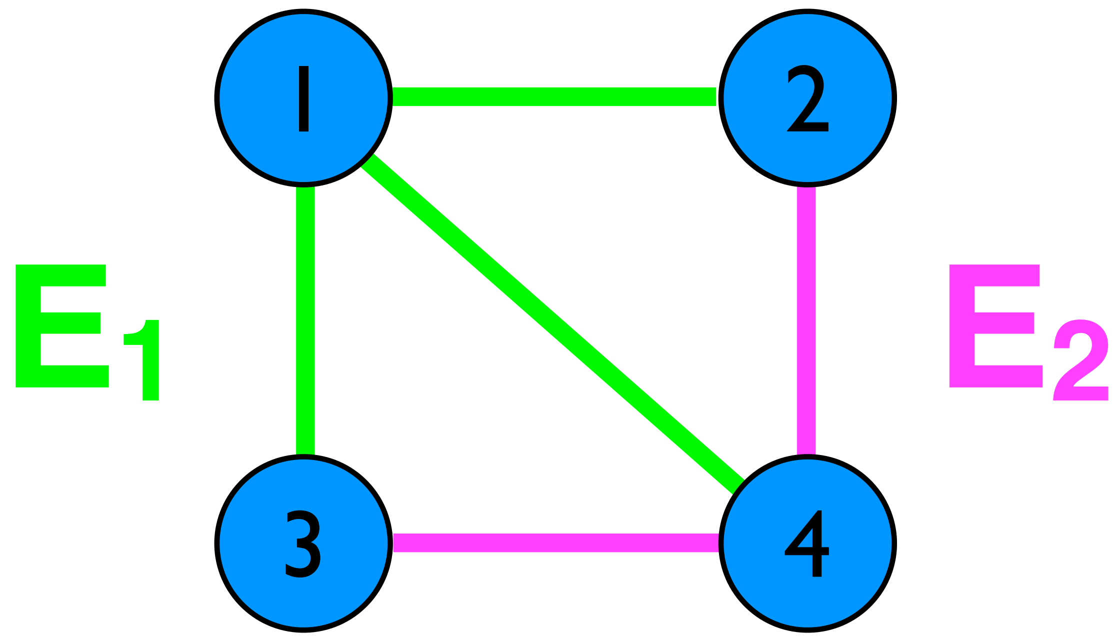
- graph **G=(V,E)**
- edge partitions $\mathbf{E_1,\ldots,E_t}$

Problem:

Is there a matching **F** that contains exactly one edge from each edge partition?

The Multiple-Choice Matching Problem

Input:

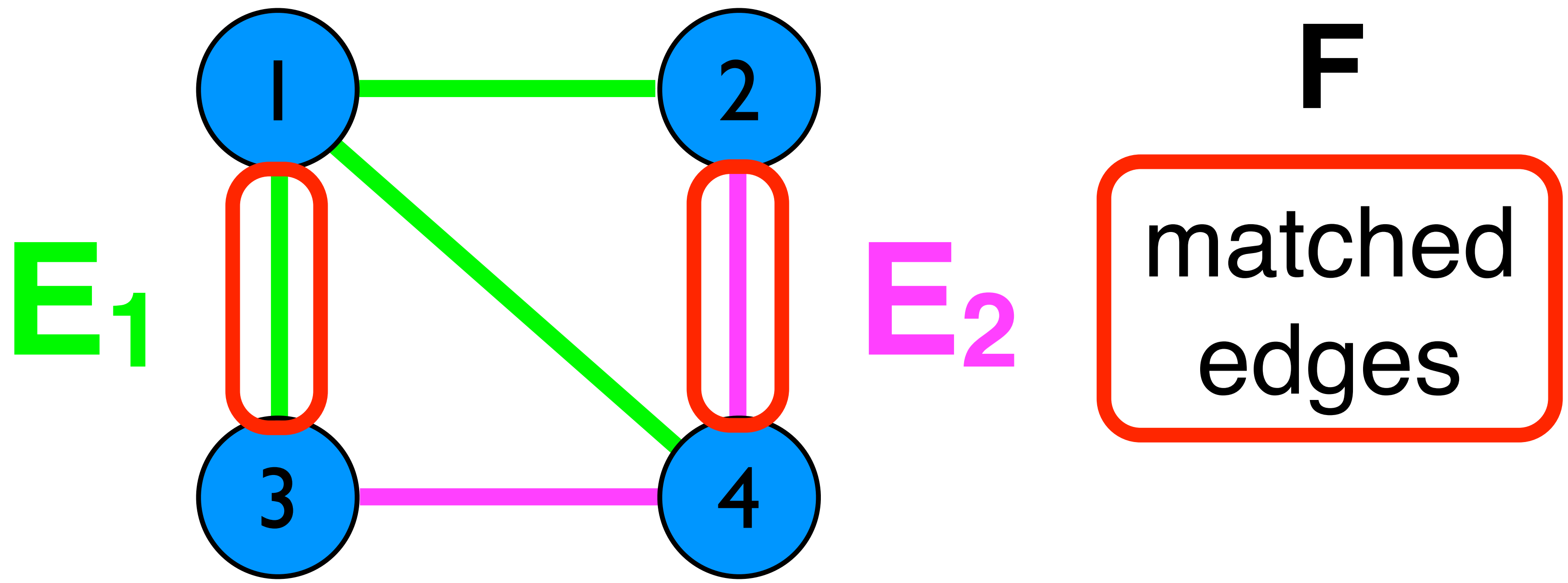


Problem:

Is there a matching **F** that contains exactly one edge from each edge partition?

The Multiple-Choice Matching Problem

Input:

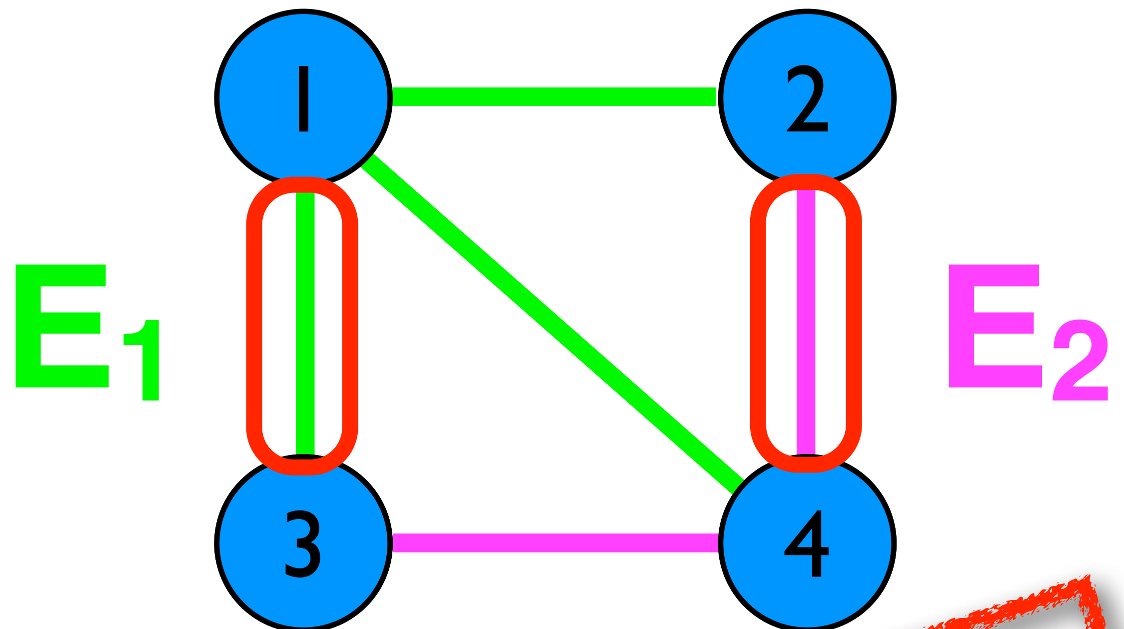


Problem:

Is there a matching **F** that contains exactly one edge from each edge partition?

The Multiple-Choice Matching Problem

Input:



F

matched edges

Problem:

Is there a matching **F** that contains exactly one edge from each edge partition?

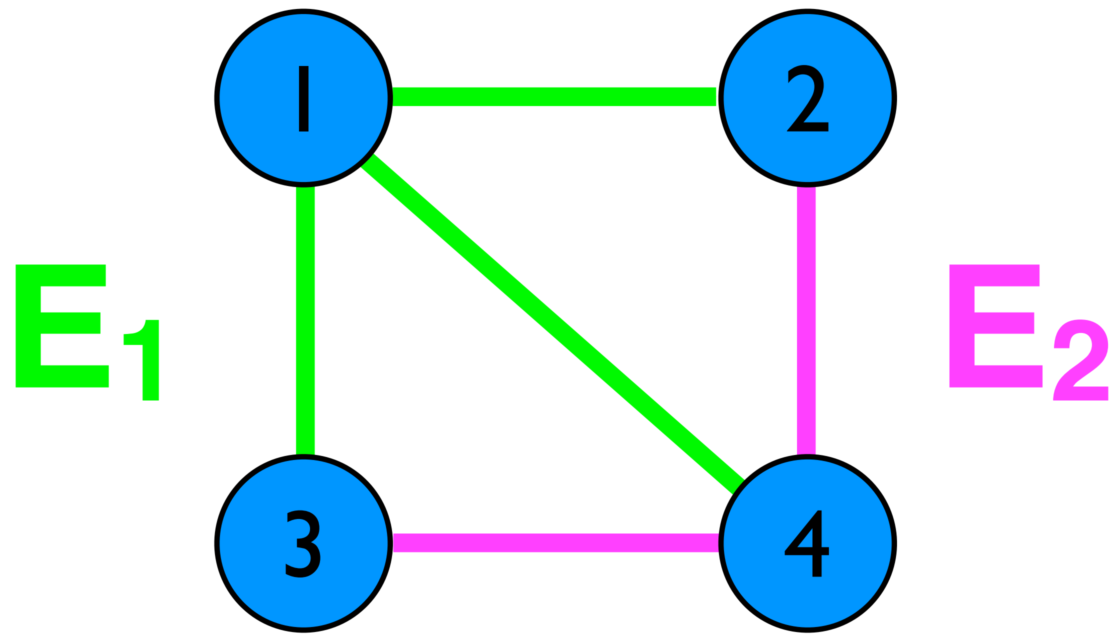
NP-HARD

Outline

- Introduction ✓
- Intuition: Why does nesting make the analysis difficult? ✓
- Reduction: From Multiple Choice Matching to Blocking Analysis
- Summary and Conclusion

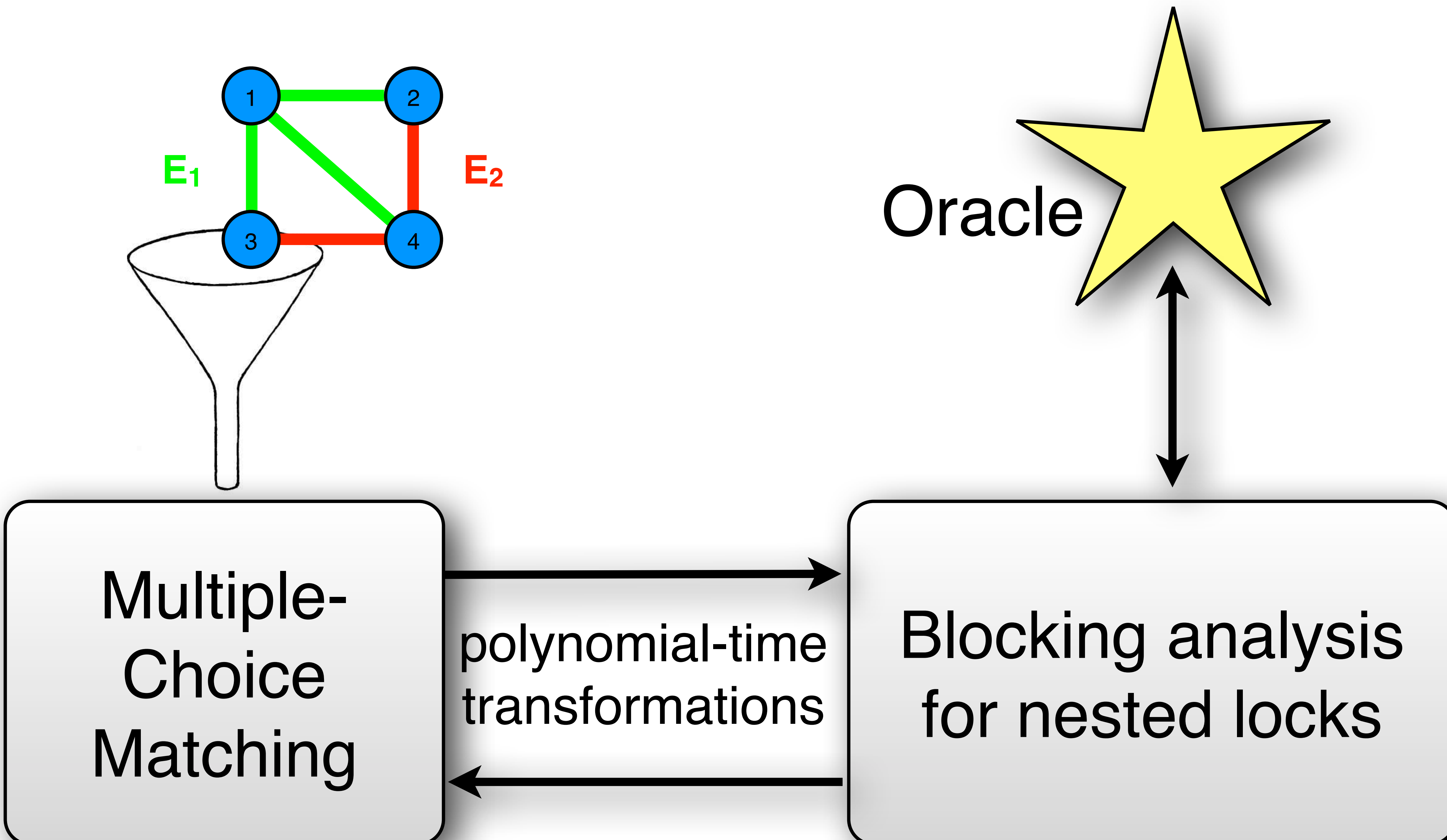
The Multiple-Choice Matching Problem

Input:

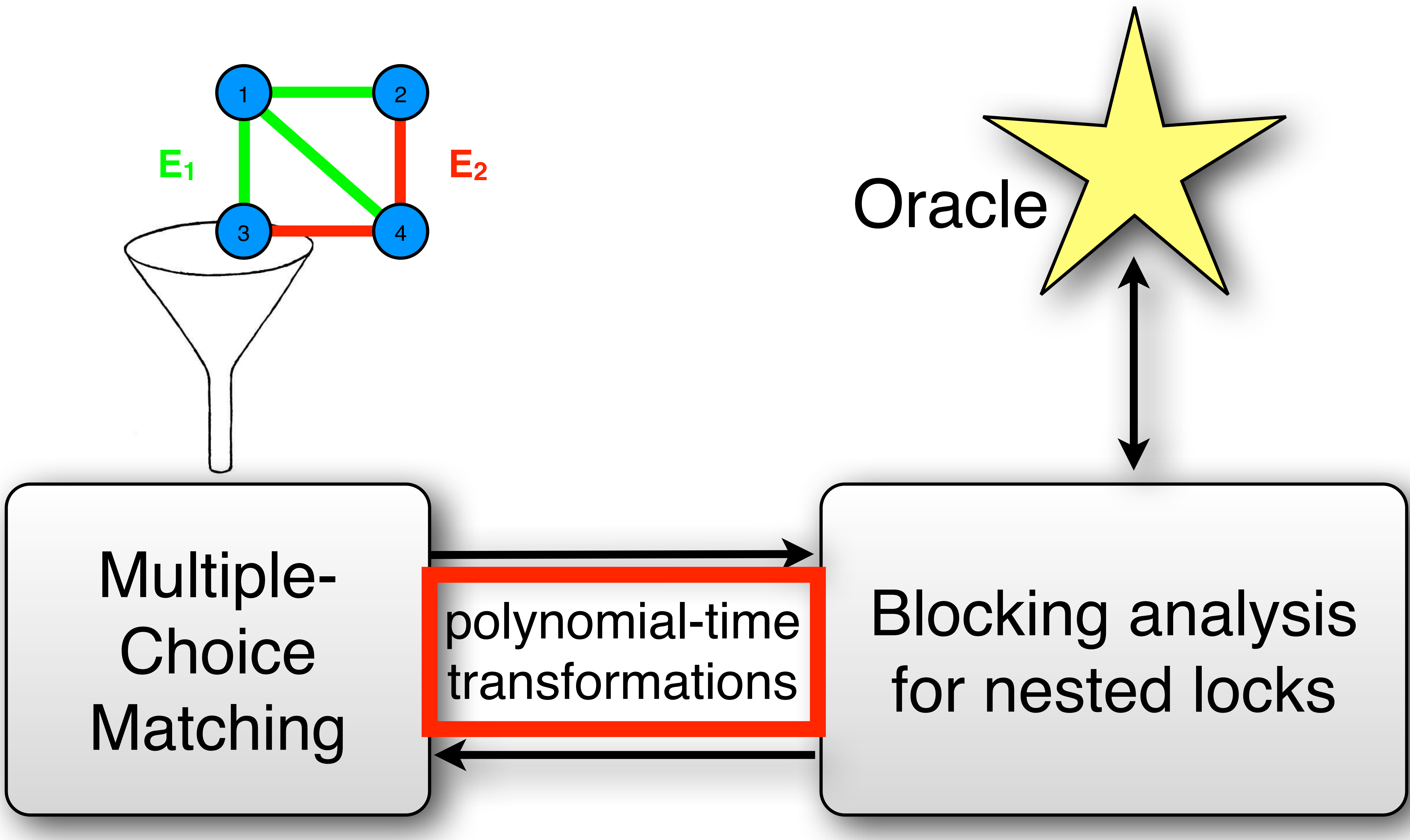


Problem:
Is there a matching **F** that contains exactly one edge from each edge partition?

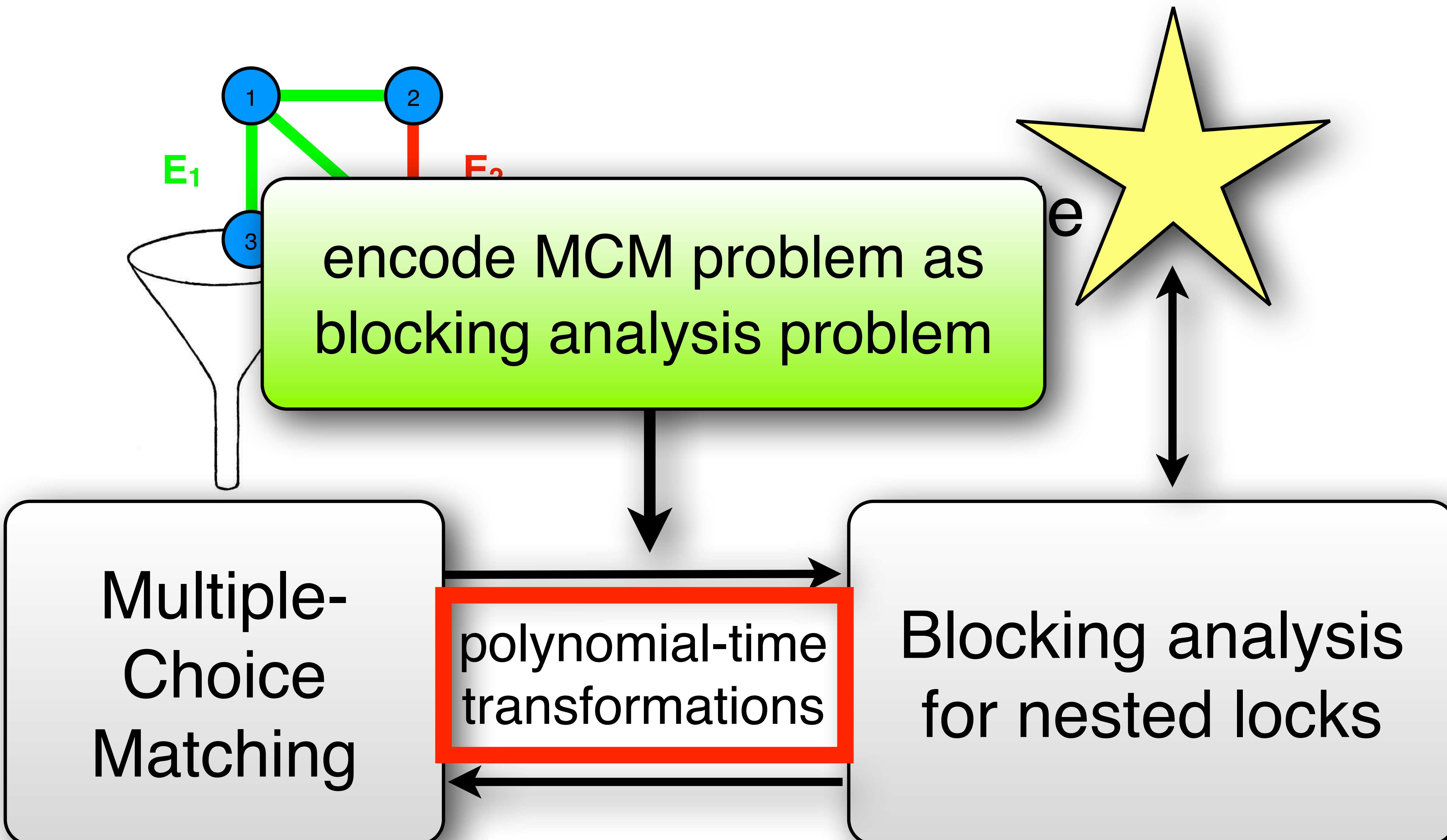
Reduction Overview



Reduction Overview



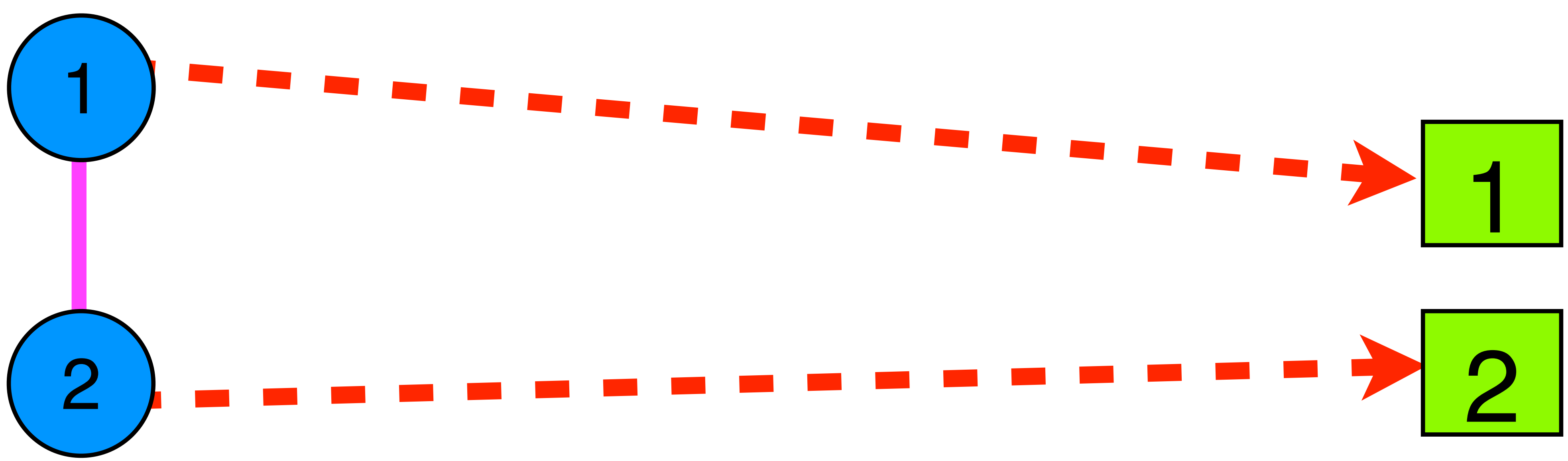
Reduction Overview



Encoding Vertices as Resources

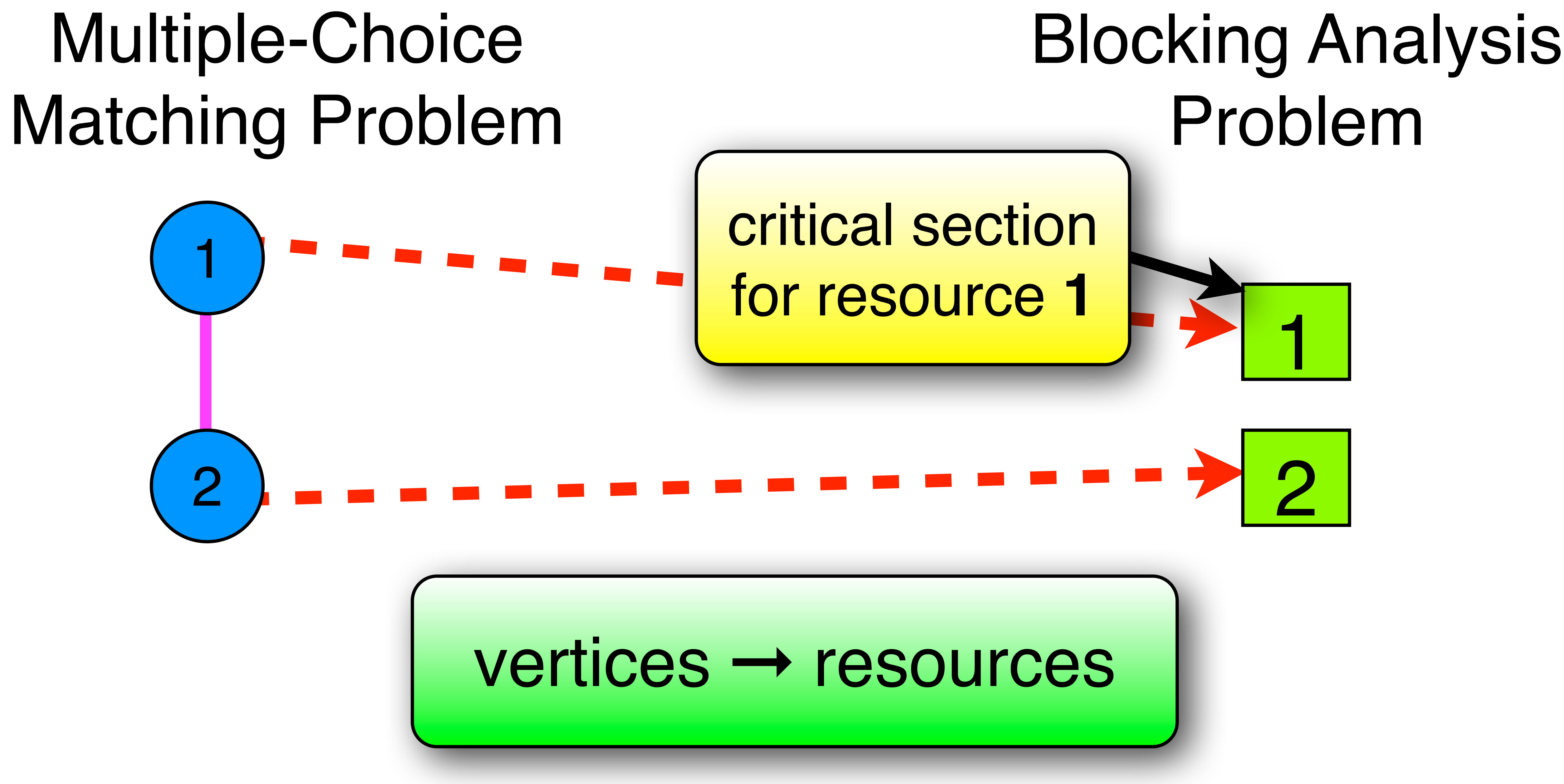
Multiple-Choice Matching Problem

Blocking Analysis Problem



vertices → resources

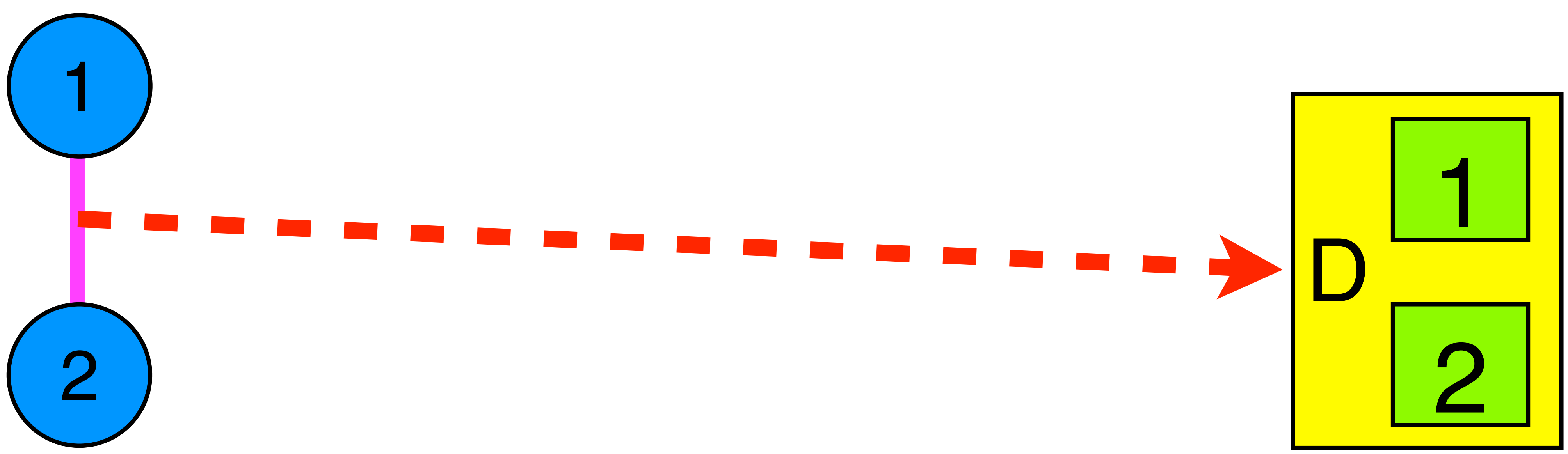
Encoding Vertices as Resources



Encoding Edges as Nested Critical Sections

Multiple-Choice Matching Problem

Blocking Analysis Problem

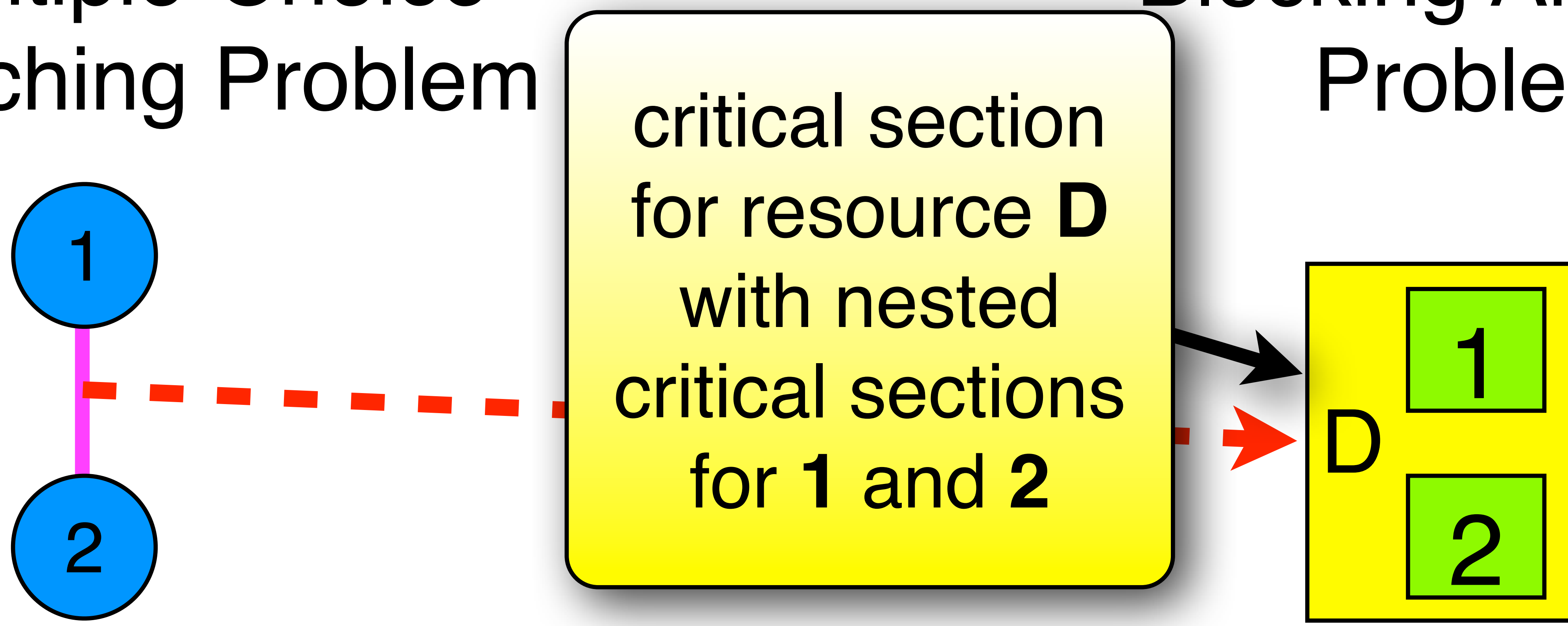


edges → outer critical sections

Encoding Edges as Nested Critical Sections

Multiple-Choice Matching Problem

Blocking Analysis Problem

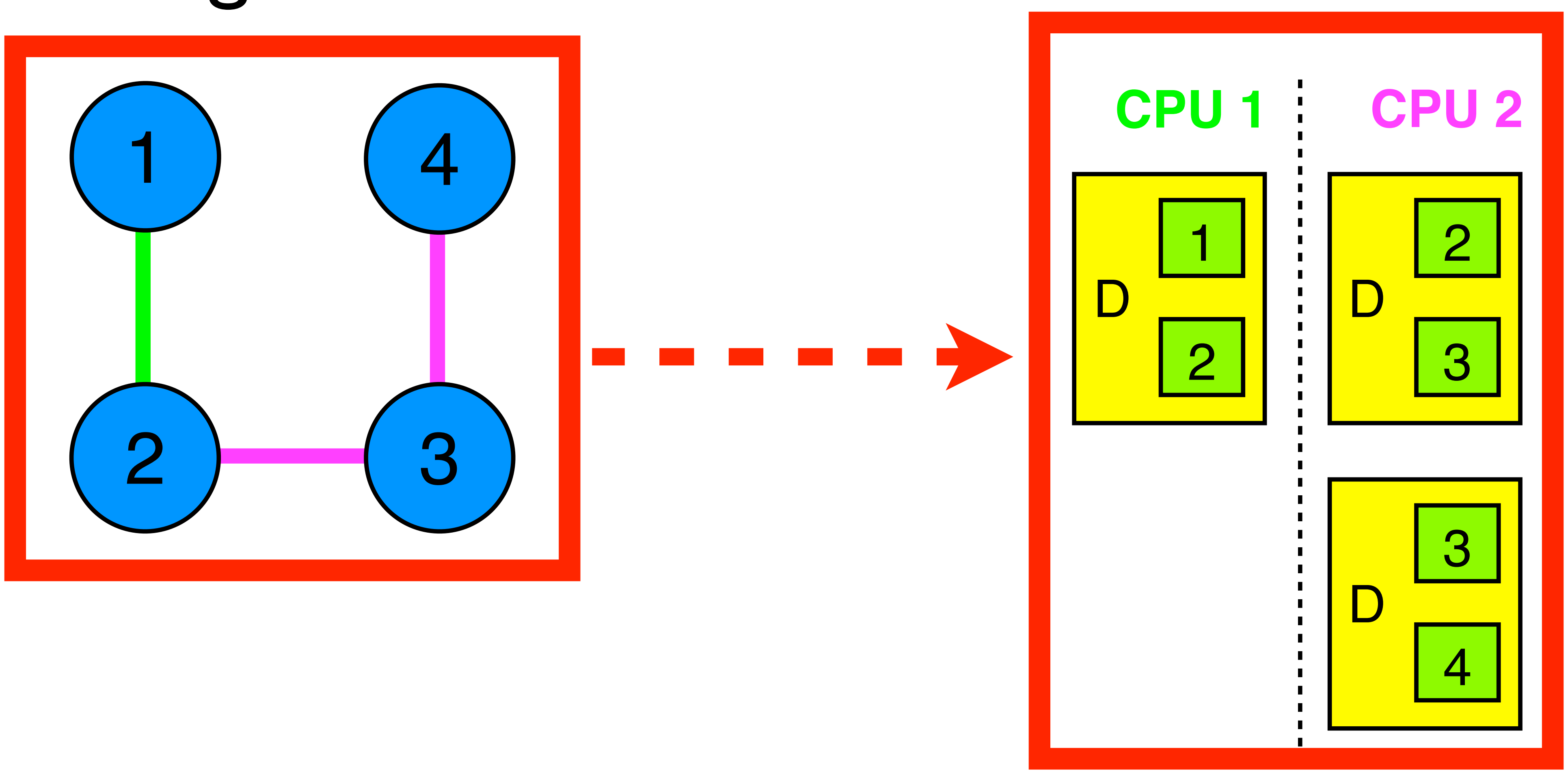


edges → outer critical sections

Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

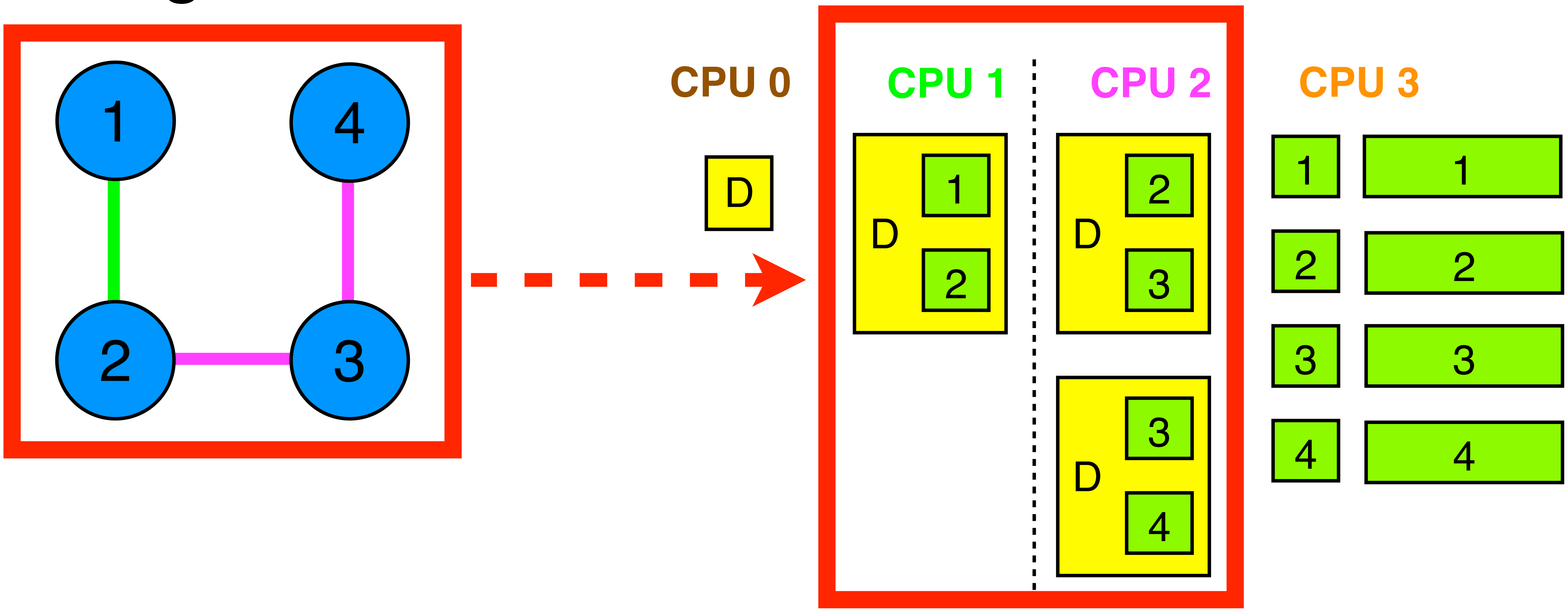
Blocking Analysis Problem



Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

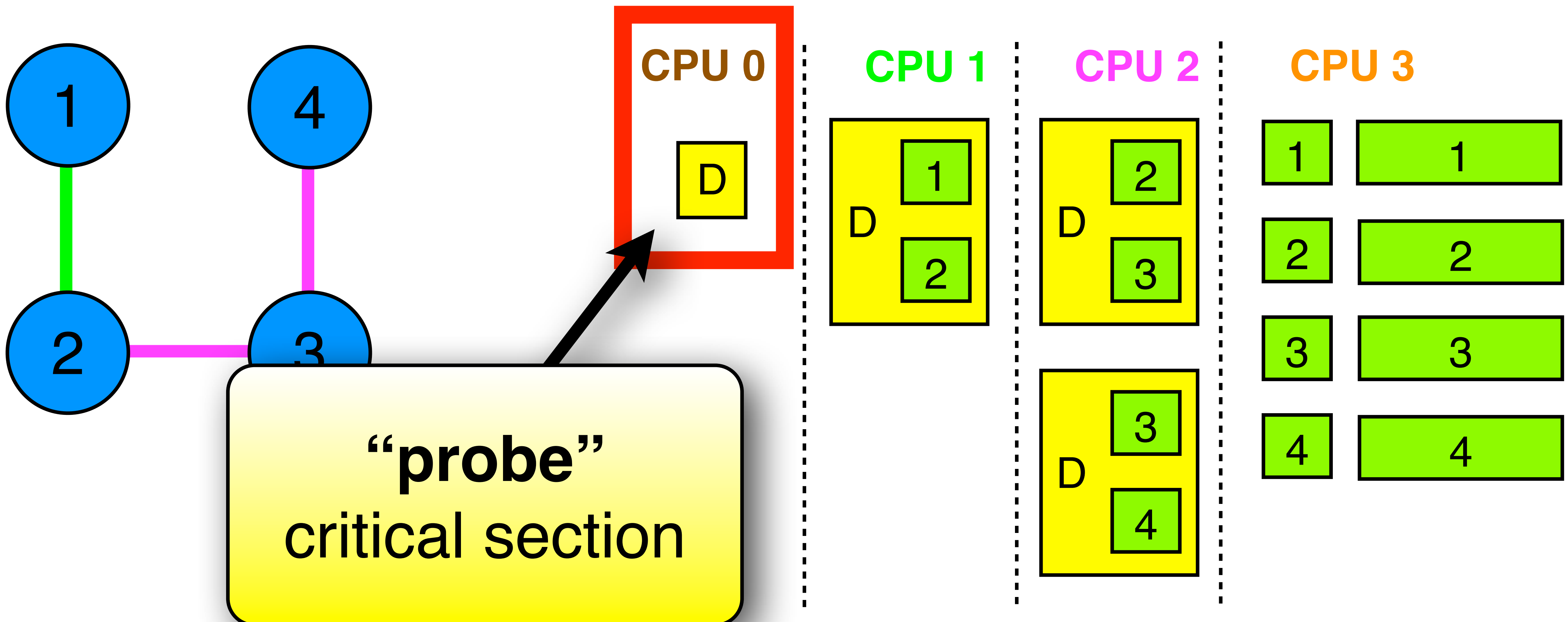
Blocking Analysis Problem



Encoding MCM as a Blocking Analysis

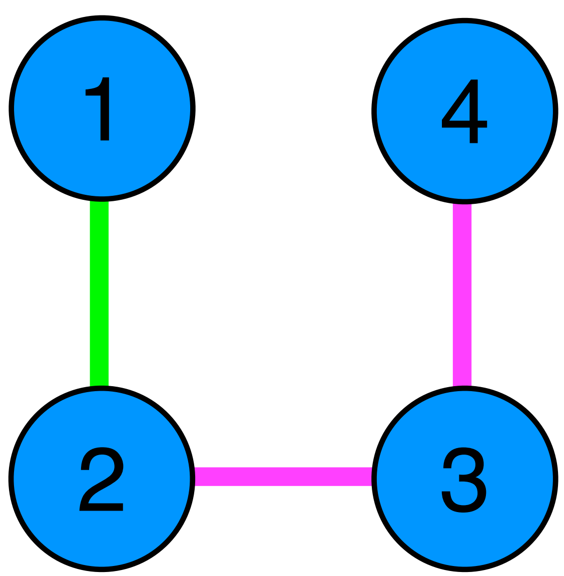
Multiple-Choice Matching Problem

Blocking Analysis Problem

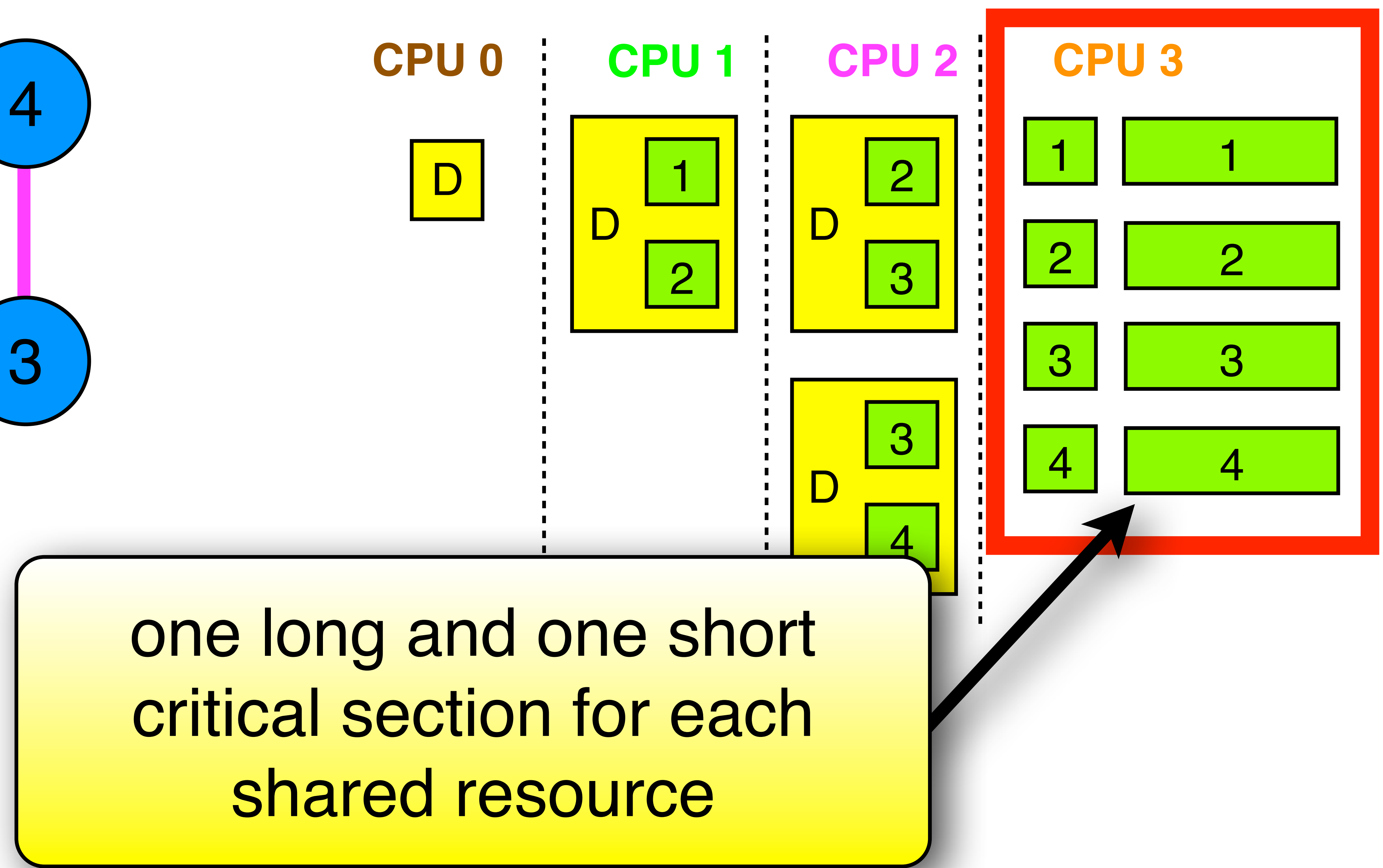


Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

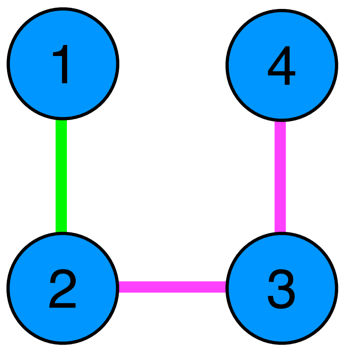


Blocking Analysis Problem

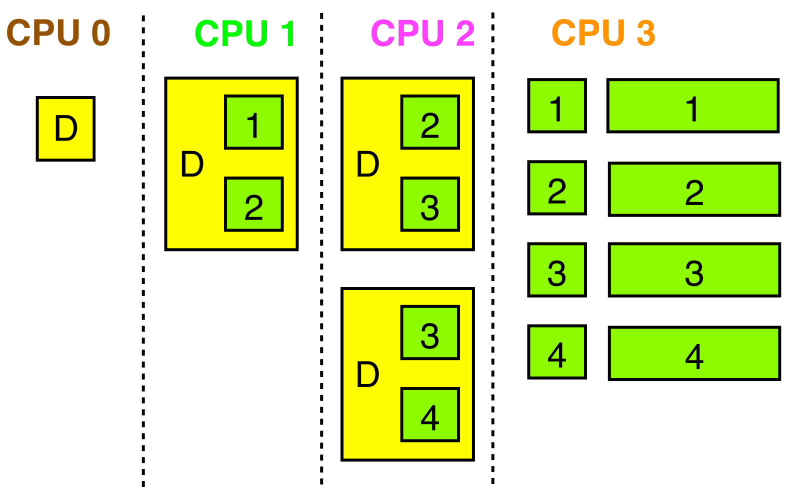


Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem

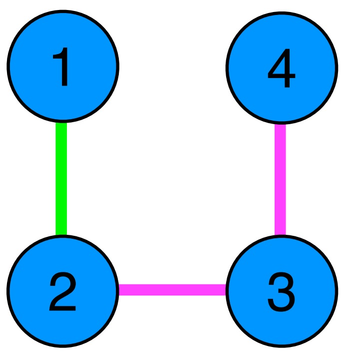


Blocking Analysis Problem



Encoding MCM as a Blocking Analysis

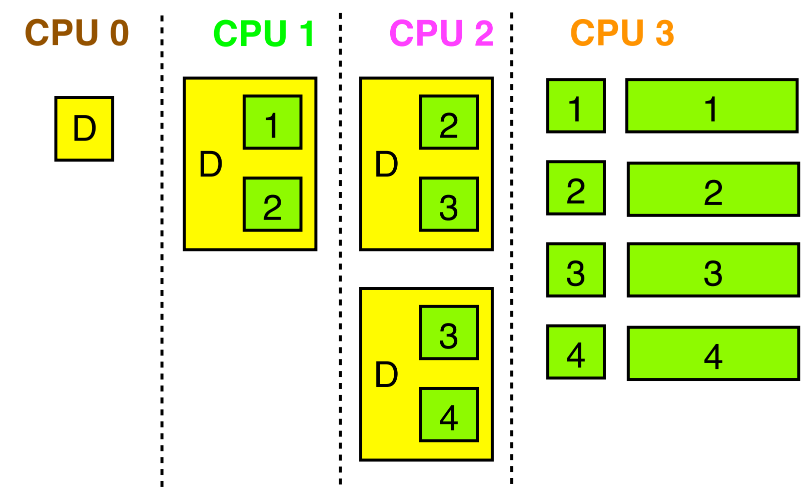
Multiple-Choice Matching Problem



Is there a set of edges **F** such that

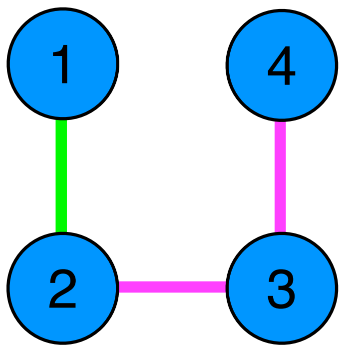
F contains exactly
one green and one purple edge

Blocking Analysis Problem



Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem



Is there a set of edges **F** such that

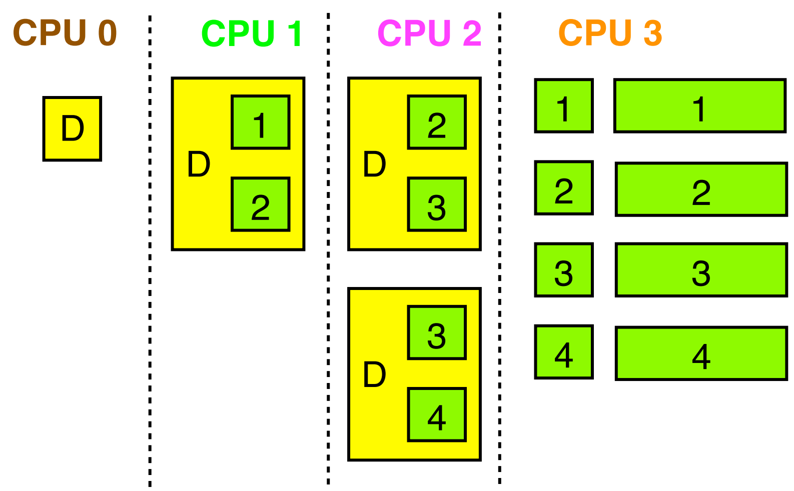
F contains exactly one green and one purple edge

and

all edges in **F** are pairwise disjoint

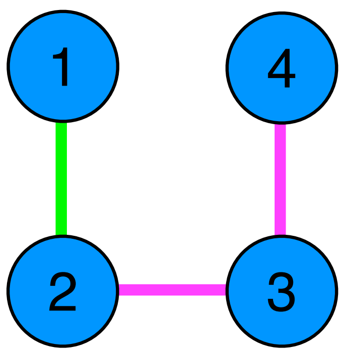
?

Blocking Analysis Problem



Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem



Is there a set of edges **F** such that

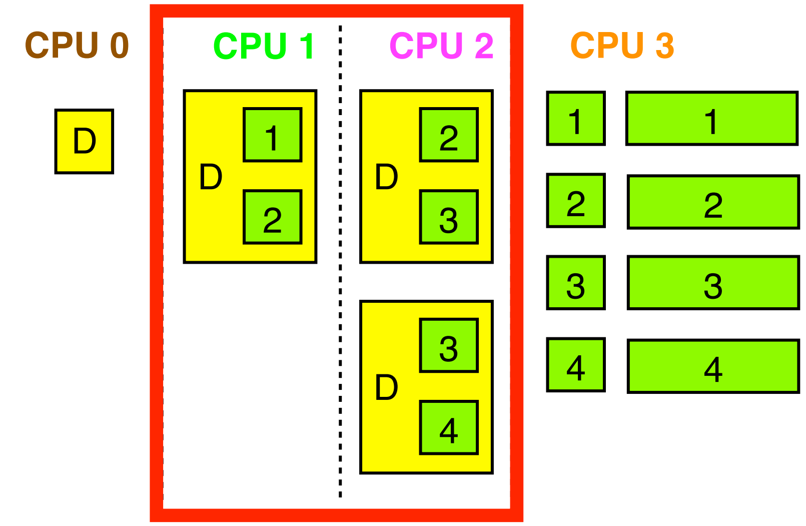
F contains exactly one green and one purple edge

and

all edges in **F** are pairwise disjoint

?

Blocking Analysis Problem

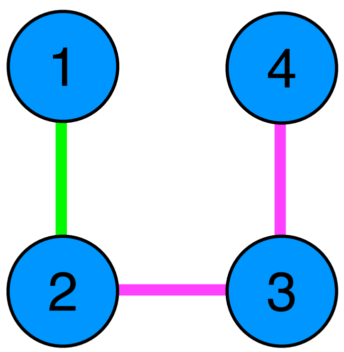


Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block

Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem



Is there a set of edges **F** such that

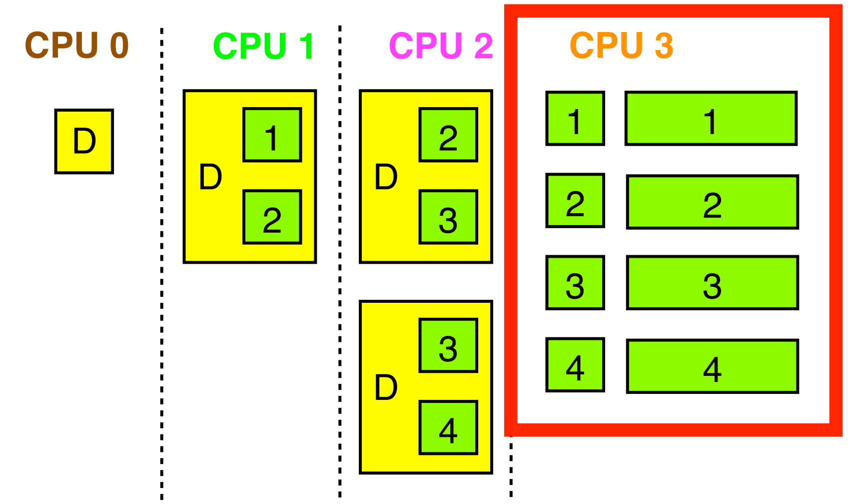
F contains exactly one green and one purple edge

and

all edges in **F** are pairwise disjoint

?

Blocking Analysis Problem



Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block

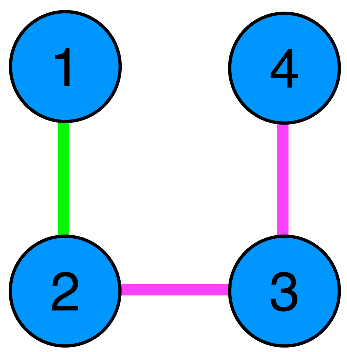
and

no short CS from CPU 3 transitively blocks

?

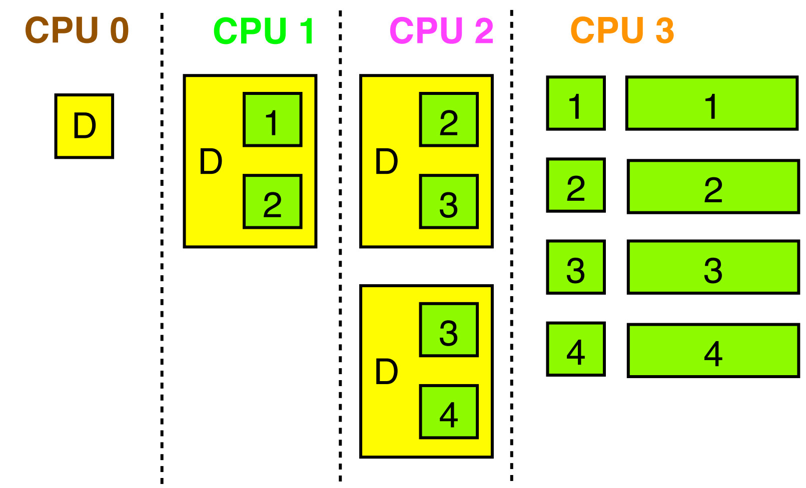
Encoding MCM as a Blocking Analysis

Multiple-Choice Matching Problem



Is there a set of edges **F** such that

Blocking Analysis Problem



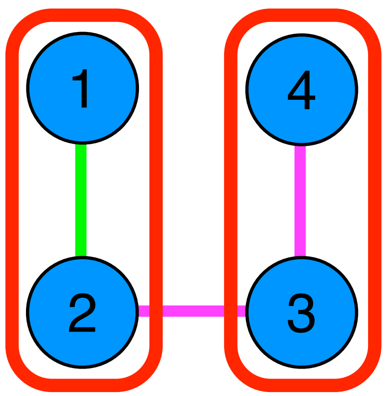
Is there a worst-case schedule such that

F contains exactly one green and one purple edge ⟺ exactly one CS for **D** from CPU 1 and CPU 2 block

all edges in **F** are pairwise disjoint ⟺ no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does** Exist

Multiple-Choice Matching Problem



Is there a set of edges **F** such that

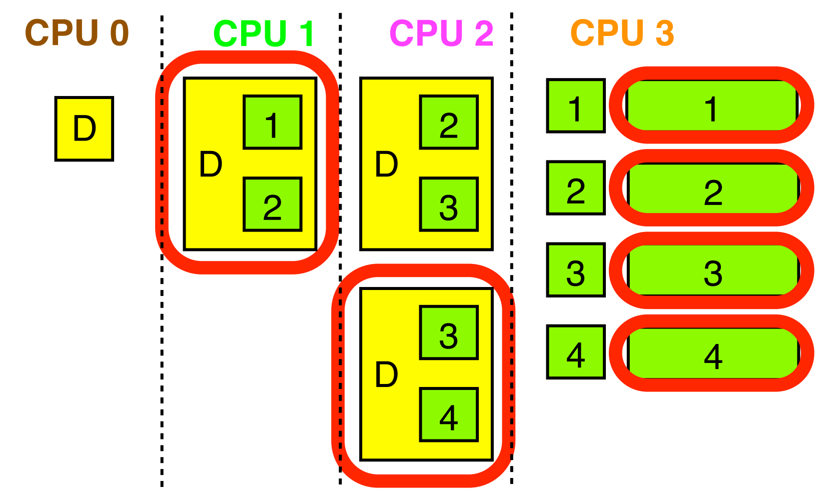
F contains exactly one green and one purple edge

and

all edges in **F** are pairwise disjoint

?

Blocking Analysis Problem



Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block

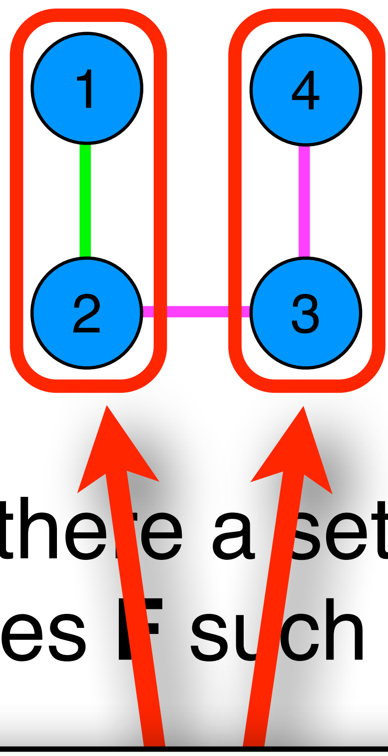
and

no short CS from CPU 3 transitively blocks

?

Example: MCM Solution **Does** Exist

Multiple-Choice Matching Problem

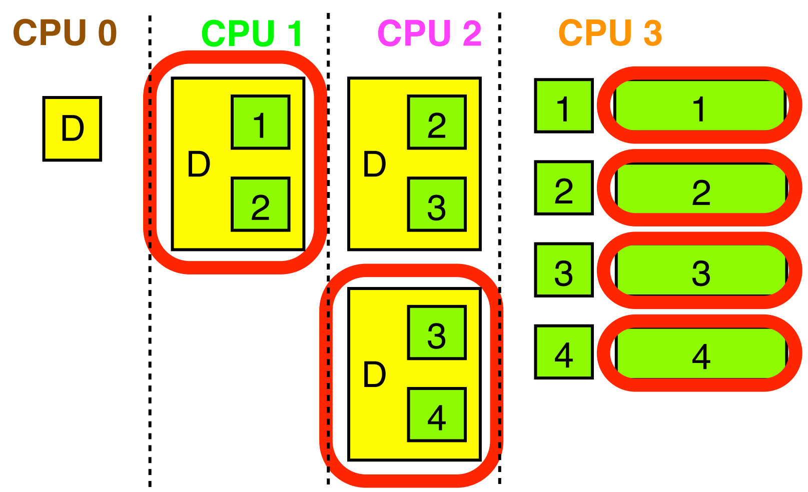


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint

Blocking Analysis Problem



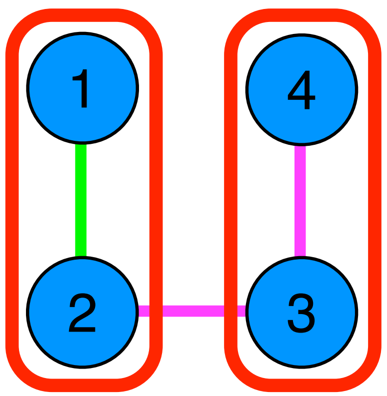
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does** Exist

Multiple-Choice Matching Problem

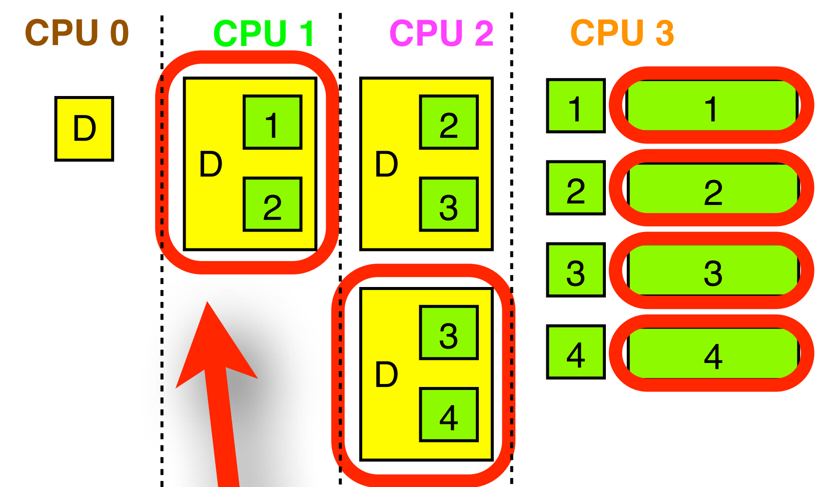


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint

Blocking Analysis Problem



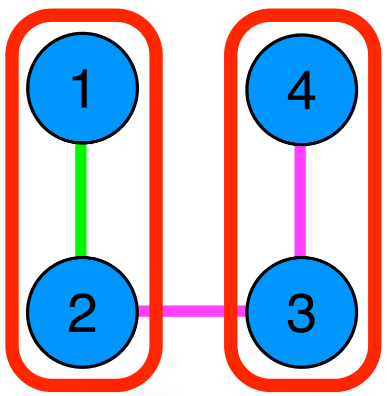
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block ✓

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does** Exist

Multiple-Choice Matching Problem

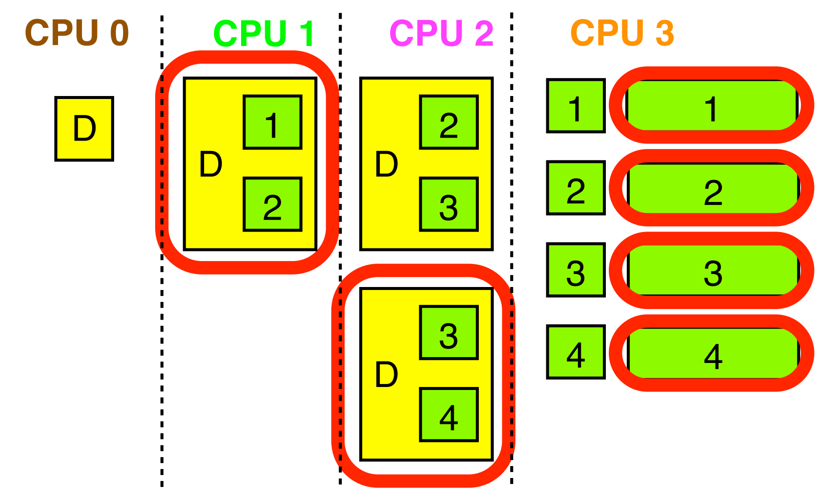


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint ✓

Blocking Analysis Problem



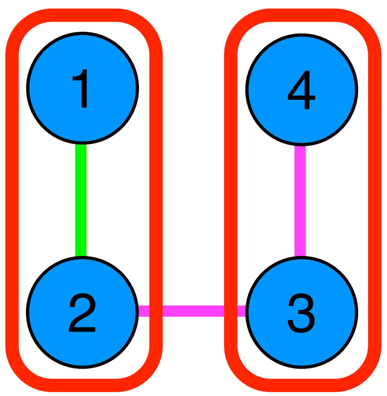
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block ✓

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does** Exist

Multiple-Choice Matching Problem

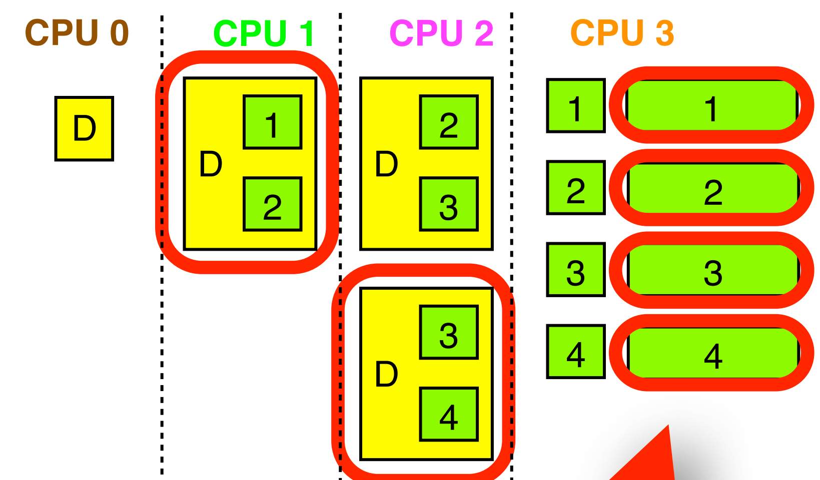


Is there a set of edges **F** such that

F contains exactly one green and one purple edge

all edges in **F** are pairwise disjoint

Blocking Analysis Problem



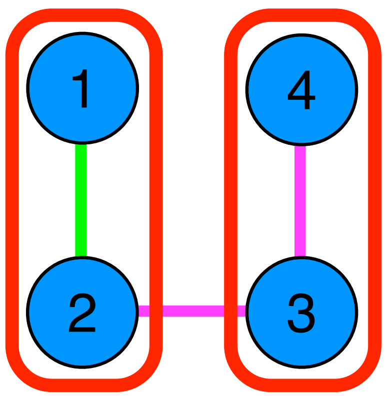
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does** Exist

Multiple-Choice Matching Problem

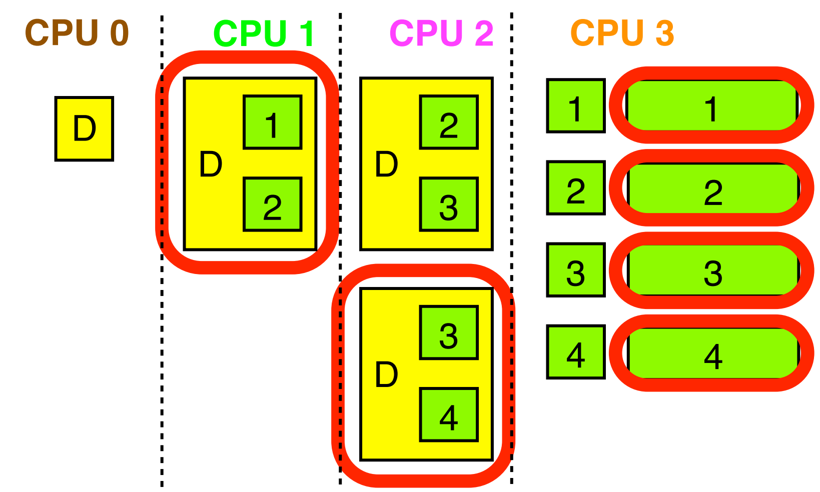


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint ✓

Blocking Analysis Problem



Is there a worst-case schedule such that

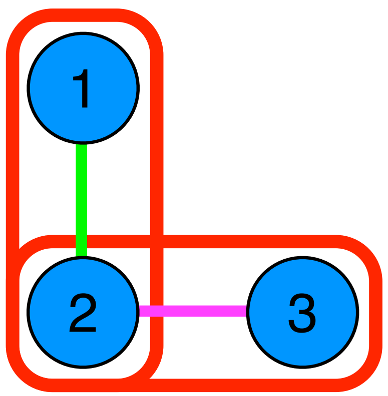
exactly one CS for **D** from CPU 1 and CPU 2 block ✓

no short CS from CPU 3 transitively blocks ✓

Example: MCM Solution **Does Not** Exist

Example: MCM Solution **Does Not** Exist

Multiple-Choice Matching Problem

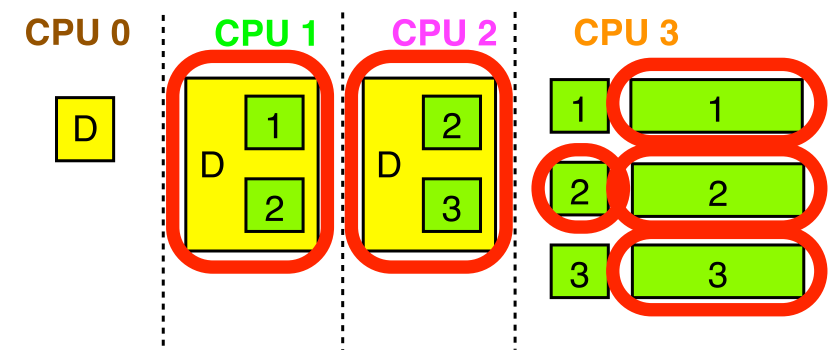


Is there a set of edges **F** such that

F contains exactly one green and one purple edge

all edges in **F** are pairwise disjoint

Blocking Analysis Problem



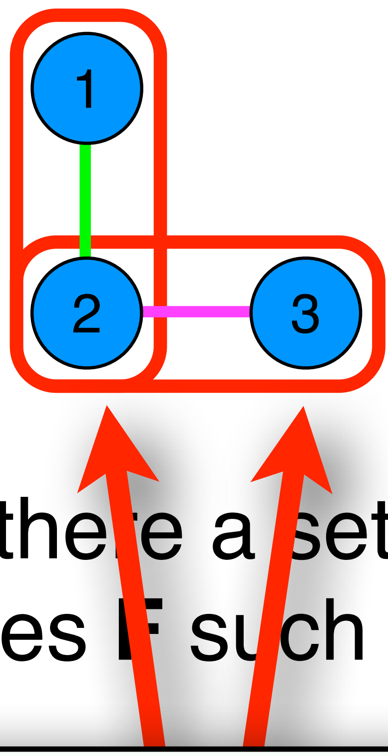
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does Not** Exist

Multiple-Choice Matching Problem

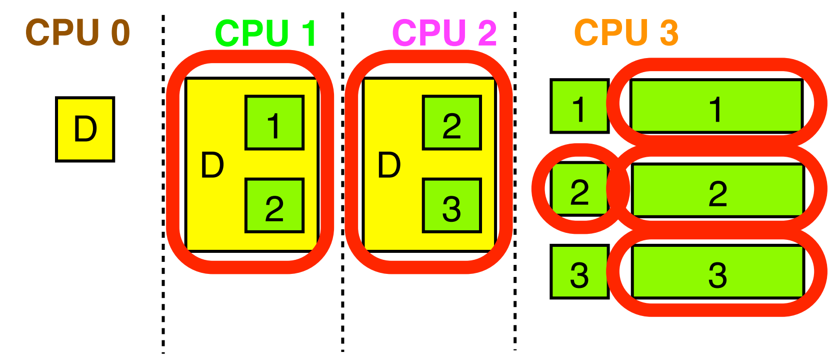


Is there a set of edges **F** such that

F contains exactly one green and one purple edge

all edges in **F** are pairwise disjoint

Blocking Analysis Problem



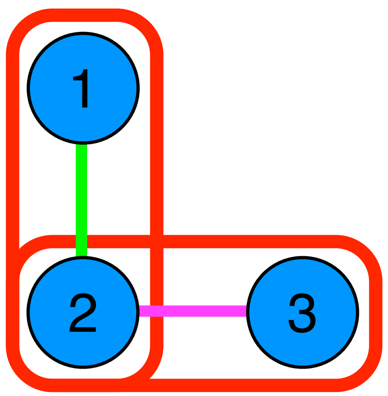
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does Not** Exist

Multiple-Choice Matching Problem

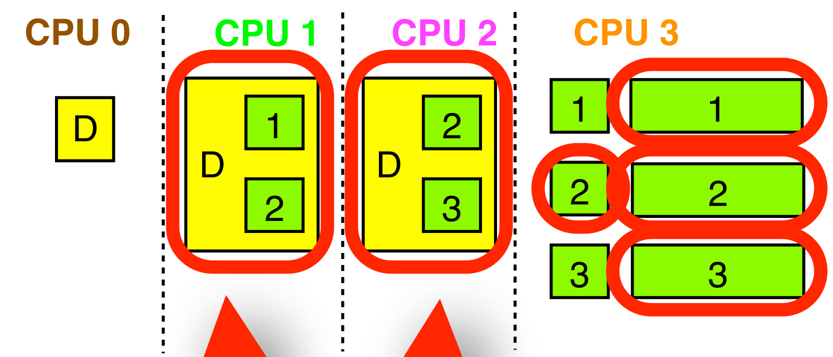


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint

Blocking Analysis Problem



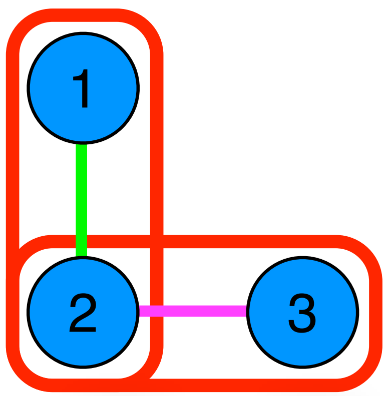
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block ✓

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does Not** Exist

Multiple-Choice Matching Problem

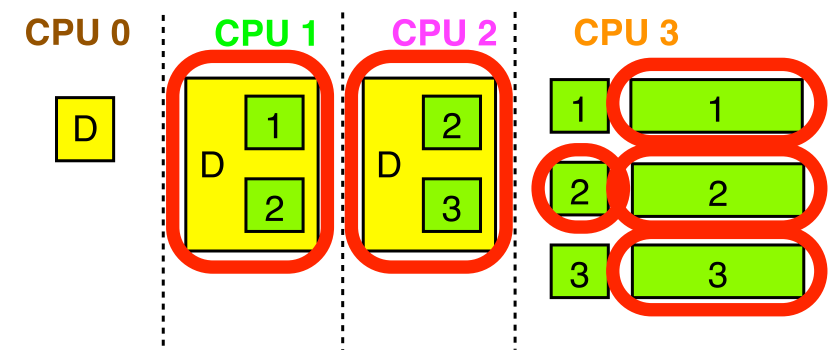


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint ✗

Blocking Analysis Problem



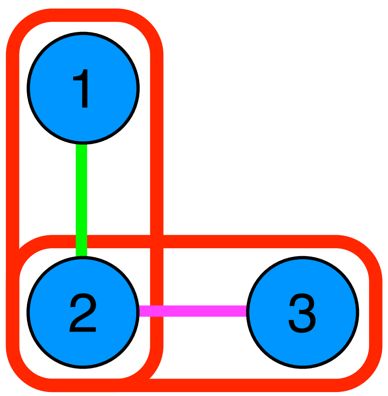
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block ✓

no short CS from CPU 3 transitively blocks

Example: MCM Solution **Does Not** Exist

Multiple-Choice Matching Problem

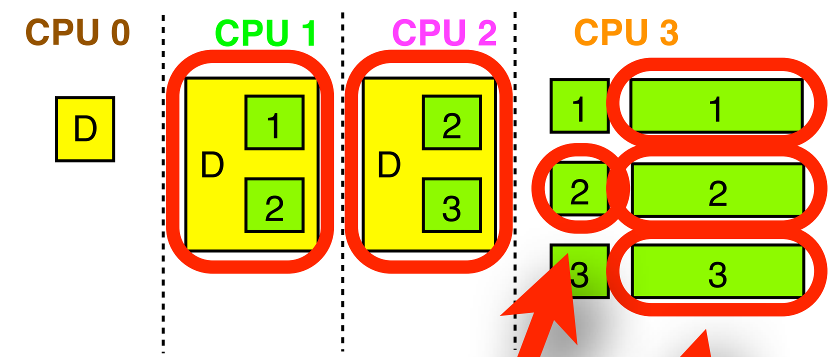


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint ✗

Blocking Analysis Problem



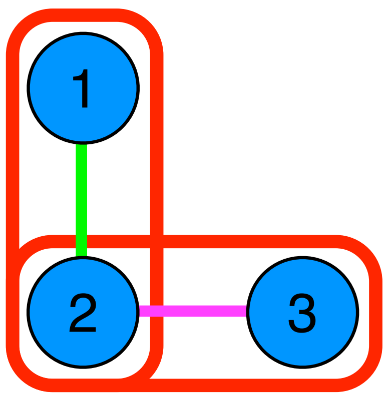
Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block ✓

no short CS from CPU 3 transitively blocks ✗

Example: MCM Solution **Does Not** Exist

Multiple-Choice Matching Problem

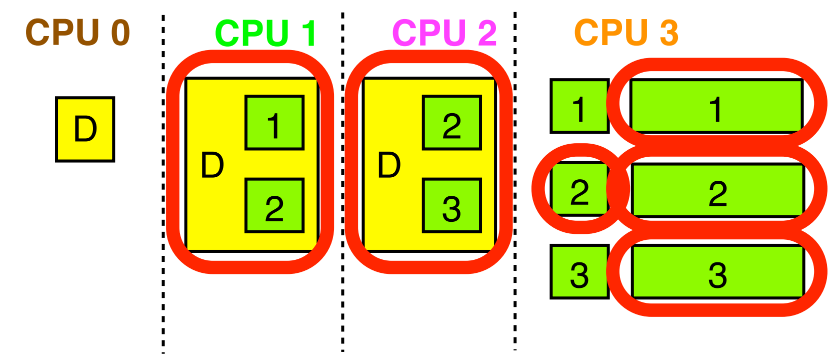


Is there a set of edges **F** such that

F contains exactly one green and one purple edge ✓

all edges in **F** are pairwise disjoint ✗

Blocking Analysis Problem



Is there a worst-case schedule such that

exactly one CS for **D** from CPU 1 and CPU 2 block ✓

no short CS from CPU 3 transitively blocks ✗

Encoding MCM as a Blocking Analysis

Result of blocking analysis is
bound on blocking duration!

Encoding MCM as a Blocking Analysis

Result of blocking analysis is
bound on blocking duration!

Is there a worst-case
schedule such that

exactly one CS for **D** from
CPU 1 and CPU 2 block

and

no short CS from CPU 3
transitively blocks

?

Encoding MCM as a Blocking Analysis

Result of blocking analysis is
bound on blocking duration!

Is there a worst-case schedule such that

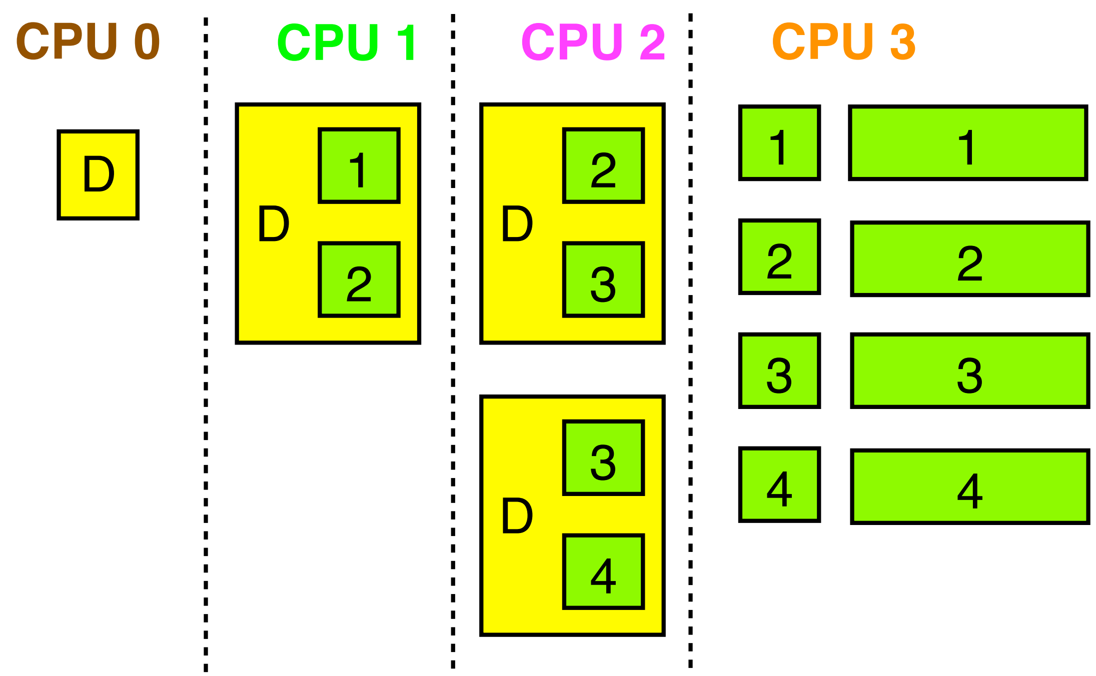
exactly one CS for **D** from CPU 1 and CPU 2 block

no short CS from CPU 3 transitively blocks

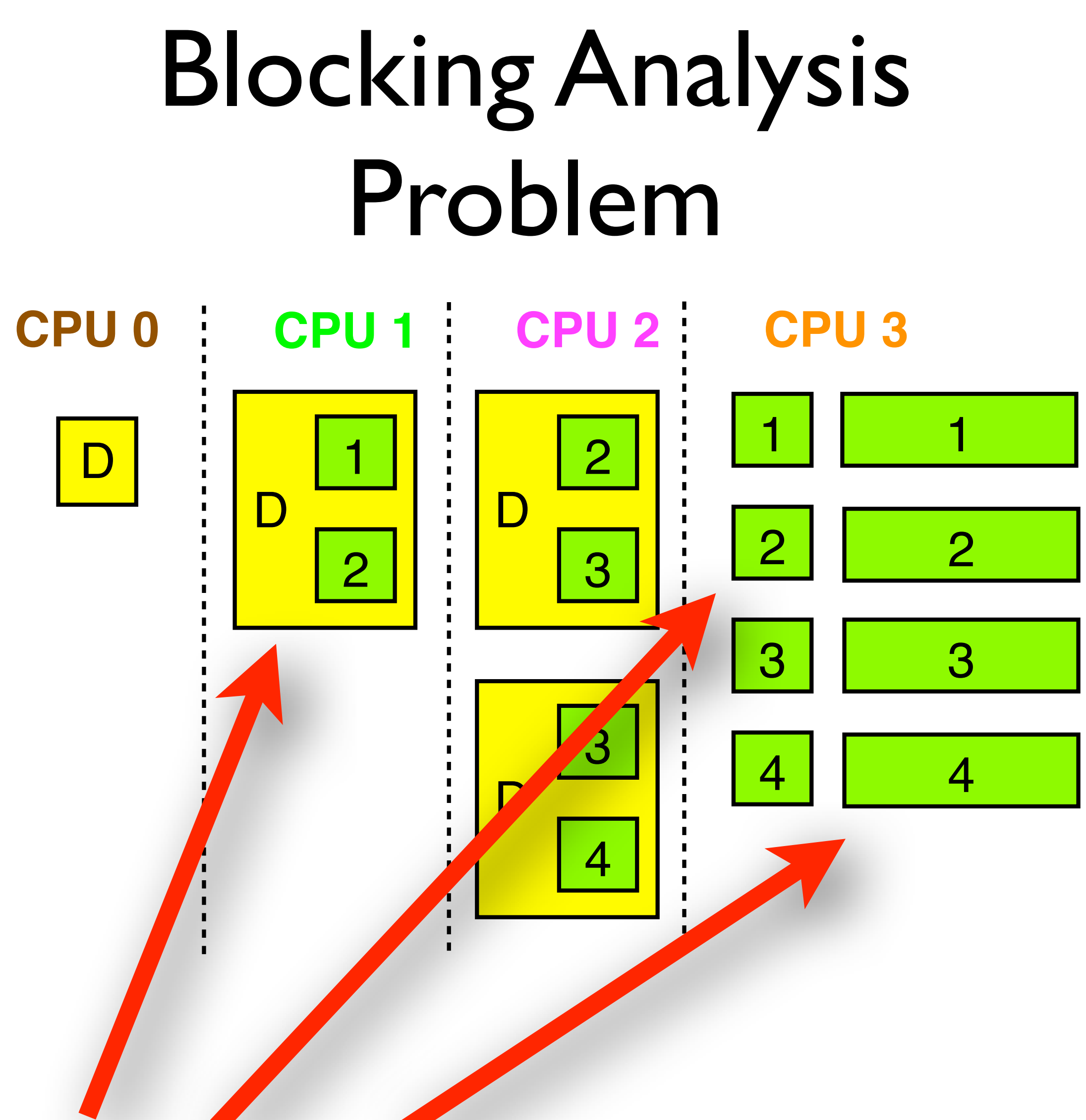
How to derive from blocking bound whether conditions hold?

Encoding MCM as a Blocking Analysis

Blocking Analysis Problem



Encoding MCM as a Blocking Analysis

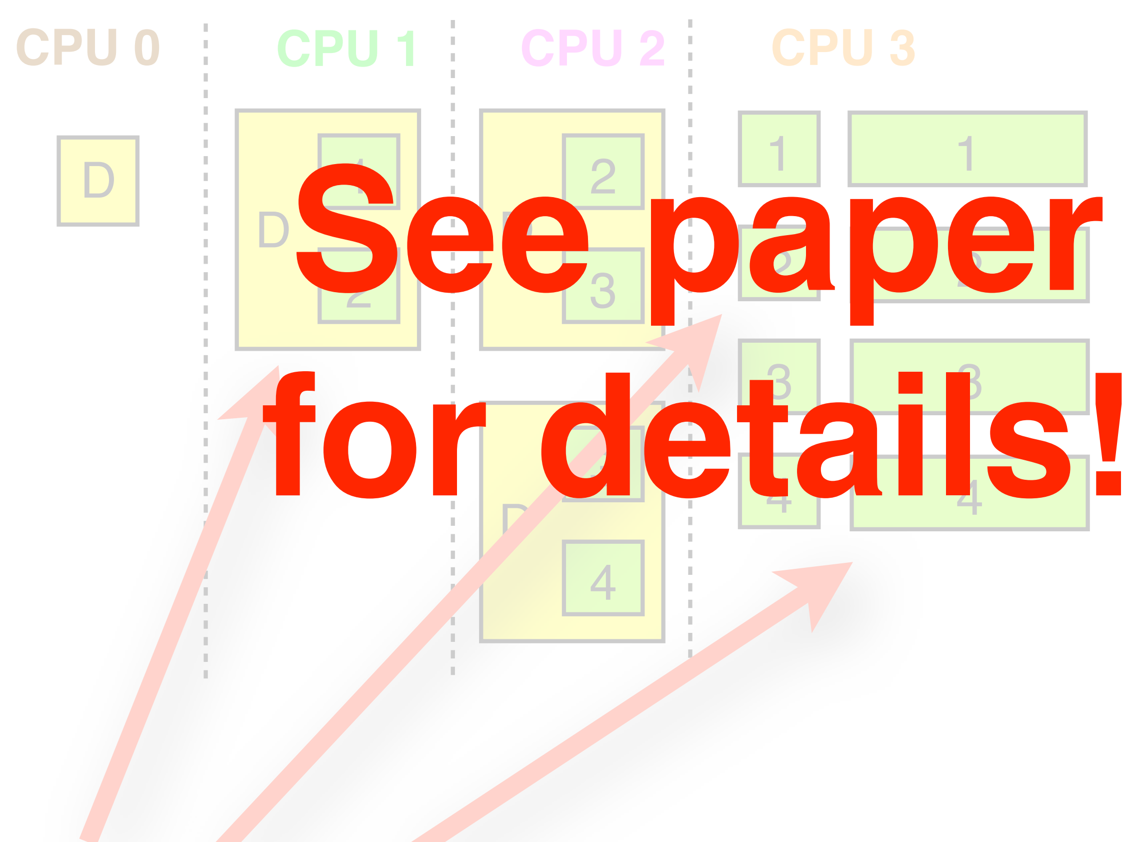


Choose critical section lengths such that it can be inferred from the blocking bound:

- how many requests from CPU 1 and CPU 2 block, and
- how many short requests from CPU 3 transitively block.

Encoding MCM as a Blocking Analysis

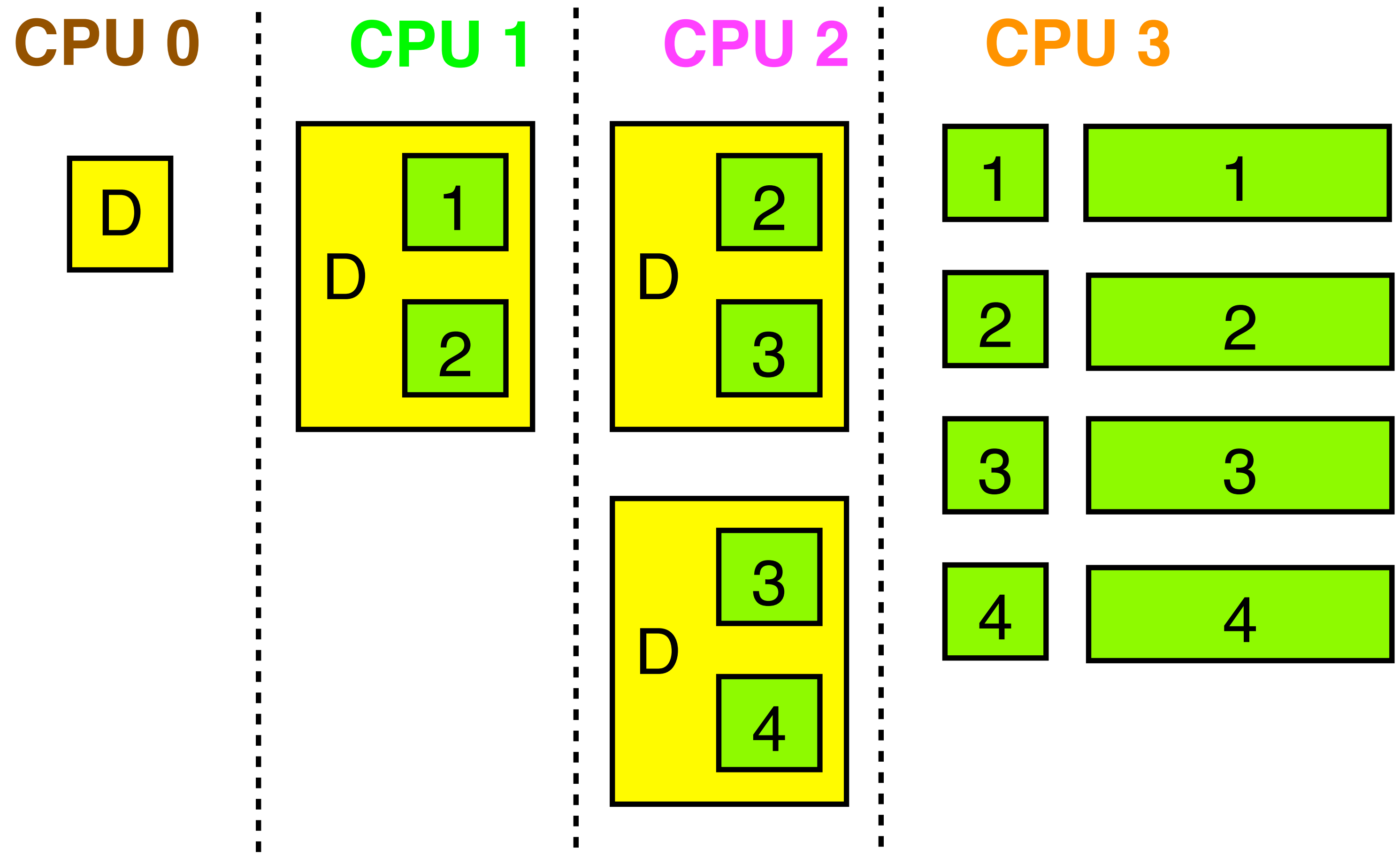
Blocking Analysis
Problem



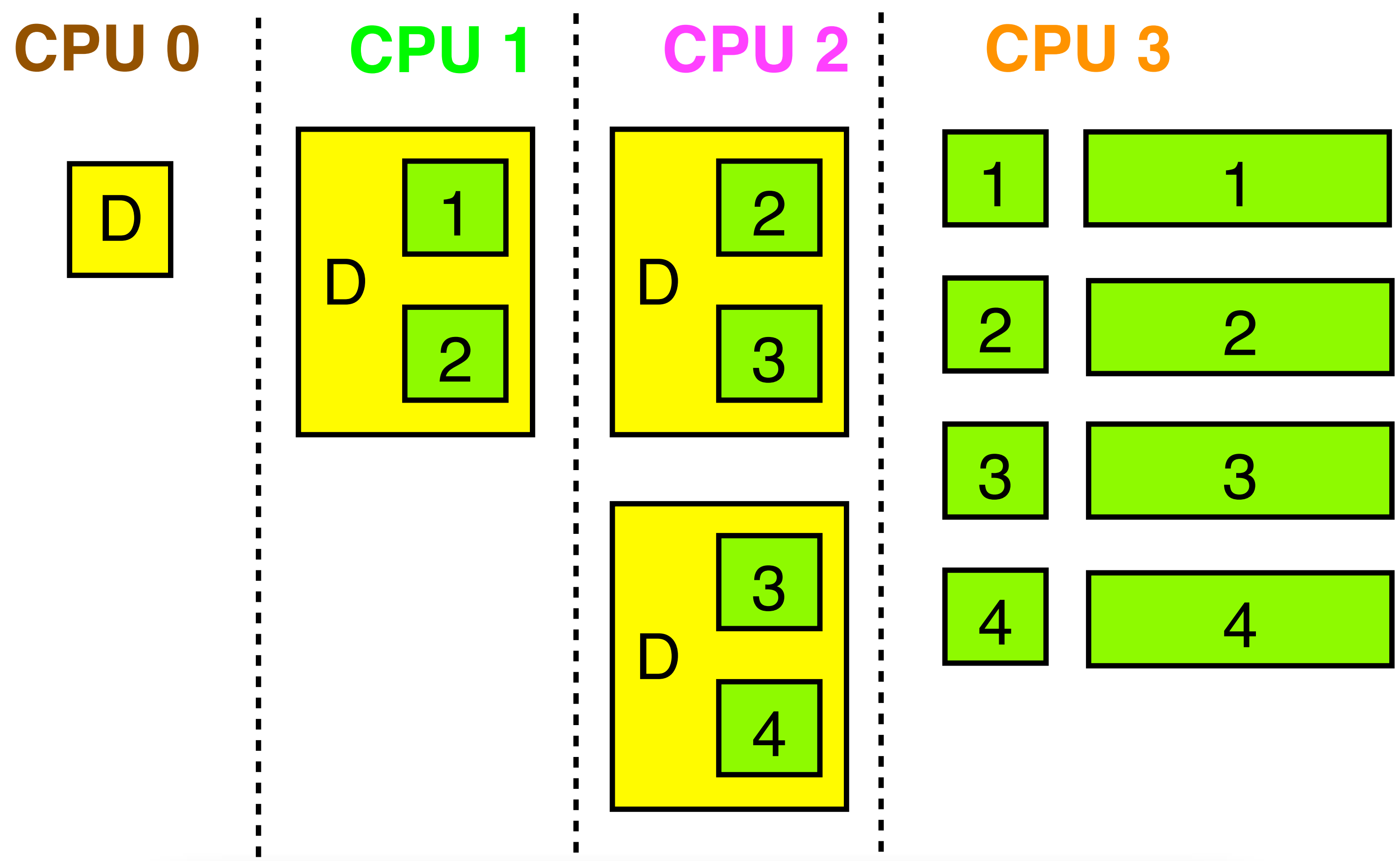
Choose critical section lengths such that it can be inferred from the blocking bound:

- how many requests from CPU 1 and CPU 2 block, and
- how many short requests from CPU 3 transitively block.

Generality of Reduction



Generality of Reduction



Reduction uses a **single job per processor**.

Generality of Reduction

Reduction uses a **single job per processor**.

Reduction oblivious to

spin based vs. suspension based locks

preemptable vs. non-preemptable spinning

any work conserving scheduler

Outline

- Introduction ✓
- Intuition: Why does nesting make the analysis difficult? ✓
- Reduction: From Multiple Choice Matching to Blocking Analysis ✓
- Summary and Conclusion

Summary

multiprocessor system

locks with **strong ordering guarantees (FIFO/Priority)**

nested critical sections

Blocking analysis
NP-hard!

Summary

multiprocessor system

locks with **strong ordering guarantees (FIFO/Priority)**

nested critical sections

Blocking analysis
NP-hard!

Reduction from
Multiple Choice Matching
Problem

Future Work

FIFO/priority-ordered locks:
Approximation hardness? PTAS?

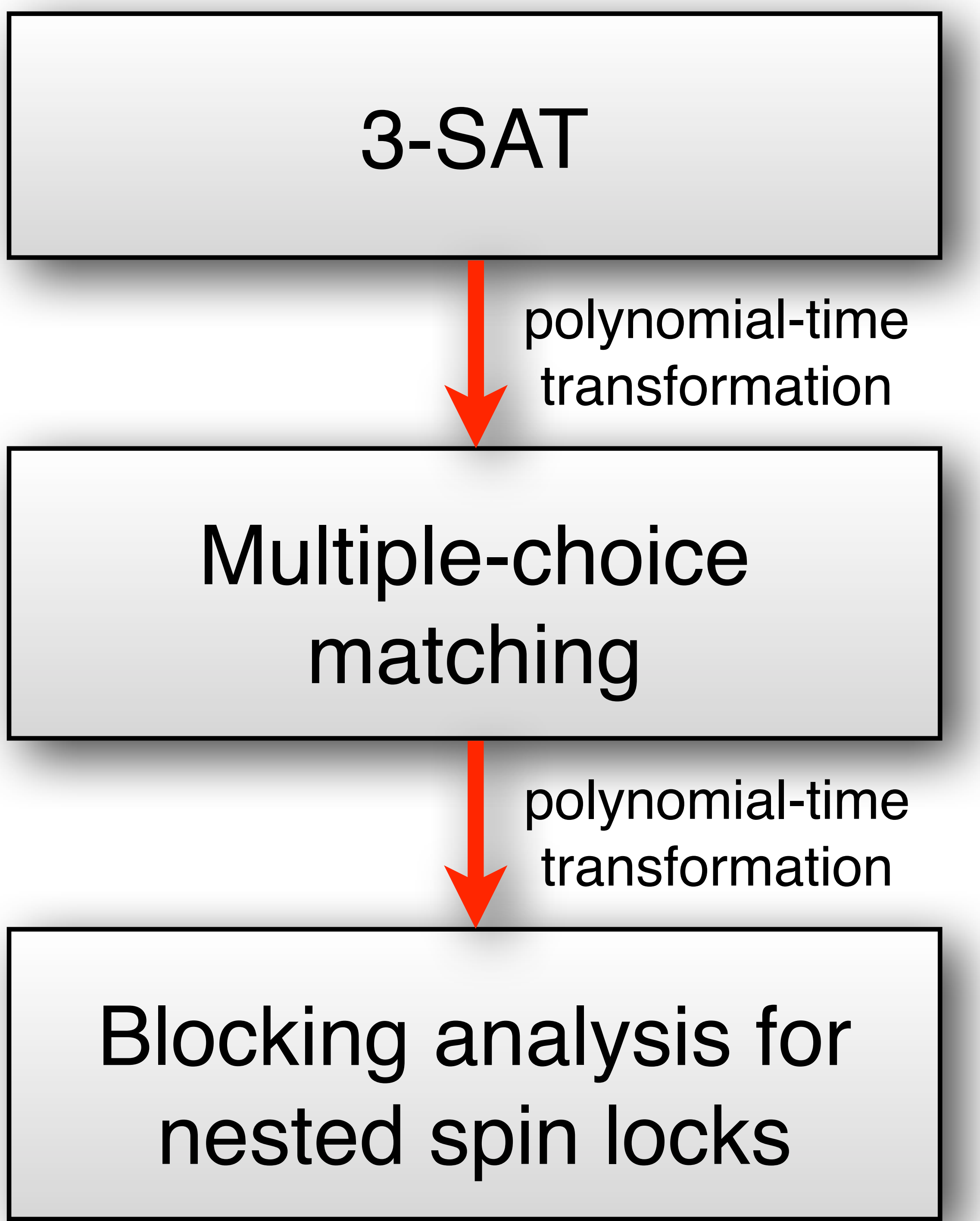
Future Work

FIFO/priority-ordered locks:
Approximation hardness? PTAS?

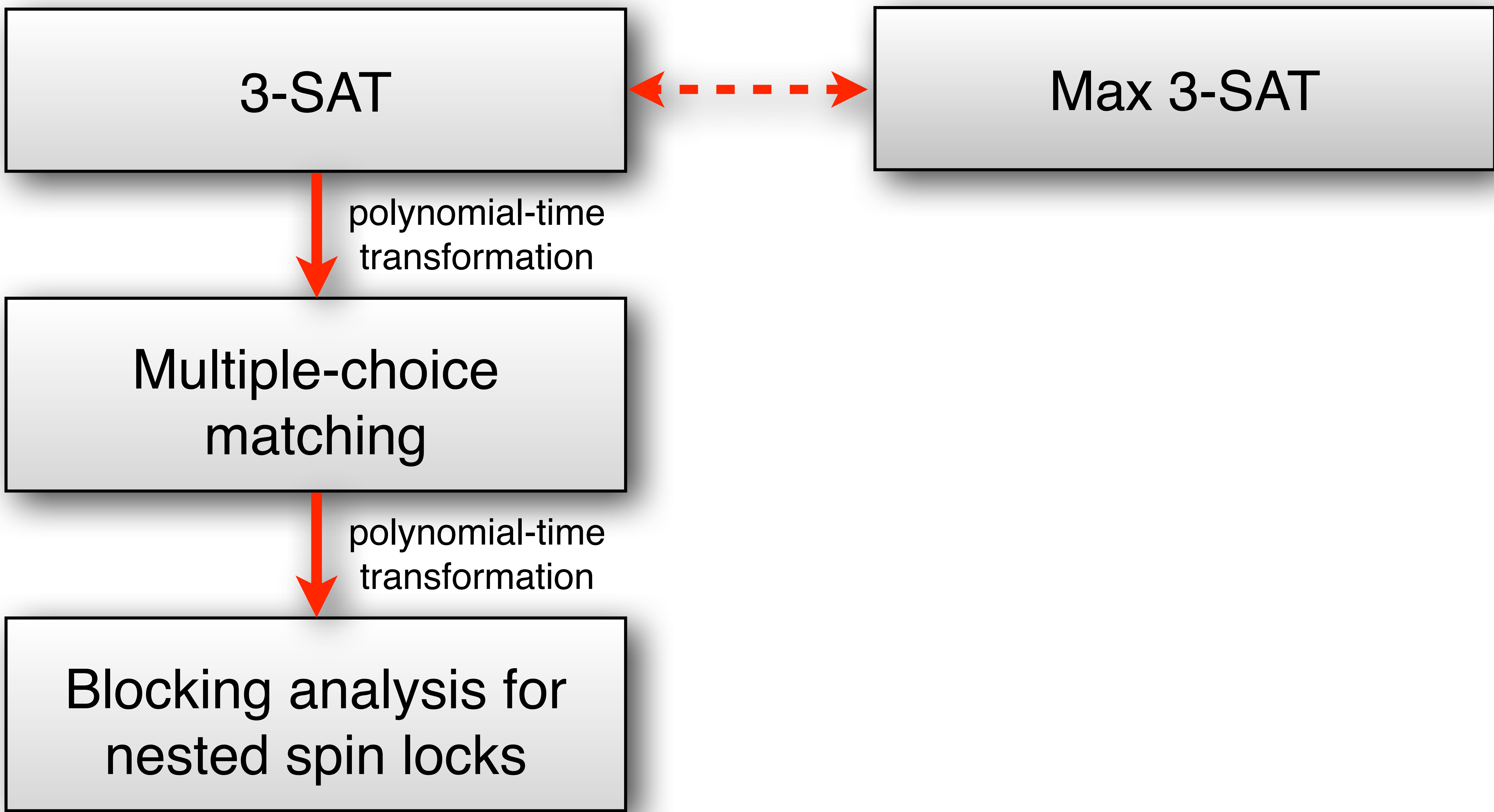
Nesting and efficient analysis possible?

This slide is intentionally left blank.

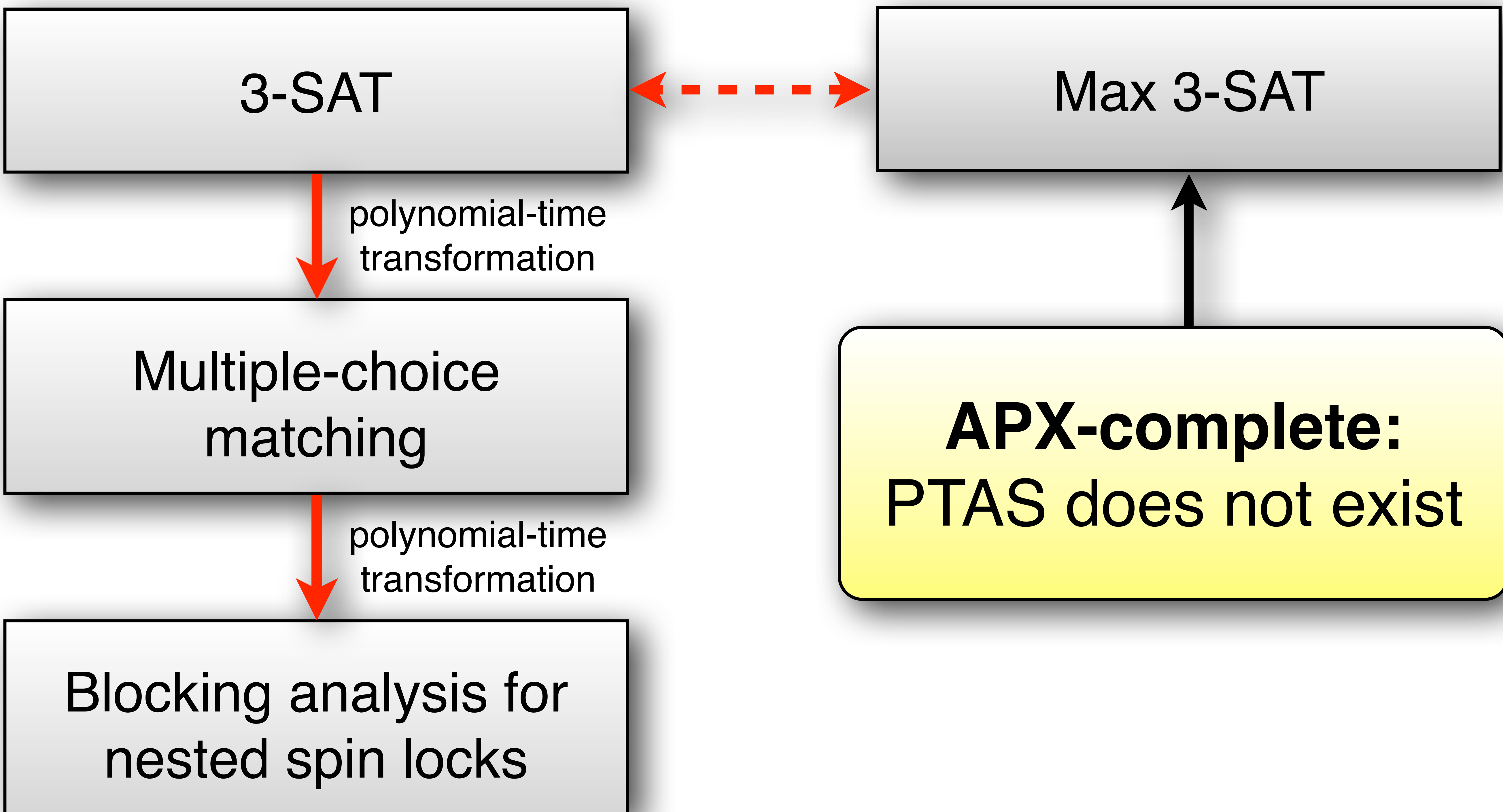
PTAS for Blocking Analysis of Nested Locks



PTAS for Blocking Analysis of Nested Locks



PTAS for Blocking Analysis of Nested Locks



PTAS for Blocking Analysis of Nested Locks