Cut-off theorems for deadlocks and serializability

Lisbeth Fajstrup

Department of Mathematics
Aalborg University

ACAT in Bremen, 2013

Cut-Off Theorems

General

Property $\mathcal{P}$ holds for all $n \in \mathbb{N}$ if and only $\mathcal{P}$ holds for $n \leq M$

Think of n as a dimension.
Here: Given thread T, T^n means n copies of T run in parallel.
The property $\mathcal{P}$ is *deadlock free* in one theorem and *serializable* in the other.

PV-programs - controlling concurrency through locks

- A set of shared resources $\mathcal{R}$ - memory, printers,...
- A capacity function $\kappa : \mathcal{R} \to \mathbb{N}$.

PV programs p are defined by the grammar

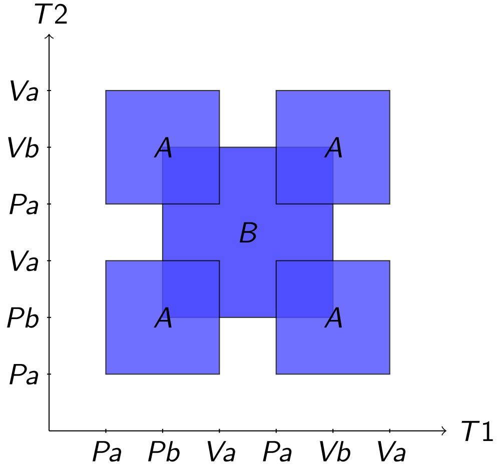
$$p ::= P_a \mid V_a \mid p.p \mid p|p \mid p+p \mid p^*$$

- P_a is a request to access the resource a, if granted acces, lock it. V_a releases the resource.
- A program without the parallel operator is a *thread*.
- Resource r may be accessed by at most $\kappa(r)$ threads at a time.
- At a final point, the program has released all resources.
- At an initial point, no resources are locked

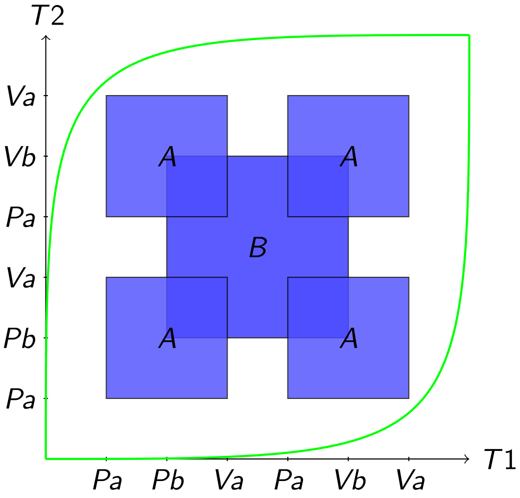
Geometrically

One thread is represented by a graph. n threads in parallel are represented by a product of n graphs with "holes" where a resource is locked above its capacity.

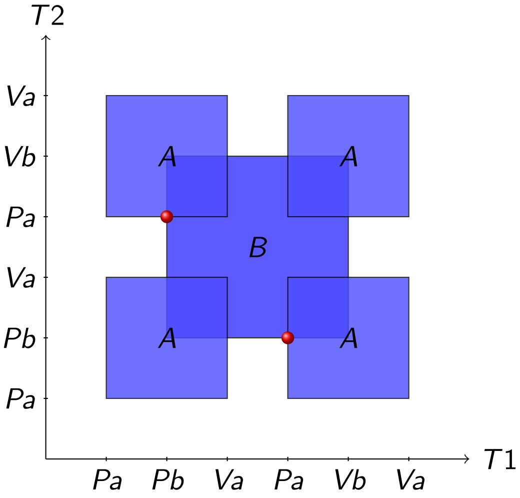
$$T1 = T2 = Pa.Pb.Va.Pa.Vb.Va \qquad \kappa \equiv 1$$



An execution is a directed path



Deadlock - no directed paths leave the point



Definition

A state $\mathbf{x} = (x_1, \ldots, x_n) \in T1 \times T2 \times \cdots \times Tn$ is a deadlock if

- The only dipaths starting in $\mathbf{x}$ are constant.
- $\mathbf{x}$ is reachable: There is a dipath from $\mathbf{0}$ to $\mathbf{x}$
- $\mathbf{x} \neq \top$ - not all Ti have finished

If $\mathbf{x}$ is a deadlock, then $x_i = \top$ or $x_i = Pr(i)$ and $\kappa(r(i))$ other threads hold a lock on $r(i)$ at $\mathbf{x}$

A cut-off theorem for deadlocks

Theorem 1

Let T be a PV thread accessing resources $\mathcal{R}$ with capacity $\kappa : \mathcal{R} \to \mathbb{N}$.
Let T^n be n copies of T run in parallel.
T^n is deadlock free for all n if and only if T^M is deadlock free, where $M = \Sigma_{r\in\mathcal{R}}\kappa(r)$.

The bound M is sharp.

Theorem 2

Given $\mathcal{R} = \{r_1, \ldots, r_k\}$ with capacity $\kappa : \mathcal{R} \to \mathbb{N}$, the thread $T = Pr_1 Pr_2 Vr_1 Pr_3 Vr_2 \ldots Pr_k Vr_{k-1} Pr_1 Vr_k Vr_1$ satisfies:

- There is a deadlock in T^M (and hence for all $n \geq M$) where $M = \Sigma_{r \in \mathcal{R}} \kappa(r)$.
- There are no deadlocks in T^n for $n \leq M$

The bound M is sharp.

Theorem 2

Given $\mathcal{R} = \{r_1, \ldots, r_k\}$ with capacity $\kappa : \mathcal{R} \to \mathbb{N}$, the thread $T = Pr_1 Pr_2 Vr_1 Pr_3 Vr_2 \ldots Pr_k Vr_{k-1} Pr_1 Vr_k Vr_1$ satisfies:

- There is a deadlock in T^M (and hence for all $n \geq M$) where $M = \Sigma_{r \in \mathcal{R}} \kappa(r)$.
- There are no deadlocks in T^n for $n \leq M$

Proof: The deadlock is $\mathbf{x} = (x_1, \ldots, x_1, x_2, \ldots, x_2, \ldots, x_k \ldots x_k)$ where for $i \neq 1$, $x_i = Pr_i$ and x_i is repeated $\kappa(r_{i-1})$ times. x_1 is the last Pr_1 and is repeated $\kappa(r_k)$ times. All resources r are locked $\kappa(r)$ times.
Hence $\mathbf{x}$ is a deadlock and $\neq \top$. (Need to check reachability and no deadlocks in lower dimensions)

Example. When $\kappa \equiv 1$

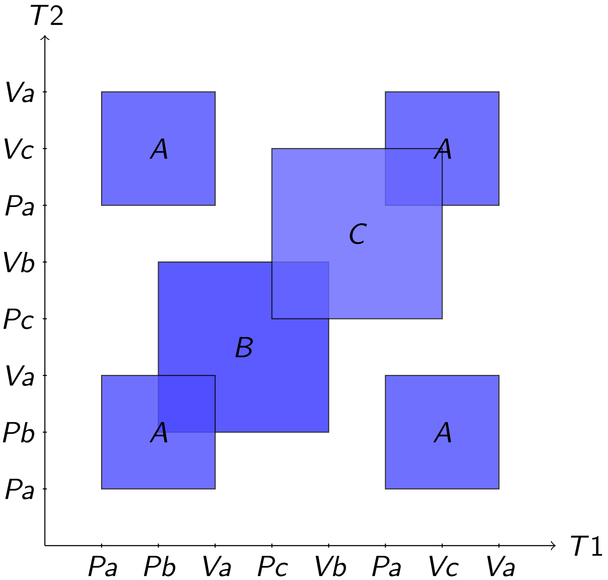
$\mathbf{x} = (x_1, x_2, \ldots, x_k)$ and $M = k$:

- $T1$ requests r_1 and has a lock on r_k
- Ti requests r_i and has a lock on r_{i+1}

There are no deadlocks in T^n for $n < M$: If $\mathbf{y}$ is a deadlock, then there is a $y_{i1} = Pr(i1)$ and another y_{i2} with a lock on $r(i1)$, so $y_{i2} = Pr(i1+1)$ (or the last $P(r1)$ if $r(i1) = r_k$.
Hence, $\mathbf{y}$ is a permutation of $(\mathbf{x}, \top, \top, \ldots, \top)$.
Similarly for general κ.
Still need to see that $\mathbf{x}$ is reachable.

Example

$T = PaPbVaPcVbPaVcVa$, $\kappa \equiv 1$. T^3 has a deadlock at $(6,2,4)$



$\mathbf{x} = (x_1, \ldots, x_1, x_2, \ldots, x_2, \ldots, x_k \ldots x_k)$ is reachable

$T = Pr_1 Pr_2 Vr_1 Pr_3 Vr_2 \ldots Pr_k Vr_{k-1} Pr_1 Vr_k Vr_1$
a dipath from $\mathbf{0}$ to $\mathbf{x}$ is composed of the pieces:

- $\gamma_0 : \mathbf{0} \to (\mathbf{x_1}, \mathbf{0})$ serially - one coordinate at a time. OBS, now hold $\kappa(r_k)$ locks on r_k

$\mathbf{x} = (x_1, \ldots, x_1, x_2, \ldots, x_2, \ldots, x_k \ldots x_k)$ is reachable

$T = Pr_1 Pr_2 Vr_1 Pr_3 Vr_2 \ldots Pr_k Vr_{k-1} Pr_1 Vr_k Vr_1$
a dipath from $\mathbf{0}$ to $\mathbf{x}$ is composed of the pieces:

- $\gamma_0 : \mathbf{0} \to (\mathbf{x_1}, \mathbf{0})$ serially - one coordinate at a time. OBS, now hold $\kappa(r_k)$ locks on r_k
- $\gamma_1 : (\mathbf{x_1}, \mathbf{0}) \to (\mathbf{x_1}, \mathbf{0}, \mathbf{x_k})$ one coordinate at a time. Now also $\kappa(r_{k-1})$ locks on r_{k-1}

$\mathbf{x} = (x_1, \ldots, x_1, x_2, \ldots, x_2, \ldots, x_k \ldots x_k)$ is reachable

$T = Pr_1 Pr_2 Vr_1 Pr_3 Vr_2 \ldots Pr_k Vr_{k-1} Pr_1 Vr_k Vr_1$
a dipath from $\mathbf{0}$ to $\mathbf{x}$ is composed of the pieces:

- $\gamma_0 : \mathbf{0} \to (\mathbf{x_1}, \mathbf{0})$ serially - one coordinate at a time. OBS, now hold $\kappa(r_k)$ locks on r_k
- $\gamma_1 : (\mathbf{x_1}, \mathbf{0}) \to (\mathbf{x_1}, \mathbf{0}, \mathbf{x_k})$ one coordinate at a time. Now also $\kappa(r_{k-1})$ locks on r_{k-1}
- $\gamma_j : (\mathbf{x_1}, \mathbf{0}, \mathbf{x_{k-j+2}}, \ldots, \mathbf{x_k}) \to (\mathbf{x_1}, \mathbf{0}, \mathbf{x_{k-j+1}}, \ldots, \mathbf{x_k})$. Now $\kappa(r_i)$ locks on $r_{k-j}, \ldots, r_k$.

For $j = 2, \ldots, k-1$.

$\mathbf{x} = (x_1, \ldots, x_1, x_2, \ldots, x_2, \ldots, x_k \ldots x_k)$ is reachable

$T = Pr_1 Pr_2 Vr_1 Pr_3 Vr_2 \ldots Pr_k Vr_{k-1} Pr_1 Vr_k Vr_1$
a dipath from $\mathbf{0}$ to $\mathbf{x}$ is composed of the pieces:

- $\gamma_0 : \mathbf{0} \to (\mathbf{x_1}, \mathbf{0})$ serially - one coordinate at a time. OBS, now hold $\kappa(r_k)$ locks on r_k
- $\gamma_1 : (\mathbf{x_1}, \mathbf{0}) \to (\mathbf{x_1}, \mathbf{0}, \mathbf{x_k})$ one coordinate at a time. Now also $\kappa(r_{k-1})$ locks on r_{k-1}
- $\gamma_j : (\mathbf{x_1}, \mathbf{0}, \mathbf{x_{k-j+2}}, \ldots, \mathbf{x_k}) \to (\mathbf{x_1}, \mathbf{0}, \mathbf{x_{k-j+1}}, \ldots, \mathbf{x_k})$. Now $\kappa(r_i)$ locks on $r_{k-j}, \ldots, r_k$.

For $j = 2, \ldots, k-1$.
Need: No resource is locked above its capacity along γ. Let $\rho_i(\mathbf{y})$ be the number of locks held on r_i at $\mathbf{y} \in T^M$.

$\mathbf{x} = (x_1, \dots, x_1, x_2, \dots, x_2, \dots, x_k \dots x_k)$ is reachable

$T = Pr_1 Pr_2 Vr_1 Pr_3 Vr_2 \dots Pr_k Vr_{k-1} Pr_1 Vr_k Vr_1$

a dipath from $\mathbf{0}$ to $\mathbf{x}$ is composed of the pieces:

- $\gamma_0 : \mathbf{0} \to (\mathbf{x_1}, \mathbf{0})$ serially - one coordinate at a time. OBS, now hold $\kappa(r_k)$ locks on r_k
- $\gamma_1 : (\mathbf{x_1}, \mathbf{0}) \to (\mathbf{x_1}, \mathbf{0}, \mathbf{x_k})$ one coordinate at a time. Now also $\kappa(r_{k-1})$ locks on r_{k-1}
- $\gamma_j : (\mathbf{x_1}, \mathbf{0}, \mathbf{x_{k-j+2}}, \dots, \mathbf{x_k}) \to (\mathbf{x_1}, \mathbf{0}, \mathbf{x_{k-j+1}}, \dots, \mathbf{x_k})$. Now $\kappa(r_i)$ locks on $r_{k-j}, \dots, r_k$.

For $j = 2, \dots, k-1$.

Need: No resource is locked above its capacity along γ. Let $\rho_i(\mathbf{y})$ be the number of locks held on r_i at $\mathbf{y} \in T^M$.

- For γ_0: $\rho_i(\gamma_0(t)) \leq 1$ for $i \neq k$ $\rho_k(\gamma_0(t)) \leq \kappa(r_k)$.
- $\rho_i(\gamma_1(t)) \leq 1$ for $i \neq k-1, k$. $\rho_k(\gamma_1(t)) = k$. $\rho_{k-1}(\gamma_1(t)) \leq \kappa(r_{k-1})$.
- $\rho_i(\gamma_j(t)) \leq 1$ for $i \leq k-j+1$. $\rho_i(\gamma_j(t)) = \kappa(r_i)$ for $i \geq k-j+2$. $\rho_{k-j+1}(\gamma_j(t)) \leq \kappa(r_{k-j+1})$.

Proof of theorem 1

Theorem 1

Let T be a PV thread accessing resources $\mathcal{R}$ with capacity $\kappa : \mathcal{R} \to \mathbb{N}$.
Let T^n be n copies of T run in parallel.
T^n is deadlock free for all n if and only if T^M is deadlock free, where $M = \Sigma_{r \in \mathcal{R}} \kappa(r)$.

Prove: If there is a deadlock in T^n, then there is a deadlock in T^M.

Proof of theorem 1

Theorem 1

Let T be a PV thread accessing resources $\mathcal{R}$ with capacity $\kappa : \mathcal{R} \to \mathbb{N}$.
Let T^n be n copies of T run in parallel.
T^n is deadlock free for all n if and only if T^M is deadlock free, where $M = \Sigma_{r \in \mathcal{R}} \kappa(r)$.

Prove: If there is a deadlock in T^n, then there is a deadlock in T^M.
For $n < M$: Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a (reachable) deadlock in T^n.
I.e. $x_i = Pr_{j(i)}$ and $\rho_{j(i)}(\mathbf{x}) = \kappa(r_{j(i)})$ or $x_i = \top$.
Then $\tilde{\mathbf{x}} = (x_1, \ldots, x_n, \top, \ldots, \top) \in T^M$ is a (reachable) deadlock in T^M, since

$$\rho_i(\tilde{\mathbf{x}}) = \rho_i(\mathbf{x}) \text{ for all } i$$

Proof of theorem 1

Theorem 1

Let T be a PV thread accessing resources $\mathcal{R}$ with capacity $\kappa : \mathcal{R} \to \mathbb{N}$.
Let T^n be n copies of T run in parallel.
T^n is deadlock free for all n if and only if T^M is deadlock free, where $M = \Sigma_{r \in \mathcal{R}} \kappa(r)$.

Prove: If there is a deadlock in T^n, then there is a deadlock in T^M.
For $n < M$: Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a (reachable) deadlock in T^n.
I.e. $x_i = Pr_{j(i)}$ and $\rho_{j(i)}(\mathbf{x}) = \kappa(r_{j(i)})$ or $x_i = \top$.
Then $\tilde{\mathbf{x}} = (x_1, \ldots, x_n, \top, \ldots, \top) \in T^M$ is a (reachable) deadlock in T^M, since

$$\rho_i(\tilde{\mathbf{x}}) = \rho_i(\mathbf{x}) \text{ for all } i$$

$\tilde{\mathbf{x}}$ is reachable: There is a serial path to $(\mathbf{0}, \top, \ldots, \top)$. Compose with $(\gamma(t), \top, \ldots, \top)$, where $\gamma(t) : \mathbf{0} \to \mathbf{x} \in T^n$

Proof of Theorem 1. $n > M$

$\mathbf{x} = (x_1, \ldots, x_n)$ is a (reachable) deadlock in T^n.
Construct the directed wait for graph $G(\mathbf{x}) = (V, E)$.

$$V = \{x_1, \ldots, x_n\}$$

There is an edge $E(x_i, x_k)$ if $x_i = Pr_j$ and $\rho_j(x_k) = 1$.

Proof of Theorem 1. $n > M$

$\mathbf{x} = (x_1, \dots, x_n)$ is a (reachable) deadlock in T^n.
Construct the directed wait for graph $G(\mathbf{x}) = (V, E)$.

$$V = \{x_1, \dots, x_n\}$$

There is an edge $E(x_i, x_k)$ if $x_i = Pr_j$ and $\rho_j(x_k) = 1$.
Properties of $G(\mathbf{x})$

- If $x_i = \top$, then the vertex x_i is isolated.
- If $x_i = Pr_j$, then the vertex x_i has $\kappa(r_j)$ outgoing edges.
- There are circuits in $G(\mathbf{x})$: Start a walk at a non-isolated vertex. If x_j is the target of an edge, then $x_j \neq \top$, so the walk continues. $G(\mathbf{x})$ is finite, so there is a circuit $\mathcal{L}$.

Proof of Theorem 1. $n > M$

$\mathbf{x} = (x_1, \ldots, x_n)$ is a (reachable) deadlock in T^n.
Construct the directed wait for graph $G(\mathbf{x}) = (V, E)$.

$$V = \{x_1, \ldots, x_n\}$$

There is an edge $E(x_i, x_k)$ if $x_i = Pr_j$ and $\rho_j(x_k) = 1$.
Properties of $G(\mathbf{x})$

- If $x_i = \top$, then the vertex x_i is isolated.
- If $x_i = Pr_j$, then the vertex x_i has $\kappa(r_j)$ outgoing edges.
- There are circuits in $G(\mathbf{x})$: Start a walk at a non-isolated vertex. If x_j is the target of an edge, then $x_j \neq \top$, so the walk continues. $G(\mathbf{x})$ is finite, so there is a circuit $\mathcal{L}$.

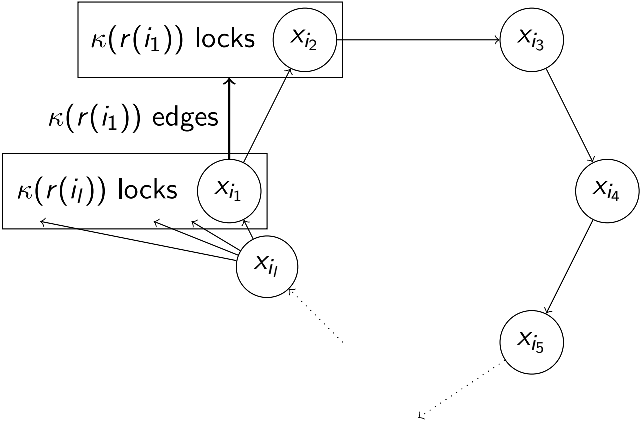
Let $\uparrow \mathcal{L}$ be the vertices reachable from $\mathcal{L}$.
Let $\tilde{\mathbf{x}} = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})$ where x_{i_j} are all the vertices in $\uparrow \mathcal{L}$

$\tilde{\mathbf{x}} = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})$ where x_{i_j} are all the vertices in $\uparrow \mathcal{L}$, future of a circuit in the wait-for-graph.
Claim:

1. $\tilde{\mathbf{x}}$ is a (reachable) deadlock.
2. $m \leq M$

Reachability: Let $\gamma : \mathbf{0} \to \mathbf{x}$ in T^n. The restriction to $Ti_1, \ldots Ti_m$ is a dipath $\mathbf{0} \to \tilde{\mathbf{x}}$

The restricted Wait for Graph - future of the loop



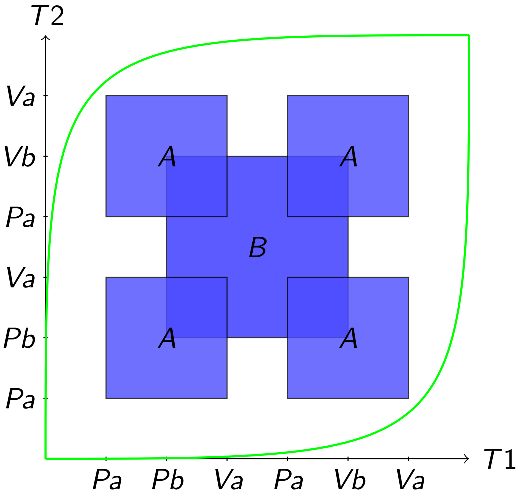
$x_{i_j} \neq \top$, so $x_{i_j} = P(r(i_j))$. In $G(\mathbf{x})$, $x_{i_j} = P(r(i_j))$ had $\kappa(r(i_j))$ outgoing edges. It still has in $G(\tilde{\mathbf{x}})$, since $\uparrow \mathcal{L}$ contains all targets.

Hence, $\tilde{\mathbf{x}}$ is a deadlock.

$m \leq M$

- In $G(\tilde{\mathbf{x}})$, all the m vertices are targets. Hence, they hold a lock on a resource.
- The maximal number of locks at an allowed state is $M = \Sigma_{r \in \mathcal{R}} \kappa(r)$
- Hence, there are at most M vertices in $G(\tilde{\mathbf{x}})$ (Less, if some x_i hold a lock on more than one resource.)

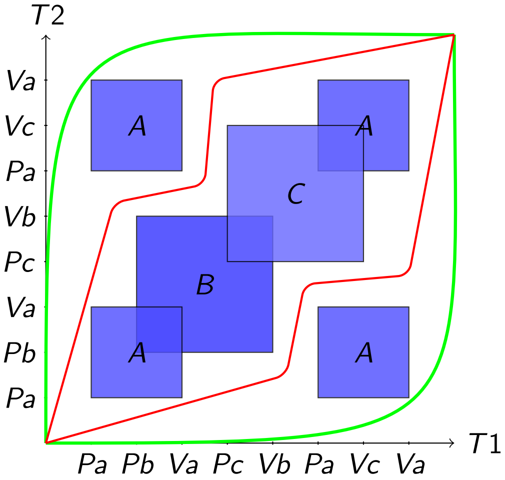
Now to something different: Serializability



Two executions up to dihomotopy. Equivalent to the serial executions $T1.T2$ and $T2.T1$

Serializability

$T = PaPbVaPcVbPaVcVa$ is not serializable.



4 Executions up to dihomotopy. Two serial executions (green). Two non-serializable executions (red).

Serializability - a cut-off theorem

Definition

$T1|T2\ldots|Tn$ is serializable if all execution paths are dihomotopic to a serial execution $Ti1.Ti2....Tin$.

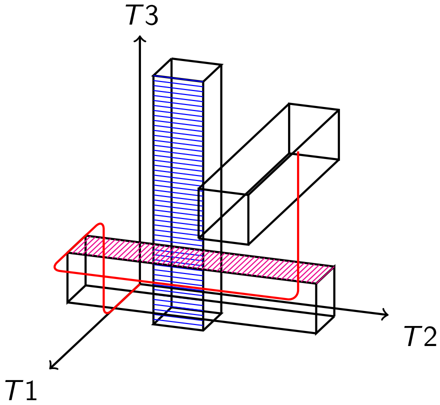
Theorem 3

Let T be a PV-thread accessing resources $\mathcal{R}$ all of capacity 1.
Let T^n be n copies of T run in parallel. Then T^n is serializable if and only is T^2 is serializable.

In general studying pairwise interaction is not enough

Example

Let $T1 = PcVcPaVa$, $T2 = PcVcPbVb$, $T3 = PaVaPbVb$,
Each pair Ti, Tj shares one resource.

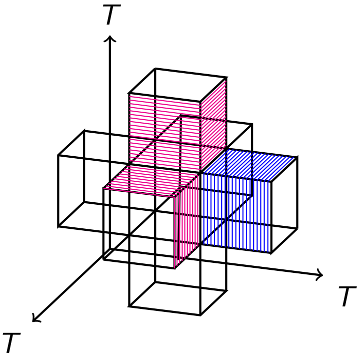


Schedules as in Raussens Trace algorithm

When $\kappa = 1$

- All conflict n-rectangles are $\times_{k=1}^{n} I_k$, $I_k = I$ except for two directions $I_i =]a_i, b_i[$, $I_j =]a_j, b_j[$
- One choice at a given such n-rectangle "above or below" in the ij-plane - i waits or j waits.

$T = PaVa$, T^3

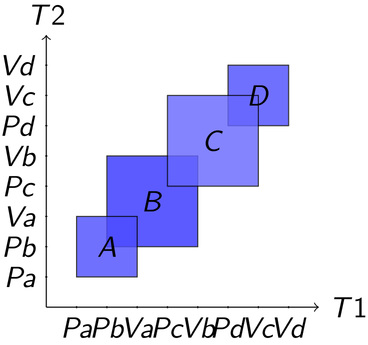


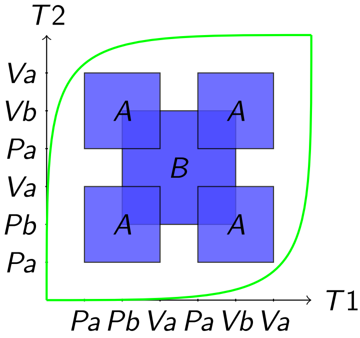
3! serial executions, pairwise inequivalent.

T^2 serializability

- If choice at one rectangle implies choice at all other rectangles - below all or above all rectangles
- Equivalently: Only schedules $J = (\{1\}, \{1\}, \dots, \{1\})$ and $J = (\{2\}, \{2\}, \dots, \{2\})$ are allowed.
- Equivalently: The closure of the forbidden area under adding unreachable and unsafe areas is connected.

OBS: Two phase locking is not required.

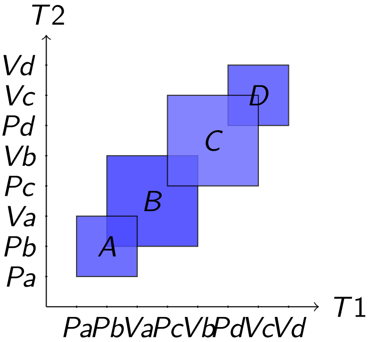


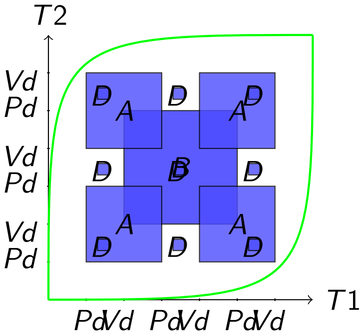


T^2 serializability

- If choice at one rectangle implies choice at all other rectangles - below all or above all rectangles
- Equivalently: Only schedules $J = (\{1\}, \{1\}, \ldots, \{1\})$ and $J = (\{2\}, \{2\}, \ldots, \{2\})$ are allowed.
- Equivalently: The closure of the forbidden area under adding unreachable and unsafe areas is connected.

OBS: Two phase locking is not required.





T^n

If T is non-trivial, all the $n!$ serial executions are non-equivalent, since

- They have different schedules wrt. the $\frac{n\cdot(n-1)}{2}$ n-rectangles induced by just one PV-interval.
- Equivalently: Their projections to at least one of the T^2 are non-equivalent.

Proof of Theorem 3

Suppose T^2 is serializable, then

1. Let $]a^r, b^r[$ correspond to a lock Pr, Vr. There are $n!$ schedules for the rectangles $\{\times_{k=1}^n I_k, I_k =]a^r, b^r[\text{ for } k = i, j, i < j \in [1:n]\}$ - corresponding to the serial executions.
2. Fix i, j There are two schedules - i last for all or j last for all the rectangles $\{I \times I \times \cdots \times]a_s^r, b_s^r[\times I \cdots \times]a_t^r, b_t^r[\times I \times \cdots \times I\}$ where $]a_s^r, b_s^r[\times]a_t^r, b_t^r[$ are forbidden rectangles in T^2
3. Choose one of the $n!$ schedules for 1); that fixes all schedules in 2). Hence, there are $n!$ inequivalent schedules.
4. These are all realized by serial executions.

Serializability for $\kappa > 1$

- There are non-serializable executions.
- All serial executions are equivalent (all boundary 2-cells are allowed)
- The scheduling algorithm calculates components of the trace space.
- More than one component $\Leftrightarrow$ non-serializable.

More to do

- Serializability for $\kappa > 1$ - guess: There is a homological obstruction . This needs to be described specifically for the symmetric case.
- Cut-off for other properties - linearizability?