WCRT for Synchronous Programs: Studying the Tick Alignment Problem

B. Bodin, M. Mendler, P. Roop, J.J. (Hugh) Wang

Structure of Talk

1. Motivation
2. Synchronous Multi-threading
3. Tick Cost Automata (TCA) and the Tick Alignment Problem (TAP)
4. Algorithms for TAP
5. Conclusion

1
MOTIVATION

Emerging Trade on Synchronous WCRT

(1) Explicit Flow Traversal [Boldt et.al. SLA++P'07, Mendler et.al. DATE'09] (Plus-Max Technique)
   – efficient (linear: sum of thread states), but large overestimates

(2) Implicit Path Enumeration [Ju et.al. DAC'09, RTS'12]
   – exact ILP constraint solving, NP-hard

(3) Model Checking [Roop et.al. CASES'09]
   – exact (exponential: product of thread states + binary search)

(4) State Exploration [Kuo et.al. DAC'11, Yip et.al. ICCPS'13]
   – exact (exponential: product of thread states)

(5) (Iterative) Narrowing [Wang et.al. CASES'13, Raymond et.al. RePP'14]
   – (i) get WCRT approximation; (ii) validate critical path in exact model; (iii) if infeasible, narrow WCRT approximation; repeat.

(6) High-level Flow Facts [Raymond et.al. RePP'14]

Emerging Trade on Synchronous WCRT

- ... shall we continue to develop
  - WCRT using general purpose algorithms (ILP, SAT, ...)
  - for particular SP languages and particular PRET architectures ?
- ... or, is there any systematics behind this work?
  - common benchmark suites to compare results ?
  - results on algorithmic complexity of WCRT ?
  - is there a canonical WCRT Problem ?

WCRT Analysis: A Many-faceted Problem

- Data-dependent control path sensitization
- Thread preemption contexts
- Bus synchronisation and
- C
- M
- Int
- Sch
- …

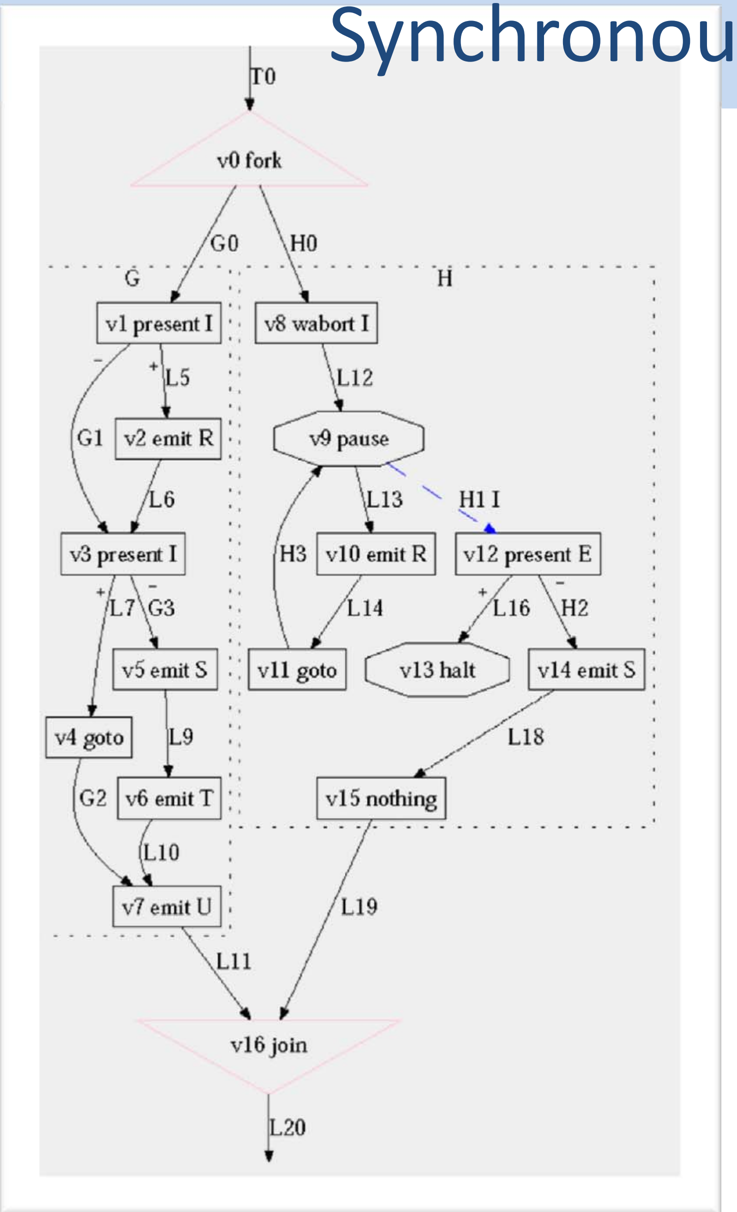
What if we abstract from all these complications ?

Q: Does WCRT become trivial ?

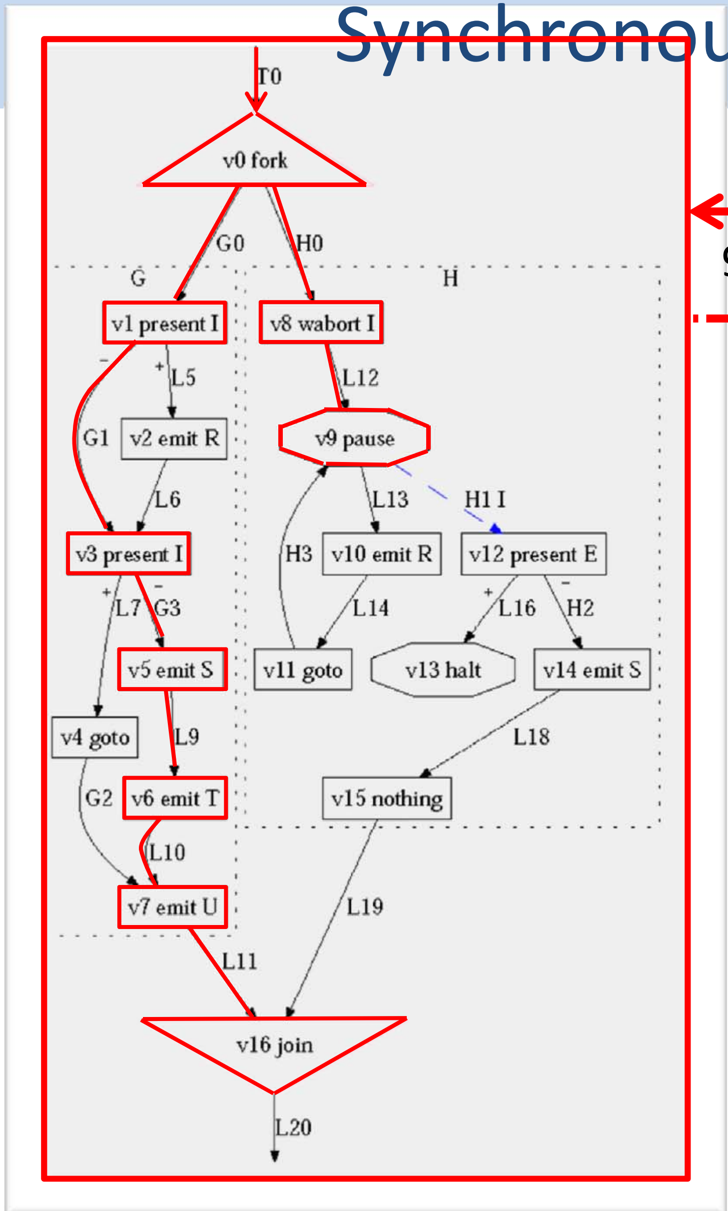
A: Open Problem …

2
SYNCHRONOUS MULTI-THREADING

Synchronous Multi-Threading



Synchronous Multi-Threading



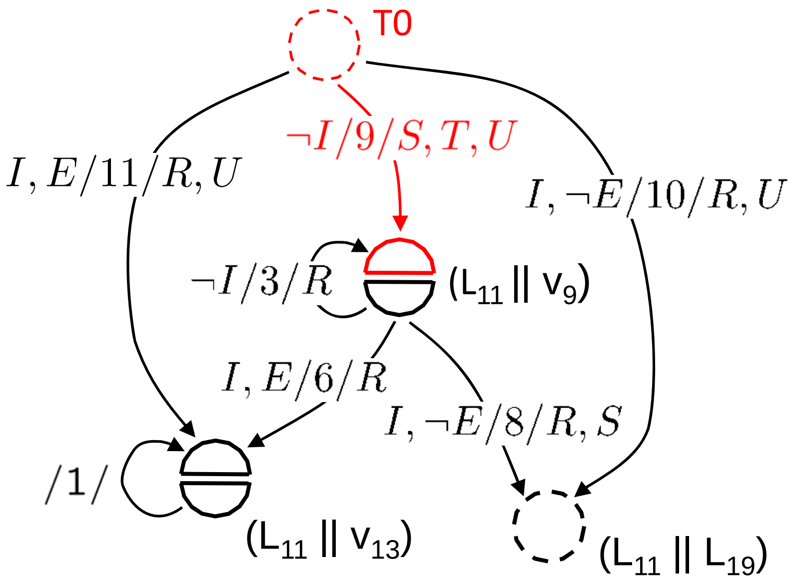
instantaneous entry

9 instruction cycles

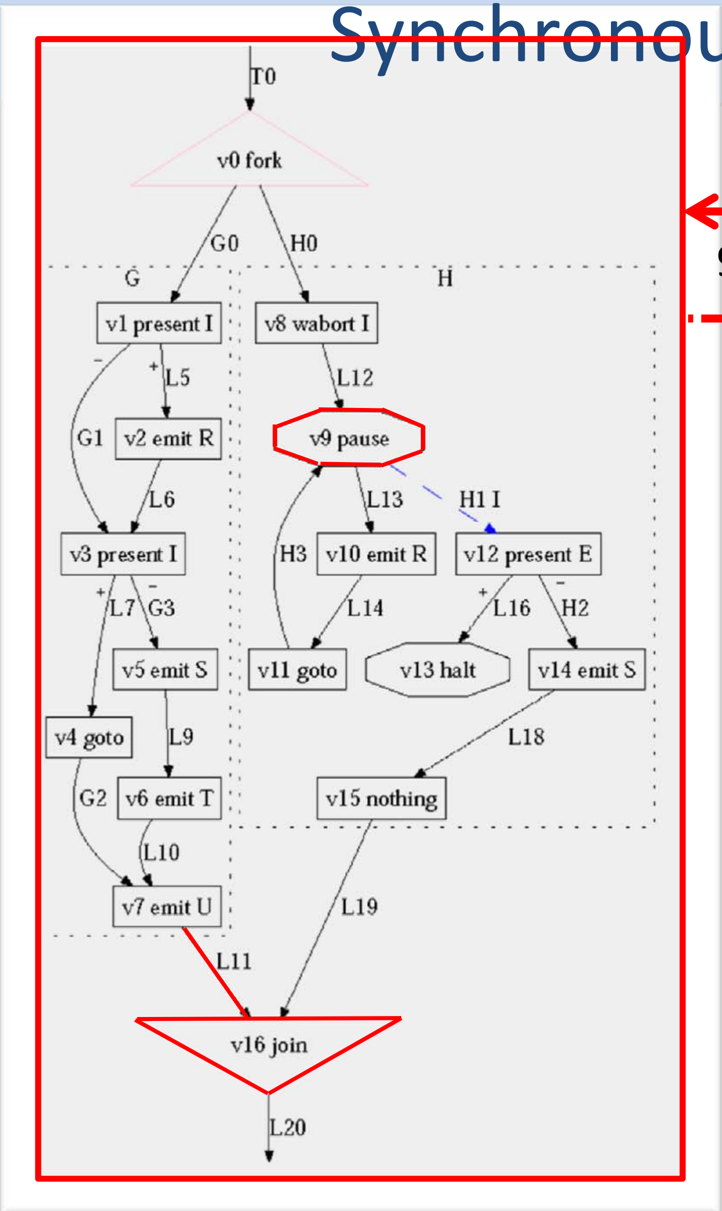
pause

I absent

S,T,U emitted



Synchronous Multi-Threading



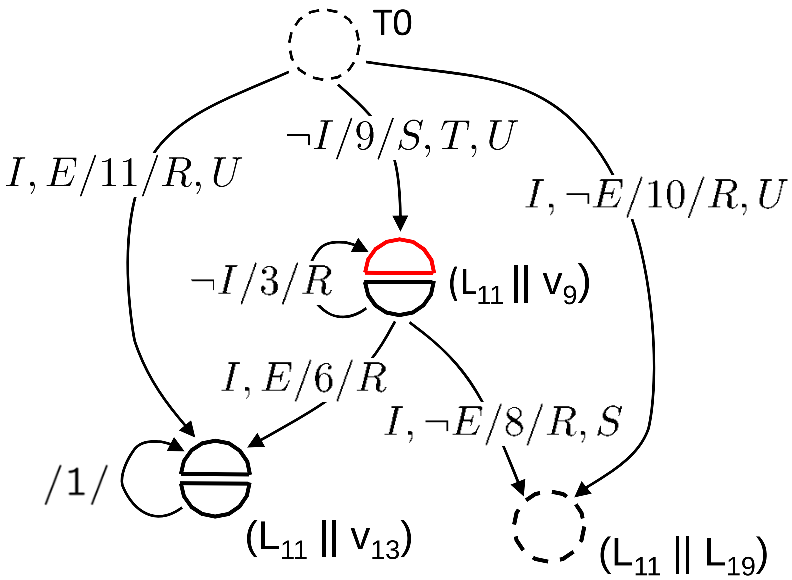
instantaneous entry

9 instruction cycles

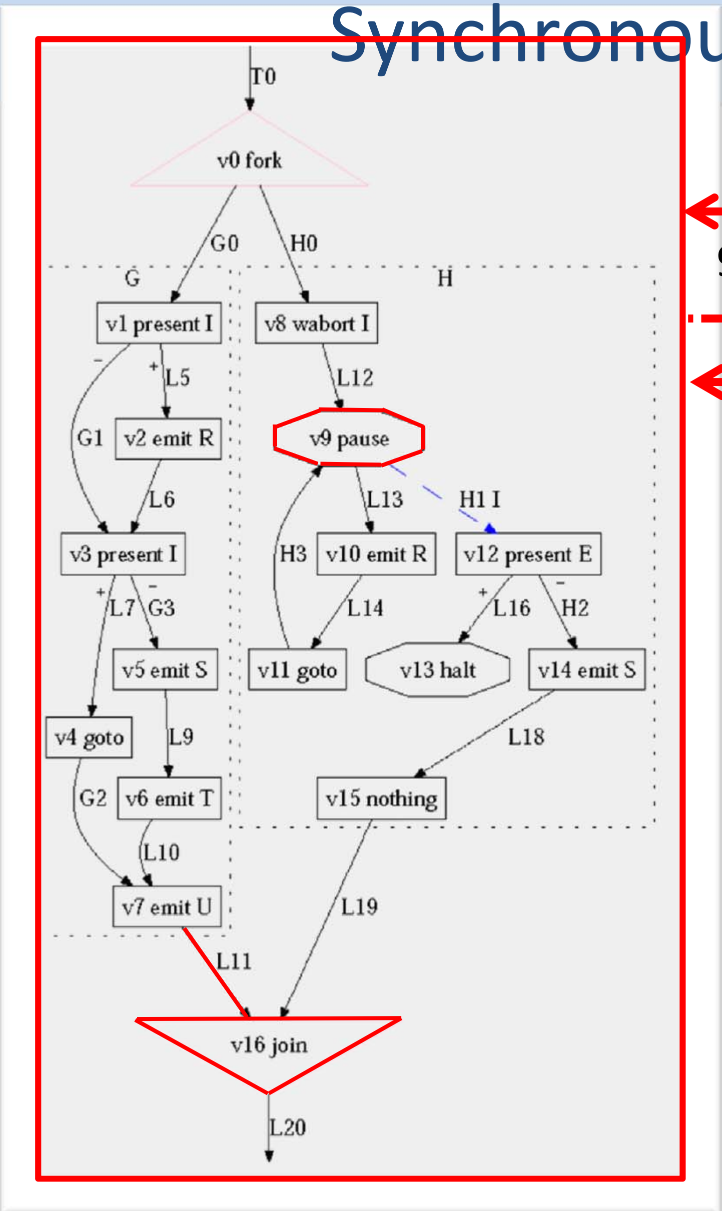
pause

I absent

S,T,U emitted



Synchronous Multi-Threading

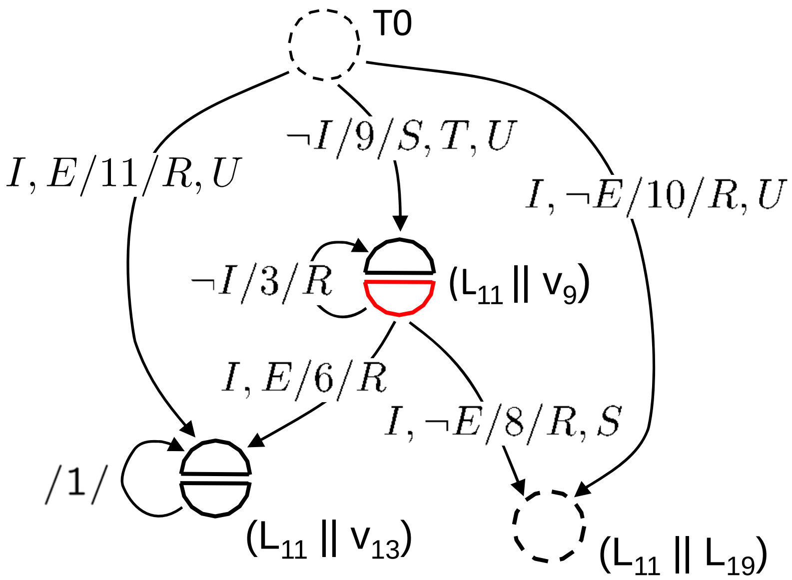


instantaneous entry — I absent

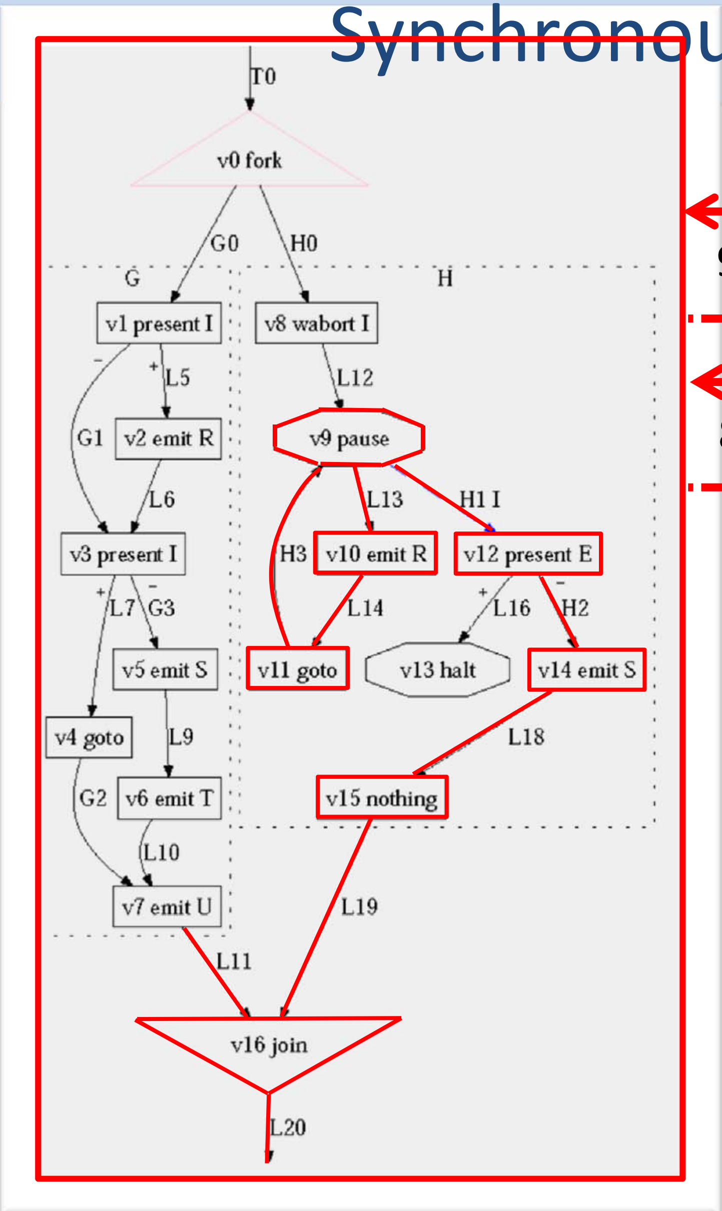
9 instruction cycles

pause — S,T,U emitted

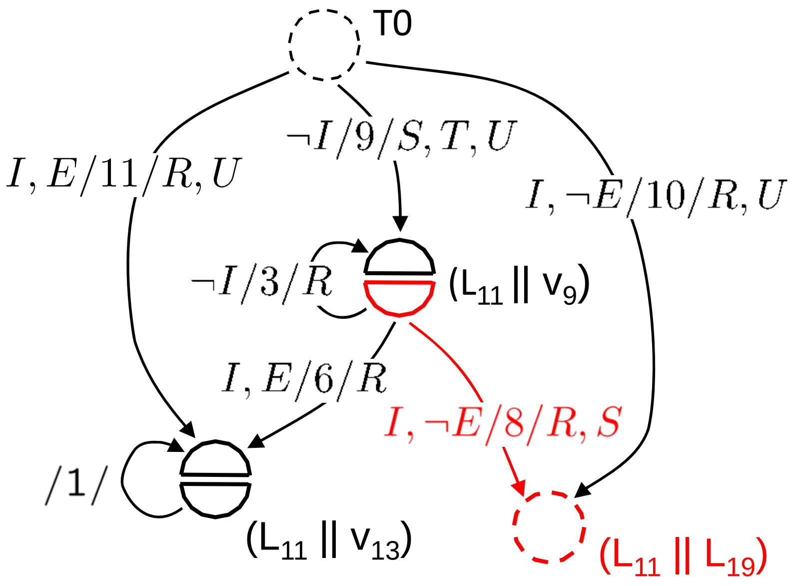
tick — I present, E absent



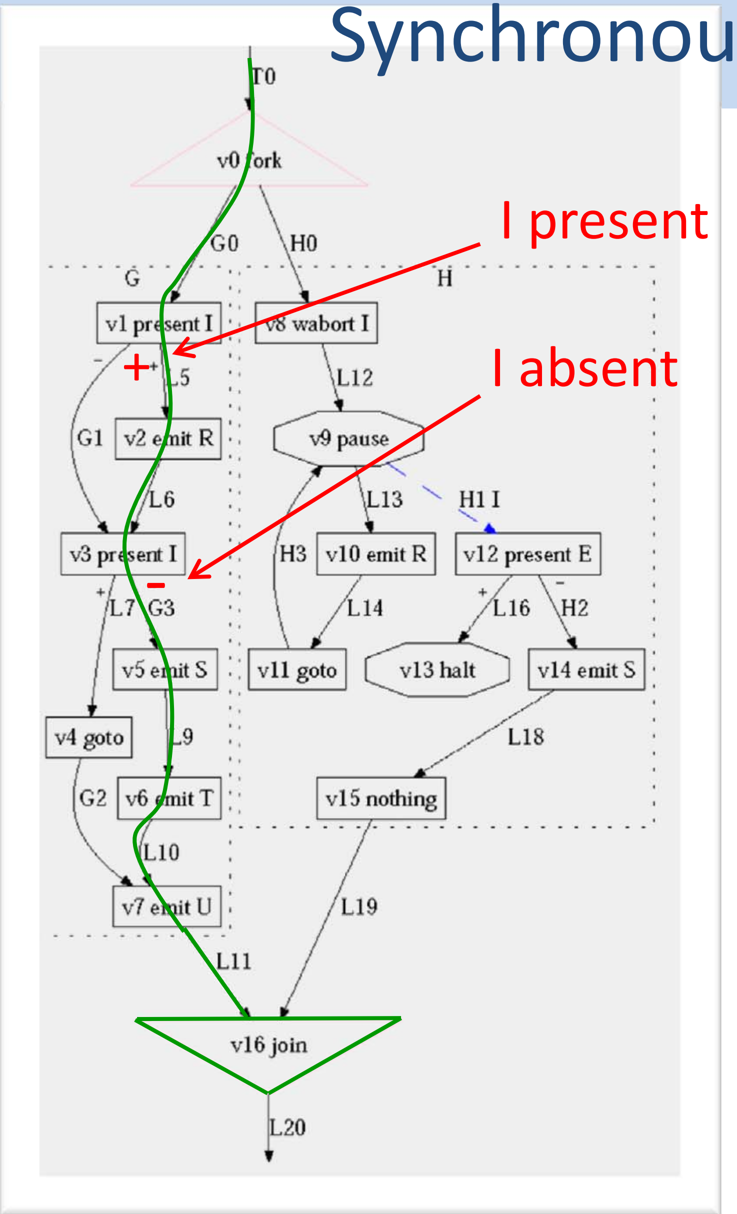
Synchronous Multi-Threading



instantaneous entry		I absent
9 instruction cycles		
pause		S,T,U emitted
tick		I present, E absent
8 instruction cycles		
instantaneous exit		R, S emitted



Synchronous Multi-Threading



(1) Intra-thread Exclusion

Path sensitization problem:

sequential flow not sensitizable

=> Precise WCRT is data-dependent

NP hard !

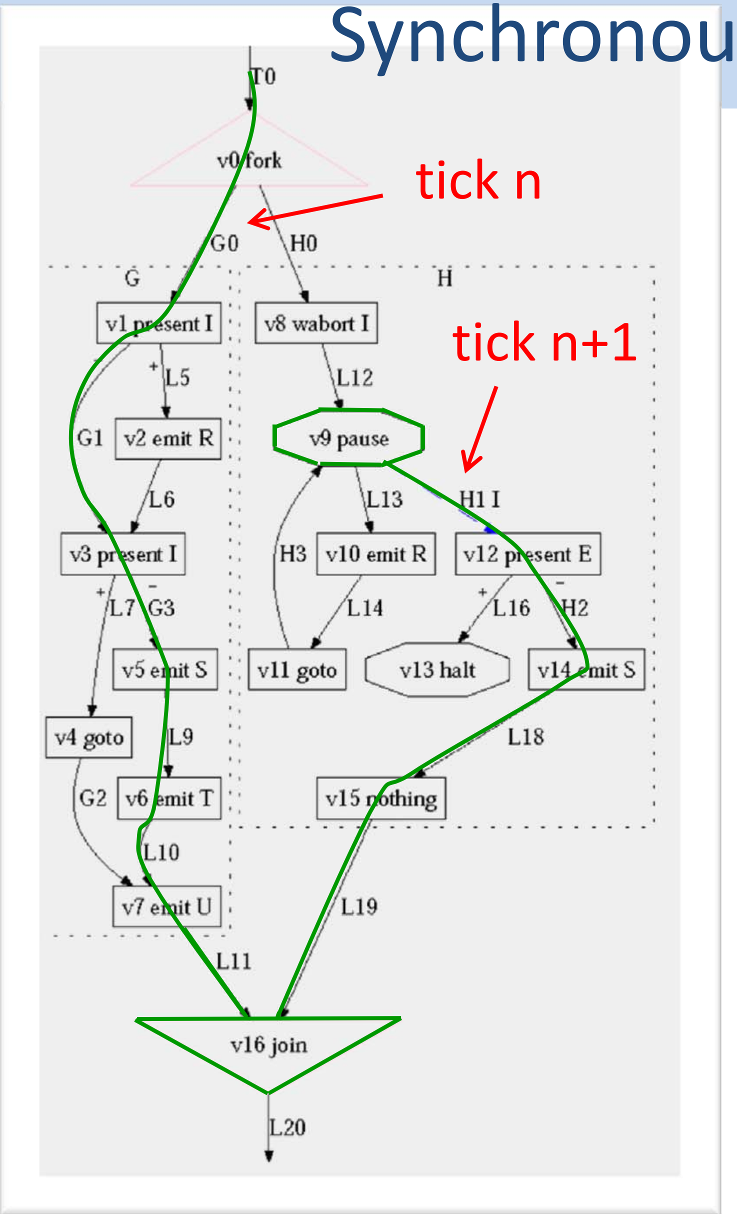
Solution: Data Abstraction

=> Over-approximation

=> Longest Path Problem

=> PTime

Synchronous Multi-Threading



(2) Inter-thread Exclusion

Tick Alignment Problem (TAP):

concurrent flow not alignable

Data Abstraction

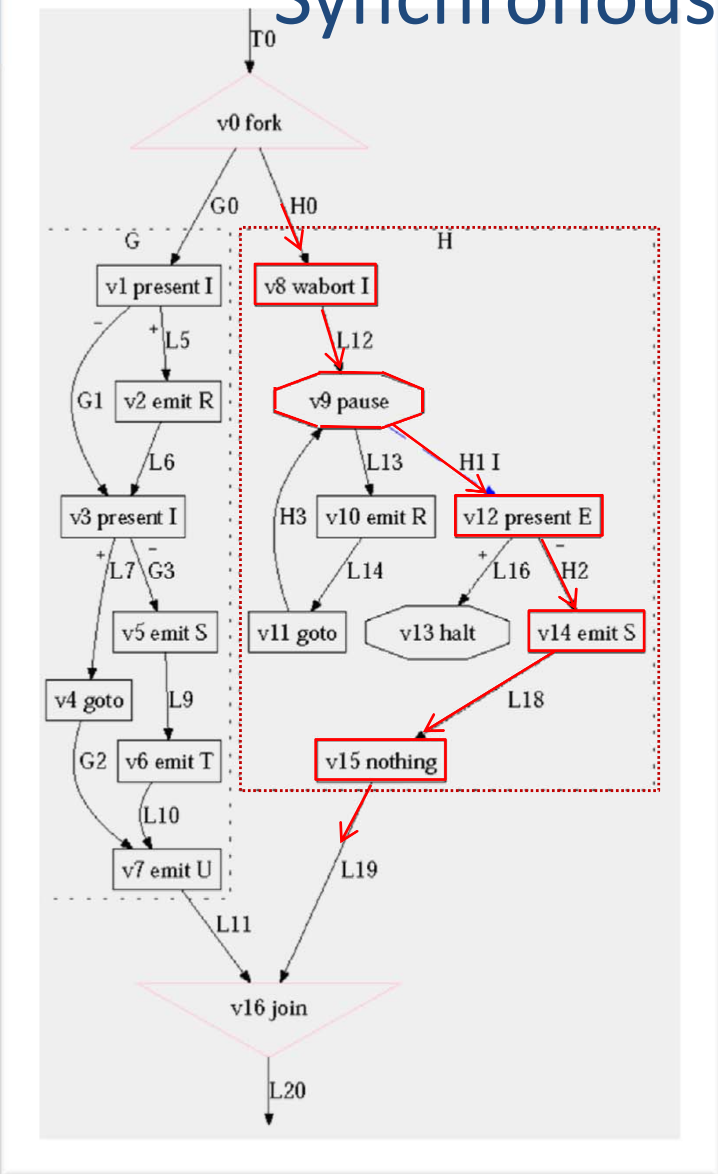
What if tick states of threads do not depend on data ?

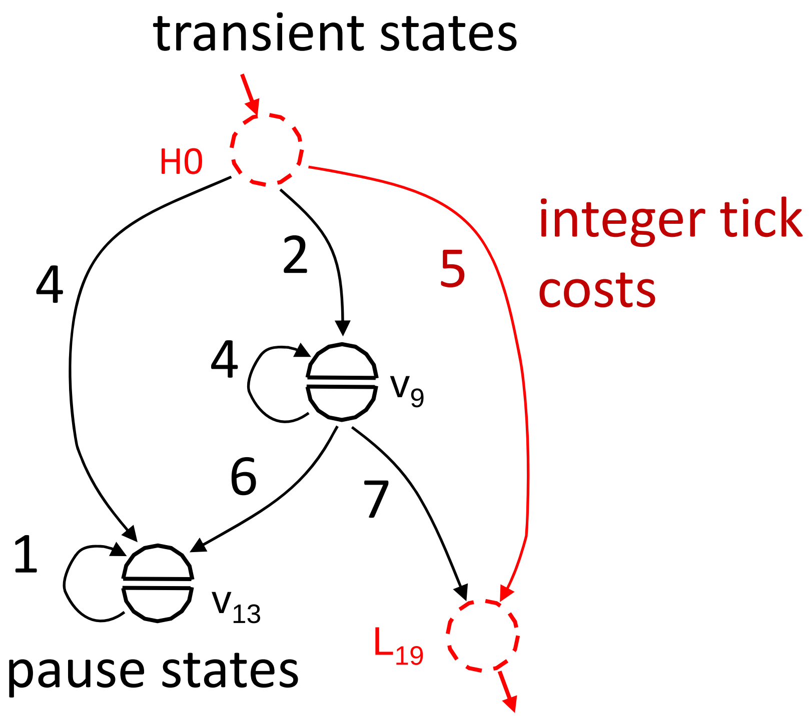
Complexity of (Pure) TAP ?

??

3
TICK COST AUTOMATA (TCA) & TICK ALIGNMENT PROBLEM (TAP)

Synchronous Tick Cost Automata





Max-Plus Formal Power Series

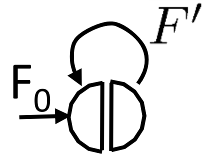
Max-Plus Algebra $\mathbb{N}_\infty = (\mathbb{N} \cup -\infty, \oplus, \odot, -\infty)$

plus

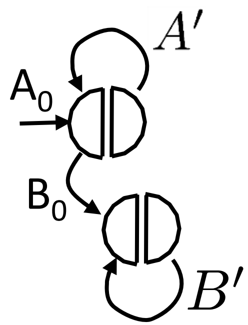
max

Formal Power Series $\mathbb{N}^*_\infty[X]$

$$\begin{aligned} F &= F_0 \oplus F_1\, X \oplus F_2\, X^2 \oplus F_3\, X^3 \cdots = F_0 \oplus X F' \\ F' &= F_1 \oplus F_2\, X \oplus F_3\, X^2 \oplus F_4\, X^3 \cdots \end{aligned}$$



Scalars $n = n \oplus -\infty X \oplus -\infty X^2 \oplus \cdots$



Sequential Composition (Expansion Law)

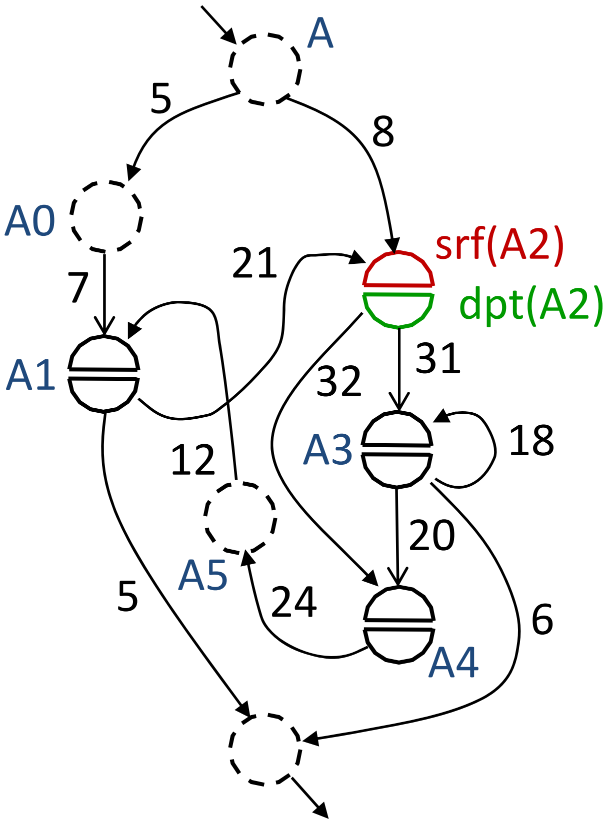
$$(A_0 \oplus X A') \,;\, (B_0 \oplus X B') = (A_0 \odot B_0) \oplus X (A' \odot B_0 \oplus B')$$

$$A_0 \,;\, (B_0 \oplus X B') = (A_0 \odot B_0) \oplus X B'$$

Synchronous Product (Expansion Law)

$$(A_0 \oplus X A') \parallel (B_0 \oplus X B') = (A_0 \odot B_0) \oplus X (A' \| B')$$

TCA = Recurrence Equations in $\mathbb{N}_\infty[X]$



Recurrence

$$A = 5 \,;\, A0 \oplus 8 \,;\, srf(A2)$$
$$A0 = 7 \,;\, srf(A1)$$
$$srf(A1) = 0 \oplus X\, dpt(A1)$$
$$srf(A2) = 0 \oplus X\, dpt(A2)$$
$$dpt(A1) = \ldots$$

Max-Plus Expansion

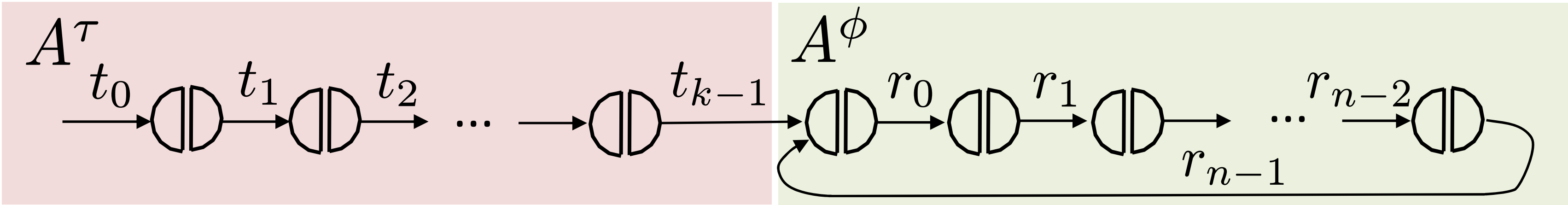
$$A[X] = 12 \oplus 32X \oplus 36X^2 \oplus 36X^3 \oplus \cdots .$$

Worst Case Reaction Time

$$\mathsf{wcrt}(A) = A[0] = 36$$

Linear TCAs (l-TCA)

$$A \quad = \quad A^\tau \oplus X^k A^\phi$$



$\tau(A) =_{df} k$ transient length

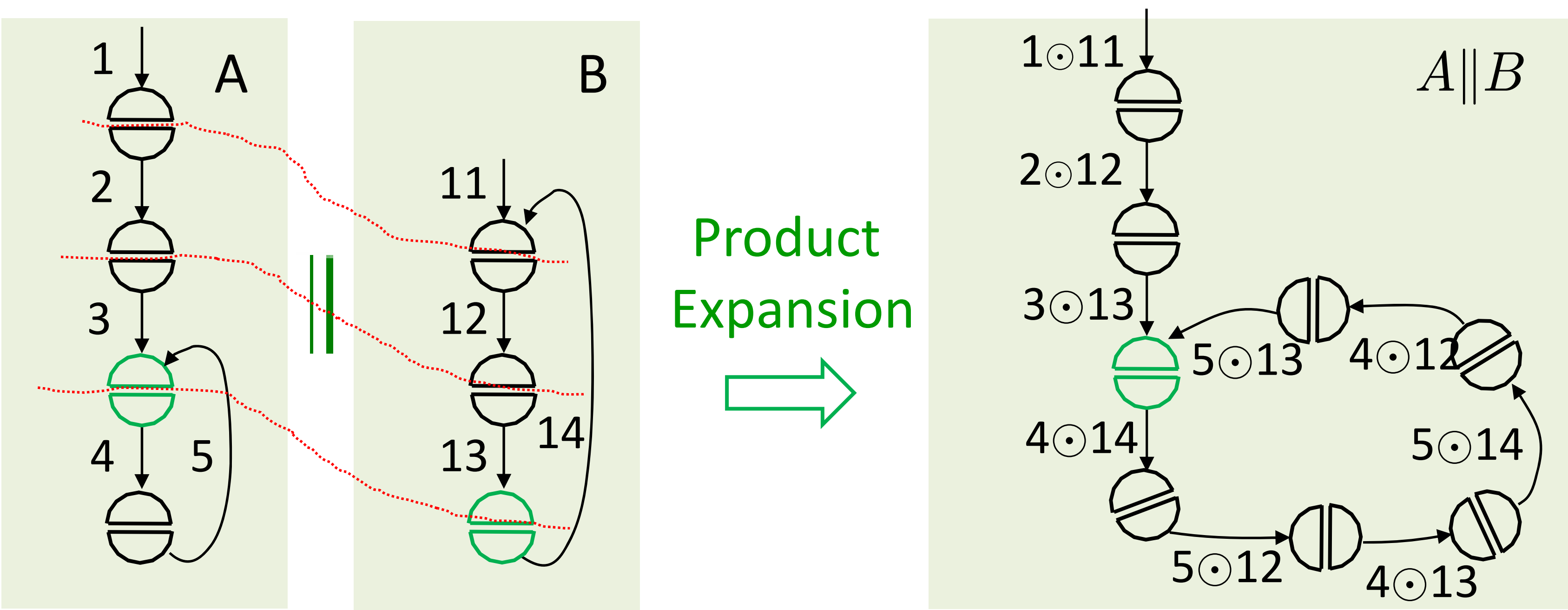
$\phi(A) =_{df} n$ cycle length

A is a monocyclic TCA (m-TCA) if $\tau(A) = 0$

Proposition 1 *Every single-threaded TCA can be reduced to an equivalent l-TCA in polynomial time.*

Synchronous Product (Multi-threaded)

$$A = \bigoplus_i A(i)X^i \qquad B = \bigoplus_i B(i)X^i \qquad (A \parallel B) \quad =_{df} \quad \bigoplus_i (A(i) \odot B(i))X^i$$



$\tau(A) = 3$
$\phi(A) = 2$

$\tau(B) = 1$
$\phi(B) = 3$

$\tau(A \| B) = 3 = max(\tau(A), \tau(B))$
$\phi(A \| B) = 6 = \phi(A)\phi(B)$

TAP by State Exploration

Definition 1 *The Tick Alignment Optimisation Problem TAP is the problem to compute wcrt*(T) *for an arbitrary parallel composition* $T = T_1 \| T_2 \| \cdots \| T_n$ *of (single-threaded) TCAs* T_i.

Proposition 2 *The TAP for l-TCAs can be reduced to TAP for m-TCAs in polynomial time.*

Algorithm 1 (Reachability, State Expansion)
Given TAP $T = T_1 \| T_2 \| \cdots \| T_n$ *with m-TCAs:*

1. *Repeatedly use the Expansion Law to obtain the reduced linear form* T^* *of* T.
2. *Compute wcrt*(T^*).

EXPONENTIAL

$\Rightarrow$ Time complexity $\Theta(n\, lcm(\phi_1, \ldots, \phi_n))$.

4
ALGORITHMS FOR TAP

m-TAP Number Theory

Proposition 3 (Chinese Remainder Theorem)
Given TAP $T = T_1 \| T_2 \| \cdots \| T_n$ *with m-TCAs* T_i. *A candidate sum of the mTAP* T,

$$T_1(S_1) + T_2(S_2) + \cdots + T_n(S_n),$$

for transition offsets $0 \leq S_i < \phi_i$, *is aligned in* T *iff*

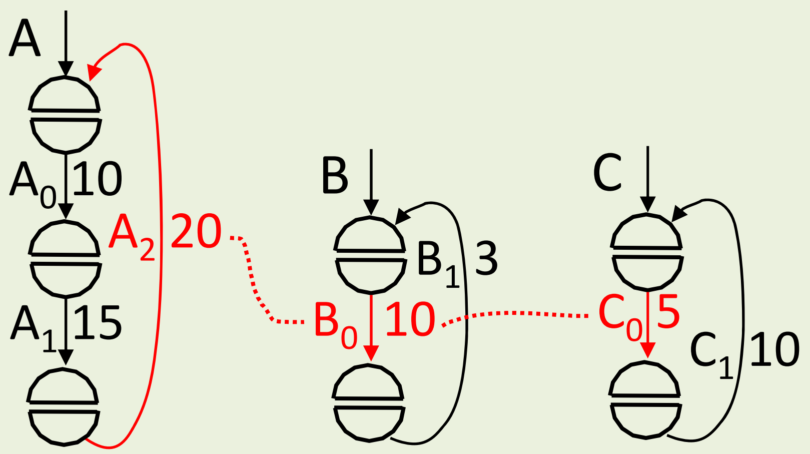
$$\forall 1 \leq i < j < n.\, S_i \equiv_{g_{ij}} S_j$$

where $g_{ij} =_{df}$ *gcd*(ϕ_i, ϕ_j).

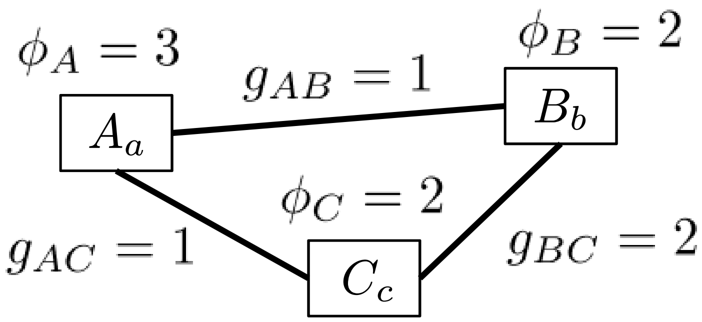
m-TAP Number Theory

$$\text{wcrt}(A \parallel B \parallel C) = max(\text{AlignedSums})$$

$$\text{AlignedSums} = \{A_{k \bmod \phi_A} + B_{k \bmod \phi_B} + C_{k \bmod \phi_C} \mid k \geq 0\}.$$



$$2 \equiv_1 0$$
$$0 \equiv_2 0$$
$$0 \equiv_1 2$$



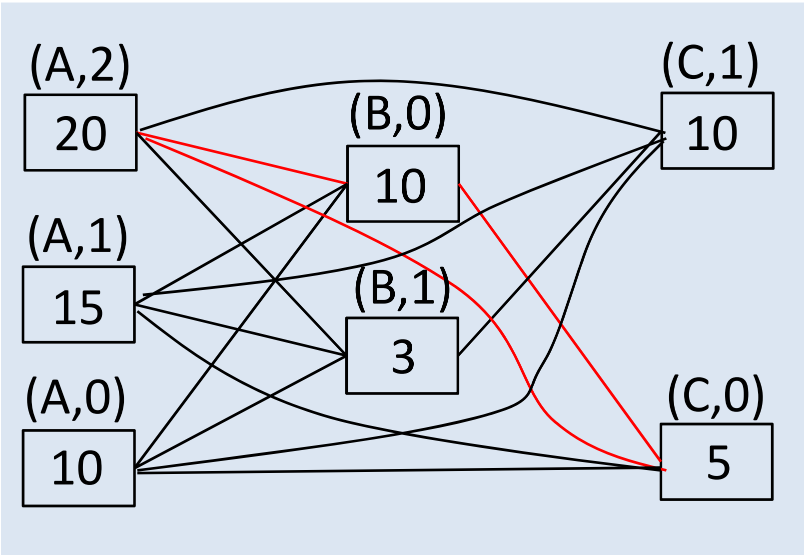
$$g_{XY} =_{df} gcd(\phi_X, \phi_Y)$$

$$a \equiv_{g_{AB}} b$$
$$b \equiv_{g_{BC}} c$$
$$c \equiv_{g_{CA}} a$$

Tick Alignment Graph TAG

TAP $T = (T_1, T_2, \ldots, T_n)$ $\quad \phi_i =_{df} \phi(T_i)$ $\quad g_{i_1 i_2} =_{df} gcd(\phi_{i_1}, \phi_{i_2})$

The weighted undirected Tick Alignment Graph G_T



TAG $G_T = (V_T, E_T, w_T)$

$$V_T =_{df} \{(i,j) \mid 1 \leq i \leq n, 0 \leq j \leq \phi_i\}$$

$$E_T =_{df} \{(i_1, j_2) \leftrightarrow (i_2, j_2) \mid j_1 \equiv_{g_{i_1 i_2}} j_2\}$$

$$w_T(i,j) =_{df} T_i(j)$$

Proposition 4 *Given a TAP T of size $O(nm)$ the associated TAG G_T can be computed in $O(n^2 m^2)$ time.*

Reduction of mTAP to Clique Problem

Proposition 5

- *A candidate sum* S *of the mTAP* T*,*

$$T_1(S_1) + T_2(S_2) + \cdots + T_n(S_n),$$

for $0 \leq S_i \leq \phi_i$*, is aligned in* T *iff the nodes*

$$S = \{(i, S_i) \mid 1 \leq i \leq n\}$$

form a clique in the TAG G_T*.*

- *To check a candidate sum* S *is a clique takes* $O(n^2)$ *time.*

- *wcrt*(T) $=$ *max*$\{w(S) \mid S$ *clique in* $G_T\}$
(Maximum Weight Clique Problem MWCP)

Exact MWCP Algorithms

MWCP is known to be NP-complete 

Algorithm 2 (ILP) [Pardalos & Xue 1992]

$$\mathit{max} \sum_{(i,j)\in V_T} w_T(i,j)\cdot x_{i,j}$$

$$\text{subject to } \; x_{i_1,j_1} + x_{i_2,j_2} \leq 1, \text{ for all } (i_1,j_1) \not\leftrightarrow_{E_T} (i_2,j_2)$$

$$x_{i,j} \in \{0,1\}, \text{ for all } (i,j) \in V_T$$

- depends on cleverness of general purpose ILP
- original problem structure lost
- difficult to twist & control search strategy

Maximum Weighted Clique Algorithms

Algorithm 3 (Branch and Bound) [K. Yamaguchi & S. Masuda 2008]

$C_{max} = MWC(G_T, -1)$ where

$MWC((V, E), \theta)$

1 compute ordering $V = \{\pi_0, \pi_1, \ldots, \pi_{|V|-1}\}$ and upper bound weights $UB(\pi_i) \geq w(D)$ any clique D with $\pi_i \in D \subseteq \{\pi_0, \pi_1, \ldots, \pi_i\}$;

2 $C := \emptyset$;

3 for $i := |V| - 1$ downto 0 do

3.1 if $UB(\pi_i) \leq \theta$ then return C

3.2 $C' := MWC((V, E)@\pi_i, \theta - w(\pi_i))$

3.3 if $C' \neq \emptyset$ then $C := C' \cup \{\pi_i\}$ and $\theta := w(C' \cup \{\pi_i\})$

Reduction of MWCP to mTAP ?

Proposition 6 *Let G be an undirected graph of size n. Subject to the time and space complexity of constructing $O(n^2)$ distinct prime numbers p_i, the MWCP for G can be reduced the mTAP for T_G of size $O(n \, \Pi_i \, p_i)$.*

= Exponential time/space reduction to "explicit" TAP

- Not much use for lower bound on complexity of mTAP
- If there is no polynomial reduction from MWCP, maybe mTAP is polynomial ?

WHO CARES ABOUT EXACT WCRT ?

Approximate MWCP

Good upper bounds may be enough !

- There are two basic techniques
  - „Plus-Max“ Approach

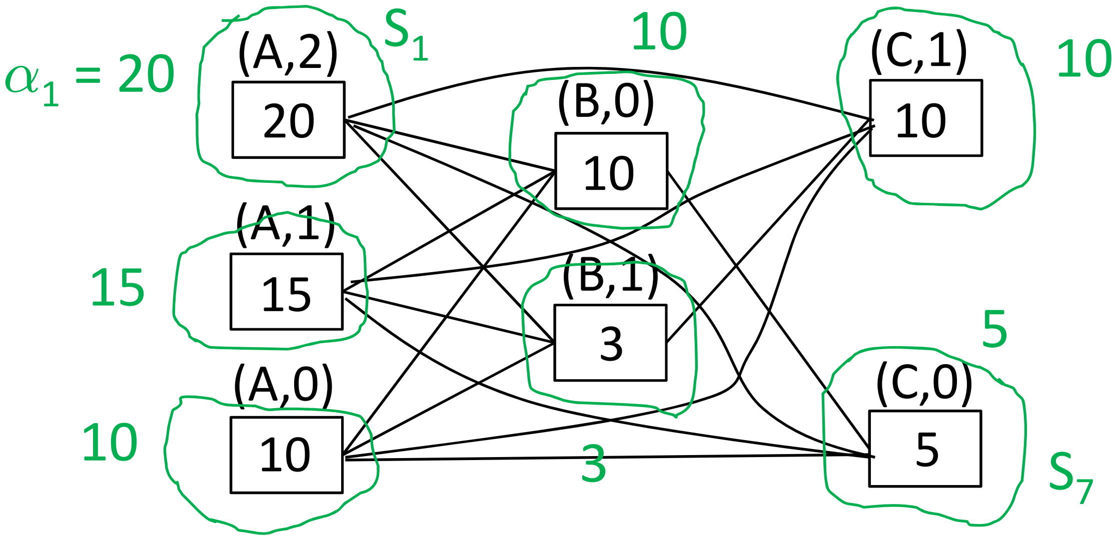
    e.g., Weighted Vertex Coloring ...
  - „Max-Plus“ Approach

    e.g., Constraint Relaxation ...

PLUS-MAX: VERTEX COLORING

Weighted Coloring Example

$$\alpha_k =_{df} \alpha(S_k)$$



Singleton coloring generates trivial upper bound for WCRT:

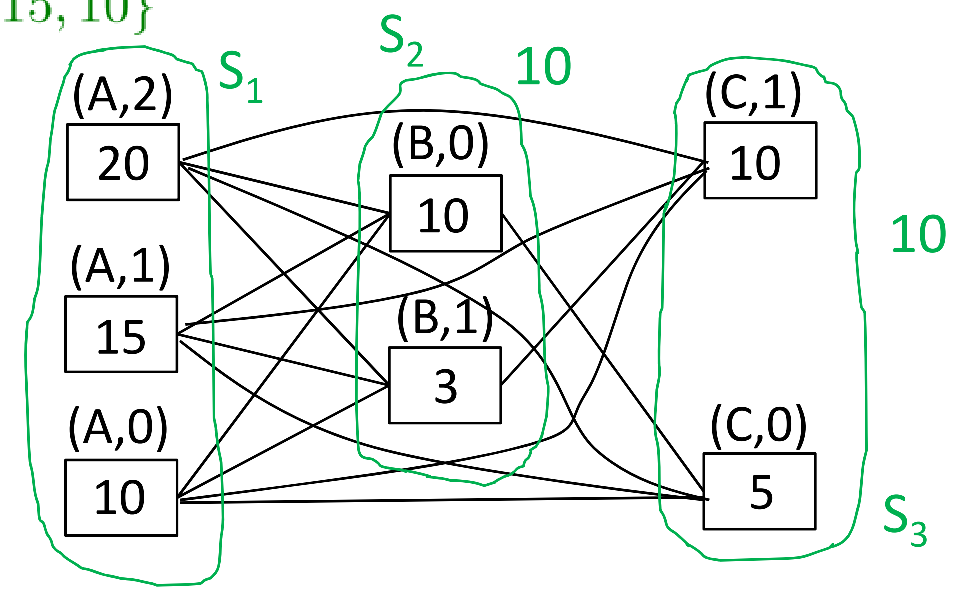
$$w(S) = \sum_{(i,j)\in V_T} w_T(i,j)$$

$$= 20 + 15 + 10 + 10 + 3 + 10 + 5 = 73$$

Weighted Coloring Example

$$\alpha_1 = max\{20, 15, 10\} = 20$$

$$\alpha_k =_{df} \alpha(S_k)$$



Thread coloring generates „Plus-Max" upper bound for WCRT:

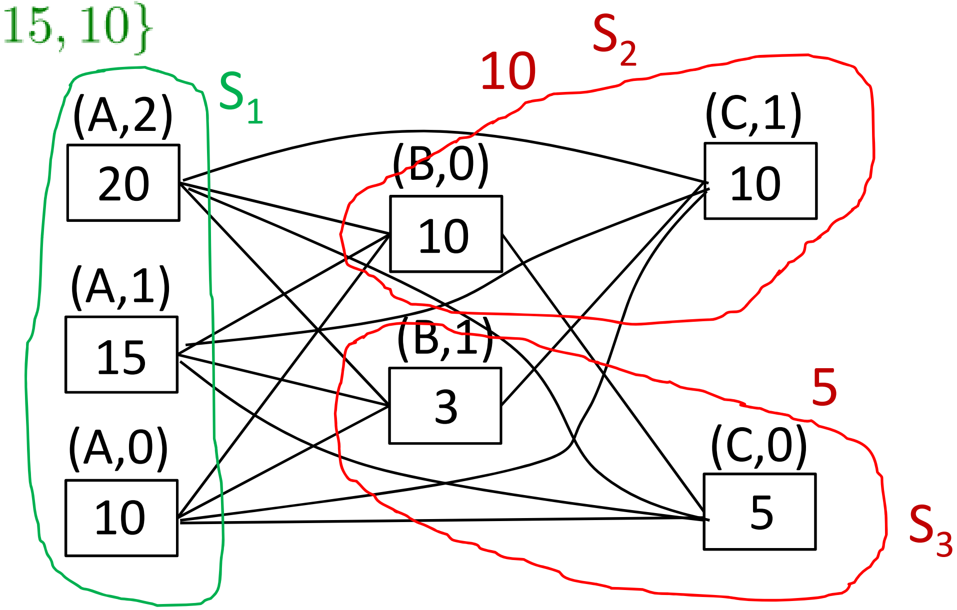
$$w(S) = \sum_i max_j \{w_T(i,j)\}$$

$$= max\{20, 15, 10\} + max\{10, 3\} + max\{10, 5\} = 40$$

Weighted Coloring Example

$$\alpha_1 = max\{20, 15, 10\} = 20$$

$$\alpha_k =_{df} \alpha(S_k)$$



General coloring generates minimal upper bound for WCRT:

$$w(S) = max\{20, 15, 10\} + max\{10, 10\} + max\{3, 5\} = 35$$

Beware: Minimal Weighted Covering is NP-complete

MAX-PLUS: CONSTRAINT RELAXATION

Relaxation for Max-Plus Upper Bounds

Relax the connectivity constraint: Maximum weight „pre-cliques“

C clique $\Rightarrow C$ pre-clique

wcrt$(T) \leq$ *max* $\{\sum_{(i,j)\in X} w_T(i,j) \mid X \subseteq V_T$ pre-clique$\}$

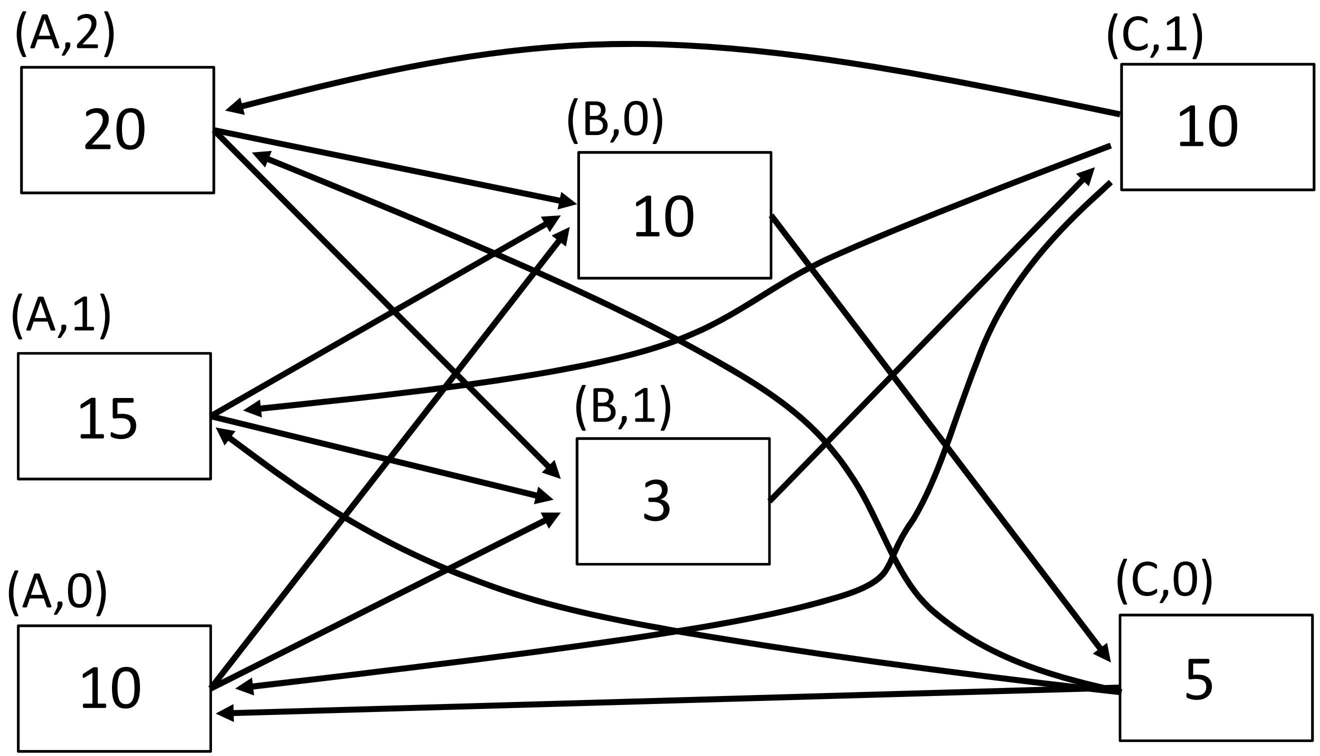
Notions of Pre-cliques

- subset: X = V_T, sum of all weights $\sum_{(i,j)\in V_T} w_T(i,j)$
- subset containing at most one node from each thread:
  same as Plus-Max $\sum_i$ *max*$_j$ $\{w_T(i,j)\}$
- triangulated subgraph
- cycle containing at most one node from each thread (Maximum Cycle Mean Problem)
- ...

Good News: These can be computed in polynomial time

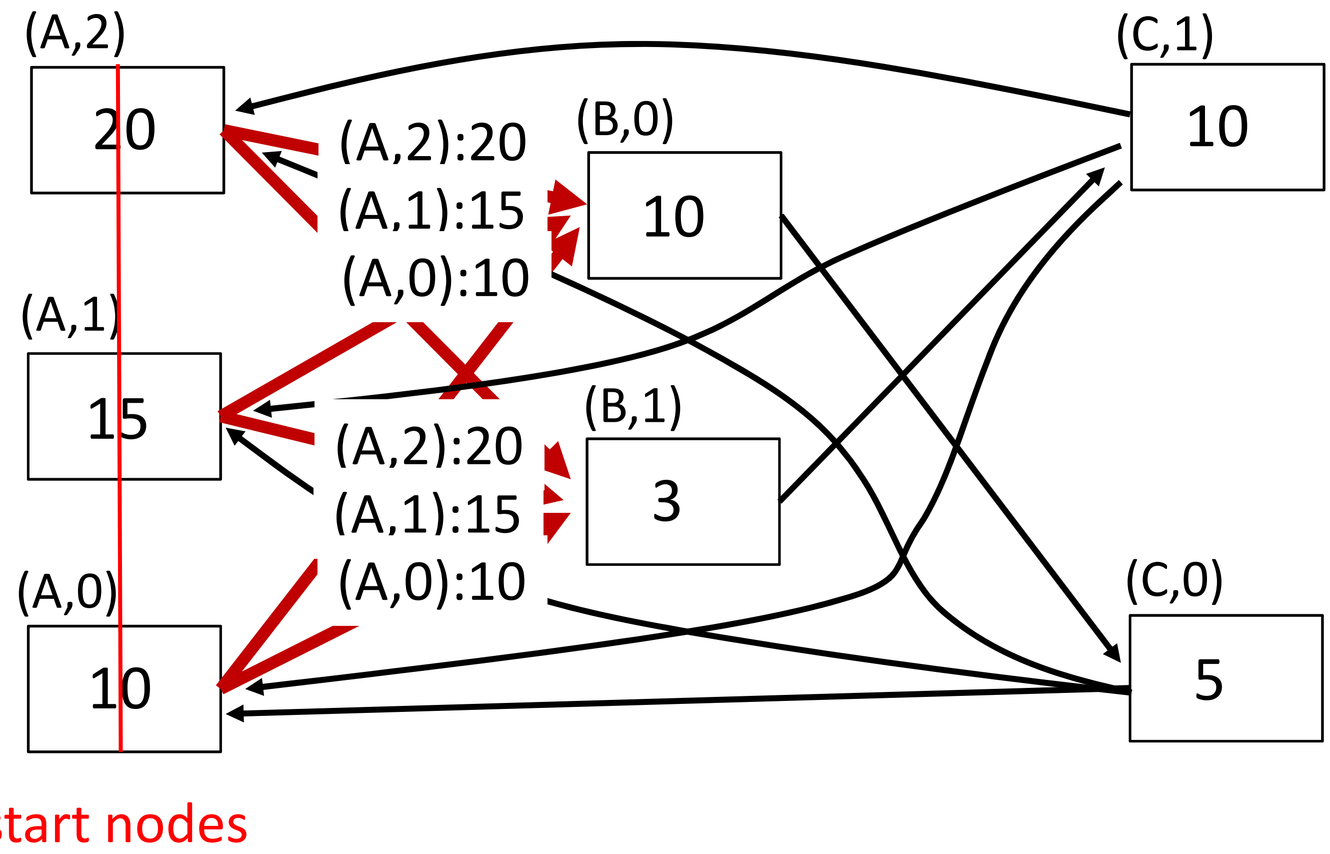
Maximum Cycle Relaxation Example

Ordered Threads, Directed edges A -> B -> C -> A



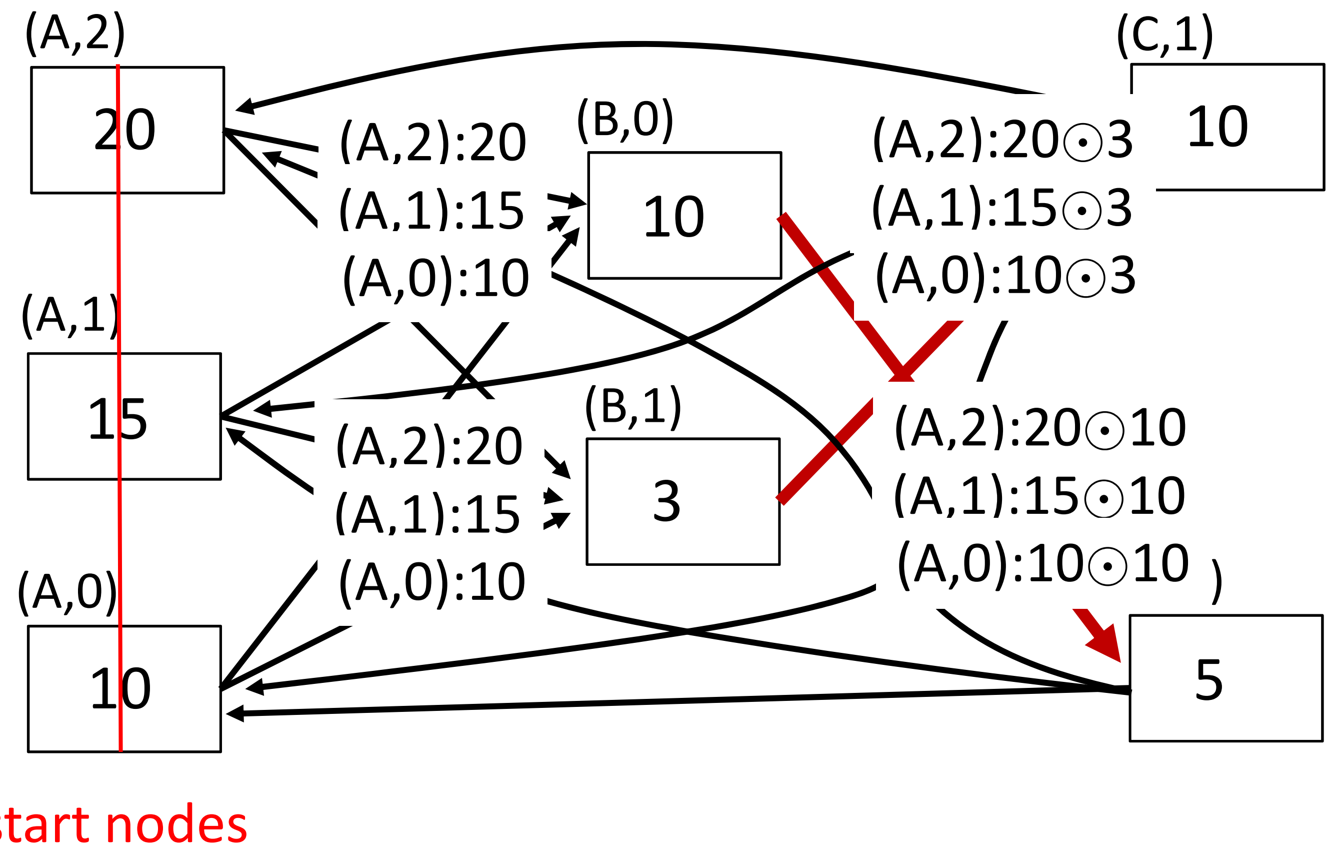
Maximum Cycle Relaxation Example

Ordered Threads, Directed edges A -> B -> C -> A



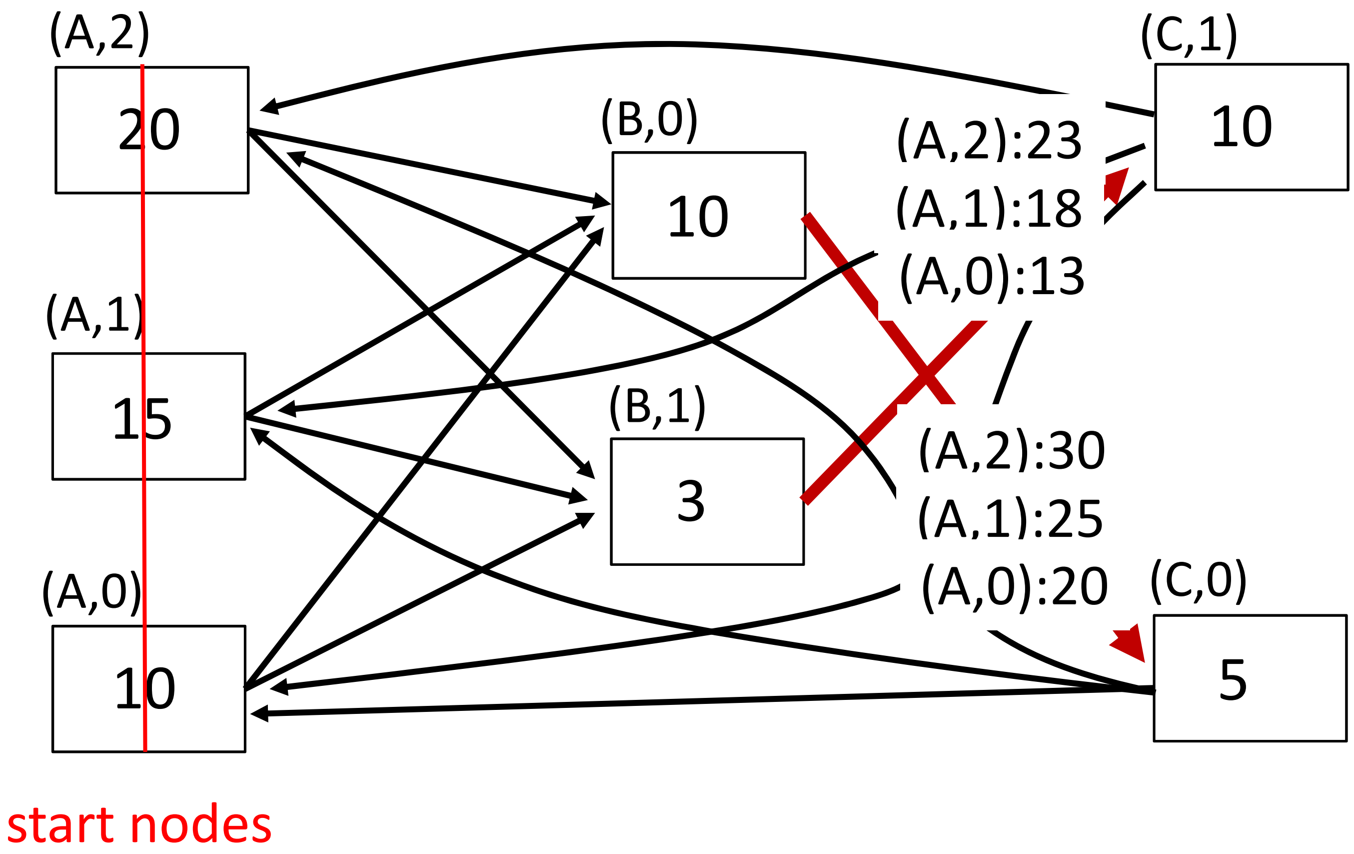
Maximum Cycle Relaxation Example

Ordered Threads, Directed edges A -> B -> C -> A



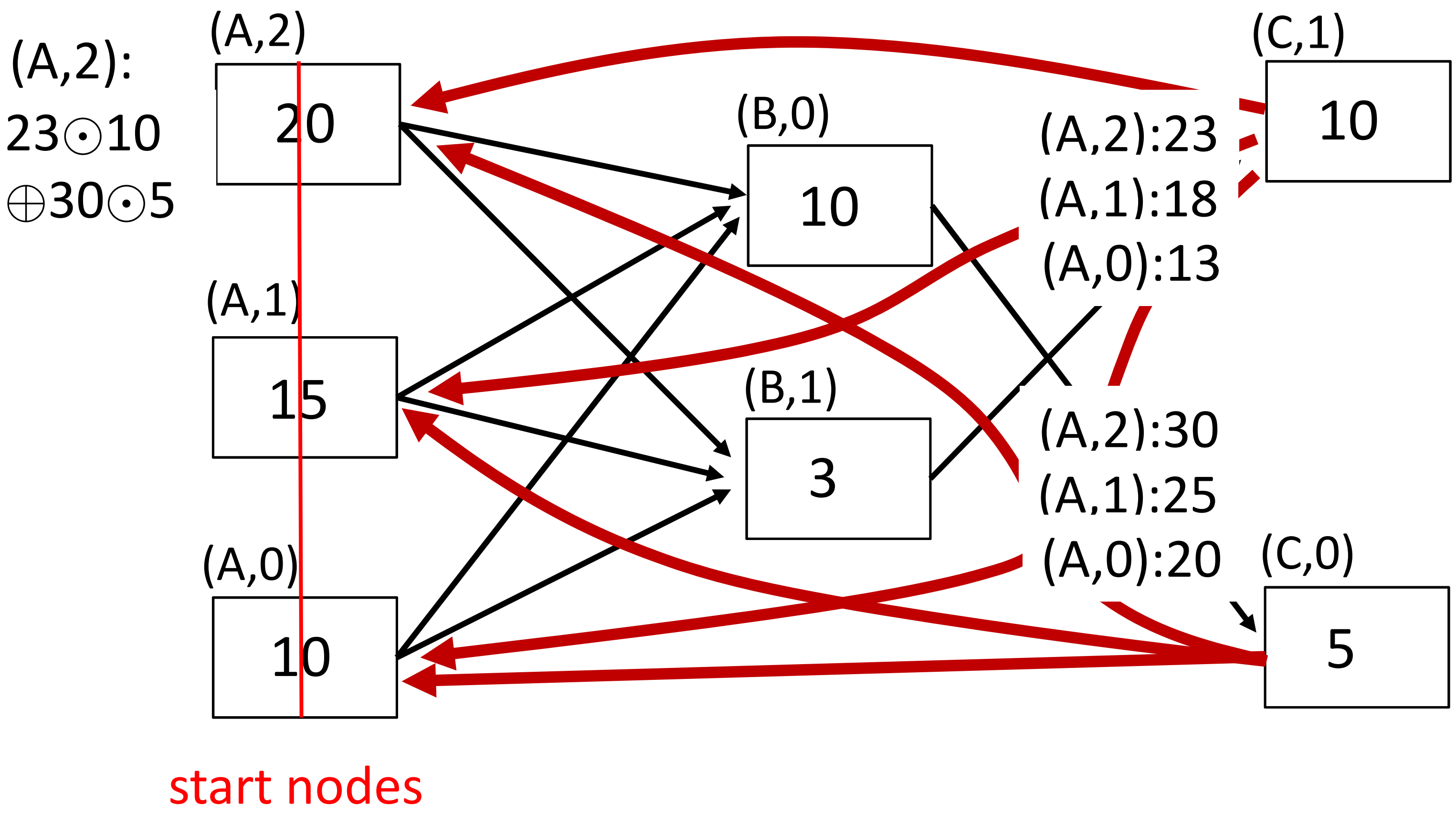
Maximum Cycle Relaxation Example

Ordered Threads, Directed edges A -> B -> C -> A



Maximum Cycle Relaxation Example

Ordered Threads, Directed edges A -> B -> C -> A



5
CONCLUSION

What‘s the Message ?

- Tick Alignment (even without data-dependency) is a non-trivial computational problem
- Conjecture
  - Unknown if TAP is NP-hard
  - Polynomial Approximations can be found which give the exact WCRT „most of the time“

    (remember Pascal Raymond‘s talk)
- Experiments still outstanding ...