

(5) Connect local layer interface (Manually proof required)		$\text{Mach}_{\text{Local}}(C, L[cid, \varepsilon_{cid}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
Multicore Linking Framework	(4) Optimize environmental context	$\sqcup$
		$\text{Mach}_{\text{sep}}(C, L[cid, \varepsilon_{sep}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{reorder}}(C, L[cid, \varepsilon'_{reorder}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{reorder}}(C, L[cid, \varepsilon_{reorder}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{split}}(C, L[cid, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{si\_big'}}(C, L[cid, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{si\_big}}(C, L[cid, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{si}}(C, L[cid, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
	(2) Introduce per-CPU machine	$\sqcup$
	(2, 3) Introduce partial machine and prove linking theorem	$\text{Mach}_{\text{env}}(C, L[cid, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{env}}(C, \parallel_{si \in \text{CoreSet}} L[\text{CoreSet}, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
		$\text{Mach}_{\text{oracle}}(C, L[\varepsilon_{\text{CoreSet}}]) \vdash \llbracket \mathbf{Prog} \rrbracket$
		$\sqcup$
	(1) Introduce hardware scheduler	$\text{Mach}_{\text{mc}}(C, L) \vdash \llbracket \mathbf{Prog} \rrbracket$
$C$ : hardware configuration		$L$ : an arbitrary layer with a certain condition

(5) Connect CompCertX Interface	$\text{Mach}_{\text{HAsm}}(C, \text{TSched}[tid, \varepsilon_{tid}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ $\sqcup$
(1, 2, 3) Introduce per-thread machine	$\text{Mach}_{\text{TAsm}}(C, \text{TLink}[tid, \varepsilon_{tid}^{MTLink}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ $\sqcup$ $\text{Mach}_{\text{IEAsm}}(C, \text{TLink}[tid, \varepsilon_{tid}^{MTLink}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ $\sqcup$
(1, 2, 3, 4) Introduce multithreaded machine and prove linking theorem	$\text{Mach}_{\text{IEAsm}}(C, \parallel tid \in T_{[cid]} \text{TLink}[cid, \varepsilon_{tid}^{MTLink}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ $\sqcup$ $\text{Mach}_{\text{EAsm}}(C, \parallel tid \in T_{[cid]} \text{TLink}[cid, \varepsilon_{cid}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ $\sqcup$ $\text{Mach}_{\text{LAsm}}(C, \text{CSched}[cid, \varepsilon_{cid}]) \vdash \llbracket \mathbf{Prog} \rrbracket$

$C$ : thread configuration

abstract relations  
between two layers



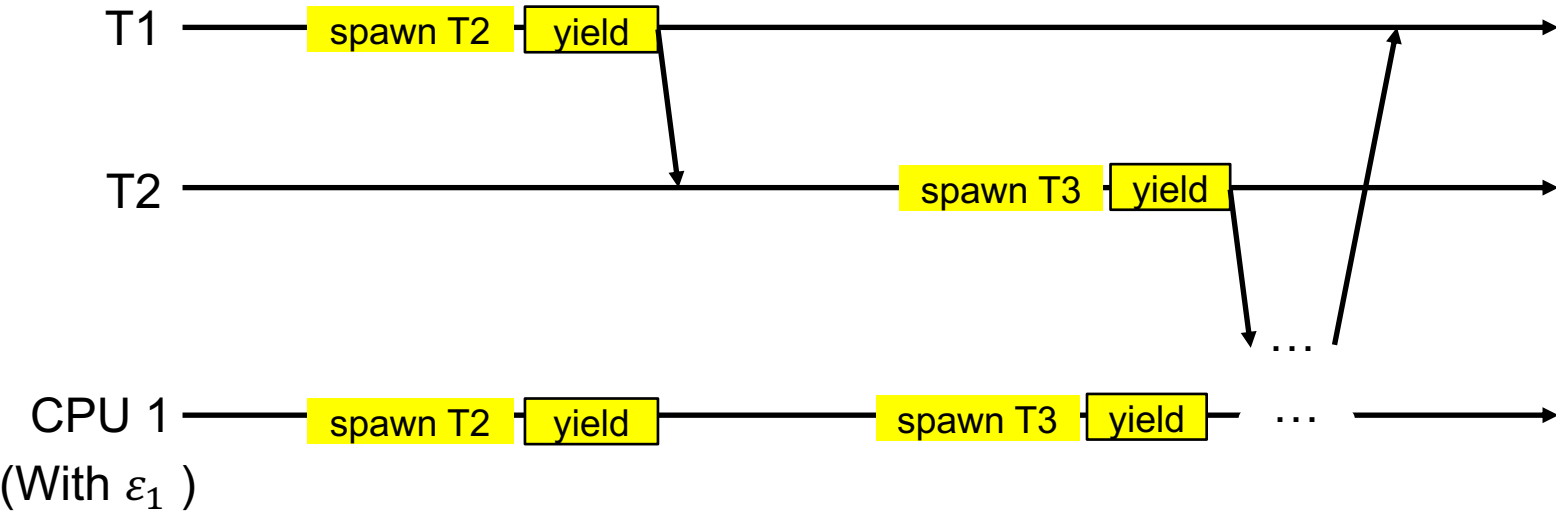
AbsRelC



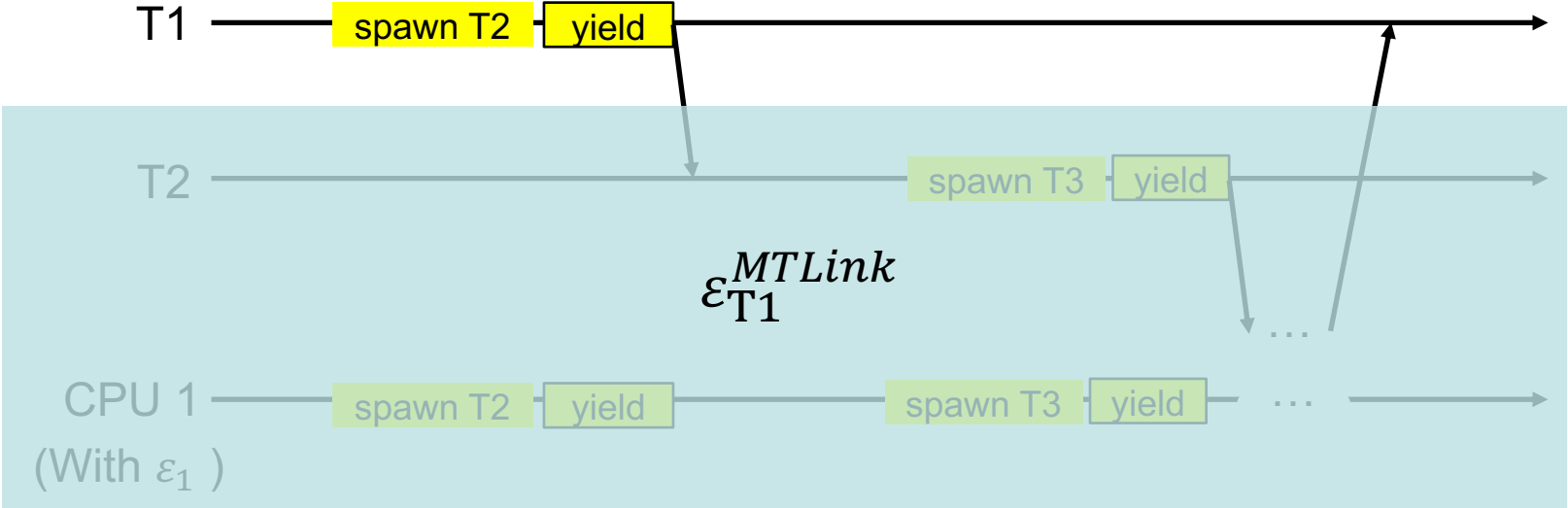
AbsRelT

- { CSched: arbitrary layer with scheduling primitives (context switching incl.)
- { TLink: arbitrary layer for intermediate machines (scheduling primitives are defined in the machine itself)
- { TSched: arbitrary layer with scheduling primitives (Scheduling has a identity behavior)

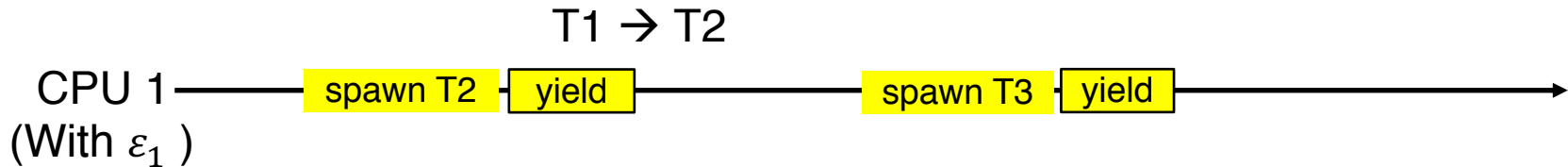
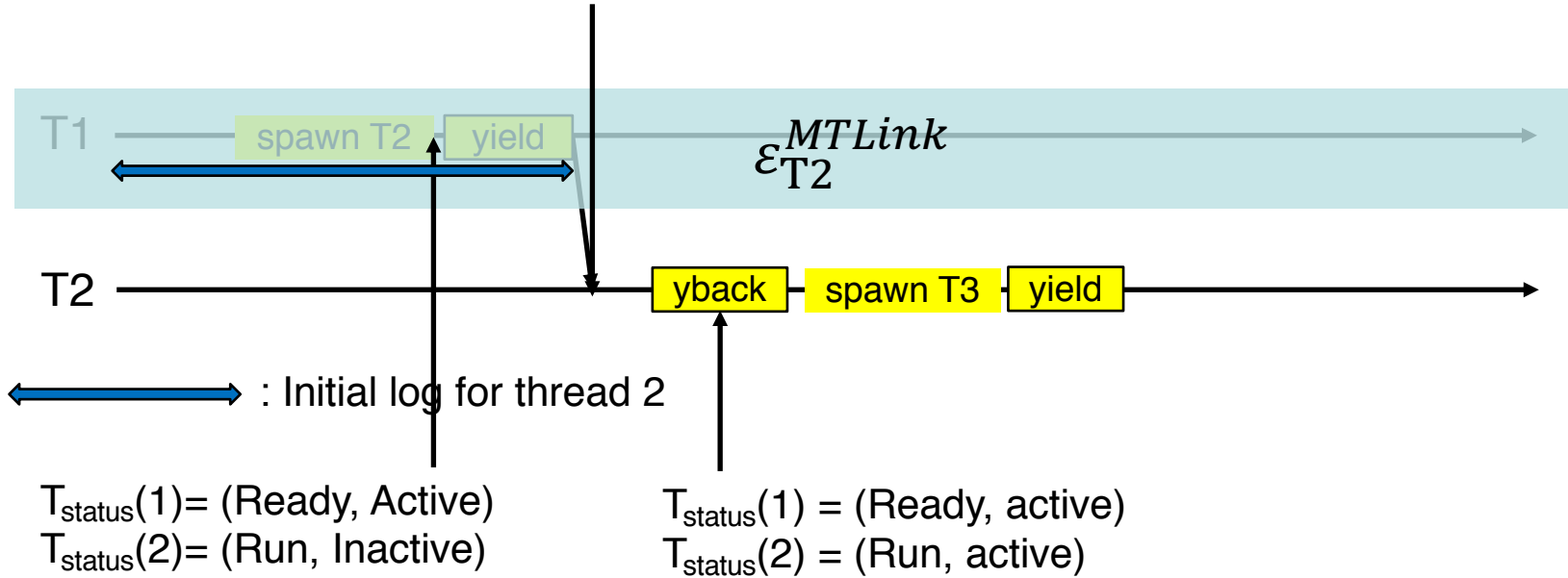
$$\text{estate}_{\text{mt}} := (tid, mem, \{ti \mapsto lst_{ti}\}, log) \ (\forall ti, ti \in T_{[cid]})$$

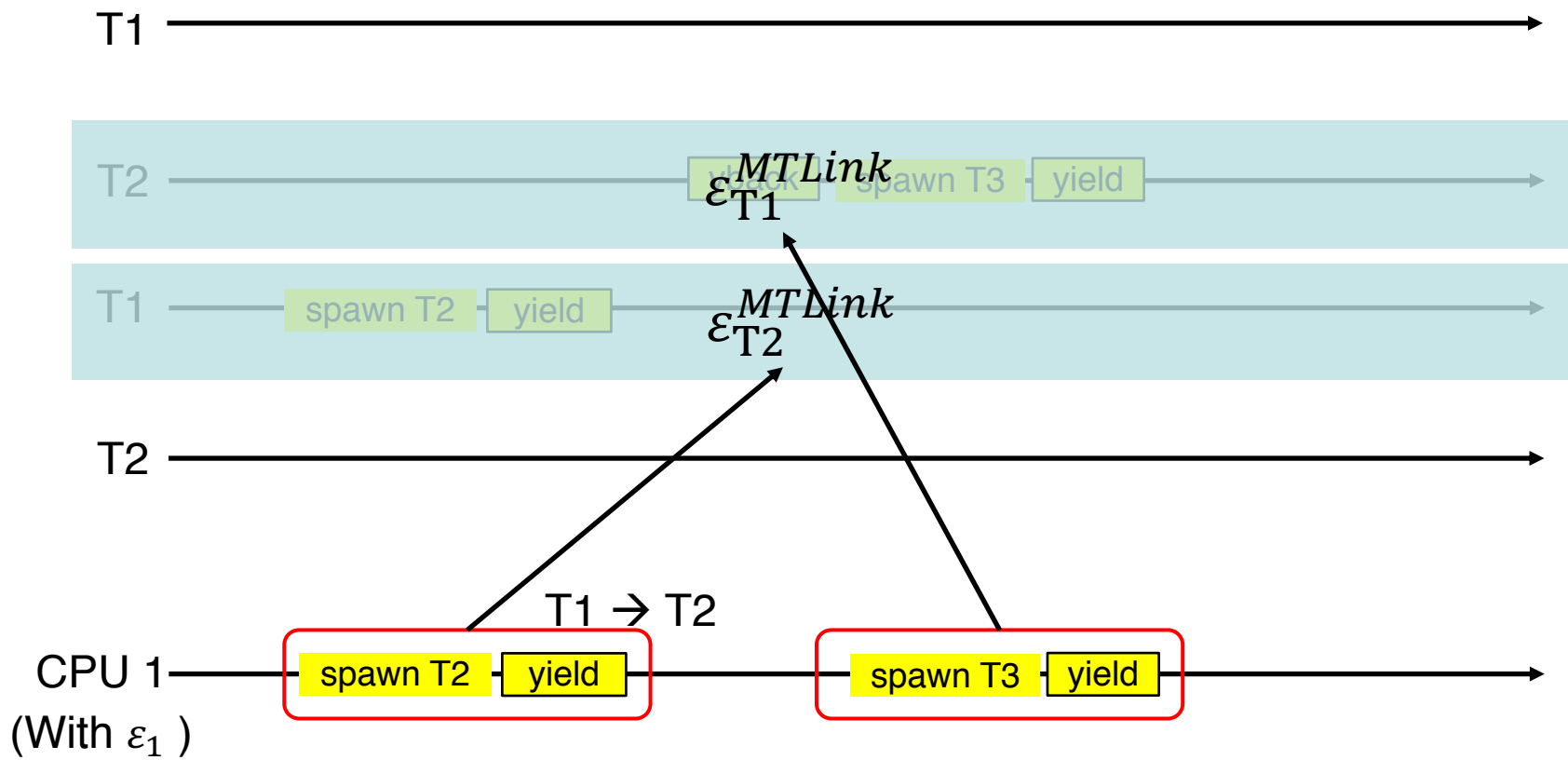


$$\text{estate}_{\text{mt}} := (tid, mem, \{ti \mapsto lst_{ti}\}, log) \ (\forall ti, ti \in T_{[cid]})$$

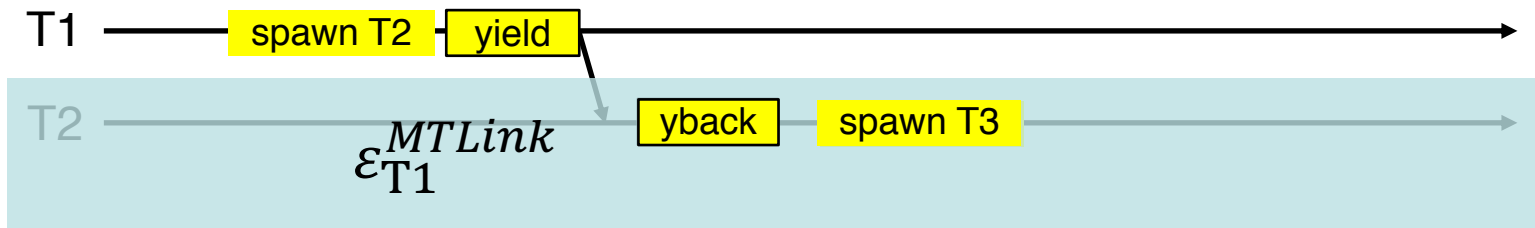


$T_{\text{status}}(1) = (\text{Run}, \text{Active})$   
 $T_{\text{status}}(2) = (\text{Ready}, \text{Inactive})$





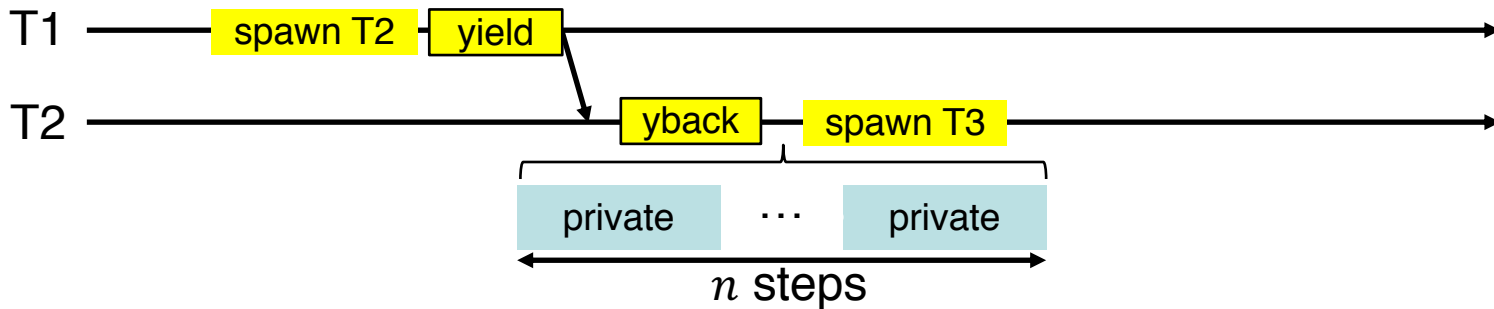
IEAsm  
with T1



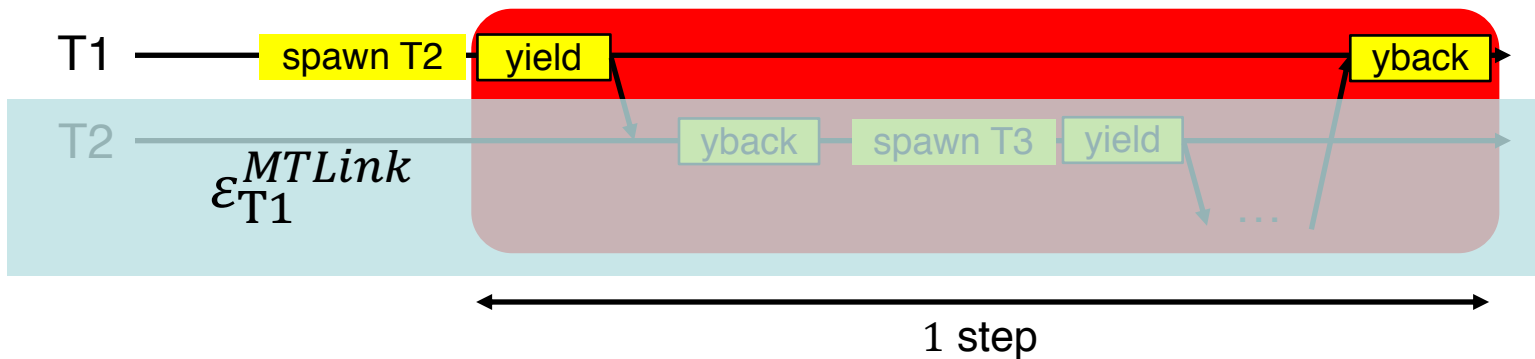
$n < progress$

Every thread will generate **at least one event** within progress steps

IEAsm  
with  $T_{[cid:=1]}$



TAsm  
with T1



$$n \leq \text{limit}$$

Every thread will be **eventually scheduled** within  $\text{limit} \times \text{progress steps}$

IEAsm  
with T1

