(5) Connect local layer interface
(Manually proof required)

Multicore Linking Framework

(4) Optimize environmental context

(2) Introduce per-CPU machine

(2, 3) Introduce partial machine and prove linking theorem

(1) Introduce hardware scheduler

$$\mathrm{Mach}_{\mathrm{Local}}(C, L[cid, \varepsilon_{cid}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{sep}}(C, L[cid, \varepsilon_{sep}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{reorder}}(C, L[cid, \varepsilon'_{reorder}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{reorder}}(C, L[cid, \varepsilon_{reorder}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{split}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{si\_big'}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{si\_big}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{si}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{env}[cid]}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{env}[\mathrm{CoreSet}]}(C, \|_{si \in CoreSet} L[CoreSet, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{oracle}}(C, L[\varepsilon_{CoreSet}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqsubseteq$$
$$\mathrm{Mach}_{\mathrm{mc}}(C, L) \vdash [\![\mathbf{Prog}]\!]$$

C: hardware configuration

L: an arbitrary layer with a certain condition

Multithreaded Linking Framework

(5) Connect CompCertX Interface

(1, 2, 3) Introduce per-thread machine

(1, 2, 3, 4) Introduce multithreaded machine and prove linking theorem

$$\mathrm{Mach}_{\mathrm{HAsm}}(C, \mathrm{TSched}[tid, \varepsilon_{tid}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqcup|$$
$$\mathrm{Mach}_{\mathrm{TAsm}}(C, \mathrm{TLink}[tid, \varepsilon_{tid}^{MTLink}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqcup|$$
$$\mathrm{Mach}_{\mathrm{IEAsm}[tid]}(C, \mathrm{TLink}[tid, \varepsilon_{tid}^{MTLink}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqcup|$$
$$\mathrm{Mach}_{\mathrm{IEAsm}[T_{[cid]}]}[(\mathrm{C}, \| \ tid \in T_{[cid]} \mathrm{TLink}[cid, \varepsilon_{tid}^{MTLink}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqcup|$$
$$\mathrm{Mach}_{\mathrm{Easm}[T_{[cid]}]}(\mathrm{C}, \| \ tid \in T_{[cid]} \mathrm{TLink}[cid, \varepsilon_{cid}]) \vdash [\![\mathbf{Prog}]\!]$$
$$\sqcup|$$
$$\mathrm{Mach}_{\mathrm{LAsm}}(C, \mathrm{CSched}[cid, \varepsilon_{cid}]) \vdash [\![\mathbf{Prog}]\!]$$

C: thread configuration

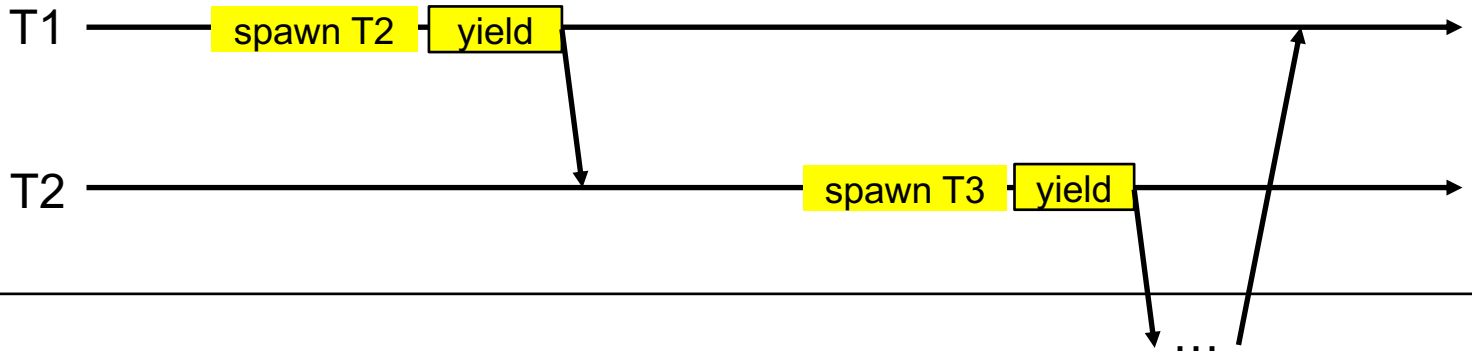
abstract relations between two layers

→ AbsRelC
- CSched: arbitrary layer with scheduling primitives (context switching incl.)

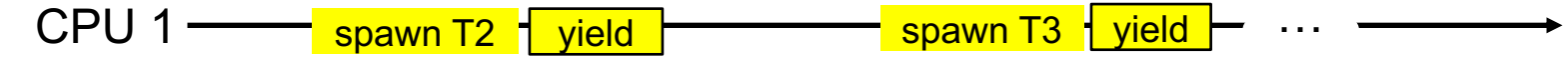
→ AbsRelT
- TLink: arbitrary layer for intermediate machines (scheduling primitives are defined in the machine itself)
- TSched: arbitrary layer with scheduling primitives (Scheduling has a identity behavior)

(2)

$\text{State}_{\text{EAsm}[T_{[1]}]} \coloneqq (tid, mem, \{ti \mapsto lst_{ti}\}, log)$ (when $\forall ti, ti \in T_{[1]}$ and $lst_{ti} = (\rho_{ti}, abs_{ti}^{T})$)



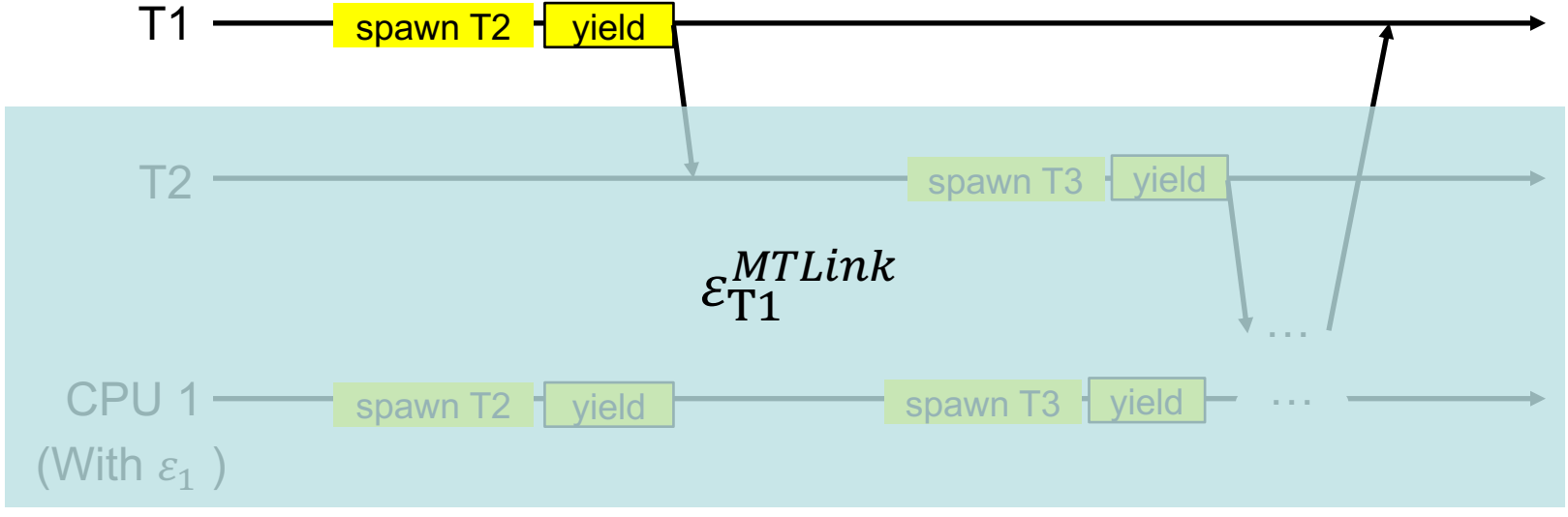
(1)



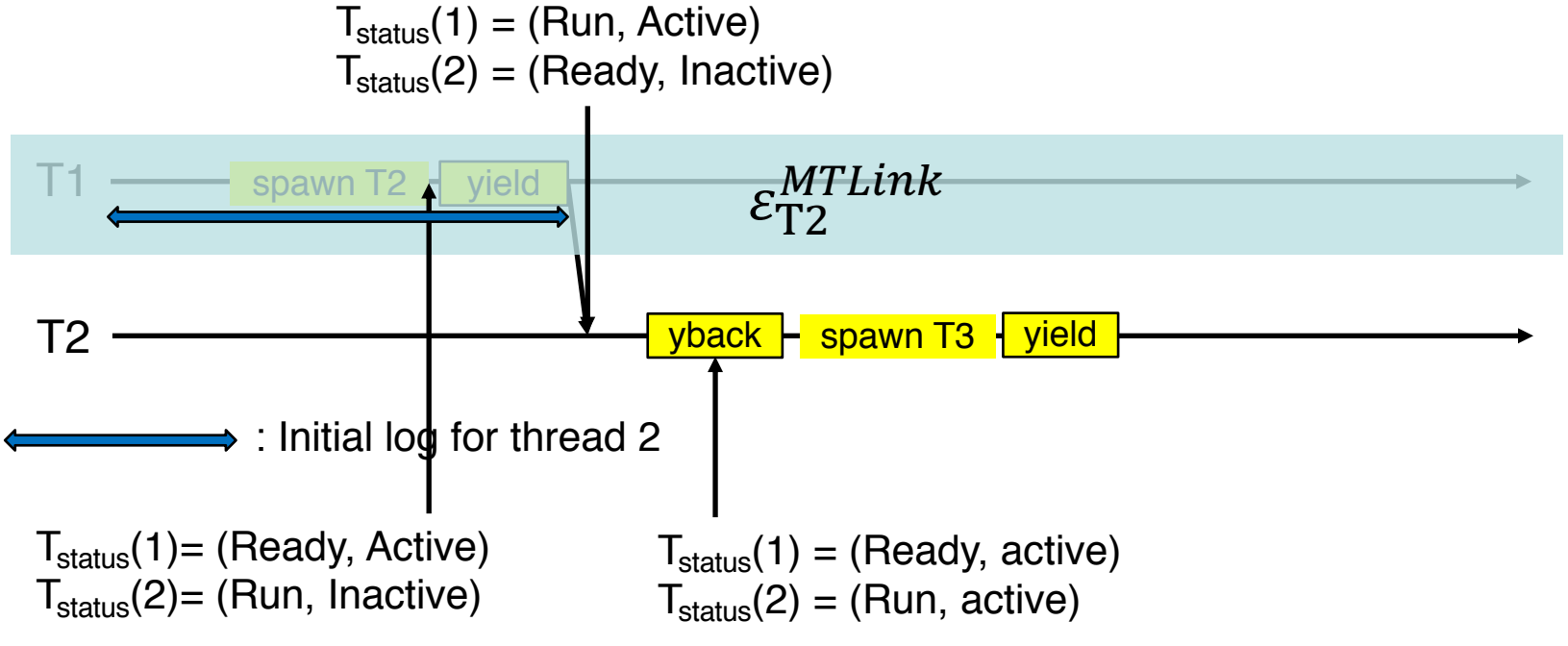
(With ε_1)

$\text{state}_{\text{LAsm}} \coloneqq (lst_{\text{cpu}}, log)$ (when $lst_{\text{cpu}} = (\rho, mem, abs)$)

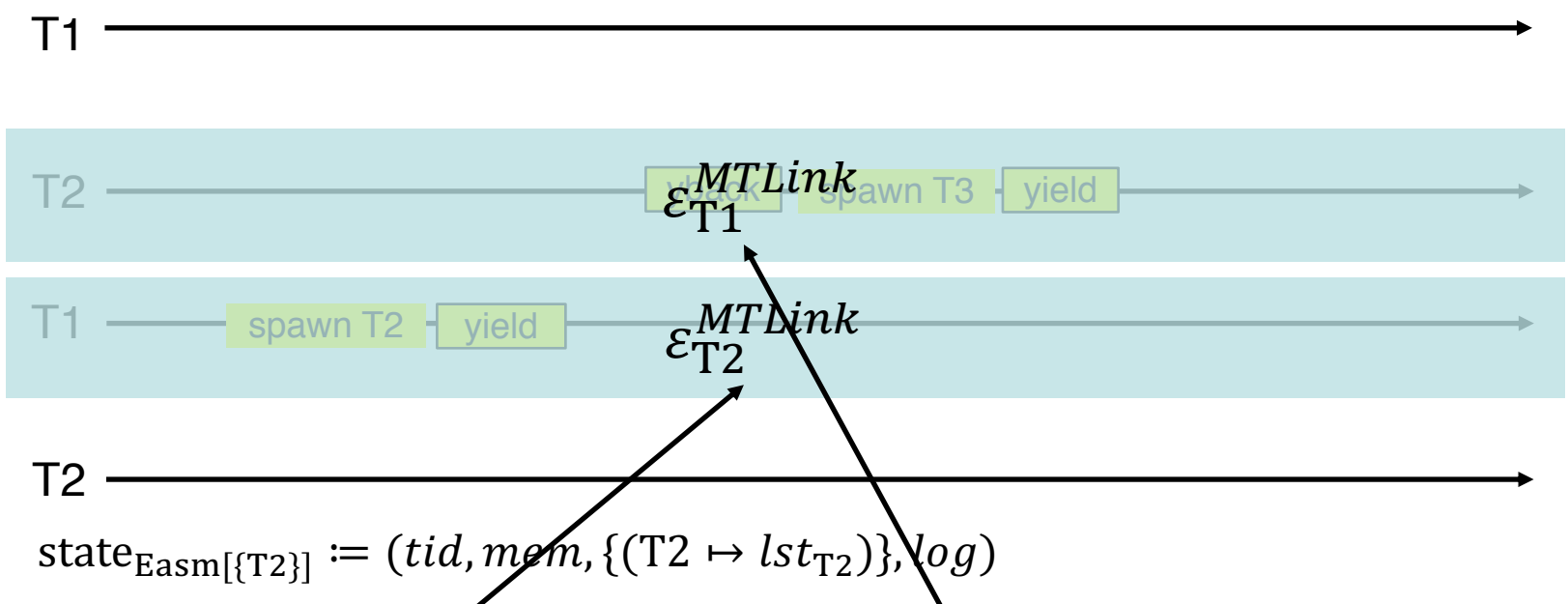
$$\text{state}_{\text{EAsm}[\{\text{T1}\}]} \coloneqq (tid, mem, \{(\text{T1} \mapsto lst_{\text{T1}})\}, log)$$



$\text{state}_{\text{EAsm}[\dots]} := (tid, mem, \{ti \mapsto lst_{ti}\}, log)$ (when $lst_{ti} = (TS_{ti}, \rho_{ti}, abs_{ti}^{T})$)

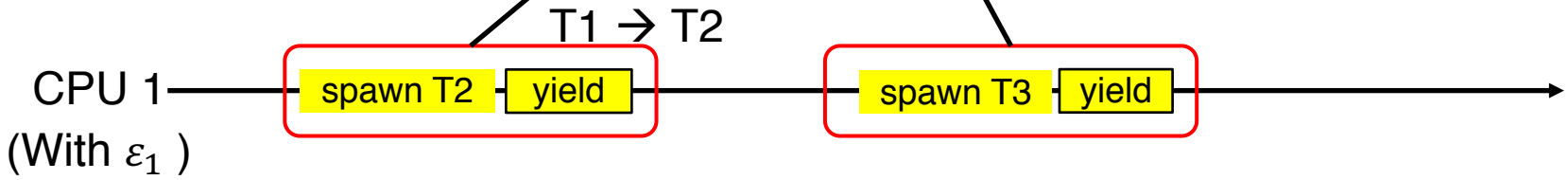


(2) $\text{state}_{\text{EAsm}[\{\text{T1}\}]} := (tid, mem, \{(\text{T1} \mapsto lst_{\text{T1}})\}, log)$



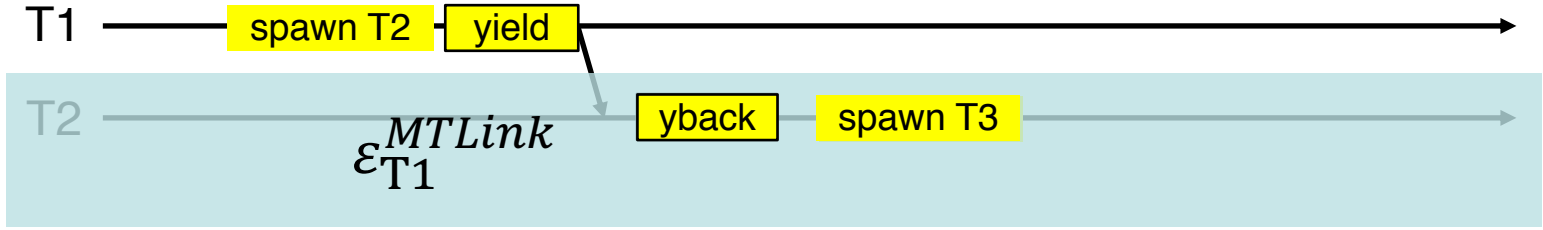
$\text{state}_{\text{Easm}[\{\text{T2}\}]} := (tid, mem, \{(\text{T2} \mapsto lst_{\text{T2}})\}, log)$

(1)



$\text{state}_{\text{LAsm}} := (lst_{\text{cpu}}, log)$

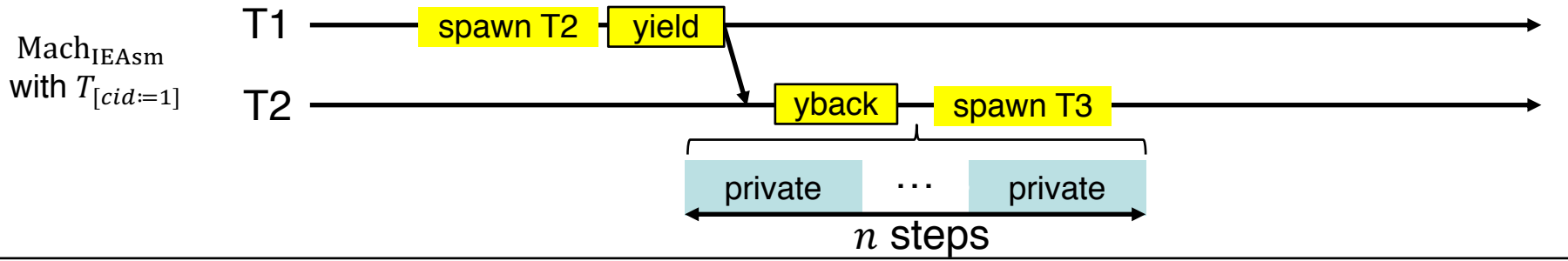
(3) $\text{state}_{\text{IEAsm}[\{T1\}]} \coloneqq (tid, (mem_{T1}, mem_{\text{others}}), \{(T1 \mapsto lst_{T1})\}, log)$



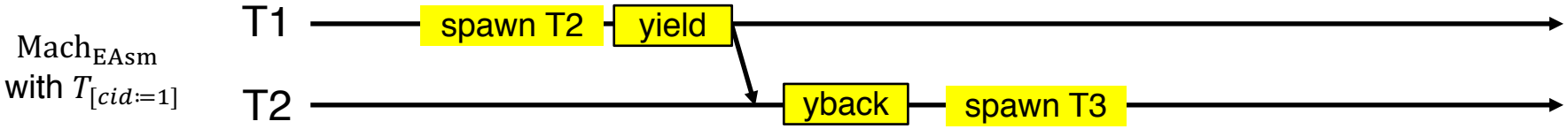
$n < progress$

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

(2)



(1)

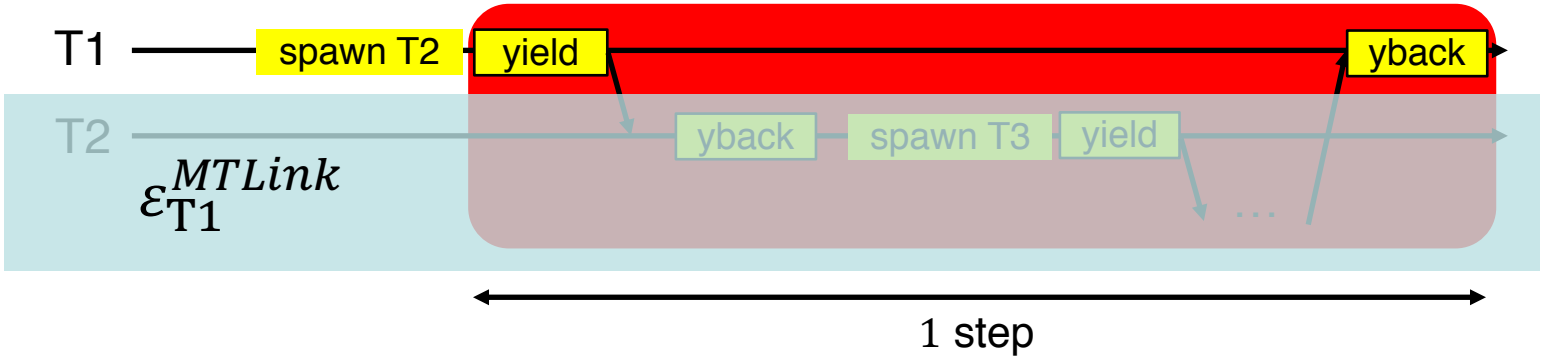


$\text{state}_{\text{EAsm}[T_{[1]}]} \coloneqq (tid, mem, \{ti \mapsto lst_{ti}\}, log)$ (when $lst_{ti} = (TS_{ti}, abs_{ti}^{T})$)

(2)

$\text{state}_{\text{TAsm}} := (tid, mem, lst, log)$

$\text{Mach}_{\text{TAsm}}$ with T1

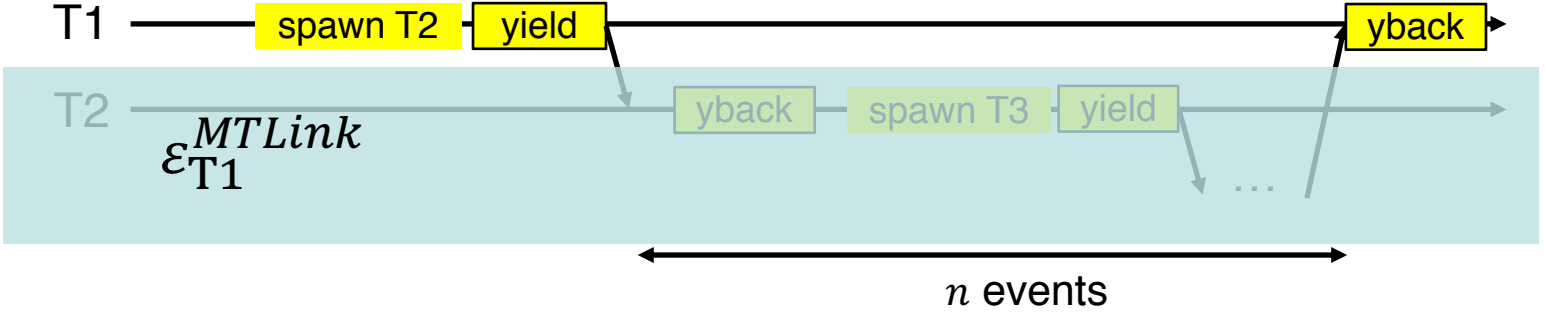


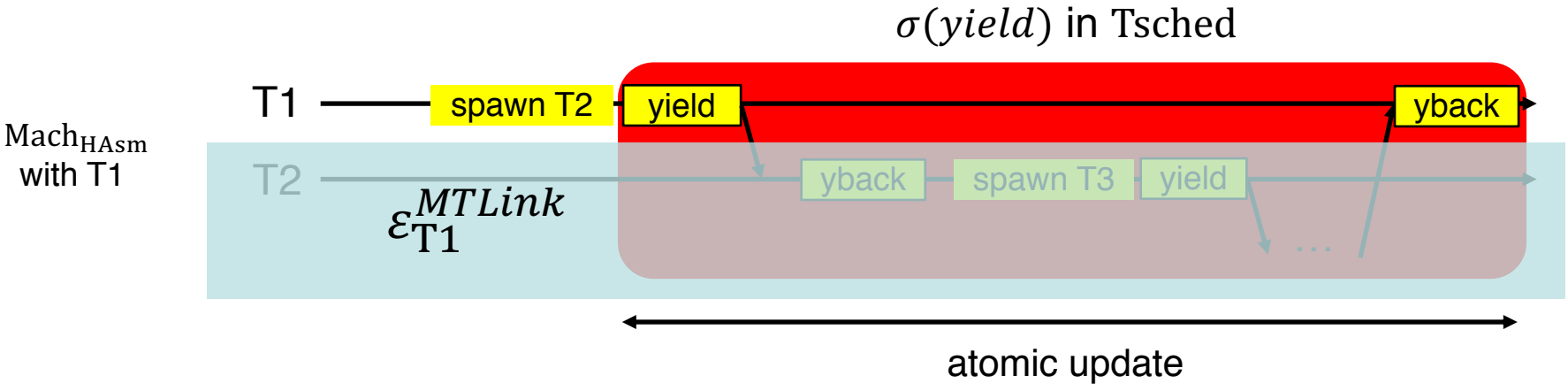
$n \leq limit$

Every thread will be eventually scheduled within $limit \times progress$ steps

(1)

$\text{Mach}_{\text{IEAsm}}$ with T1





$$\text{state}_{\text{HAsm}} \coloneqq (lst_{\text{thrd}}, log)\ (lst_{\text{thrd}} = (\rho, mem, abs))$$