(5) Connect local layer interface (manual proof required)		$\mathrm{Mach_{Local}}(C, L[cid, \varepsilon_{cid}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
(4)	Optimize environmental context	$\mathrm{Mach_{sep}}(C, L[cid, \varepsilon_{sep}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
		$\mathrm{Mach_{reorder}}(C, L[cid, \varepsilon'_{reorder}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
		$\mathrm{Mach_{reorder}}(C, L[cid, \varepsilon_{reorder}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
		$\mathrm{Mach_{split}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
		$\mathrm{Mach_{si\_big'}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
		$\mathrm{Mach_{si\_big}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
		$\mathrm{Mach_{si}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$
(2)	Introduce per-CPU machine	$\sqcup$
		$\mathrm{Mach_{env[cid]}}(C, L[cid, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$
(2, 3)	Introduce partial machine and prove linking theorem	$\sqcup$
		$\mathrm{Mach_{env[CoreSet]}}(C, \|_{si \in CoreSet} L[CoreSet, \varepsilon_{si}]) \vdash [\![\mathbf{Prog}]\!]$
		$\sqcup$
		$\mathrm{Mach_{oracle}}(C, L[\varepsilon_{CoreSet}]) \vdash [\![\mathbf{Prog}]\!]$
(1)	Introduce hardware scheduler	$\sqcup$
		$\mathrm{Mach_{mc}}(C, L) \vdash [\![\mathbf{Prog}]\!]$

Multicore-Linking Framework

C: hardware configuration L: an arbitrary layer with a certain condition

Multithreaded-Linking Framework

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

C: thread configuration

abstract relations between two layers $\longrightarrow$ AbsRelC

$\longrightarrow$ AbsRelT
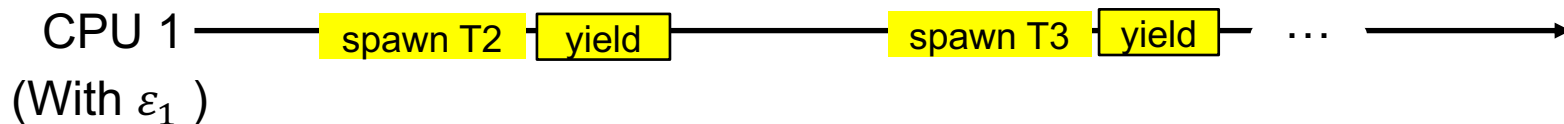
CSched: arbitrary layer with scheduling primitives (context switching primitives are included)

TLink: arbitrary layer for intermediate machines (scheduling primitives are defined in the machine itself)

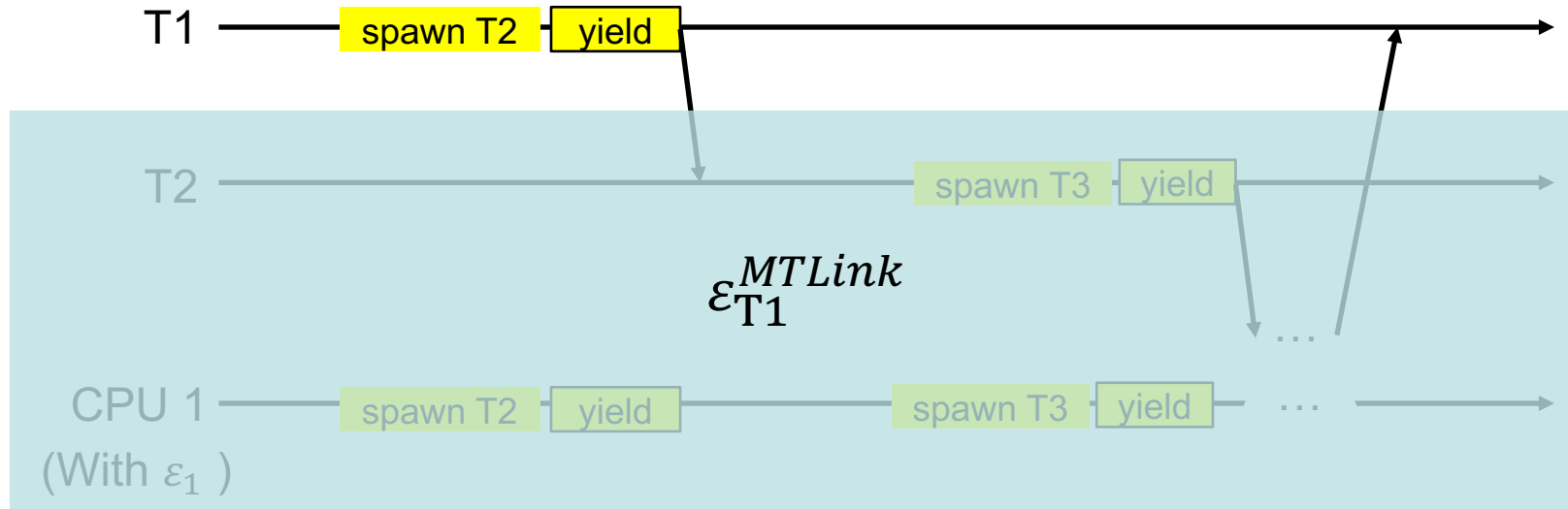TSched: arbitrary layer with scheduling primitives (scheduling has a identity behavior)

(2)

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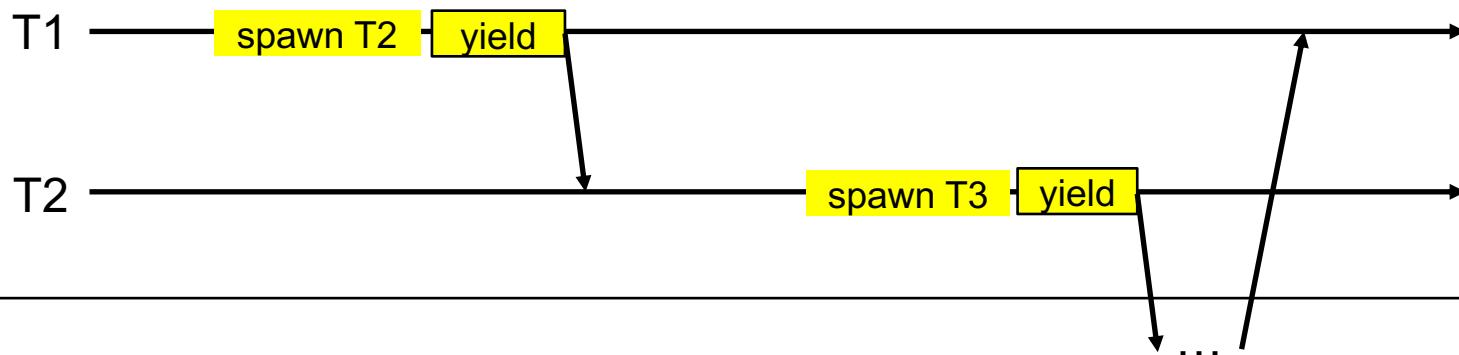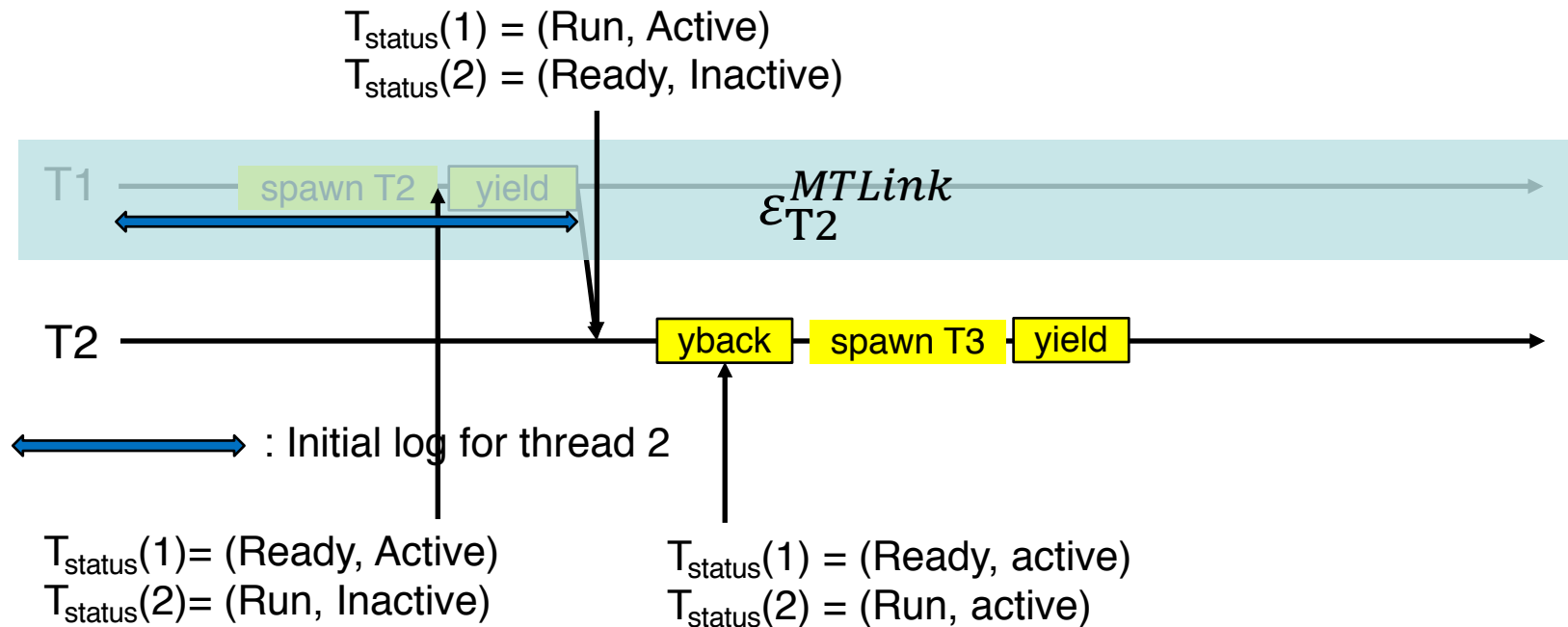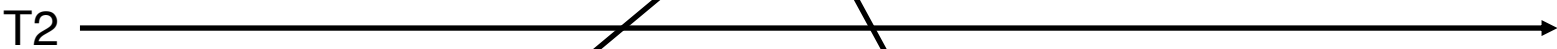

(1)

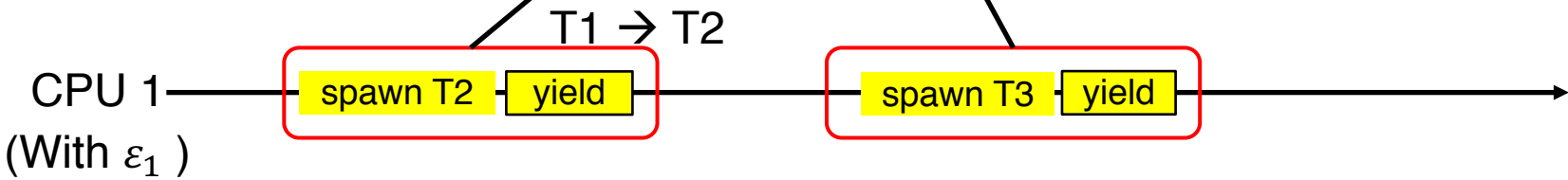$$\text{state}_{\text{LAsm}} \coloneqq (lst_{\text{cpu}}, log) \text{ (when } lst_{\text{cpu}} = (\rho, mem, abs))$$

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

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

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

T1   spawn T2   yield $\mathcal{E}_{T2}^{MTLink}$

T2   yback   spawn T3   yield

⟷ : Initial log for thread 2

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

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

(2)

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

T1

T2 $\mathcal{E}_{\text{T1}}^{MTLink}$ yield   spawn T3   yield

T1   spawn T2   yield $\mathcal{E}_{\text{T2}}^{MTLink}$

T2

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

(1)

CPU 1
(With $\mathcal{E}_1$)

T1 → T2

spawn T2   yield   spawn T3   yield

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

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

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

T1 — spawn T2 — yield

T2 — yback — spawn T3

$\mathcal{E}_{\text{T1}}^{MTLink}$
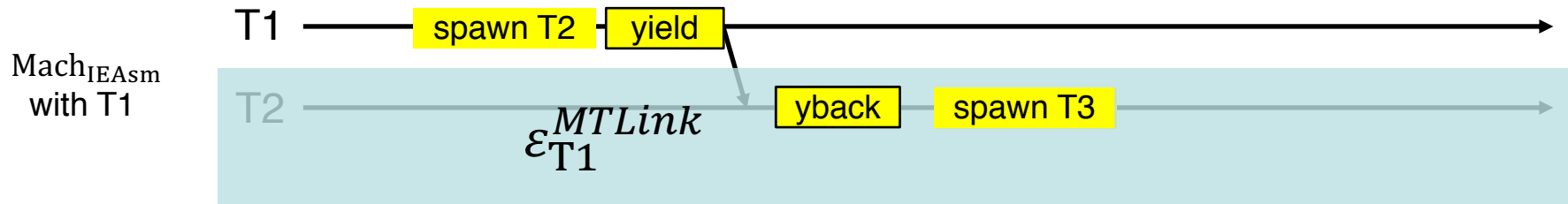
$n < progress$

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

(2) $\text{Mach}_{\text{IEAsm}}$ with $T_{[cid\coloneqq1]}$

T1 — spawn T2 — yield

T2 — yback — spawn T3

private $\cdots$ private

n steps

(1) $\text{Mach}_{\text{EAsm}}$ with $T_{[cid\coloneqq1]}$

T1 — spawn T2 — yield

T2 — yback — spawn T3
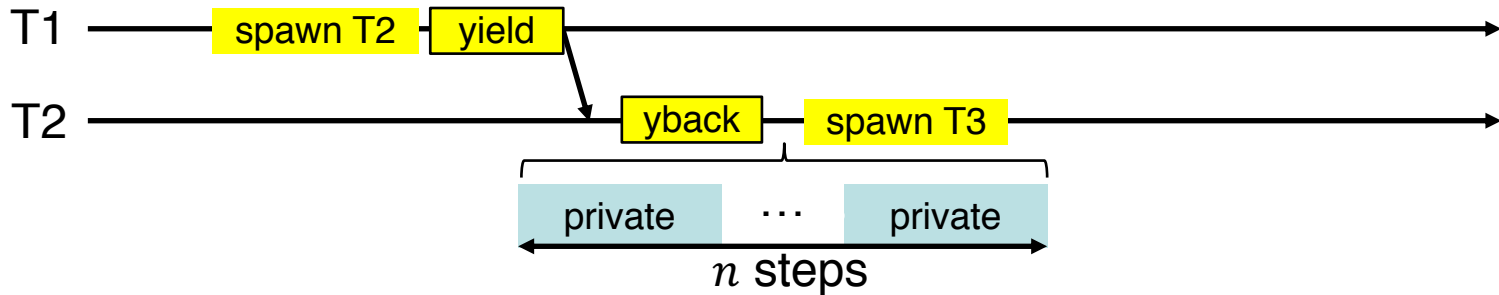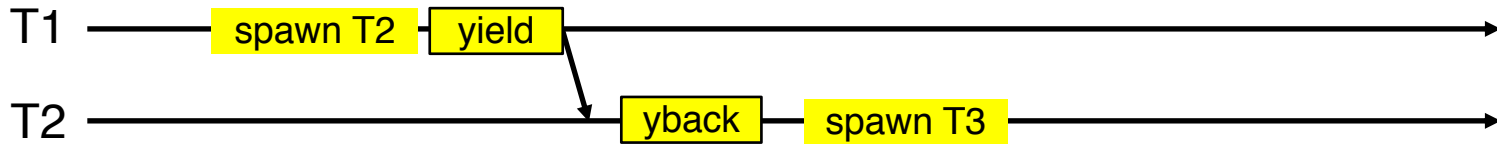
$\text{state}_{\text{EAsm}[T_{[1]}]} \coloneqq (tid, mem, \{ti \mapsto lst_{ti}\}, log)$ (when $lst_{ti} = (TS_{ti}, abs_{ti}^{T})$)

(2)

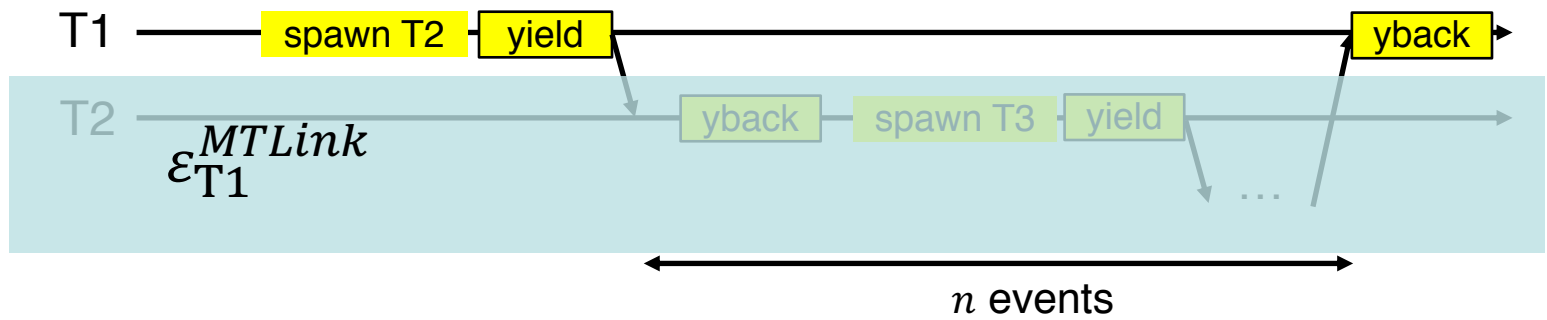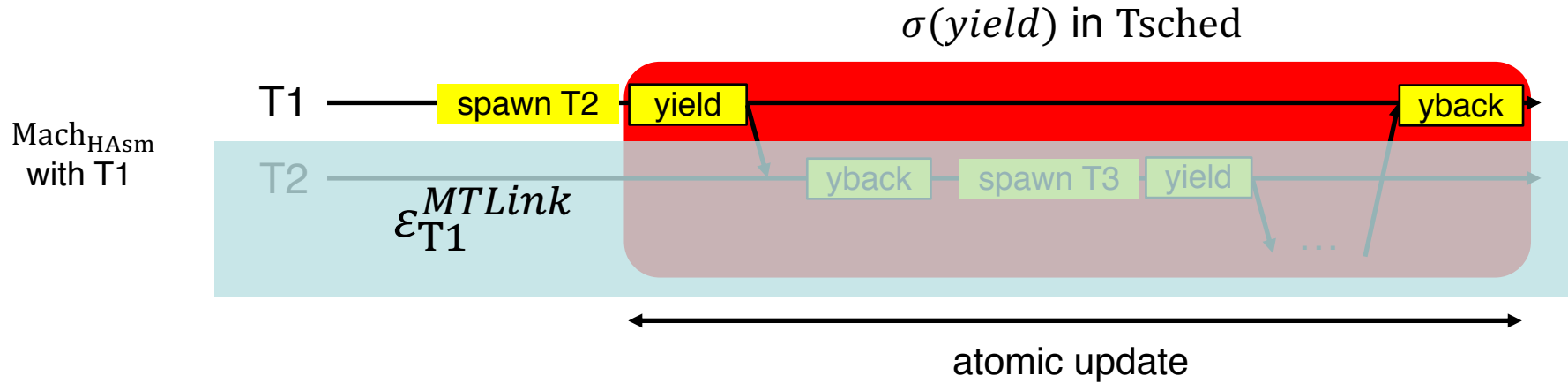$$state_{TAsm} := (tid, mem, lst, log)$$

Mach_{TAsm} with T1

T1 — spawn T2 — yield — yback

T2 — $\mathcal{E}_{T1}^{MTLink}$ — yback — spawn T3 — yield — ...

1 step
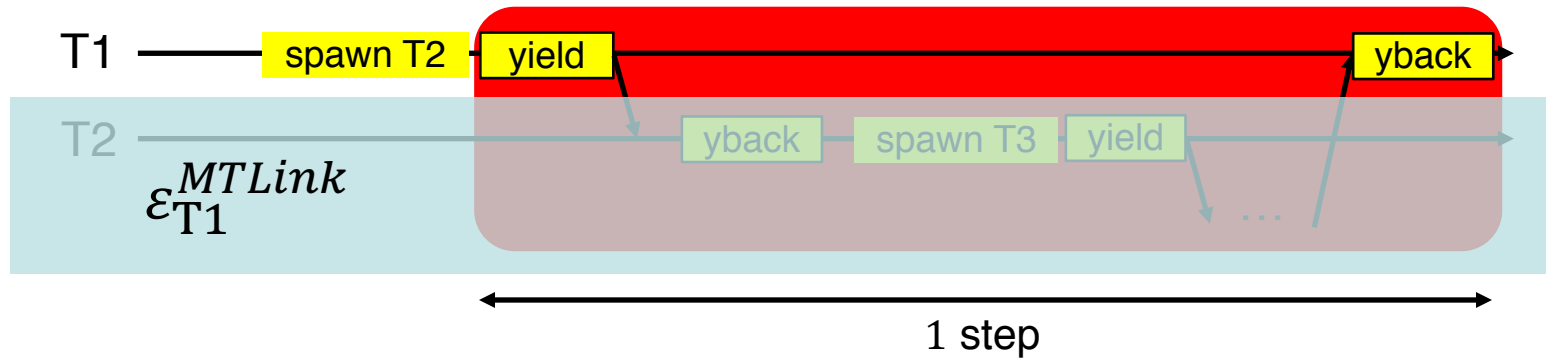
$$n \leq limit$$

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

(1)

Mach_{IEAsm} with T1

T1 — spawn T2 — yield — yback

T2 — $\mathcal{E}_{T1}^{MTLink}$ — yback — spawn T3 — yield — ...

n events

$\sigma(yield)$ in Tsched

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

T1 — spawn T2 — yield — yback

T2 — yback — spawn T3 — yield — ...

$\mathcal{E}_{\text{T1}}^{MTLink}$

atomic update

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