

| | | |
|--|--|--|
| (5) Connect local layer interface (manual proof required) | | $\text{Mach}_{\text{Local}}(C, L[cid, \varepsilon_{cid}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ \sqcup |
| Multicore Linking Framework | (4) | $\text{Mach}_{\text{sep}}(C, L[cid, \varepsilon_{sep}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ \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 | $\text{Mach}_{\text{env}[cid]}(C, L[cid, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ |
| | (2, 3) Introduce partial machine and prove linking theorem | $\text{Mach}_{\text{env}[CoreSet]}(C, \parallel_{si \in CoreSet} L[CoreSet, \varepsilon_{si}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ \sqcup |
| | (1) Introduce hardware scheduler | $\text{Mach}_{\text{oracle}}(C, L[\varepsilon_{CoreSet}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ \sqcup $\text{Mach}_{\text{mc}}(C, L) \vdash \llbracket \mathbf{Prog} \rrbracket$ |
| | 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 $\text{Mach}_{\text{IEAsm}[tid]}(C, \text{TLink}[tid, \varepsilon_{tid}^{MTLink}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ \sqcup |
| (1, 2, 3, 4) Introduce multithreaded machine and prove linking theorem | $\text{Mach}_{\text{IEAsm}[T_{[cid]}]}((C, \parallel tid \in T_{[cid]} \text{TLink}[cid, \varepsilon_{tid}^{MTLink}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ \sqcup $\text{Mach}_{\text{Easm}[T_{[cid]}]}(C, \parallel tid \in T_{[cid]} \text{TLink}[cid, \varepsilon_{cid}]) \vdash \llbracket \mathbf{Prog} \rrbracket$ \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

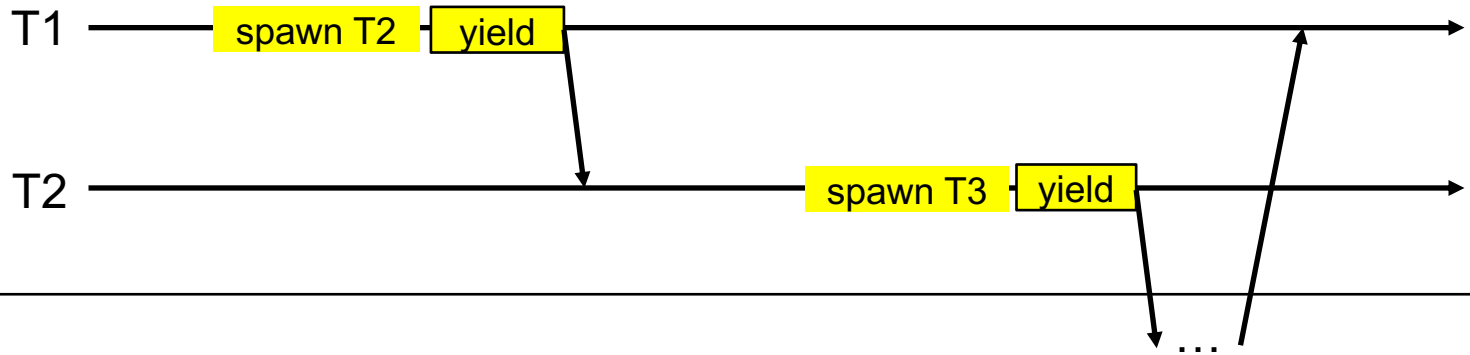


AbsRelT

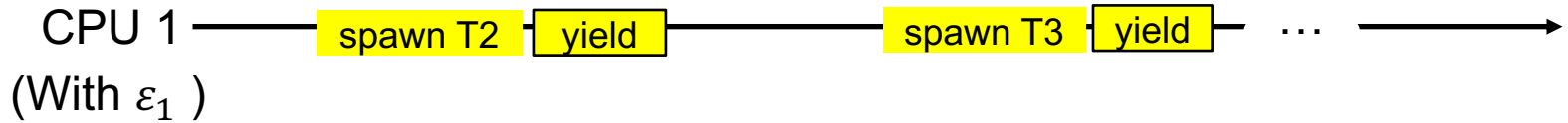
- CSched: arbitrary layer with scheduling primitives (context switching incl.)
- TLink: arbitrary layer for intermediate machines (scheduling primitives are defined in the machine itself)
- TSched: arbitrary layer with scheduling primitives (Scheduling has a identity behavior)

(2)

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

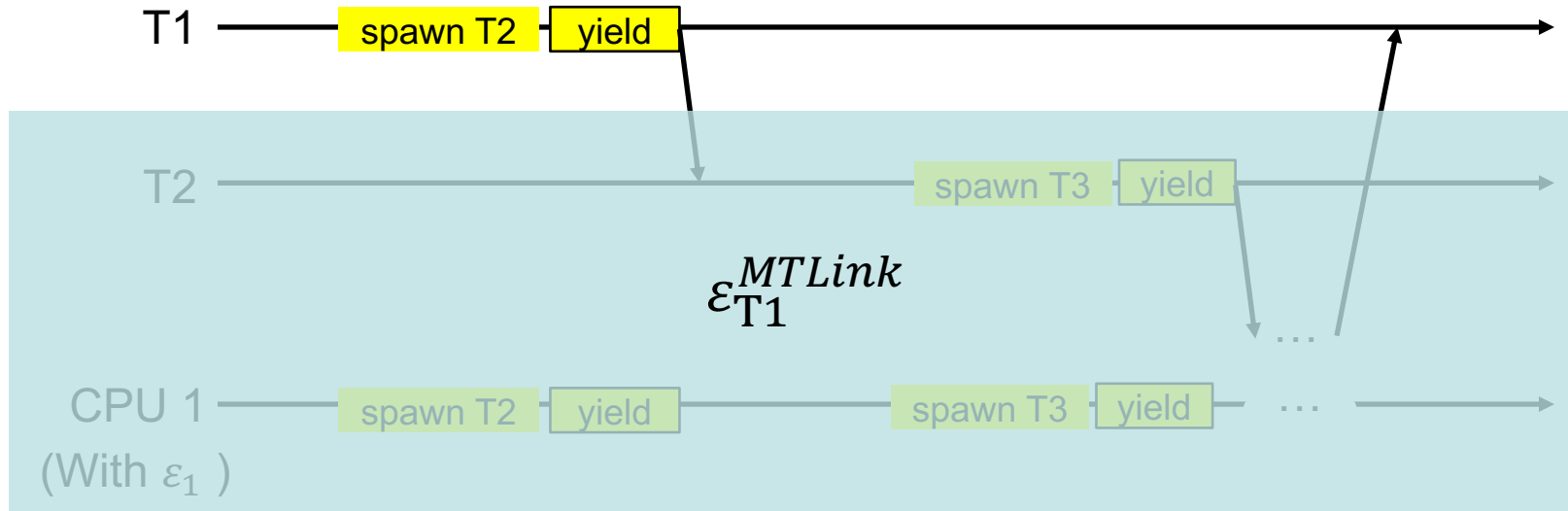


(1)

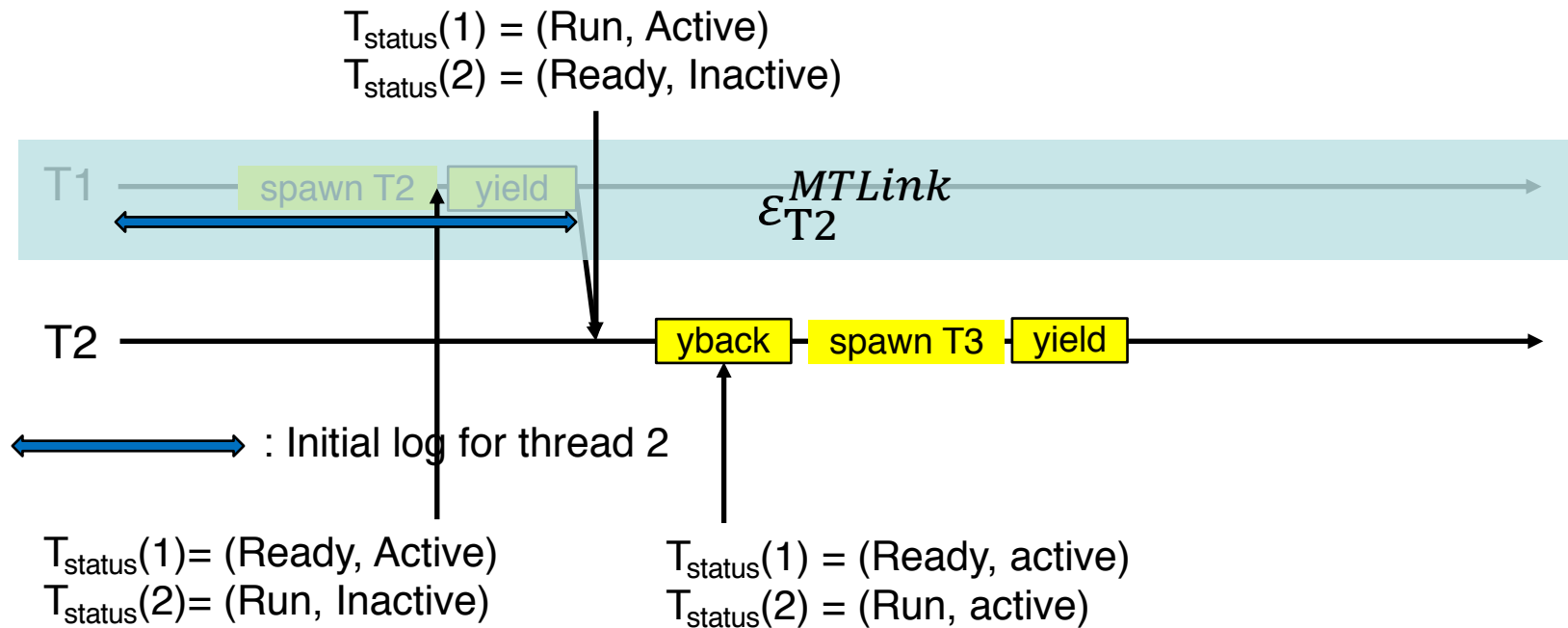


$\text{state}_{\text{LAsm}} := (lst_{\text{cpu}}, log)$ (when $lst_{\text{cpu}} = (\rho, mem, abs)$)

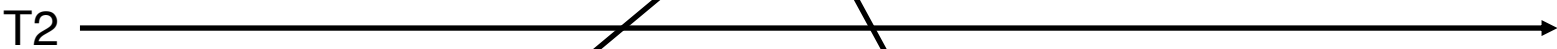
$$\text{state}_{\text{EAsm}}[\{T_1\}] := (tid, mem, \{(T_1 \mapsto lst_{T_1})\}, log)$$



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

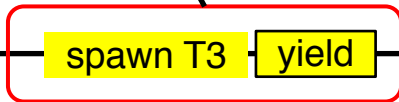
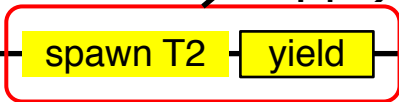


(2) $\text{state}_{\text{EAsm}}[\{T1\}] := (tid, mem, \{(T1 \mapsto lst_{T1})\}, log)$



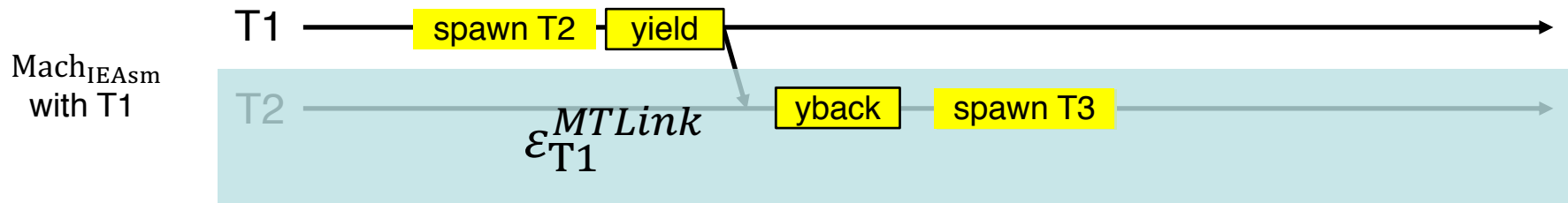
$\text{state}_{\text{Easm}}[\{T2\}] := (tid, mem, \{(T2 \mapsto lst_{T2})\}, log)$

(1) CPU 1 — (With ϵ_1)



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

(3) $\text{state}_{\text{IEAsm}}[\{T_1\}] := (tid, (\text{mem}_{T_1}, \text{mem}_{\text{others}}), \{(T_1 \mapsto \text{lst}_{T_1})\}, \log)$

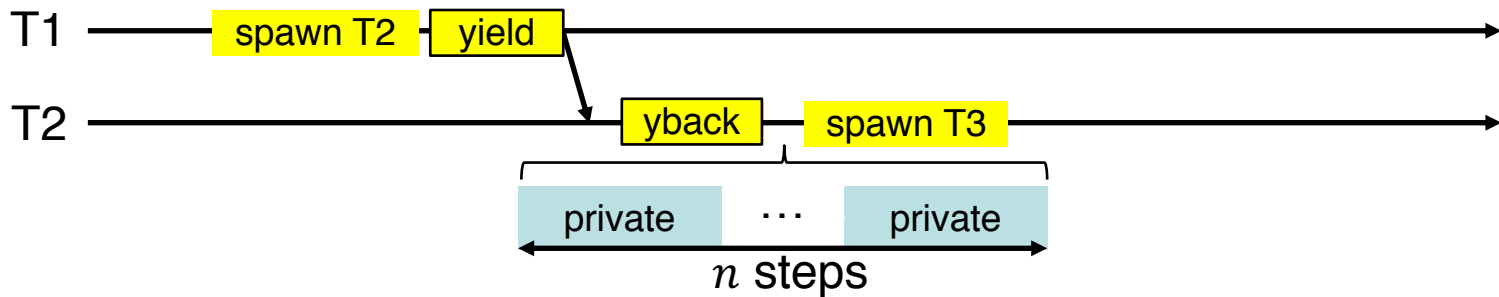


$n < \text{progress}$

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

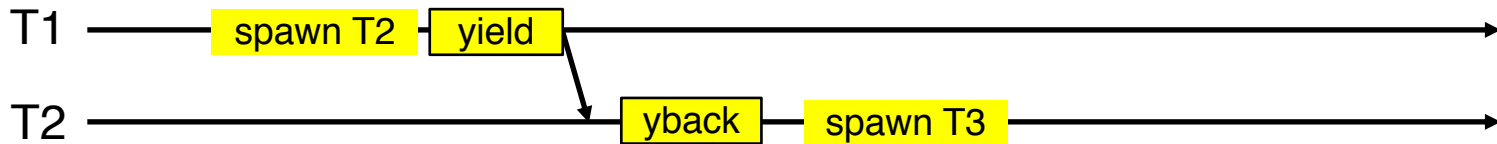
(2)

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



(1)

Mach_{EAsm} with $T_{[cid:=1]}$

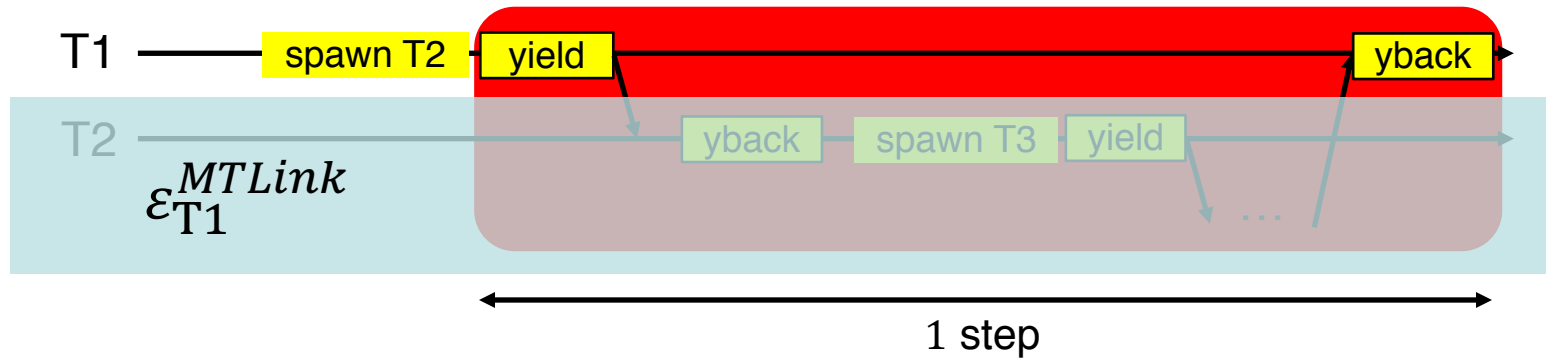


$\text{state}_{\text{EAsm}}[T_{[1]}] := (tid, \text{mem}, \{ti \mapsto \text{lst}_{ti}\}, \log)$ (when $\text{lst}_{ti} = (TS_{ti}, \text{abs}_{ti}^T)$)

(2)

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

$Mach_{TAsm}$
with T1

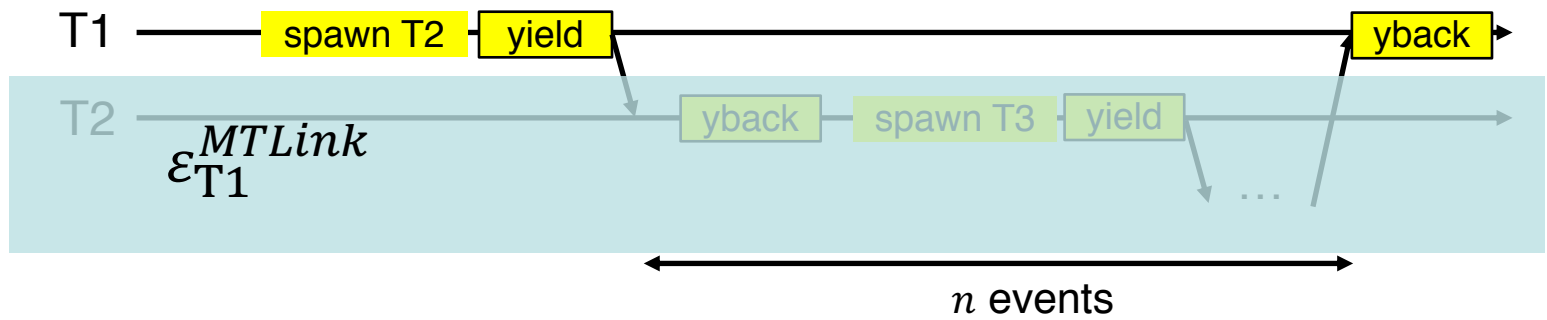


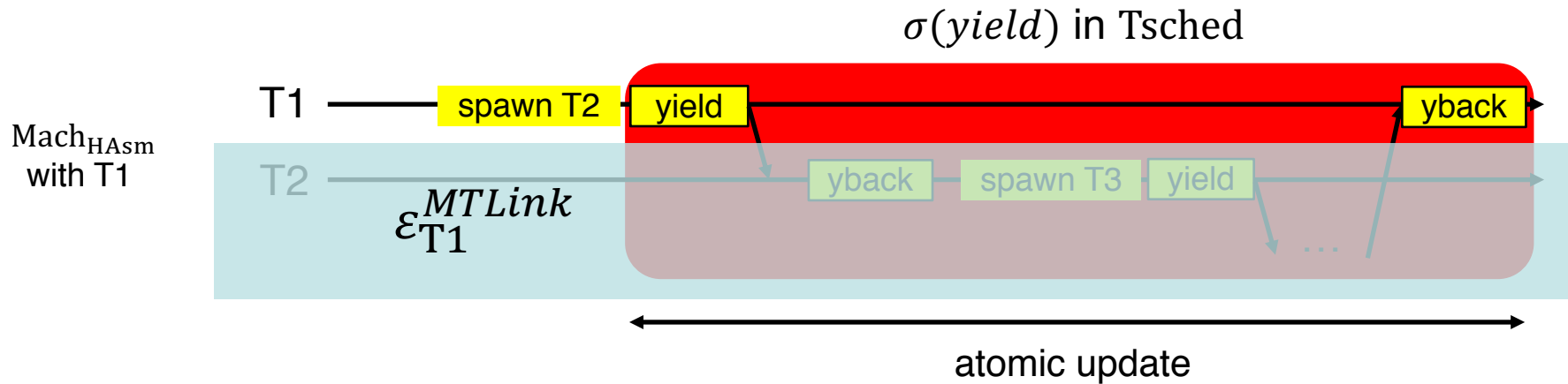
$n \leq limit$

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

(1)

$Mach_{IEAsm}$
with T1





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