(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

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

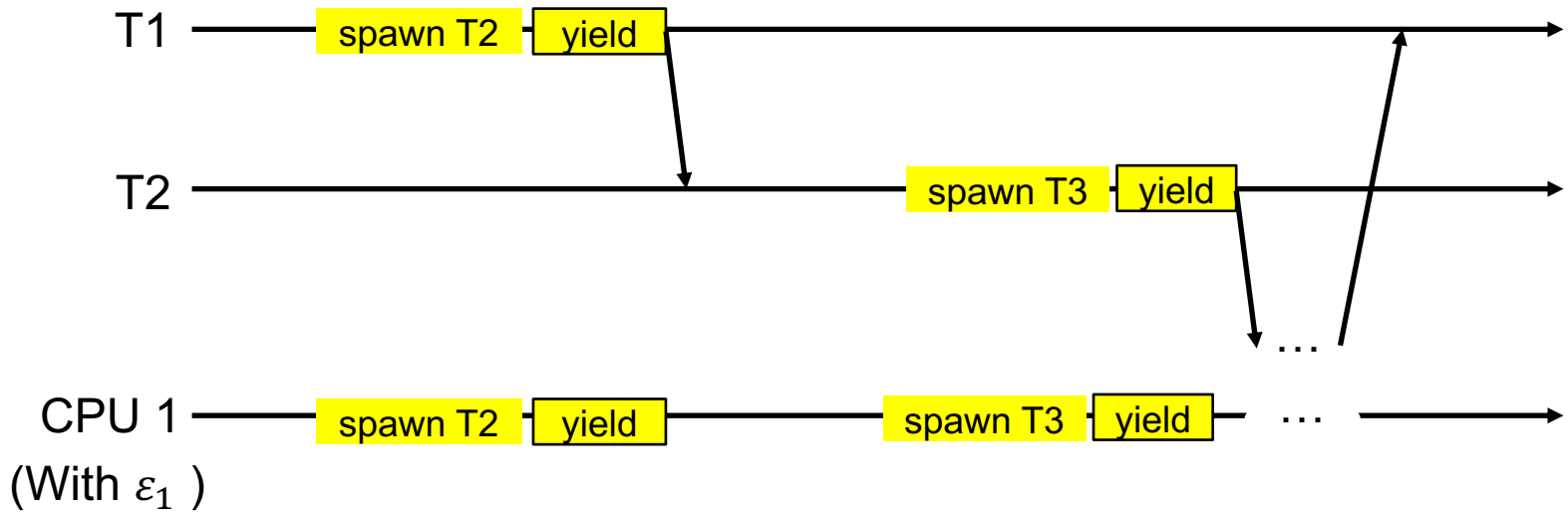
C: thread configuration

abstract relations between two layers

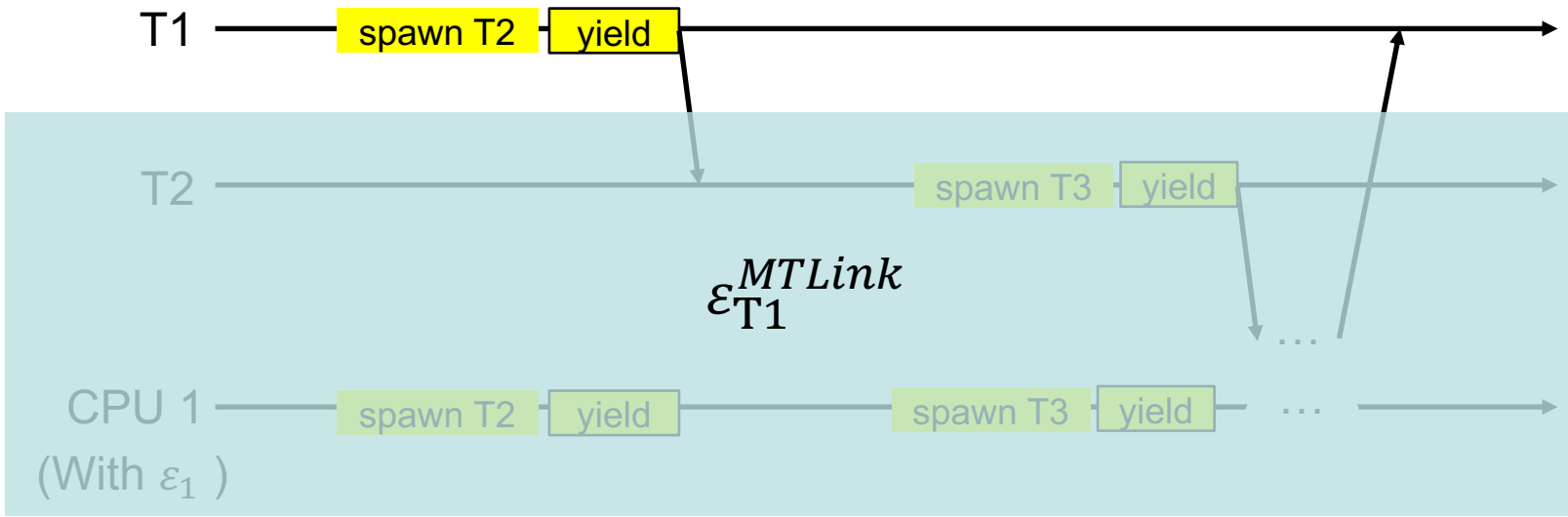
→ AbsRelC
- CSched: arbitrary layer with scheduling primitives (context switching incl.)

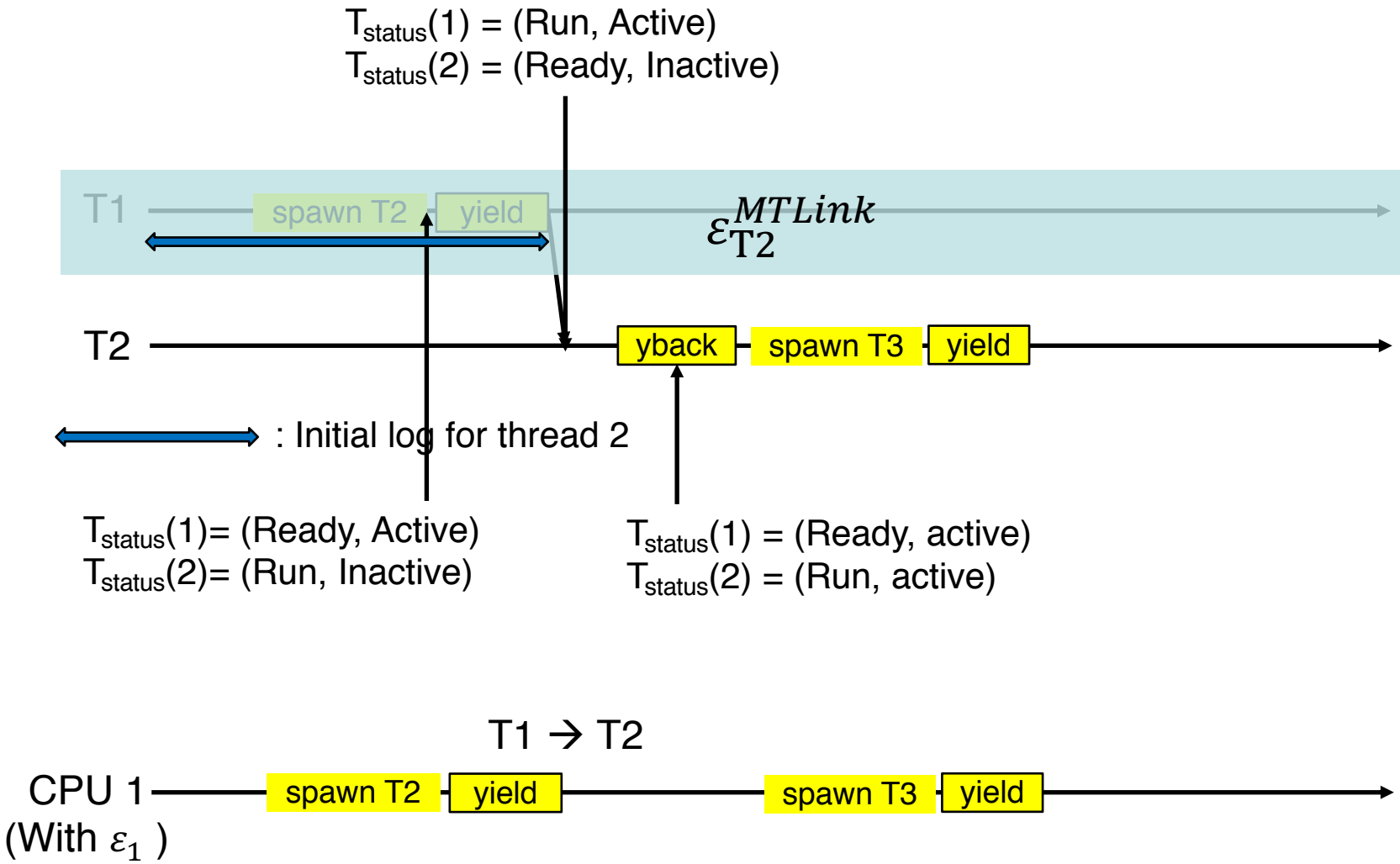
→ AbsRelT
- TLink: arbitrary layer for intermediate machines (scheduling primitives are defined in the machine itself)
- TSched: arbitrary layer with scheduling primitives (Scheduling has a identity behavior)

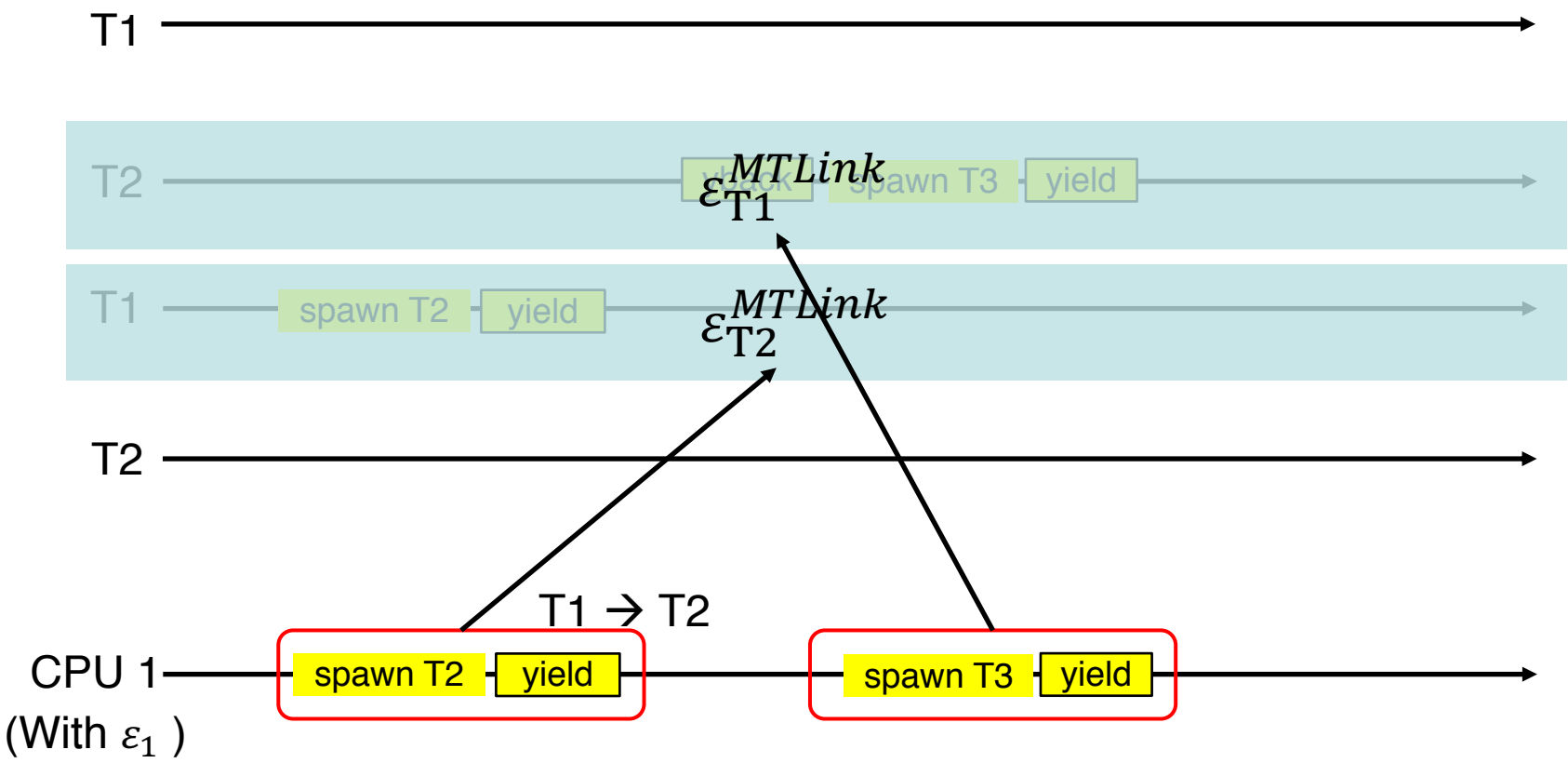
$\text{estate}_{\text{mt}} \coloneqq (tid, mem, \{ti \mapsto lst_{ti}\}, log)\ (\forall ti, ti \in T_{[cid]})$



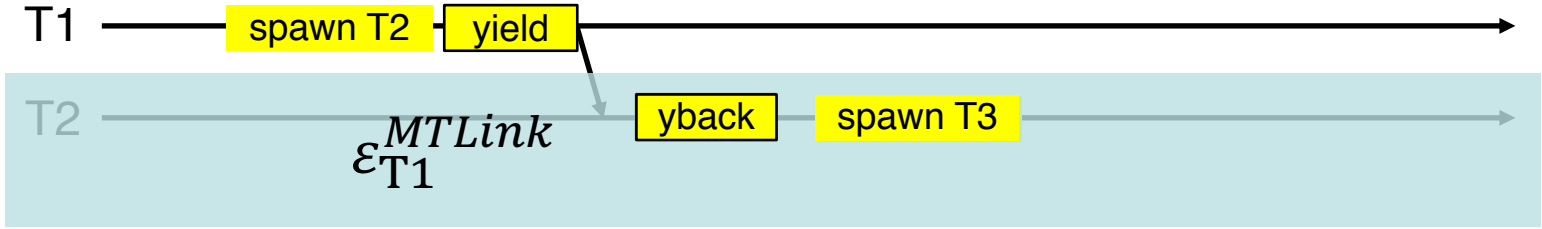
$\text{estate}_{\text{mt}} \coloneqq (tid, mem, \{ti \mapsto lst_{ti}\}, log)\ (\forall ti, ti \in T_{[cid]})$




$T_{status}(1)$ = (Run, Active)
$T_{status}(2)$ = (Ready, Inactive)
T1
spawn T2
yield
$\varepsilon_{T2}^{MTLink}$
T2
yback
spawn T3
yield
: Initial log for thread 2
$T_{status}(1)$= (Ready, Active)
$T_{status}(2)$= (Run, Inactive)
$T_{status}(1)$ = (Ready, active)
$T_{status}(2)$ = (Run, active)
T1 → T2
CPU 1
(With ε_1)
spawn T2
yield
spawn T3
yield


T1
T2
spawn T3
yield
$\mathcal{E}_{T1}^{MTLink}$
T1
spawn T2
yield
$\mathcal{E}_{T2}^{MTLink}$
T2
T1 → T2
CPU 1
(With ε_1)
spawn T2
yield
spawn T3
yield

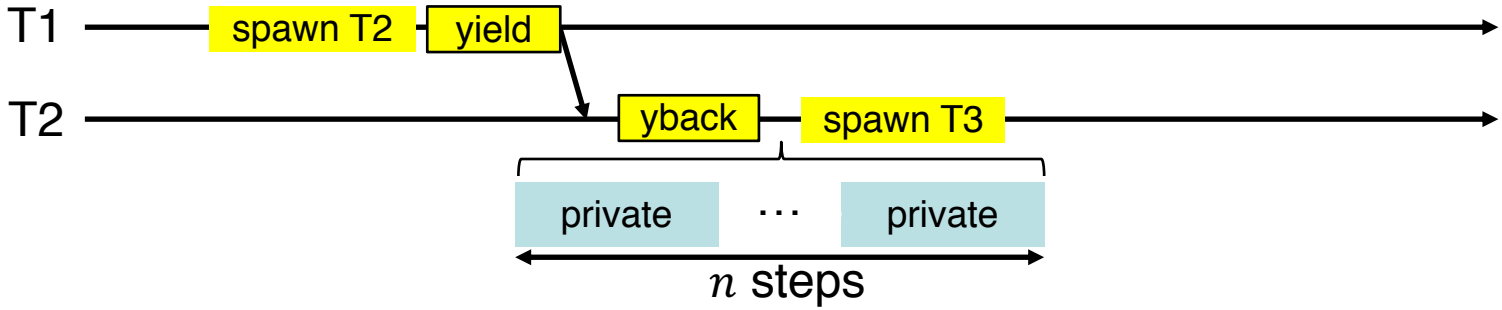
IEAsm
with T1



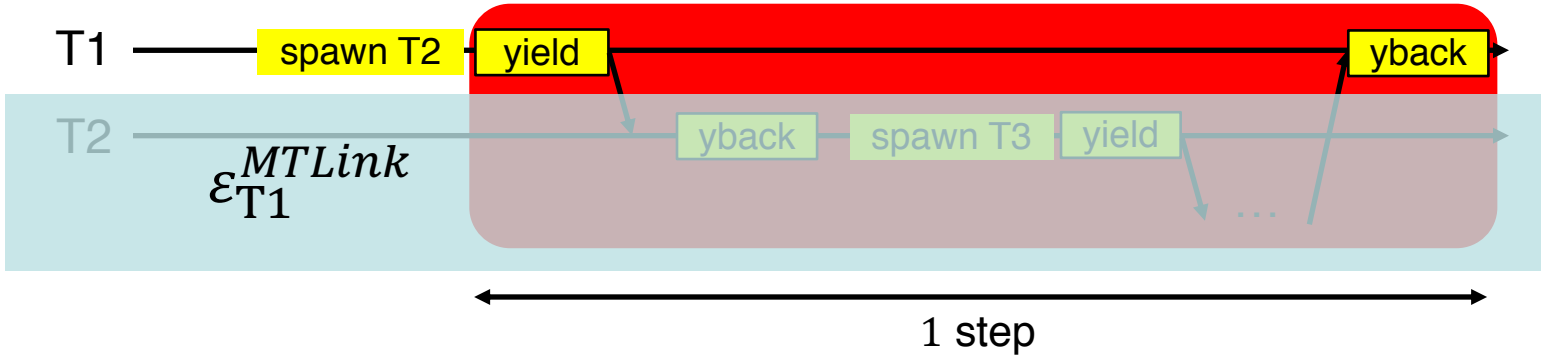
$n < progress$

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

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



TAsm with T1



$n \leq limit$

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

IEAsm with T1