

Fairness in LTL

Lecture #14 of Model Checking

Joost-Pieter Katoen

Lehrstuhl 2: Software Modeling & Verification

E-mail: `katoen@cs.rwth-aachen.de`

May 21, 2007

What did we treat so far?

- LTL semantics: for words, states and transition systems
- LTL equivalence: idempotence, duality, absorption, and expansion
- Dual operators to until: weak until and release
- Expansion law as characteristic equation for until and weak until
- Positive normal form
 - for weak until: exponential blow-up of formula
 - for release: linear transformation
- LTL is a specification formalism for LT properties

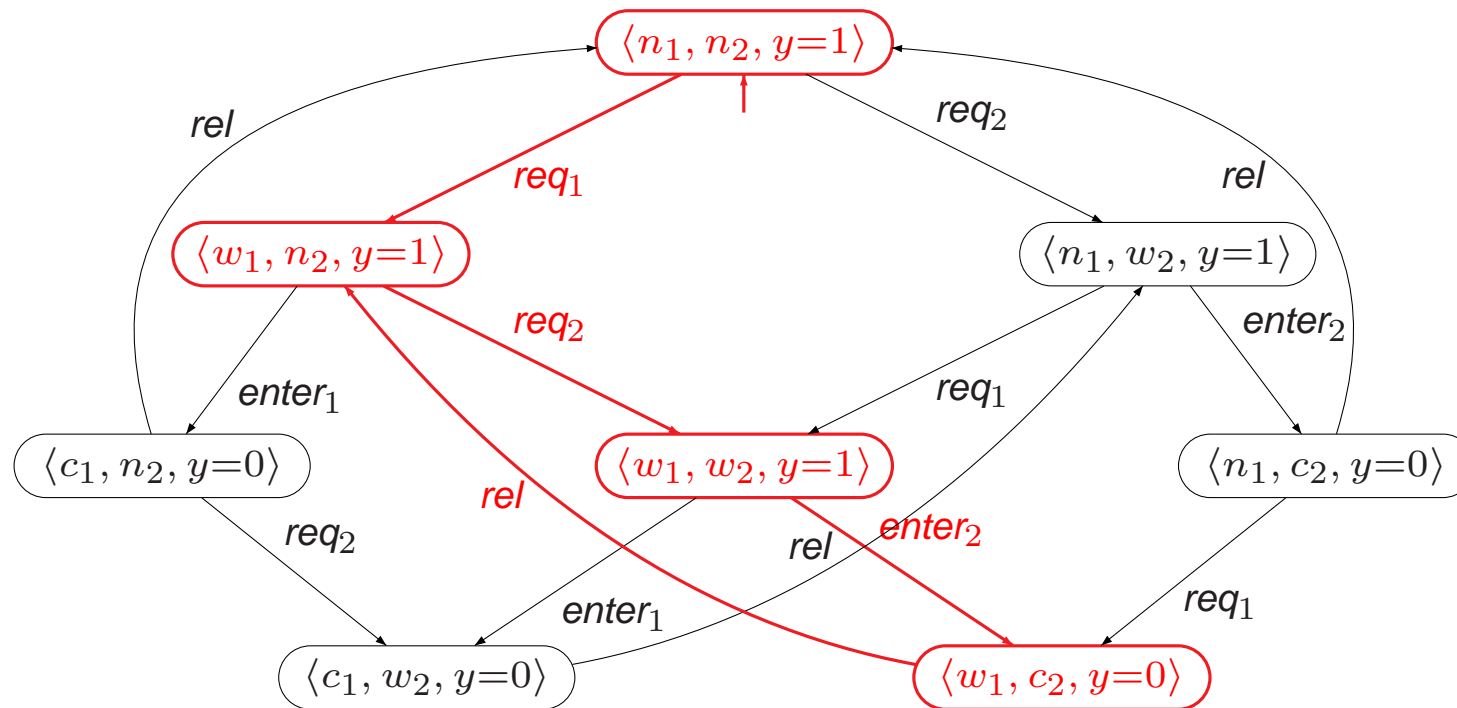
what about fairness in LTL?

Overview Lecture #14

⇒ Repetition: action-based fairness

- State-based fairness in LTL
- Action-based versus state-based fairness
- LTL with fairness constraints

Process one starves



Fairness

- Starvation freedom is often considered under **process fairness**
 - ⇒ there is a fair scheduling of the execution of processes
- **Fairness is typically needed to prove liveness**
 - not for safety properties!
 - to prove some form of progress, progress needs to be possible
- Fairness is concerned with a **fair resolution of nondeterminism**
 - such that it is not biased to consistently ignore a possible option
- Problem: liveness properties constrain infinite behaviours
 - but some traces—that are unfair—refute the liveness property

Summary of fairness

- *Fairness constraints* rule out unrealistic executions
 - by putting constraints on the *actions* that occur along infinite executions
- Unconditional, strong, and weak fairness constraints
 - unconditional \Rightarrow strong fair \Rightarrow weak fair
 - weak fairness rules out the least number of runs; unconditional the most
- *Fairness assumptions* allow distinct constraints on distinct action sets
- (Realizable) fairness assumptions are irrelevant for safety properties
 - important for the verification of liveness properties

Action-based fairness constraints

For set A of actions and infinite run ρ :

- *Unconditional fairness*

some action in A occurs infinitely often along ρ

- *Strong fairness*

if actions in A are *infinitely often* enabled (not necessarily always!)
then some action in A has to occur infinitely often in ρ

- *Weak fairness*

if actions in A are *continuously enabled* (no temporary disabling!)
then it has to occur infinitely often in ρ

Action-based fairness constraints

For $TS = (S, Act, \rightarrow, I, AP, L)$ without terminal states, $A \subseteq Act$,
and infinite execution fragment $\rho = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} \dots$ of TS :

1. ρ is **unconditionally A -fair** whenever: $\underbrace{\forall k \geq 0. \exists j \geq k. \alpha_j \in A}_{\text{infinitely often } A \text{ is taken}}$

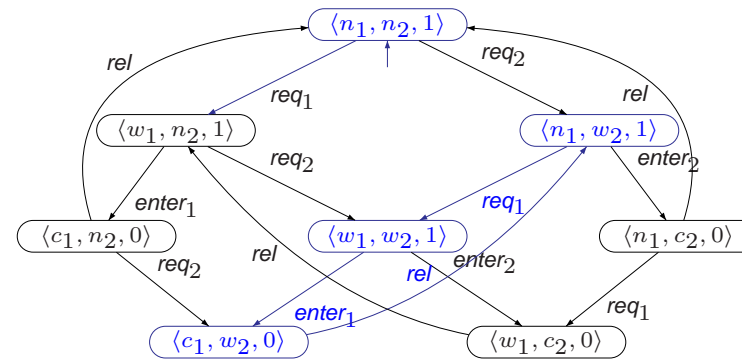
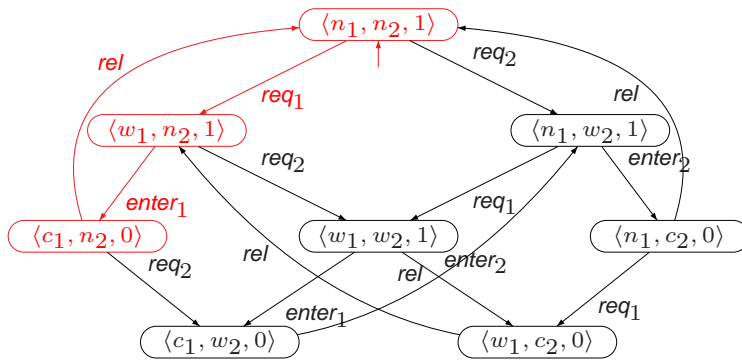
2. ρ is **strongly A -fair** whenever:

$$\underbrace{(\forall k \geq 0. \exists j \geq k. Act(s_j) \cap A \neq \emptyset)}_{\text{infinitely often } A \text{ is enabled}} \implies \underbrace{(\forall k \geq 0. \exists j \geq k. \alpha_j \in A)}_{\text{infinitely often } A \text{ is taken}}$$

3. ρ is **weakly A -fair** whenever:

$$\underbrace{(\exists k \geq 0. \forall j \geq k. Act(s_j) \cap A \neq \emptyset)}_{A \text{ is eventually always enabled}} \implies \underbrace{(\forall k \geq 0. \exists j \geq k. \alpha_j \in A)}_{\text{infinitely often } A \text{ is taken}}$$

Examples



- Run $\langle n_1, n_2, 1 \rangle \xrightarrow{req_1} \langle w_1, n_2, 1 \rangle \xrightarrow{enter_1} \langle c_1, n_2, 0 \rangle \xrightarrow{rel} \langle n_1, n_2, 1 \rangle \xrightarrow{req_1} \dots$
 - is not unconditionally A -fair for $A = \{ enter_2 \}$
 - but strongly A -fair, as in no ρ -state, the action $enter_2$ is enabled
- Run $\langle n_1, n_2, 1 \rangle \xrightarrow{req_2} \langle n_1, w_2, 1 \rangle \xrightarrow{req_1} \langle w_1, w_2, 1 \rangle \xrightarrow{enter_1} \langle c_1, w_2, 0 \rangle \xrightarrow{rel} \langle n_1, w_2, 1 \rangle \dots$
 - is not strongly A -fair: along ρ , $enter_2$ is infinitely often enabled but never taken
 - but weakly A -fair, since $enter_2$ is always not enabled along ρ

Fairness assumptions

- A *fairness assumption* for Act is a triple

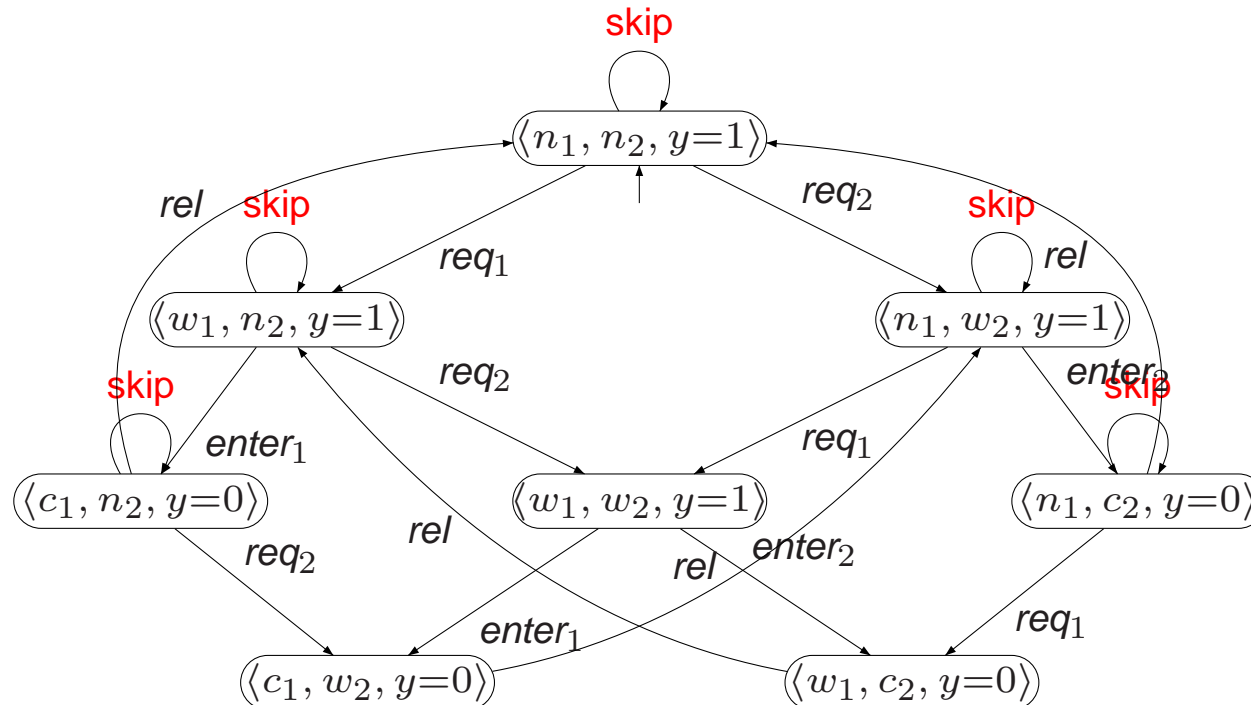
$$\mathcal{F} = (\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak})$$

with $\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak} \in 2^{Act}$.

- Execution ρ is \mathcal{F} -fair if:
 - it is unconditionally A -fair **for all** $A \in \mathcal{F}_{ucond}$, and
 - it is strongly A -fair **for all** $A \in \mathcal{F}_{strong}$, and
 - it is weakly A -fair **for all** $A \in \mathcal{F}_{weak}$
- \mathcal{F} is **realizable** for TS if for any $s \in Reach(TS)$: $FairPaths_{\mathcal{F}}(s) \neq \emptyset$

fairness assumption $(\emptyset, \mathcal{F}', \emptyset)$ denotes strong fairness; $(\emptyset, \emptyset, \mathcal{F}')$ weak, etc.

Example: fairness assumption for mutual exclusion



$$\mathcal{F}' = \left(\emptyset, \underbrace{\{\{enter_1\}, \{enter_2\}\}}_{\mathcal{F}_{strong}}, \underbrace{\{\{req_1\}, \{req_2\}\}}_{\mathcal{F}_{weak}} \right)$$

in any \mathcal{F}' -fair execution each process infinitely often requests access

Fair paths and traces

- Let fairness assumption $\mathcal{F} = (\mathcal{F}_{ucond}, \mathcal{F}_{strong}, \mathcal{F}_{weak})$
- Path $s_0 \rightarrow s_1 \rightarrow s_2 \dots$ is \mathcal{F} -fair if
 - there exists an \mathcal{F} -fair execution $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \dots$
 - $FairPaths_{\mathcal{F}}(s)$ denotes the set of \mathcal{F} -fair paths that start in s
 - $FairPaths_{\mathcal{F}}(TS) = \bigcup_{s \in I} FairPaths_{\mathcal{F}}(s)$
- Trace σ is \mathcal{F} -fair if there exists an \mathcal{F} -fair execution ρ with $trace(\rho) = \sigma$
 - $FairTraces_{\mathcal{F}}(s) = trace(FairPaths_{\mathcal{F}}(s))$
 - $FairTraces_{\mathcal{F}}(TS) = trace(FairPaths_{\mathcal{F}}(TS))$

Fair satisfaction

- TS *satisfies* LT-property P :

$$TS \models P \quad \text{if and only if} \quad \text{Traces}(TS) \subseteq P$$

- TS *fairly satisfies* LT-property P wrt. fairness assumption \mathcal{F} :

$$TS \models_{\mathcal{F}} P \quad \text{if and only if} \quad \text{FairTraces}_{\mathcal{F}}(TS) \subseteq P$$

- TS satisfies the LT property P if *all* its *fair* observable behaviors are admissible

Overview Lecture #14

- Repetition: action-based fairness

⇒ State-based fairness in LTL

- Action-based versus state-based fairness
- LTL with fairness constraints

LTL fairness constraints

Let Φ and Ψ be propositional logic formulas over AP .

1. An *unconditional LTL fairness constraint* is of the form:

$$ufair = \Box \Diamond \Psi$$

2. A *strong LTL fairness condition* is of the form:

$$sfair = \Box \Diamond \Phi \longrightarrow \Box \Diamond \Psi$$

3. A *weak LTL fairness constraint* is of the form:

$$wfair = \Diamond \Box \Phi \longrightarrow \Box \Diamond \Psi$$

Φ stands for “something is enabled”; Ψ for “something is taken”

LTL fairness assumption

- *LTL fairness assumption* = conjunction of LTL fairness constraints
 - the fairness constraints are of any arbitrary type
- Strong fairness assumption: $sfair = \bigwedge_{0 < i \leq k} (\Box \Diamond \Phi_i \longrightarrow \Box \Diamond \Psi_i)$
 - compare this to an action-based strong fairness constraint over A with $|A| = k$
- General format: $fair = unfair \wedge sfair \wedge wfair$
- Rules of thumb:
 - strong (or unconditional) fairness assumptions are useful for solving contentions
 - weak fairness suffices for resolving nondeterminism resulting from interleaving

Fair satisfaction

For state s in transition system TS (over AP) without terminal states, let

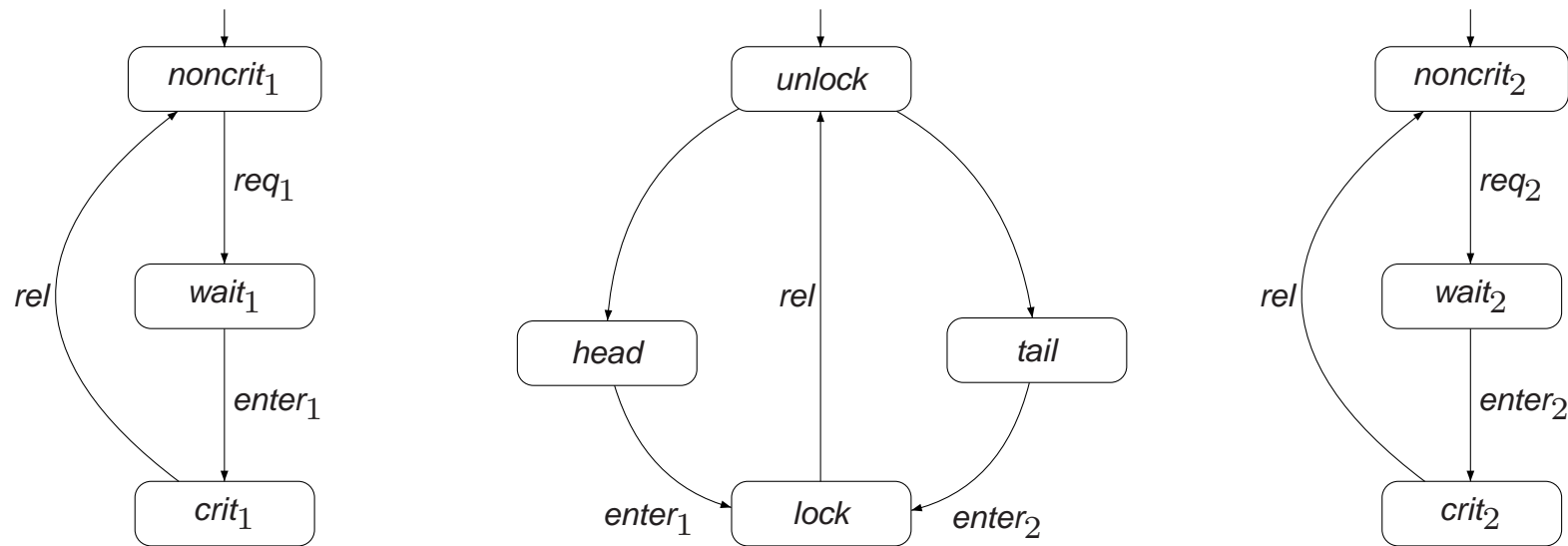
$$\begin{aligned} FairPaths_{fair}(s) &= \{ \pi \in Paths(s) \mid \pi \models_{fair} \text{fair} \} \\ FairTraces_{fair}(s) &= \{ trace(\pi) \mid \pi \in FairPaths_{fair}(s) \} \end{aligned}$$

For LTL-formula φ , and LTL fairness assumption $fair$:

$$\begin{aligned} s \models_{fair} \varphi &\text{ if and only if } \forall \pi \in FairPaths_{fair}(s). \pi \models \varphi \text{ and} \\ TS \models_{fair} \varphi &\text{ if and only if } \forall s_0 \in I. s_0 \models_{fair} \varphi \end{aligned}$$

\models_{fair} is the *fair satisfaction relation* for LTL; \models the standard one for LTL

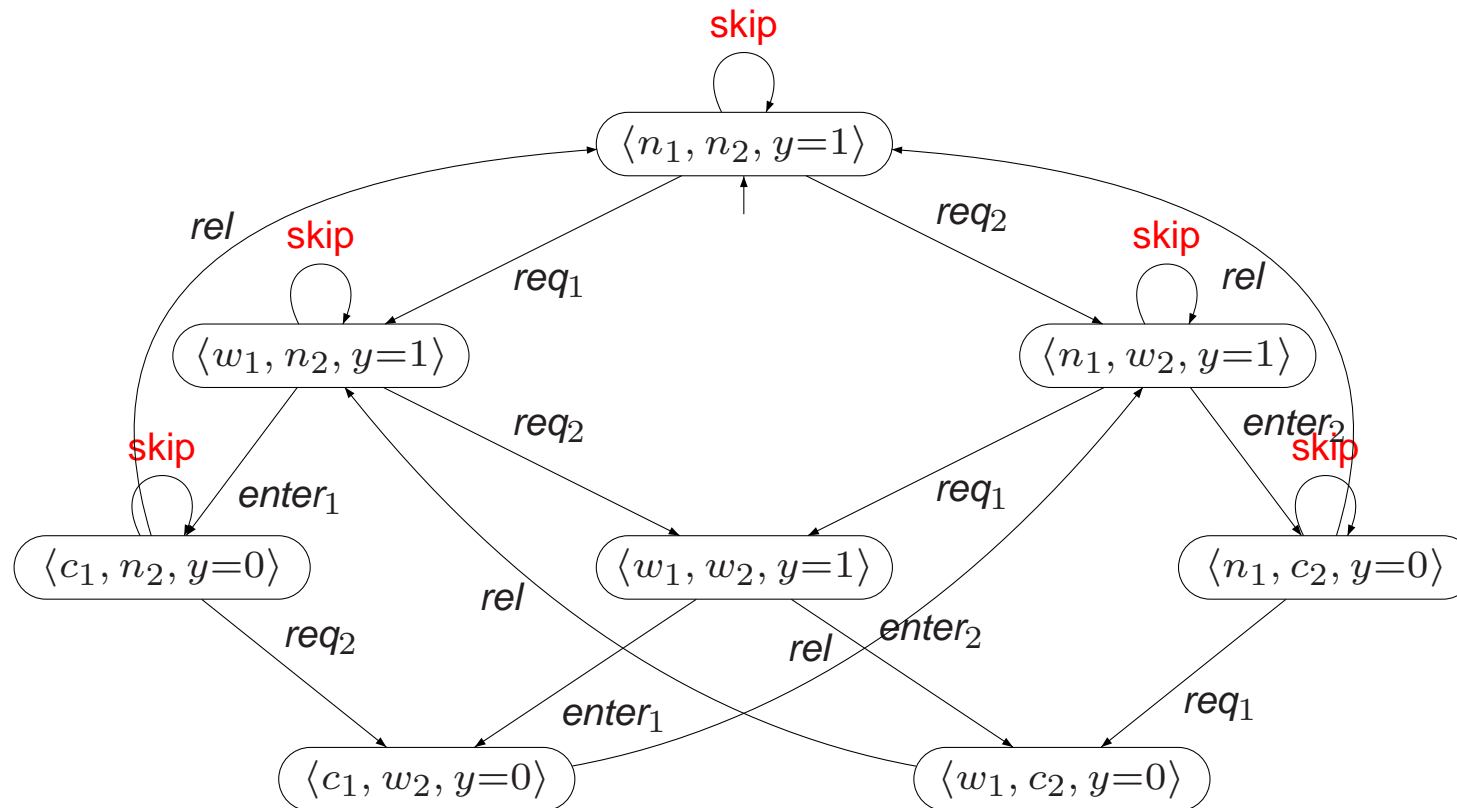
Randomized arbiter



$$TS_1 \parallel \text{Arbiter} \parallel TS_2 \not\models \Box \Diamond \text{crit}_1$$

But: $TS_1 \parallel \text{Arbiter} \parallel TS_2 \models_{\text{fair}} \Box \Diamond \text{crit}_1 \wedge \Box \Diamond \text{crit}_2$ with *fair* = $\Box \Diamond \text{head} \wedge \Box \Diamond \text{tail}$

Semaphore-based mutual exclusion



on black board: some action- versus state-based fairness assumptions

State- versus action-based fairness

- From action-based to (state-based) LTL fairness assumptions:
 - premise: deduce from state label, the possible enabled actions
 - conclusion: deduces from state label, the just executed actions
 - General scheme:
 - copy each non-initial state s and keep track of action used to enter s
 - copy $\langle s, \alpha \rangle$ means s has been entered via action α
- ⇒ Any action-based fairness assumption can be transformed into an equivalent LTL fairness assumption
- the reverse, however, does not hold

Turning action-based into state-based fairness

For $TS = (S, Act, \rightarrow, I, AP, L)$ let $TS' = (S', Act \cup \{begin\}, \rightarrow', I', AP', L')$ with:

- $S' = I \times \{begin\} \cup S \times Act$ and $I' = I \times \{begin\}$
- \rightarrow' is the smallest relation satisfying:

$$\frac{s \xrightarrow{\alpha} s'}{\langle s, \beta \rangle \xrightarrow{\alpha'} \langle s', \alpha \rangle} \quad \text{and} \quad \frac{s_0 \xrightarrow{\alpha} s \quad s_0 \in I}{\langle s_0, begin \rangle \xrightarrow{\alpha'} \langle s, \alpha \rangle}$$

- $AP' = AP \cup \{ enabled(\alpha), taken(\alpha) \mid \alpha \in Act \}$
- labeling function:
 - $L'(\langle s_0, begin \rangle) = L(s_0) \cup \{ enabled(\beta) \mid \beta \in Act(s_0) \}$
 - $L'(\langle s, \alpha \rangle) = L(s) \cup \{ taken(\alpha) \} \cup \{ enabled(\beta) \mid \beta \in Act(s) \}$

it follows: $Traces_{AP}(TS) = Traces_{AP}(TS')$

State- versus action-based fairness

- Strong A -fairness is described by the LTL fairness assumption:

$$sfair_A = \Box \Diamond \bigvee_{\alpha \in A} enabled(\alpha) \rightarrow \Box \Diamond \bigvee_{\alpha \in A} taken(\alpha)$$

- The fair traces of TS and its action-based variant TS' are equal:

$$\begin{aligned} & \left\{ trace_{AP}(\pi) \mid \pi \in Paths(TS), \pi \text{ is } \mathcal{F}\text{-fair} \right\} \\ &= \left\{ trace_{AP}(\pi') \mid \pi' \in Paths(TS'), \pi' \models fair \right\} \end{aligned}$$

- For every LT-property P (over AP): $TS \models_{\mathcal{F}} P$ iff $TS' \models_{fair} P$

Example

Reducing \models_{fair} to \models

For:

- transition system TS without terminal states
- LTL formula φ , and
- LTL fairness assumption $fair$

it holds:

$$TS \models_{fair} \varphi \quad \text{if and only if} \quad TS \models (fair \rightarrow \varphi)$$

verifying an LTL-formula under a fairness assumption can be done
using standard verification algorithms for LTL