Defining Fairness and Fairly Correct Systems

Hagen Völzer

joint work with Daniele Varacca (Paris)

with contributions from Ekkart Kindler (Paderborn)
and Matthias Schmalz (Lübeck)

IBM Zurich Research Laboratory

RWTH Aachen, October 17, 2007

Outline

- We define fairness,
  - i.e., the set of all fairness properties wrt a given system,
  - in line with definitions of *safety* and *liveness* by Lamport, Alpern and Schneider,
  - fairness properties are similar to — but different from — properties of measure 1
- We motivate and define fairly correct systems
  = a system that is correct under *some* fairness assumption
- We study fair model checking:
  - has usually the same complexity as standard model checking, but can be less expensive
  - can be seen as a useful approximation of standard model checking, where no specification of a fairness assumption is needed

Road Map

What is fairness? – A: Examples

- Traditional fairness notions
- Stronger fairness notions

What is fairness? – B: Characterisation

- A first, language-theoretical characterisation
- A game-theoretical characterisation
- A topological characterisation

Fairness and probability

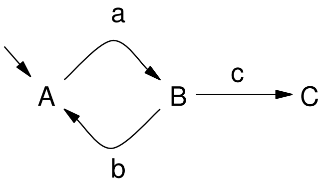
Fairly correct systems

- Motivation and definition
- Fair model checking
- Complete fairness

Road Map

Model of a system

Safety as a transition system



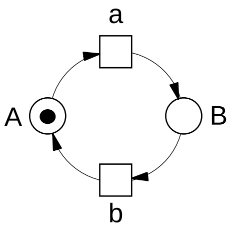
What *may* happen;
generate set of all runs

Liveness as a *fairness assumption*

- *Maximality* ∩
- *Strong fairness* wrt. *c*

What *must* happen;
selects a subset of runs

Sequential maximality

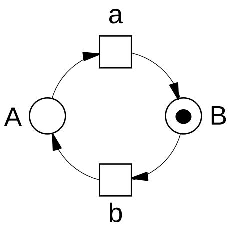


$A, a, B, b, A, \ldots$

Linear-time semantics

- All runs: $a, ab, aba, \ldots, (ab)^\omega$
- Undesired: e.g.. aba
- Assume: Maximality
- Unique maximal run: $(ab)^\omega$

Sequential maximality

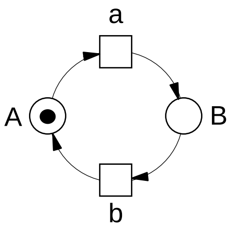


$A, a, B, b, A, \ldots$

Linear-time semantics

- All runs: $a, ab, aba, \ldots, (ab)^\omega$
- Undesired: e.g.. aba
- Assume: Maximality
- Unique maximal run: $(ab)^\omega$

Sequential maximality

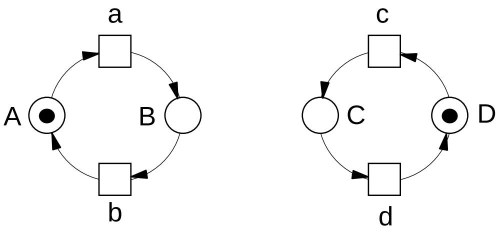


$A, a, B, b, A, \ldots$

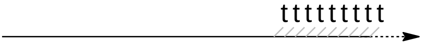
Linear-time semantics

- All runs: $a, ab, aba, \ldots, (ab)^\omega$
- Undesired: e.g.. aba
- Assume: Maximality
- Unique maximal run: $(ab)^\omega$

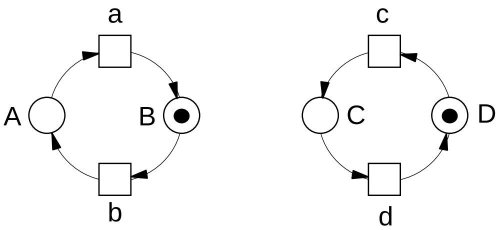
Weak fairness



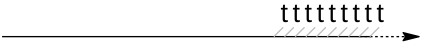
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Weak fairness wrt. t: $\Diamond \Box\, \mathit{enabled}(t) \Longrightarrow \Box \Diamond\, \mathit{taken}(t)$



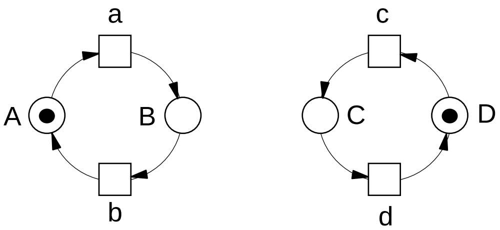
Weak fairness



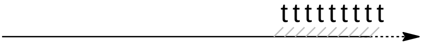
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Weak fairness wrt. t: $\Diamond \Box \, enabled(t) \Longrightarrow \Box \Diamond \, taken(t)$



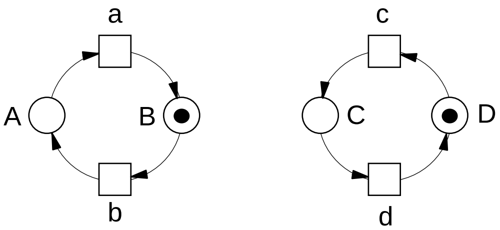
Weak fairness



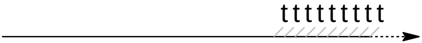
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Weak fairness wrt. t: $\Diamond \Box\, \mathit{enabled}(t) \implies \Box \Diamond\, \mathit{taken}(t)$



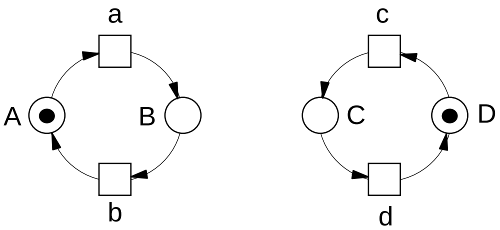
Weak fairness



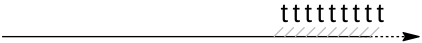
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Weak fairness wrt. t: $\Diamond \Box \, \mathit{enabled}(t) \Longrightarrow \Box \Diamond \, \mathit{taken}(t)$



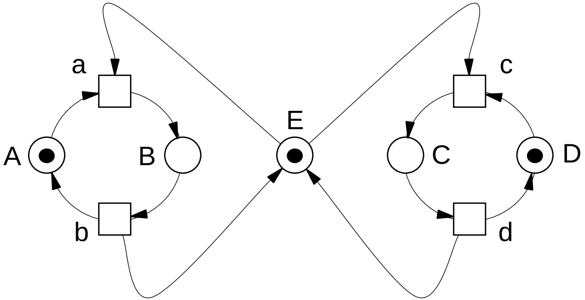
Weak fairness



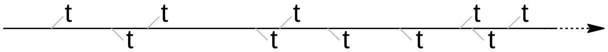
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Weak fairness wrt. t: $\Diamond\Box\, \mathit{enabled}(t) \Longrightarrow \Box\Diamond\, \mathit{taken}(t)$



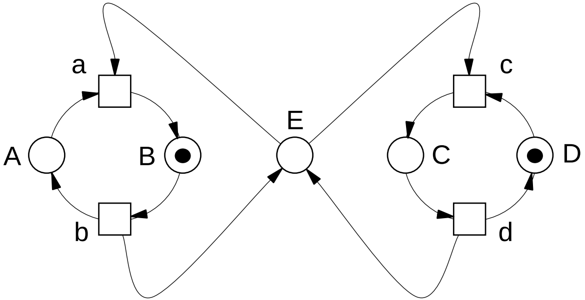
Strong fairness



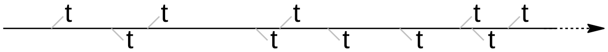
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Strong fairness wrt. t: $\Box\Diamond\, enabled(t) \Longrightarrow \Box\Diamond\, taken(t)$



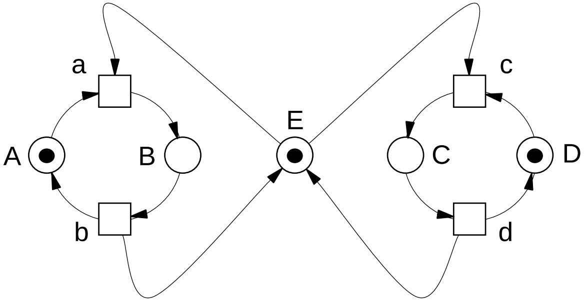
Strong fairness



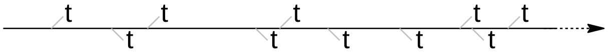
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Strong fairness wrt. t: $\Box\Diamond\, \mathit{enabled}(t) \Longrightarrow \Box\Diamond\, \mathit{taken}(t)$



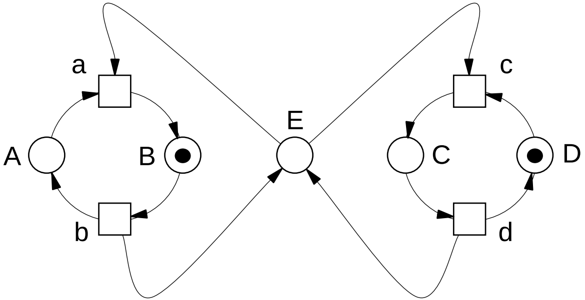
Strong fairness



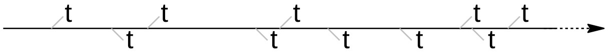
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Strong fairness wrt. t: $\Box\Diamond\, enabled(t) \implies \Box\Diamond\, taken(t)$



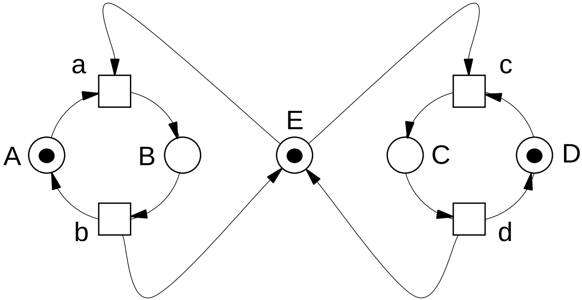
Strong fairness



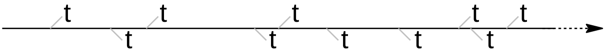
- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Strong fairness wrt. t: $\Box\Diamond\, enabled(t) \implies \Box\Diamond\, taken(t)$



Strong fairness

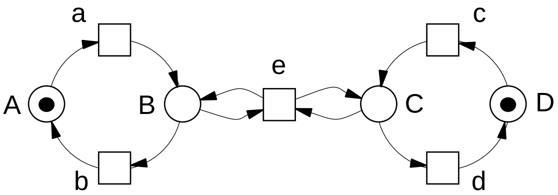


- Undesired: e.g. $(ab)^\omega, (cd)^\omega$
- Strong fairness wrt. t: $\Box\Diamond\, \mathit{enabled}(t) \Longrightarrow \Box\Diamond\, \mathit{taken}(t)$

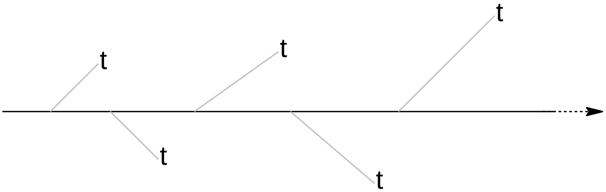


Road Map

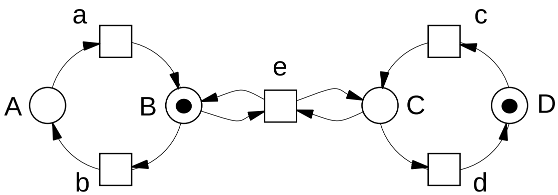
∞-Fairness (E. Best 84)



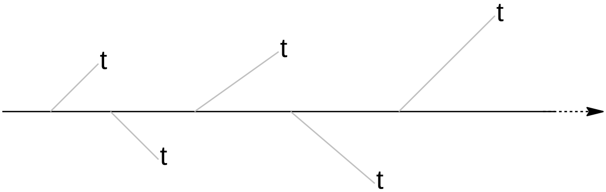
- Undesired: e.g. $(abcd)^\omega$
- ∞-fairness wrt. t: $\Box\Diamond\, reachable(t) \Longrightarrow \Box\Diamond\, taken(t)$



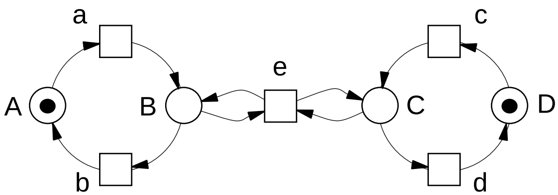
∞-Fairness (E. Best 84)



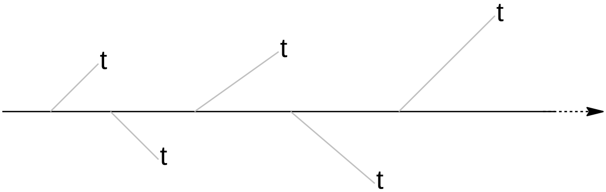
- Undesired: e.g. $(abcd)^\omega$
- ∞-fairness wrt. t: $\Box\Diamond\, \mathit{reachable}(t) \Longrightarrow \Box\Diamond\, \mathit{taken}(t)$



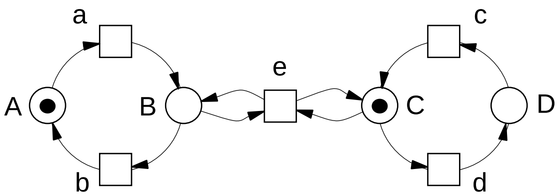
∞-Fairness (E. Best 84)



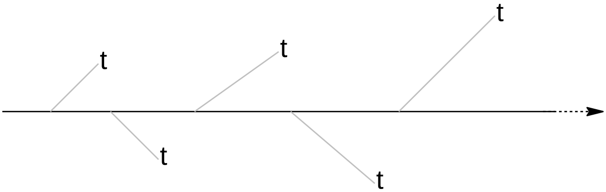
- Undesired: e.g. $(abcd)^{\omega}$
- ∞-fairness wrt. t: $\Box\Diamond\, \mathit{reachable}(t) \Longrightarrow \Box\Diamond\, \mathit{taken}(t)$



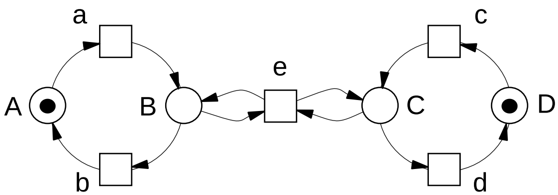
∞-Fairness (E. Best 84)



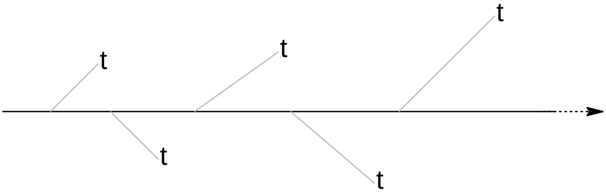
- Undesired: e.g. $(abcd)^\omega$
- ∞-fairness wrt. t: $\Box\Diamond\, reachable(t) \Longrightarrow \Box\Diamond\, taken(t)$



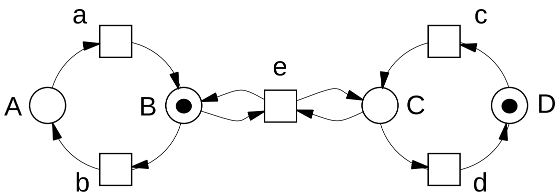
∞-Fairness (E. Best 84)



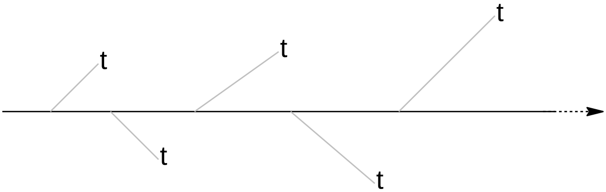
- Undesired: e.g. $(abcd)^\omega$
- ∞-fairness wrt. t: $\Box\Diamond\, \mathit{reachable}(t) \Longrightarrow \Box\Diamond\, \mathit{taken}(t)$



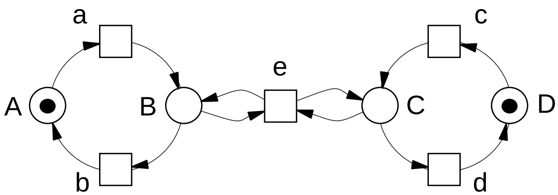
∞-Fairness (E. Best 84)



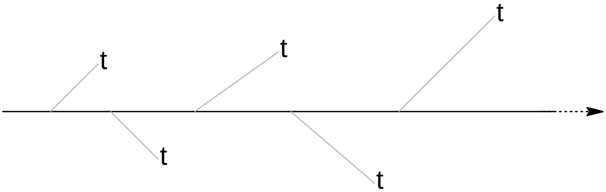
- Undesired: e.g. $(abcd)^\omega$
- ∞-fairness wrt. t: $\Box \Diamond \mathit{reachable}(t) \Longrightarrow \Box \Diamond \mathit{taken}(t)$



∞-Fairness (E. Best 84)

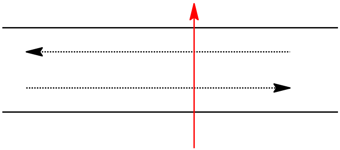


- Undesired: e.g. $(abcd)^\omega$
- ∞-fairness wrt. t: $\Box \Diamond \mathit{reachable}(t) \Longrightarrow \Box \Diamond \mathit{taken}(t)$



What does fairness mean?

A two-lane road



- Always prefer the weaker assumption
- Stronger (than traditional) assumptions can still be ok

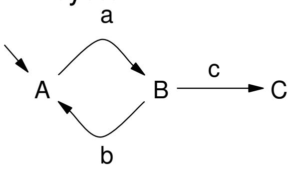
The stronger the fairness assumption, the stronger the potential performance problems

Road Map

Setting

$$x = s_0, s_1, \ldots$$

- Run: finite or infinite sequence of states: $x \in \Sigma^\infty = \Sigma^+ \cup \Sigma^\omega$
- Temporal property: $E \subseteq \Sigma^\infty$
- System $S \subseteq \Sigma^\infty$ = all runs generated by a given transition system



$$\{A, AB, ABA, ABC, ABAB, \ldots\}$$

Question

When is a temporal property F a fairness property for a given system S?

Common pattern:

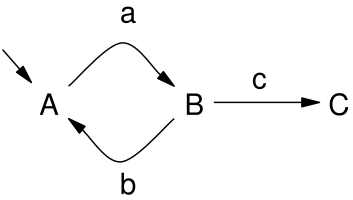
Sufficiently often enabled $\Longrightarrow$ sufficiently often taken.

Defining fairness – History

- Lamport 85, Alpern and Schneider 85: Defined *safety* and *liveness*
- No characterisation of fairness
- Apt, Francez, Katz 88: Necessary criteria for fairness
  - machine closure,
  - *equivalence robustness*, and *liveness enhancement*
- Lamport 2000: Machine closure is the only relevant criterion

Requirement: Machine closure of (S, F)

= each finite run of S can be extended into $S \cap F$



- $\Diamond$ *taken*(b) is not m.c.
- $\Diamond$ *taken*(c) is m.c.

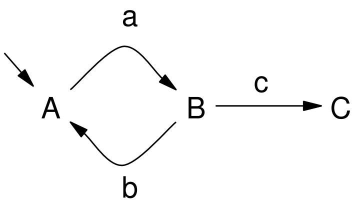
= Fairness does not rule out finite runs of the transition system

  (Transition system cannot 'paint itself into a corner')

- If (S, F) is an implementation (S, F) should be m.c.

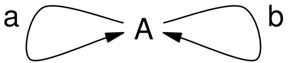
Requirement: Closure under intersection

= intersection of two (countably many) fairness is fairness



- x-Fairness wrt transition 1 $\cap$
- y-Fairness wrt process 2 $\cap$
- z-Fairness wrt ...

Machine closure is not enough



- $E_1 = \Box\Diamond\, taken(a)$
- $E_2 = \Diamond\Box\, taken(b)$
- $E_1 \cap E_2 = \emptyset$

- E_2 prescribes that some choice is not taken sufficiently often
- E_1, E_2 are both machine closed
- $E_1 \cap E_2$ is not machine closed

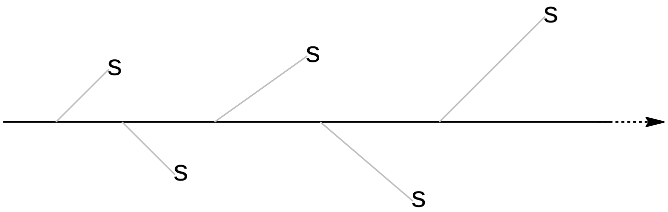
$\Longrightarrow$ machine-closed properties are not closed under intersection (bad for composition)

What do we want?

1. Machine-closed : Minimal requirement for implementability
2. Modular : Intersection of two fairness assumptions is a fairness assumption
3. Popular existing fairness notions fit
4. Otherwise as liberal as possible

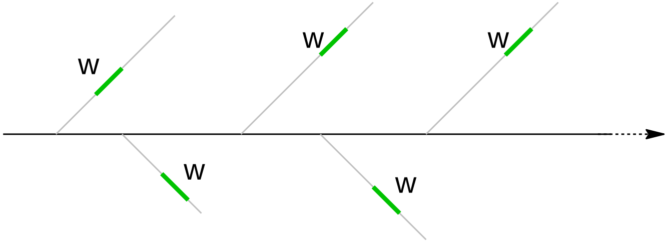
∞-Fairness wrt a state $s \in \Sigma$

Strongest fairness wrt a state



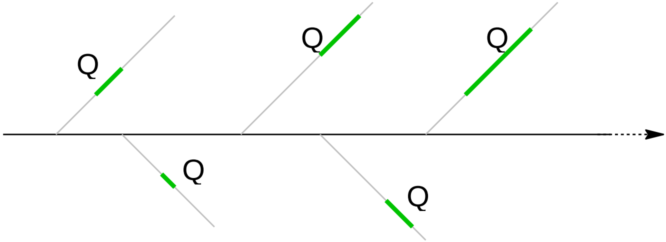
$\Box\, reachable_S(s) \implies \Box \Diamond\, taken(s)$

∞-Fairness wrt a word $w \in \Sigma^+$



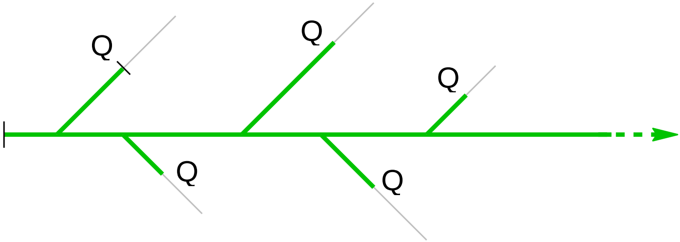
$$\Box\, reachable_S(w) \Longrightarrow \Box \Diamond\, taken(w)$$

∞-Fairness wrt $Q \subseteq \Sigma^+$



$\Box\, reachable_S(Q) \Longrightarrow \Box\Diamond\, taken(Q)$ (informal notation)

Memoryful ∞-fairness wrt $Q \subseteq \Sigma^+$

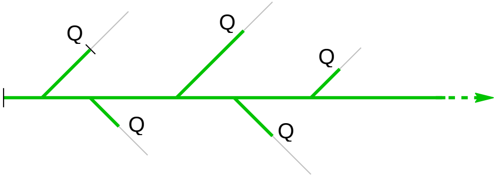


$\Box\, live_S(Q) \Longrightarrow \Box \Diamond Q$ (informal notation)

Examples:

- $Q = \Sigma^+ w$ (∞-Fairness wrt w)
- $Q =$ "$\#a = \#b$" (truly memoryful) "∞-Equifairness"

Defining fairness



Definition

$E \subseteq \Sigma^\infty$ is a fairness property for S iff it contains a property of the form $\Box\, live_S(Q) \Longrightarrow \Box \Diamond Q$ for some $Q \subseteq \Sigma^+$.

Example:

- $\Box \Diamond(\Phi \wedge enabled(t)) \Longrightarrow \Box \Diamond(\Phi \wedge taken(t))$ where Φ is a past formula (α-Fairness (Lichtenstein, Pnueli, Zuck 85))

What do we want? — Revisited

1. Machine-closed : Minimal requirement for implementability ✓
2. Modular : Intersection of two fairness assumption is a fairness assumption ?
3. Popular existing fairness notions fit ✓
4. Otherwise as liberal as possible ?
5. Moreover:
   - Fairness is closed under superset (and arbitrary union) ✓
   - Is $\Diamond \Box \mathit{taken}(b)$ a fairness property ?

Road Map

Game-theoretical characterisation

- Helps to prove or disprove that a given property is a fairness property
- Helps to construct model checking algorithms

Banach-Mazur game

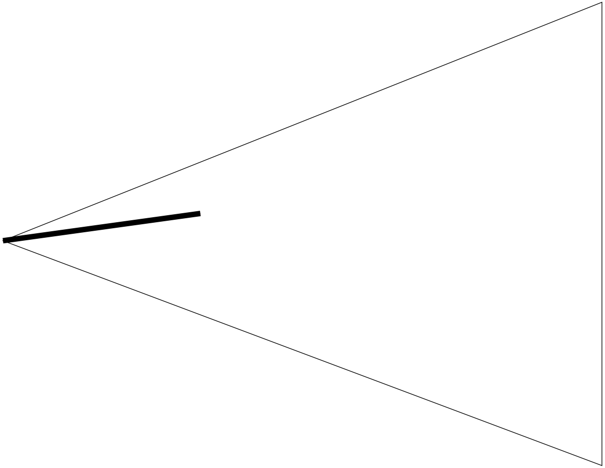
Two players: Scheduler and Opponent

- *Target:* (fairness) property E
- Opponent tries to produce an unfair run $x \notin E$
- Scheduler tries to produce a fair run $x \in E$

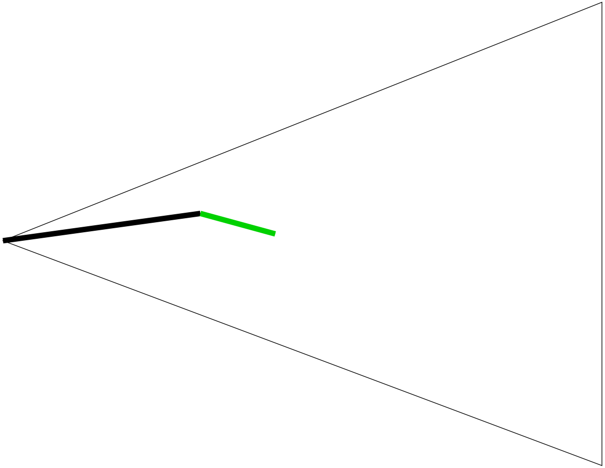
Banach-Mazur game

Run x:

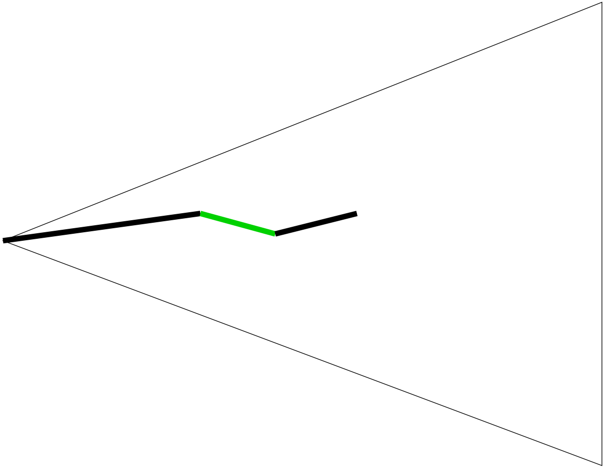
Banach-Mazur game



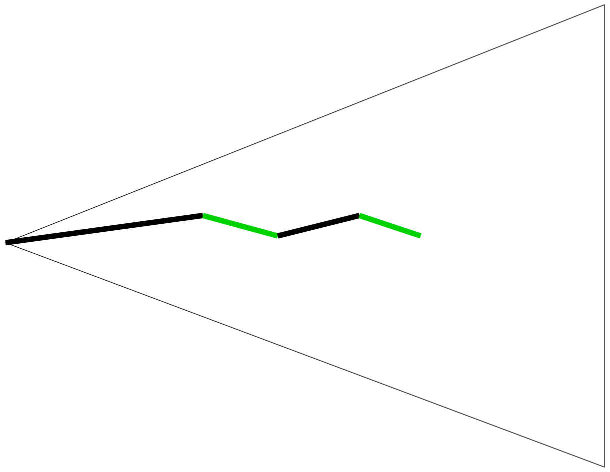
Banach-Mazur game



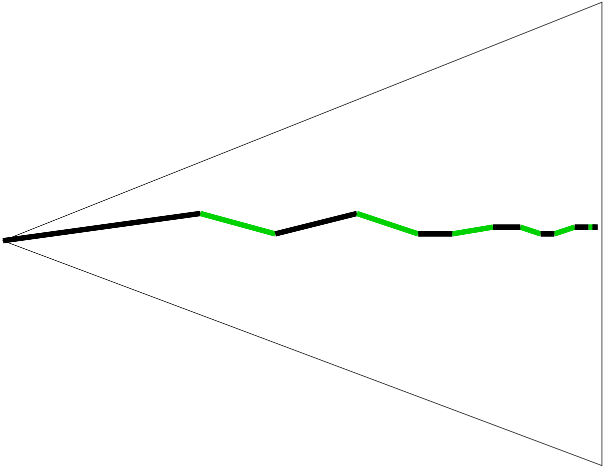
Banach-Mazur game



Banach-Mazur game

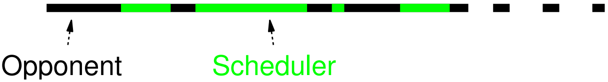


Banach-Mazur game



Banach-Mazur game

Run x:



- Target: $E \subseteq S$
  - scheduler wins if $x \in E$
  - otherwise, opponent wins
- Scheduler can enforce a finite behaviour to be taken infinitely often
- It cannot prevent another finite behaviour from being taken infinitely often

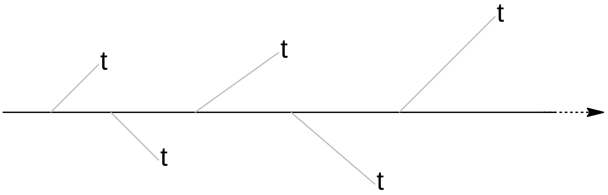
Theorem

F is a fairness property for S iff the Scheduler has a winning strategy for F.

Scheduler has winning strategy for F

Examples

- ∞-Fairness wrt. transition t



- Any weaker property

Counterexamples

- Σ^+
- $\Diamond \Box \, taken(b)$
- $\{\alpha x \mid \alpha \in \Sigma^+\}$ for $x \in \Sigma^\omega$

Closure under Intersection

Strategy: $f : \Sigma^+ \to \Sigma^+$ s.t. α is prefix of $f(\alpha)$.

Theorem

Fairness is closed under countable intersection.

Proof: Let f_i be a winning strategy for $E_i, i = 0, \ldots$. Define

$$f(\alpha) = f_k(f_{k-1}(\ldots f_0(\alpha)\ldots)) \quad \text{where } k = |\alpha|$$

f is a winning strategy for $\bigcap_i E_i$

What do we want? — Revisited

1. Machine-closed : Minimal requirement for implementability ✓
2. Modular : Intersection of two fairness assumption is a fairness assumption ✓
3. Popular existing fairness notions fit ✓
4. Otherwise as liberal as possible ?
5. Moreover:
   - Fairness is closed under superset (and arbitrary union) ✓
   - Is $\Diamond\Box\,\mathit{taken}(b)$ a fairness property ✓

Determinacy

$E \subseteq \Sigma^{\infty}$ is **determinate** if either Scheduler or Opponent has a winning strategy for it.

Determinate

$\uparrow$

Borel

$\uparrow$

Definable by
LTL, Buchi-automata, Linear time μ-calculus

Existence of indeterminate property can be shown using the axiom of choice.

NB. Determinacy yields complete proof strategy for fairness.

Maximality theorem

Theorem

Fairness is a maximal class of determinate properties such that fairness is machine-closed wrt the system and fairness is closed under finite intersection.

Maximality theorem

Theorem

Fairness is a maximal class of determinate properties such that fairness is machine-closed wrt the system and fairness is closed under finite intersection.

Suppose: Scheduler has no strategy for E.

$\Longrightarrow$ Opponent has winning strategy for E, let α be its first move.

$\Longrightarrow$ Scheduler has strategy for $F = \overline{E} \cup \overline{\alpha\uparrow}$

$\Longrightarrow$ α has no extension into $E \cap F$.

$\Longrightarrow$ $E \cap F$ is not machine-closed wrt the system

What do we want? — Revisited

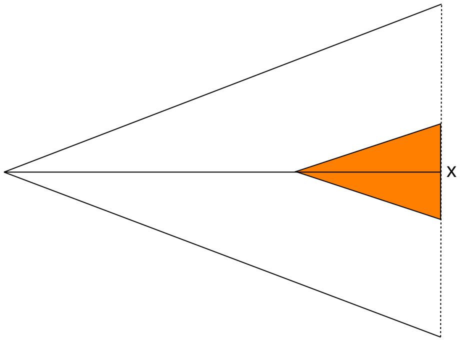
1. Machine-closed : Minimal requirement for implementability $\checkmark$
2. Modular : Intersection of two fairness assumption is a fairness assumption $\checkmark$
3. Popular existing fairness notions fit $\checkmark$
4. Otherwise as liberal as possible $\checkmark$
5. Moreover:
   - Fairness is closed under superset (and arbitrary union) $\checkmark$
   - Is $\Diamond \Box \mathit{taken}(b)$ a fairness property $\checkmark$

Road Map

Topological characterisation

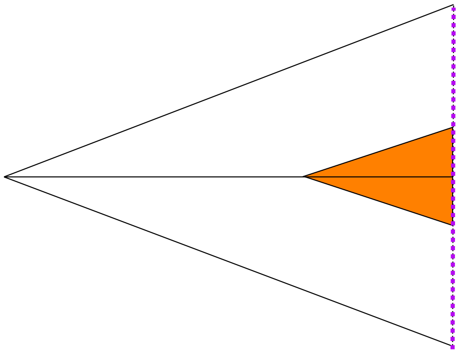
- Fairness properties are the large sets from a topological point of view
- Formalises that most runs are fair
- Leads to an important link to probability theory

Neighbourhood of a run *x*



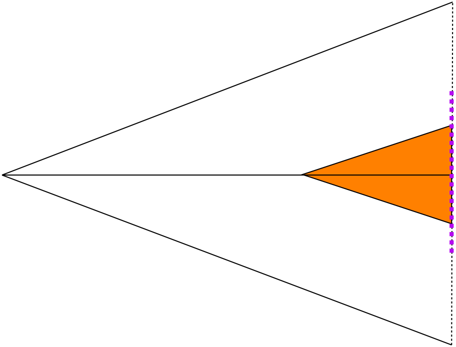
- = includes the set of all runs that share a specific prefix with *x* (*a basic open set*)
- the longer the prefix the smaller the neighbourhood

Dense set E



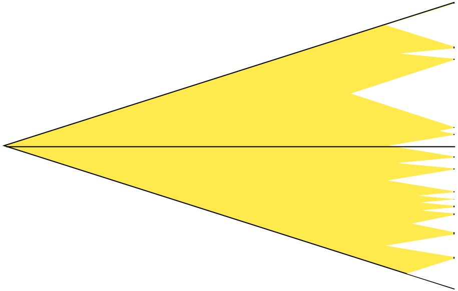
= intersects every neighbourhood
= machine closure
= pollutes the whole space (every neighbourhood is polluted)

Dense set E



- Somewhere dense = dense inside some basic open set (pollutes part of the space)

Nowhere dense set



= not somewhere dense

- Clean runs can stay clear of dirty runs
- Nowhere dense $\Longrightarrow$ small
- Full of holes (holes reachable from everywhere)

Topological characterisation of fairness

Meager set = small:

- = union of countably many nowhere dense sets
- No neighbourhood is meager

Co-meager set = large:

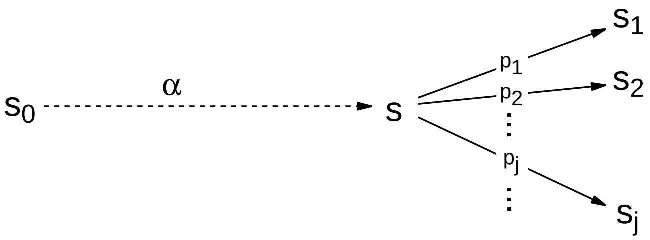
- = complement of a meager set
- $\Leftrightarrow$ fairness

Intermediate set:

- = neither large nor small

Road Map

Probability measure μ on a system S

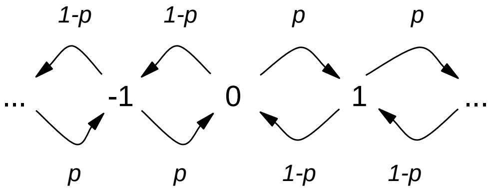


- $\sum_j p_j = 1$
- Assume $p_j \neq 0$
- Bounded: $\exists c > 0$: for all α and p_j : $p_j > c$
- Markov: p_j depends on last state only

Shared properties of topological and probabilistic largeness

- If a set is large its complement is not.
- Any superset of a large set is large.
- The intersection of countably many large sets is large.
- Intersection with a large set preserves size.
- Every large set is dense.
- ...

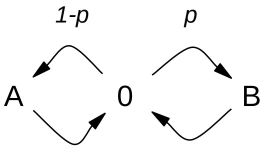
Notions do not coincide! (1/2)



$p \neq \frac{1}{2}$

- $E = \Box \Diamond 0$
- $\mu(E) = 0$ but E is co-meager
- $\mu(\overline{E}) = 1$ but $\overline{E}$ is meager
- System is infinite!

Notions do not coincide! (2/2)



$p \neq \frac{1}{2}$

- $E = \Box \Diamond (\#A = \#B)$
- $\mu(E) = 0$ but E is co-meager
- Property is not ω-regular, hence not expressible in LTL!

Coincidence — Main theorem

Theorem

If S is finite-state, E is ω-regular, μ a bounded probability measure on S then

$$E \text{ is co-meager in } S \Leftrightarrow \mu(E) = 1$$

Coincidence — Main theorem

Theorem

If S is finite-state, E is ω-regular, μ a bounded probability measure on S then

$$E \text{ is co-meager in } S \Leftrightarrow \mu(E) = 1$$

In particular true when μ is a Markov measure and E is LTL expressible.

Proof sketch (E is co-meager in $S \Leftrightarrow \mu(E) = 1$)

Suppose: Scheduler has strategy for E.

$\Longrightarrow$ Scheduler has memoryless strategy for E. (Berwanger, Grädel, and Kreutzer 2003)

$\Longrightarrow$ Scheduler has bounded strategy for E.

$\Longrightarrow$ $\mu(E) = 1$

Proof sketch (E is co-meager in $S \Leftrightarrow \mu(E) = 1$)

Suppose: Scheduler has strategy for E.

$\Longrightarrow$ Scheduler has memoryless strategy for E.
(Berwanger, Grädel, and Kreutzer 2003)

$\Longrightarrow$ Scheduler has bounded strategy for E.

$\Longrightarrow$ $\mu(E) = 1$

Suppose: Scheduler has no strategy for E.

$\Longrightarrow$ Opponent has winning strategy for E, let α be its first move.

$\Longrightarrow$ Scheduler has strategy for $F = \overline{E} \cup \overline{\alpha\uparrow}$

$\Longrightarrow$ $\mu(\overline{E} \cup \overline{\alpha\uparrow}) = 1$ (furthermore: $\mu(\overline{\alpha\uparrow}) < 1$)

$\Longrightarrow$ $\mu(\overline{E}) > 0$ hence $\mu(E) < 1$

Some consequences

- Any ω-regular fairness property has probability 1 under randomised scheduling
- Obtain alternative characterisations for probability 1 (language-theoretic, game-theoretic, topological) in the considered case
- Obtain nice proof for the known fact that concrete values of probabilities do not matter for probability 1

Some consequences

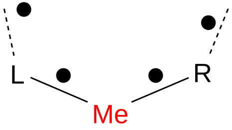
- Any ω-regular fairness property has probability 1 under randomised scheduling
- Obtain alternative characterisations for probability 1 (language-theoretic, game-theoretic, topological) in the considered case
- Obtain nice proof for the known fact that concrete values of probabilities do not matter for probability 1

- Obtain algorithms for model checking fairly correct systems ...

Road Map

Introduction

Five Philosophers



System is not correct!

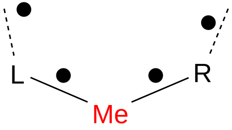
- L and R may ‘conspire’ against Me

SPEC

- mutual exclusion and
- starvation-freedom

Introduction

Five Philosophers



SPEC

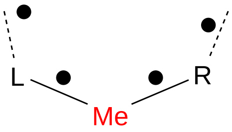
- mutual exclusion and
- starvation-freedom

Get a better system!

- May not be possible (e.g. fault-tolerant consensus, fault-tolerant dining philosophers)
- May not be desirable (price to pay)
- May not be necessary (SPEC is satisfied in practice)

Introduction

Five Philosophers



SPEC

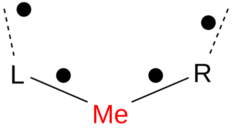
- mutual exclusion and
- starvation-freedom

Live with the system at hand!

- System is almost correct
- ‘Most’ runs satisfy SPEC
- Occurs e.g. in fault-tolerant distributed algorithms

Introduction

Five Philosophers



SPEC

- mutual exclusion and
- starvation-freedom

How to verify the system?

- System is not correct wrt SPEC
  - Pragmatic approach: weaken SPEC
- System is almost correct (‘most’ runs satisfy SPEC)
  - How to formalize this?
  - How to verify this?

Relaxations of correctness

Let S be the set of all runs of the system.

Almost Correct

- SPEC is probabilistically large (i.e. $\mu(SPEC) = 1$)
- needs probability measure μ on S

Fairly Correct (New!)

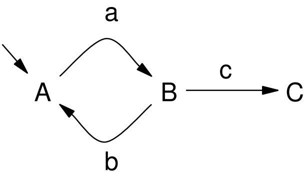
- SPEC is topologically large (i.e. SPEC is co-meager wrt S)

$\Leftrightarrow$ there is a fairness assumption F for S such that $S \cap F \subseteq SPEC$

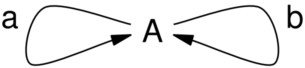
$\Leftrightarrow$ $SPEC$ is a fairness property for S !

Coincide for finite-state systems and ω-regular specifications.

Technical examples



- SPEC: No *a* after a *c*, correct and fairly correct
- SPEC: Termination, not correct but fairly correct



- SPEC: $\Diamond\Box\, taken(a)$, not correct and not fairly correct
- SPEC: $\Box\Diamond\, taken(a)$, not correct but fairly correct

Road Map

What is fairness? – A: Examples

Traditional fairness notions

Stronger fairness notions

What is fairness? – B: Characterisation

A first, language-theoretical characterisation

A game-theoretical characterisation

A topological characterisation

Fairness and probability

Fairly correct systems

Motivation and definition

Fair model checking

Complete fairness

Fair linear-time model checking

$$M \models_{\approx} \phi$$

- Fair model checking for LTL is PSPACE-complete; time linear in $|M|$ and exponential in $|\phi|$ (Courcoubetis and Yannakakis 95, Vardi 85)
- The same is true for specifications given by Büchi automata (Vardi 85)
- Standard model checking has the same complexity in both cases
- No implementation available

Alternative: Use of existing technology

Assume ϕ is in reactivity normal form:

$$\phi = \bigwedge_{i=1}^{n} (\Box \Diamond h_i \vee \Diamond \Box g_i)$$

where h_i and g_i are *past formulas*.

- We have linear translation of ϕ into CTL+past formula Φ s.t. M is fairly correct wrt ϕ if and only if M is correct wrt Φ
- There is a model checker for CTL+past: TLV
- Model checking CTL+past is PSPACE-complete

Branching time: CTL, CTL*

- We obtained optimal algorithms for “relaxed” versions of CTL and CTL*
  (Replace ‘for all paths’ by ‘for almost all paths’)
- Complexity is the same as for standard model checking for these languages
- Algorithm for CTL uses translation into standard CTL

Fair model checking can be less expensive than standard model checking

- Usually they have the same complexity (see above)
- Remains true for many subclasses, e.g. $L(\Diamond)$ is co-NP-complete (New: Schmalz 2007)
- But not always:

Theorem

Fair model checking of $L(\Box\Diamond)$ can be done in linear time (while standard model checking of $L(\Box\Diamond)$ is co-NP-complete)

More results from Schmalz 2007

- Algorithm for $L(\Box\Diamond)$ can be integrated with algorithm of Courcoubetis and Yannakakis (sometimes exponentially better, never worse)
- Algorithm can be extended to handle past operators
- Algorithm can be extended to return diagnostic information
- Crucial tool: Game-theoretic characterisation

Road Map

What is fairness? – A: Examples

- Traditional fairness notions
- Stronger fairness notions

What is fairness? – B: Characterisation

- A first, language-theoretical characterisation
- A game-theoretical characterisation
- A topological characterisation

Fairness and probability

Fairly correct systems

- Motivation and definition
- Fair model checking
- Complete fairness

Question

Is there a strongest fairness property F for S, i.e.,

S is fairly (almost) correct wrt $SPEC$ iff $F \cap S \subseteq SPEC$?

Benefit: Reduces proving fair correctness to proving satisfaction conditioned on F.

Answer (1/2)

- No, not in general.
  (Fairness is not closed under arbitrary intersection.)

Answer (1/2)

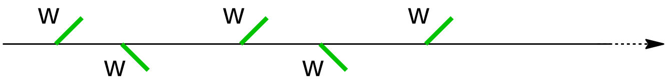
- No, not in general.
  (Fairness is not closed under arbitrary intersection.)
- Yes, if we are interested in a countable class $\mathcal{F}$ of properties only (e.g. LTL, ω-regular)

$$F_{\mathcal{F}} = \bigcap \{F \in \mathcal{F} \mid F \text{ is a fairness property for } S\}$$

is complete for $\mathcal{F}$.

Answer (2/2)

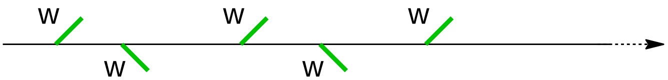
- Word fairness is complete for LTL and ω-regular



- α-Fairness is also known to be complete

Answer (2/2)

- Word fairness is complete for LTL and ω-regular



- α-Fairness is also known to be complete
- Word fairness is not ω-regular
- No ω regular-property is complete in general
- State fairness is complete for — as well as expressible in $L(\diamond)$

Conclusion (1/3)

Performance

$\uparrow$

Full functional correctness (under fairness)

$\uparrow$

Fair correctness

$\uparrow$

Machine closure

$\uparrow$

Safety

Higher assurance

Lower assurance

There are more generic relaxations of correctness (Berwanger et al. 2003, Pistore and Vardi 2003)

Conclusion (2/3)

Fair model checking

- allows to verify systems that are only fairly correct
- can be used as an approximation to standard model checking
  - has the same complexity for LTL, can even be better for subclasses
  - no need to specify any fairness assumption
  - difference in bugs covered is small

Conclusion (3/3)

- Definition of fairness carries over to other domains (Mazurkiewicz traces, pomsets, ...)
- Game-theoretic characterisation could help to simplify other algorithms in probabilistic model checking

References

- Schmalz, V., Varacca 2007: Model checking almost all paths can be less expensive than checking all paths. FSTTCS 2007
- Varacca and V.: Temporal logics and model checking for fairly correct systems. LICS 2006
- V., Varacca, and Kindler: Defining Fairness. CONCUR 2005
- Pistore and Vardi: The planning spectrum ..., LICS 2003
- Berwanger, Grädel, and Kreutzer: Once upon a time in a west ..., LPAR 2003
- Oxtoby: Measure and Category ..., Springer 1971