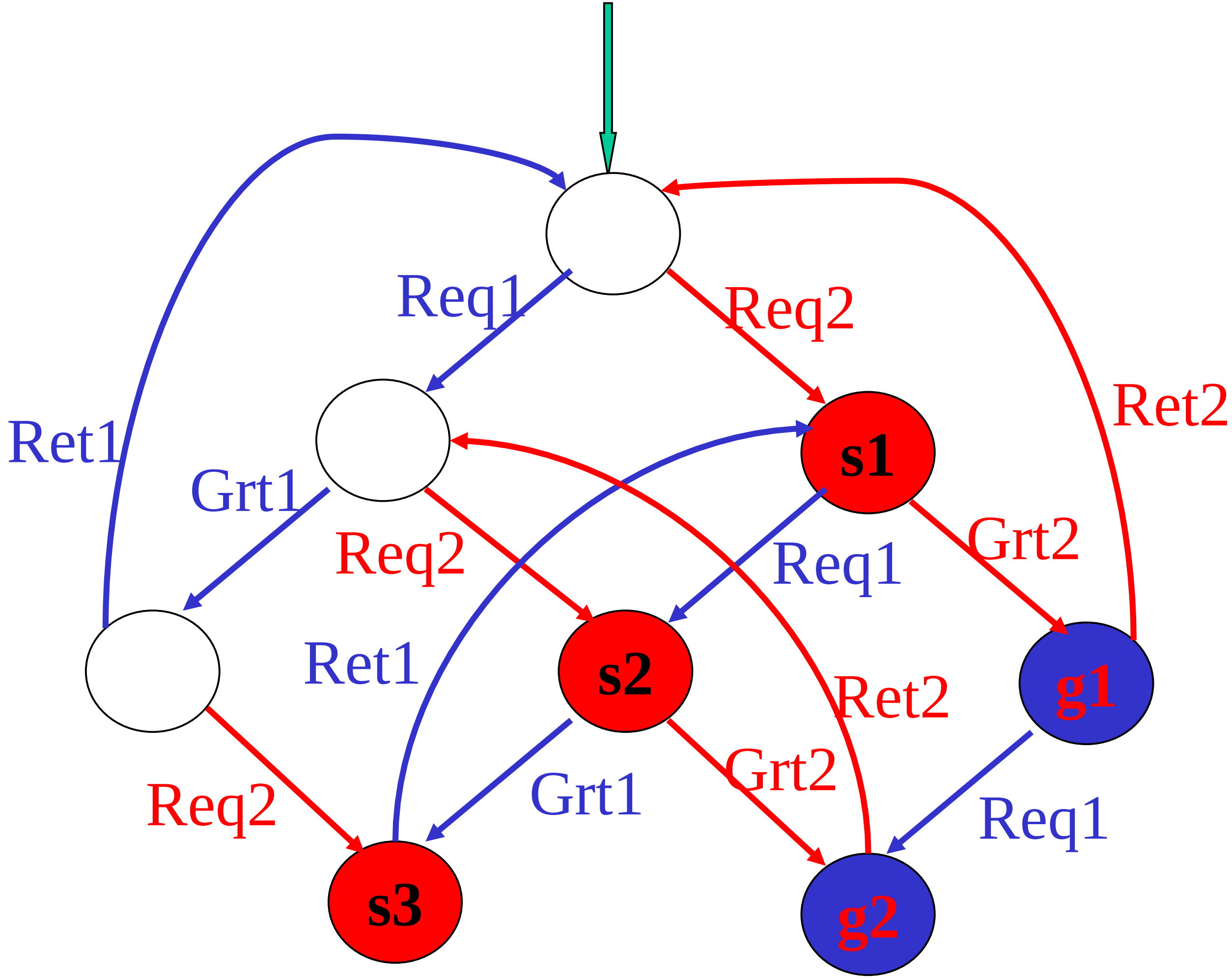
Tecniche di Specifica e di Verifica

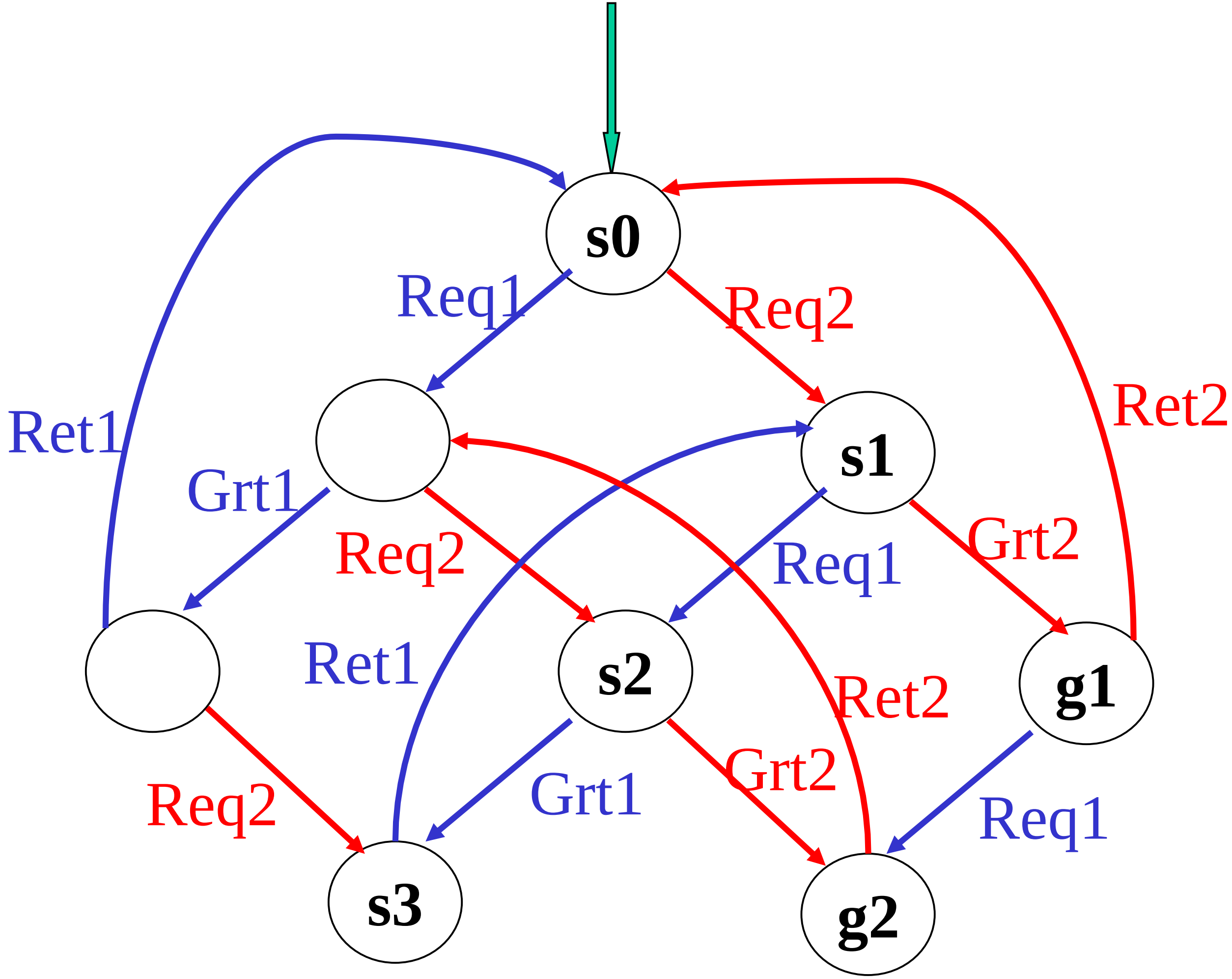
Model Checking under Fairness

Fairness

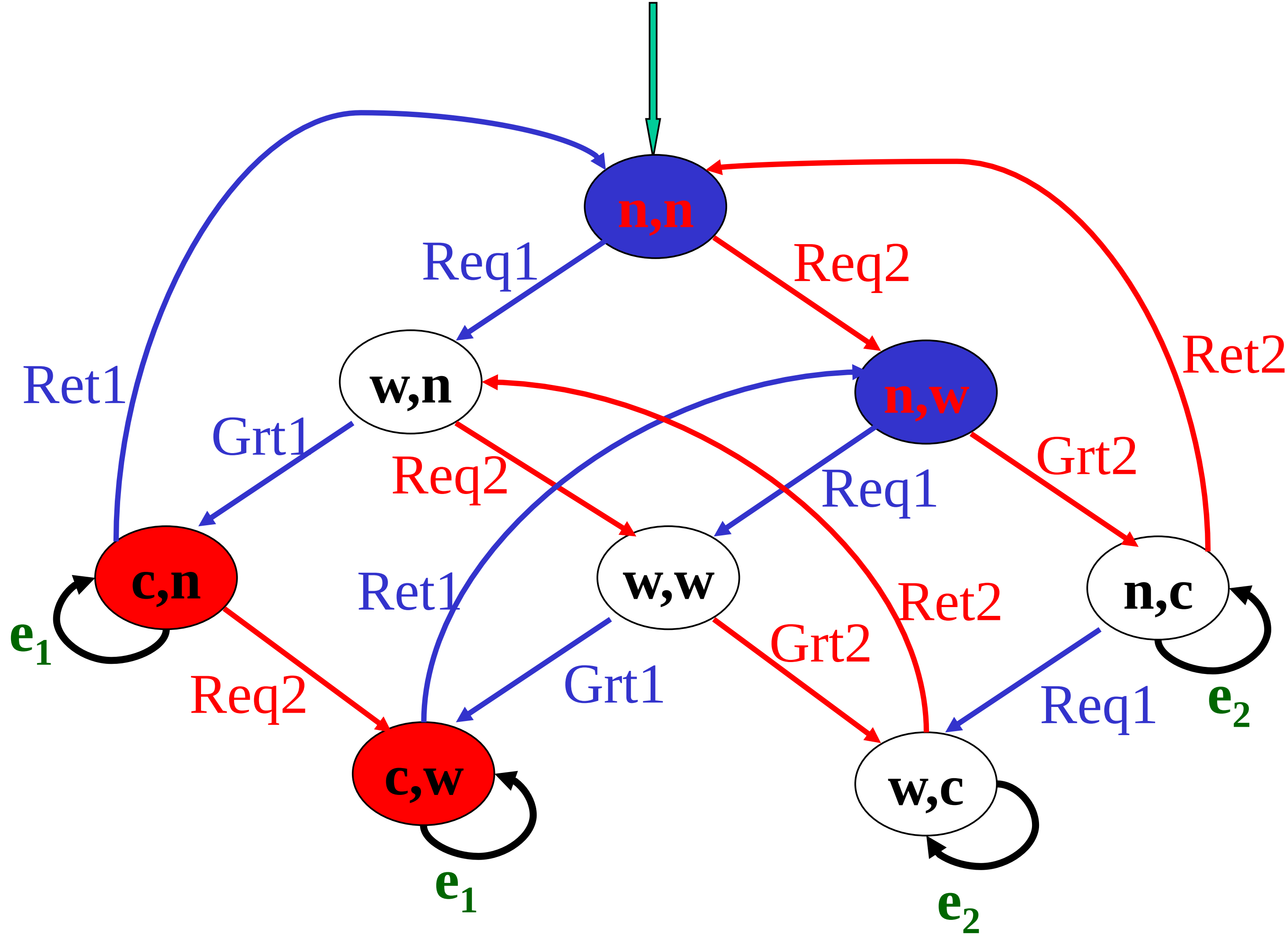
- **K =(S, S_0, R, AP, L)**
- **K** may ***not*** be able to capture ***exactly*** the desired executions.
  - Too generous.
- Use ***fairness constraints*** to rule out **undesired executions**.



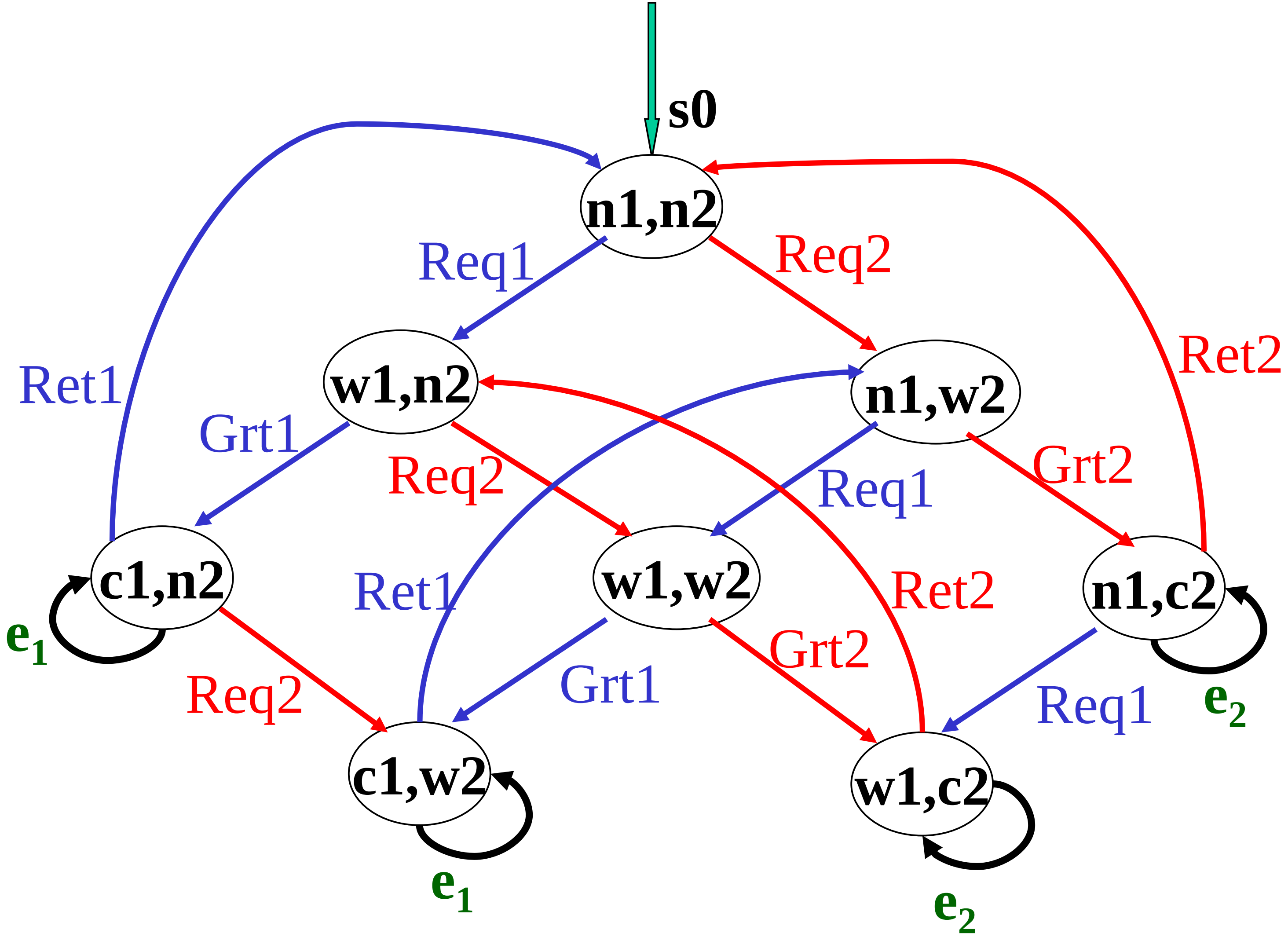
a **computation** in which **s1** or **s2** or **s3** is visited **infinitely** often but **g1** and **g2** are visited only **finitely often** is **unfair**.



$K, s0 \nvDash AG\ (\ req2 \rightarrow AF\ grt2\ )$



A computation in which **(c,n)** or **(c,w)** is visited infinitely often but **(n,n)** and **(n,w)** are visited only finitely often.


s0
n1,n2
Req1
Req2
w1,n2
n1,w2
Ret1
Ret2
Grt1
Req2
Grt2
Req1
c1,n2
w1,w2
n1,c2
Ret1
Ret2
e1
Req2
Grt1
Grt2
Req1
e2
c1,w2
w1,c2
e1
e2

K, s0 ⊨ EF EG c1 !

Fairness

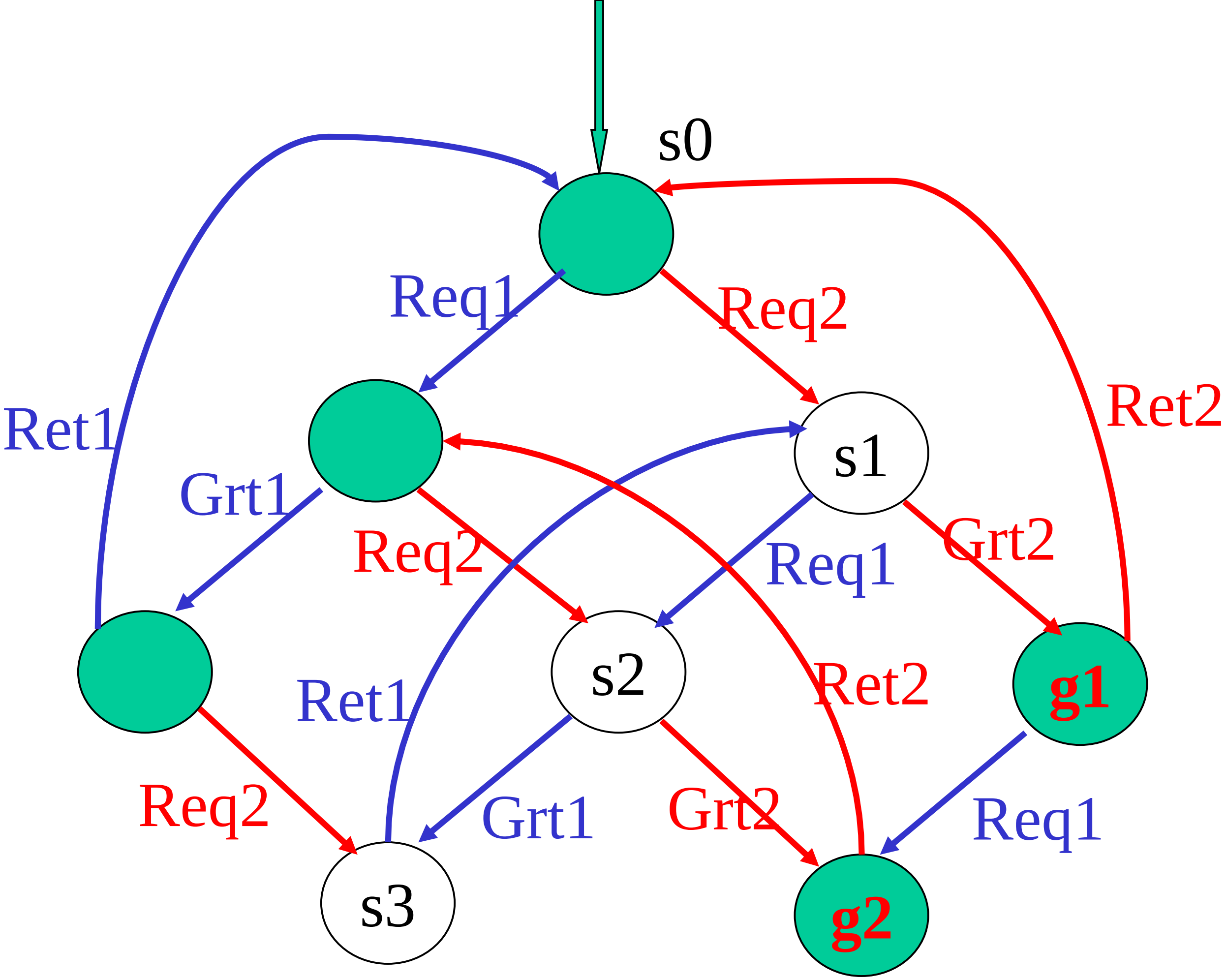
- The ***first kind of unfairness*** has to do with a ***bad scheduling policy***.
  - Find a better allocation scheme.
    - Turn-based.
- The ***second kind of unfairness*** is unavoidable.
- ***Solution***:
  - Consider only ***fair computations***.

Fairness

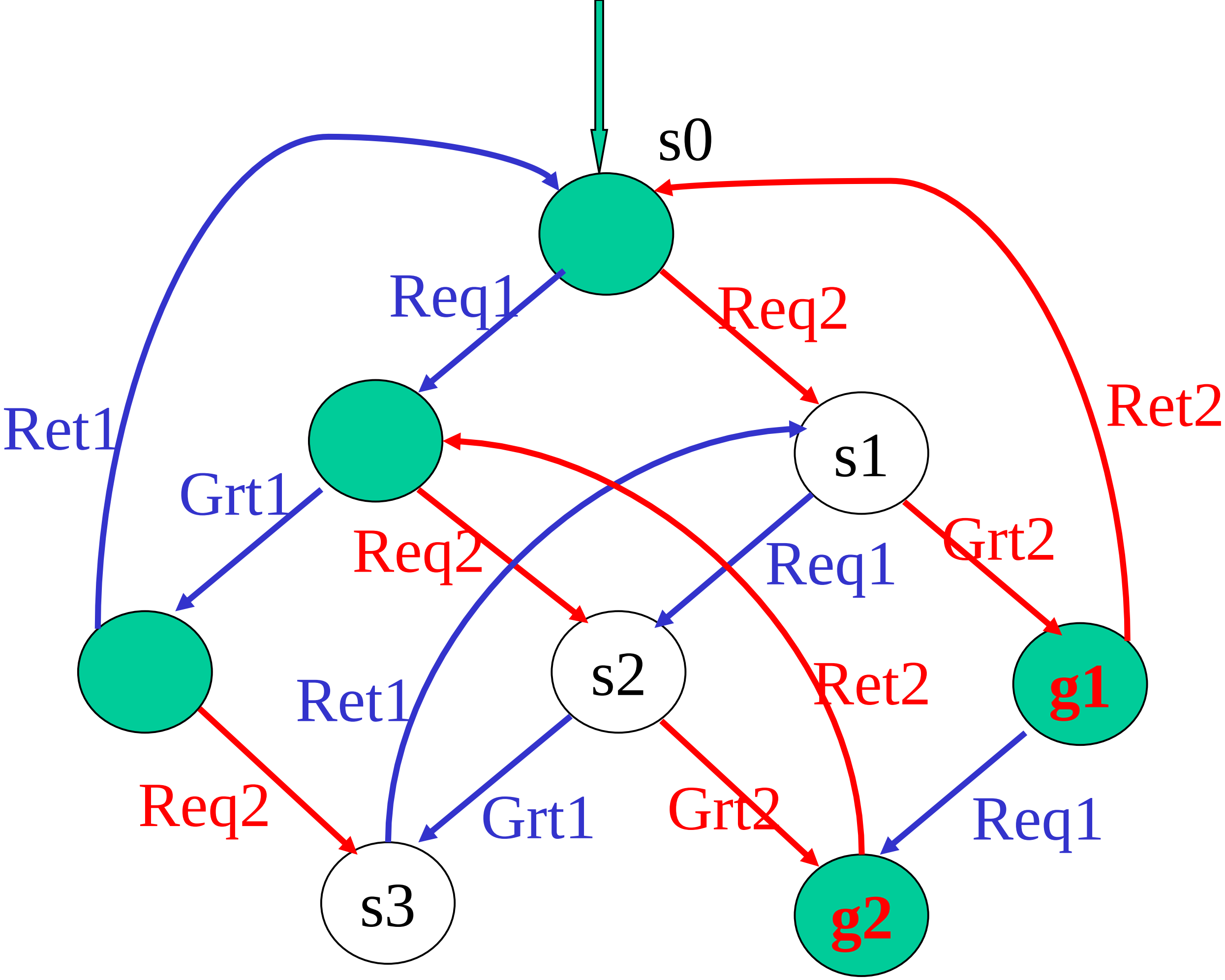
- ***Fair Kripke Structures***.
- First Attempt:
  - **K = (S, S_0, R, AP, L, $\mathcal{F}$)**
  - $\mathcal{F} \subseteq$ **S** (***fairness constraint***)
- π is a ***fair computation iff***:
  - It is a computation.
  - **inf(π)** $\cap\ \mathcal{F} \neq \emptyset$
  - **inf(π) = {s : s** appears infinitely often in **π}**

Fairness

- ***Fair Kripke Structures***.
- **$K = (S, S_0, R, AP, L, \mathcal{F}_1, \mathcal{F}_2,..,\mathcal{F}_n)$**
  - $\mathcal{F}_i \subseteq S$ (***fairness constraints***)
- π is a ***fair computation*** ***iff***:
  - It is a computation.
  - $\mathbf{inf}(\pi) \cap \mathcal{F}_i \neq \emptyset$ for each $i = 1, 2,..,n$
  - $\mathbf{inf}(\pi) = \{s : s$ appears infinitely often in $\pi\}$



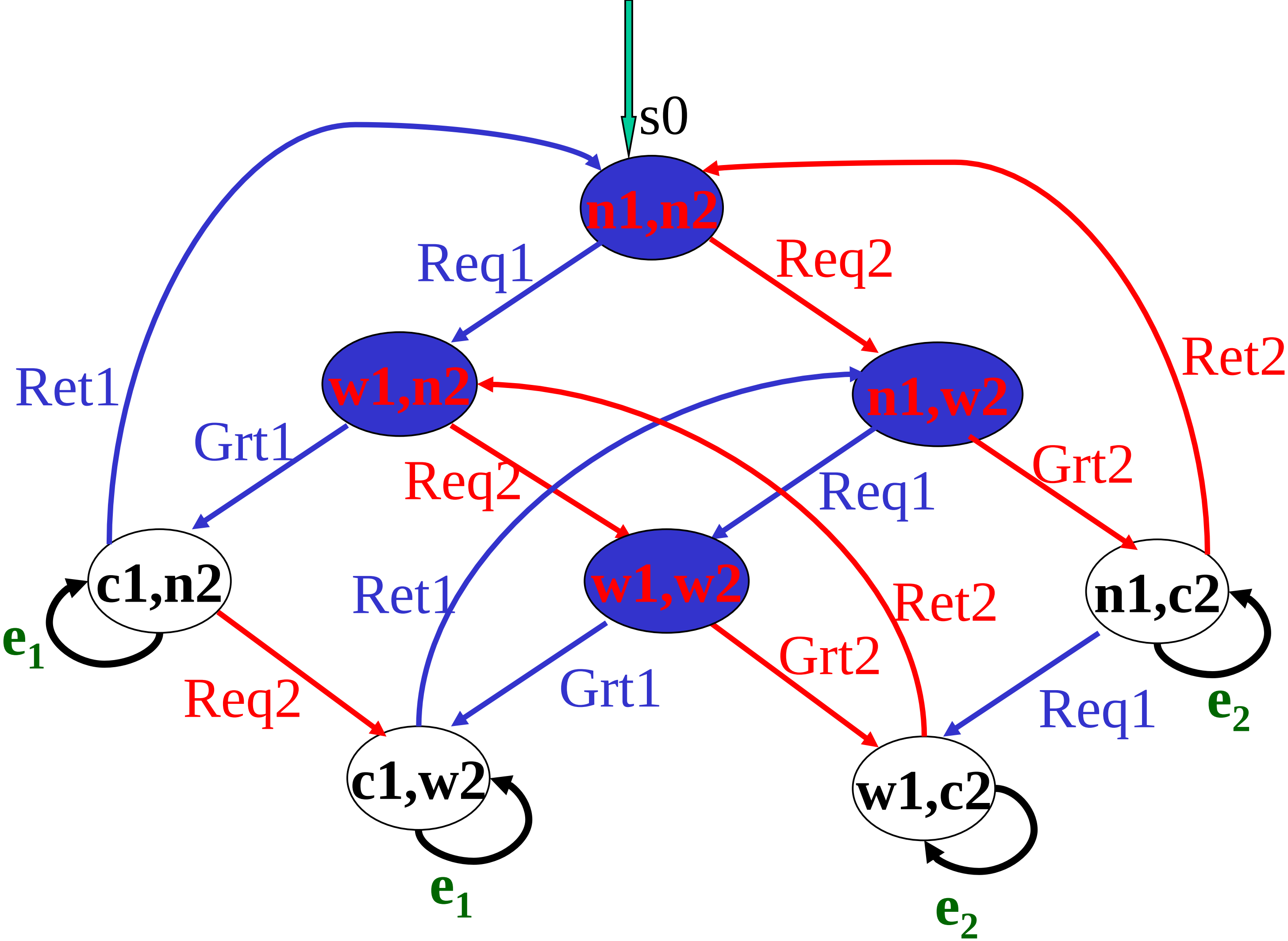
K, s0 ⊨ AG(req2 → AF grt2) with above
fairness constraint !



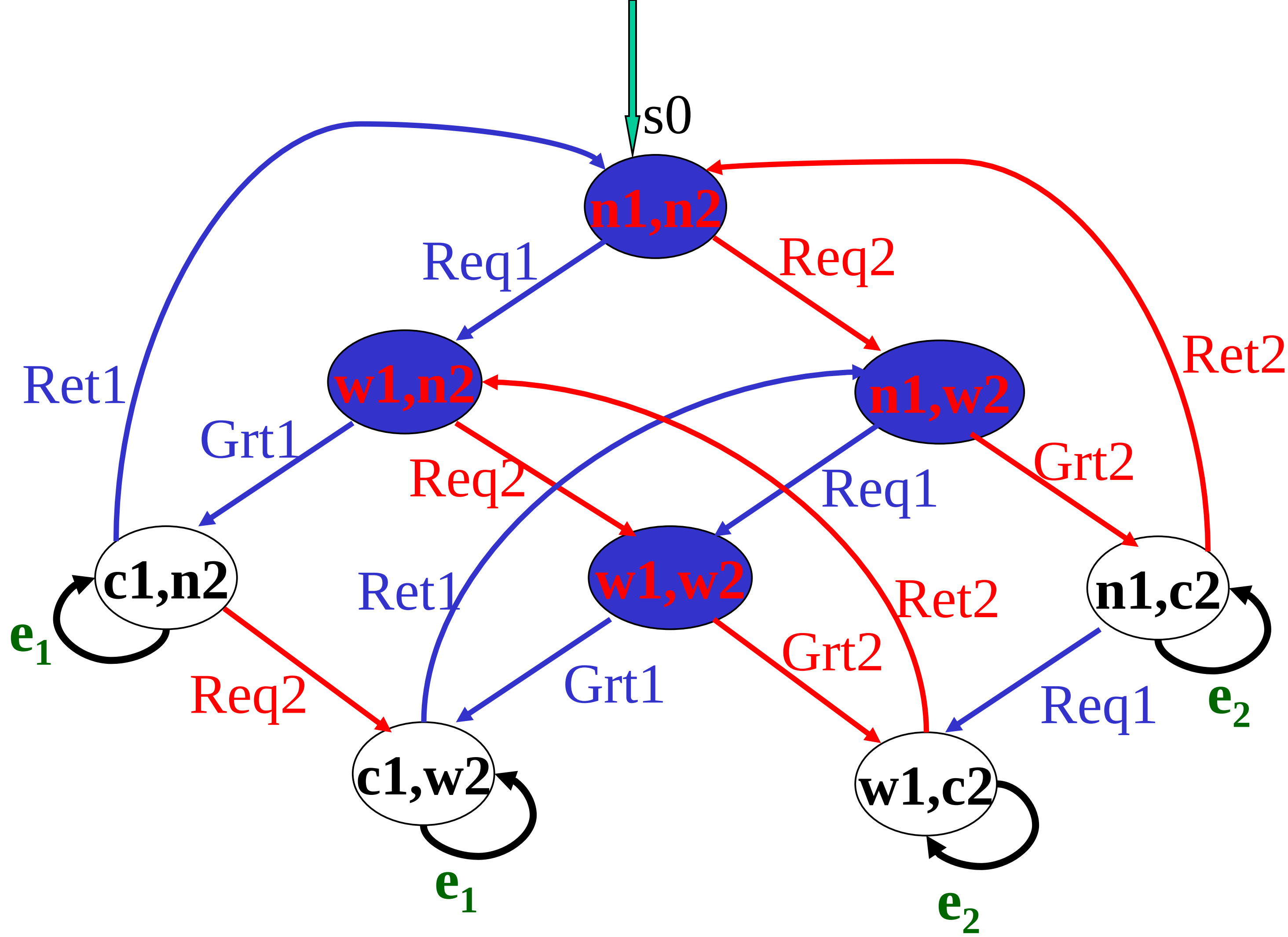
K, s0 ⊨ AG(req2 → AF grt2)

F ---- ¬req2 ∨ grt2

(notice that **s1**,**s2**,**s3** satisfy **req2 and g1**,**g2 satisfy grt2)**



K, s0 $\nvDash$ EF (EG c1 $\vee$ EG c2) with the above ***fairness constraint*** **!**



K, s0 $\not\models$ EF (EG c1 $\vee$ EG c2) with the above ***fairness constraint*** **!**

F ---- $\neg$c1 $\wedge$ $\neg$c2

NuSMV Fairness

- Can't always use sets of states to specify fairness.
  - State space is often defined implicitly.
- Use formulas!
- ϕ ---- Property ϕ is true ***infinitely often***.
- ***Model check*** along only ***fair computation paths***.

Model Checking CTL with Fairness

- $\mathcal{C} = \{P_1, P_2, \ldots, P_n\}$
  - Fairness constraints.
- **K = (S, S_0, R, AP, L, $\mathcal{C}$)**
- **s0 s1 s2 …..** is a ***fair computation*** ***iff***:
  - ***It is a computation.***
  - ***For each i, there are infinitely many j such that***

$$K, s_j \models P_i$$

Model Checking with Fairness.

- $\mathcal{C} = \{P_1, P_2, \ldots, P_n\}$
  - Fairness constraints.
- $K = (S, S_0, R, AP, L, \mathcal{C})$

$K, s \vDash_{\mathcal{C}} \psi$ is defined as follows:

- $K, s \vDash_{\mathcal{C}} p$ ***iff*** $p \in L(s)$
- $K, s \vDash_{\mathcal{C}} \neg\psi$ ***iff*** $K, s \nvDash_{\mathcal{C}} \psi$
- $K, s \vDash_{\mathcal{C}} \psi_1 \wedge \psi_2$ ***iff*** $K, s \vDash_{\mathcal{C}} \psi_1$ and $K, s \vDash_{\mathcal{C}} \psi_2$

Model Checking with Fairness.

- $K, s \models_{\mathcal{C}} EX\psi$ ***iff*** there exists a ***fair path*** from **s** and there exists **s'** along that path with **R(s, s')** and $K, s' \models_{\mathcal{C}} \psi$.
- $K, s \models_{\mathcal{C}} EU(\psi_1, \psi_2)$ ***iff*** there exists a ***fair path*** from **s** which satisfies ψ_2 at some state and ψ_1 at all previous states.
- $K, s \models_{\mathcal{C}} EG\psi$ ***iff*** there exists a ***fair path*** from **s** which satisfies ψ at every state along this fair path.

Model Checking with Fairness.

- $\mathcal{C} = \{P_1, P_2, \ldots, P_n\}$
  - Fairness constraints.
- **K = (S, S_0, R, AP, L, $\mathcal{C}$)**
- It is possible to adapt the **NuSMV** model checking procedure for the problem
  - $K, s \models \psi$

    to the problem
  - $K, s \models_{\mathcal{C}} \psi$.

Fair Strongly Connected Comp.

A non-trivial strongly connected component C of* K *is fair with respect to the fair set* $\mathcal{C} = \{P_1, P_2, \ldots, P_n\}$ *iff for each* $P_i \in \mathcal{C}$ *there is a state* $s \in C$ *such that

$$K, s \models P_i$$

M. C. with Fairness: EG(β)

Let **K' = (S',R',L', $\mathcal{C}$)** be the sub-graph of **K** where

- **S' = { s | K, s $\models_{\mathcal{C}}$ β }**
- **R' = R$|_{S' \times S'}$** (the restriction of **R** to **S'**)
- **L' = L$|_{S'}$** (the restriction of **L** to **S'**)

Lemma: K, s $\models_{\mathcal{C}}$ EG(β) ***iff***

1. **s ∈ S' = { s' | K, s' $\models_{\mathcal{C}}$ β } and**

2. ***there exists a path*** in **K'** leading from **s** to a ***non-trivial fair strongly connected component*** **C** of the graph **(S',R')** ***w.r.t.*** $\mathcal{C}$.

Computing the labeling for EG(β)

Complexity: $O(|K||\mathcal{C}|)$

Algorithm Check_Fair_EG(β)

$S' := \{s \mid \beta \in Labels_{\mathcal{C}}(s)\}$;

$SCC := \{X \mid X$ is a ***fair*** non trivial SCC of $S'\}$;

$T := \bigcup_{X \in SCC} \{s \mid s \in X\}$;

for each $s \in T$ do $Labels_{\mathcal{C}}(s) := Labels_{\mathcal{C}}(s) \cup \{EG(\beta)\}$;

while $T \neq \emptyset$ do

chose $s \in T$;

$T := T \setminus \{s\}$;

for each $t \in S'$ with $t \rightarrow s$ do

if $EG(\beta) \notin Lables_{\mathcal{C}}(t)$ then

$Labels_{\mathcal{C}}(t) := Labels_{\mathcal{C}}(t) \cup \{EG(\beta)\}$;

$T := T \cup \{t\}$;

The Labels function

Let *fair* be a new *atomic proposition* and let us use the algorithm Check_Fair_EG(*true*) to label *K* with this new proposition (i.e. *fair* = *EG true* where *true* $\in$ Labels$_{\mathcal{C}}$(s), for all s)

Then

- $K, s \models_{\mathcal{C}} p$ **iff** $K, s \models p$
- $K, s \models_{\mathcal{C}} \neg\phi$ **iff** $K, s \not\models_{\mathcal{C}} \phi$
- $K, s \models_{\mathcal{C}} EX\phi$ **iff** $K, s \models EX(\phi \wedge \mathit{fair})$
- $K, s \models_{\mathcal{C}} EU(\psi, \phi)$ **iff** $K, s \models EU(\psi, \phi \wedge \mathit{fair})$

Symbolic MC for $EG_f \phi$

Let us start by noting that

$$\mathbf{EG}\ \phi \equiv \phi \wedge \mathbf{EX}\ \mathbf{EG}\ \phi \equiv \phi \wedge \mathbf{EX}\ \mathbf{EU}\ (\phi, \mathbf{EG}\ \phi)$$

Therefore

$$\mathbf{EG}\ \phi = \nu \mathbf{Z}.\ \phi \wedge \mathbf{EX}\ \mathbf{EU}(\phi, \mathbf{Z})$$

The fixpoint **Z** is then the ***largest set*** of states with the following two properties:

1. all the states in **Z** satisfy ϕ, and
2. for all states $\mathbf{s} \in \mathbf{Z}$
   - there is a ***non-empty*** sequence of states (a ***path***) from **s** ***leading*** to a state in **Z**, and
   - all states in this sequence ***satisfy*** the formula ϕ.

Symbolic MC for $EG_f\ \phi$

Let us generalize the previous result, and consider **Z** the ***largest set*** of states with the following two properties:

1. all the states in **Z** satisfy ϕ, and
2. for all $P_k \in \mathcal{C}$ and all states $s \in Z$
   - there is a ***non-empty*** sequence of states (a ***path***) from **s** leading to a state in **Z** satisfying P_k, and
   - all states in this sequence ***satisfy*** the formula ϕ.

It can be shown that:

- each state in **Z** is the beginning of a path along which ϕ is ***always true***, and
- every formula in $\mathcal{C}$ holds ***infinitely often*** along this path.

Symbolic MC for $EG_f\ \phi$

It follows that $\mathbf{EG_f}\ \phi$ can be expressed as a greatest fixed point of the following function:

$$\mathbf{EG_f}\ \phi = \nu \mathbf{Z}.\ \phi \wedge \bigwedge_{k=1\ldots n} \mathbf{EX}\ \mathbf{EU}(\phi, \mathbf{Z} \wedge \mathbf{P_k})$$

This equation can be used to compute the set of states that satisfy $\mathbf{EG_f}\ \phi$ according to the ***fair semantics***.

Symbolic MC for $EX_f\ \phi$ and $EU_f(\phi,\psi)$

All other temporal operators can be computed by combining $\mathbf{EG_f}$ and the standard semantics of ***non-fair*** operators.

Let us define the ***set of all states*** that are the starting state of some ***fair computation*** as the set of states satisfying a new proposition ***fair*** such that:

$$\textbf{\textit{fair}} = \mathbf{EG_f}\ \textbf{\textit{true}}$$

Hence,

$$\mathbf{EX_f}\ \phi = \mathbf{EX}(\phi \wedge \textbf{\textit{fair}});$$

$$\mathbf{EU_f}(\phi, \psi) = \mathbf{EU}(\phi, \psi \wedge \textbf{\textit{fair}})$$

Counter-example/Witness Generation

- A formula with a ***universal path quantifier*** has a counter-example consisting of one trace (path)
- A formula with an ***existential path quantifier*** has a witness consisting of one trace
- Due to the dualities in **CTL**, we only have to consider witnesses for existential formulae. That is:
  - a two states trace witnessing **EX** ϕ (this is trivial)
  - a finite trace π witnessing **EU**(ϕ,ψ)
  - an infinite trace π witnessing **EG** ϕ
  - for finite systems, the latter must be a ***lasso***, that is π is a path consisting of a (finite) prefix σ and a (finite) loop ρ, such that $\pi = \sigma\rho^{\omega}$
- For ***fair counter examples*** we need that the loop which contains a state ***from each fairness constraint***.

Witness for EU(ϕ,ψ)

Recall that:

$$\mathbf{EU}(\phi,\psi) = \mu \mathbf{Q}.\ \psi \vee (\phi \wedge \mathbf{EX}\ \mathbf{Q})$$

Unfolding the recursion, we get:

$$\mathbf{Q_0} = \mathit{False}$$
$$\mathbf{Q_1} = \psi \vee (\phi \wedge \mathbf{EX}\ \mathit{False}) = \psi$$
$$\mathbf{Q_2} = \psi \vee (\phi \wedge \mathbf{EX}\ \psi)$$
$$\mathbf{Q_3} = \psi \vee (\phi \wedge \mathbf{EX}\ (\psi \vee (\phi \wedge \mathbf{EX}\ \psi)))$$

- The fixed point computation follows a process of backward reachability.
- Each $\mathbf{Q_i}$ contains the states that can reach ψ in at most $\boldsymbol{i}$-1 steps (transitions), while ϕ holds in between.
- We can generate a witness (path) by performing a forward reachability within the sequence of $\mathbf{Q_i}$'s.

Witness for EU(ϕ,ψ)

- Assume the initial state $\mathbf{s_0} \vDash \mathbf{EU(\phi,\psi)}$
- To find a minimal witness from state s_0, we start in the smallest $\boldsymbol{n}$ such that $s_0 \in \mathbf{Q}_{\boldsymbol{n}}$.
- The desired witness is a path of the form

$$\pi = s_0 \rightarrow s_1 \rightarrow \cdots \rightarrow s_n$$

such that $s_i \in \mathbf{Q}_{\boldsymbol{n-i}} \cap R(s_{i-1})$ and $s_n \in \mathbf{Q_1} = \psi$ (where $R(s_{i-1})$ denotes the set $\{s \mid R(s_{i-1},s)\}$)

- Notice that this path is guaranteed to exist since $s_0 \in \mathbf{Q}_{\boldsymbol{n}}$, $\mathbf{Q}_{\boldsymbol{n-i}}$ contains states reachable in one step from some state in $\mathbf{Q}_{\boldsymbol{n-i+1}}$, and each such state satisfies ϕ.
- Then π is a path (i.e. $(s_i,s_{i+1}) \in R$ for $0 \le i \le \boldsymbol{n}$-1) such that $s_n \vDash \psi$ and $s_i \vDash \phi$, for each $0 \le i < \boldsymbol{n}$.

Witness for EU(ϕ,ψ)

This can easily be implemented symbolically using BDDs as follows:

- Given s_0 the BDD representation of state s_0.
- For $i \in \{1,\ldots,\boldsymbol{n}\}$, we can ***pick*** any state s_i as any assignment which makes true the following function:

$$\mathbf{Q}_{n-i}(\boldsymbol{v'}) \wedge R(s_{i-1},\boldsymbol{v'})$$

($\boldsymbol{v'}$ denotes the vector of primed vars and s_{i-1} the assignment to the current vars for state s_{i-1})

- Any s_i is the BDD representation of a state s_i that:
  - can reach ψ (with ϕ true in between) in at most $\boldsymbol{n-i}$ steps and
  - is a successor of a state s_{i-1} that can reach ψ (with ϕ true in between) in at most $\boldsymbol{n-i+1}$ steps ..., and so on.

Witness for $EG_f\ \phi$

- We want an path from an intial state s_0 to a cycle on which each fairness constraint P_1, P_2 , ... , P_n occurs.

$$\mathbf{EG_f}\ \phi = \nu \mathbf{Z}.\ \phi \wedge \bigwedge_{k=1\ldots n} \mathbf{EX}\ \mathbf{EU}(\phi, \mathbf{Z} \wedge \mathbf{P_k})$$

- Unfolding the recursion we obtain:

$$Z_0 = \mathit{True}$$

$$Z_1 = \phi \wedge \bigwedge_{k=1\ldots n} \mathbf{EX}\ \mathbf{EU}(\phi, \mathit{True} \wedge \mathbf{P_k})$$

$$\ldots$$

$$Z_m = \phi \wedge \bigwedge_{k=1\ldots n} \mathbf{EX}\ \mathbf{EU}(\phi, Z_{m-1} \wedge \mathbf{P_k})$$

- Let $\check{\mathbf{Z}} = Z_m = Z_{m-1} = EG_{\mathbf{f}}\ \phi$ be the fixpoint.

Witness for $EG_f\ \phi$

- Let $\check{Z} = Z_m = Z_{m-1} = EG_f\ \phi$ be the fixpoint.
- While computing $\check{Z}$ in the last iteration, it was also computed, for each $k \in \{1,\ldots,n\}$, the set of states satisfying $\mathbf{EU}(\phi, \check{Z} \wedge P_k)$.
- This amounts to computing, for each $k \in \{1,\ldots,n\}$, the following sequence of sets, using backward reachability:

$$Q^k{}_0 \subseteq Q^k{}_1 \subseteq Q^k{}_2 \subseteq \ldots \subseteq Q^k{}_{j_k}$$

  – where each $Q^k{}_i$ is an (under) approximation of the set of states satisfying $\mathbf{EU}(\phi, \check{Z} \wedge P_k)$

  – and each state in $Q^k{}_i$ can reach $\check{Z} \wedge P_k$ with no more than i steps (transitions).

Witness for $EG_f\ \phi$

Let the sequences of approximantions

$$Q^k_0 \subseteq Q^k_1 \subseteq Q^k_2 \subseteq \ldots \subseteq Q^k_{j_k}$$

be given for each $k \in \{1,\ldots,n\}$ (we can save them during the last iteration of the outer fixpoint of $EG_f\ \phi$)

- Assume now that the initial state $\mathbf{s_0 \models EG_f\ \phi}$
- We can first construct a path

$$s_0 \longrightarrow^* s_1 \longrightarrow^* \cdots \longrightarrow^* s_n$$

(where $\longrightarrow^*$ is the transitive closure of R), such that:

  – the formula ϕ holds invariantly, and

  – for each $k \in \{1,\ldots,n\}$, $s_k \in \mathbf{Ž} \wedge \mathbf{P_k}$

- The path above is then guaraneed to exist and to pass through each fairness constraint, while holding ϕ true.

Witness for $EG_f\ \phi$

To build the path we start setting $k=1$ and then:

1. determine the minimal z such that s_{k-1} has a successor $t^k_0 \in Q^k_z$
2. using the witness procedure for **EU**, construct a witness for $\mathbf{EU}(\phi, \check{\mathbf{Z}} \wedge \mathbf{P_k})$, namely a path of the form:

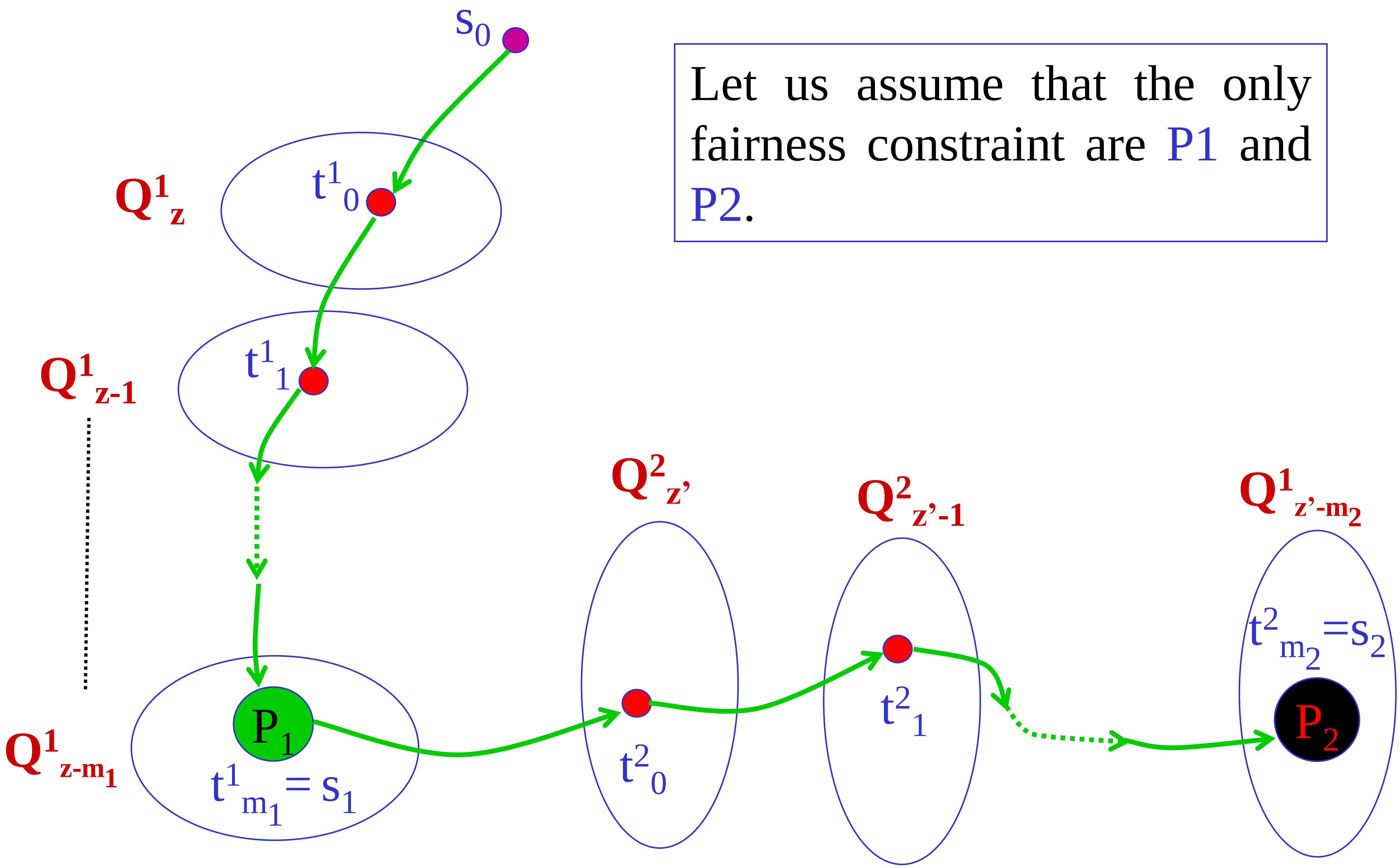
$$s_{k-1} \rightarrow t^k_0 \rightarrow t^k_1 \cdots \rightarrow t^k_{m_k} \in \check{\mathbf{Z}} \wedge \mathbf{P_k}$$

3. finally set $s_k = t^k_{m_k}$ and proceed to build the path for $\mathbf{P_{k+1}}$ going back to step 1 (until $k = n$).

Notice that, each t^k_j (with $j \geq 1$) will be found in Q^k_{z-j}, and will satisfy ϕ.

Building a fair path from s_0

Let us assume that the only fairness constraint are P1 and P2.



Witness for $EG_f\ \phi$

Once we have generated the path

$$s_0 \longrightarrow^* s_1 \longrightarrow^* \cdots \longrightarrow^* s_n$$

we need to check if s_n can reach (non trivially) s_1 while holding ϕ true, i.e. check whether

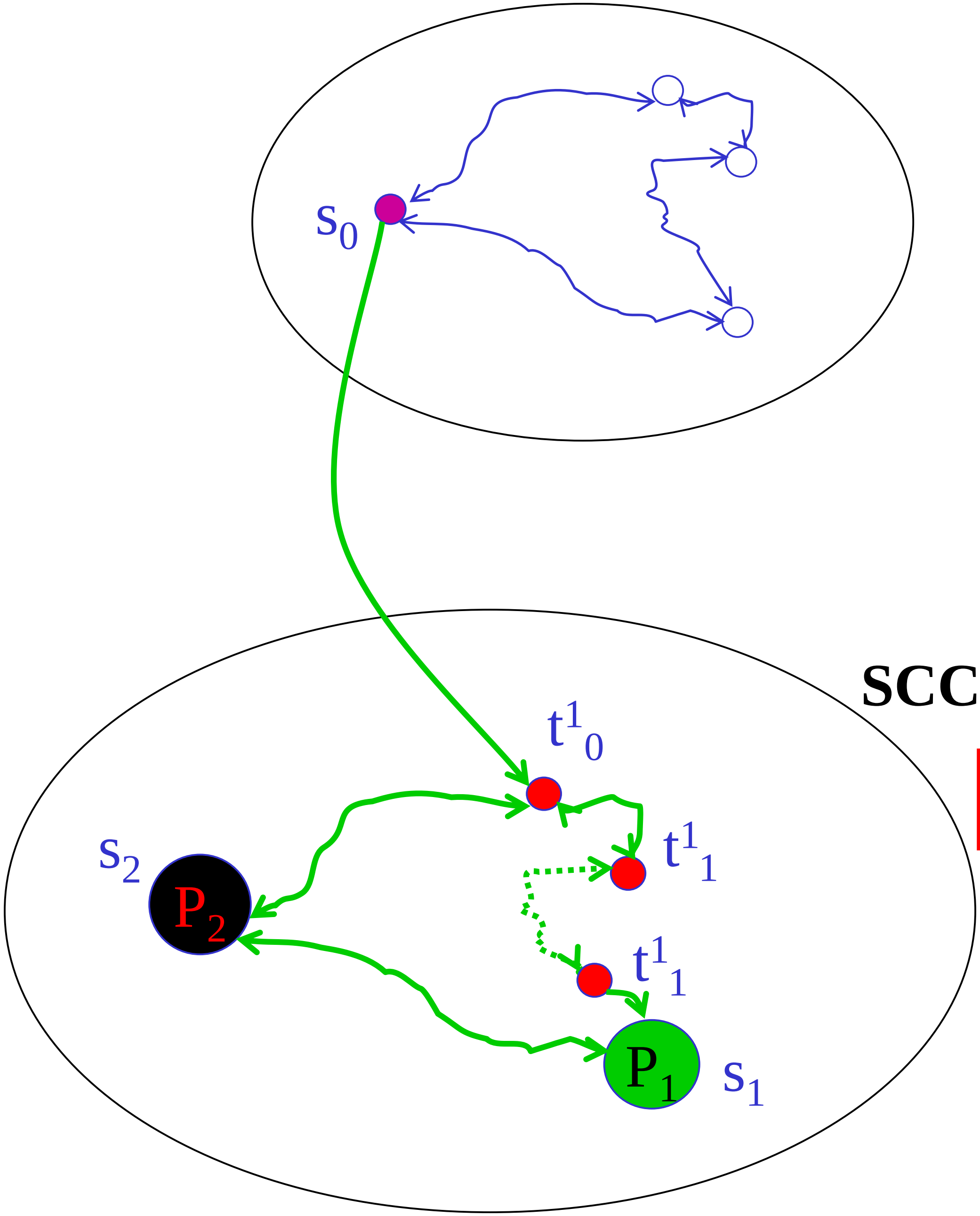
$$s_n \in \mathbf{EX}\ \mathbf{EU}(\phi, \{s_1\})$$

If this is the case, then we have found a (non trivial) cycle from s_1 back to s_1 passing through all the fairness constraints and which invariantly satisfies ϕ.

This means that $s_1, s_2, \ldots, s_n$ all belong to the same **SCC** satisfying ϕ and reachable from s_0.

Therefore, the prefix going from s_0 to s_1 (σ) in $s_0 \longrightarrow^* s_1$ concatenated with the cycle from s_1 to s_1 (ρ^ω) forms the desired witness $\pi = \sigma\rho^\omega$.

Witness contained in the first SCC



Let us assume that the only fairness constraint are P1 and P2.

$s_2 \models$ **EX EU**$(\phi, \{s_1\})$

Witness for $EG_f\ \phi$

If, in the other hand,

$$s_n \notin \mathbf{EX}\ \mathbf{EU}(\phi, \{s_1\})$$

then s_1 and s_n do not belong to the same **SCC** and the cycle cannot be closed.

This means that $s_1, s_2 \ldots, s_n$ belong to the prefix $\boldsymbol{\sigma}$ of the desired witness $\boldsymbol{\pi}$.

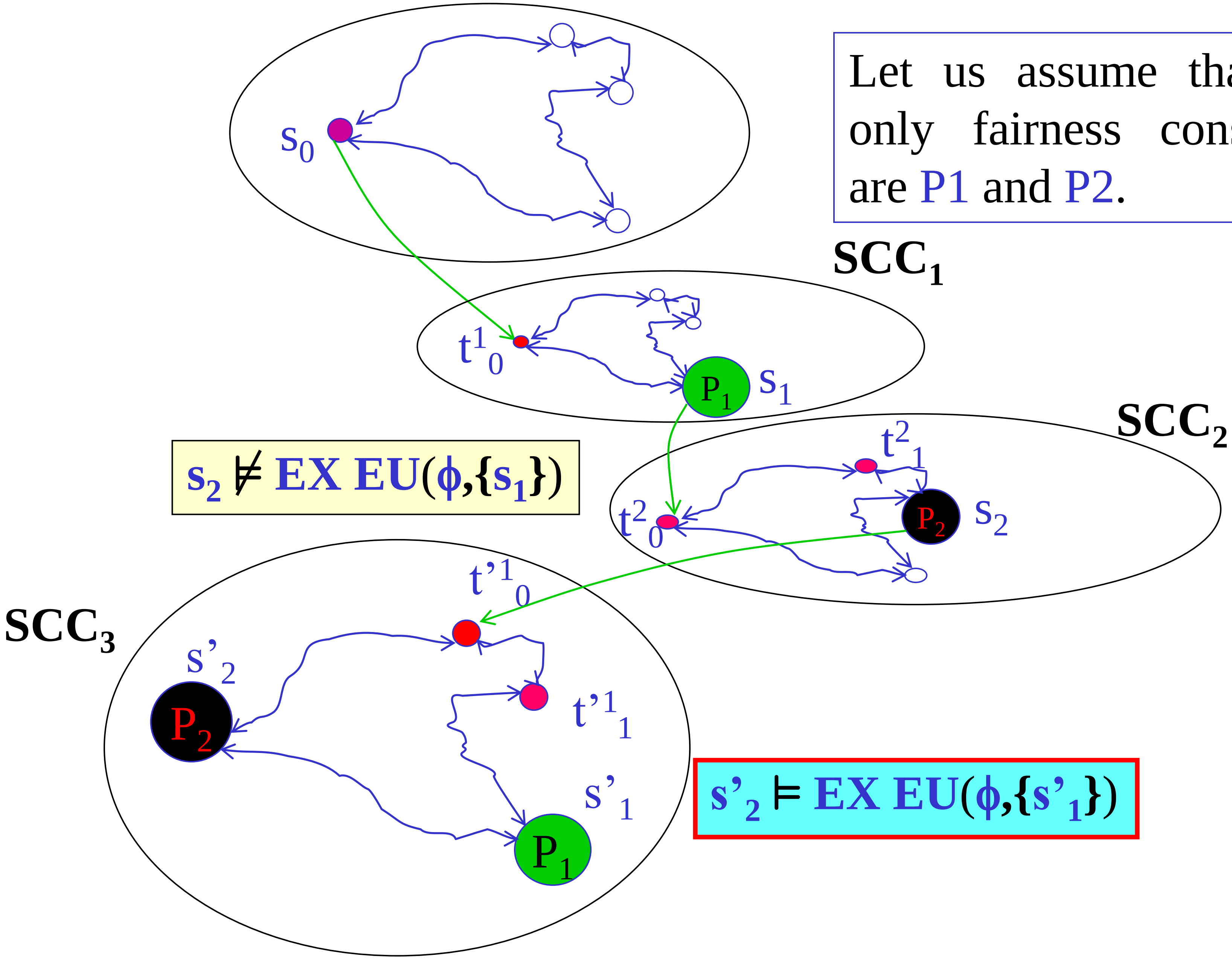
In this case, we can restart the process starting from s_n as we have already done from s_0, building another seuqence

$$s_n \rightarrow^{*} s'_1 \rightarrow^{*} \cdots \rightarrow^{*} s'_n$$

passing through all the fairness constraints and then check if $s'_n \in \mathbf{EX}\ \mathbf{EU}(\phi, \{s'_1\})$, i.e. another **SCC**.

Witness over multiple SCCs

Let us assume that the only fairness constraint are P1 and P2.



$\mathbf{s_2 \not\models EX\ EU(\phi,\{s_1\})}$

$\mathbf{s'_2 \models EX\ EU(\phi,\{s'_1\})}$

Witness for $EG_f\ \phi$

The process above must terminate since:

1. the Kripke structure is finite, therefore so is also the number of **SCC**s.
2. the algorithm, while looking for the fair cycle, essentially moves from one **SCC** to another within the graph of th **SCC**s, following non trivial paths.
3. the ***graph of the*** **SCC**s is always acyclic.

Therefore, if the witness $\pi = \sigma\rho^\omega$ is not found earlier, then ρ^ω must be contained in some ***terminal*** **SCC**, i.e. one which has no outgoing arc to some other **SCC**.

The graph of the SCCs