Probabilistic CTL

David N. Jansen

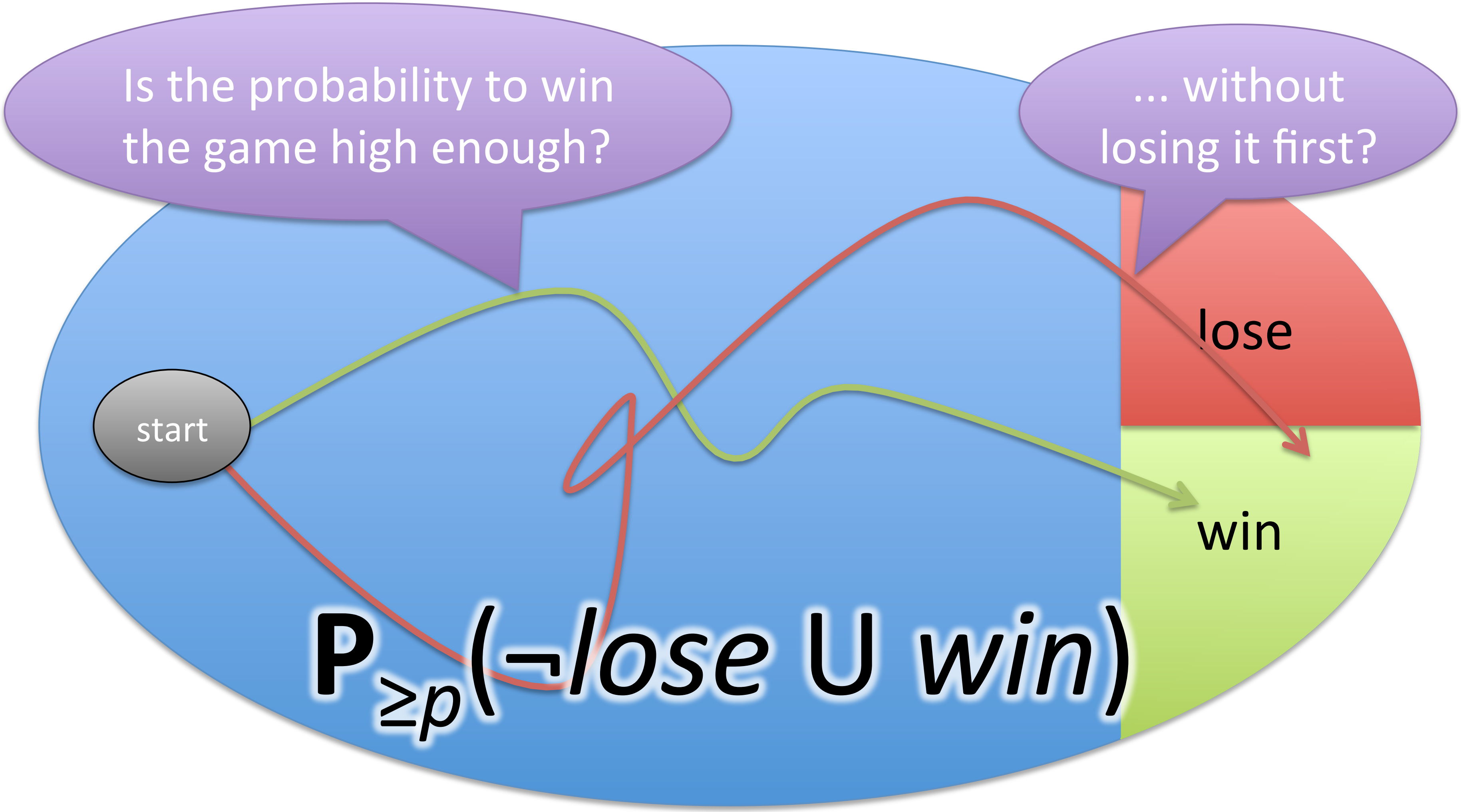
Recapitulation: DTMCs

- Markov chains describe
  the behaviour of discrete-state systems.
- Discrete-time Markov chains
  count the number of steps,
  but not how long a step takes.
- Transient state and steady-state analysis
  serve to calculate state probabilities.

Probabilistic CTL

- a logic to describe properties
  of Markov chains
- extends CTL
- strictly speaking, not a quantitative logic
  (truth values are Boolean: true or false)

Constrained Probabilistic Reachability



PCTL syntax

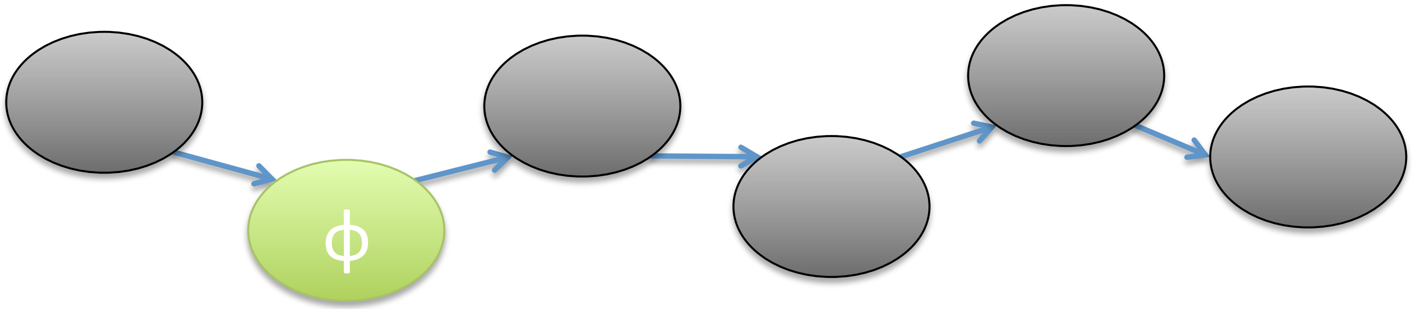
- state formulas ϕ, ψ
  - a — atomic proposition
  - $\neg\phi$ — negation
  - $\phi \vee \psi$ — disjunction
  - $\mathbf{P}_{<p}(\Pi)$, $\mathbf{P}_{\leq p}(\Pi)$, $\mathbf{P}_{\geq p}(\Pi)$, $\mathbf{P}_{>p}(\Pi)$ — probability constraint

$p \in [0,1]$

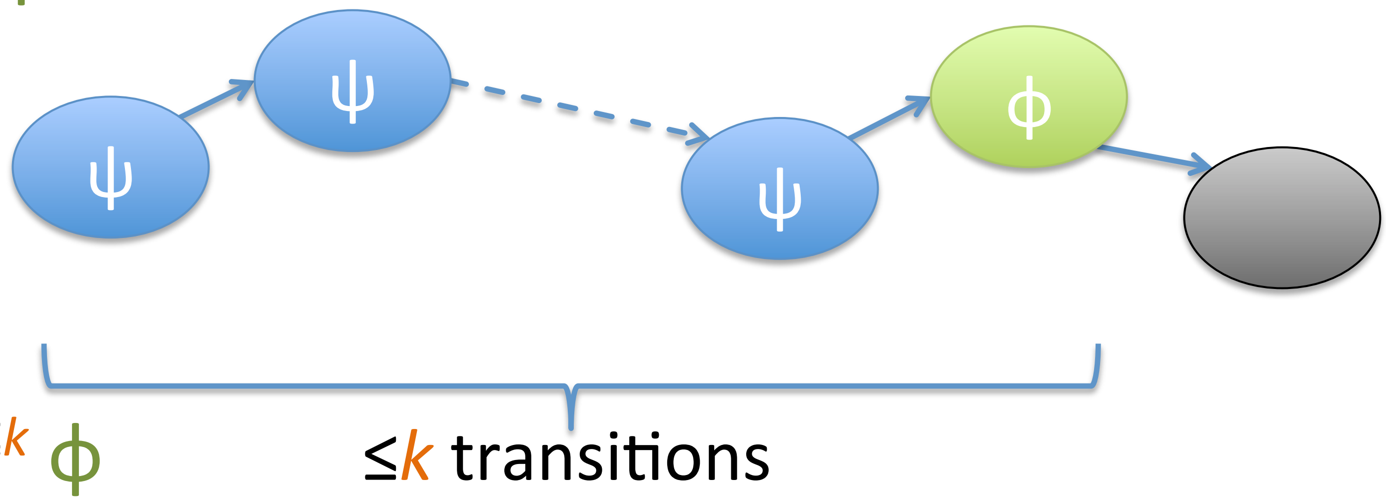
- path formulas Π
  - $X\ \phi$ — next state
  - $\psi\ U\ \phi$ — unbounded until
  - $\psi\ U^{\leq k}\ \phi$ — bounded until

Path formulas

- X ϕ



- ψ U ϕ



- $\psi\ U^{\leq k}\ \phi$

Example formulas...

- simple communication protocol:
  sender sends a message;
  it gets through with 90% probability;
  otherwise it is resent until success.
  - Model this Markov chain.
  - Formulate a useful PCTL property for this MC.
- Craps:
  - Formulate a useful PCTL property for Craps.

PCTL semantics

- Formal definition:
- When does a state satisfy a state formula?
- When does a path satisfy a path formula?

Formal definition: labelled DTMC

- A Markov chain consists of:
  - S — finite set of states (often $S = \{1, 2, \ldots n\}$)
  - $\mathbf{P}: S \times S \rightarrow [0,1]$ — transition probability matrix (with row sums = 1)
  - $\pi_0: S \rightarrow [0,1]$ — initial state distribution (sometimes)
  - $L: S \rightarrow 2^{AP}$ — labelling with atomic propositions

PCTL semantics: state formulas

Sat(ϕ) = set of states that satisfy ϕ:

- Sat(a) = $\{s \mid a \in L(s)\}$
- Sat($\neg\phi$) = $S \setminus$ Sat(ϕ)
- Sat($\phi \vee \psi$) = Sat(ϕ) $\cup$ Sat(ψ)

- Sat($\mathbf{P}_{\geq p}(\Pi)$) = $\{s \mid \text{Prob}_s\ \text{Sat}(\Pi) \geq p\}$ etc.

- Prob_s Sat(Π) = probability to get a Π-path, if you start in s

PCTL semantics: path formulas

Sat(Π) = set of paths that satisfy Π:

- Sat(X ϕ) = $\{\sigma \mid \sigma[1] \in \text{Sat}(\phi)\}$
- Sat(ψ U ϕ) = $\{\sigma \mid \exists\, i \geq 0, \sigma[i] \in \text{Sat}(\phi)$ and $\forall\, j < i, \sigma[j] \in \text{Sat}(\psi)\}$
- Sat(ψ $U^{\leq k}$ ϕ) = $\{\sigma \mid \exists\, i \geq 0, i \leq k$ and $\sigma[i] \in \text{Sat}(\phi)$ and $\forall\, j < i, \sigma[j] \in \text{Sat}(\psi)\}$

Measurability

- For every PCTL path formula,
  the set of paths satisfying the formula
  is measurable (i. e. is in the σ-algebra).

- Proof: one can construct the set
  as a (disjoint) union of cylinder sets...
  (details on board)

PCTL model checking

- construct satisfaction sets bottom-up
  (as with CTL model checking)

- new step: probabilistic operator $\mathbf{P}_{\geq p}(\Pi)$
  → constrained probabilistic reachability

Example...

- Craps, formula $\mathbf{P}_{>0.3}(\neg \mathit{state}_4 \ U^{\leq 6} \ \mathit{win})$
- “Is the probability to win with up to 6 rolls > 0.3, but without the first roll being 4?”

- Need to calculate the probability of Sat($\neg \mathit{state}_4 \ U^{\leq 6} \ \mathit{win}$) = union of cylinder sets... (on the blackboard)

How to calculate the probability of Sat($\neg$*lose* $U^{\leq k}$ *win*)

- Make all states in Sat(*win*) absorbing.
  Reason: All paths reaching a *win*-state satisfy the formula.
- In the modified MC,
  Sat($\neg$*lose* $U^{\leq k}$ *win*) = Sat($\neg$*lose* $U^{=k}$ *win*)

- Make all states in Sat(*lose*) absorbing.
  Reason: All paths reaching a *lose*-state falsify the formula.
- In the modified MC,
  Sat($\neg$*lose* $U^{=k}$ *win*) = Sat(*true* $U^{=k}$ *win*)

How to calculate the probability of Sat($\color{red}{true}$ $U^{=k}$ *win*)

- Sat(*true* $U^{=k}$ *win*) = set of paths that reach *win* in *k* steps
- Transient probability for m… chain: Pr(Sat(*true* $U^{=k}$ *win*)) = $\pi_k = \pi_0 \cdot P^k$

  $\pi_k(s)$ = probability to be in state *s* after *k* steps
- sum over all *win*-states: $\Sigma_{s \in \mathrm{Sat}(win)} \pi_k(s)$
- probability = $\pi_0 \cdot P^k \cdot \mathbf{1}_{win}$

  vector: 1 in *win*-states, 0 in other states
- calculate this for all initial states

  $\pi_0 = (1, 0, 0, \ldots)$ $\pi_0 = (0, 1, 0, 0, \ldots)$ etc.

  → vector $P^k \cdot \mathbf{1}_{win}$

How to calculate $\mathrm{Sat}(\mathbf{P}_{\geq p}(\neg \mathit{lose}\ \mathrm{U}^{\leq k}\ \mathit{win}))$

- modify Markov chain:
  make all *win*- and all *lose*-states absorbing
- calculate $\pi_{win} = P^{k} \cdot \mathbf{1}_{win}$
- For every state *s*:
  If $\pi_{win}(s) \geq p$, then $s \in \mathrm{Sat}(P_{\geq p}(\neg \mathit{lose}\ \mathrm{U}^{\leq k}\ \mathit{win}))$

$\pi_{win}(s)$ = probability that $(\neg \mathit{lose}\ \mathrm{U}^{\leq k}\ \mathit{win})$, if starting in *s*

- Advantage: calculate for each state *s* at once!

Example...

- calculate the probabilities for the Craps example and $\mathbf{P}_{>0.3}(\neg(\mathit{state}\ 4)\ \mathrm{U}^{\leq 6}\ \mathit{win})$

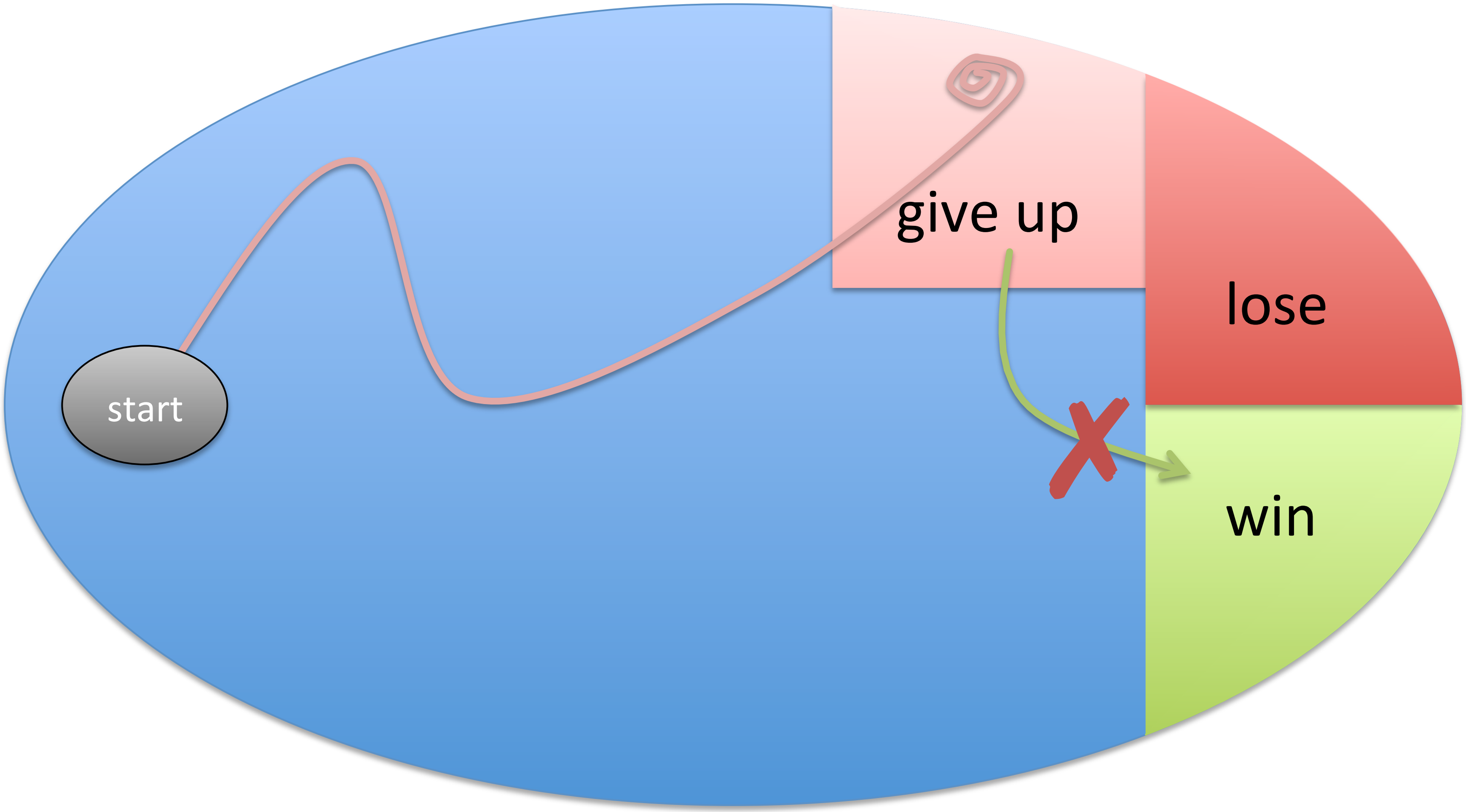
- Be careful: “lose” gets a new meaning here.

Optimization

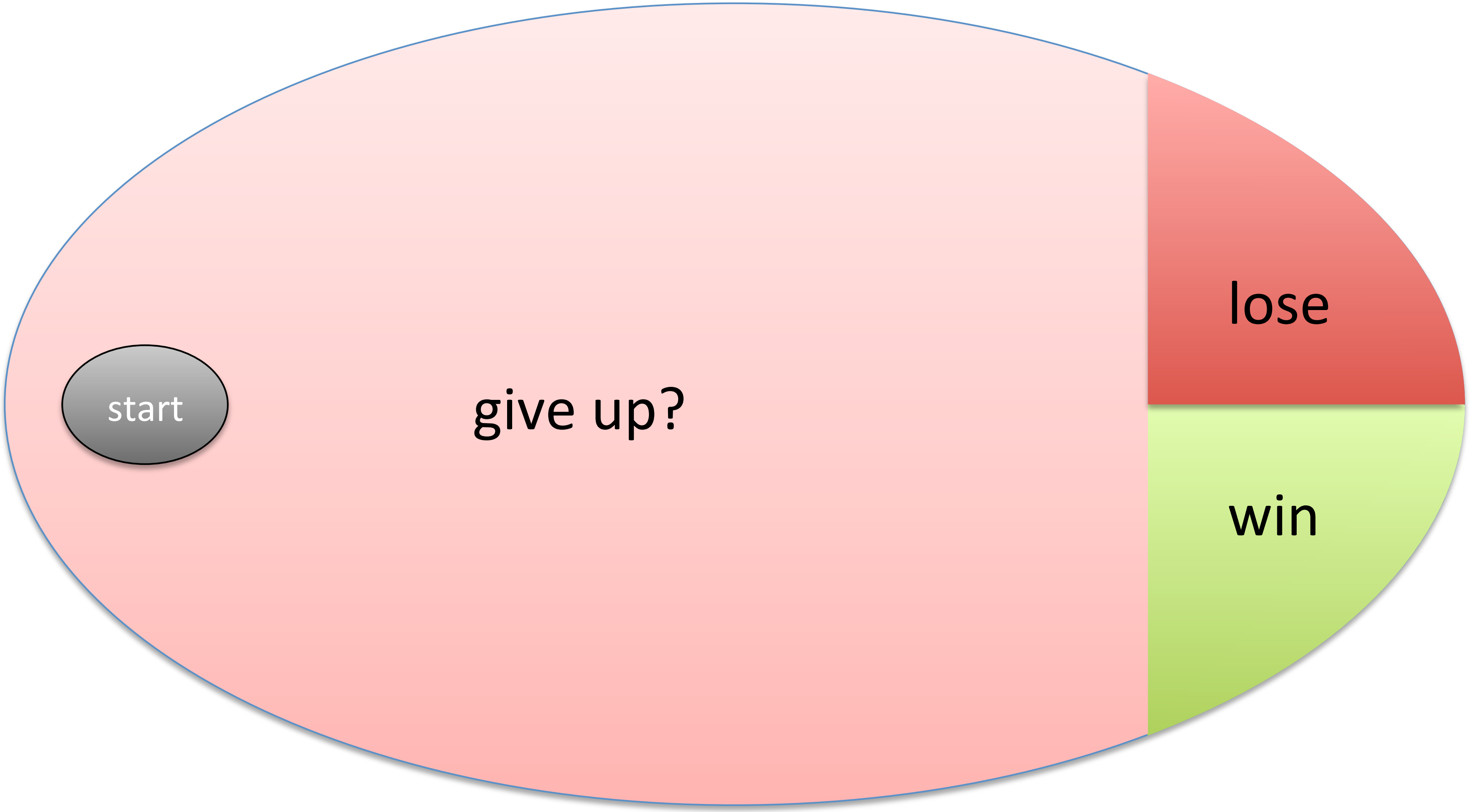
- There may be stalemate states
  (Poland is not yet lost, but it will never win)
- In these states, probability to reach *win* is = 0
- Finding stalemate states is simpler
  than matrix calculations

- Therefore:
  also make stalemate states absorbing.

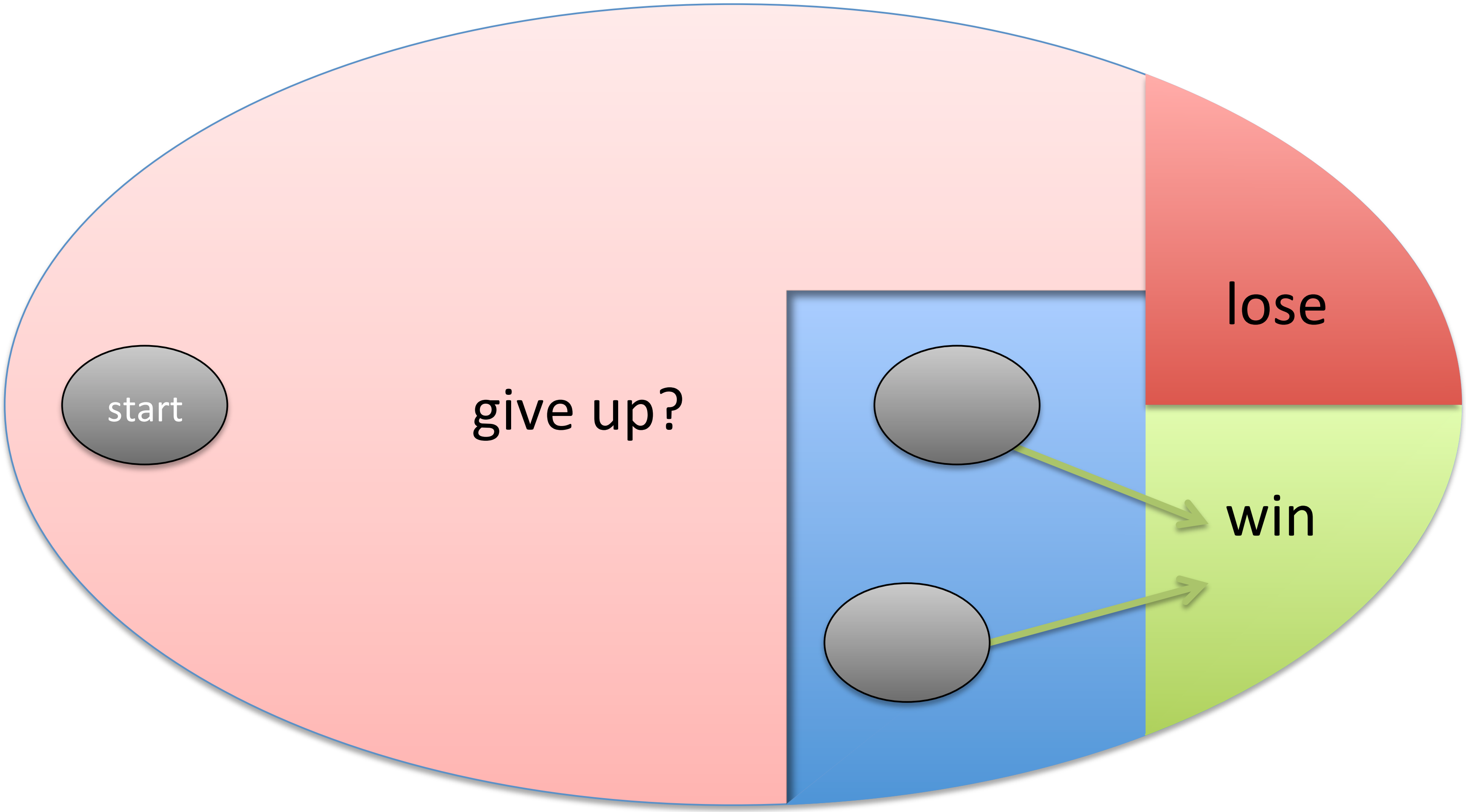
Stalemate states



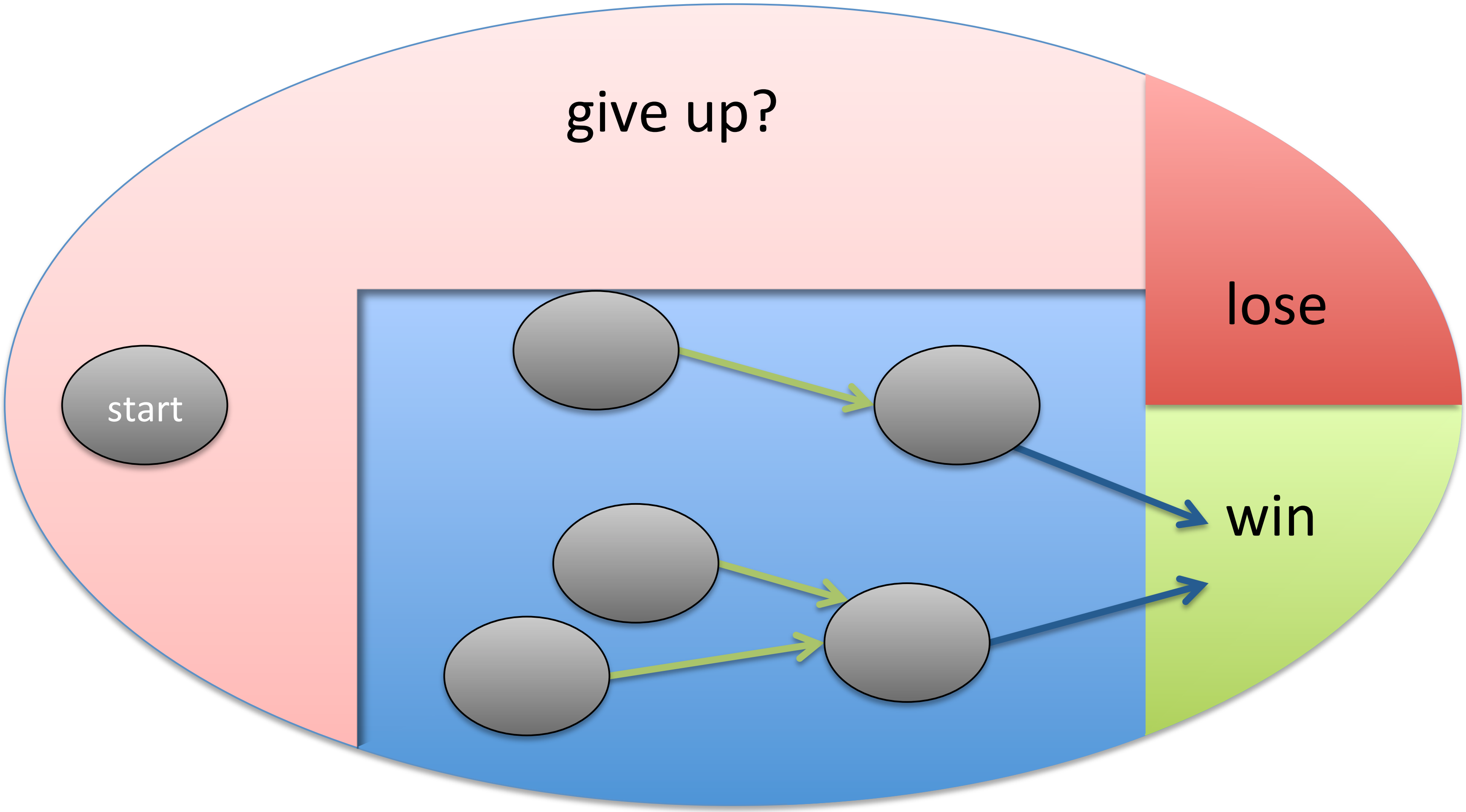
How to find stalemate states



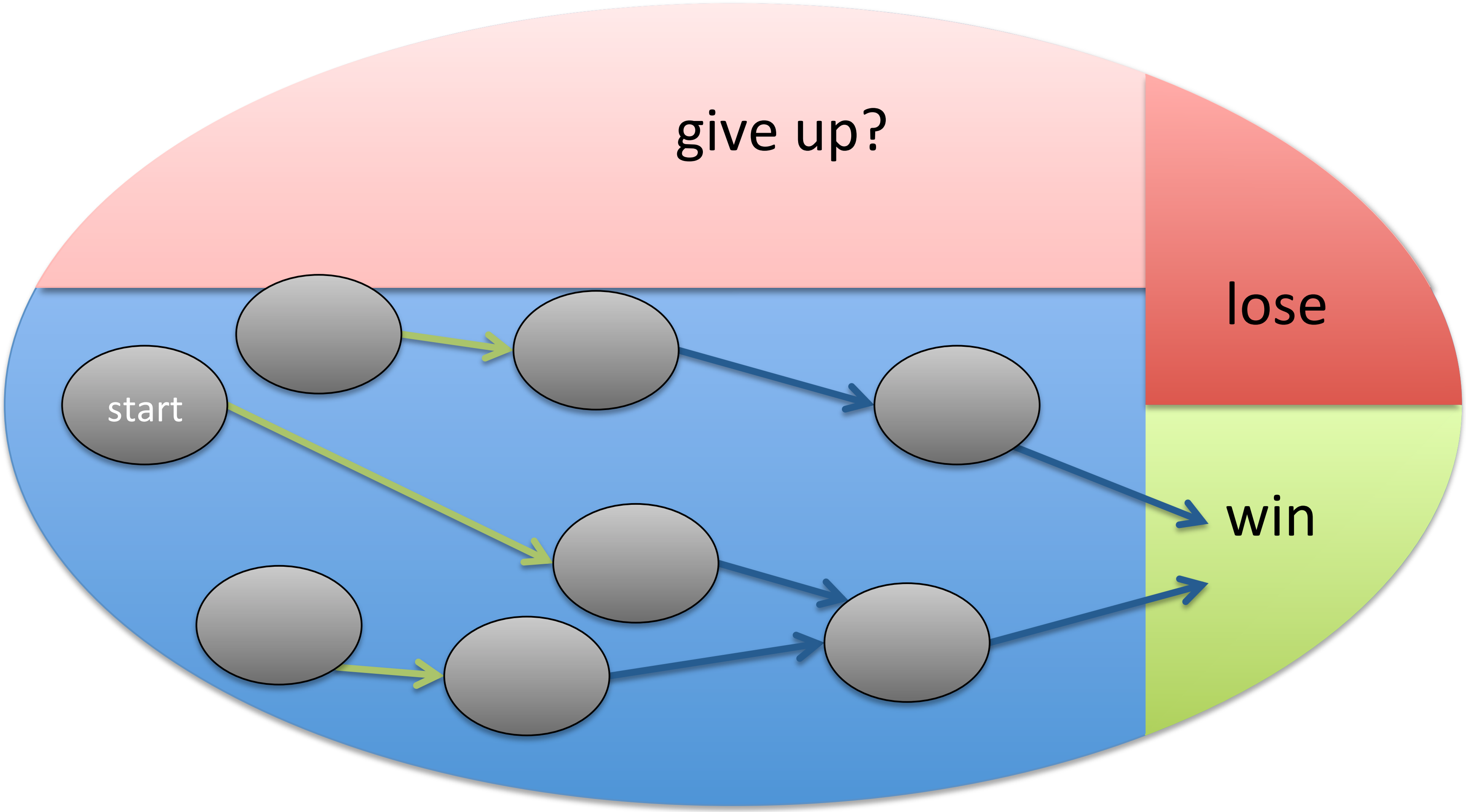
How to find stalemate states



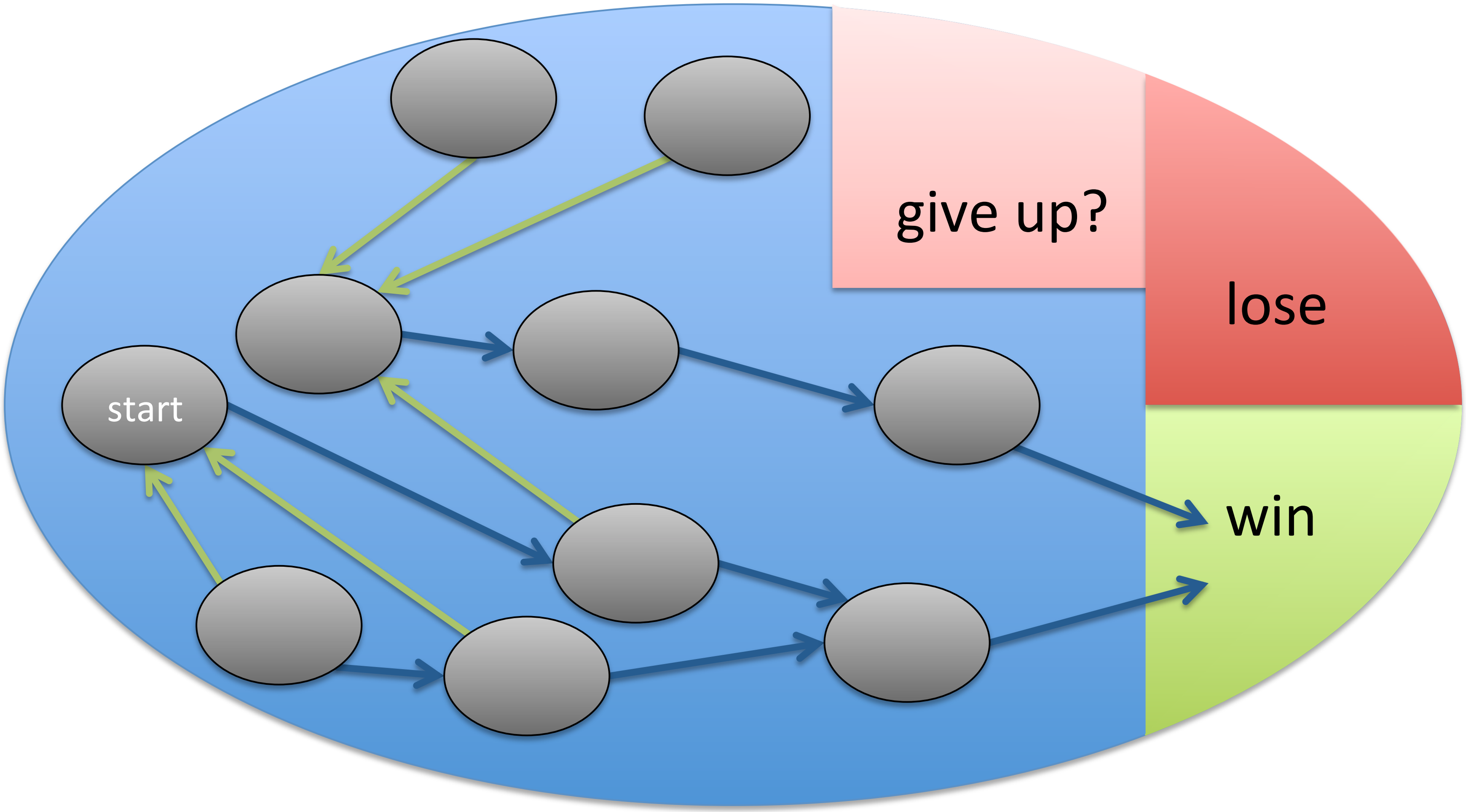
How to find stalemate states



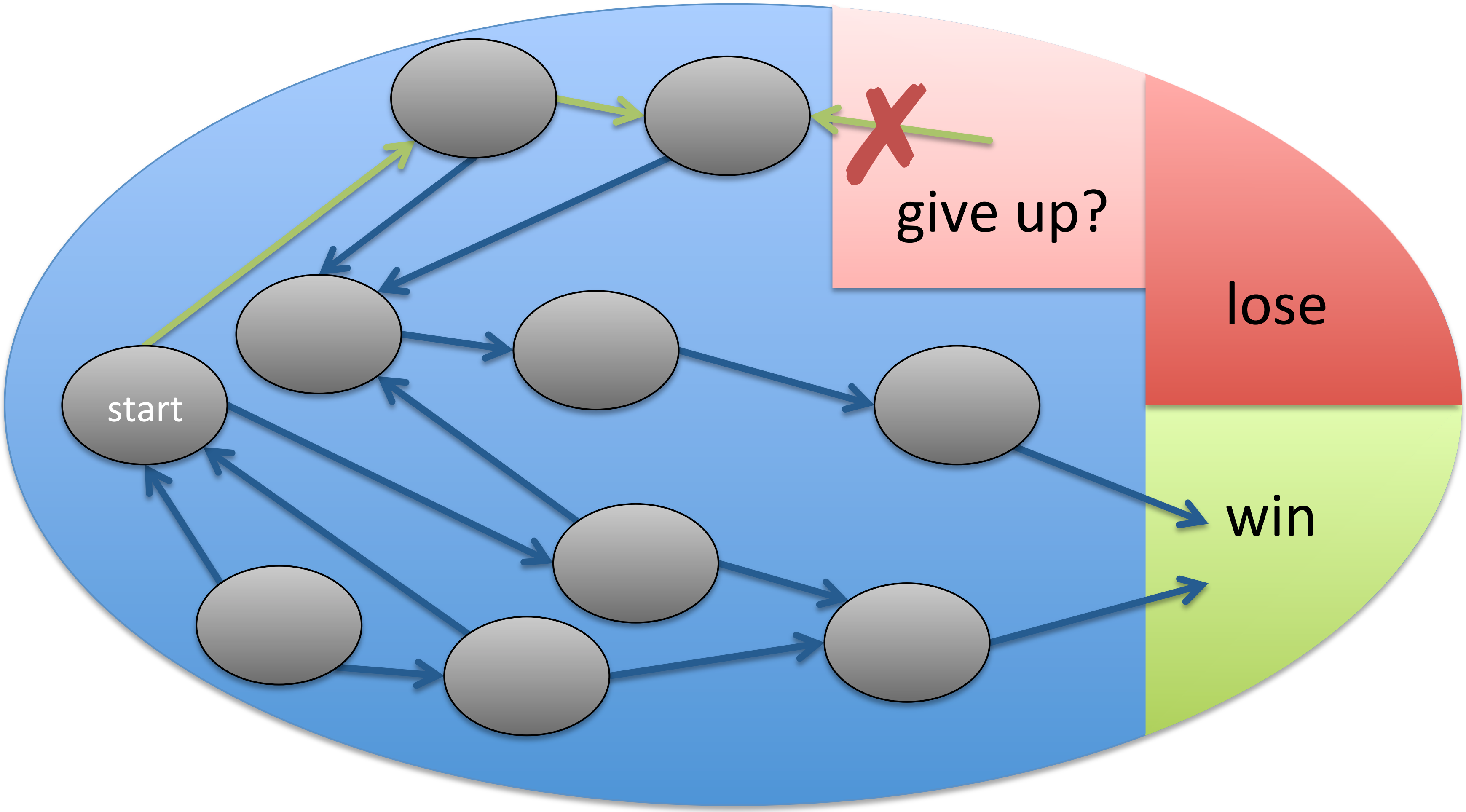
How to find stalemate states



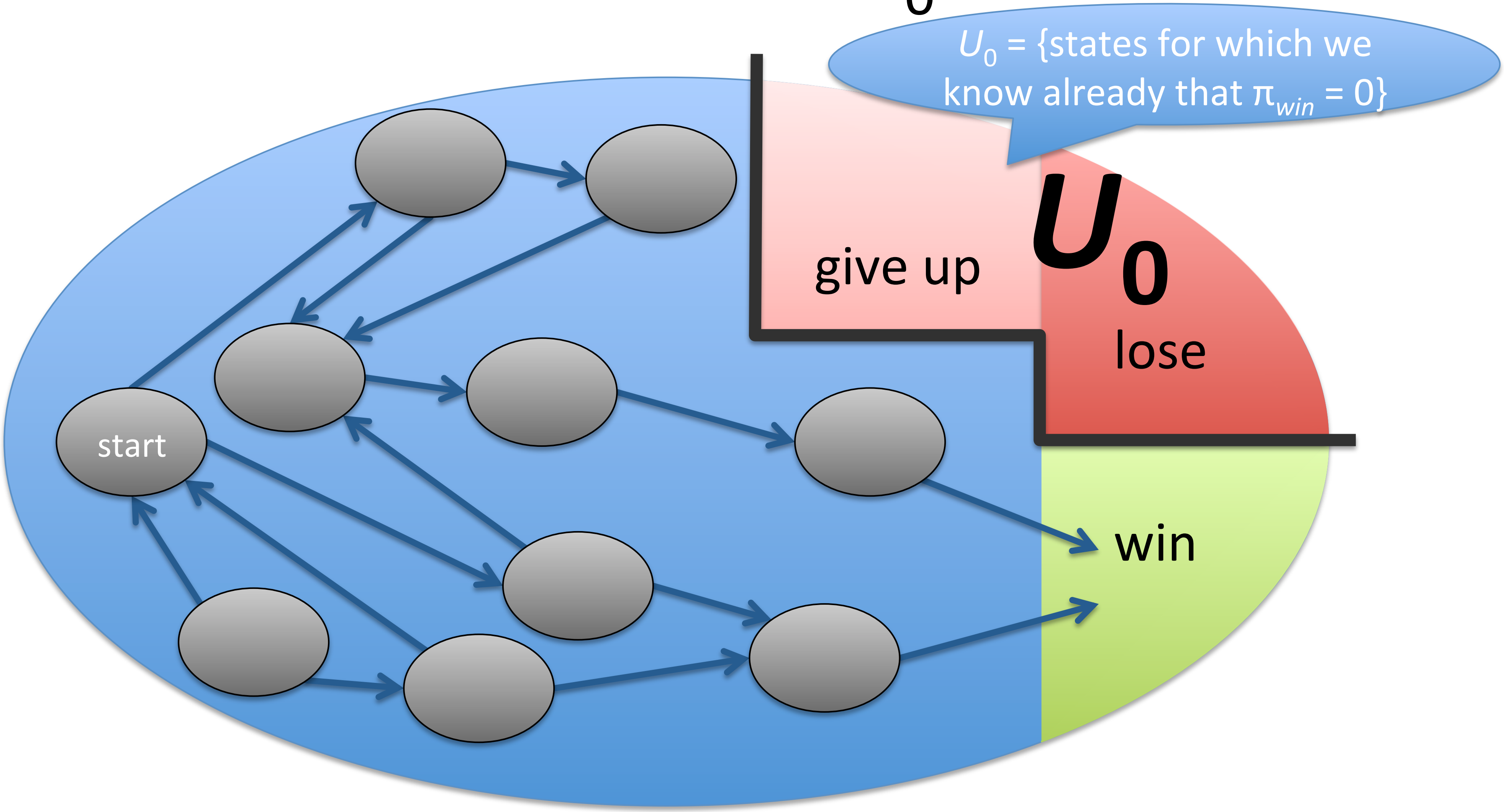
How to find stalemate states



How to find stalemate states



How to find U_0



How to find U_0

- create list of winning states
- iterate through list from start to end:
  for each state,
  add non-losing predecessors to end of list
- when iteration is complete,
  the list contains all states except U_0

Recapitulation

- PCTL: a logic to describe properties of Markov chains
- most important property: until formula = constrained probabilistic reachability
- compute probability with equation system
  - solution is unique if ...