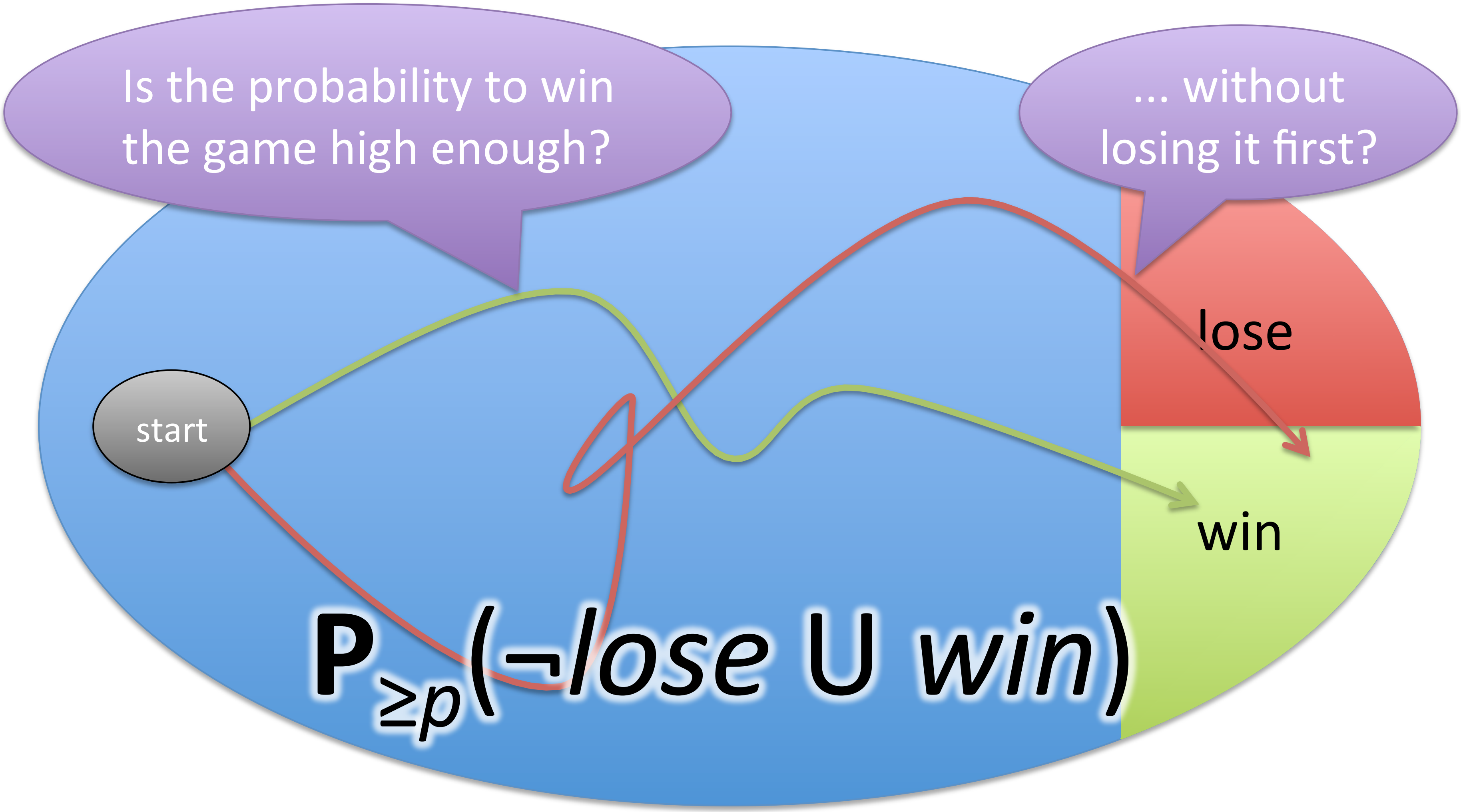
# Probabilistic CTL

David N. Jansen

# Constrained Probabilistic Reachability



# PCTL syntax

- state formulas $\phi$, $\psi$
  - $a$ — 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 $\Pi$
  - $X\ \phi$ — next state
  - $\psi\ U\ \phi$ — unbounded until
  - $\psi\ U^{\leq k}\ \phi$ — bounded until

# 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

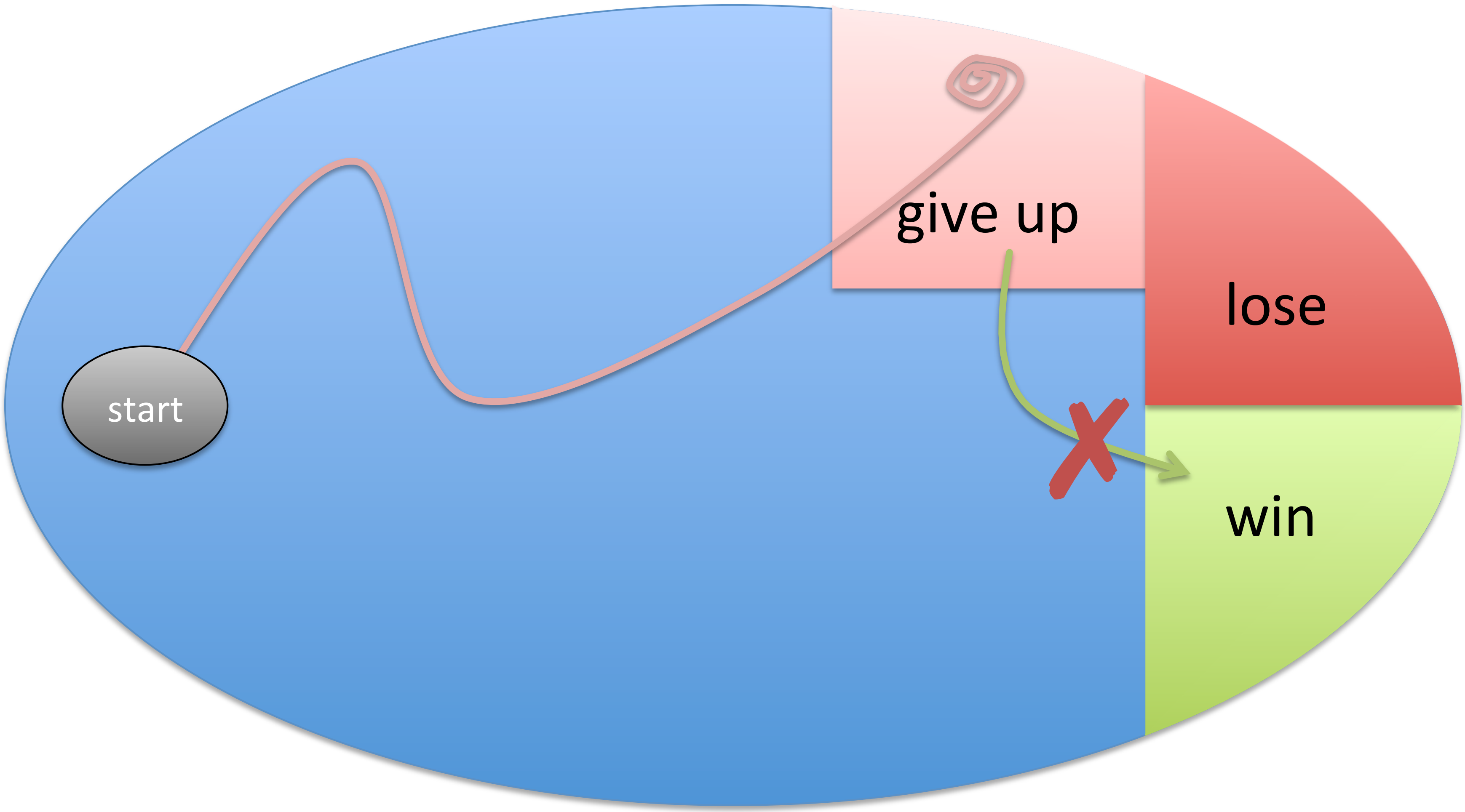
# How to calculate $\mathrm{Sat}(\mathbf{P}_{\geq p}(\neg \mathit{lose}\ \mathrm{U}^{\leq k}\ \mathit{win}))$

- modify Markov chain:
  make all *win*- and all *lose*-states absorbing
- calculate $\pi_{win} = P^{\mathrm{k}} \cdot \mathbf{1}_{win}$
- 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!

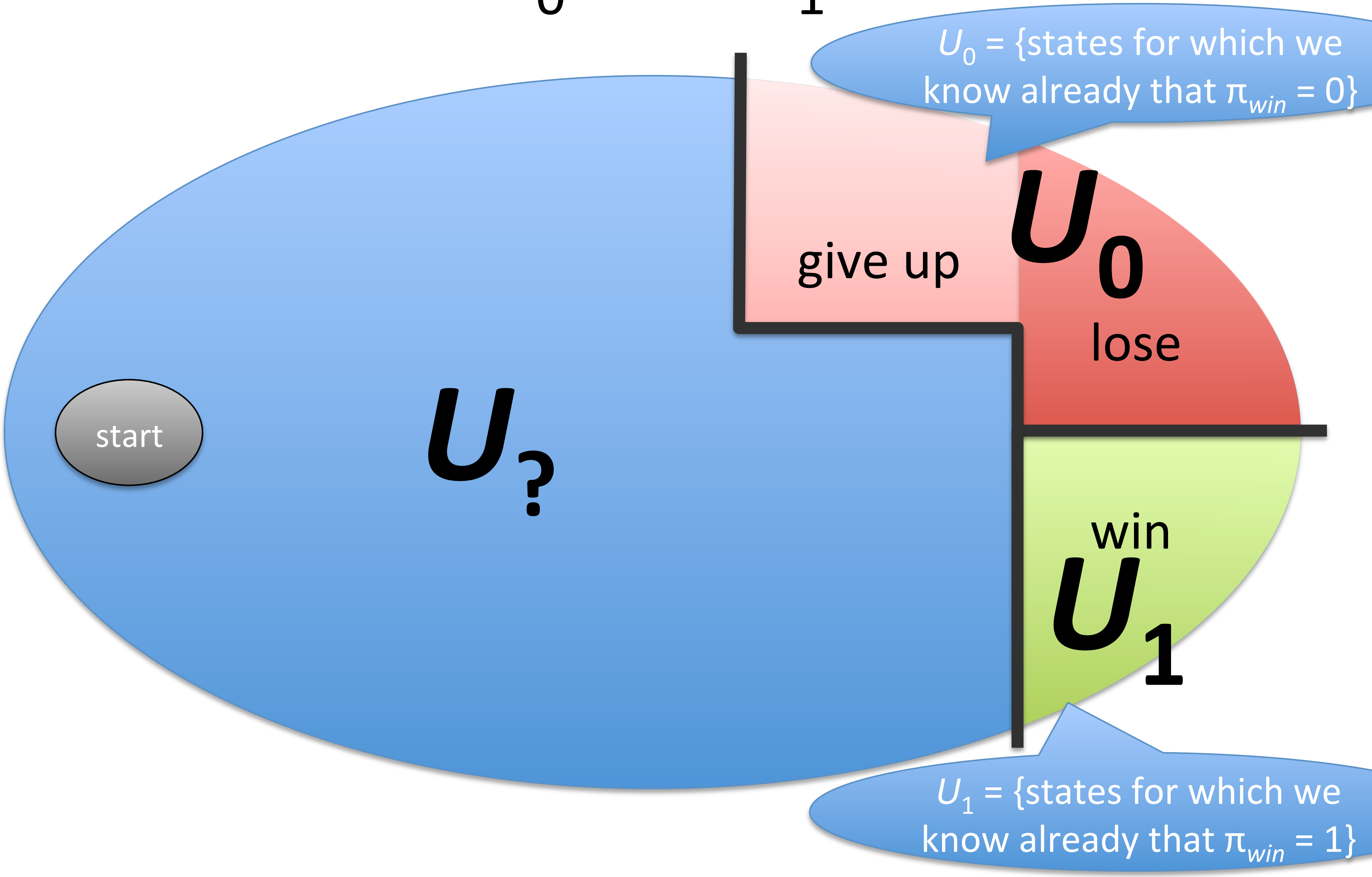
# Stalemate states



# 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 $U_0$

# $U_0$ and $U_1$


$U_0$ = {states for which we know already that $\pi_{win}$ = 0}
give up
$U_0$
lose
start
$U_?$
win
$U_1$
$U_1$ = {states for which we know already that $\pi_{win}$ = 1}

# Step-bounded reachability

- Assume $U_1$ = {winning states}
- $x_n(s)$ := probability of $(\neg lose\ \mathrm{U}^{\leq n}\ win)$,
  if $s$ is the start state

- $x_0(U_1) = 1$ $\quad\quad$ $x_0(U_0 \cup U_?) = 0$
- $x_{n+1}(U_1) = 1$ $\quad\quad$ $x_{n+1}(U_0) = 0$

$$x_{n+1}(s) = \sum_{t \in S} \mathbf{P}(s,t) \cdot x_n(t)$$

# How to calculate the probability of Sat(¬*lose* U *win*)

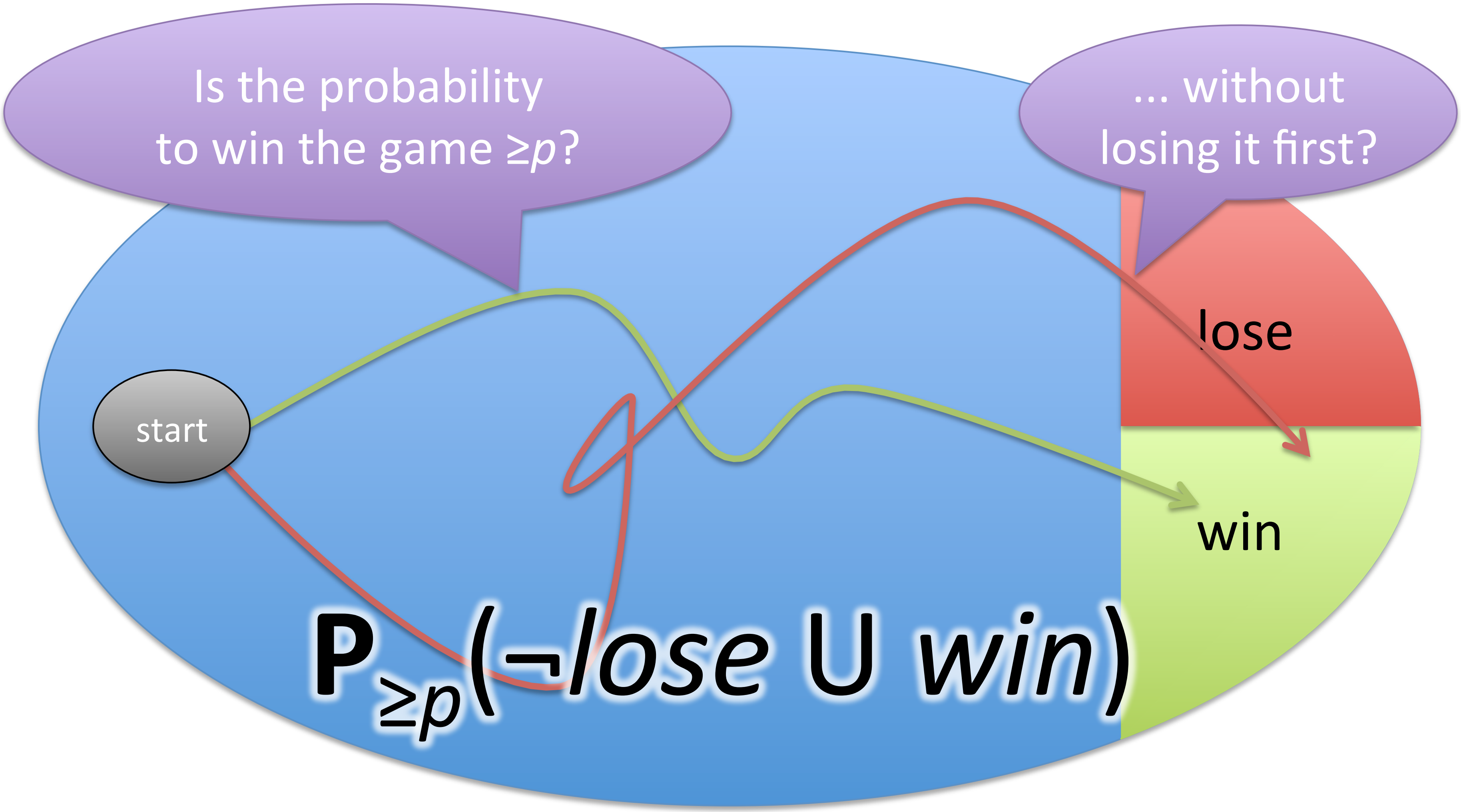
- Make all states in Sat(*win*) absorbing.
  Reason: All paths reaching a *win*-state satisfy the formula.
- Make all states in Sat(*lose*) absorbing.
  Reason: All paths reaching a *lose*-state falsify the formula.
- In the modified MC,
  Sat(¬*lose* U *win*) = Sat(*true* U *win*)

# How to calculate the probability of Sat(*true* U *win*)

- Only states in Sat(*win*) $\cup$ $U_0$ are recurrent
  (i.e. have steady-state probability ≠ 0)
- use steady-state analysis
  to find the probability to reach Sat(*win*)

- Two weeks ago:
  steady-state analysis for irreducible MCs
- Today:
  steady-state analysis for reducible MCs

# Constrained Probabilistic Reachability




$$\mathbf{P}_{\geq p}(\neg \mathit{lose}\ \mathrm{U}\ \mathit{win})$$

# How to Calculate Constrained Probabilistic Reachability

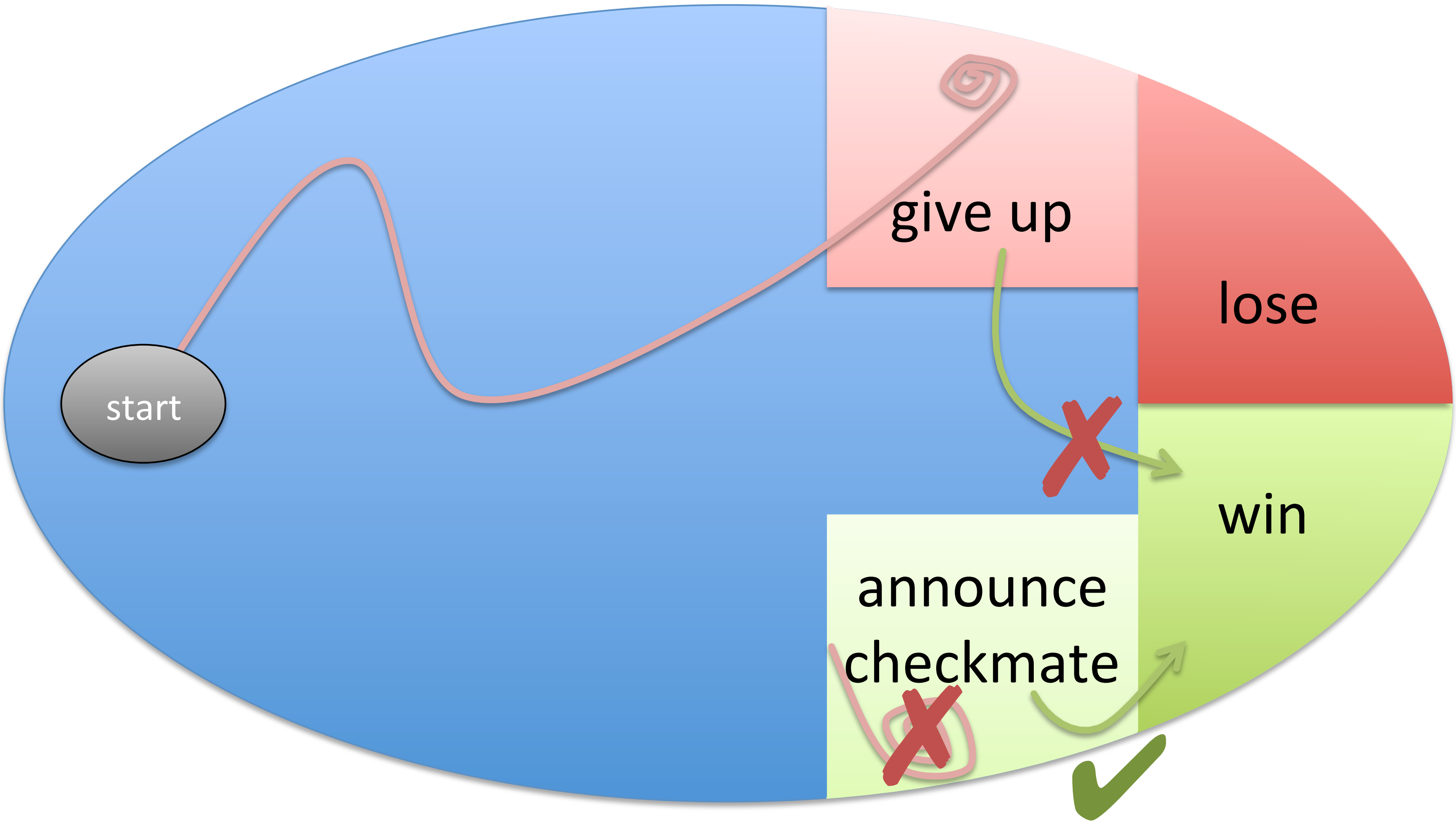
- $x(s)$ := probability to win without losing first, if $s$ is the start state

- $x(win) = 1$ $\qquad\qquad$ $x(lose) = 0$

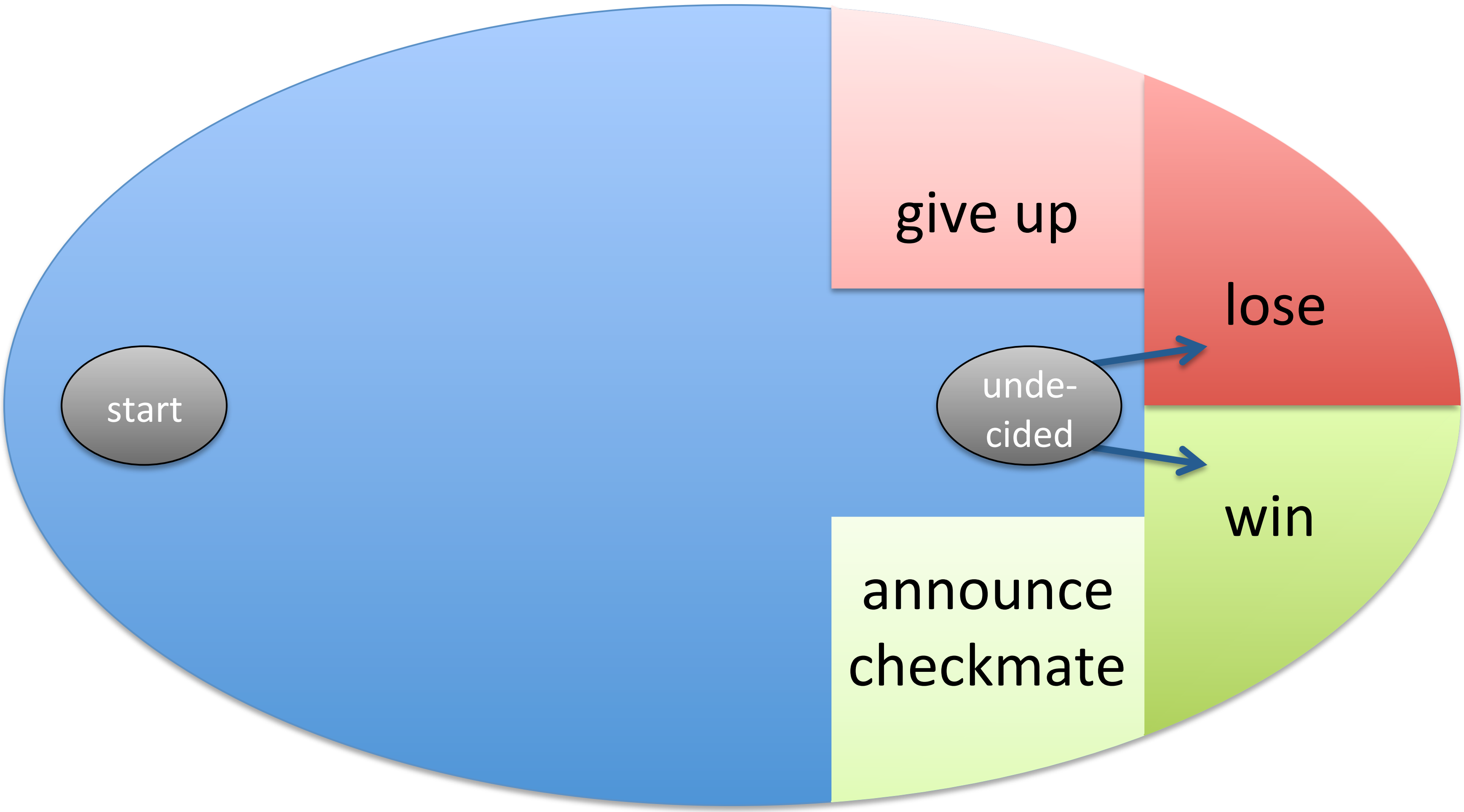
- $$x(s) = \sum_{t \in S} \mathbf{P}(s,t) \cdot x(t)$$

Linear equation system

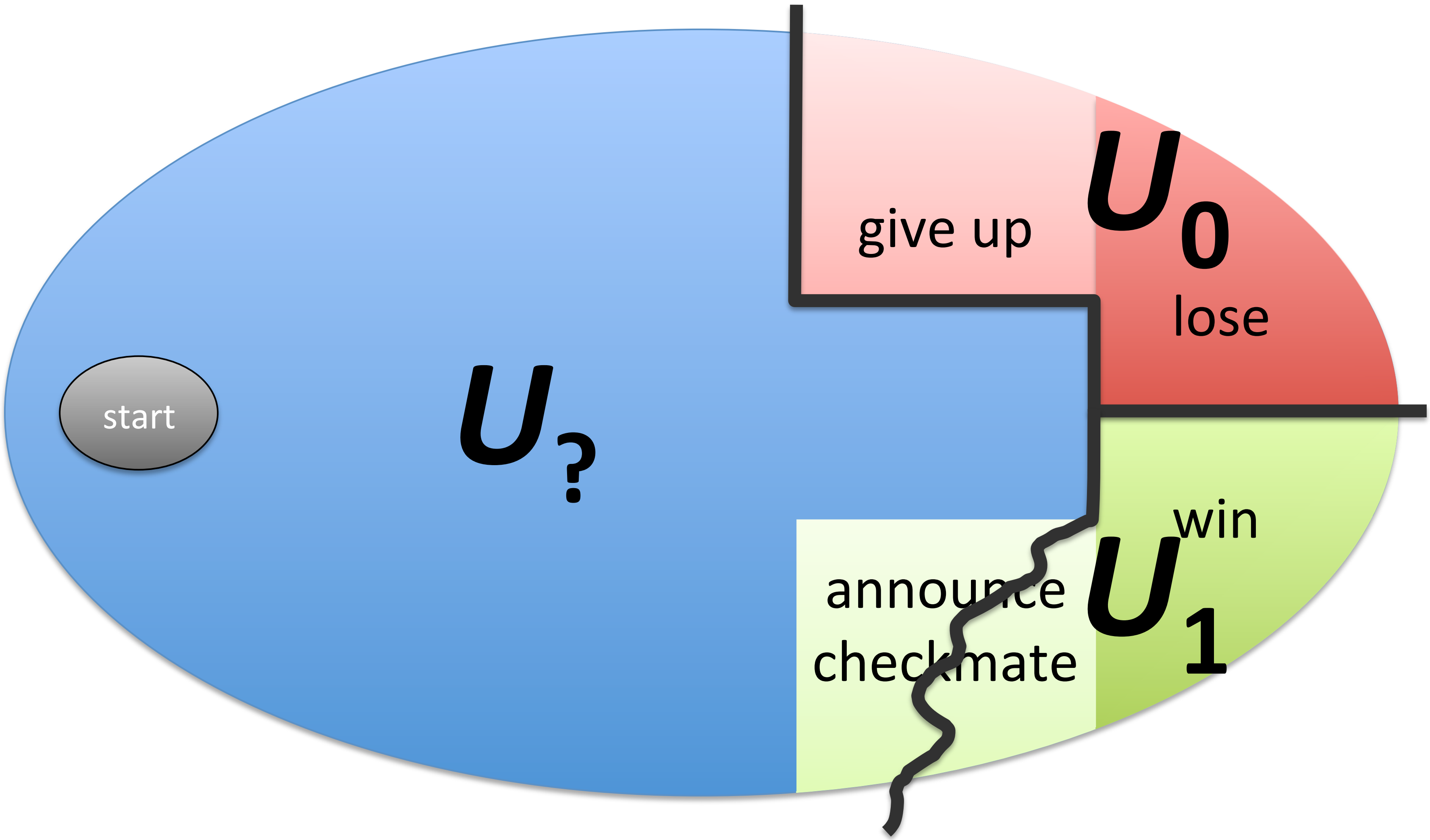
# States where you cannot lose



# Undecided States May Exist



# Non-Unique Solution: Stalemate

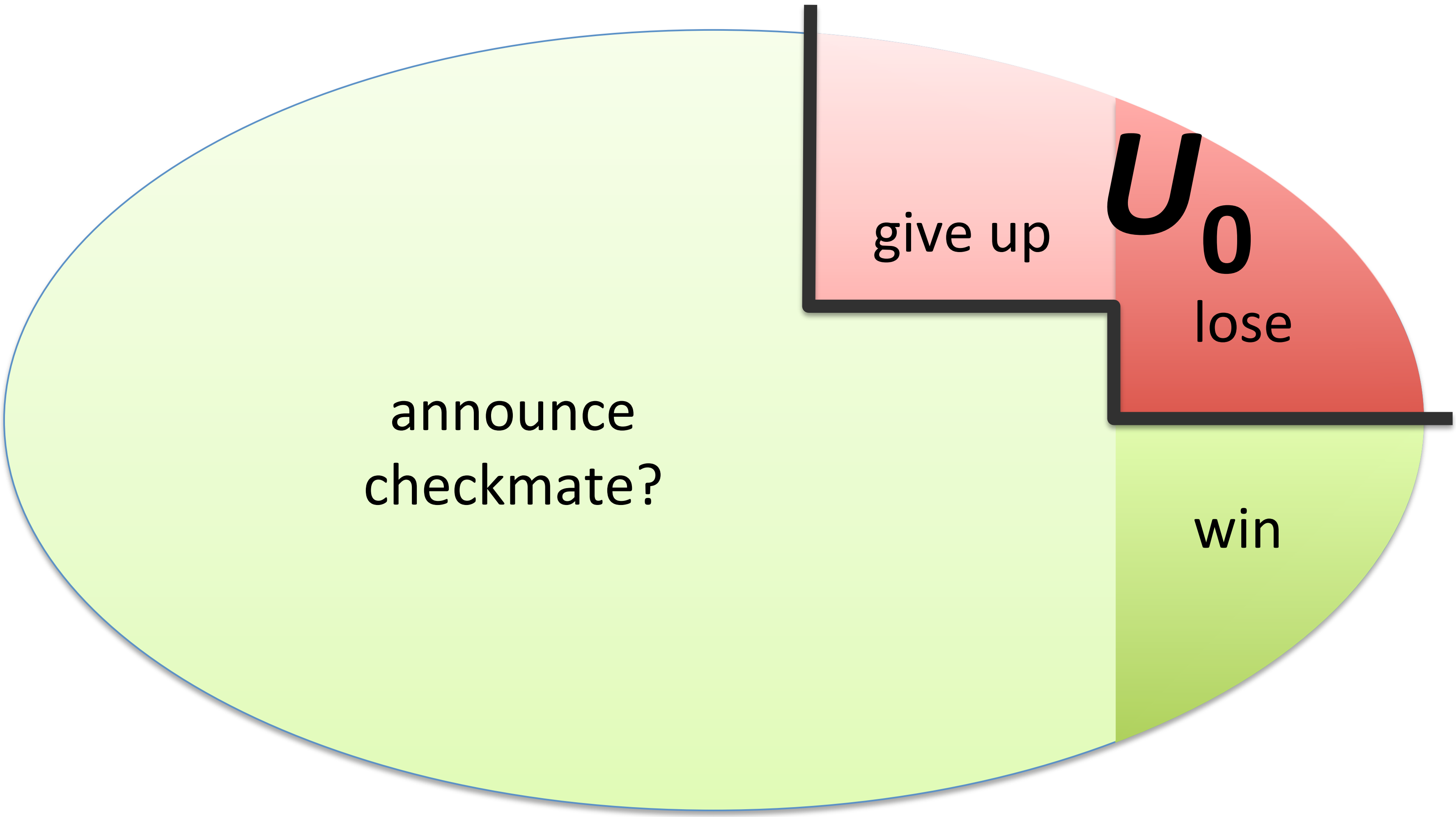


# Unique Solution

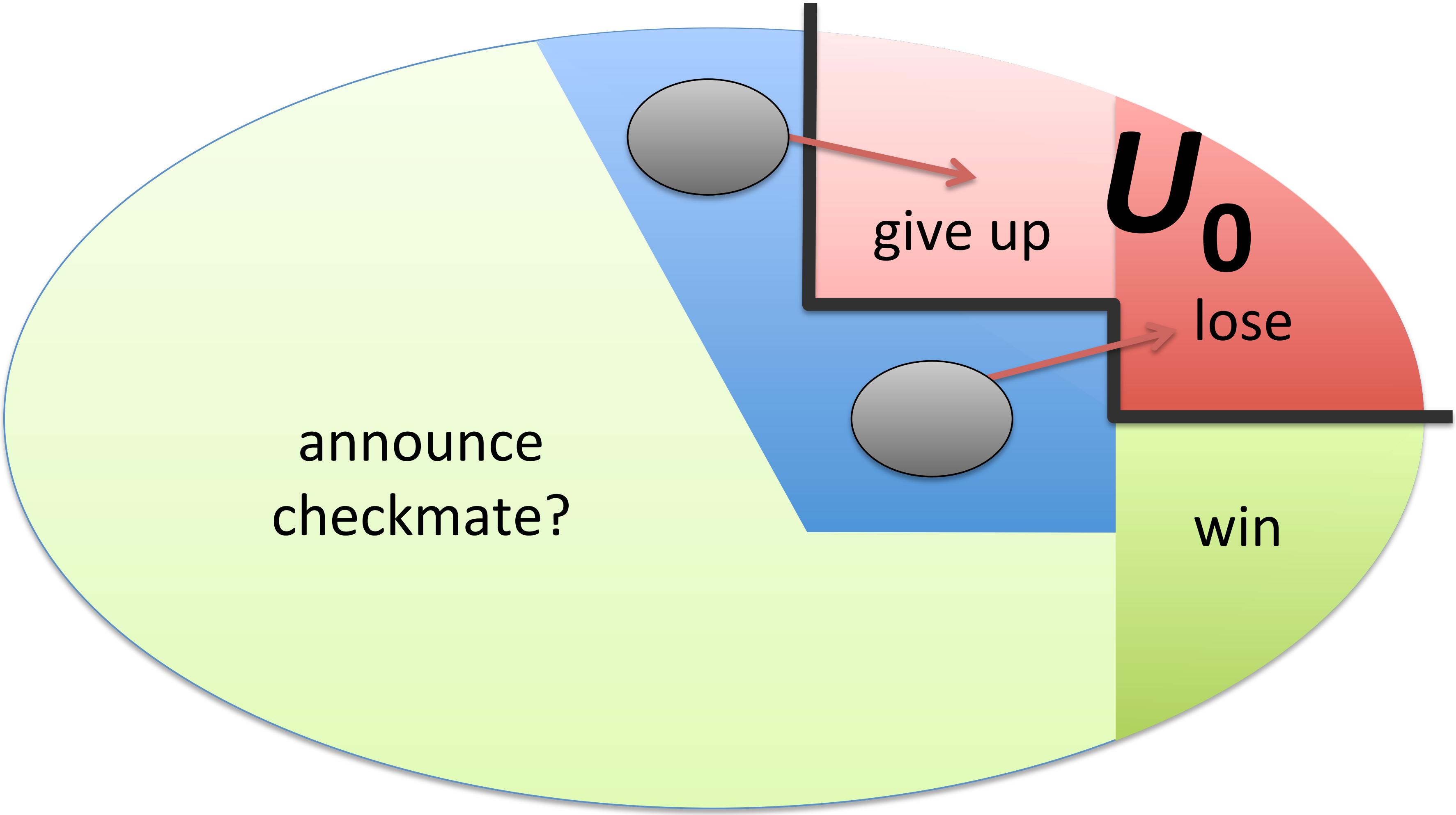
- $x(s) = 1$ if $s \in U_1$
- $x(s) = 0$ if $s \in U_0$
- $x(s) = \sum_{t \in S_?} \mathbf{P}(s,t) \cdot x(t) + \sum_{t \in S_1} \mathbf{P}(s,t)$ if $s \in U_?$
- **If all “stalemate” states $\in U_0$,**
  **this equation system has unique solution.**

- Proof of Theorem: on the board...
- (Baier/Katoen, p. 766)

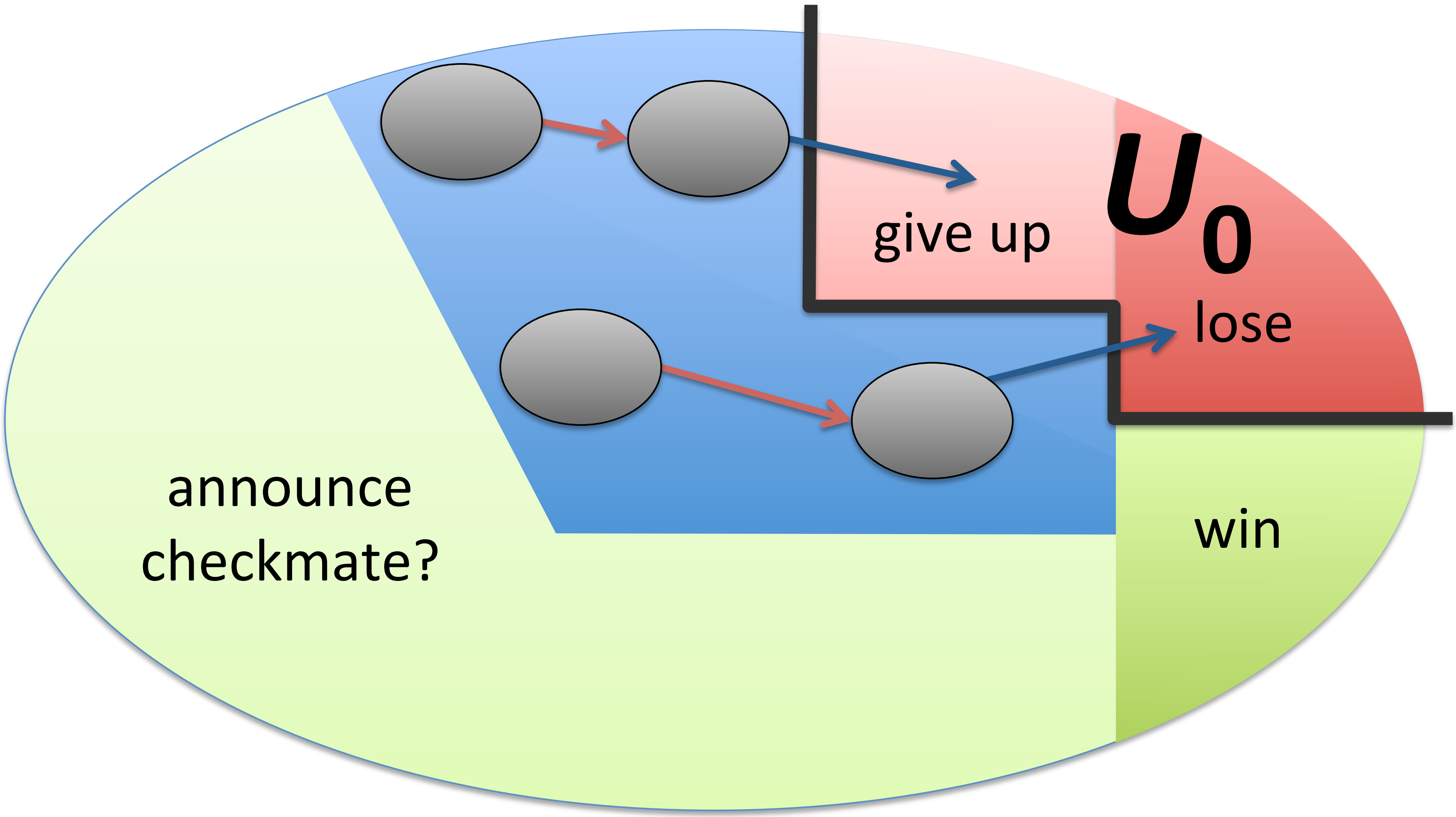
# How to find $U_1$ (optimal)



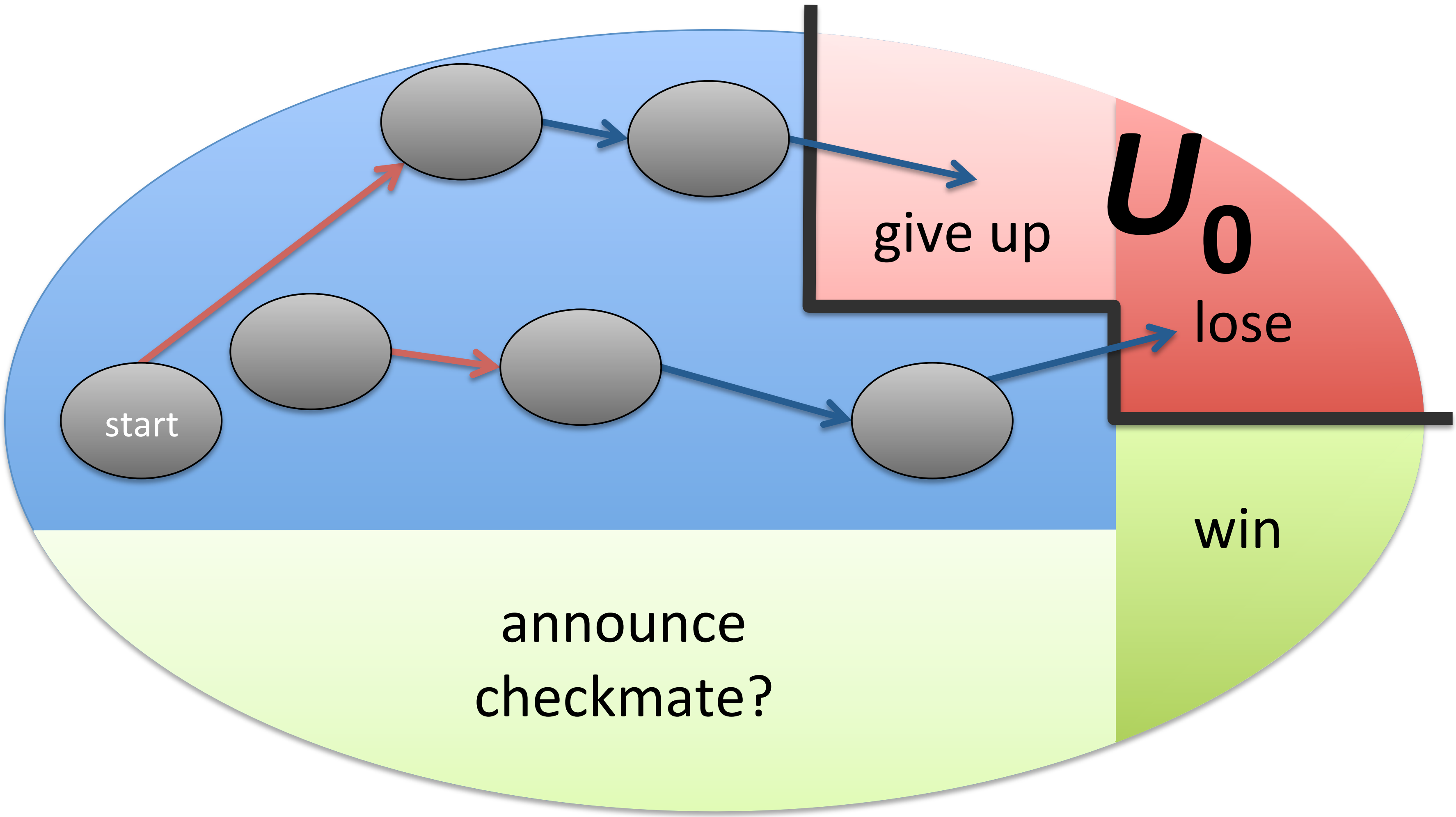
# How to find $U_1$ (optimal)



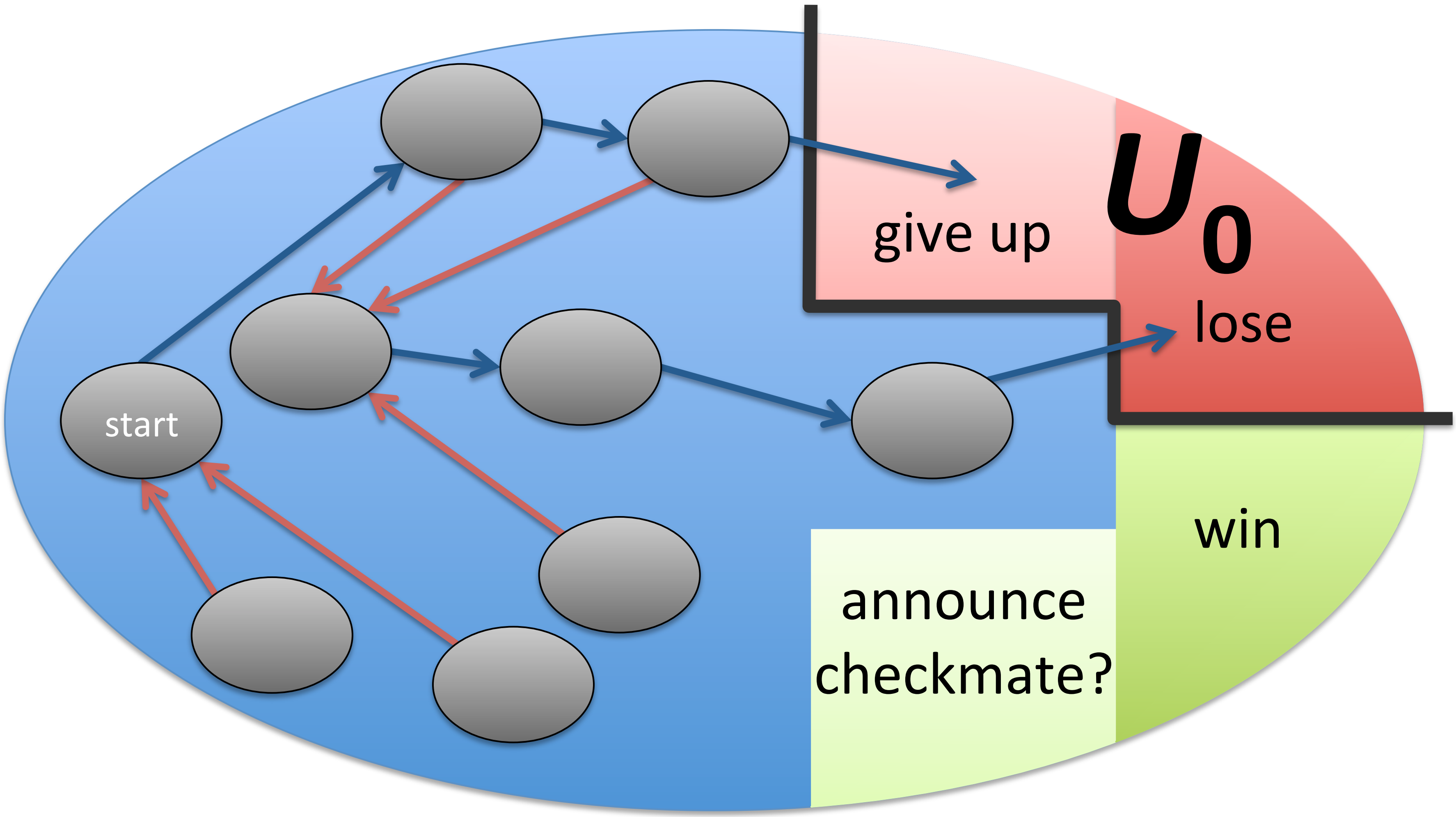
# How to find $U_1$ (optimal)



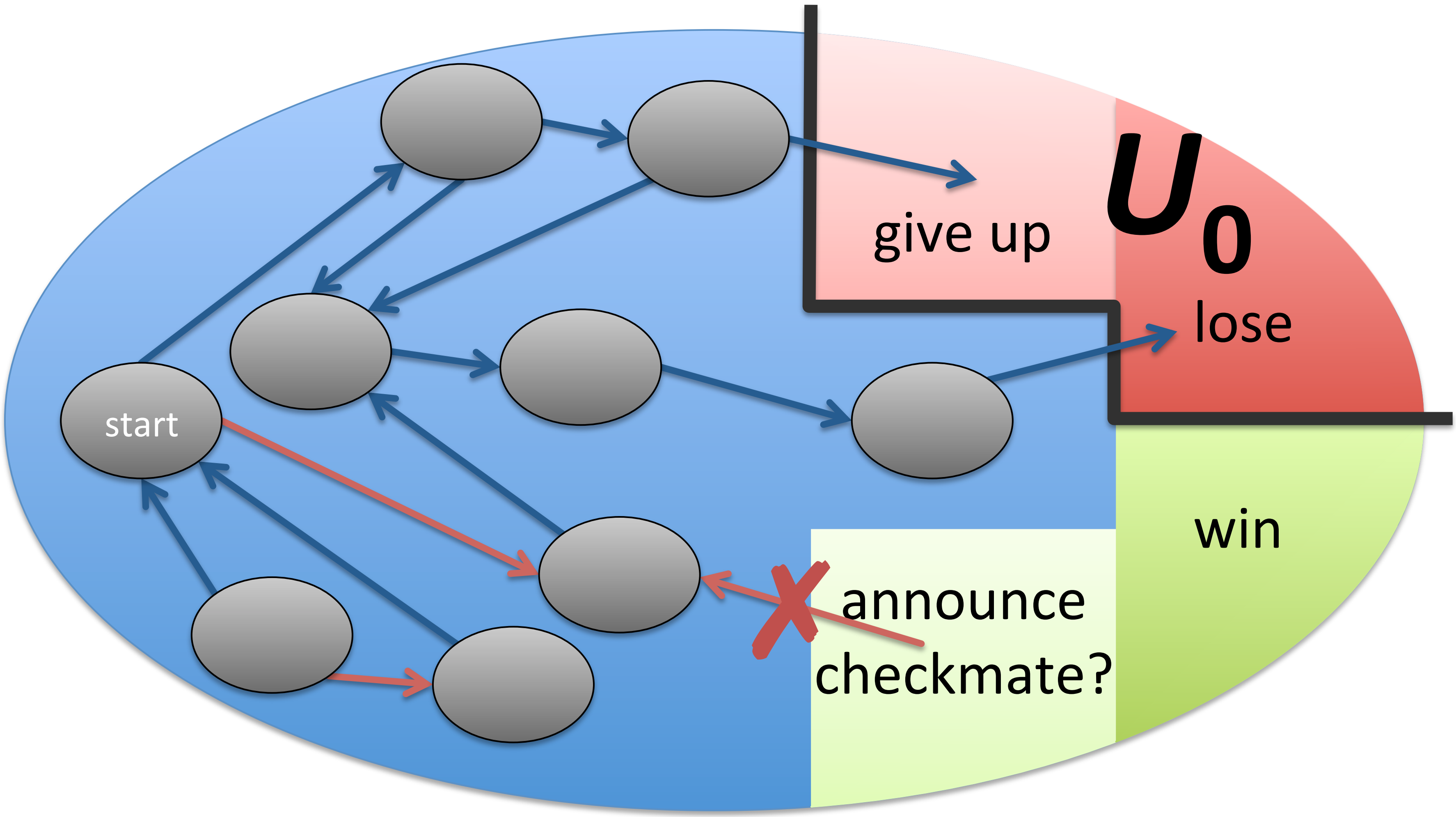
# How to find $U_1$ (optimal)



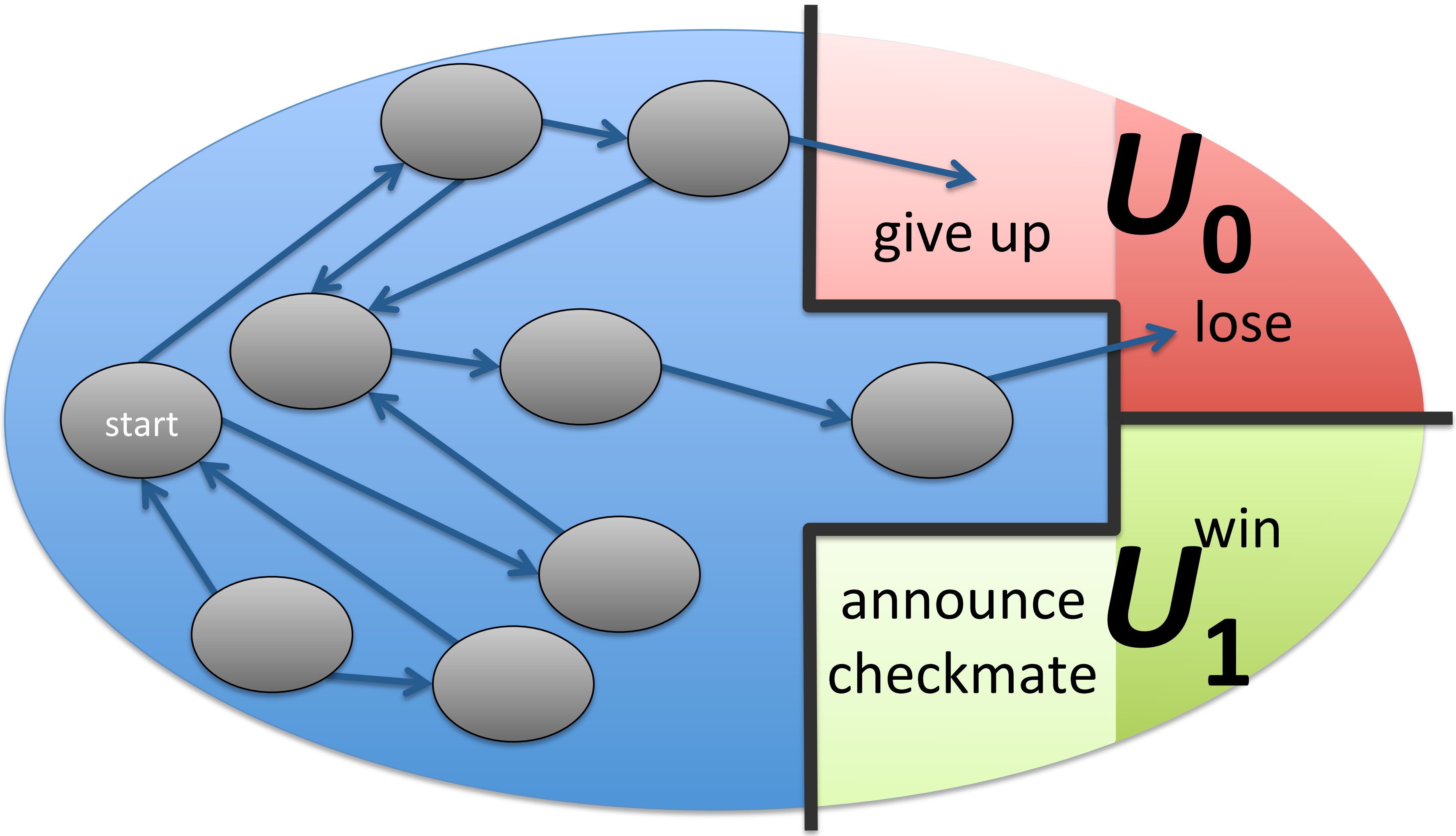
# How to find $U_1$ (optimal)



# How to find $U_1$ (optimal)



# How to find $U_1$ (optimal)



# How to find $U_1$ (optimal)

- create list of $U_0$-states
- iterate through list from start to end:
  for each state,
  add non-winning predecessors to end of list
- when iteration is complete,
  the list contains all states except $U_1$

# Numerical Stability

- Direct solution (= invert matrix) is numerically unstable
- Better: iterative solution
- several iterations exist:
  - $\Upsilon$, power method
  - Jacobi or Gauss-Seidel iteration (general iterations for linear equation systems)

# Power method iteration

- Define a functional
  $\Upsilon: (x: S \rightarrow [0,1]) \mapsto (x': S \rightarrow [0,1])$
- $x'(\text{win}) = 1$ $x'(\text{lose}) = 0$
- $$x'(s) = \sum_{t \in S} \mathbf{P}(s,t) \cdot x(t)$$
- Theorem 10.15: **Least fixed point of $\Upsilon$ is the solution.**

- Proof of Theorem: on the board...

- The fixpoint is unique: (Knaster–Tarski)

# Model checking procedure

- Assume given a DTMC and a formula $\phi$
- start with simple subformulas of $\phi$:
  - For each subformula $\phi'$, find $Sat(\phi')$
  - Reuse results of even simpler subformulas, as in semantics (e.g. $Sat(\neg\phi) = S \setminus Sat(\phi)$ )
- Continue until you reach $\phi' = \phi$

# Example

$$start \wedge \mathbf{P}_{\geq 0.3}(true \; \mathrm{U}^{\leq 6} \; win) \wedge \mathbf{P}_{=1}(true \; \mathrm{U} \; state_4 \vee \mathbf{P}_{\geq 0.4}(true \; \mathrm{U} \; win) \vee state_{10})$$

# 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 ...