

Controlling a Population

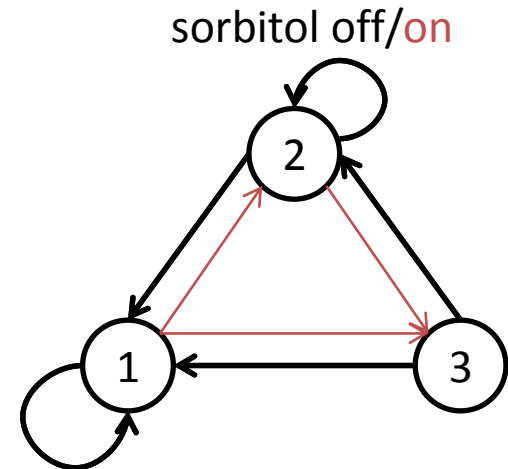
Nathalie Bertrand, Miheer Dewaskar,
Blaise Genest, Hugo Gimbert

Idealized Setting:

Question: Can we *surely* send *all* yeasts to State 1 in the same time?
(no matter how each yeast react)

Control at each step:
same action for every yeast.

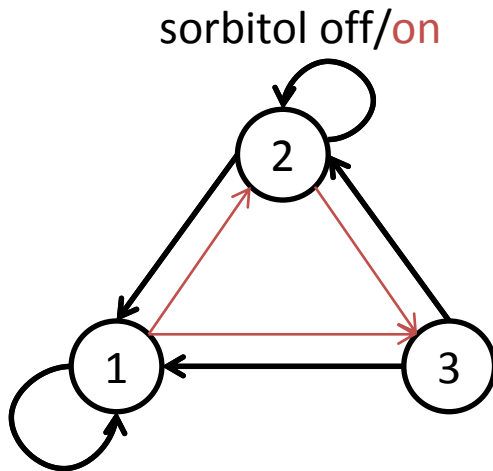
e.g.: sorbitol *off* for all yeast,
sorbitol *on* for all yeast



2 player *param.* game on NFA:
Controller player +
Agents resolving non det.

Problem: For all number N of yeasts, does there exist a controller σ_N
($\sigma_N(\text{history}) = \text{action to play}$)
such that no infinite history compatible with σ_N avoiding State $(N,0,0)$

Basic Properties:



Prop: If controller for N yeasts, then controller for $N-1$ yeasts.

Idea of the proof: σ_{N-1} plays σ_N simulating a fake yeast.

Either exists a strategy σ_N for all N , or **cut-off** N_0 with strategy for $N < N_0$, and no strategy for $N > N_0$.

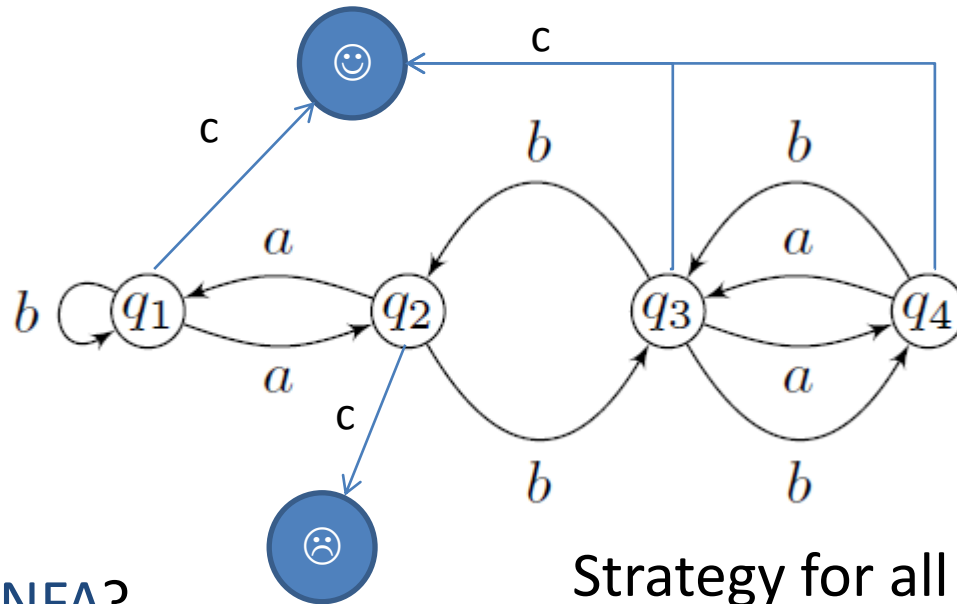
Result 1:

Either exists a strategy σ_N for all N , or **cut-off** N_0 with strategy for $N < N_0$, and no strategy for $N > N_0$.

Th1: there exists NFA for which the **cut-off** is **doubly exponential** in the size of the **NFA**

Corollary: using « small model » properties would entail a **2EXPTIME** or more complexity.

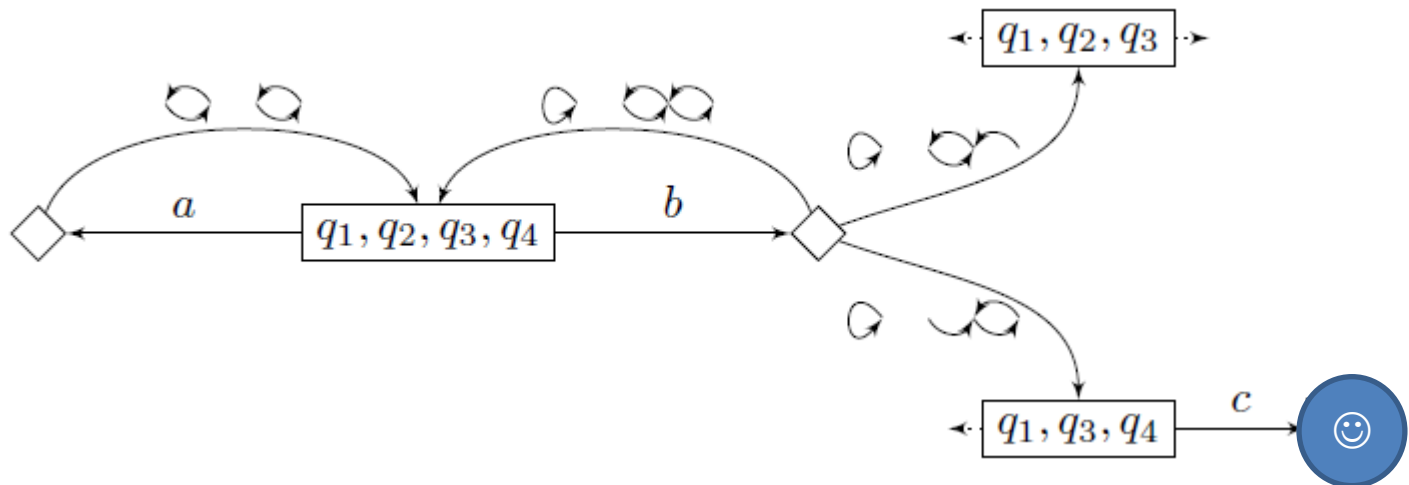
Solving control problem symbolically



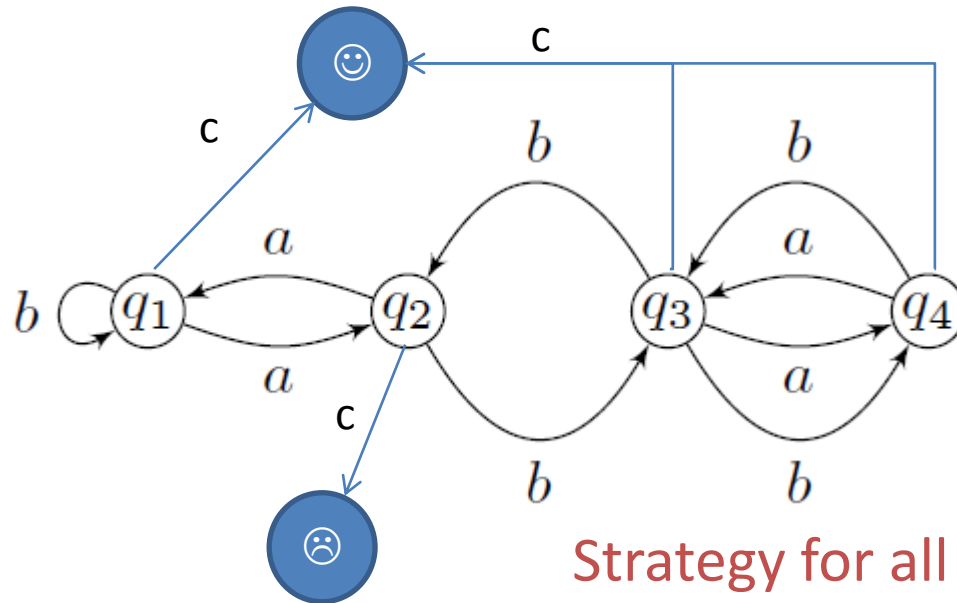
Use support NFA?

Strategy for all N.

No strategy in the support NFA.



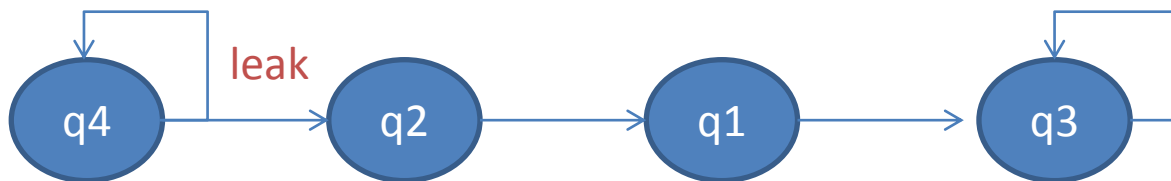
Solving control problem **symbolically**



Strategy for all N .

No strategy in the support NFA.

play **ab**:

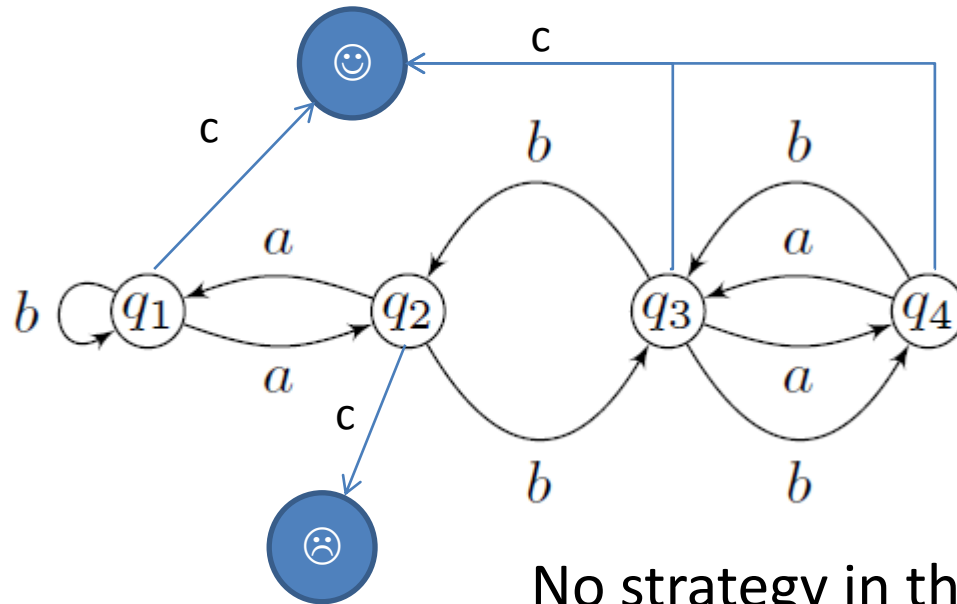


Playing **(ab)*** long enough till no token in q_2 .

And then we can play **c** and reach 😊.

Hence for all N , there is a strategy σ_N leading to 😊.

Solving control problem **symbolically**



Notice that playing a^* or b^* is not winning:

a^* : $q1 \rightarrow q2 \rightarrow q1, q3 \rightarrow q4 \rightarrow q3$

b^* : counter strategy : $q1 \rightarrow q1, q3 \rightarrow q2+q4, q4 \rightarrow q3, q2 \rightarrow q3$

Playing **positionally** on support is **not sufficient** to win.

Solving control problem symbolically

Prop:

If winning strategy in the support game, then winning strategy for all N.

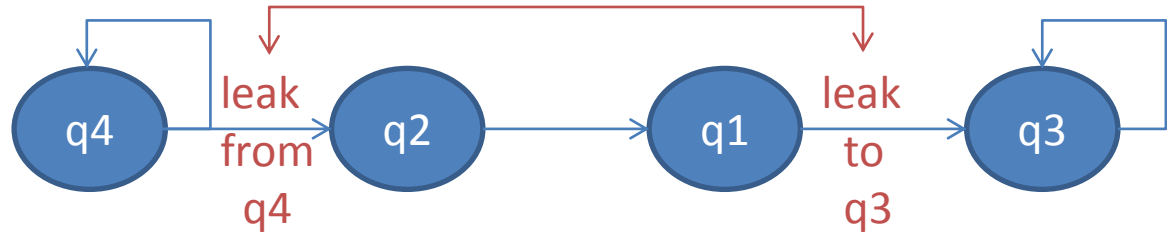
Converse not necessarily true: agents strategy can be leaking(cheating)

Change the **winning condition** in support game:

An infinite play is **B-winning for controller** in support game if reach {😊} or there is an « **unbounded leak** ».

Defining Leaks

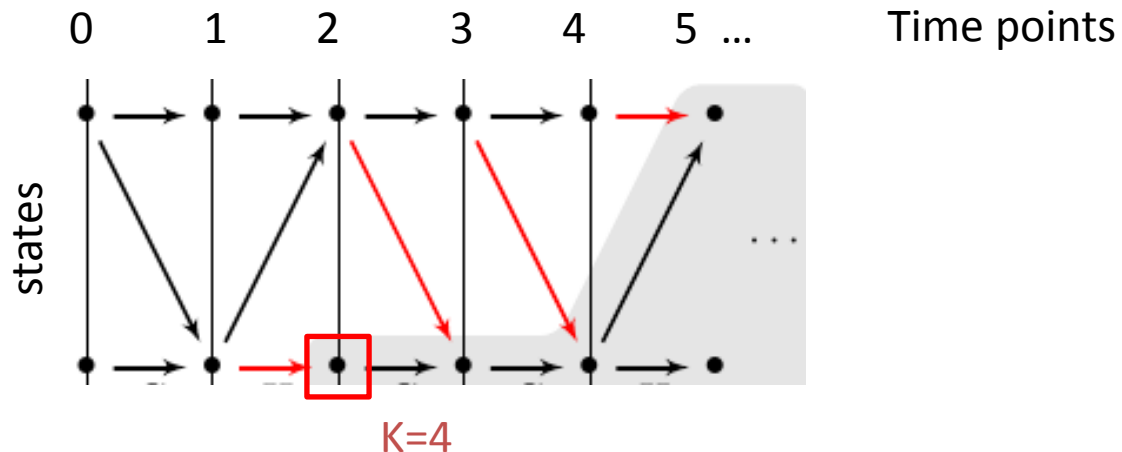
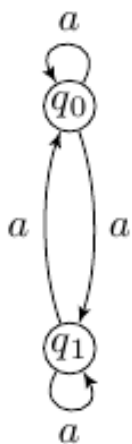
For repeated graph G^* :
 No unbounded leak iff
 Union of BSCCs



In general state $q1, q3$ can change.

Look at how each state transform along an infinite play.

Unbounded leak if for all K , exists a time point and state q with K or more entries in the future of q



Solving control problem symbolically

An infinite play is **B-winning** for controller in support game if reach {😊} or there is an « **unbounded leak** ».

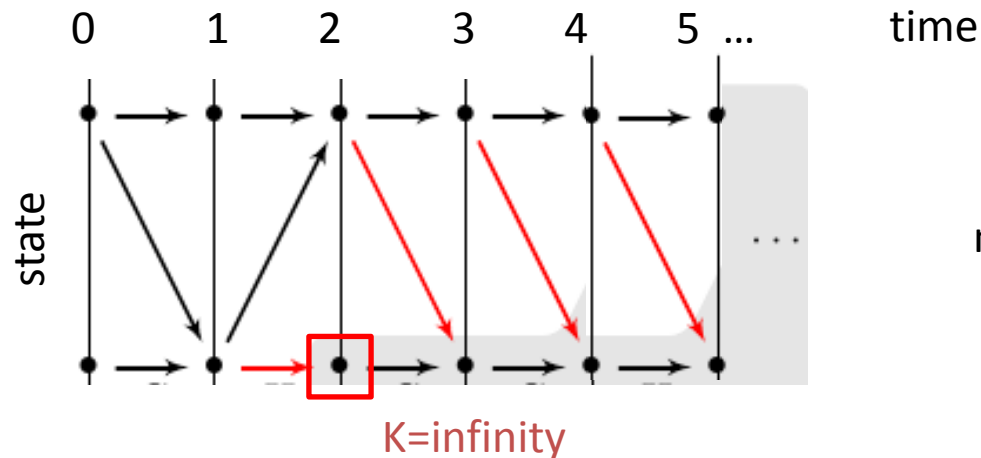
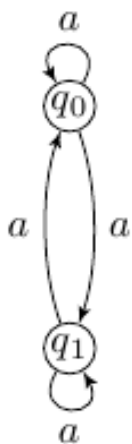
Prop: for all **N**, controller has a **winning strat** for **N tokens** iff Controller has a **B-winning strat** in the **support game**.

But **Unbounded** condition hard (Colcombet Bojanczik) to check

Defining Leaks

Unbounded leak if for all K , exists a time point and state q with K or more entries in the future of q

Infinite leak if exists a time point and state q with **Infinite** number of **entries** in the future of q



Unbounded leak
not equivalent with
infinite leak

Solving control problem symbolically

⇒ replace unbounded leak by **infinite** leak (Buchi condition, easier).

An infinite play is **I-winning for controller** in support game if reach {😊} or there is an « **infinite leak** ».

For **finite memory** strategies of Agents,
if there is **no infinite leak**, there is no unbounded leak

Reach-or-Leak

Question: How to solve reach-or-leak game?

Infinite Leaking accepted by a **exp. non-deterministic Buchi automata**:

Guess (q,i) and check that (q,i) has an infinite number of entries.

Can determinize it as a **doubly exp** Parity automaton with **exp.** Parities

Can run this parity automaton to tell if the play is winning.

\Rightarrow **Doubly exp** complexity. Finite memory M is also doubly exp.

\Rightarrow Cut-off **triple exponential** $N = 2^M 2^{2^{|A|+1}}$

Can we do better than $2^{EXPTIME}$?

Easy lower bound: $PSPACE$ (using exponentially many supports)

Th: Population control is $EXPTIME$ -complete

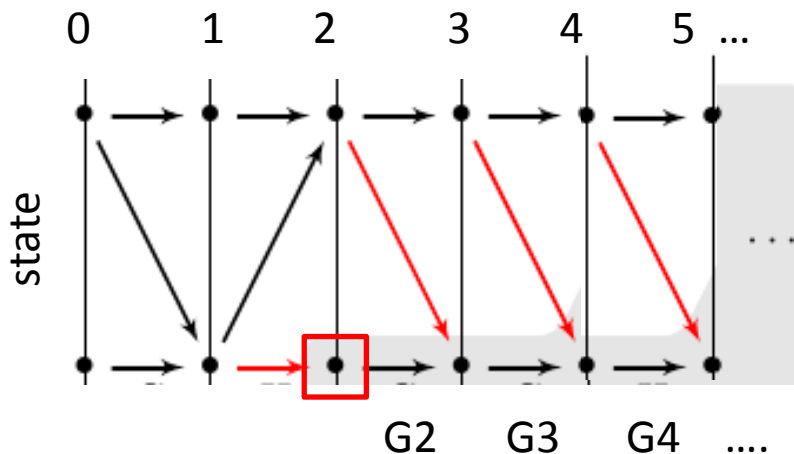
Exponential size Parity Automaton

Build ad-hoc deterministic automaton:

Compute deterministic parity automaton: **exp.** size, **poly** parities

⇒ EXPTIME, exponential memory, cut-off doubly exponential $N = 2^{M 2^{|A|}}$

Main issue: Determinise the choice of starting leak « i »



$G[2,4] = G2 \circ G3 \circ G4$ leaks at G5

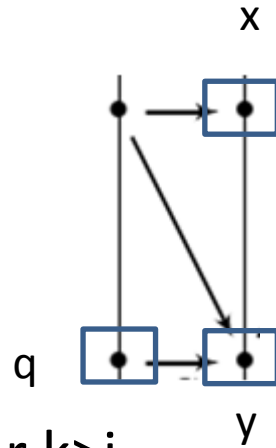
$i=2$



Separation

Determinise the choice of starting leak « i »

Def: We say that $G[i,j]=G_i..G_j$ **separates** (x,y) if for some q ,
 $(q,y) \in G[i,j]$ and $(q,x) \notin G[i,j]$



Lemma: If $G[i,j]$ separates (x,y) then $G[k,j]$ separates (x,y) for $k>i$

Lemma: Take the **minimal** i s.t. run is i -leaking.
Then exist $(x_j, y_j)_{j \geq i}$ such that :

$G[i,j]$ separates **all** $(x_j, y_j)_{j \geq i}$ and

$G[i-1,k]$ separates a **finite number of** $(x_j, y_j)_{j \geq i}$

Exponential size Parity Automaton

Determinise the choice of starting leak « i »

Keep in memory

$G[i_1, j], G[i_2, j] \dots G[i_n, j]$ such that

$G[i_k, j]$ separates some (x, y) not separated by $G[i_{k'}, j]$ for $k' < k$.

Only $|A|^2$ graphs to keep in memory. => **Exp. number of states**

One parity for each graph, to detect infinite leak from this graph.

EXPTIME, exponential memory, Cut off doubly exponential $N = 2^M 2^{2^{|A|+1}}$

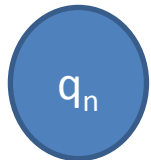
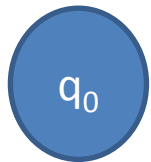
PSPACE-hardness:

Turing machine with polyspace M .



NFA (need at least $M+2$ agents)

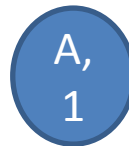
Control
state



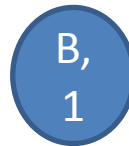
Head
position



positions



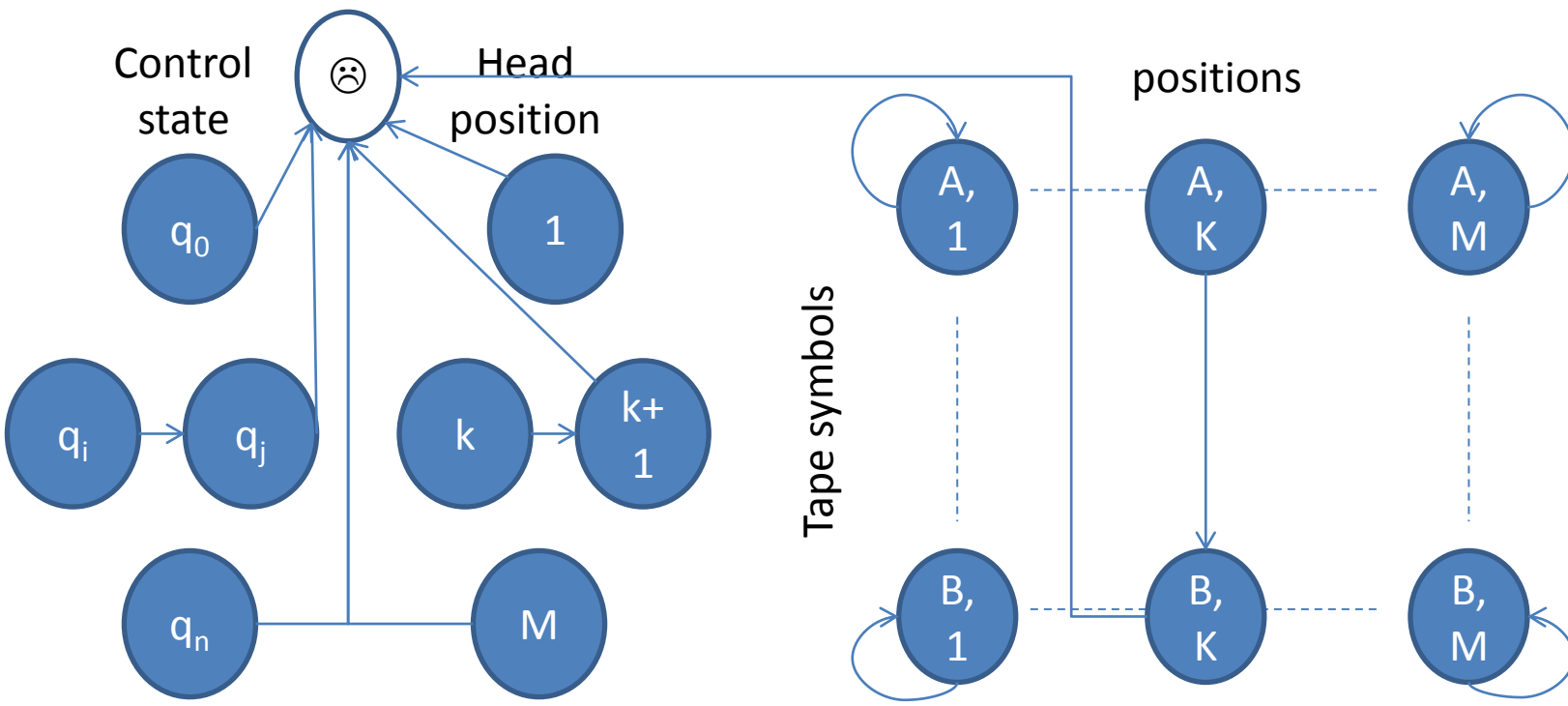
Tape symbols



PSPACE-hardness:

Rule r : $q_i, \text{head}==A$ then $\text{head}:=B, \text{pos}++$, goto q_j

M transitions $(t_{r,k})_{k=1..M}$

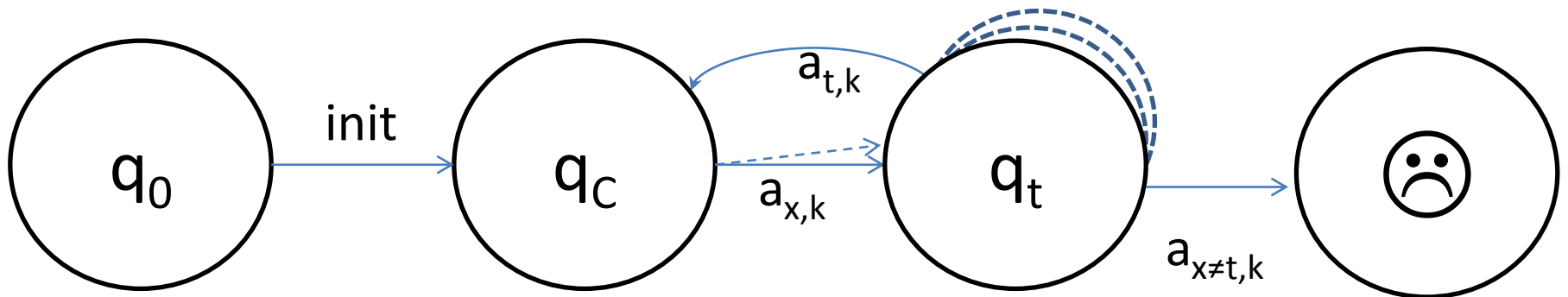


EXPTIME-hardness:

Alternating Turing machine with polyspace M.

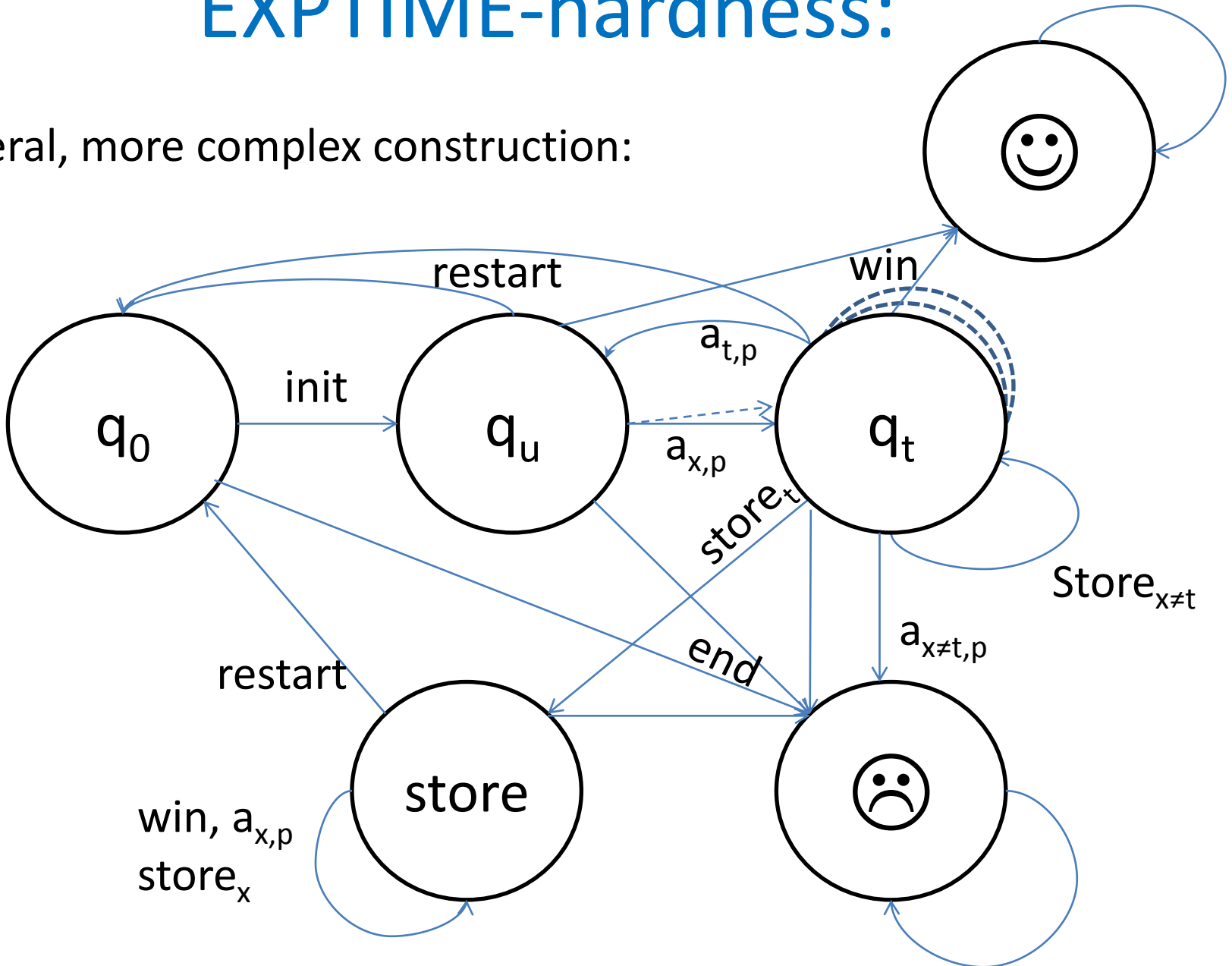
↓
NFA

Same NFA as before + assume 1 token in q_c after init.



EXPTIME-hardness:

In general, more complex construction:



Sum-up

Deciding whether for all N , there exists controller σ_N is
EXPTIME-complete.

If yes, we have a **symbolic** controller with **finite exponential memory**

If not, there exists $N_0 \leq 2^{2^{|A|^2}} 2^{2^{|A|+1}}$ such that:
controller up to N_0 and no controller after N_0

And there are cases where we have $N_0 = 2^{2^{|A|/2-10}}$