Example exam tasks: Basic formalisms, temporal logics, model checking, statecharts

dr. István Majzik
BME Department of Measurement and Information Systems

Theoretical questions

1. Argue if the formula A (X Stop V F Start) is a syntactically correct formula in CTL and in CTL*!
2. Describe the main idea of bounded model checking!
3. Describe the use of ROBDDs in model checking!
4. Define the term conflict in case of statecharts (which transitions are in conflict) and describe how conflicts are resolved!

Theoretical questions

Argue if the following LTL equivalences are correct or not:

1. (F Stop) V (F Start) ≡ F (Stop V Start)
2. G Stop ≡ not F (not Stop)

Argue if the following CTL equivalences are correct or not:

1. AF (Start $\vee$ Stop) ≡ (AF Start) $\vee$ (AF Stop)
2. AF (Start $\wedge$ Stop) ≡ (AF Start) $\wedge$ (AF Stop)
3. EF (Start $\wedge$ Stop) ≡ (EF Start) $\wedge$ (EF Stop)

Requirement formalization: Railway crossing

- We model the behavior of a railway crossing signal with the following atomic labels: {off, white, red}
- The behavior of the driver arriving at the crossing is modeled with the following atomic labels: {arriving, looking, stopping, crossing}
- Use LTL expressions to formalize the following requirements which apply to the behavior of the driver in every case:
  1. If the signal is off, the driver will be looking and then in the next moment either stopping or crossing.
  2. The driver will eventually cross the crossing.
  3. If upon arrival, the signal is red, the driver will not cross until the signal is white.

Requirement formalization: Server room

- We model the states of a server performing complex simulation with the following atomic labels: {off, waiting, warm-up, simulation}
- The air-conditioning system is modeled with the following atomic labels: {standby, normal, maximal}
- Use LTL expressions to formalize the following requirements which apply to the behavior of the server in every case:
  1. If in any moment the simulation is permormed with the air-conditioning system being in the stand-by state, then in the next moment, the server will move to the waiting state.
  2. Eventually, the simulation will be started.
  3. The simulation can be performed only if there has been a warm-up phase with the air-conditioning system in the normal state.

Model checking: Servers

- An IT system has two types of resource,a database server and an application server, both of which can be on or off.
- During normal operation, the resources are turned on and off simultaneously.
- Initially, both resources are off. The system is functional if both resources are on.
- If – in the functional state – the database server is turned off due to an error, the system becomes unfunctional. After this, the application server is also turned off, then the system is restarted by turning both servers on again.
- Tasks:
  1. Create a Kripke structure modeling the behavior of the system described above with regard to the states of the resources! Label the states with the following atomic propositions (based on the informal description):
     {initial, functional, unfunctional}
  2. Check if the following CTL formula holds if the initial state of the Kripke structure if the functional state:
     E(¬unfunctional U initial)
     Explain your solution!

Model checking: Computer engineering student

- The behavior of a computer engineering student is described by two predicates: drinking coffee or not, and sleeping or not.
- The student has three states:
  - During studying, he/she is drinking coffee and not sleeping;
  - After this he/she takes an exam, where he/she is not drinking coffe and not sleeping either;
  - After the exam, he/she rests, when he/she is sleeping and not drinking coffee.
- The initial state of the student is studying, which is continuous until taking an exam. He/she will not take an exam without studying and will study only after resting.
- Tasks:
  1. Create a Kripke structure modeling the behavior of the student described above with regard to drinking coffee and sleeping! Label the states with the following atomic propositions (based on the informal description): {resting, studying, taking an exam}
  2. Check if the following CTL formula holds on the model (the initial state is studying, as specified in the text):

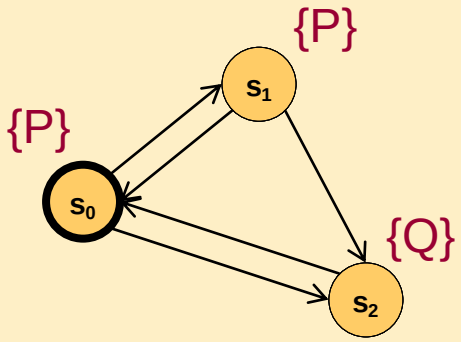
     E(¬taking an exam U rests)

Model checking with the tableau method

Consider the Kripke structure on the right.

Perform the model checking of the following formula with the tableau method:

$\neg$ (P U Q)

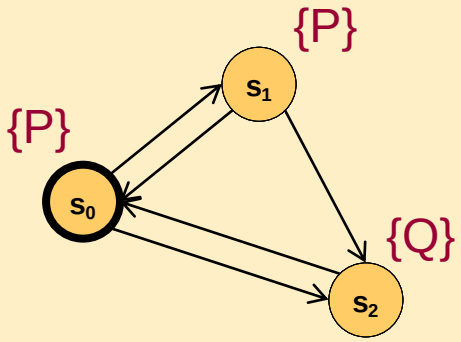


Model checking with the tableau method

Consider the Kripke structure on the right.

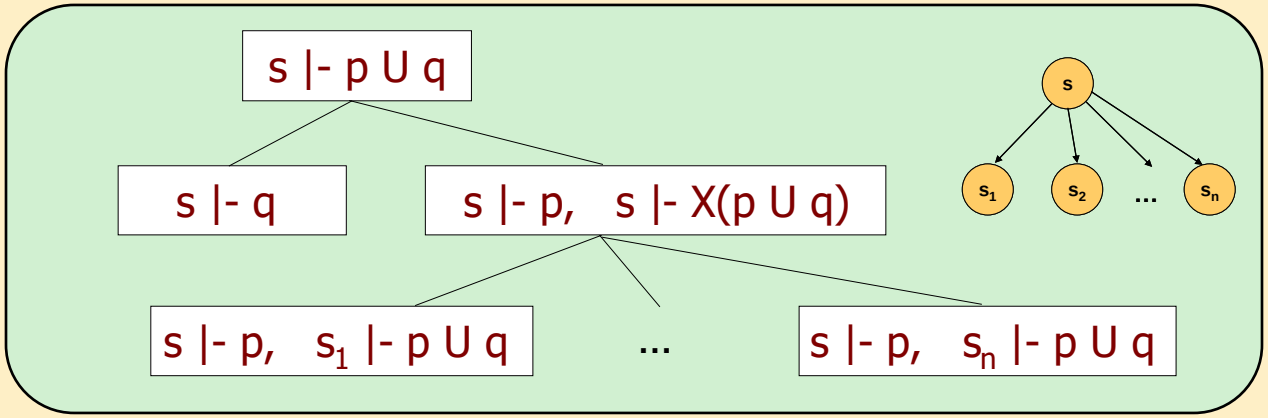
Perform the model checking of the following formula with the tableau method:

$\neg$ (P U Q)



Things to know:

- Negation (to look for counterexamples): (P U Q)
- Tableau rule: (P U Q) = Q $\vee$ (P $\wedge$ X(P U Q))

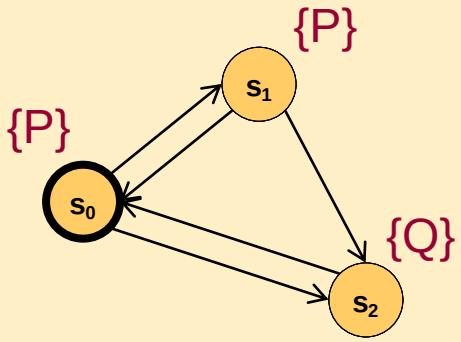


Model checking with the tableau method

Consider the Kripke structure on the right.

Perform the model checking of the following formula with the tableau method:

$$\neg (P \; U \; Q)$$



Things to know:

- Negation (to look for counterexamples): $(P \; U \; Q)$
- Tableau rule: $(P \; U \; Q) = Q \vee (P \wedge X(P \; U \; Q))$
- Contradicting branch if:
  - Atomic proposition does not hold in state
  - X operator with deadlock
  - Cycle with P, but without Q
- Satisfying branch (giving counterexamples) if:
  - Only atomic propositions and all of them hold in the state
  - Cycle without contradiction

ROBDD: Building ROBDD

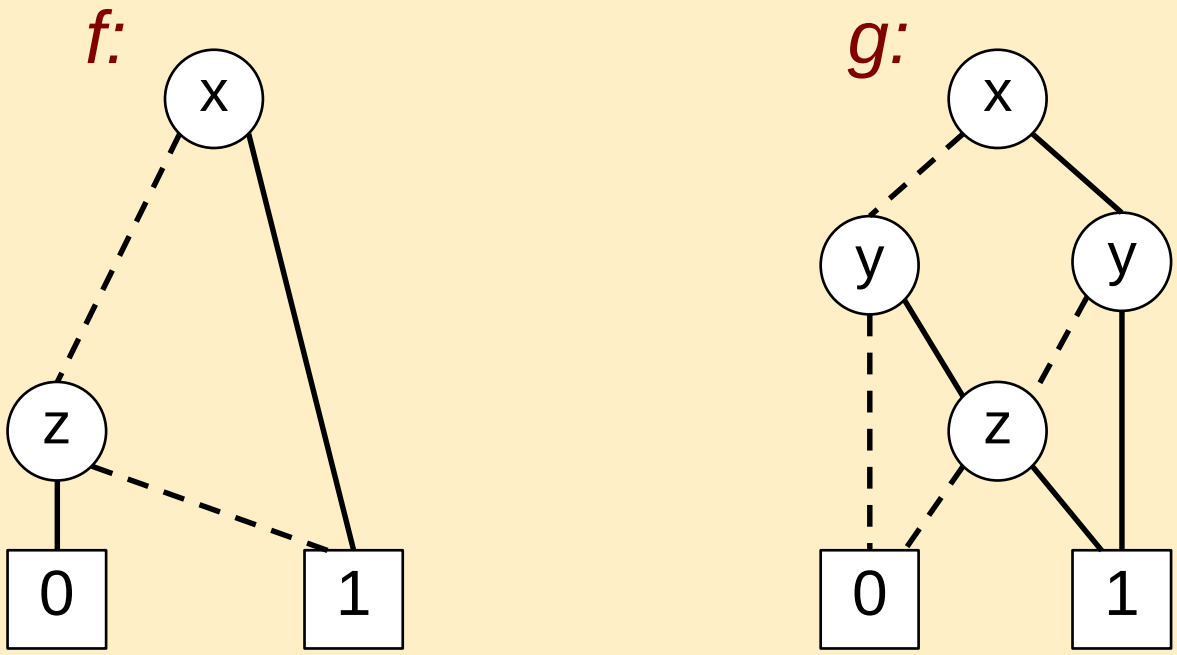
Consider the following Boolean function *g*:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

1. Construct the decision tree representing *g*! Use the variable ordering used in the table: *x*, *y*, *z*.
2. Based on this, construct the reduced ordered binary decision diagram (ROBDD) representation of *g*!
3. Give the algebraic form of the function!

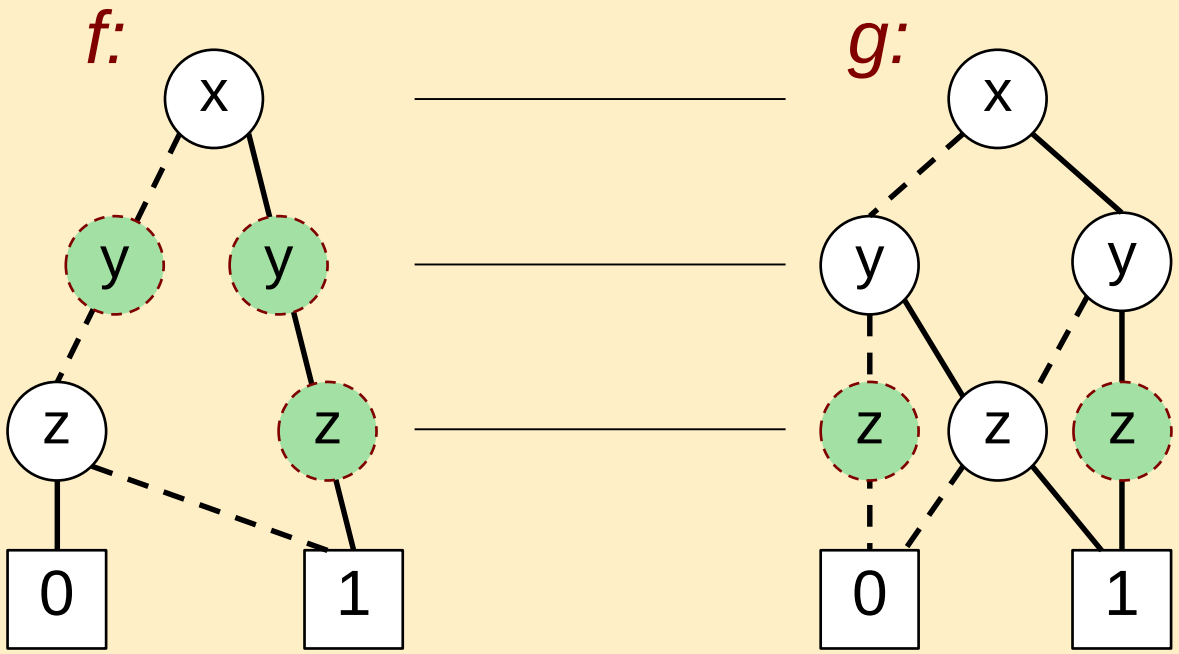
ROBDD: Operations on functions

Consider the following functions f and g given in ROBDD form. Construct the ROBDD representing $f \wedge g$!



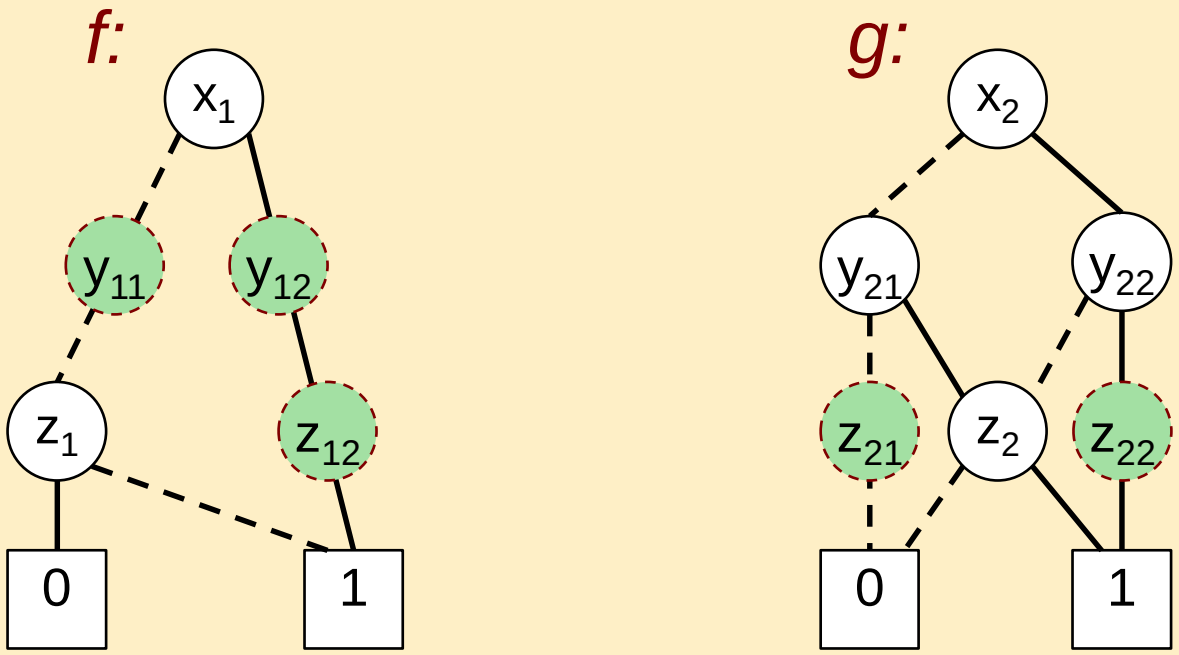
ROBDD: Operations on functions

Consider the following functions f and g given in ROBDD form. Construct the ROBDD representing $f \wedge g$!

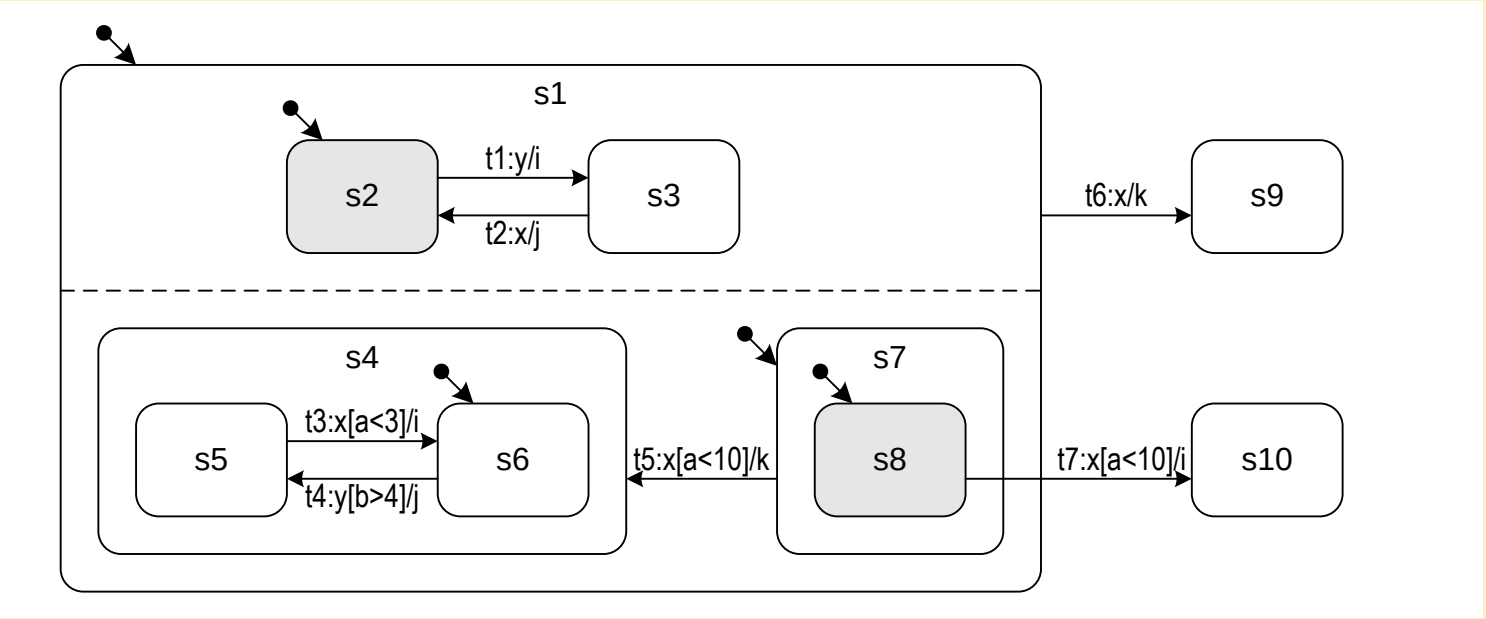


ROBDD: Operations on functions

Consider the following functions f and g given in ROBDD form. Construct the ROBDD representing $f \wedge g$!



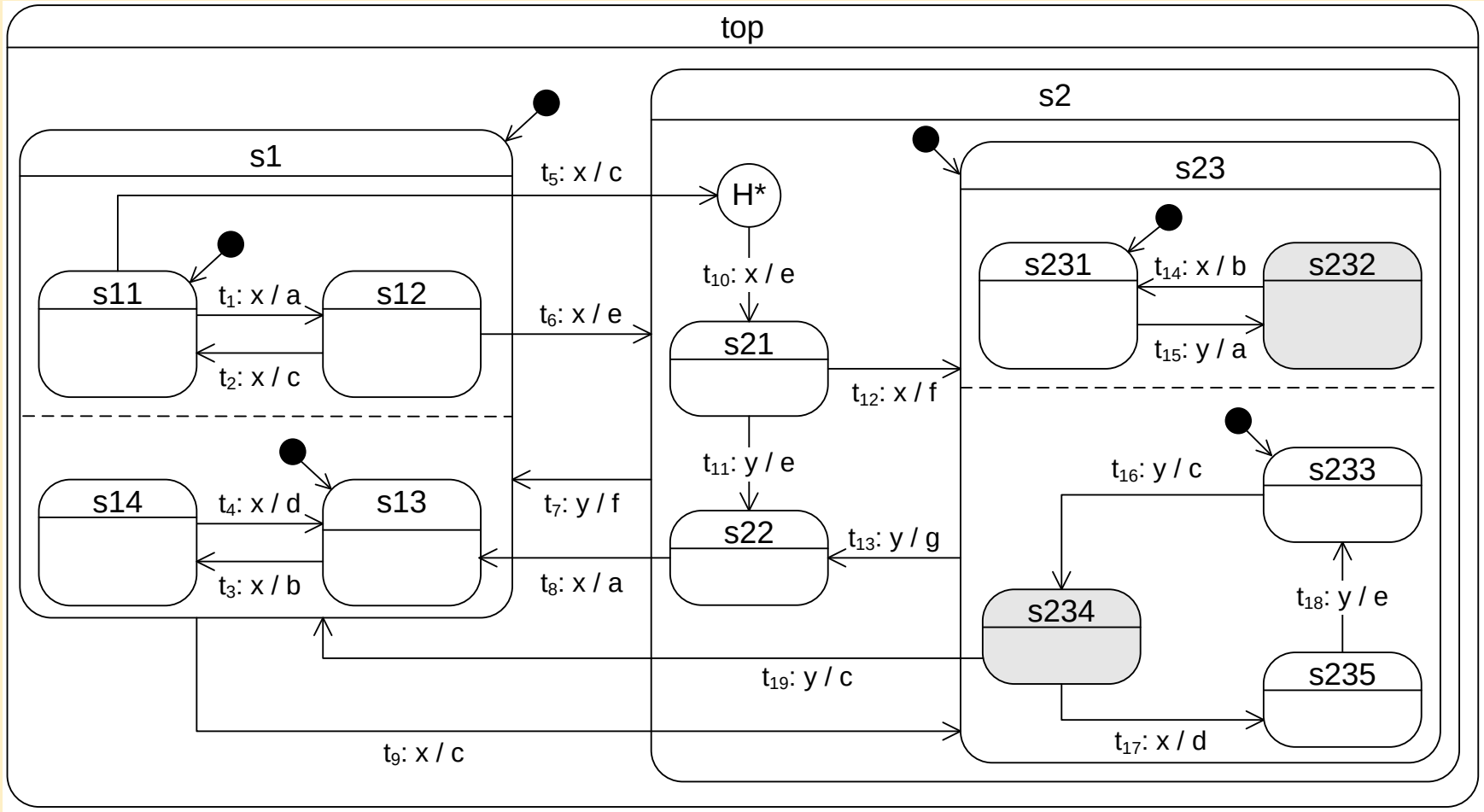
Statecharts 1.



In the initial state configuration, the value of variable *a* is 8 and an event "*x*" occurs.

1. Which transitions are enabled?
2. Which enabled transitions are in conflict?
3. What is the set of fireable transitions?
4. What is (are) the next stable state configuration(s)?
5. What actions are executed and in what order?

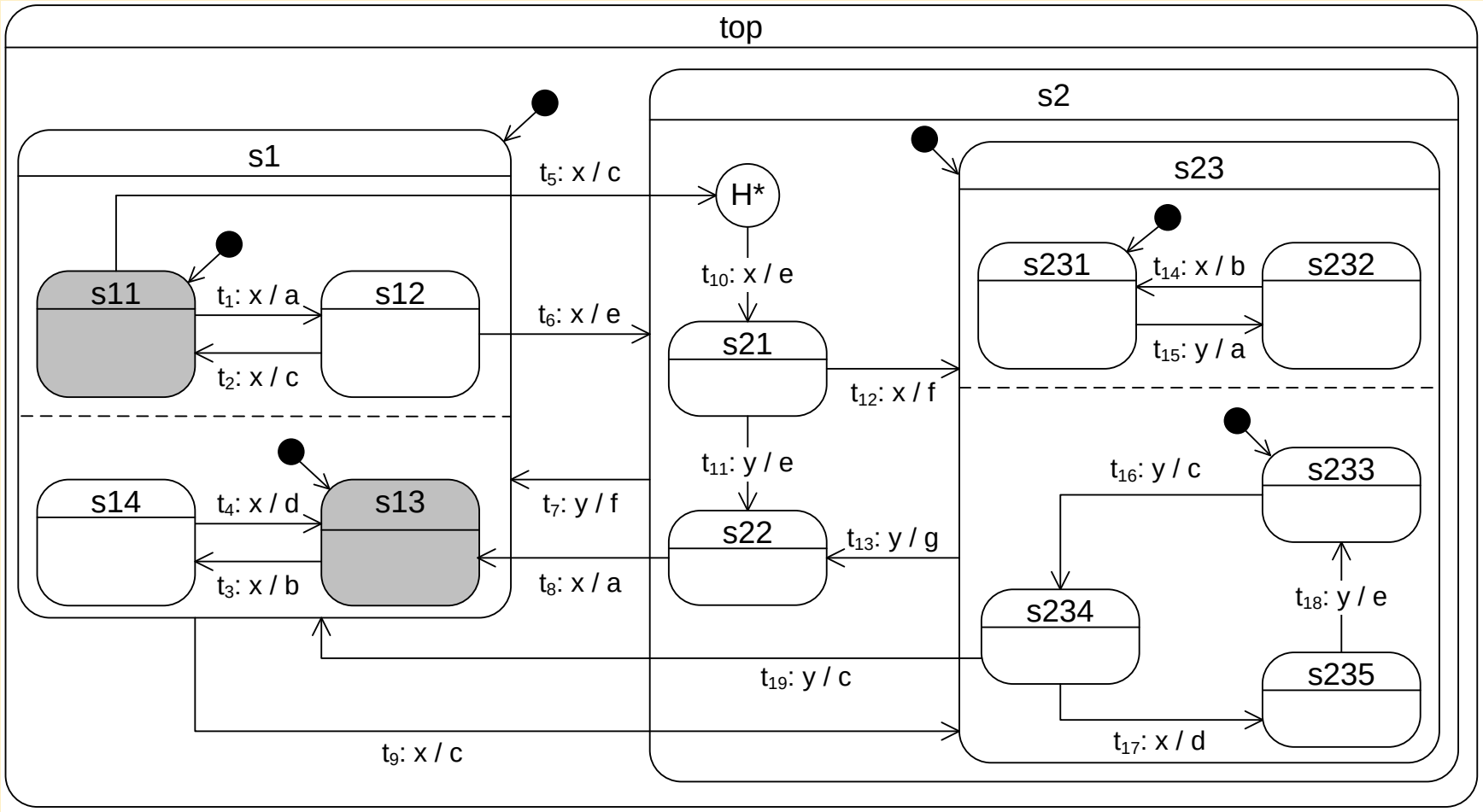
Statecharts 2.1.



In the state configuration {top, s2, s23, s232, s234} the event *y* is passed by the scheduler.

- What will be the new state configuration?

Statecharts 2.2.



After this, the event *x* is passed by the scheduler.

1. Which transitions are enabled, in conflict and fireable?
2. What will be the new state configuration? What actions are executed?