Feedback — Quiz week 2

Help Center

Thank you. Your submission for this quiz was received.

You submitted this quiz on **Mon 26 Oct 2015 10:33 PM CET**. You got a score of **4.10** out of **5.00**.

Please note that quiz questions might change between attempts!

Question 1

Based on the same data, different views can be taken to create process models with different perspectives.

Given the data below, which column/attribute should be selected as the case identifier to analyze in which order each of the doctors executes activities?

patient	activity	timestamp	doctor	age	cost
5781	make X-ray	23-1-2014@10.30	Dr. Jones	45	70.00
5541	blood test	23-1-2014@10.18	Dr. Scott	61	40.00
5833	blood test	23-1-2014@10.27	Dr. Scott	24	40.00
5781	blood test	23-1-2014@10.49	Dr. Scott	45	40.00
5781	CT scan	23-1-2014@11.10	Dr. Fox	45	1200.00
5833	surgery	23-1-2014@12.34	Dr. Scott	24	2300.00
5781	handle payment	23-1-2014@12.41	Carol Hope	45	0.00
5541	radiation therapy	23-1-2014@13.57	Dr. Jones	61	140.00
5541	radiation therapy	23-1-2014@13.08	Dr. Jones	61	140.00
...	…	…	…	…	…

Your Answer		Score	Explanation
◉ case id = doctor	✔	0.50	The doctor is indeed the case that 'flows through' different activities

○ case id = patient	
○ case id = activity	
○ case id = timestamp	
Total	0.50 / 0.50

Question Explanation

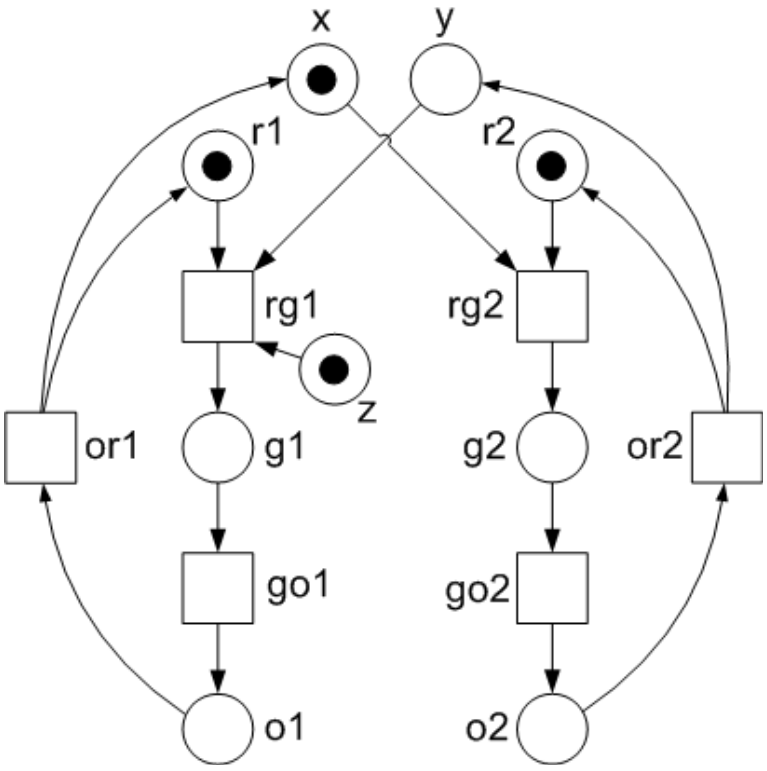
Doctors are the 'object' or case that executes activities in different orders. At the same time, the doctor is also the resource.

For completeness, the full mapping would be: activity = activity, timestamp = timestamp, patient=data, age=data and cost=data.

More information on how to map data to event log artefacts is discussed in the 'Event Logs and Process Models' lecture in week 2.

Question 2

Please indicate all **reachable dead markings** of the marked Petri net shown below.



Your Answer		Score	Explanation
☑ [x, r1, r2]	✖	0.00	This marking is not dead because the transition 'rg2' is still enabled since there is a token in places 'x' and 'r2'.

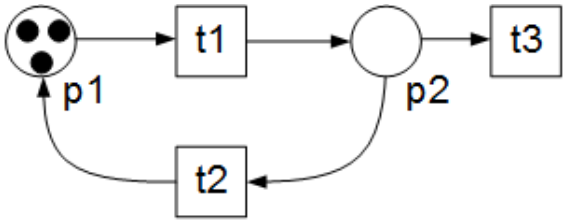
☐ [x, y, r1, r2]	✔	0.12	This marking is not reachable because there can never be a token in places 'x' and 'y' at the same time.
☐ [x, z, r1 ,r2]	✔	0.12	This is the initial marking, however it is not dead since transition rg1 is enabled since there are tokens in places 'r1', 'x' and 'z'.
☐ [y, r1, r2]	✖	0.00	This is indeed the only dead marking that is reachable by firing the following execution sequence is rg2, go2, or2, rg1 (which removes the only token in place z), go1, or1, rg2, go2, or2.
Total		0.25 / 0.50	

Question Explanation

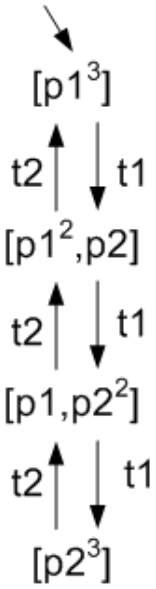
A dead marking is a marking of the Petri net at which no further transition is able to fire. The only possible execution sequence is rg2, go2, or2, rg1 (which removes the only token in place z), go1, or1, rg2, go2, or2. At this point the marking is y, r1, r2, but there is no token in z blocking rg1 to fire, and hence no transition is able to fire.
More about Petri nets, firing sequences and final markings is explained in week 2 in the lecture 'Petri Nets (1/2)'.

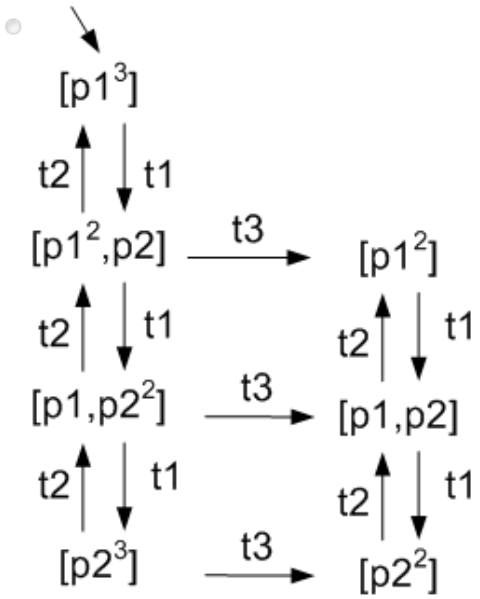
Question 3

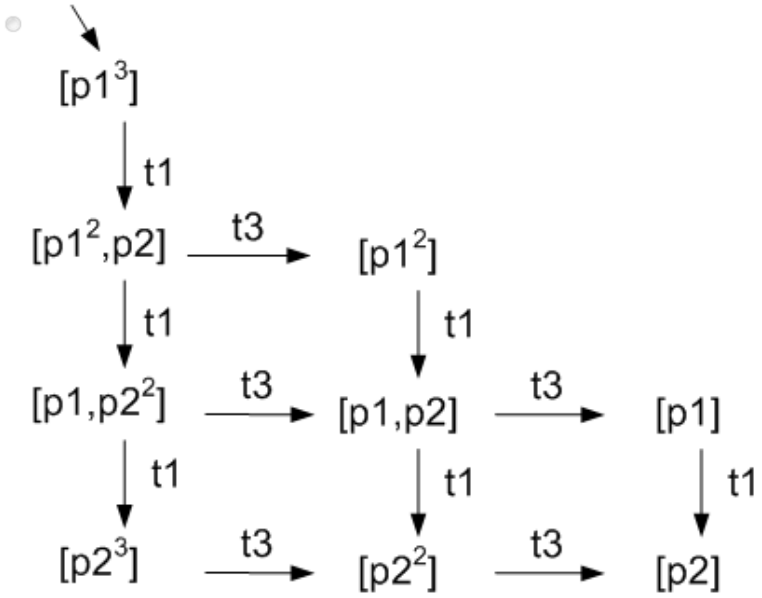
Given the marked Petri net below, which of the following is the correct reachability graph?

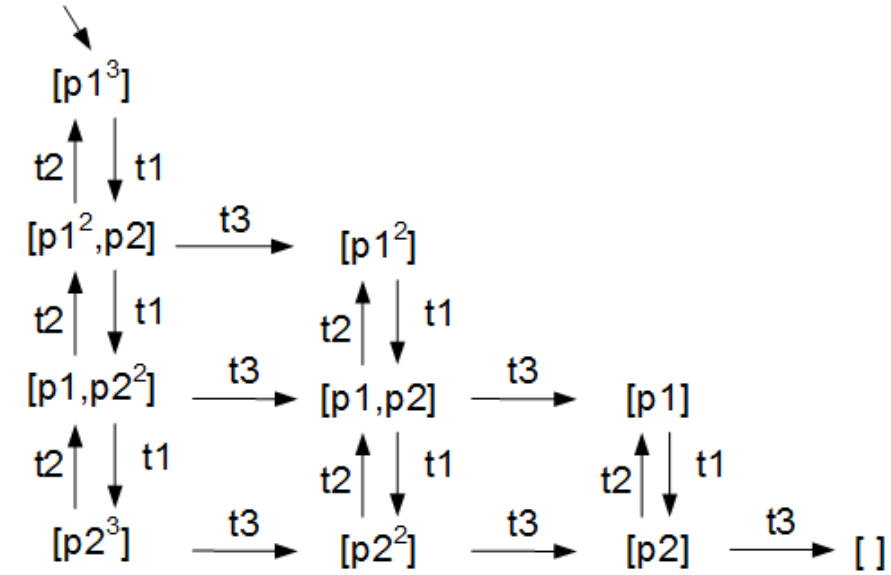


Your Answer **Score** **Explanation**









✔ 0.50

In the initial marking, only t1 is possible, which moves one token to p2. Now t1, t2 and t3 are enabled. When we fire t2 we return to the initial marking. When t3 is fired we remove a token from

	the Petri net. In general t1 and t2 can fire in turn, moving tokens between p1 and p2. However, when t3 is fired 3 times the Petri net is empty and nothing is possible any more.
Total	0.50 / 0.50

Question Explanation

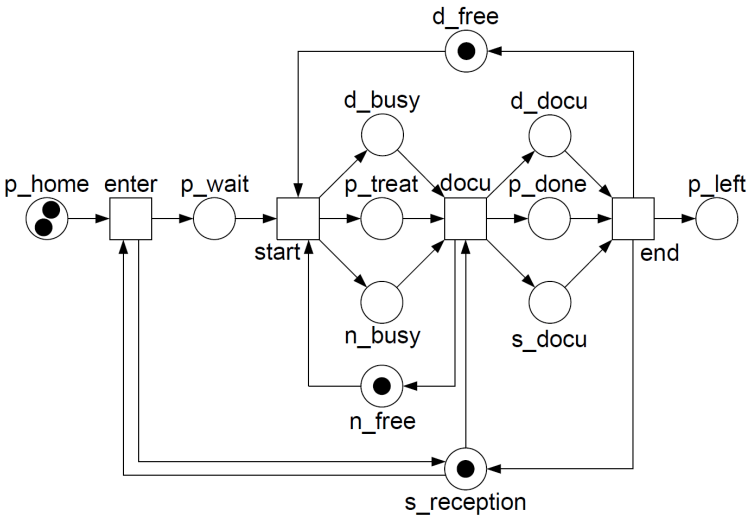
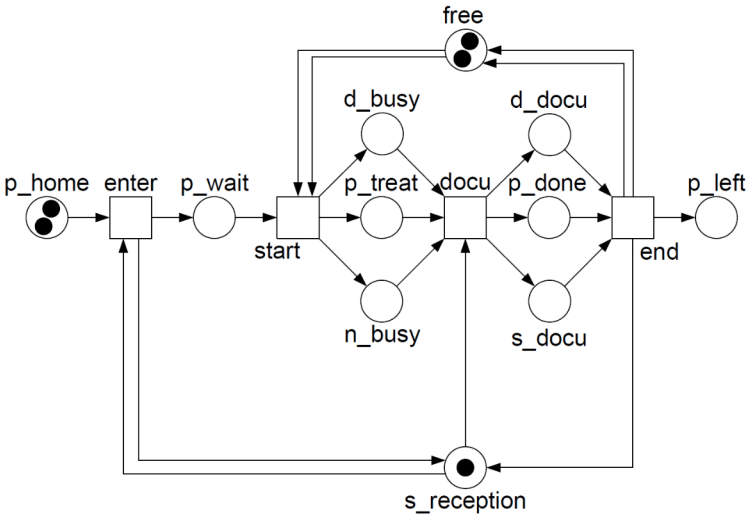
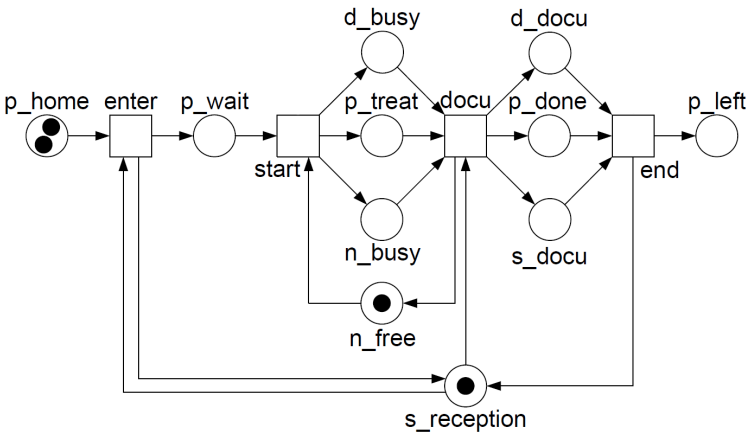
How to construct a reachability graph is explained in more detail in the lecture 'Petri Nets (2/2)' of week 2.

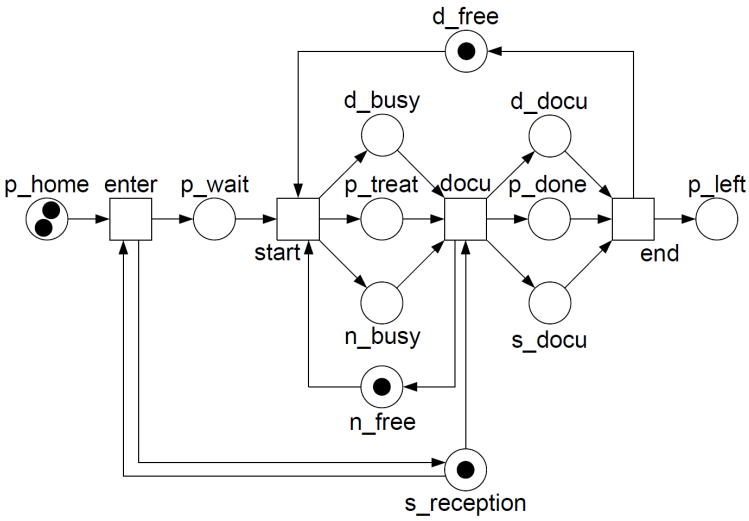
Question 4

Which of the following Petri nets best describes the following process of treatment of a patient at a dentist?

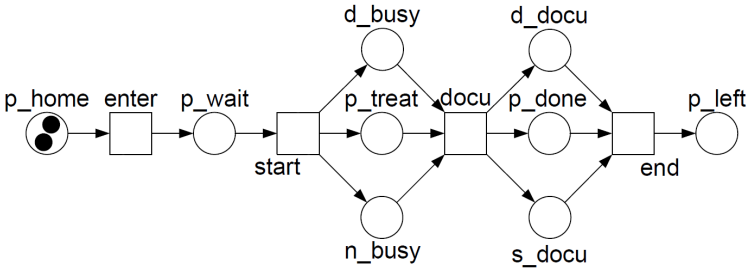
If we look at the patient separately, then each patient can be in one of five states: the patient is at home (state "p_home"), the patient is sitting in the waiting room (state "p_wait"), the patient is treated (state "p_treat"), the treatment is finished (state "p_done"), and the patient has left the practice (state "p_left"). There are four events: the patient enters the practice ("enter"); the treatment starts ("start"); the treatment is documented ("docu"); and the treatment ends ("end"). The dentist can be in three states: "d_free", "d_busy", and "d_docu". There are three respective events: "start", "docu", and "end". The nurse can be either in state "n_free" or in state "n_busy". There are two events where the nurse is involved: "start" and "docu". Finally, there is also a secretary involved. Either she sits at the reception (state "s_reception") or she helps the dentist documenting the treatment and writing a prescription (state "s_docu"). For the documentation, there are two events relevant: "docu" and "end". Furthermore, when a patient enters the practice, the secretary is taking care of the reception.
Suppose that initially there are two patients at home, the dentist and the nurse are in state "free",

and the secretary is sitting at the reception. Note that you can click on the process models shown to enlarge them.

Your Answer	Score	Explanation
		
		
		
	✖ 0.00	This Petri net is not correct since the secretary is required when



executing the "docu" activity, it is not made available again (e.g., put back in the place "s_reception"). This means that only one patient can be treated after which the Petri net can deadlock with patients waiting for documentation for instance.



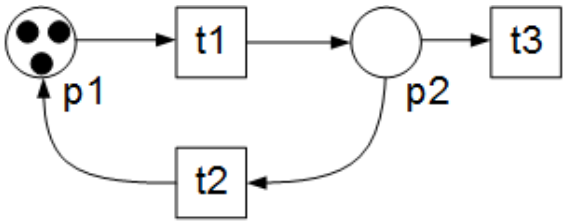
Total	0.00 / 0.50

Question Explanation

More examples of how to model certain processes based on their descriptions are discussed in the lecture 'Petri Nets (2/2)' in week 2.

Question 5

Given the Petri net below, which of the following properties hold:



Your Answer		Score	Explanation
☐ All transitions are dead (i.e. can never fire).	✔	0.10	Not all transitions are dead since in the initial marking transition t1 is enabled.

☑ The marked Petri net has a reachable dead marking.	✔	0.10	The Petri net contains a deadlock since t3 can remove all tokens from the Petri net after which no further transition is enabled.
☑ All places are bounded.	✔	0.10	All places, and therefore the Petri net, are indeed bounded since at no point in time p1 or p2 contains more than 3 tokens.
☑ All transitions are live	✖	0.00	None of the transitions are live since there is a reachable marking, the empty marking, after which none of the transitions are enabled.
☐ All places are safe.	✔	0.10	A place is safe when it is 1-bounded, which is not the case here since both places are 3-bounded.
Total		0.40 / 0.50	

Question Explanation

Different Petri net properties are discussed in 'Transition Systems and Petri Net Properties' of week 2.

Question 6

Workflow nets are a special class of Petri nets. What are the requirements for a workflow net?

Your Answer		Score	Explanation
☑ The Petri net has one source place.	✔	0.08	
☑ The Petri net has one sink place.	✔	0.08	
☐ Each transition in the Petri net has at most 2 input places and at most 2 output places.	✔	0.08	
☐ Each transition in the Petri net has a unique label.	✔	0.08	
☐ Each place in the Petri net has at most 2 input transitions and 2 output transitions.	✔	0.08	
☑ All nodes (except source and sink) in the Petri net are on a path from source to sink.	✔	0.08	
Total		0.50 / 0.50	

Question Explanation

Workflow nets are discussed in the lecture 'Workflow Nets and Soundness' of week 2.

Question 7

Which of the following properties always hold for **sound workflow nets**?

Your Answer		Score	Explanation
☐ The workflow net does not allow for loops.	✔	0.07	This is not a necessary requirement for soundness.
☑ When the workflow net reaches the desired final state, marking the sink place, no other places are marked.	✔	0.07	
☐ The workflow net is unbounded.	✔	0.07	This is not a requirement since bounded workflow nets can also be sound.
☑ The workflow net is safe.	✔	0.07	
☐ Two transitions cannot be enabled at the same time.	✔	0.07	This is not a necessary requirement for soundness since this would reduce all choice and parallel construct.
☑ The workflow net can always reach the desired final state marking the sink place.	✔	0.07	
☑ The workflow net has no dead parts.	✔	0.07	
Total		0.50 / 0.50	

Question Explanation

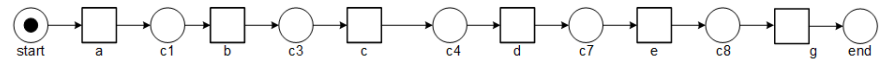
The requirements for soundness are explained in the lecture 'Workflow Nets and Soundness' of week 2.

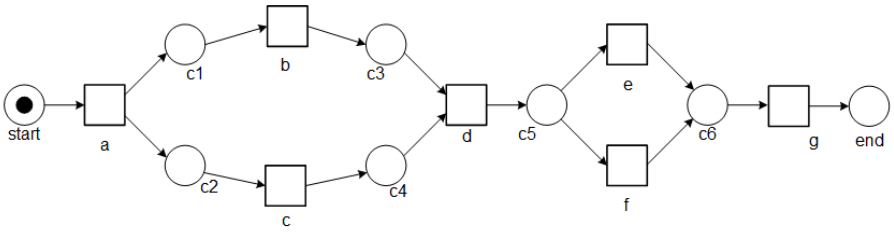
Question 8

Given the event log L = [< a,b,c,d,e,g > ^{99},< a,c,b,d,f,g >^{1}], which result does the Alpha algorithm produce?

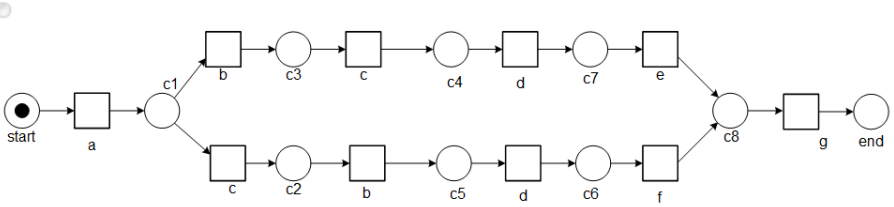
Ideally you would execute the Alpha algorithm on the event log mentioned above. Otherwise, think of the differences between the process models and why the Alpha algorithm can (not) discover a certain process model.

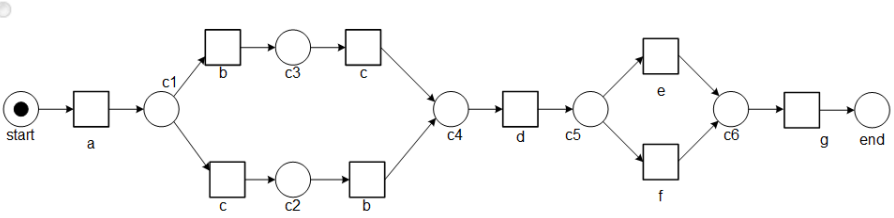
Your Answer	Score	Explanation





✔ 0.50





Total	0.50 / 0.50

Question Explanation

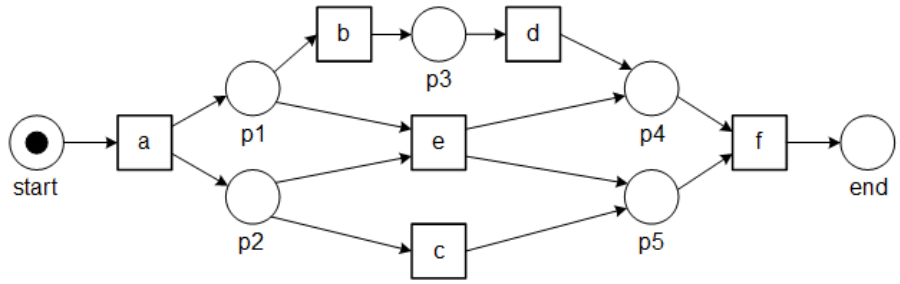
The Alpha algorithm will detect that all traces start with activity 'a' and that activities 'b' and 'c' are in parallel and are always followed by activity 'd'. Activity 'd' is followed by either activity 'e' of 'f', which are always followed by 'g'. This results in a Petri net with 'b' and 'c' in parallel and 'e' and 'f' in a choice construct.
The Alpha algorithm is explained in more detail in the lecture 'Alpha Algorithm: A Process Discovery Algorithm' of week 2.

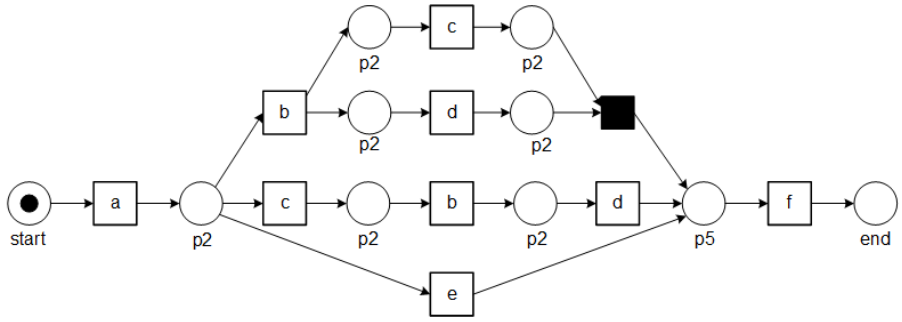
Question 9

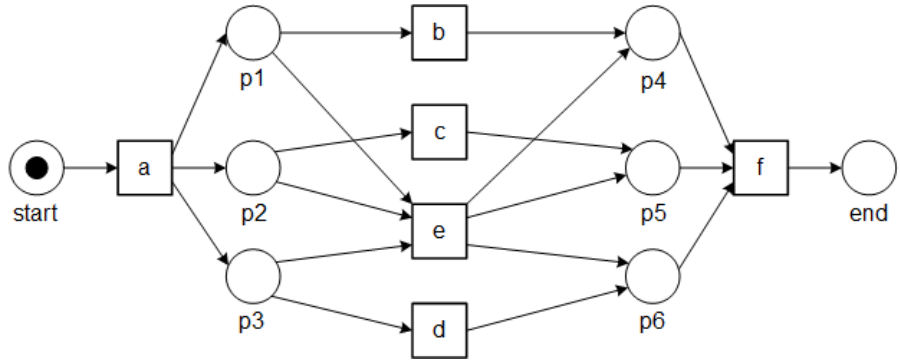
Given the event log L = [< a,b,c,d,f > , < a,b,d,c,f > , < a,c,b,d,f > , < a,e,f >], which result does the Alpha algorithm produce?

Ideally you would execute the Alpha algorithm on the event log mentioned above. Otherwise, think of the differences between the process models and why the Alpha algorithm can (not) discover a certain process model.

Your Answer	Score	Explanation
	✔ 0.50	







Total	0.50 / 0.50

Question Explanation

The Alpha algorithm does not have enough 'evidence' to conclude that activities 'b', 'c' and 'd' are in parallel, since it has never observed activity 'd' before 'b'. It did however detect that activities 'b' and 'c' are in parallel and 'c' and 'd'. It also found the exclusive choice between activities 'b', 'c' and 'd', and activity 'e'.

Question 10

Which of the following are properties of the Alpha algorithm?

Your Answer		Score	Explanation
☑ It cannot discover loops of length 1.	✔	0.05	When an activity is repeated the Alpha algorithm cannot detect this.
☑ It may discover implicit/redundant places.	✔	0.05	Some places in the Petri net discovered by the Alpha algorithm can be removed without

			changing the behavior.
☑ It cannot discover loops of length 2.	✔	0.05	When two activities are repeated the Alpha algorithm cannot detect this.
☑ It can only discover unique activity labels, e.g. an activity label occurs exactly once in a discovered process model.	✔	0.05	The alpha algorithm is not able to duplicate activity names when they are used in a different context.
☐ It may create deadlocked/unsound models.	✖	0.00	There is no overall check whether the constructed process model is free of certain errors such as deadlocks.
☑ It is sensitive to noise in the event log.	✔	0.05	The Alpha algorithm does not ignore anything and therefore is highly sensitive to noise.
☐ It cannot detect choices.	✔	0.05	Choices are easily detected by the Alpha algorithm.
☐ It cannot detect parallelism.	✔	0.05	When the event log contains 'enough' behavior (in essence, is complete) then the Alpha algorithm has no problems in discovering parallelism.
☐ It requires user input besides the event log.	✔	0.05	The Alpha algorithm is one of very few process discovery algorithms that require no user input at all (this is not necessarily a good thing).
☐ It needs all possible traces or execution sequences to be able to rediscover a process model.	✔	0.05	The Alpha algorithm does not require all possible behavior to be observed. If k activities are in parallel for instance, the Alpha algorithm only requires k(k-1) observations to rediscover this parallel construct.
Total		0.45 / 0.50	

Question Explanation

The limitations of the Alpha algorithm are explained in more detail in the lecture 'Lecture 27: Alpha Algorithm: Limitations' of week 2.