CS445 / CS645 / ECE451
Fall 2018 — Final Exam

6 December 2018, 7:30pm–10:00pm

Instructor: Daniel M. Berry

Time allowed: 2.5 hours = 150 minutes

No aids allowed (*i.e.,* closed book).

Answer all of the questions on this exam paper.

There are 10 questions for a total of 150 marks.

Plan your time wisely: 1 minute per mark

Your Name and Student Number

In the immortal words of Daniel Berry of Earth,

Knock our socks off!

Q1		scaled to 28
Q2		scaled to 8
Q3		scaled to 10
Q4		scaled to 8
Q5		scaled to 16
Q6		scaled to 8
Q7		scaled to 16
Q8		scaled to 10
Q9		scaled to 26
Q10		scaled to 20
TOTAL		scaled to 150

In this exam, a short underscore of 1 inch (= 2.54 cm) should be filled with one word. A long underscore of 3 inches (= 7.63 cm) should be filled with a phrase consisting of one to several words. In the either case, if you cannot think of exactly the right number of words, then give the best answer that you can and we'll give it as many marks as we can, possibly even full credit! If you cannot even think of just words to fill in, then write an answer as a sentence, and we'll give it as many marks as we can.

"(complete the sentence)" means that in the space following the current line, you are to complete the sentence that was started and interrupted with the "(complete the sentence)"

If you are asked a question, which ends with a "?", you are to answer that question in the space following the current line.

In this exam, if you are asked for a simple answer, you need not justify it, unless you are also asked explicitly "Why?". However, you may always write down assumptions that can help us give you partial credit.

If an exam question directs you to list, describe, write, explain, draw, mark, put, change, modify, or anything like that, just do so.

In the exam questions,
"CBS" means "computer-based system".
"SW" means "software".
"NFR" means "non-functional requirement", a.k.a. "quality attribute".
"RE" means "requirements engineering".
"SRS" means "software requirements specification, written according to some standard, e.g., IEEE".
"UM" means "user's manual".

Note the difference between a serif front like that used for *this* sentence
and a sans serif font like that used for *this* sentence.

1. **[28 total marks]** RE Reference Model

(a) When one is considering the reliability of a CBS, the definition of the expected cost of an event X is

$$\textit{expectedCost}(X) = \textit{probability}\ (X) \times \textit{cost}\ (X)$$

An important event that affects reliability is that the domain assumption necessary for a CBS to entail its requirements fails to hold.

Suppose you have two different CBSs, one specified by S_1 and the other specified by S_2. Under slightly different domain assumptions, D_1 and D_2, respectively, both CBSs entail the same requirements R. That is,

$D_1, S_1 \vdash R$
$D_2, S_2 \vdash R$

How do you decide which of the CBSs is the better one to use?

(b) On the same theme, suppose you have an aircraft with some automated flying that is under control of a so-called autopilot, that is described by $D, S \vdash R$.

In which of D, S, and R is a description of the behavior of the autopilot, its features and what the features do?

In which of D, S, and R are the instructions to the pilots on how to operate the aircraft in general, how to engage and use the autopilot, and what hardware and other problems to watch out for?

Suppose that the aircraft manufacturer produces a new version of *this* aircraft with improved safety with a new autopilot. Describe the new version, which satisfies the same R in terms of a variation of the formula $D, S \vdash R$.

As a result of the change in the aircraft, what must be done to the pilots who fly the aircraft?

In claiming that the new version of the aircraft is safer than the old version, what is the manufacturer saying about the expected costs of failure of the domain assumptions to hold?

(c) Let

$D =$ There are birds in the air
$S =$ Aircraft made of aluminum flies through the air
$R =$ Flying aircraft does not crash

One cannot prove

$D, S \vdash R$

because if an aircraft made of aluminum hits a bird while flying through the air, the aircraft will break apart and crash.

Give one possible change to each of D, S, and R to get D', S', and R', respectively, such that one *is able to prove*.

$D', S \vdash R$
$D, S' \vdash R$
$D, S \vdash R'$

$D' =$

$S' =$

$R' =$

Explain how D' is in some sense the same as R'.

2. **[8 total marks]** Use Cases and Scenarios

   When we covered use cases, I mentioned that (Sindre and Opdahl) and Alexander have proposed *misuse cases* to capture use cases that a system must be protected against, that you do not want to happen at all, e.g., a security breach.

   I know that we did not spend much time talking about them in class, but I would like to see how you deal with a new version of a problem that you are very familiar with.

   List 4 misuse cases for your group's IEKS in addition to the one that has been provided as a hint.

   Break car window to gain entry.

   Give a meaningful name to the actor that performs all of these misuse cases, a name that reminds the analyst of the negative nature of these misuse cases.

   Are all of these misuse cases necessarily performed by someone that the Prius should be protected from? ("Yes" or "No")

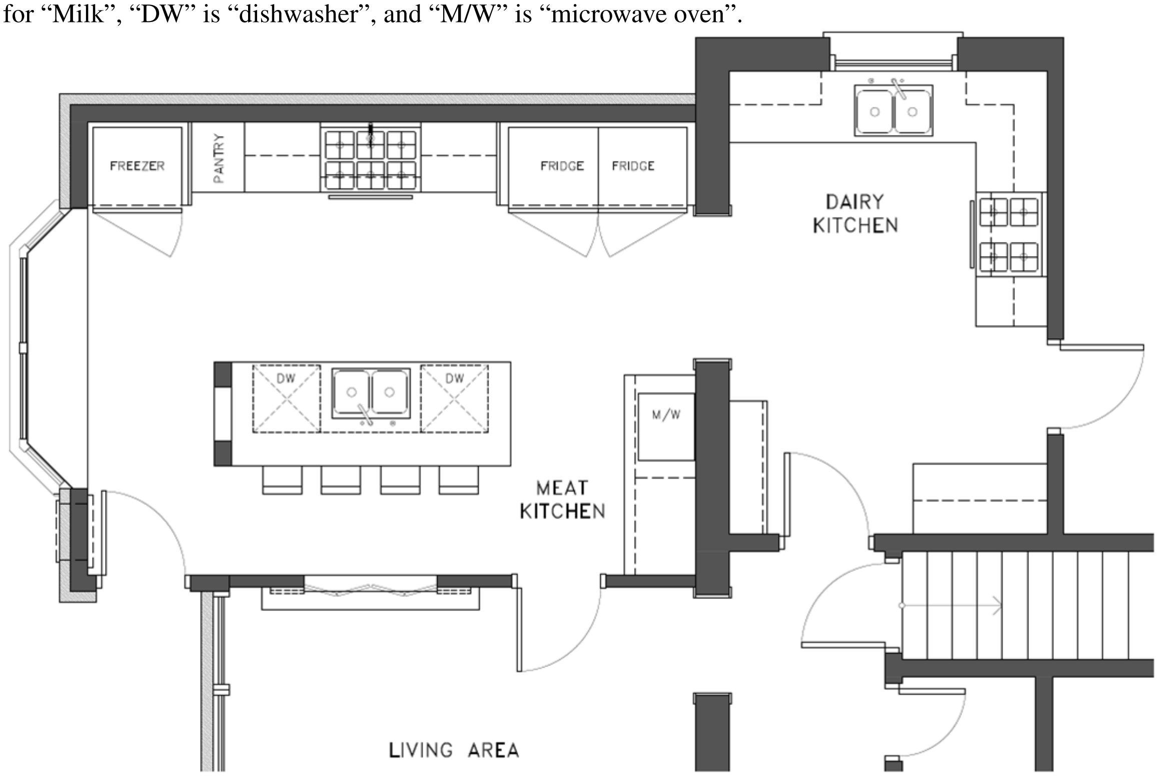
   If your answer was "No", then explain *who* should be allowed to do some of these misuse cases, and why.

3. **[10 total marks]** Elicitation

A main requirement for a kosher kitchen is that it permits and does not prevent (There is no way for any kitchen to force someone to be kosher if he or she does not want to be.) either temporal or spatial separation of milk products and meat products during food preparation, cooking, eating, and cleaning up. A synagogue kitchen may have to be preparing a milk and a meat meal at the same time, e.g., a grilled cheese lunch and a slow-cooked meat stew for dinner the same day. So a synagogue kitchen needs to implement spatial separation of milk products and meat products. Thus, we have what is known as Requirement R for the rest of this question!

> "A synagogue kitchen must permit and not prevent spatial separation of milk products and meat products during food preparation, cooking, eating, and cleaning up."

Consider the floor plan P for a synagogue kitchen that the Rabbi sent to Berry. Note that "Dairy" is a synonym for "Milk", "DW" is "dishwasher", and "M/W" is "microwave oven".



Give 4 reasons that P fails to meet requirement R.

P has some defects that make the kitchen hard to use or potentially damaging to the food, independent of any consideration of being kosher. Give 3 of these defects.

Had the architect explored typical kitchen scenarios such as (1) "putting groceries from the supermarket into the refrigerator" and (2) "getting milk from refrigerator while cooking a meat meal" with the clients? ("Yes" or "No")

The Rabbi told Berry that the kitchen would be built according to the plan P, and not the one that Berry offered. Berry told the Rabbi that the cost to fix the kitchen to meet Requirement R after it was built according to P was about \$20K, given that the cost of building to P was about \$20K.

How did Berry arrive at the \$20K estimate the cost for fixing the already built kitchen?

4. **[8 total marks]** Requirements Determination is Unstoppable

   Recall the distinction between a *scope determininG Requirement*, GR, and a *scope determineD Requirement*, DR, for a CBS.

   Define each.
   A scope determininG Requirement GR is one that

   A scope determineD Requirement DR is one that

   For the IEKS, give one example of each.
   GR:

   DR:

   Once you have picked the scope determininG Requirements GRs for an agile build B, there are two approaches for discovering the scope determineD Requirements DRs of B.

   AgileBuilds: discovering the scope determineD Requirements DRs of B in a series of follow-on agile builds, and

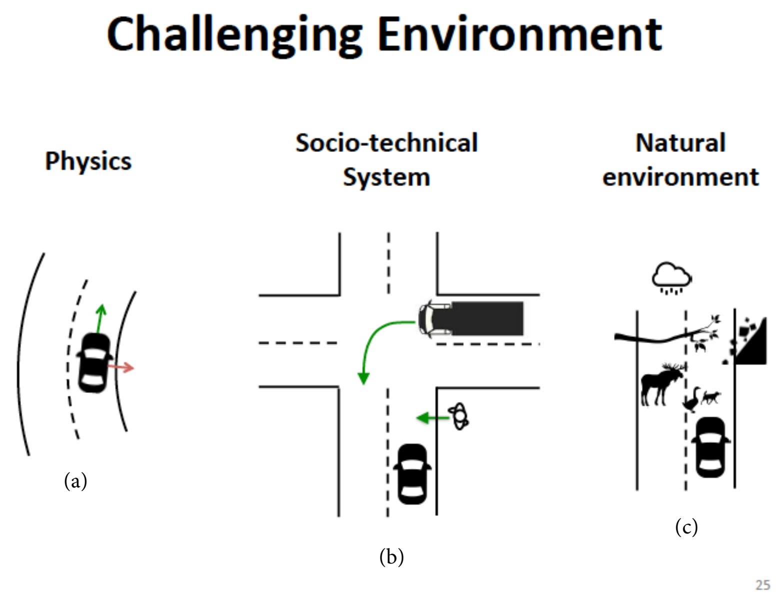
   UpfrontRE: discovering the scope determineD Requirements DRs of B in an upfront exhaustive requirements analysis that is completed before starting the coding of the build B

   Which of the approaches, **Agile** or **UpfrontRE**, is more likely to deliver in less time, some running build, albeit not very robust?

   Which of the approaches, **AgileBuilds** or **UpfrontRE**, is more likely to deliver in less time, the running build B' that has all the scope determineD Requirements DRs of B implemented?

5. **[16 total marks]** Guest and Graduate Student Lectures

(a) Consider this slide from Prof Czarnecki's talk on 18 October.



It is the second last of the slides that he presented.

"(a)", "(b)", and "(c)" have been added to the slide to identify 3 situations that affect the requirements for self-driving vehicles.

Assume that R is

> Everyone in a vehicle V gets to where he or she wants to go
> (1) without anyone
> (i) in V or
> (ii) elsewhere on or near the roads being driven on by V
> being injured or killed and
> (2) without V being damaged.

Pick only *one* of these situations to answer below, and answer it under the situation's label. Describe how your selected situation affects S and D in $D, S \vdash R$. That is, (1) explain what knowledge about the situation must be included in S order that with the proper D, it can be demonstrated that $D, S \vdash R$, and (2) explain what must be in D if S does *not include the knowledge* that is your answer for part 1.

(a)
Either S includes

or D includes the assumption that

(b)
Either S includes

or D includes the assumption that

(c)
Either S includes

or D includes the assumption that

(b) According to Michael Samborski, UW's Parking Lot C is a pay-and-display kind of lot that has no gate, but it has a meter that gives you a dated ticket that you display on your dash. The proper way to operate under the pay-and-display method to pay for parking is a seven step procedure:
1. Find a parking spot.
2. Park your car.
3. Walk to the meter.
4. At the meter, pay $5.00 for and get a dated ticket that is valid until 3:00am the next day.
5. Walk back to your car.
6. Place the ticket on the dash board in your car.
7. Leave your car.

Two problems that Samborski saw as he used the lot every time he came to attend CS645 were

Problem 1: Parkers who wanted to park closer to the corner closest to the university would first drive to the meter, leave their cars blocking the lane for a minute while they leave the car to buy the ticket at the meter, and then drive to their spaces.

Problem 2: Parkers leaving the lot early in the day had no use for a parking ticket that lasts until 3:00 am and would pass them off to cars waiting at the meter, perhaps for a payment of $2.00.

What is a reason for a parker doing what parkers are doing in Problem 1.

Explain the bad effect on the parkers of Problem 1.

Explain the bad effect on the UW Parking Service of Problem 2.

Explain the good effect on each parker, the giver and the receiver, of the parking ticket exchange in Problem 2.
The giver

The receiver

Samborski described an app-based payment system that a parker can install on his or her smart phone. This app knows the parker's license plate number and credits money from his or her credit card to the parker's chosen parking spot for that plate. Use of this app can be combined with license plate recognition (LPR) to simplify enforcement by the UW Parking Services.

On top of that, parkers will be able pay by the hour, so that most of them will be paying a lot less than the current flat $5.00/day fee.

Explain how Samborski's solution solves Problem 1.

Explain how Samborski's solution solves Problem 2.

Explain how payment by the hour affects Problem 2.

6. **[8 total marks]** NFRs

Some believe that the *BIGGEST* problem about nonfunctional, quality requirements is that many of them are what linguists call *vague*.

van Rooij defines vagueness this way:

> "An expression is vague, if its meaning is not precise. For vagueness at the sentence-level this means that a vague sentence does not give rise to precise truth conditions."

In other words a predicate (a property) is said to be vague if its truth is not determinable with certainty.

Explain the problem that a vague requirement causes to attempts to validate the holding of the requirement.

Name one nonfunctional or quality requirement that is vague.

Explain in one sentence why this nonfunctional or quality requirement is vague.

Describe one way to make a test case for verifying that a program meets this vague requirement. *That a program passes or does not pass this test case must be certain even though the tested requirement is vague.*

Does your way to make a certain test case really eliminate the vagueness of the original vague requirement?

Explain your answer.

7. **[16 total marks]** Ambiguity

(a) Consider the 5 variations of the sentence My e-key unlocks my Prius., obtained by distributing only to different places in the sentence.

_______ 1. Only my e-key unlocks my Prius.

_______ 2. My only e-key unlocks my Prius.

_______ 3. My e-key only unlocks my Prius.

_______ 4. My e-key unlocks only my Prius.

_______ 5. My e-key unlocks my only Prius.

Items B through H below are sentences, each of which might clarify, interpret, explain, or exemplify, *but not contradict* the meaning of one of 5 sentences above. Item A is simply the claim that a sentence above is classified as meaningless. Please, in the underscore next to each sentence S above, write the letter labeling every of the 8 items below that clarifies, interprets, explains, or exemplifies S. Each letter appears in the underscores at most once, and some letters may not be used at all. *Do not worry yet what is true. Worry only about matching meaning. Later we shall consider truth.*

A. Meaningless
B. My e-key does not lock my Prius.
C. My e-key scratches my Prius.
D. My e-key does not unlock Joe's Prius.
E. I have only one e-key.
F. I do not have a second Prius.
G. Joe's e-key does not unlock my Prius.
H. My e-key locks my Prius.

Suppose I have only one e-key, this e-key is the e-key for my Prius, no one else has a e-key for my Prius, I have only one Prius, and everything that is specified in the original Prius user's manual about a Prius e-key is true.
Which of the sentences 1 – 5 are true under this supposition?

Make a sentence that is true under this supposition, with as many "only"s as possible, by crossing off in this not true sentence as few "only"s as possible:

Only my only e-key only unlocks only my only Prius.

(b) Suppose that in a single room, there are multiple motion sensors, there is a single alarm that is to be issued when the sensors detect an incursion. One possible statement of the requirements is

R: The sensors issue an alarm when they detect an incursion.

From a programmer's viewpoint, what is wrong with this R? (It's not enough to say that R is ambiguous; you have to say what the programmer does not know given R.)

Thus, R is ambiguous! It has at least 3 different meanings. By careful use of a singular, possibly collective, subject, rewrite this sentence 3 different ways to illustrate 3 of these meanings. *Each sentence must be of the form*

. . . sensor. . . issues an alarm when . . . detect. . . an incursion.

and must answer the programmer's question.

i.

ii.

iii.

8. **[10 total marks]** Cost Estimation

(a) Why do we need an estimate of the cost to build a CBS during the analysis of the CBS's requirements, before we even begin to design the modules of its implementation?

Why, despite this need, is it so difficult to estimate the cost to build a CBS during the analysis of the CBS's requirements, before we even begin to design the modules of its implementation?

(b) Suppose that we are going to do a Function Point Analysis of the *addition* to the Prius's software to accommodate your group's specified IEKS. List all the use cases that would have to be counted among the inputs in the function-point count.

List all internal processing that are driven by these use cases and that would have to be counted among the user query and data access routines in the function-point count.

List all outputs that are driven by these use cases and that would have to be counted among the output routines in the function-point count.

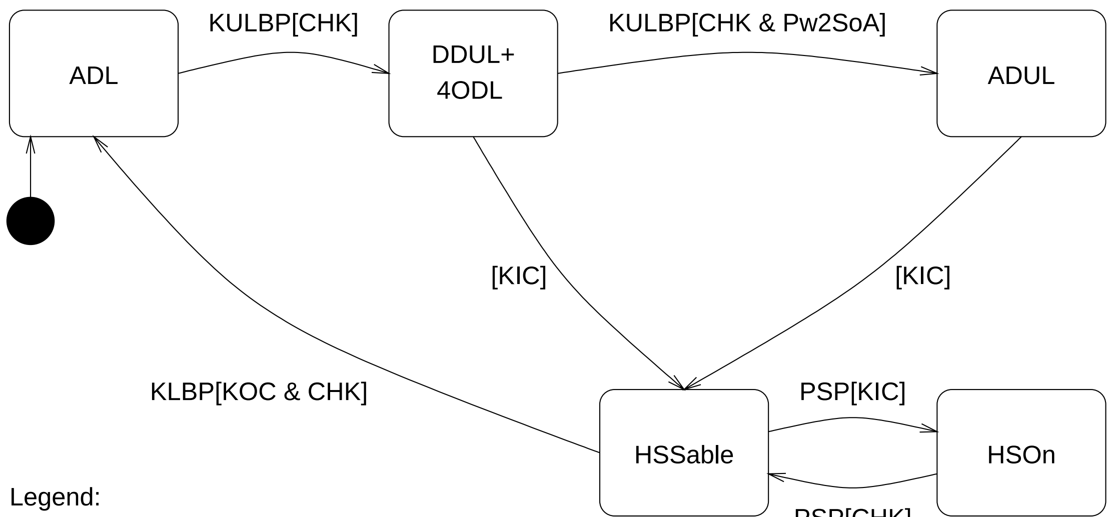
How does the fact that an input routine to type a character on a virtual keyboard on a touch screen *is executed an arbitrary number of times*, e.g., in a loop, affect the function counting?

9. **[26 total marks]** State Machines and Linear Temporal Logic

(a) One unfortunate circumstance is that a *system state of any model* and a *state-machine state* are *not* the same thing. They are quite different and we need to qualify any use of the word "state" with the adjective "system" or "state-machine" to disambiguate. Nevertheless, the two kinds of states *are* related.

A *system state* σ of a model M is "an assignment of values to the model's variables. Intuitively, the system state is a snapshot of the system's execution. In this snapshot, every variable has some value."

Consider the state machine SM that was given as part of the project's vision document



Legend:
ADL: all doors locked
DDUL+4ODL: driver door unlocked and 4 other doors locked
ADUL: all doors unlocked
HSSable: hybrid system startable
HSOn: hybrid system on

KULBP: key's unlock button pushed
KLBP: key's lock button pushed
PSP: power switch pushed

CHK: car hears key
Pw2SoA: pushed within 2 seconds of another
KIC: key in cabin
KOC: key outside cabin

In the following, you may consider each condition to be its own state-machine variable of type Boolean. Describe a system state of SM as an n-tuple. You may use *value*(V) to mean the "the current value of V".

Give two particular system states of SM

(b) Complete the following incomplete Linear Temporal Logic specification of the state machine SM

$\square(\text{DDUL+4ODL} \Rightarrow (\text{DDUL+4ODL} \;\mathcal{W}\; ((\text{KULBP} \wedge (\text{CHK} \wedge \text{Pw2SoA})) \vee (\text{KIC}))))$
$\square((\text{DDUL+4ODL} \wedge (\text{KULBP} \wedge (\text{CHK} \wedge \text{Pw2SoA}))) \Rightarrow \bigcirc\, \text{ADUL})$
$\square((\text{DDUL+4ODL} \wedge (\text{KIC})) \Rightarrow \bigcirc\, \text{HSSable})$

(c) A CBS is said to be *statemachiny* if its overall behavior is naturally described by a state machine with a finite number of states and events.

Which of the following CBSs are statemachiny? Write "Yes" or "No" in the provided underscore.

- IEKS (additional ways to start a Prius when the key battery is dead) ________
- MS Word (formatting the input text according to a style definition) ________
- Sort (sorting the input file to produce a sorted output file) ________
- Dating (finding mutual matches among profiles, in which each person provides a profile of him/herself and a profile of his/her ideal mate) ________
- Factorial (compute $n!$ where n is the input) ________
- DVR (digital video recorder and its remote controller) ________
- Weather predictor (computes a prediction from a collection of data gathered over a long period of time) ________
- Google translator (translates text from one language to another using machine learning from known correct translations) ________
- Turnstile (automated coin operated gate keeper for Waterloo parks) ________

10. **[20 total marks]** The Requirements Iceberg

(a) In terms of human ambition, explain why a silver bullet remains a silver bullet for only about a year.

(b) What a piece of code is supposed to do is also known as the piece's ______________
.

(c) What is the latest time that one can determine the requirements for a line of code?

(d) When one software development company X is developing a CBS A, its employees typically

- develop a preliminary version V of A and
- write a requirements specification R for A

concurrently. X calls this concurrent development-and-specification "getting a head start on the implementation while learning its requirements". When the development and specification are done, X has

- an incomplete and not completely correct version V of A and
- a requirements specification R for A

It costs C to build V, and it costs D to write R.

Experience has shown that between 10% and 40% of V has to be changed to get the resulting V' to match R. *Moreover, we have evidence that modifying in a consistent manner an already written piece of code costs 10 times what that piece cost to write from scratch.*
In terms of C and D, explain the total costs involved if as little as 10% of V has to be changed to make the resulting V' match R.

In terms of C and D, explain the total costs involved if as much as 40% of V has to be changed to make the resulting V' match R.

In terms of C and D, explain the total costs involved if V were not developed at all. Instead, X waited until R was done to start building V' to match R, from scratch. Assume that the costs to build V and V' are the same.

If you believe that time = money and the 10% minumum for the changes, then is the so-called head start really a head start, i.e., does X finish building V' sooner with the so-called head start than without the so-called head start?

(e) Berry claimed that updating existing software to meet new requirements is like lying. In every lie, there is the lie and there is the truth. Explain what is the lie when you are updating existing software to meet new requirements.

Explain what is the truth when you are updating existing software to meet new requirements.

Why is making sure that no one can discover that you have lied is very difficult (for a lie in general)?

What makes the lie that is updating existing software to meet new requirements so difficult to pull off?

(f) Why is code for flying an airplane more complex than a mathematical model of flying that is based on fluid mechanics?