Dr. Gates: Did you talk about the open [interval] on the right?

Dr. Salamah: Oh! That’s a good point. So remember when we say that the scope for the before R, so remind me, the scope for the before R does it, what’s special about it?

Stops right before the R.

Good. With the after L it includes that state of the data reading where L holds. So it’s not the same, with the after L it does include that reading where L holds. Where the before R does not include the data reading where R holds. And this is the same for between L and R. It includes the reading for L but it doesn’t include the reading for R. Yeah? And if you see the definitions also the textual definition not only the diagrams that specify them. The after L until R is similar, very similar to between L and R. The only difference here is if R never holds then the scope is between the first L all the way to the end of the arrow. Or until you get an R if that’s the infinite truth. Or if you don’t you keep checking until you get an R then it’s until the very end of the readings. We’re good? Yeah? Okay.

Alright so, let’s move to patterns, for today we decided to just give you a subset of this and hopefully elicit some questions from you and we can go to more examples. So, we talked about scope is a range of interest, what I want my property to hold, well my property is specified by a pattern. A pattern is a sort of property that repeats over and over. So we are breaking these into time dependent and not time dependent and today we are giving you four examples of the not time dependent ones. So for example universality, that’s the first example that we have. Over that scope, now the scope can be whatever one of the five we described before. Over that scope, over that set of data readings I want that universality pattern to hold which means I want that property to hold over every reading in that scope. So if I have a global scope I want that reading, universality property to be true over every reading in my data set. So we talked about scope as array of edges where I want my property to hold, where my property is specified by a pattern. Pattern that you will specify over and over and over. So we are breaking this into time dependent and not-time dependent and today we will give you four examples of the not-time dependent once. So for example, universality that’s the first example that we have. Over that scope, I mean scope it can be one of the five we described before, over that scope, over that set of data, data reading, I want that universality pattern to hold, which means I want that property to hold over every reading in that scope. So if I have a global scope I want that universality property to be true over every reading in my data set. Okay? If I have universality between L and R, then I don’t care what happens before L, I don’t care what happens after R, I only care that this property holds in every reading between the L up until just before the R. That’s what universality means, that property and for example, here I said that temperature has to be smaller or equal to 35, that temperature has to be smaller or equal to 35 in every reading within the scope. So again, if I am doing global scope, the temperature has to be 35 or less on every reading over all my set of data reading or if I am doing after L, I don’t care what happens before L, but the temperature better be 35 or less after L until the end of the reading, until the end of my data set. Easy so far? Okay.

Absence is a little bit the opposite. That property should never hold in my scope. So we were saying universality is it should hold for every reading, absence should never hold in any of my readings, again within the scope you are always going to read by the scope and existence it has to happen at some point, it has to happen at least once. If it doesn’t happen at least once, then there is an anomaly or there’s a violation, and then the interesting one is response. If you notice on all the ones before this one, before the response, in all the ones we have discussed so far we are only targeting one type of reading, right? It’s either temperature or its humidity or something, but we are doing just one type of reading, with response we have two scopes because we have two data readings. I am reading about temperature, I am reading about humidity for example, and I am saying in response it’s a cost-effect, for example if temperature falls to 15, then humidity will behave in a different way. That is why we are saying that we are performing or we are checking two readings, two data sets: the data set for temperature and the data set for humidity. Now, somehow your job will be to calibrate those two data sets to make sure that that order is preserved, if the temperature happens at reading number 5, the temperature becomes 15 in reading number 5, then make sure that humidity becomes 80% at that same reading number 5 within the humidity data set or at reading number 6. It has to happen at that same reading or the reading immediately next. Questions?

Student: I think I’ll wait until you’re done.

Student: Yeah I think we can wait until the end.

Dr. Salamah: Well, I think this is my last scope.

Dr. Gates: Yeah maybe. In that handout, there are several things you’ll notice. One is that it tells you what the scope is, and it tells you what the Boolean statement is and the parameters. Let’s see what else . I want you to look at absence for instance.

Dr. Salamah: Absence is number 2.

Dr. Gates: If you look at absence, it is giving you a property right? It is giving you a Boolean relationship, but notice that absence is saying that that [the property] does not hold. I am not putting the “not” there. I am just saying this is what should not hold.

Dr. Salamah: Right, so we are not negating that property. What we are saying is that the temperature should never be smaller or equal to 15. What Dr. Gates is saying, when you are specifying the Boolean, the Boolean does not say that temperature is not smaller or equal to 15. That is something that has to be done during your implementation based on what the pattern that has been selected. Okay?

Dr. Gates: What you are trying to do is to figure out how to get this information, or how to help the scientist specify these properties, and there are a lot ways you can do it. You may use a pattern to somehow cue them, or maybe questions that can cue and these are things we want you to think about. We are at the point of saying what you need to do, the “what”, not the “how”, but you still should be thinking about what the system is able to gather. You are going to have to come up with some relationship, right? We are going to give you some other examples that are a little more complicated and it may be that you will have to do some math or some transformation with the variables that you are checking, sensor reading that you are checking. But we will--we are going to add another layer next week. We want to make sure that you understand the basics right now.

Dr. Salamah: One thing that I want to clear up and I was hoping that you will be asking this and maybe you were going to ask but I will clear it up anyway. When we say “not time dependent”, those are the type of properties that we are explaining today, then you see the example where it says “during the daytime of May 12th”, that is time right?, but that is not the time we are talking about. Time dependent we are talking about are within the property itself, so the property holds multiple times, the property holds a certain number of times. That is the time dependency we are talking about. Of course we didn’t give you any of these examples today, because we gave you the not dependent today, but time dependent doesn’t mean that there is no time associated with the scope because the scope has to be built with some sort of a time stamp over reading. So don’t let that confuse you. “Not time dependent” doesn’t mean that the property will not have any time in the textual description of it. Is that clear? Okay. All right.