What does it mean to say an algorithm is Sound and Complete?

I heard different interpretations of *sound and complete*. I understand that *completeness* means finding a solution if there is one. What does it mean to say an algorithm is *sound*.

What does it mean to say an algorithm is Sound and Complete?

algorithms terminology

edited Feb 15 '13 at 7:37
gnat
18.2k 12 59 121

asked Mar 20 '12 at 18:25
mutelogan
203 1 2 5

I suggest you reevaluate what answer you accepted given that one is wrong. – BlackJack Mar 20 '12 at 19:44

Just did that :) – mutelogan Mar 20 '12 at 20:13

3 Answers

These are very specific terms as related to logic.

Here are some starting points:

http://en.wikipedia.org/wiki/Soundness

http://en.wikipedia.org/wiki/Completeness_(logic)

Basically, soundness (of an algorithm) means that the algorithm doesn't yield any results that are untrue. If, for instance, I have a sorting algorithm that sometimes does not return a sorted list, the algorithm is not sound.

Completeness, on the other hand, means that the algorithm addresses all possible inputs and doesn't miss any. So, if my sorting algorithm never returned an unsorted list, but simply refused to work on lists that contained the number 7, it would not be complete.

It is sound and complete if it works on all inputs (semantically valid in the world of the program) and always gets the answer right.

edited Nov 28 '14 at 15:02
Hi I'm Frogatto
103 4

answered Mar 20 '12 at 18:48
Erik Dietrich
4,515 3 20 30

Thanks. I was really confused about what *soundness* means. I was getting multiple answers. – mutelogan Mar 20 '12 at 20:10

Happy if it helped... :) – Erik Dietrich Mar 20 '12 at 20:11

10 An example would be the Binary Search, It's sound, but it's not complete. It can't work on non-sorted lists. – Malfist Mar 20 '12 at 20:36

1 @Malfist but isn't the 'world of the program' sorted lists? – Andres Oct 27 '13 at 23:02

@Malfist Why it cannot? I can easily supply an unsorted list. The fact that I will get the wrong results means that it is unsound rather than incomplete. So that definitions are interchangable, as they are in CS. Isn't it? – Valentin Tihomirov Jan 31 at 22:20

I find Erik Dietrich's answer a tad confusing. The following is better:

An algorithm is *sound* if, anytime it returns an answer, that answer is true. An algorithm is *complete* if it guarantees to return a correct answer for any arbitrary input (or, if no answer exists, it guarantees to return failure).

Two important points:

1. Soundness is a weak guarantee. It does not promise that A will terminate.
2. Soundness and Completeness are related concepts; infact they are the logical converse of each other. i.e. Soundness says that if an answer is returned that answer is true. Completeness says that an answer is true if it is returned.

Consider for an example a sorting algorithm A that receives as input a list of numbers. We say that A is sound if every time it returns a result that result is a sorted list. Likewise, we say that A is complete if guarantees to return a sorted list any time we give it a list of numbers.

edited Feb 15 '13 at 6:40

answered Feb 15 '13 at 6:04

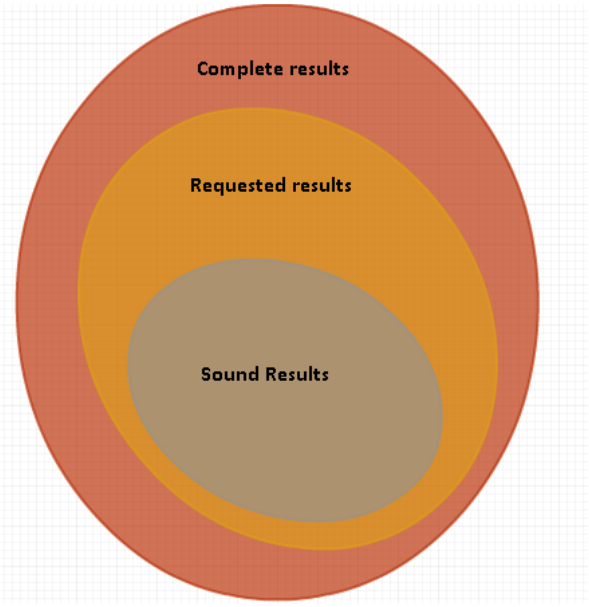
Daniel
97 1 2

Why are you confused? *"An algorithm is sound if, anytime it returns an answer, that answer is true."* means the same as *"Basically, soundness (of an algorithm) means that the algorithm doesn't yield any results that are untrue."* These mean the same thing. As for your (very brief) definition of Completeness, it matches nothing in the wikipedia link and you cite no reference of your own. I have to say, Erik's definitions are more practically useful. If yours are correct, you have to provide better evidence and more meat. – itsbruce Nov 28 '14 at 18:31

1 Just to clarify, when you say "Completeness says that an answer is true if it is returned", you mean that the answer is "correct" right? – Dois Jun 8 '15 at 1:45

There are much better answers at the SO. Basically, you provide some data collection and criteria to search. *Sound algorithm* catches you only the fish that matches the criteria but it may miss some data items. *Complete algorithm* produces a superset of requested results, which means that you receive some garbage on top of requested results. Sound algorithm is more conservative.

Statistician would probably say that sound algorithm is biased towrads type I errors (it does not accept the correct candidates), whereas complete algorithm is biased towards type II errors (to accept the false candidates).



edited Feb 1 at 7:31

answered Jan 31 at 22:46

Valentin Tihomirov
168 8