Definitions

We can define an algorithm as successfully binding a series of variables to some Find requirements, given certain properties about other variables.

A meta-algorithm is the description of how to reach such an algorithm given the problem.

Meta-Algorithms

Stub

If we were given the answer, just return the answer. Also, if we were given a task which we can solve without even using the input, just solve the task.

If all the requirements for a Find variable are satisfied by a Given variable, the algorithm is to return the Given variable.

pseudocode

when trying to bind variable x

if exists given y st forall requirements r of x, y satisfies r

yield algorithm {bind x to y}

Theorem code

Option one (unbound)

Given x st enumerable(x), Find y st child(x,y)

can be simplified to

Given x,y st enumerable(x) & child(x,y)

Option two (bound)

Given x,y st enumerable(x), Find y st child(x,y)

can be simplified to

Given x,y,y' st enumerable(x) & child(x,y), Find y' st y==y'

output runtime

O(1), just return the input value.

examples

example 1

Given x st type\_list(x), Find y st type\_list(y)

Obvious, just return the given list.

example 2

Given x st type\_list(x), Find y st type\_collection(y)

Pretty simple as well, may involve a casting operation in some languages. We perform forward-chaining with logic on the input, then when we check for satisfiers for y we have x.

Simple Testing

If we just need to perform a simple test, then do it.

If our only constraint is an atomic which is simply testable, just return an algorithm that tests it.

If the constraint is an atomic this same principle applies, but the checking time is linear across the bound variable instead of being O(1)

psuedocode

when trying to bind variable x

if requirement r(y) on x is testable atomic

yield algorithm {

if test(r)

x=y

}

Brute Force

Try all possible answers

We can eliminate some of our requirements if they are enumerable. Just enumerate over all of them and try to solve the reduced problem for each one. If we fail to come up with a solution from this, there is no solution to the problem.

pseudocode

when trying to bind variable x

if exists requirement r of x st r is ANDing and

child r' of r is enumerable

yield algorithm {

forall x' st x' satisfies r' for x

try to solve (r-r') with x = x'

}

output runtime

O(enumeration\_size(r' wrt x) \* runtime of solving r-r' )

Brute Force over children

This is a revised version of brute-force.

application

when a Given G contains g = enumerable(x), and Goal F contains f = find(y st child(x,y))

The problem can be reduced to

Given G - g + y st child(x,y), test(F - f)

by applying an iteration over all y st child(x,y)

pseudocode

foreach child ny of x:

run test code, with pass result of returning y = ny

Testing

This is a revised version of testing

application

When a Goal F contains f = test(Z(y)) st Z is testable

The problem can be reduced to

test(F-f)

By simply testing for f

pseudocode

if(Z(y))

do rest of test, if the rest is empty, then return binding for y

Search through a graph