Program breed(CRProgram to, CRProgram from, size\_t toRootI, size\_t fromRootI) {

auto newProgram(to);

killChildren(newProgram, toRootI);

newProgram[toRootI] = getTerminal(); // to make newProgram valid tree after killing children

auto branchIntervals = markChildren(from, fromRootI);

auto levelStarts = getLevelStarts(newProgram);

Indexes insertionPoints;

auto point = toRootI;

auto levelI = getLevelI(toRootI, getLevelStops(newProgram));

for (size\_t intervalI = 0; intervalI < branchIntervals.size(); intervalI += 2) {

++levelI;

if (levelI < levelStarts.size()) {

auto prevPoint = point;

point = levelStarts[levelI];

for (size\_t i = levelStarts[levelI - 1]; i < prevPoint; ++i) {

point += arities[to[i]];

}

insertionPoints.push\_back(point);

}

else {

insertionPoints.push\_back(to.size());

}

}

newProgram[toRootI] = from[fromRootI];

for (int i = insertionPoints.size() - 1; i >= 0; --i){

newProgram.insert(newProgram.begin() + insertionPoints[i],

from.begin() + branchIntervals[i \* 2],

from.begin() + branchIntervals[i \* 2 + 1]);

}

return newProgram;

}