```
initialize;
repeat
    propagate completion, loop, and recorded nogoods;
    if no conflict then
        if not all variables assigned ∧ no nogood recorded then
            vsids ← ComputeVsidsLiteral();
            return CallExternalHeuristic(vsids);
        end
    else
        if conflict on top level then
            return unsatisfiable
        else
            AnalyzeConflict(conflict);
            Backjump;
        end
    end
until model found;
```

```
Function init is
    for a ∈ symbolic atoms do
        signature ← signature of a;
        ℓ ← solver literal of a;
        if signature matches some watched or result signature then
            register watch for ℓ;
        end
        if a is a fact then
            append a to heuristic program;
        end
    end
end
Function propagate(changes) is
    for l ∈ changes do
        update internal assignment value for l;
        mark l as untouchable;
    end
end
Function undo(changes) is
    decisions = ∅;
    for l ∈ changes do
        update internal assignment value for l;
        unmark l as untouchable;
    end
end
```

```
Function decide(vsids) is
    switch decision mode do
        case online do
            initialize solver with heuristic program;
            ground and solve heuristic program;
            let m = first model of heuristic;
            d ← heuristic/4 atoms in m;
            ℓ ← literal corresponding to best decision in d not marked
              untouchable;
            return ℓ
        case offline do
            if decisions = ∅ then
                initialize solver with heuristic program;
                ground and solve heuristic program;
                let m = first model of heuristic;
                decisions = sorted heuristic/4 atoms in m;
            end
            while decisions ≠ ∅ do
                d ← decisions.pop();
                if d not marked untouchable then return
                  makeDecision(d,vsids);
            end
            return vsids
        case resigned do return vsids;
    end
end
Function makeDecision(heuristic atom h, vsids) is
    switch atom in h do
        case "vsids" do
            return vsids
        case "resign" do
            set decision mode to resigned;
            return vsids
        end
        otherwise do
            ℓ ← literal corresponding to atom and sign in h;
            return ℓ
        end
    end
end
```