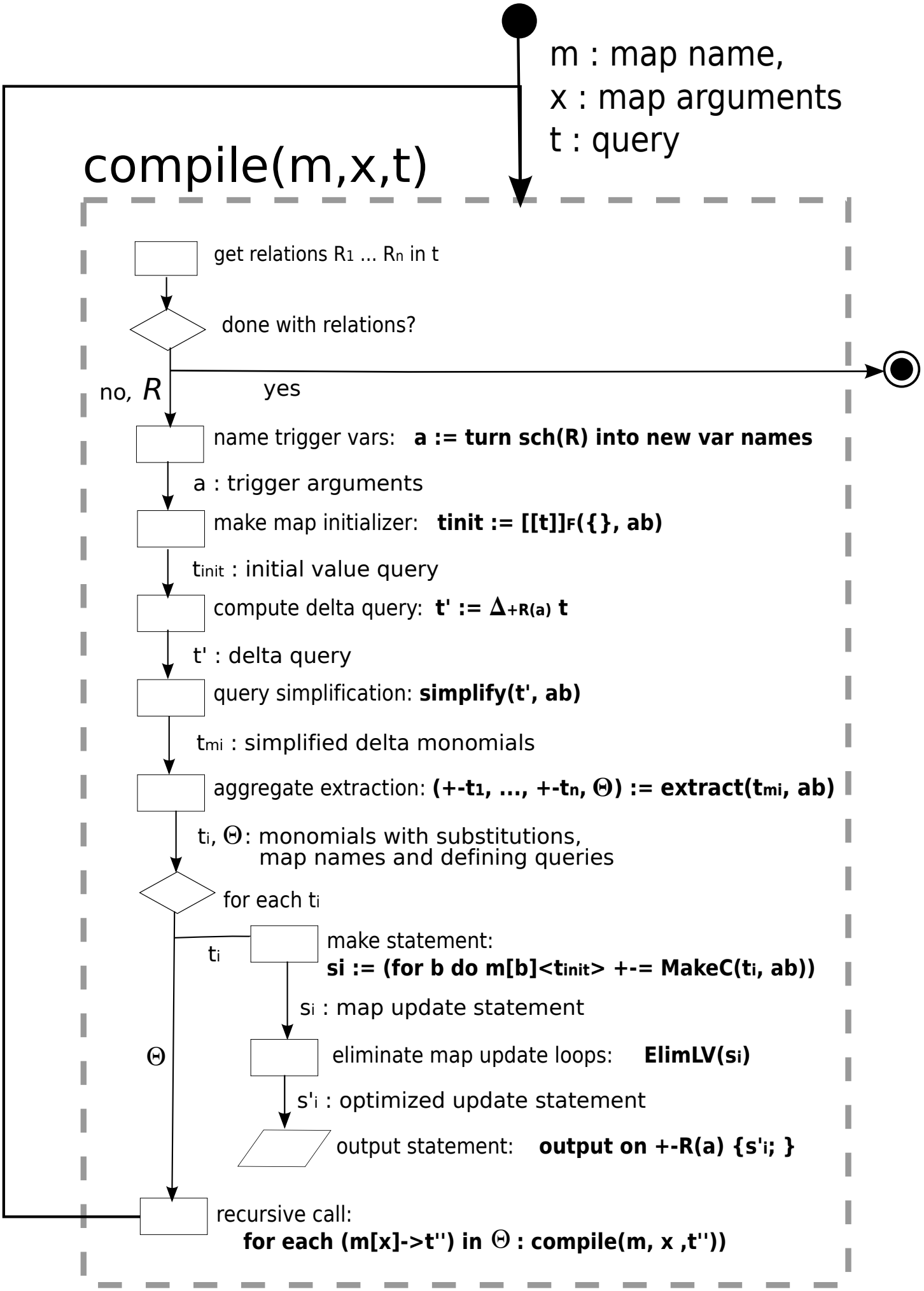
m : map name,
x : map arguments
t : query
compile(m,x,t)
get relations R1 ... Rn in t
done with relations?
no, R
yes
name trigger vars: a := turn sch(R) into new var names
a : trigger arguments
make map initializer: tinit := [[t]]F({}, ab)
tinit : initial value query
compute delta query: t' := Δ+R(a) t
t' : delta query
query simplification: simplify(t', ab)
tmi : simplified delta monomials
aggregate extraction: (+-t1, ..., +-tn, Θ) := extract(tmi, ab)
ti, Θ: monomials with substitutions,
map names and defining queries
for each ti
ti
make statement:
si := (for b do m[b]<tinit> +-= MakeC(ti, ab))
si : map update statement
Θ
eliminate map update loops: ElimLV(si)
s'i : optimized update statement
output statement: output on +-R(a) {s'i; }
recursive call:
for each (m[x]->t'') in Θ : compile(m, x ,t''))