AIFA: PLANNING

02/04/2024

Koustav Rudra

POP Example: Get Tea, Biscuits, Book

Initial State:

Op(ACTION: Start,
EFFECT: At(Home) ∧ Sells(BS, Book)
∧ Sells(TS, Tea)
∧ Sells(TS, Biscuits))

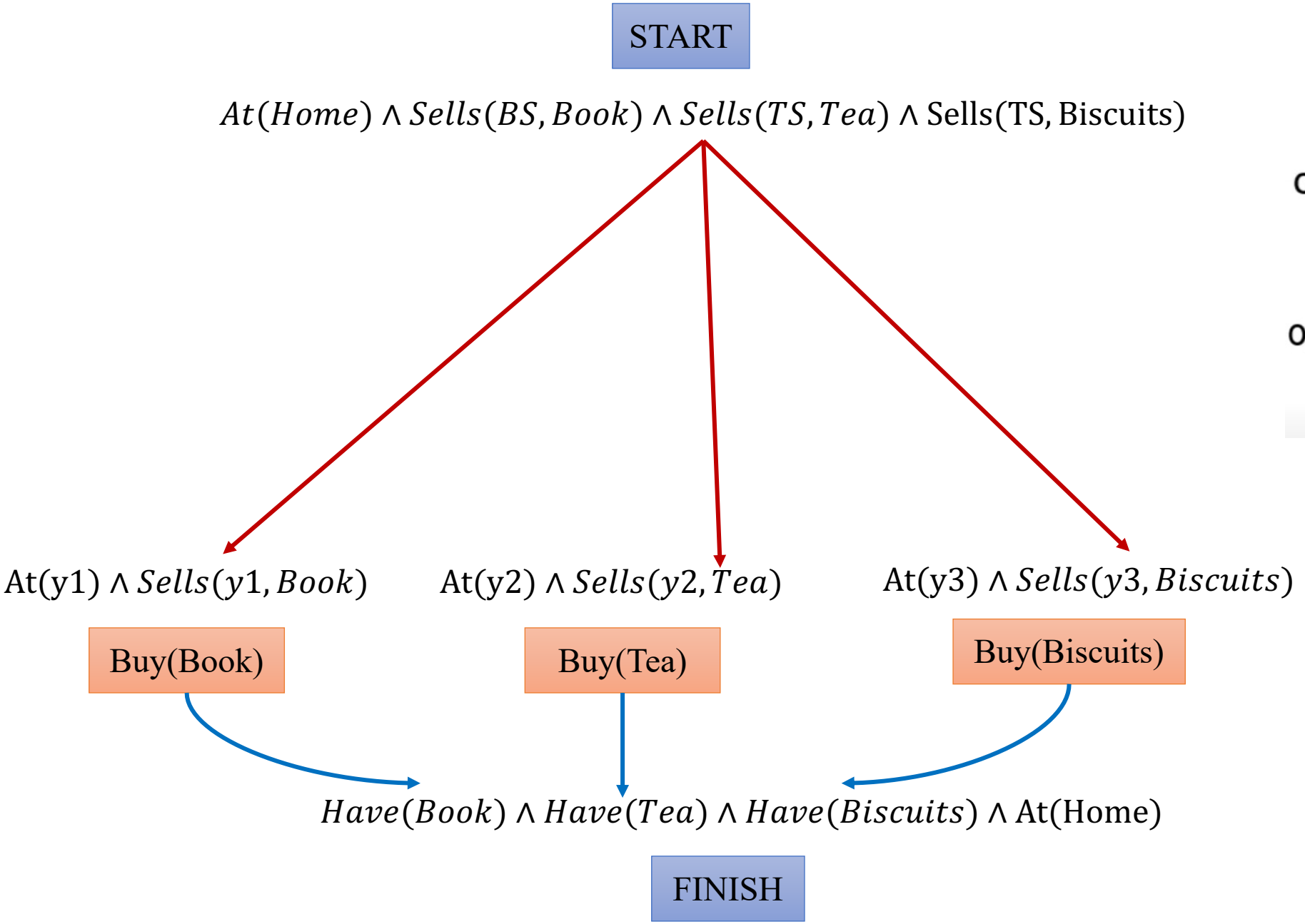
Goal State:

Op(ACTION: Finish,
PRECOND: At(Home) ∧ Have(Tea)
∧ Have(Biscuits)
∧ Have(Book))

Actions:

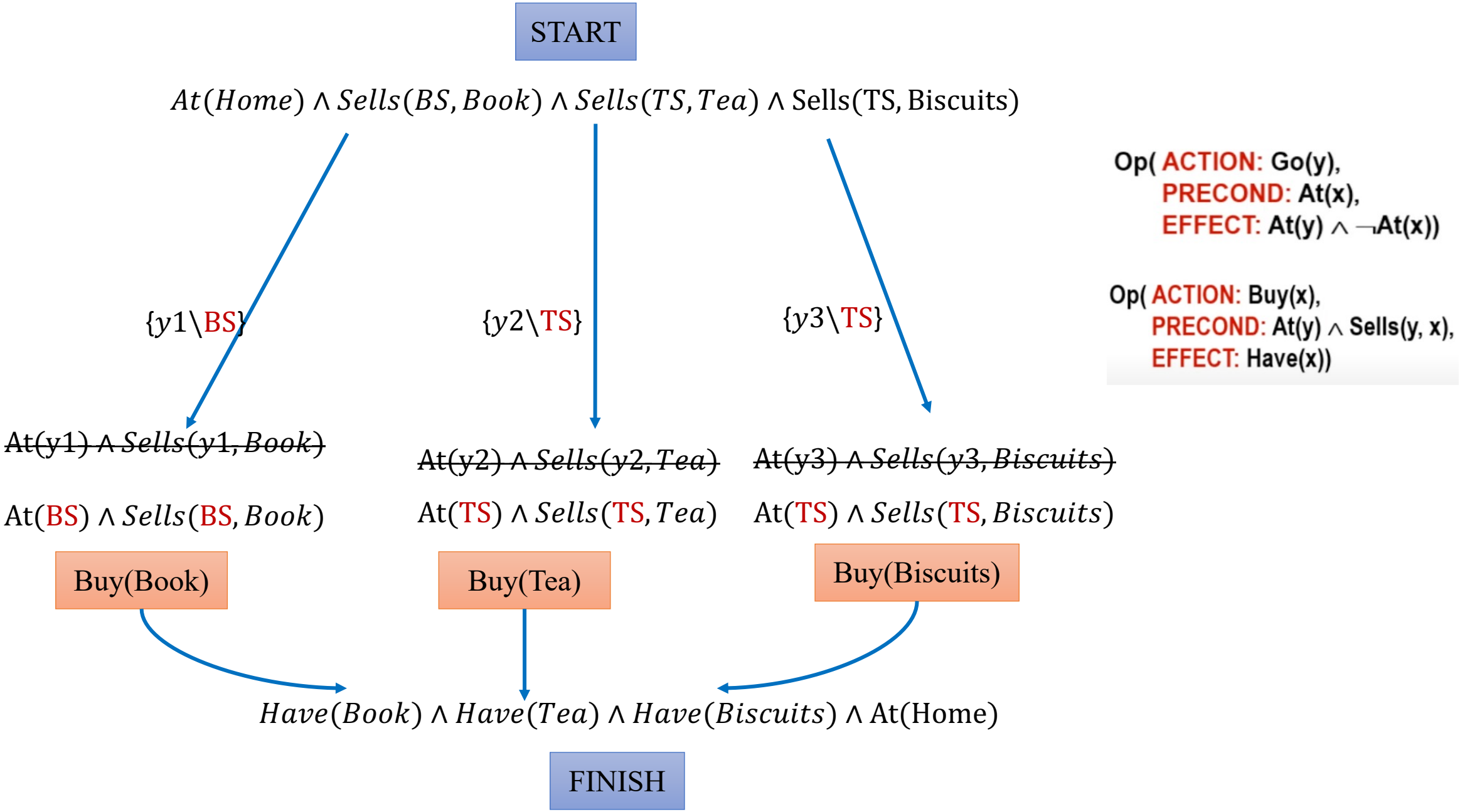
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) ∧ ¬At(x))

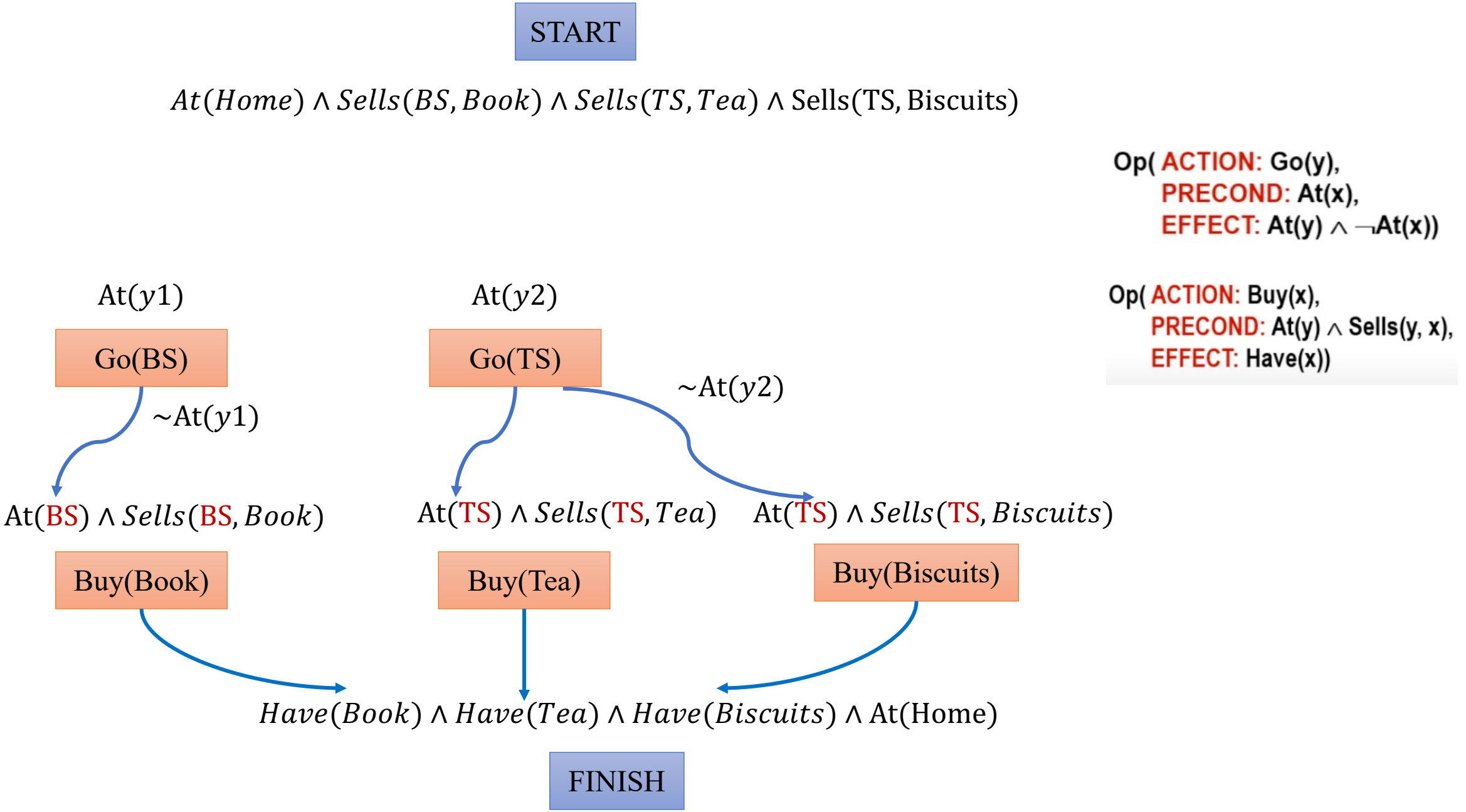
Op(ACTION: Buy(x),
PRECOND: At(y) ∧ Sells(y, x),
EFFECT: Have(x))

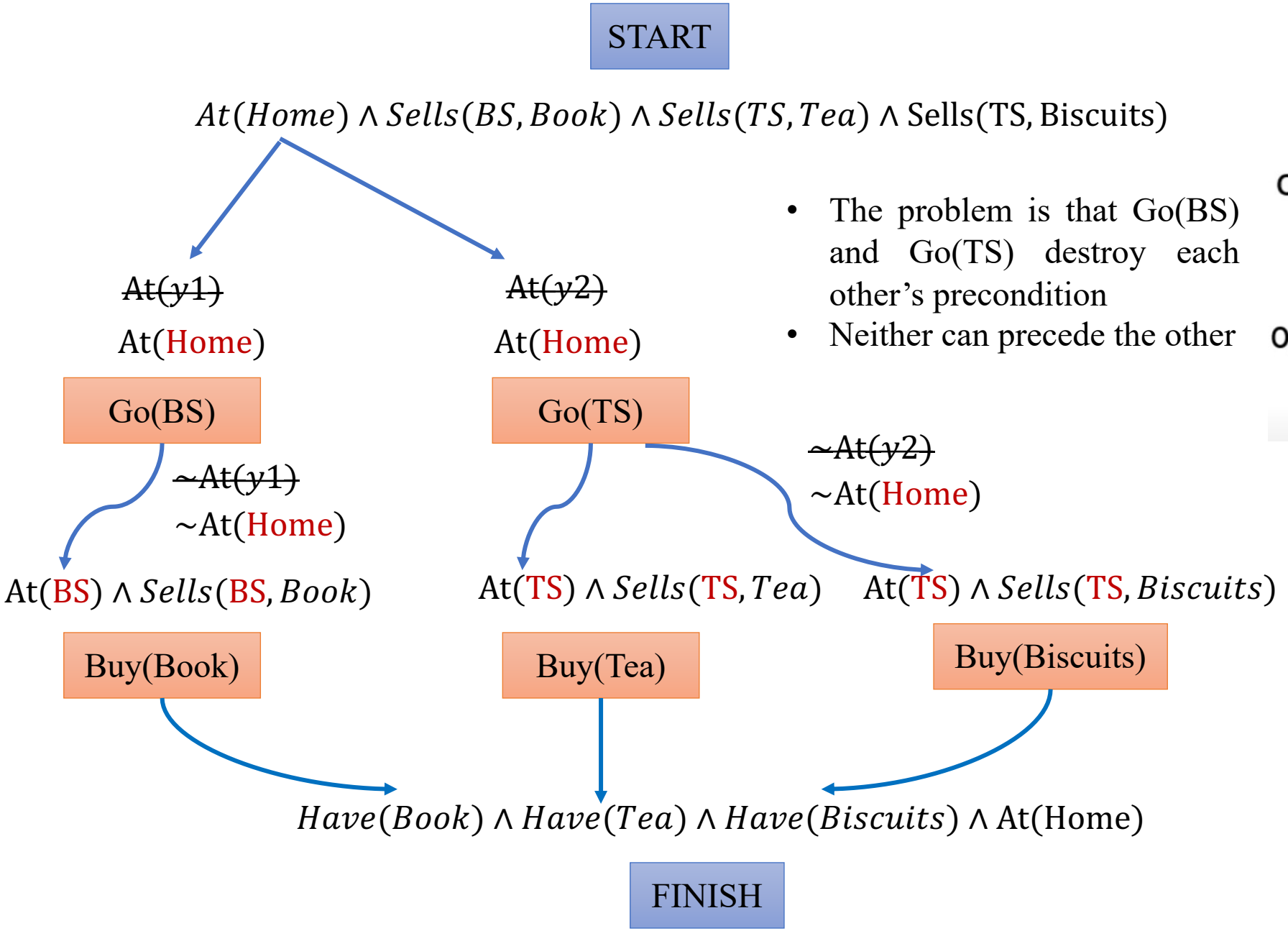


Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) $\wedge \neg$At(x))

Op(ACTION: Buy(x),
PRECOND: At(y) $\wedge$ Sells(y, x),
EFFECT: Have(x))


START
$At(Home) \wedge Sells(BS, Book) \wedge Sells(TS, Tea) \wedge$ Sells(TS, Biscuits)
{y1\BS}
{y2\TS}
{y3\TS}
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) ∧ ¬At(x))
Op(ACTION: Buy(x),
PRECOND: At(y) ∧ Sells(y, x),
EFFECT: Have(x))
~~At(y1) ∧ Sells(y1, Book)~~
At(BS) ∧ Sells(BS, Book)
~~At(y2) ∧ Sells(y2, Tea)~~
At(TS) ∧ Sells(TS, Tea)
~~At(y3) ∧ Sells(y3, Biscuits)~~
At(TS) ∧ Sells(TS, Biscuits)
Buy(Book)
Buy(Tea)
Buy(Biscuits)
$Have(Book) \wedge Have(Tea) \wedge Have(Biscuits) \wedge$ At(Home)
FINISH

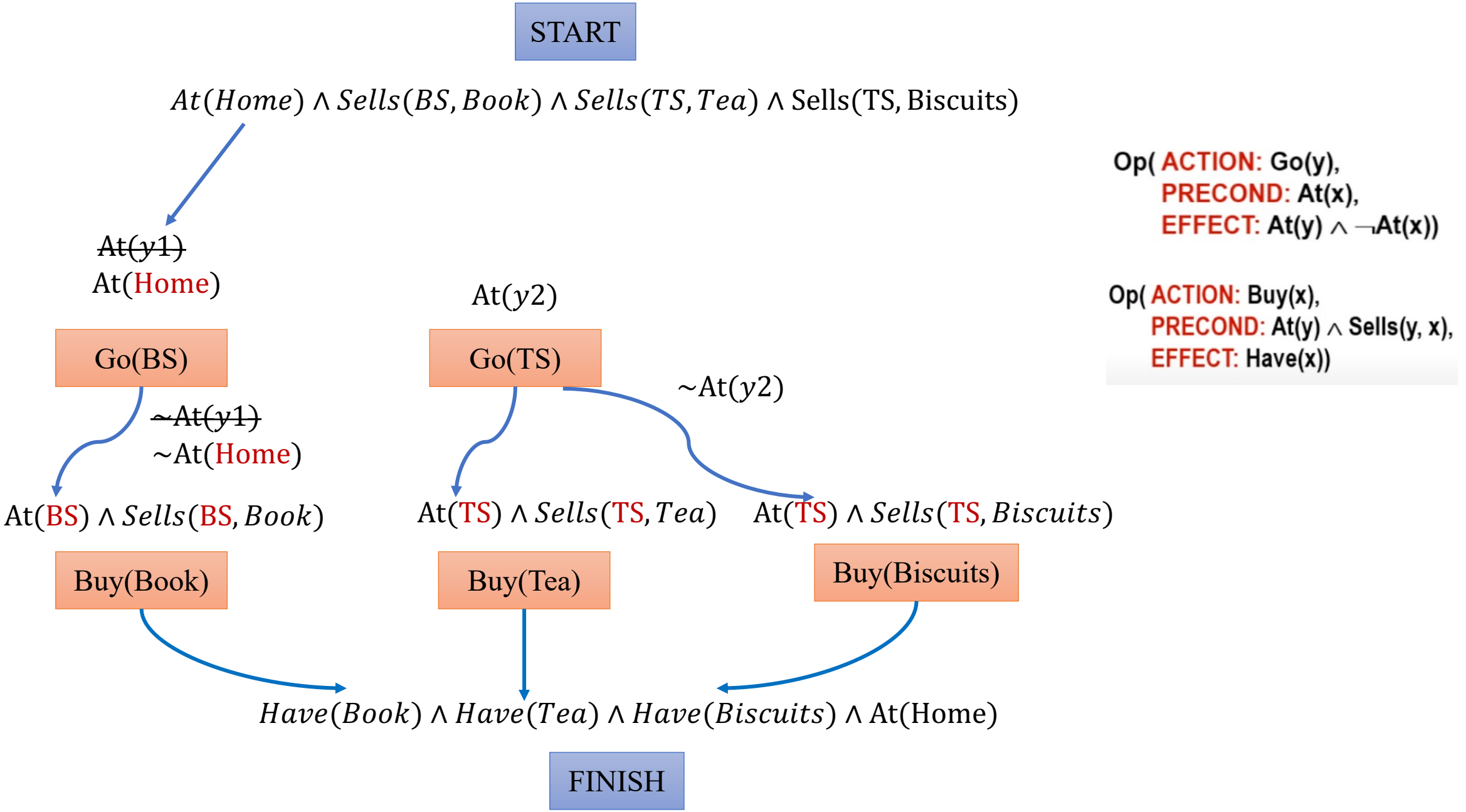

START
$At(Home) \wedge Sells(BS, Book) \wedge Sells(TS, Tea) \wedge$ Sells(TS, Biscuits)
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) ∧ ¬At(x))
Op(ACTION: Buy(x),
PRECOND: At(y) ∧ Sells(y, x),
EFFECT: Have(x))
At($y1$)
At($y2$)
Go(BS)
Go(TS)
~At($y1$)
~At($y2$)
At(BS) ∧ $Sells$(BS, $Book$)
At(TS) ∧ $Sells$(TS, Tea)
At(TS) ∧ $Sells$(TS, $Biscuits$)
Buy(Book)
Buy(Tea)
Buy(Biscuits)
$Have(Book) \wedge Have(Tea) \wedge Have(Biscuits) \wedge$ At(Home)
FINISH

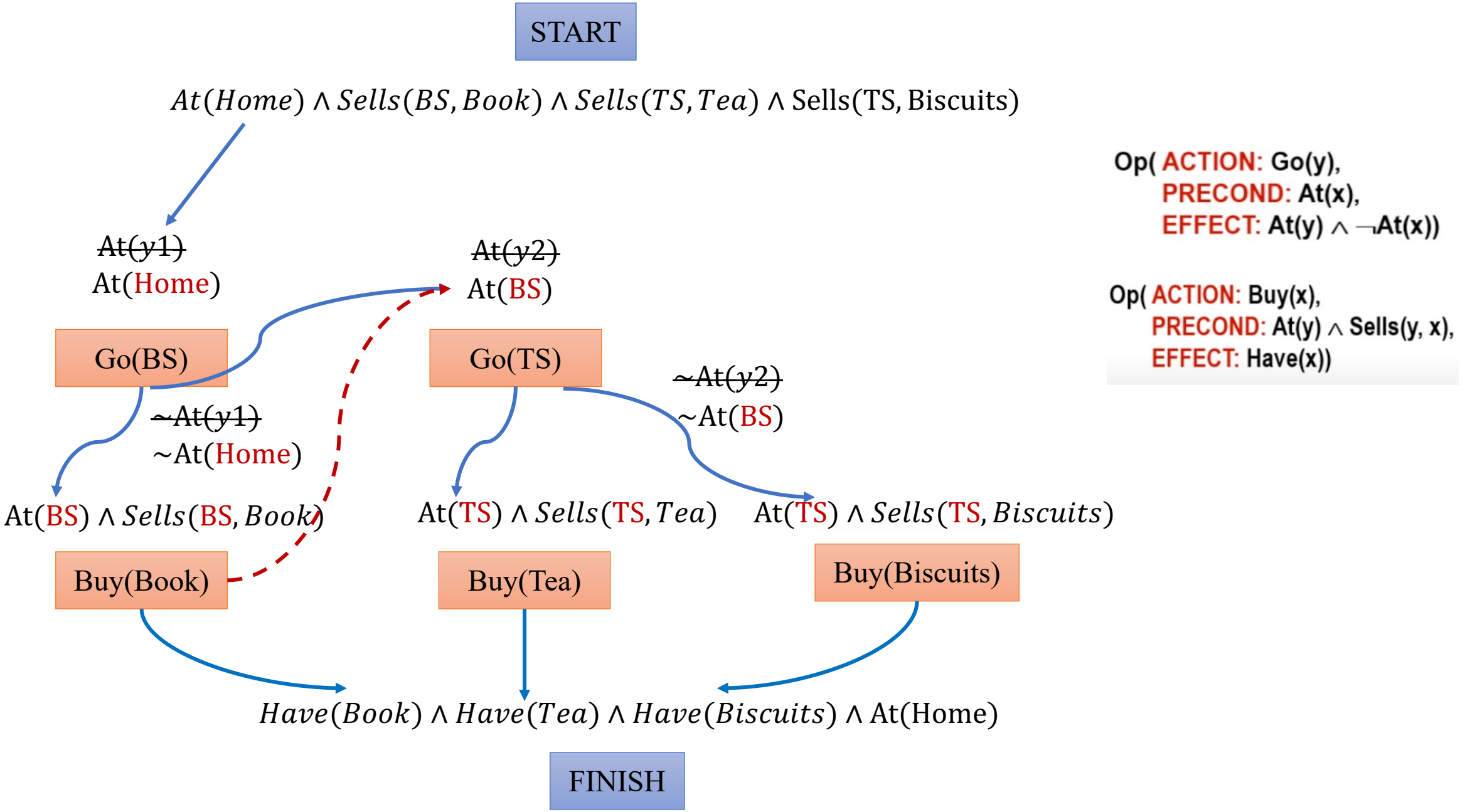


- The problem is that Go(BS) and Go(TS) destroy each other's precondition
- Neither can precede the other

Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) $\wedge \neg$At(x))

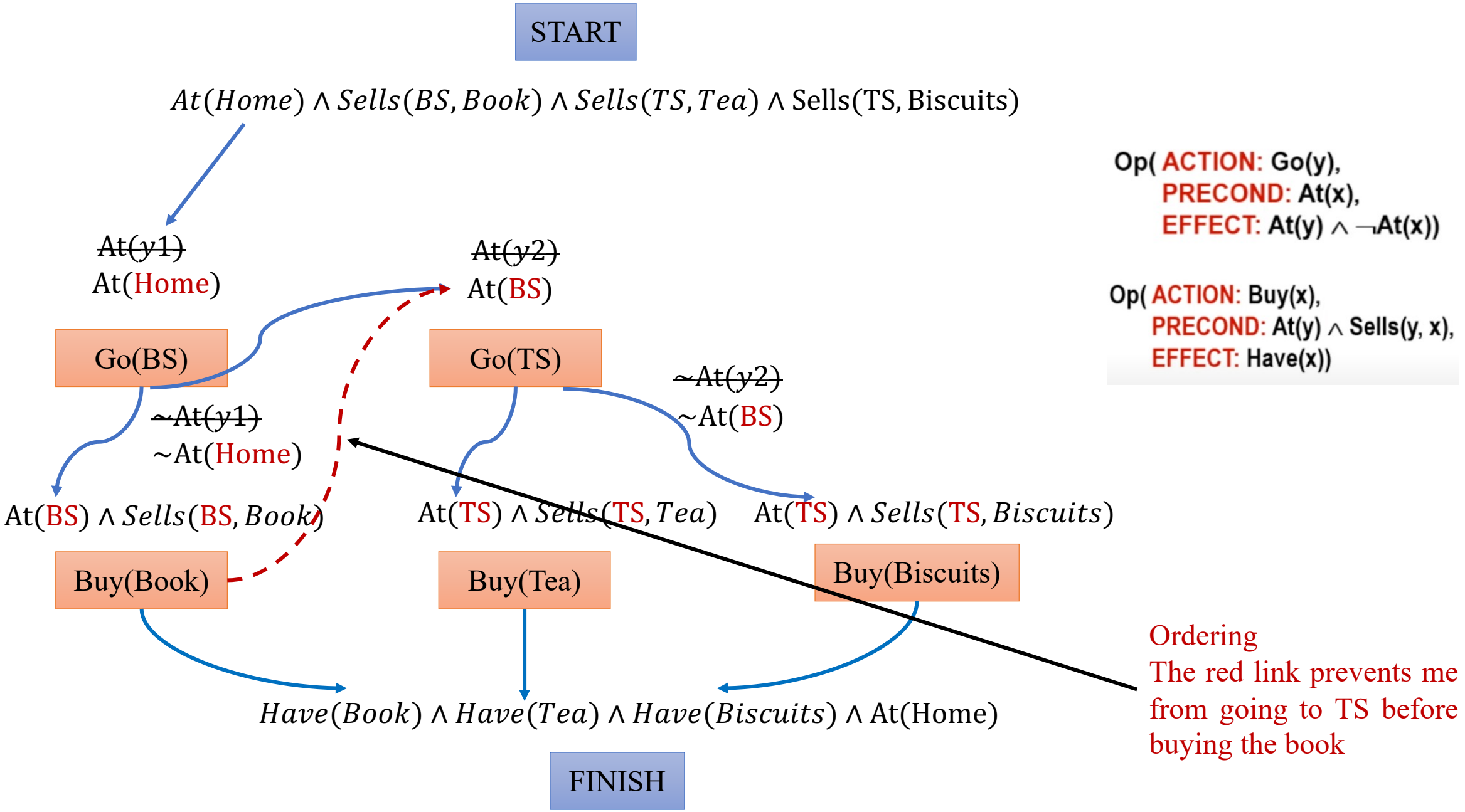
Op(ACTION: Buy(x),
PRECOND: At(y) $\wedge$ Sells(y, x),
EFFECT: Have(x))

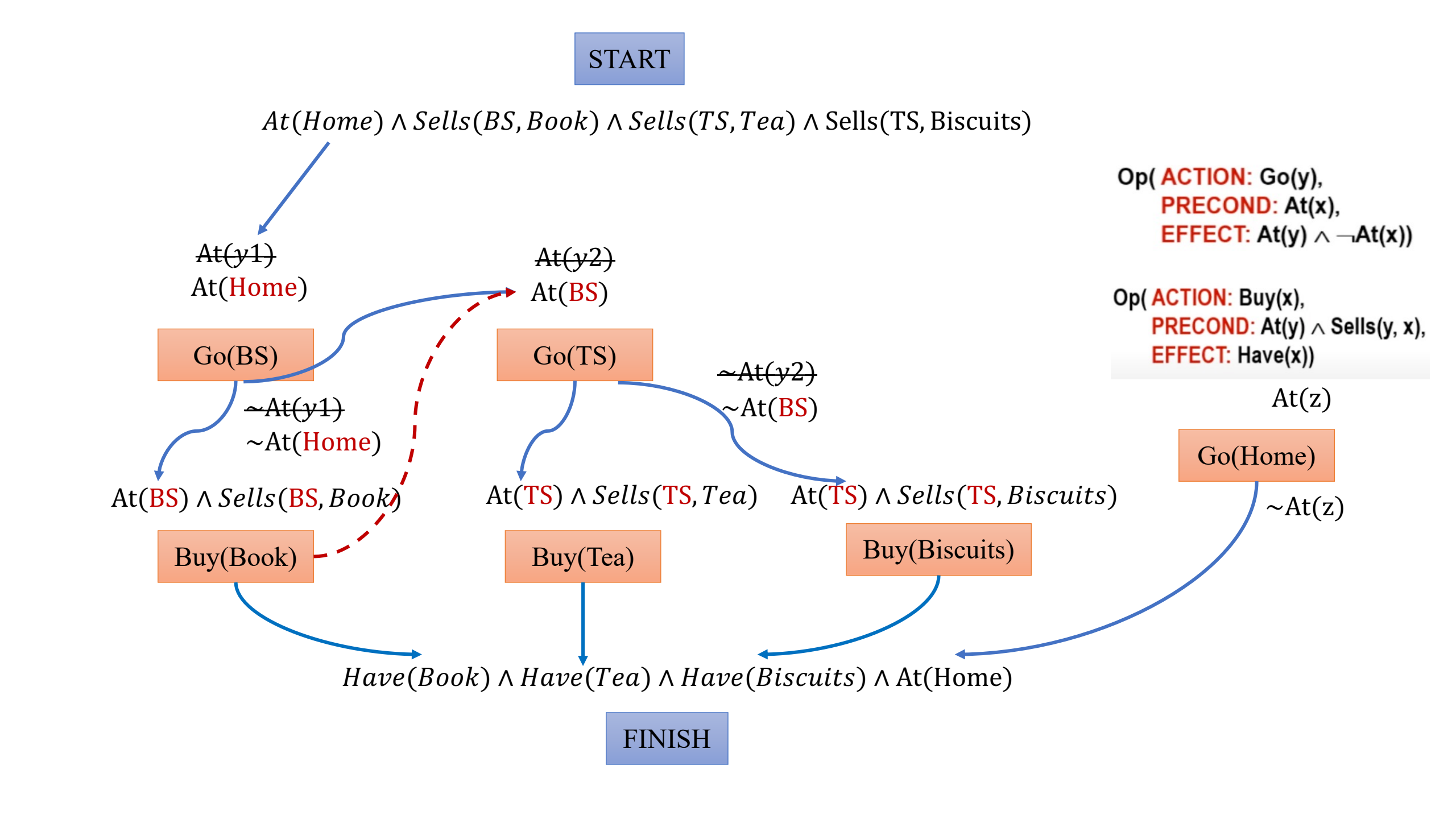

START
At(Home) ∧ Sells(BS, Book) ∧ Sells(TS, Tea) ∧ Sells(TS, Biscuits)
At(y1)
At(Home)
At(y2)
Go(BS)
Go(TS)
~At(y2)
~At(y1)
~At(Home)
At(BS) ∧ Sells(BS, Book)
At(TS) ∧ Sells(TS, Tea)
At(TS) ∧ Sells(TS, Biscuits)
Buy(Book)
Buy(Tea)
Buy(Biscuits)
Have(Book) ∧ Have(Tea) ∧ Have(Biscuits) ∧ At(Home)
FINISH
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) ∧ ¬At(x))
Op(ACTION: Buy(x),
PRECOND: At(y) ∧ Sells(y, x),
EFFECT: Have(x))

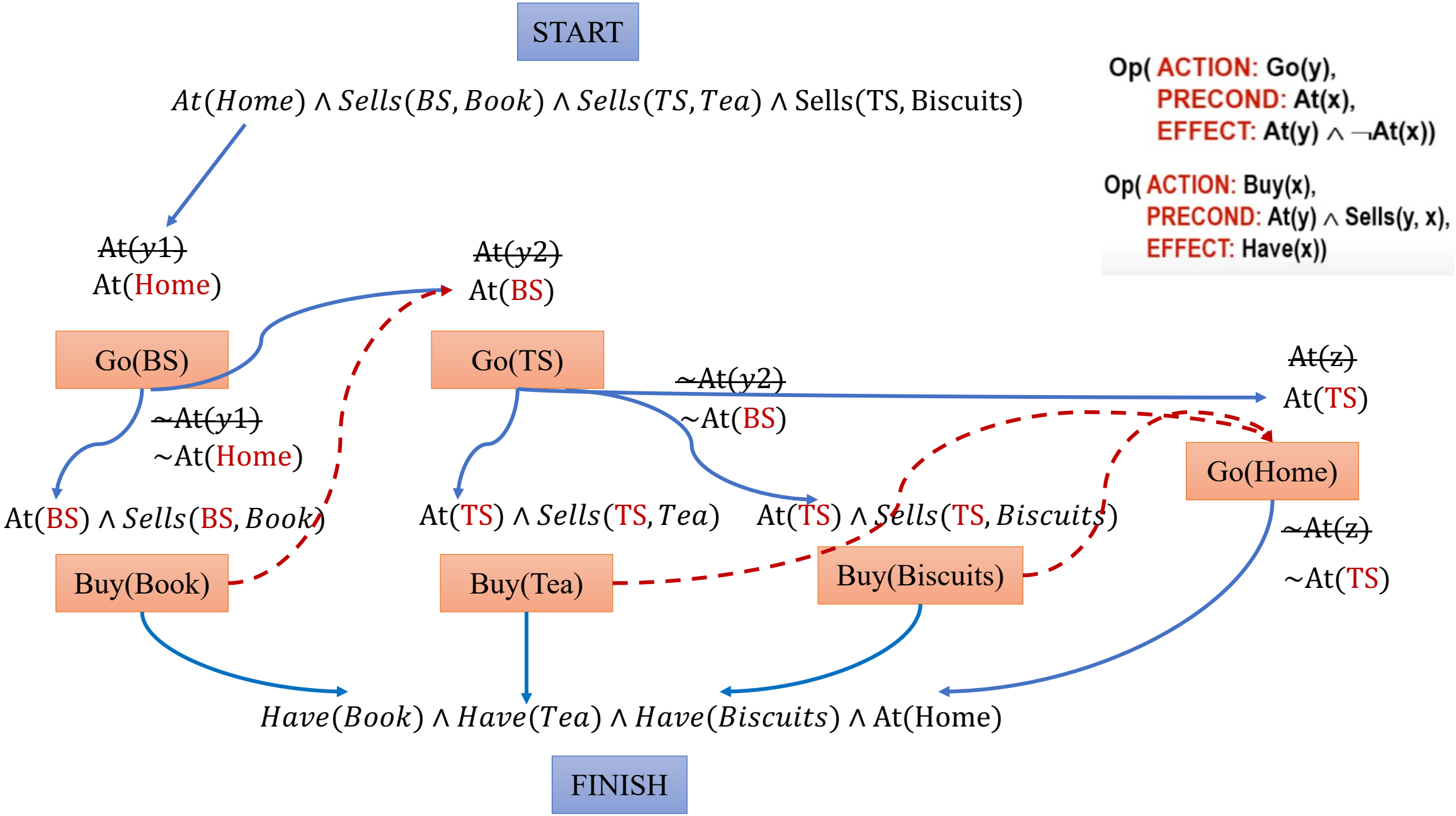


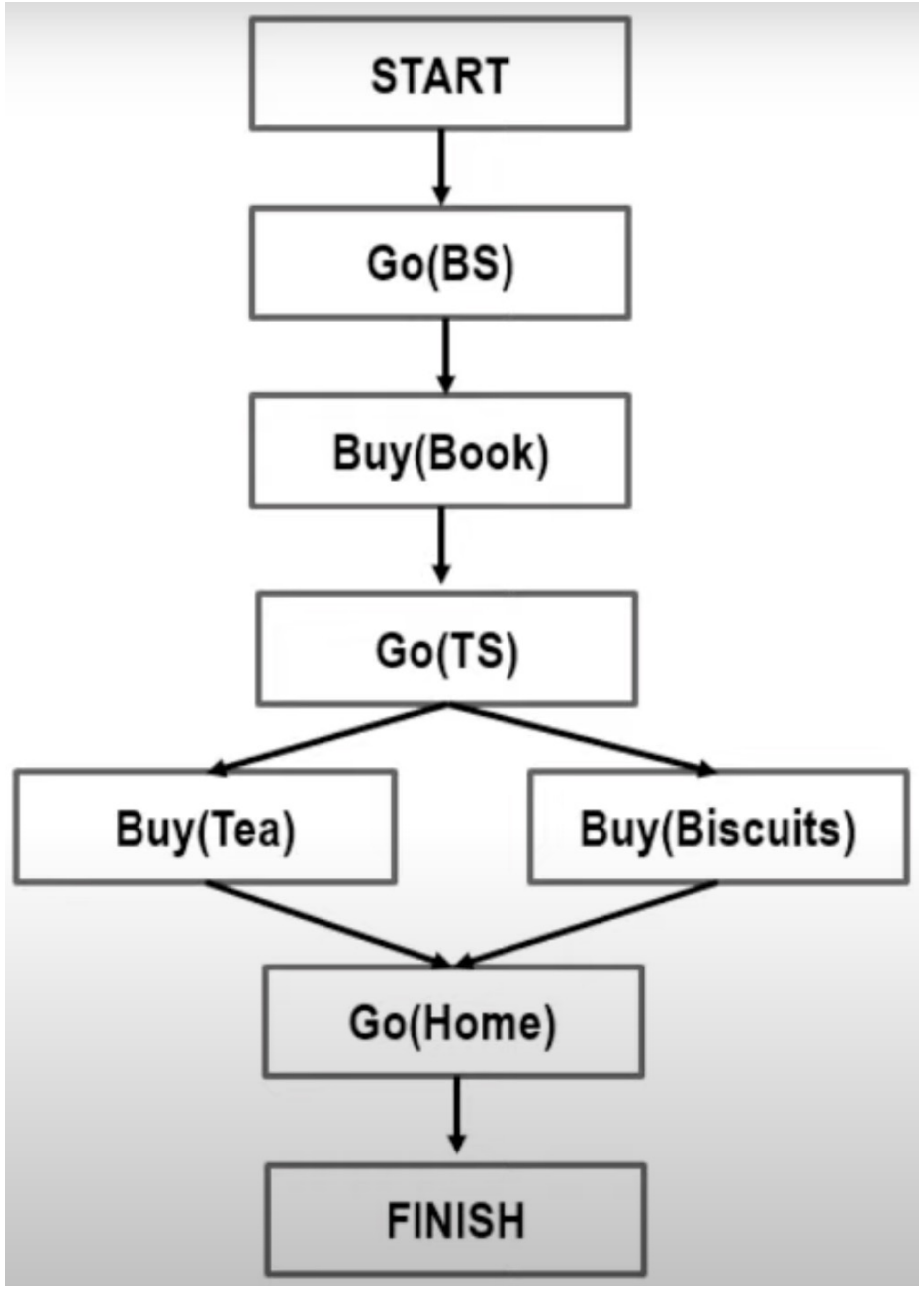
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) $\wedge \neg$At(x))

Op(ACTION: Buy(x),
PRECOND: At(y) $\wedge$ Sells(y, x),
EFFECT: Have(x))


START
$At(Home) \wedge Sells(BS, Book) \wedge Sells(TS, Tea) \wedge$ Sells(TS, Biscuits)
~~At(y1)~~
At(Home)
~~At(y2)~~
At(BS)
Go(BS)
Go(TS)
~~~At(y1)~~
~At(Home)
~~~At(y2)~~
~At(BS)
At(BS) $\wedge Sells(BS, Book)$
At(TS) $\wedge Sells(TS, Tea)$
At(TS) $\wedge Sells(TS, Biscuits)$
Buy(Book)
Buy(Tea)
Buy(Biscuits)
$Have(Book) \wedge Have(Tea) \wedge Have(Biscuits) \wedge$ At(Home)
FINISH
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) ∧ ¬At(x))
Op(ACTION: Buy(x),
PRECOND: At(y) ∧ Sells(y, x),
EFFECT: Have(x))
Ordering
The red link prevents me from going to TS before buying the book


START
At(Home) ∧ Sells(BS, Book) ∧ Sells(TS, Tea) ∧ Sells(TS, Biscuits)
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) ∧ ¬At(x))
Op(ACTION: Buy(x),
PRECOND: At(y) ∧ Sells(y, x),
EFFECT: Have(x))
At(y1)
At(Home)
At(y2)
At(BS)
Go(BS)
Go(TS)
~At(y1)
~At(Home)
~At(y2)
~At(BS)
At(z)
Go(Home)
At(BS) ∧ Sells(BS, Book)
At(TS) ∧ Sells(TS, Tea)
At(TS) ∧ Sells(TS, Biscuits)
~At(z)
Buy(Book)
Buy(Tea)
Buy(Biscuits)
Have(Book) ∧ Have(Tea) ∧ Have(Biscuits) ∧ At(Home)
FINISH


START
At(Home) ∧ Sells(BS, Book) ∧ Sells(TS, Tea) ∧ Sells(TS, Biscuits)
Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) ∧ ¬At(x))
Op(ACTION: Buy(x),
PRECOND: At(y) ∧ Sells(y, x),
EFFECT: Have(x))
At(y1)
At(Home)
At(y2)
At(BS)
Go(BS)
Go(TS)
At(z)
At(TS)
~At(y2)
~At(BS)
~At(y1)
~At(Home)
Go(Home)
At(BS) ∧ Sells(BS, Book)
At(TS) ∧ Sells(TS, Tea)
At(TS) ∧ Sells(TS, Biscuits)
~At(z)
~At(TS)
Buy(Book)
Buy(Tea)
Buy(Biscuits)
Have(Book) ∧ Have(Tea) ∧ Have(Biscuits) ∧ At(Home)
FINISH



- Ordering
- Causal Constraints

Example

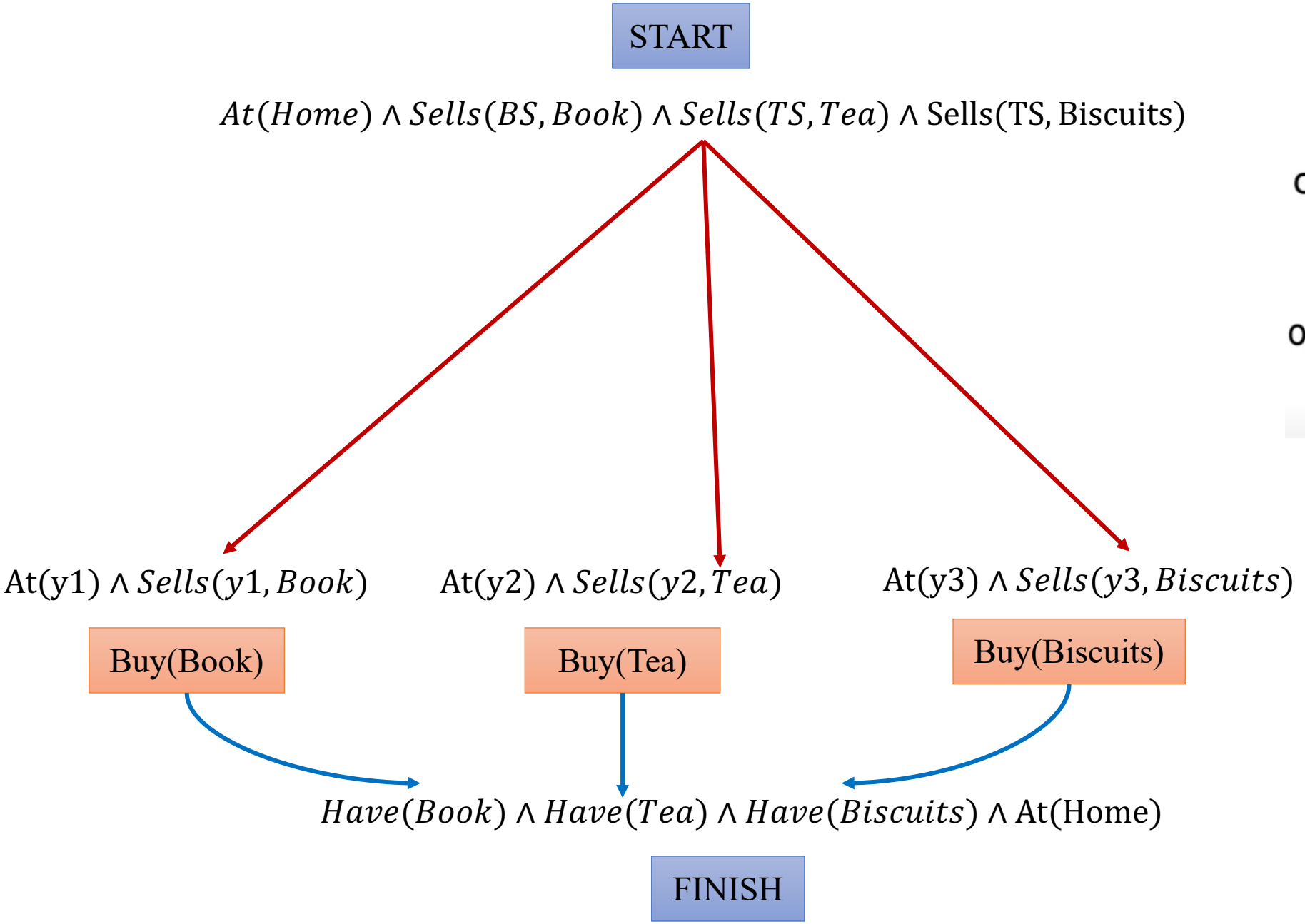
- Initial Plan
- Plan(
  - STEPS: {
    - $S1$: $Op($ $ACTION$: $Start$,
      - $EFFECT$: $At(Home) \wedge Sells(BS, Book) \wedge Sells(TS, Tea) \wedge Sells(TS, Biscuits)$),
    - $S2$: $Op($ $ACTION$: $Finish$,
      - $PRECOND$: $At(Home) \wedge Have(Tea) \wedge Have(Biscuits) \wedge Have(Book)$),
    - },
  - ORDERINGS: $\{S_1 \prec S_2\}$,
  - BINDINGS: {},
  - LINKS: {}
  -)

Partial Order Planning Algorithm

- Function POP(initial, goal, operators)
- // Returns plan
  - $plan \leftarrow Make - Minimal - Plan(initial, goal)$
  - $Loop\ do$
    - $if\ Solution(plan)\ then\ return\ plan$
    - $S, c \leftarrow Select - Subgoal(plan)$
    - $Choose - Operator(plan, operators, S, c)$
    - $Resolve - Threats(plan)$
  - End

POP: Selecting-Subgoals

- Function Select-Subgoal(plan)
- //Returns S, c
  - $pick\ a\ plan\ step\ S\ from\ STEPS(plan)$
    - $with\ a\ precondition\ c\ that\ has\ not\ been\ achieved$
  - $Return\ S, c$



Op(ACTION: Go(y),
PRECOND: At(x),
EFFECT: At(y) $\wedge \neg$At(x))

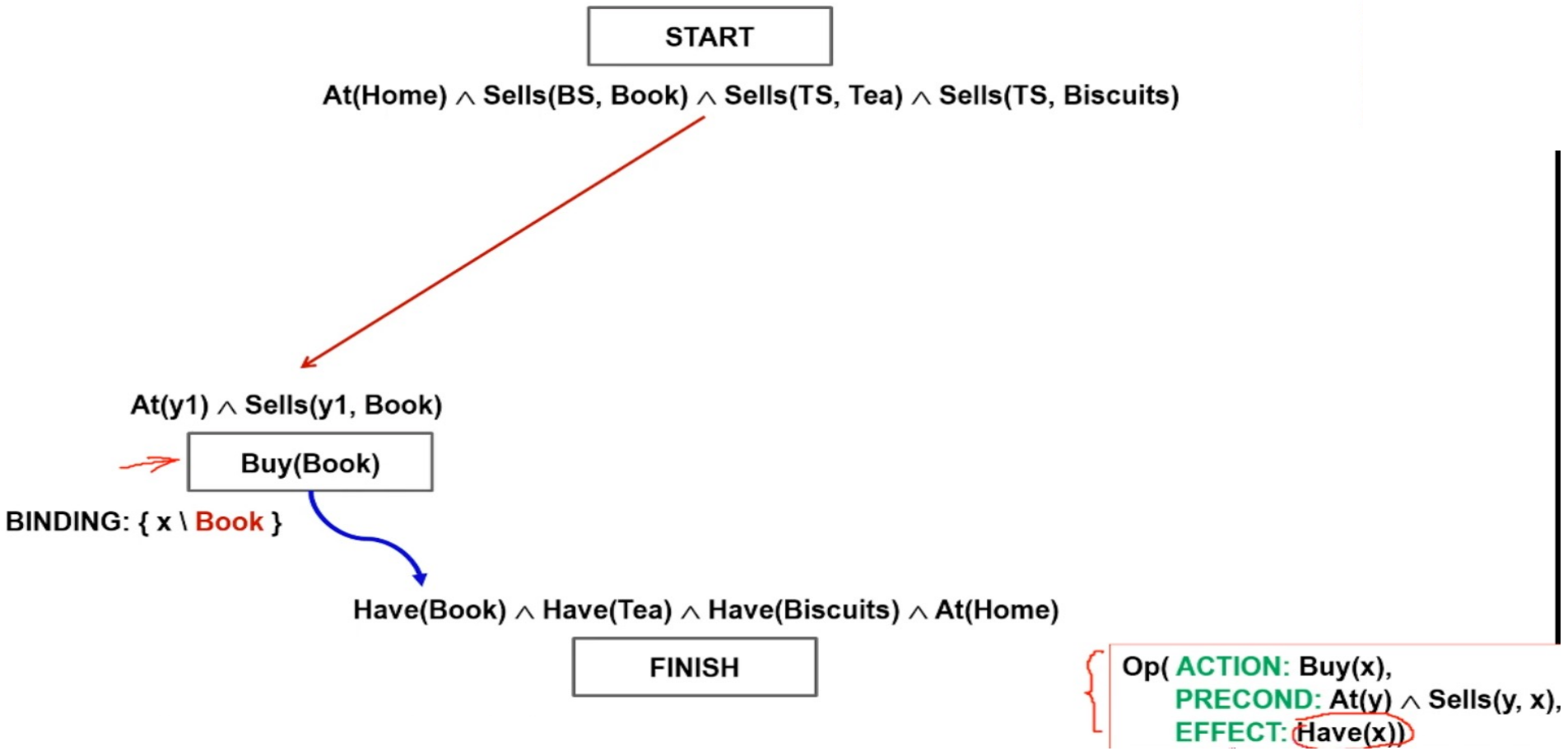
Op(ACTION: Buy(x),
PRECOND: At(y) $\wedge$ Sells(y, x),
EFFECT: Have(x))

POP: Choosing Operators

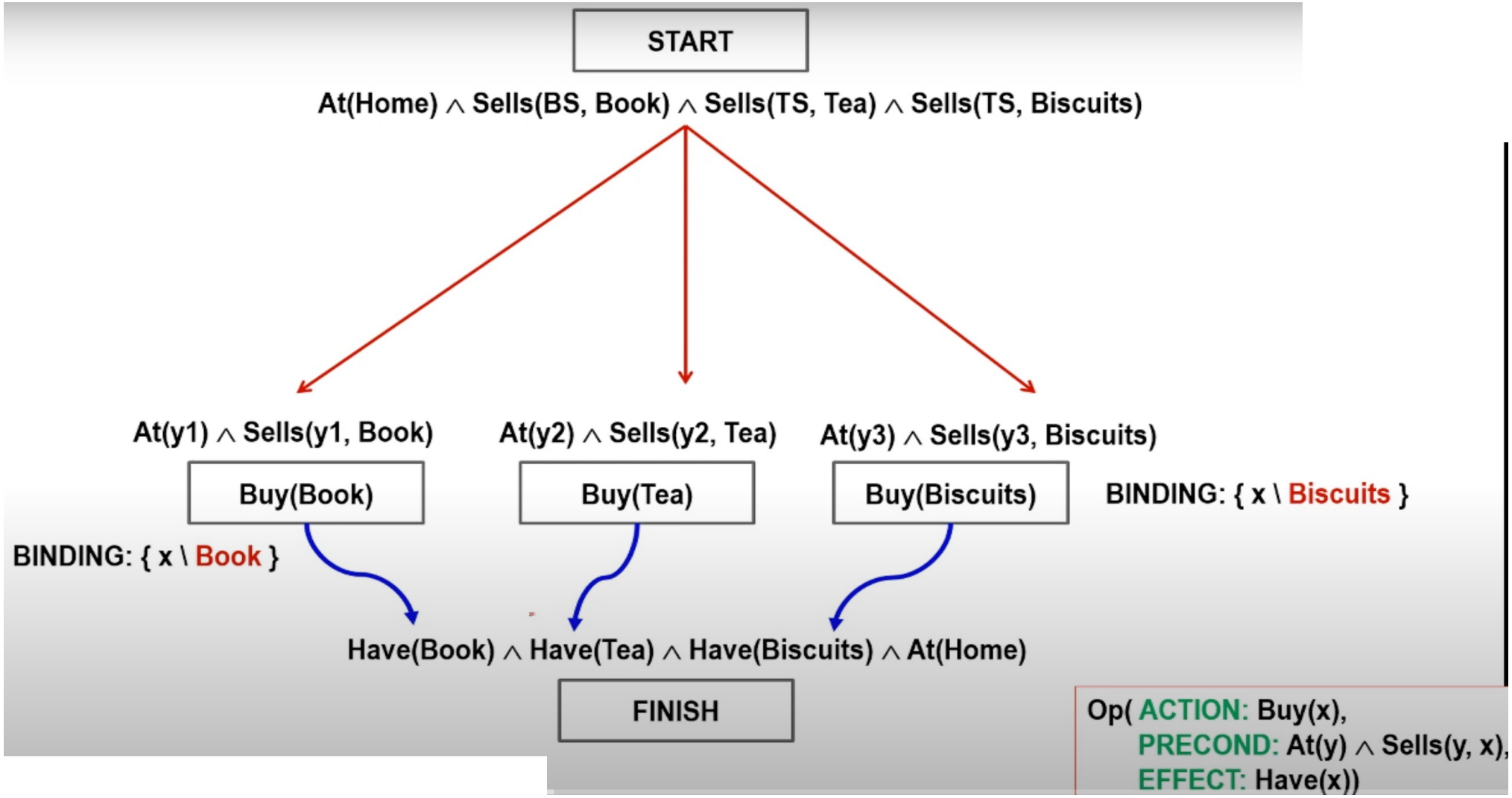
- Procedure Choose-Operator(plan, operators, S, c)
  - $choose\ a\ step\ S'\ from\ operators\ or\ STEPS(plan)\ that\ has\ c\ as\ an\ effect$
  - $if\ there\ is\ no\ such\ step\ \rightarrow fail$
  - $add\ the\ causal\ link\ S' \rightarrow c: S\ to\ LINKS(plan)$
  - $add\ the\ ordering\ constraint\ S' \prec S\ to\ ORDERINGS(plan)$

  - $if\ S\ is\ a\ newly\ added\ step\ from\ operators\ then\ add\ S\ to\ STEPS(plan)\ and\ add$
  - $Start \prec S' \prec Finish\ to\ ORDERINGS(plan)$

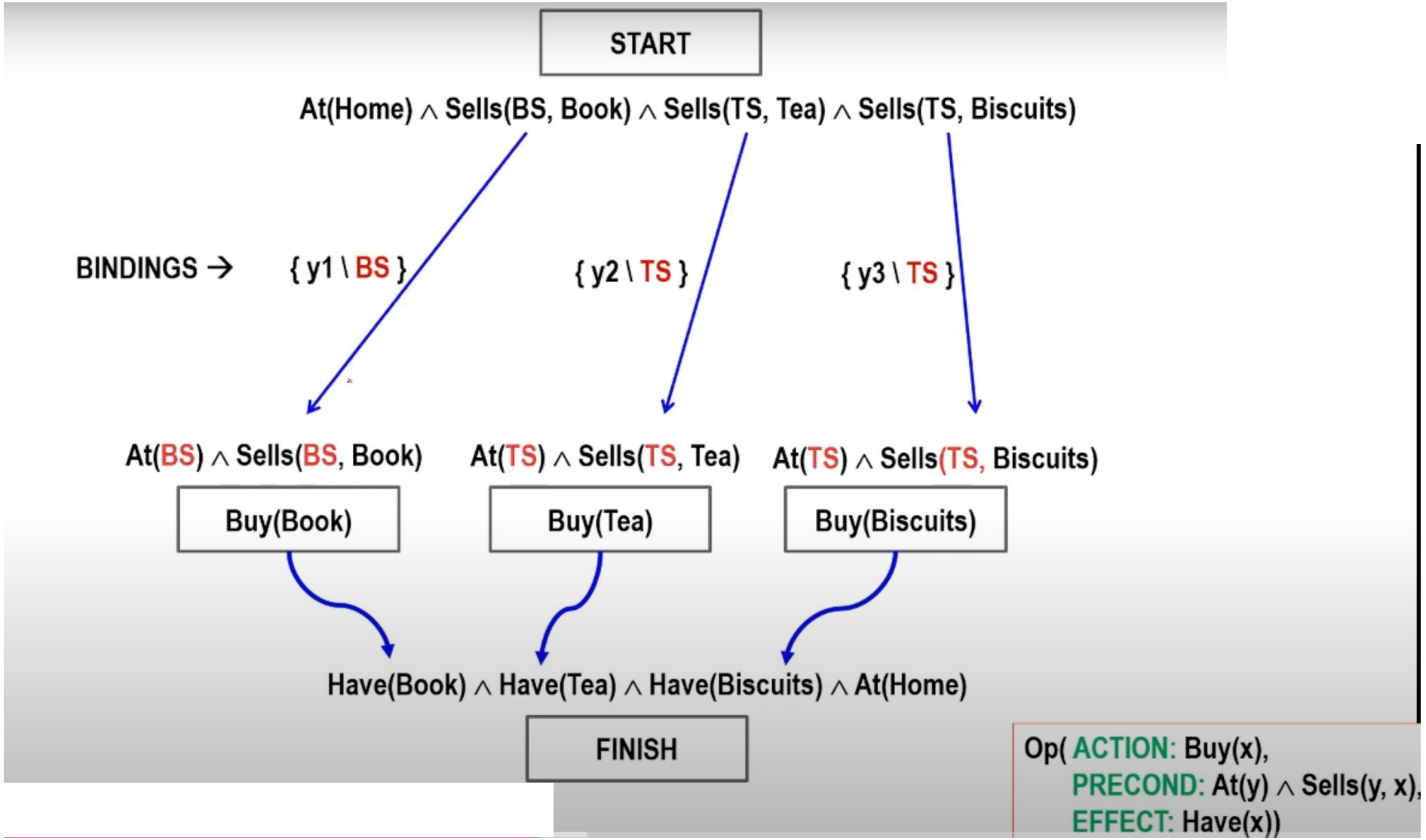
POP: Choosing Operators



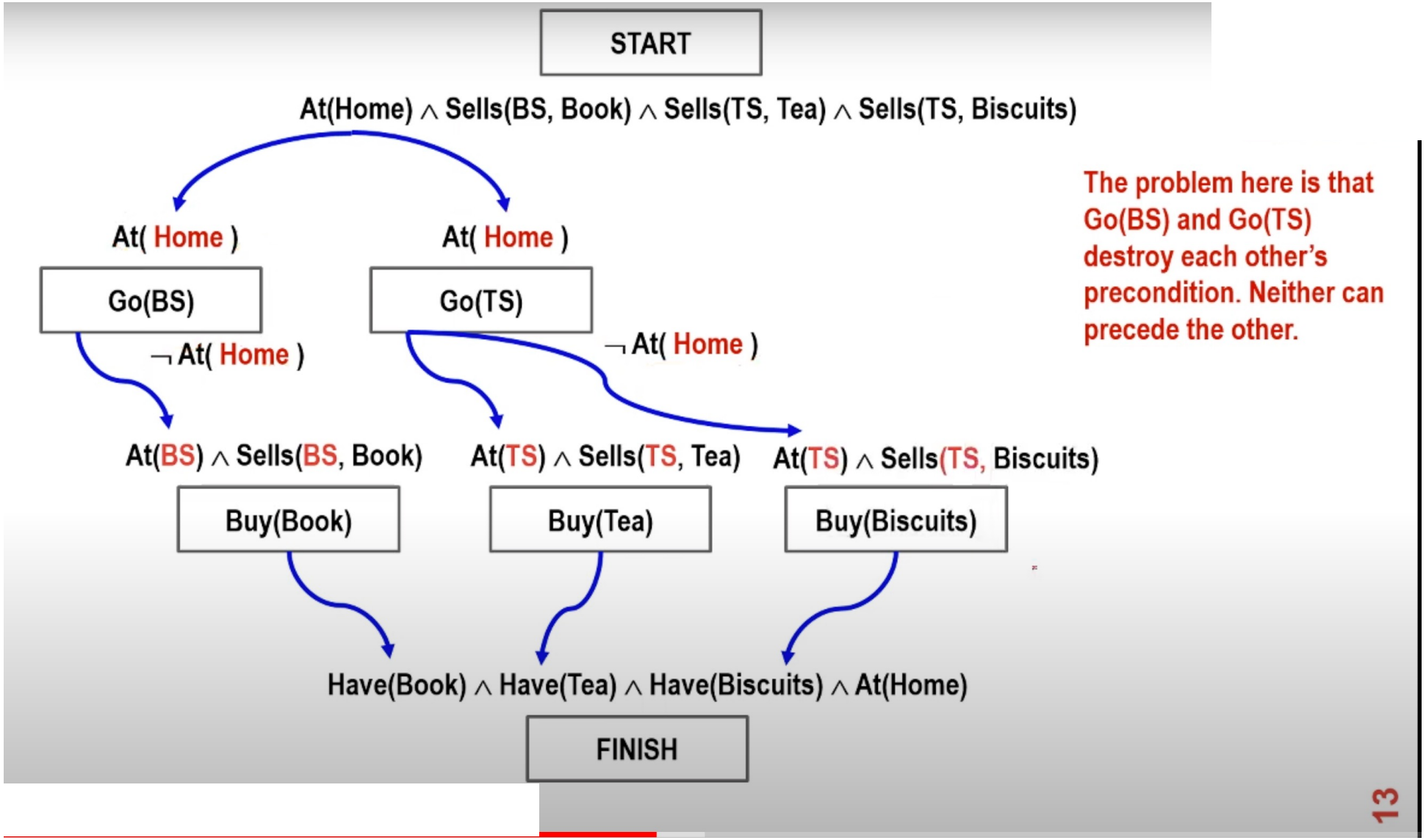
POP: Choosing Operators



POP: Solve Goal via Variable Binding

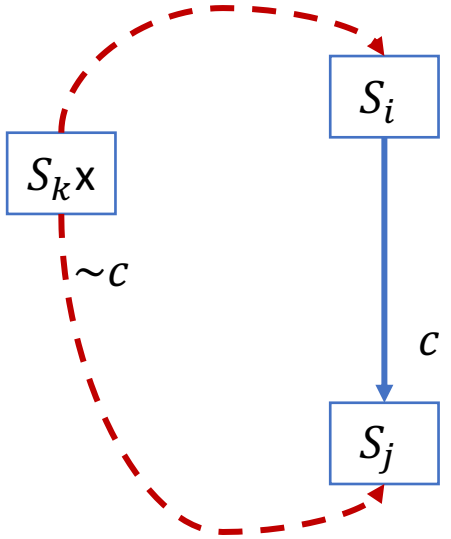


POP: Choosing Operators

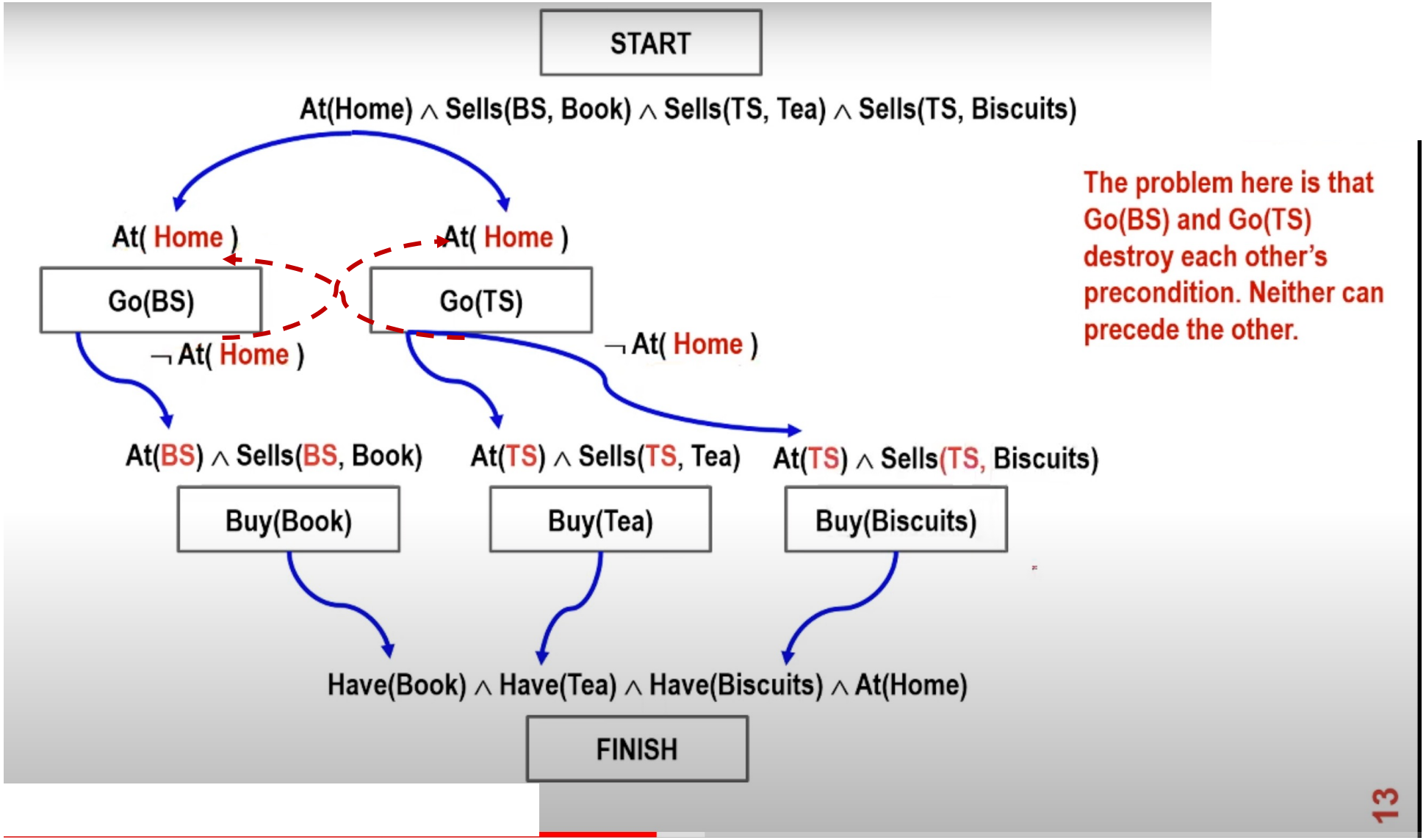


POP: Resolving Threats

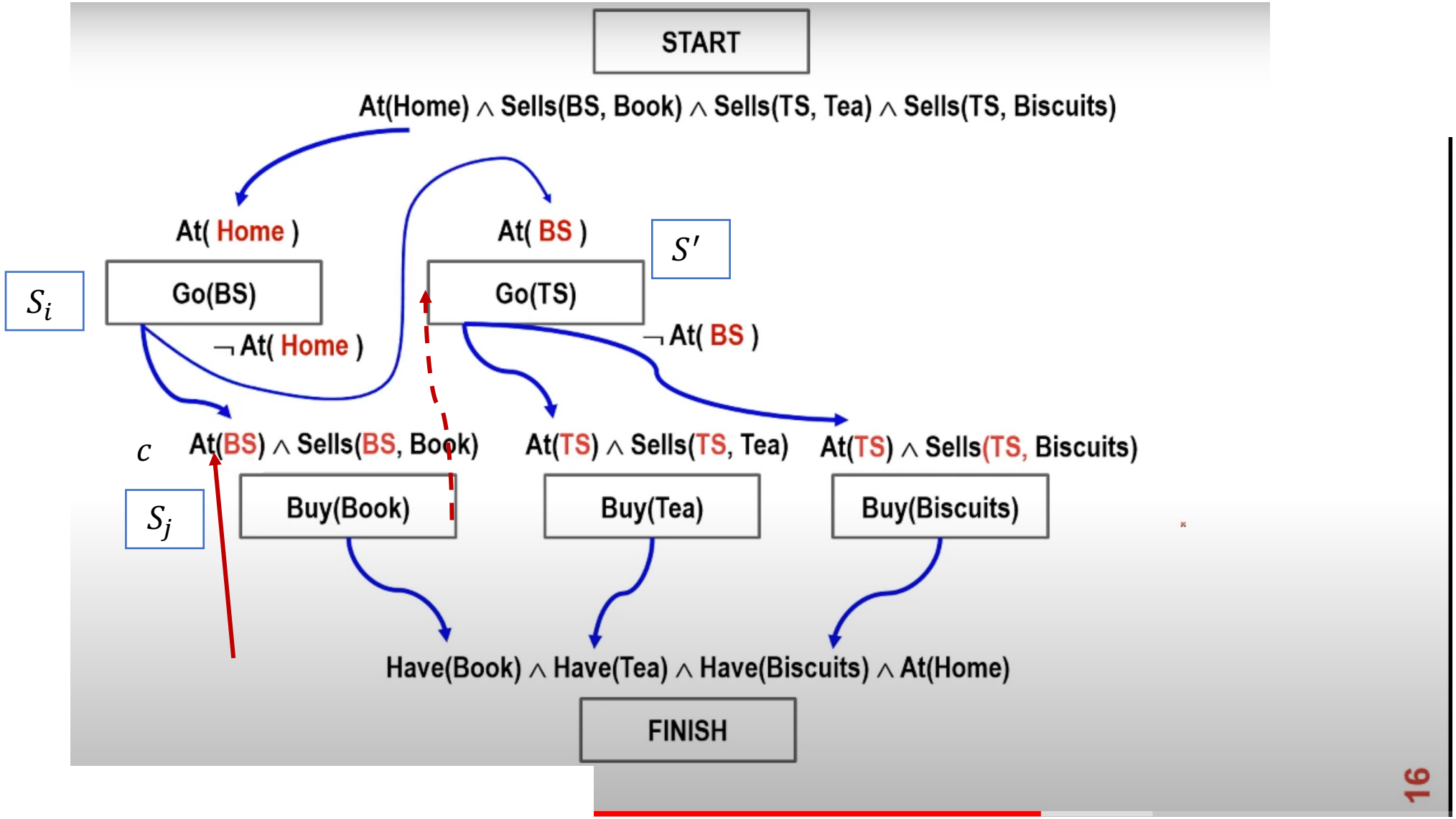
- Procedure Resolve-Threats(plan)
  - $for\ each\ S'\ that\ threatens\ a\ link\ S_i \rightarrow c: S_j\ in\ LINKS(plan)\ do$
    - Choose either
      - $Promotion: Add\ S' \prec S_i\ to\ ORDERINGS(plan)$
      - $Demotion: Add S_j \prec S'\ to\ ORDERINGS(plan)$
  - $if\ not\ Consistent(plan) \rightarrow fail$



POP: Resolving Threats

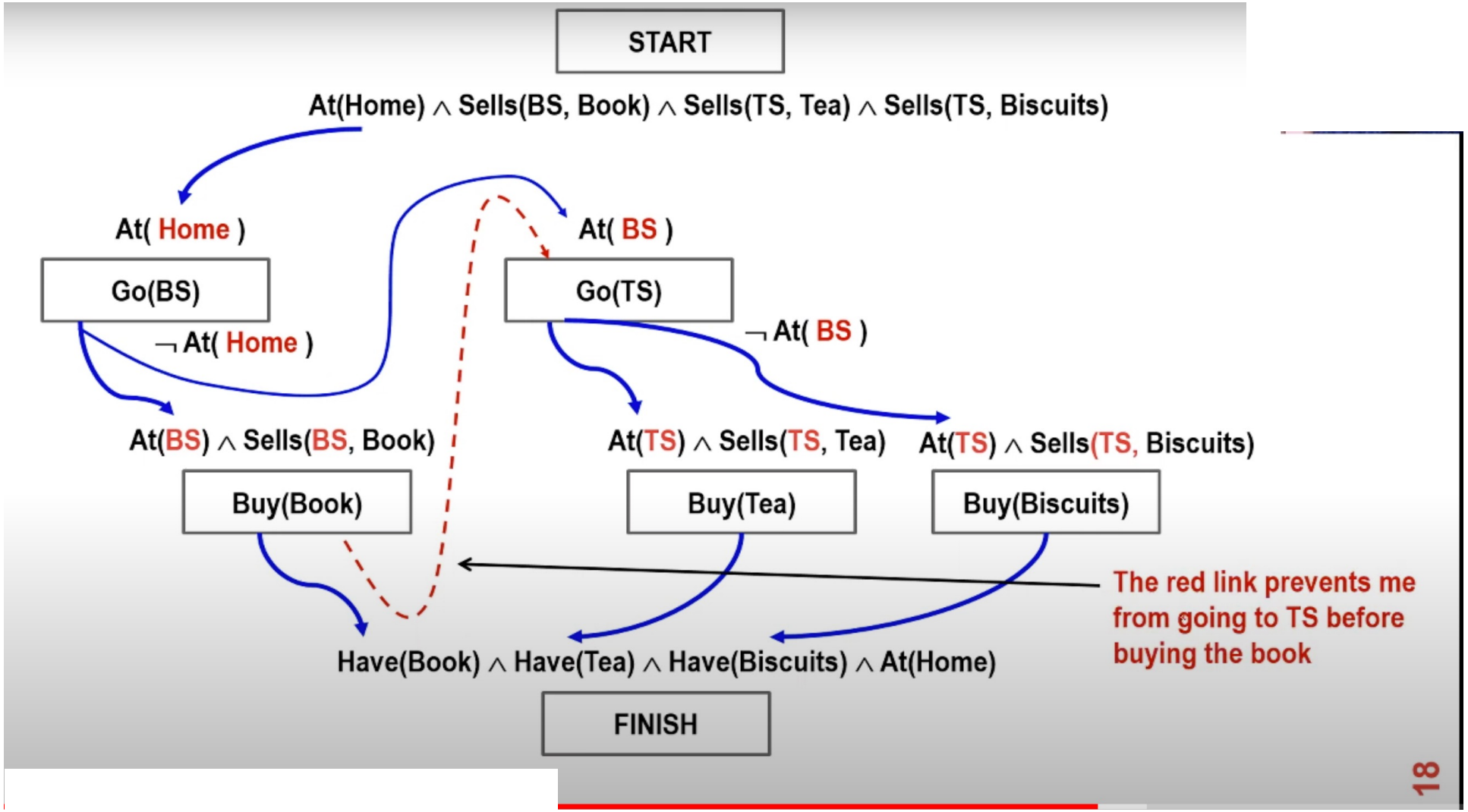


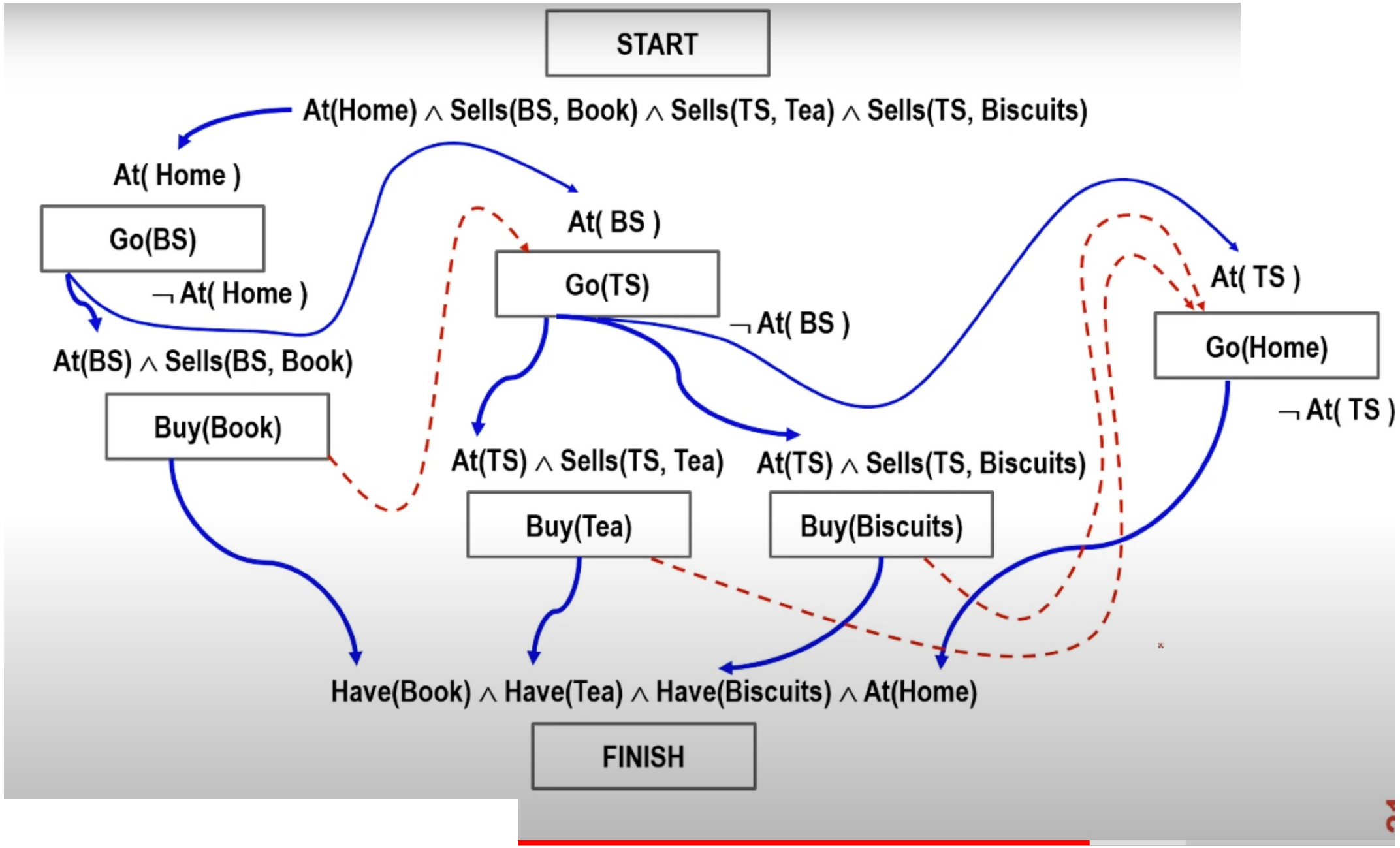
POP: Resolving Threats



Demotion of S'

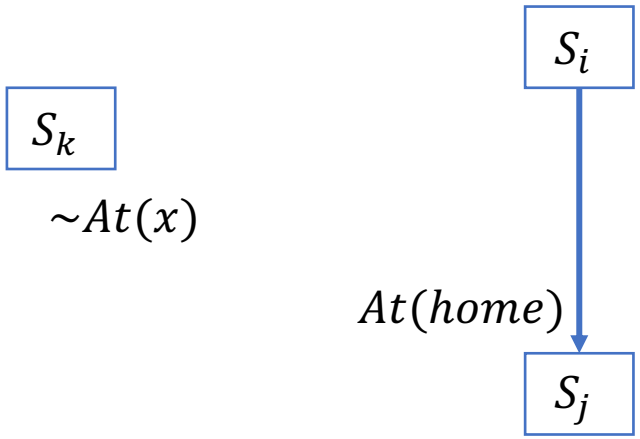
POP: Resolving Threats




START
At(Home) ∧ Sells(BS, Book) ∧ Sells(TS, Tea) ∧ Sells(TS, Biscuits)
At(Home)
Go(BS)
¬ At(Home)
At(BS) ∧ Sells(BS, Book)
Buy(Book)
At(BS)
Go(TS)
¬ At(BS)
At(TS) ∧ Sells(TS, Tea)
Buy(Tea)
At(TS) ∧ Sells(TS, Biscuits)
Buy(Biscuits)
At(TS)
Go(Home)
¬ At(TS)
Have(Book) ∧ Have(Tea) ∧ Have(Biscuits) ∧ At(Home)
FINISH

Partially Instantiated Operators

- So far we have not mentioned anything about binding constraints
- Should an operator that has the effect, say, $\sim At(x)$ be considered a threat to the condition $At(Home)$?
  - Indeed it is a possible threat because x may be bound to Home



Dealing with potential threats

- Resolve now with an equality constraint
  - Blind x to something that resolves the threat (say $x = TS$)

- Resolve now with an inequality constraint
  - Extend the language of variable binding to allow $x \neq Home$

- Resolve later
  - Ignore possible threats
  - If $x = Home$ is added later into the plan, then we will attempt to resolve the threat (by promotion or demotion)

POP: Choosing Operators

- Procedure Choose-Operator(plan, operators, S, c)
  - $choose\ a\ step\ S'\ from\ operators\ or\ STEPS(plan)\ that\ has\ c\ as\ an\ effect$
  - $if\ there\ is\ no\ such\ step\ \rightarrow fail$
  - $add\ the\ causal\ link\ S' \rightarrow c: S\ to\ LINKS(plan)$
  - $add\ the\ ordering\ constraint\ S' \prec S\ to\ ORDERINGS(plan)$

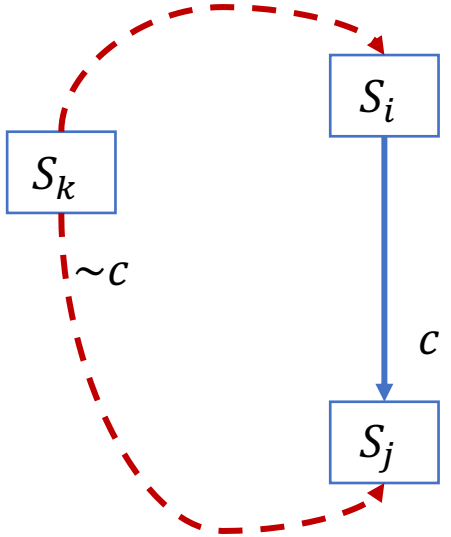
  - $if\ S\ is\ a\ newly\ added\ step\ from\ operators\ then\ add\ S\ to\ STEPS(plan)\ and\ add$
  - $Start \prec S' \prec Finish\ to\ ORDERINGS(plan)$

POP: Choosing Operators

- Procedure Choose-Operator(plan, operators, S, c)
  - $choose\ a\ step\ S'\ from\ operators\ or\ STEPS(plan)\ that\ has\ c'\ as\ an\ effect$
    - $such\ that\ u = UNIFY(c, c', BINDINGS(plan))$
  - $if\ there\ is\ no\ such\ step\ \rightarrow fail$
  - $add\ u\ to\ BINDINGS(plan)$
  - $add\ the\ causal\ link\ S' \rightarrow c{:}\ S\ to\ LINKS(plan)$
  - $add\ the\ ordering\ constraint\ S' \prec S\ to\ ORDERINGS(plan)$
  - $if\ S\ is\ a\ newly\ added\ step\ from\ operators\ then\ add\ S\ to\ STEPS(plan)\ and\ add$
  - $Start \prec S' \prec Finish\ to\ ORDERINGS(plan)$

POP: Resolving Threats

- Procedure Resolve-Threats(plan)
  - $for\ each\ S'\ that\ threatens\ a\ link\ S_i \rightarrow c: S_j\ in\ LINKS(plan)\ do$
    - Choose either
      - $Promotion: Add\ S' \prec S_i\ to\ ORDERINGS(plan)$
      - $Demotion: Add S_j \prec S'\ to\ ORDERINGS(plan)$
  - $if\ not\ Consistent(plan) \rightarrow fail$



POP: Resolving Threats

- Procedure Resolve-Threats(plan)
  - $for\ each\ S_i \rightarrow c : S_j\ in\ LINKS(plan)\ do$
    - $for\ each\ S'\ in\ STEPS(plan)\ do$
      - $for\ each\ c'\ in\ EFFECTS(S')\ do$
        - $if\ SUBST(BINDINGS(plan), c) = SUBST(BINDINGS(plan), \sim c')$
        - Choose either
          - $Promotion: Add\ S' \prec S_i\ to\ ORDERINGS(plan)$
          - $Demotion: Add\ S_j \prec S'\ to\ ORDERINGS(plan)$
        - $if\ not\ Consistent(plan) \rightarrow fail$

Thank You