Project Part 1 requires you to complete an **anytime, forward-searching, depth-bounded**, **utility-driven** scheduler.

```
procedure Search(G, S, ~~goal)~~        D: depth bound
    Inputs
        G: graph with nodes N and arcs A
        s: start node                      U: utility function (applied to a path, not a single node)
        goal: Boolean function of nodes
    Output
        path from s to a node for which goal is true
        or ⊥ if there are no solution paths
    Local
        Frontier: set of paths             ; Solutions: Priority Queue of solutions organized by solution "quality", presumably by U
    Frontier := {⟨s⟩}                       ; Solutions := Empty Priority Queue
    while Frontier ≠ {} do
        select and remove ⟨n_0, ..., n_k⟩ from Frontier
        if depth(n_k) >= D then
            ~~return ⟨n_0, ..., n_k⟩~~       ; add <n0,…,nk> to Solutions using solution "quality", presumably by U
    else Frontier := Frontier ∪ {⟨n_0, ..., n_k, n⟩ : ⟨n_k, n⟩ ∈ A}    //generate successors    + scheduled operations that transition from state to state
    return ⊥
```

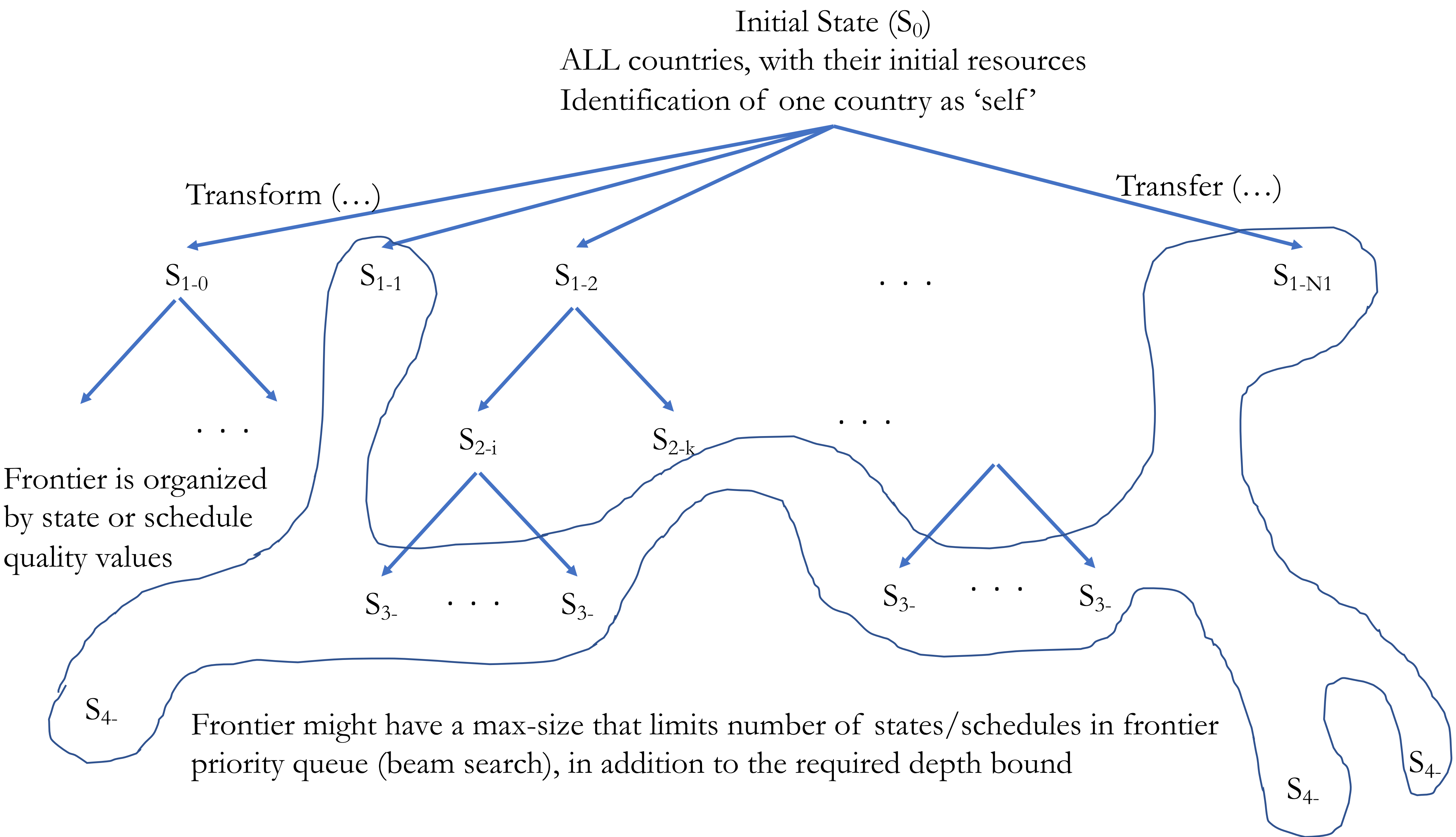
Figure 3.4: Search: generic graph searching algorithm

Adapted from http://artint.info/2e/html/ArtInt2e.Ch3.S4.html

Comments on Part 1 Specification

https://my.vanderbilt.edu/cs4269aiproject/project-description-and-logistics/

and https://my.vanderbilt.edu/cs4269aiproject/project-part-1-spring2021/



	A	B	C	D	E	F
1	Resource	Weight	Notes			
2	R1	0	analog to population			
3	R2	0	analog to metallic elements			
4	R3	0	analog to timber			
5	R21	0.2	analog to metallic alloys			
6	R22	0.5	analog to electronics			
7	R23	0.8	analog to housing (and housing sufficiency)			
8	R21'	-0.5	waste			
9	R22'	-0.8	waste			
10	R23'	-0.4	waste			

resources file

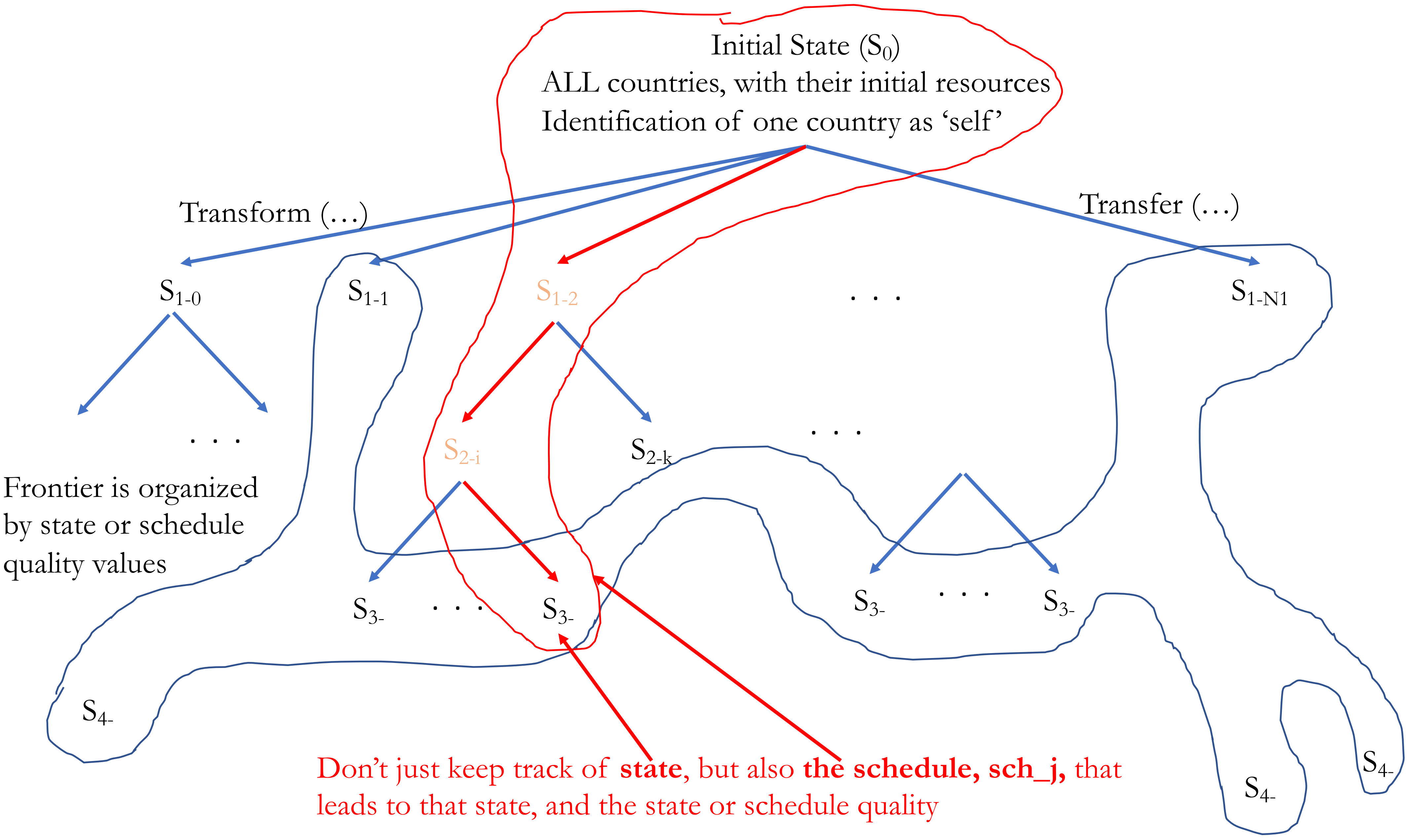
You can modify with additional or alternative columns, with explanation.

	A	B	C	D	E	F	G	H	I	J
1	Country	R1	R2	R3	R21	R22	R23	R21'	R22'	R23'
2	Atlantis	100	700	2000	0	0	0	0	0	0
3	Brobdingnag	50	300	1200	0	0	0	0	0	0
4	Carpania	25	100	300	0	0	0	0	0	0
5	Dinotopia	30	200	200	0	0	0	0	0	0
6	Erewhon	70	500	1700	0	0	0	0	0	0

initial_state

Modify as needed by additional/alternative resources and countries.

Self = Atlantis, for example


Initial State (S_0)
ALL countries, with their initial resources
Identification of one country as 'self'
Transform (…)
Transfer (…)
S_{1-0}
S_{1-1}
S_{1-2}
. . .
S_{1-N1}
. . .
S_{2-i}
S_{2-k}
. . .
Frontier is organized by state or schedule quality values
S_{3-}
. . .
S_{3-}
S_{3-}
. . .
S_{3-}
S_{4-}
S_{4-}
S_{4-}
Don't just keep track of **state**, but also **the schedule, sch_j,** that leads to that state, and the state or schedule quality

Possible Pseudocode for Generate Successors

```
Successors ← { }
For each (skeletal, variablized) operator (i.e., TRANSFER and each TRANSFORM template), ?Op {
    For each variable ?X in ?Op {
        For each constant, K, of the appropriate type (i.e., country, resource, amount) {
            Substitute K for ?X in ?Op
        }
    } // when done, all variables in ?Op replaced by constants, yielding Op
    If preconditions of Op satisfied, apply Op to current world, and add successor to set of successors
}
```

How many successors (ballpark) will there be: (P ?ops) * (M vars per ?op) * (N vals per var) = P*M*N

So, in our **toy problem** of 6 countries, 9 resources, and assuming only 3 possible values per resource (lets say and average of 6 values per variable), that's

4 templates * 4 variables per template * 6 values per variable, or say 4 * 4 * 6, on the order of **100 successors**

Alloys Template

((TRANSFORM ?C (INPUTS (R1 1) (R2, 2)) (OUTPUTS (R1 1) (R21, 1) (R21’ 1)),
preconditions are of the form ?ARj <= ?C(?Rj)

. . .

Electronics Template

(TRANSFORM ?C (INPUTS (R1 3) (R2 2) (R21 2)) (OUTPUTS (R22 2) (R22’ 2) (R1 3)),
preconditions are of the form ?ARj <= ?C(?Rj)

. . .

Housing Template

(TRANSFORM ?C (INPUTS (R1 5) (R2, 1) (R3 5) (R21 3) (OUTPUTS (R1 5) (R23, 1) (R23’ 1)),
preconditions are of the form ?AIk <= ?C(?Rk)

. . .

(TRANSFER ?Cj1 ?Cj2 ((?Ri ?ARi)), where ?ARi <= ?Cj1(?Ri)

. . .

A(tlantis)	E(rewon)
R1: 500	R1: 100
R2: 700	R2: 50
R3: 100	R3: 2000
R21: 0	R21: 30
R21’: 0	R21’: 0
R22: 0	R22: 0
R22’: 0	R22’: 0
R23: 0	R23: 0
R23’: 0	R23’: 0

State, n_k

A(tlantis)	E(rewon)
R1: 500	R1: 100
R2: 700	R2: 50
R3: 100	R3: 2000
R21: 0	R21: 30
R21’: 0	R21’: 0
R22: 0	R22: 0
R22’: 0	R22’: 0
R23: 0	R23: 0
R23’: 0	R23’: 0

Alloys Template

((TRANSFORM ?C (INPUTS (R1 1) (R2, 2)) (OUTPUTS (R1 1) (R21, 1) (R21’ 1)),
preconditions are of the form ?ARj <= ?C(?Rj)

(TRANSFORM **A** (INPUTS (R1 50*1) (R2, 50*2)) (OUTPUTS (R1 50) (R21, 50) (R21’ 50)),
preconditions 50 <= 500, 100 <= 700

• • •

Electronics Template

(TRANSFORM ?C (INPUTS (R1 3) (R2 2) (R21 2)) (OUTPUTS (R22 2) (R22’ 2) (R1 3)),
preconditions are of the form ?ARj <= ?C(?Rj)

(TRANSFORM **A** (INPUTS (R1 30) (R2 20) (R21 20)) (OUTPUTS (R22 20) (R22’ 20) (R1 30)),
preconditions 30 <= 500, 20 <= 700, 20 !<= 0

• • •

Housing Template

(TRANSFORM ?C (INPUTS (R1 5) (R2, 1) (R3 5) (R21 3) (OUTPUTS (R1 5) (R23, 1) (R23’ 1)),
preconditions are of the form ?AIk <= ?C(?Rk)

(TRANSFORM **E** (INPUTS (R1 10*5) (R2, 10*1) (R3 10*5) (R21 10*3) (OUTPUTS (R1 10*5) (R23, 10*1) (R23’ 10*1)),
preconditions are of the form 50 <= 100, 10 <= 50, 50 <= 2000, 30 <= 30

• • •

(TRANSFER ?Cj1 ?Cj2 ((?Ri ?ARi)), where ?ARi <= ?Cj1(?Ri)

(TRANSFER E A ((R3 500)), preconditions 500 <= 2000

• • •

A(tlantis)	E(rewon)
R1: 500	R1: 100
R2: 700	R2: 50
R3: 100	R3: 2000
R21: 0	R21: 30
R21’: 0	R21’: 0
R22: 0	R22: 0
R22’: 0	R22’: 0
R23: 0	R23: 0
R23’: 0	R23’: 0

Alloys Template

((TRANSFORM ?C (INPUTS (R1 1) (R2, 2)) (OUTPUTS (R1 1) (R21, 1) (R21’ 1)),
preconditions are of the form ?ARj <= ?C(?Rj)

(TRANSFORM **A** (INPUTS (R1 50*1) (R2, 50*2)) (OUTPUTS (R1 50) (R21, 50) (R21’ 50)),
preconditions 50 <= 500, 100 <= 700

A(tlantis)	E(rewon)
R1: 500	R1: 100
R2: 600	R2: 50
R3: 100	R3: 2000
R21: 50	R21: 30
R21’: 50	R21’: 0
R22: 0	R22: 0
R22’: 0	R22’: 0
R23: 0	R23: 0
R23’: 0	R23’: 0

. . .

Electronics Template

(TRANSFORM ?C (INPUTS (R1 3) (R2 2) (R21 2)) (OUTPUTS (R22 2) (R22’ 2) (R1 3)),
preconditions are of the form ?ARj <= ?C(?Rj)

(TRANSFORM **A** (INPUTS (R1 30) (R2 20) (R21 20)) (OUTPUTS (R22 20) (R22’ 20) (R1 30)),
preconditions 30 <= 500, 20 <= 700, 20 !<= 0

X No successor

. . .

Housing Template

(TRANSFORM ?C (INPUTS (R1 5) (R2, 1) (R3 5) (R21 3) (OUTPUTS (R1 5) (R23, 1) (R23’ 1)),
preconditions are of the form ?AIk <= ?C(?Rk)

(TRANSFORM **E** (INPUTS (R1 10*5) (R2, 10*1) (R3 10*5) (R21 10*3) (OUTPUTS (R1 10*5)
(R23, 10*1) (R23’ 10*1)),
preconditions are of the form 50 <= 100, 10 <= 50, 50 <= 2000, 30 <= 30

A(tlantis)	E(rewon)
R1: 500	R1: 10
R2: 700	R2: 40
R3: 100	R3: 1950
R21: 0	R21: 0
R21’: 0	R21’: 0
R22: 0	R22: 0
R22’: 0	R22’: 0
R23: 0	R23: 10
R23’: 0	R23’: 10

. . .

(TRANSFER ?Cj1 ?Cj2 ((?Ri ?ARi)), where ?ARi <= ?Cj1(?Ri)

(TRANSFER E A ((R3 500)), preconditions 500 <= 2000

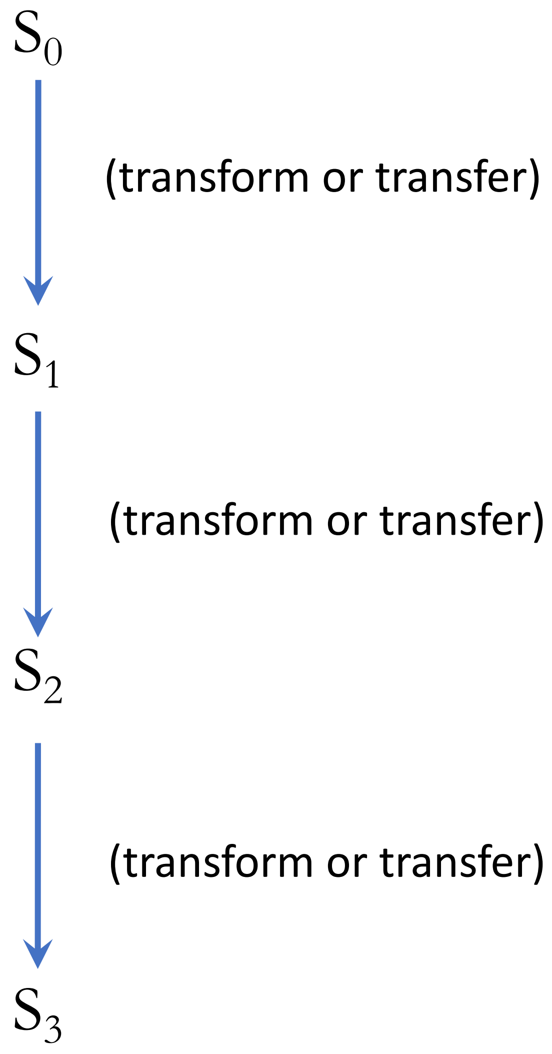
A(tlantis)	E(rewon)
R1: 500	R1: 100
R2: 700	R2: 50
R3: 600	R3: 1500
R21: 0	R21: 30
R21’: 0	R21’: 0
R22: 0	R22: 0
R22’: 0	R22’: 0
R23: 0	R23: 0
R23’: 0	R23’: 0

. . .

This shows only a few of the many successors in our domain

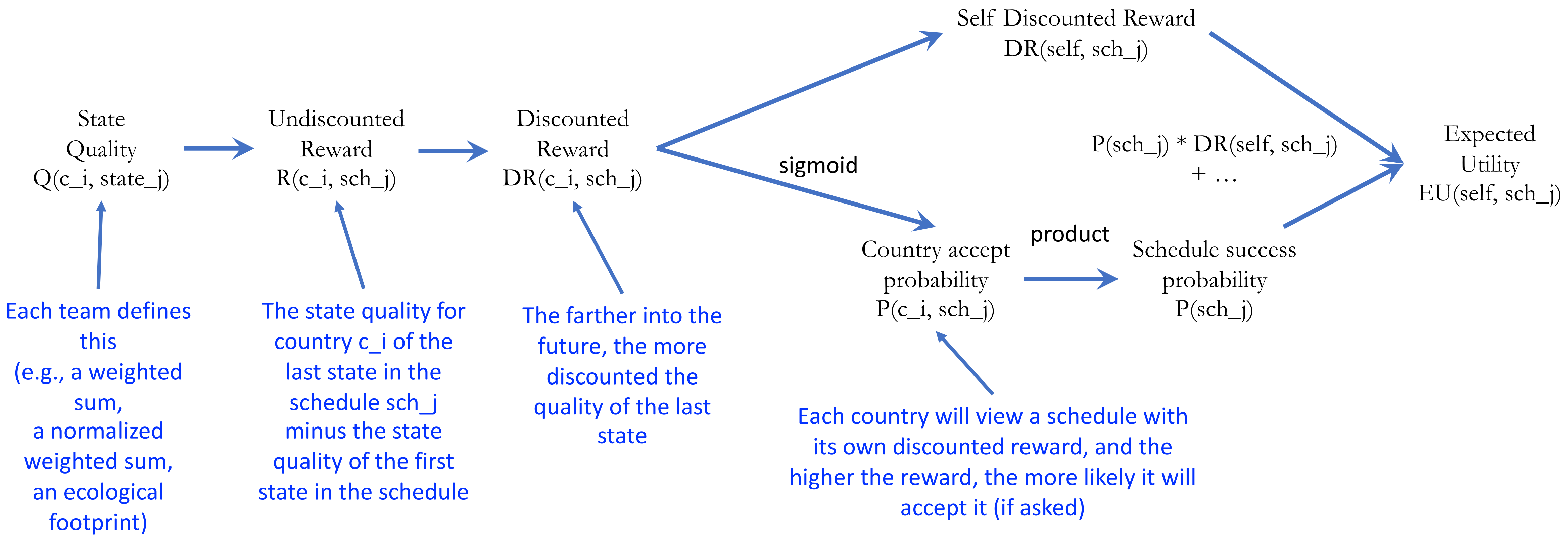
A partial schedule with all variables bound to constants (e.g., ?C1 bound to Atlantis, ?C2 bound to Carpania)

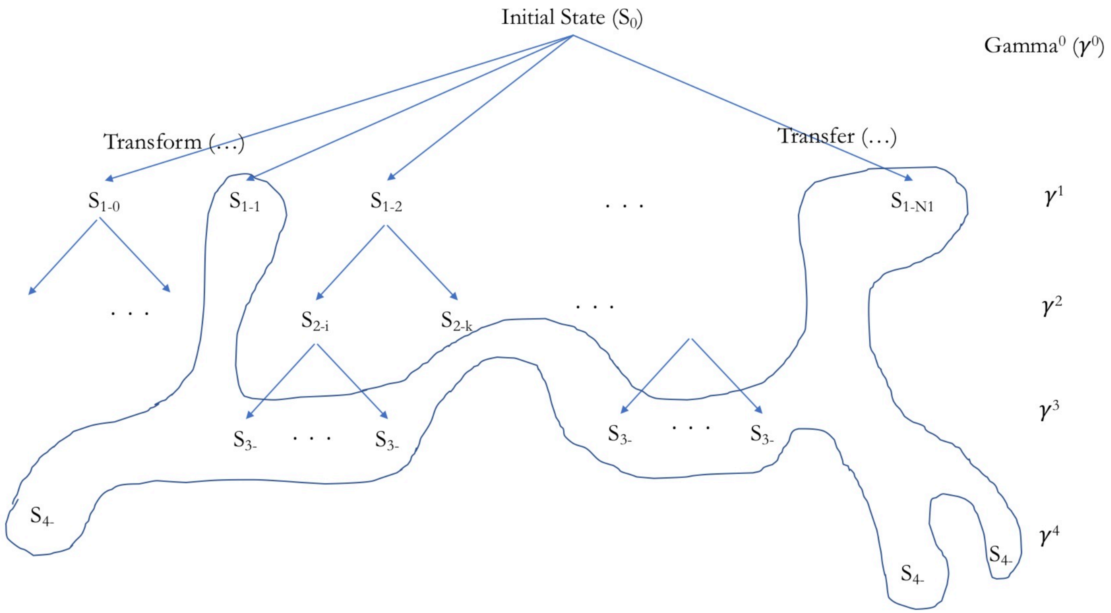
```
[  (TRANSFORM Atlantis (INPUTS (AvailableLand 5)
                             (Population 25)
                             (Water 25)
                             (MetallicElements 5)
                             (Timber 25)
                             (MetallicAlloys 15)
                             (PotentialEnergyUsable 25))
                    (OUTPUTS (Housing 5)
                             (HousingWaste 5)
                             (Population 25)
                             (Water 20)))
   (TRANSFER Atlantis Carpania ((Housing 3))
   ...
]
```



Dependencies between measures

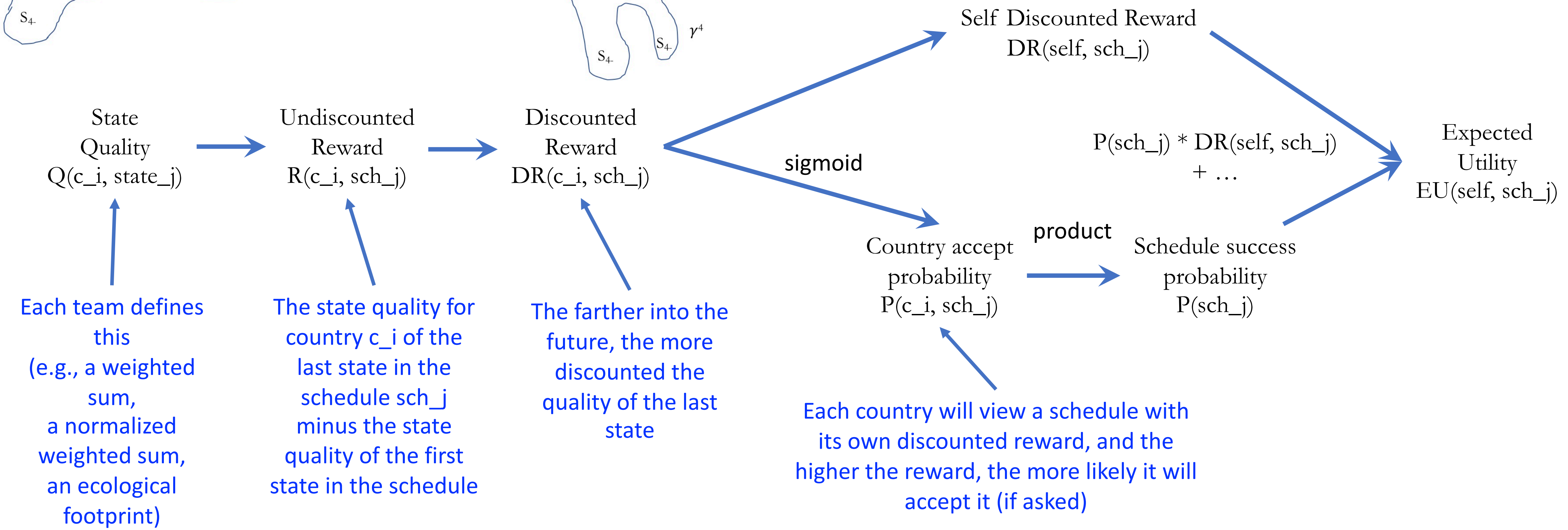
https://my.vanderbilt.edu/cs4269aiproject/project-part-1-spring2021/

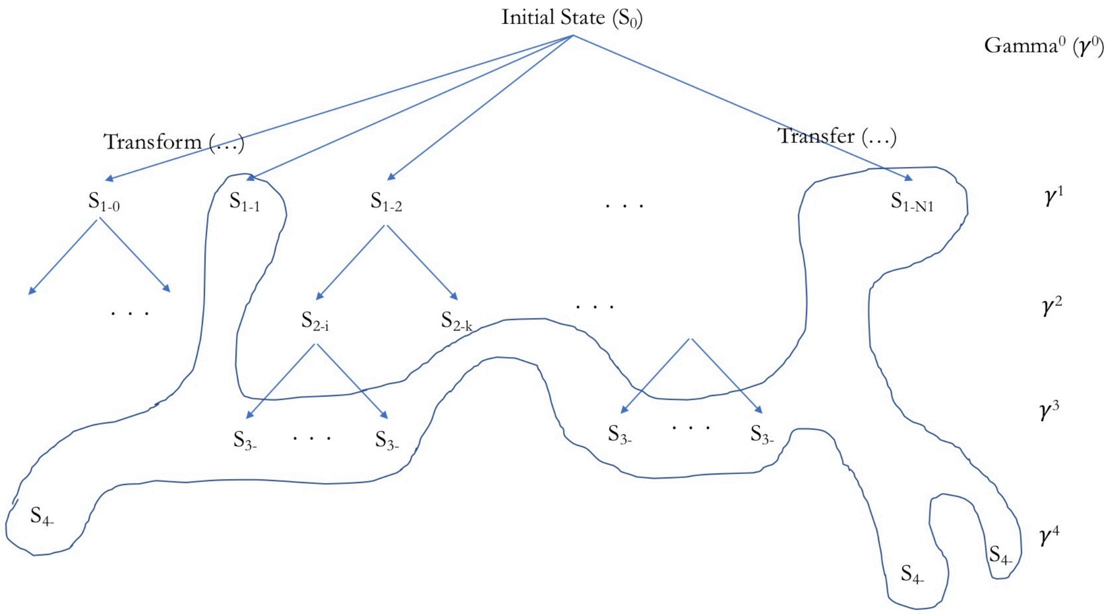


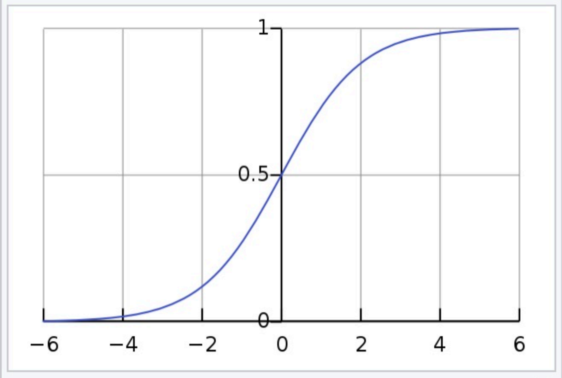


	A	B	C	D	E
1	Country	R1	R2	R3	R21
2	Atlantis	100	700	2000	0
3	Brobdingnag	50	300	1200	0
4	Carpania	25	100	300	0
5	Dinotopia	30	200	200	0
6	Erewhon	70	500	1700	0

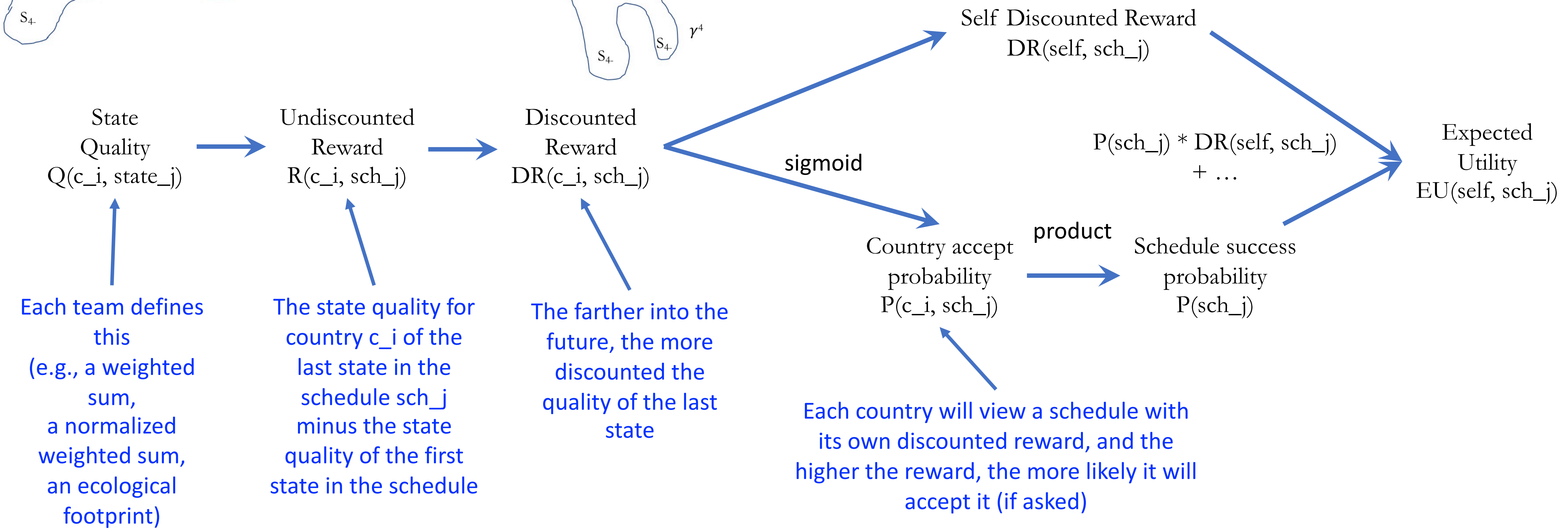
• • •







Standard logistic sigmoid function i.e. $L = 1, k = 1, x_0 = 0$



S_0 $EU(\text{self}, S_0)$

(transform or transfer)

S_1 $EU(\text{self}, S_0 \rightarrow S_1)$

(transform or transfer)

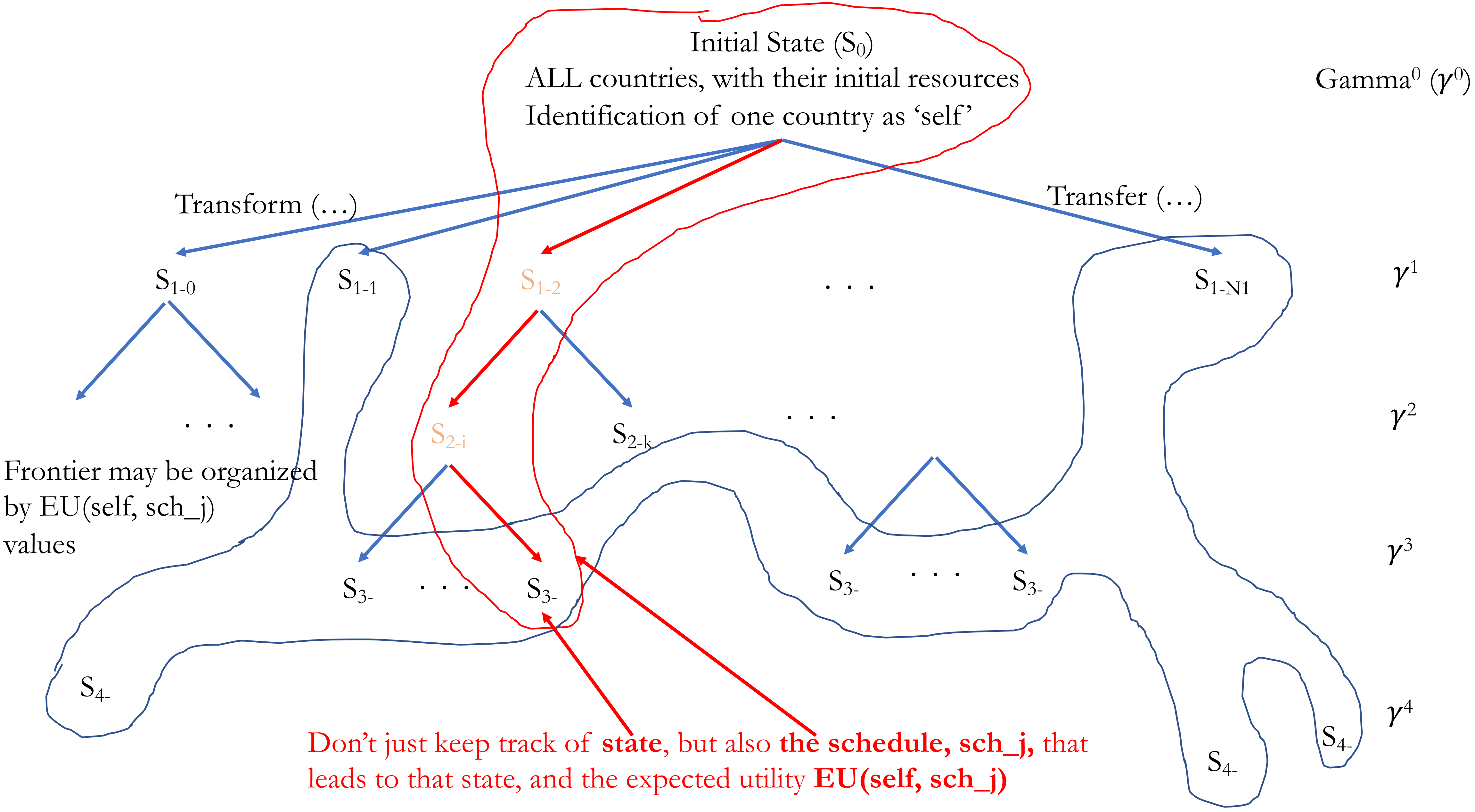
S_2 $EU(\text{self}, S_0 \rightarrow S_1 \rightarrow S_2)$

(transform or transfer)

S_3 $EU(\text{self}, S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3)$

You will want to track intermediate schedules and their EU scores so that you can see if these scores monotonically improve or not

If search depth-bound is 3, for example, must always search to depth 3. You will output the depth 3 schedule, even if we would want to execute the best sub-schedule when the time came to act, whether it be depth 0, 1, 2, or 3



Initial State (S_0)
ALL countries, with their initial resources
Identification of one country as 'self'
Gamma^0 (γ^0)
Transform (…)
Transfer (…)
$S_{1\text{-}0}$
$S_{1\text{-}1}$
$S_{1\text{-}2}$
. . .
$S_{1\text{-}N1}$
γ^1
. . .
$S_{2\text{-}i}$
$S_{2\text{-}k}$
. . .
γ^2
Frontier may be organized by EU(self, sch_j) values
$S_{3\text{-}}$
. . .
$S_{3\text{-}}$
$S_{3\text{-}}$
. . .
$S_{3\text{-}}$
γ^3
$S_{4\text{-}}$
$S_{4\text{-}}$
$S_{4\text{-}}$
γ^4
Don't just keep track of **state**, but also **the schedule, sch_j,** that leads to that state, and the expected utility **EU(self, sch_j)**

Near-term Tasks

1. Represent Resources, Countries, Initial States (test problems)
    a) Start with resources that correspond to required resources
    b) Enable additions of resources and countries
    c) Research and cite real-world resources as loose justification
2. Code to read files
3. Generate Successors
    a) Accumulate schedules that correspond to paths
    b) Compute measures, notably
        i) state utility, **your choice (each country)**
        ii) undiscounted reward, **pre-specified form (each country)**
        iii) discounted reward, **pre-specified form (each country)**
        iv) Expected Utility (EU) **pre-specified form (your country, self)**
        v) Be able to easily modify constants γ (discounted reward), x_0 and k (logistic function), C (Expected Utility)