Phd course on

Formal modelling and analysis of interactive systems

Part 4
Tasks and Task Failures

Tasks, Tasks Failures, Decomposition, ATC System Example

Antonio Cerone

United Nations University

International Institute for Software Technology

Macau SAR China

email: antonio@iist.unu.edu

web: www.iist.unu.edu

Contents

Task Modelling and Task Analysis

Goal and Task

- Goal: what the user expects to achieve as the output as the interaction

Goal and Task

- Goal: what the user expects to achieve as the output as the interaction
- Task: the activity that has to be performed to achieve a goal

Goal and Task

- Goal: what the user expects to achieve as the output as the interaction
- Task: the activity that has to be performed to achieve a goal
  - may be decomposed into sub-tasks

Goal and Task

- Goal: what the user expects to achieve as the output as the interaction
- Task: the activity that has to be performed to achieve a goal
  - may be decomposed into sub-tasks
  - top-level task includes the goal as a sub-task

Goal and Task

- Goal: what the user expects to achieve as the output as the interaction
- Task: the activity that has to be performed to achieve a goal
  - may be decomposed into sub-tasks
  - top-level task includes the goal as a sub-task
  - lower-level tasks may be associated with sub-goal a sub-task

Goal and Task

- Goal: what the user expects to achieve as the output as the interaction
- Task: the activity that has to be performed to achieve a goal
  - may be decomposed into sub-tasks
  - top-level task includes the goal as a sub-task
  - lower-level tasks may be associated with sub-goal a sub-task
  - $\Longrightarrow$ closure may be nested / sequentialised

Goal and Task

- Goal: what the user expects to achieve as the output as the interaction
- Task: the activity that has to be performed to achieve a goal
  - may be decomposed into sub-tasks
  - top-level task includes the goal as a sub-task
  - lower-level tasks may be associated with sub-goal a sub-task
  - $\Longrightarrow$ closure may be nested / sequentialised
  - lower-level tasks are actions

Categories of Task

- user task
  cognitive activities performed by the user

Categories of Task

- user task
  cognitive activities performed by the user
- application task
  outputs provided to the user by the machine

Categories of Task

- user task
  cognitive activities performed by the user
- application task
  outputs provided to the user by the machine
- interaction task
  performed by the user by interacting with the system

Categories of Task

- user task
  cognitive activities performed by the user
- application task
  outputs provided to the user by the machine
- interaction task
  performed by the user by interacting with the system
- abstract task
  high-level activities which involve more than one category above

Categories of Task — ATM

- user task
  cognitive activities performed by the user
  recall the pin, decide how much to withdraw
- application task
  outputs provided to the user by the machine
  deliver cash, return the card
- interaction task
  performed by the user through interaction
  insert the card, enter the pin
- abstract task
  high-level activities
  get cash, get authenticated

Task Modelling

- indentify the activities
  that have to be performed to achieve a goal

Task Modelling

- indentify the activities
  that have to be performed to achieve a goal
- decompose activities
  to produce a task hierarchy

Task Modelling

- indentify the activities
  that have to be performed to achieve a goal
- decompose activities
  to produce a task hierarchy
- stop the decomposition
  when basic actions have been reached

Task Analysis

The process of analysing the way people perform tasks:

- what people do
- what things they work with
- what they must know (task knowledge)

Purpose

of Task Modelling

- define a conceptual model of a new system from a user's perspective
  - $\Longrightarrow$ build a formal model of the system
  - $\Longrightarrow$ convert it to a formal task specification

Purpose

of Task Modelling

- define a conceptual model of a new system from a user's perspective
  - $\Longrightarrow$ build a formal model of the system
  - $\Longrightarrow$ convert it to a formal task specification

of Task Analysis

- production of training material and documentation
- requirement capture
- design / generation of user interfaces

Task Decomposition

- describe the actions people do
- structure them within task-subtask hierarchy
- describe order of subtasks

Task Decomposition

- describe the actions people do
- structure them within task-subtask hierarchy
- describe order of subtasks

Used in both Task Modelling and Task Analysis

Tools for Task Decomposition

Hierarchical Task Analysis (HTA)

- text and diagrams to show hierarchy
- plans to describe order

Tools for Task Decomposition

Hierarchical Task Analysis (HTA)

- text and diagrams to show hierarchy
- plans to describe order

CuncurTaskTrees [Paternò 00]

- text, icons and diagrams to show hierarchy
- temporal relationships (using process algebra style operators) to describe order

HTA: Textual Notation

Hierarchy description:

0. make a cup of tea
    1. boil water
    2. empty pot
    3. put tea leaves in pot
    4. pour in boiling water
    5. wait 5 minutes
    6. pour tea

HTA: Textual Notation

Hierarchy description:

0. make a cup of tea
    1. boil water
    2. empty pot
    3. put tea leaves in pot
    4. pour in boiling water
    5. wait 5 minutes
    6. pour tea

Plans

Plan 0. do 1
at the same time, if pot is full do 2
then do 3 – 4 – 5
after 5 minutes do 6

Generating the Hierarchy

- get list of tasks
- group tasks into higher level tasks
- decompose lower level tasks further

Generating the Hierarchy

- get list of tasks
- group tasks into higher level tasks
- decompose lower level tasks further

How to know when to stop?

Generating the Hierarchy

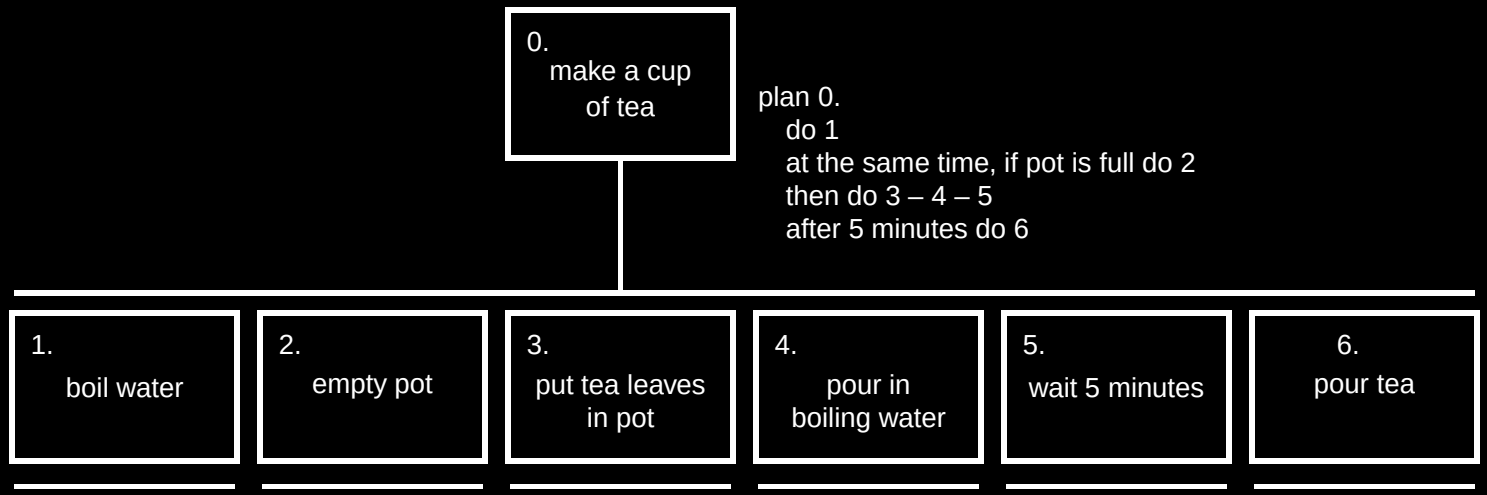
- get list of tasks
- group tasks into higher level tasks
- decompose lower level tasks further

How to know when to stop?

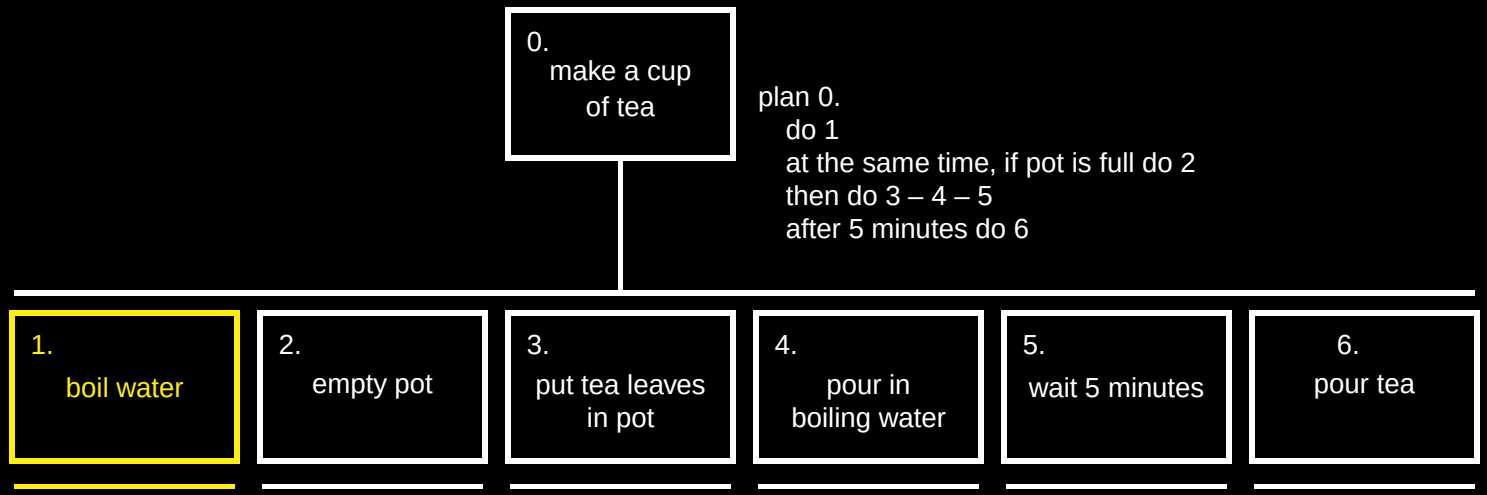
Stopping rules:

- Simplicity: Is the task simple enough?
- Purpose: Is the task relevant?
- Motor Action: lowest sensible level

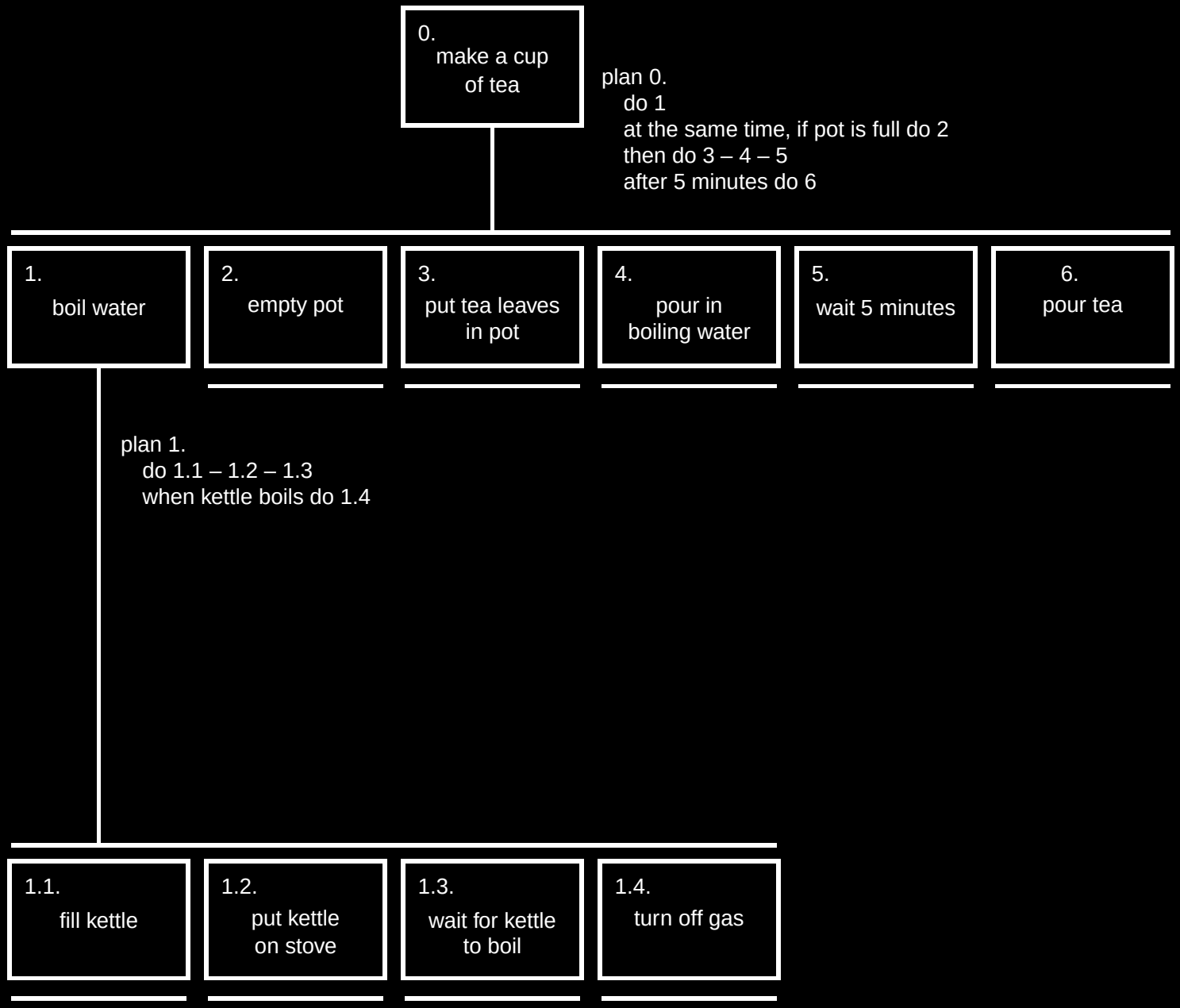
HTA: Diagrammatic Notation



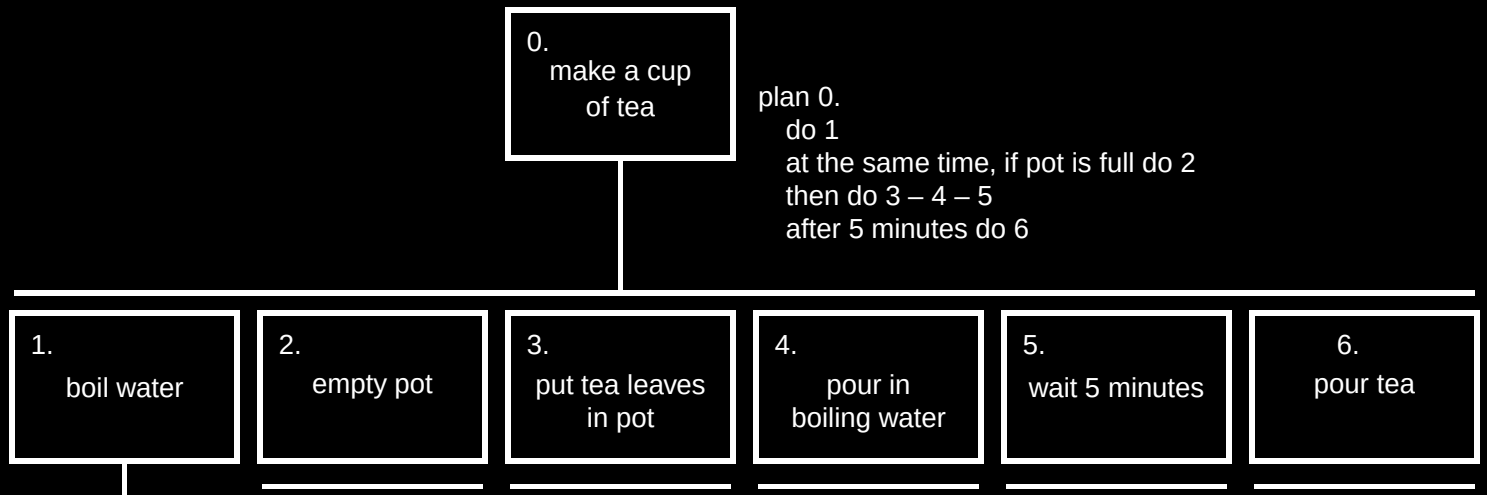
HTA: Decomposition



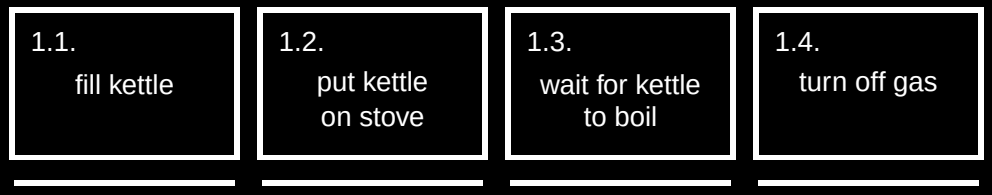
HTA: Decomposition



HTA: Domain Expert



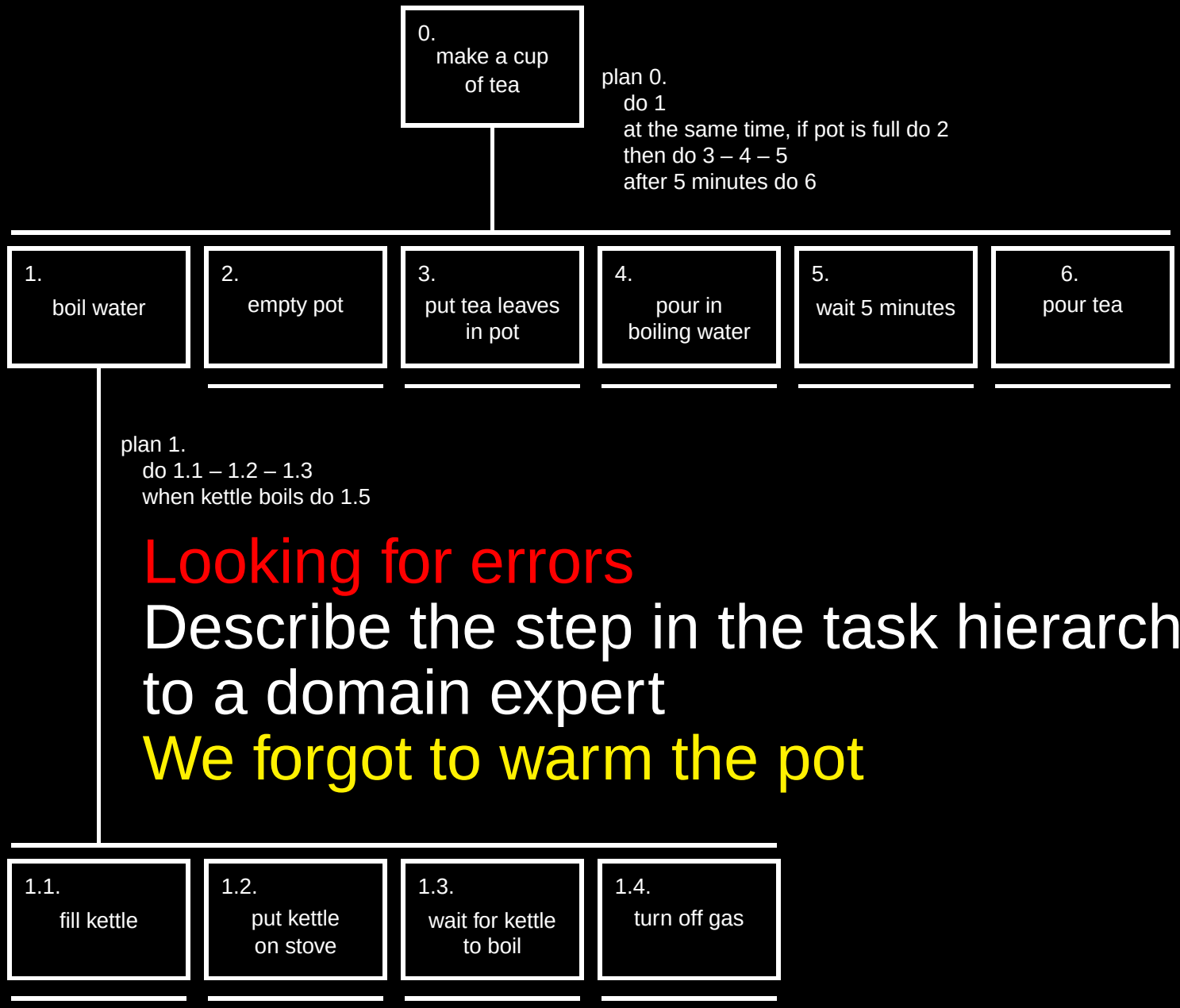
plan 1.
do 1.1 – 1.2 – 1.3
when kettle boils do 1.5



Looking for errors

Describe the step in the task hierarchy to a domain expert

HTA: Domain Expert

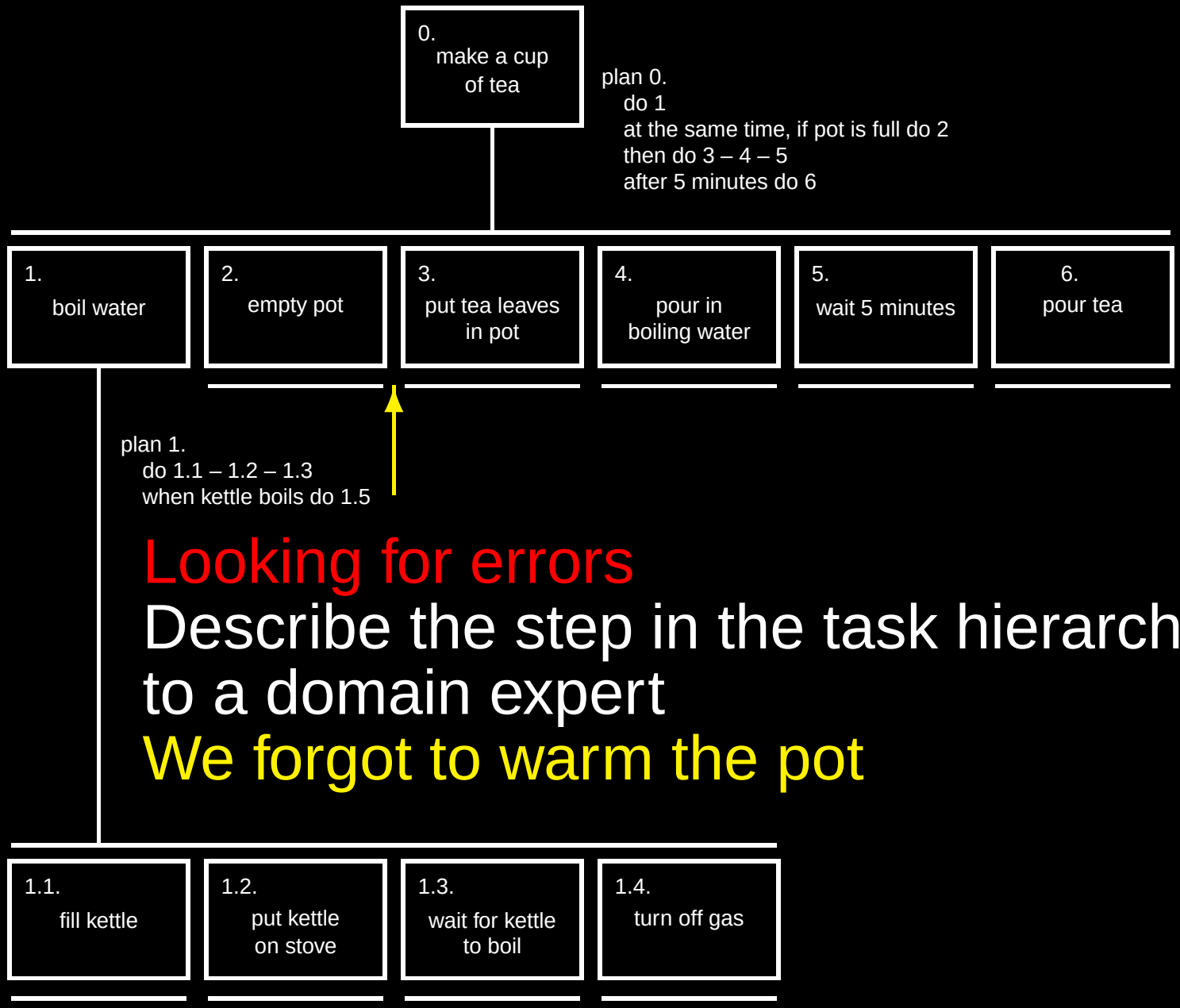


Looking for errors

Describe the step in the task hierarchy to a domain expert

We forgot to warm the pot

HTA: Domain Expert

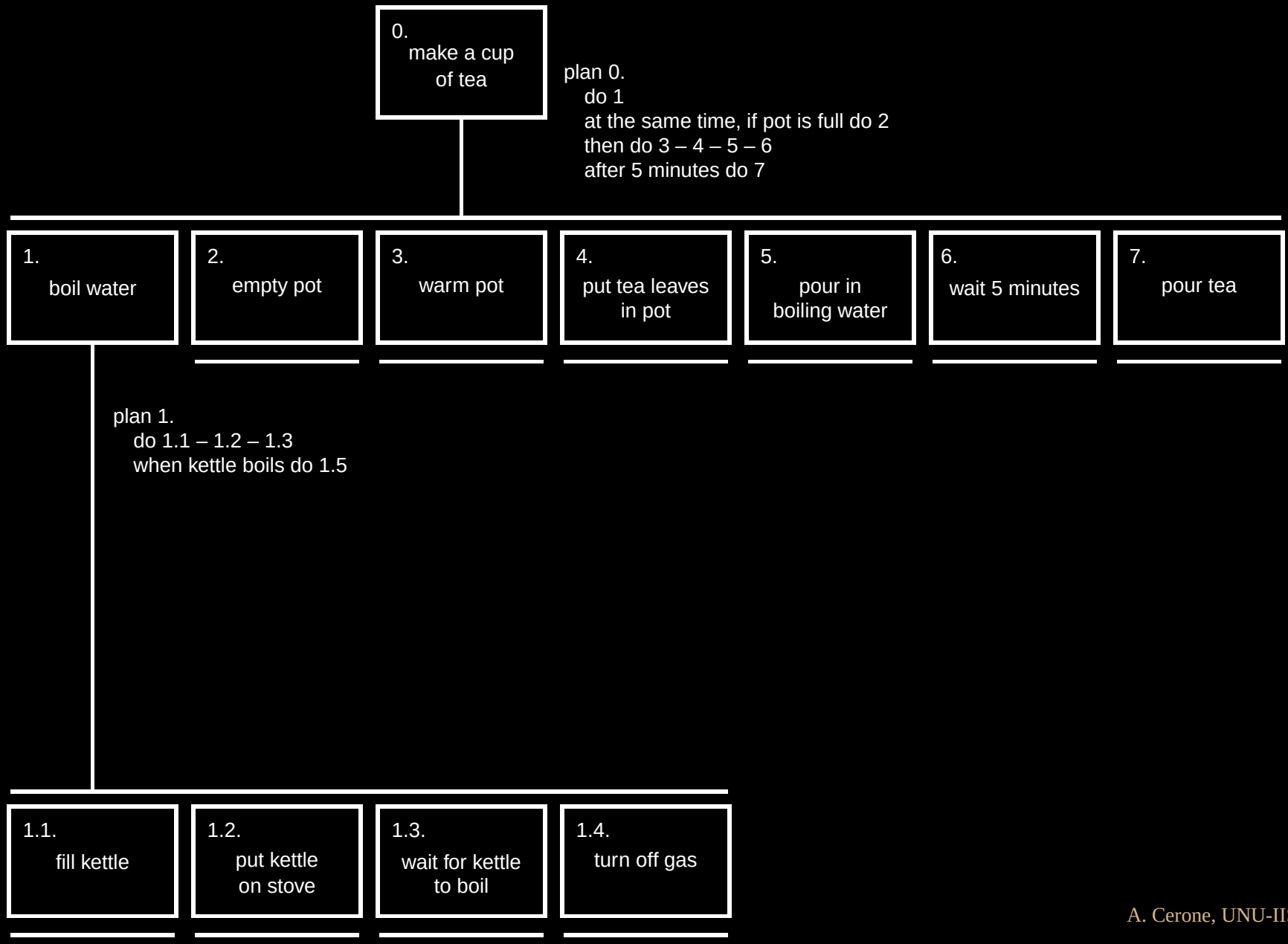


Looking for errors

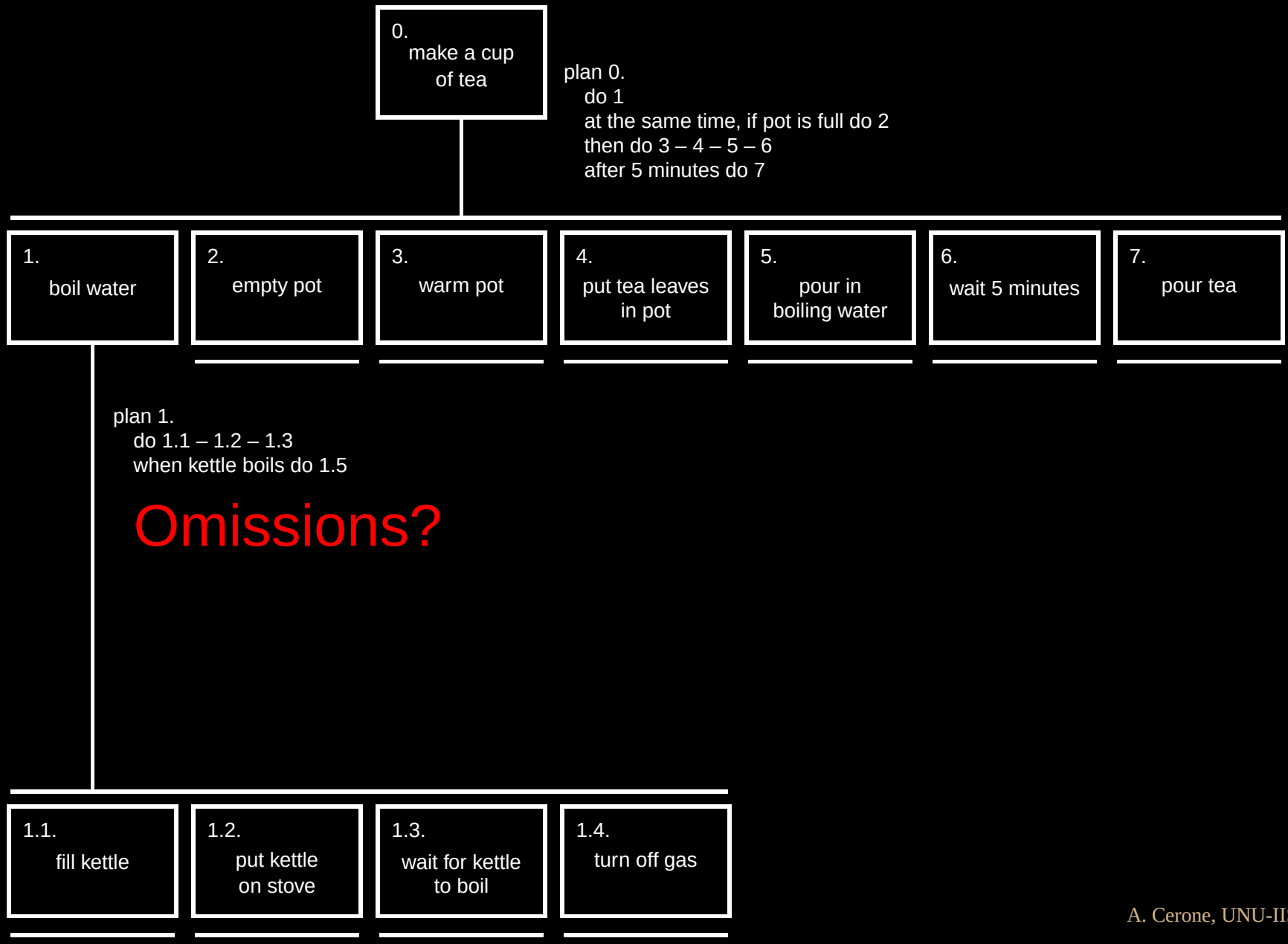
Describe the step in the task hierarchy to a domain expert

We forgot to warm the pot

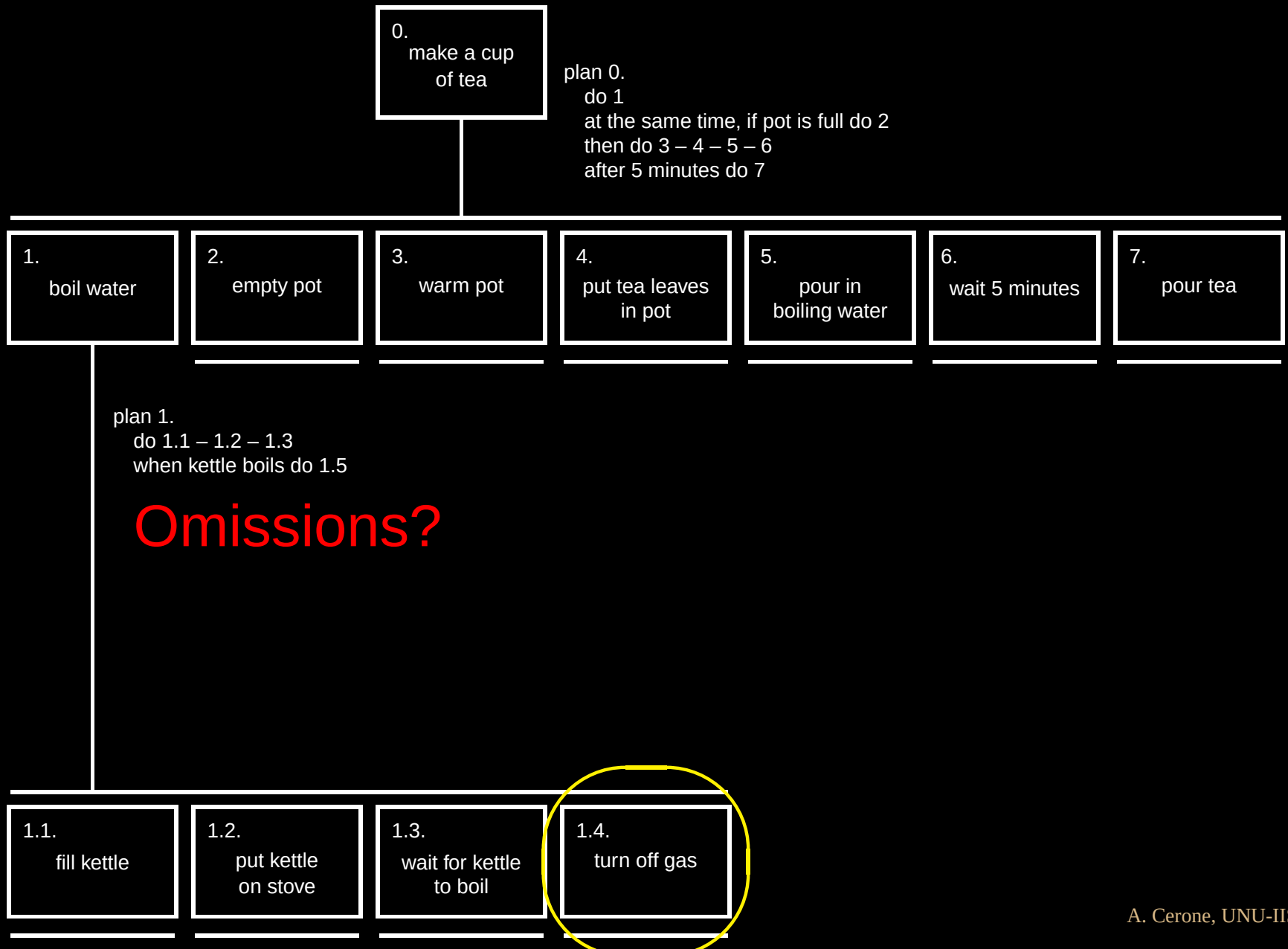
HTA: Omissions



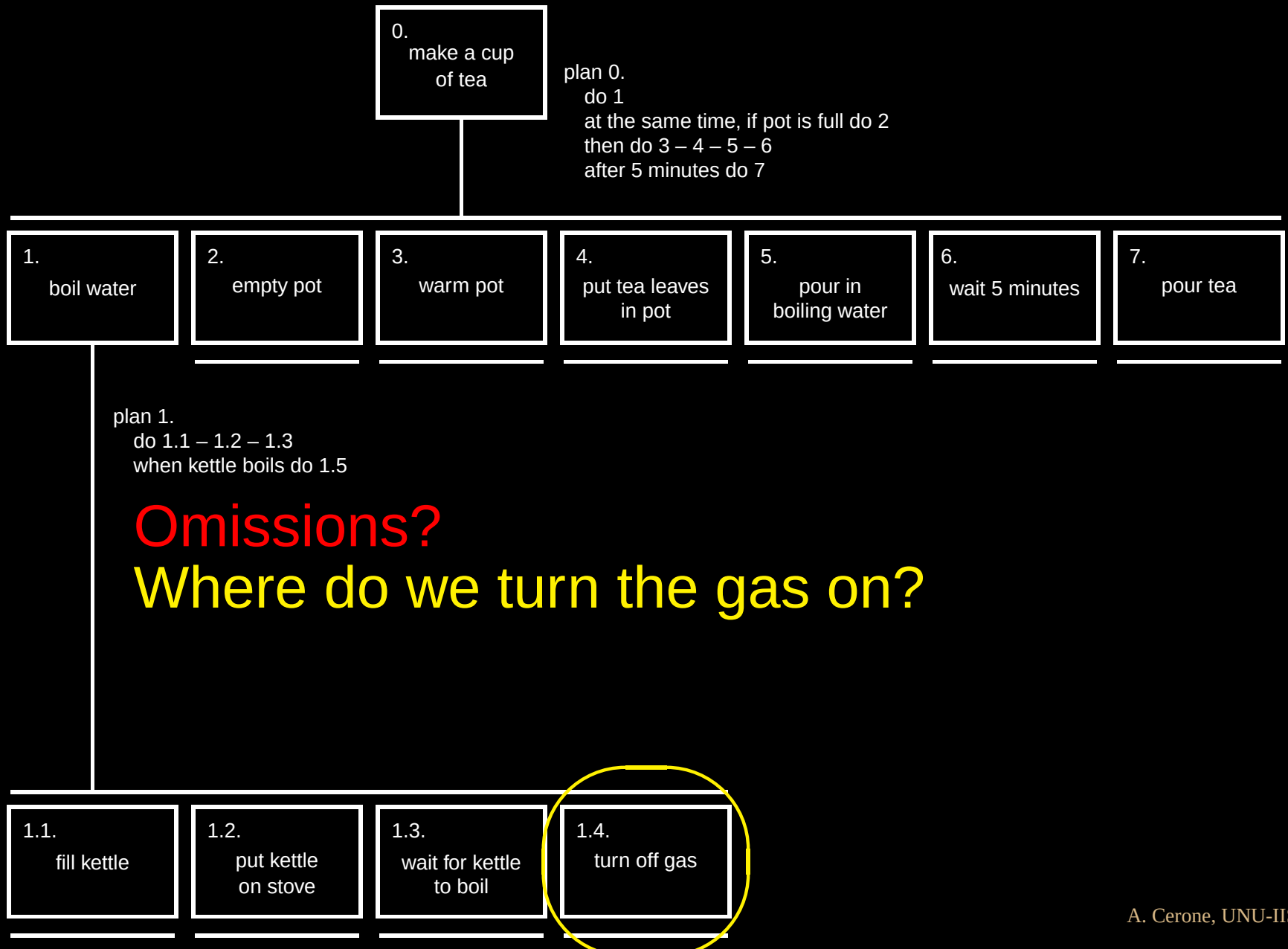
HTA: Omissions



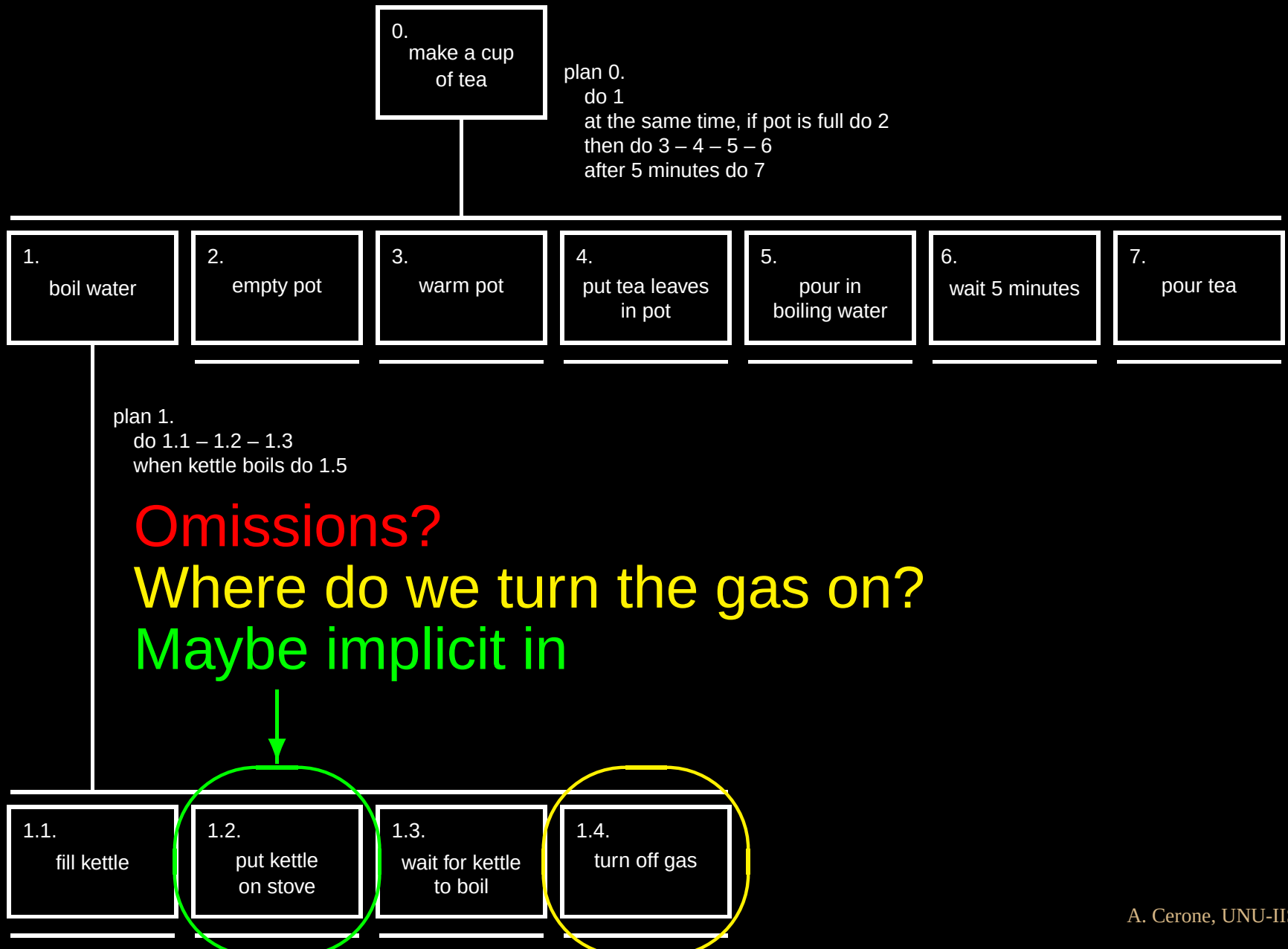
HTA: Omissions



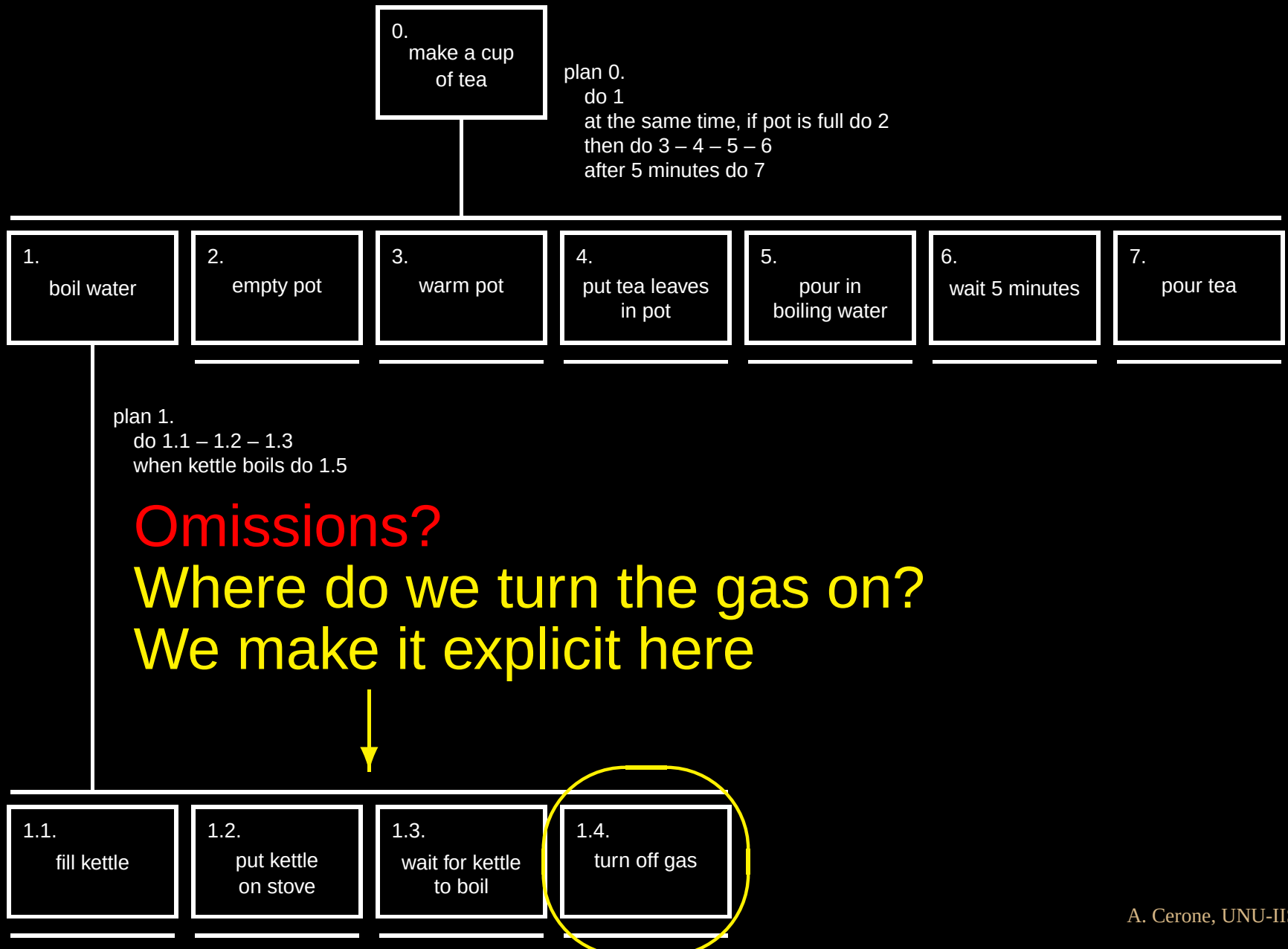
HTA: Omissions



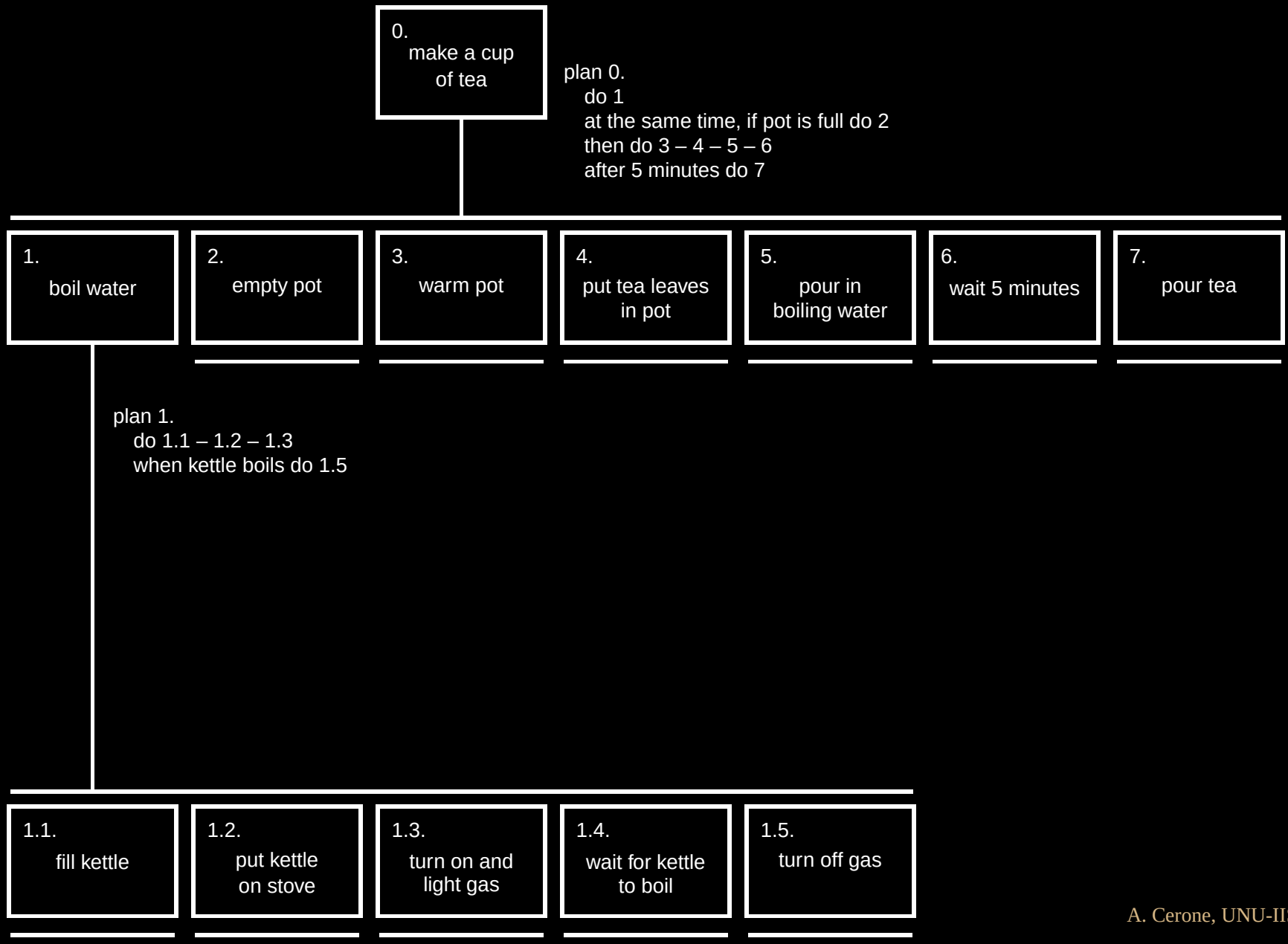
HTA: Omissions



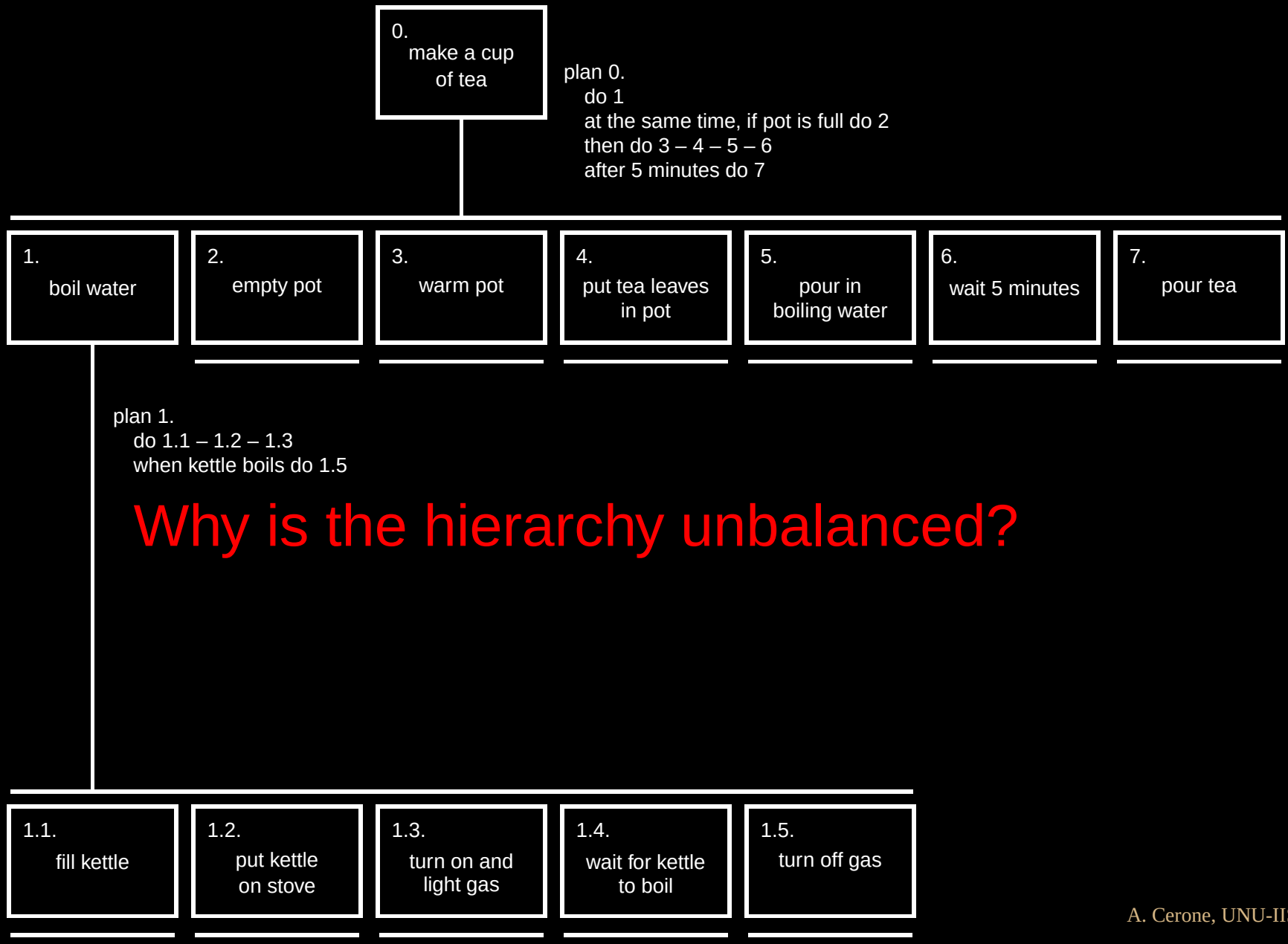
HTA: Omissions



HTA: Umbalanced Hierarchy

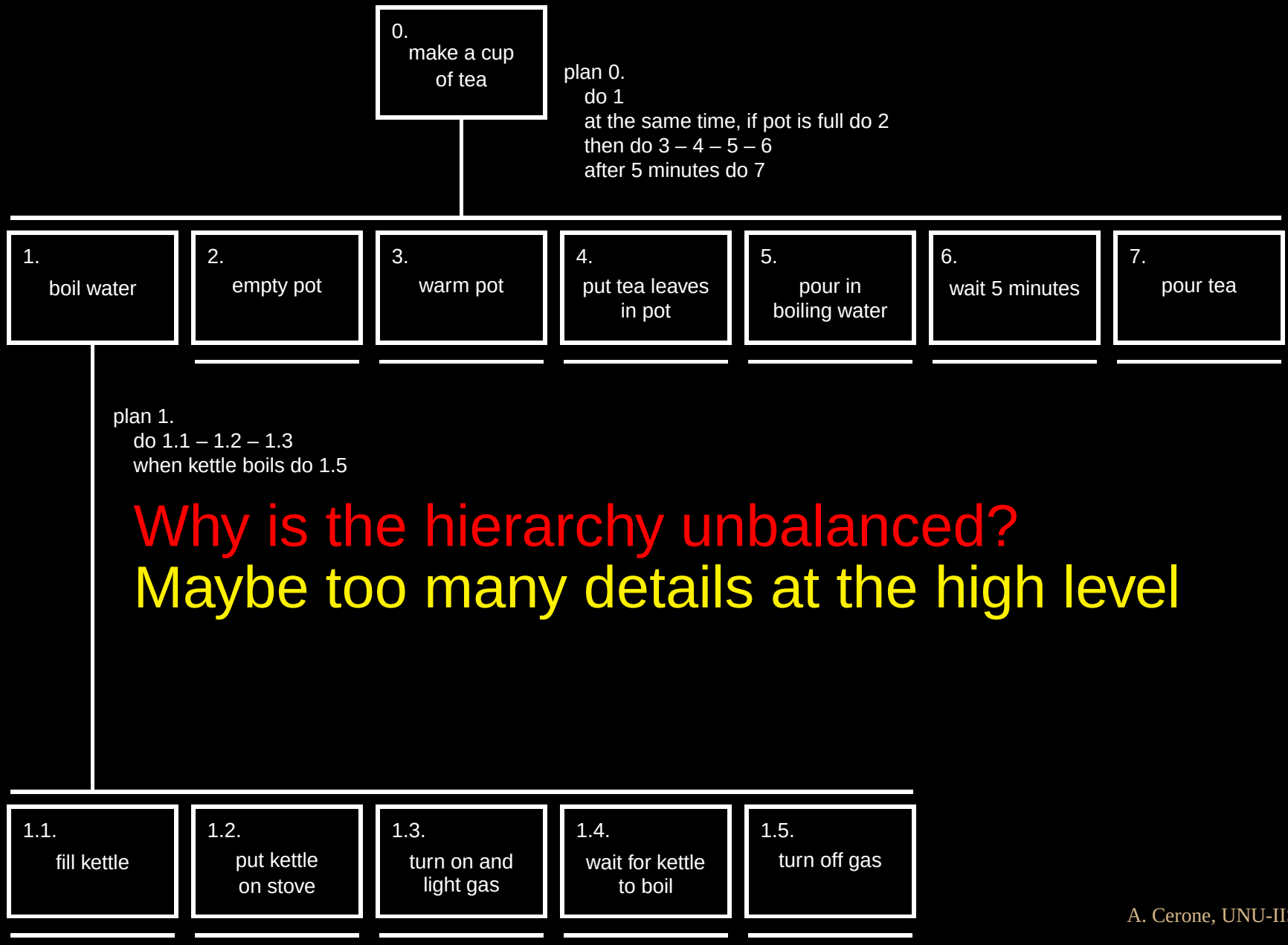


HTA: Umbalanced Hierarchy



Why is the hierarchy unbalanced?

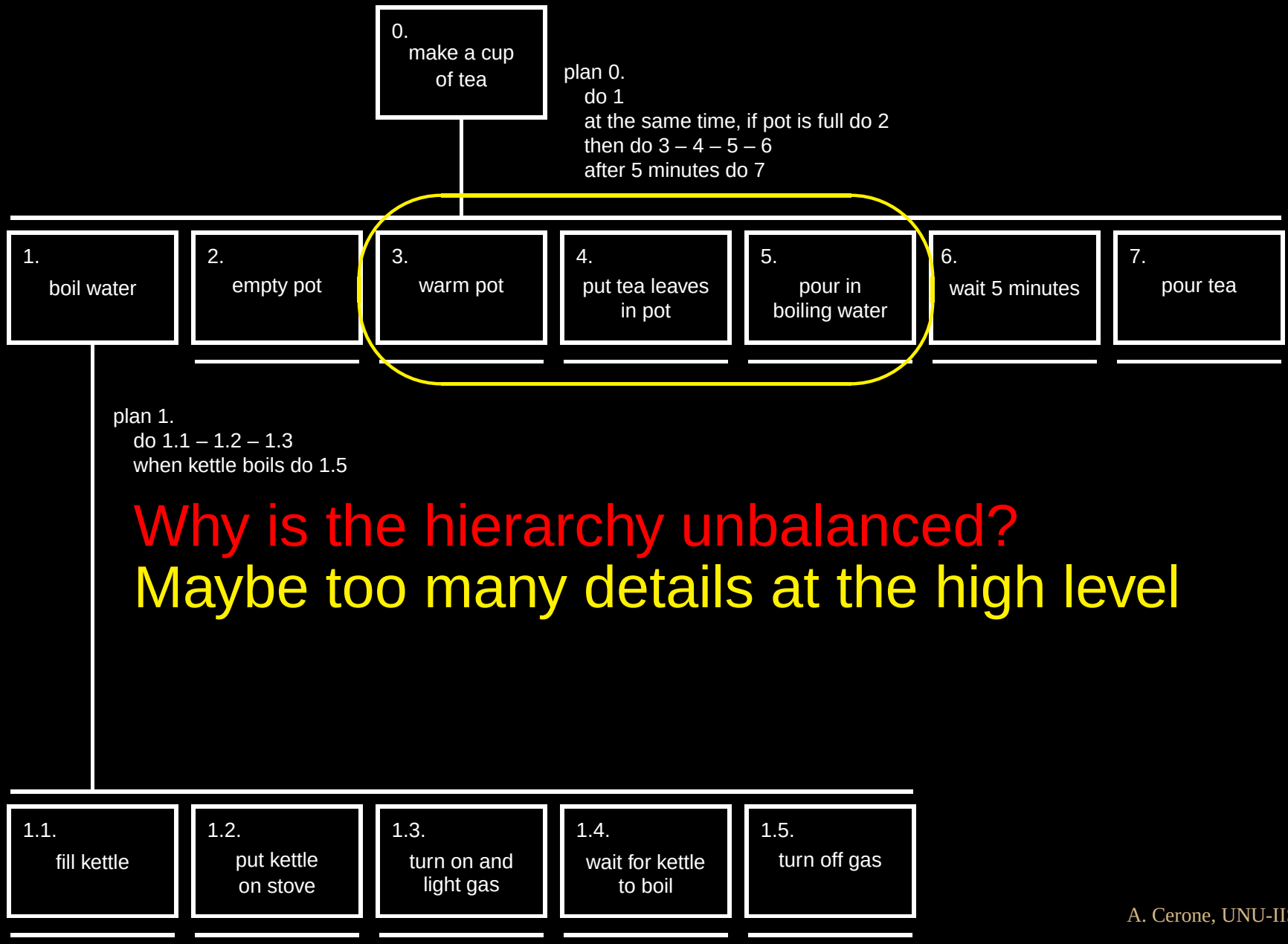
HTA: Umbalanced Hierarchy



Why is the hierarchy unbalanced?

Maybe too many details at the high level

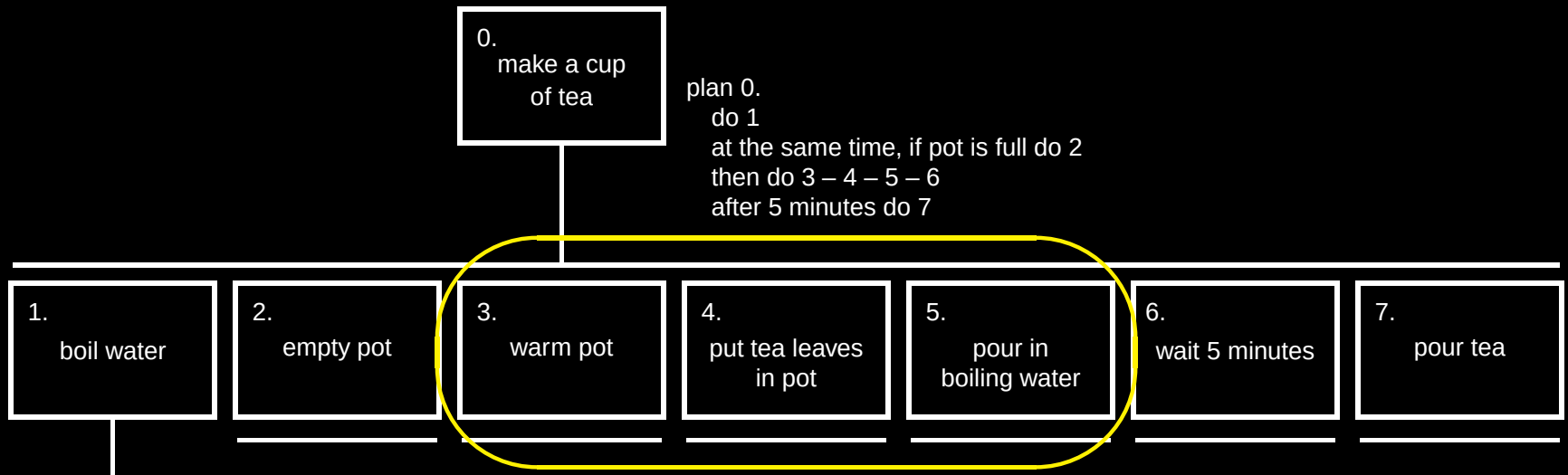
HTA: Umbalanced Hierarchy



Why is the hierarchy unbalanced?

Maybe too many details at the high level

HTA: Umbalanced Hierarchy

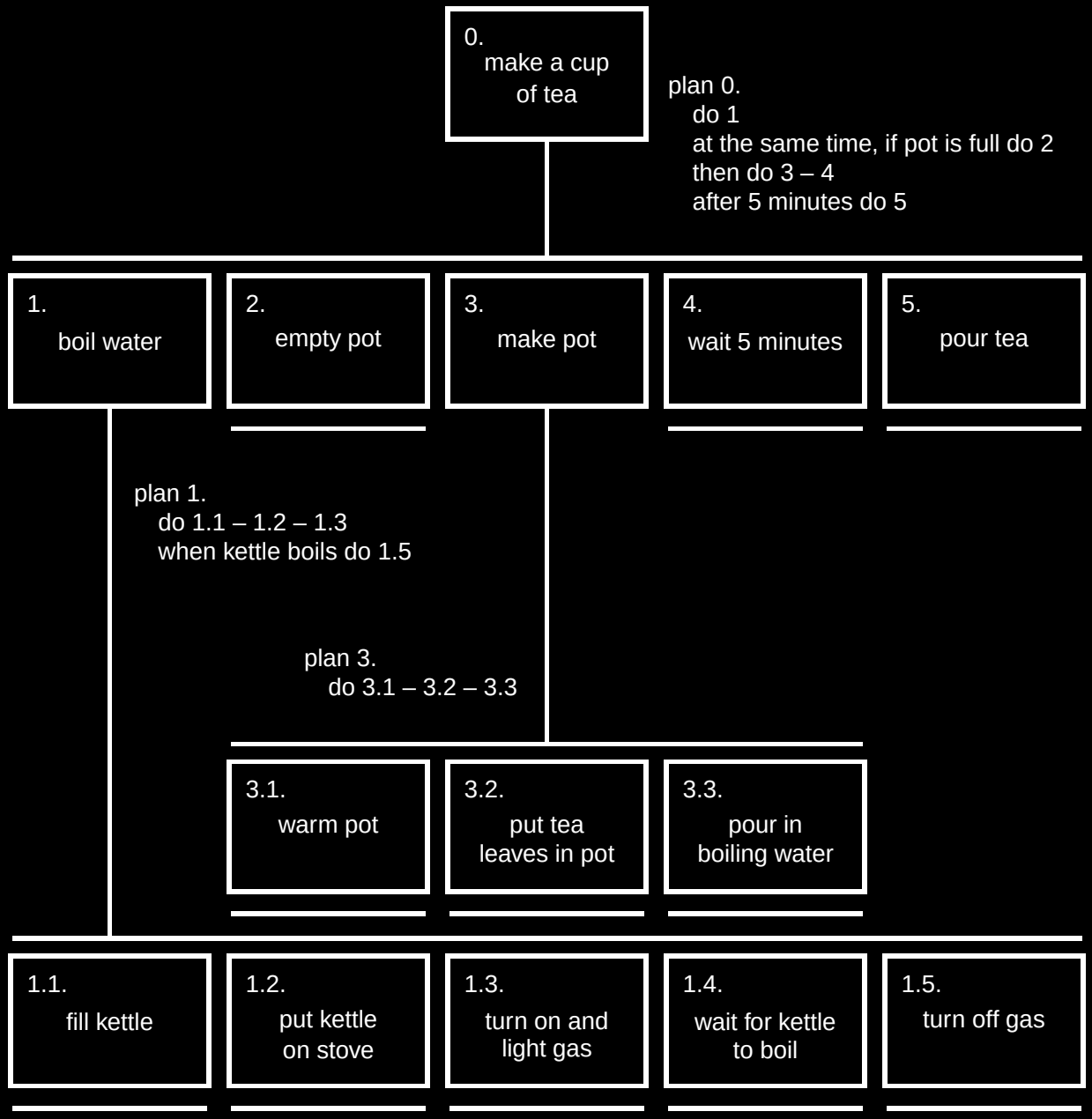


Why is the hierarchy unbalanced?

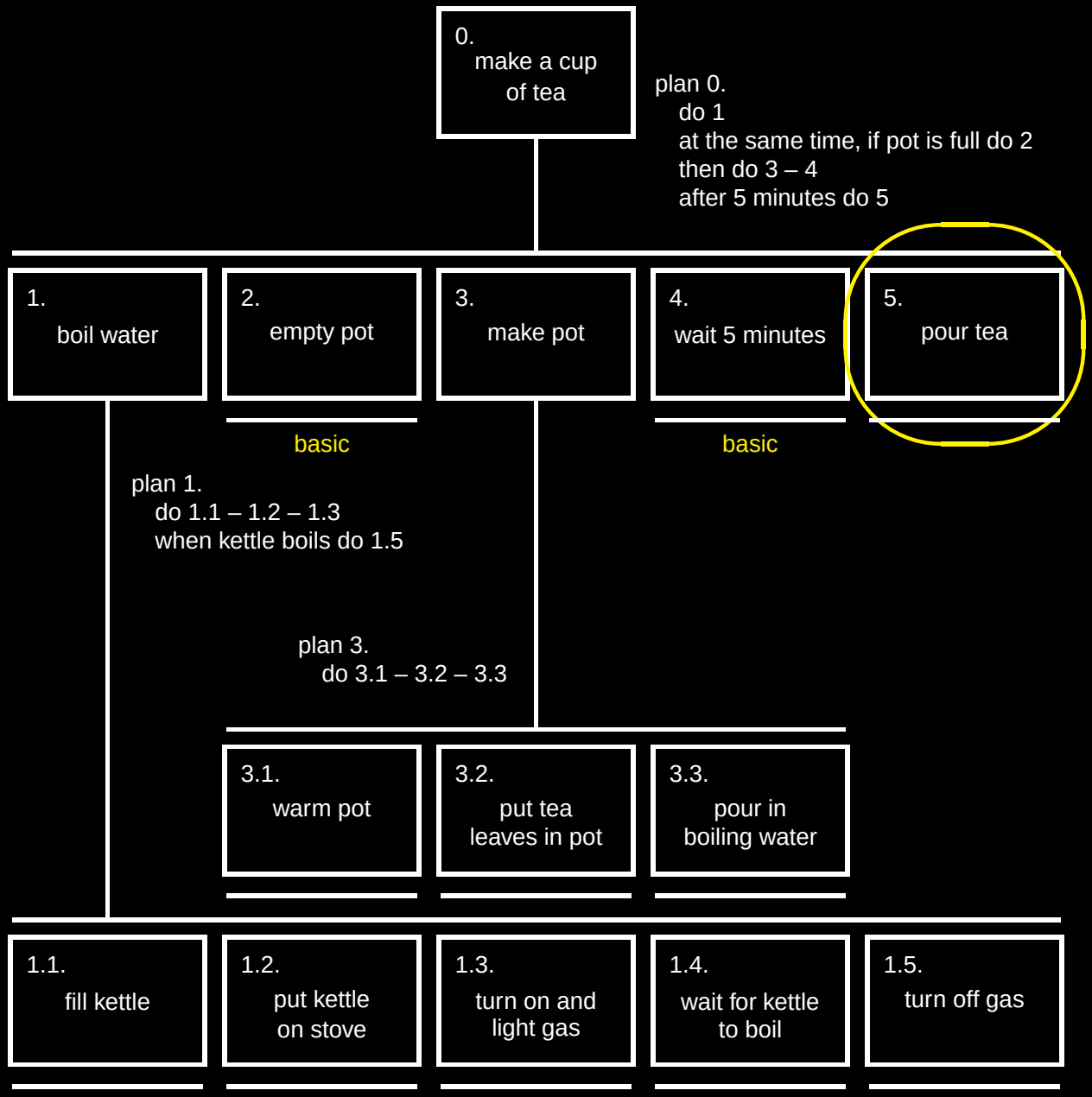
Maybe too many details at the high level

We add new make pot node which encompass tasks 3, 4, 5. Why not 2 and 6?

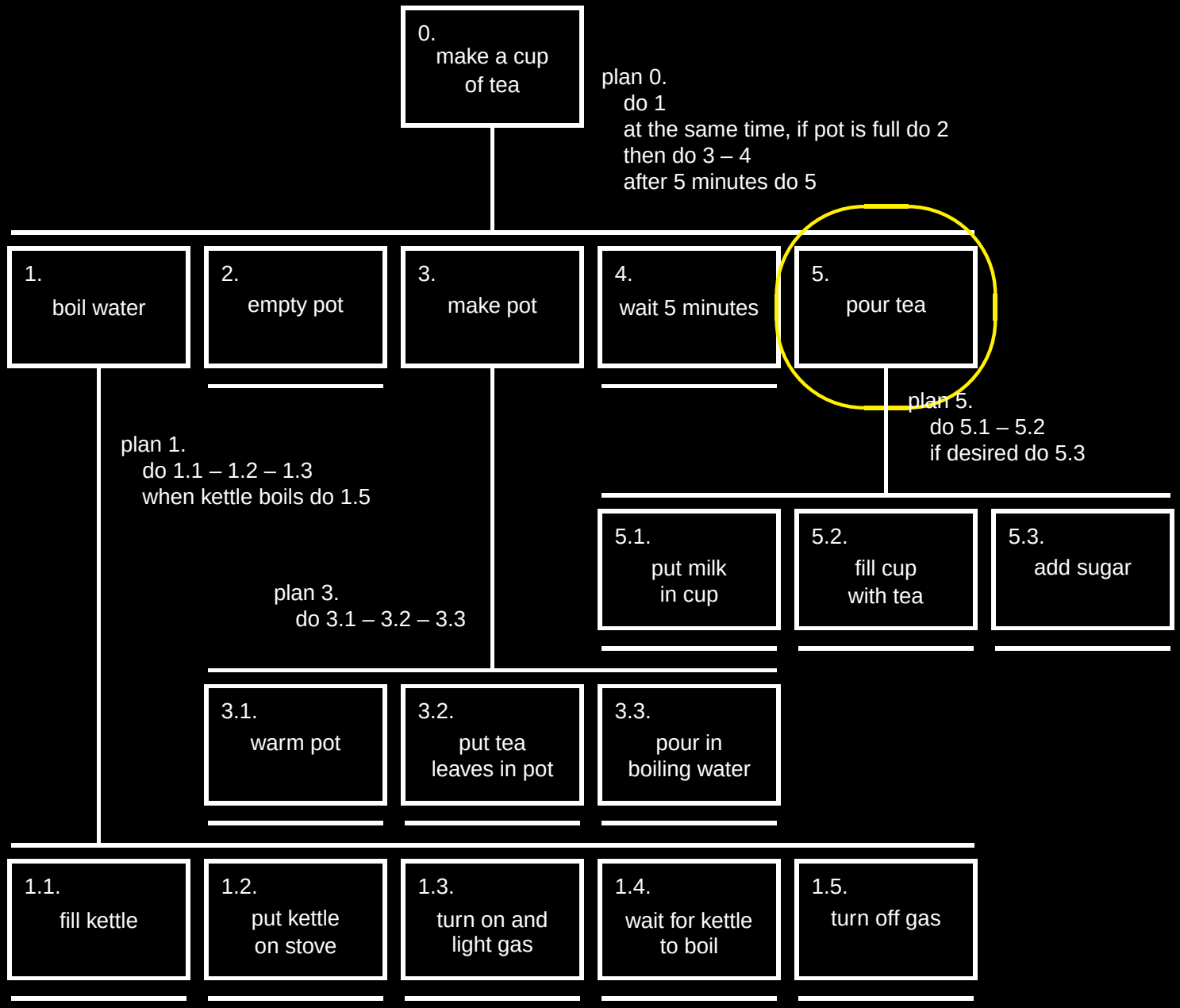
HTA: Further Decompositions



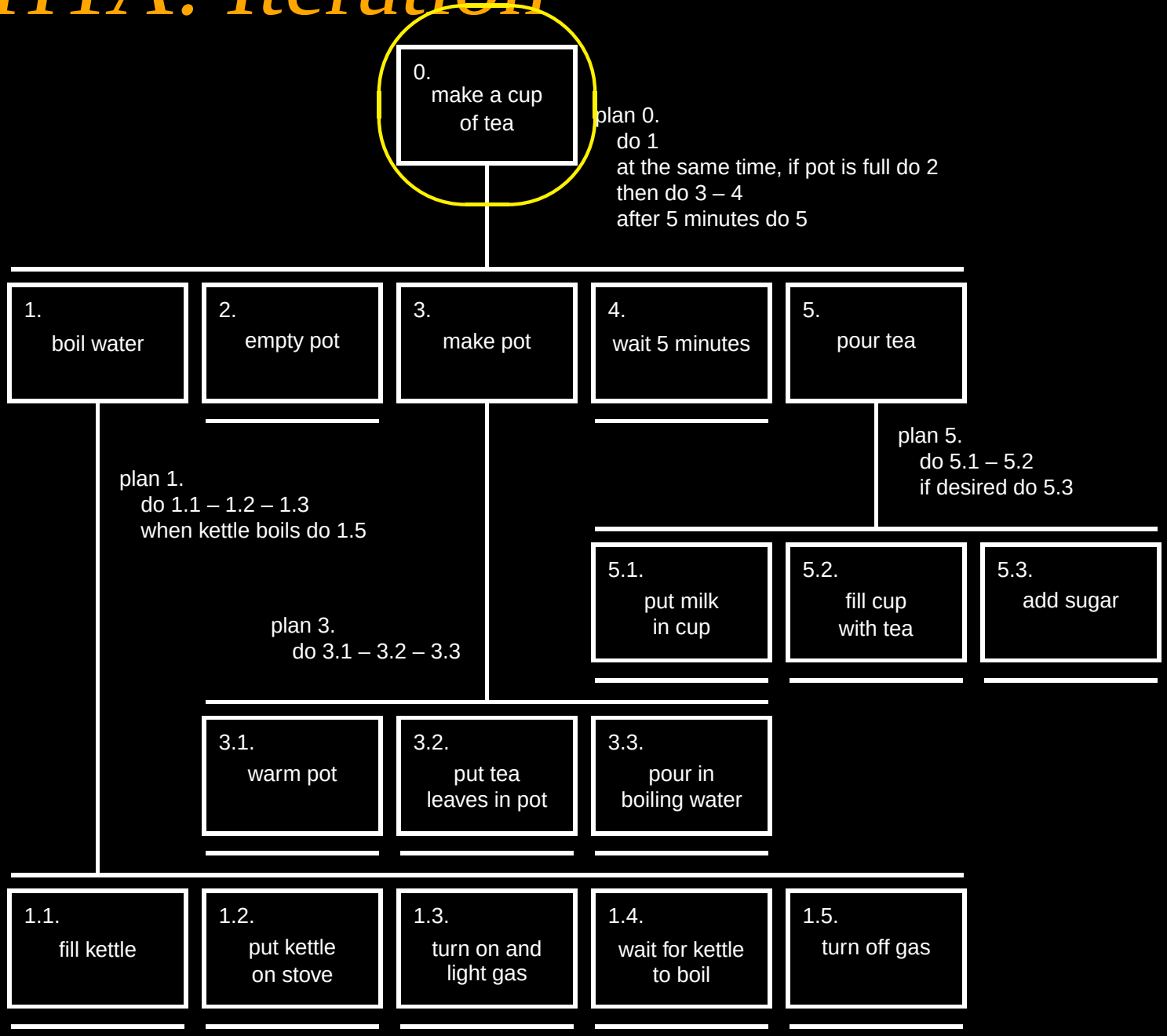
HTA: Further Decompositions



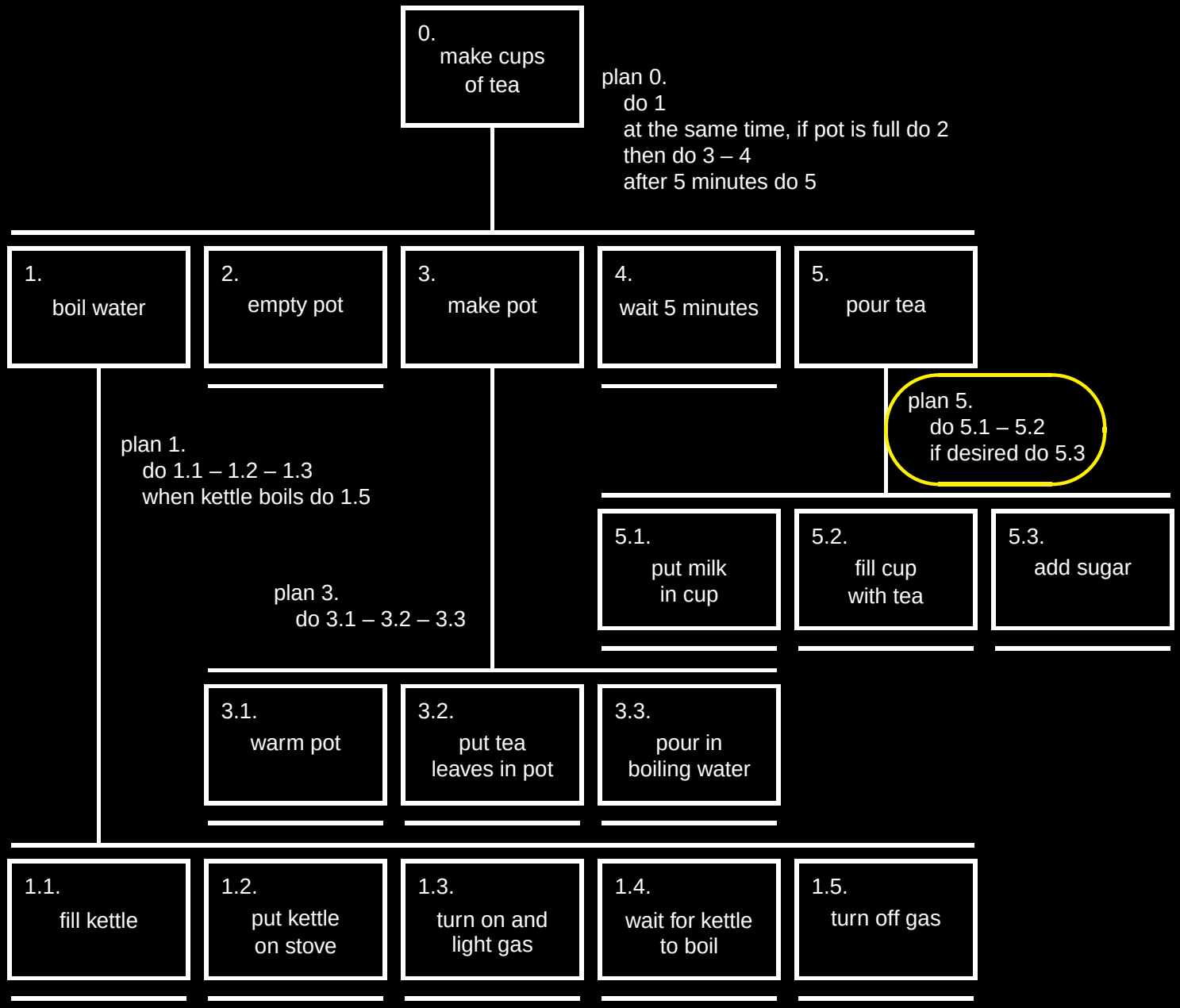
HTA: Further Decompositions



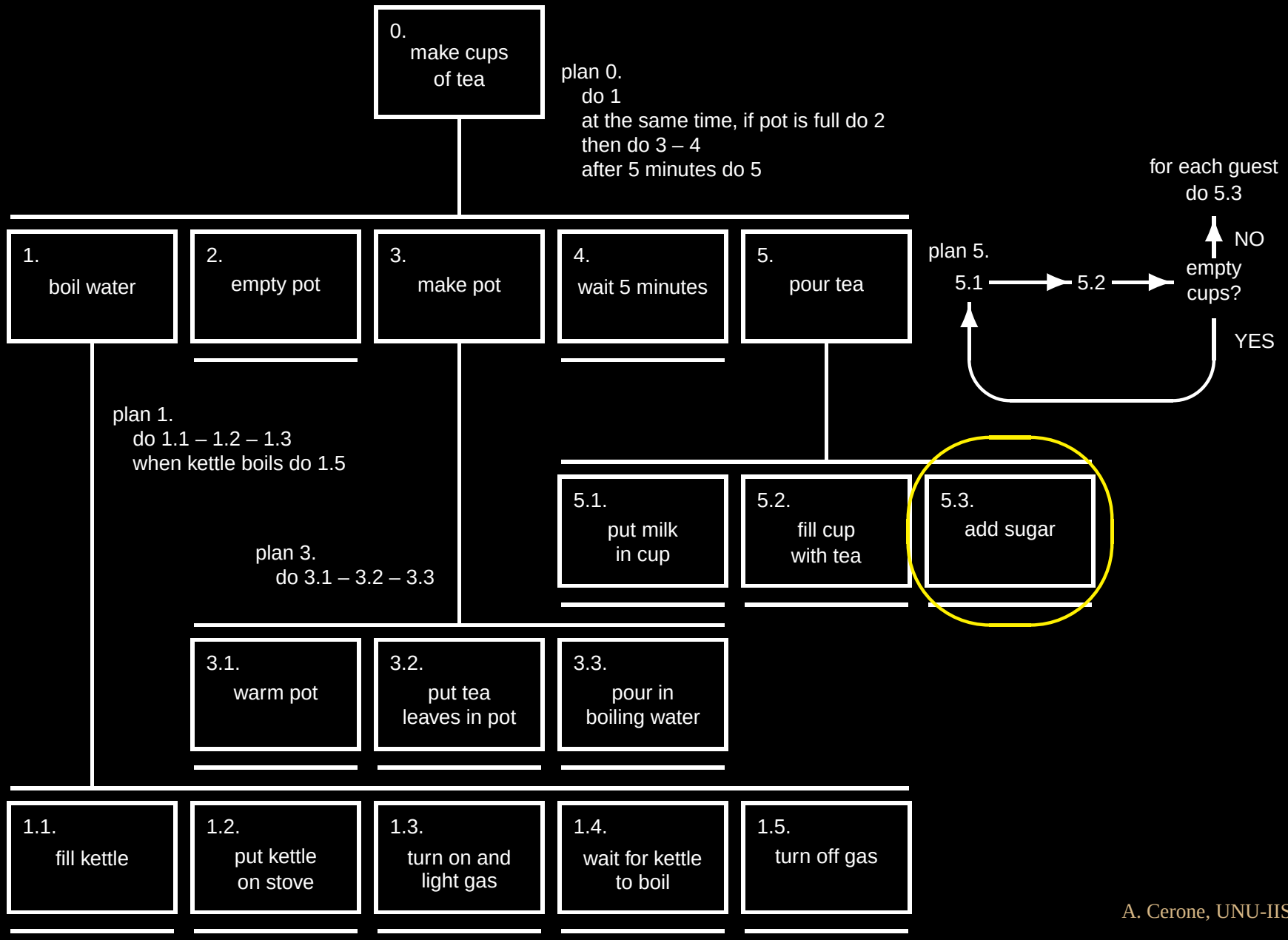
HTA: Iteration



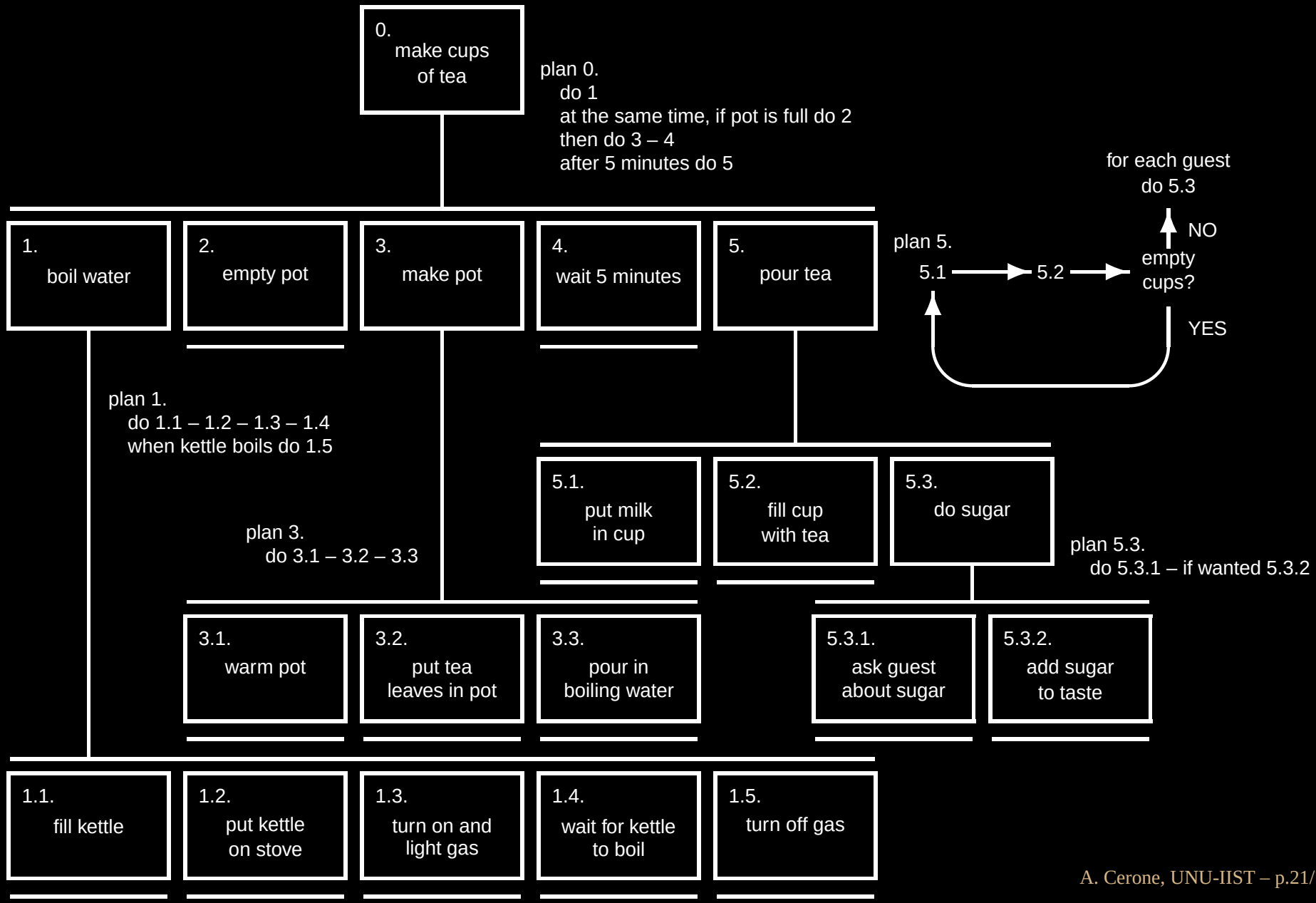
HTA: Iteration



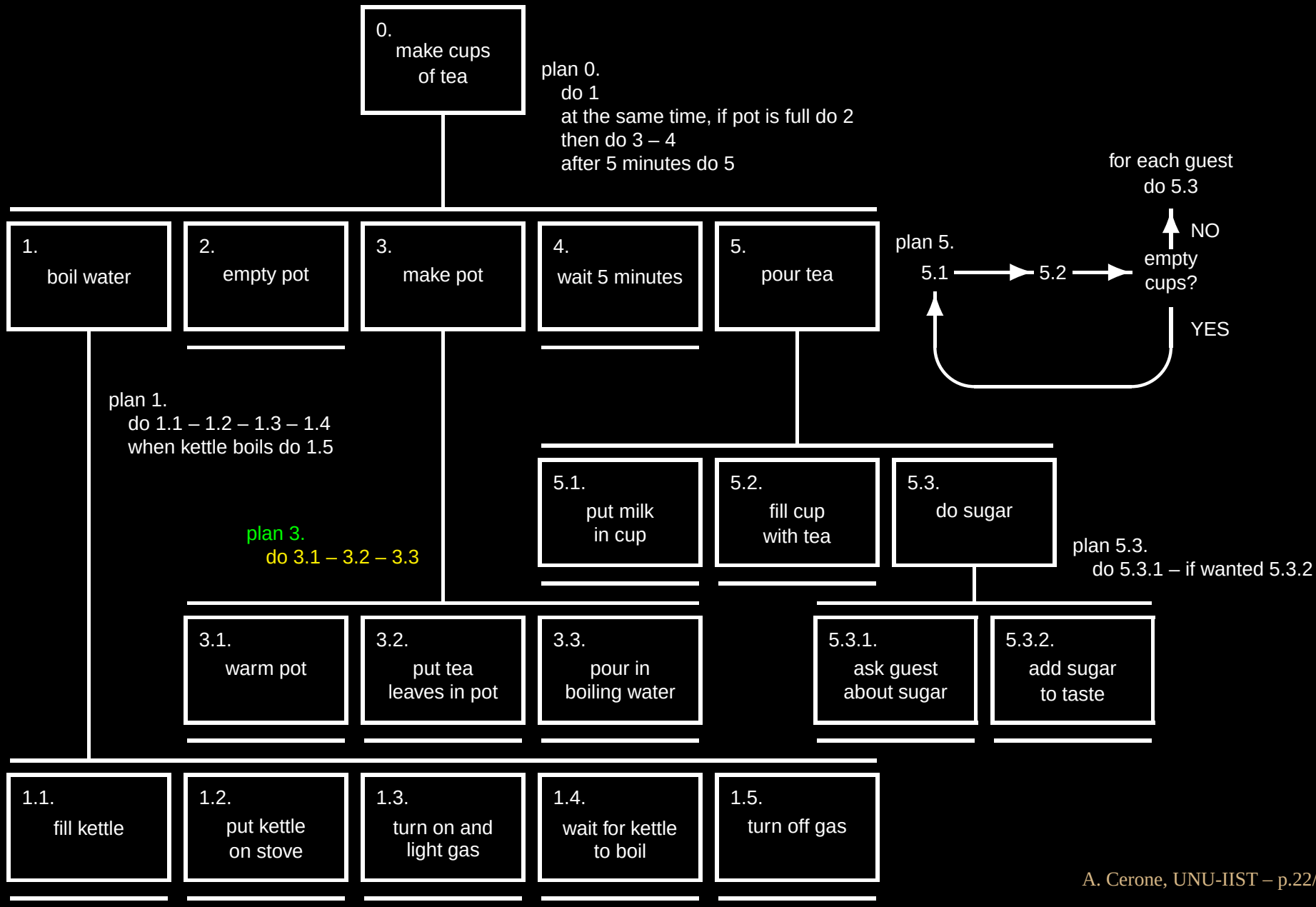
HTA: Iteration



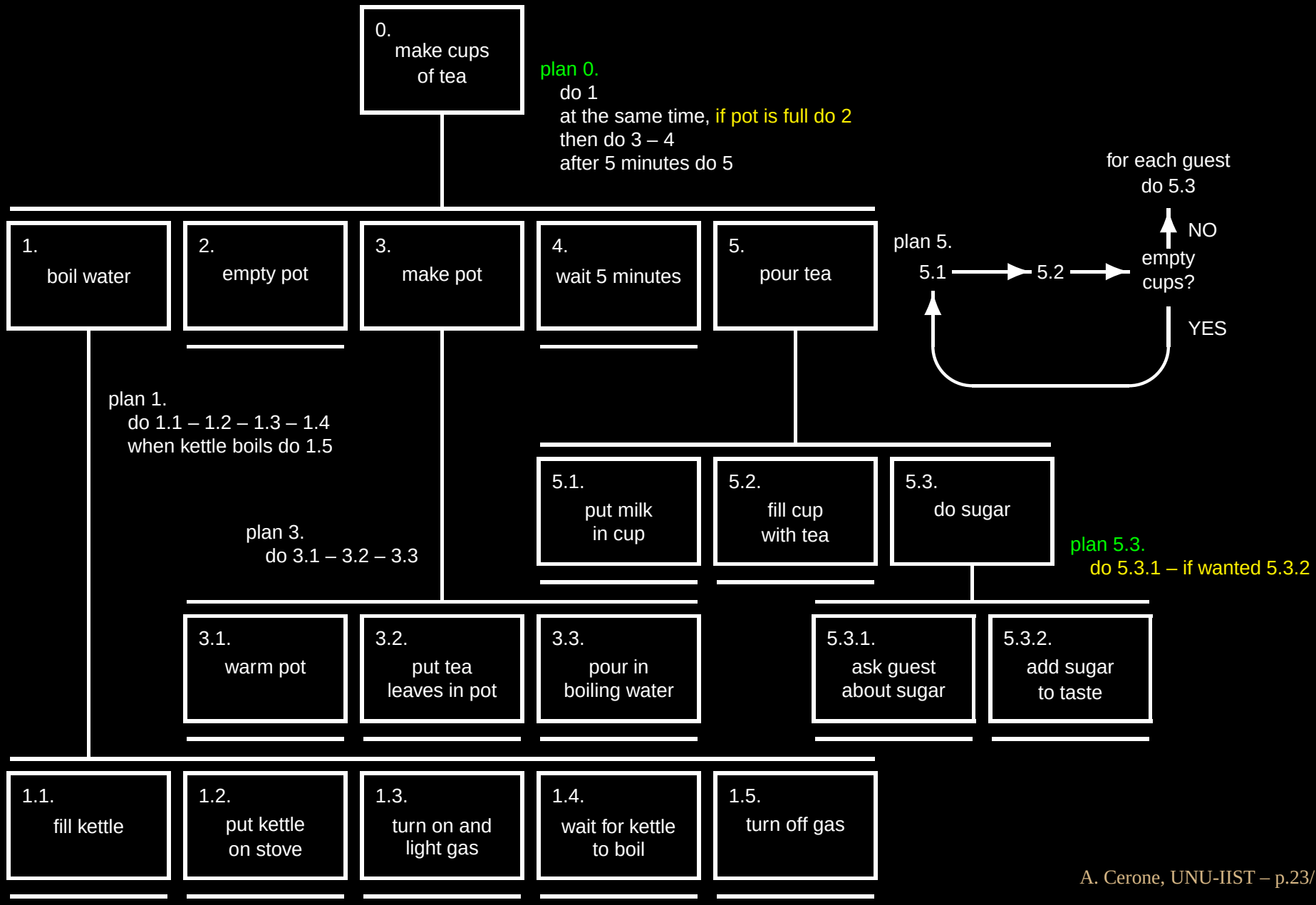
HTA: Final Decomposition



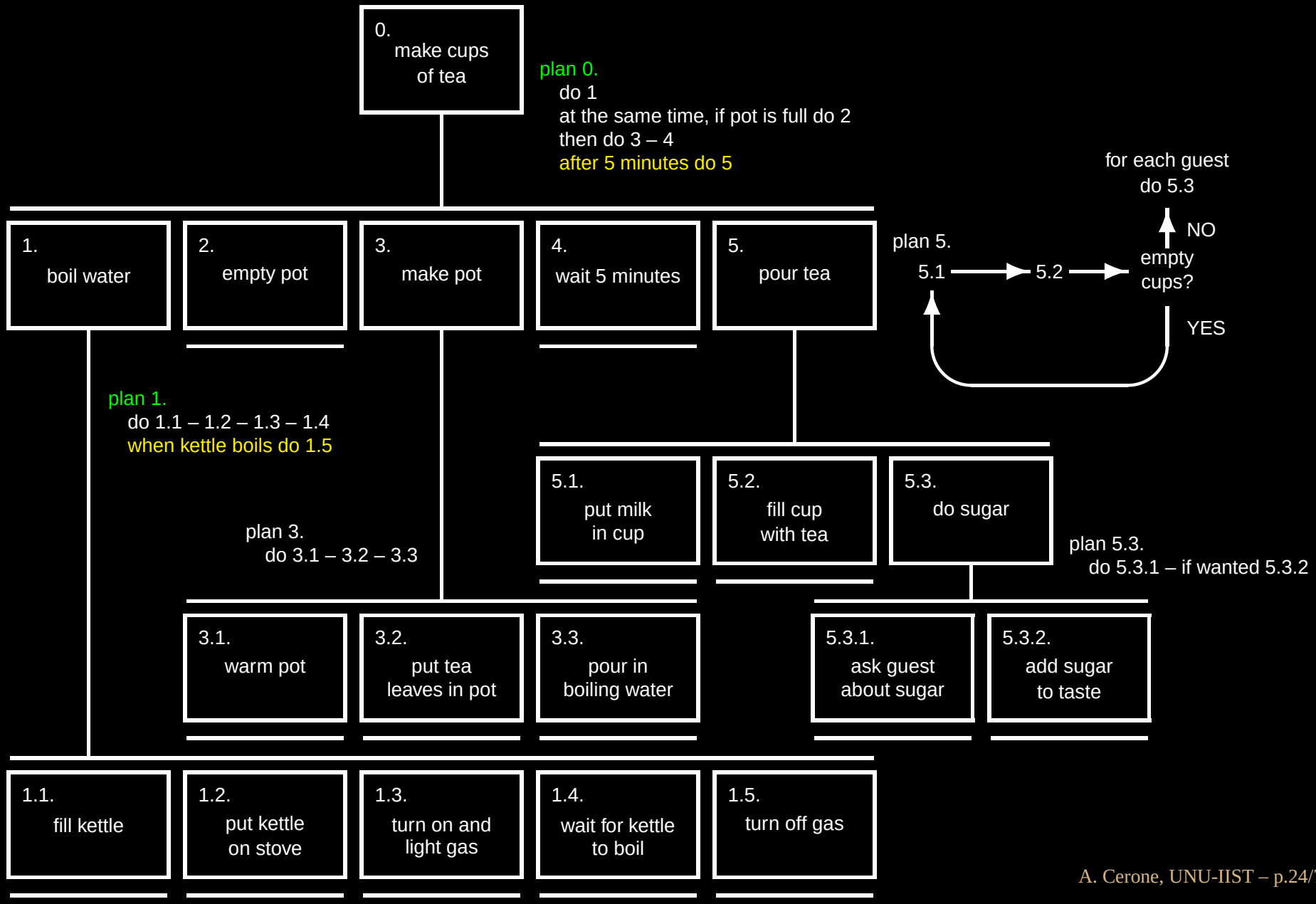
HTA: Fixed Sequence



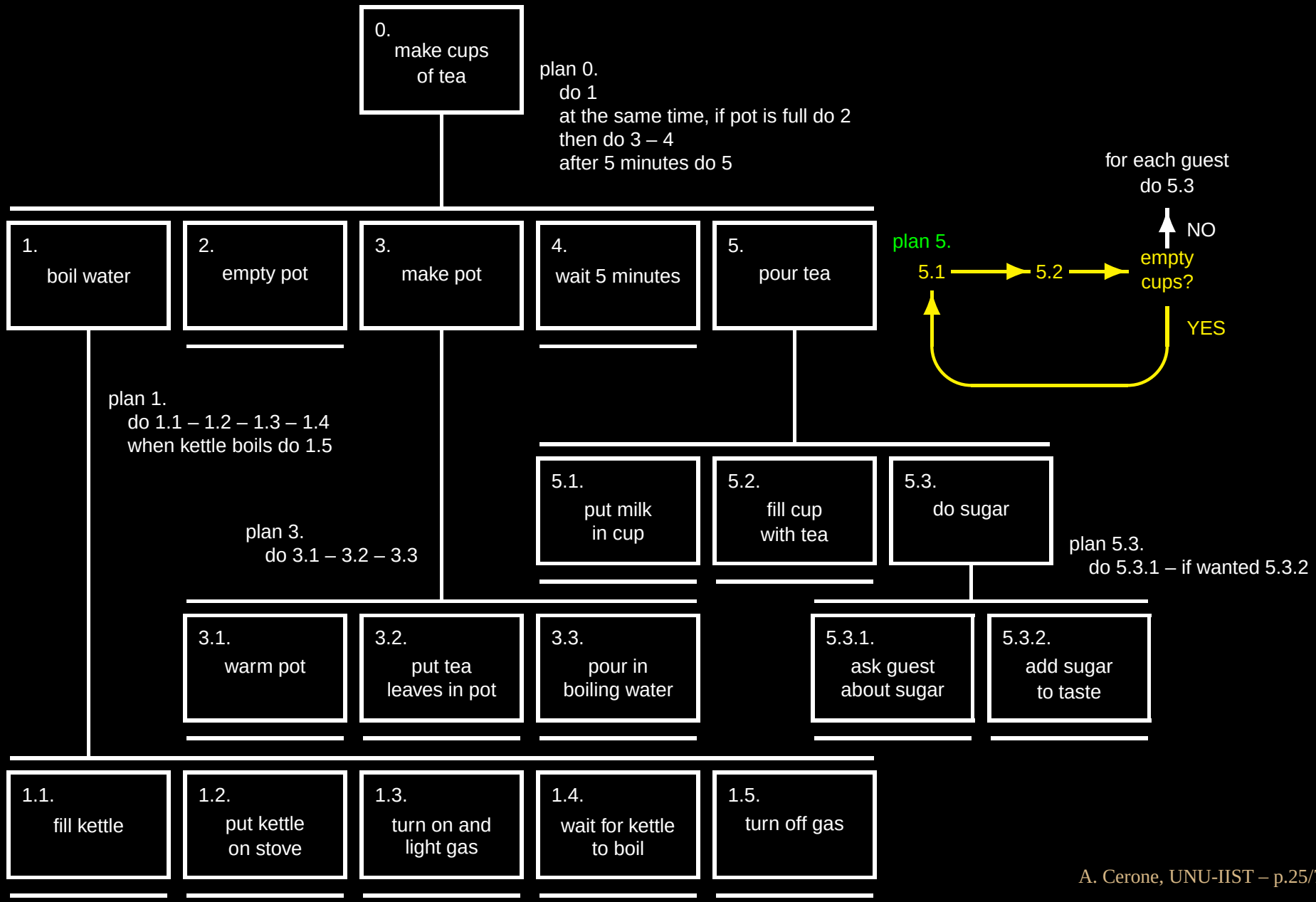
HTA: Optional Tasks



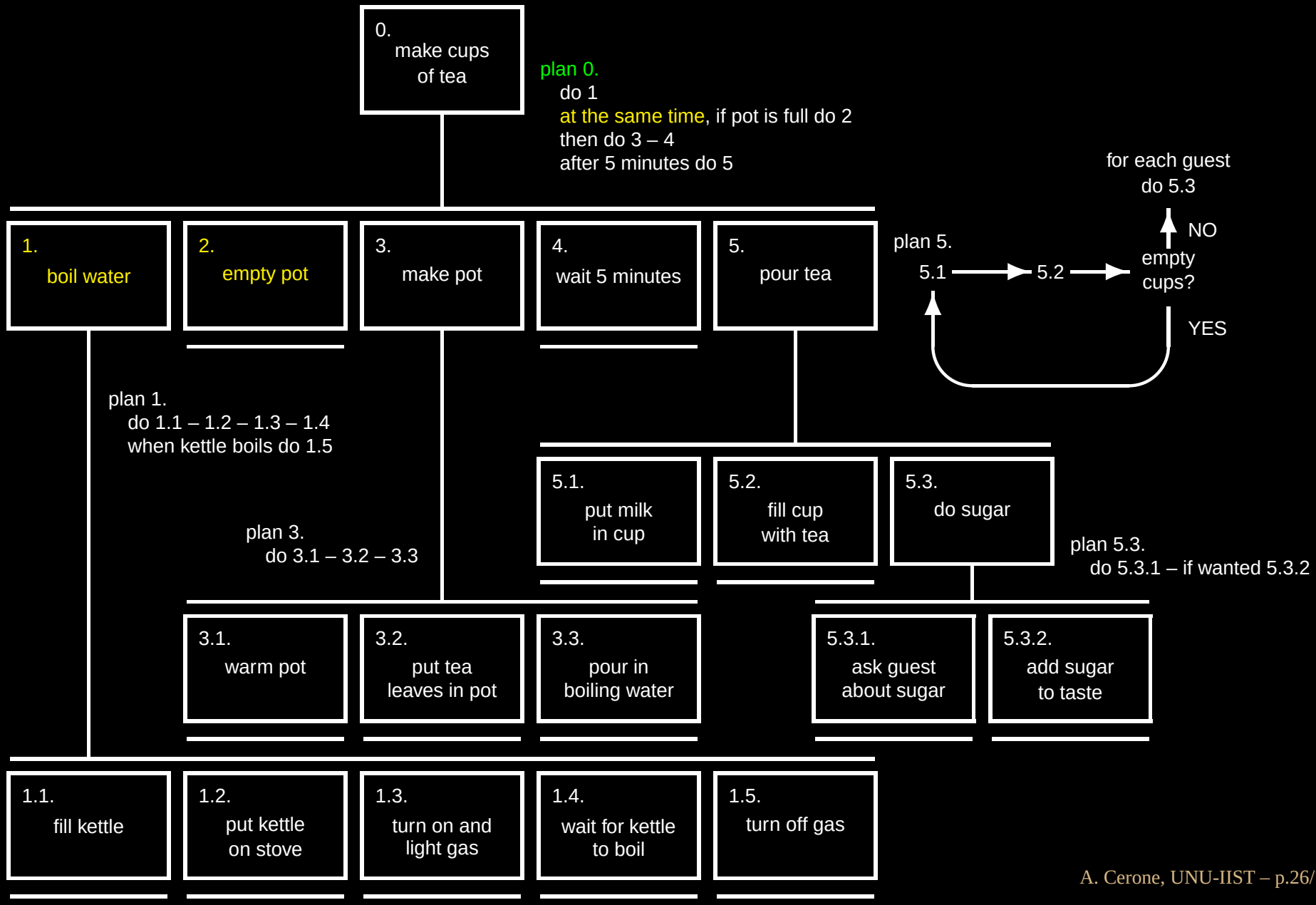
HTA: Waiting for Events



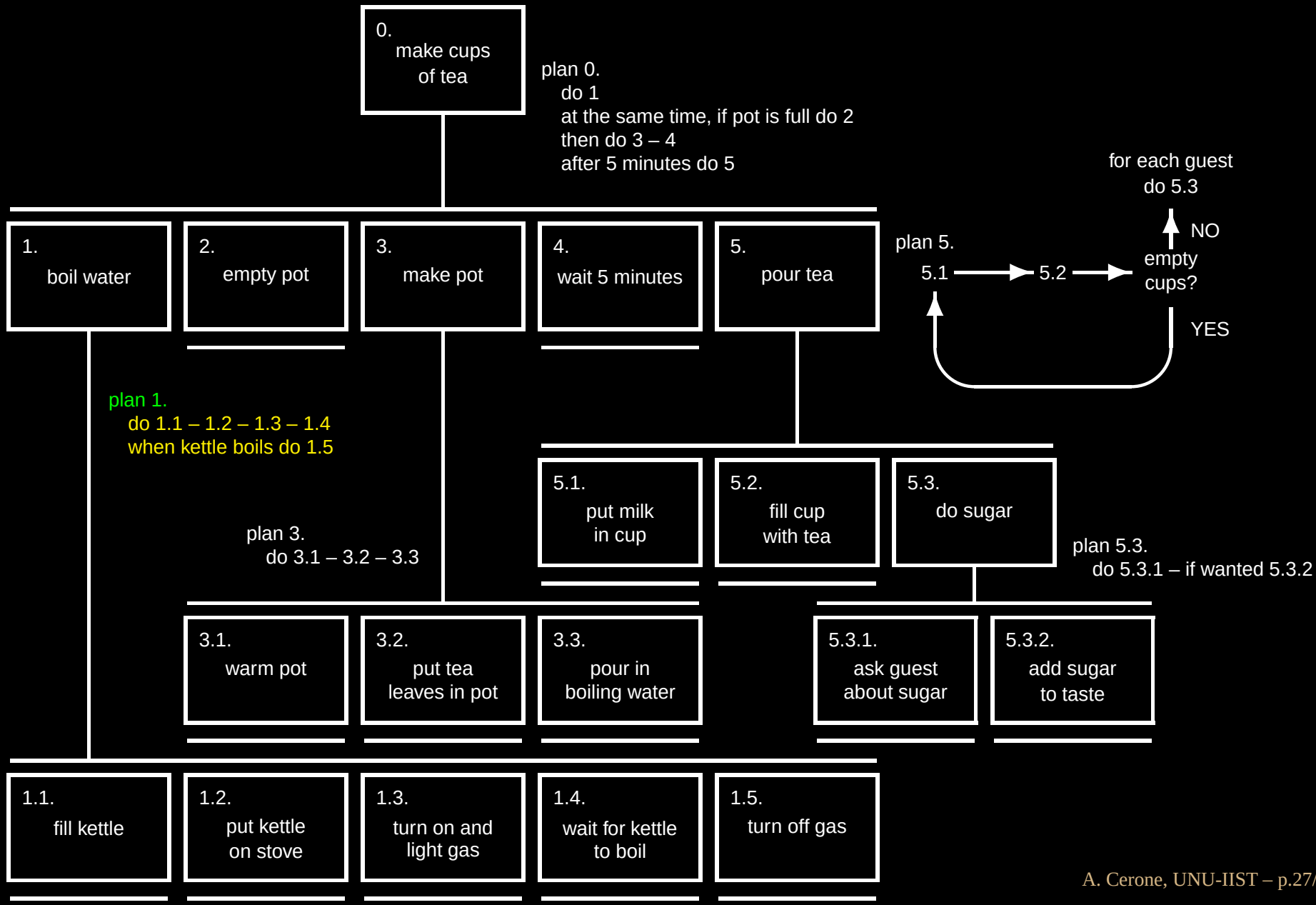
HTA: Cycles


0.
make cups of tea
plan 0.
do 1
at the same time, if pot is full do 2
then do 3 – 4
after 5 minutes do 5
1.
boil water
2.
empty pot
3.
make pot
4.
wait 5 minutes
5.
pour tea
plan 5.
5.1
5.2
empty cups?
NO
YES
for each guest
do 5.3
plan 1.
do 1.1 – 1.2 – 1.3 – 1.4
when kettle boils do 1.5
plan 3.
do 3.1 – 3.2 – 3.3
5.1.
put milk in cup
5.2.
fill cup with tea
5.3.
do sugar
plan 5.3.
do 5.3.1 – if wanted 5.3.2
3.1.
warm pot
3.2.
put tea leaves in pot
3.3.
pour in boiling water
5.3.1.
ask guest about sugar
5.3.2.
add sugar to taste
1.1.
fill kettle
1.2.
put kettle on stove
1.3.
turn on and light gas
1.4.
wait for kettle to boil
1.5.
turn off gas

HTA: Time-sharing



HTA: Mixtures



Decomposition Heuristics

- paired actions
  e.g., turn on and turn off gas
- restructure/balance e.g., generate make pot and decompose pour tea
- generalise
  e.g., from make a cup of tea to make cups of tea

Task Failure Decompositions

ATM Properties in LTL

Functional Correctness:

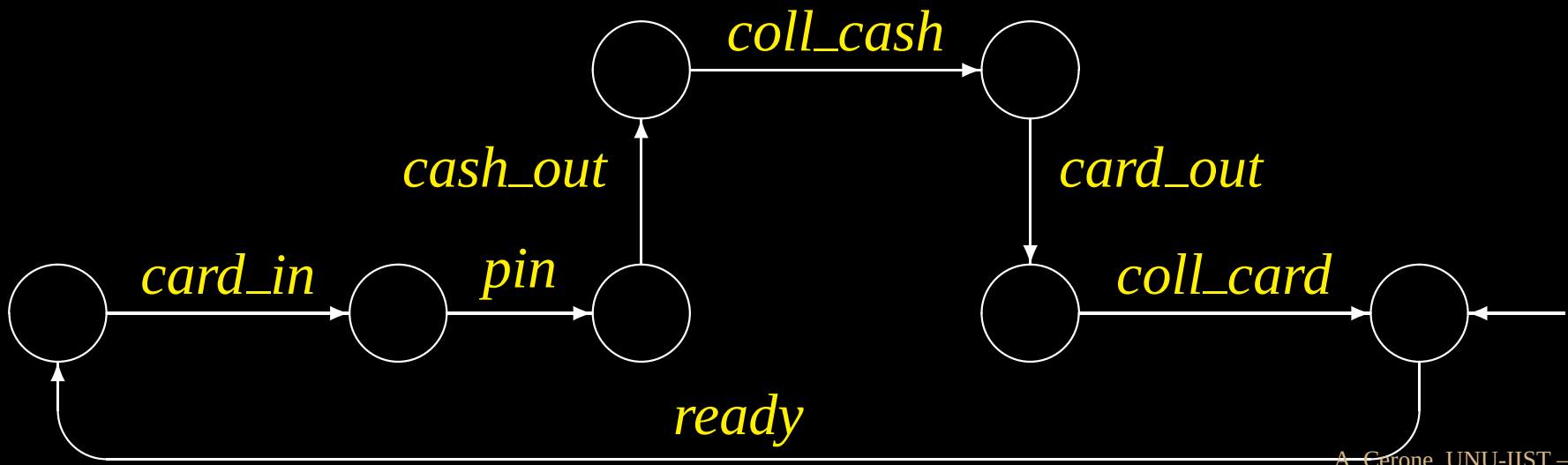
The ATM machine will eventually deliver cash

$$\Box(ready \rightarrow \Diamond cash\_out)$$

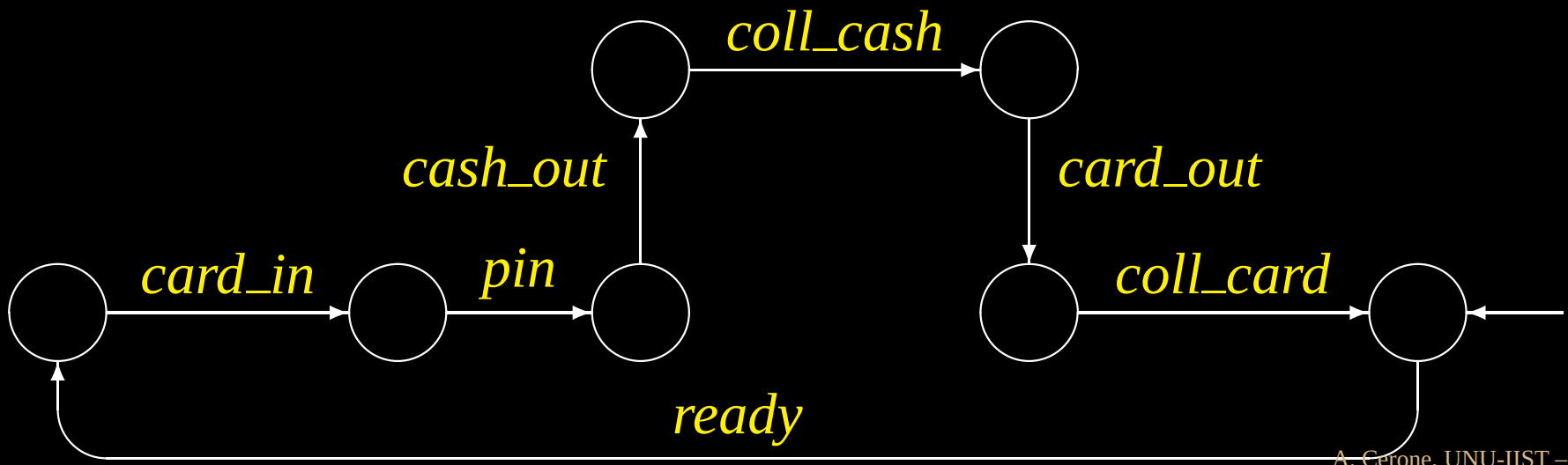
Safety:

The ATM machine will eventually return the card

$$\Box(ready \rightarrow \Diamond card\_out)$$

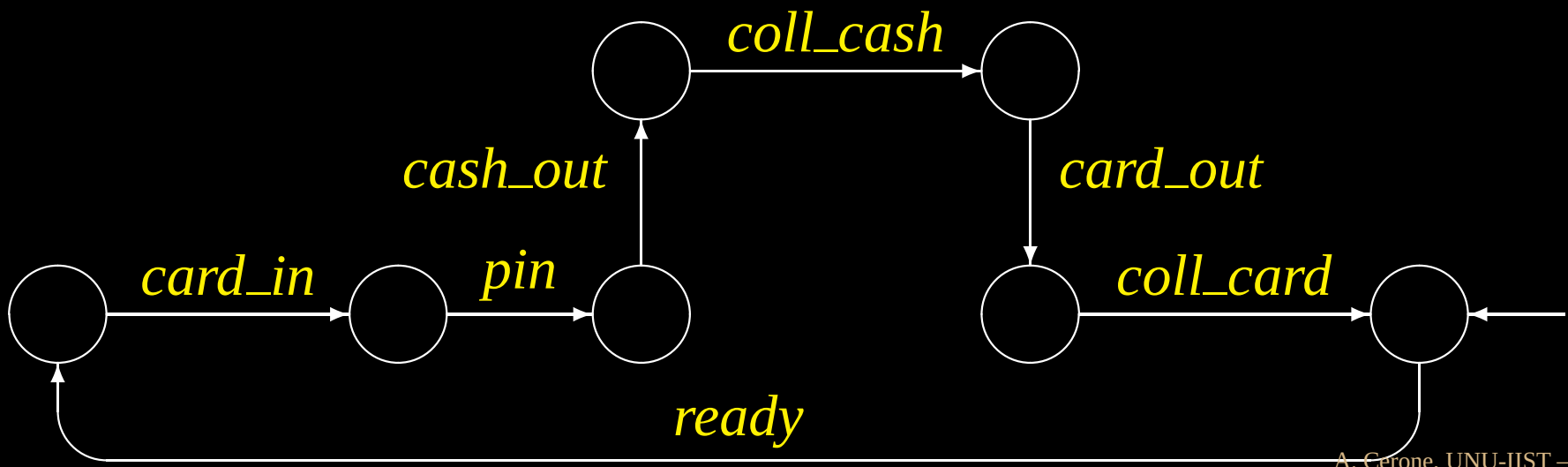


Example: ATM Machine



Example: ATM Machine

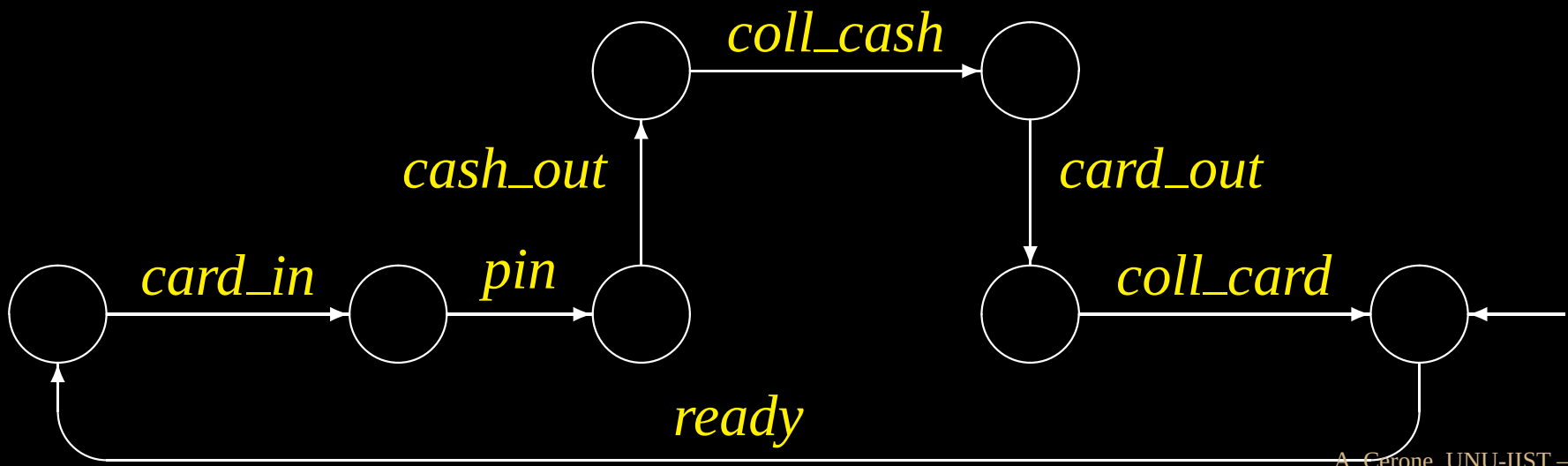
Goal: $\Box(ready \rightarrow \Diamond coll\_cash)$



Example: ATM Machine

Goal: $\Box(ready \rightarrow \Diamond coll\_cash)$

Safety: $\Box(ready \rightarrow \Diamond coll\_card)$

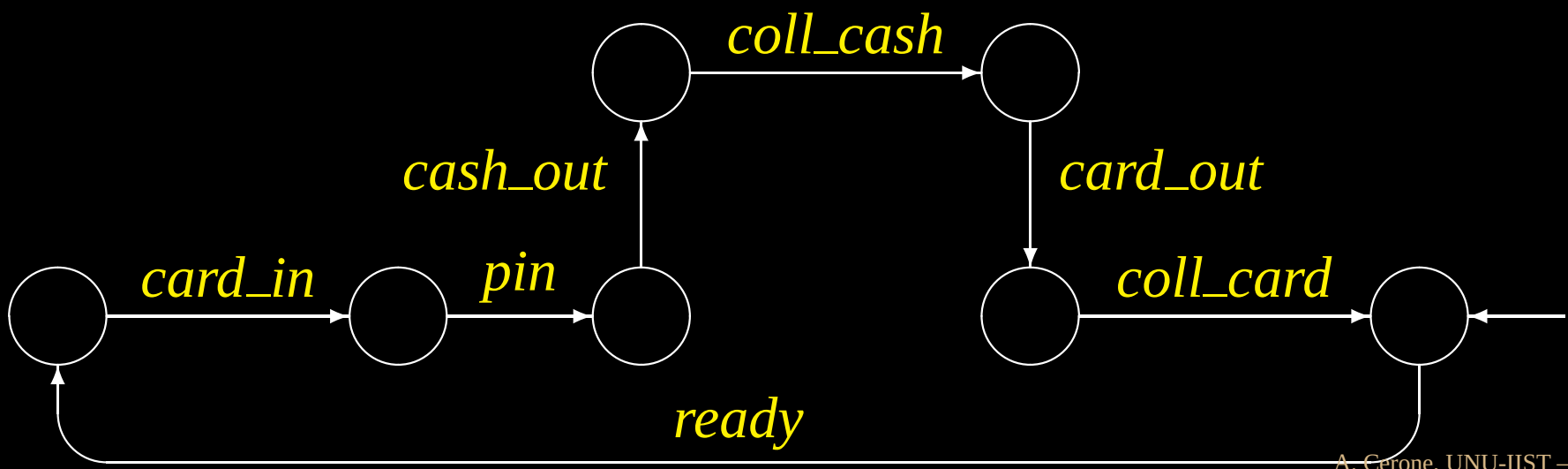


Example: ATM Machine

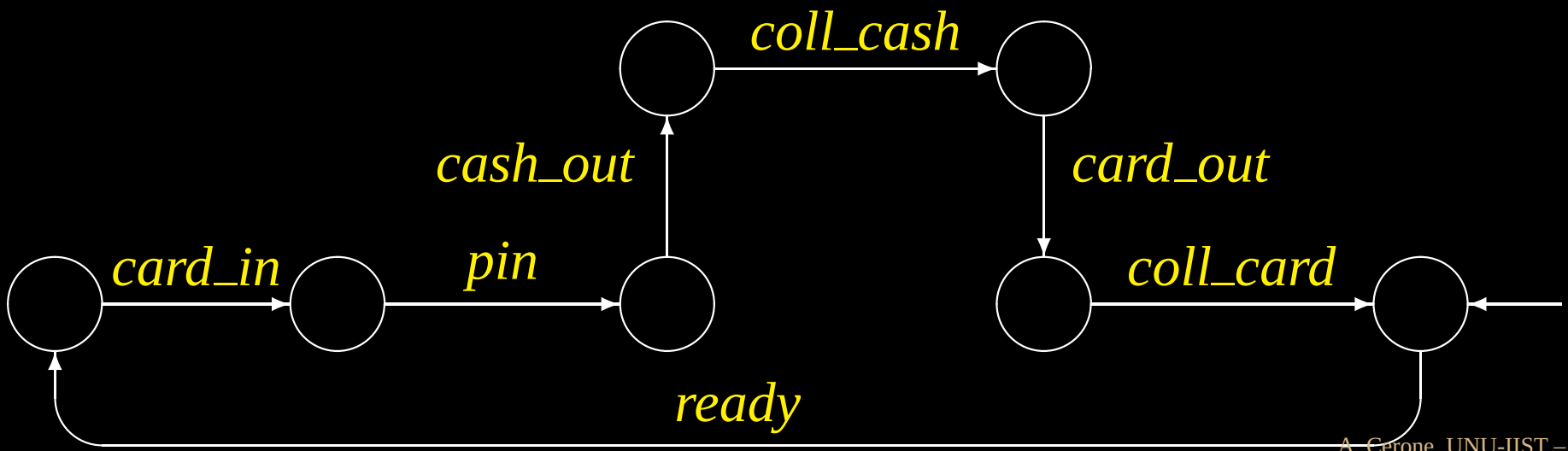
Goal: $\Box(ready \rightarrow \Diamond coll\_cash)$

Safety: $\Box(ready \rightarrow \Diamond coll\_card)$

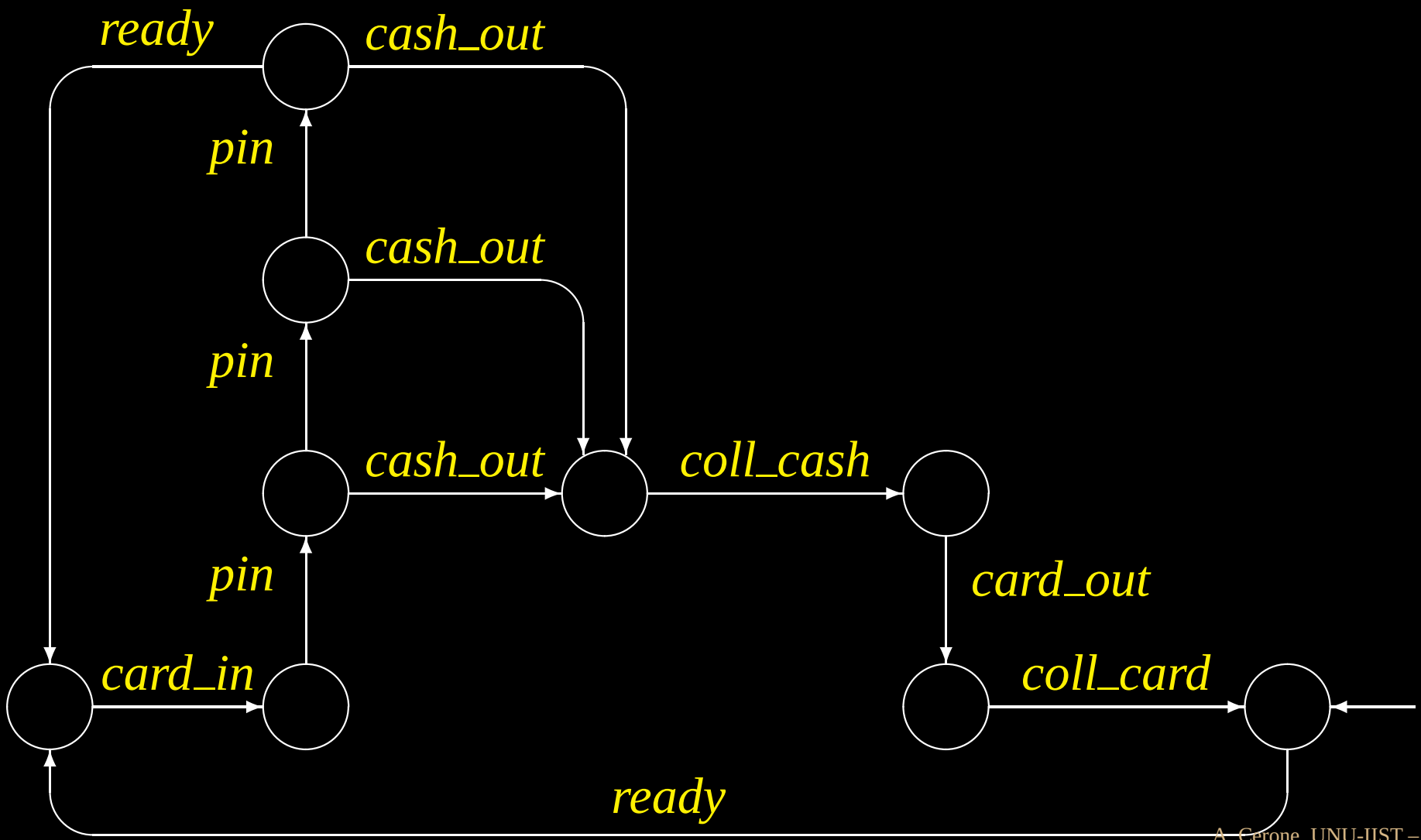
Task: $\Box(ready \rightarrow ((\Diamond coll\_cash) \wedge (\Diamond coll\_card)))$



Refined ATM Machine



Refined ATM Machine



ATM: Task Failure

Goal: $\Box(ready \rightarrow \Diamond coll\_cash)$

Safety: $\Box(ready \rightarrow \Diamond coll\_card)$

Task: $\Box(ready \rightarrow ((\Diamond coll\_cash) \wedge (\Diamond coll\_card)))$

ATM: Task Failure

Goal: $\Box(ready \rightarrow \Diamond coll\_cash)$

Safety: $\Box(ready \rightarrow \Diamond coll\_card)$

Task: $\Box(ready \rightarrow ((\Diamond coll\_cash) \wedge (\Diamond coll\_card)))$

Task Failure:

$$\neg\, \Box((\Diamond coll\_cash) \wedge (\Diamond coll\_card))$$

ATM: Task Failure

Goal: $\Box(ready \rightarrow \Diamond coll\_cash)$

Safety: $\Box(ready \rightarrow \Diamond coll\_card)$

Task: $\Box(ready \rightarrow ((\Diamond coll\_cash) \wedge (\Diamond coll\_card)))$

Task Failure:

$$\Diamond(\ (\Box\neg\ coll\_cash) \vee (\Box\neg\ coll\_card))$$

ATM: Task Failure

Goal: $\Box(ready \rightarrow \Diamond coll\_cash)$

Safety: $\Box(ready \rightarrow \Diamond coll\_card)$

Task: $\Box(ready \rightarrow ((\Diamond coll\_cash) \wedge (\Diamond coll\_card)))$

Task Failure:

$$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$$

Task Failure Decomposition

Top-level Task Failure:

$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$

Task Failure Decomposition

Top-level Task Failure:

$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$

1. input wrong pin three times in a row
   $\Longrightarrow$ card confiscated and cash not collected

Task Failure Decomposition

Top-level Task Failure:

$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$

1. input wrong pin three times in a row
   $\Longrightarrow$ card confiscated and cash not collected
   $(\Box\neg\, coll\_cash) \wedge (\Box\neg\, coll\_card))$

Task Failure Decomposition

Top-level Task Failure:

$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$

1. input wrong pin three times in a row
   $\Longrightarrow$ card confiscated and cash not collected
   $(\Box\neg\, coll\_cash) \wedge (\Box\neg\, coll\_card))$
2. collect cash but not card

Task Failure Decomposition

Top-level Task Failure:

$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$

1. input wrong pin three times in a row
   $\Longrightarrow$ card confiscated and cash not collected
   $(\Box\neg\, coll\_cash) \wedge (\Box\neg\, coll\_card))$
2. collect cash but not card
   $(\Diamond coll\_cash) \wedge (\Box\neg\, coll\_card))$

Task Failure Decomposition

Top-level Task Failure:

$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$

1. input wrong pin three times in a row
   $\Longrightarrow$ card confiscated and cash not collected
   $(\Box\neg\, coll\_cash) \wedge (\Box\neg\, coll\_card))$
2. collect cash but not card
   $(\Diamond coll\_cash) \wedge (\Box\neg\, coll\_card))$
3. collect card but not cash

Task Failure Decomposition

Top-level Task Failure:

$(\Box\neg\, coll\_cash) \vee (\Box\neg\, coll\_card))$

1. input wrong pin three times in a row
   $\Longrightarrow$ card confiscated and cash not collected
   $(\Box\neg\, coll\_cash) \wedge (\Box\neg\, coll\_card))$
2. collect cash but not card
   $(\Diamond coll\_cash) \wedge (\Box\neg\, coll\_card))$
3. collect card but not cash
   $(\Diamond coll\_card) \wedge (\Box\neg\, coll\_cash))$

TF Psyc. Interpretation

Top-level Task Failure:

either card or cash is not collected

1. input wrong pin three times in a row
   $\Longrightarrow$ card confiscated and cash not collected
   $\Longleftarrow$ pin forgotten
2. collect cash but not card
   $\Longleftarrow$ forget to collect card due to postcompletion error
3. collect card but not cash
   $\Longleftarrow$ forget to collect cash

Purpose

of Task Failure Decomposition

- define specific failures (phenotype errors) that cause the top-level task failure

Purpose

of Task Failure Decomposition

- define specific failures (phenotype errors) that cause the top-level task failure
- use phenotype errors to improve the interface

Purpose

of Task Failure Decomposition

- define specific failures (phenotype errors) that cause the top-level task failure
- use phenotype errors to improve the interface
- give phenotype errors psychological interpretations (genotype errors)

Purpose

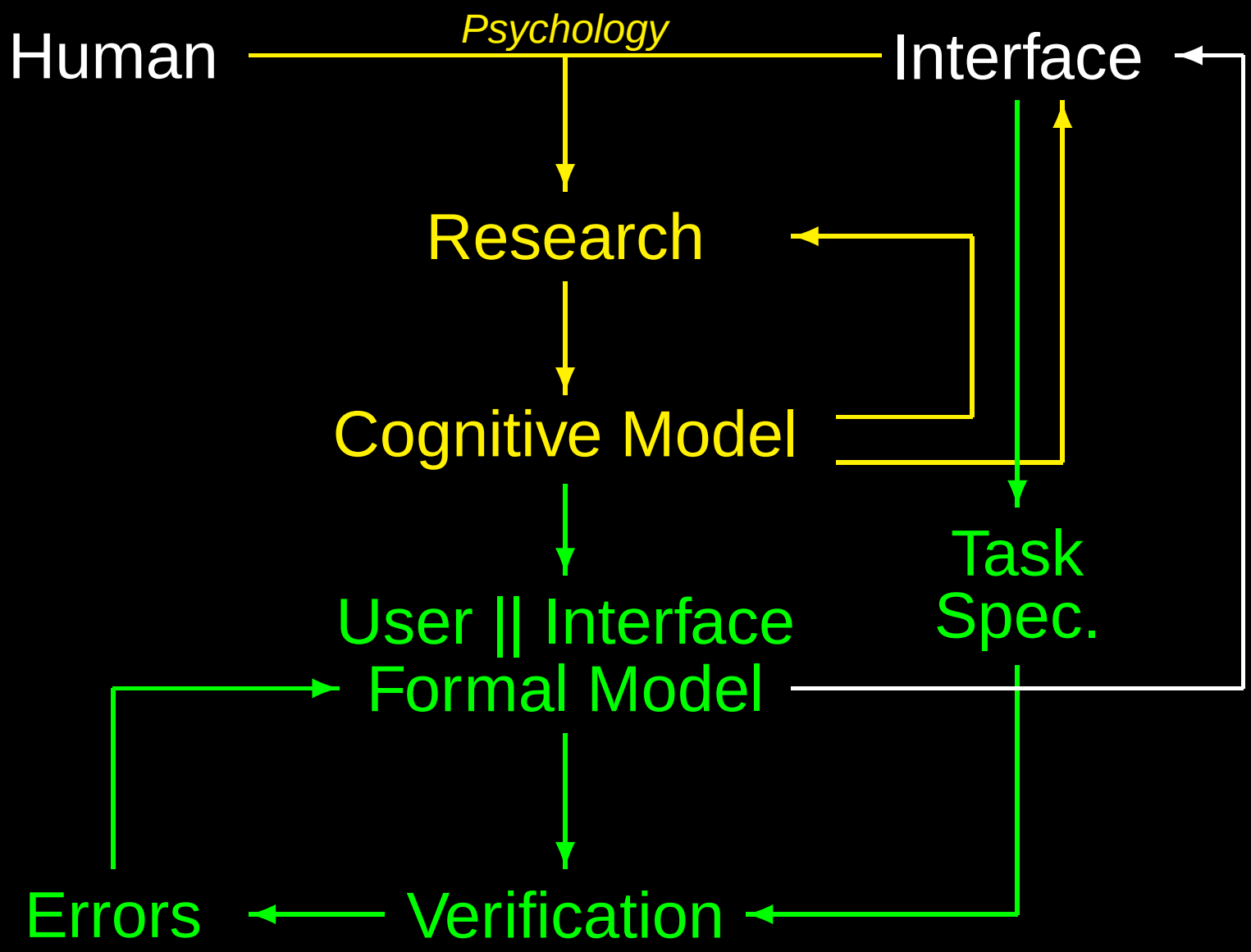
of Task Failure Decomposition

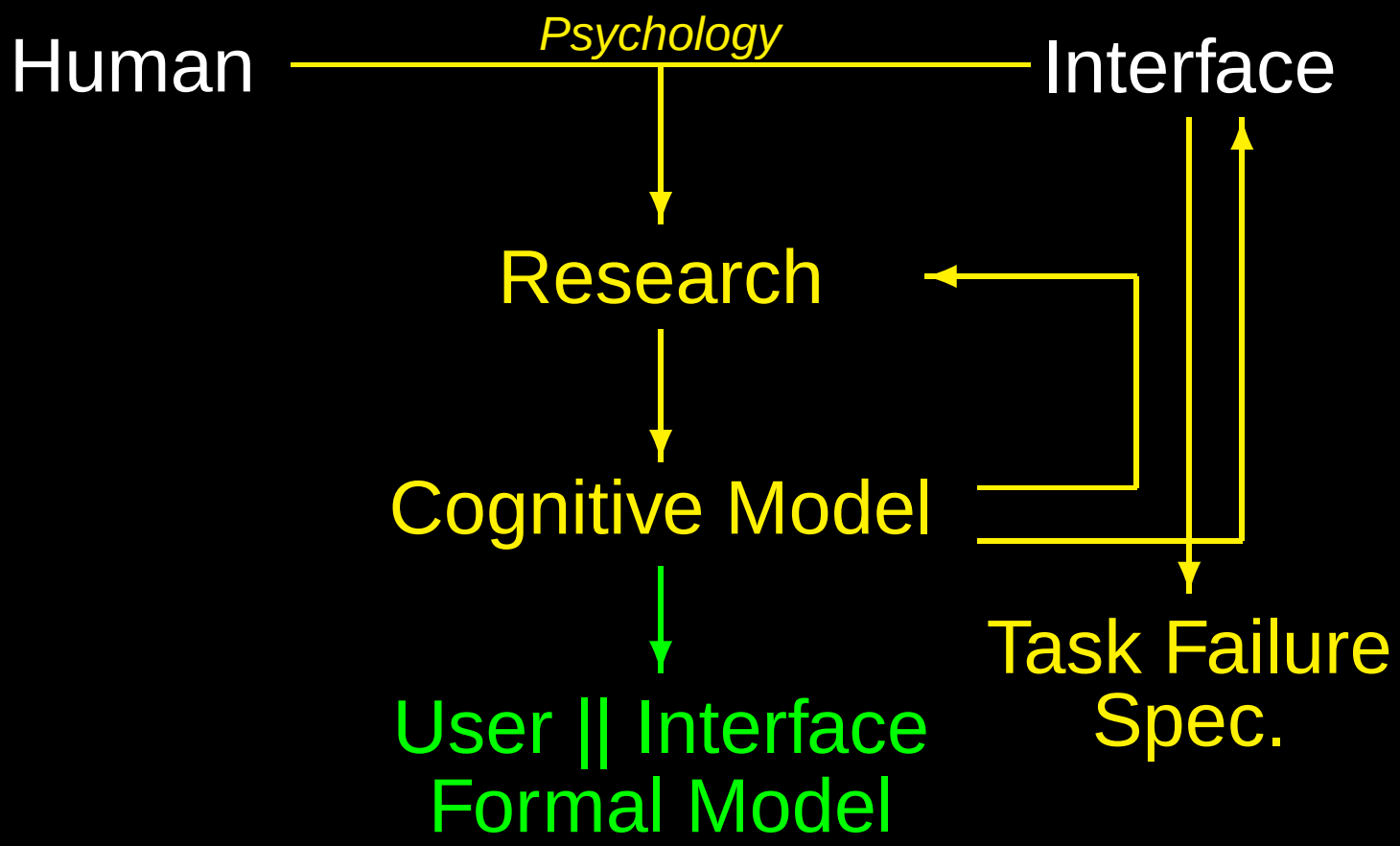
- define specific failures (phenotype errors) that cause the top-level task failure
- use phenotype errors to improve the interface
- give phenotype errors psychological interpretations (genotype errors)
- perform experiment to confirm causality genotype error $\Longrightarrow$ phenotype error

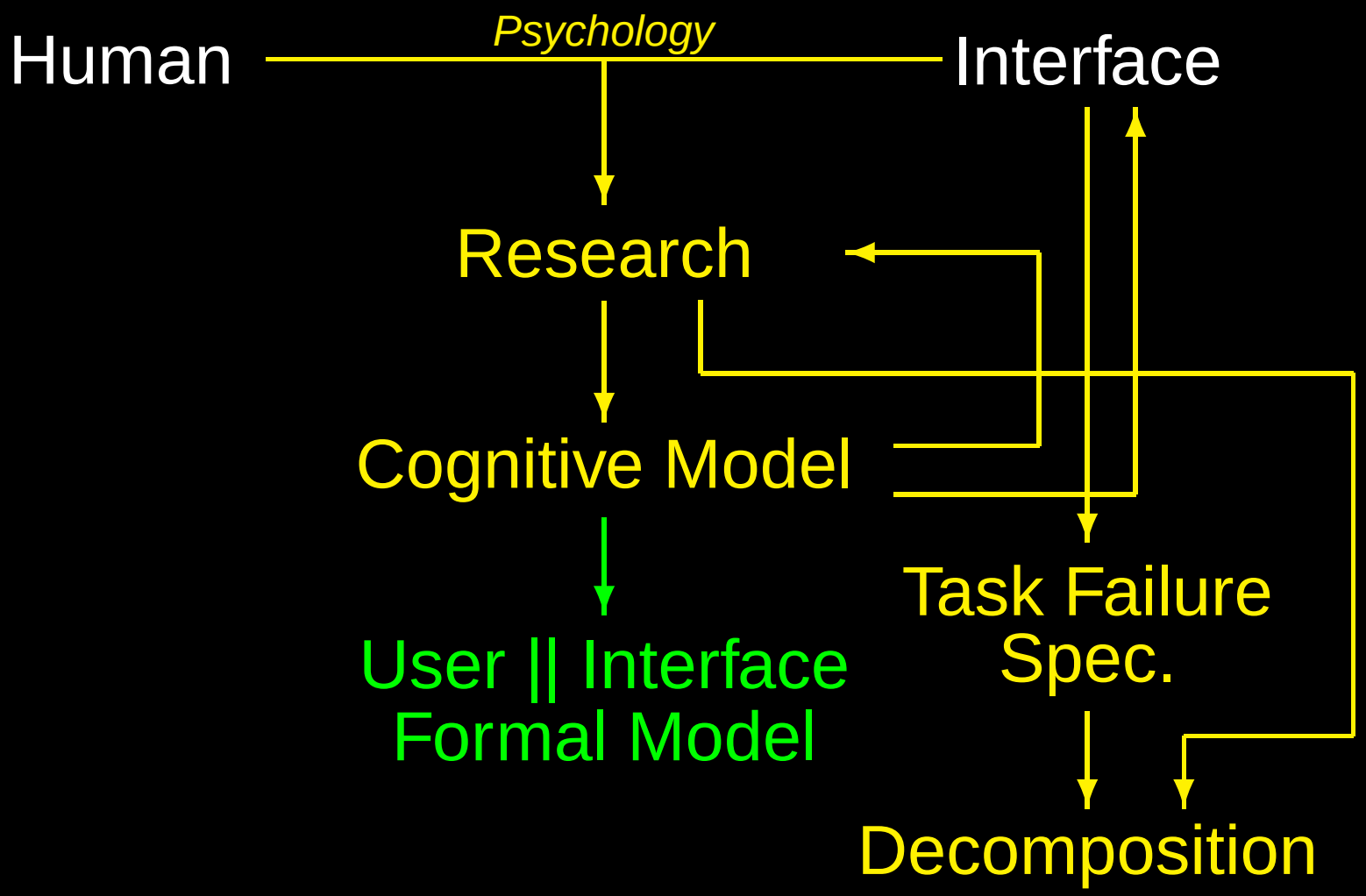
Purpose

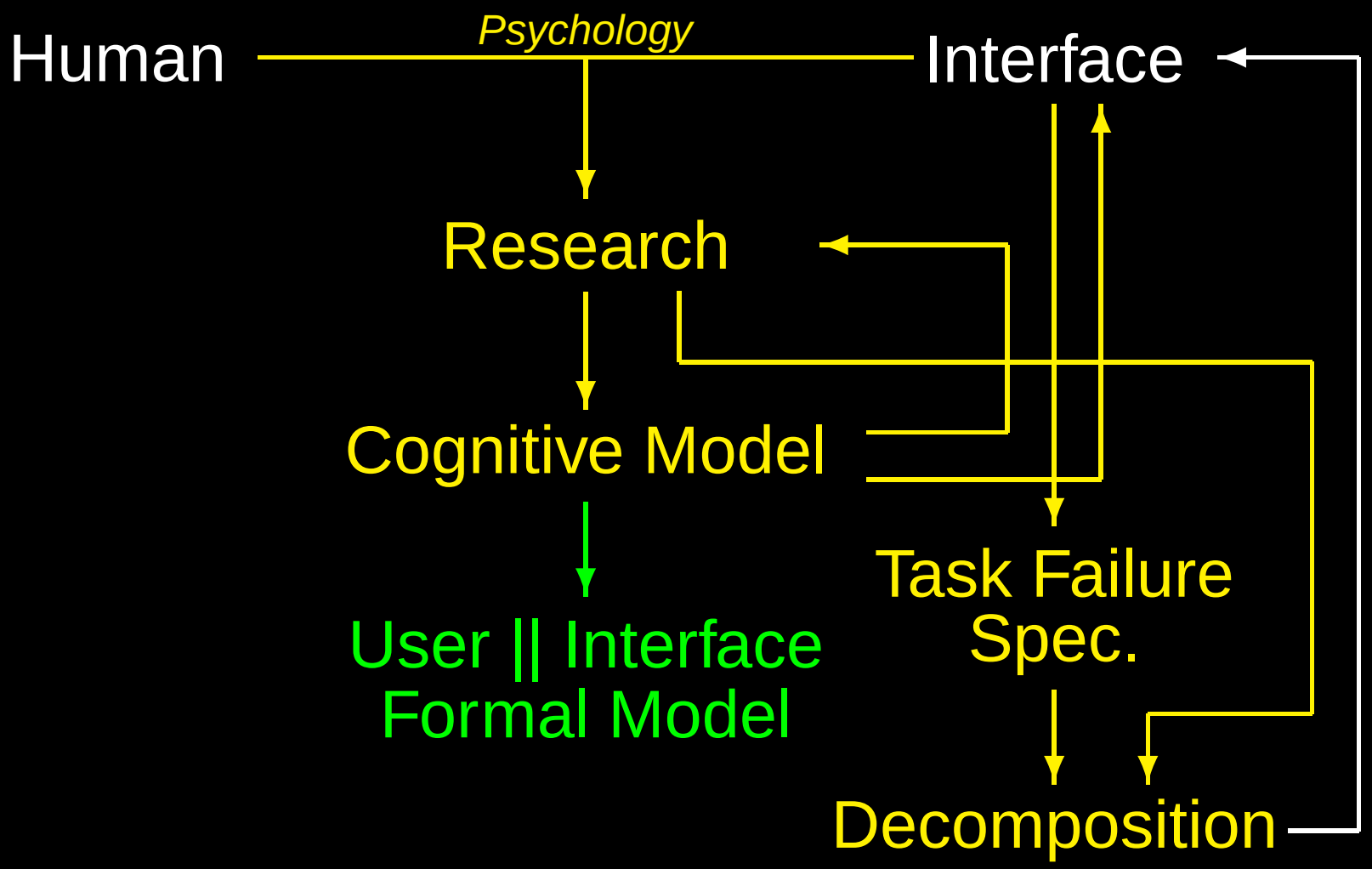
of Task Failure Decomposition

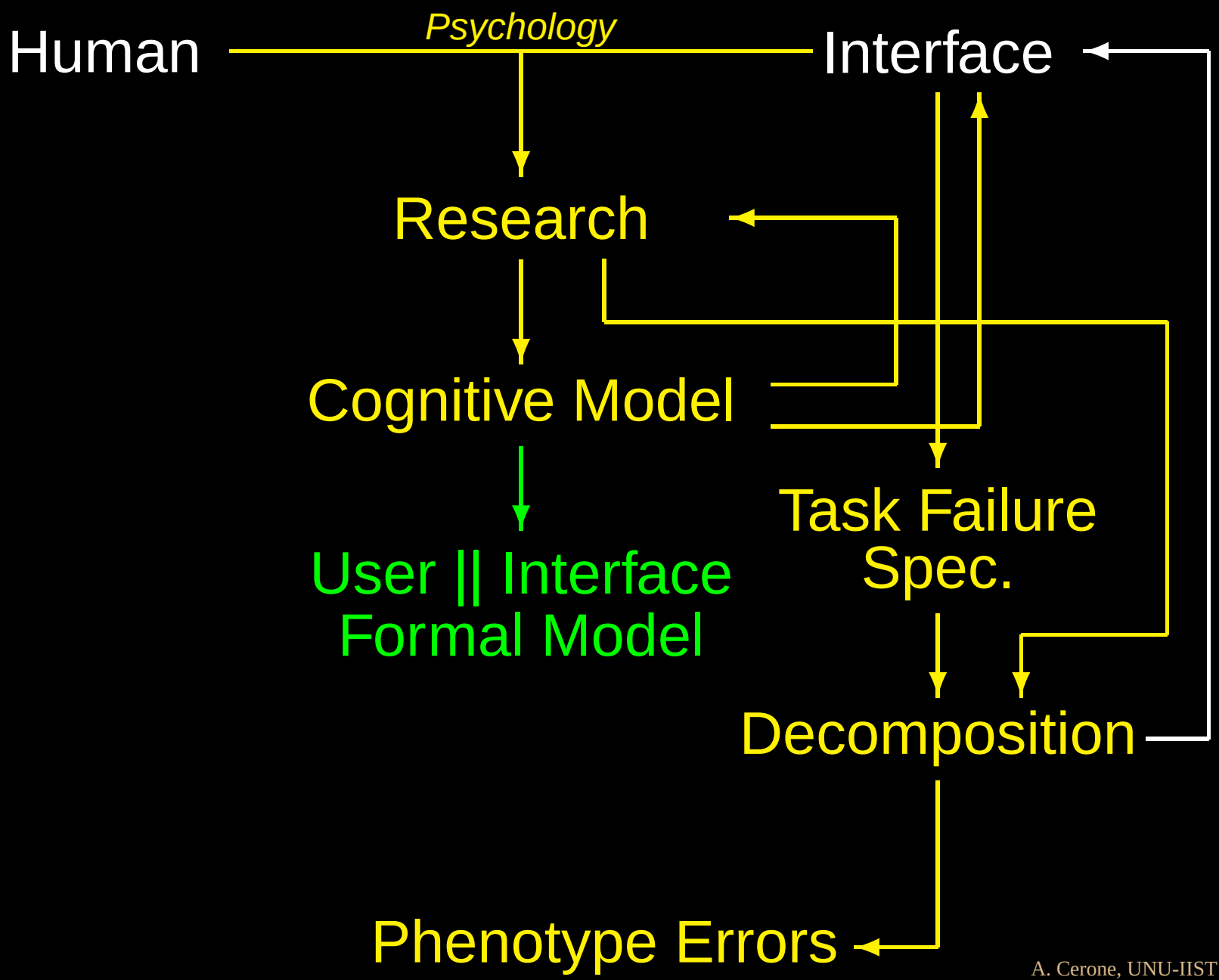
- define specific failures (phenotype errors) that cause the top-level task failure
- use phenotype errors to improve the interface
- give phenotype errors psychological interpretations (genotype errors)
- perform experiment to confirm causality genotype error $\Longrightarrow$ phenotype error
- use genotype errors to improve the cognitive model

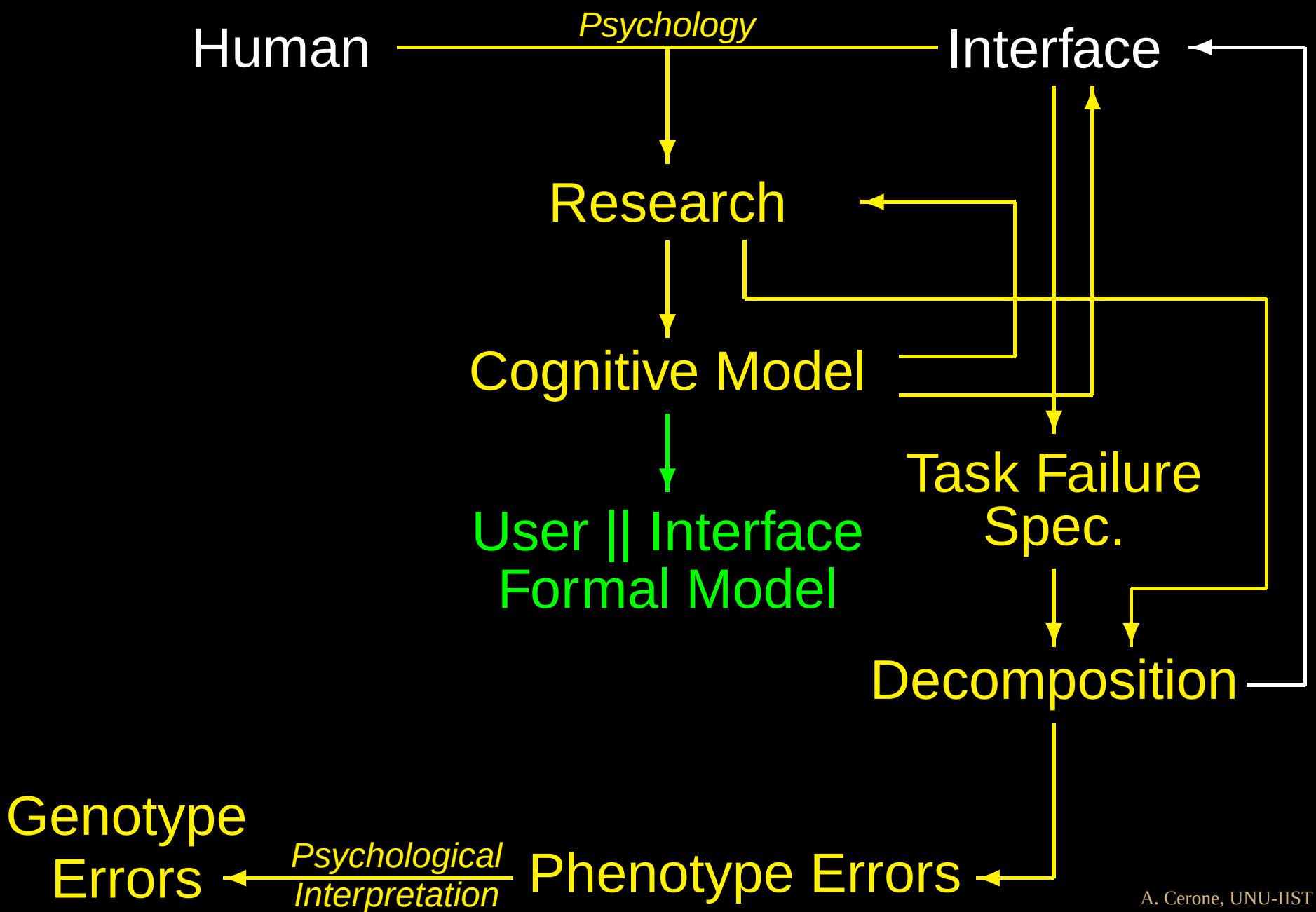

Human
Psychology
Interface
Research
Cognitive Model
Task
Spec.
User || Interface
Formal Model
Errors
Verification

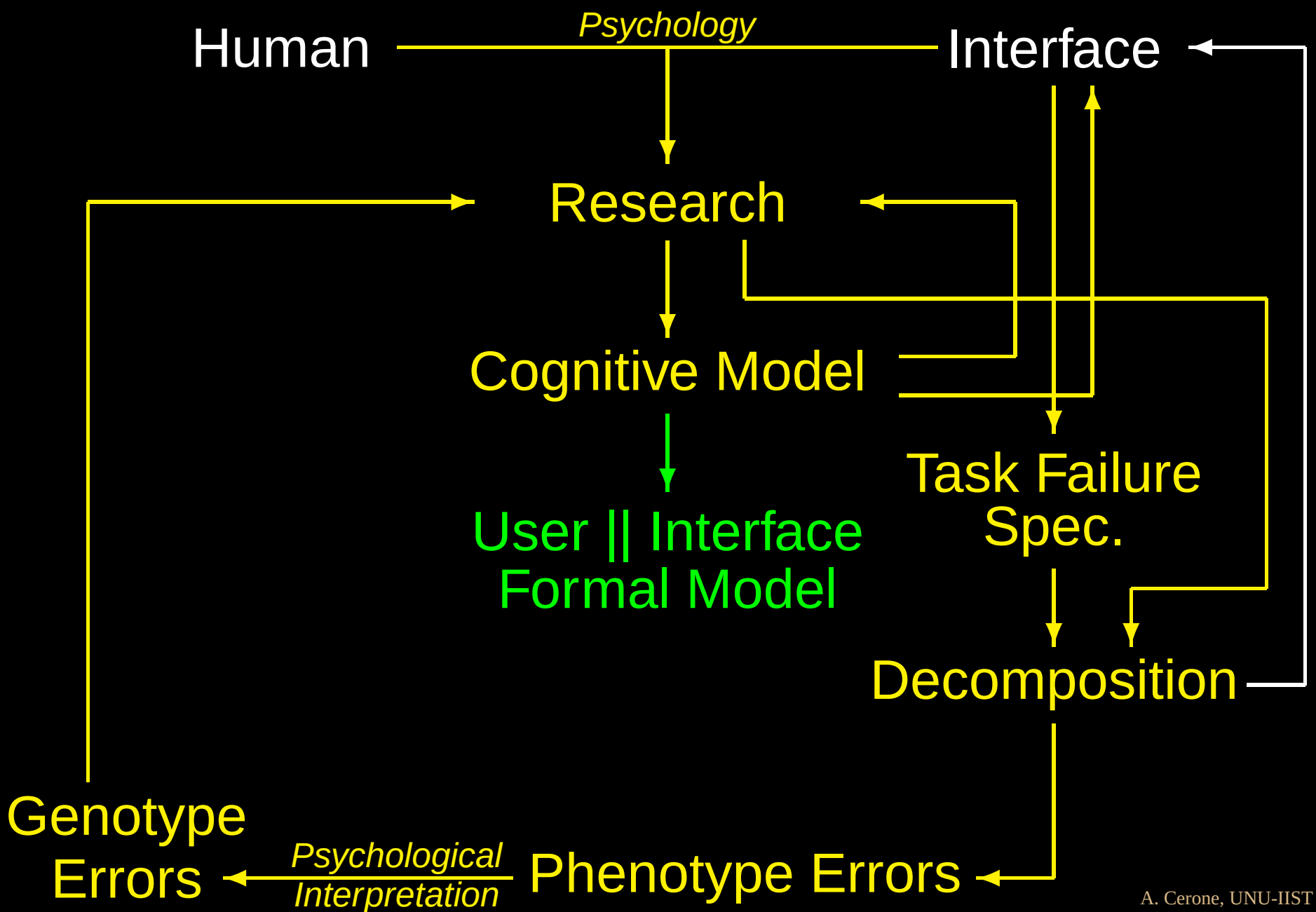

Human
Psychology
Interface
Research
Cognitive Model
Task Failure
Spec.
User || Interface
Formal Model

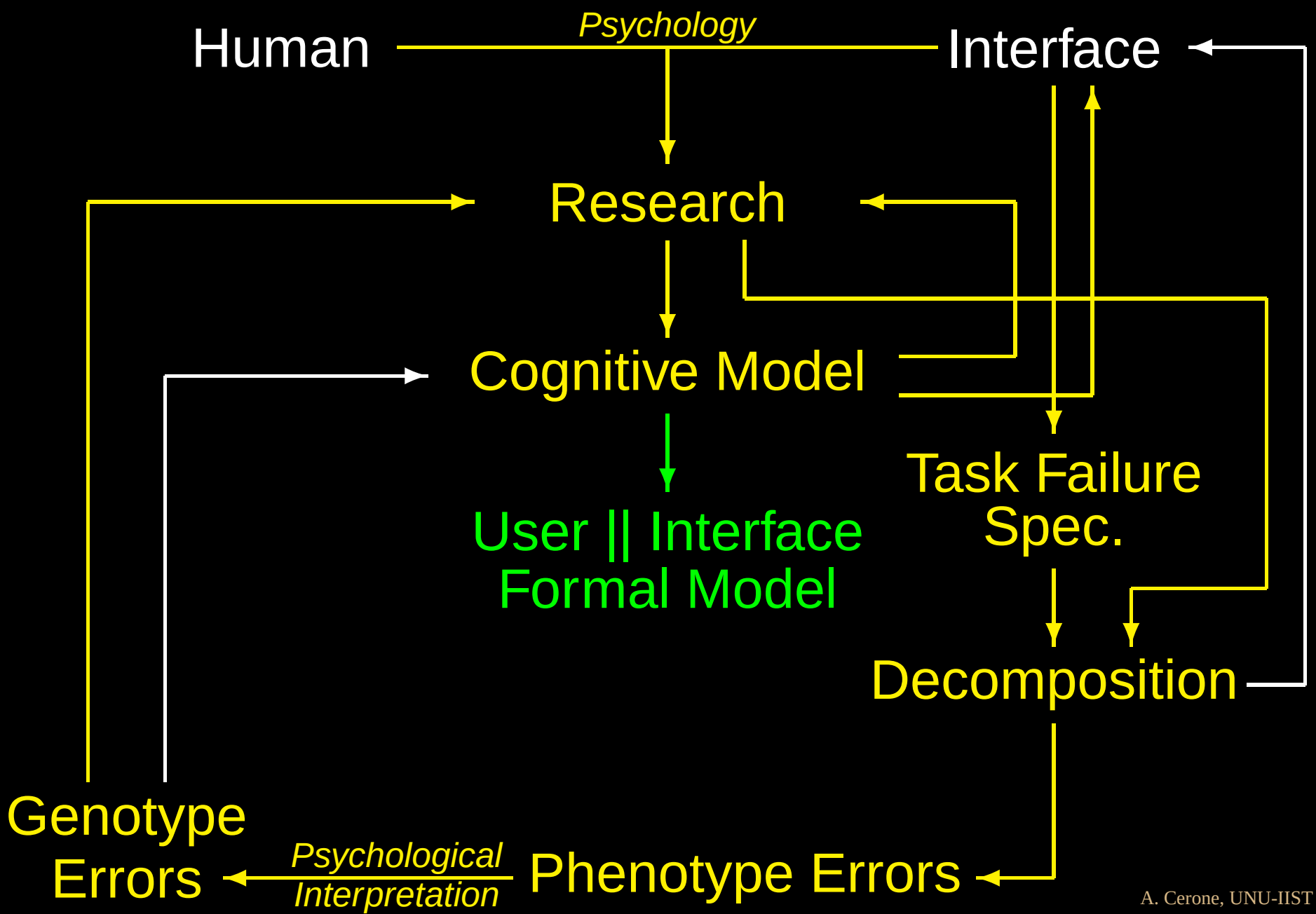

Human
Psychology
Interface
Research
Cognitive Model
User || Interface
Formal Model
Task Failure
Spec.
Decomposition

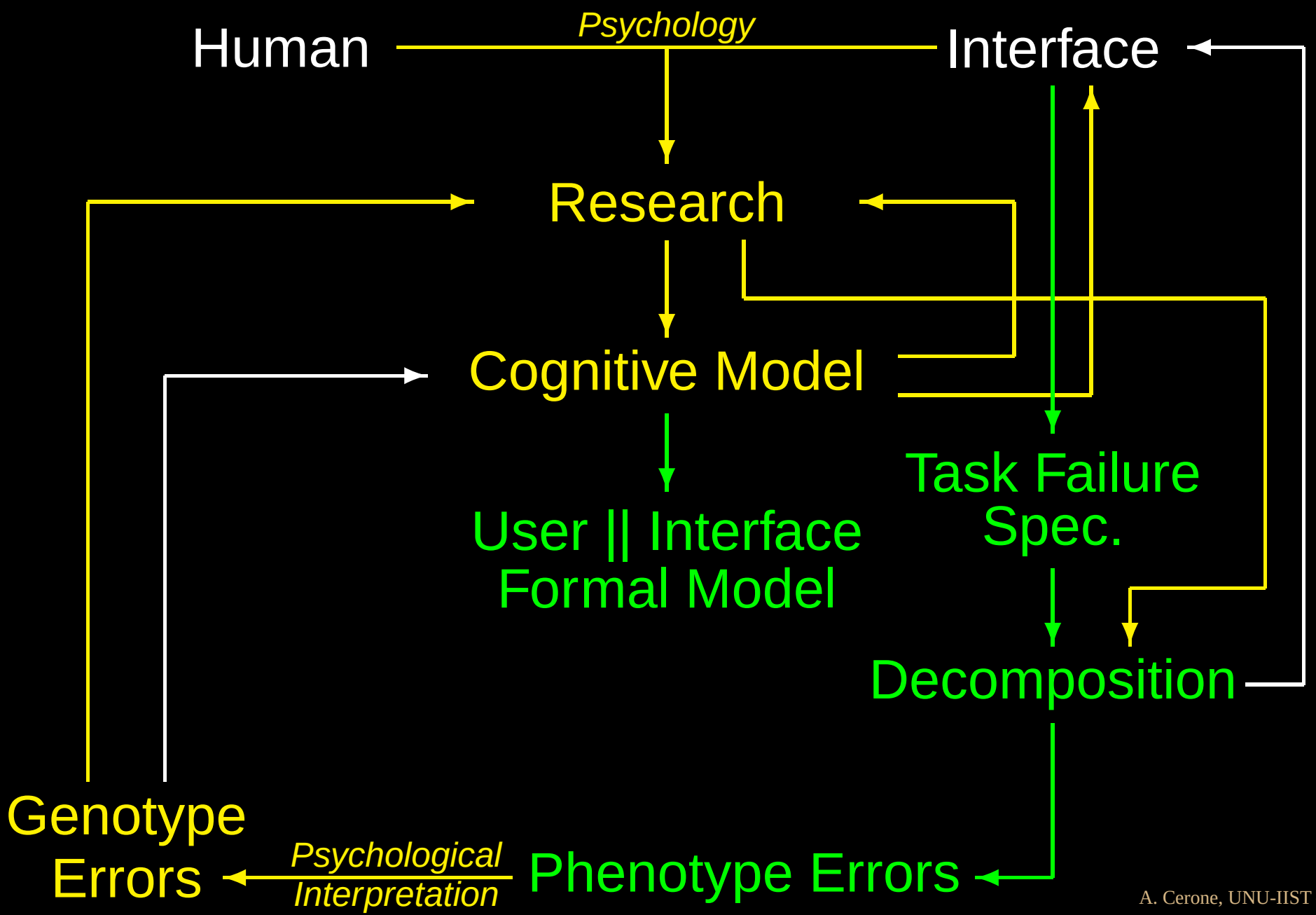

Human
Psychology
Interface
Research
Cognitive Model
User || Interface
Formal Model
Task Failure
Spec.
Decomposition

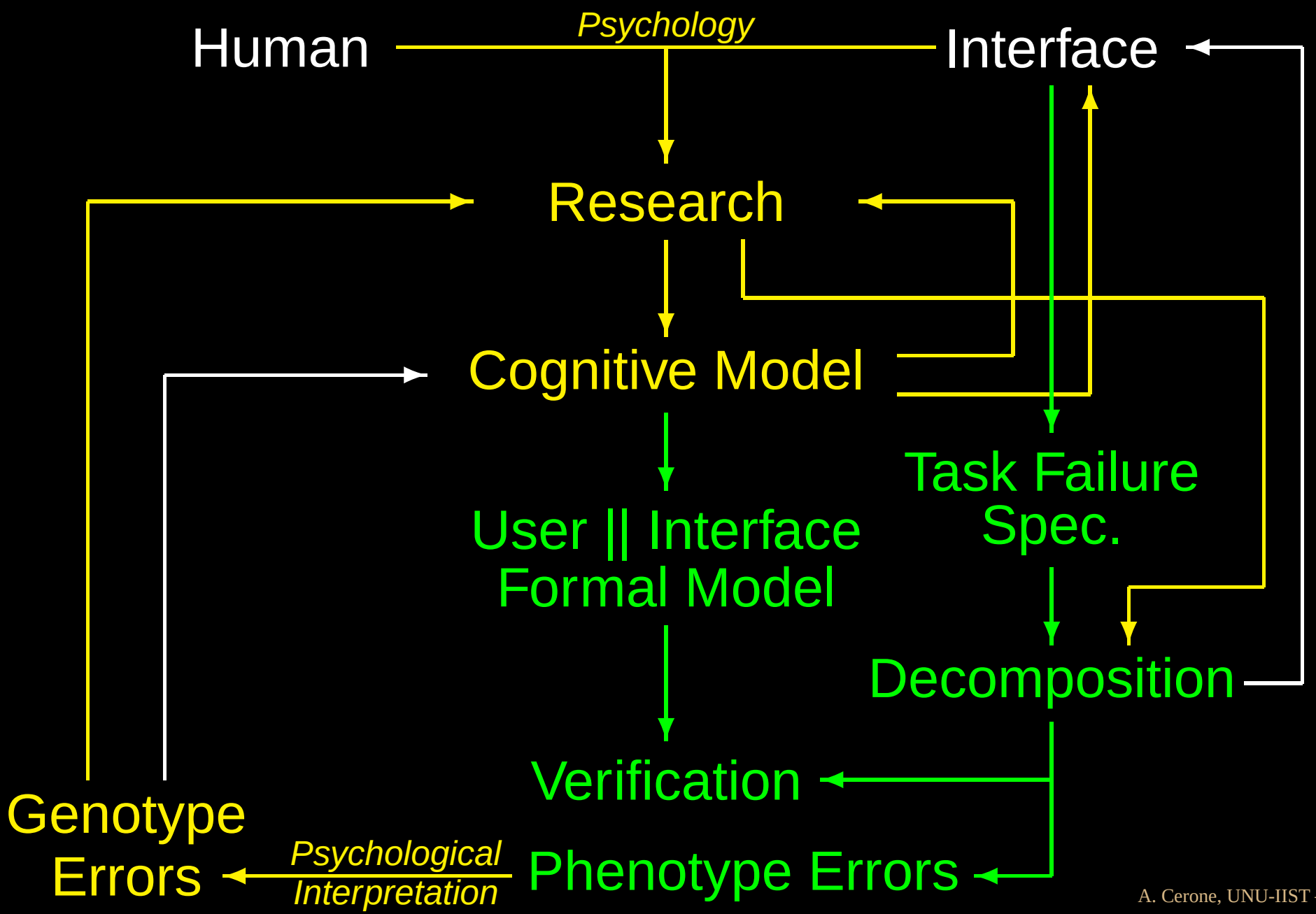

Human
Psychology
Interface
Research
Cognitive Model
User || Interface
Formal Model
Task Failure
Spec.
Decomposition
Phenotype Errors

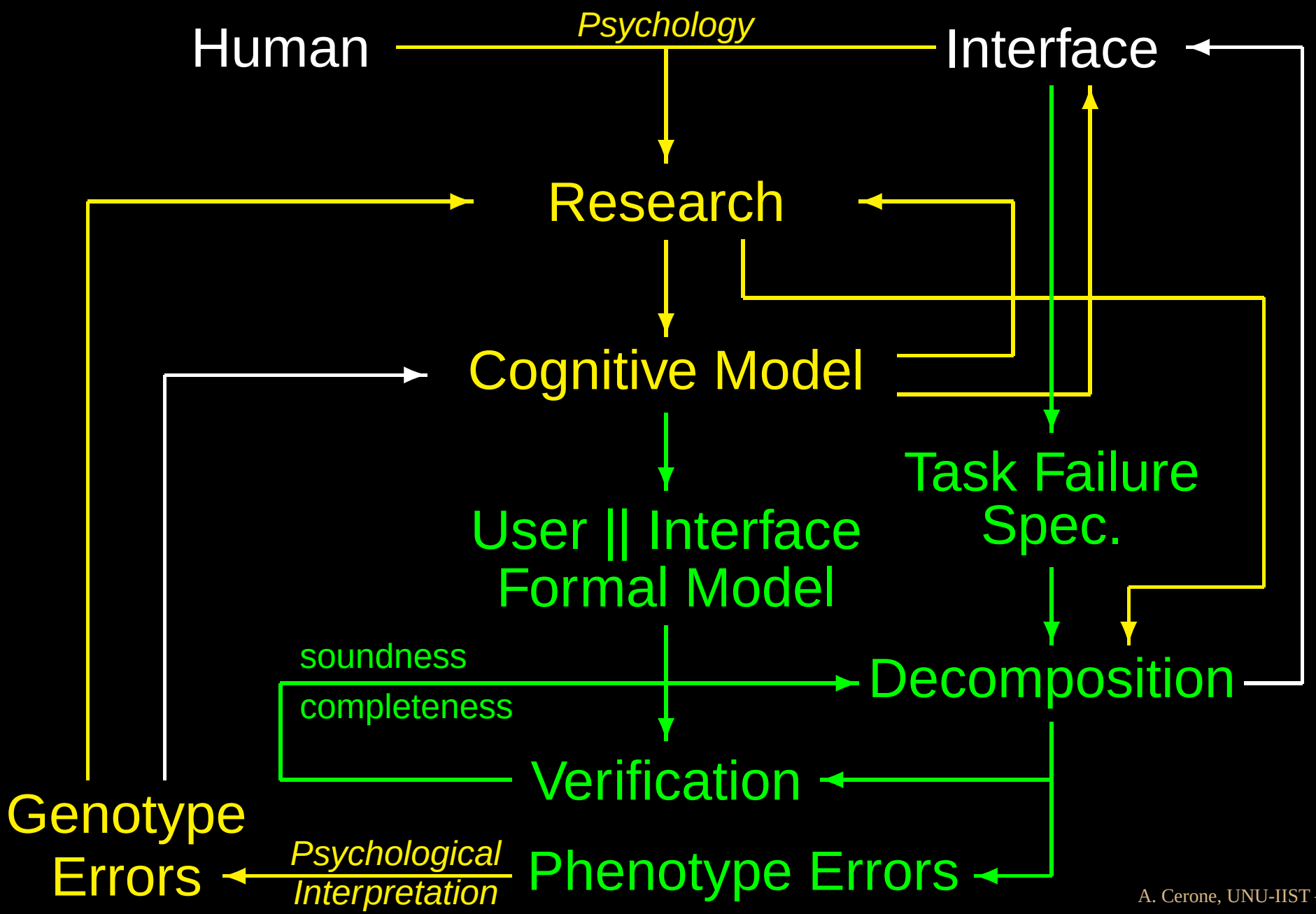

Human
Psychology
Interface
Research
Cognitive Model
User || Interface
Formal Model
Task Failure
Spec.
Decomposition
Genotype
Errors
Psychological
Interpretation
Phenotype Errors


Human
Psychology
Interface
Research
Cognitive Model
User || Interface
Formal Model
Task Failure
Spec.
Decomposition
Genotype
Errors
Psychological
Interpretation
Phenotype Errors


Human
Psychology
Interface
Research
Cognitive Model
User || Interface
Formal Model
Task Failure
Spec.
Decomposition
Genotype
Errors
Psychological
Interpretation
Phenotype Errors


Human
Psychology
Interface
Research
Cognitive Model
User || Interface
Formal Model
Task Failure
Spec.
Decomposition
Genotype
Errors
Psychological
Interpretation
Phenotype Errors


Human
Psychology
Interface
Research
Cognitive Model
Task Failure Spec.
User || Interface Formal Model
Decomposition
Verification
Genotype Errors
Psychological Interpretation
Phenotype Errors


Human
Psychology
Interface
Research
Cognitive Model
Task Failure Spec.
User || Interface Formal Model
soundness
completeness
Decomposition
Verification
Genotype Errors
Psychological Interpretation
Phenotype Errors

Operator Choice Model (OCM)

Operator Choice Model

Scanning: The operator searches the interface for a certain property.

Operator Choice Model

Scanning: The operator searches the interface for a certain property.

Identification: The operator identifies part of the interface that may represent the property.

Operator Choice Model

Scanning: The operator searches the interface for a certain property.

Identification: The operator identifies part of the interface that may represent the property.

Classification: The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

Operator Choice Model

Scanning: The operator searches the interface for a certain property.

Identification: The operator identifies part of the interface that may represent the property.

Classification: The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

Decision on how to resolve the situation.

Operator Choice Model

Scanning: The operator searches the interface for a certain property.

Identification: The operator identifies part of the interface that may represent the property.

Classification: The operator

- assesses whether the property is in need of further interest;
- if so, gives some form of priority to the property.

Decision on how to resolve the situation.

Action to be performed as a series of interaction with the interface.

OCM for Nuclear Plant

Scanning: The operator scans among each of the individual reactor readouts on the interface searching for any anomalies.

Identification: The operator identifies a particular readout.

Classification: The operator

- assesses whether the identified readout describes a normal or abnormal operation of the plant;
- if abnormal, gives a priority to the operation according to its urgency to be resolved.

Decision on how to resolve the abnormal situation.

Action to be performed as a series of interaction with the interface and with internal and/or external authorities.

OCM for Air Traffic Control

Scanning: The operator scans among each pair of aircraft searching for a pair that may violate separation.

Identification: The operator identifies a pair of aircraft.

Classification: The operator

- assesses whether the identified pair of aircraft will eventually violate separation (in conflict) or not (not in conflict);
- if so, gives a priority to the conflict according to its urgency to be resolved.

Decision on how to resolve the conflict.

Action to be performed as a series of interaction with the interface.

Air Traffic Control (ATC) System Example

Air Traffic Control (ATC)

- Aircraft fly along straight-line segments — called *flight paths* — between *waypoints* within a fixed sector of airspace.

Air Traffic Control (ATC)

- Aircraft fly along straight-line segments — called *flight paths* — between *waypoints* within a fixed sector of airspace.
- Aircraft *horizontal separation* must be at least 5 miles.

Air Traffic Control (ATC)

- Aircraft fly along straight-line segments — called *flight paths* — between *waypoints* within a fixed sector of airspace.
- Aircraft *horizontal separation* must be at least 5 miles.
- A *pair* of aircraft *violate separation* when the horizontal distance between them is less than 5 miles (*separation violation*).

Air Traffic Control (ATC)

- Aircraft fly along straight-line segments — called *flight paths* — between *waypoints* within a fixed sector of airspace.
- Aircraft *horizontal separation* must be at least 5 miles.
- A *pair* of aircraft *violate separation* when the horizontal distance between them is less than 5 miles (*separation violation*).
- A *pair* of aircraft is in *conflict* when their pathways are such that the two aircraft will eventually violate separation.

ATC Simulator

- The ATC operator's task involves monitoring the movement of aircraft on a screen, looking for pair of aircraft that *may violate separation*.

ATC Simulator

- The ATC operator's task involves monitoring the movement of aircraft on a screen, looking for pair of aircraft that *may violate separation*.
- When such a conflict is detected, the operator uses a mouse to select one of the aircraft and change its speed using a pulldown menu.

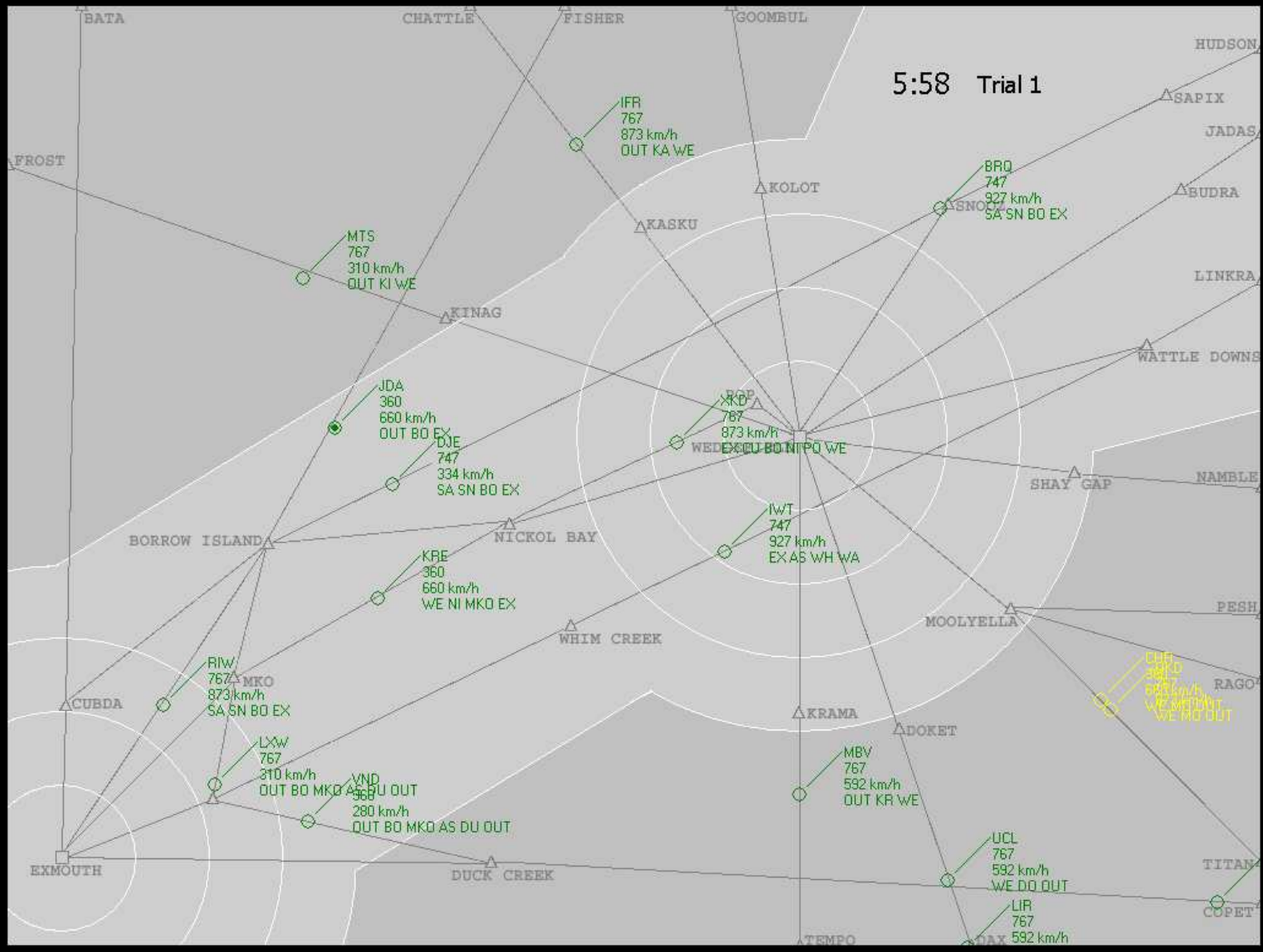
ATC Simulator

- The ATC operator's task involves monitoring the movement of aircraft on a screen, looking for pair of aircraft that *may violate separation*.
- When such a conflict is detected, the operator uses a mouse to select one of the aircraft and change its speed using a pulldown menu.
- The goal of the task is to resolve all conflicts before they violate separation, while not introducing any new conflict.

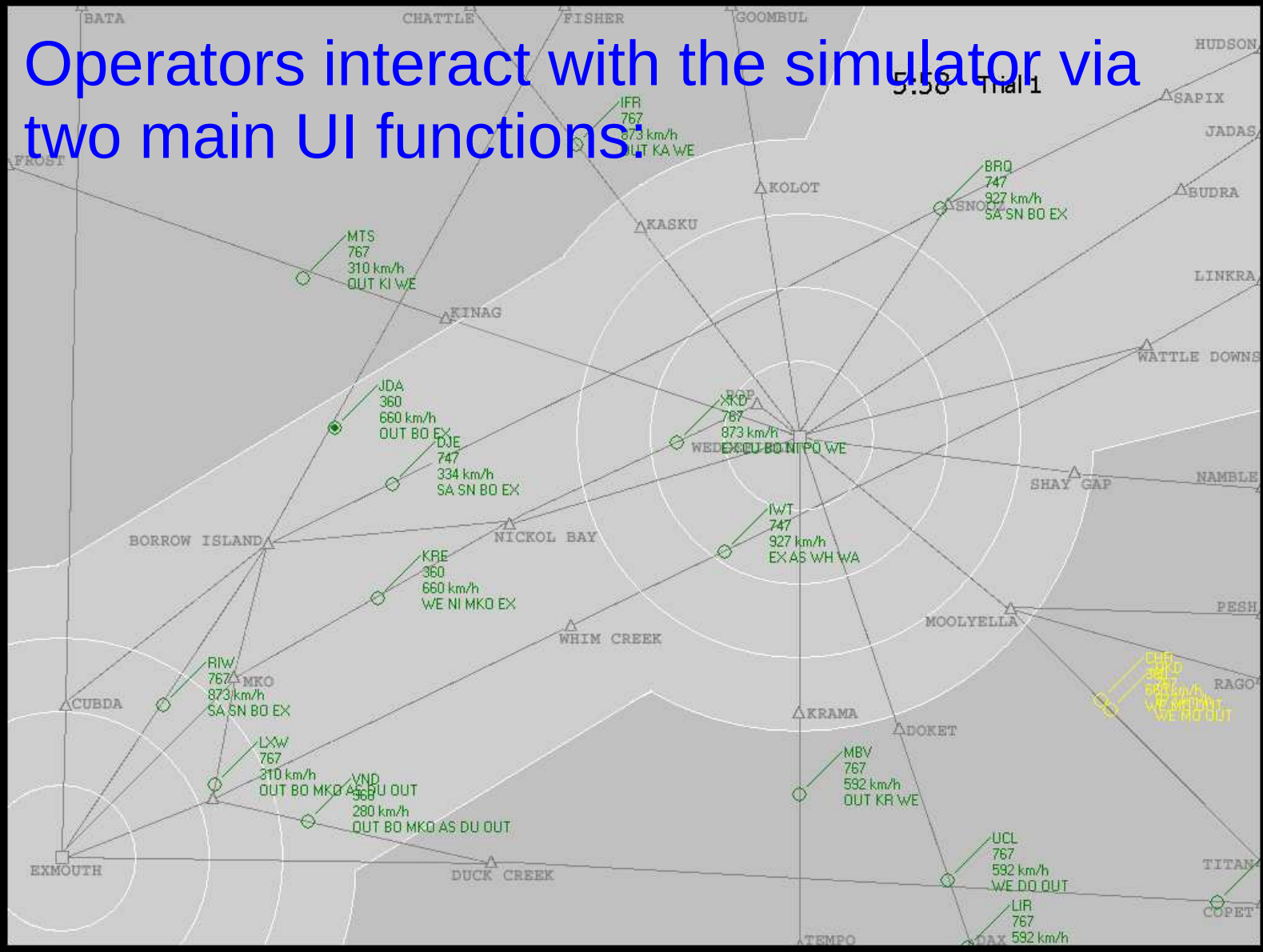
ATC Simulator

- The ATC operator's task involves monitoring the movement of aircraft on a screen, looking for pair of aircraft that *may violate separation*.
- When such a conflict is detected, the operator uses a mouse to select one of the aircraft and change its speed using a pulldown menu.
- The goal of the task is to resolve all conflicts before they violate separation, while not introducing any new conflict.
- We have a task failure when separation is violated.

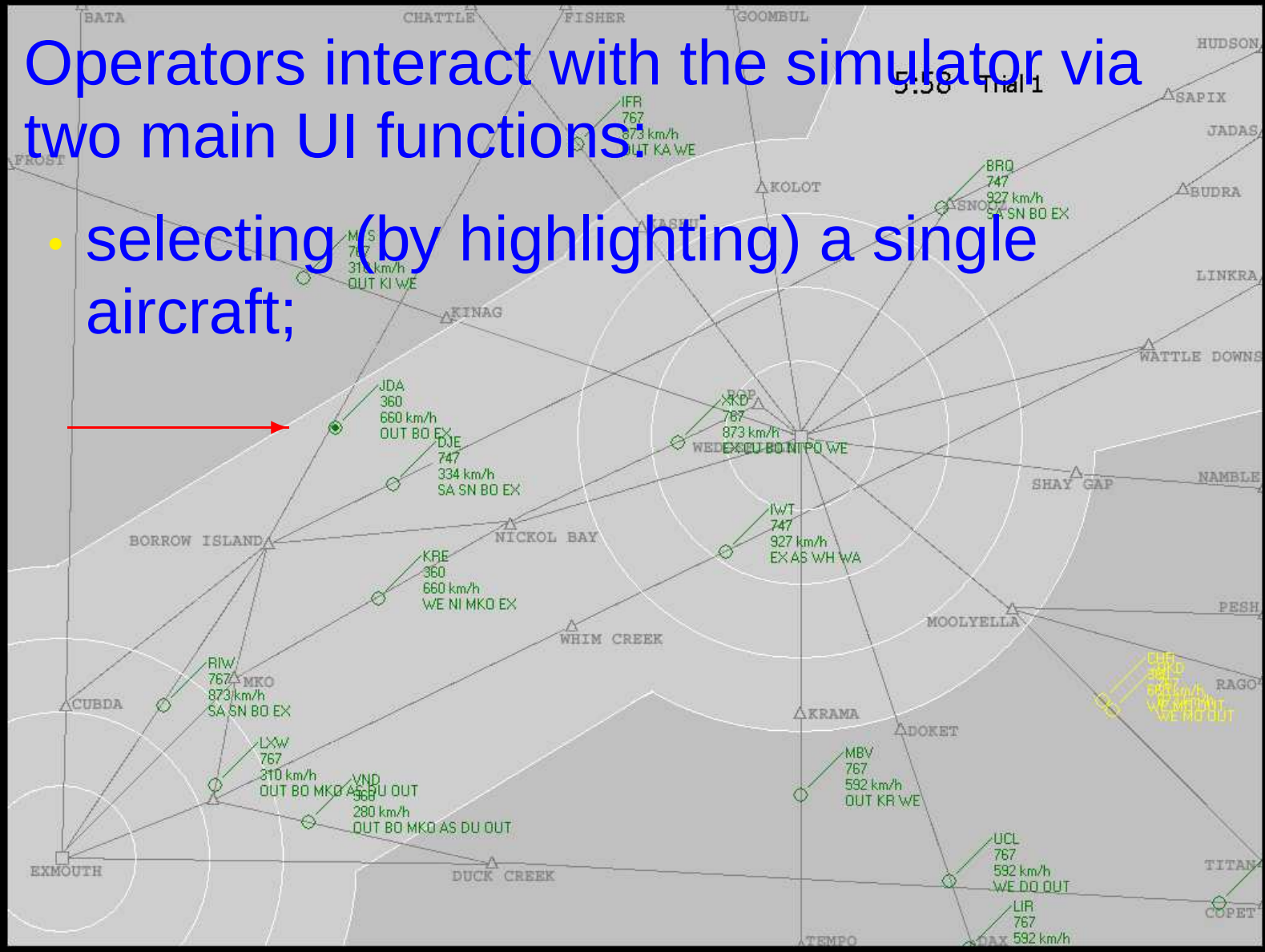
ATC Simulator Screenshot



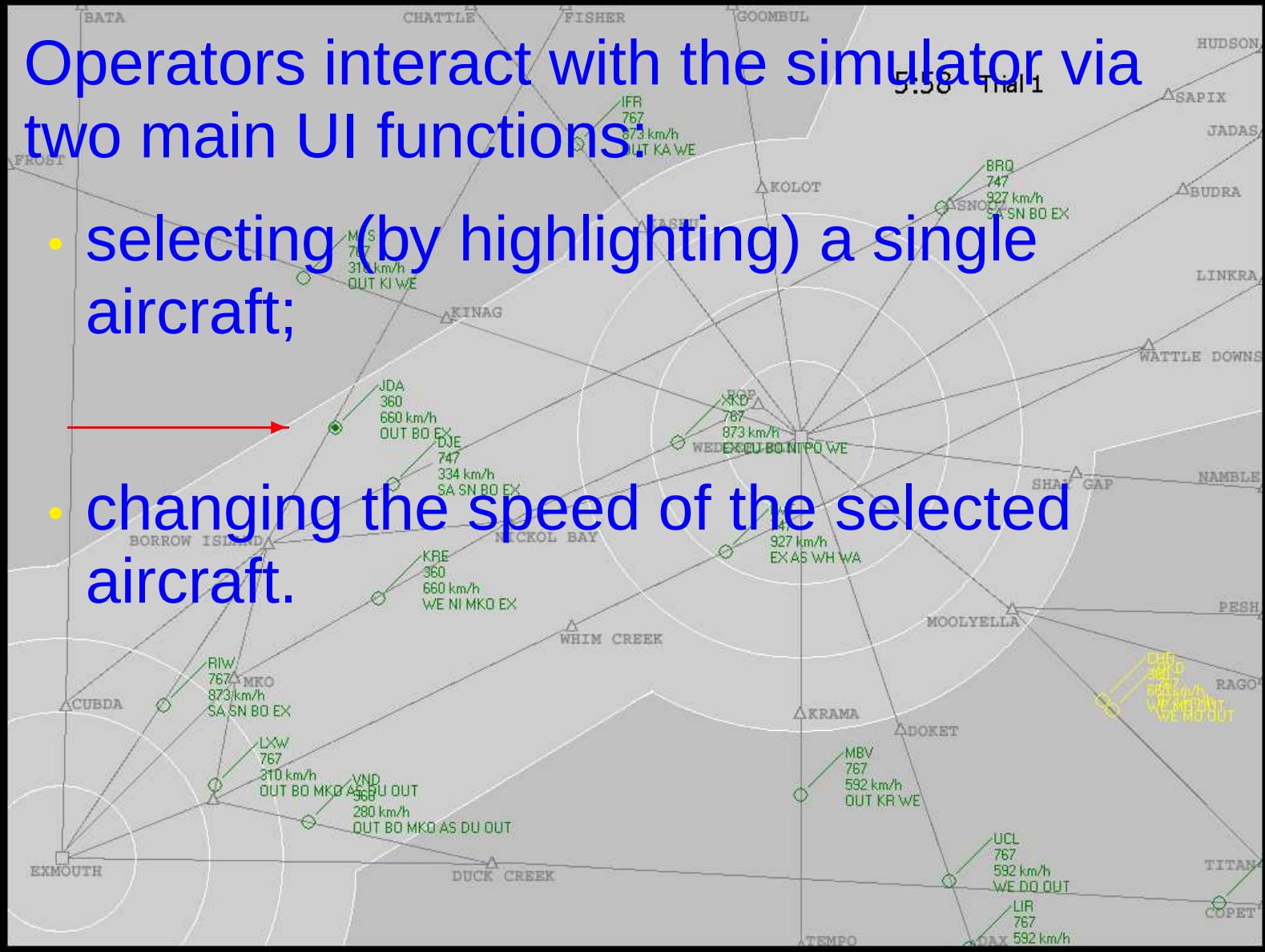
ATC Simulator Screenshot



ATC Simulator Screenshot



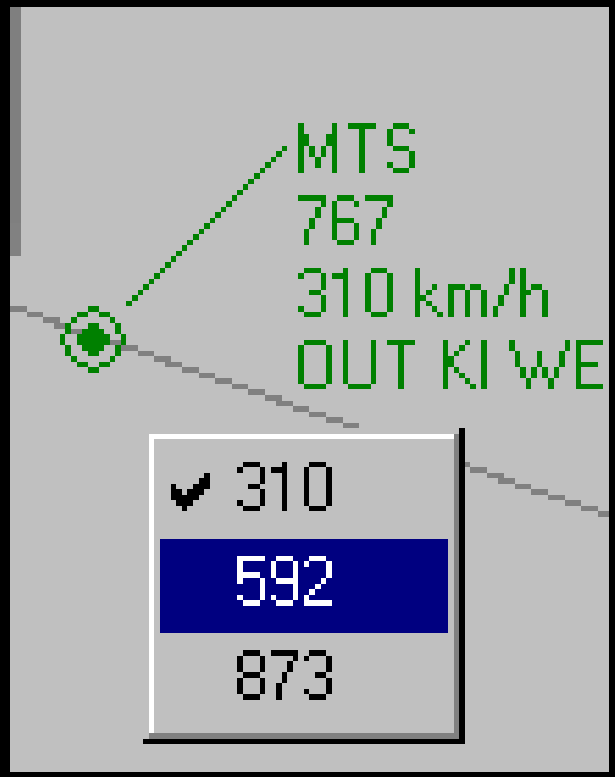
ATC Simulator Screenshot



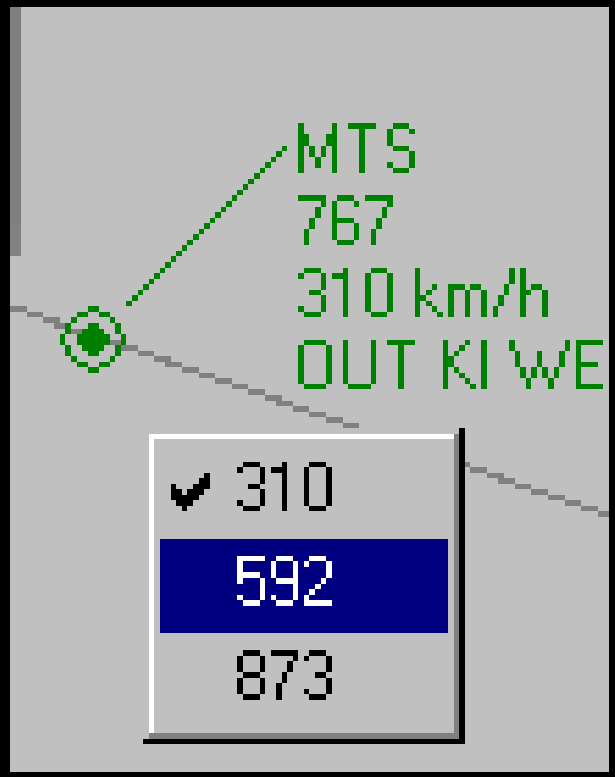
Operators interact with the simulator via two main UI functions:

- selecting (by highlighting) a single aircraft;
- changing the speed of the selected aircraft.

Speed Menu

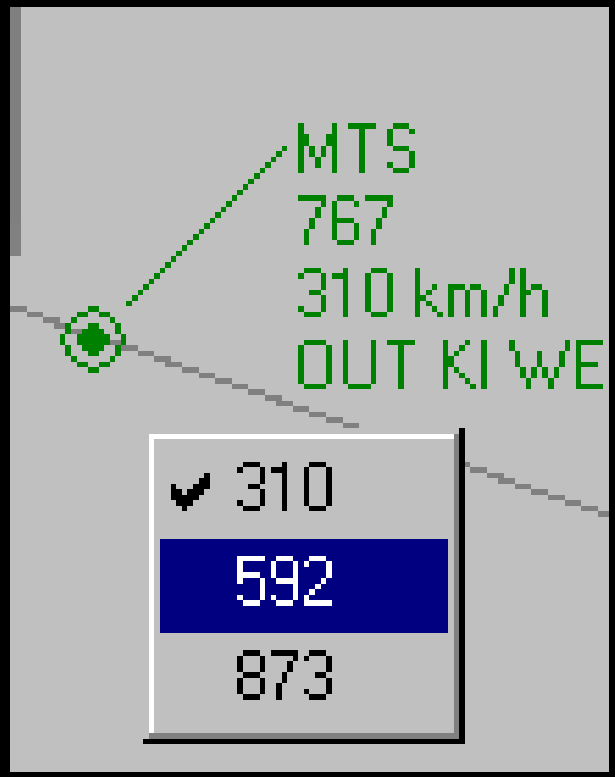


Speed Menu



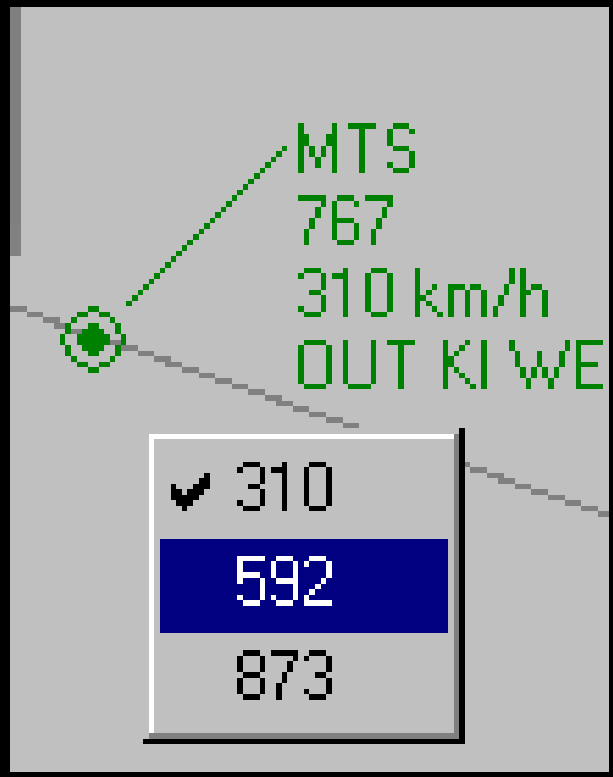
Open the menu by clicking the right button.

Speed Menu



Open the menu by clicking the right button.
The menu appears at the position of the cursor.

Speed Menu



Open the menu by clicking the right button.
The menu appears at the position of the cursor.
Selected the speed by left clicking on the desired menu entry.

Operator Erros

- slip: inadvertedly select a wrong or the current speed

Operator Erros

- slip: inadvertedly select a wrong or the current speed
  $\Longleftarrow$ selction task closure (cognitive problem)

Operator Erros

- slip: inadvertedly select a wrong or the current speed
  $\Longleftarrow$ selction task closure (cognitive problem)
- mistaken identity: change the spead of an aircraft different from the intended one

Operator Erros

- slip: inadvertedly select a wrong or the current speed
  $\Longleftarrow$ selction task closure (cognitive problem)
- mistaken identity: change the spead of an aircraft different from the intended one
  $\Longleftarrow$ the menu appears at the position of the cursor (usability problem)

Operator Erros

- slip: inadvertedly select a wrong or the current speed
  $\Longleftarrow$ selction task closure (cognitive problem)
- mistaken identity: change the spead of an aircraft different from the intended one
  $\Longleftarrow$ the menu appears at the position of the cursor (usability problem)
- mis-classification, mis-prioritization, conflict generation

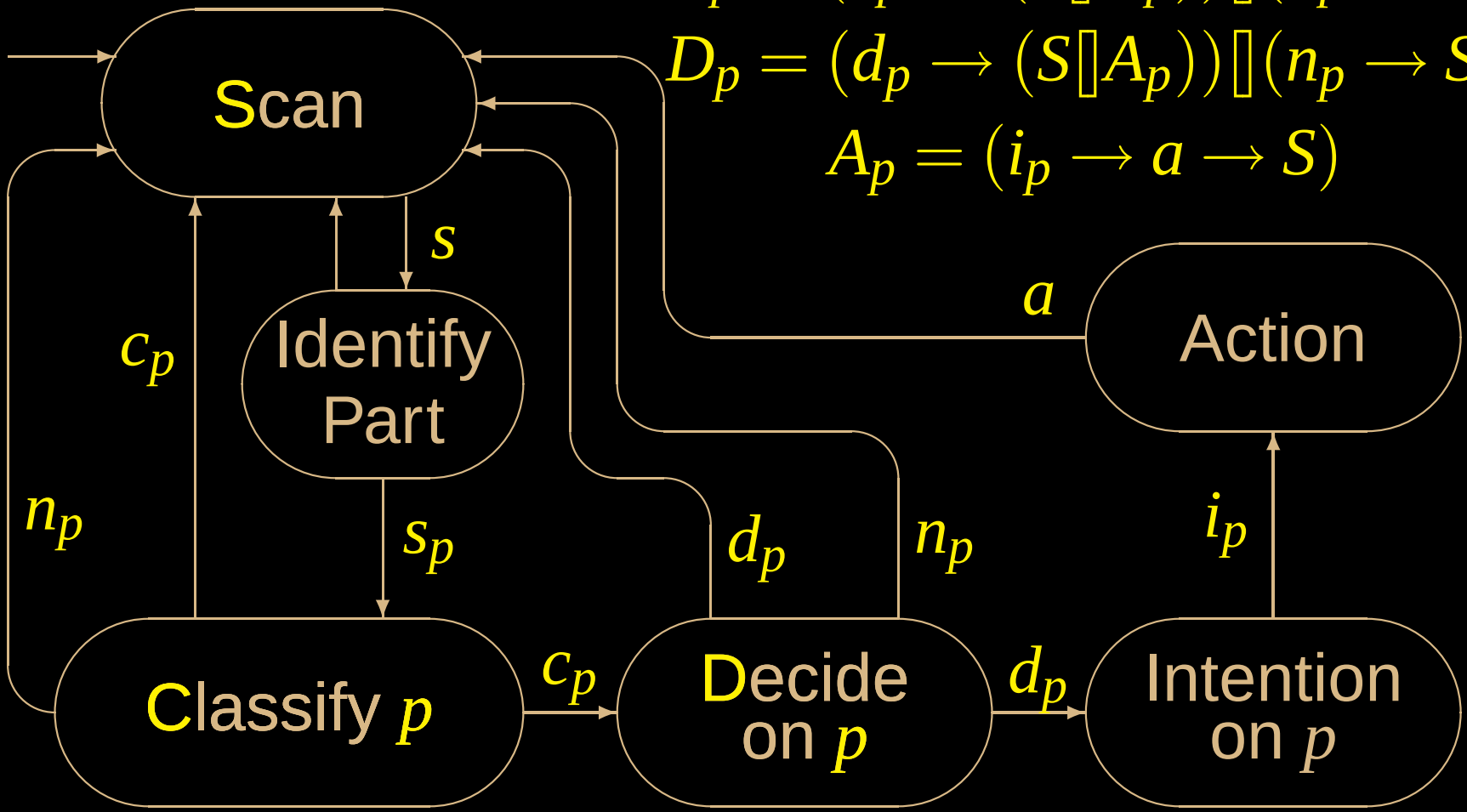
Operator Erros

- slip: inadvertedly select a wrong or the current speed
  $\Longleftarrow$ selction task closure (cognitive problem)
- mistaken identity: change the spead of an aircraft different from the intended one
  $\Longleftarrow$ the menu appears at the position of the cursor (usability problem)
- mis-classification, mis-prioritization, conflict generation

The operator can recover from these errors without causing separation violation (task failure)

Model Interpretation for ATC

$$S = s \rightarrow (([]_{p:Part}(s_p \rightarrow C_p)) [] S)$$
$$C_p = (c_p \rightarrow (S [] D_p)) [] (n_p \rightarrow S)$$
$$D_p = (d_p \rightarrow (S [] A_p)) [] (n_p \rightarrow S)$$
$$A_p = (i_p \rightarrow a \rightarrow S)$$



Model Interpretation for ATC

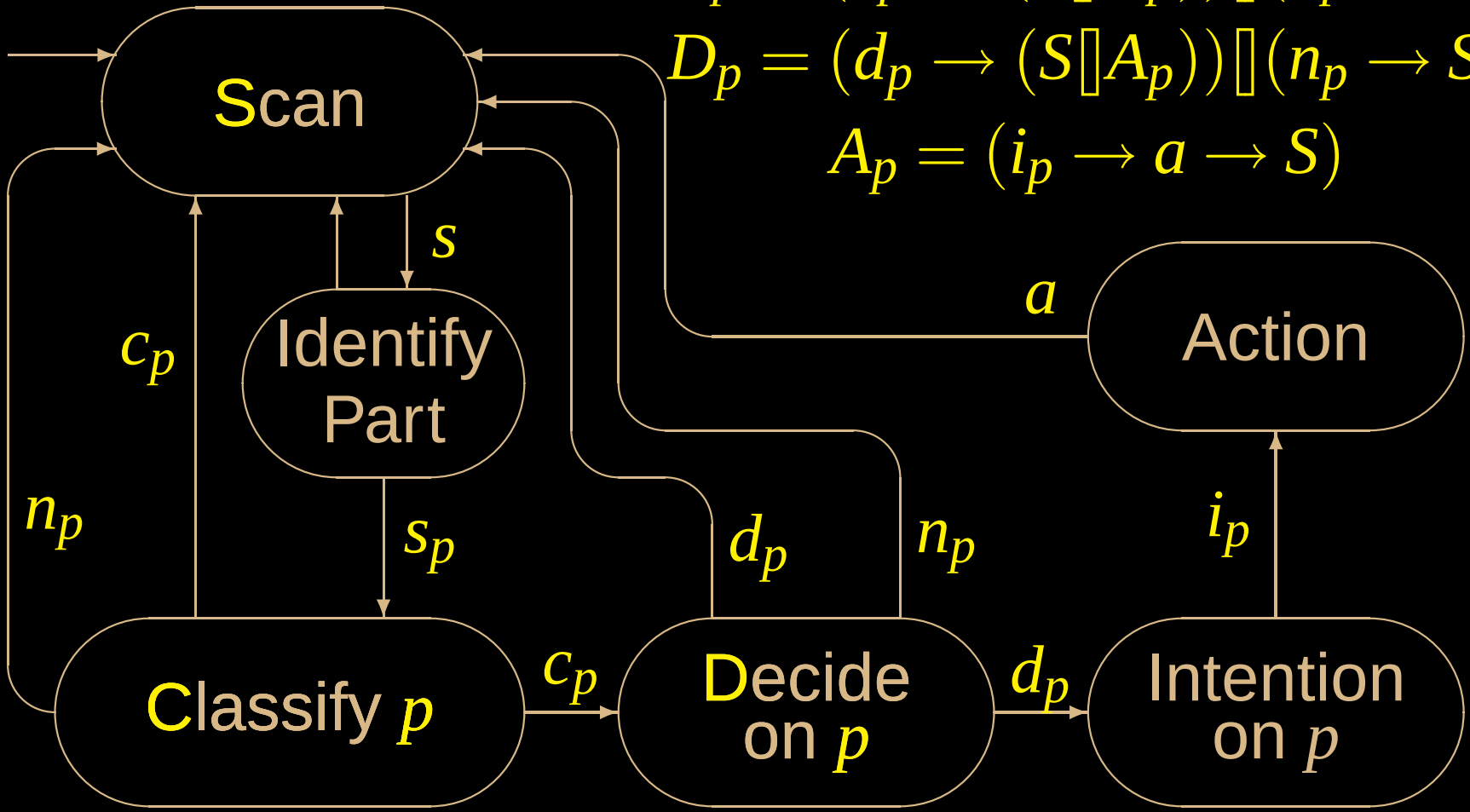
$p =$ Pair of aircraft

$$S = s \rightarrow (([]_{p:Pairs}(s_p \rightarrow C_p))[]S)$$

$$C_p = (c_p \rightarrow (S[]D_p))[](n_p \rightarrow S)$$

$$D_p = (d_p \rightarrow (S[]A_p))[](n_p \rightarrow S)$$

$$A_p = (i_p \rightarrow a \rightarrow S)$$

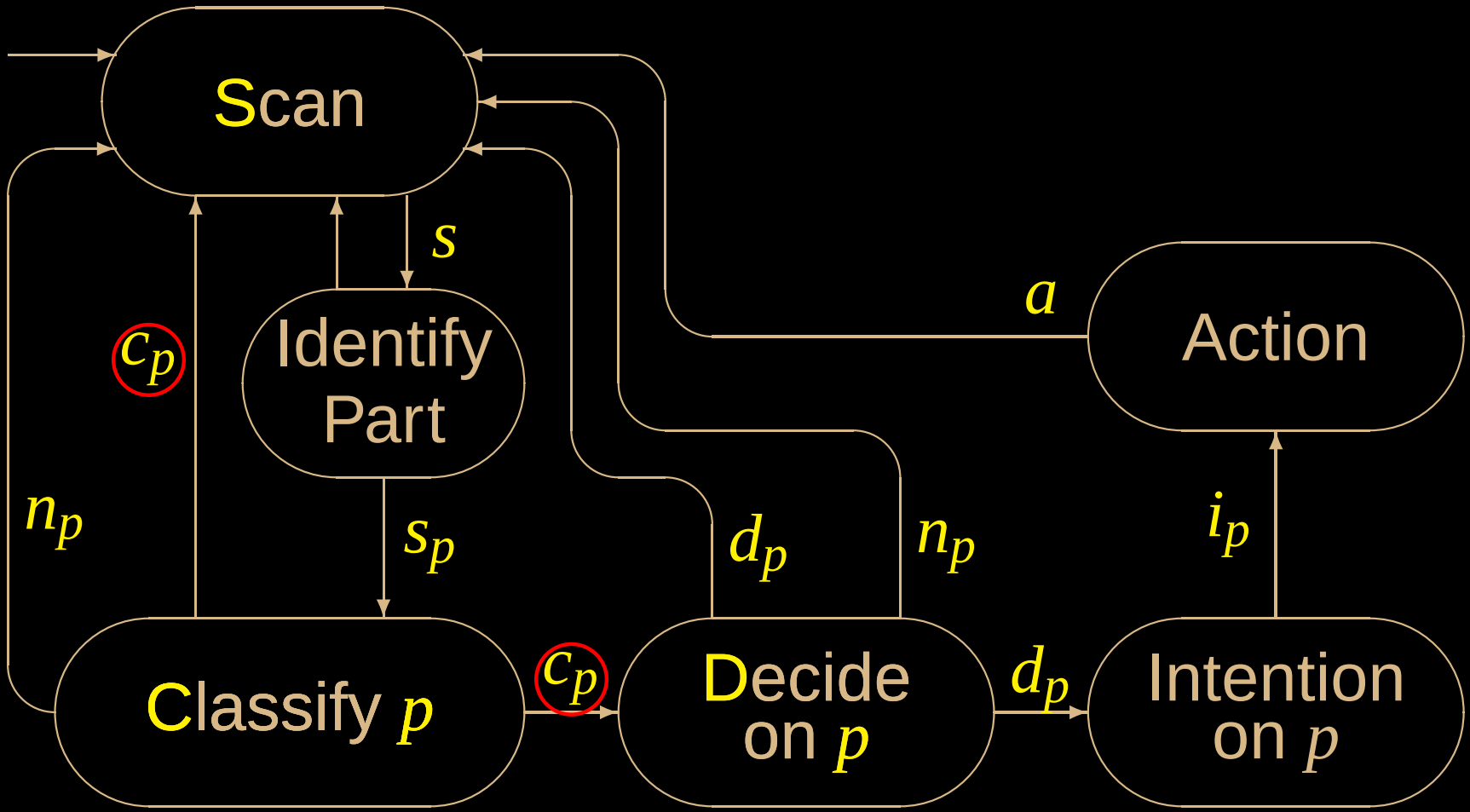


Model Interpretation for ATC

$p =$ Pair of aircraft

The pair is classified

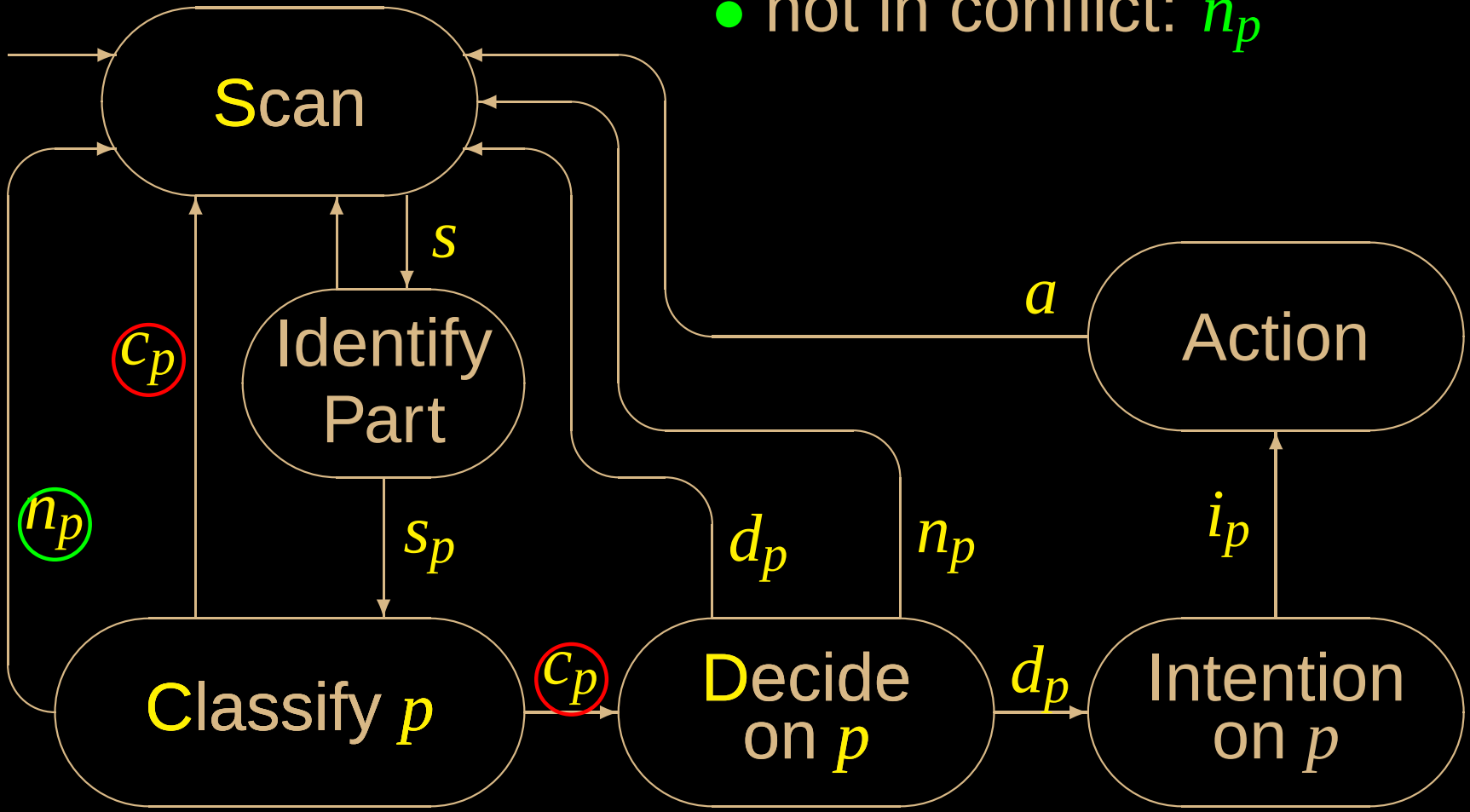
- in conflict: c_p



Model Interpretation for ATC

$p =$ Pair of aircraft

The pair is classified
- in conflict: c_p
- not in conflict: n_p

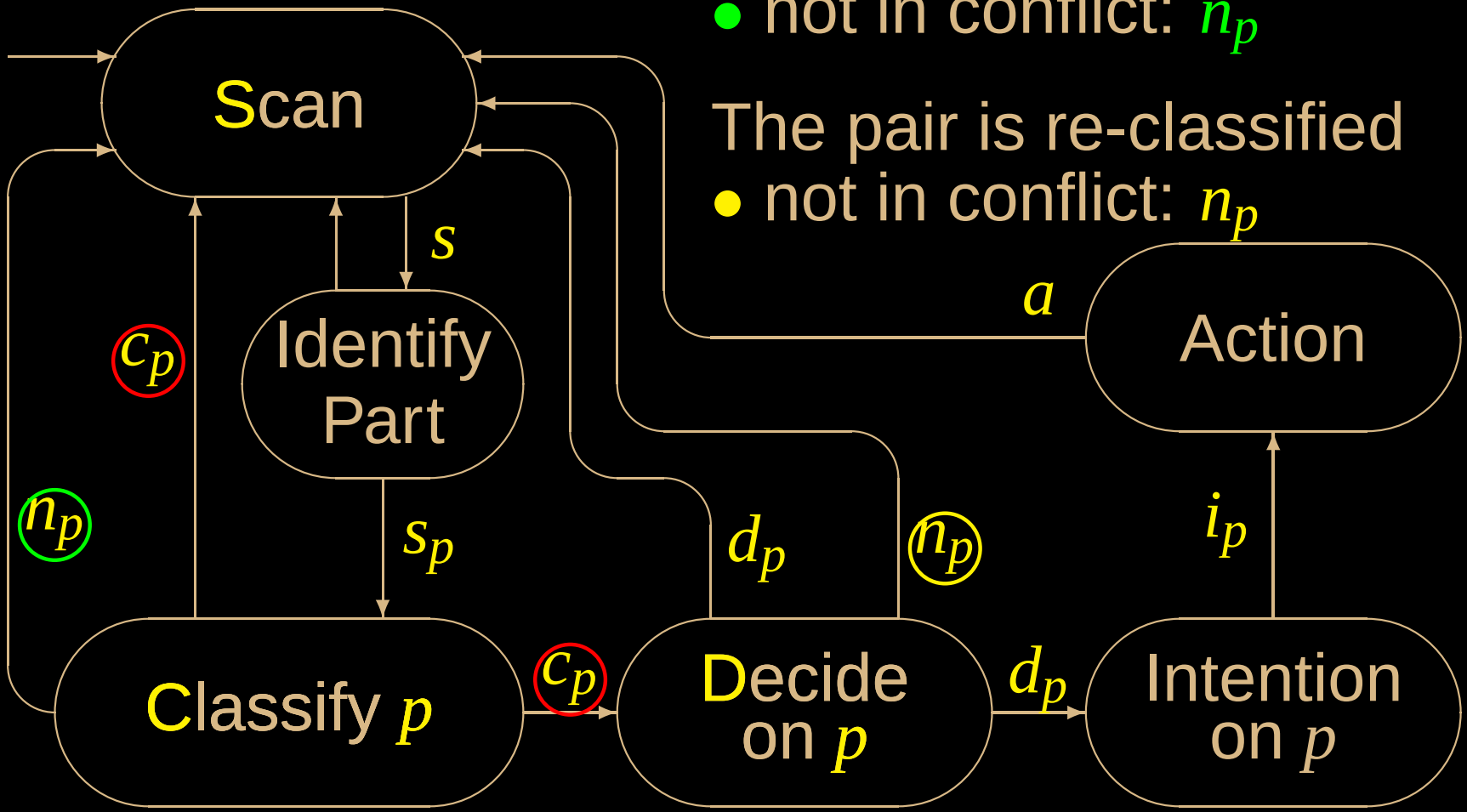


Model Interpretation for ATC

$p =$ Pair of aircraft

The pair is classified
- in conflict: c_p
- not in conflict: n_p

The pair is re-classified
- not in conflict: n_p



OCM for Air Traffic Control

Scanning: The operator scans among each pair of aircraft searching for a pair that may violate separation.

Identification: The operator identifies a pair of aircraft.

Classification: The operator

- assesses whether the identified pair of aircraft will eventually violate separation (in conflict) or not (not in conflict);
- if so, gives a priority to the conflict according to its urgency to be resolved.

Decision on how to resolve the conflict.

Action to be performed as a series of interaction with the interface.

Environment Model

$$
\begin{aligned}
S &= s \to ((\Box_{p:Pairs}(s_p \to C_p)) \Box S) \\
C_p &= (c_p \to (S \Box D_p)) \Box (n_p \to S) \\
D_p &= (d_p \to (S \Box A_p)) \Box (n_p \to S) \\
A_p &= (i_p \to a \to S)
\end{aligned}
$$

Environment Model

$$
\begin{aligned}
S &= s \rightarrow (([]_{p:Pairs}(s_p \rightarrow C_p)) [] S) \\
C_p &= (c_p \rightarrow (S [] D_p)) [] (n_p \rightarrow S) \\
D_p &= (d_p \rightarrow (S [] A_p)) [] (n_p \rightarrow S) \\
A_p &= (i_p \rightarrow a \rightarrow S)
\end{aligned}
$$

$$
\begin{aligned}
I_p = s \rightarrow a \rightarrow (&(unresolved_p \rightarrow I_p) [] \\
&(resolved_p \rightarrow N_p) [] \\
&(noeffect_p \rightarrow I_p))
\end{aligned}
$$

Environment Model

$$
\begin{aligned}
S &= s \rightarrow ((\Box_{p:Pairs}(s_p \rightarrow C_p)) \Box S) \\
C_p &= (c_p \rightarrow (S \Box D_p)) \Box (n_p \rightarrow S) \\
D_p &= (d_p \rightarrow (S \Box A_p)) \Box (n_p \rightarrow S) \\
A_p &= (i_p \rightarrow a \rightarrow S)
\end{aligned}
$$

$$
\begin{aligned}
I_p = s \rightarrow a \rightarrow ((&unresolved_p \rightarrow I_p) \Box \\
&(resolved_p \rightarrow N_p) \Box \\
&(noeffect_p \rightarrow I_p))
\end{aligned}
$$

$$
\begin{aligned}
N_p = s \rightarrow a \rightarrow ((&unnecessary_p \rightarrow N_p) \Box \\
&(adverse_p \rightarrow I_p) \Box \\
&(noeffect_p \rightarrow N_p))
\end{aligned}
$$

Environment Model

$$
\begin{aligned}
S &= s \rightarrow ((\Box_{p:Pairs}(s_p \rightarrow C_p)) \Box S) \\
C_p &= (c_p \rightarrow (S \Box D_p)) \Box (n_p \rightarrow S) \\
D_p &= (d_p \rightarrow (S \Box A_p)) \Box (n_p \rightarrow S) \\
A_p &= (i_p \rightarrow a \rightarrow S)
\end{aligned}
$$

$$
\begin{aligned}
I_p = s \rightarrow a \rightarrow ((&unresolved_p \rightarrow I_p) \Box \\
&(resolved_p \rightarrow N_p) \Box \\
&(noeffect_p \rightarrow I_p))
\end{aligned}
$$

$$
\begin{aligned}
N_p = s \rightarrow a \rightarrow ((&unnecessary_p \rightarrow N_p) \Box \\
&(adverse_p \rightarrow I_p) \Box \\
&(noeffect_p \rightarrow N_p))
\end{aligned}
$$

$$
OCM = S \parallel (\parallel_{p:Init_I} I_p) \parallel (\parallel_{p:Init_N} N_p)
$$

Task Failure Analysis

Three levels of decomposition of task failures.

Task Failure Analysis

Three levels of decomposition of task failures.

A first decomposition of task failures is based on

- the intention of the operator to resolve a conflict (i_p);

Task Failure Analysis

Three levels of decomposition of task failures.

A first decomposition of task failures is based on

- the intention of the operator to resolve a conflict (i_p);

and on the result, benign or adverse, of the operator's action:

Task Failure Analysis

Three levels of decomposition of task failures.

A first decomposition of task failures is based on

- the intention of the operator to resolve a conflict (i_p);

and on the result, benign or adverse, of the operator's action:

- the fact that the initial conflict I_p is effectively resolved ($resolved_p$);

Task Failure Analysis

Three levels of decomposition of task failures.

A first decomposition of task failures is based on

- the intention of the operator to resolve a conflict (i_p);

and on the result, benign or adverse, of the operator's action:

- the fact that the initial conflict I_p is effectively resolved ($resolved_p$);
- the fact that in absence of initial conflict (N_p) a new conflict is created ($adverse_p$).

Decomposition of Task Failures

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$conflict\_created_p = \Diamond adverse_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$conflict\_created_p = \Diamond adverse_p$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$non\_response_p = non\_resolved_p \wedge \Box \neg i_p$$

$$conflict\_created_p = \Diamond adverse_p$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$non\_response_p = non\_resolved_p \wedge \Box \neg i_p$$

$$ineffective\_response_p = non\_resolved_p \wedge \Diamond i_p$$

$$conflict\_created_p = \Diamond adverse_p$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$non\_response_p = non\_resolved_p \wedge \Box \neg i_p$$
$$= non\_resolved_p \wedge no\_intended\_response_p$$

$$ineffective\_response_p = non\_resolved_p \wedge \Diamond i_p$$

$$conflict\_created_p = \Diamond adverse_p$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$
$$= (\Diamond resolved_p) \wedge no\_intended\_response_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$non\_response_p = non\_resolved_p \wedge \Box \neg i_p$$
$$= non\_resolved_p \wedge no\_intended\_response_p$$

$$ineffective\_response_p = non\_resolved_p \wedge \Diamond i_p$$

$$conflict\_created_p = \Diamond adverse_p$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$
$$= (\Diamond resolved_p) \wedge no\_intended\_response_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$non\_response_p = non\_resolved_p \wedge \Box \neg i_p$$
$$= non\_resolved_p \wedge no\_intended\_response_p$$

$$ineffective\_response_p = non\_resolved_p \wedge \Diamond i_p$$

$$conflict\_created_p = \Diamond adverse_p$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$
$$= (\Diamond resolved_p) \wedge no\_intended\_response_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$non\_response_p = non\_resolved_p \wedge \Box \neg i_p$$
$$= non\_resolved_p \wedge no\_intended\_response_p$$

$$ineffective\_response_p = non\_resolved_p \wedge \Diamond i_p$$

$$conflict\_created_p = \Diamond adverse_p$$

$$\mathcal{D}(non\_response_p)$$
$$= \{f \wedge non\_resolved_p \mid f \in \mathcal{D}(no\_intended\_response_p)\}$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$
$$= (\Diamond resolved_p) \wedge no\_intended\_response_p$$

Decomposition of Task Failures

$$non\_resolved_p = \neg \Diamond resolved_p$$

$$non\_response_p = non\_resolved_p \wedge \Box \neg i_p$$
$$= non\_resolved_p \wedge no\_intended\_response_p$$

$$ineffective\_response_p = non\_resolved_p \wedge \Diamond i_p$$

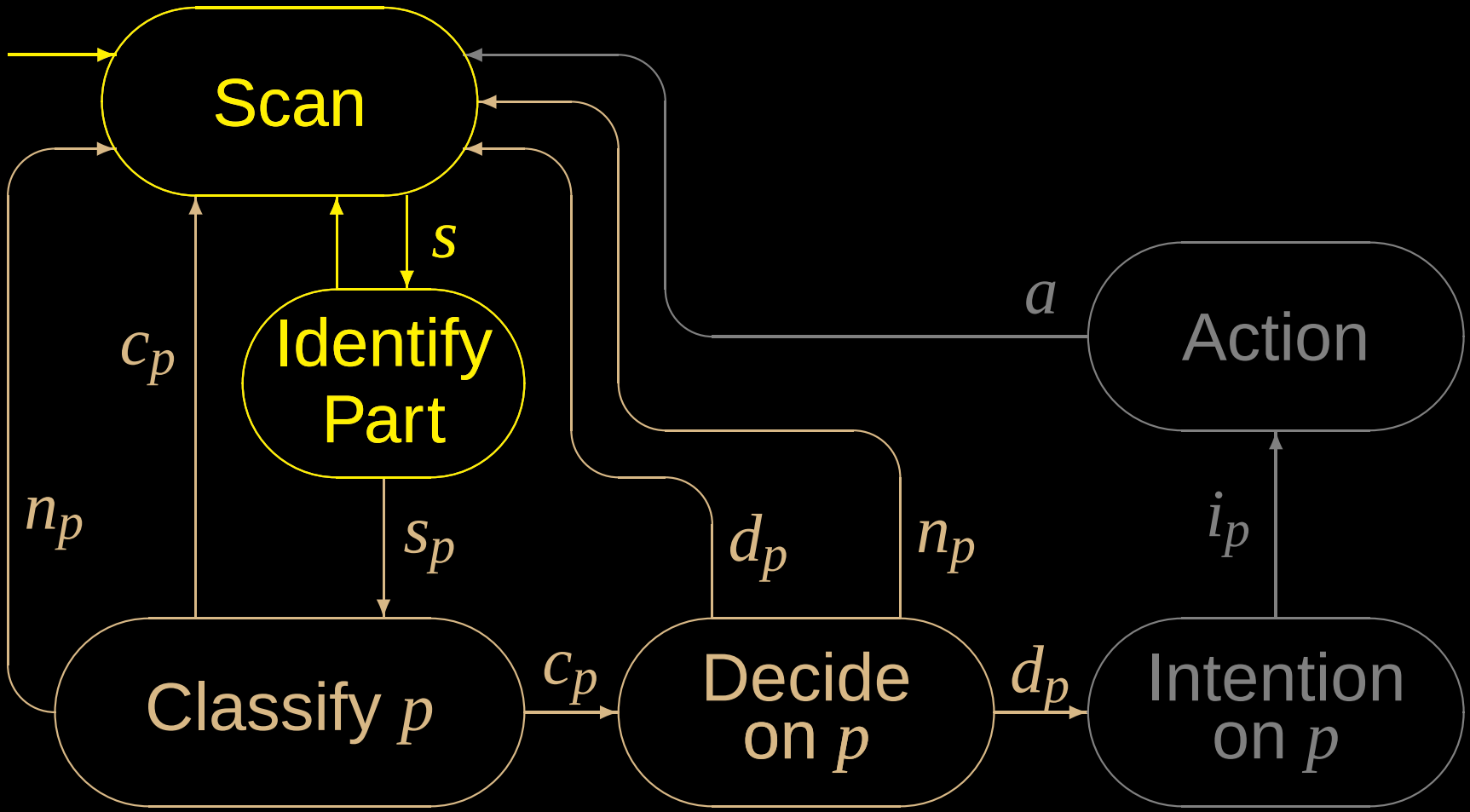
$$conflict\_created_p = \Diamond adverse_p$$

$$\mathcal{D}(fluke_p)$$
$$= \{f \wedge (\Diamond resolved_p) \mid f \in \mathcal{D}(no\_intended\_response_p)\}$$

$$fluke_p = (\Diamond resolved_p) \wedge \Box \neg i_p$$
$$= (\Diamond resolved_p) \wedge no\_intended\_response_p$$

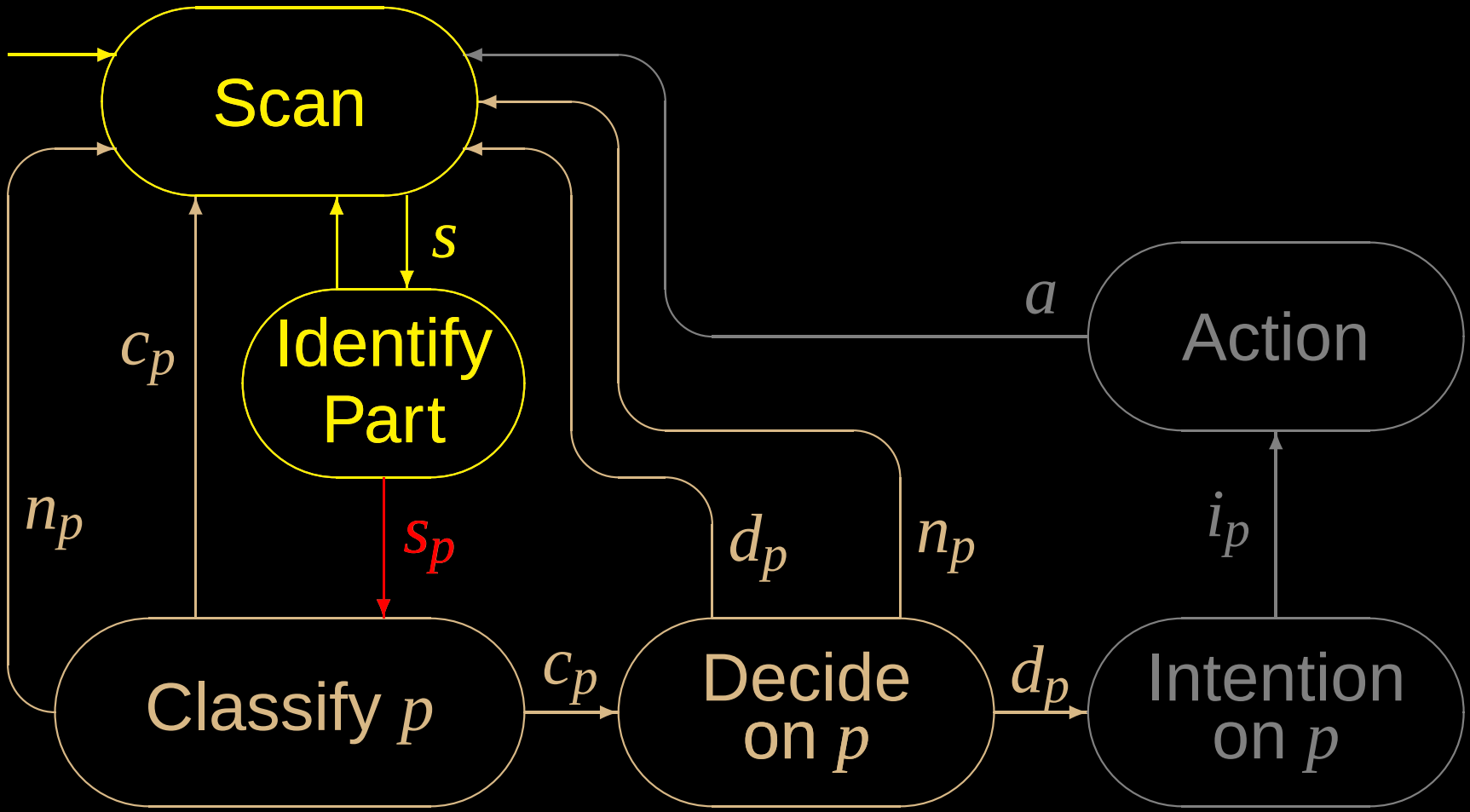
Failure of Scanning

no_intended_response$_p$:



Failure of Scanning

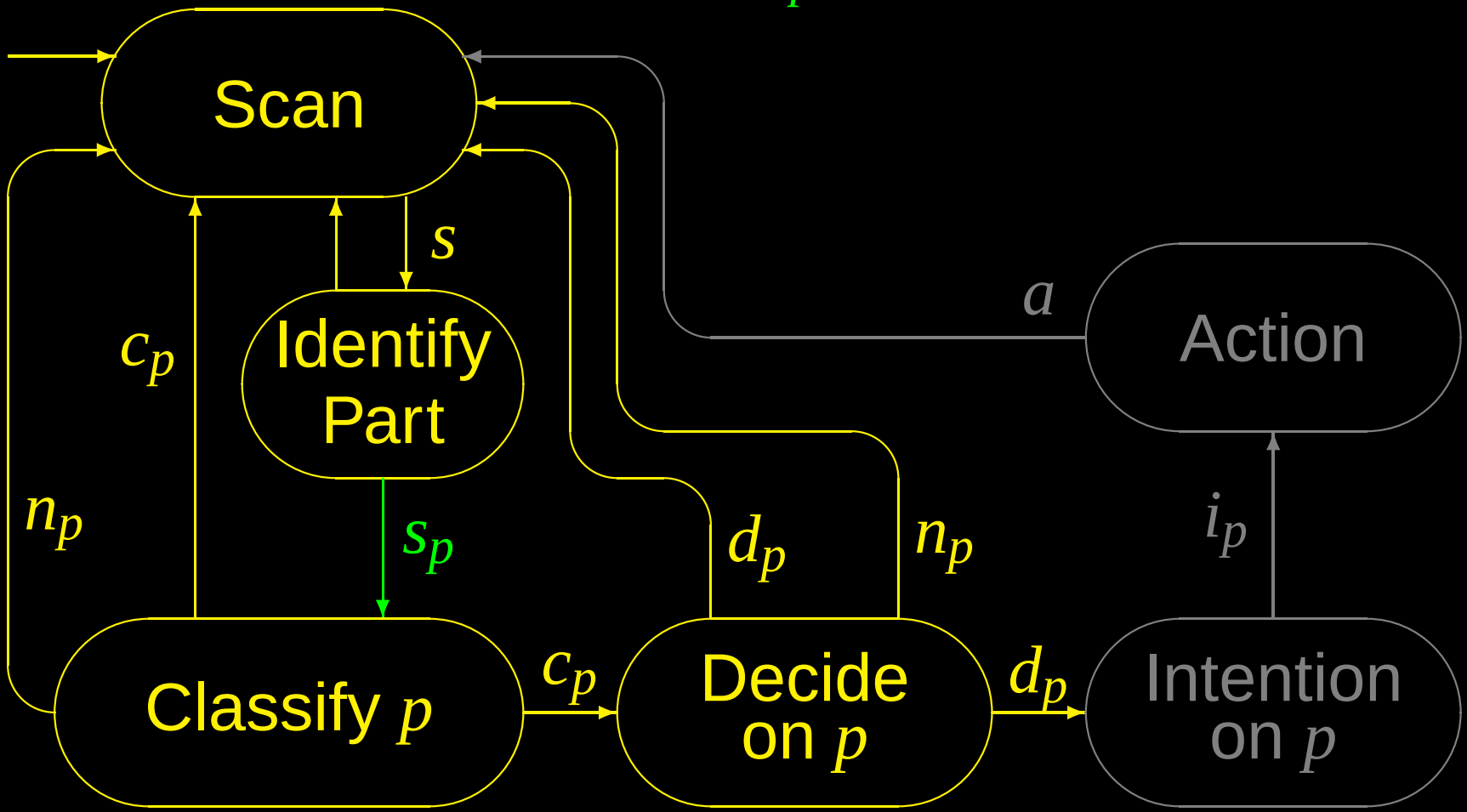
$no\_intended\_response_p$: $\Box \neg s_p$



Failure of Making Decision

$no\_intended\_response_p$: $\Box \neg s_p$

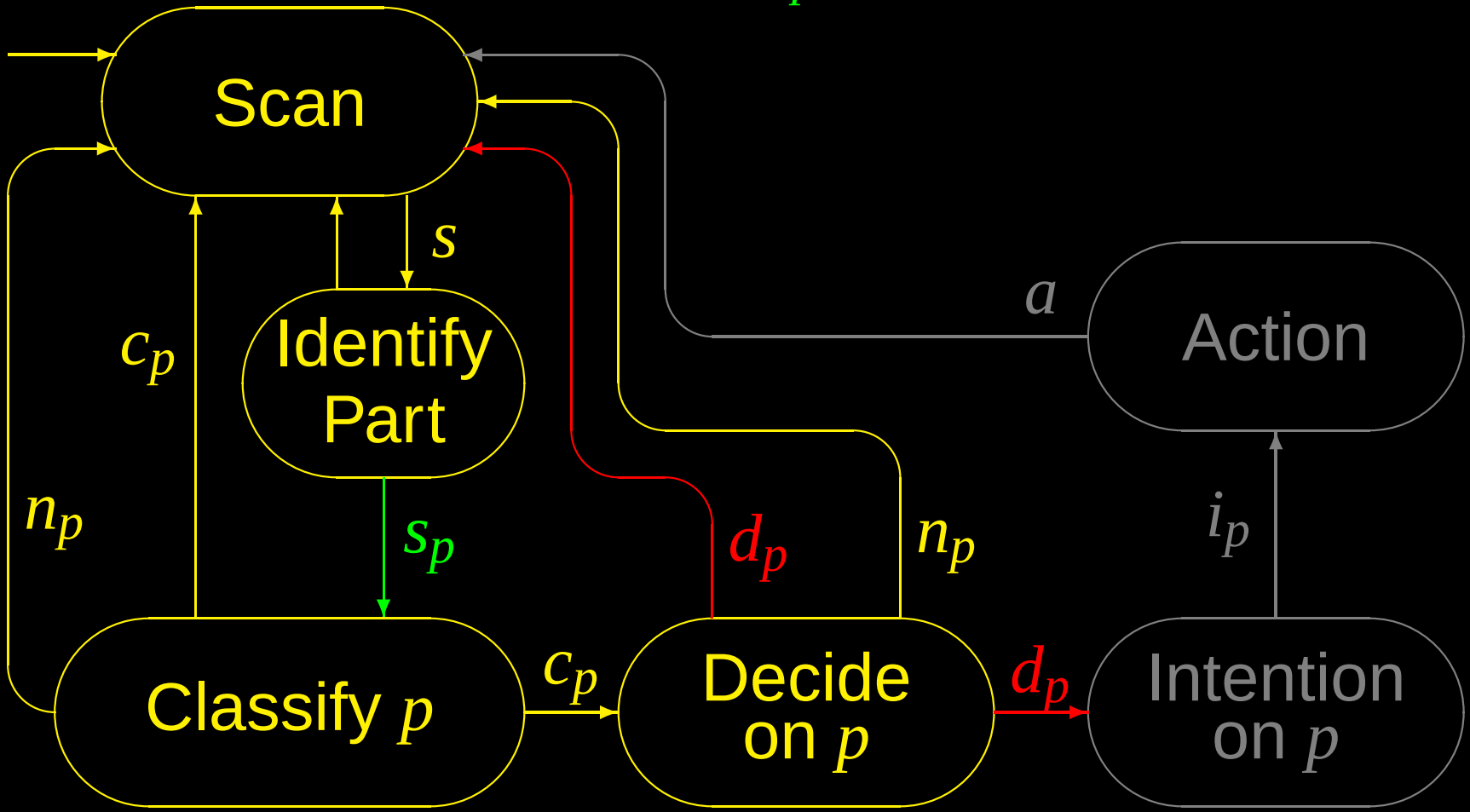
$\Diamond s_p \wedge \ldots$



Failure of Making Decision

$no\_intended\_response_p$: $\Box \neg s_p$

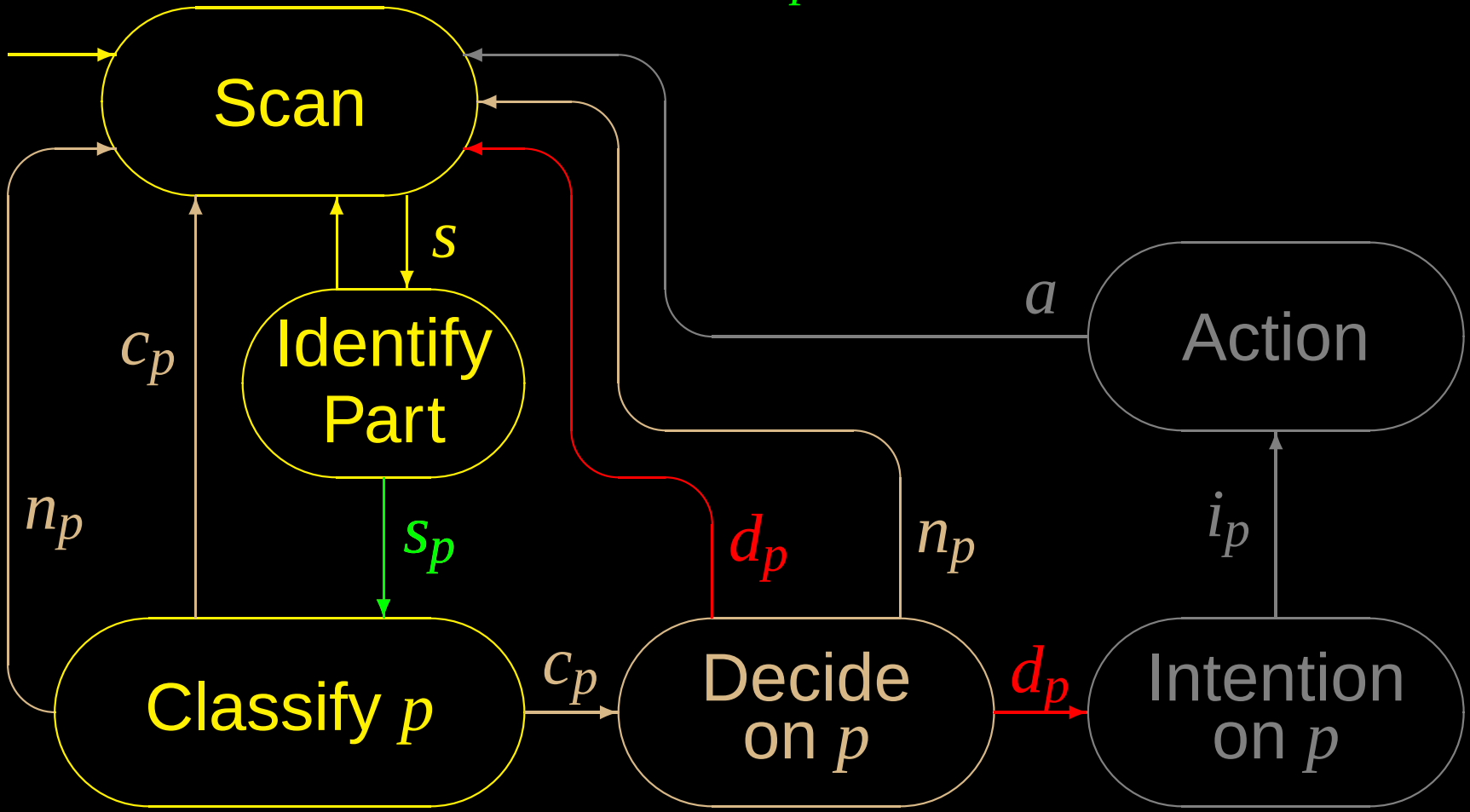
$\Diamond s_p \wedge \ldots$



Persistent Mis-classification

$no\_intended\_response_p$: $\Box\neg\, s_p$

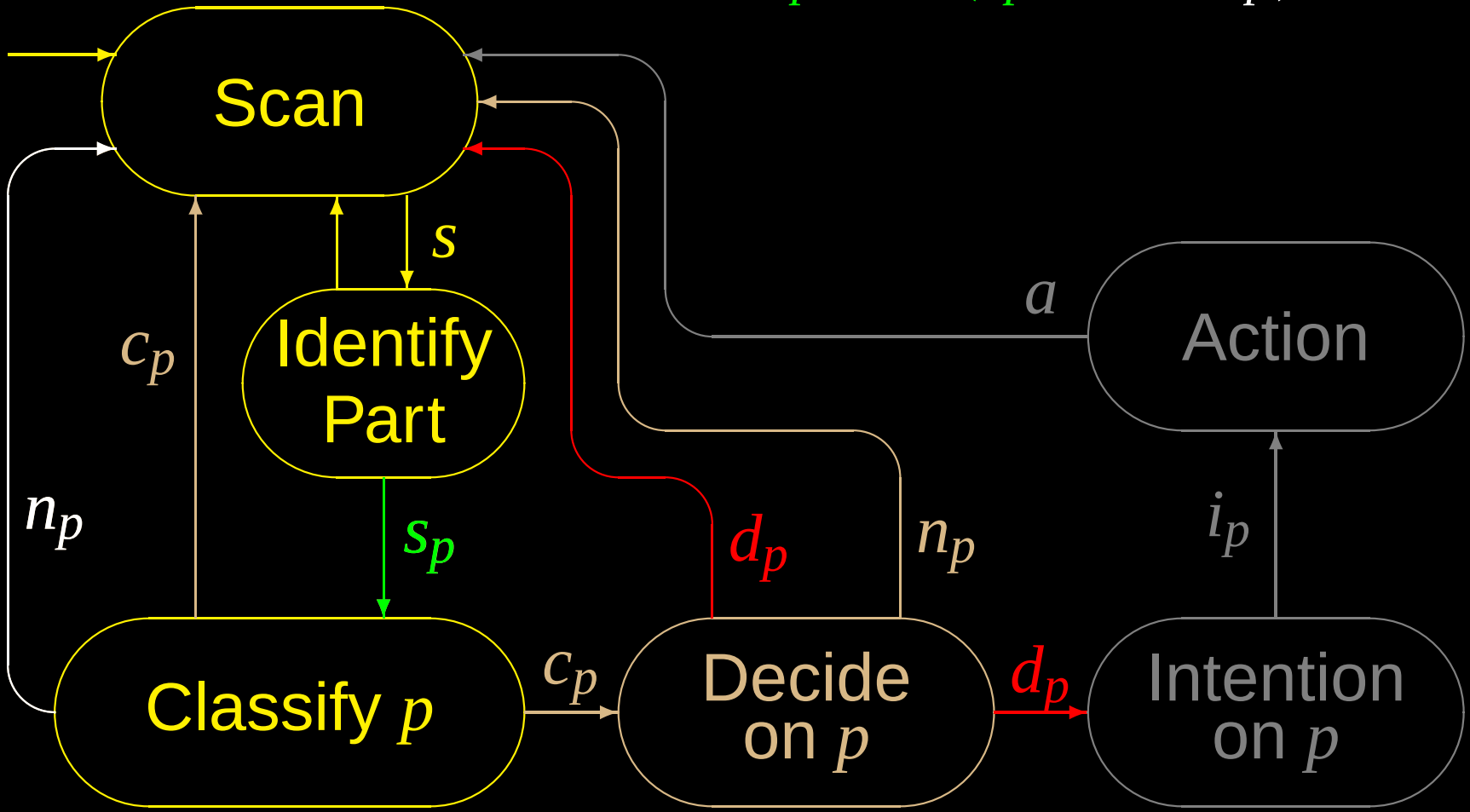
$\Diamond s_p \wedge \ldots$



Persistent Mis-classification

$no\_intended\_response_p$: $\Box\neg\, s_p$

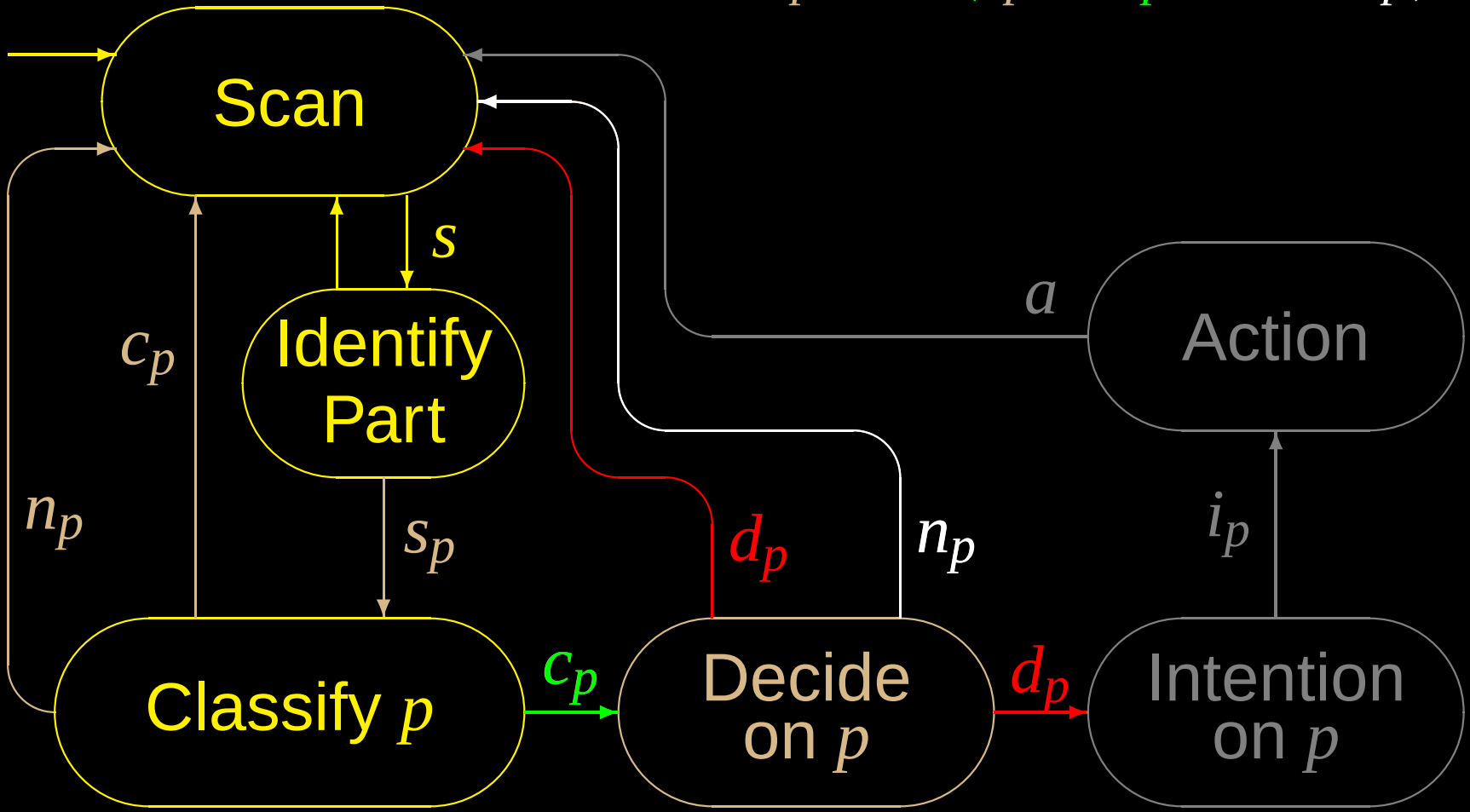
$\Diamond s_p \wedge \Box(s_p \rightarrow \bigcirc n_p)$



Persistent Mis-classification

$no\_intended\_response_p$: $\Box\neg\, s_p$

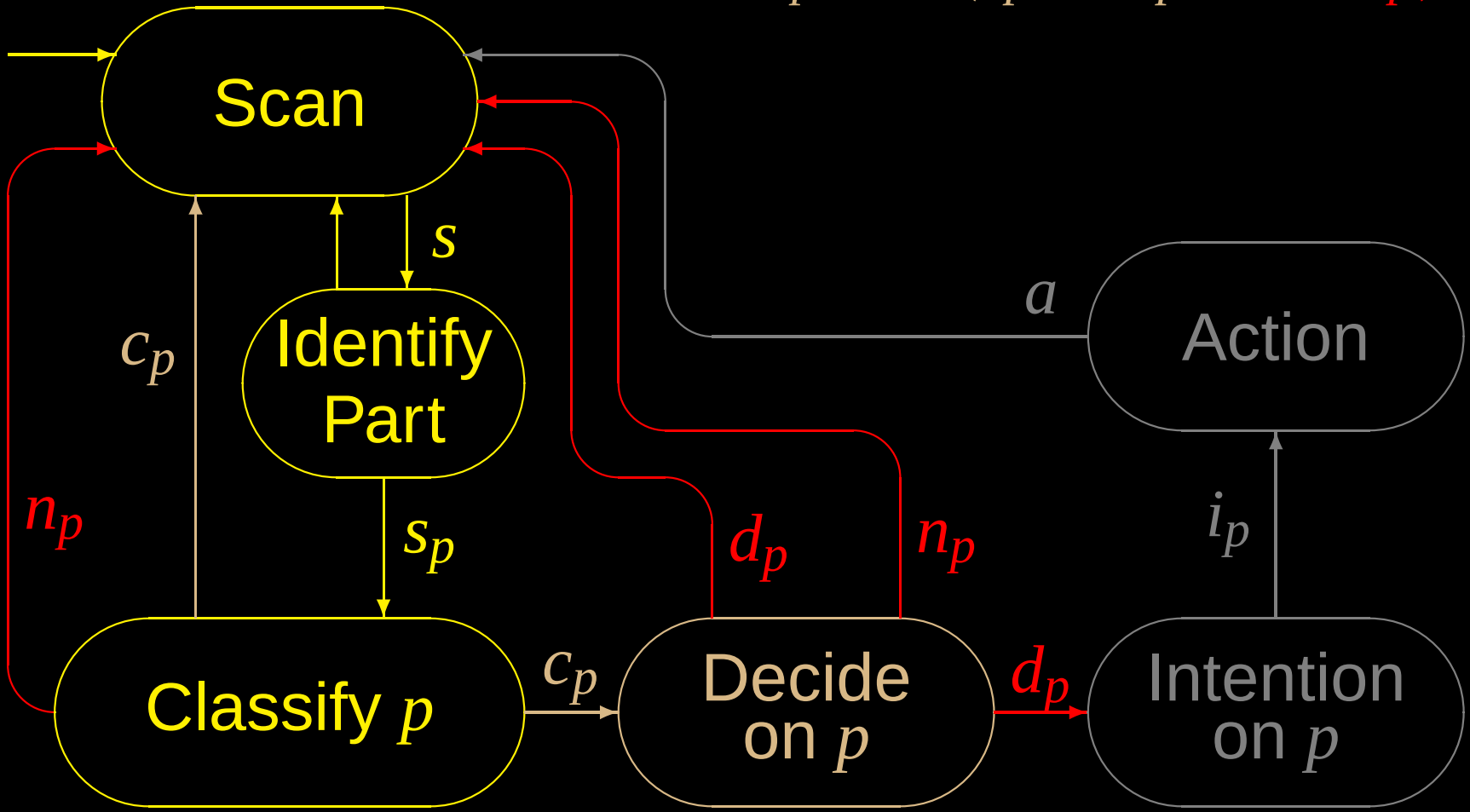
$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$



Persistent Mis-prioritisation

$no\_intended\_response_p$: $\Box\neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$



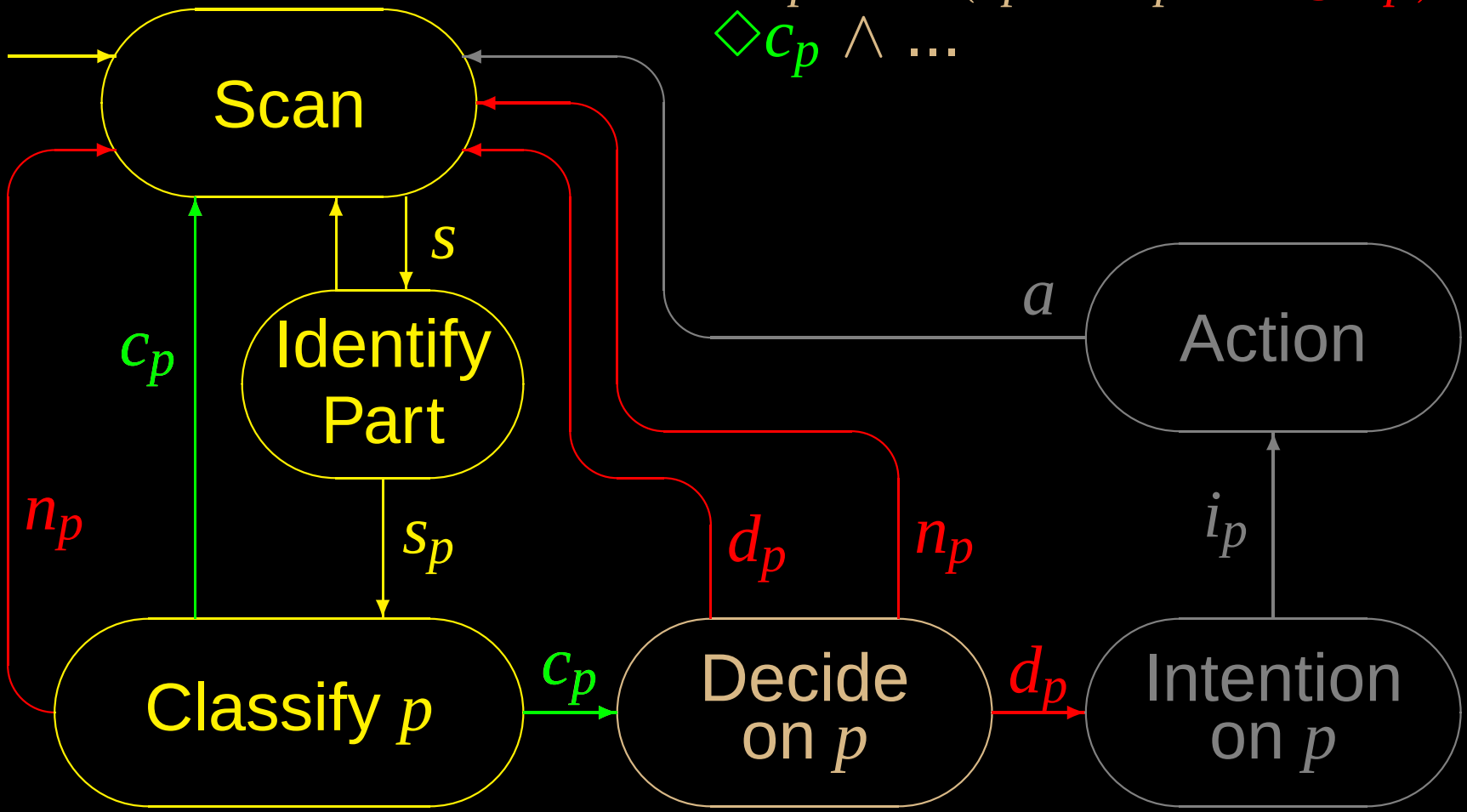
Persistent Mis-prioritisation

$no\_intended\_response_p$:

$\Box\neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

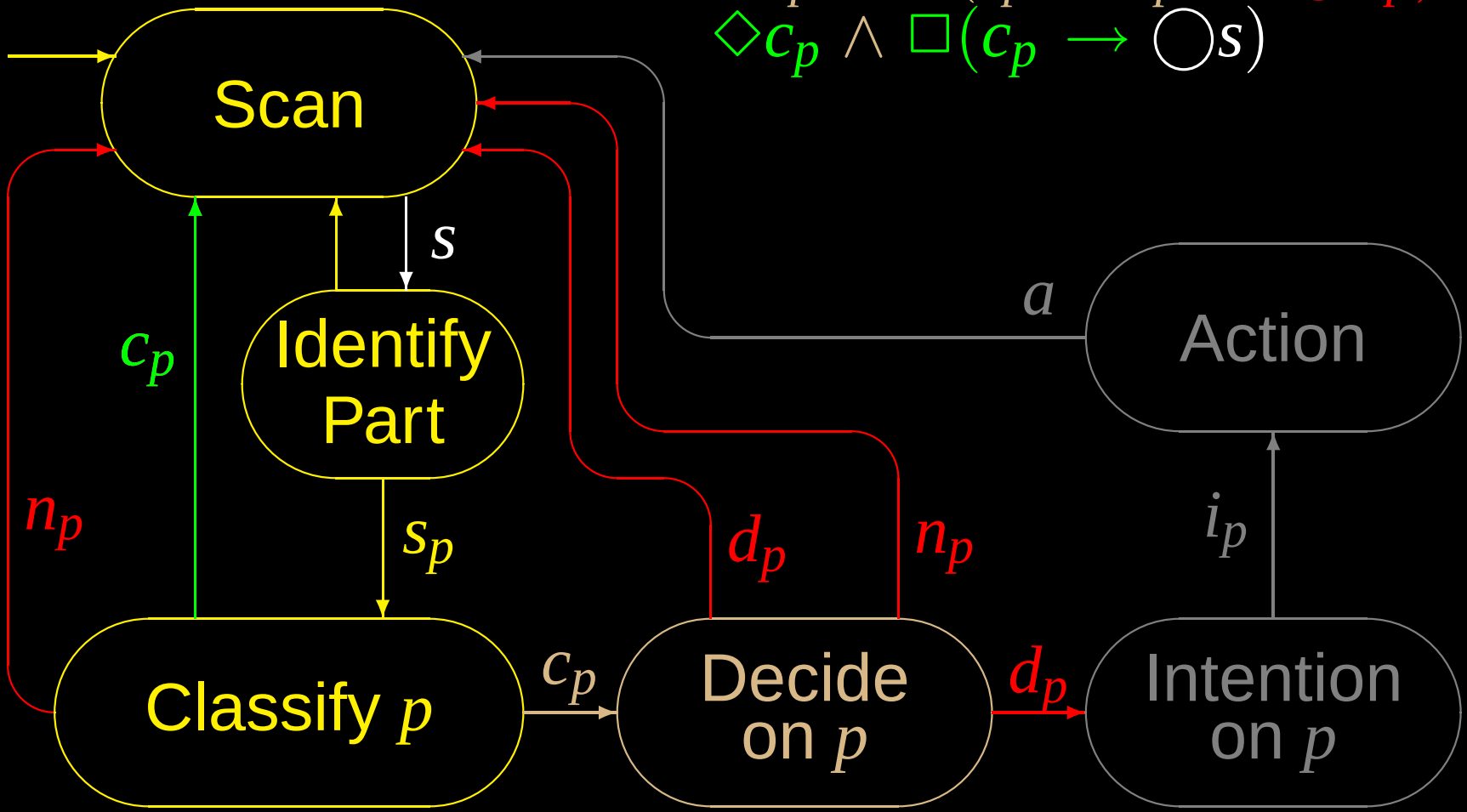
$\Diamond c_p \wedge \ldots$



Persistent Mis-prioritisation

$no\_intended\_response_p$:

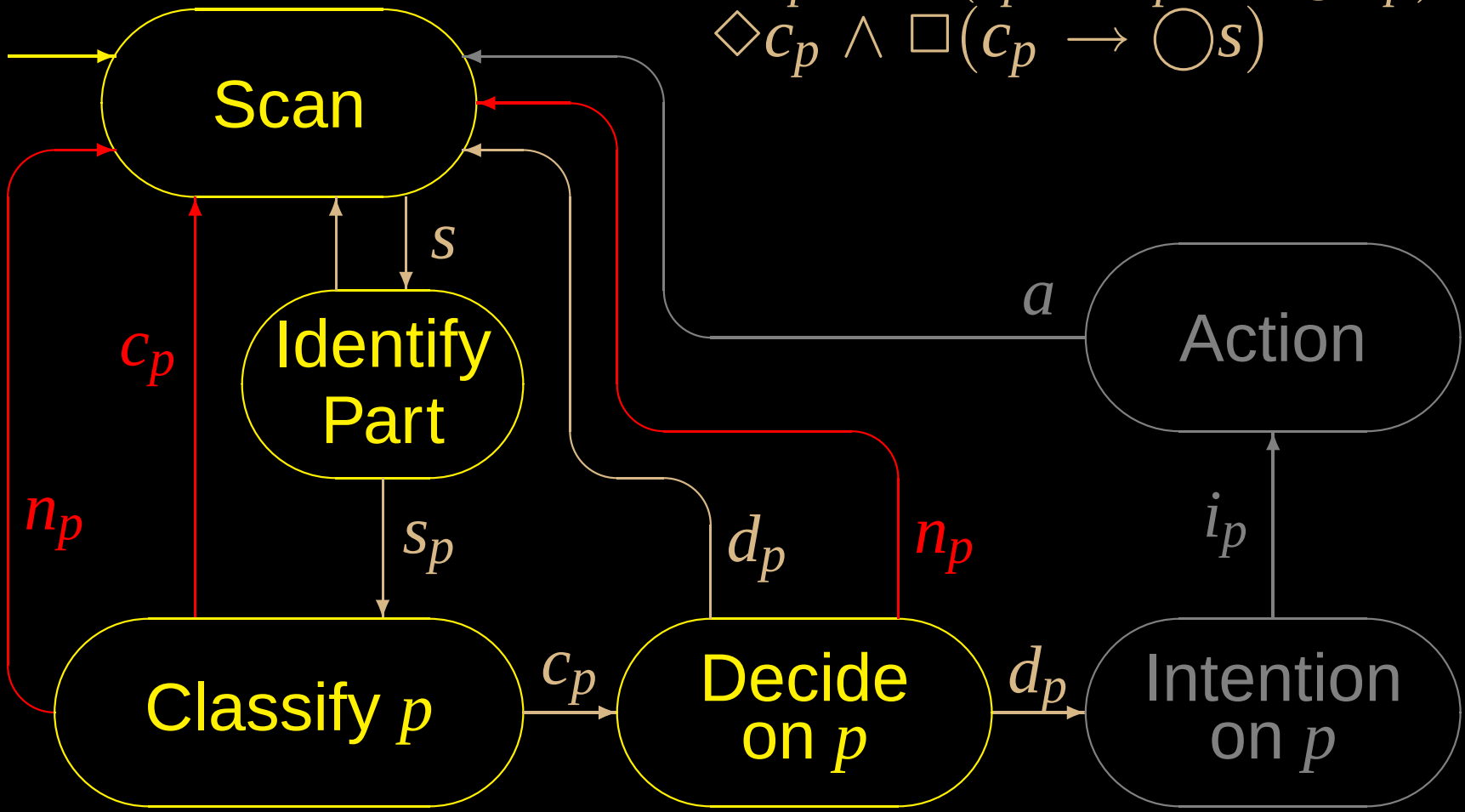
$$\Box\neg\, s_p$$
$$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$$
$$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$$



Defer Action for Too Long

$no\_intended\_response_p$:

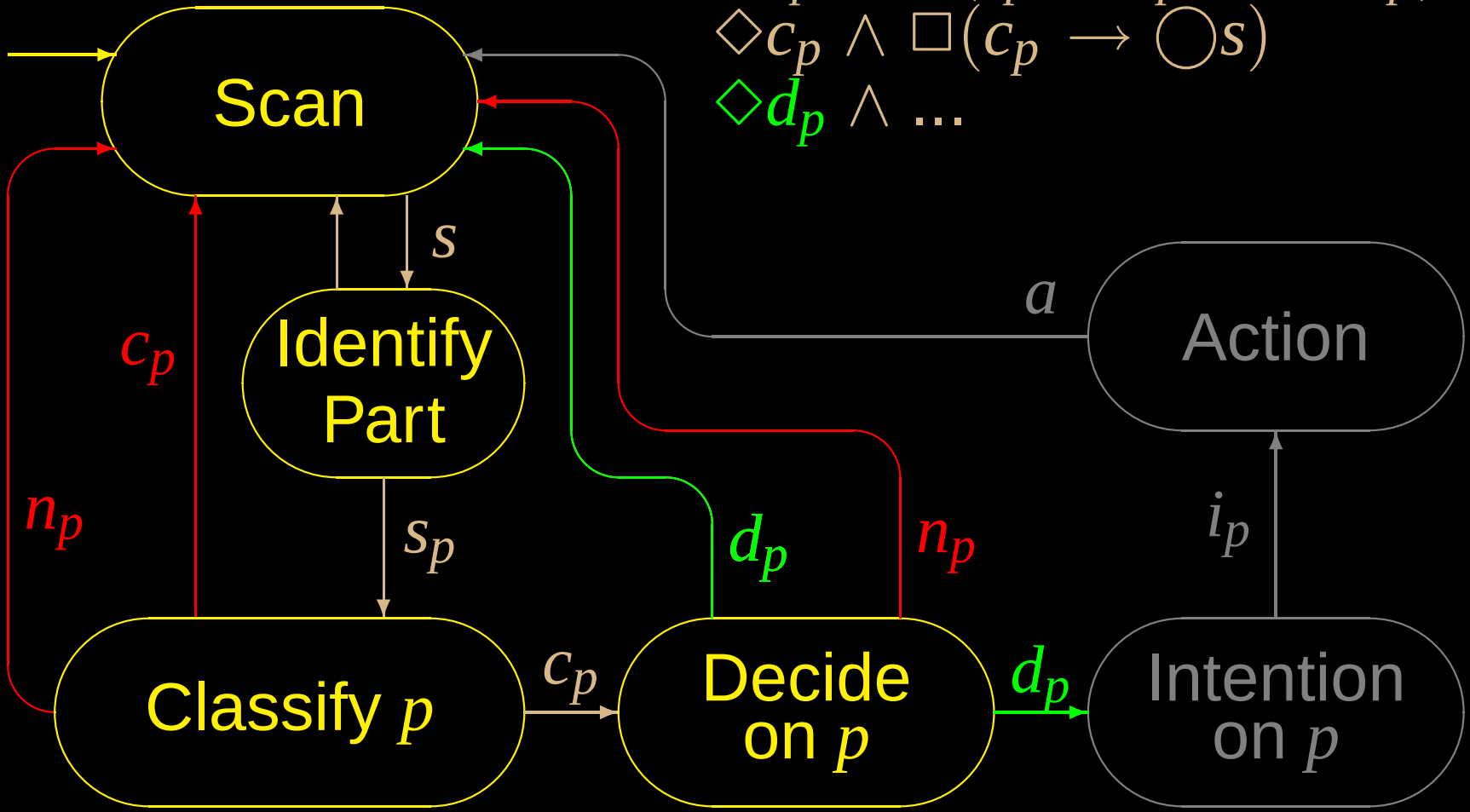
$$\Box\neg\, s_p$$
$$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$$
$$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$$



Defer Action for Too Long

$no\_intended\_response_p$:

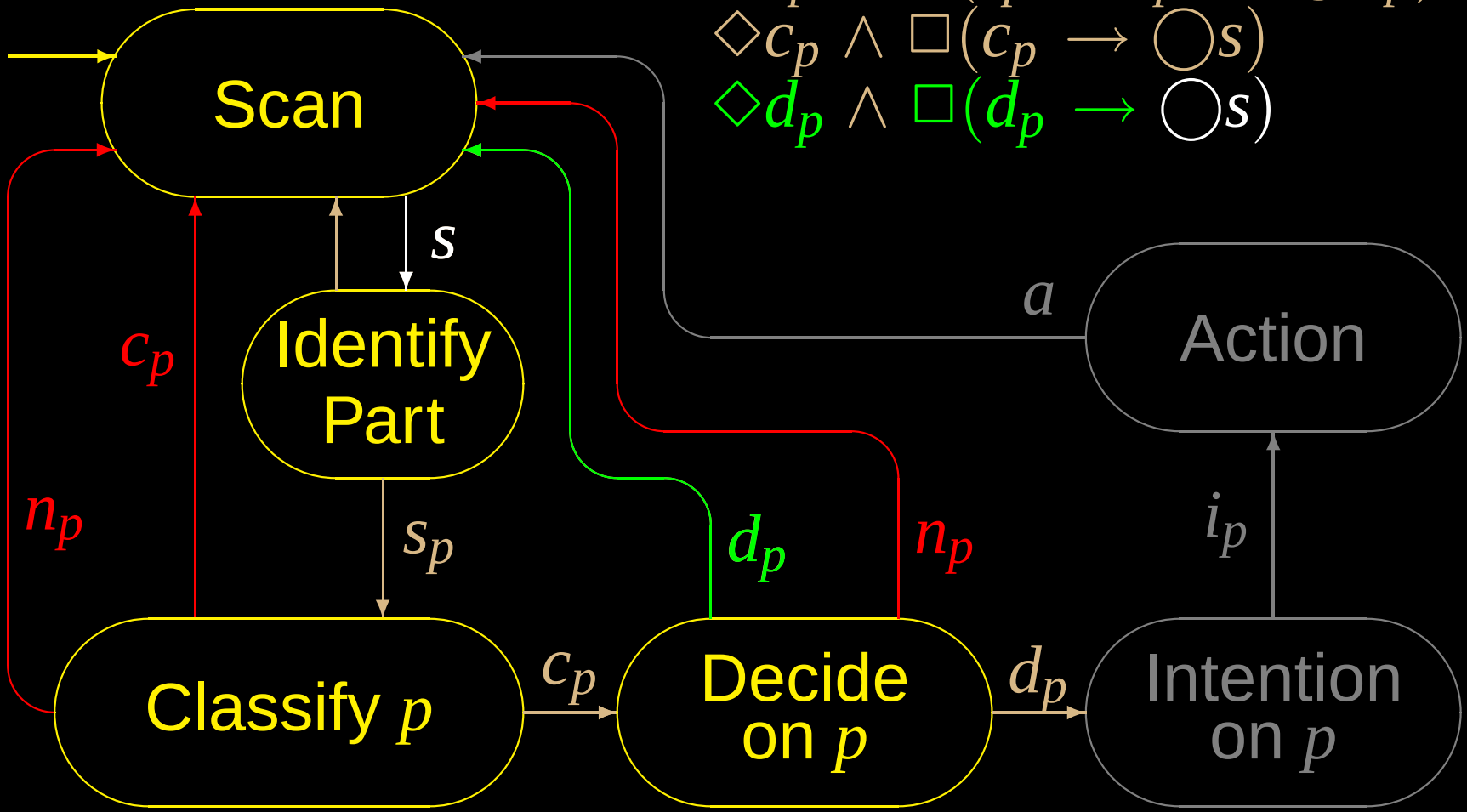
$$\Box\neg\, s_p$$
$$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$$
$$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$$
$$\Diamond d_p \wedge \ldots$$



Defer Action for Too Long

$no\_intended\_response_p$:

$$\Box\neg\, s_p$$
$$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$$
$$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$$
$$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$$



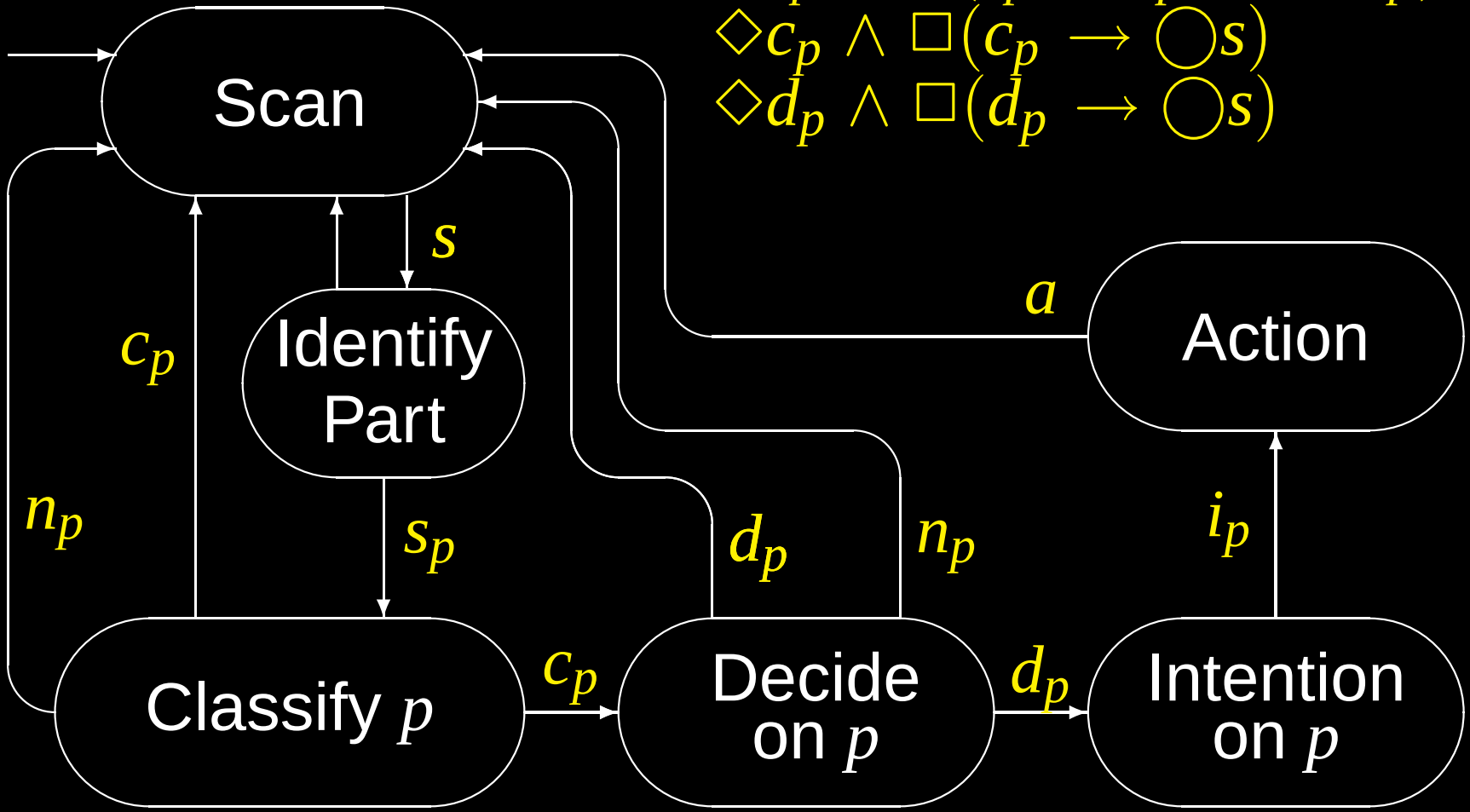
Task Failure Decomposition

$\Box\neg\, i_p$ is decomposed as $\Box\neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$



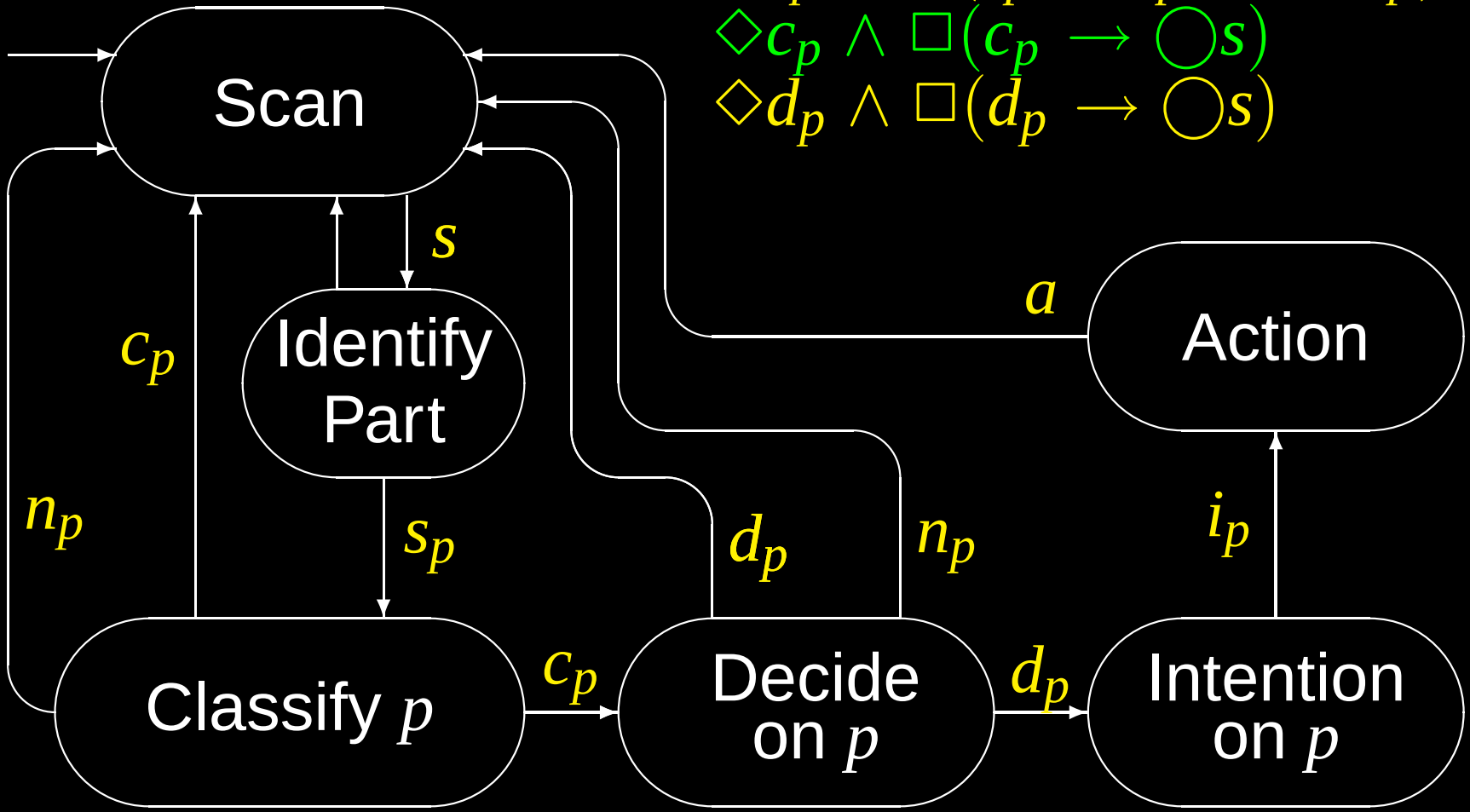
Task Failure Decomposition

$\Box\neg\, i_p$ is decomposed as $\Box\neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$



Single Mis-prioritisation

$c_p \rightarrow \bigcirc s$

(phenotype error)

Single Mis-prioritisation

$c_p \rightarrow \bigcirc s$

(phenotype error)

Possible genotype errors are

- mis-calculation
- mis-storage
- mis-retrival

of the time planned for corrective actions

Single Mis-prioritisation

$c_p \rightarrow \bigcirc s$

(phenotype error)

Possible genotype errors are

- mis-calculation
- mis-storage
- mis-retrival

of the time planned for corrective actions

$\Longrightarrow$ possible recovery

(through new calculation at a next scan)

Persistent Mis-prioritisation

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

(phenotype error)

Persistent Mis-prioritisation

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

(phenotype error)

Possible genotype errors

- perception distorted $\Longrightarrow$ memory of result of previous mis-calculation keeps emerging

due to

- distraction
- similarity with observed non-conflicts
- high workload

Final Decomposition

$$\mathcal{D}(\mathit{no\_intended\_response}_p) = \{\Box\neg\, s_p\ ,\\ \Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)\ ,\\ \Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)\ ,\\ \Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)\ \}$$

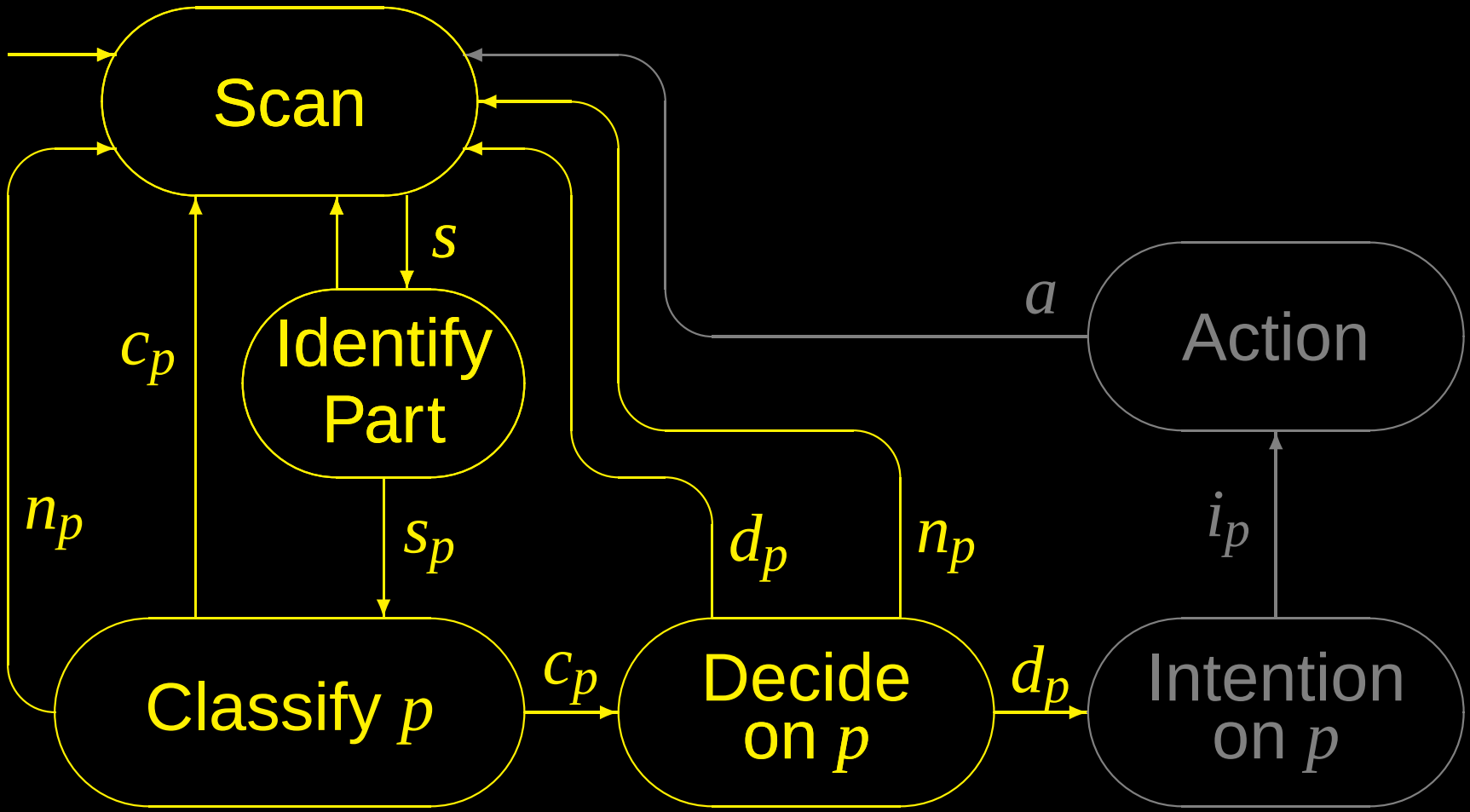
Final Decomposition

$$\mathcal{D}(\mathit{no\_intended\_response}_p) = \{\Box\neg\, s_p\ ,\ \Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)\ ,\ \Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)\ ,\ \Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)\ \}$$

$$\begin{aligned}
&\mathcal{D}(\mathit{non_response}_p)\\
&= \{f \wedge \mathit{non_resolved}_p \mid f \in \mathcal{D}(\mathit{no_intended_response}_p)\}\\
&= \{\Box\neg\, s_p \wedge \mathit{non_resolved}_p,\\
&\quad \Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p\ ) \wedge \mathit{non_resolved}_p,\\
&\quad \Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s) \wedge \mathit{non_resolved}_p,\\
&\quad \Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s) \wedge \mathit{non_resolved}_p\}
\end{aligned}$$

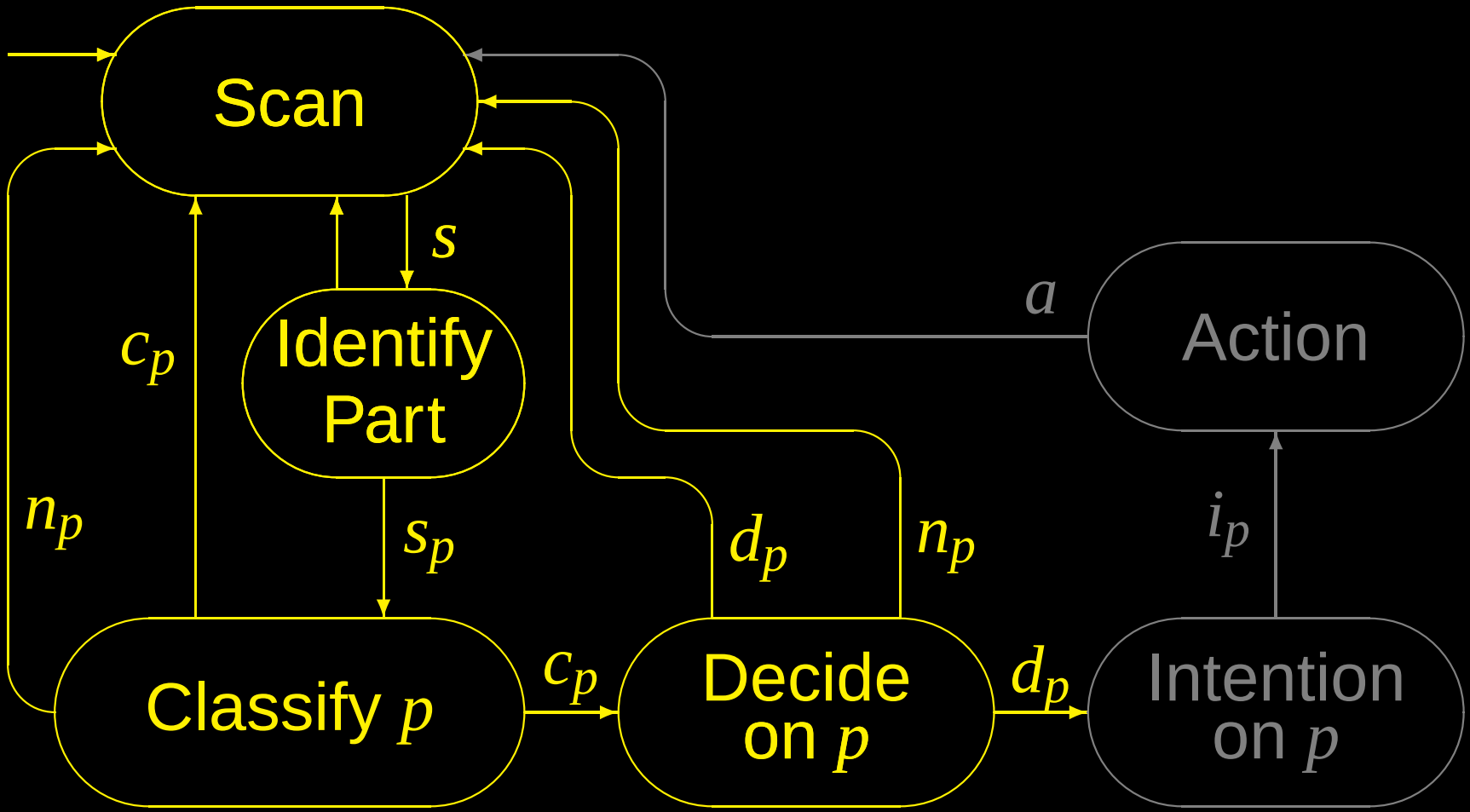
Soundness

Soundness


Scan
s
c_p
Identify Part
a
Action
n_p
s_p
d_p
n_p
i_p
Classify p
c_p
Decide on p
d_p
Intention on p

Soundness

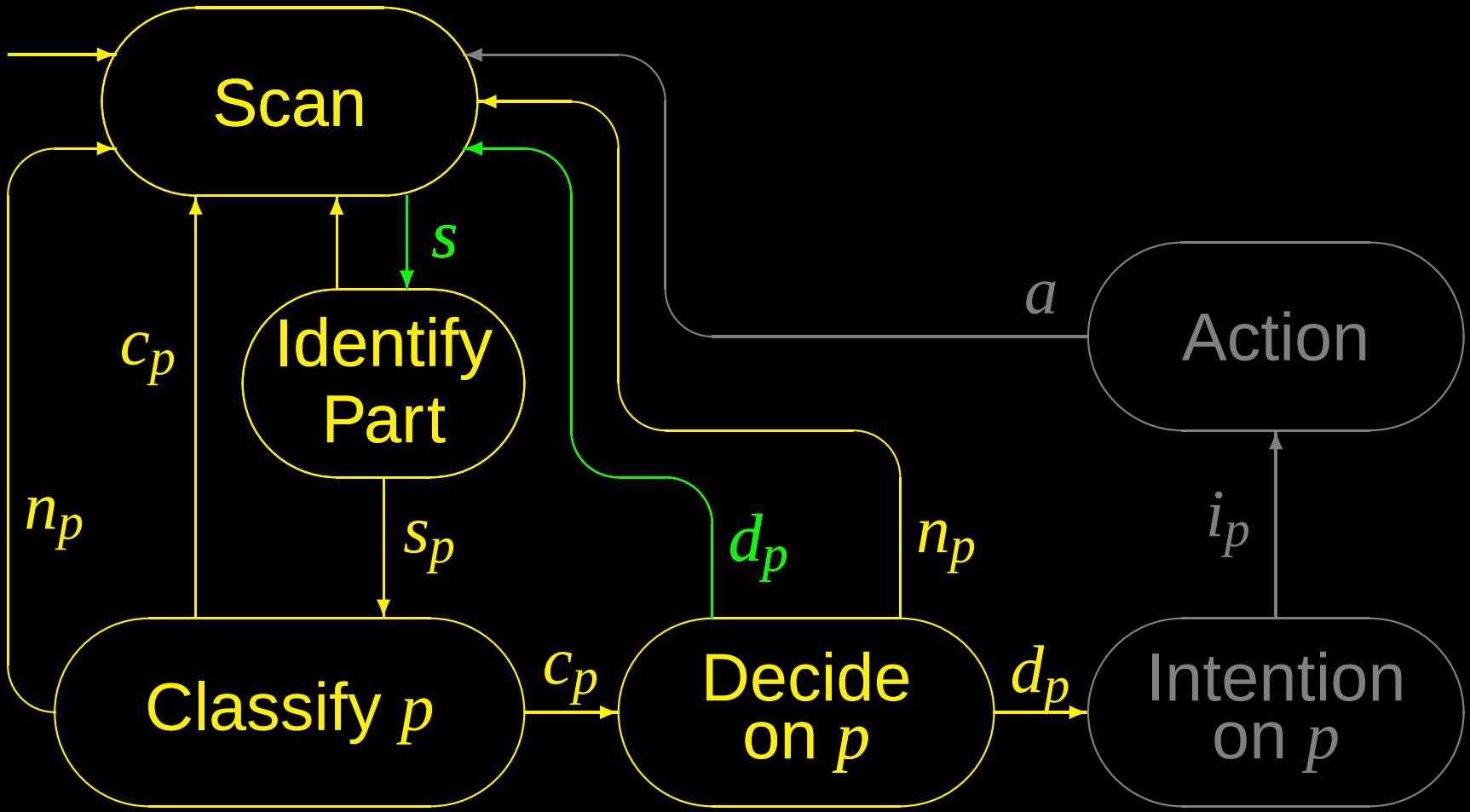
Example: Defer action for too long



Soundness

Example: Defer action for too long

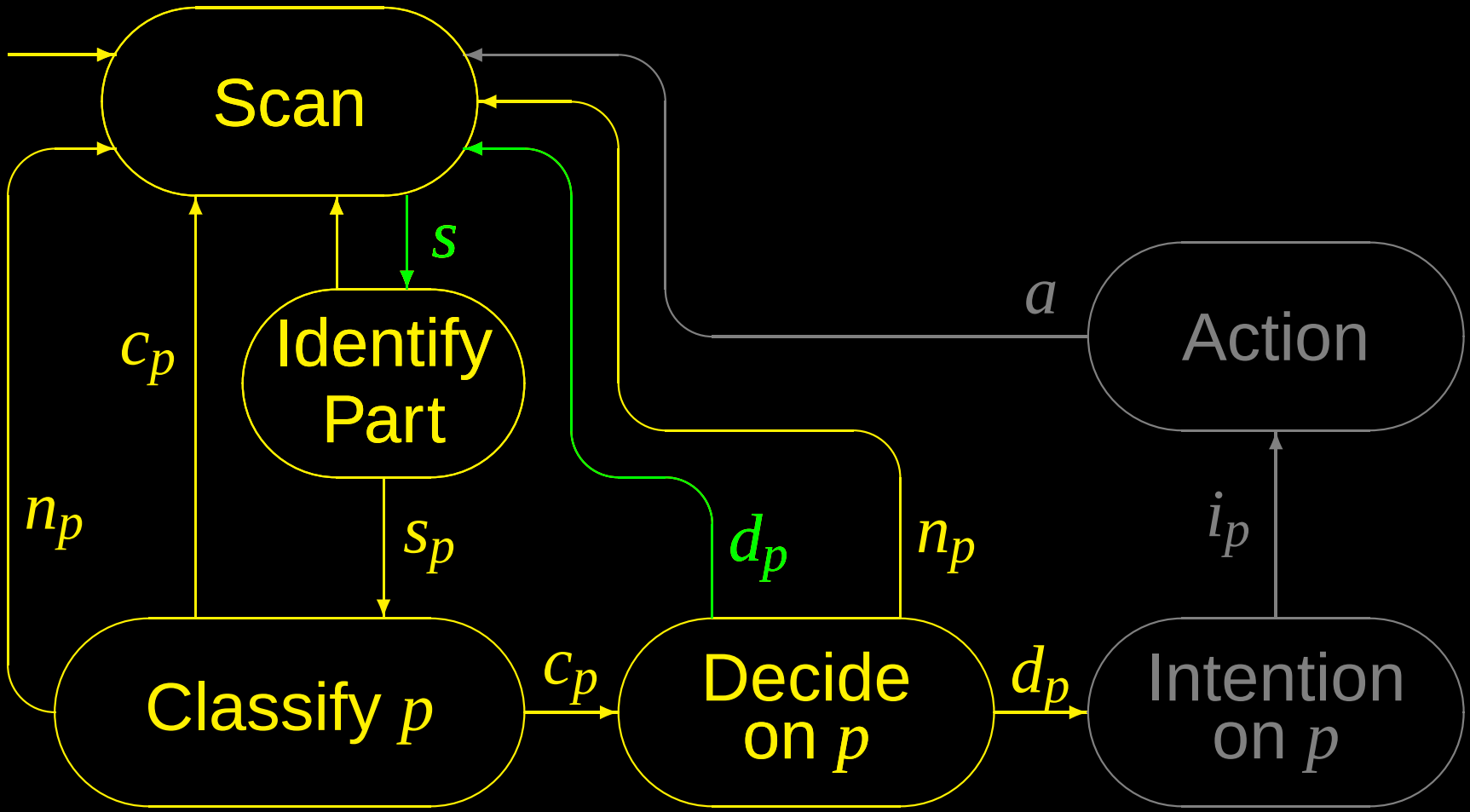
$$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$$



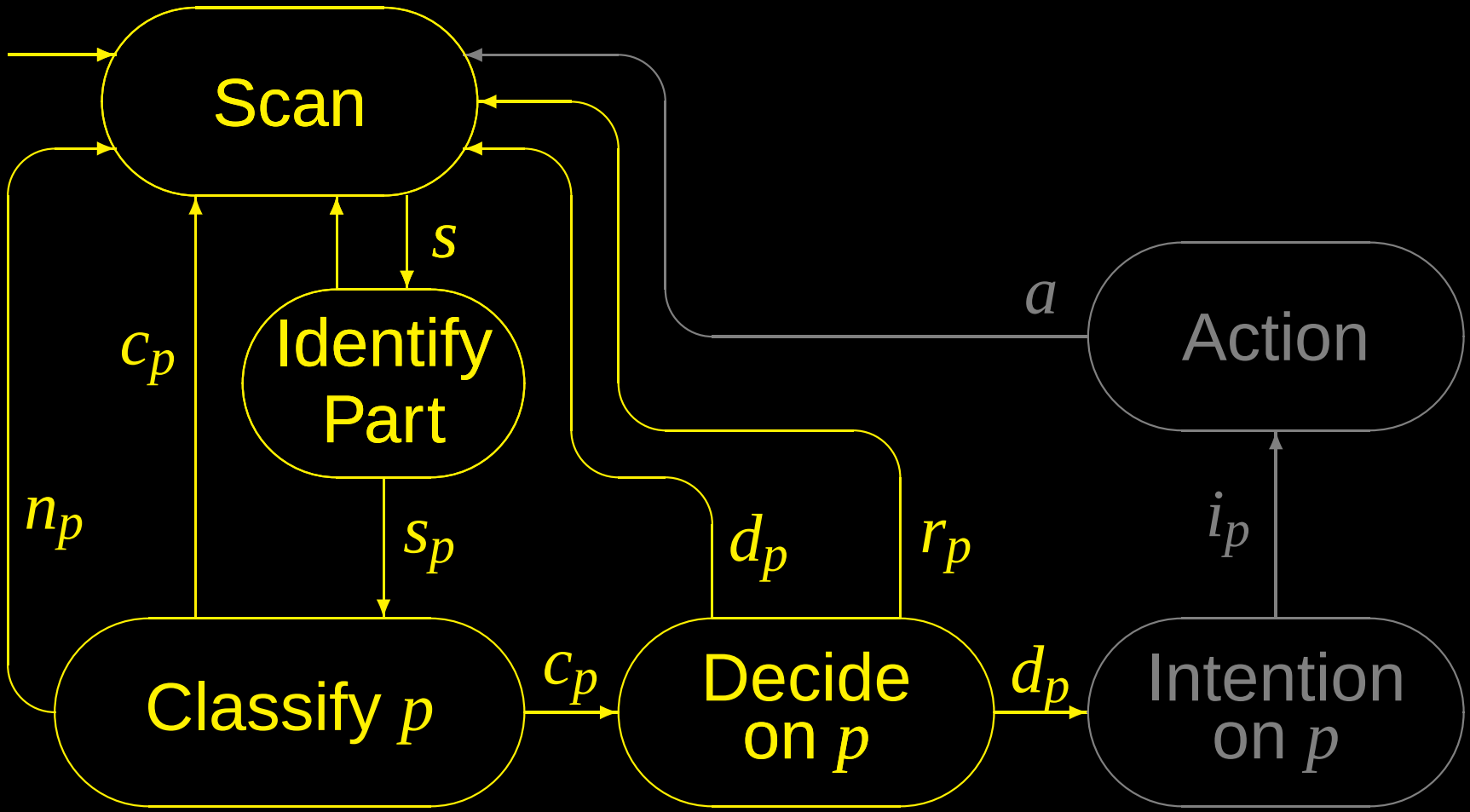
Soundness

Example: Defer action for too long

$$(\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)) \rightarrow \Box\neg\, i_p$$

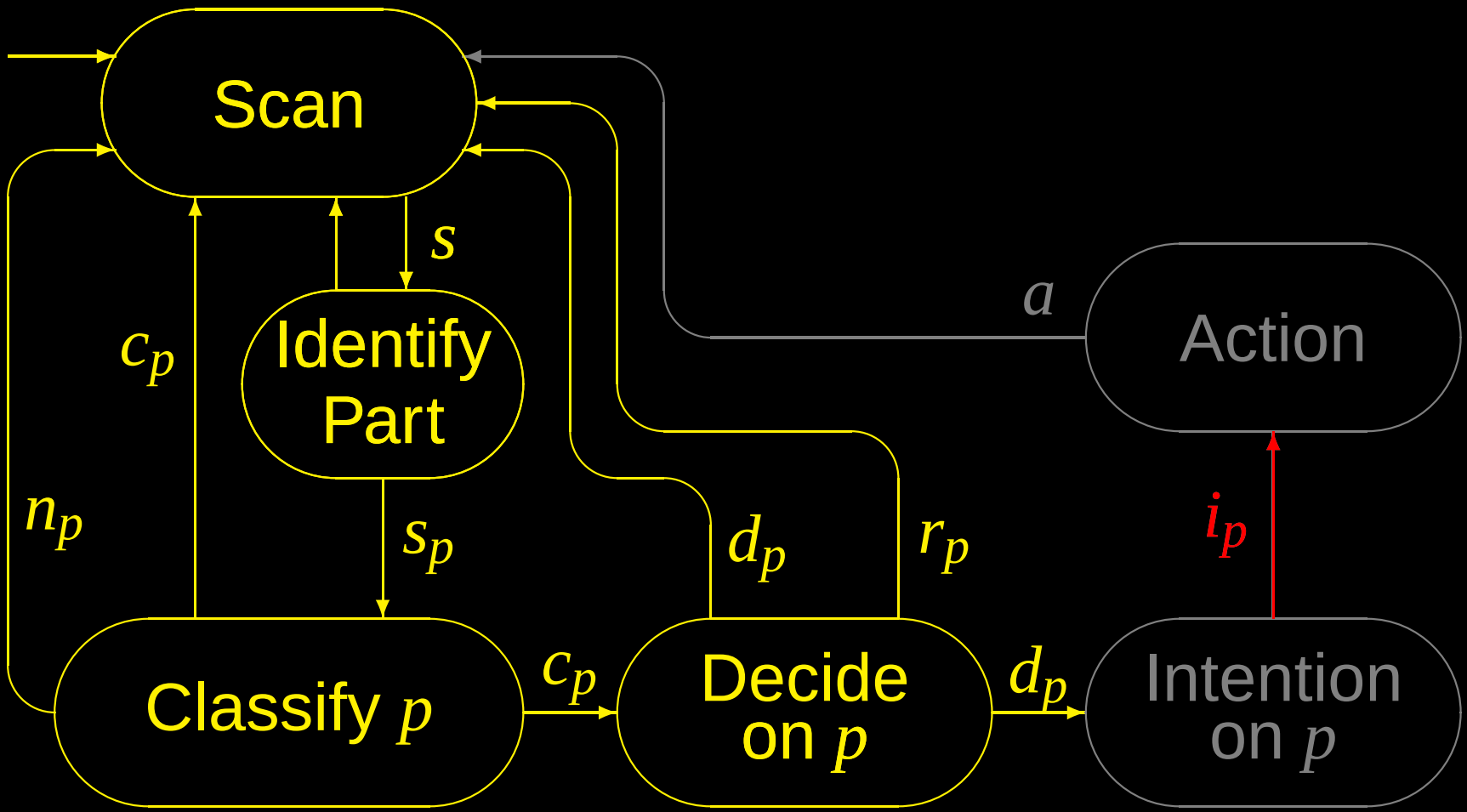


Completeness



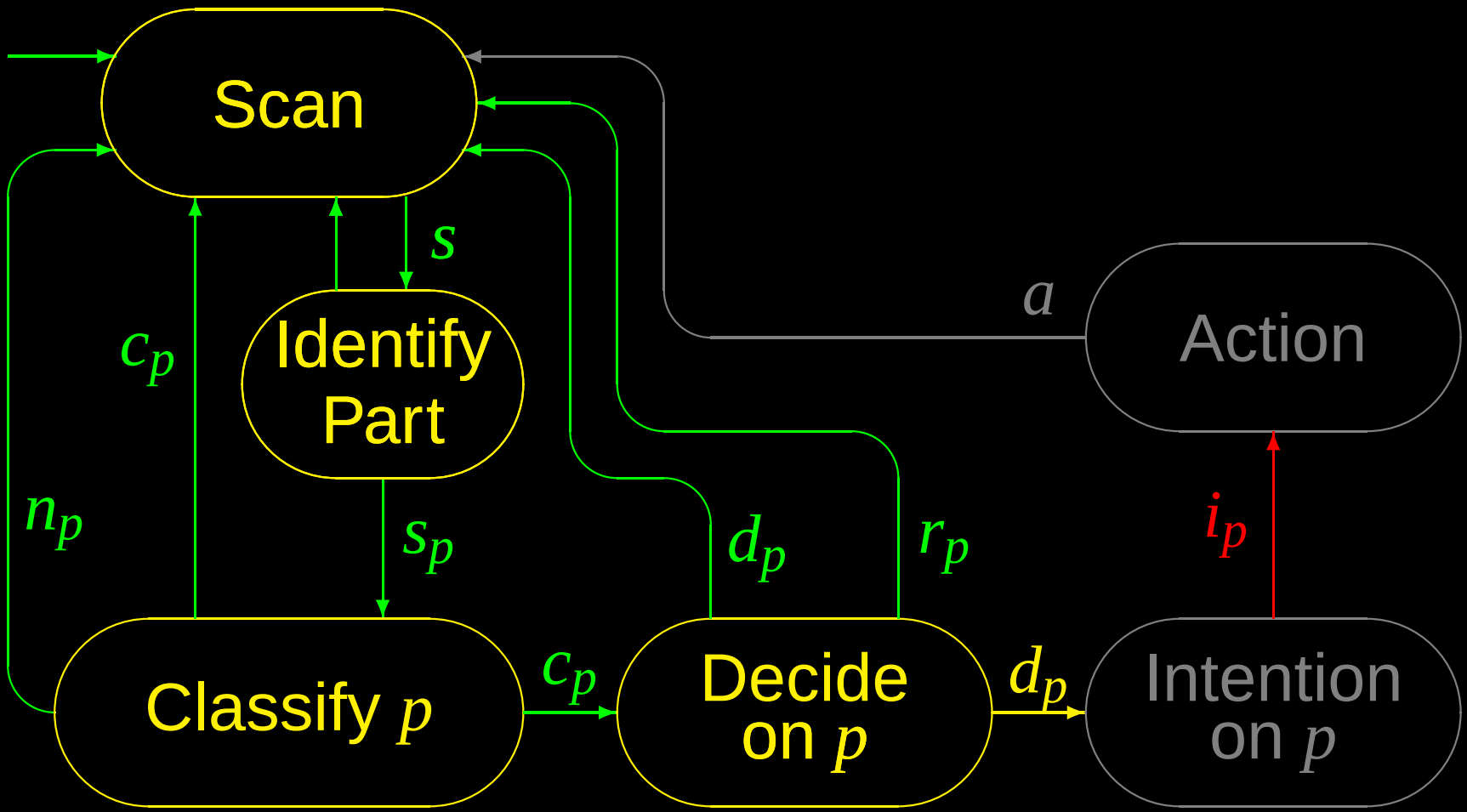
Completeness

$\Box \neg i_p \ \ldots$



Completeness

$\Box \neg i_p \rightarrow \ldots$



Completeness

$$\Box\neg\, i_p \;\rightarrow\; \bigvee_{f \in \mathcal{F}} f$$

where

$$\mathcal{F} = \mathcal{D}(\Box\neg\, i_p) = \mathcal{D}(no\_intended\_response_p) =$$

Completeness

$$\Box\neg\, i_p \;\rightarrow\; \bigvee_{f \in \mathcal{F}} f$$

where

$$\mathcal{F} = \mathcal{D}(\Box\neg\, i_p) = \mathcal{D}(\mathit{no\_intended\_response}_p) =$$

$$\{\, \Box\neg\, s_p \;,$$
$$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p) \;,$$
$$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s) \;,$$
$$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s) \,\}$$

ATC: References

Peter Lindsay and Simon Connelly. Modelling Erroneous Operator Behaviours for an Air-Traffic Control Task. AUIC 2002.

ATC: References

Peter Lindsay and Simon Connelly. Modelling Erroneous Operator Behaviours for an Air-Traffic Control Task. AUIC 2002.

Antonio Cerone, Peter Lindsay and Simon Connelly. Formal Analysis of Human-computer Interaction using Model-checking. SEFM 2005.

Antonio Cerone, Simon Connelly and Peter Lindsay. Formal Analysis of Human Operator Behavioural Patterns in Interactive Surveillance Systems. Software and System Modeling 7(3), Springer, pages 273–286, 2008.

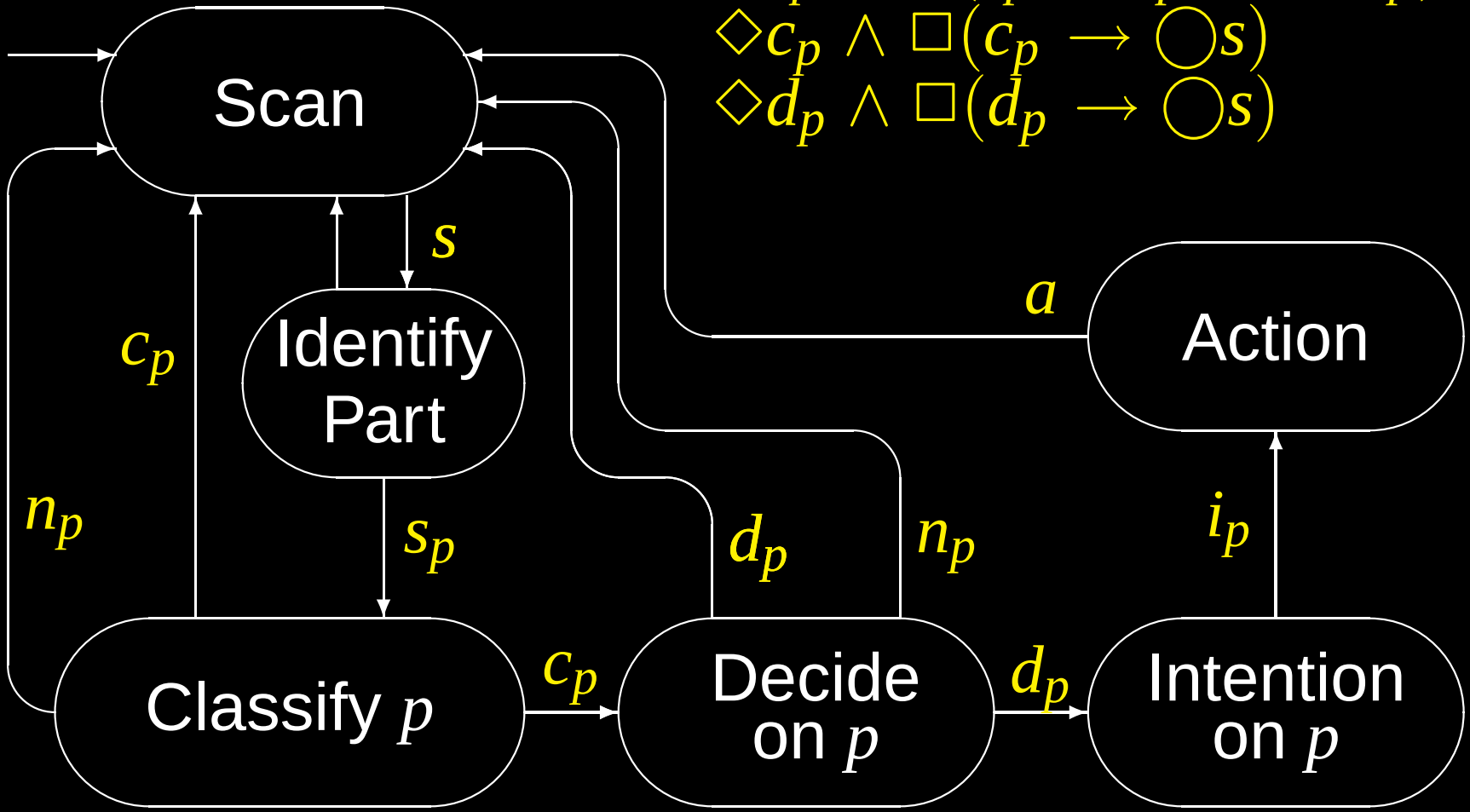
Task Failure Decomposition

$\Box\neg\, i_p$ is decomposed as $\Box\neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$



Complete Decomposition

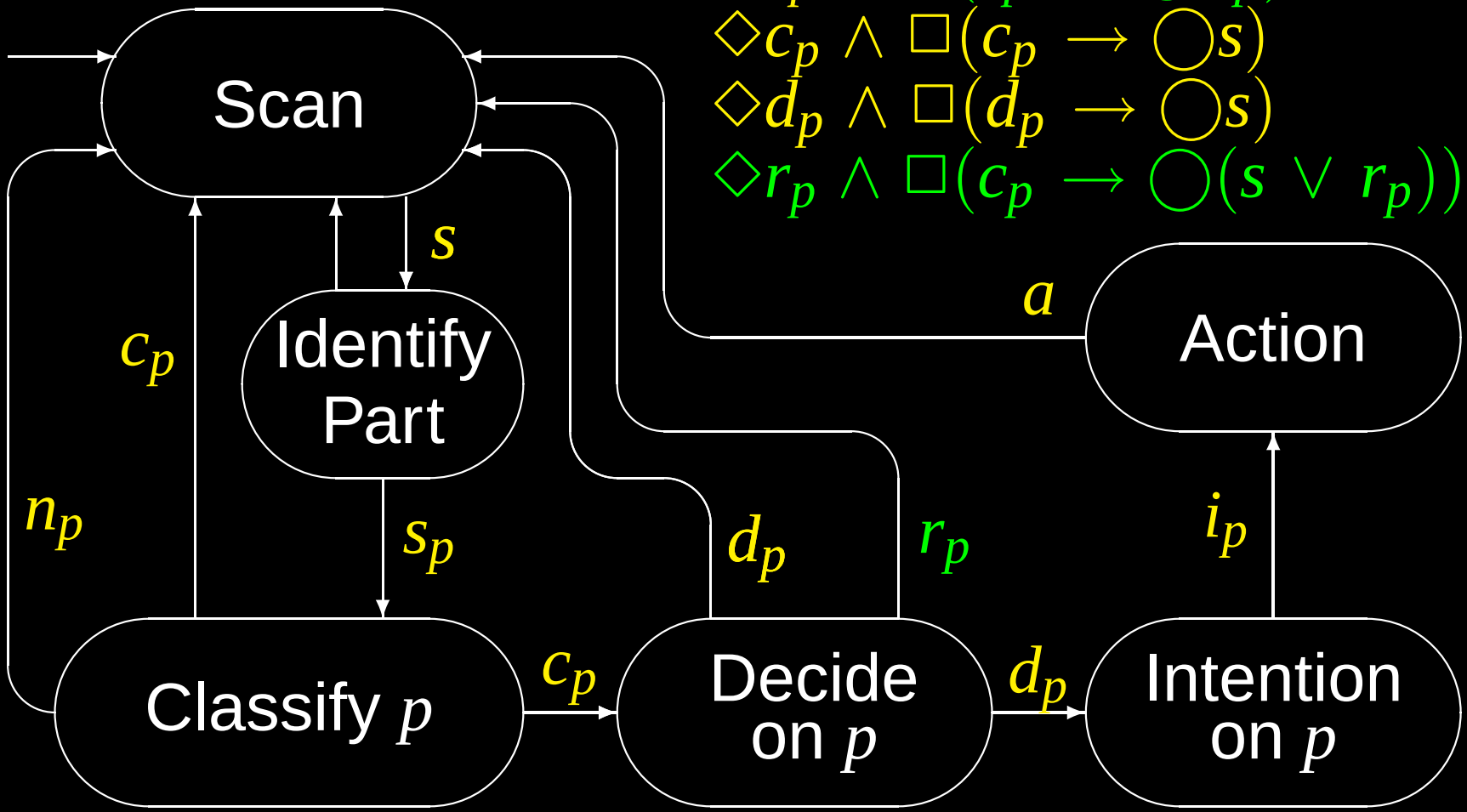
$\Box\neg\, i_p$ is decomposed as $\Box\neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \rightarrow \bigcirc n_p)$

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$

$\Diamond r_p \wedge \Box(c_p \rightarrow \bigcirc(s \vee r_p))$



Contrary Decision Process

$\Diamond r_p \wedge \Box(c_p \rightarrow \bigcirc(s \vee r_p))$

(phenotype error)

Possible genotype error is

- memory of previous decisions on similar pairs resulting in unnecessary actions

due to

- fear
- high workload

Error Cause

What caused such an error?

Error Cause

What caused such an error?

- use of the same action name n to denote the results of two cognitive processes

Error Cause

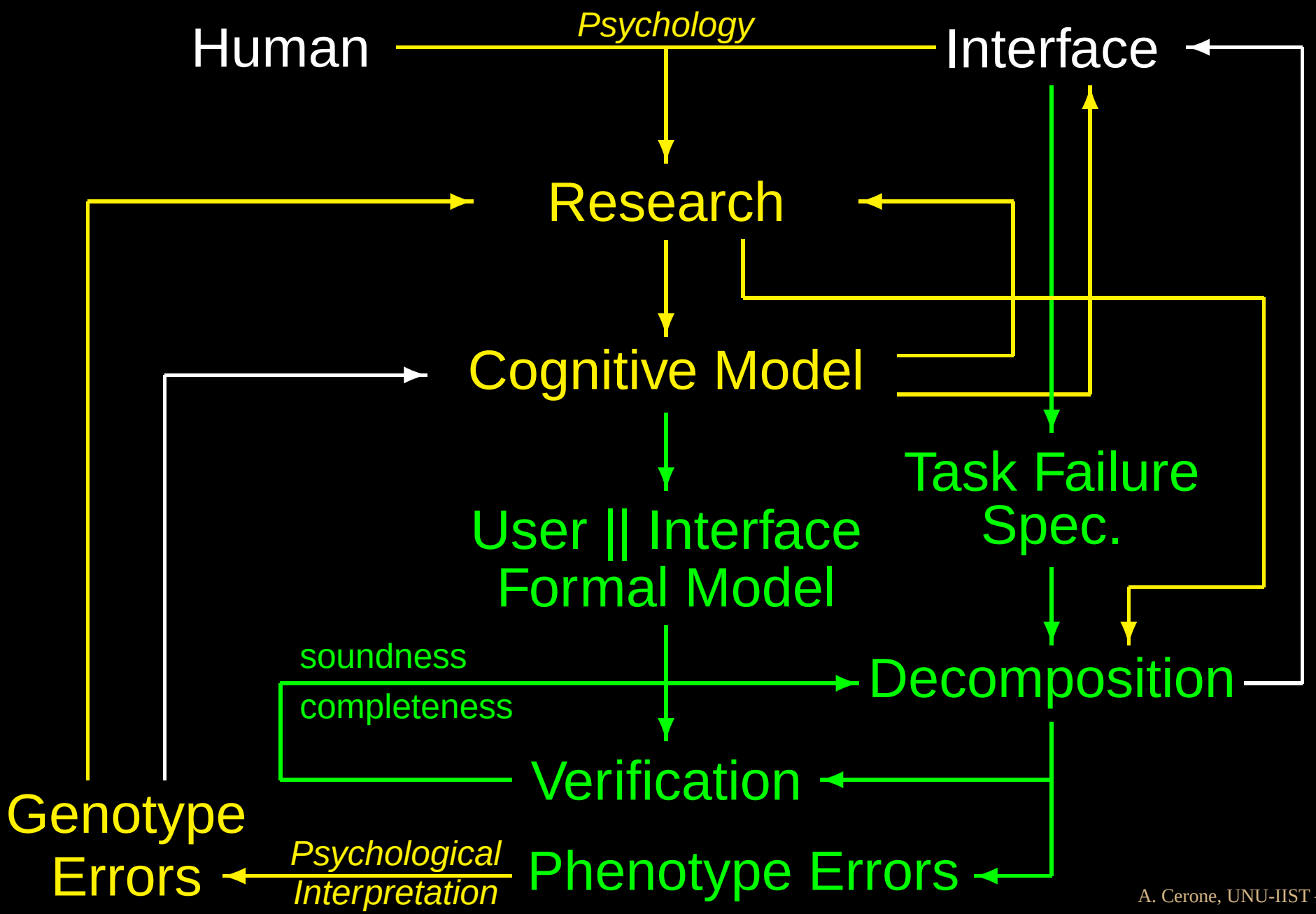
What caused such an error?

- use of the same action name n to denote the results of two cognitive processes
- aim at an elegant and easy to understand (to psychologists) formal model

Error Cause

What caused such an error?

- use of the same action name n to denote the results of two cognitive processes
- aim at an elegant and easy to understand (to psychologists) formal model
  $\Longrightarrow$ focus on syntactical look of formulae rather than on their interpretation on the model

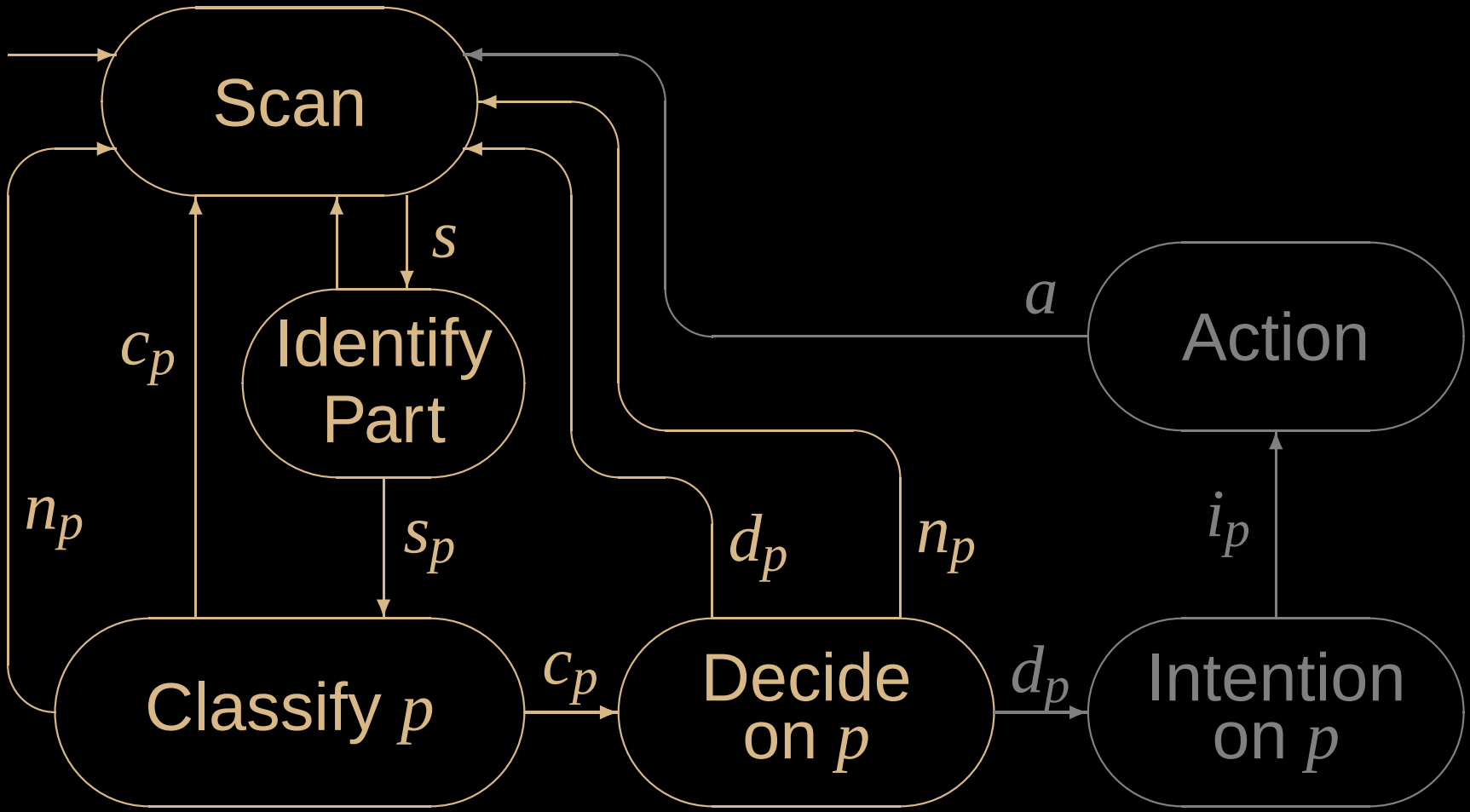

Human
Psychology
Interface
Research
Cognitive Model
Task Failure Spec.
User || Interface Formal Model
soundness
completeness
Decomposition
Verification
Genotype Errors
Psychological Interpretation
Phenotype Errors

Exercise: Counterexample?

Find and analyse the counterexample which falsifies completeness

Exercise: Counterexample?

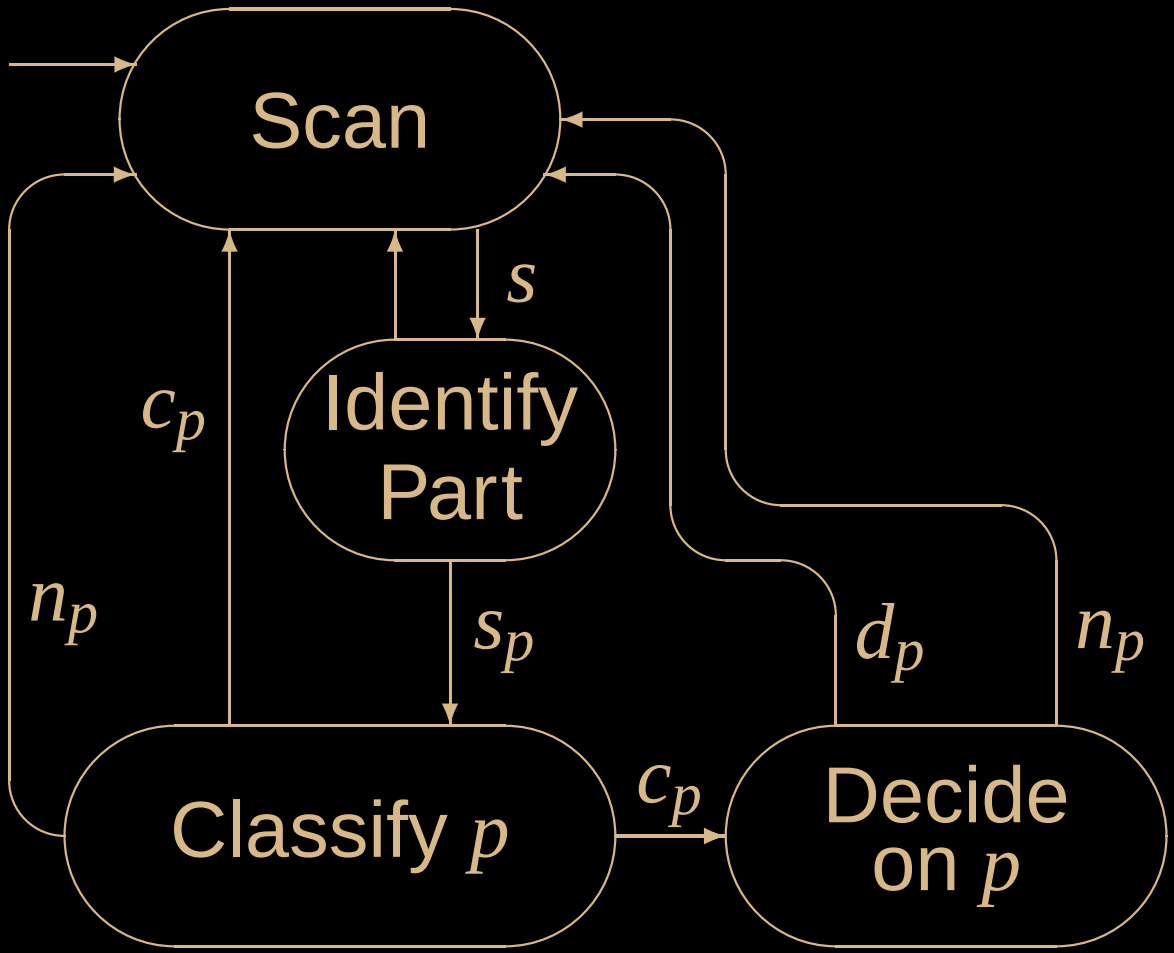
Find and analyse the counterexample



Exercise: Counterexample?

Find and analyse the counterexample

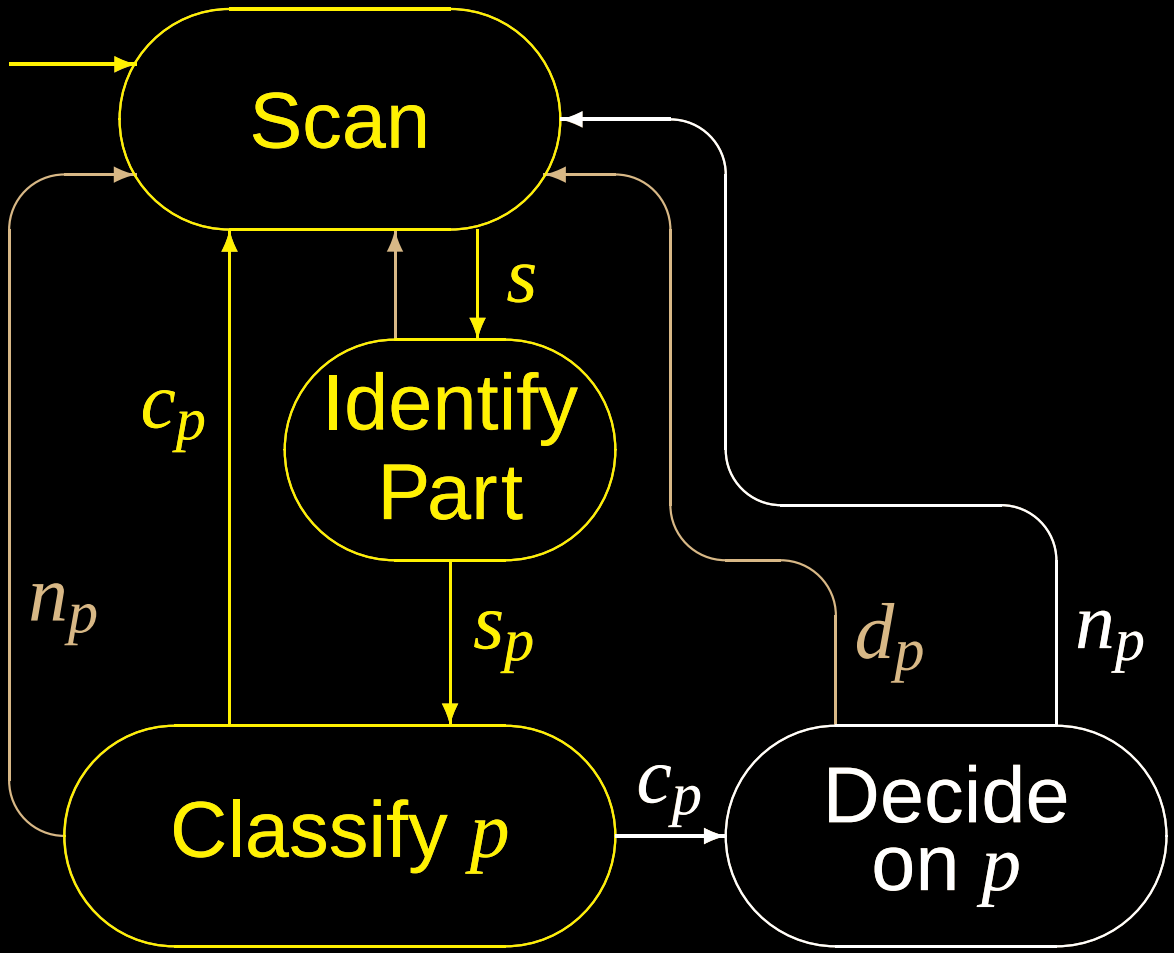
$$s \longrightarrow s_p \longrightarrow c_p \longrightarrow s \longrightarrow s_p \longrightarrow c_p \longrightarrow n_p \longrightarrow s \longrightarrow \ldots$$



Exercise: Counterexample?

Find and analyse the counterexample

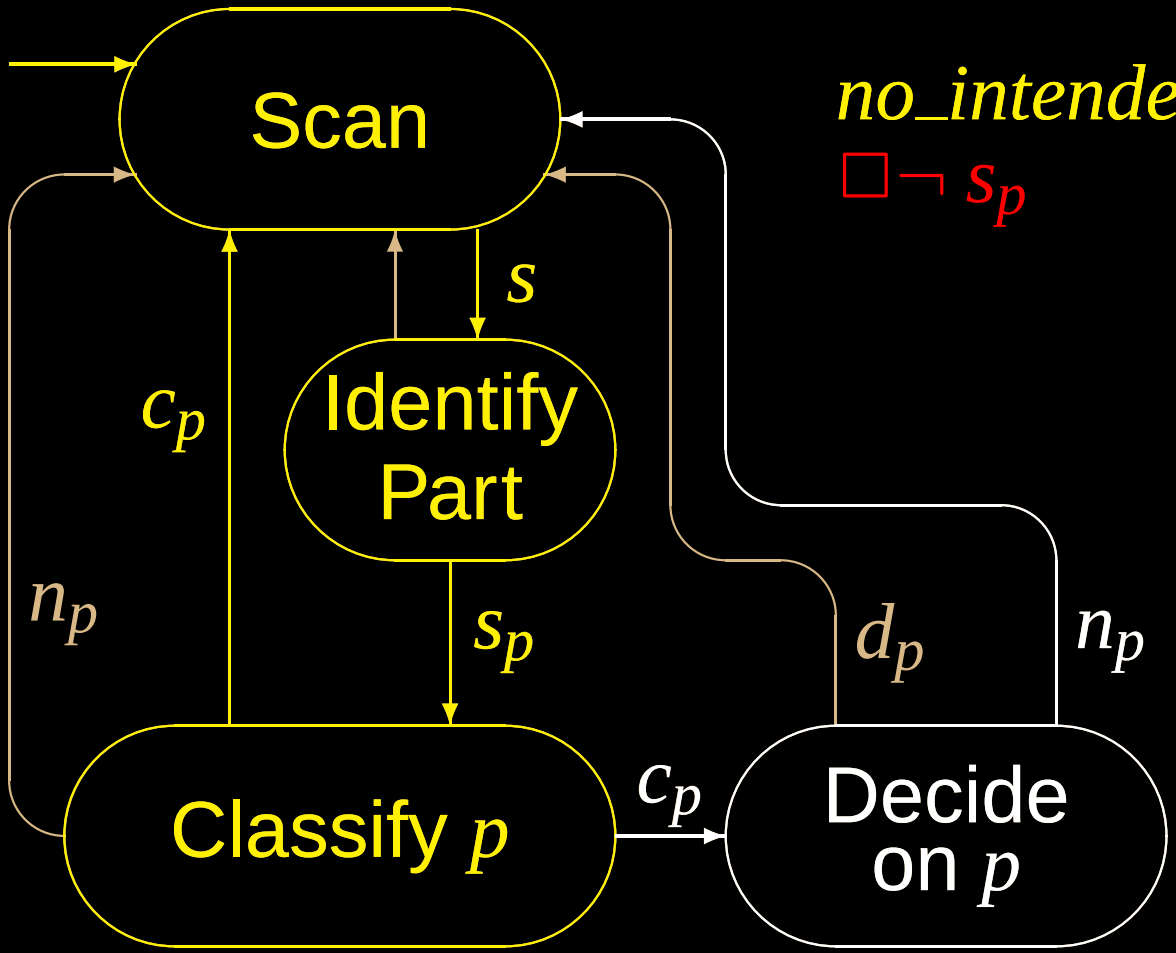
$$s \longrightarrow s_p \longrightarrow c_p \longrightarrow s \longrightarrow s_p \longrightarrow c_p \longrightarrow n_p \longrightarrow s \longrightarrow \ldots$$



Exercise: Counterexample?

Find and analyse the counterexample

$$s \longrightarrow s_p \longrightarrow c_p \longrightarrow s \longrightarrow s_p \longrightarrow c_p \longrightarrow n_p \longrightarrow s \longrightarrow \ldots$$



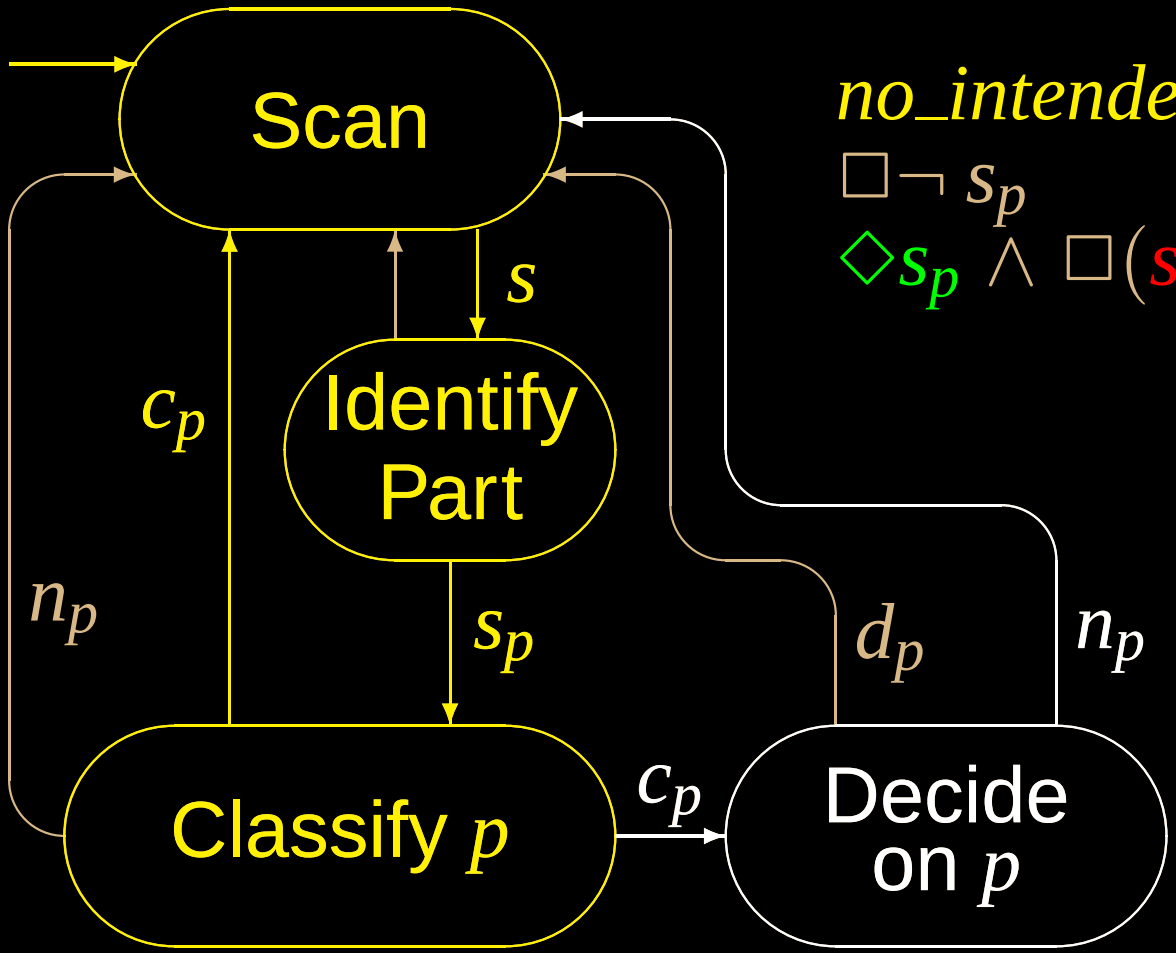
$no\_intended\_response_p$:

$\Box \neg s_p$

Exercise: Counterexample?

Find and analyse the counterexample

$$s \longrightarrow s_p \longrightarrow c_p \longrightarrow s \longrightarrow s_p \longrightarrow c_p \longrightarrow n_p \longrightarrow s \longrightarrow \ldots$$



$no\_intended\_response_p$:

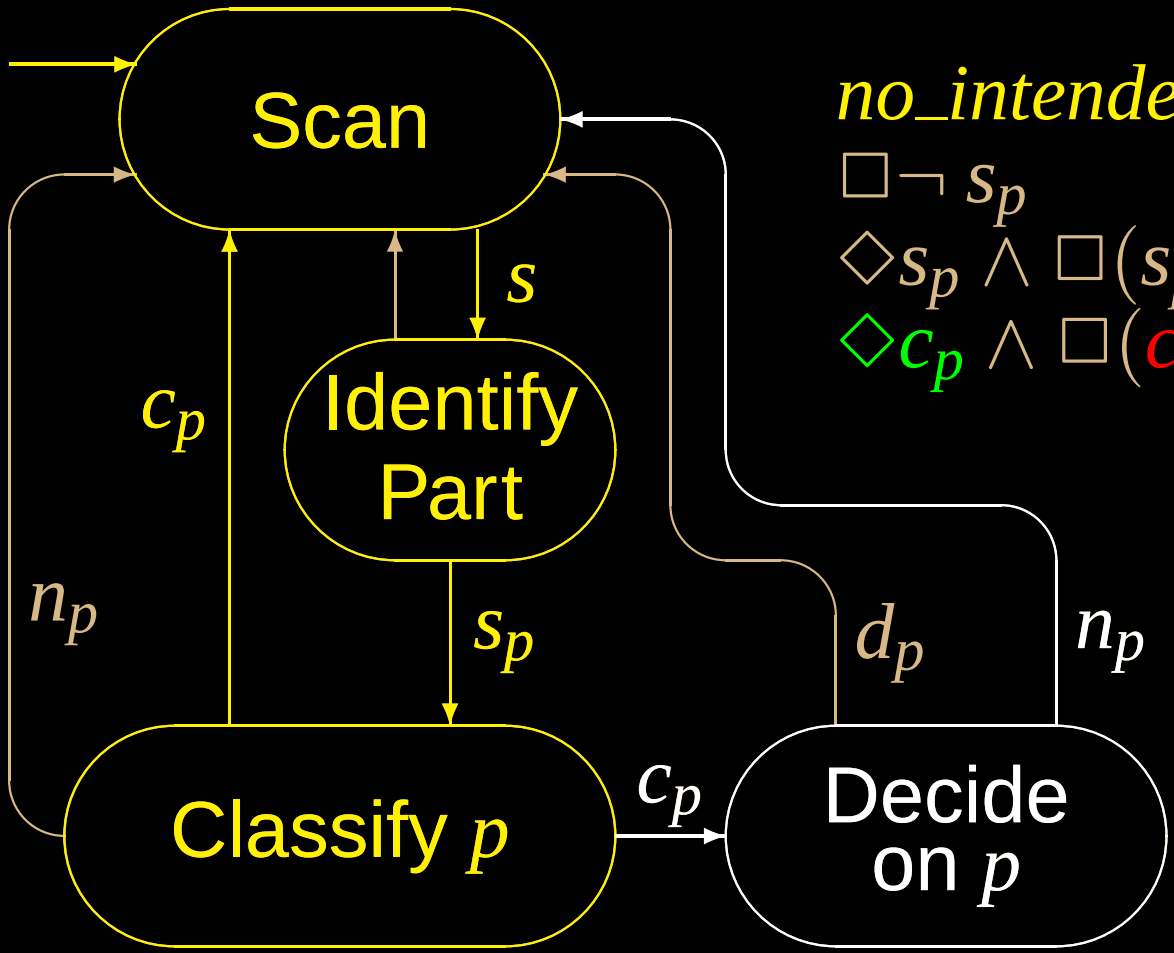
$\Box \neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

Exercise: Counterexample?

Find and analyse the counterexample

$$s \longrightarrow s_p \longrightarrow c_p \longrightarrow s \longrightarrow s_p \longrightarrow c_p \longrightarrow n_p \longrightarrow s \longrightarrow \ldots$$



$no\_intended\_response_p$:

$\Box \neg s_p$

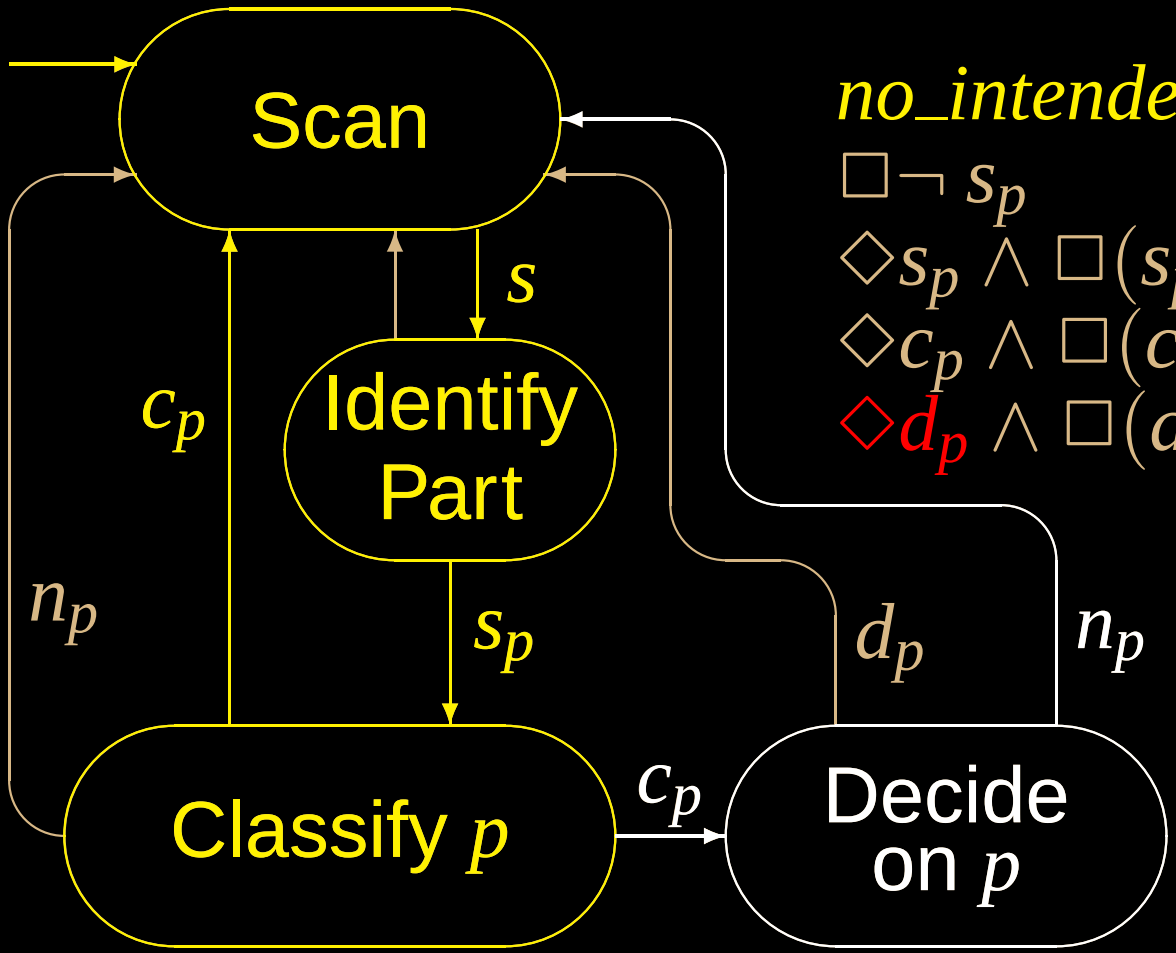
$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

Exercise: Counterexample?

Find and analyse the counterexample

$$s \longrightarrow s_p \longrightarrow c_p \longrightarrow s \longrightarrow s_p \longrightarrow c_p \longrightarrow n_p \longrightarrow s \longrightarrow \ldots$$



$no\_intended\_response_p$:

$\Box\neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

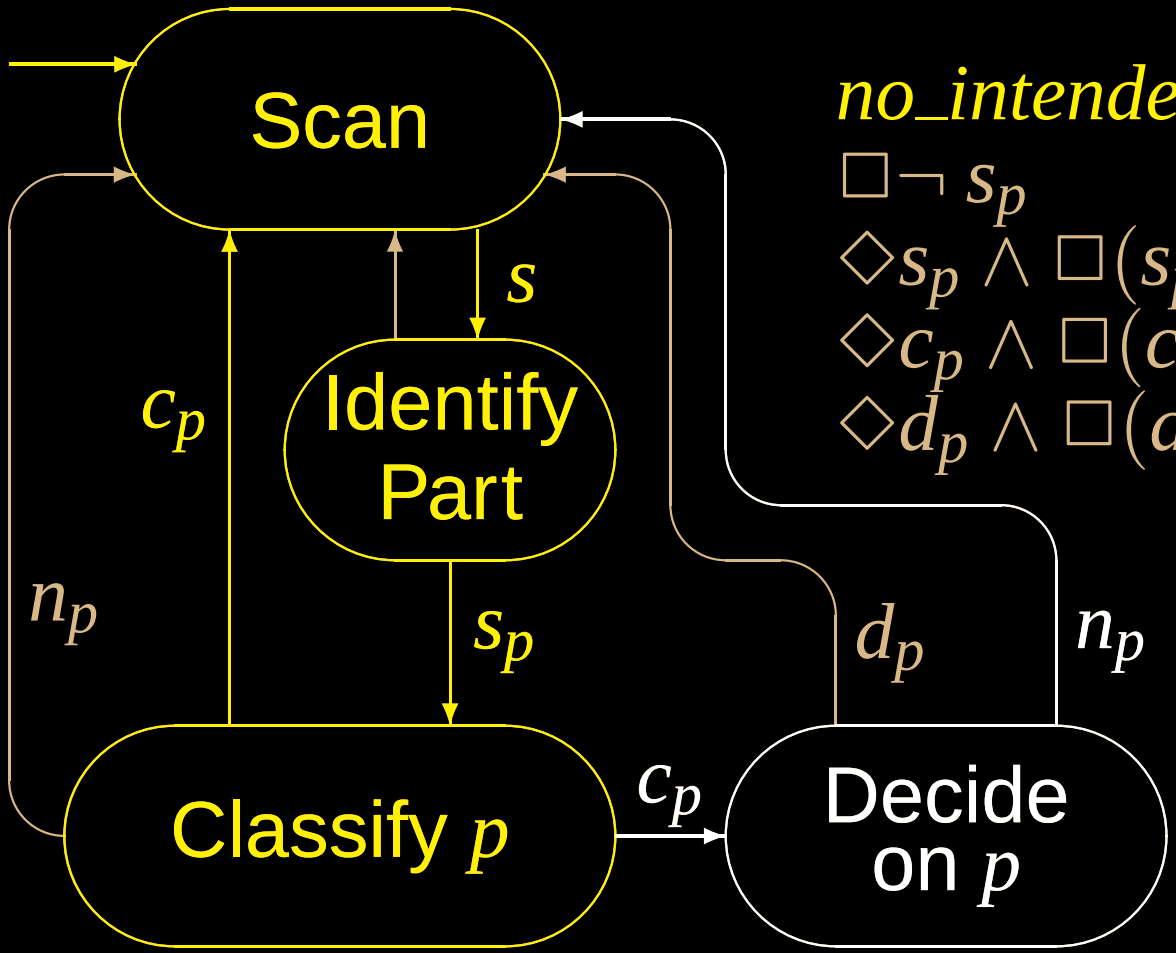
$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$

Exercise: Counterexample?

Find and analyse the counterexample

$$s \longrightarrow s_p \longrightarrow c_p \longrightarrow s \longrightarrow s_p \longrightarrow c_p \longrightarrow n_p \longrightarrow s \longrightarrow \ldots$$



$no\_intended\_response_p$:

$\Box \neg\, s_p$

$\Diamond s_p \wedge \Box(s_p \vee c_p \rightarrow \bigcirc n_p)$

$\Diamond c_p \wedge \Box(c_p \rightarrow \bigcirc s)$

$\Diamond d_p \wedge \Box(d_p \rightarrow \bigcirc s)$

References

[Paternò 00]

Fabio Paternò.
Model-based Design and Evaluation of Interactive Applications.
Springer, 2000.

Book
Introduces ConcurtaskTrees.

[Lindsay et. al. 02]

P. Lindsay and S. Connelly.
Modelling Erroneous Operator Behaviour for an Air-traffic Control Task.
AUIC 2002, Conferences in Research and Practice in Information Technology, Vol. 7, Australian Computer Society, pages 43–54.

Formal Methods Paper
Incorrect decomposition for the ATC Example.

[Cerone et al. 05 and 08]

A. Cerone, P. Lindsay and S. Connelly.
Formal Analysis of Human-computer Interaction using Model-checking.
SEFM 2005, IEEE Comp. Soc., 2005, pages 352–361.

A. Cerone, S. Connelly and P. Lindsay.
Formal Analysis of Human Operator Behavioural Patterns in Interactive Surveillance Systems.
Software and Systems Modeling, Vol.7, No. 3, Springer, 2008, pages 273–286.

Formal Methods Papers

On the correct decomposition for the ATC Example, but the second is the most complete.

End