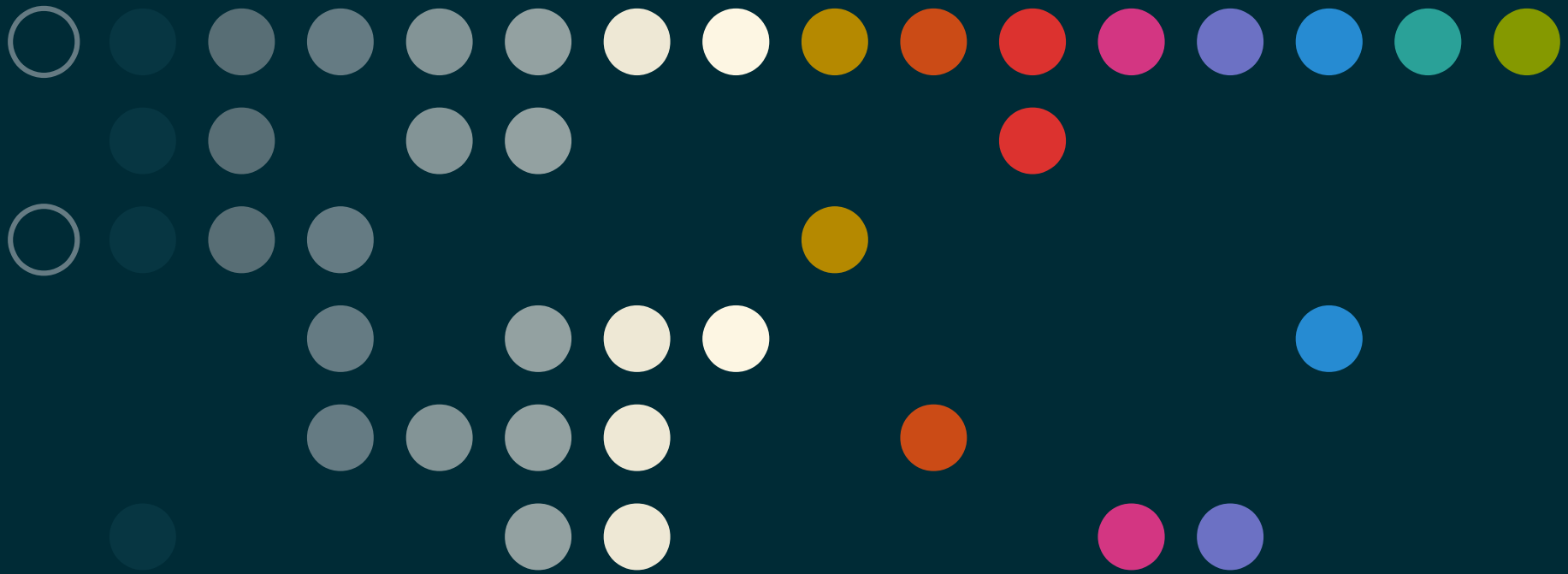
Endomorphic metalanguage and abstract planning for real-time intent recognition

Antoine Gréa

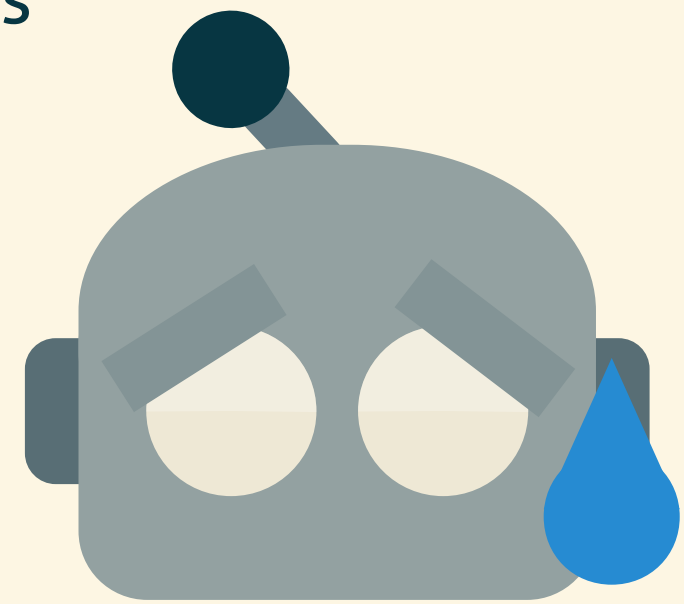
LIRIS Lyon 1 Lyon 2 POLYTECH

1 Introduction



A what ?

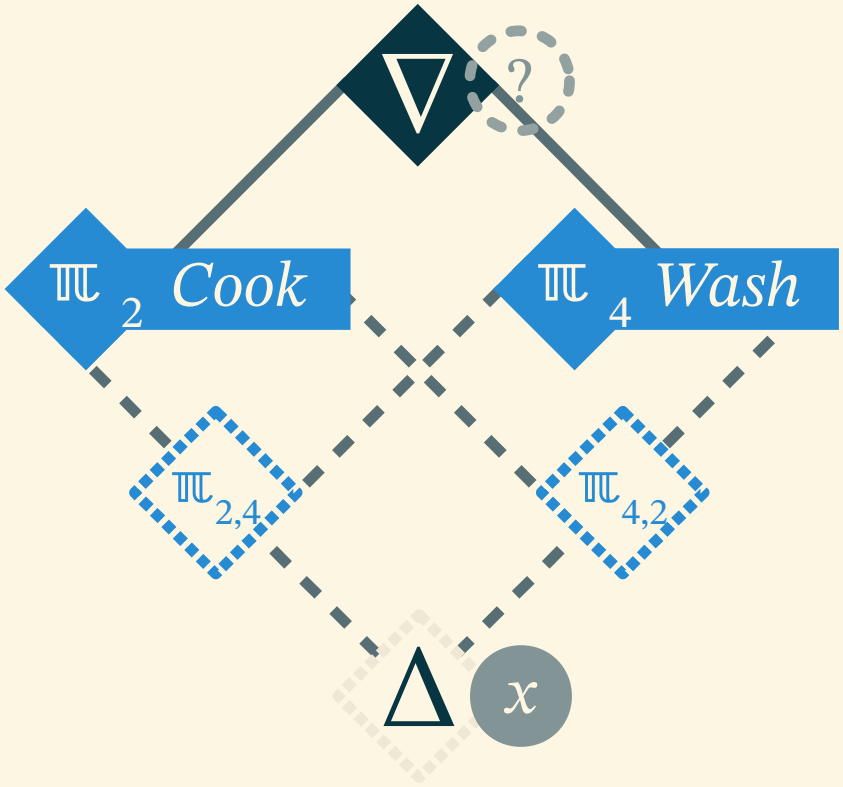
- *Dependent people need help !*
  - Not **annoying** the person
  - Can't see *everything* they are doing
- How to help without asking ?
  - Guessing the intent somehow
- **Intent recognition**
  - Observed behavior → Goal
  - Using action sequences: Plans

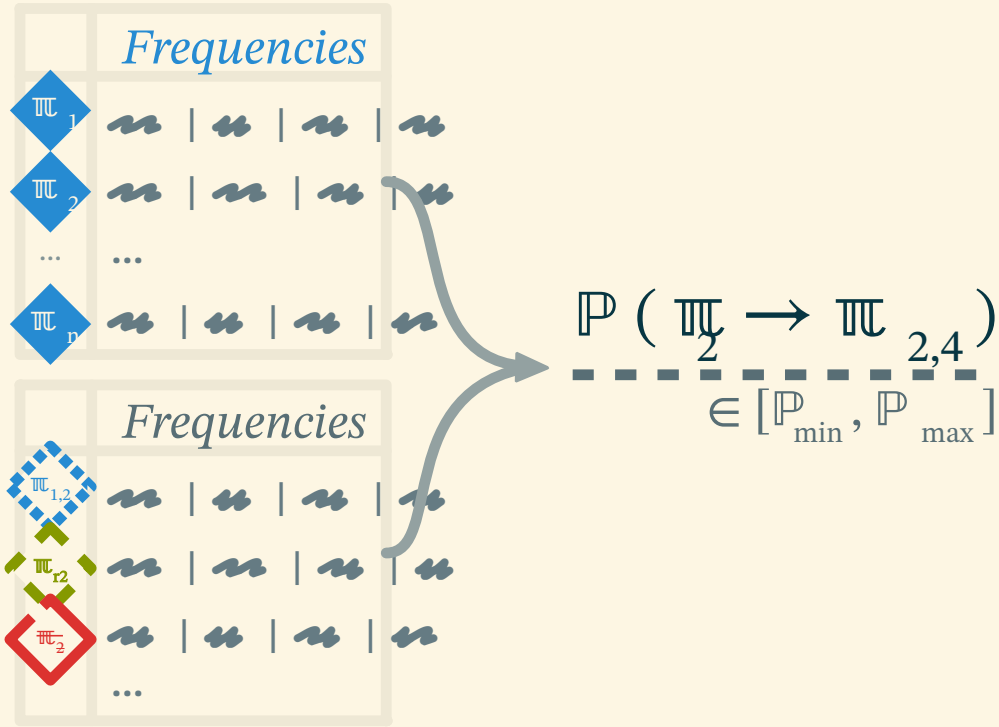


2 Intent Recognition

2.1 Logic Approach

- Lattice Based : ✔ Fast computations ✘ Exponential growth

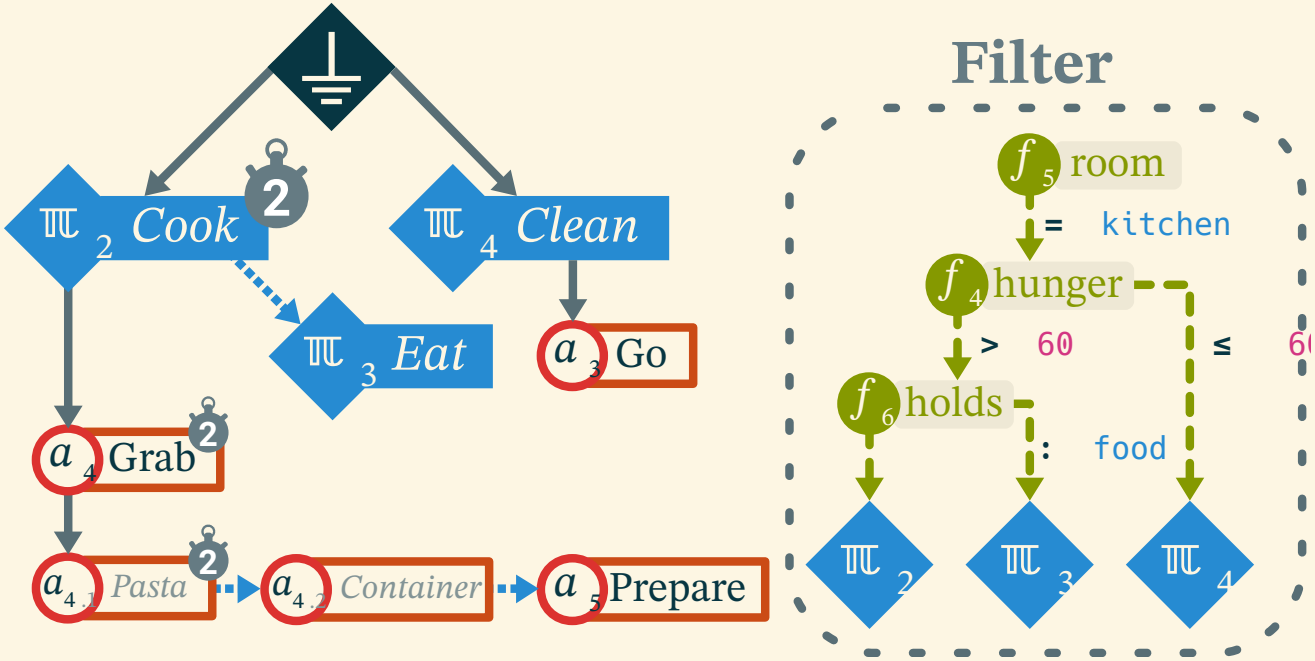




[@BOUCHARD_2006]

2.2 Stochastic Approach

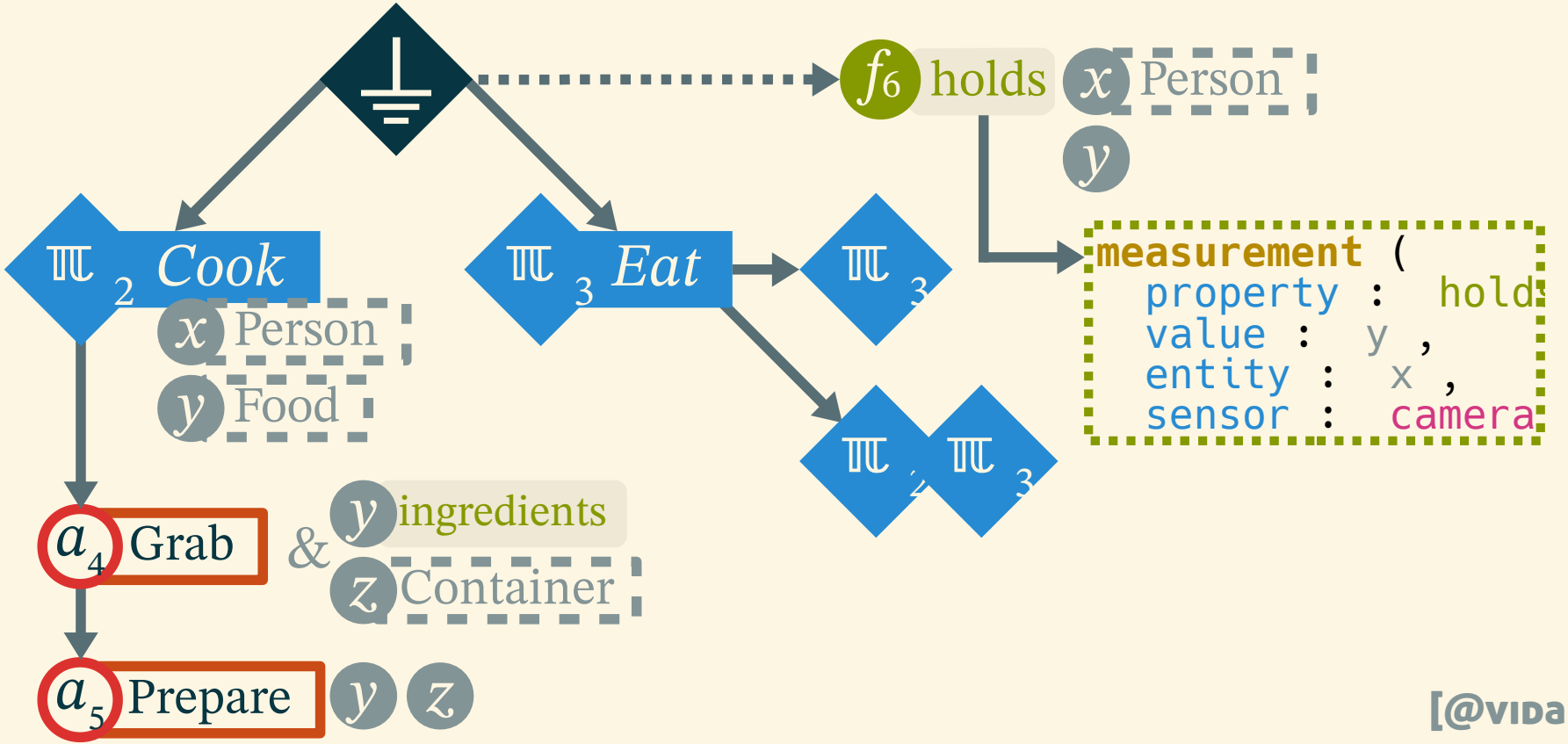
- And/Or and decision tree :
  - ✔ Accurate and efficient
  - ✘ Handmade plan library and tree



[@avrahami_2006]

2.3 Grammatical Approach

- Valued Grammar : ✔ Versatile ✘ Slow refresh rate (~40s)



[@VIDAL_2010]

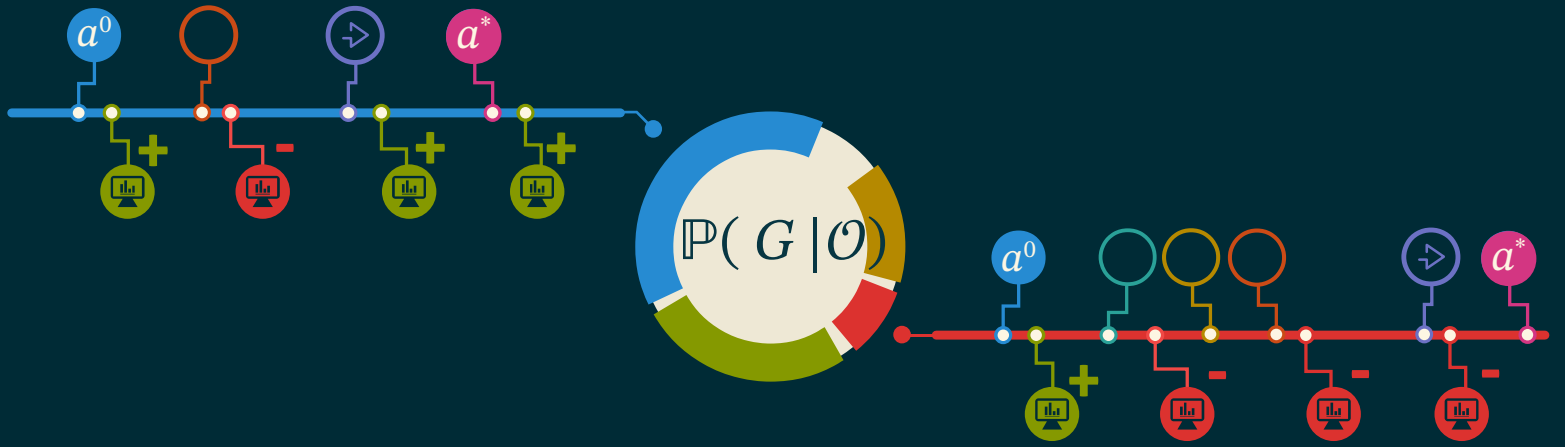
2.4 Invert Planning

- **Theory of Mind :**

> *"The easier the plan, the more likely the goal*

✔ Flexible

✘ More complex



Intent recognition

[@Ramirez_2008]

2.5 Framework Stacks

- Existing
- Contributions

Plan

1 Intent Recognition

2 Knowledge Representation

3 General Planning

4 Flexible Online Planning

5 Perspectives

6 Conclusion

3 Knowledge Representation

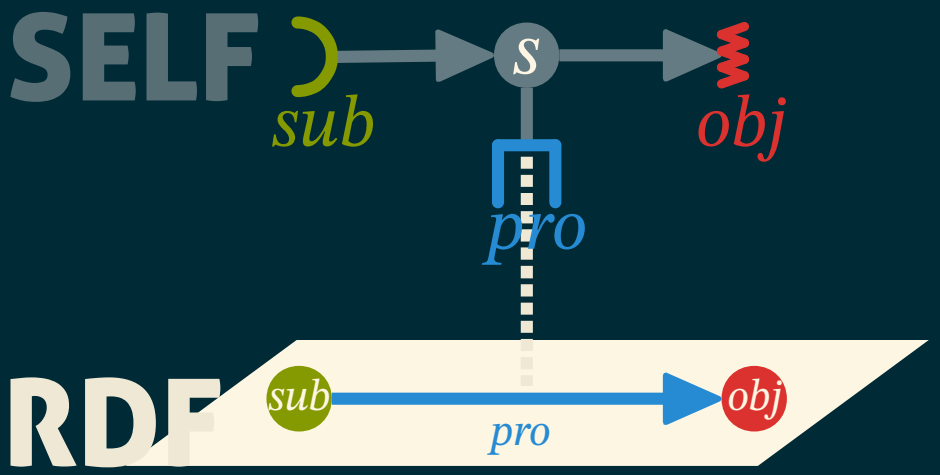
How to Know

- Abstraction
  - *How to **refer** to something*
- Formalization
  - *How to **talk** about something*
- Interpretation
  - *How to **know** about something*

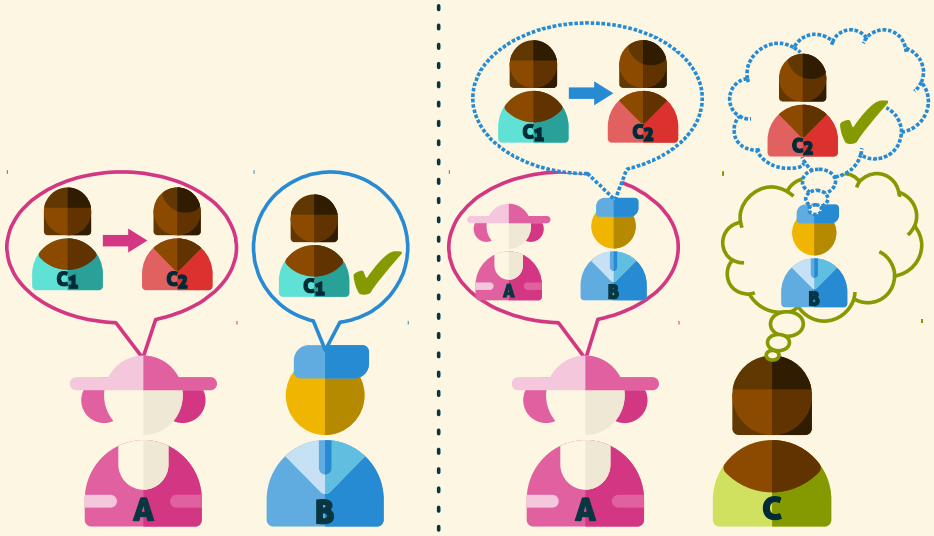
Issues Expressing Knowledge

- Abstraction
  - Incomplete information
- Formalization
  - Informal bases
- Interpretation
  - Non defined terms

SELF



Example of Modal Logic



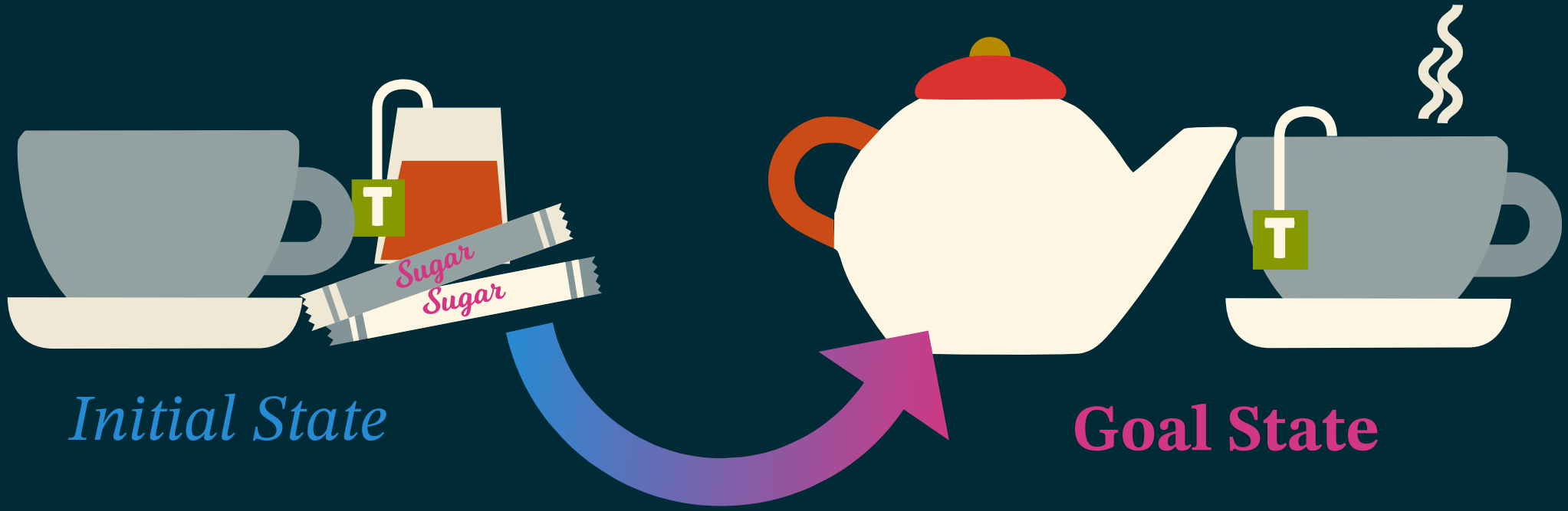
4 General Planning

Classical Planning

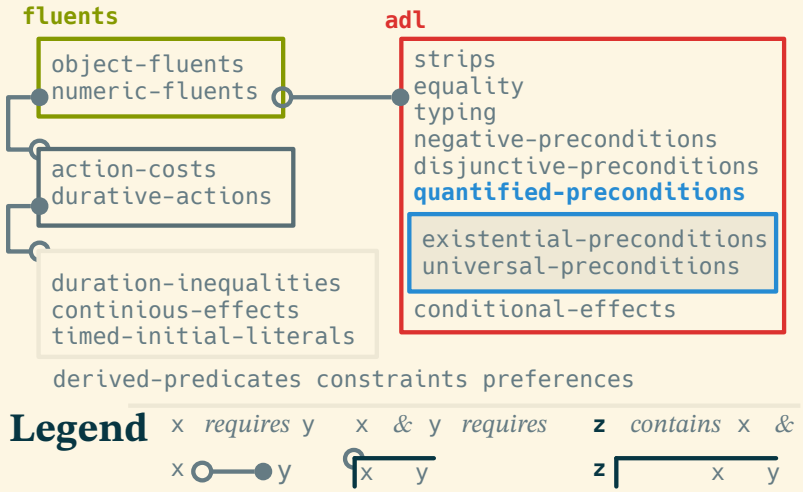
- Domain
  - Fluents
    - Formula over objects
  - States
    - Properties of the world
    - Formula over fluents
  - Actions
    - Precondition
    - Effects
- Problem
  - Initial state
  - Goal state
- Plan (solution)
  - Action sequence
  - Order
    - Total
    - Partial

Example

- Having some tea, aren't we ?

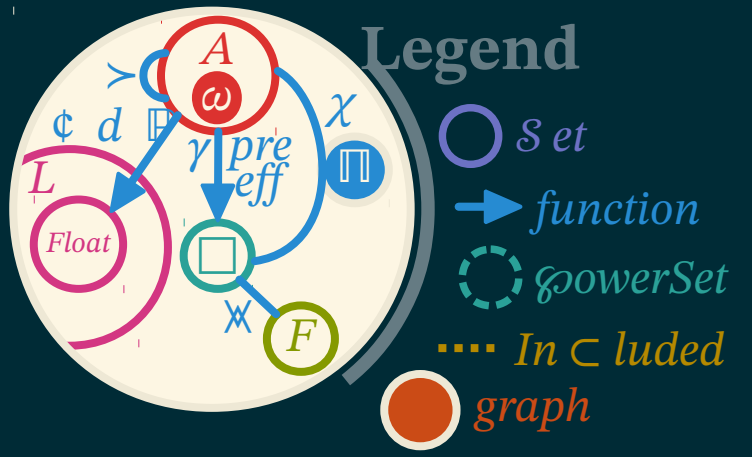


Existing Frameworks



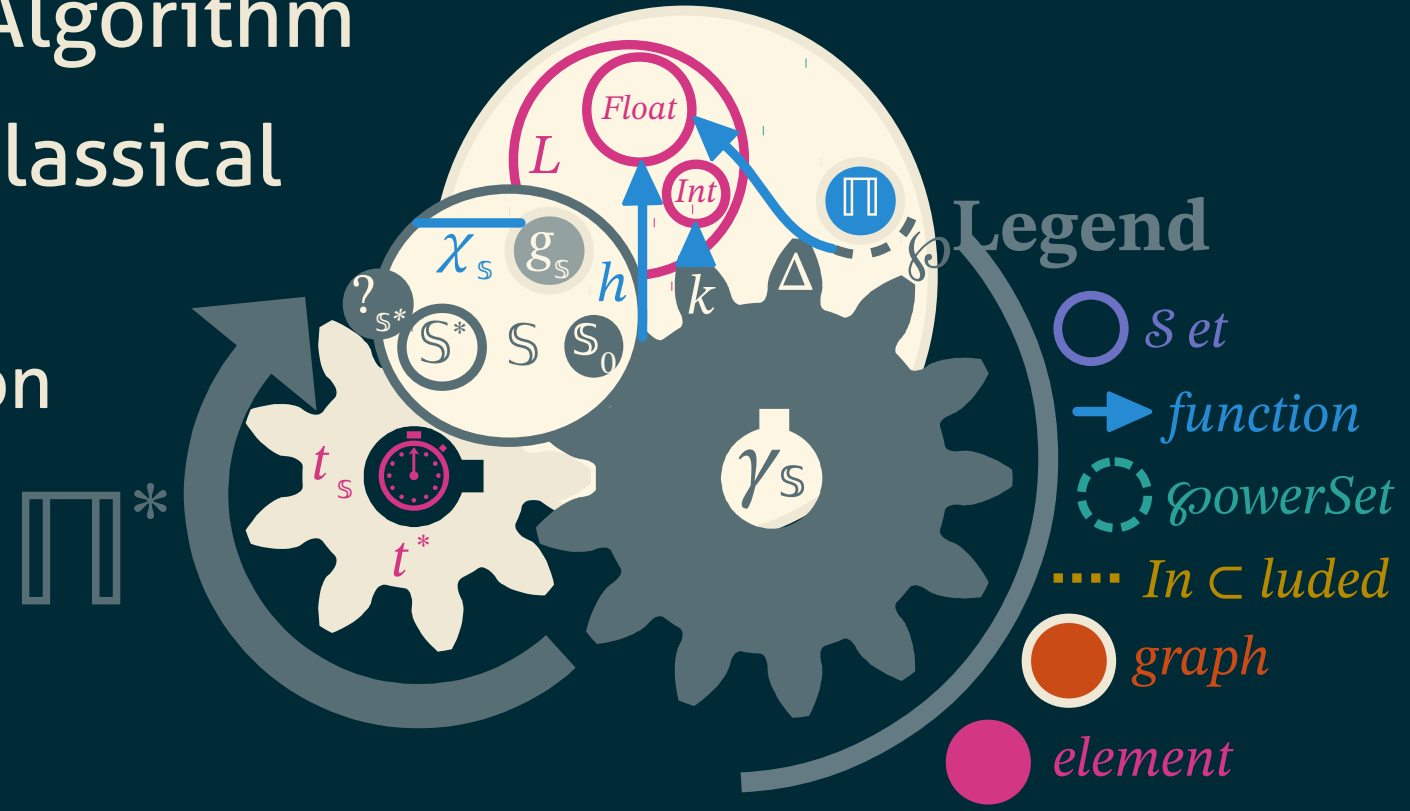
Planning Formalism Revisited

- States
  - And/Or trees of Fluents
  - Verifying
  - Applying
- Actions
  - Precondition, Effects
  - Constraints
  - Cost, Duration, Probability
  - Methods
- Search Space
  - Starting point
  - Iterator
  - Solution predicate



General Planning Algorithm

- Shortest Path Algorithm
- Instances for Classical Approaches
  - State-transition
  - Plan space
  - Case based
  - Probabilistic
  - Hierarchical



Π*

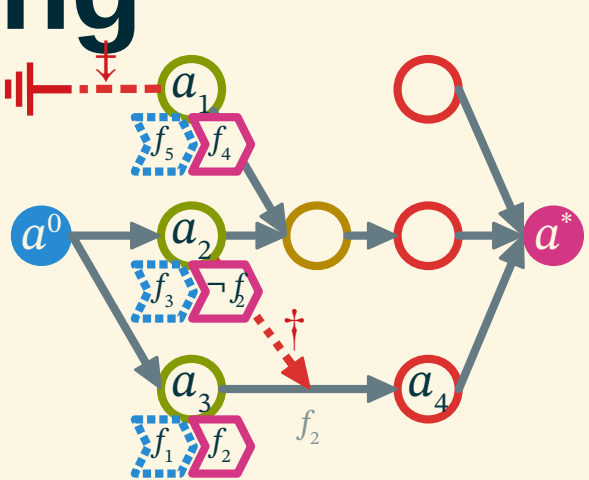
COLOR Framework

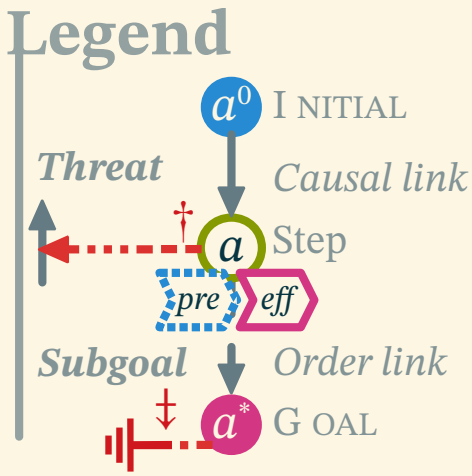
5 Flexible Online Planning

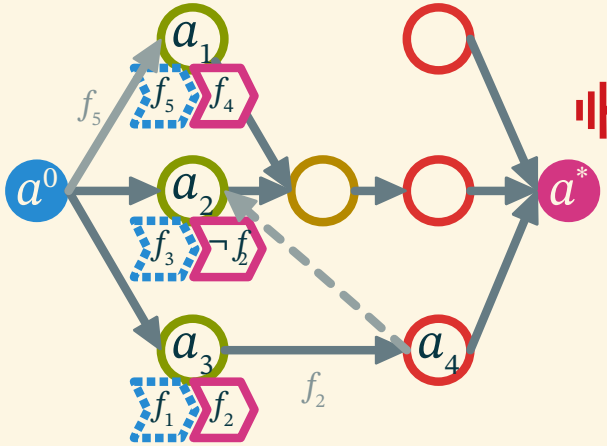
Planning Phases

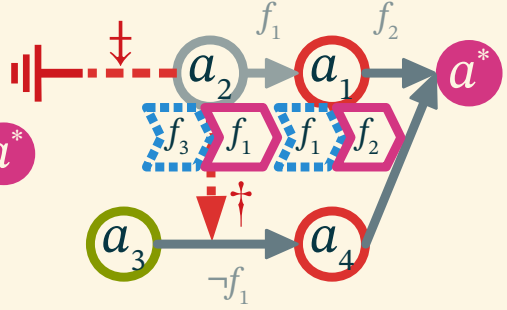
- Phases dependent on
  - Available information
  - Timing constraints
  - Planning paradigm

Plan Space Planning



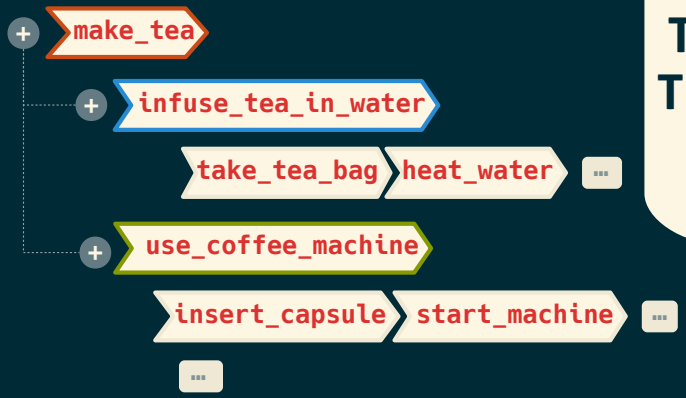




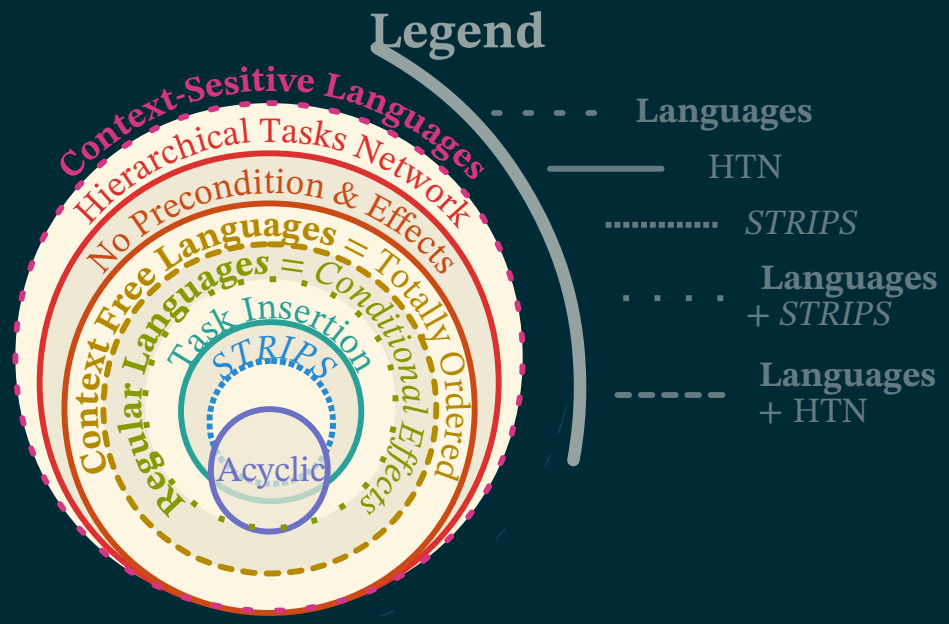


Hierarchical Task Networks

- Based on tasks
- Decomposition
- Vary in complexity

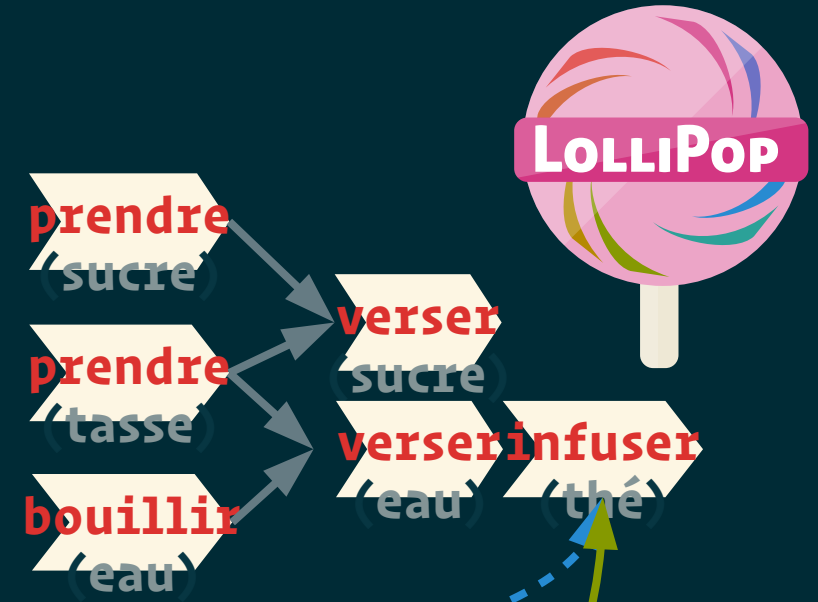


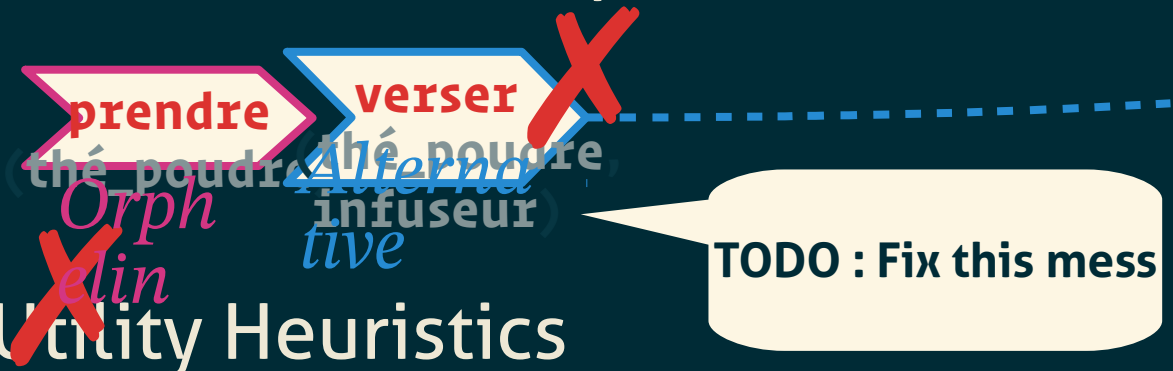
TODO : Citation of
The difference with
planning



Plan Repair Prototype

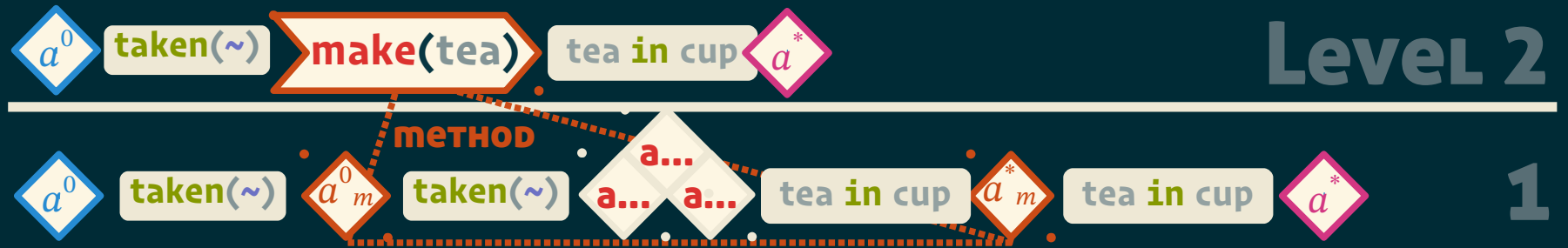
- Partial Order Planner (POP)
- Operator dependency graph
- Negative refinements
- Alternatives & Orphans
- Utility Heuristics



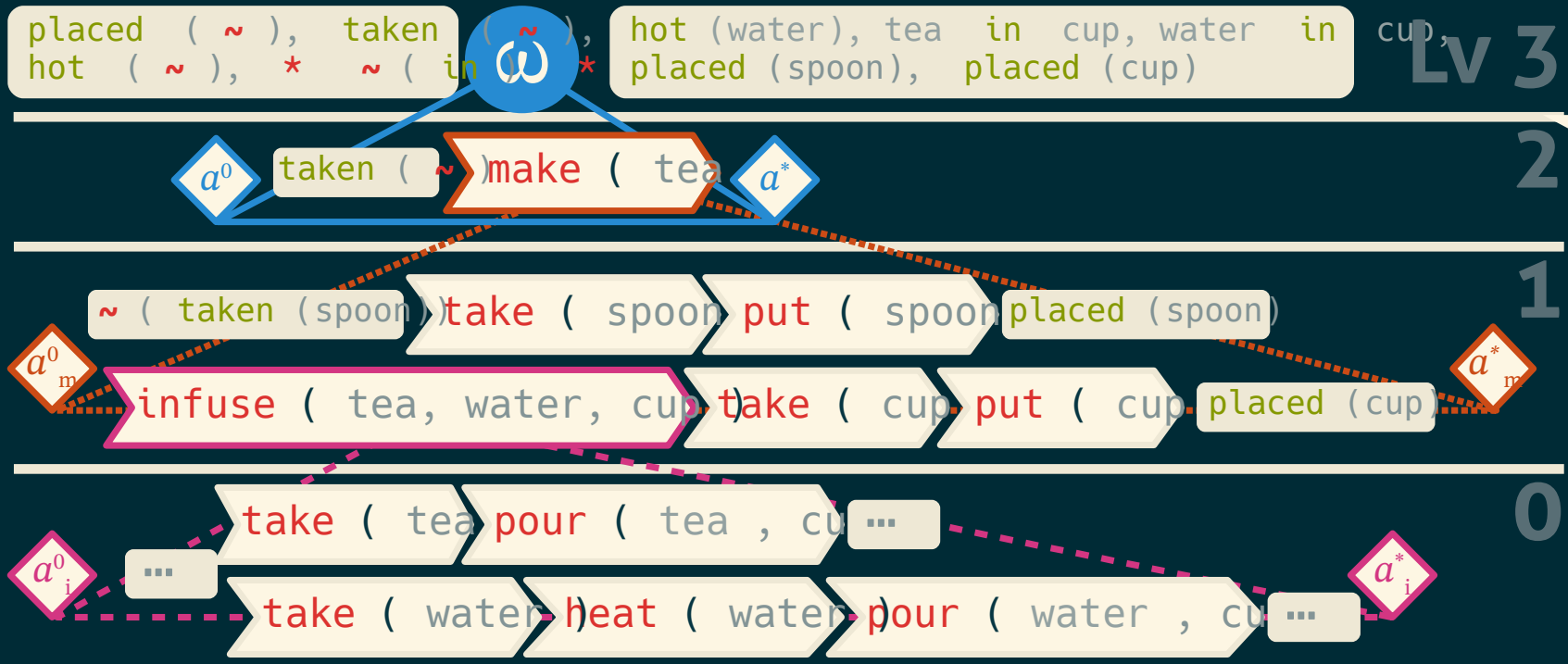


Abstract Planning

- HTN + POP planning
- Partial Resolution
  - An abstract solution at every level of abstraction
- Search by level
  - Expansion after completion :

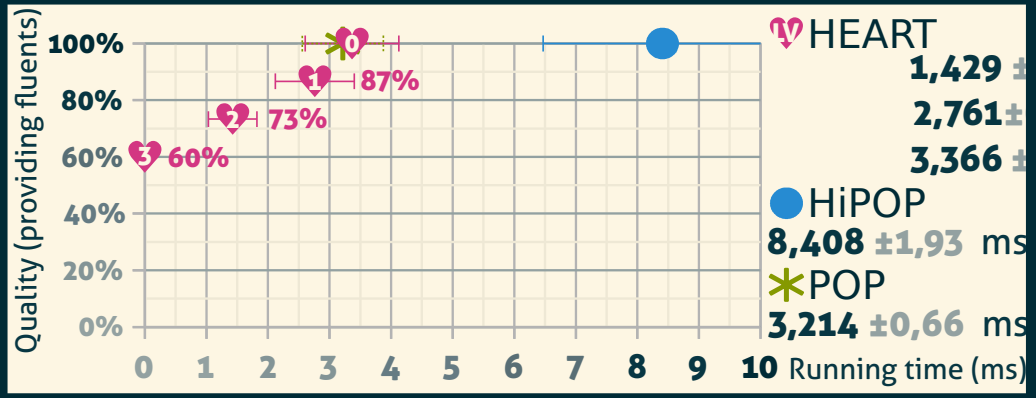


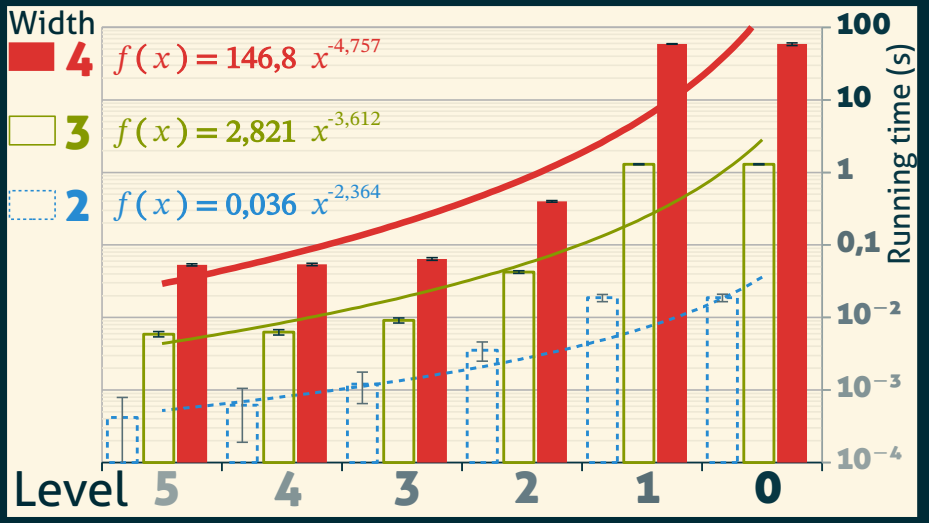
HEART


placed (~), taken (~),
hot (~), * ~ (in (*)
ω
hot (water), tea in cup, water in cup,
placed (spoon), placed (cup)
Lv 3
2
1
0
a^0
taken (~)
make (tea
a^*
~ (taken (spoon))
take (spoon
put (spoon
placed (spoon)
a^0_m
infuse (tea, water, cup
take (cup
put (cup
placed (cup)
a^*_m
take (tea
pour (tea , cu
…
a^0_i
…
take (water
heat (water)
pour (water , cu
…
a^*_i
TODO
Animate for
step by step

Results

- TODO





6 Conclusion

Contributions & Results

SELF Improvement

Planning Colorized

Fixing Planning Domains

Toward Intent Recognition

Thanks for listening !