



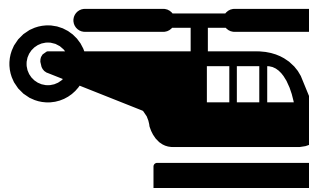
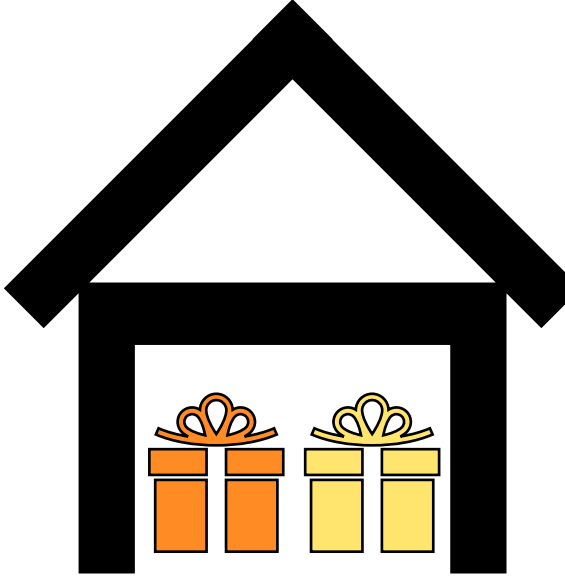
MAX PLANCK INSTITUTE
FOR SOFTWARE SYSTEMS

A Negotiation Framework for Distributed Reactive Synthesis

Rupak Majumdar
Kaushik Mallik
Anne-Kathrin Schmuck
Damien Zufferey

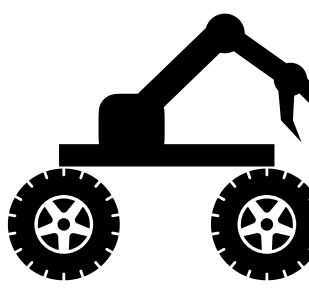
Max Planck Institute for Software Systems, Germany

A Distributed Synthesis Problem for Delivery Robots



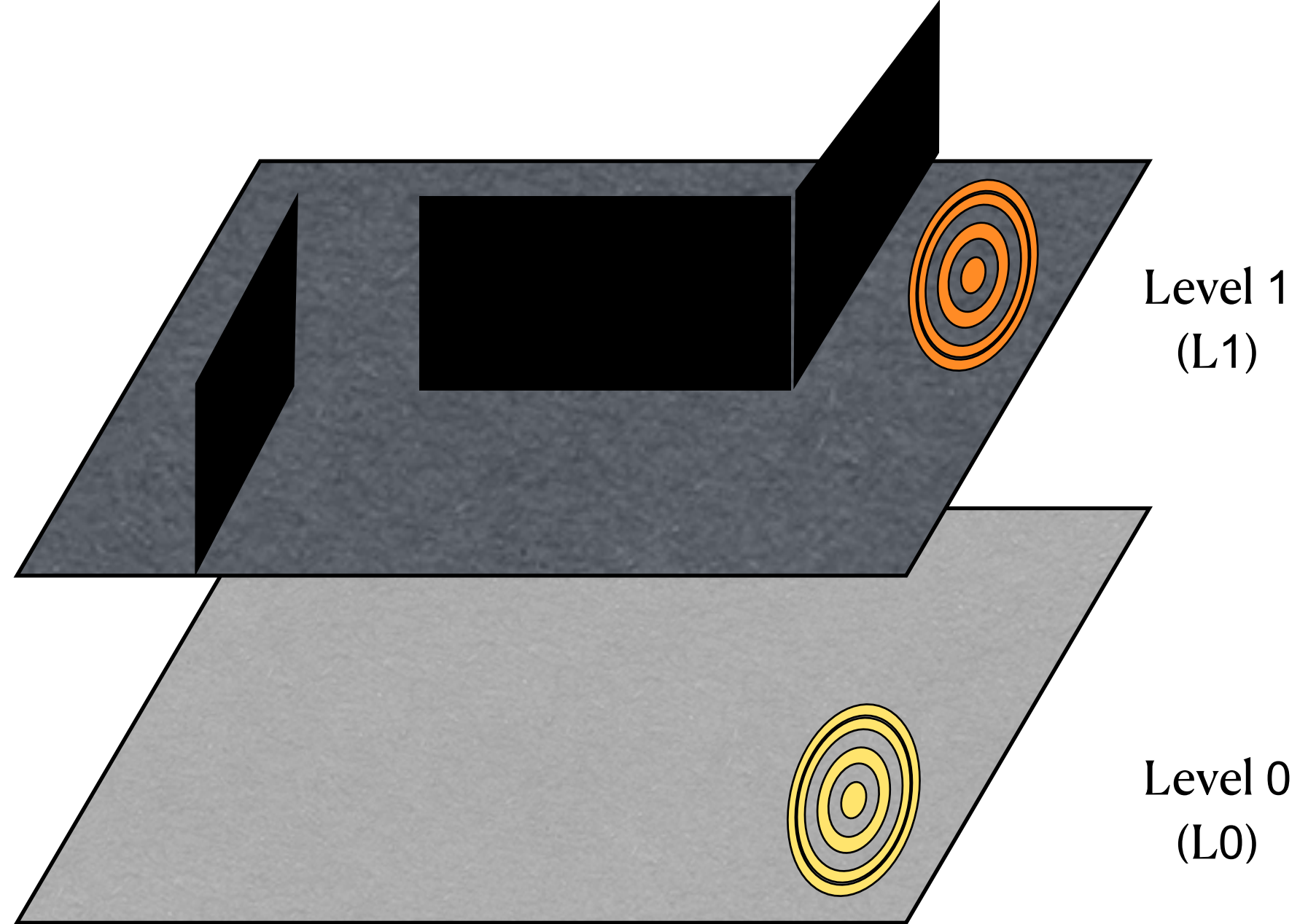
Drone

Task: Both boxes need to be delivered.

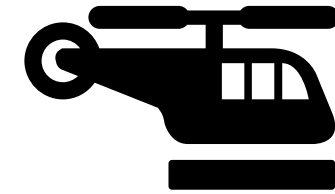
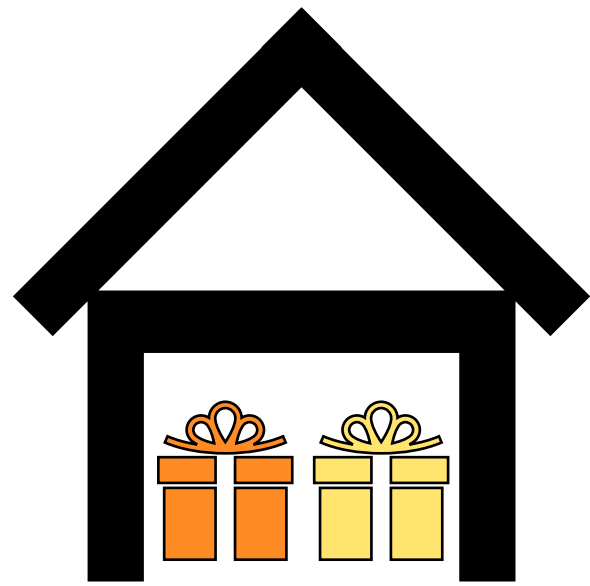


Vehicle

Task: Both boxes need to be delivered.

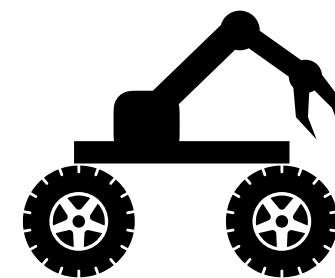


A Distributed Synthesis Problem for Delivery Robots



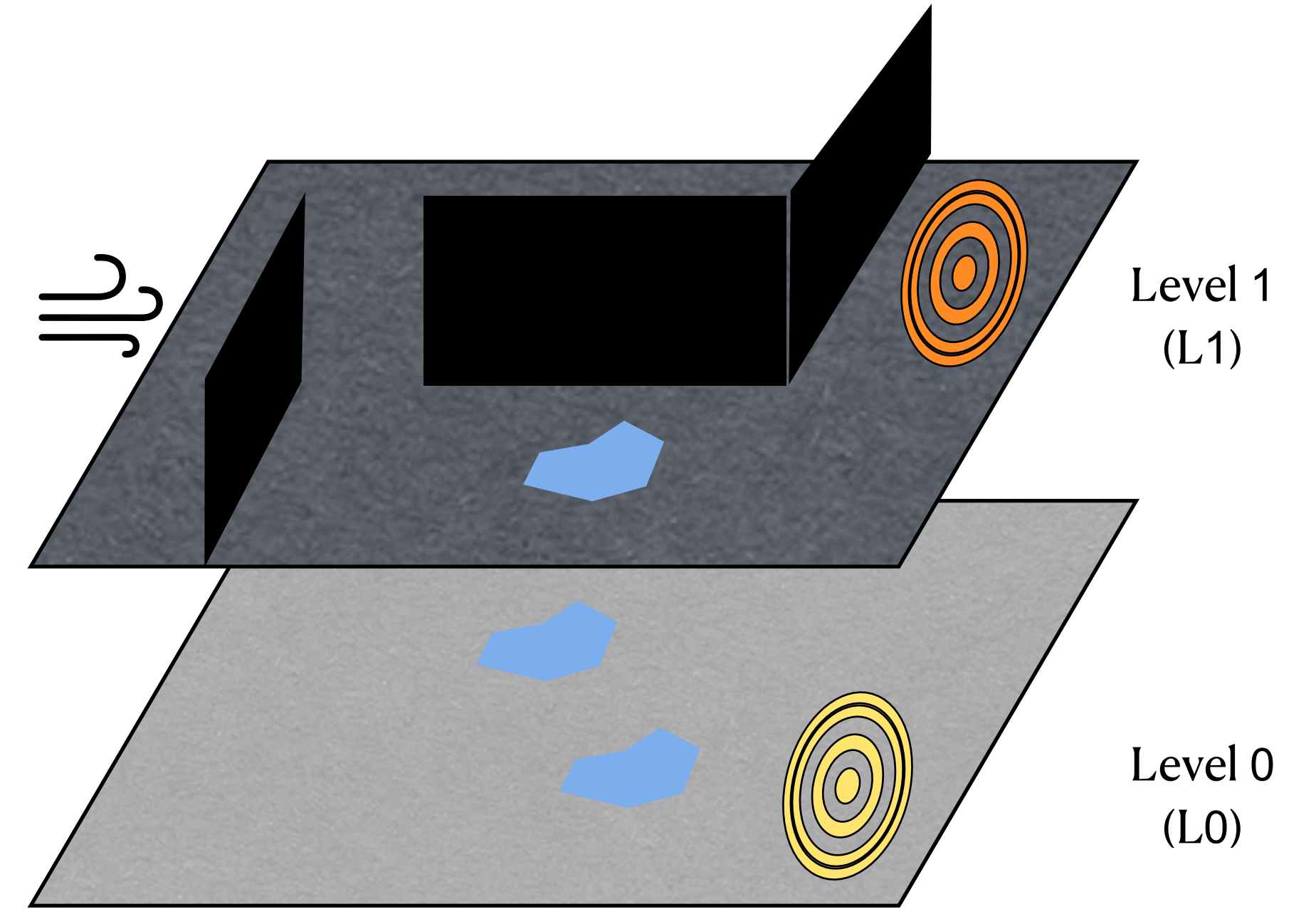
Drone

Task: Both boxes need to be delivered.

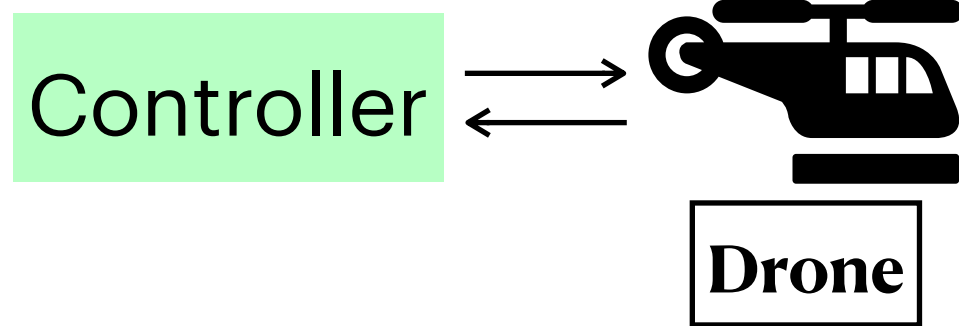


Vehicle

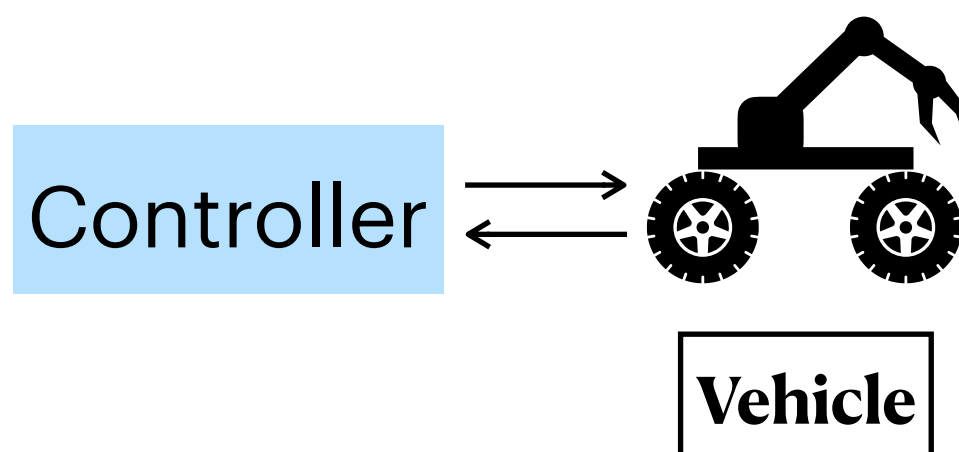
Task: Both boxes need to be delivered.



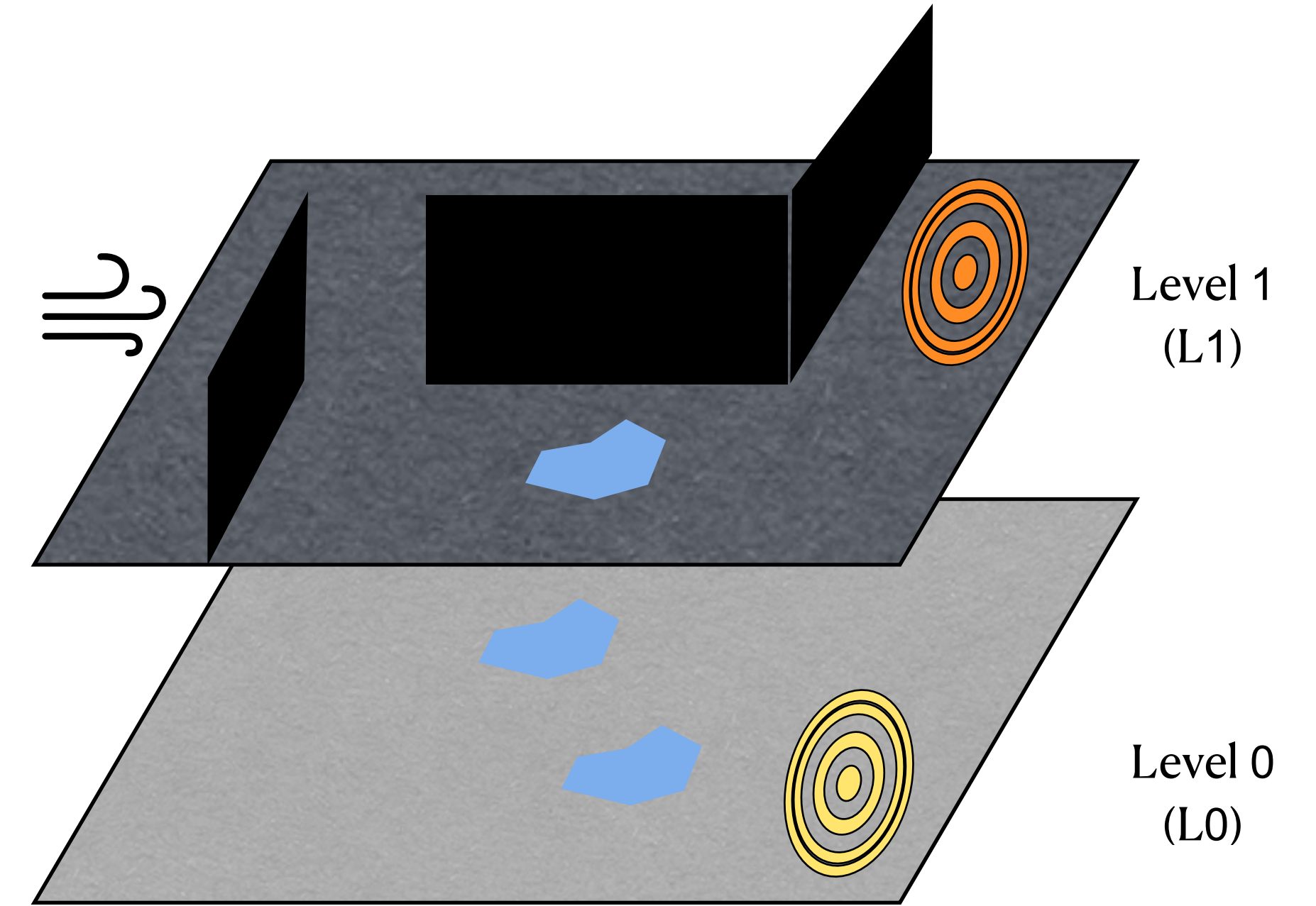
A Distributed Synthesis Problem for Delivery Robots



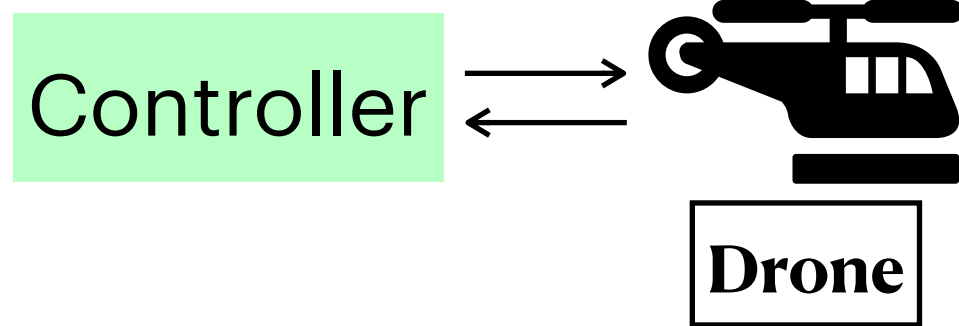
Task: Both boxes need to be delivered.



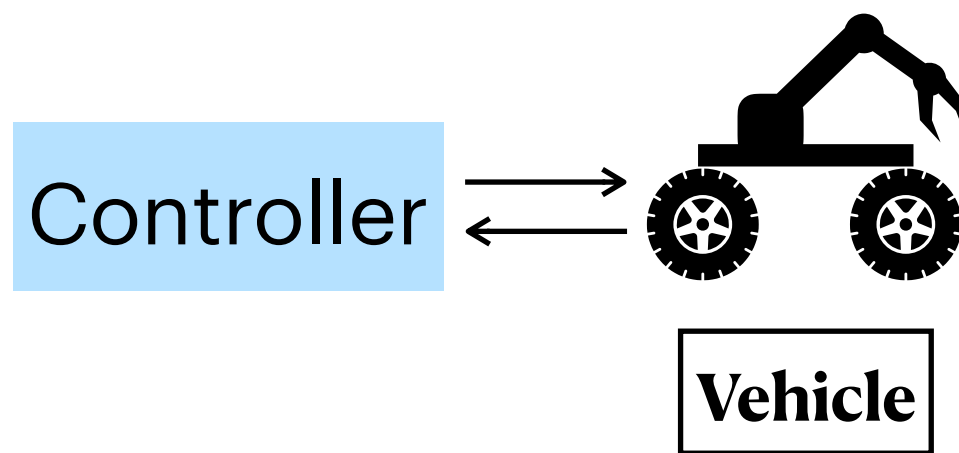
Task: Both boxes need to be delivered.



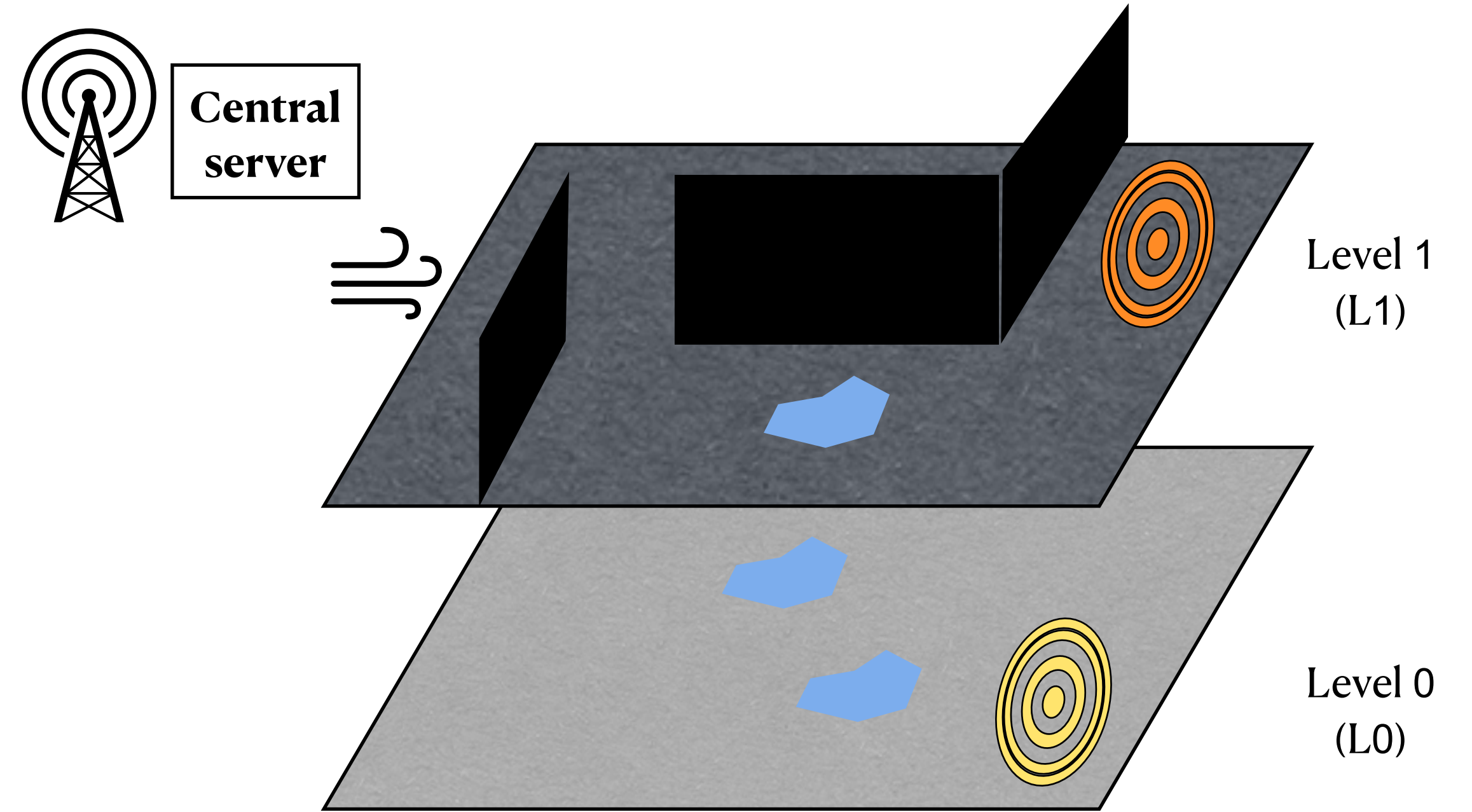
A Distributed Synthesis Problem for Delivery Robots



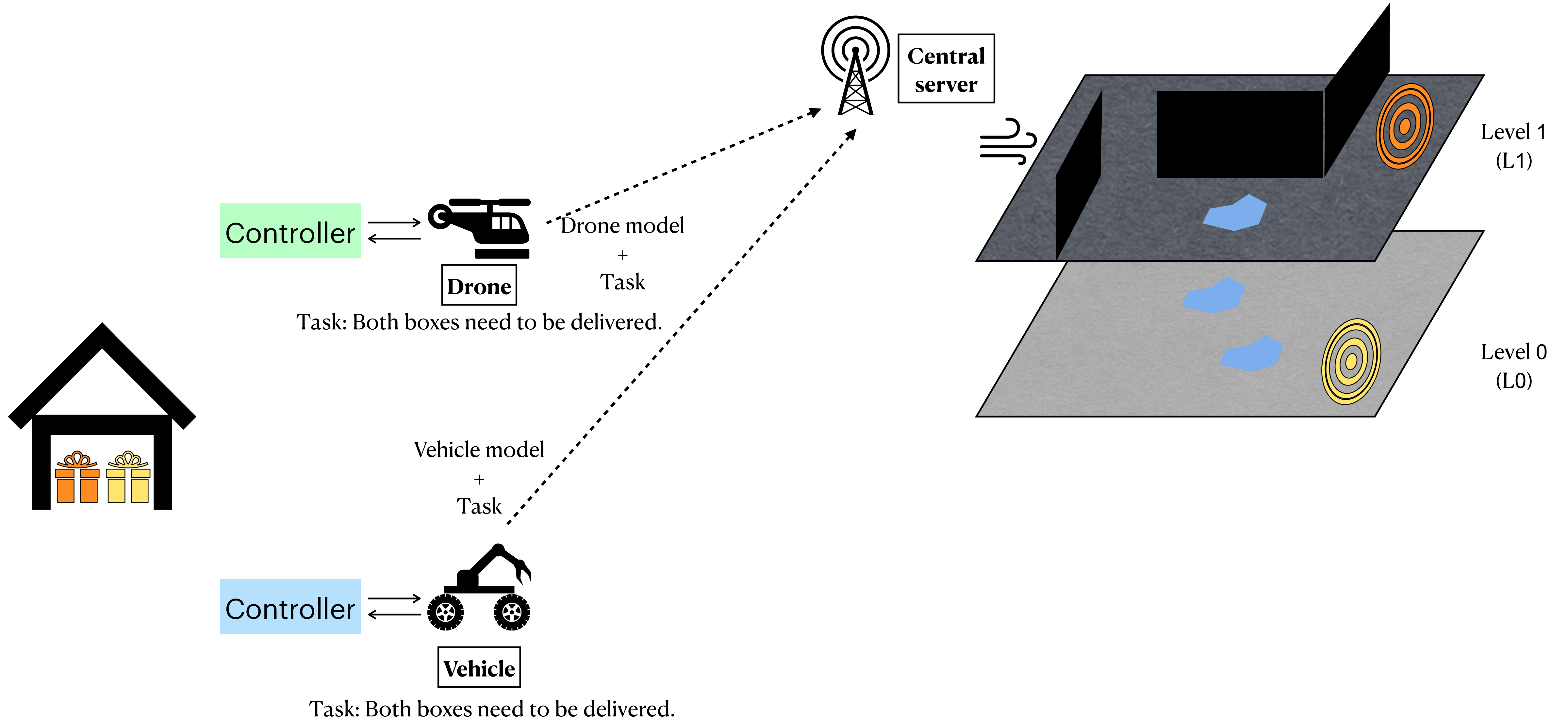
Task: Both boxes need to be delivered.



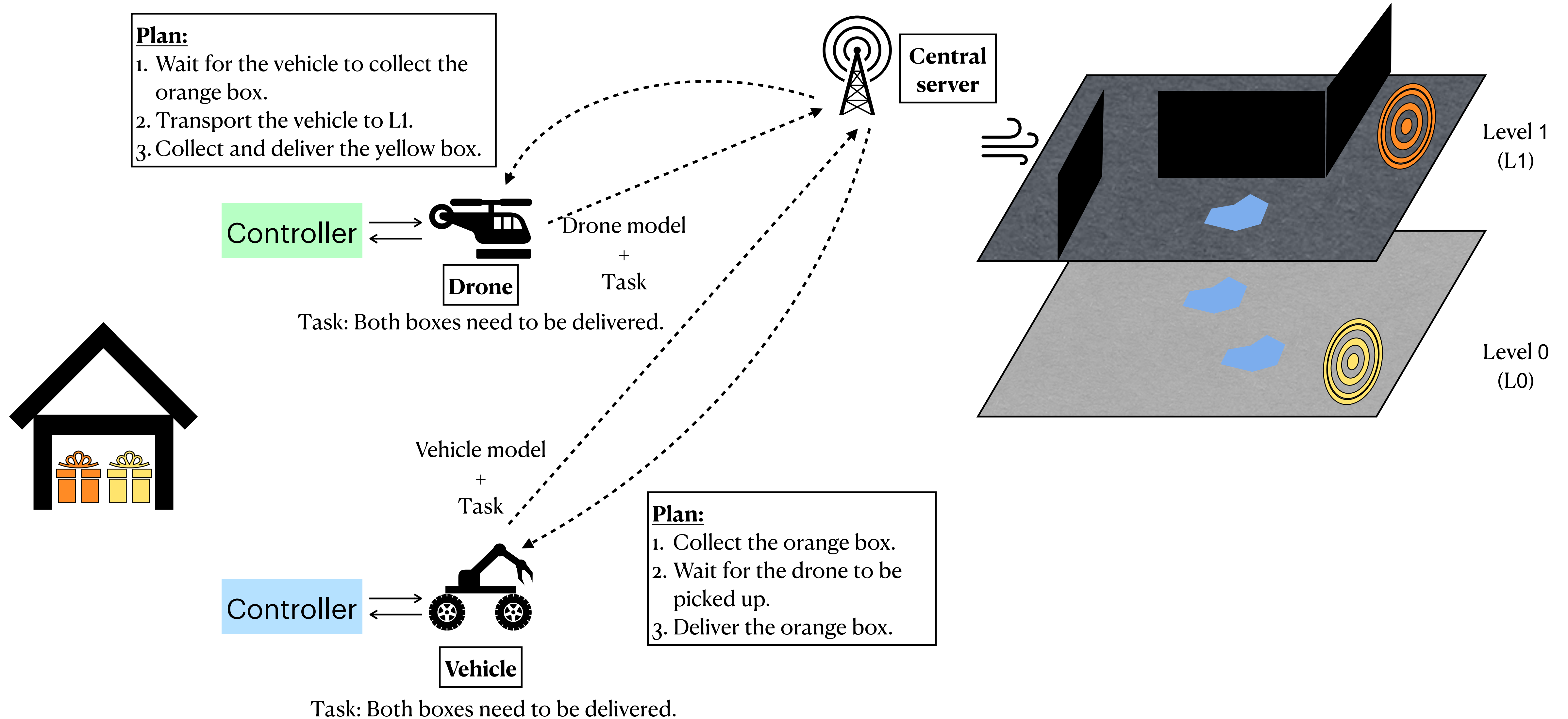
Task: Both boxes need to be delivered.



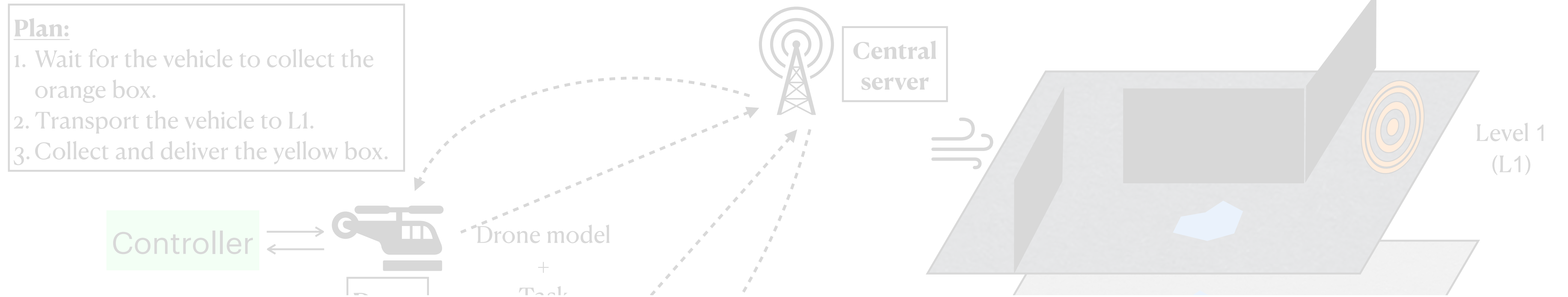
A Distributed Synthesis Problem for Delivery Robots



A Distributed Synthesis Problem for Delivery Robots

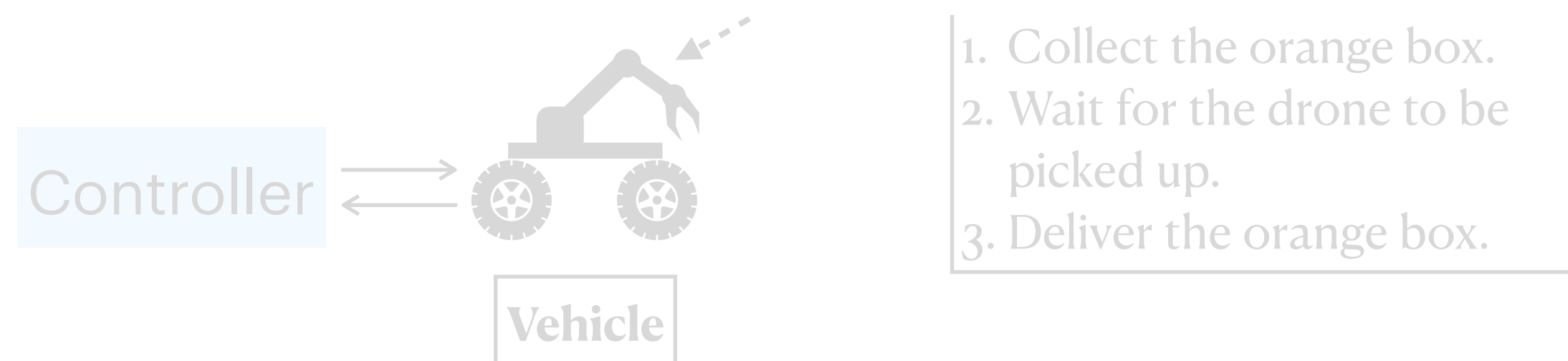


A Distributed Synthesis Problem for Delivery Robots



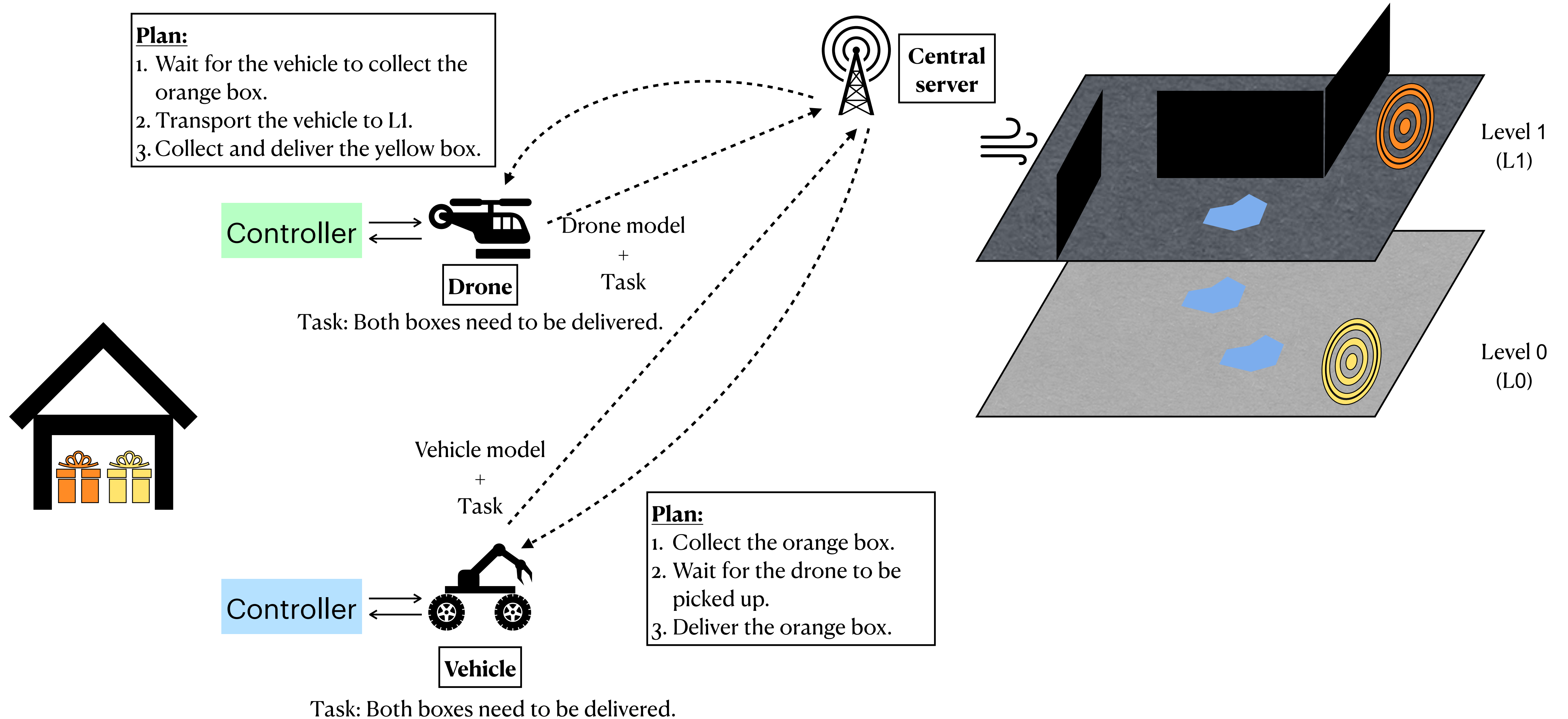
Distributed synthesis methods which use semi-centralized approach:

- Sven Schewe and Bernd Finkbeiner. "Bounded synthesis." ATVA 2007
- Rajeev Alur, Salar Moarref, and Ufuk Topcu. "Compositional and symbolic synthesis of reactive controllers for multi-agent systems." Information and Computation 2018.

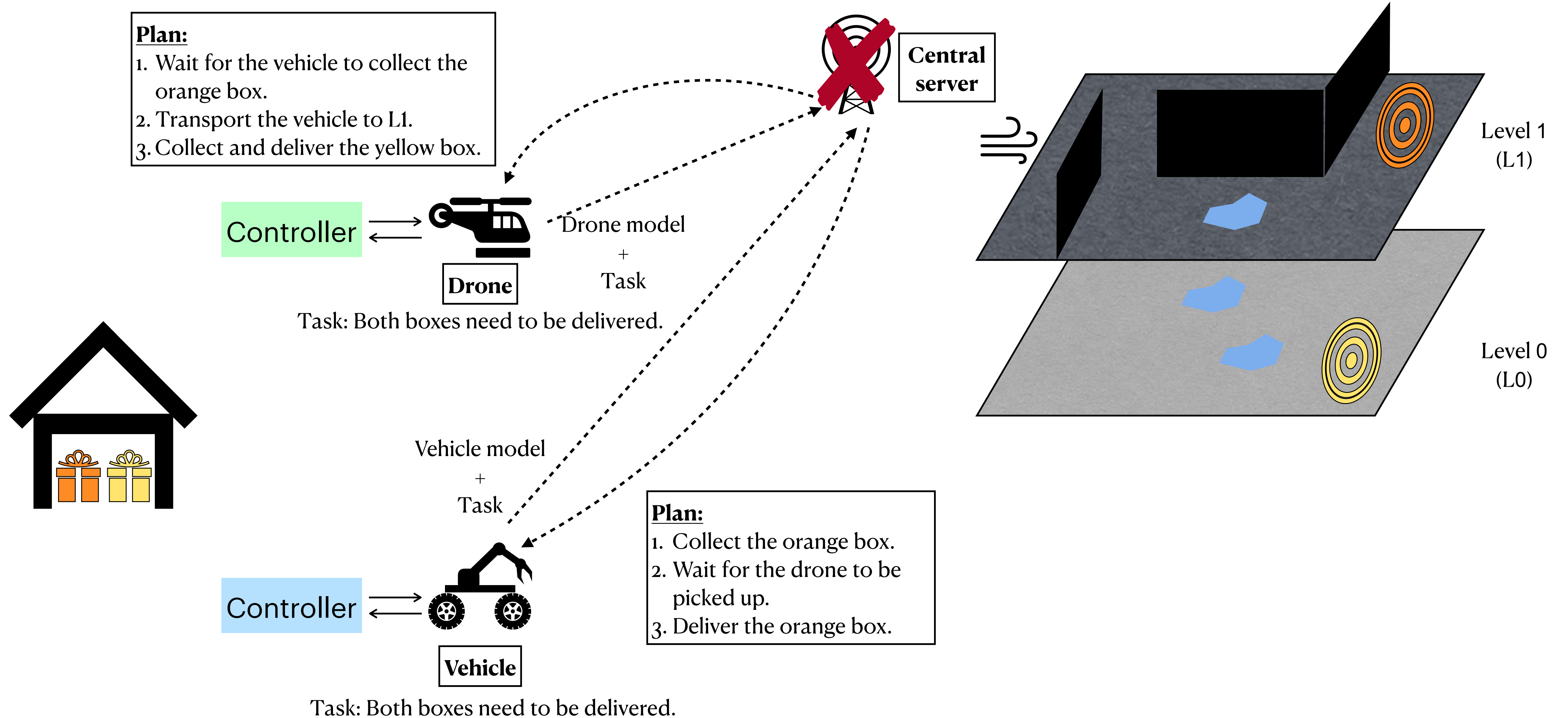


Task: Both boxes need to be delivered.

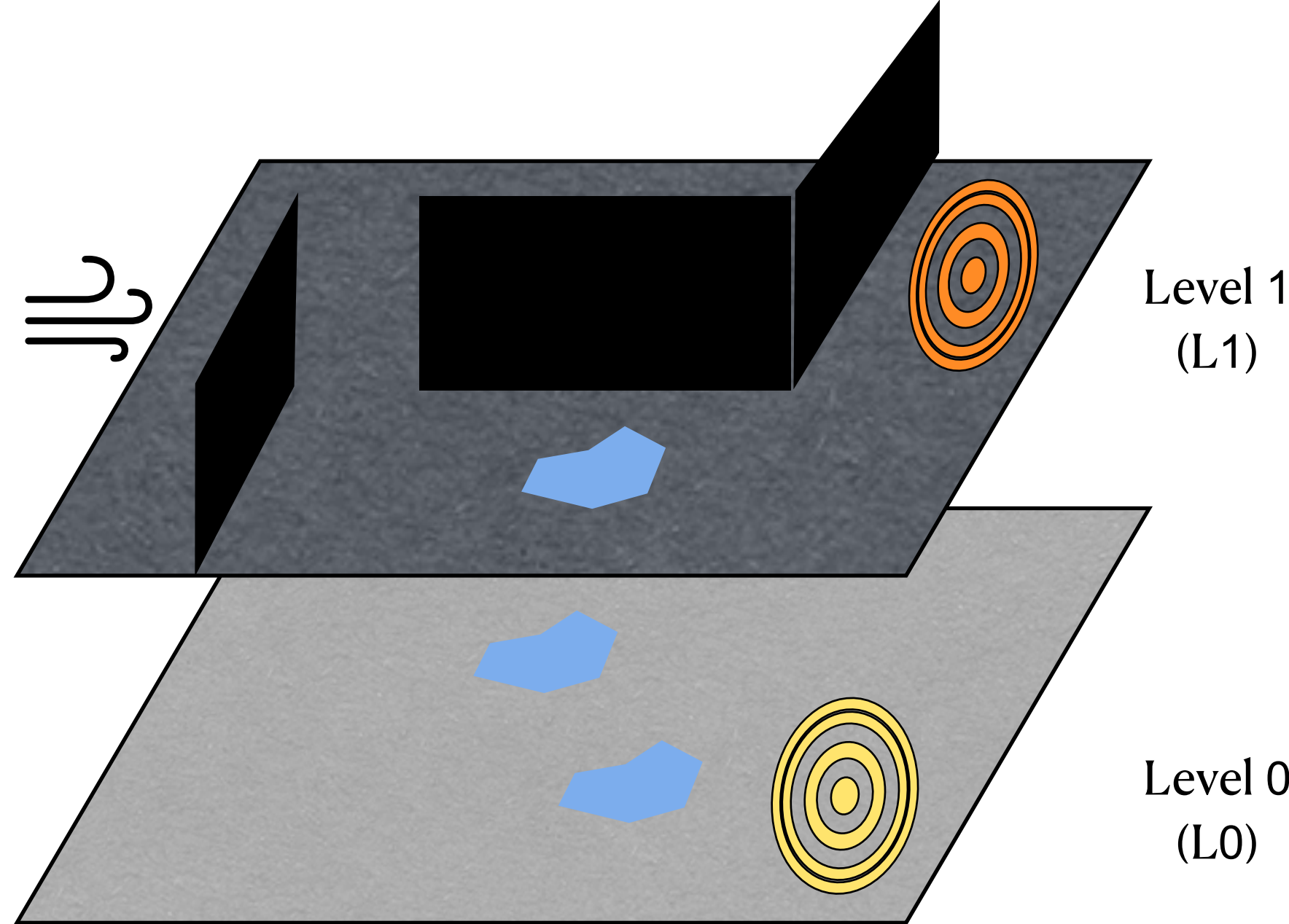
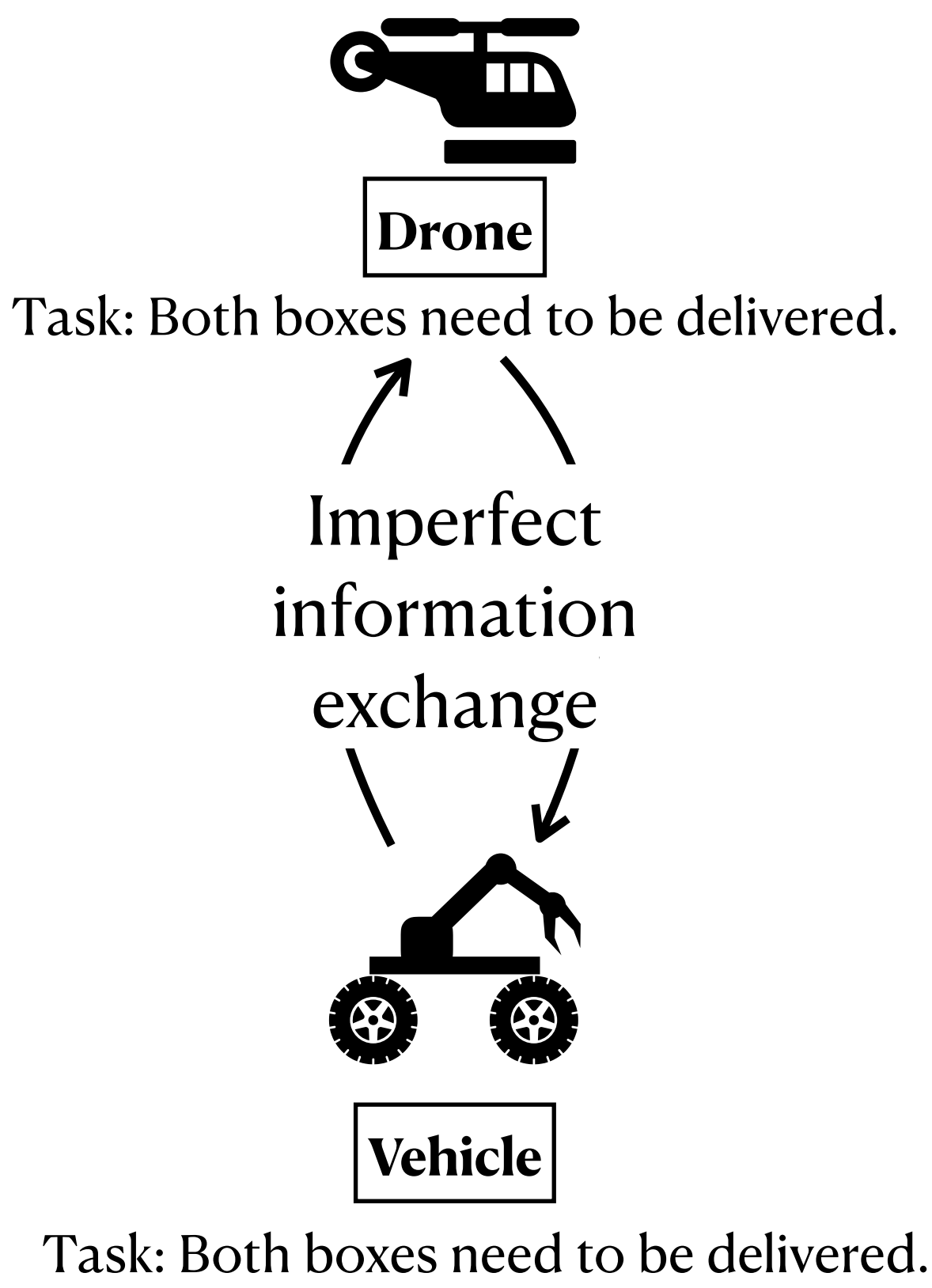
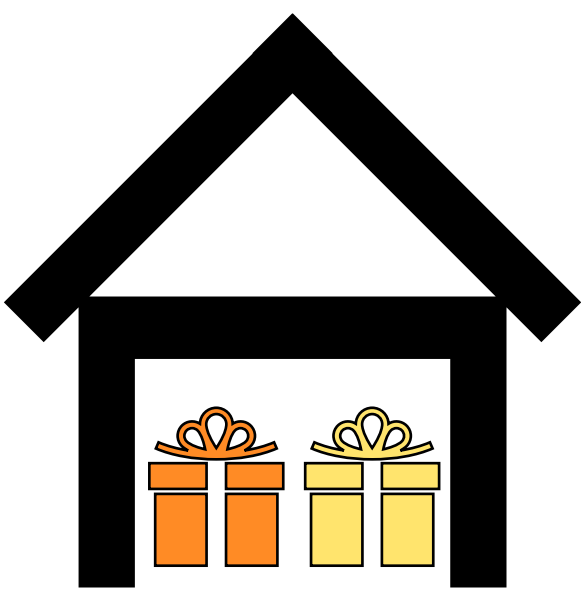
A Distributed Synthesis Problem for Delivery Robots



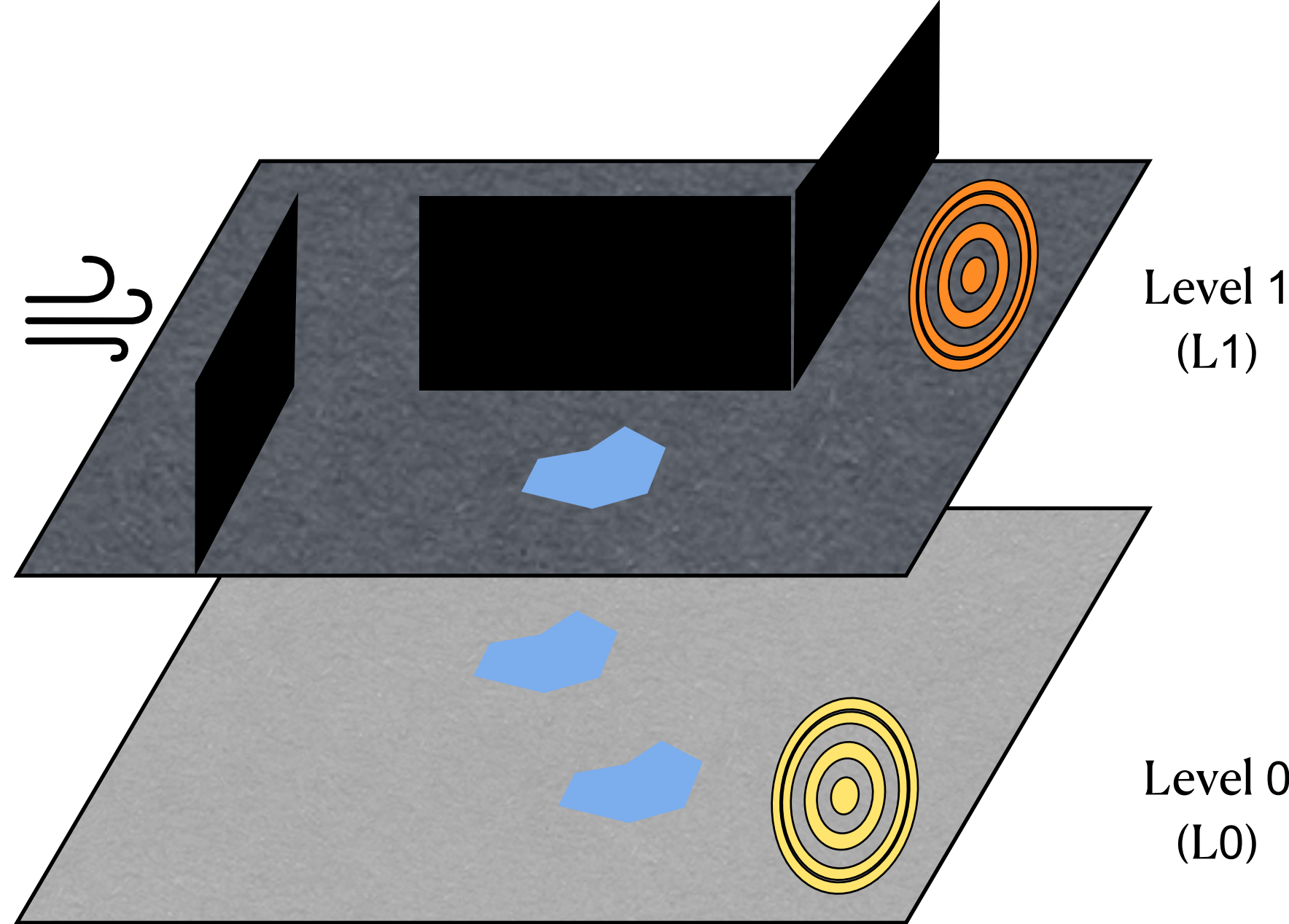
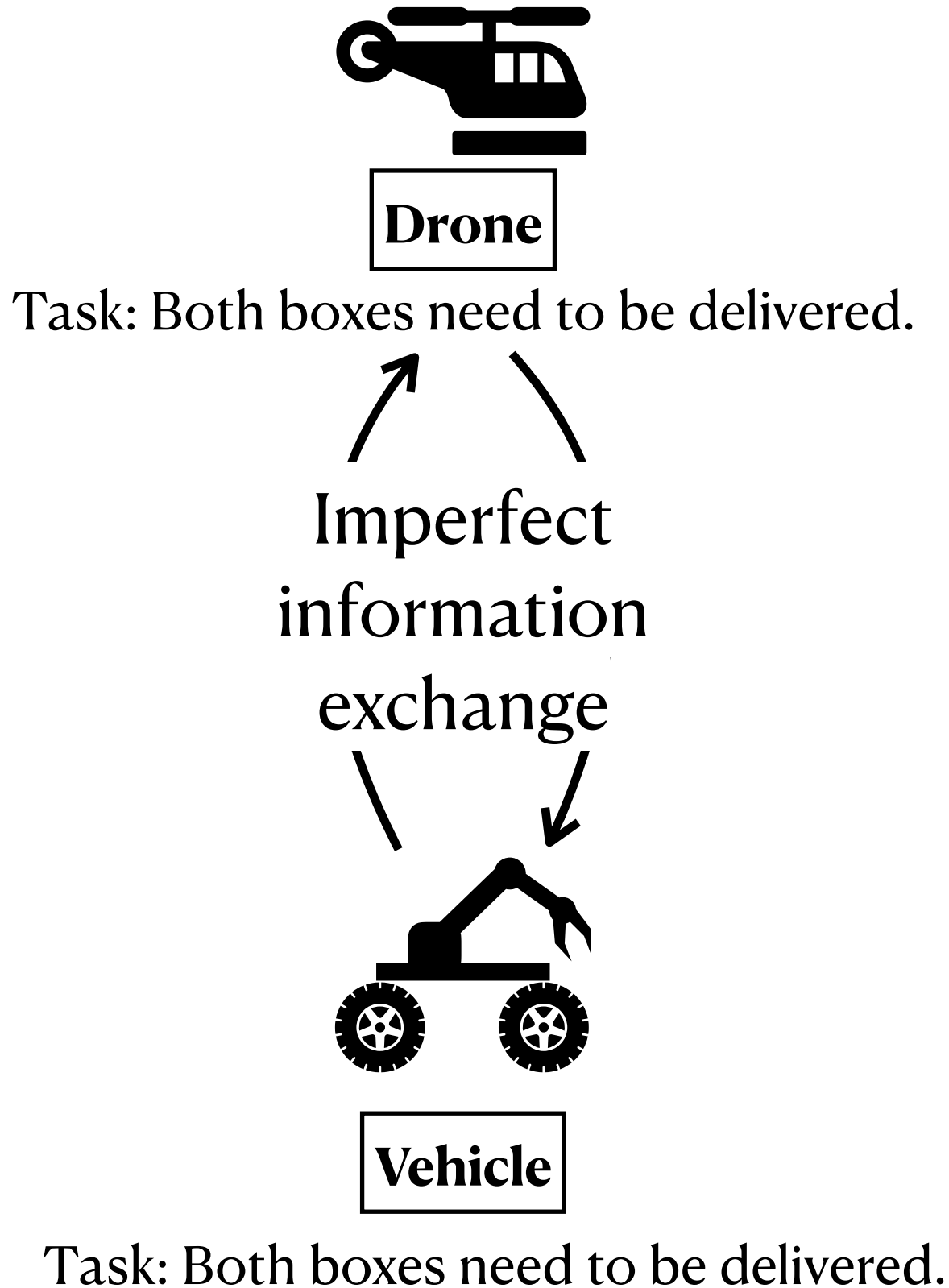
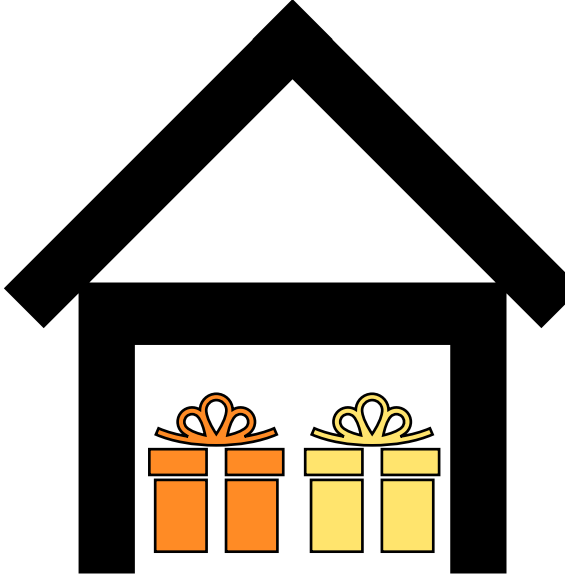
A Distributed Synthesis Problem for Delivery Robots



A Distributed Synthesis Problem for Delivery Robots

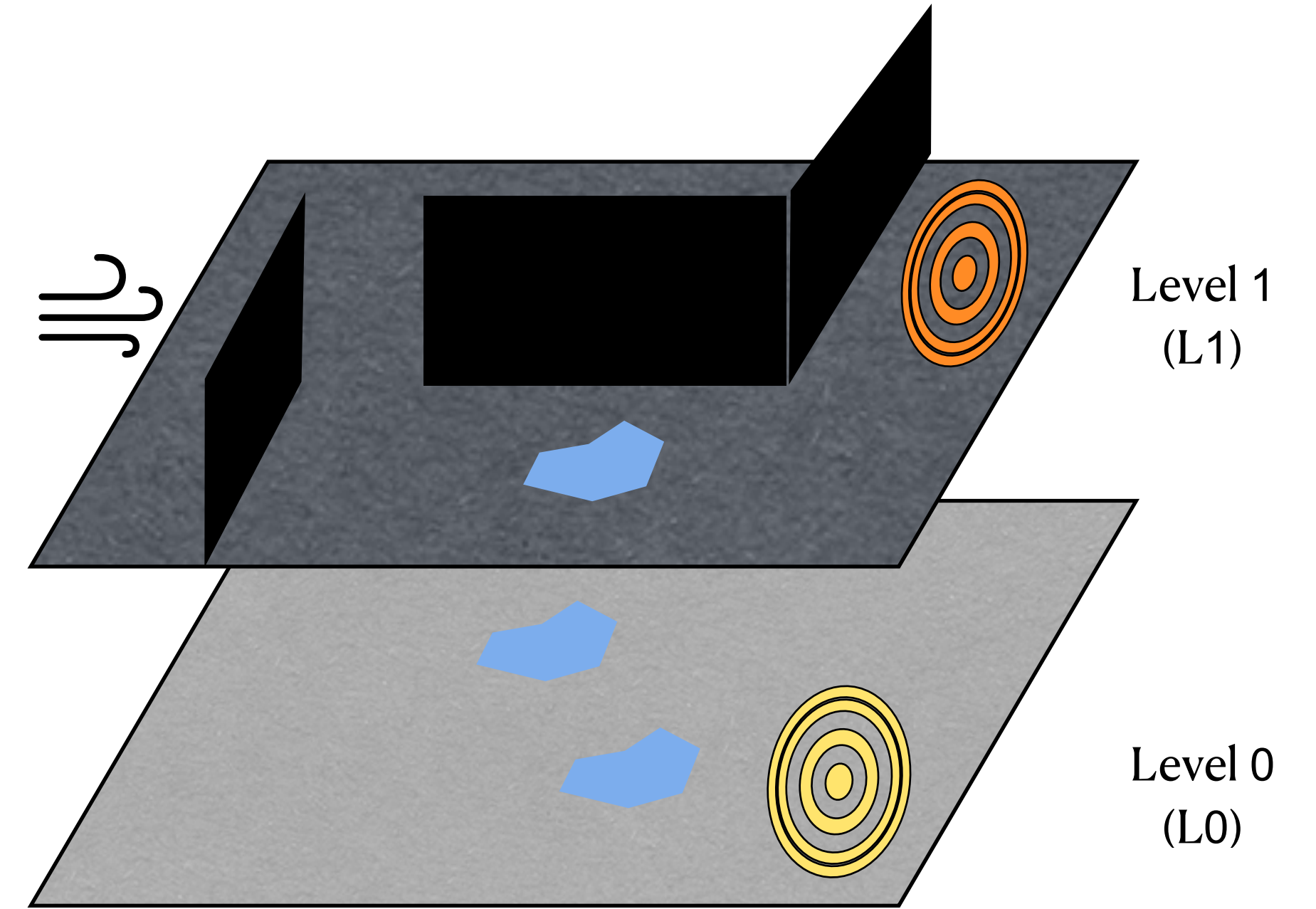
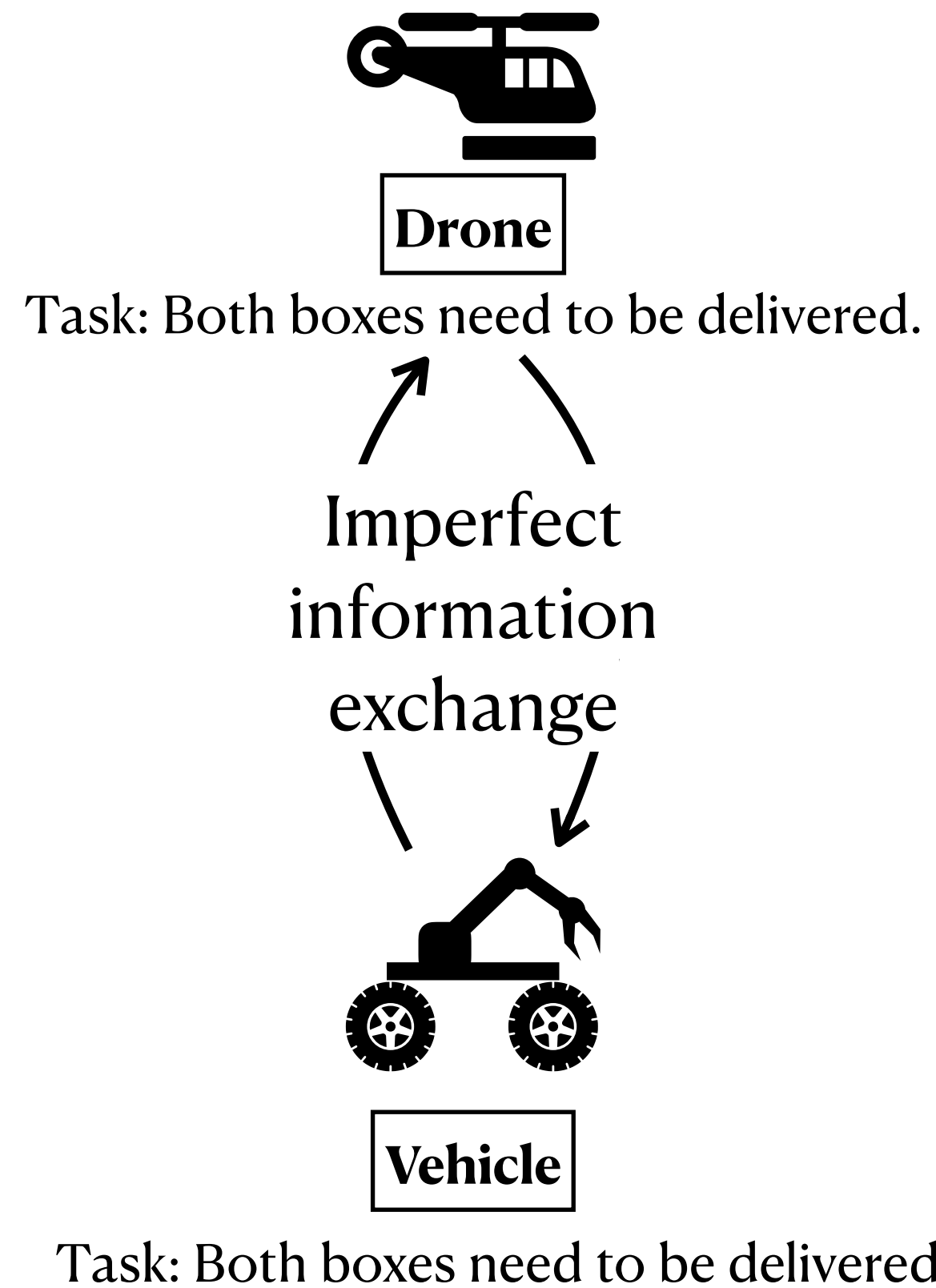


A Distributed Synthesis Problem for Delivery Robots



Distributed reactive synthesis is undecidable
[Pnueli and Rosner, 1990].

Negotiation: A Modular Approach

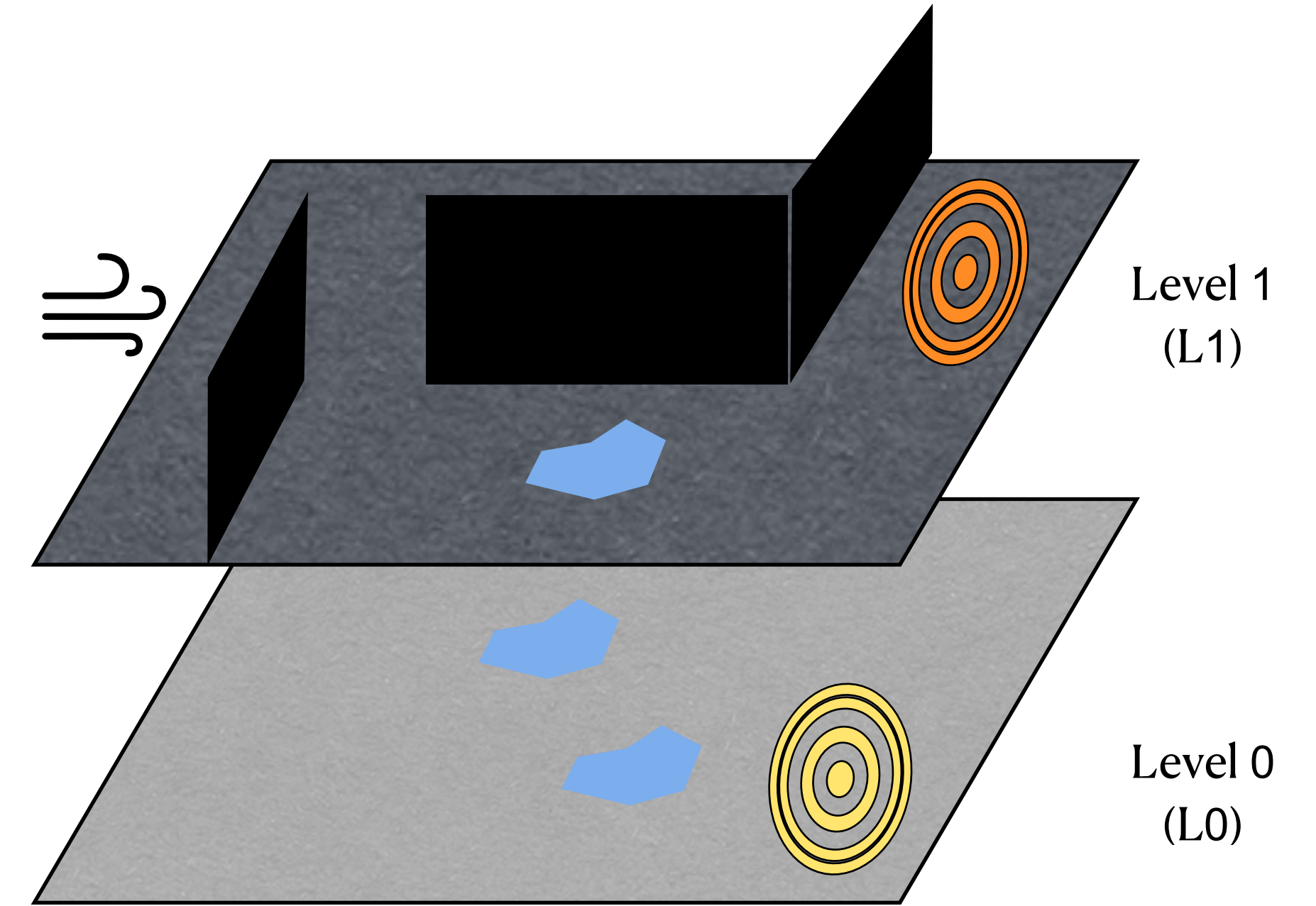
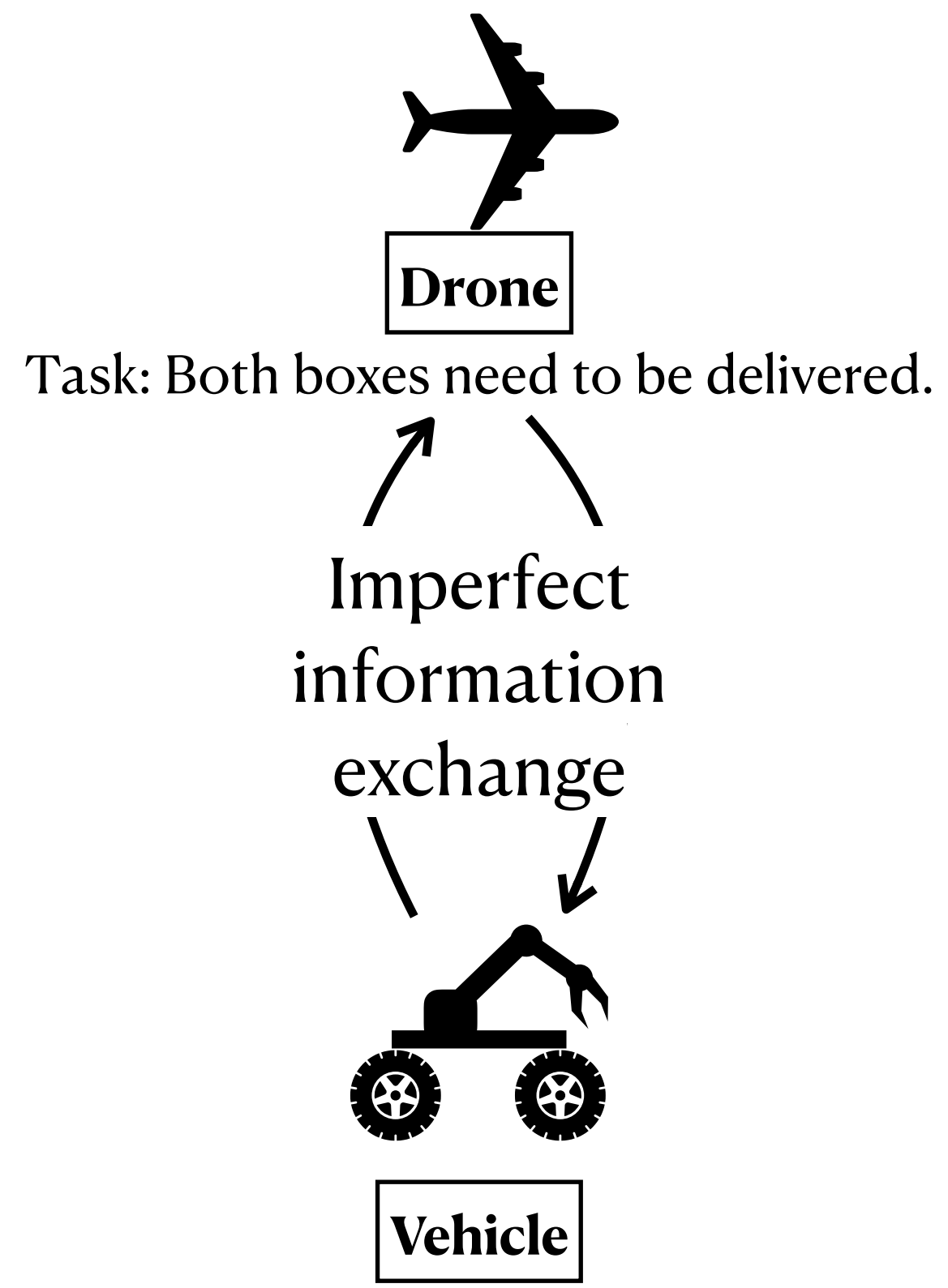


Distributed reactive synthesis is undecidable

[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

Negotiation: A Modular Approach

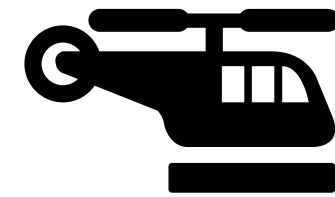


Distributed reactive synthesis is undecidable

[Pnueli and Rosner, 1990].

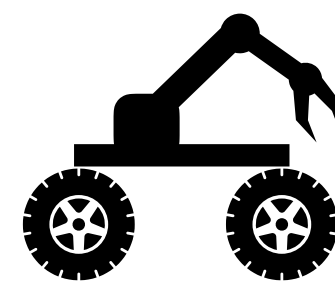
**Negotiation:
a sound, incomplete, and modular approach.**

Negotiation: A Modular Approach



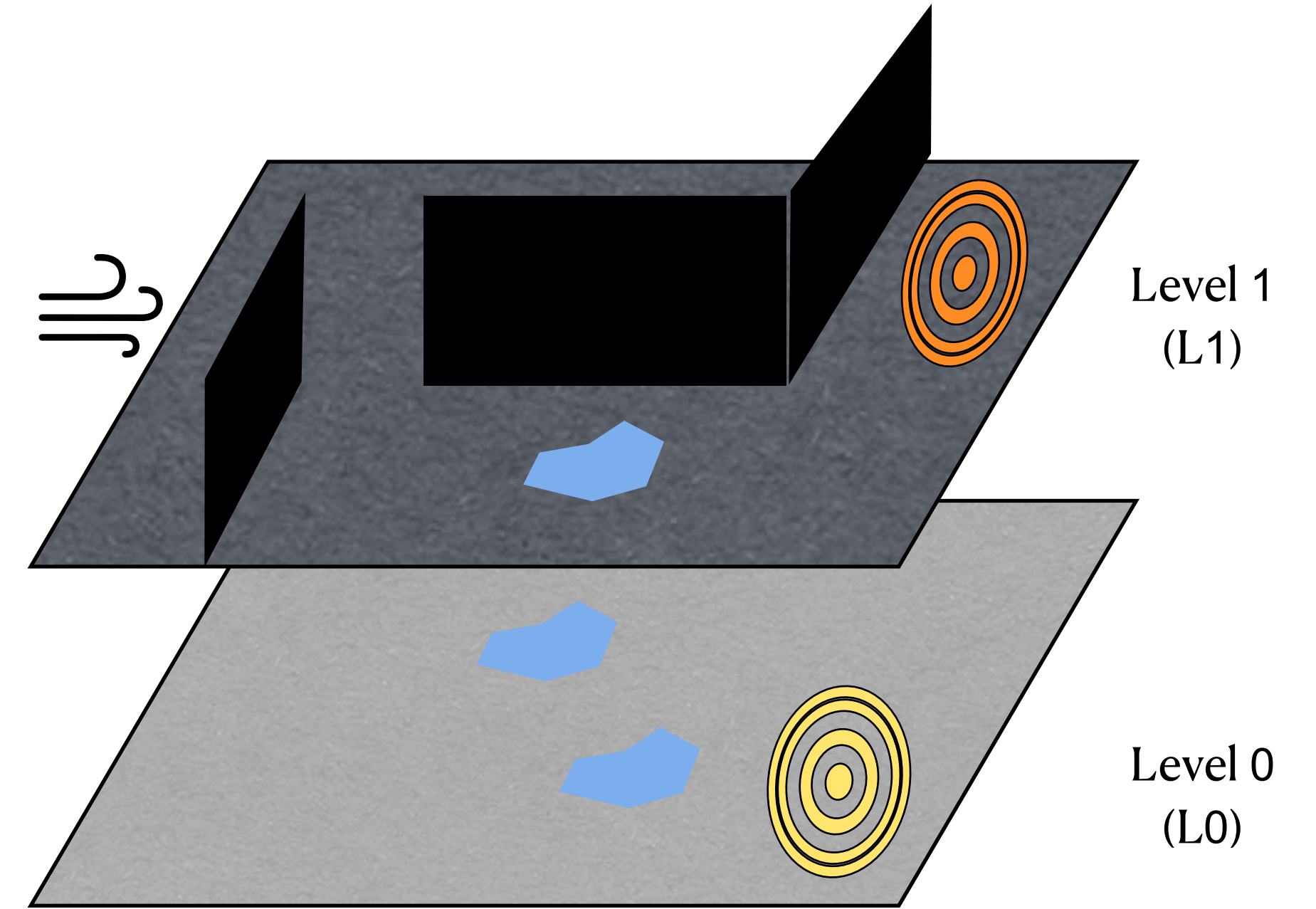
Drone

Task: Both boxes need to be delivered.



Vehicle

Task: Both boxes need to be delivered.

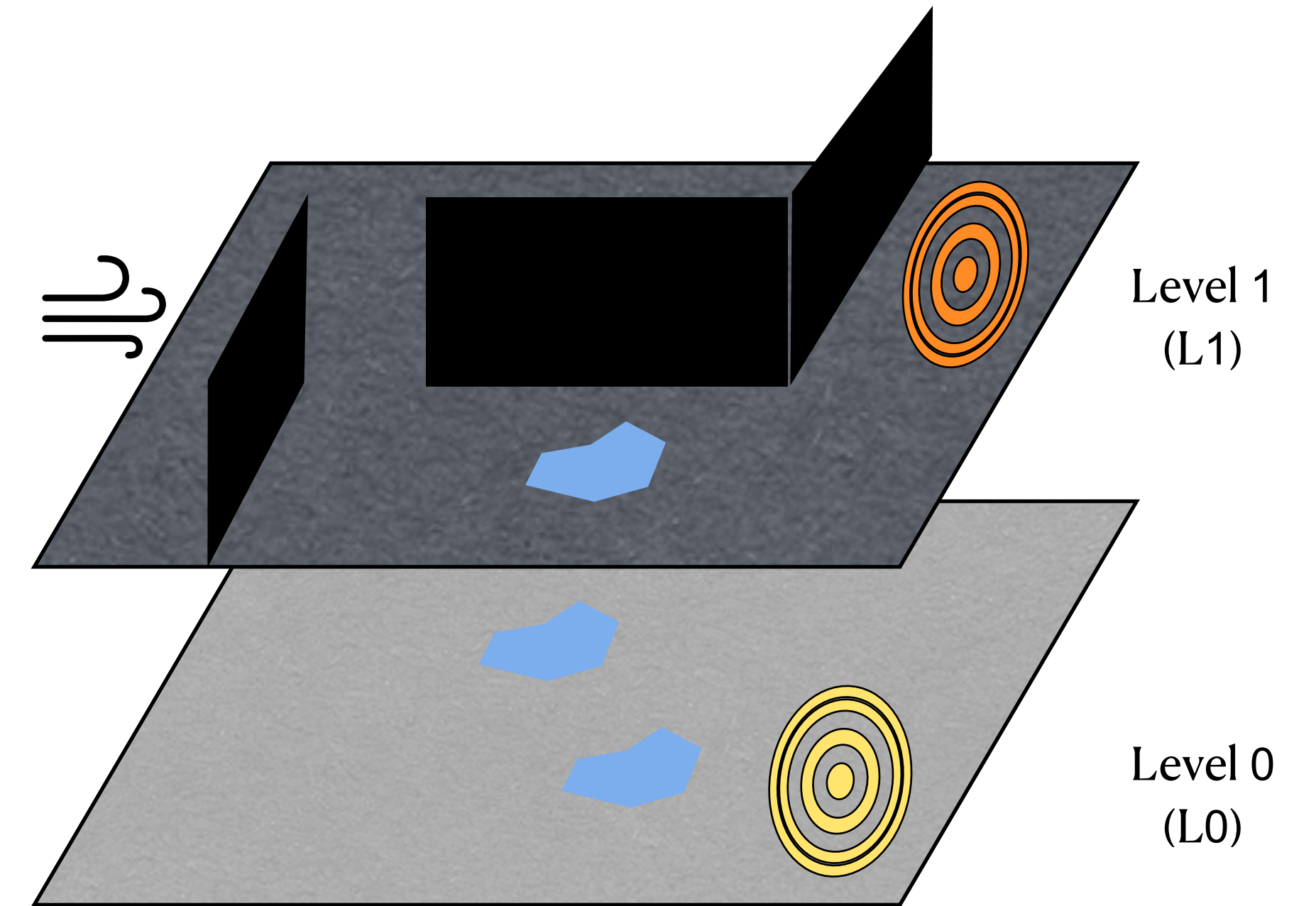
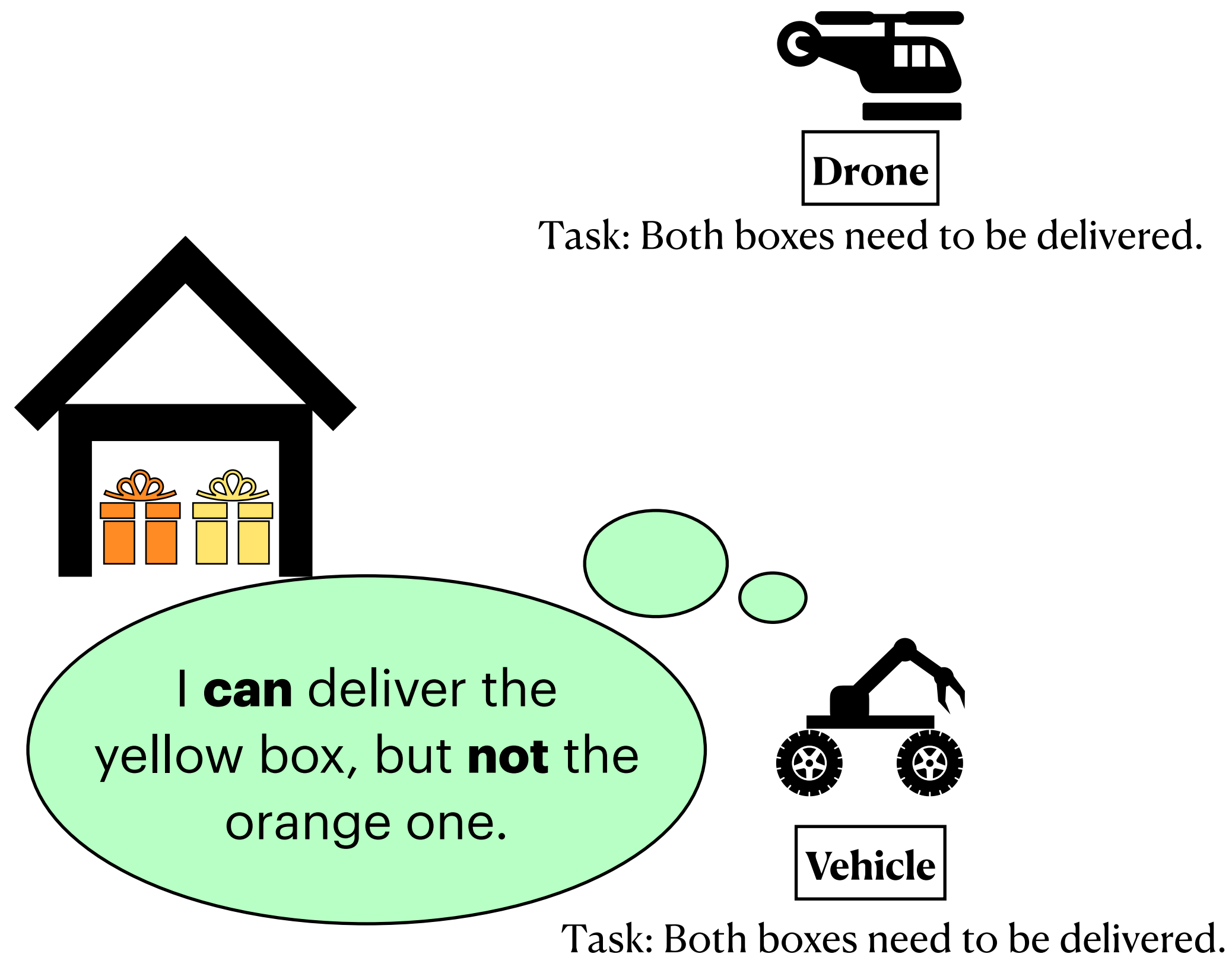


Distributed reactive synthesis is undecidable

[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

Negotiation: A Modular Approach

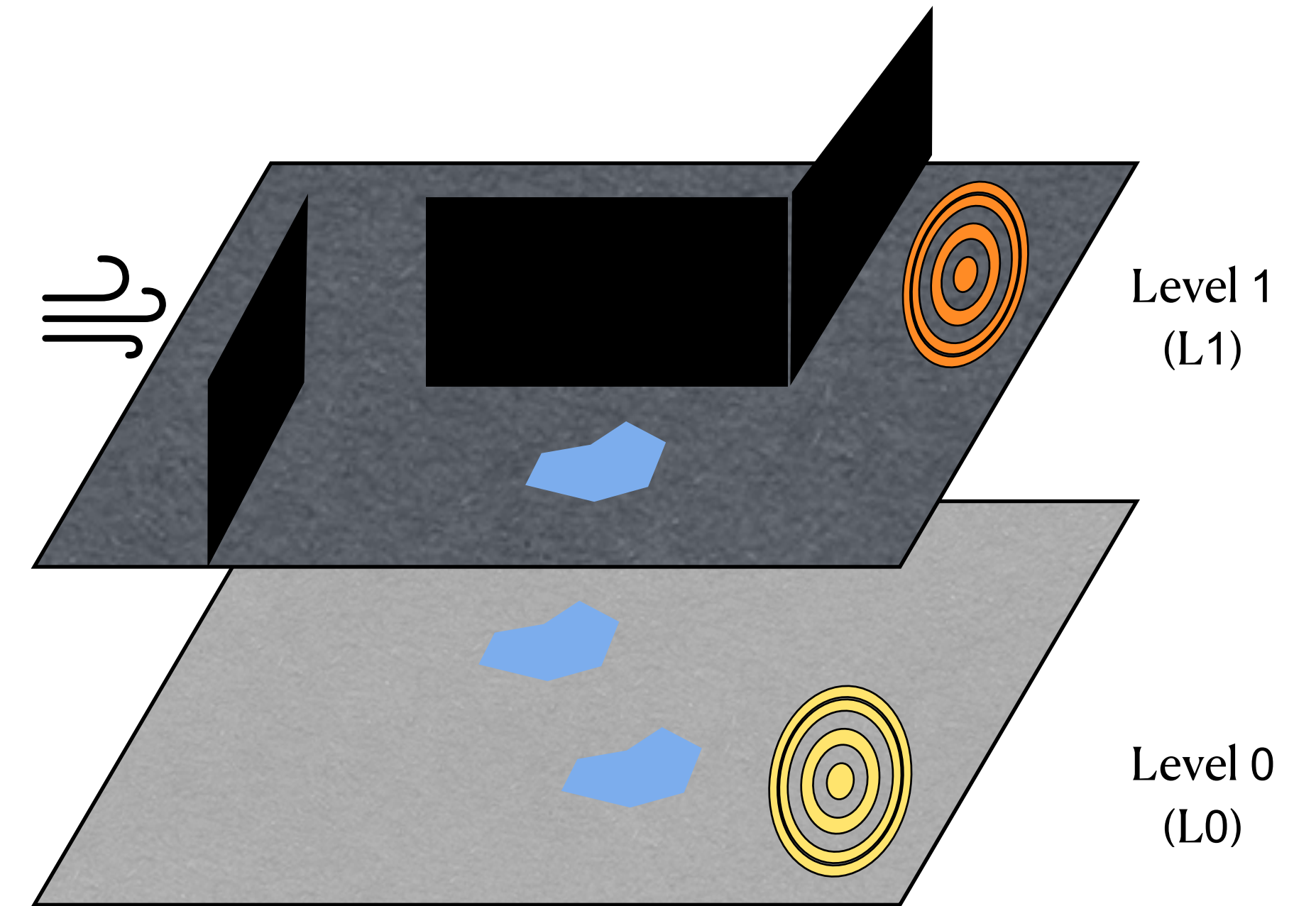
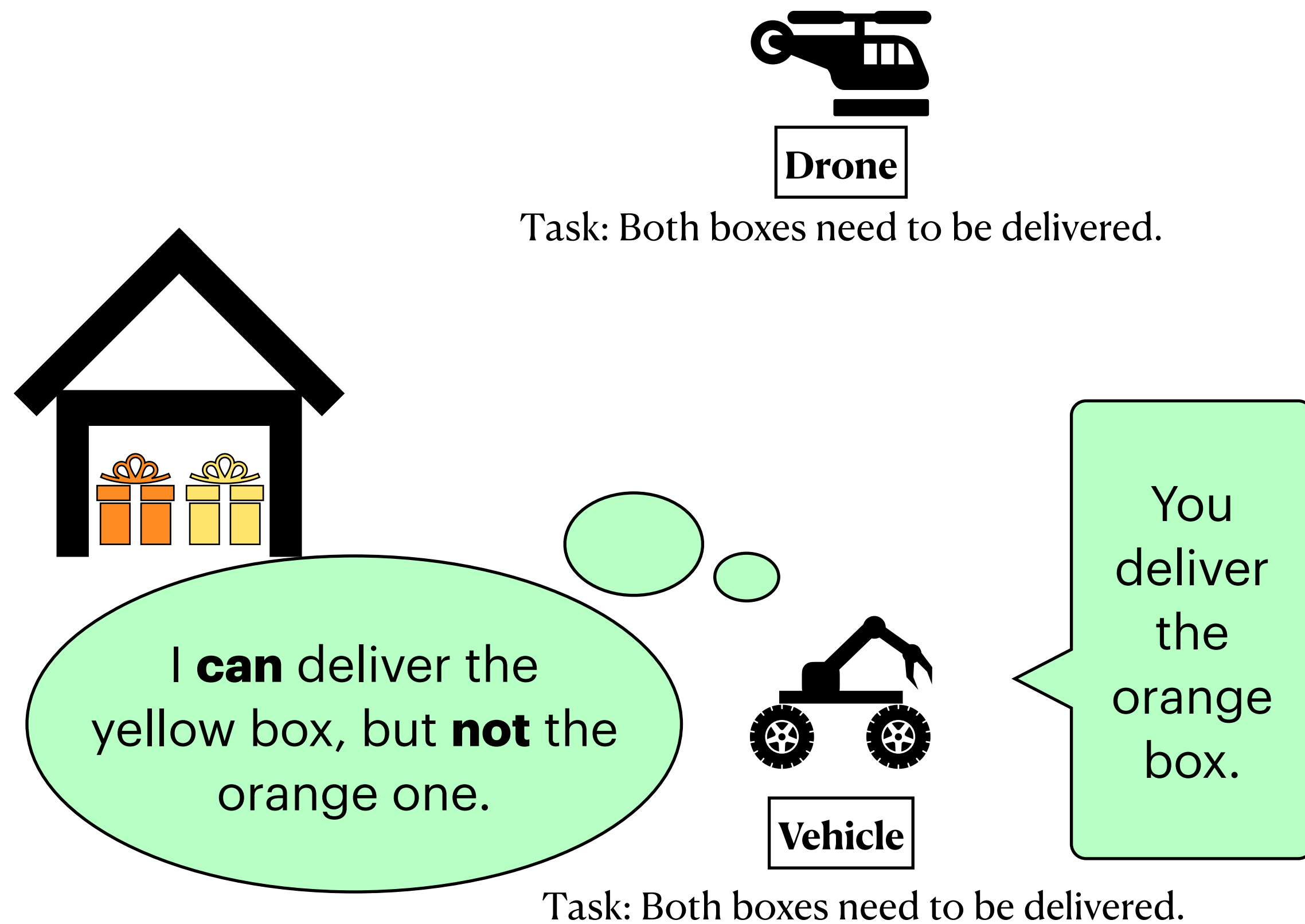


Distributed reactive synthesis is undecidable

[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

Negotiation: A Modular Approach

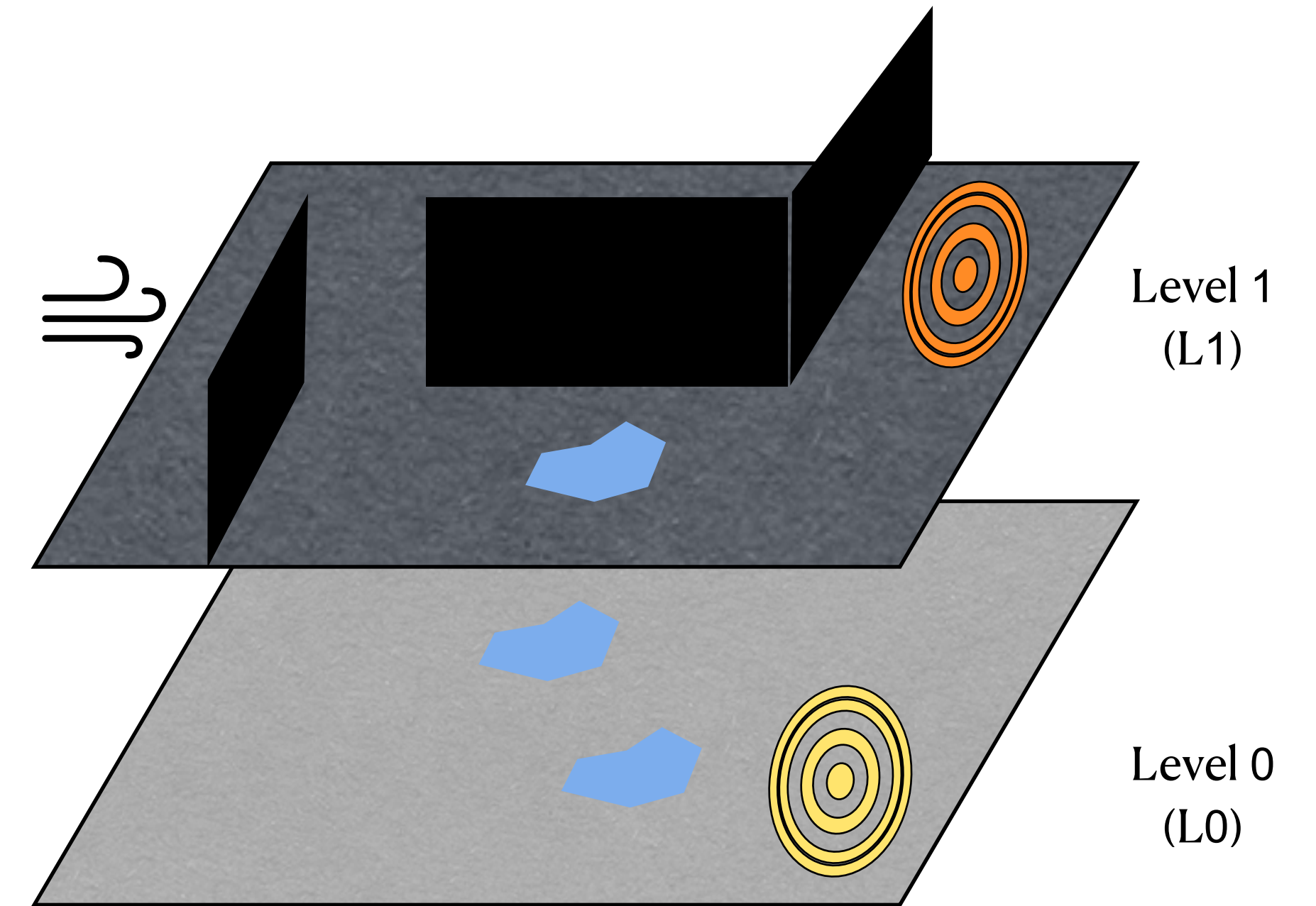
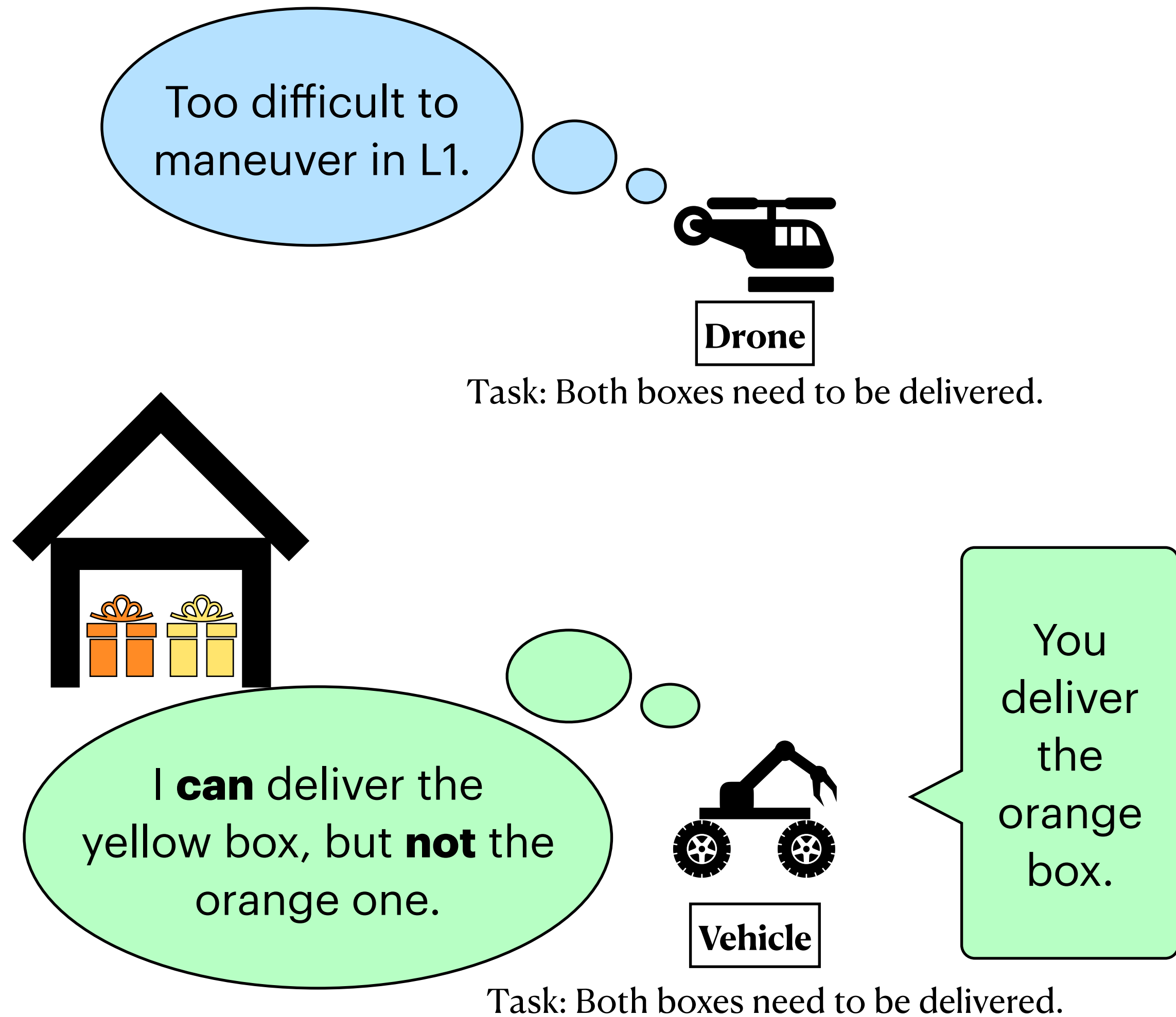


Distributed reactive synthesis is undecidable

[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

Negotiation: A Modular Approach

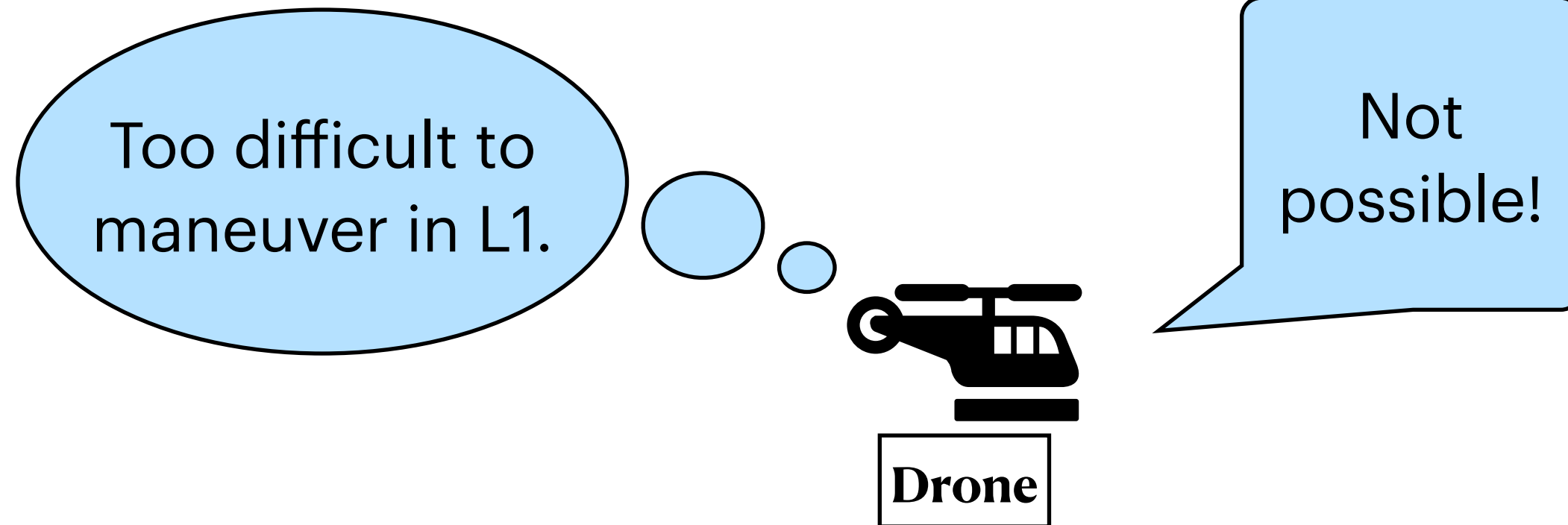


Distributed reactive synthesis is undecidable

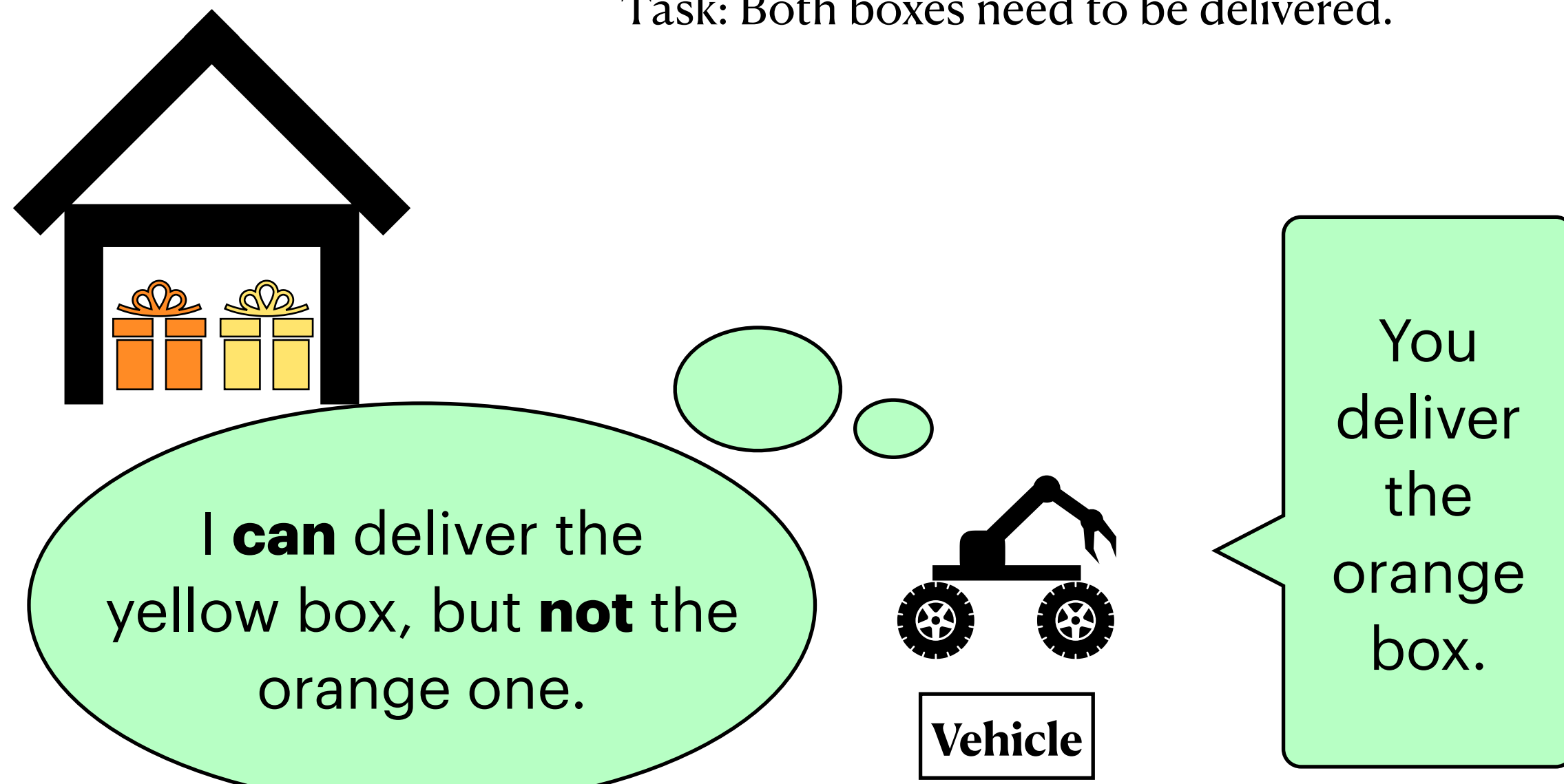
[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

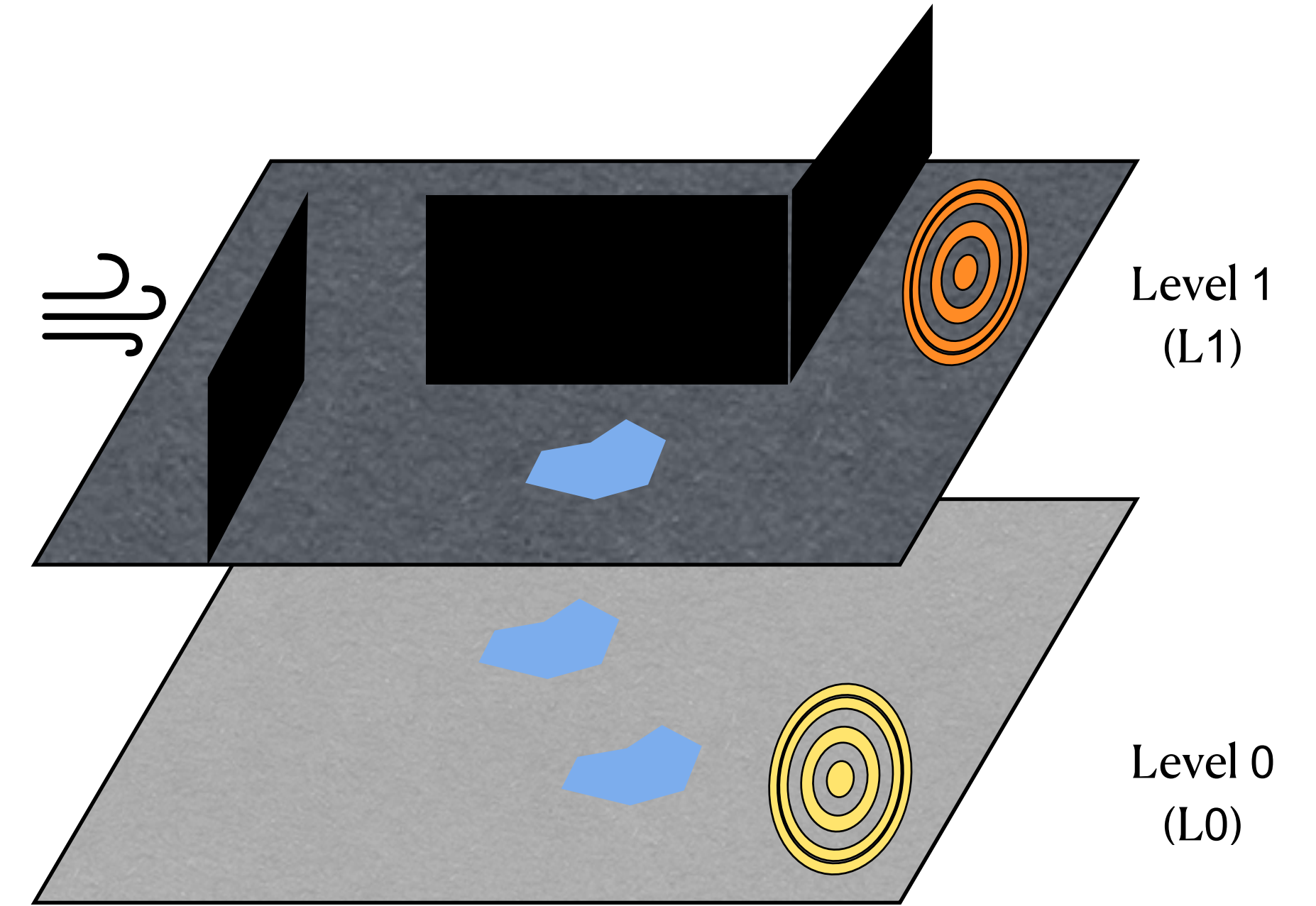
Negotiation: A Modular Approach



Task: Both boxes need to be delivered.



Task: Both boxes need to be delivered.

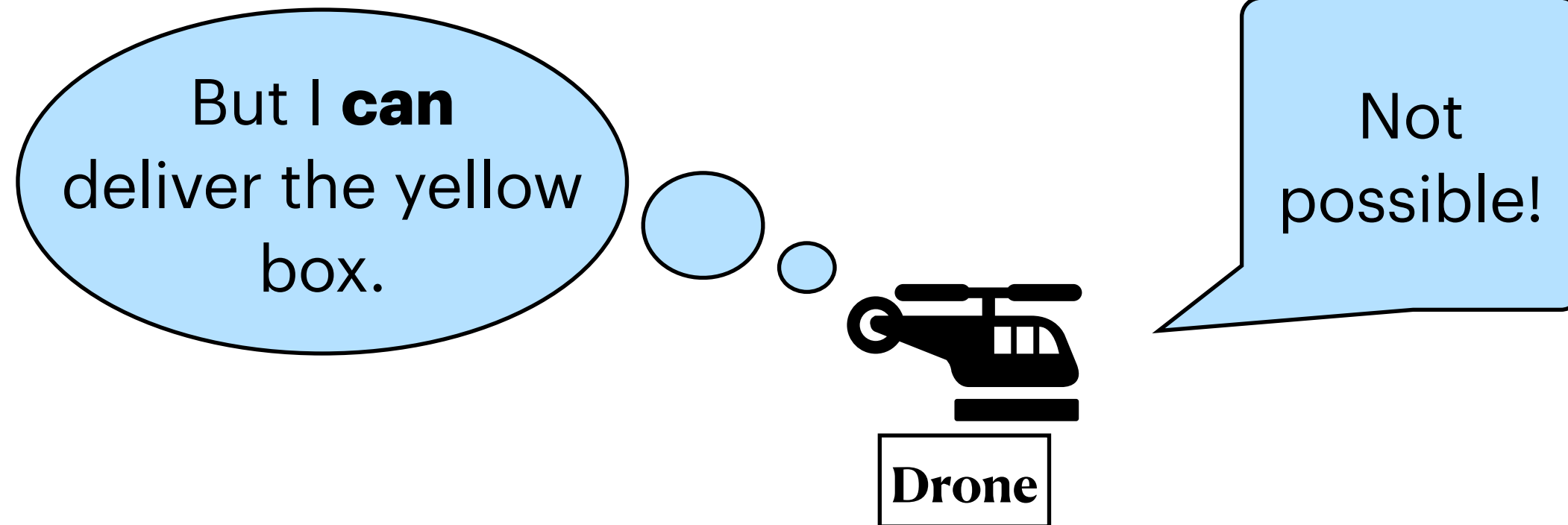


Distributed reactive synthesis is undecidable

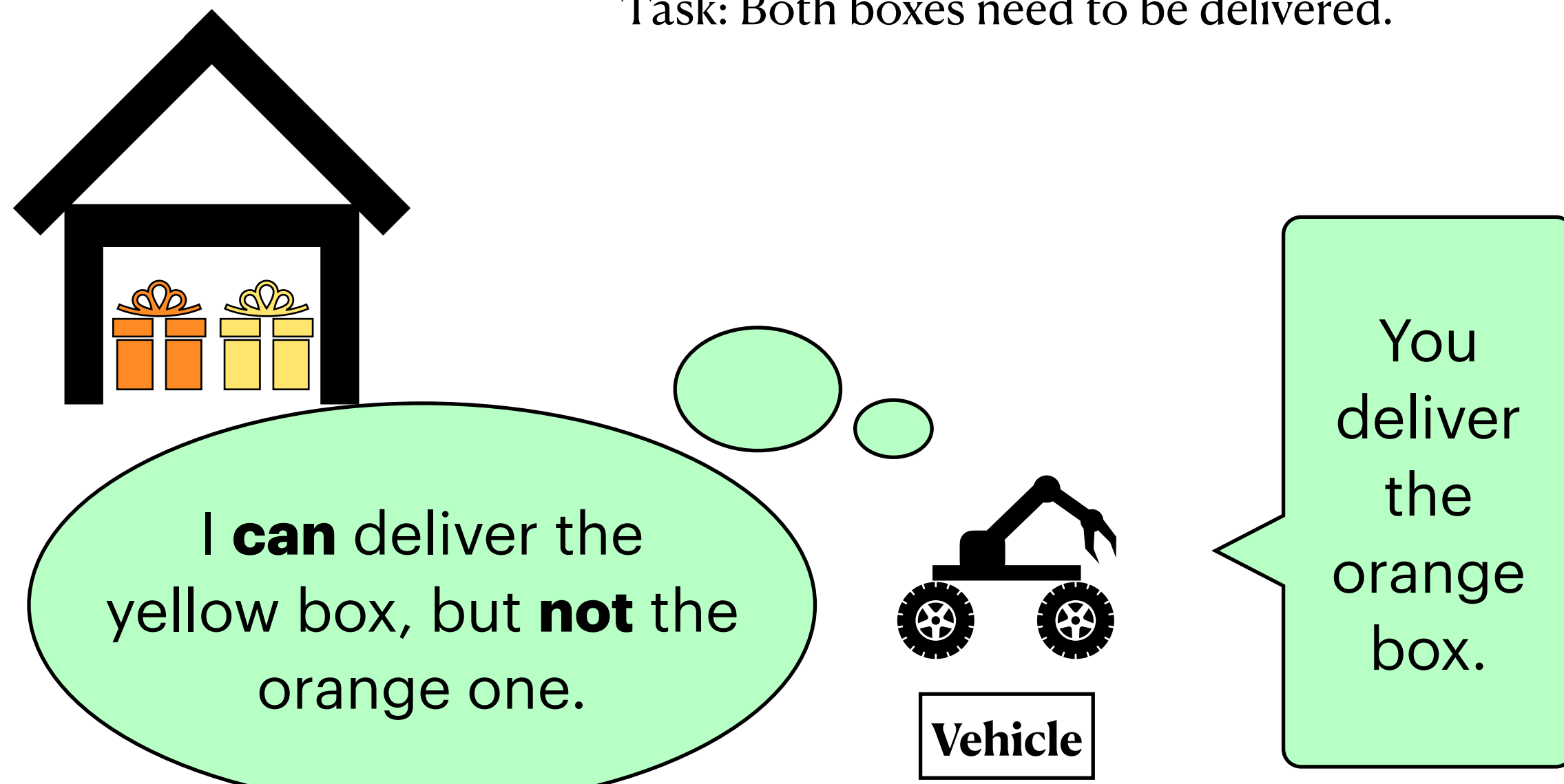
[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

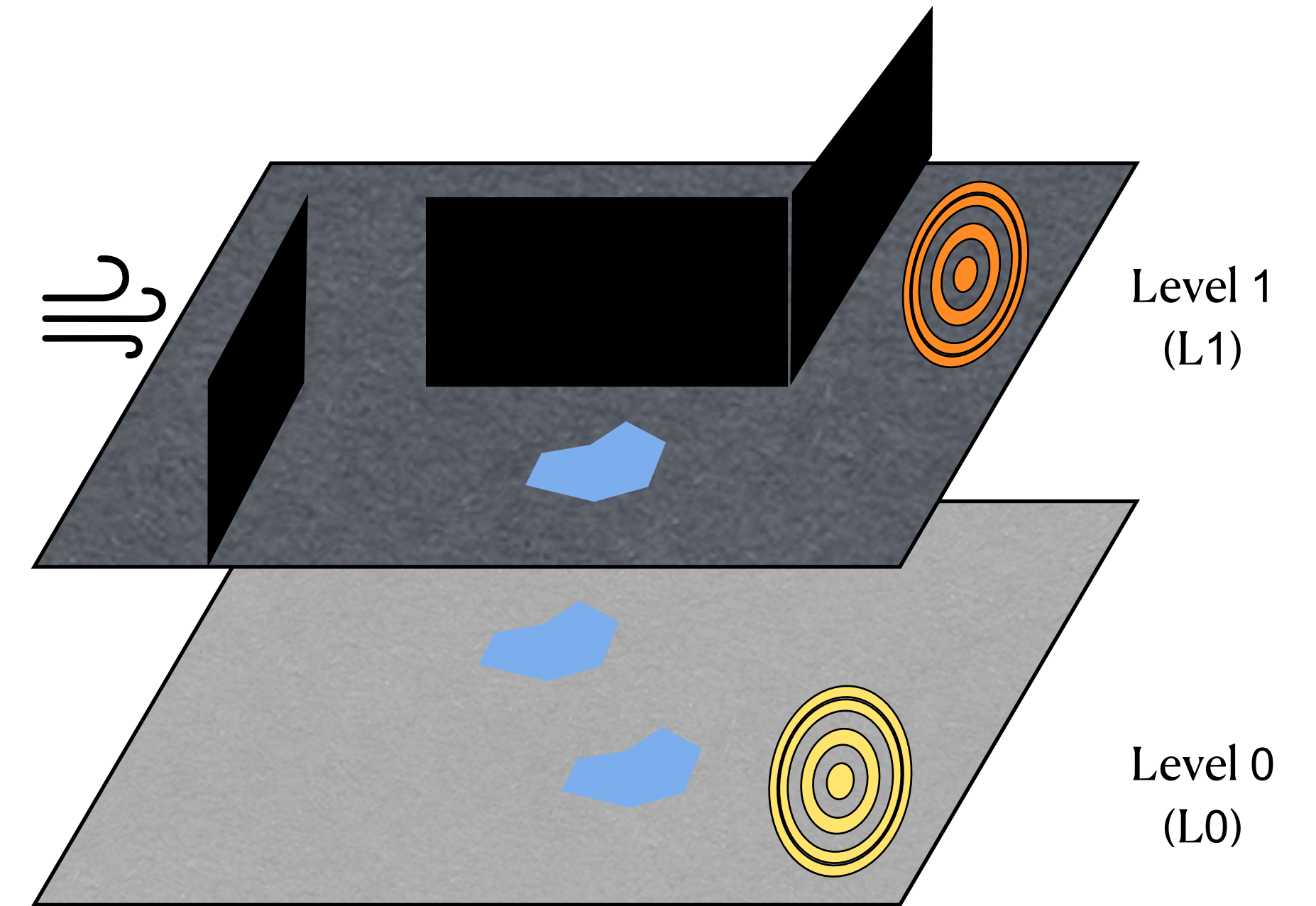
Negotiation: A Modular Approach



Task: Both boxes need to be delivered.



Task: Both boxes need to be delivered.



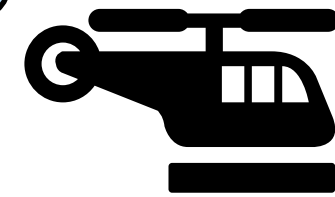
Distributed reactive synthesis is undecidable

[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

Negotiation: A Modular Approach

But I **can** deliver the yellow box.



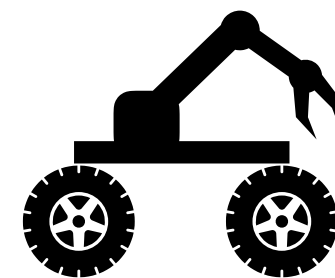
Drone

You deliver the orange box.

Task: Both boxes need to be delivered.



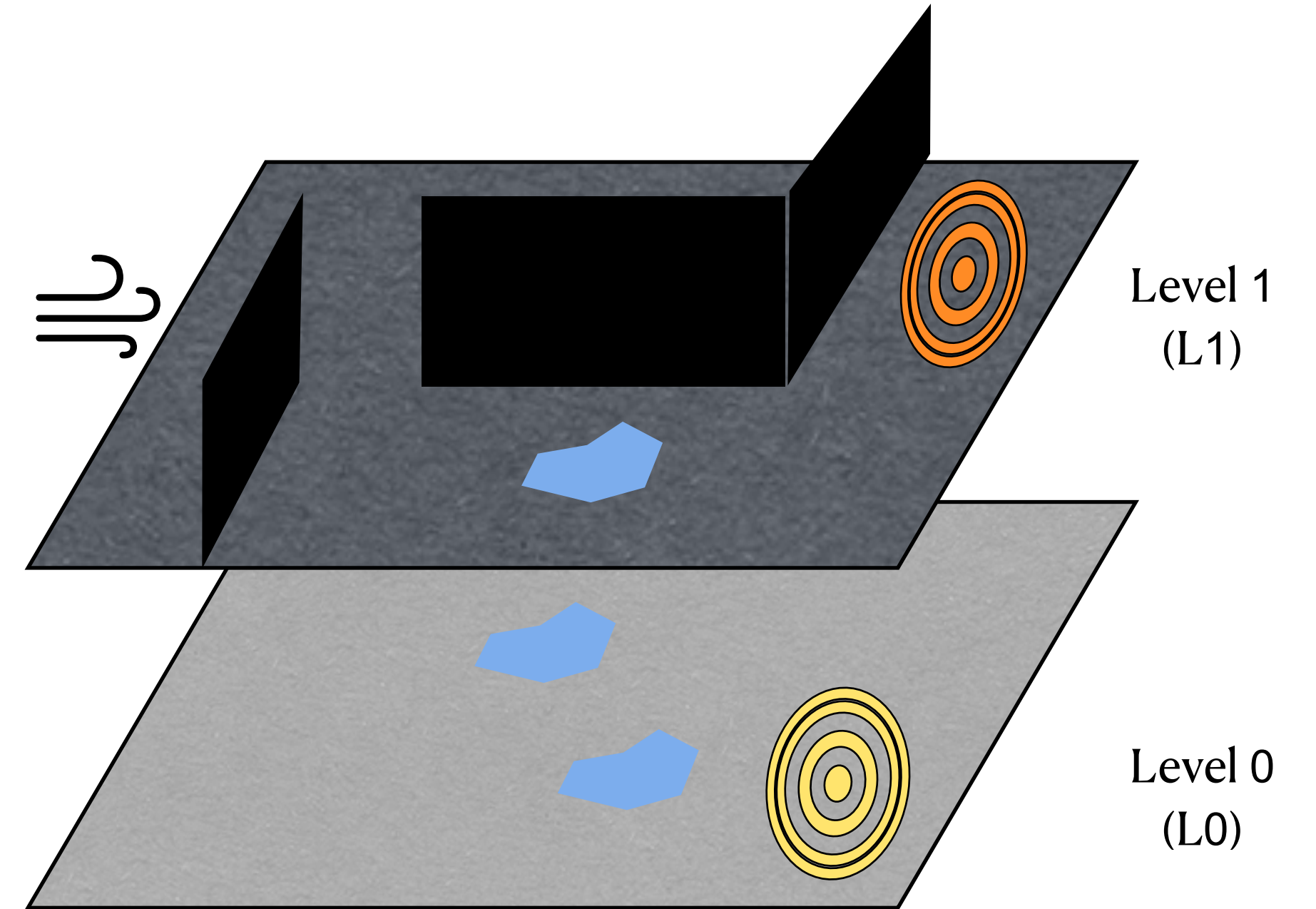
I **can** deliver the yellow box, but **not** the orange one.



Vehicle

You deliver the orange box.

Task: Both boxes need to be delivered.

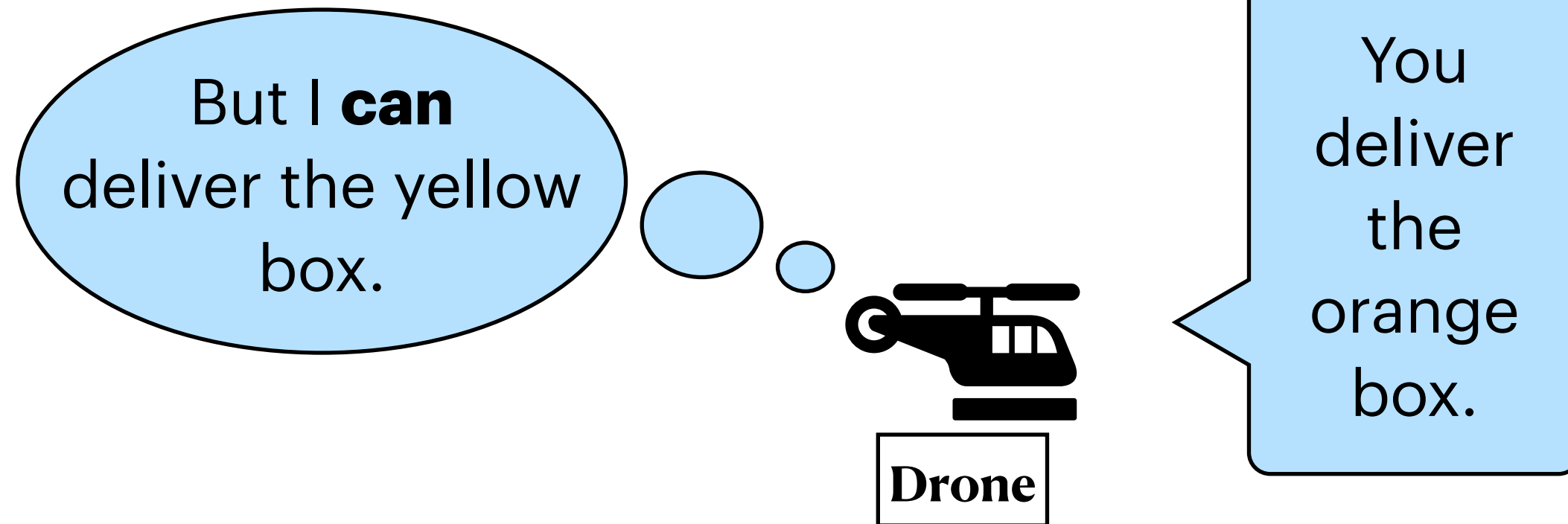


Distributed reactive synthesis is undecidable

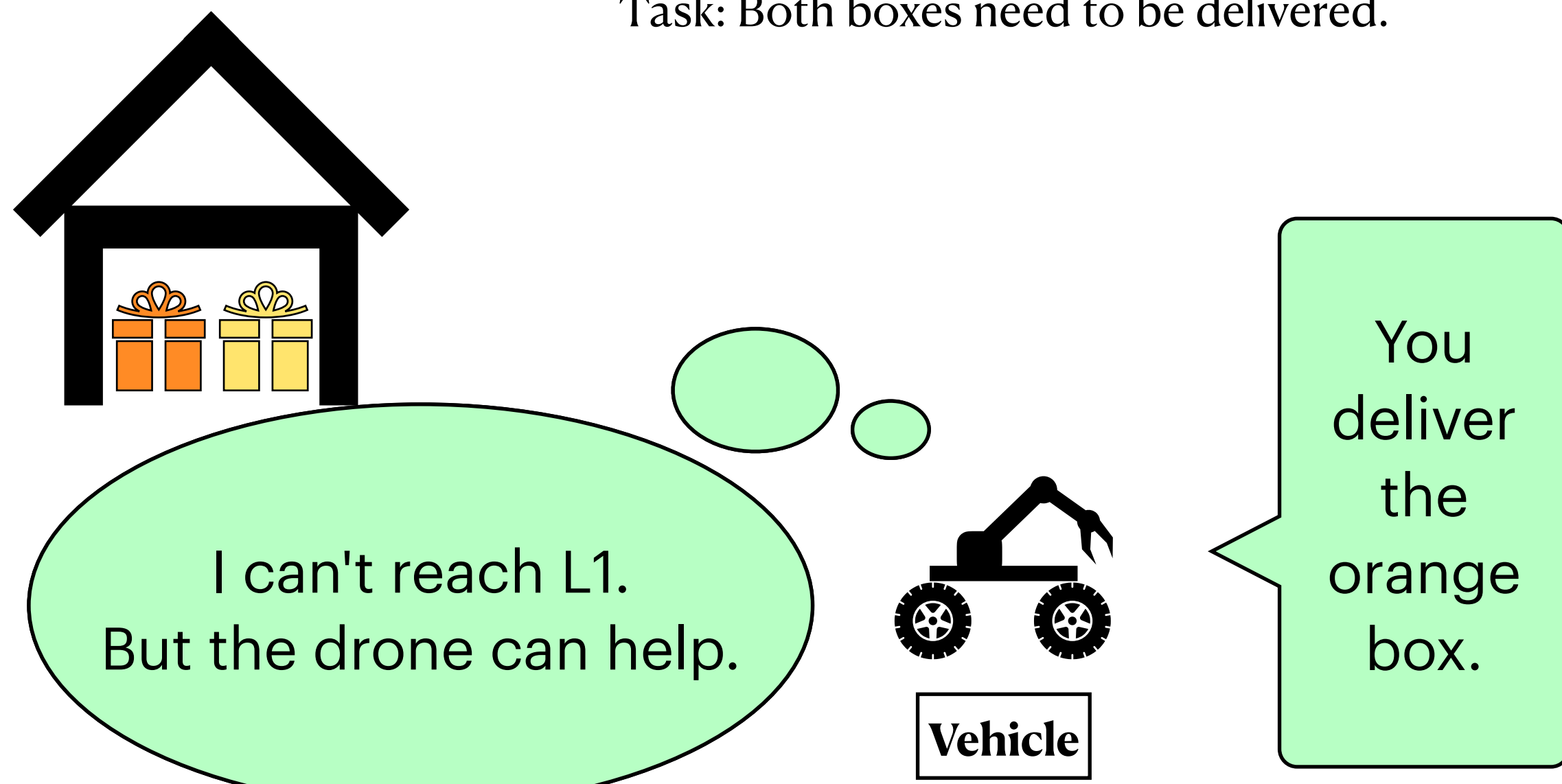
[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

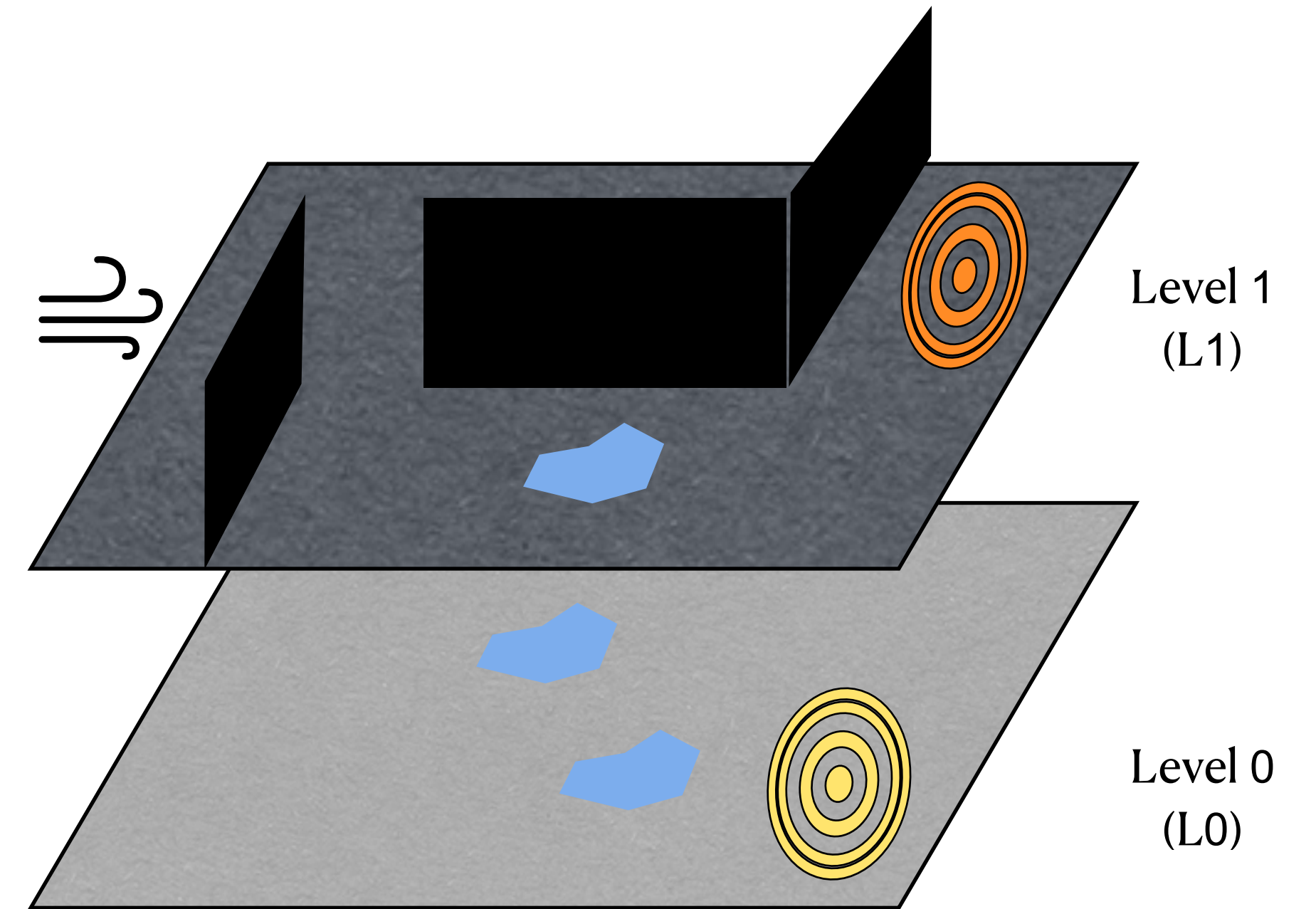
Negotiation: A Modular Approach



Task: Both boxes need to be delivered.



Task: Both boxes need to be delivered.

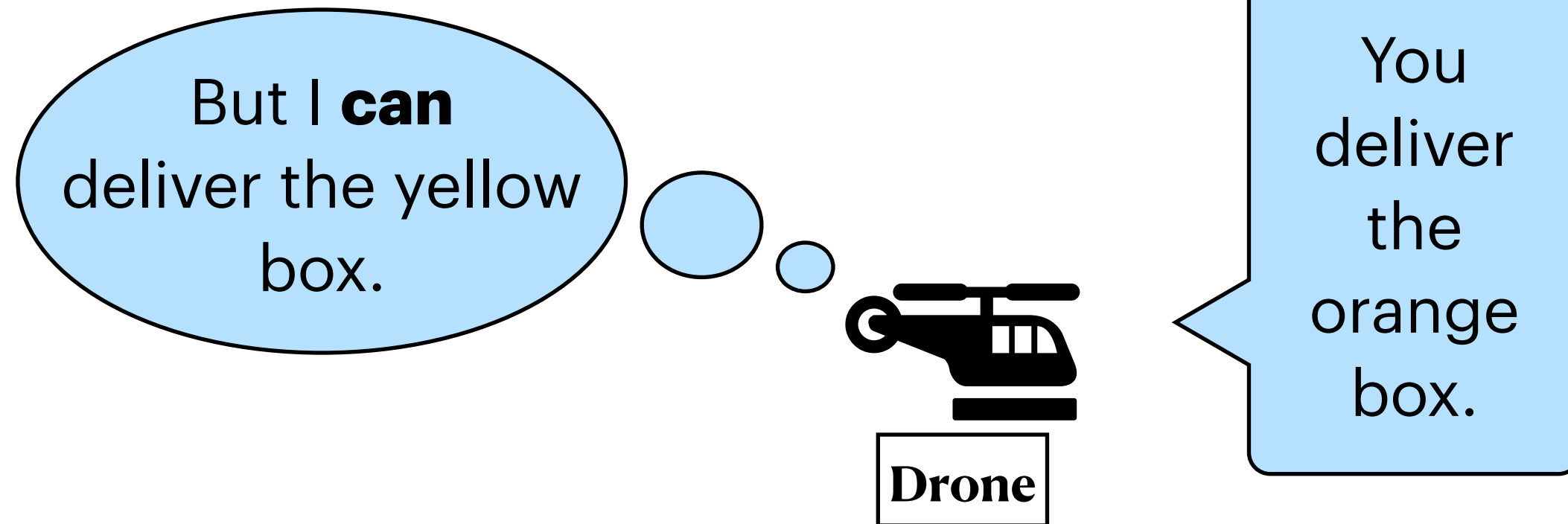


Distributed reactive synthesis is undecidable

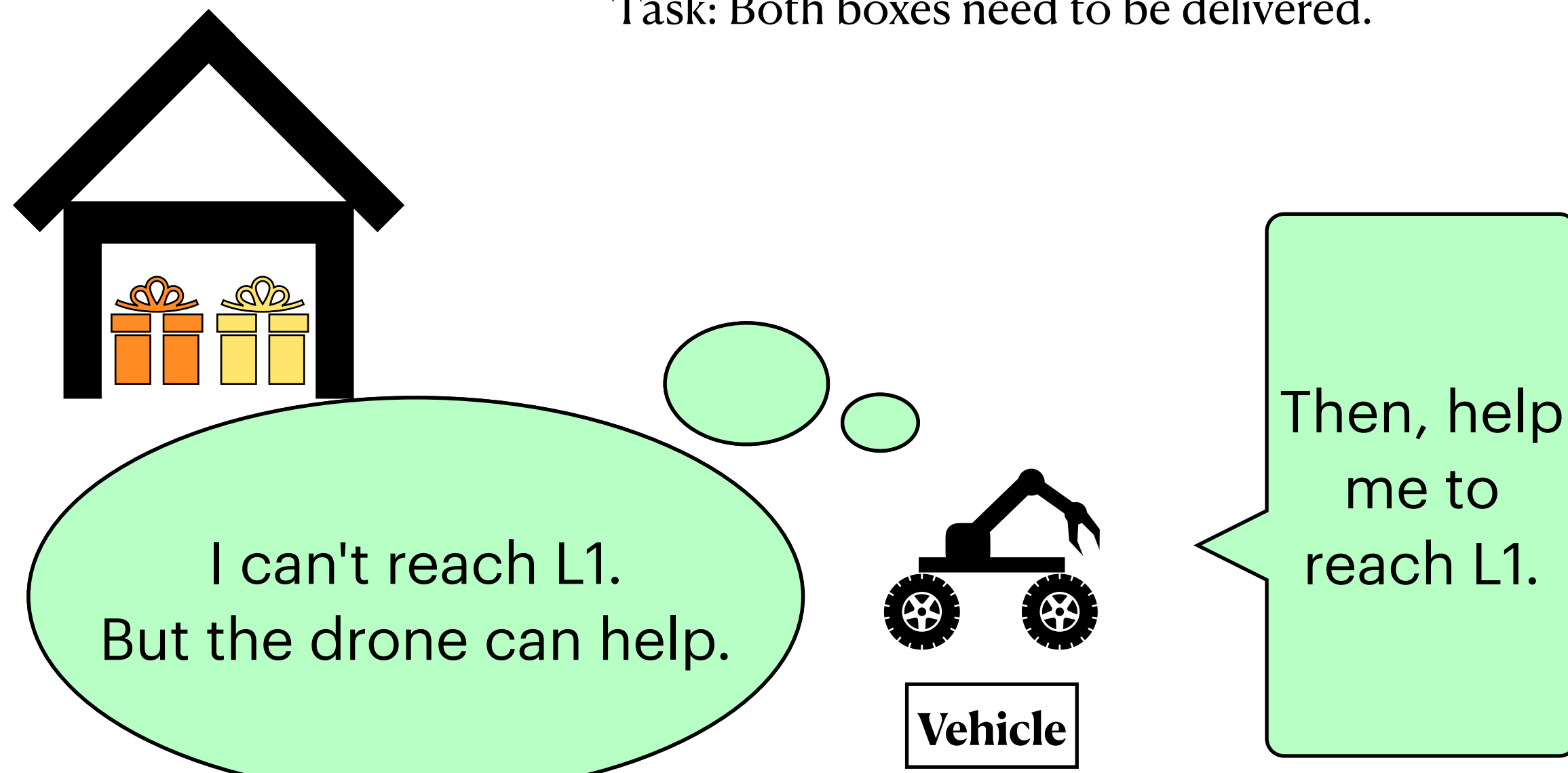
[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

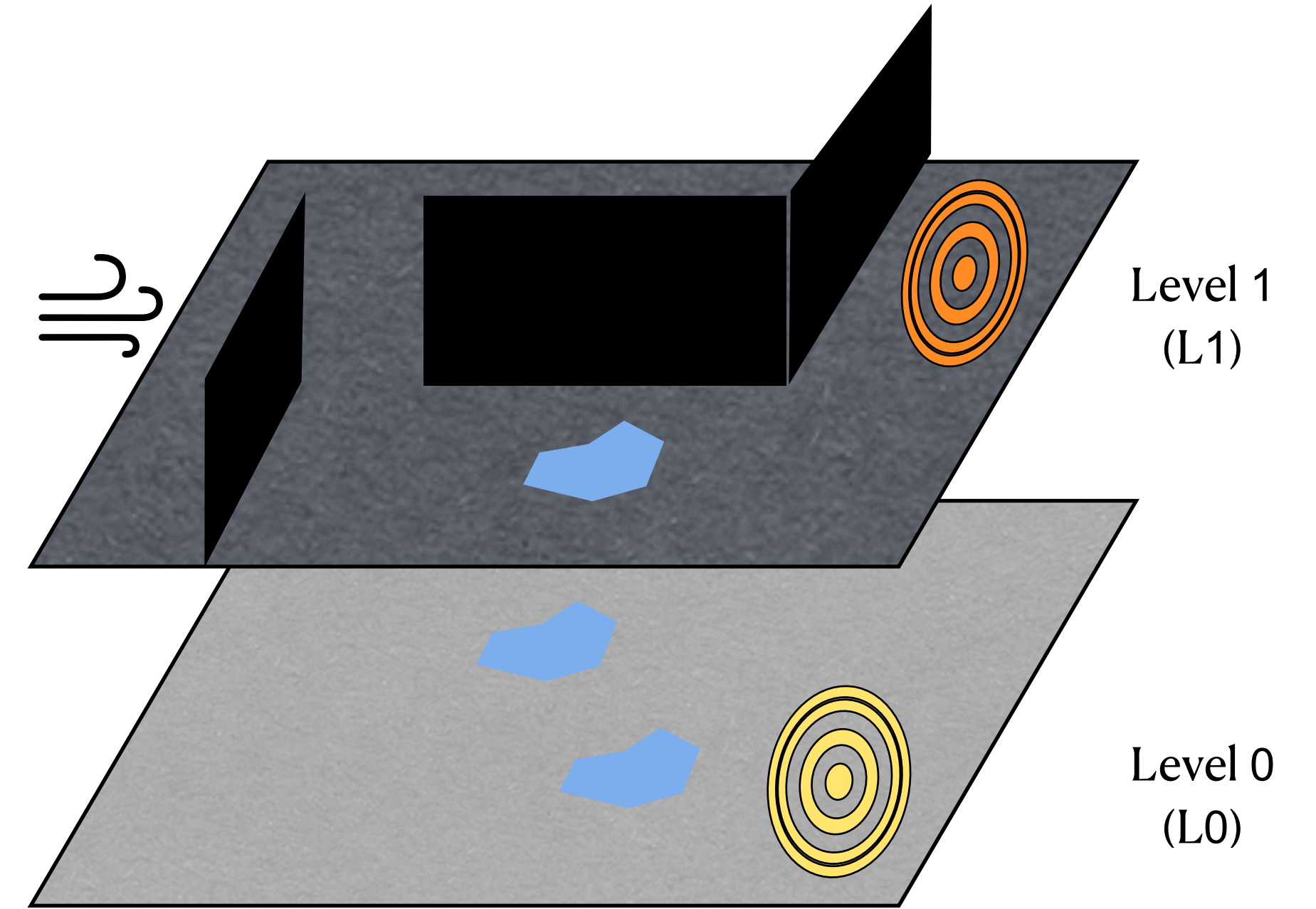
Negotiation: A Modular Approach



Task: Both boxes need to be delivered.



Task: Both boxes need to be delivered.

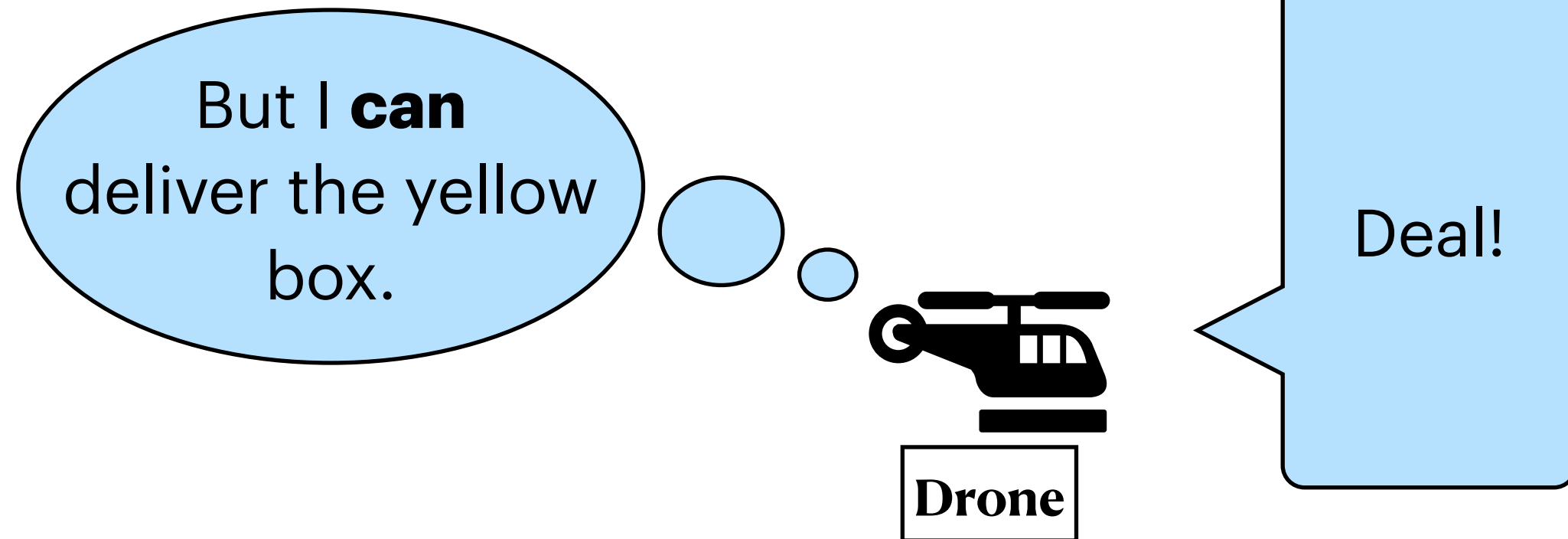


Distributed reactive synthesis is undecidable

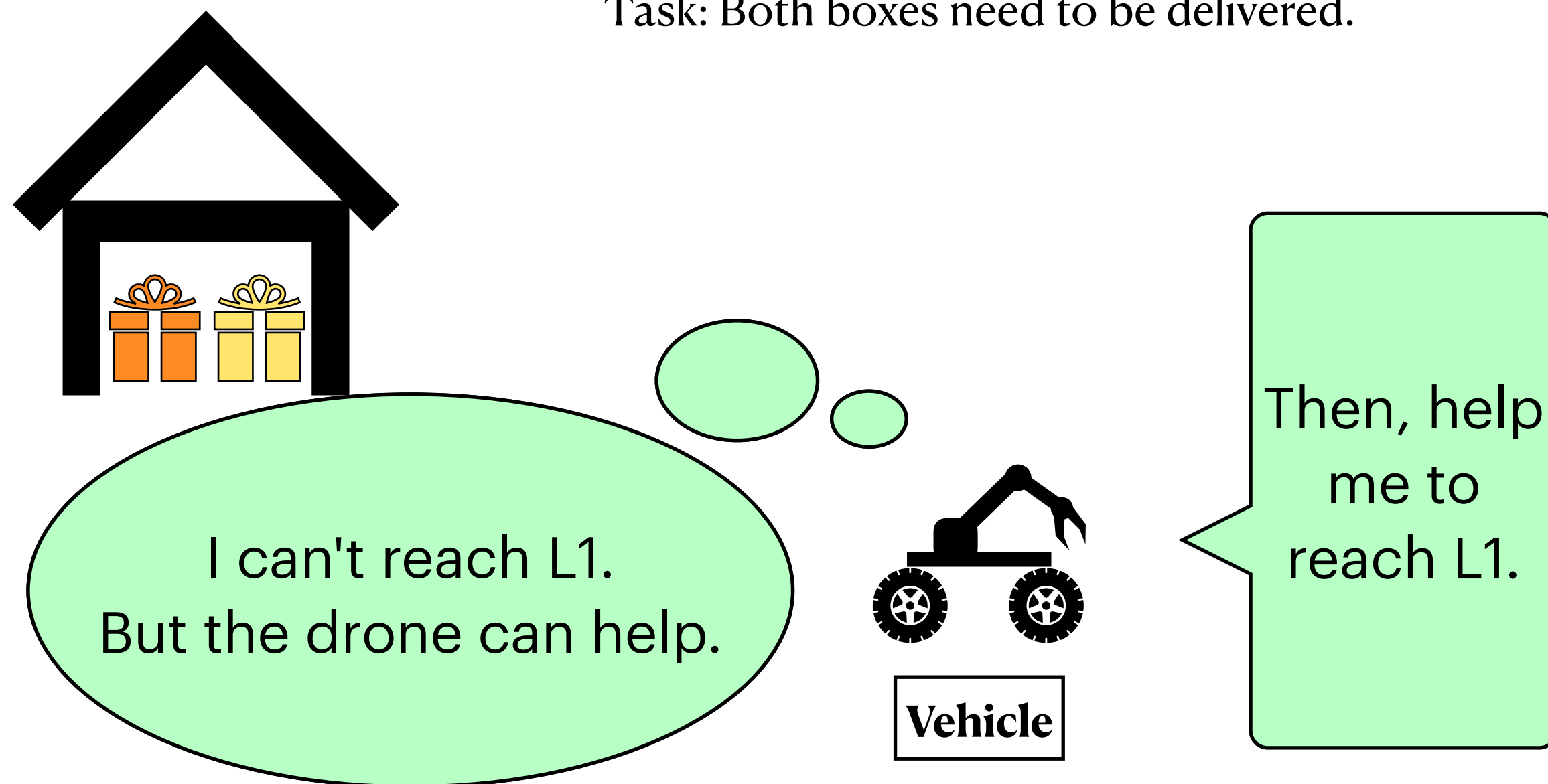
[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

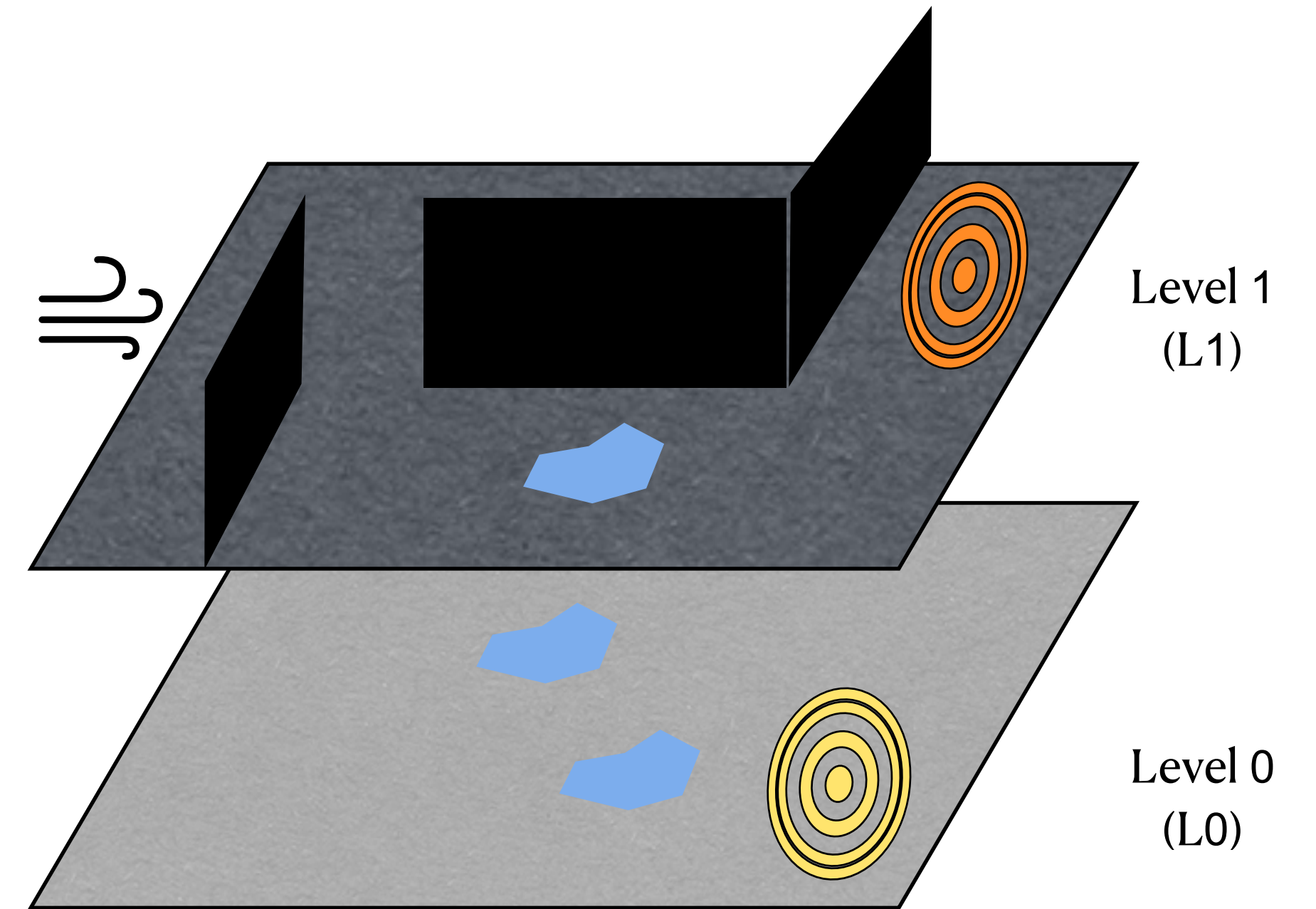
Negotiation: A Modular Approach



Task: Both boxes need to be delivered.



Task: Both boxes need to be delivered.

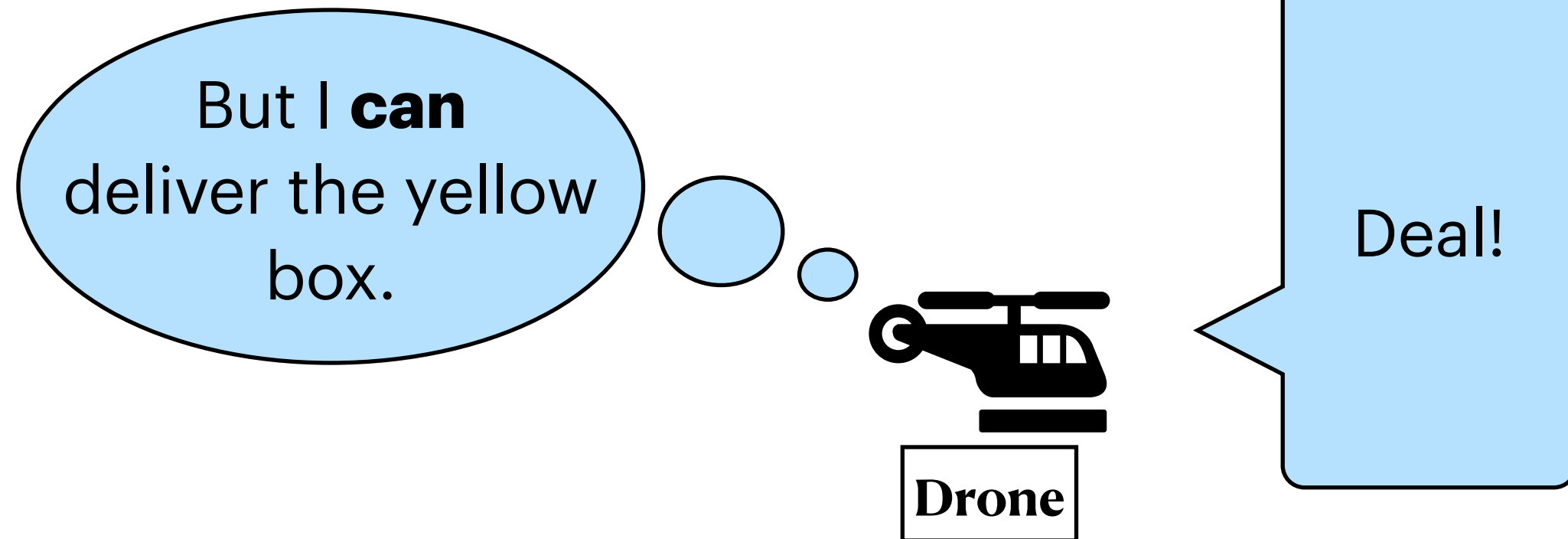


Distributed reactive synthesis is undecidable

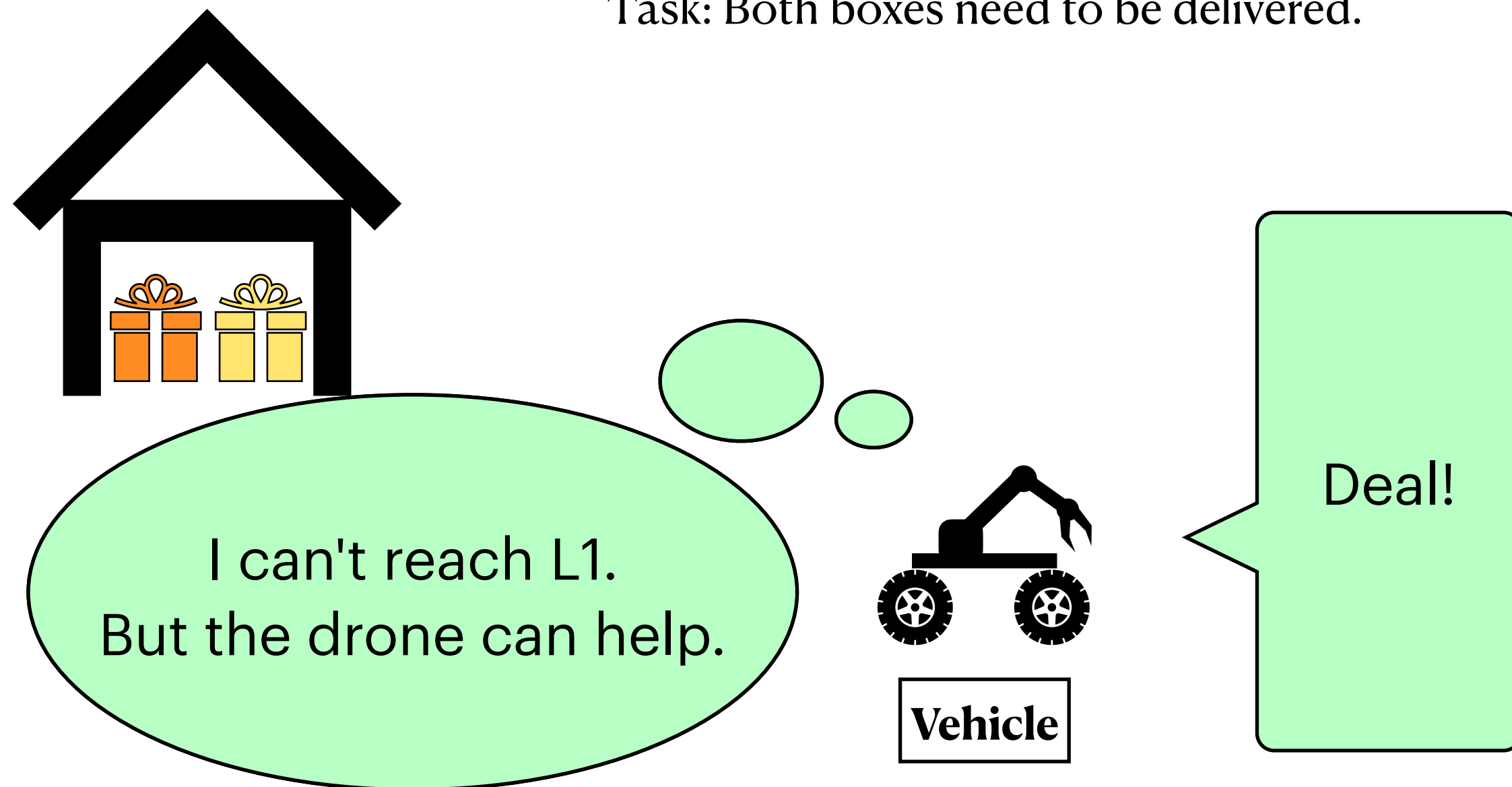
[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

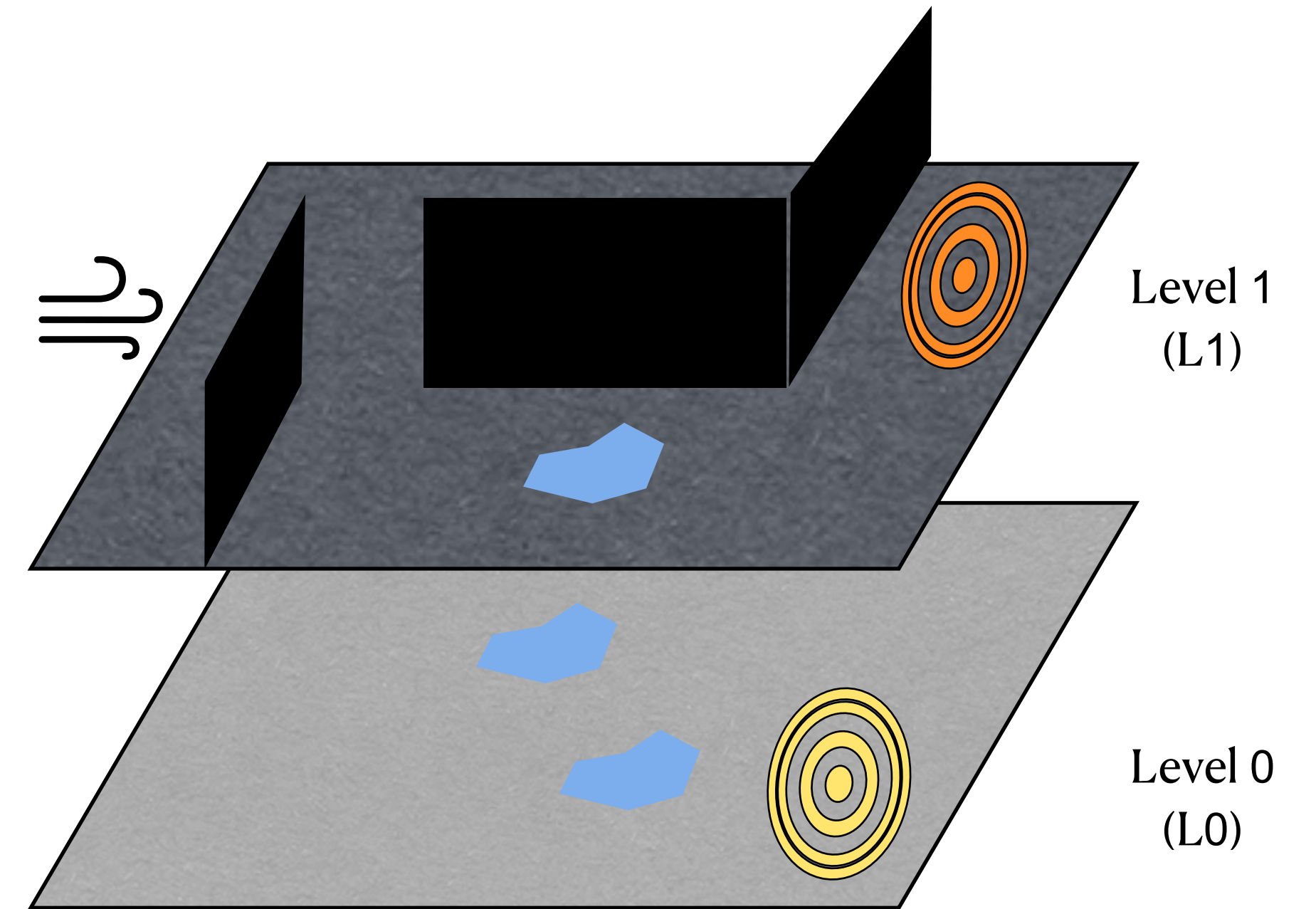
Negotiation: A Modular Approach



Task: Both boxes need to be delivered.



Task: Both boxes need to be delivered.

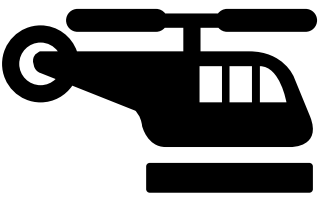


Distributed reactive synthesis is undecidable

[Pnueli and Rosner, 1990].

**Negotiation:
a sound, incomplete, and modular approach.**

Outcome of Negotiation: Assume-Guarantee Contracts



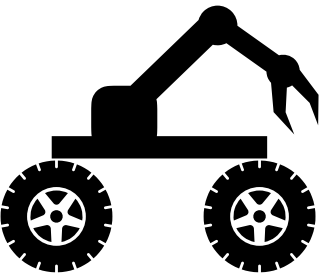
Drone

Task: given specification

You deliver the orange box.

The assume-guarantee *contract* for the drone:

(promise made by the vehicle, promise made by the drone)



Vehicle

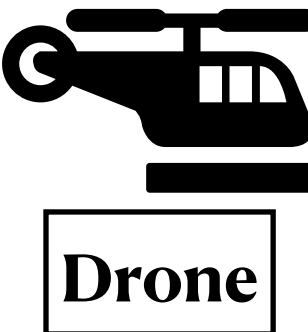
Task: given specification

Then, help me to reach L1.

The assume-guarantee *contract* for the vehicle:

(promise made by the drone, promise made by the vehicle)

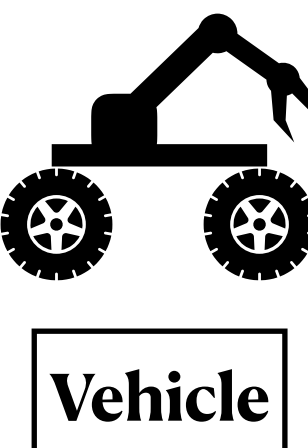
Outcome of Negotiation: Assume-Guarantee Contracts



You deliver the orange box.

Task: ~~given specification~~
(assumption → specification) ∧ contract

The assume-guarantee *contract* for the drone:
(promise made by the vehicle, promise made by the drone)

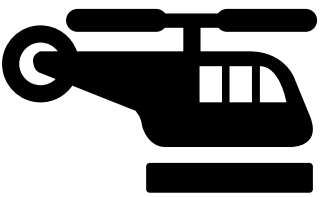


Then, help me to reach L1.

Task: ~~given specification~~
(assumption → specification) ∧ contract

The assume-guarantee *contract* for the vehicle:
(promise made by the drone, promise made by the vehicle)

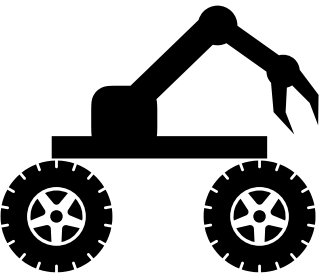
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
(assumption → specification) ∧ contract



Vehicle

Then, help me to reach L1.

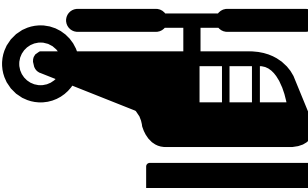
Task: ~~given specification~~
(assumption → specification) ∧ contract

The assume-guarantee *contract* for the drone:
(promise made by the vehicle, promise made by the drone)



The assume-guarantee *contract* for the vehicle:
(promise made by the drone, promise made by the vehicle)

Outcome of Negotiation: Assume-Guarantee Contracts

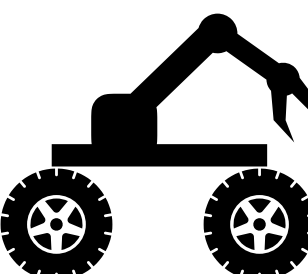


Drone

You deliver the orange box.

Task: ~~given specification~~

(assumption \rightarrow specification) \wedge contract



Vehicle

Then, help me to reach L1.

Task: ~~given specification~~

(assumption \rightarrow specification) \wedge contract

The assume-guarantee *contract* for the drone:
(promise made by the vehicle, promise made by the drone)



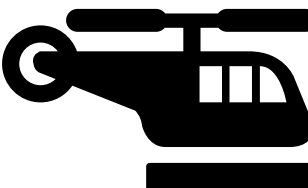
fulfill promise

if

vehicle kept its promise in the past

The assume-guarantee *contract* for the vehicle:
(promise made by the drone, promise made by the vehicle)

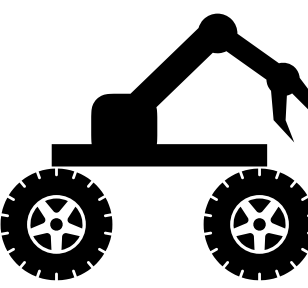
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
(assumption → specification) ∧ contract

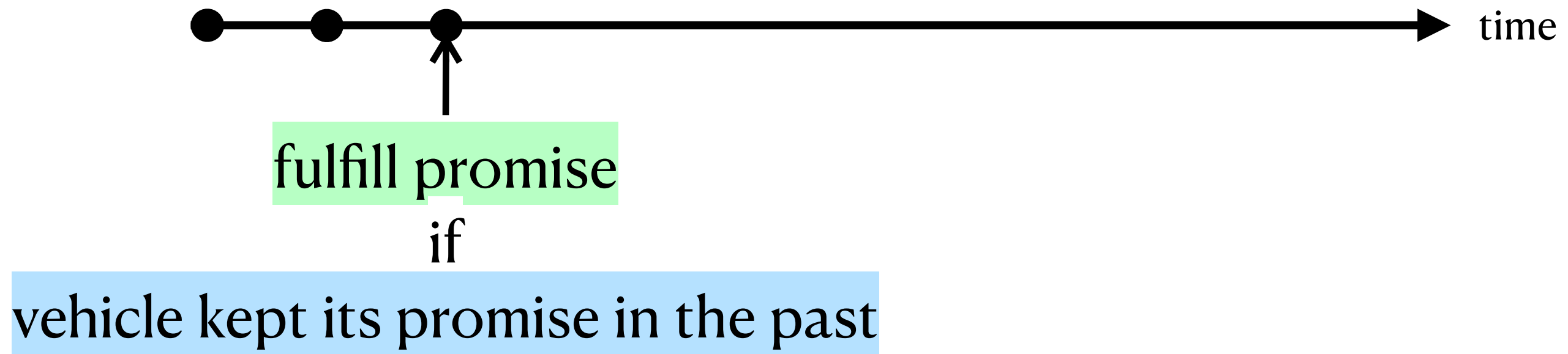


Vehicle

Then, help me to reach L1.

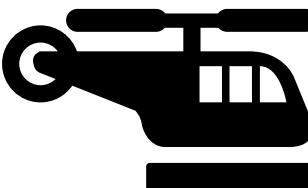
Task: ~~given specification~~
(assumption → specification) ∧ contract

The assume-guarantee *contract* for the drone:
(promise made by the vehicle, promise made by the drone)



The assume-guarantee *contract* for the vehicle:
(promise made by the drone, promise made by the vehicle)

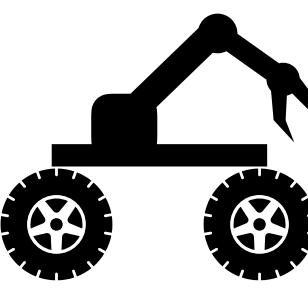
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
(assumption → specification) ∧ contract



Vehicle

Then, help me to reach L1.

Task: ~~given specification~~
(assumption → specification) ∧ contract

The assume-guarantee *contract* for the drone:
(promise made by the vehicle, promise made by the drone)



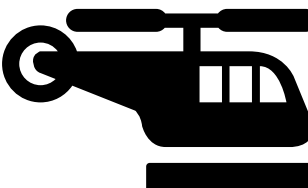
fulfill promise

if

vehicle kept its promise in the past

The assume-guarantee *contract* for the vehicle:
(promise made by the drone, promise made by the vehicle)

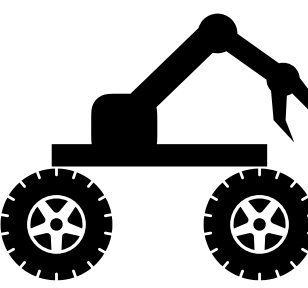
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
(assumption → specification) ∧ contract

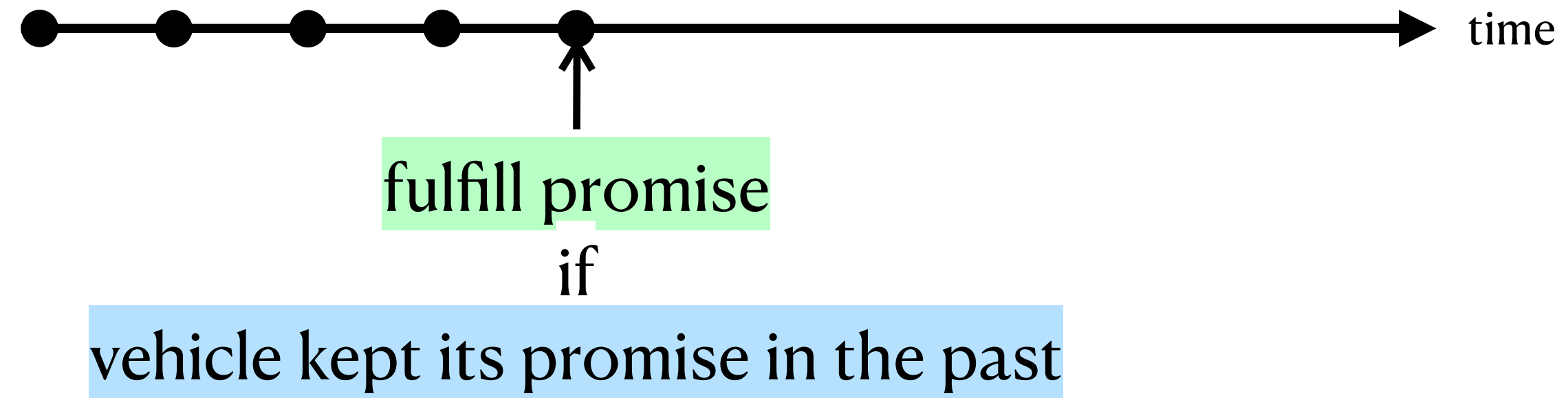


Vehicle

Then, help me to reach L1.

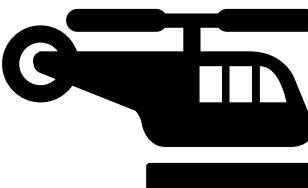
Task: ~~given specification~~
(assumption → specification) ∧ contract

The assume-guarantee *contract* for the drone:
(promise made by the vehicle, promise made by the drone)



The assume-guarantee *contract* for the vehicle:
(promise made by the drone, promise made by the vehicle)

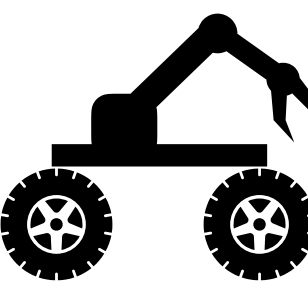
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

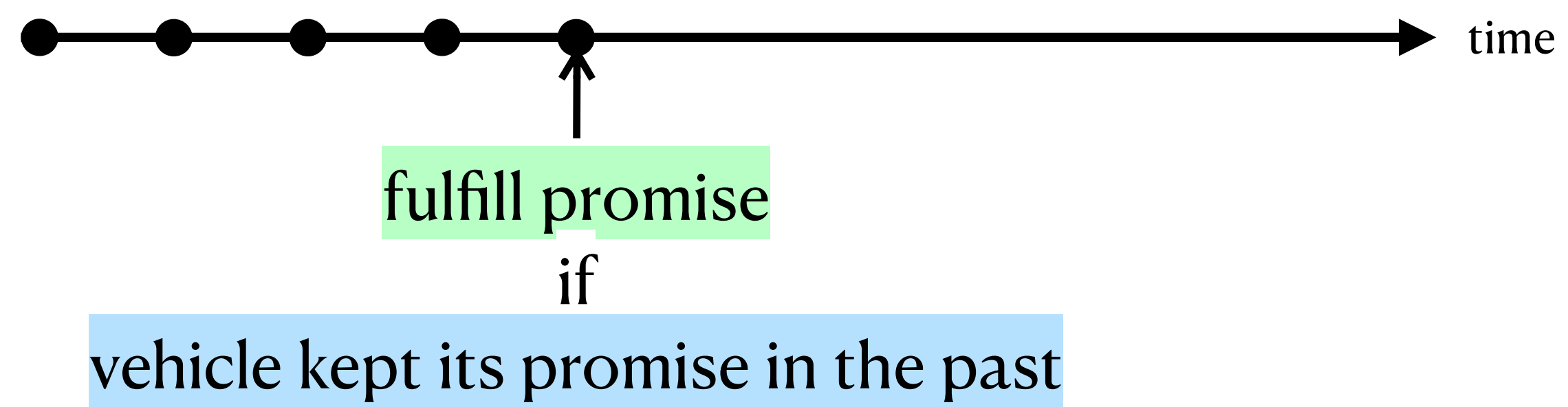


Vehicle

Then, help me to reach L1.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

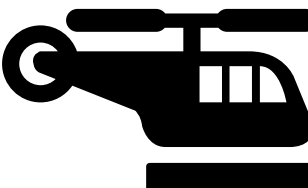
The assume-guarantee *contract* for the drone:
 (promise made by the vehicle, promise made by the drone)



The assume-guarantee *contract* for the vehicle:
 (promise made by the drone, promise made by the vehicle)



Outcome of Negotiation: Assume-Guarantee Contracts

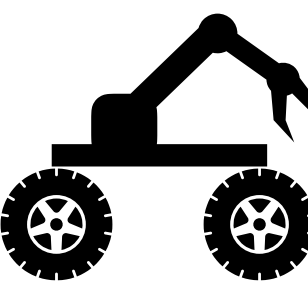


Drone

You deliver the orange box.

Task: ~~given specification~~

(assumption \rightarrow specification) \wedge contract



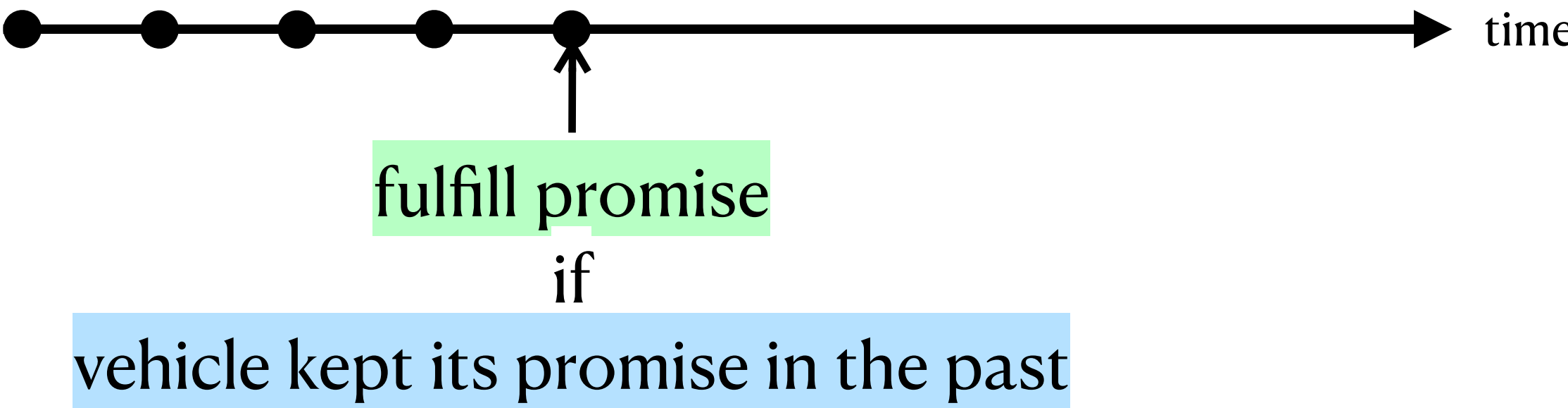
Vehicle

Then, help me to reach L1.

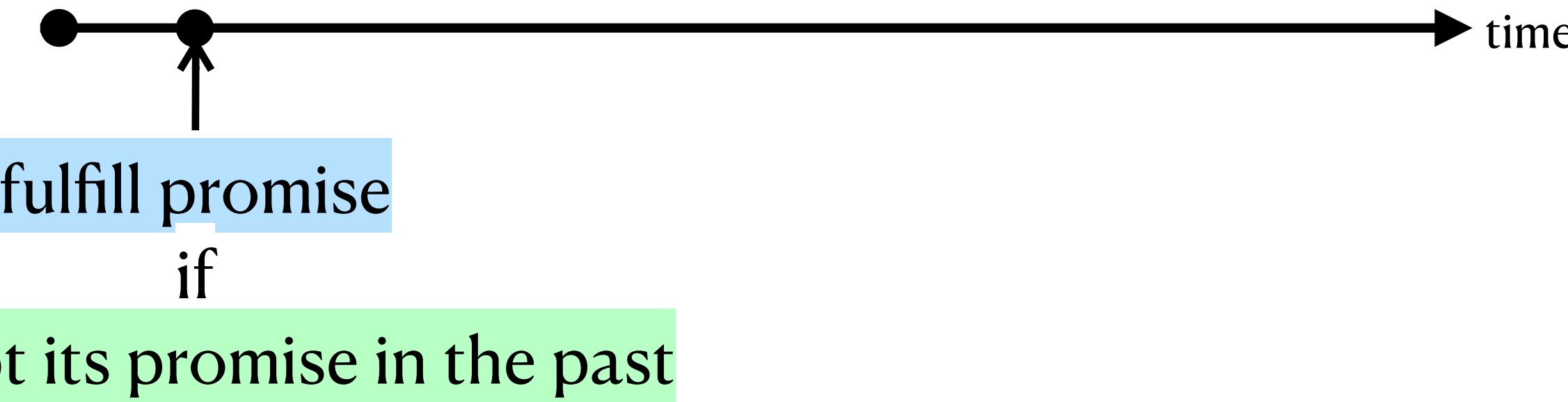
Task: ~~given specification~~

(assumption \rightarrow specification) \wedge contract

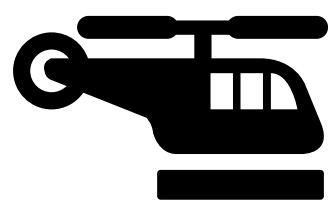
The assume-guarantee *contract* for the drone:
 (promise made by the vehicle, promise made by the drone)



The assume-guarantee *contract* for the vehicle:
 (promise made by the drone, promise made by the vehicle)



Outcome of Negotiation: Assume-Guarantee Contracts

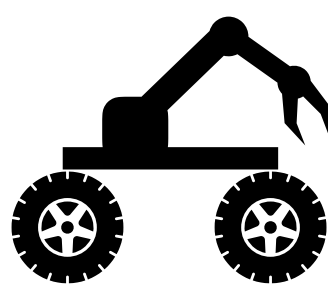


Drone

You deliver the orange box.

Task: ~~given specification~~

(assumption \rightarrow specification) \wedge contract



Vehicle

Then, help me to reach L1.

Task: ~~given specification~~

(assumption \rightarrow specification) \wedge contract

The assume-guarantee *contract* for the drone:

(promise made by the vehicle, promise made by the drone)



fulfill promise

if

vehicle kept its promise in the past

The assume-guarantee *contract* for the vehicle:

(promise made by the drone, promise made by the vehicle)

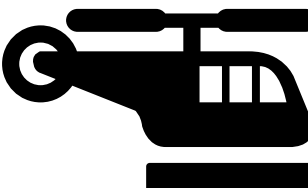


fulfill promise

if

drone kept its promise in the past

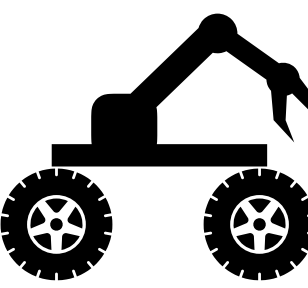
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

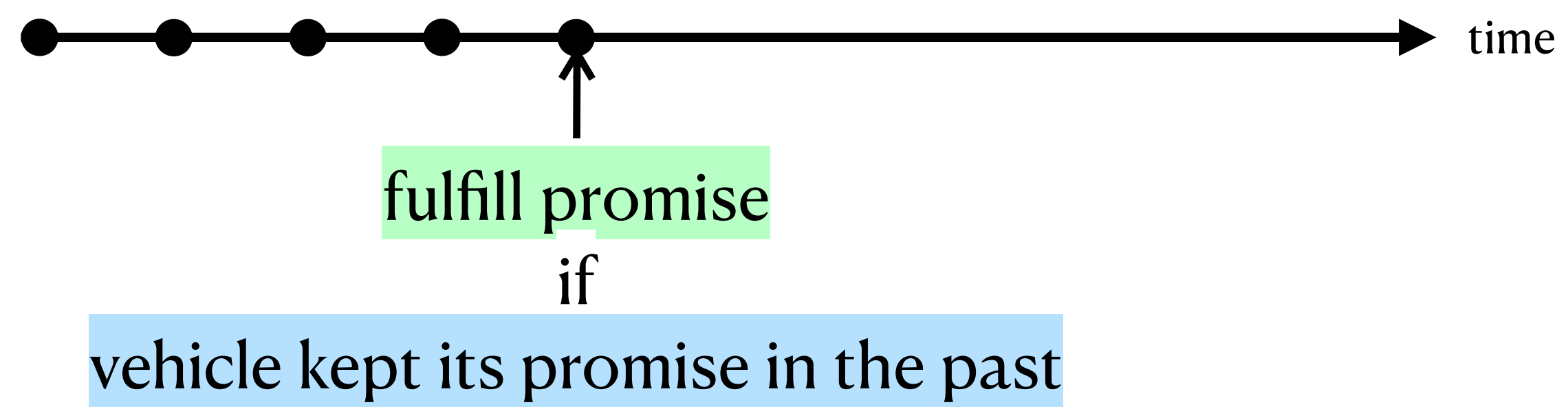


Vehicle

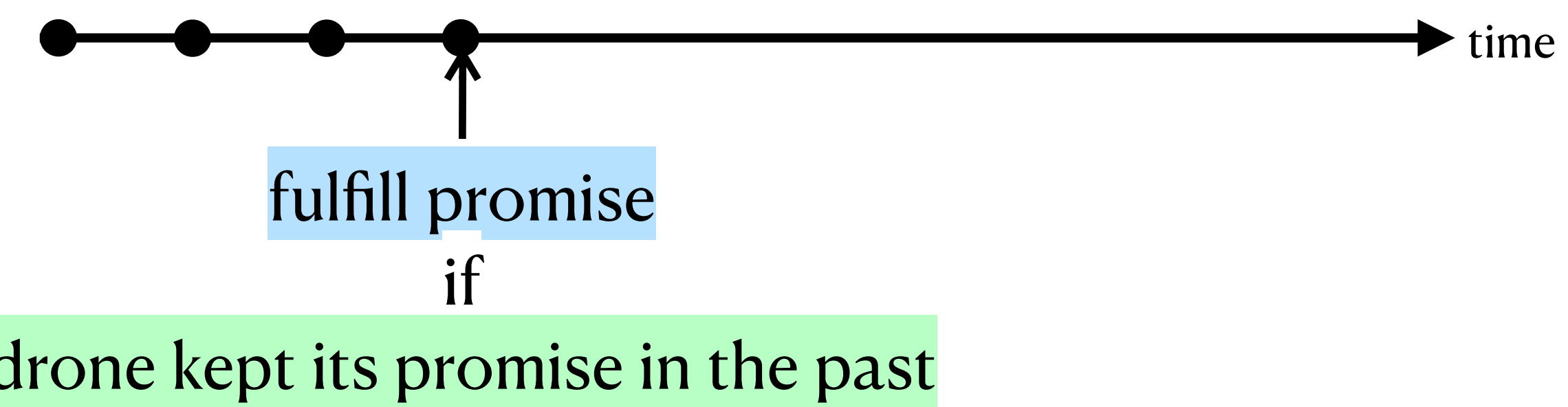
Then, help me to reach L1.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

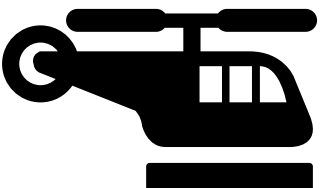
The assume-guarantee *contract* for the drone:
 (promise made by the vehicle, promise made by the drone)



The assume-guarantee *contract* for the vehicle:
 (promise made by the drone, promise made by the vehicle)



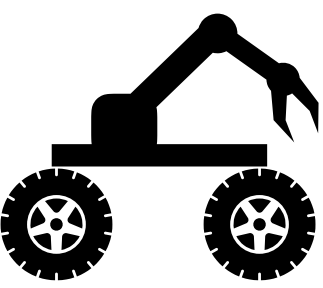
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

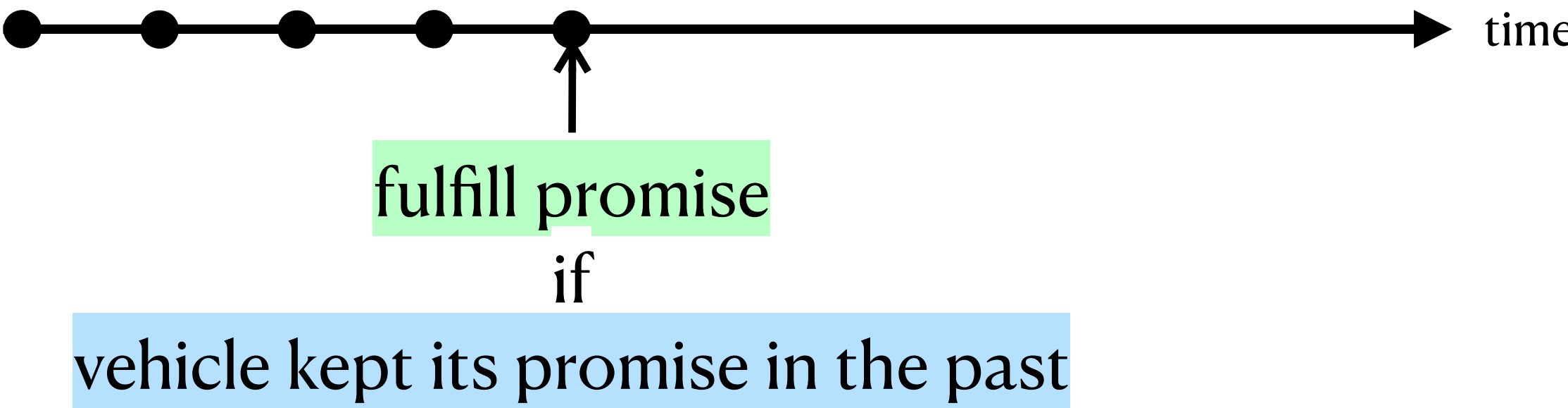


Vehicle

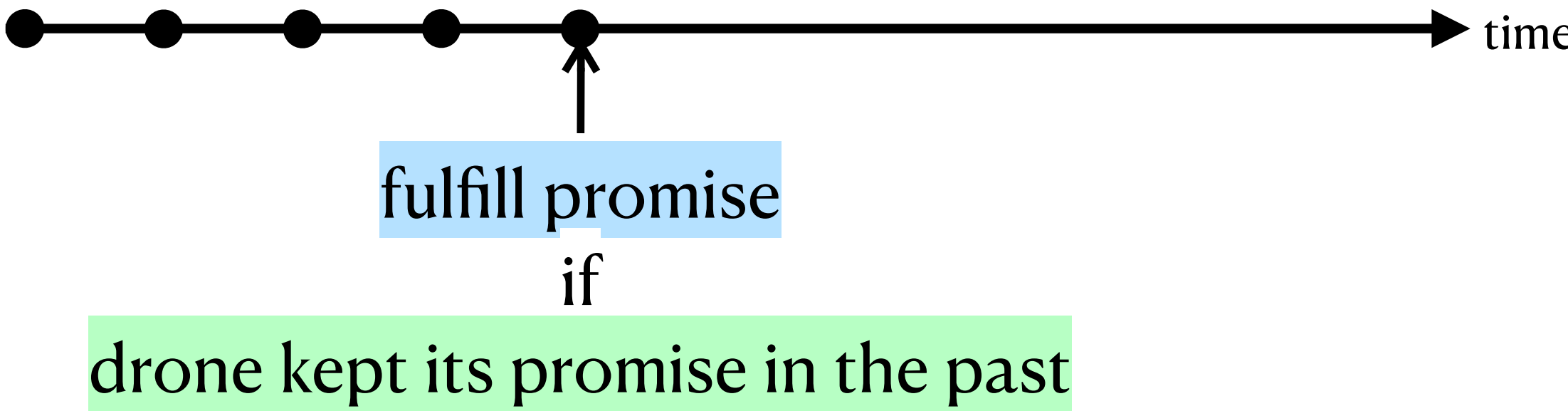
Then, help me to reach L1.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

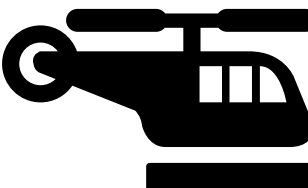
The assume-guarantee contract for the drone:
 (promise made by the vehicle, promise made by the drone)



The assume-guarantee contract for the vehicle:
 (promise made by the drone, promise made by the vehicle)



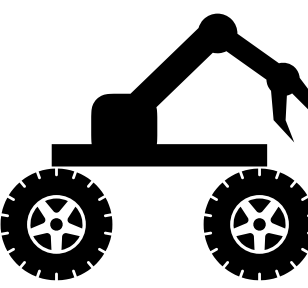
Outcome of Negotiation: Assume-Guarantee Contracts



Drone

You deliver the orange box.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

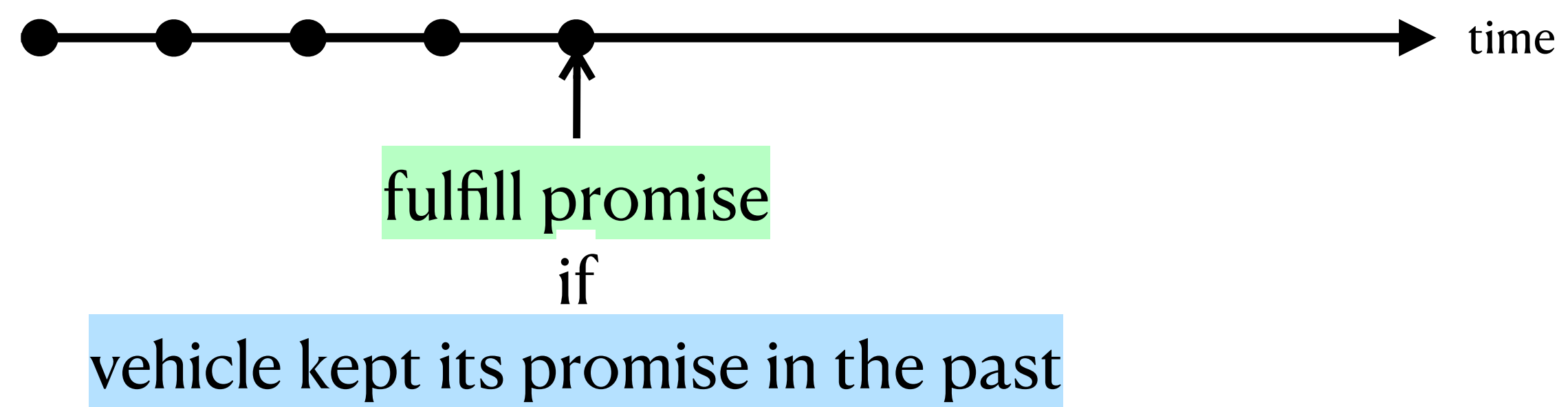


Vehicle

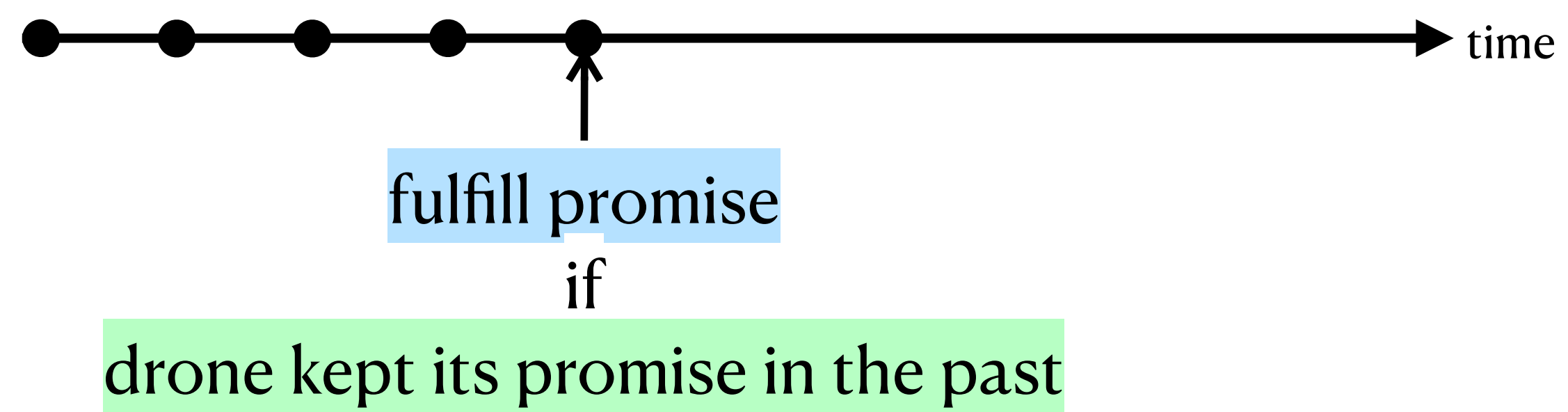
Then, help me to reach L1.

Task: ~~given specification~~
 (assumption \rightarrow specification) \wedge contract

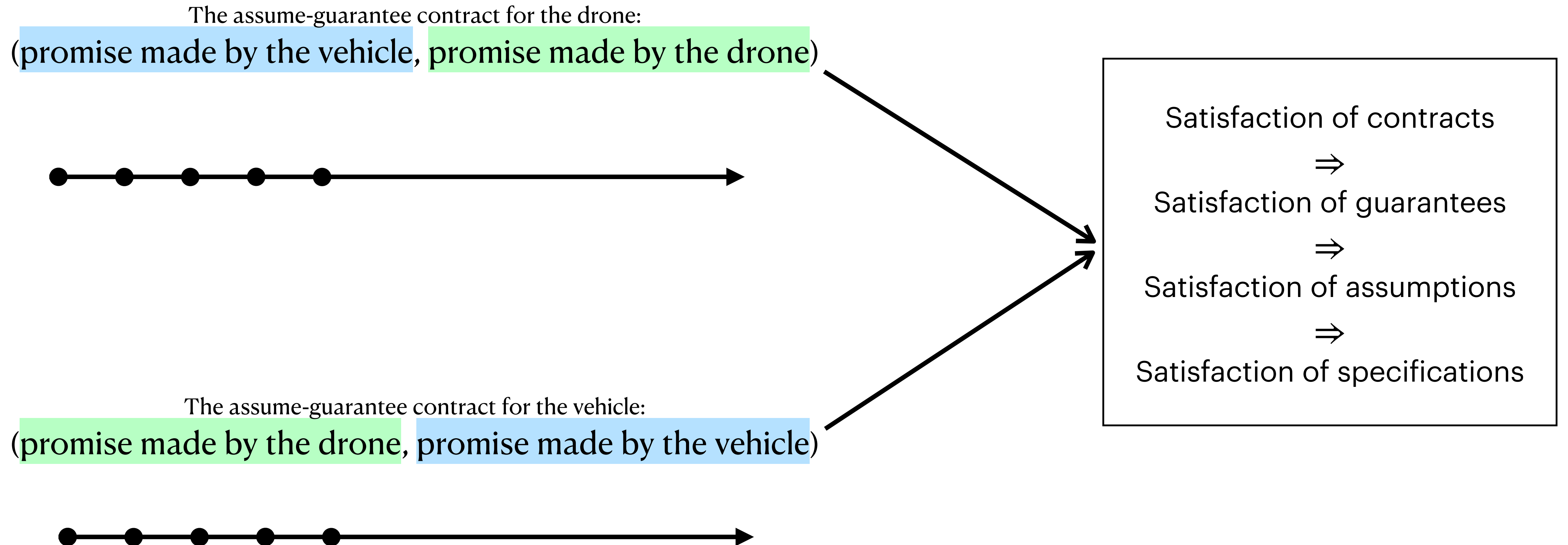
The assume-guarantee *contract* for the drone:
 (promise made by the vehicle, promise made by the drone)



The assume-guarantee *contract* for the vehicle:
 (promise made by the drone, promise made by the vehicle)



Satisfaction of Contracts



Satisfaction of Contracts

The assume-guarantee contract for the drone:

(promise made by the vehicle, promise made by the drone)

Satisfaction of contracts

● Related work (*one-way* interaction):

- Rajeev Alur, Salar Moarref, and Ufuk Topcu. "Pattern-based refinement of assume-guarantee specifications in reactive synthesis." TACAS 2015.

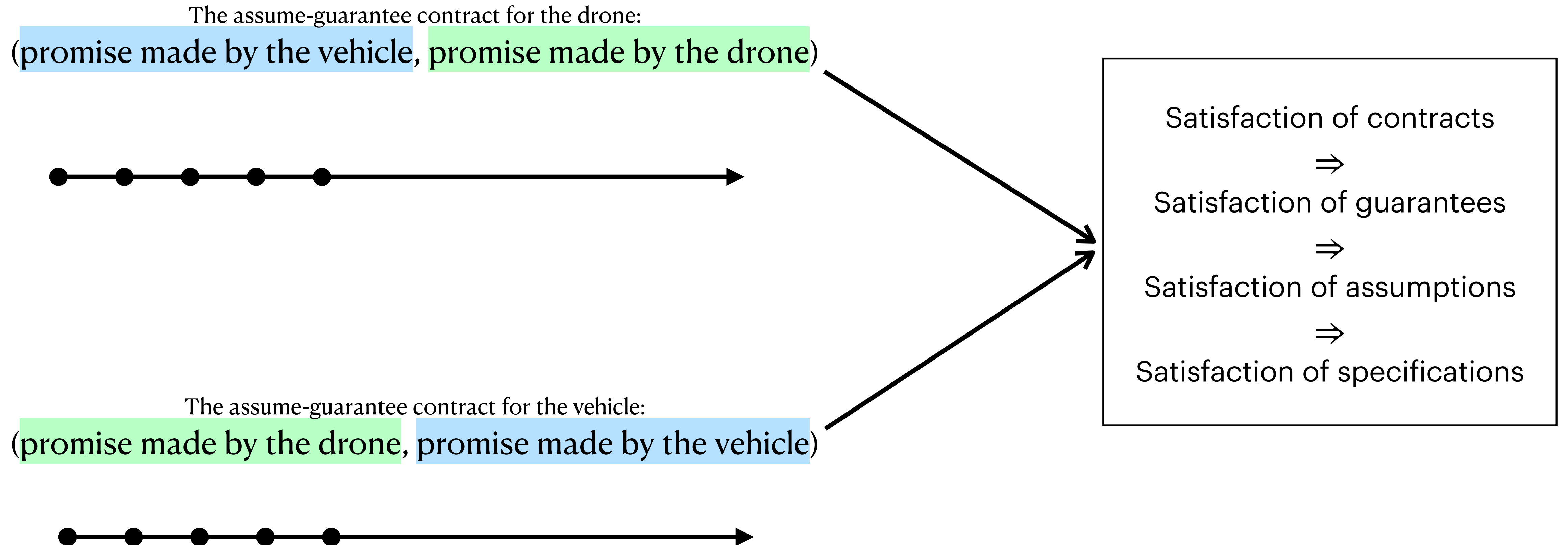
Satisfaction of specifications

The assume-guarantee contract for the vehicle:

(promise made by the drone, promise made by the vehicle)



Satisfaction of Contracts



The Problem

The *Negotiation* Algorithm

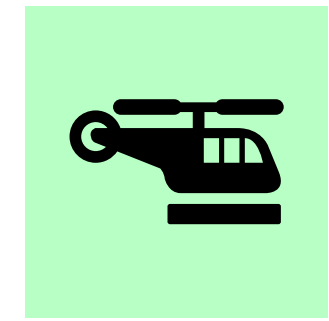
Experiments

The Problem

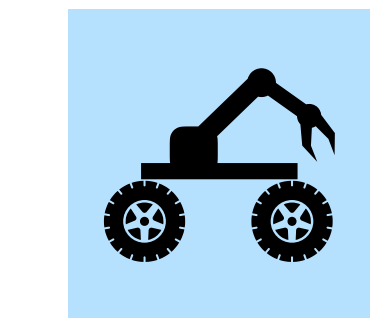
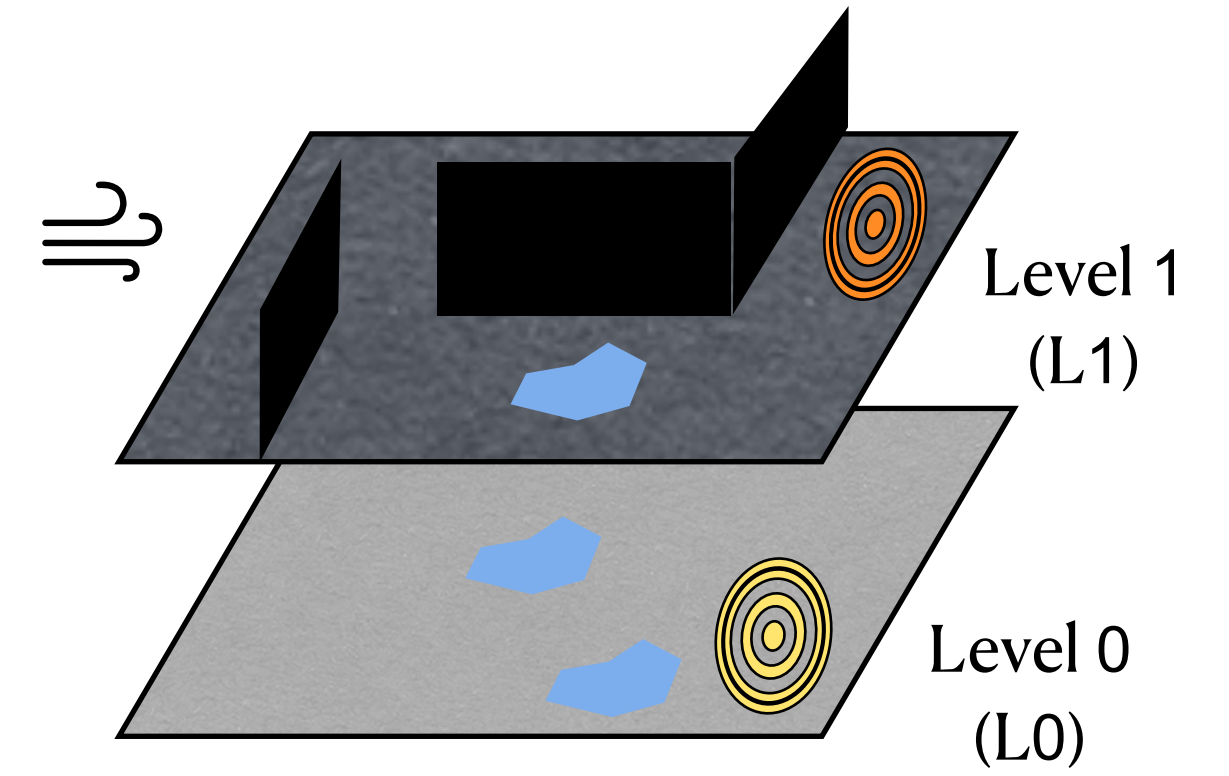
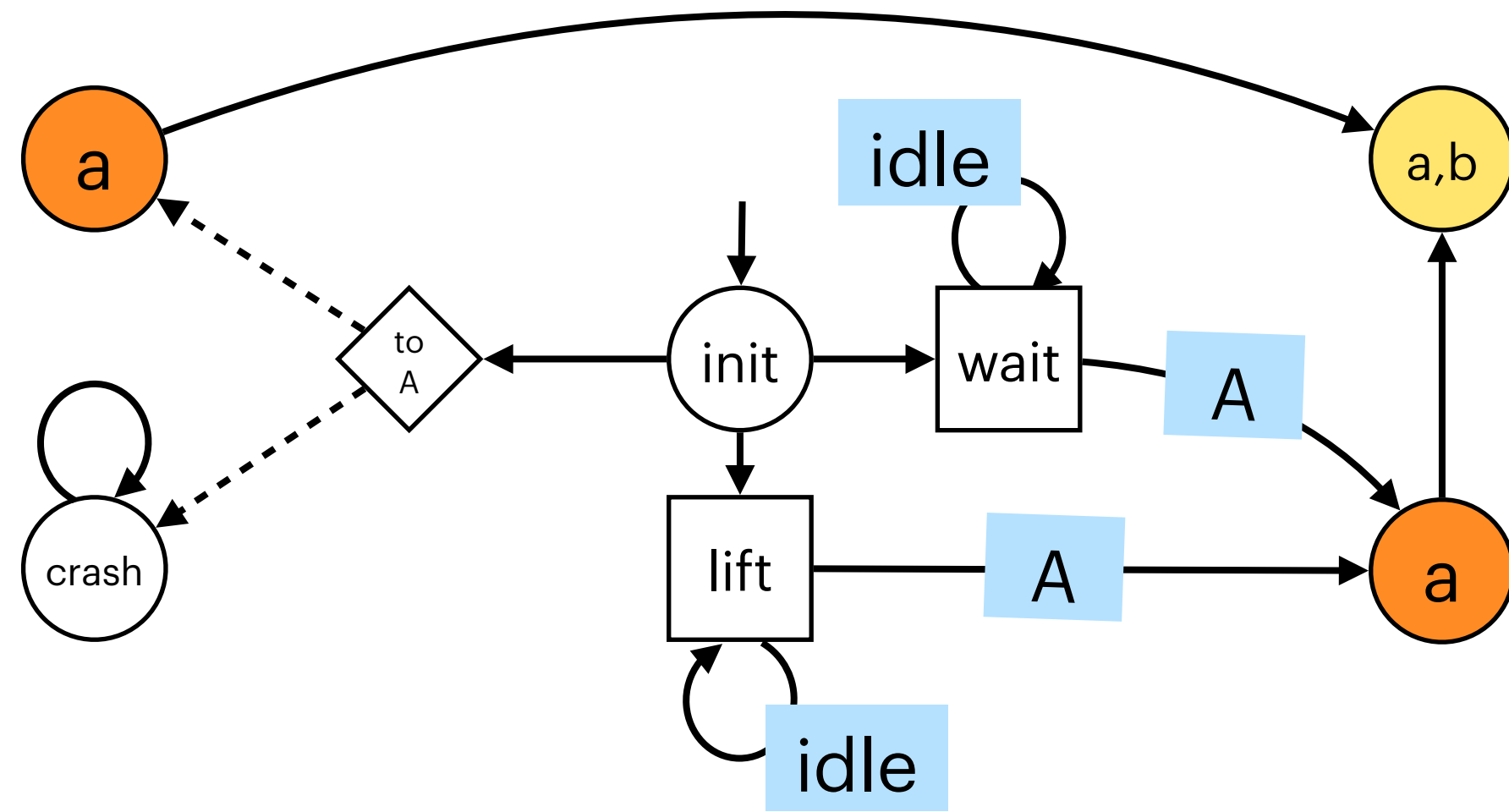
The *Negotiation* Algorithm

Experiments

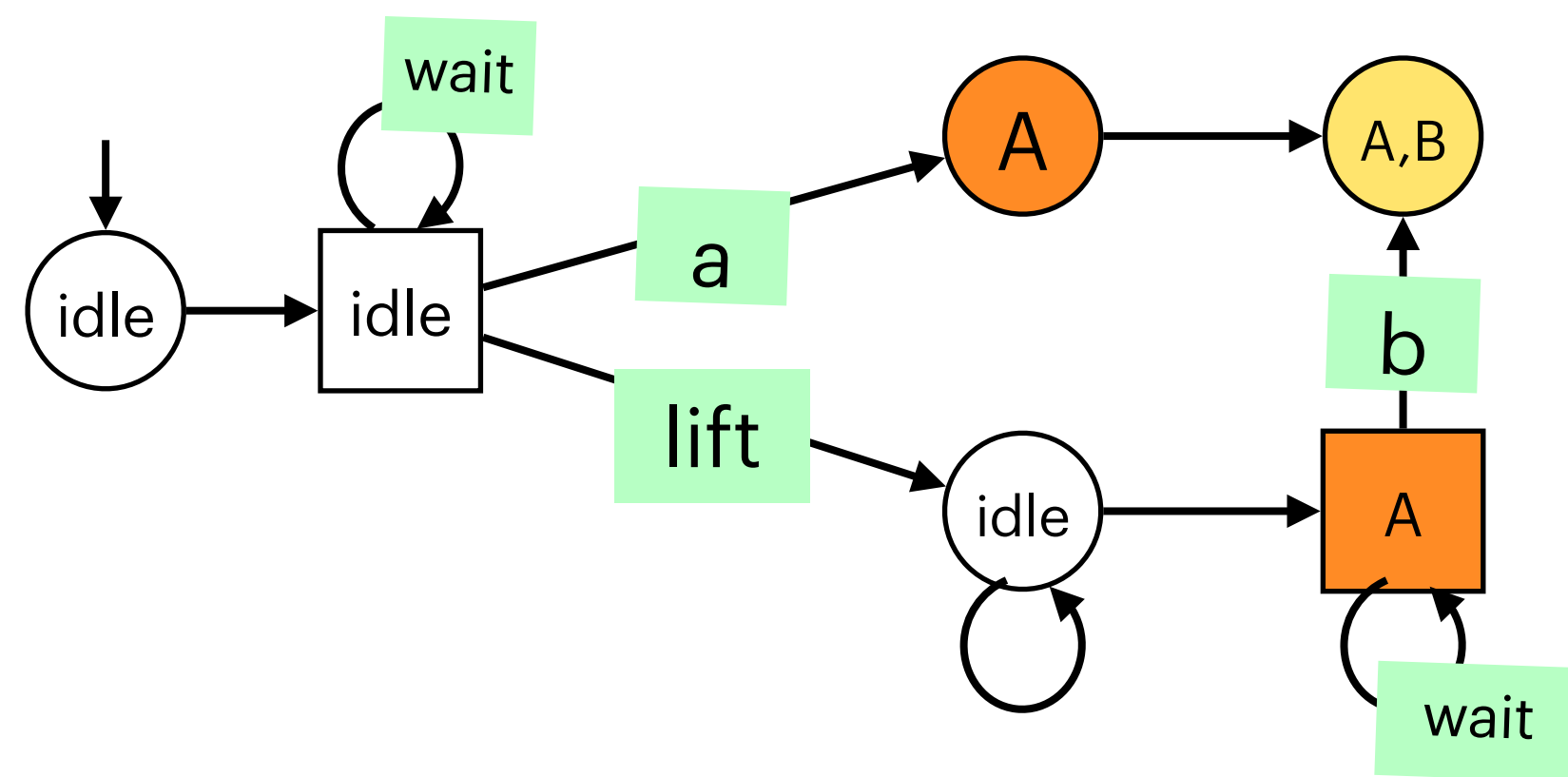
The Problem



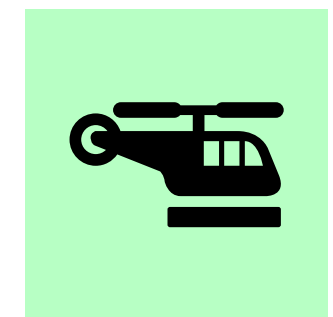
$$\varphi = \diamond(a, b)$$



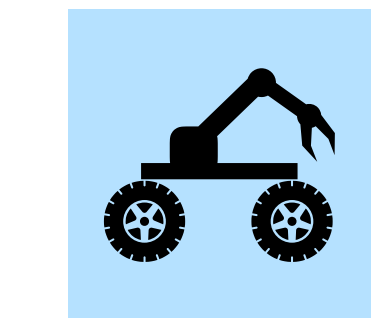
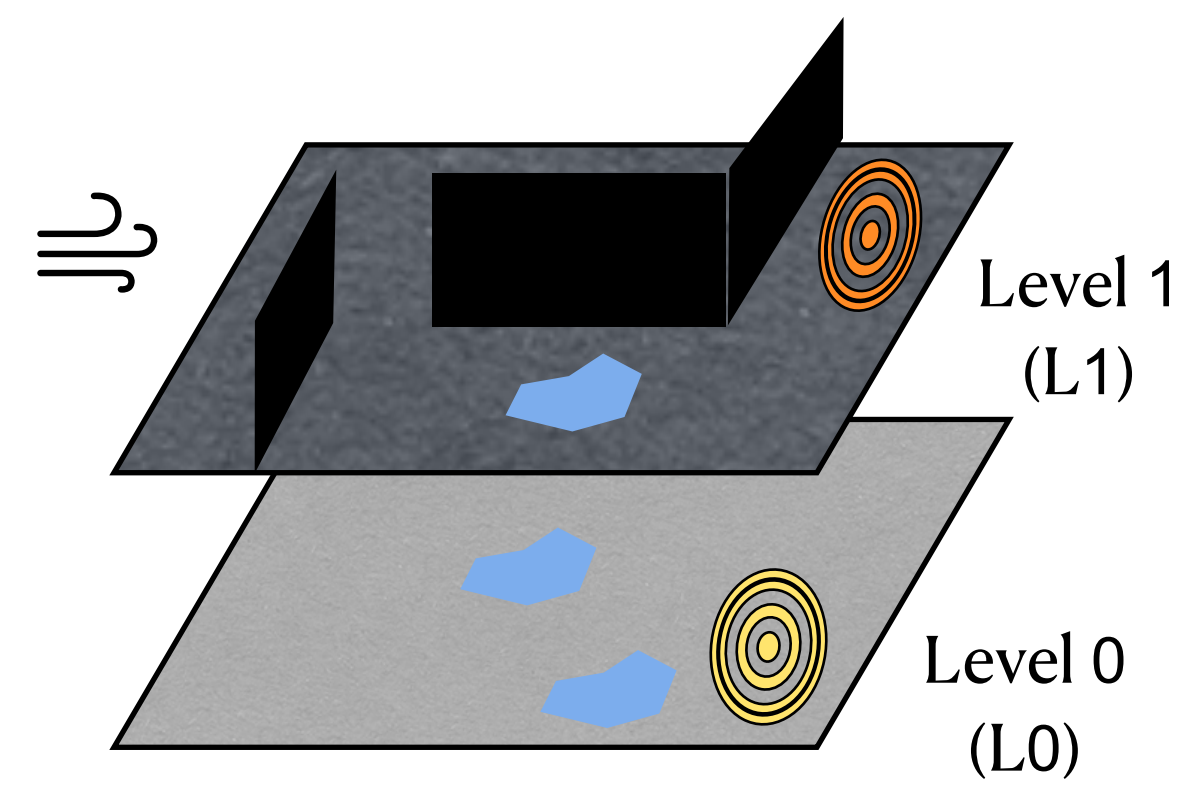
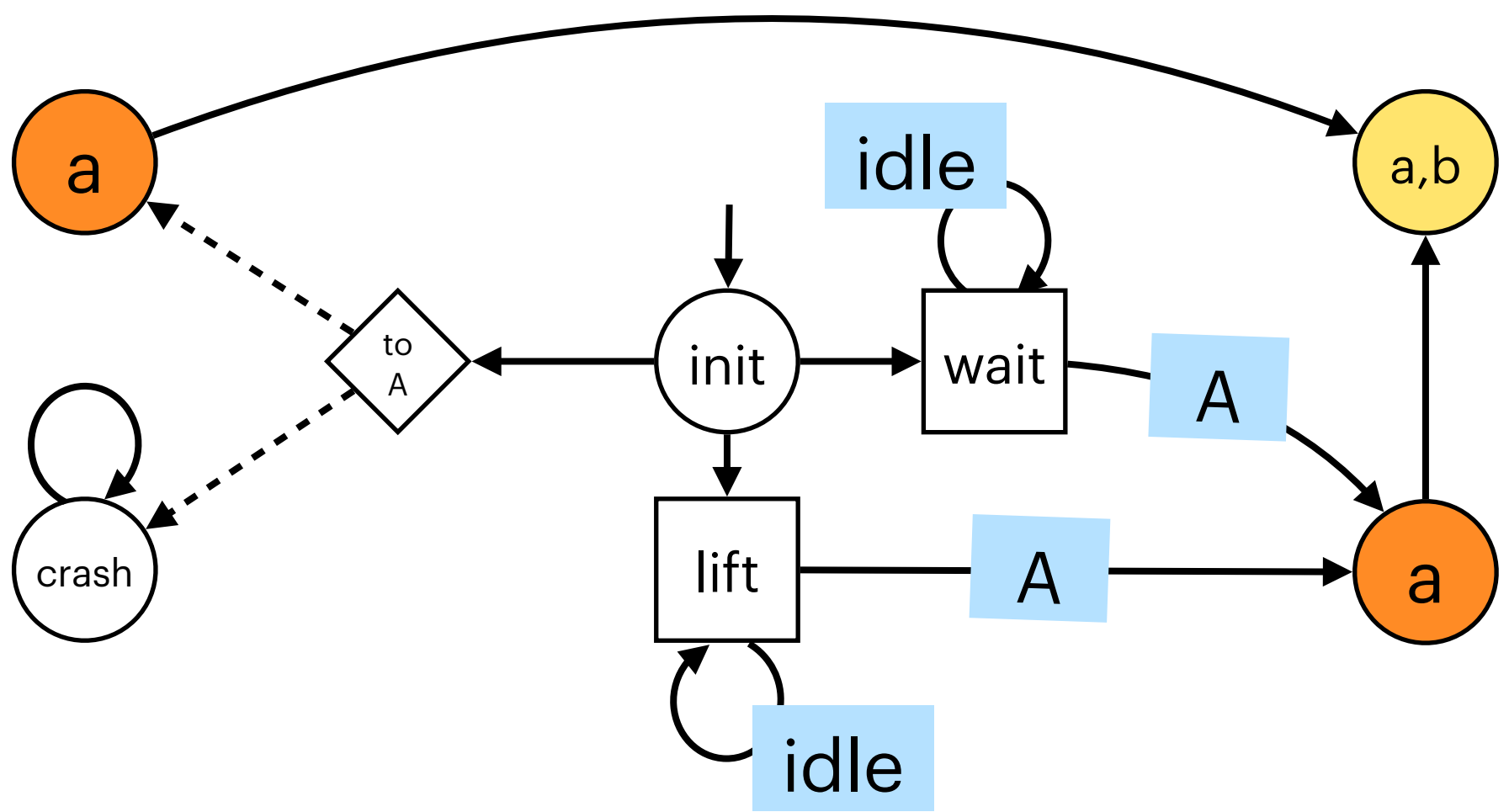
$$\psi = \diamond(A, B)$$



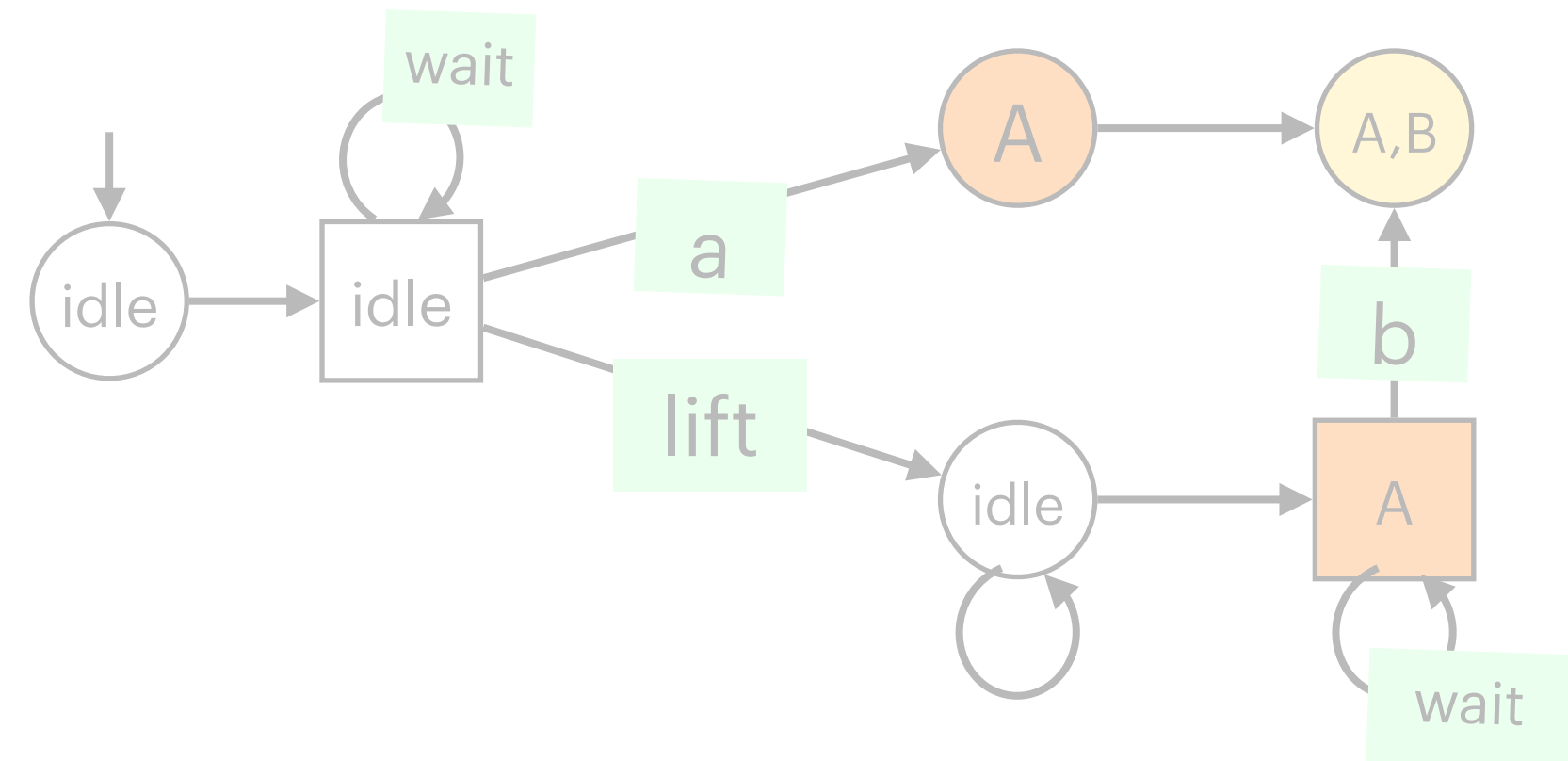
The Problem



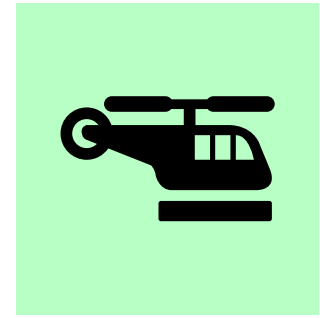
$$\varphi = \diamond(a, b)$$



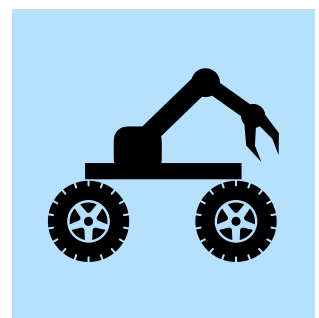
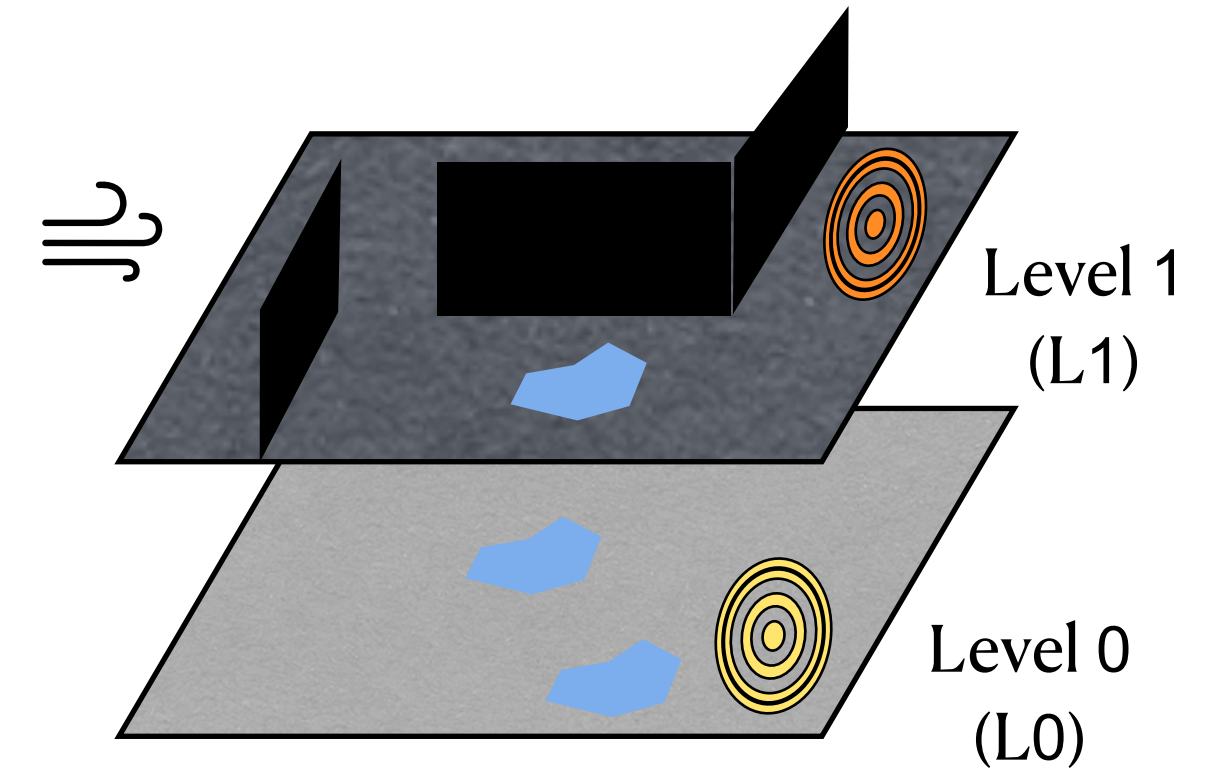
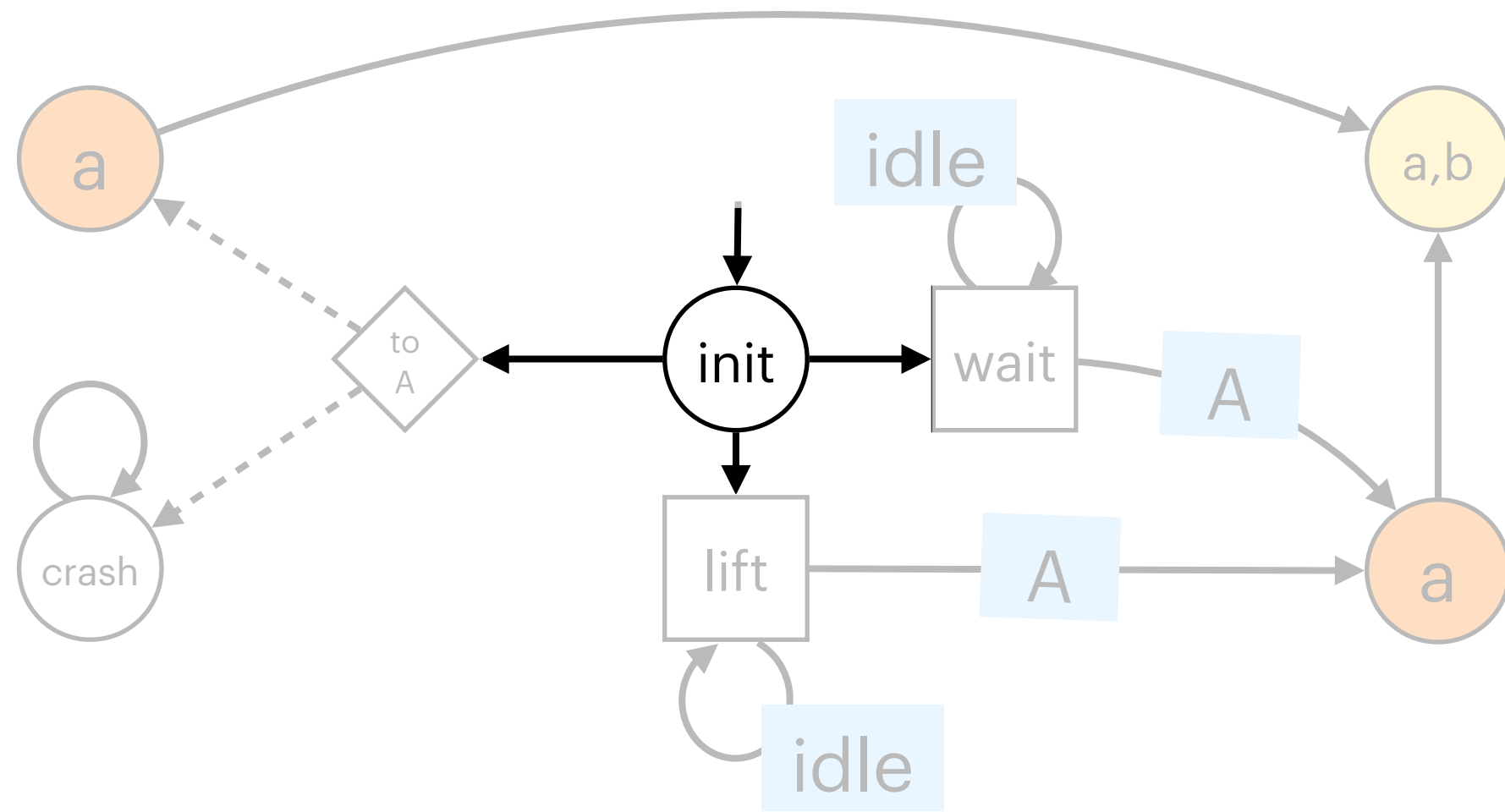
$$\psi = \diamond(A, B)$$



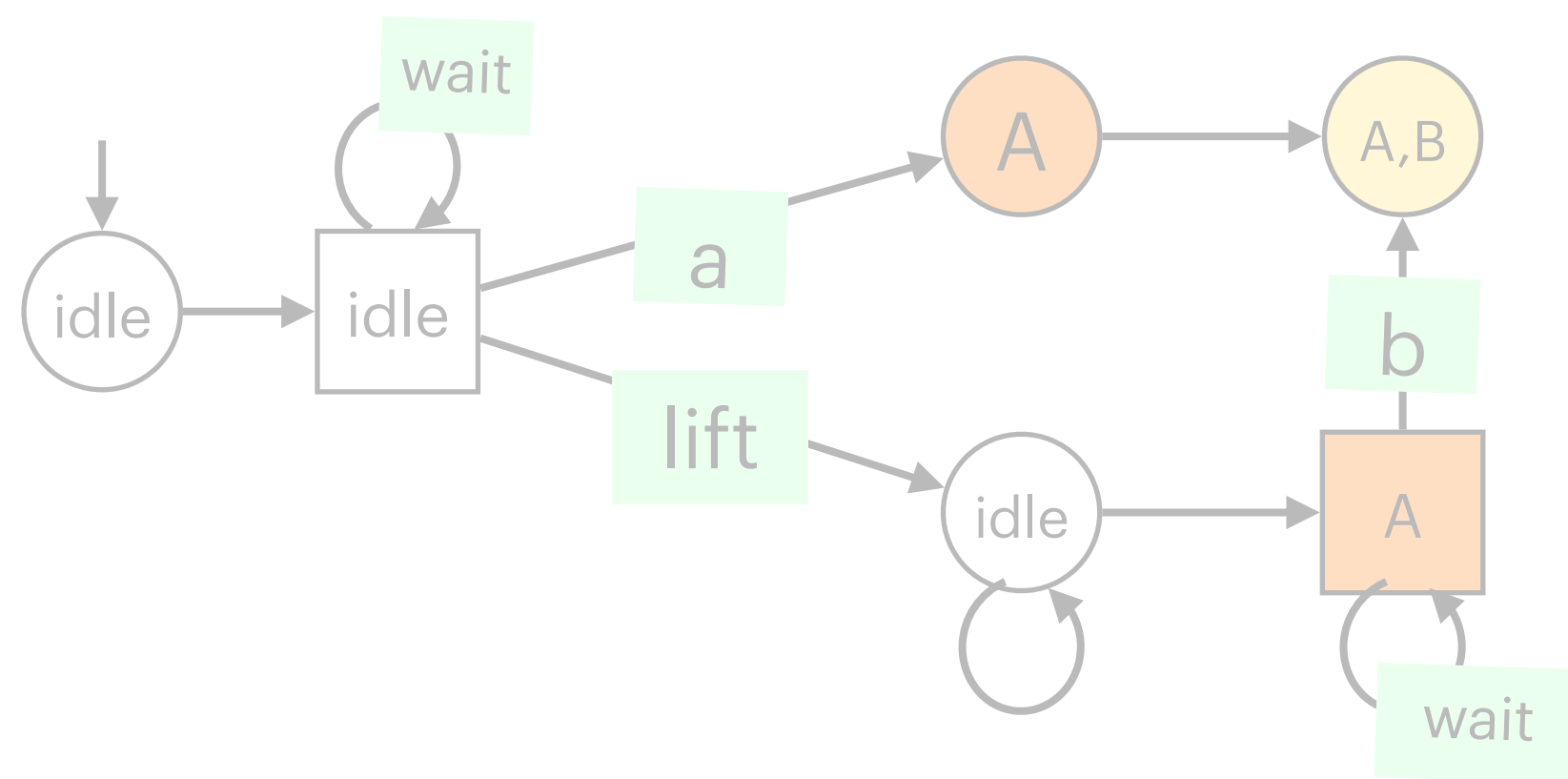
The Problem



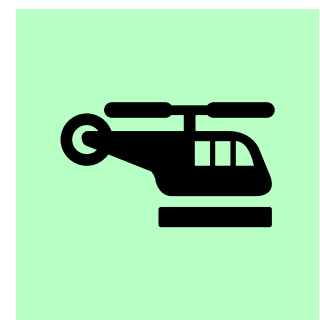
$$\varphi = \diamond(a, b)$$



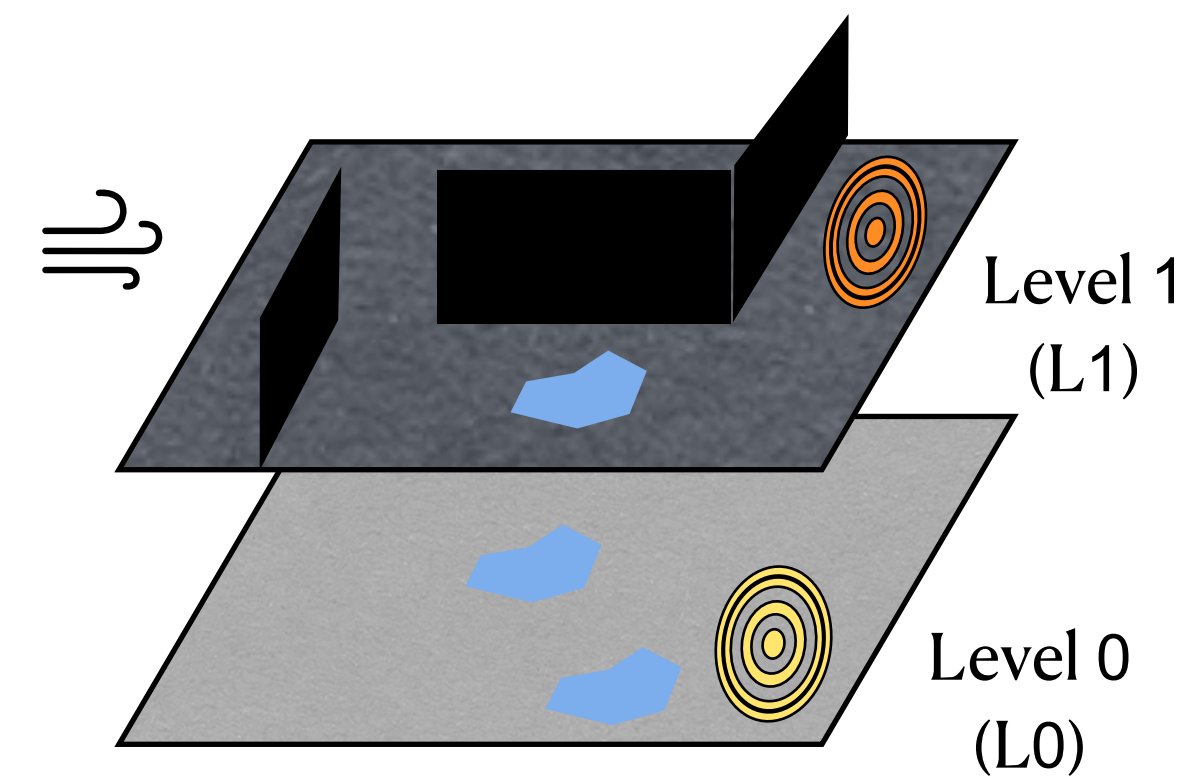
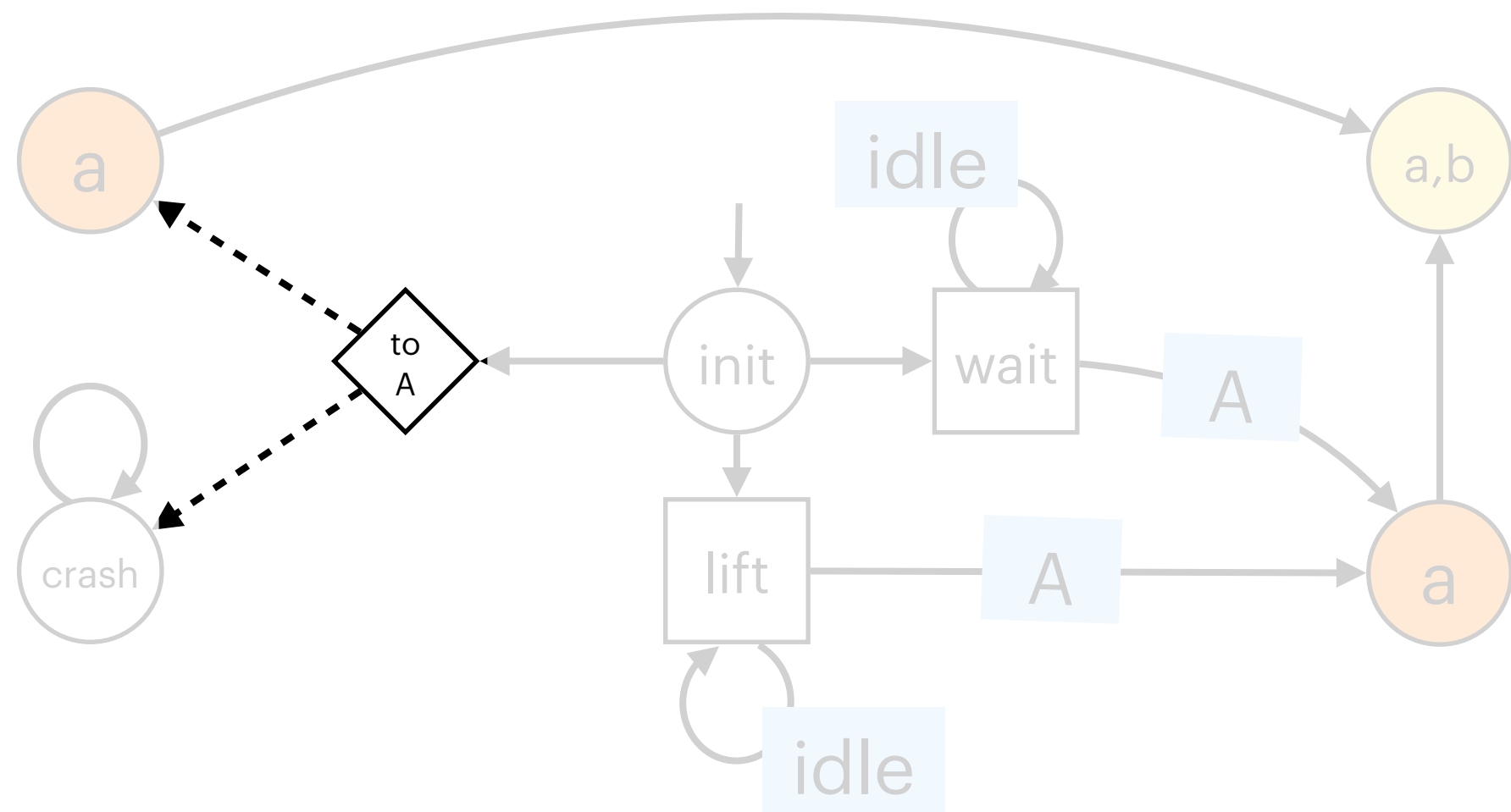
$$\psi = \diamond(A, B)$$



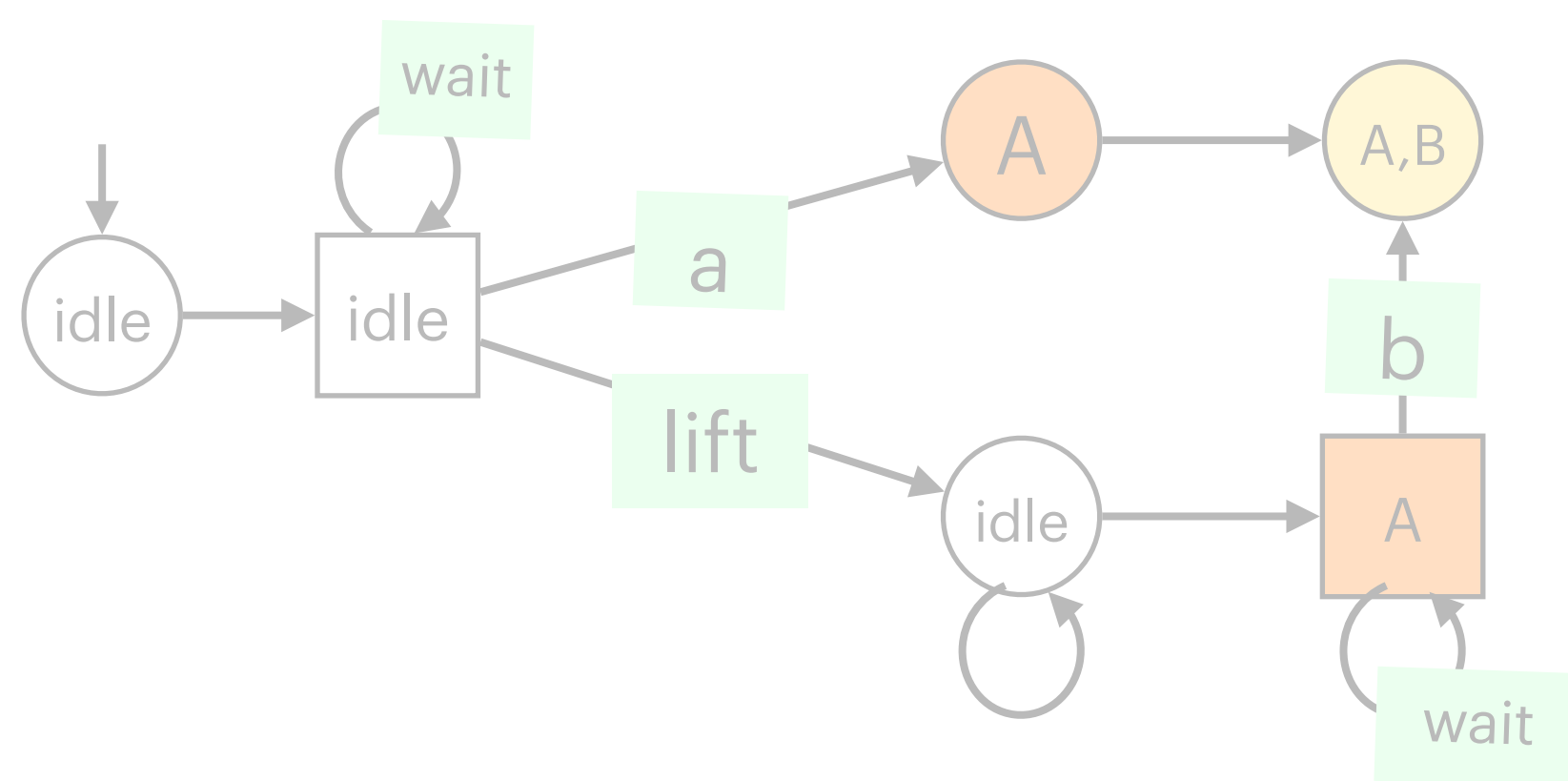
The Problem



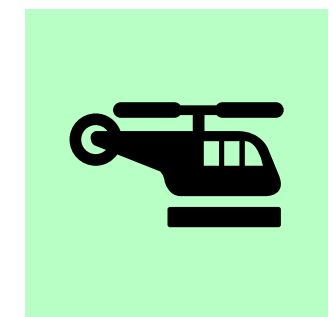
$$\varphi = \diamond(a, b)$$



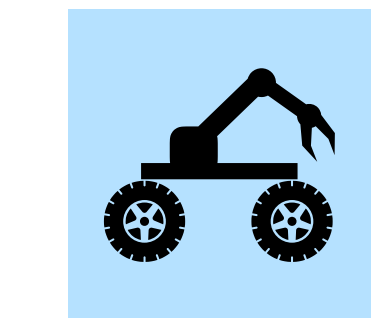
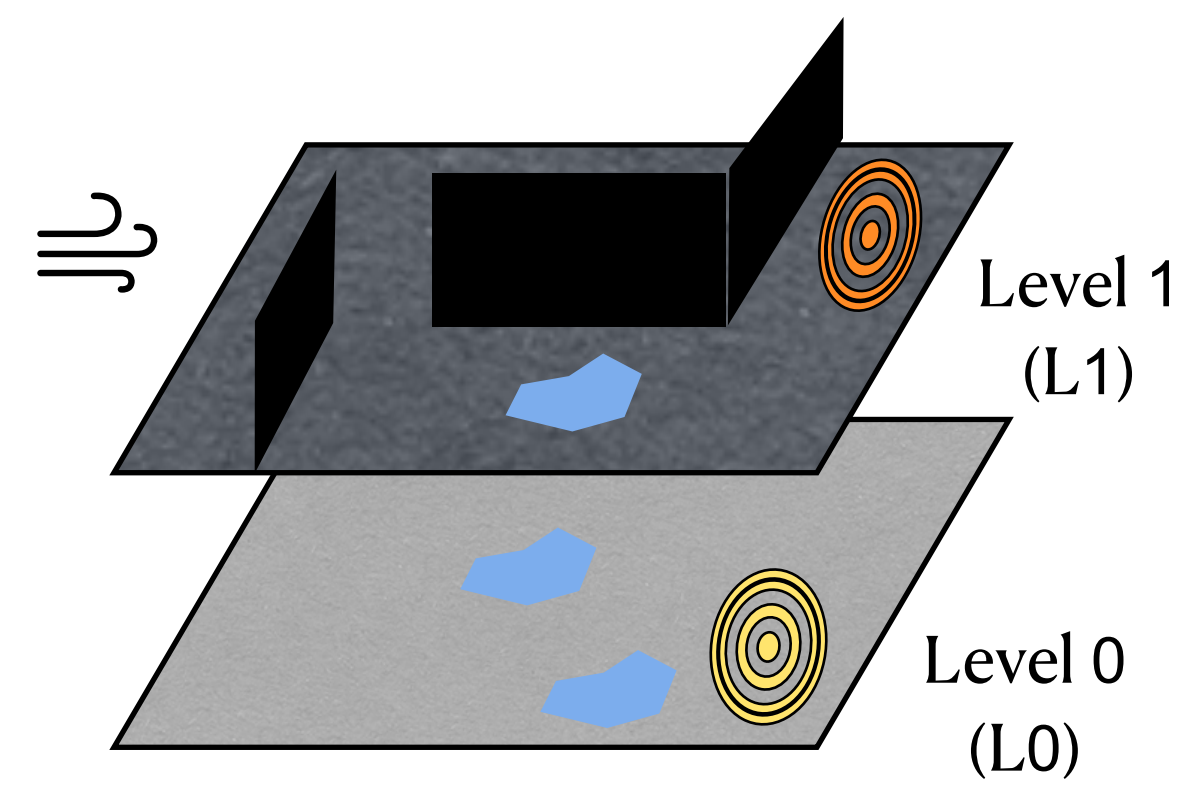
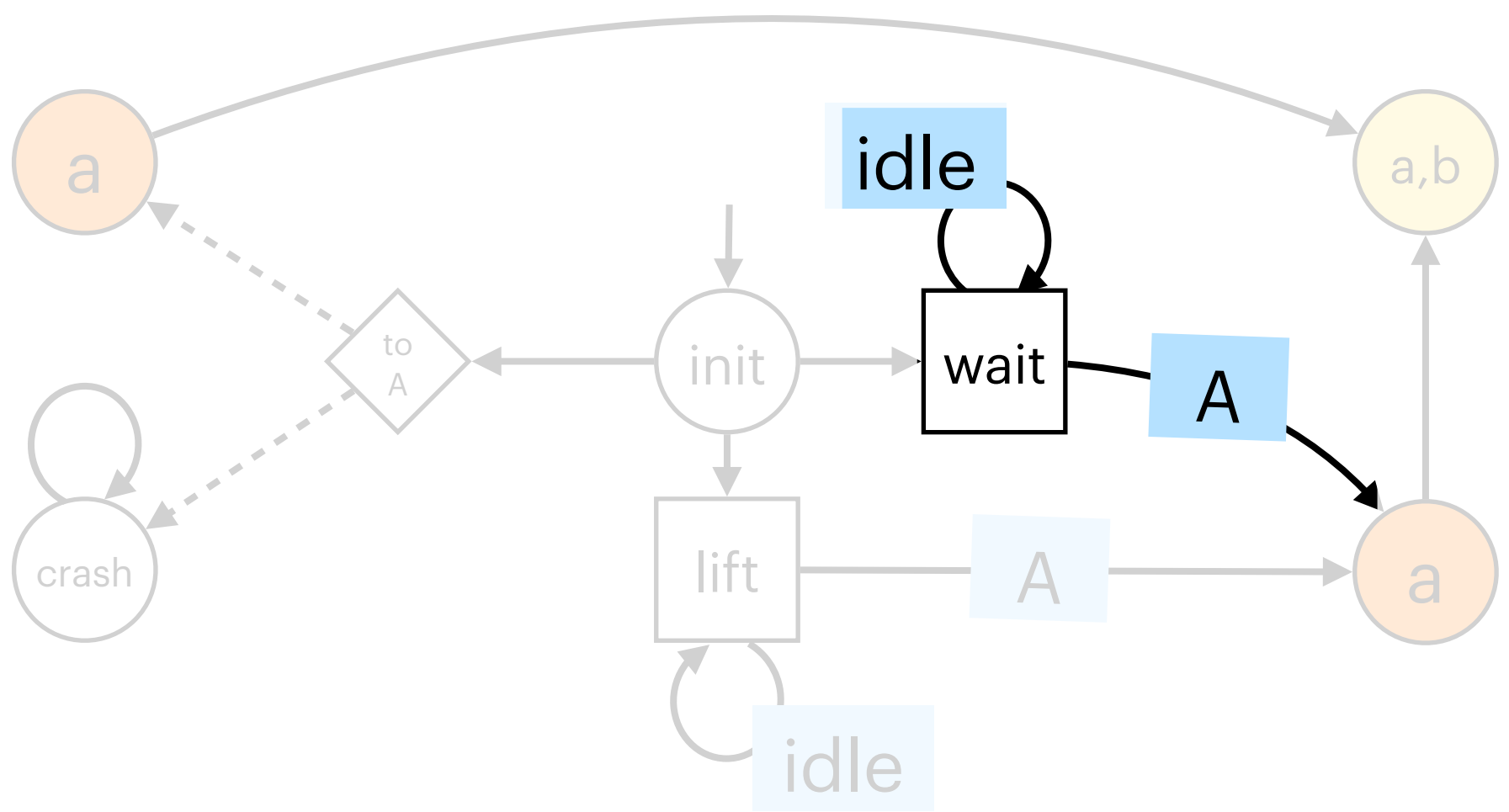
$$\psi = \diamond(A, B)$$



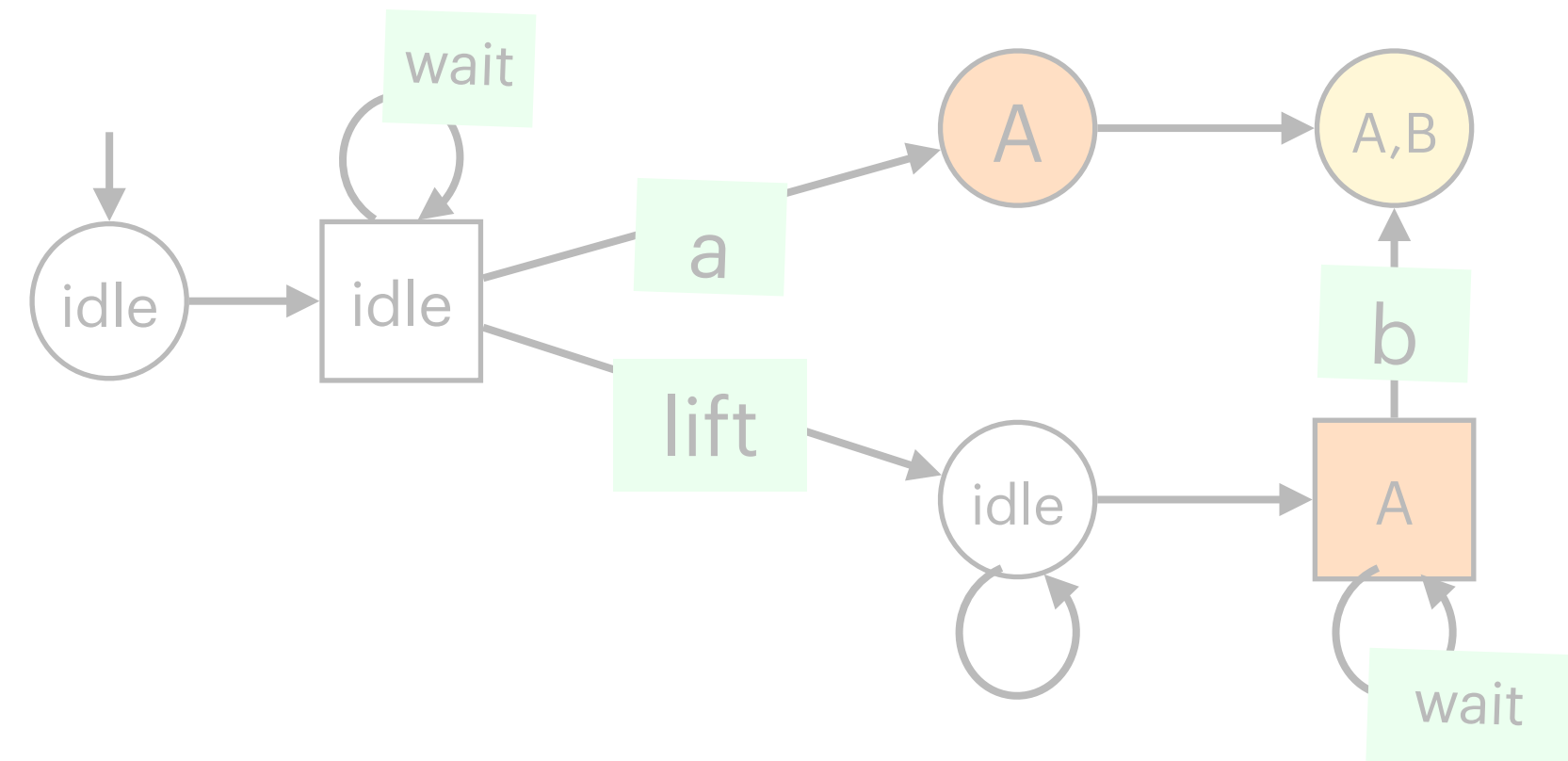
The Problem



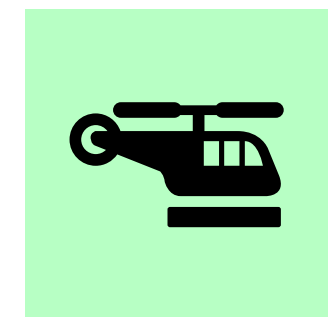
$$\varphi = \diamond(a, b)$$



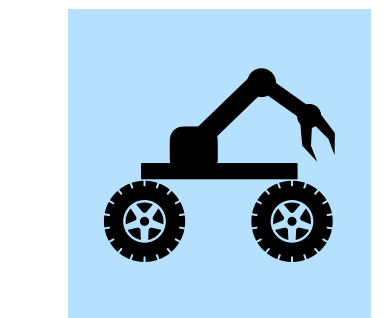
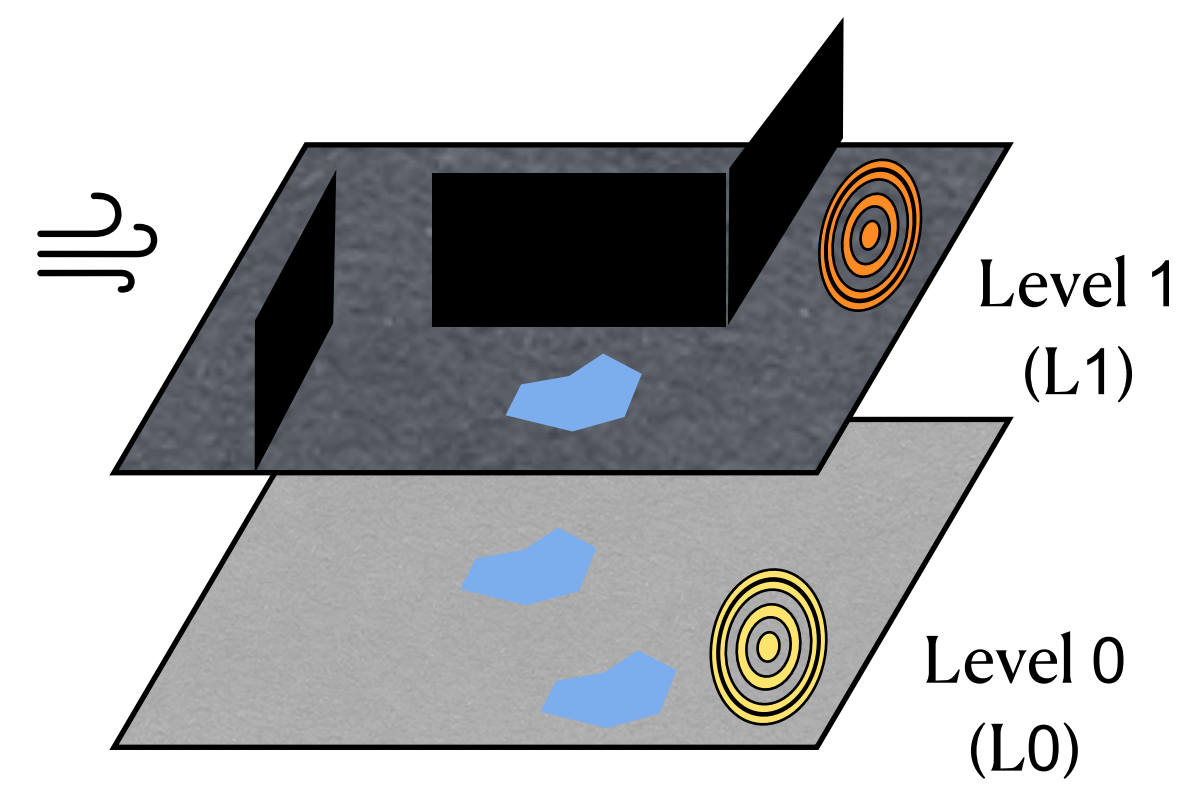
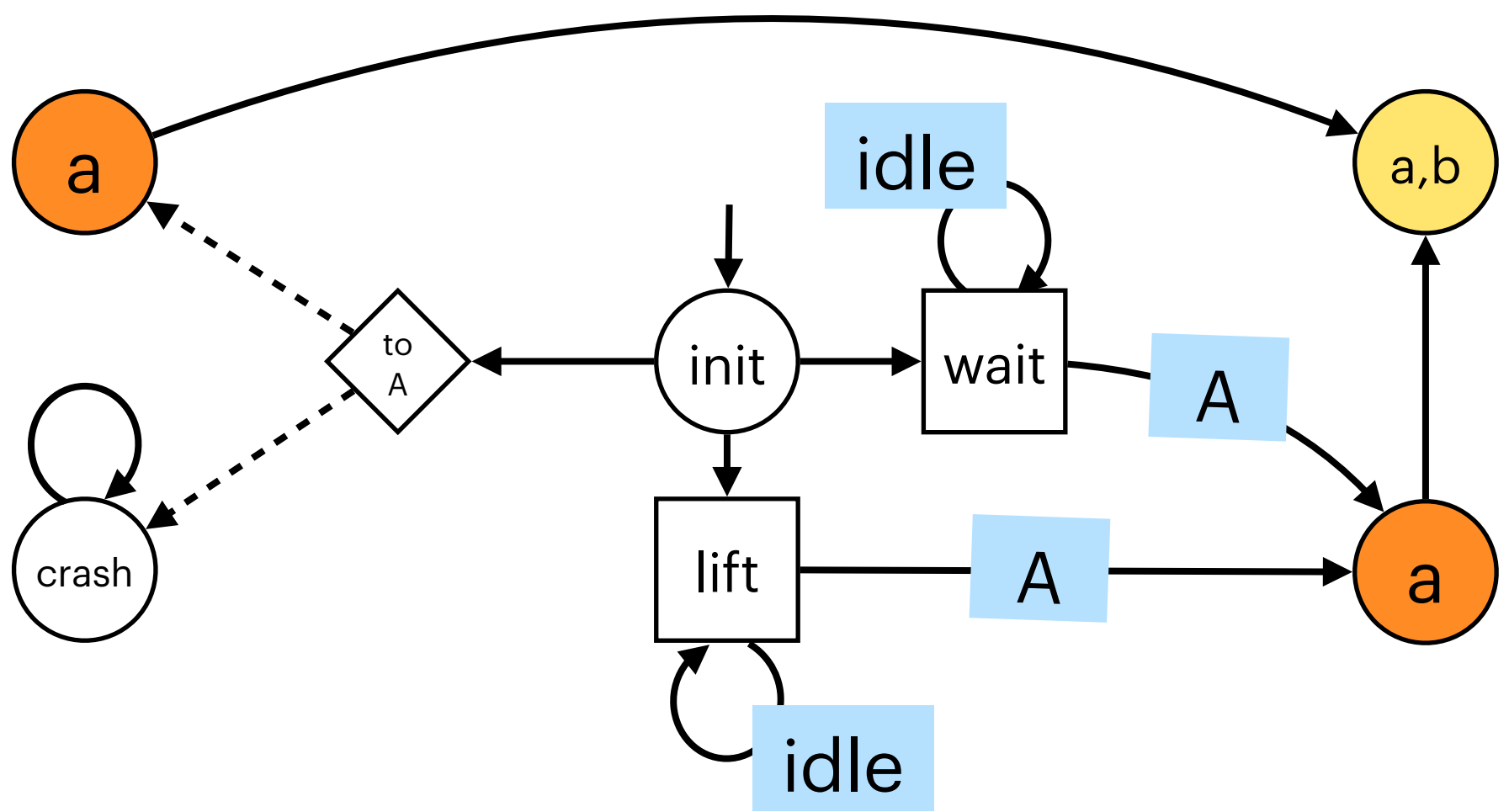
$$\psi = \diamond(A, B)$$



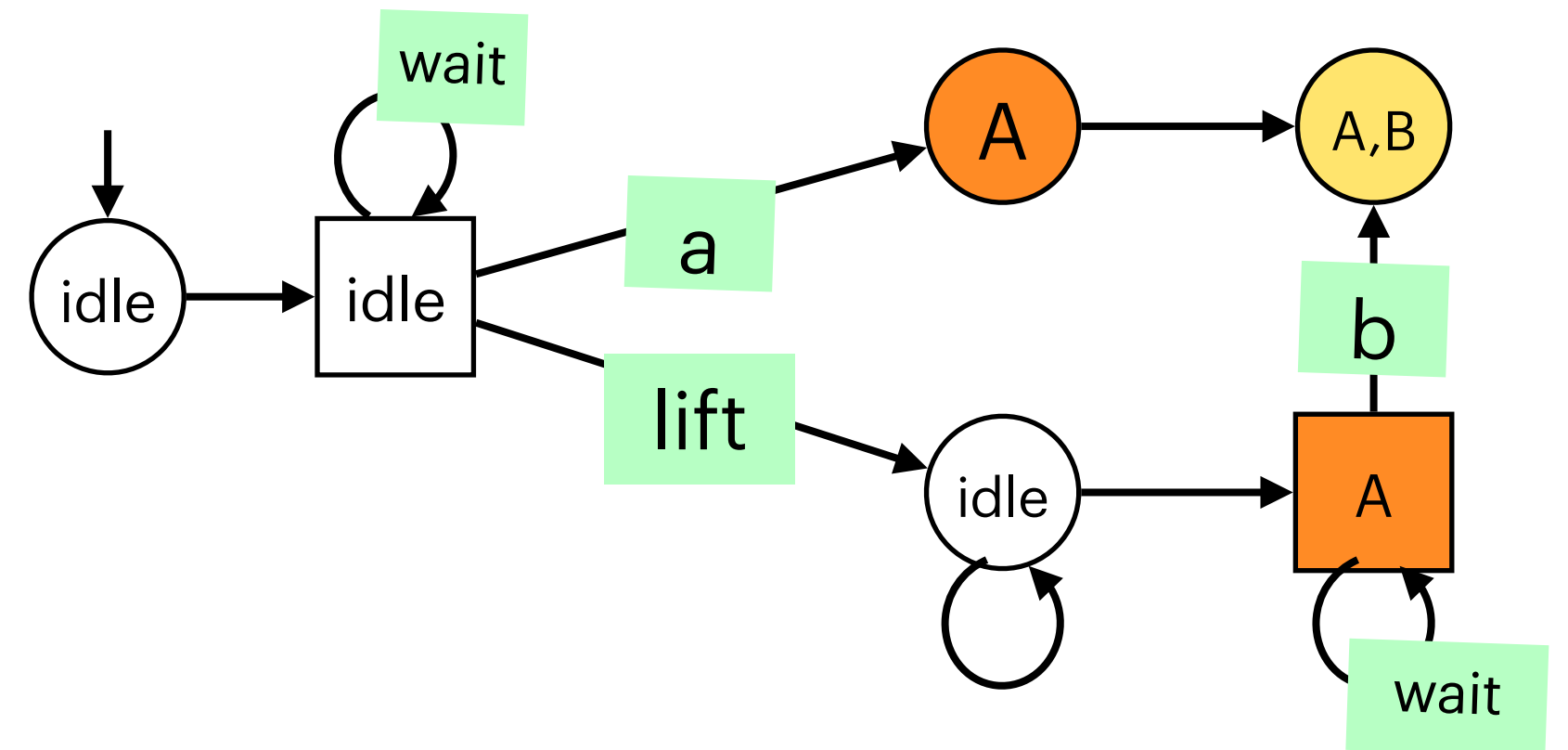
The Problem



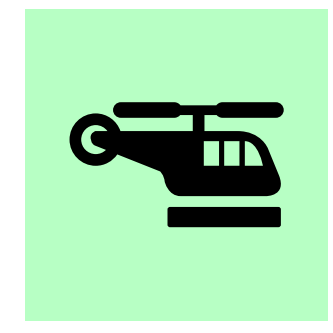
$$\varphi = \diamond(a, b)$$



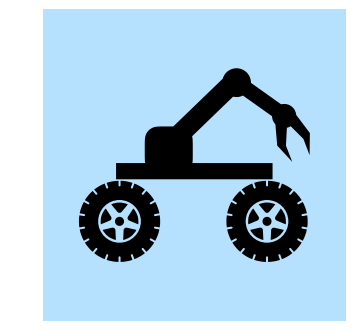
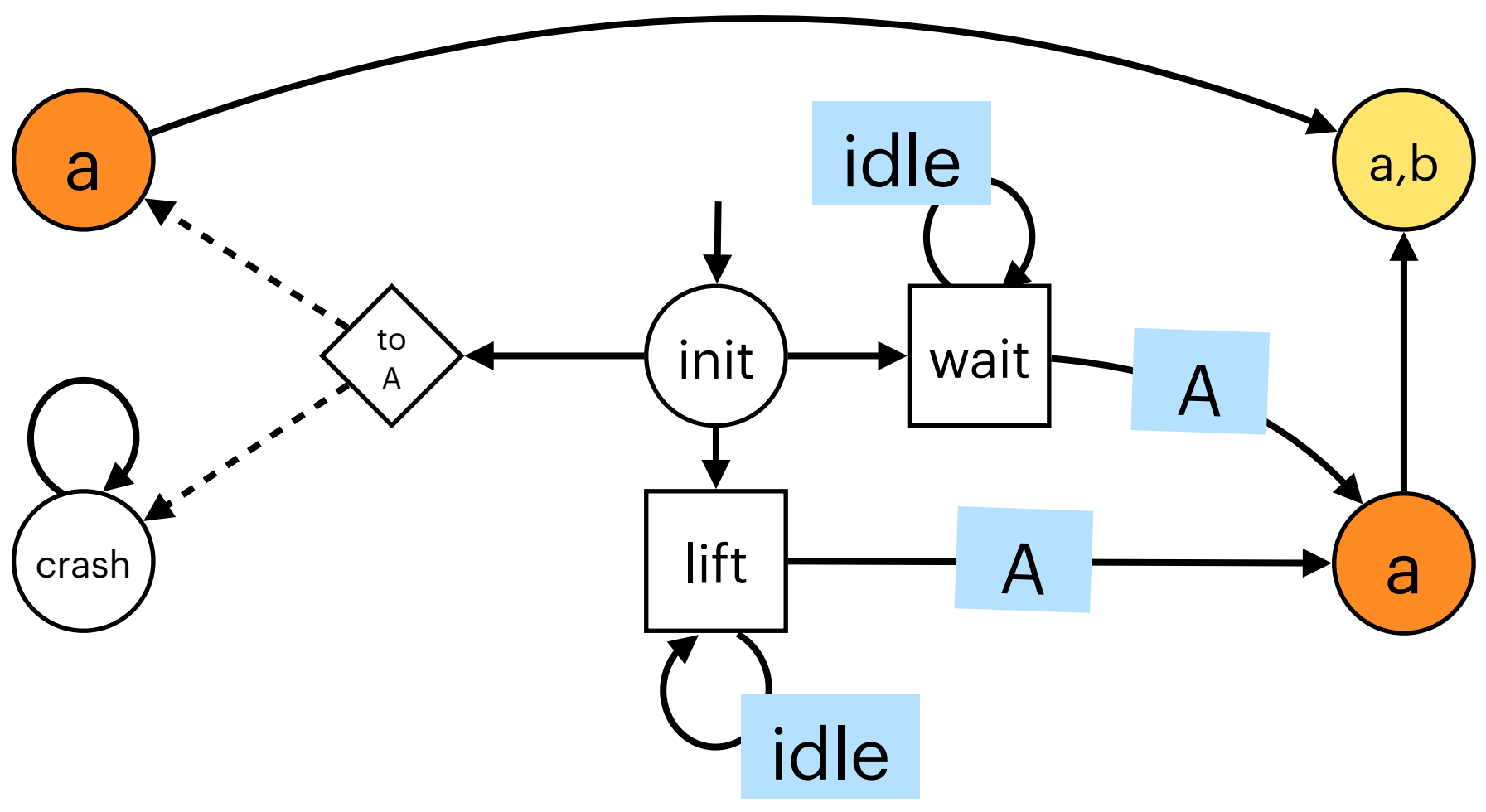
$$\psi = \diamond(A, B)$$



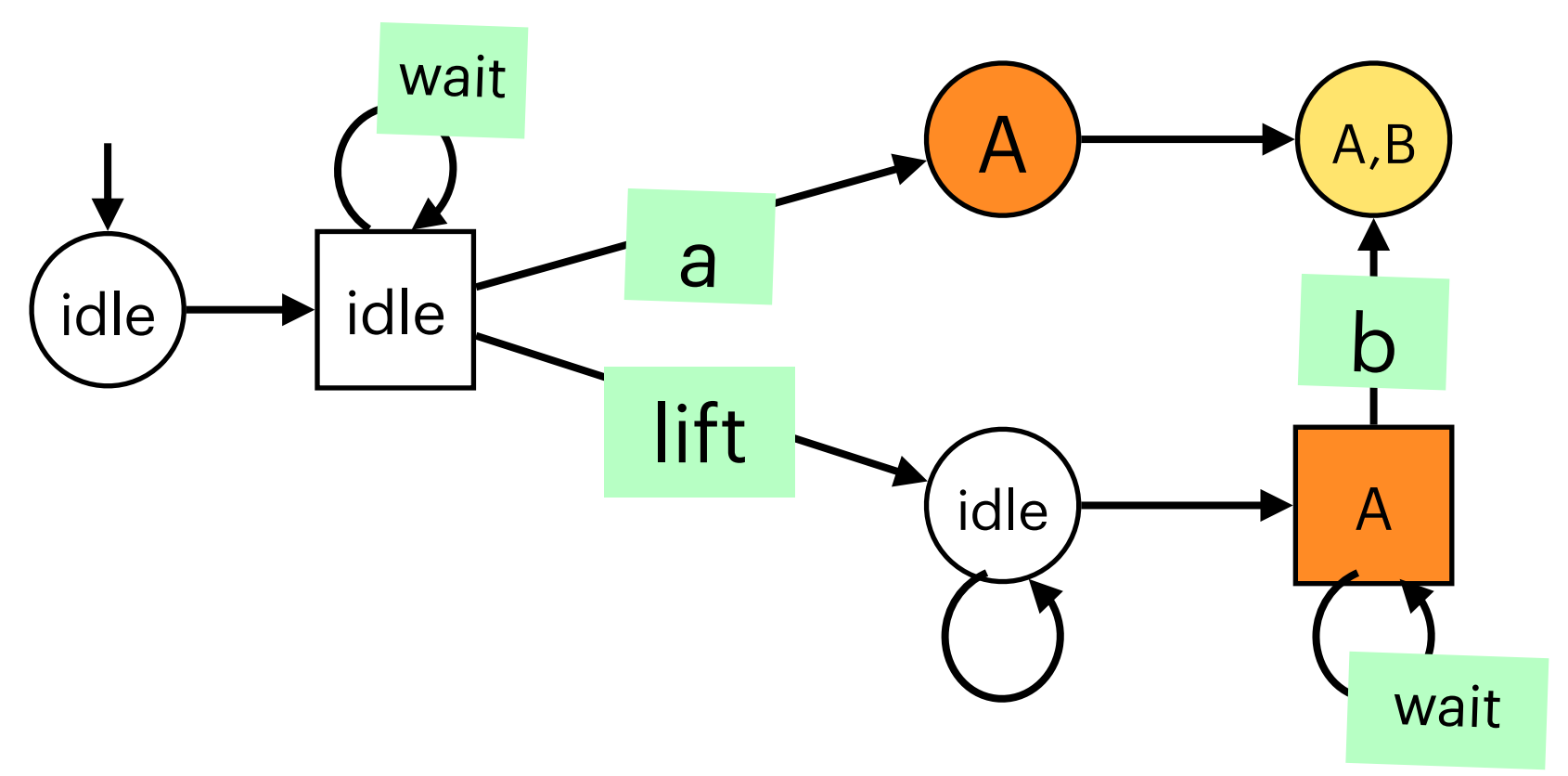
The Problem



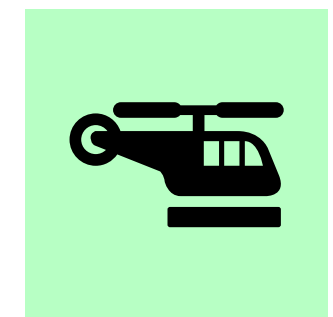
$$\varphi = \diamond(a, b)$$



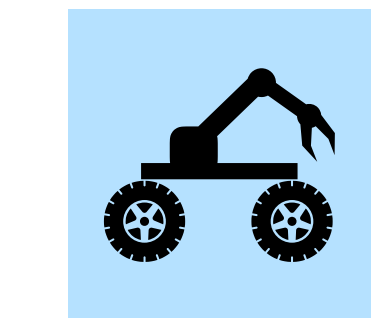
$$\psi = \diamond(A, B)$$



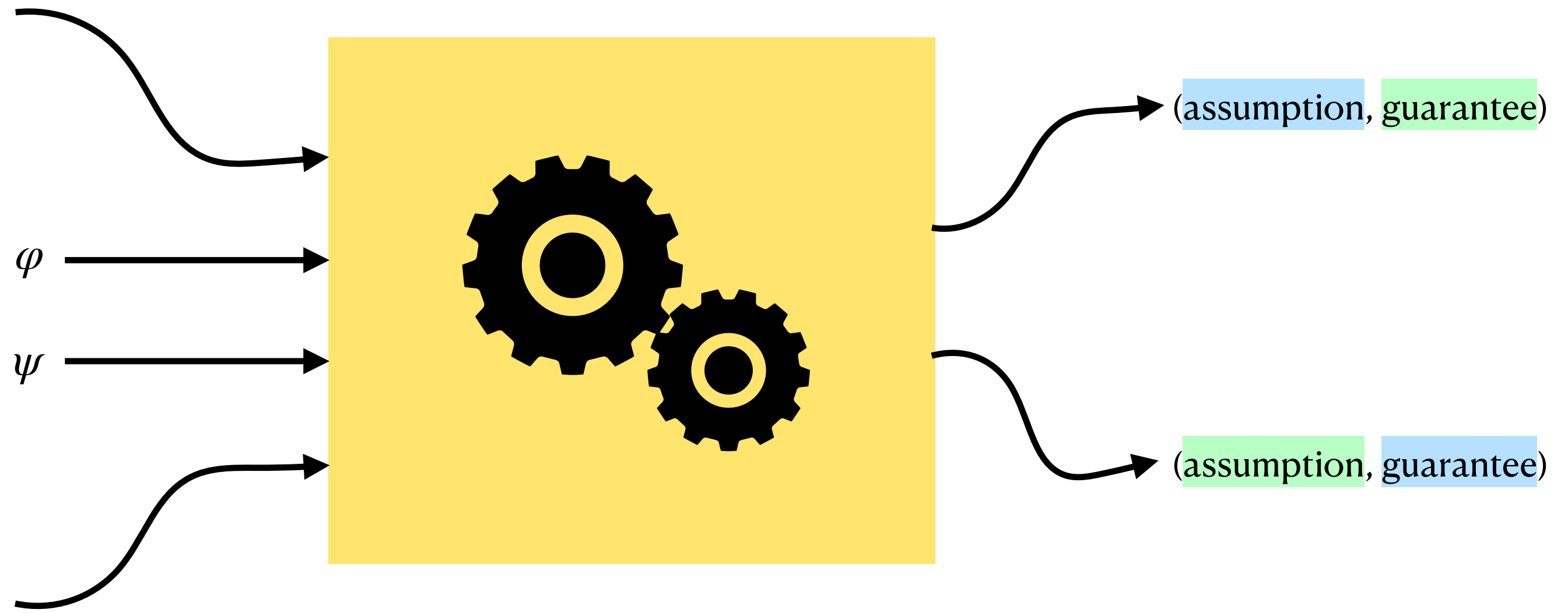
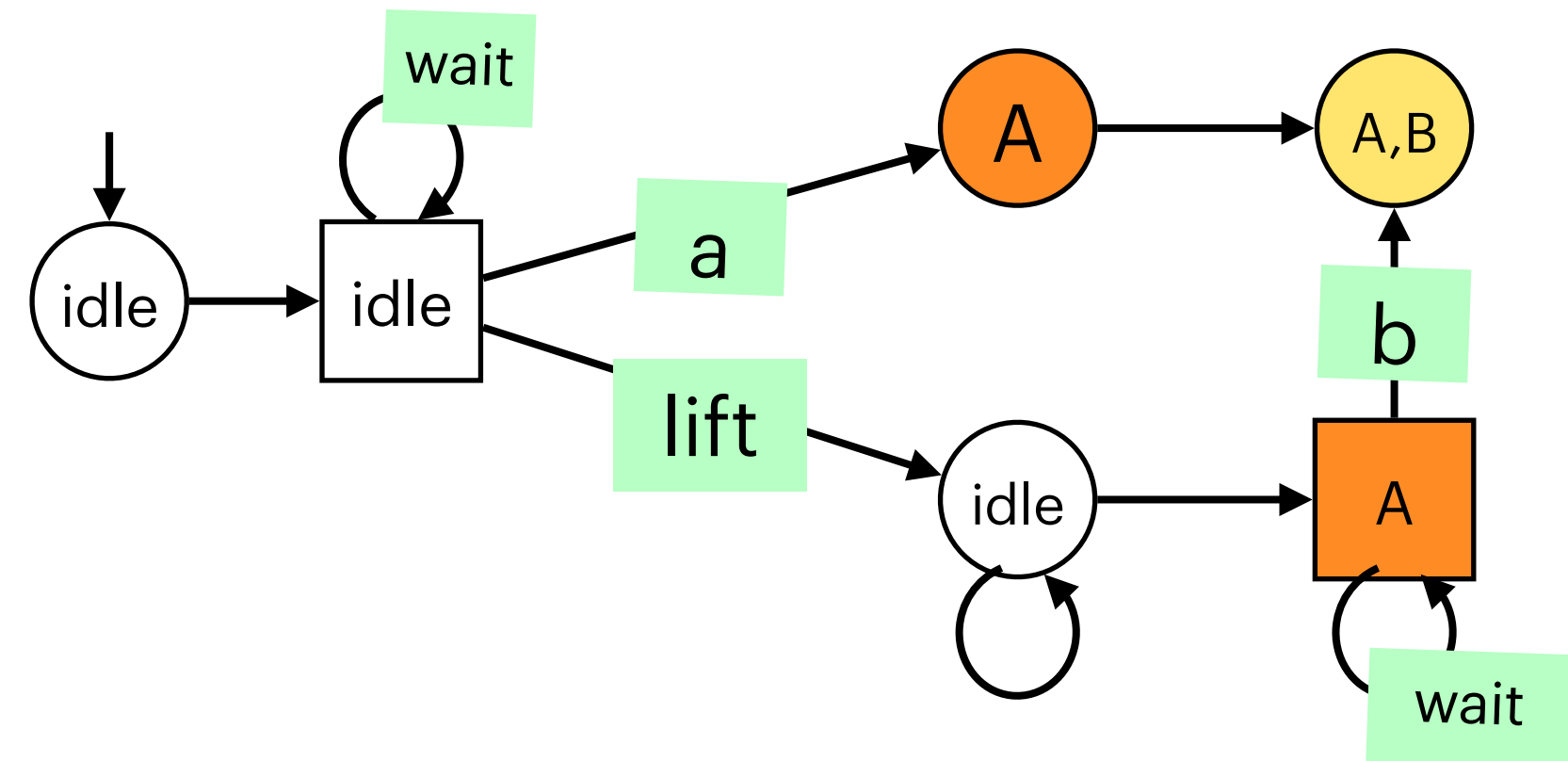
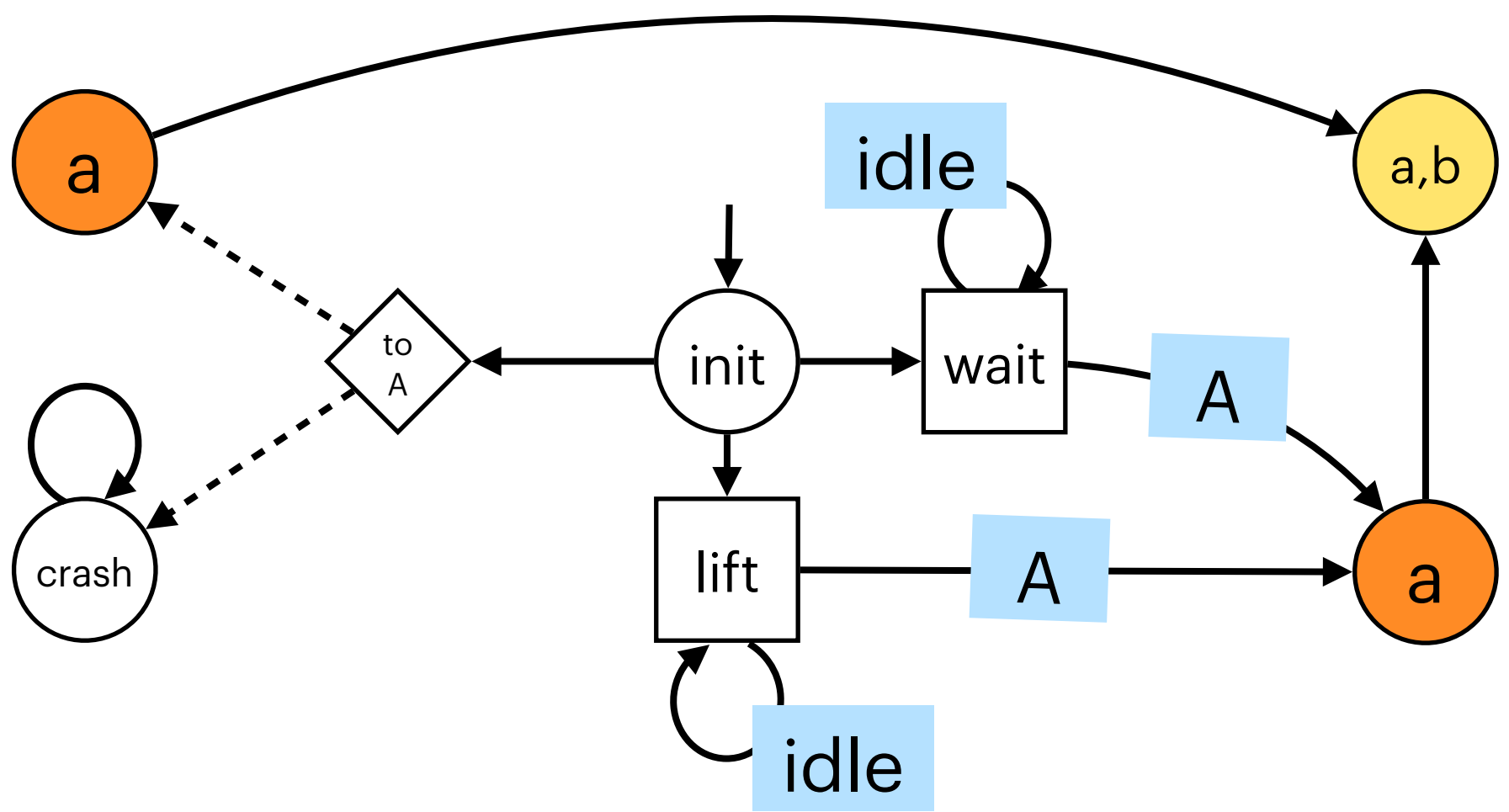
The Problem



$$\varphi = \diamond(a, b)$$



$$\psi = \diamond(A, B)$$



The Problem

The *Negotiation* Algorithm

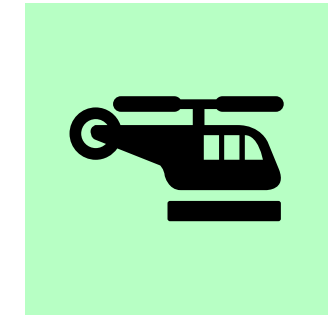
Experiments

The Problem

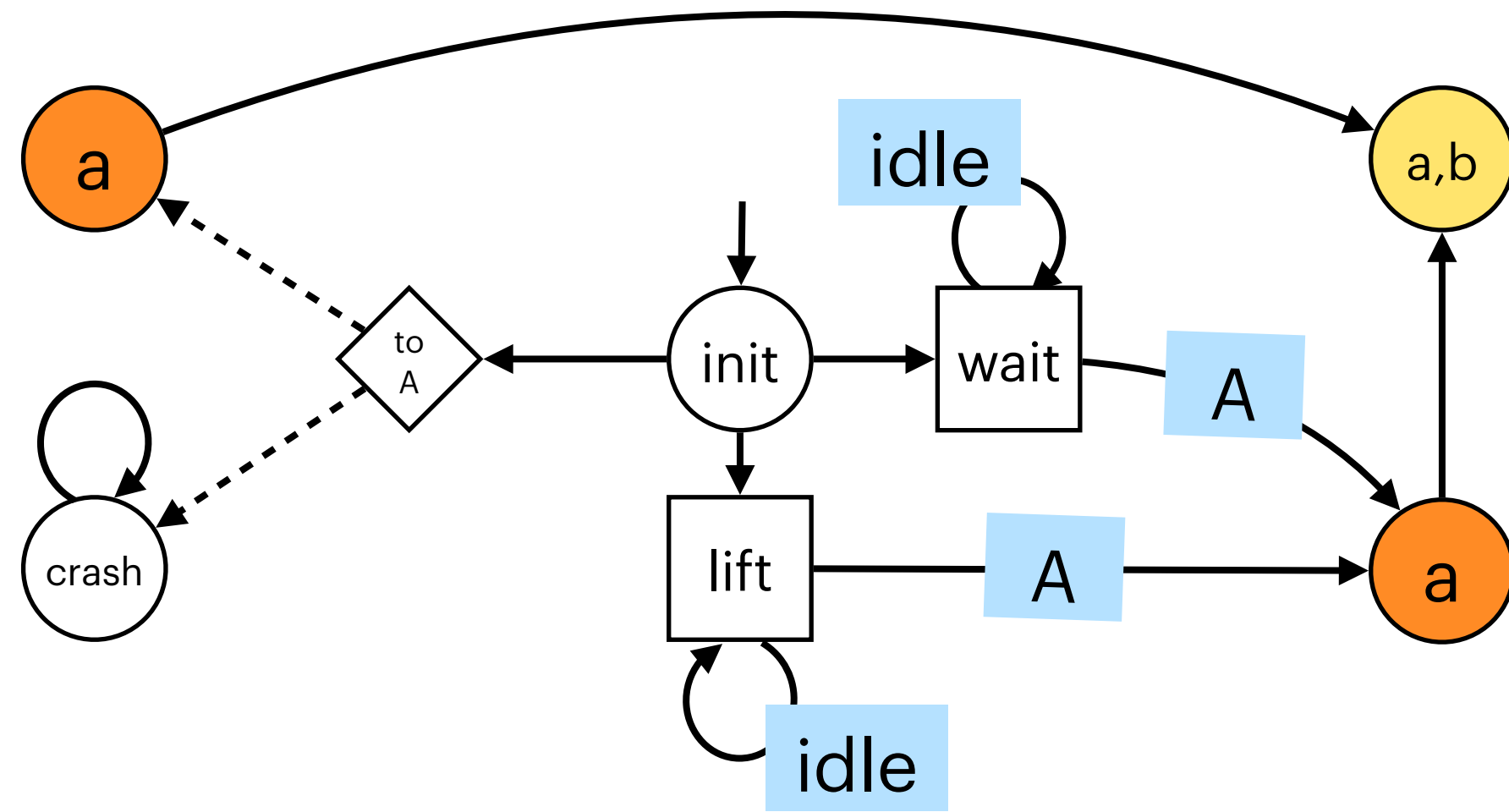
The *Negotiation* Algorithm

Experiments

The *Negotiation* Algorithm



$$\varphi = \diamond(a, b)$$

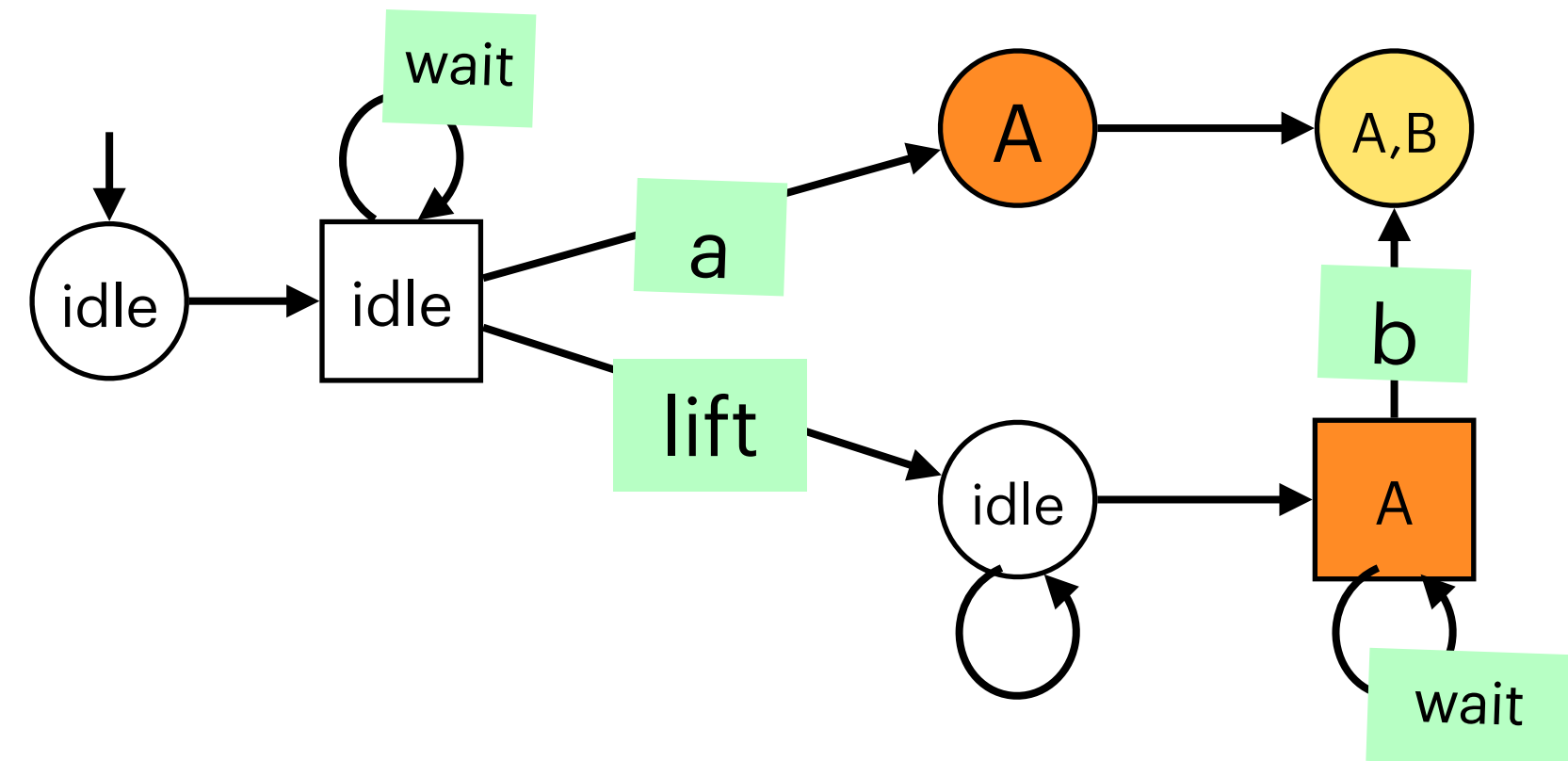


Negotiation rounds

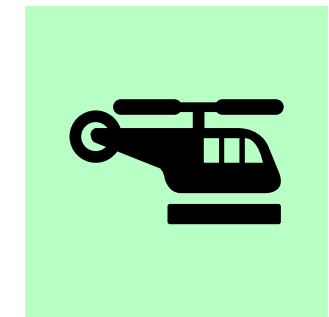
$$\varphi = \diamond(a, b) \wedge (\top, \top)$$



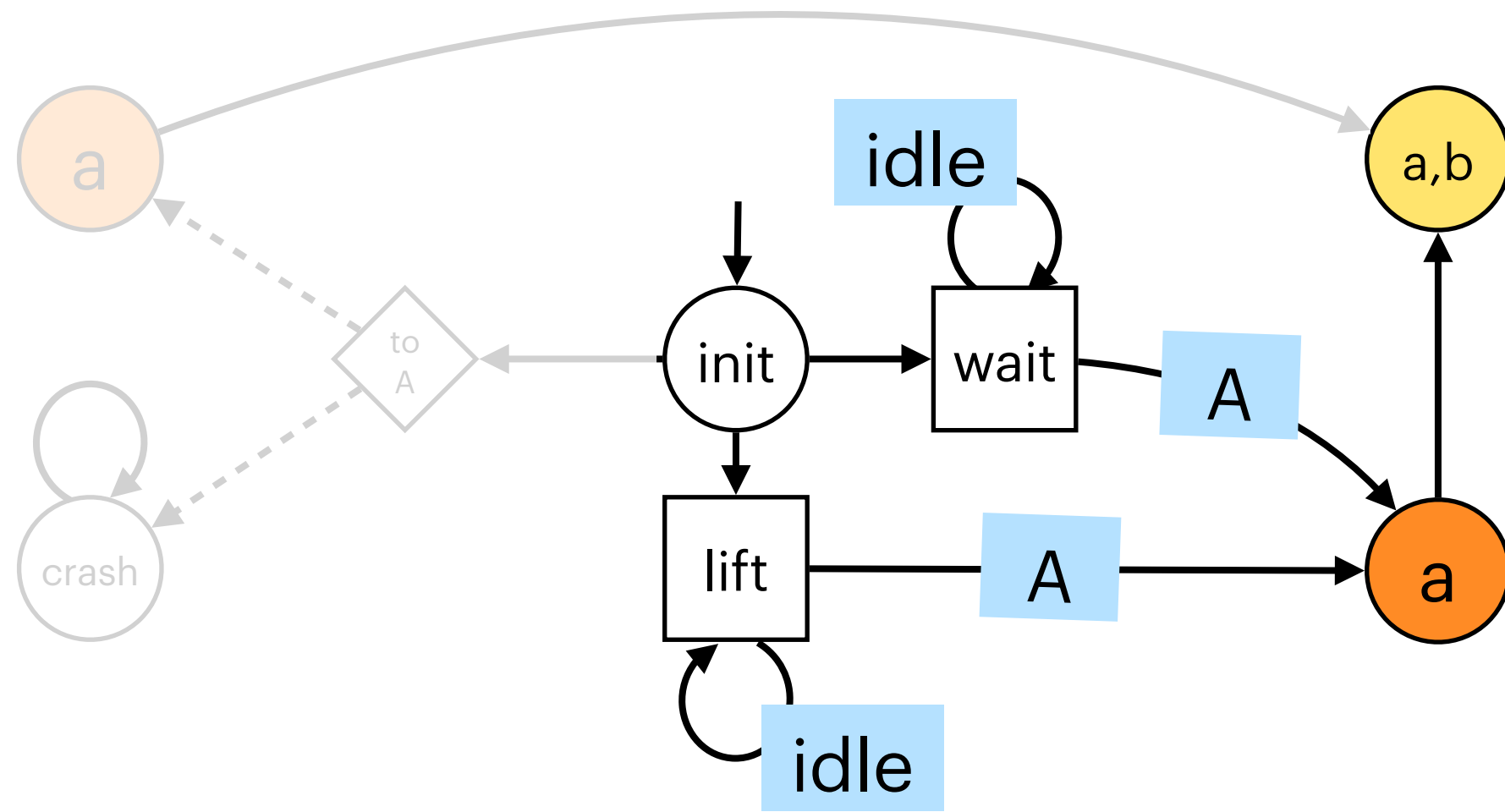
$$\psi = \diamond(A, B)$$



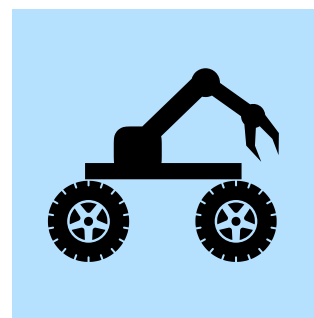
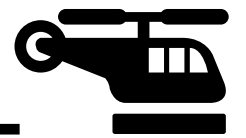
The *Negotiation* Algorithm



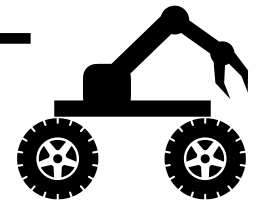
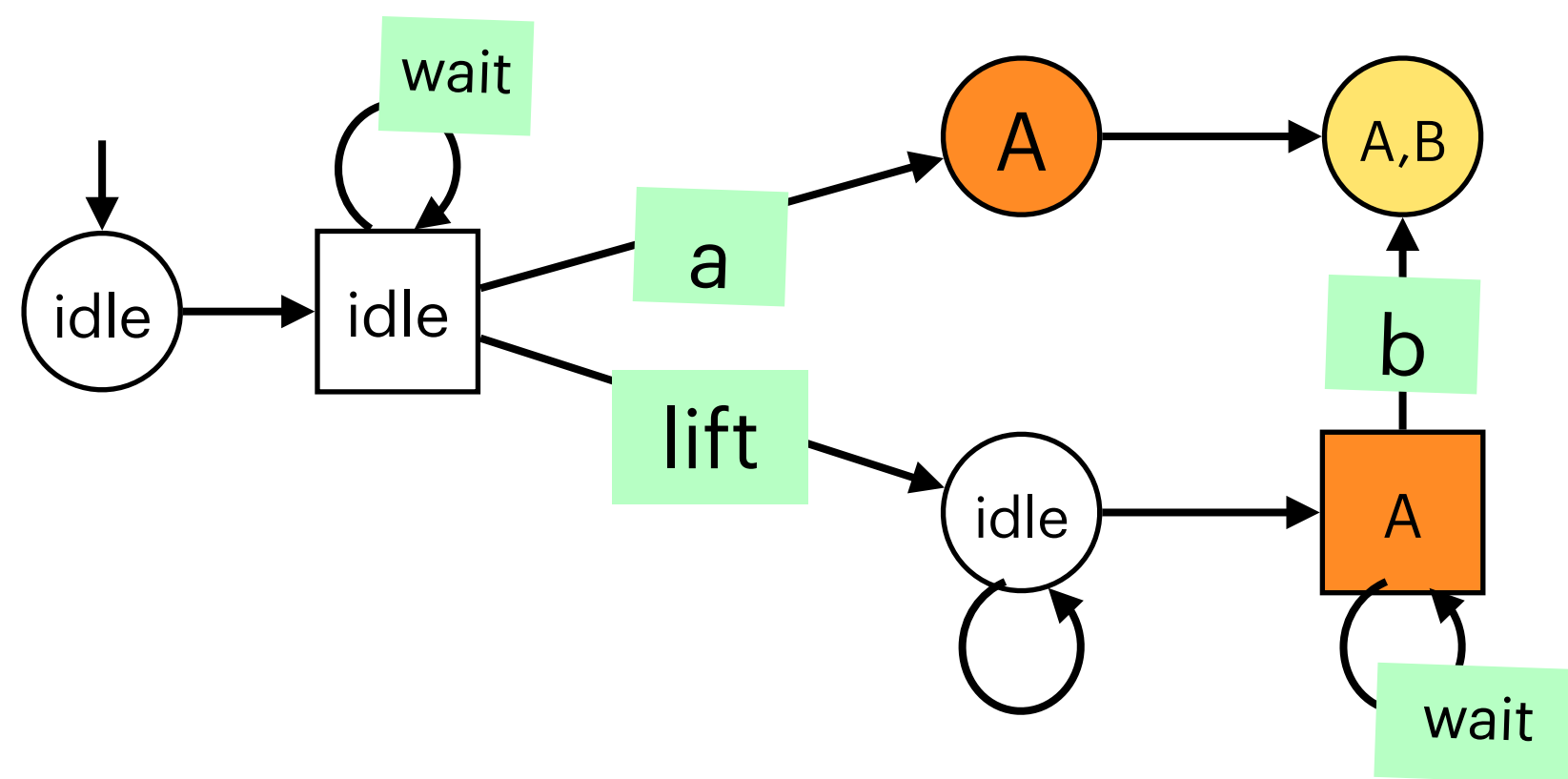
$$\varphi = \diamond(a, b)$$



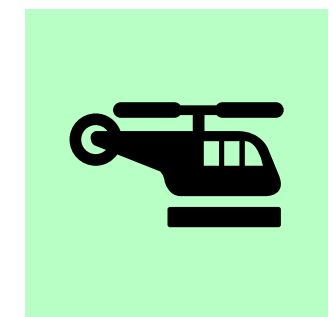
Negotiation rounds
 $\varphi = \diamond(a, b) \wedge (\top, \top)$



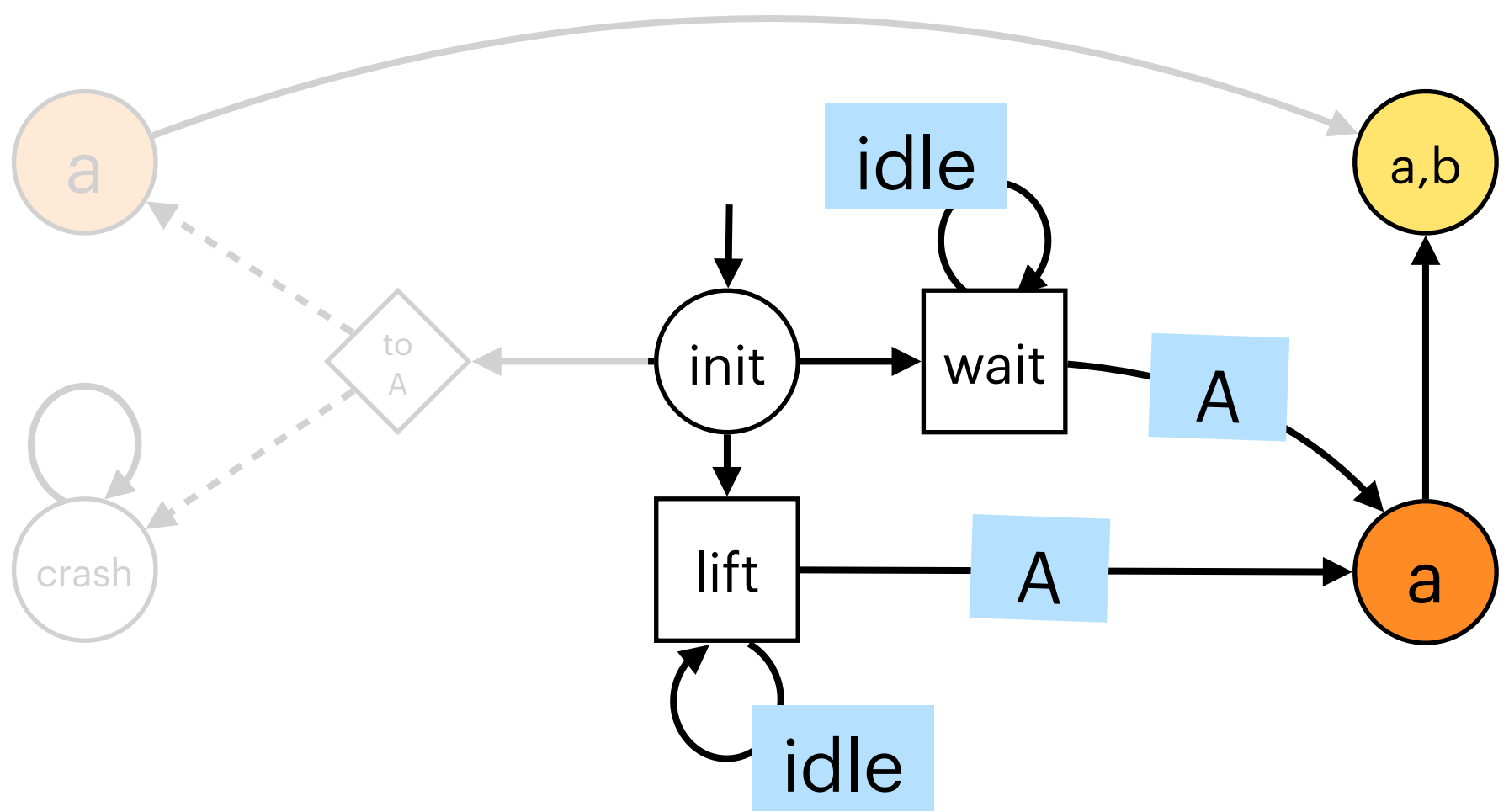
$$\psi = \diamond(A, B)$$



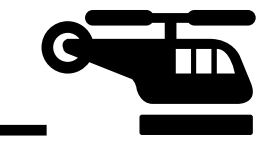
The *Negotiation* Algorithm



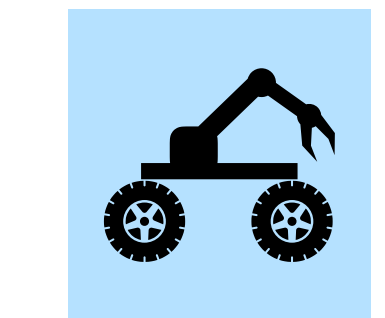
$$\varphi = \diamond(a, b)$$



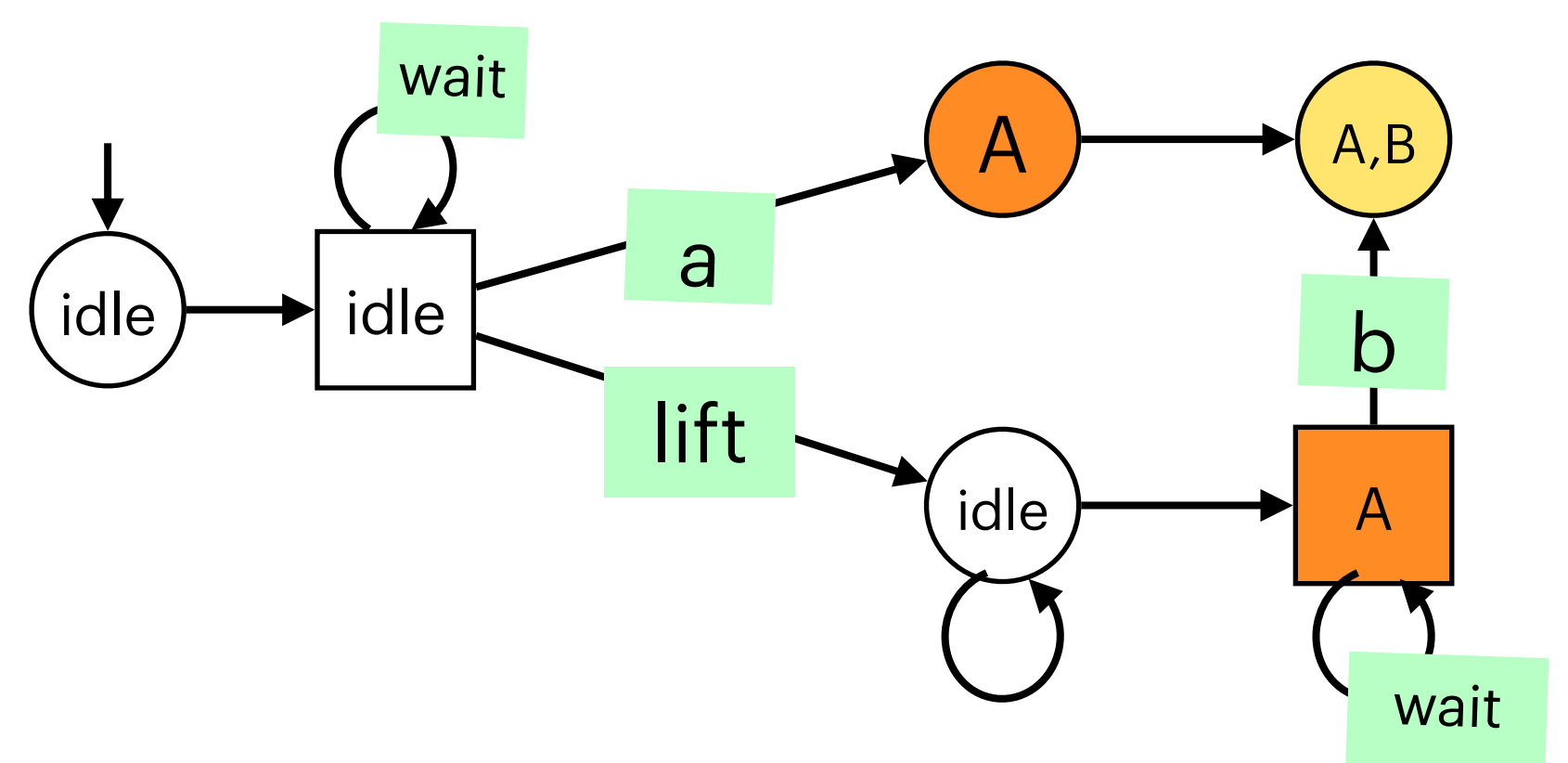
Negotiation rounds
 $\varphi = \diamond(a, b) \wedge (\top, \top)$



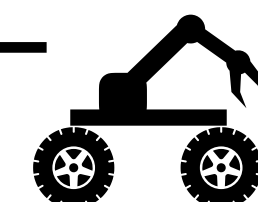
Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$



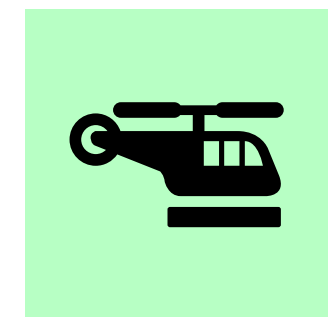
$$\psi = \diamond(A, B)$$



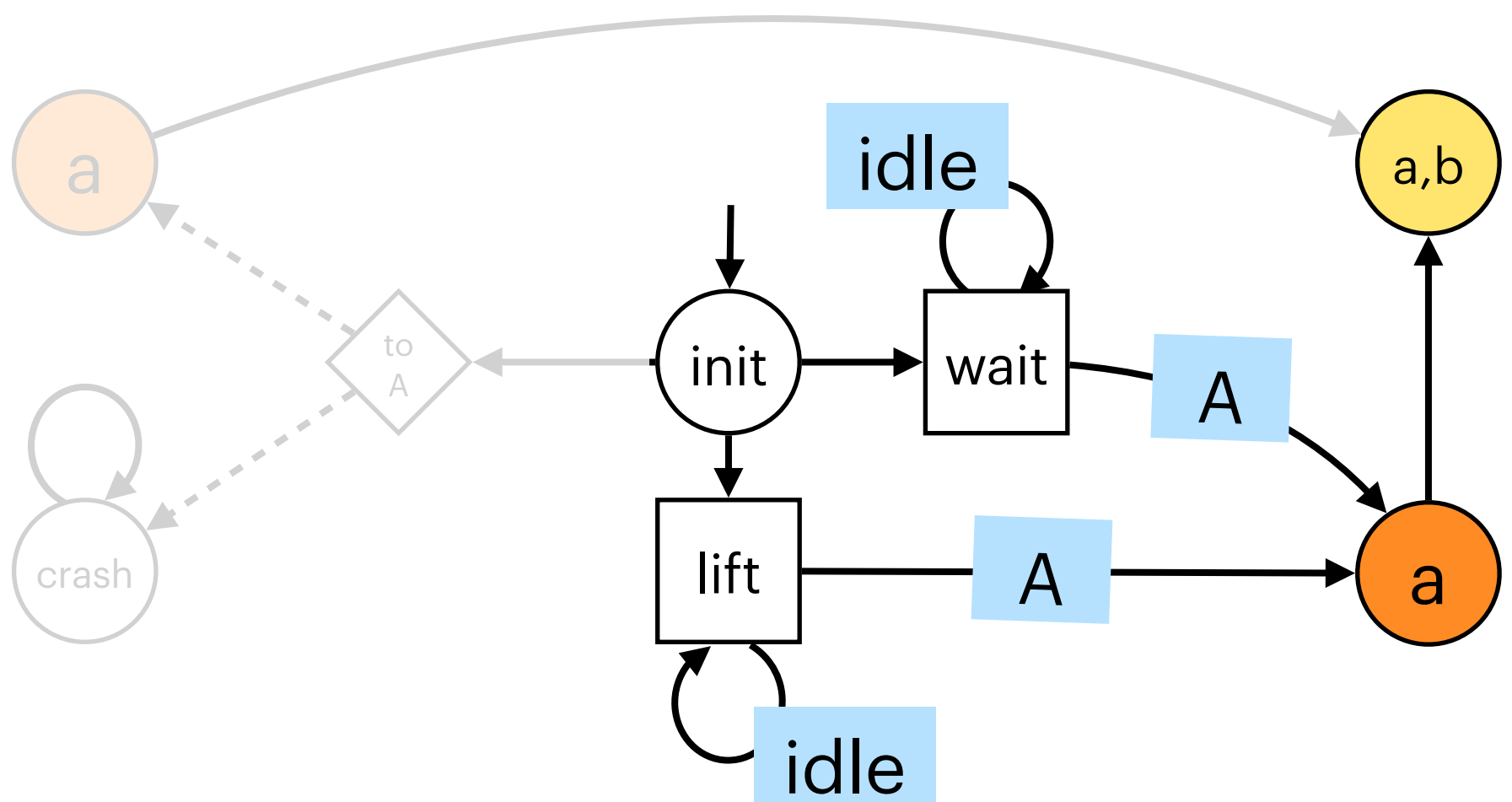
$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle})$



The *Negotiation* Algorithm



$$\varphi = \diamond(a, b)$$

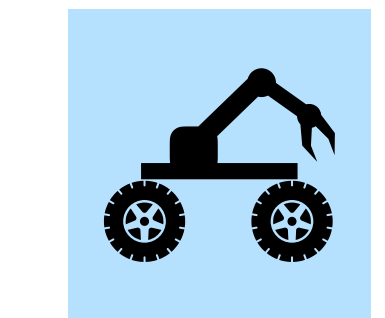


Negotiation rounds \rightarrow

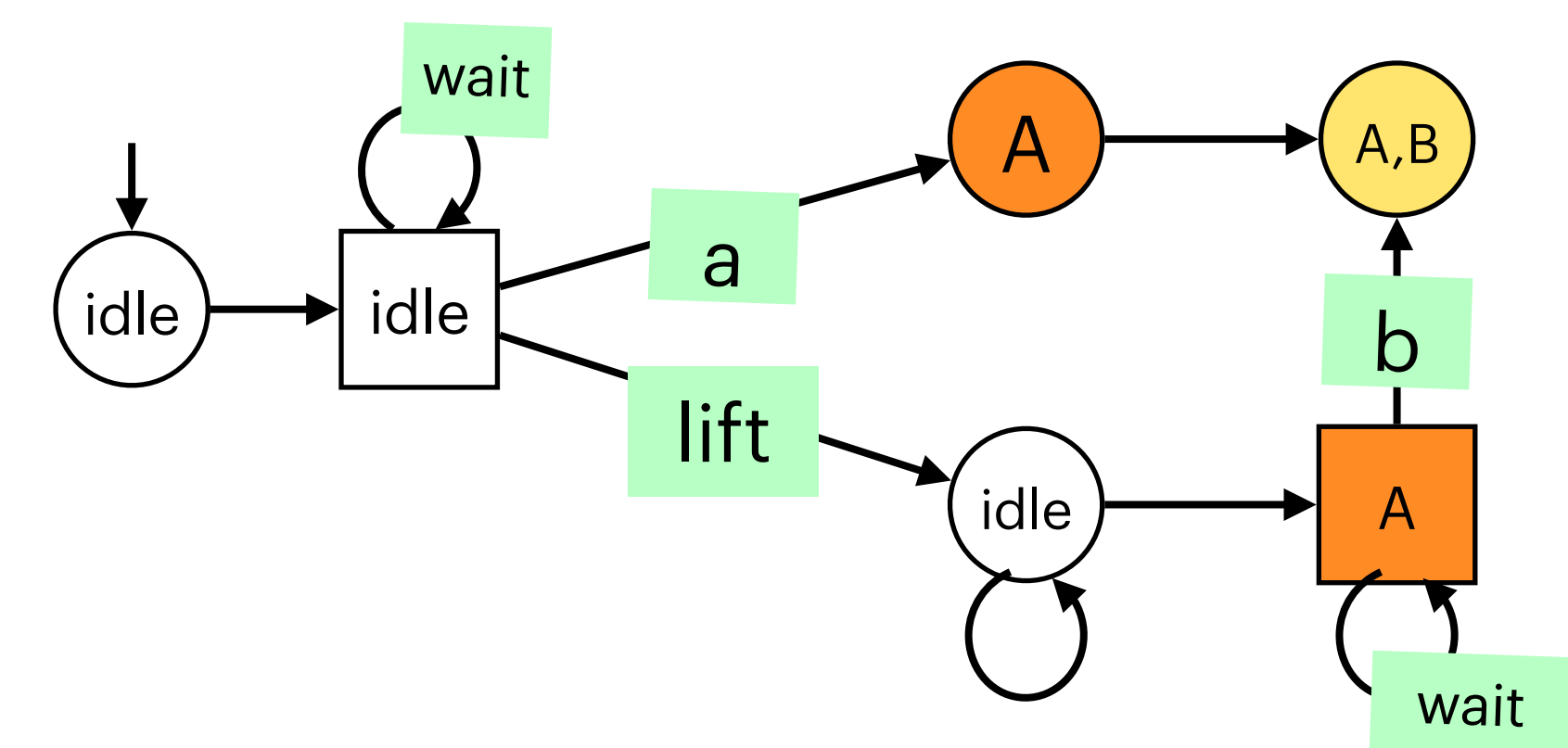
$$\varphi = \diamond(a, b) \wedge (\top, \top)$$



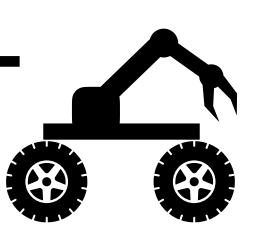
Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$



$$\psi = \diamond(A, B)$$

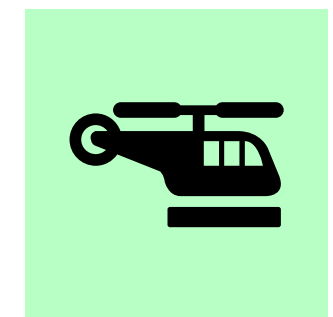


$$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle})$$

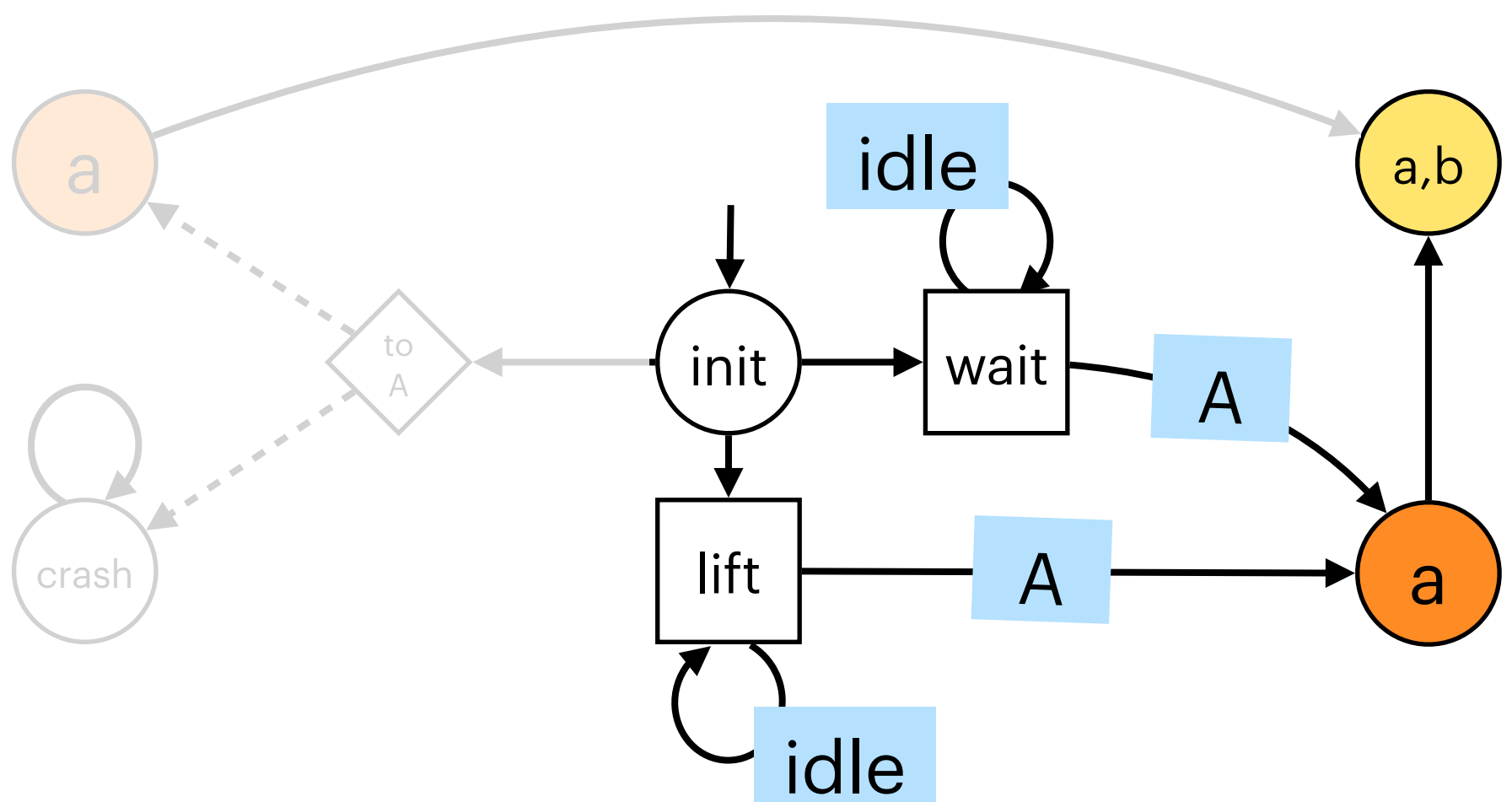


*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

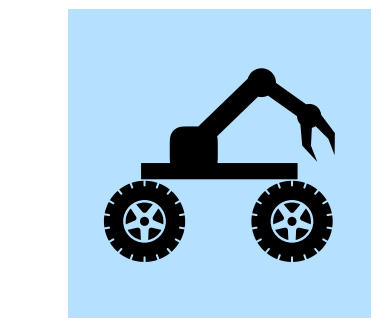
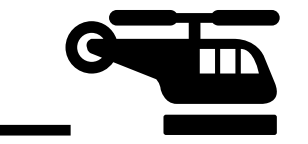
The *Negotiation* Algorithm



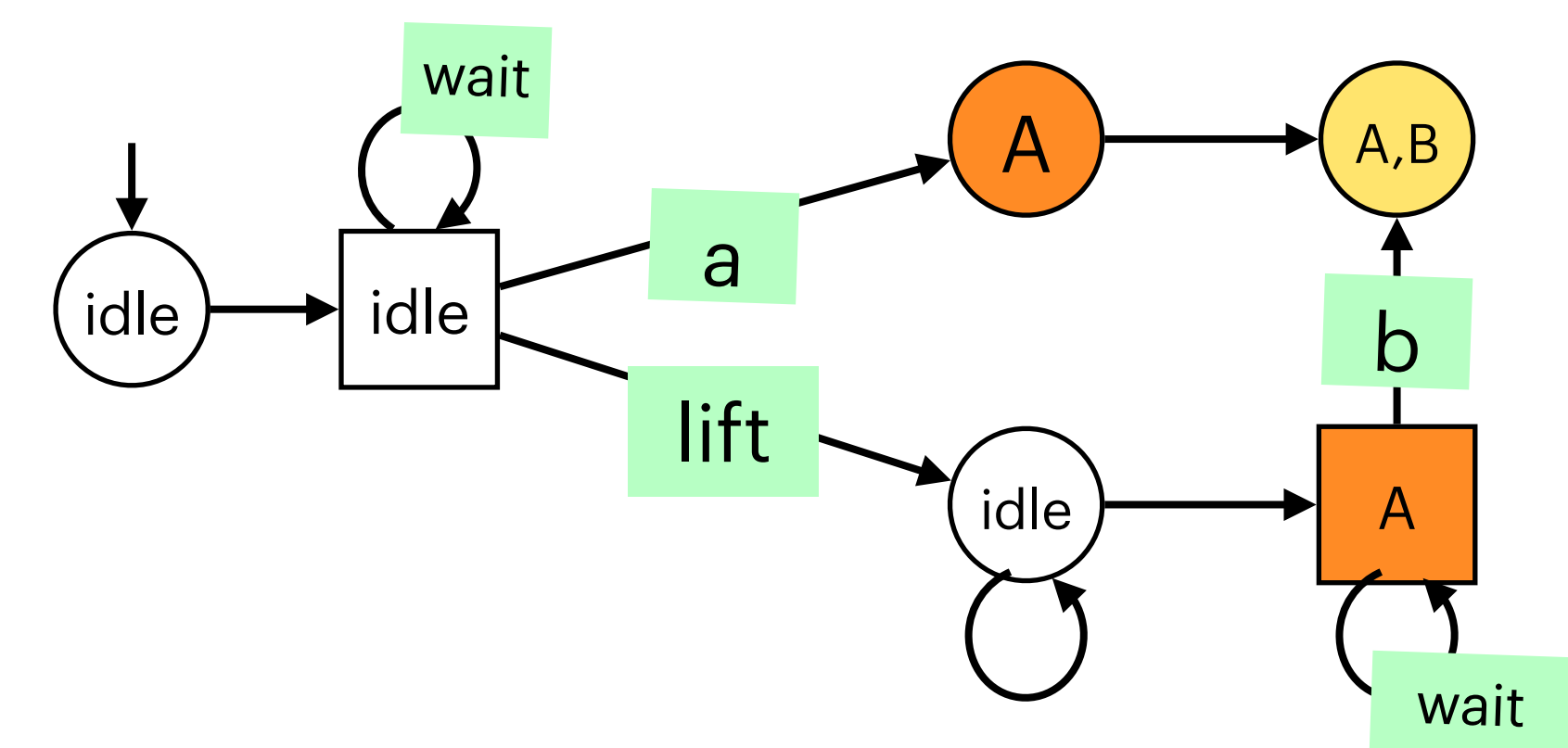
$$\varphi = \diamond(a, b)$$



Negotiation rounds
 $\varphi = \diamond(a, b) \wedge (\top, \top)$



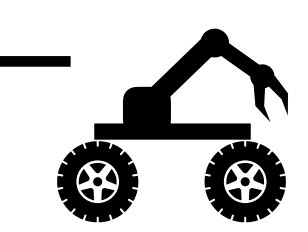
$$\psi = \diamond(A, B)$$



Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$

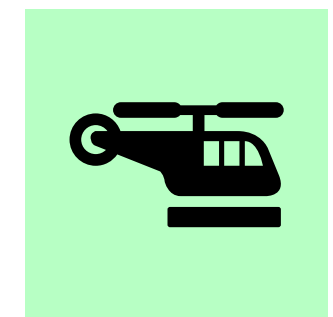
$$\bigcirc^2 \neg \text{wait}$$

$$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle})$$

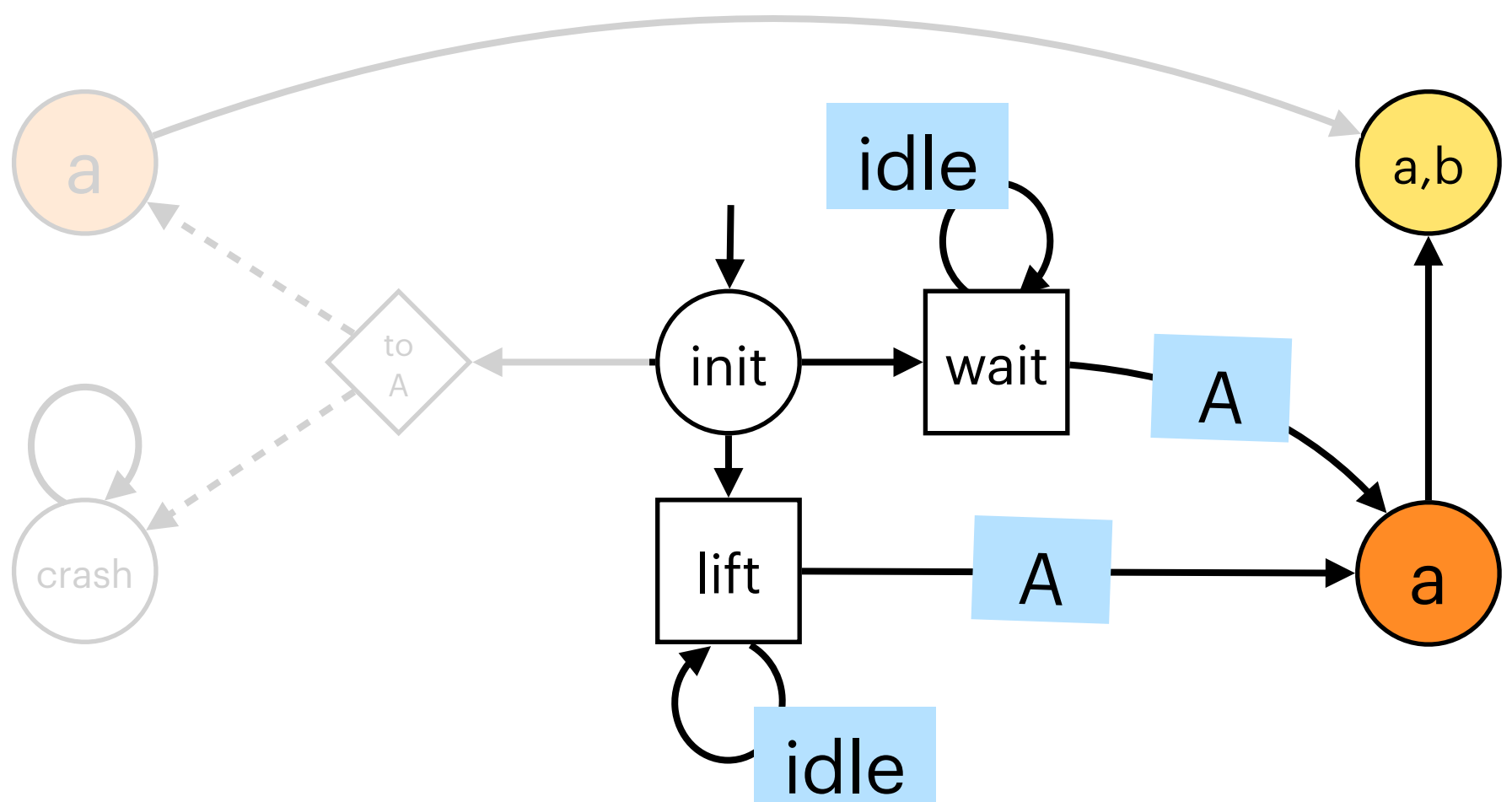


*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

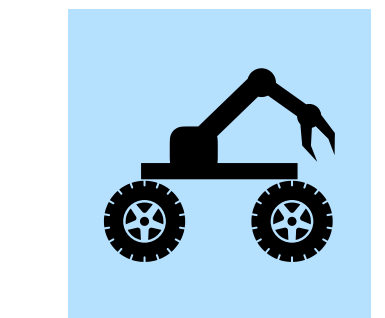
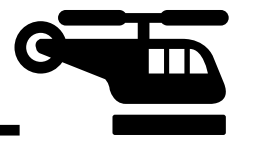
The *Negotiation* Algorithm



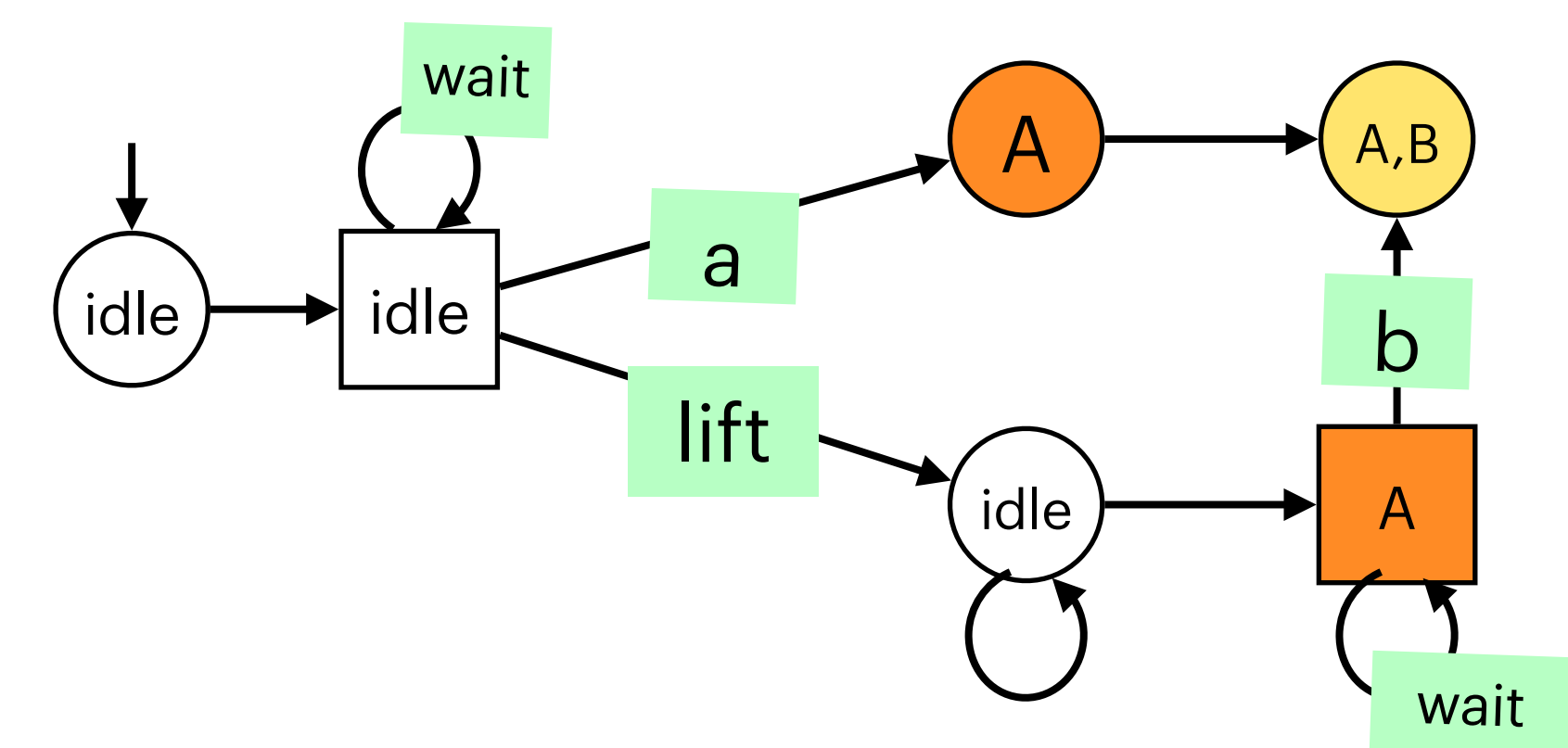
$$\varphi = \diamond(a, b)$$



Negotiation rounds
 $\varphi = \diamond(a, b) \wedge (\top, \top)$



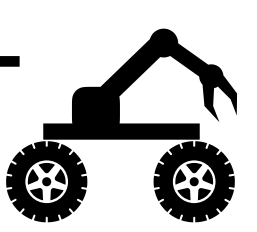
$$\psi = \diamond(A, B)$$



Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$

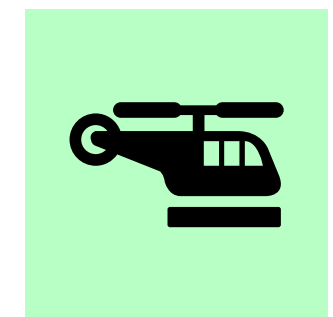
$$\bigcirc^2 \neg \text{wait} \wedge \square (\text{lift} \rightarrow \bigcirc^2 \neg \text{wait})$$

$$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle})$$

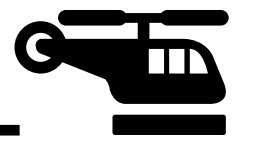
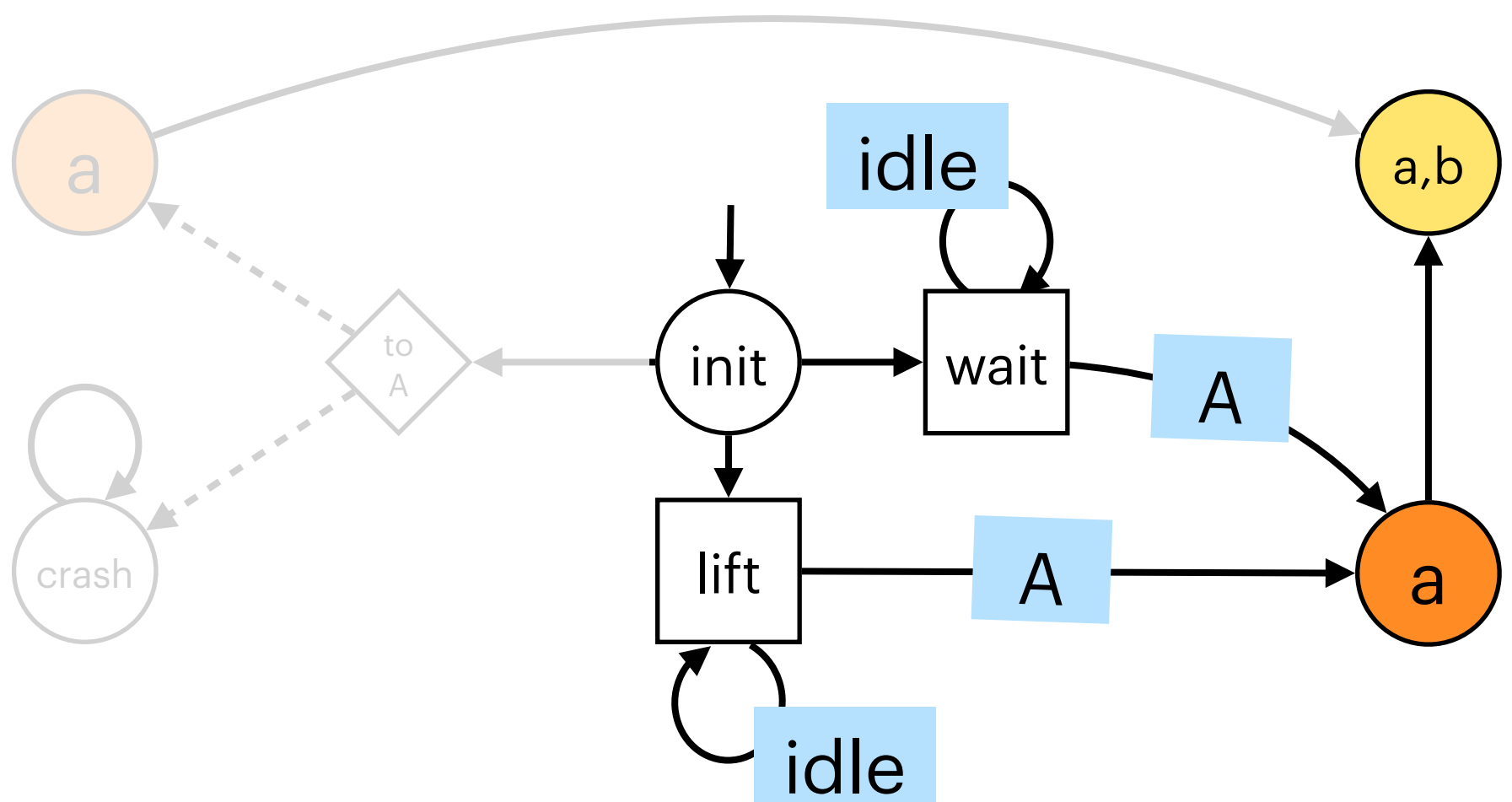


*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

The *Negotiation* Algorithm

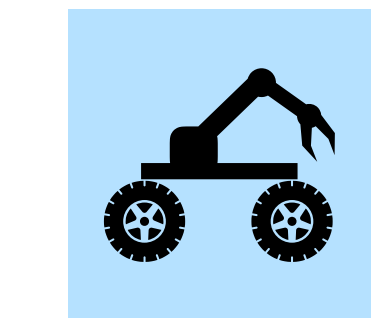


$$\varphi = \diamond(a, b)$$

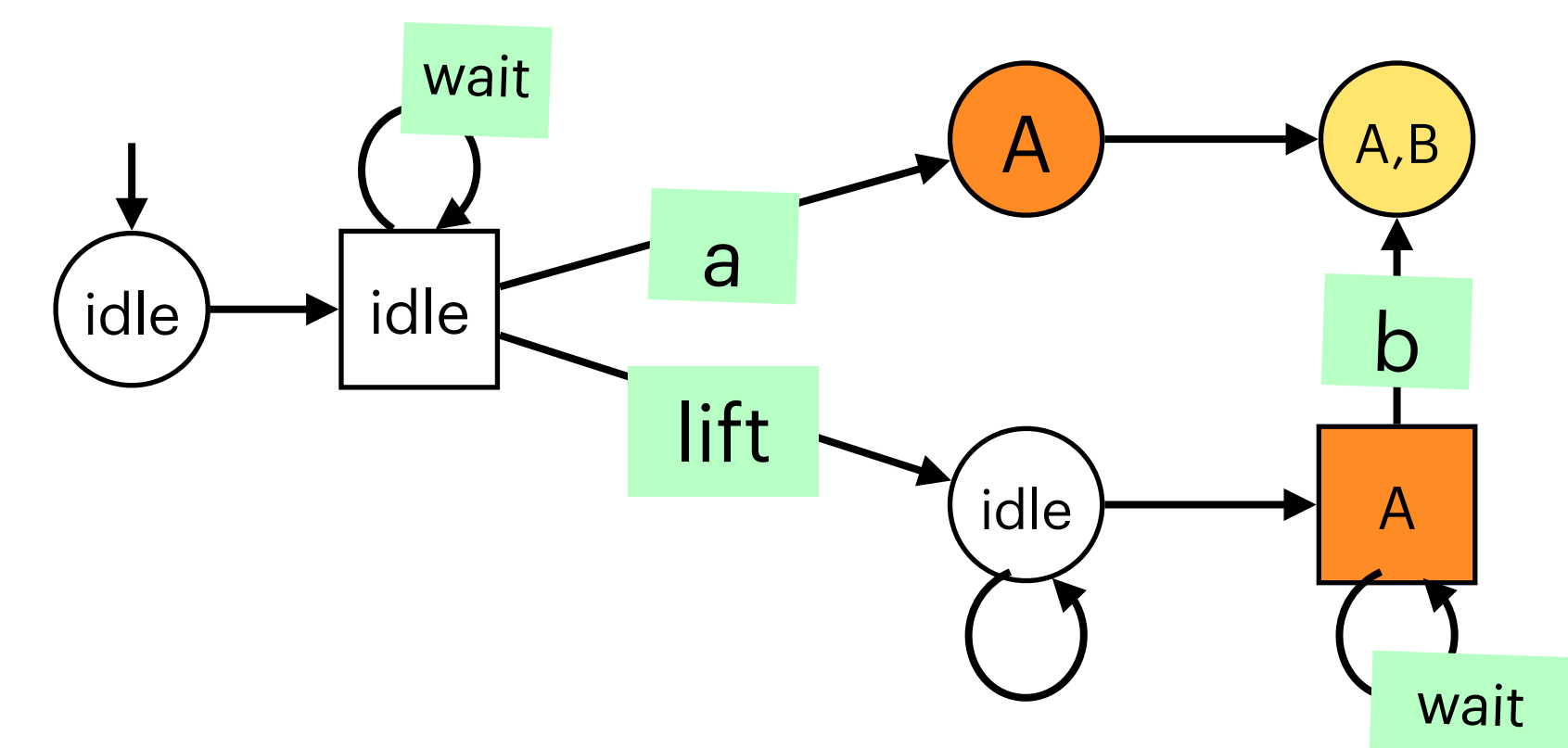


Negotiation rounds

$$\varphi = \diamond(a, b) \wedge (\top, \top)$$



$$\psi = \diamond(A, B)$$



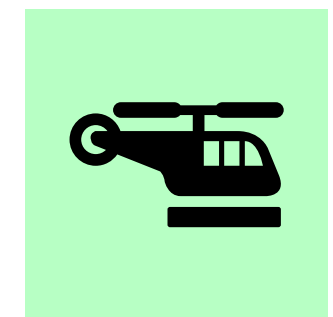
Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$

$$\lambda \quad \bigcirc^2 \neg \text{wait} \wedge \square (\text{lift} \rightarrow \bigcirc^2 \neg \text{wait})$$

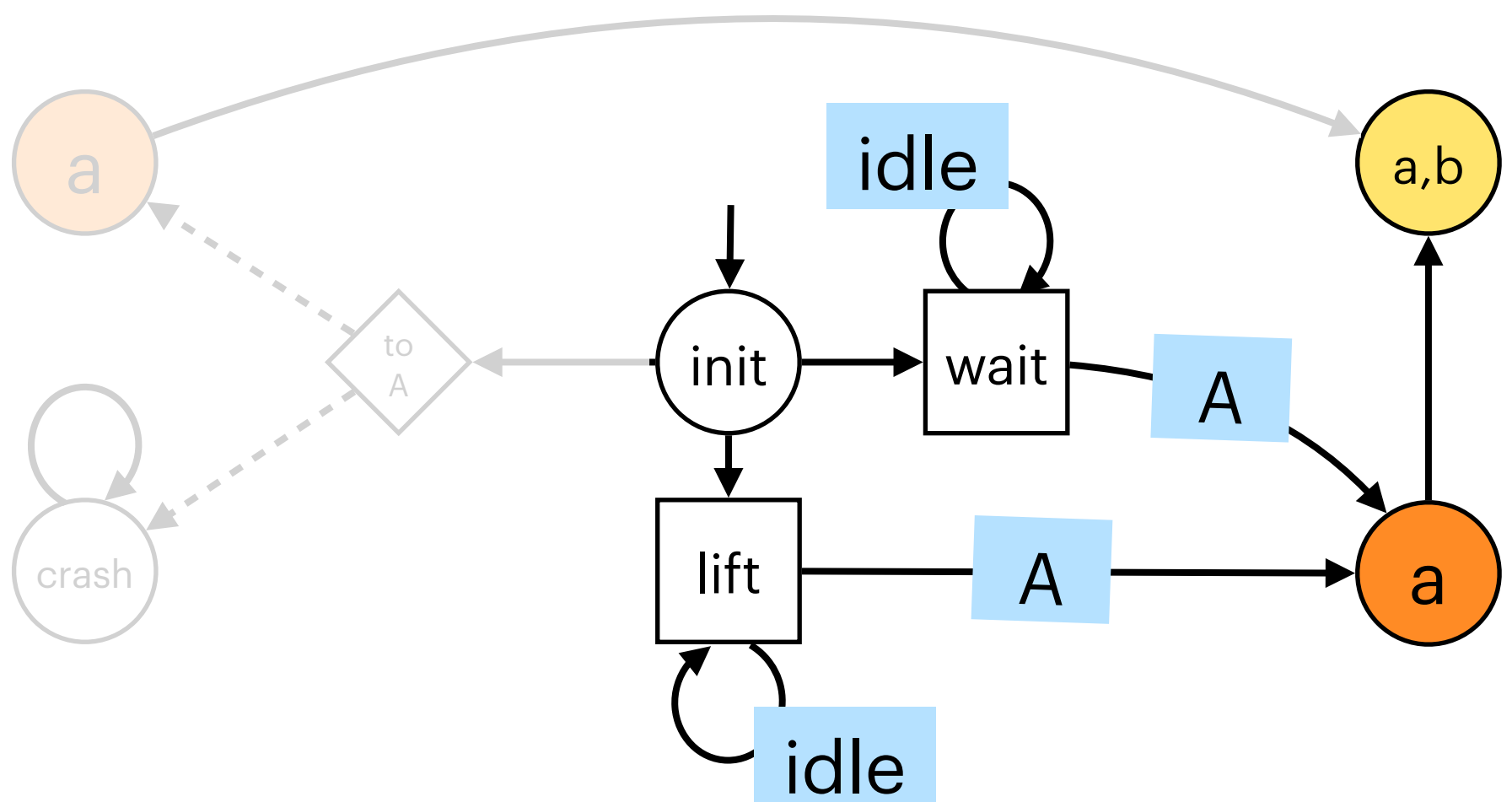
$$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle})$$

*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

The *Negotiation* Algorithm



$$\varphi = \diamond(a, b)$$



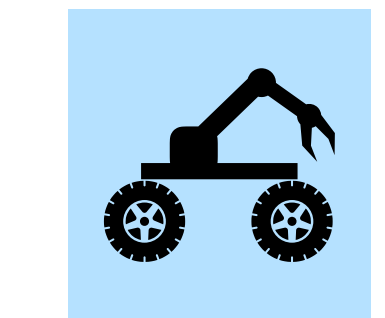
Negotiation rounds

$$\varphi = \diamond(a, b) \wedge (\top, \top) \quad \varphi' = (\bigcirc^2 \neg \text{idle} \rightarrow \diamond(a, b)) \wedge (\bigcirc^2 \neg \text{idle}, \lambda)$$

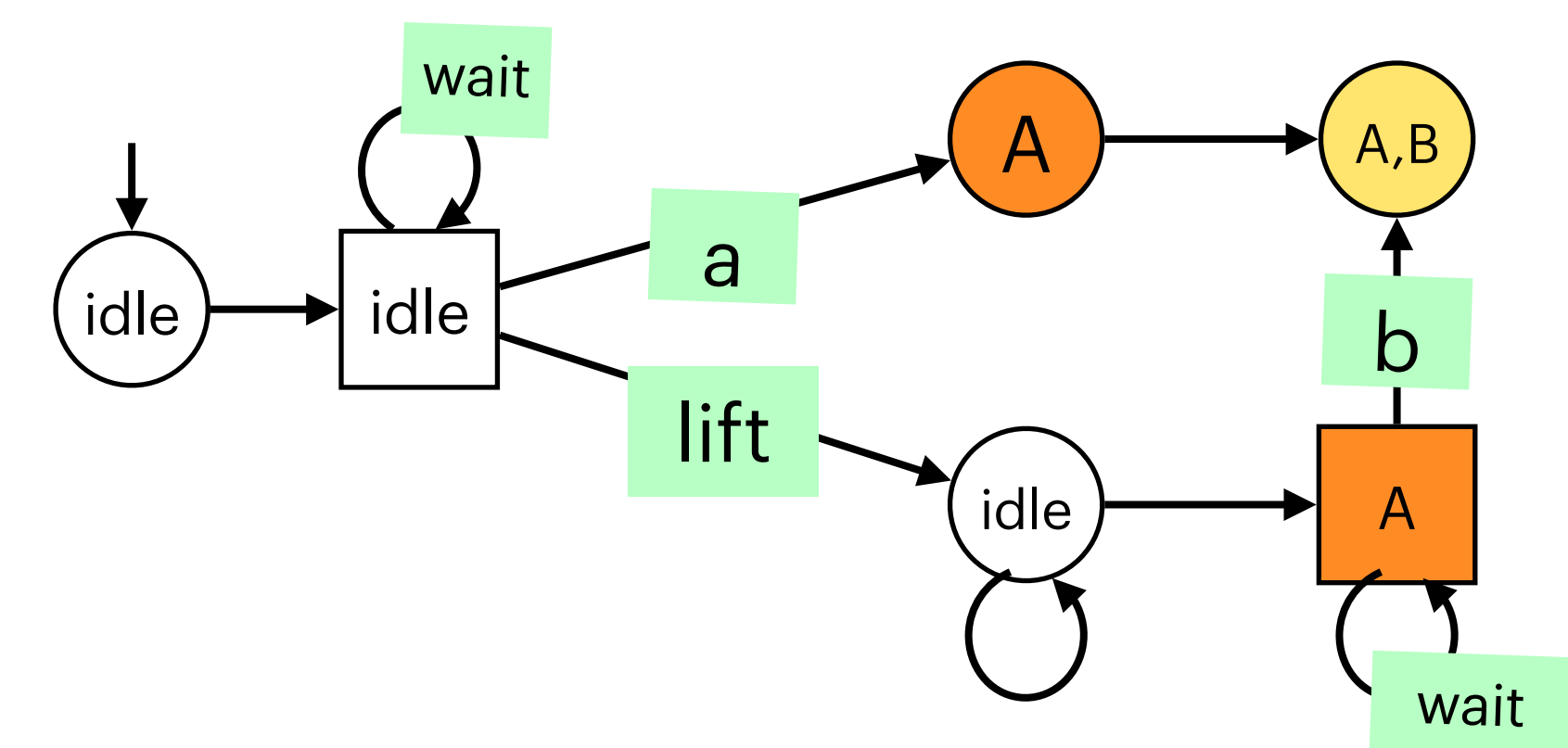
Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$

$$\bigcirc^2 \neg \text{wait} \wedge \square (\text{lift} \rightarrow \bigcirc^2 \neg \text{wait})$$

$$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle})$$

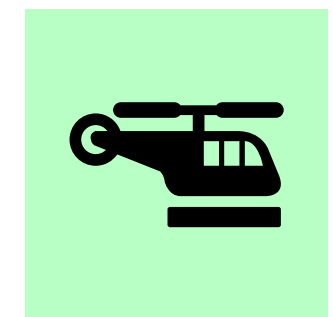


$$\psi = \diamond(A, B)$$

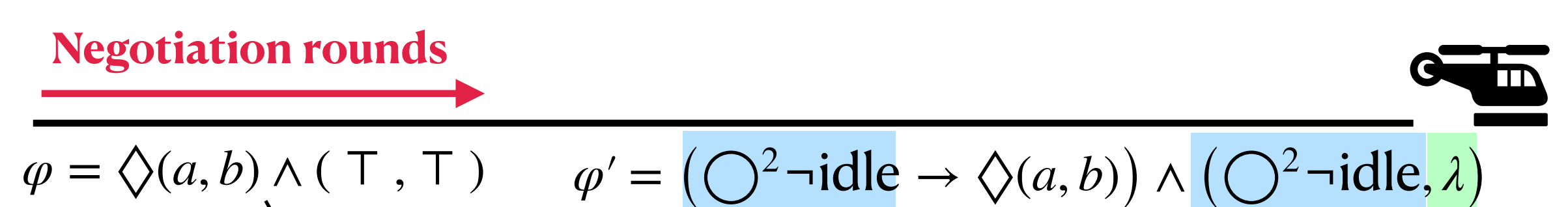
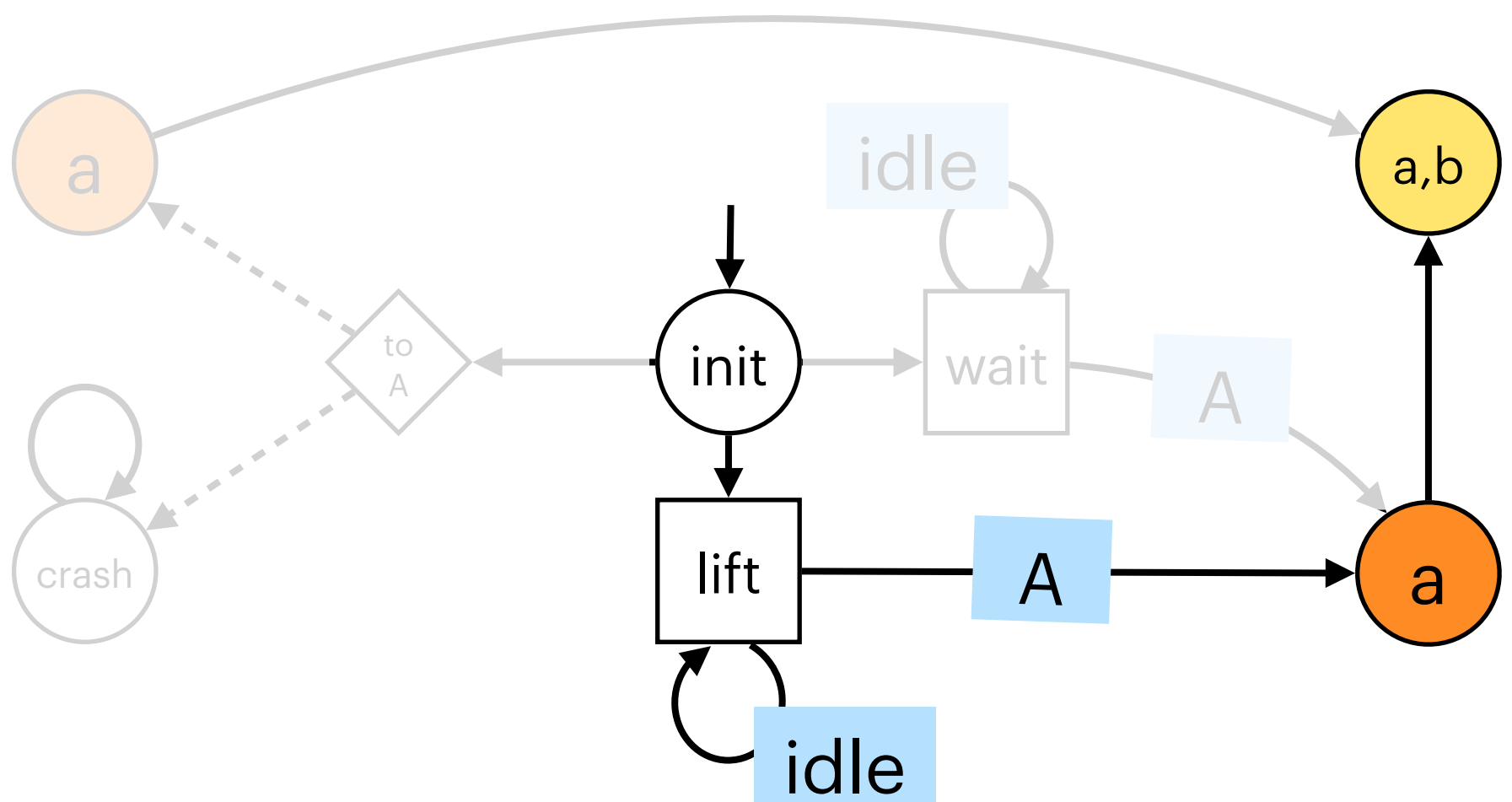


*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

The *Negotiation* Algorithm



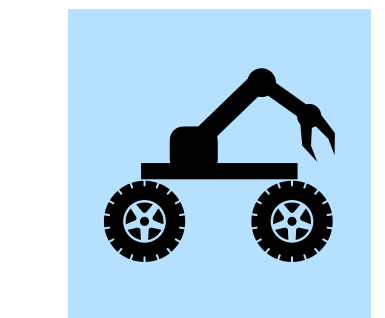
$$\varphi = \diamond(a, b)$$



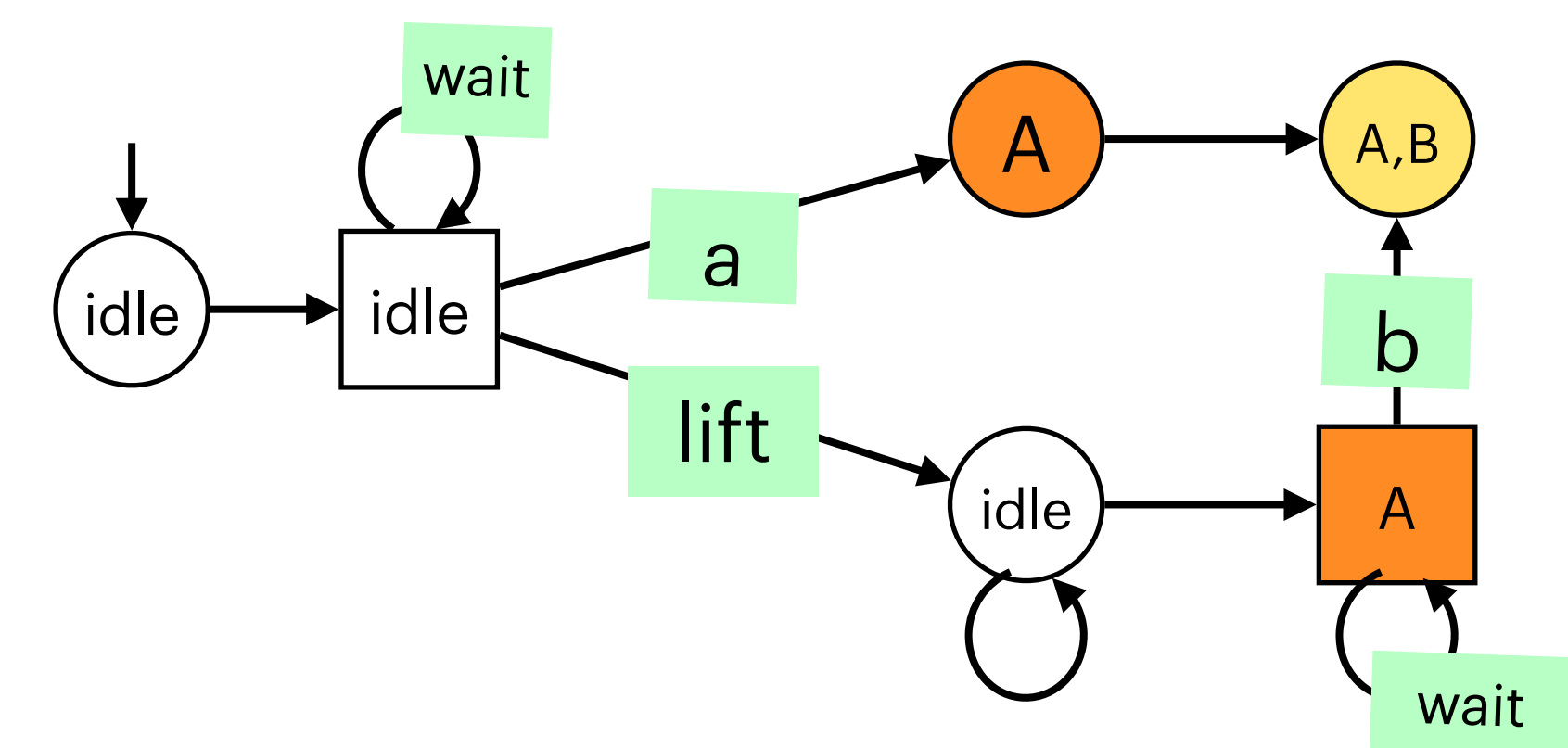
Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$

$$\begin{aligned} & \bigcirc^2 \neg \text{wait} \\ & \wedge \\ & \square (\text{lift} \rightarrow \bigcirc^2 \neg \text{wait}) \end{aligned}$$

$$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle})$$

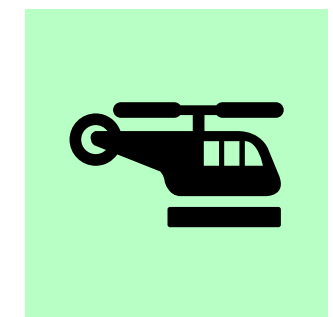


$$\psi = \diamond(A, B)$$

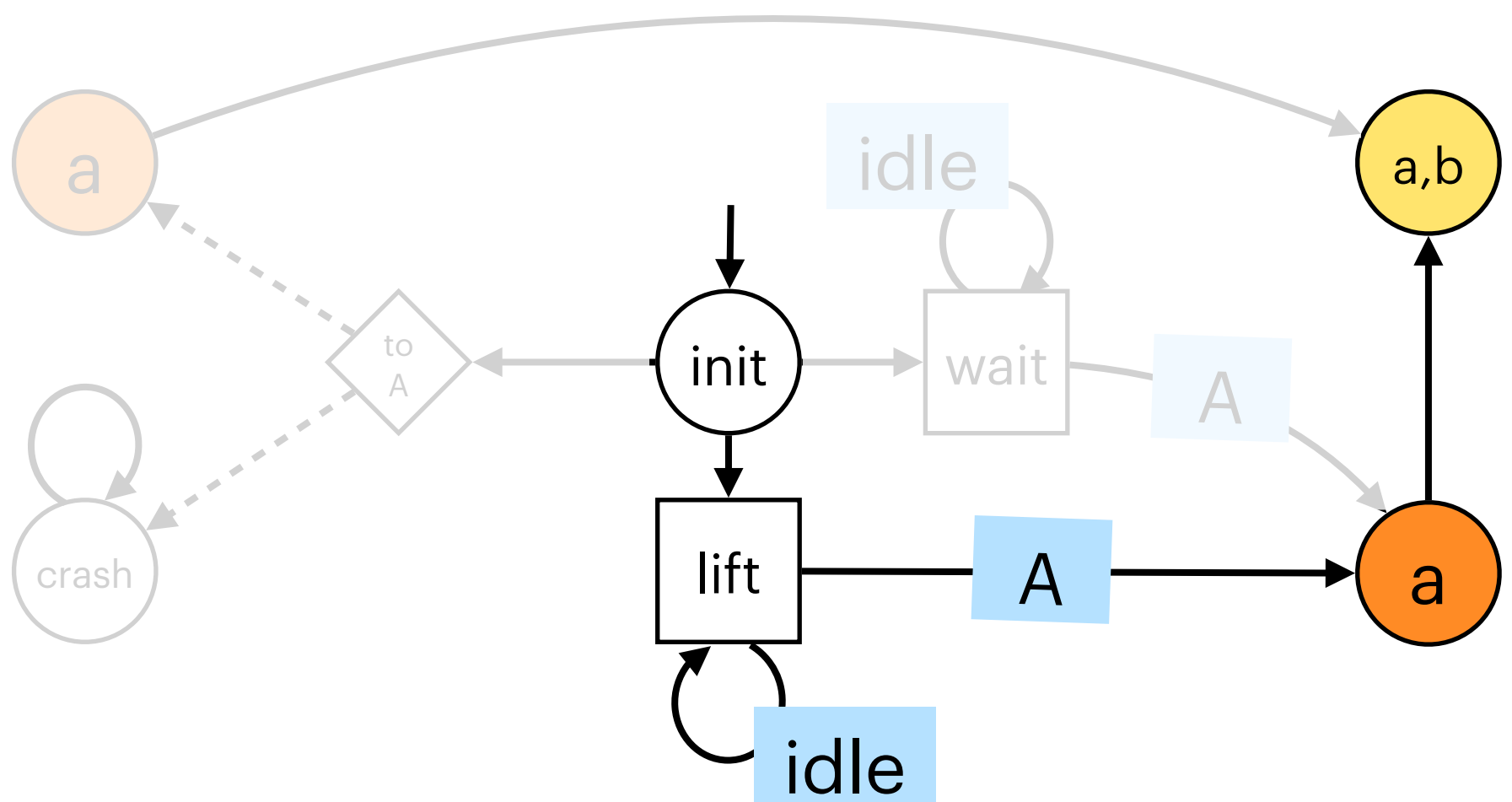


*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

The *Negotiation* Algorithm



$$\varphi = \diamond(a, b)$$



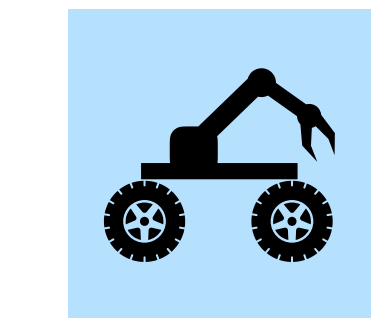
Negotiation rounds

$$\varphi = \diamond(a, b) \wedge (\top, \top) \quad \varphi' = (\bigcirc^2 \neg \text{idle} \rightarrow \diamond(a, b)) \wedge (\bigcirc^2 \neg \text{idle}, \lambda)$$

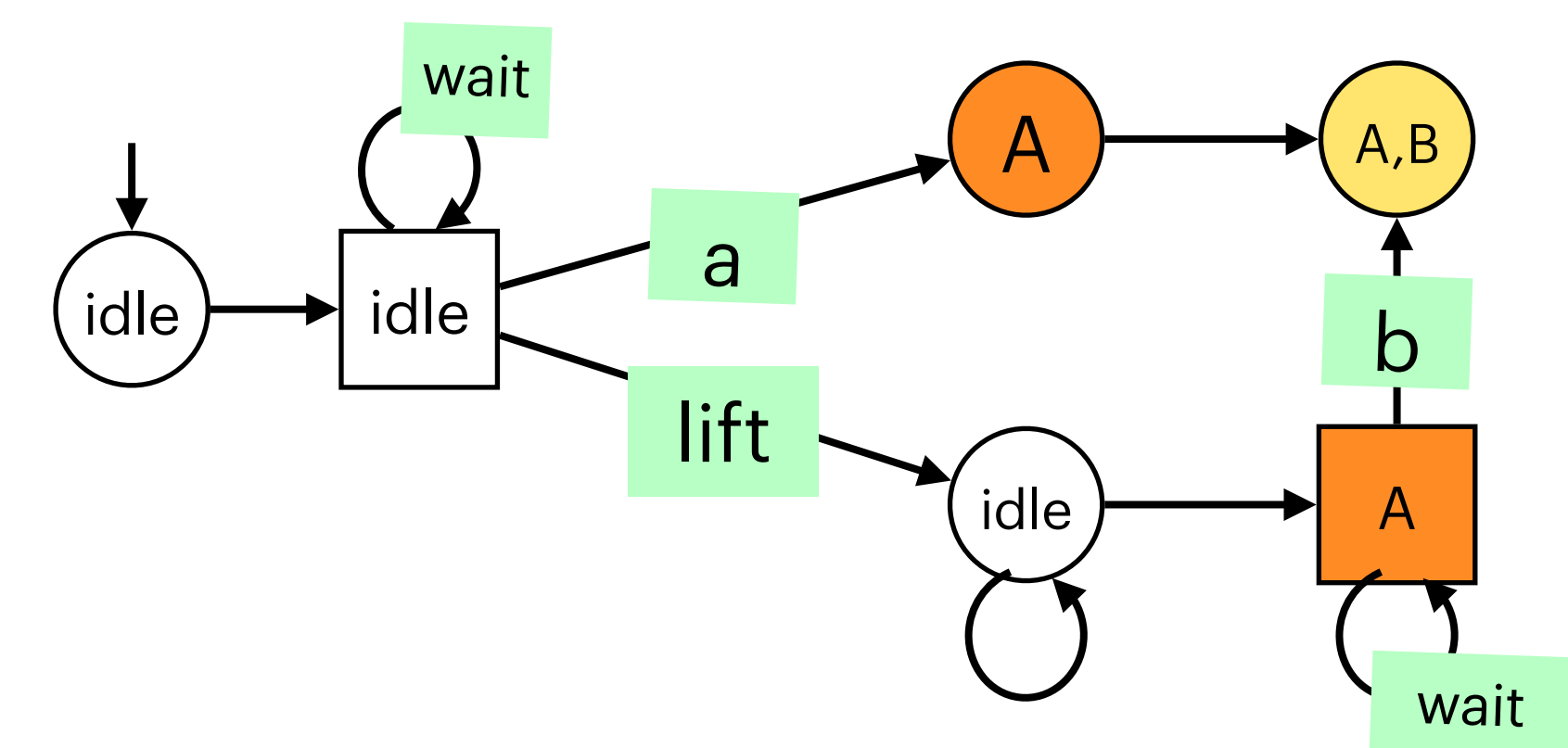
Safe sufficient assumption: $\bigcirc^2 \neg \text{idle}$

$$\bigcirc^2 \neg \text{wait} \wedge \square(\text{lift} \rightarrow \bigcirc^2 \neg \text{wait})$$

$$\psi' = \diamond(A, B) \wedge (\top, \bigcirc^2 \neg \text{idle}) \quad \psi'' = (\lambda \rightarrow \diamond(A, B)) \wedge (\lambda, \bigcirc \neg \text{idle})$$

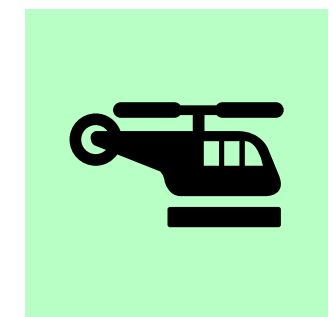


$$\psi = \diamond(A, B)$$

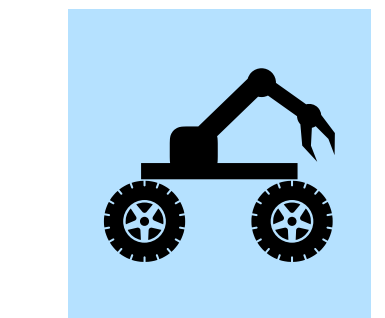
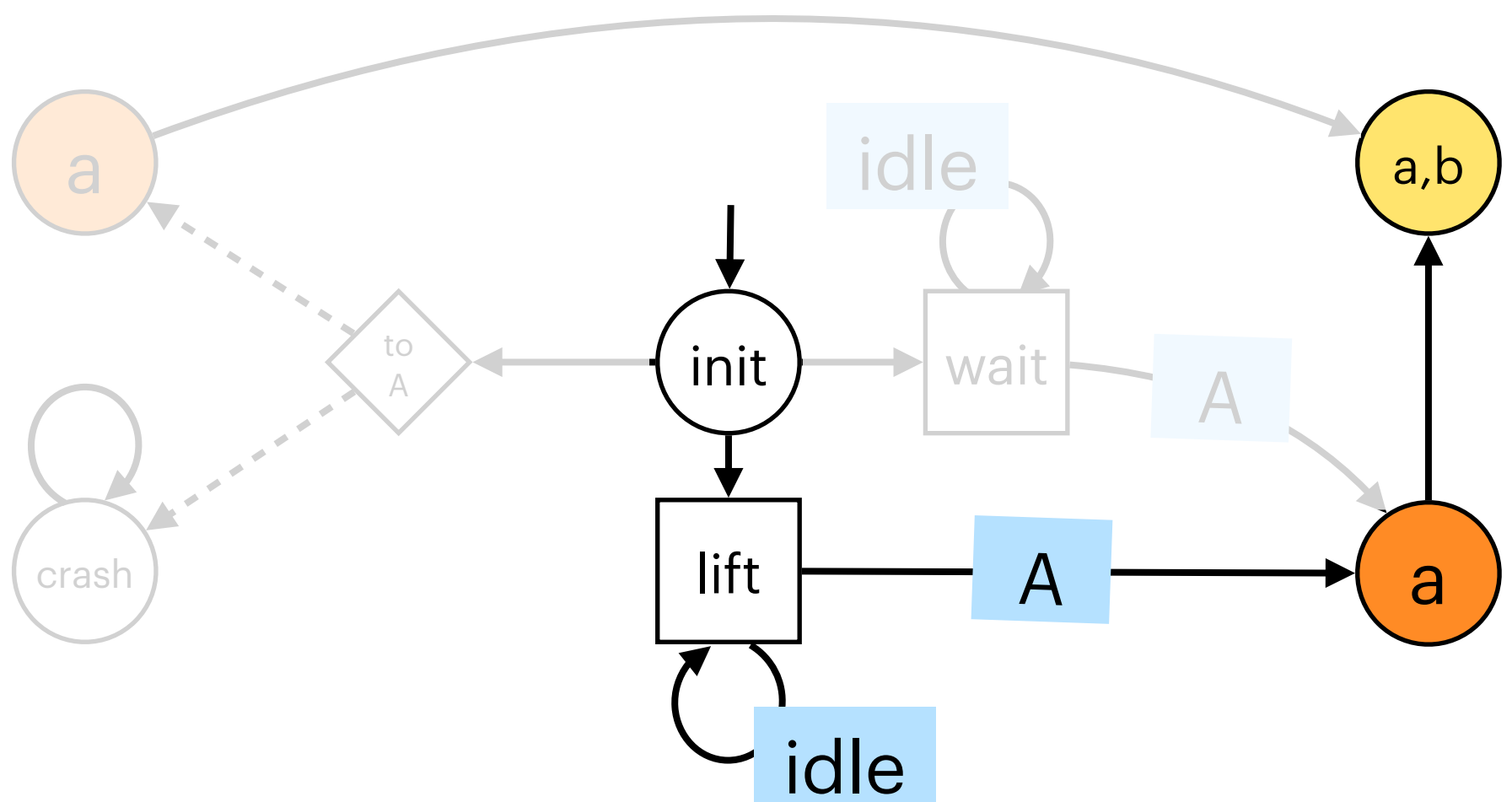


*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

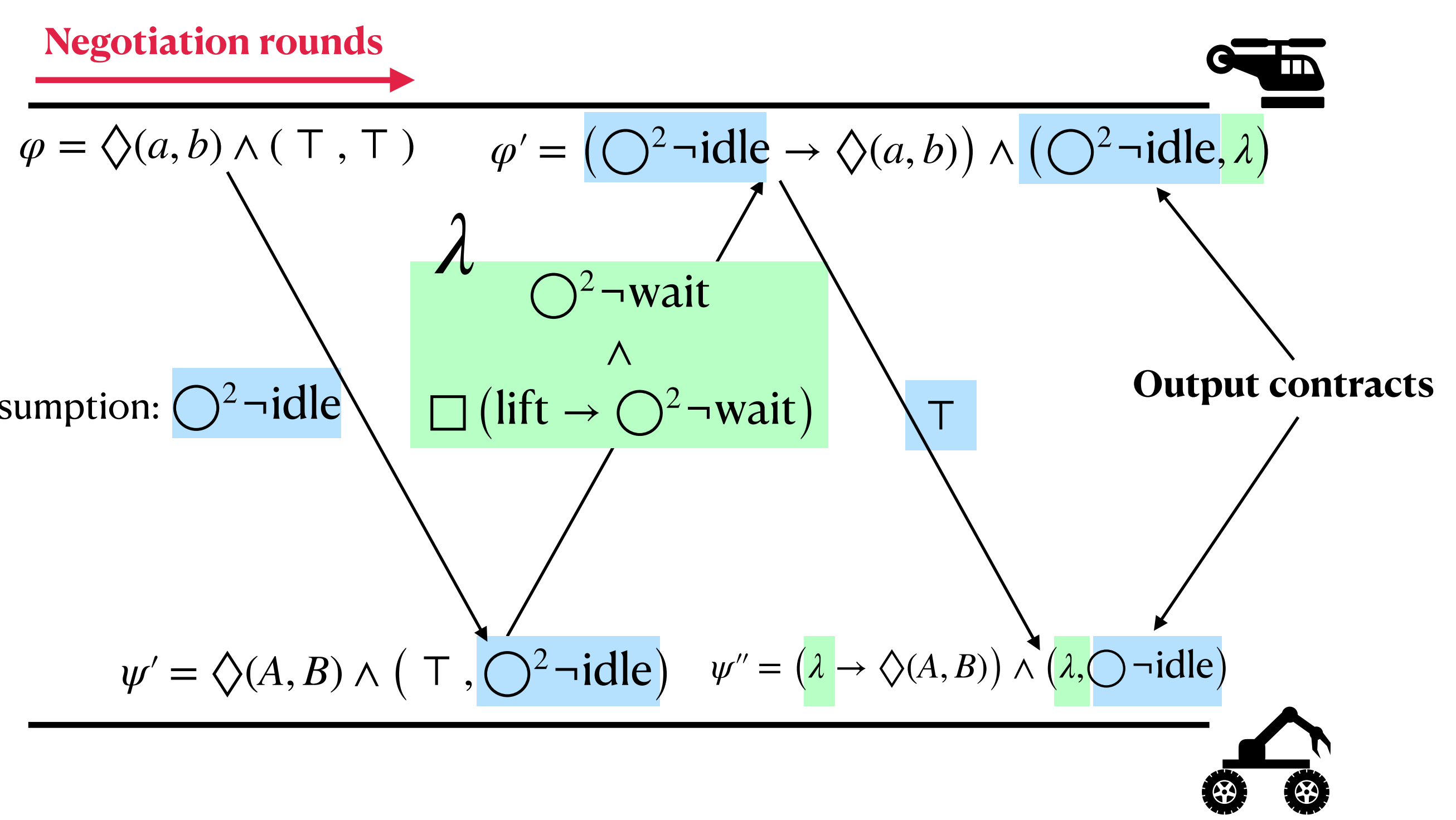
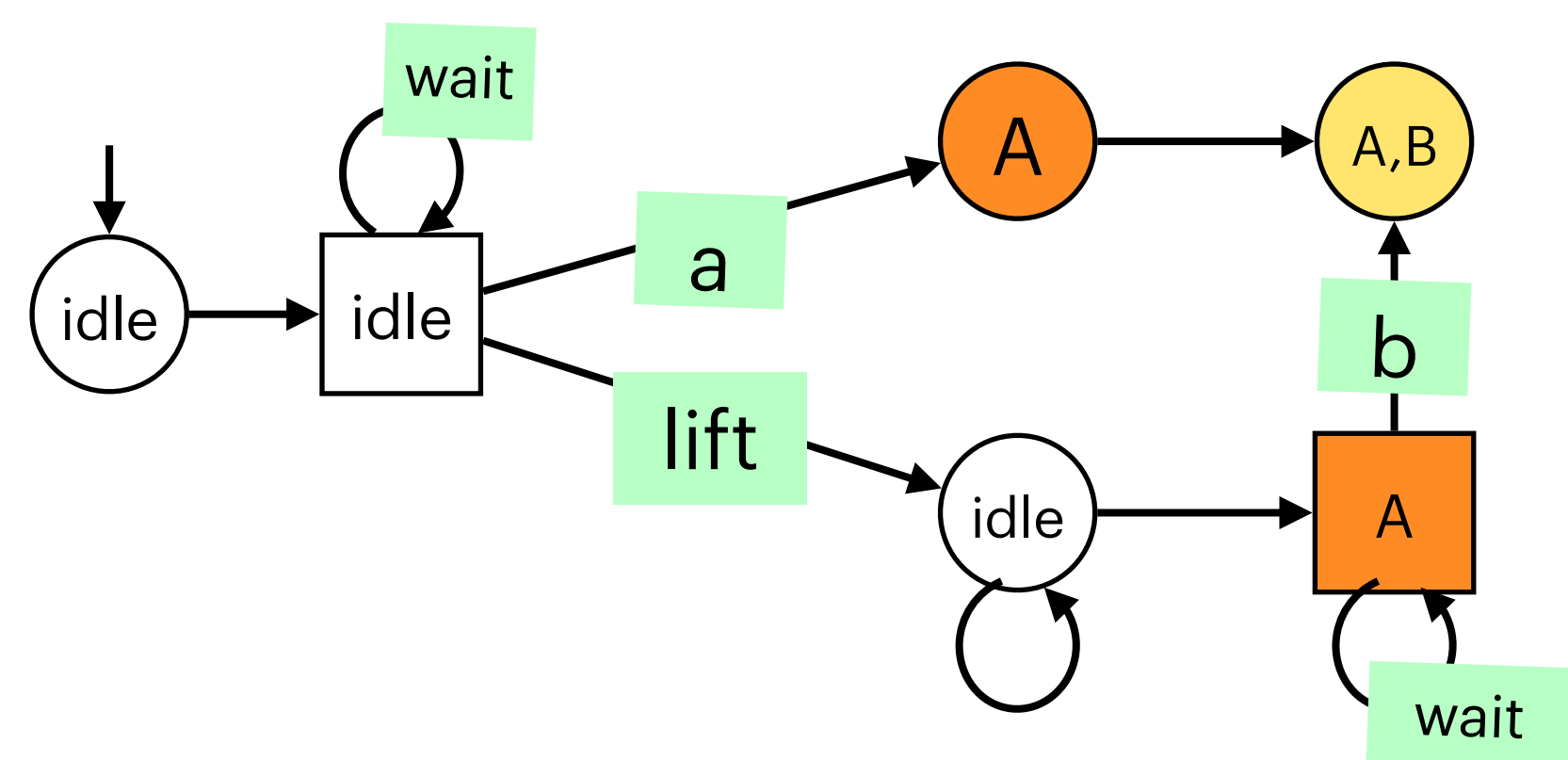
The *Negotiation* Algorithm



$$\varphi = \diamond(a, b)$$



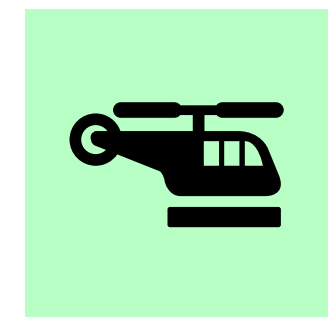
$$\psi = \diamond(A, B)$$



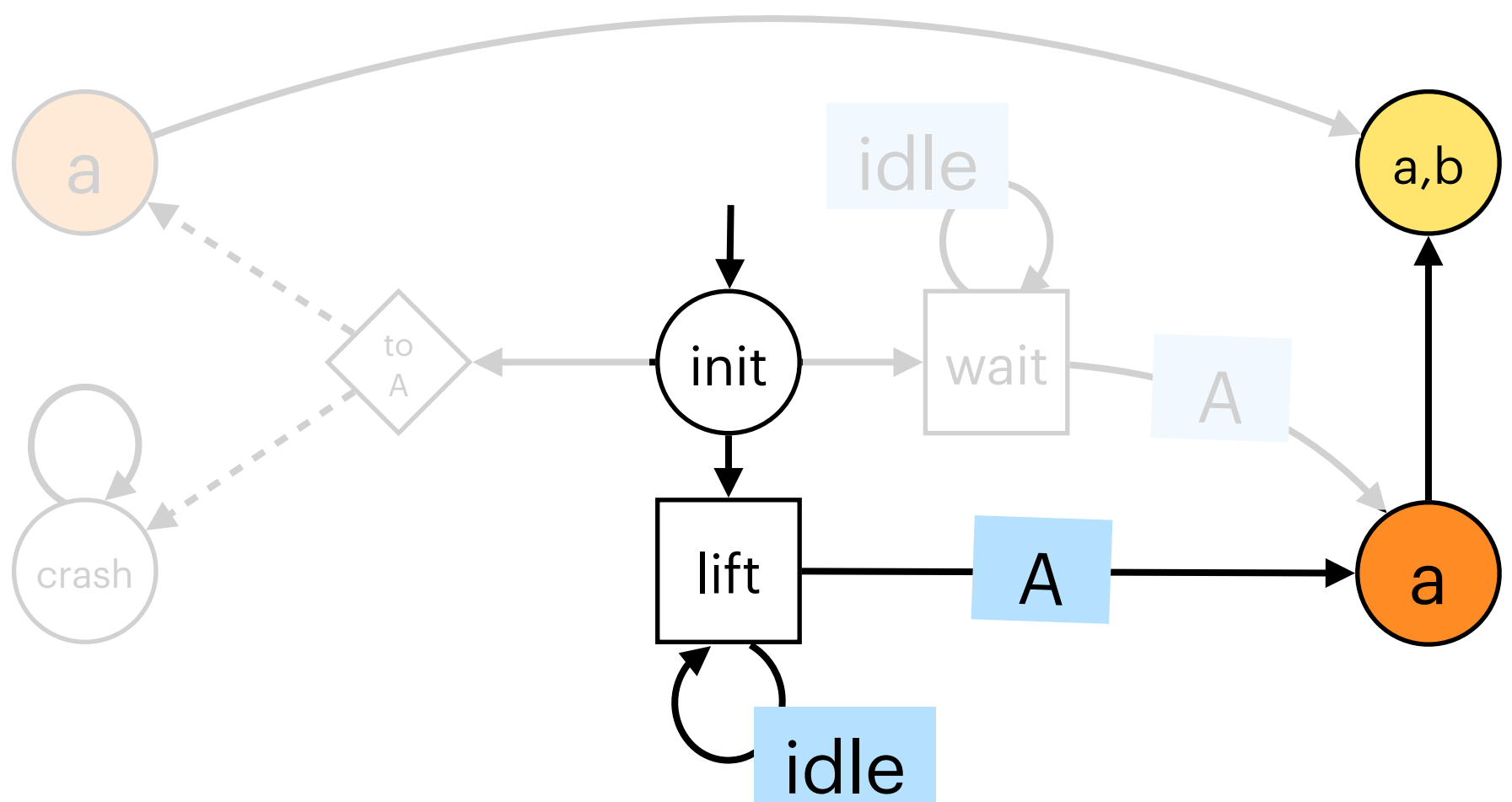
*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

The *Negotiation* Algorithm

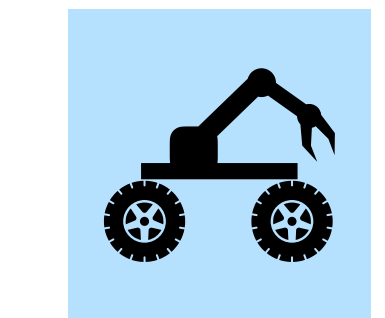
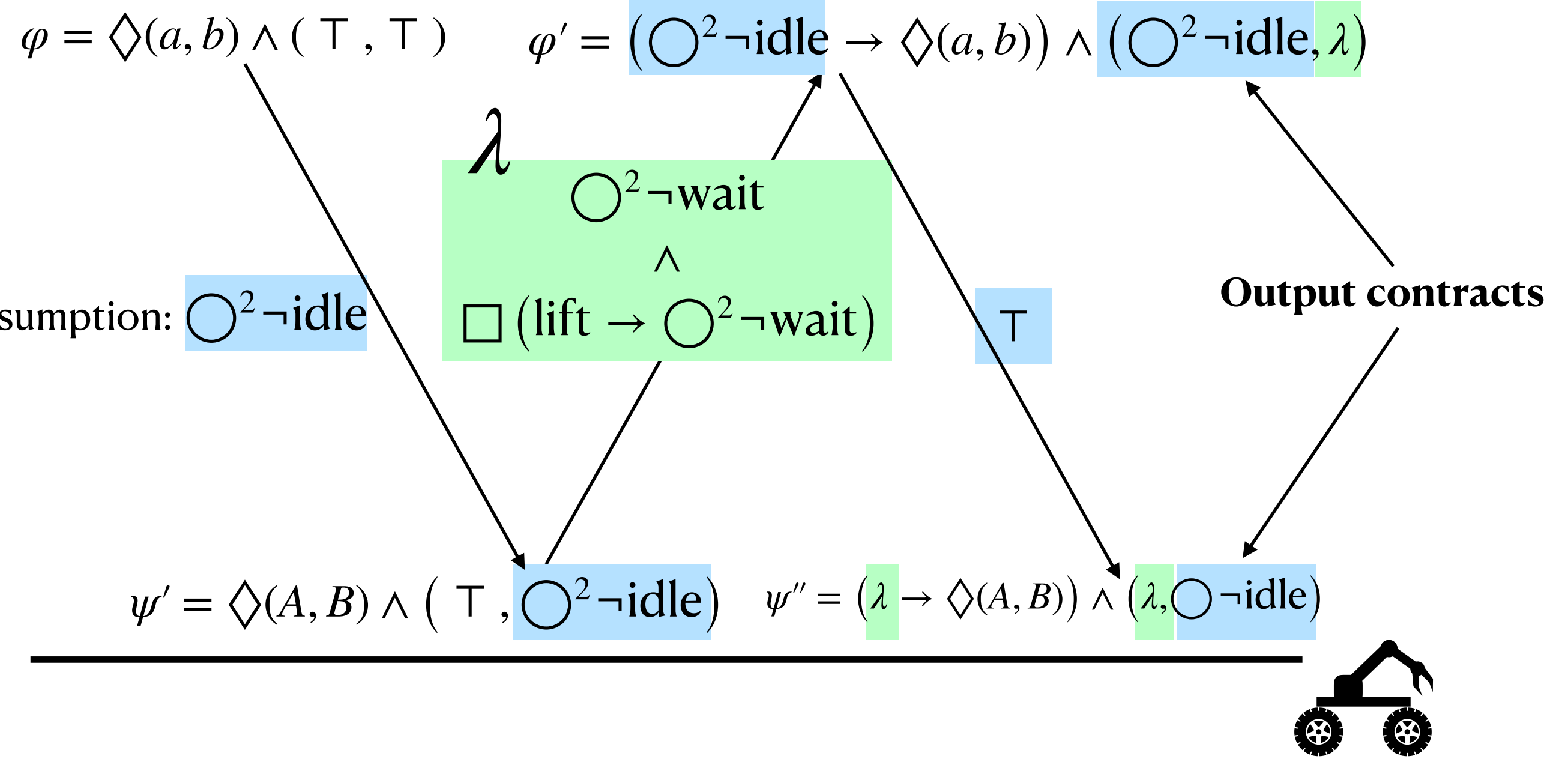
- Features:
- ✓ Sound
 - ✗ Incomplete



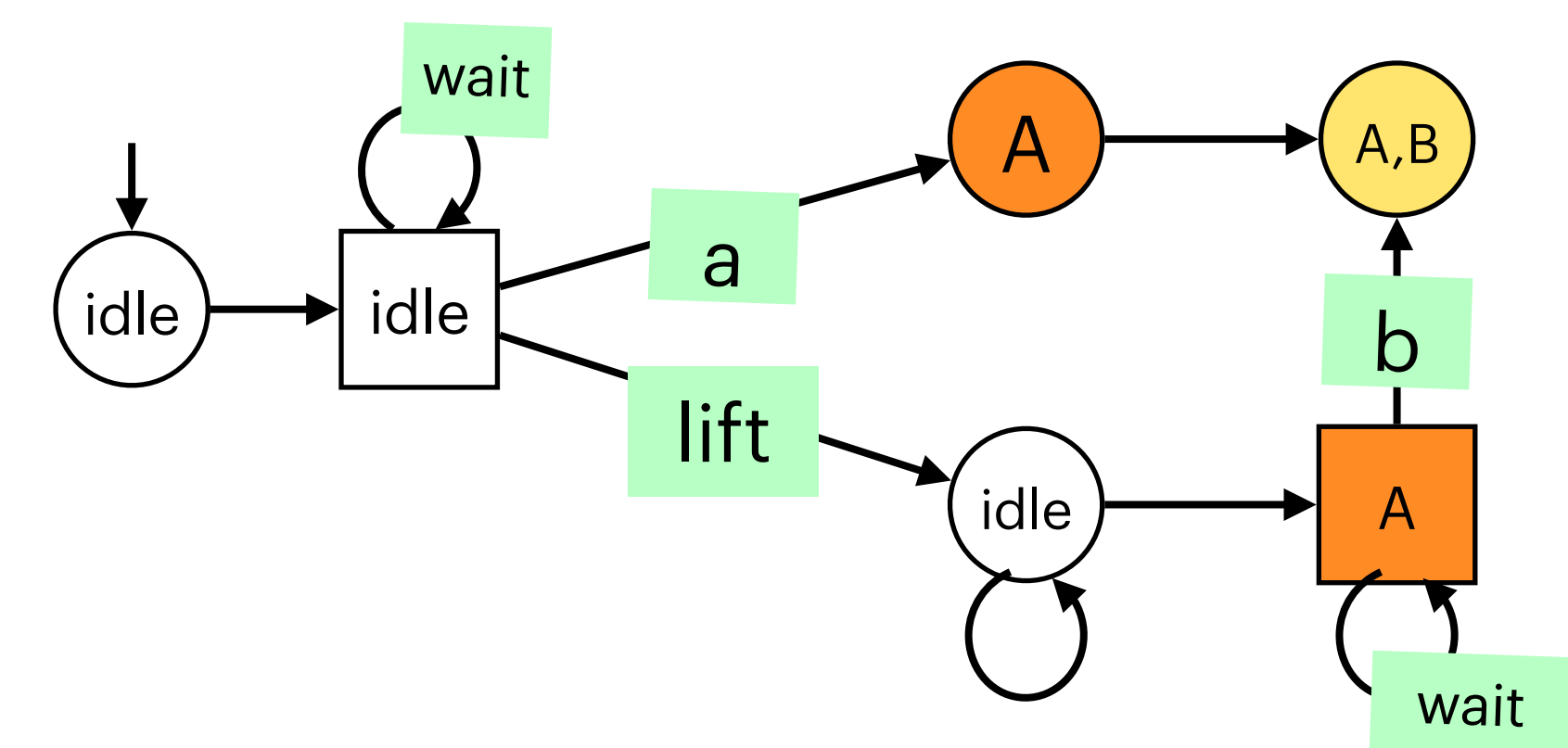
$$\varphi = \diamond(a, b)$$



Negotiation rounds \rightarrow



$$\psi = \diamond(A, B)$$



*Krishnendu Chatterjee, Thomas A. Henzinger, and Barbara Jobstmann. "Environment assumptions for synthesis." *CONCUR*, 2008.

The Problem

The *Negotiation* Algorithm

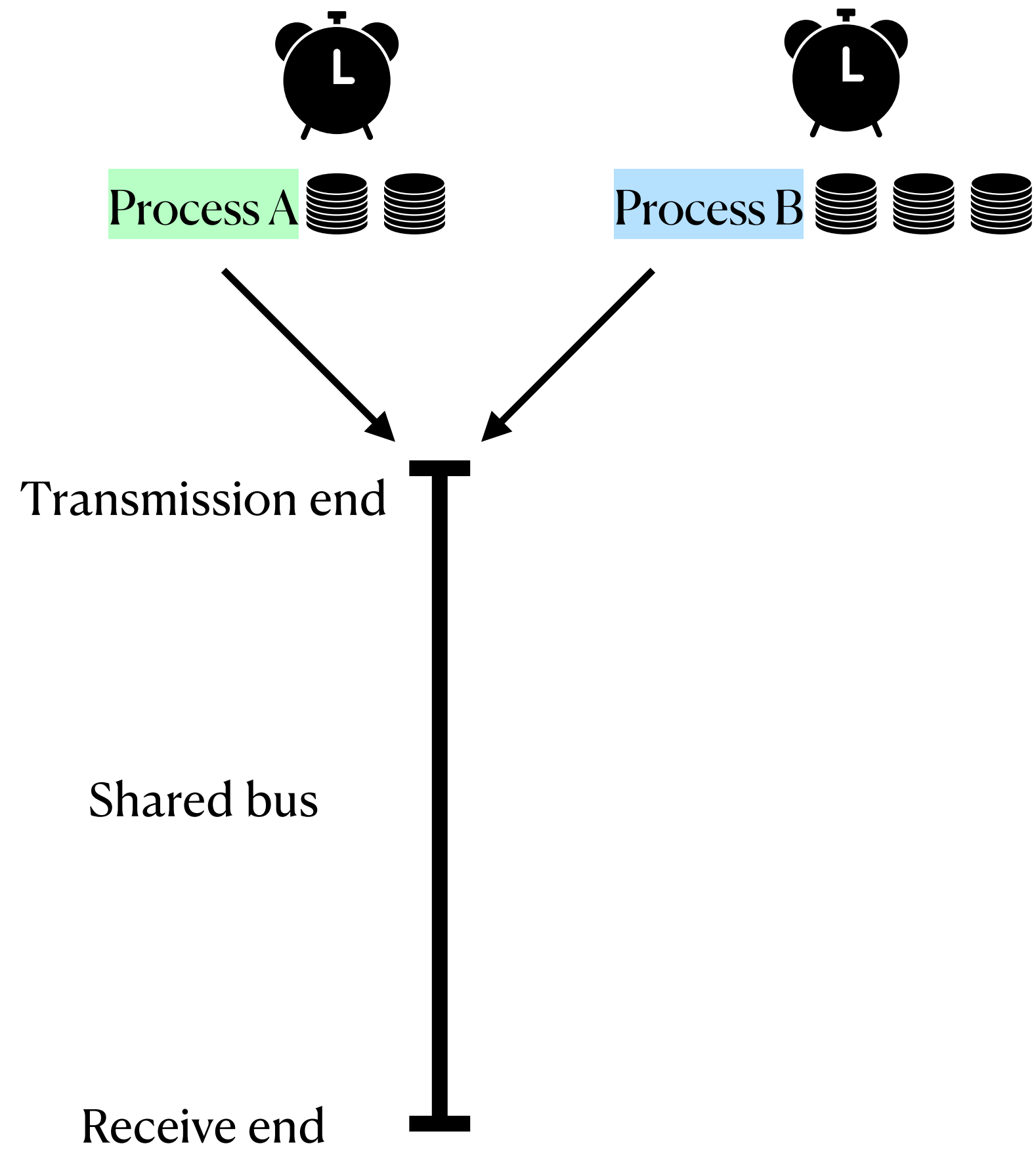
Experiments

The Problem

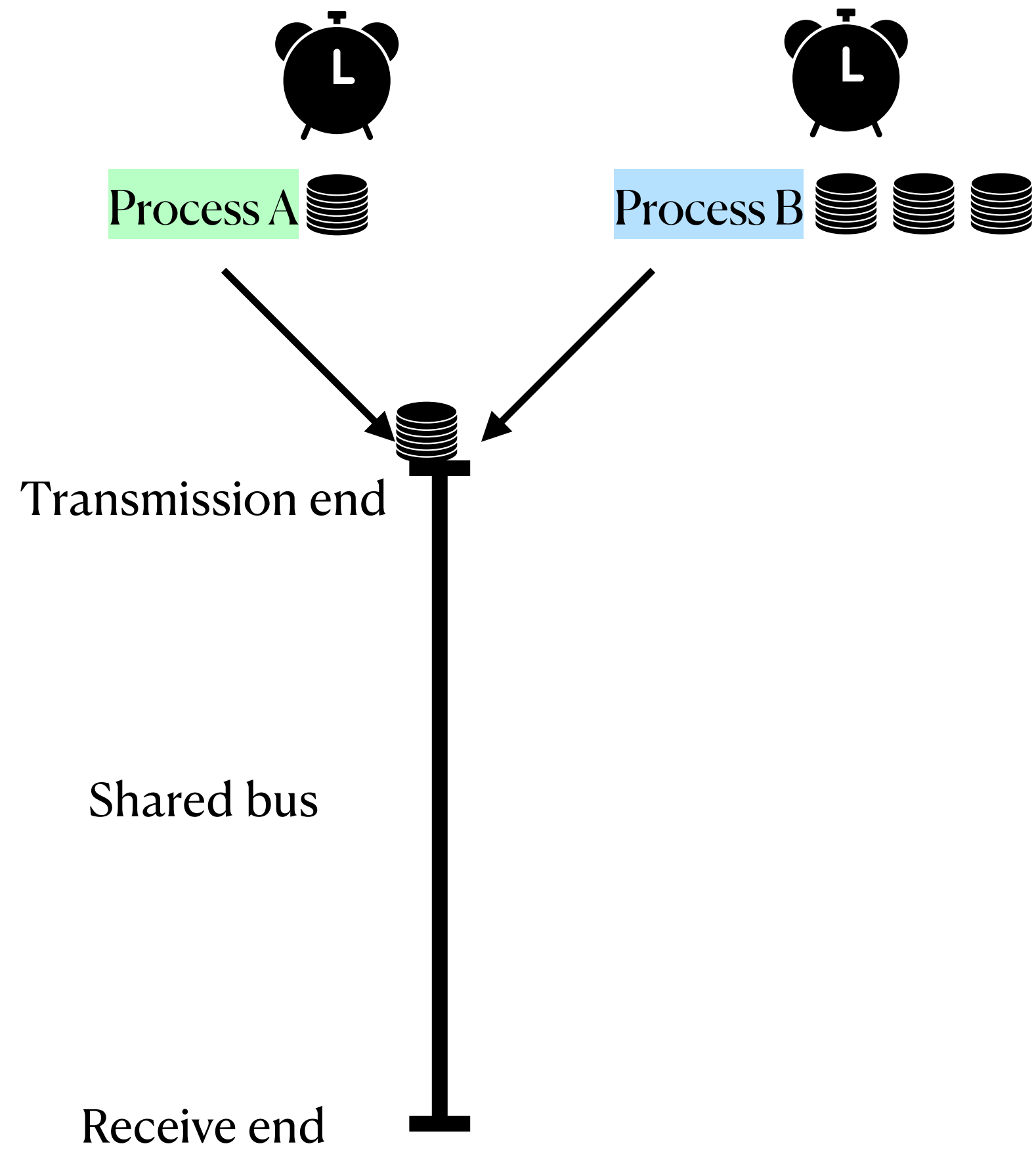
The *Negotiation* Algorithm

Experiments

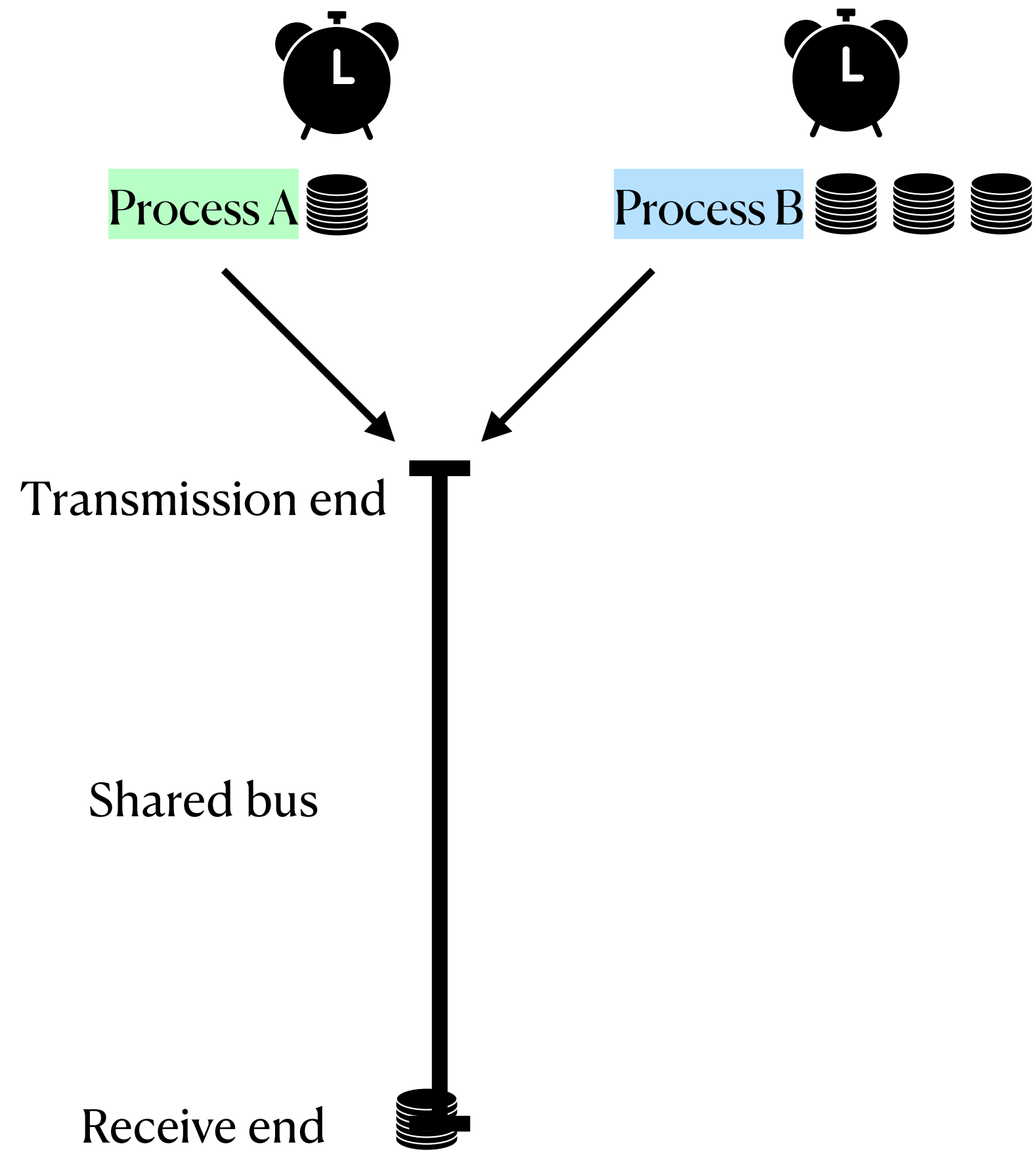
The Mutual Exclusion Example



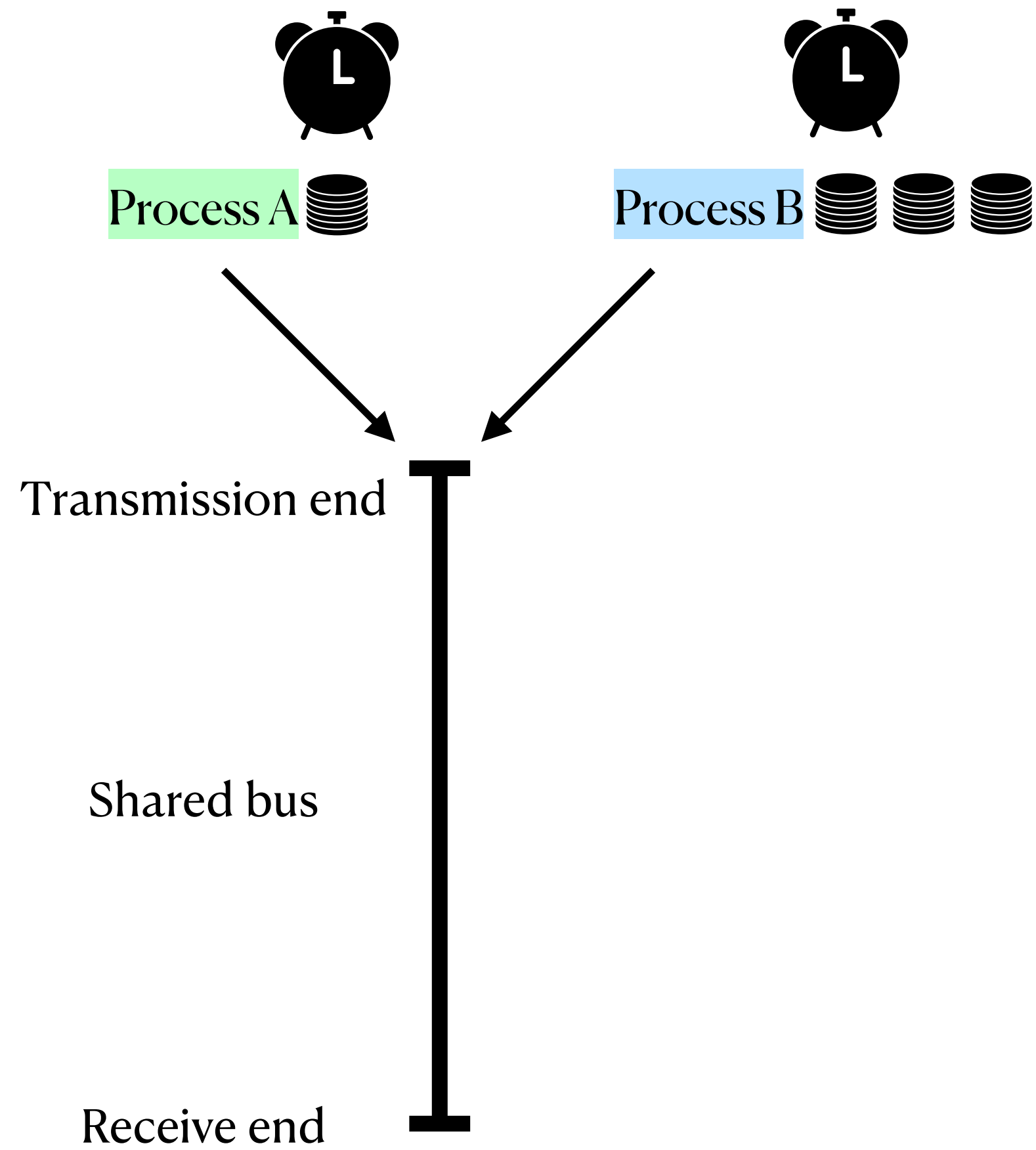
The Mutual Exclusion Example



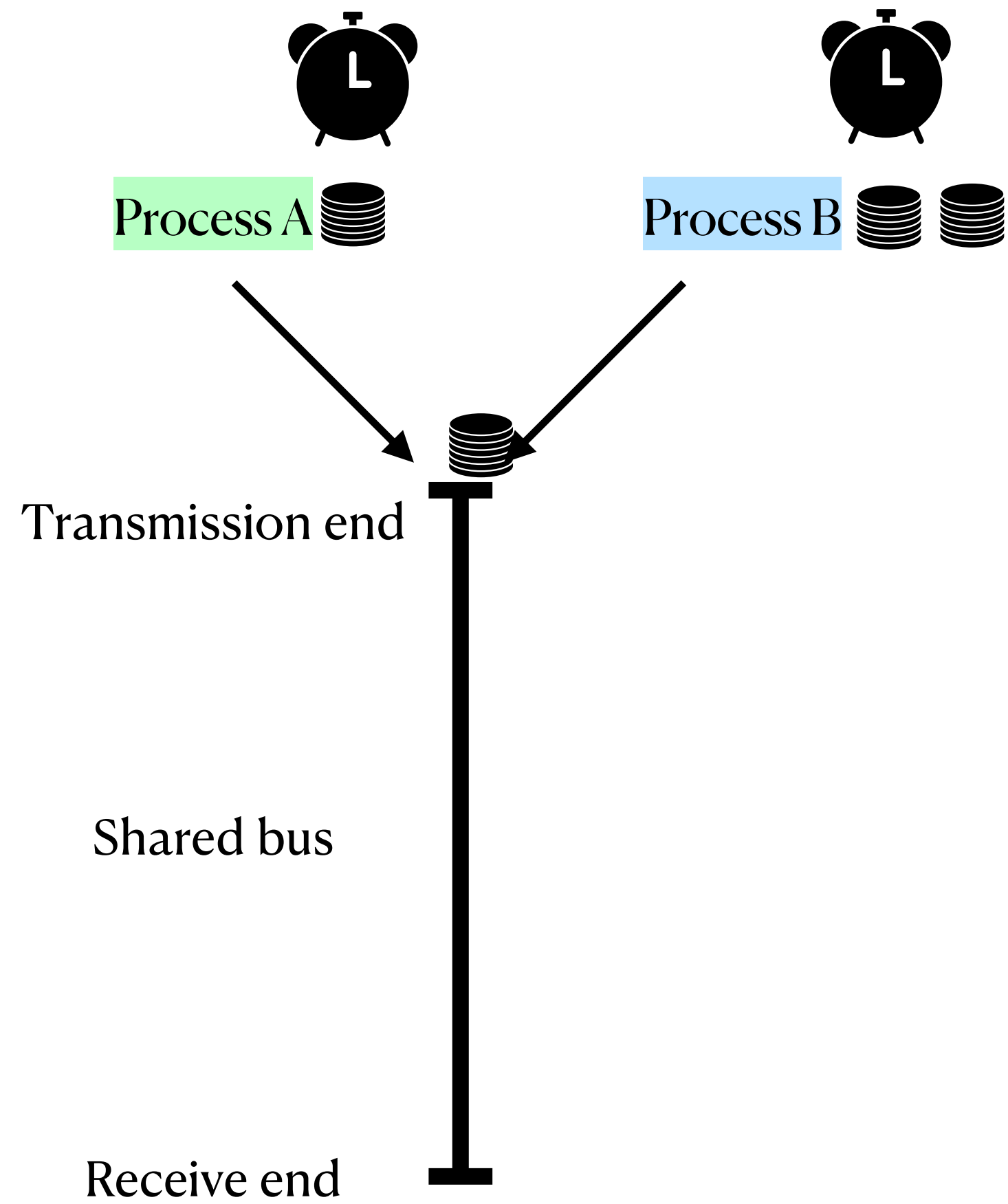
The Mutual Exclusion Example



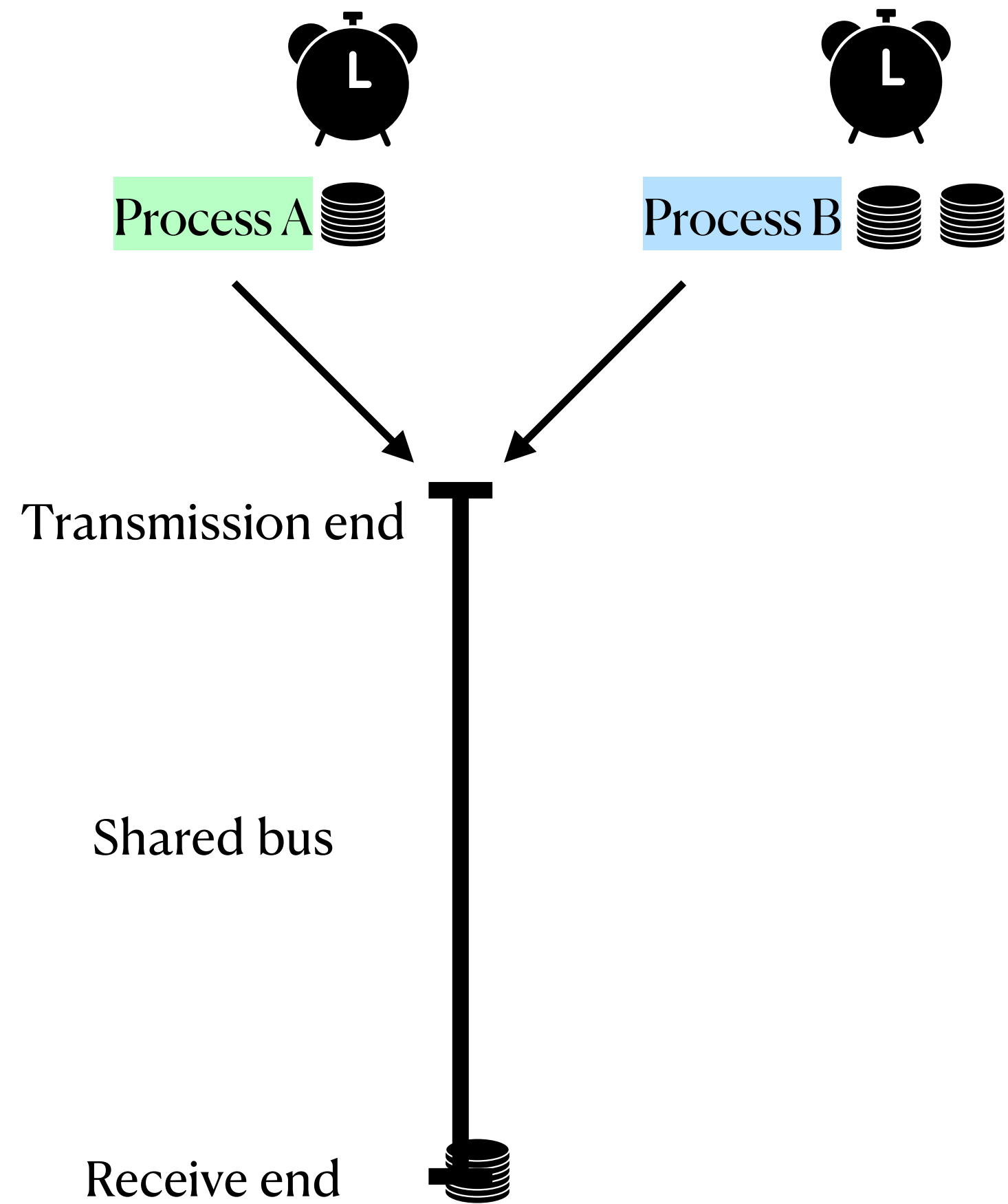
The Mutual Exclusion Example



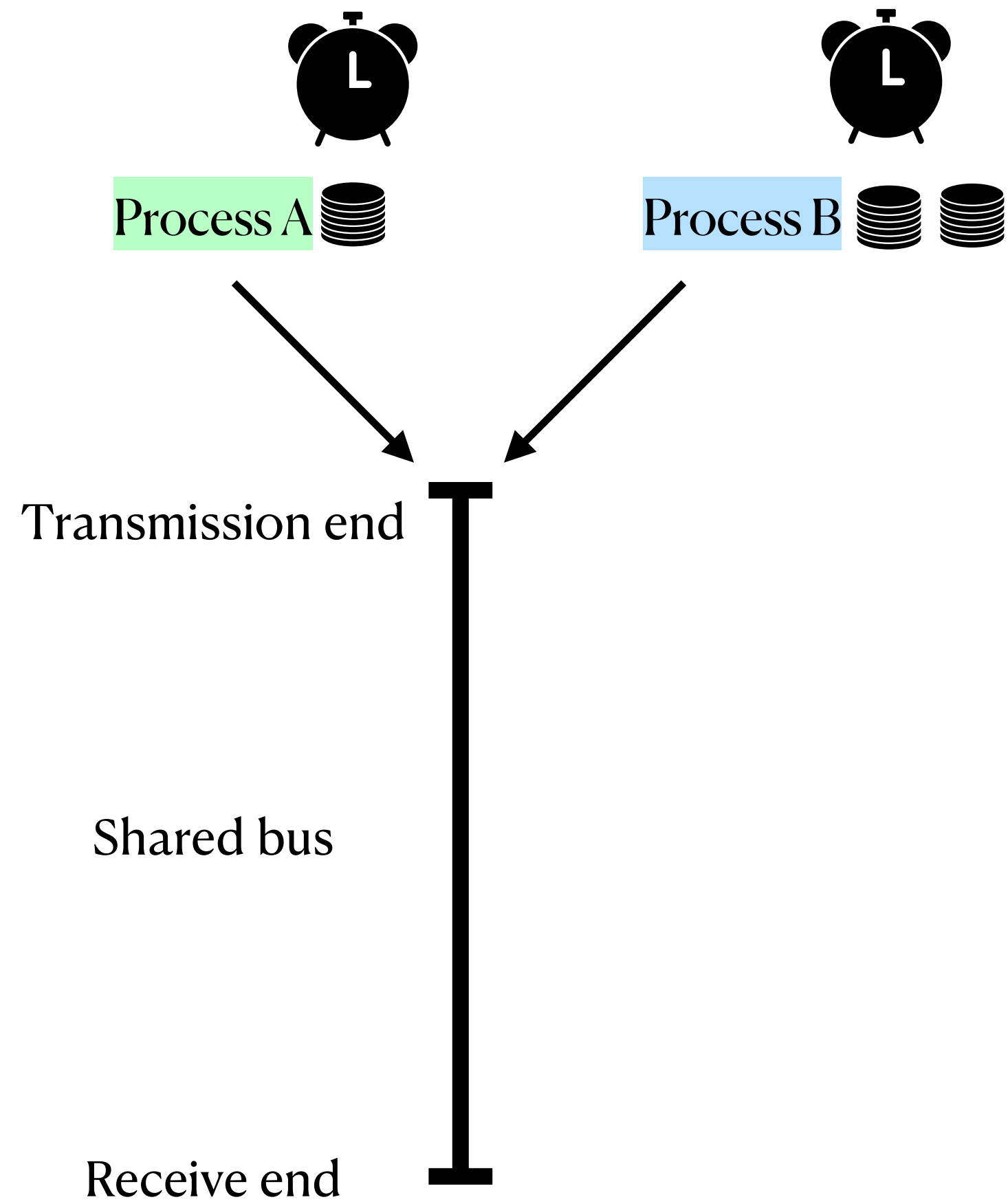
The Mutual Exclusion Example



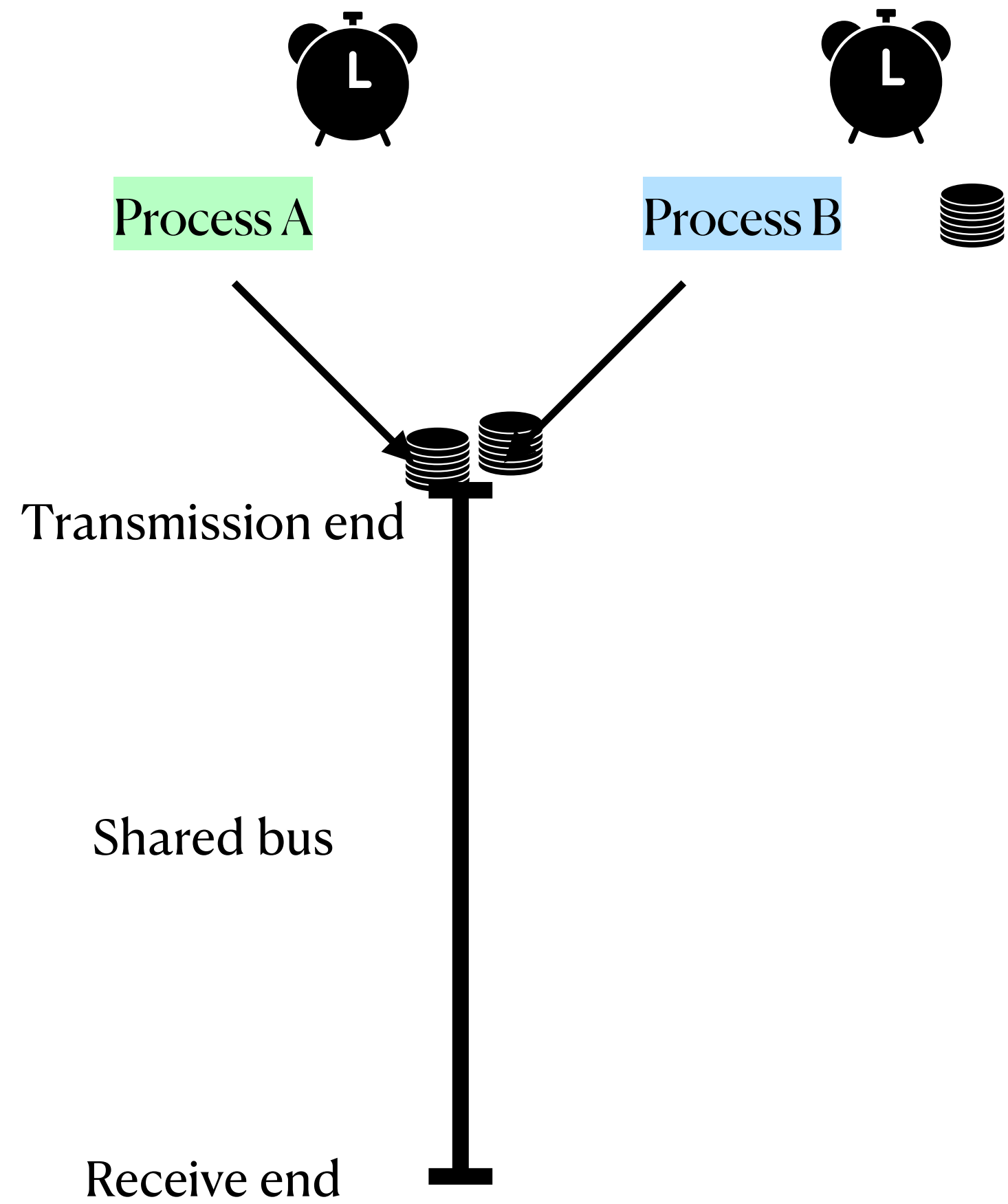
The Mutual Exclusion Example



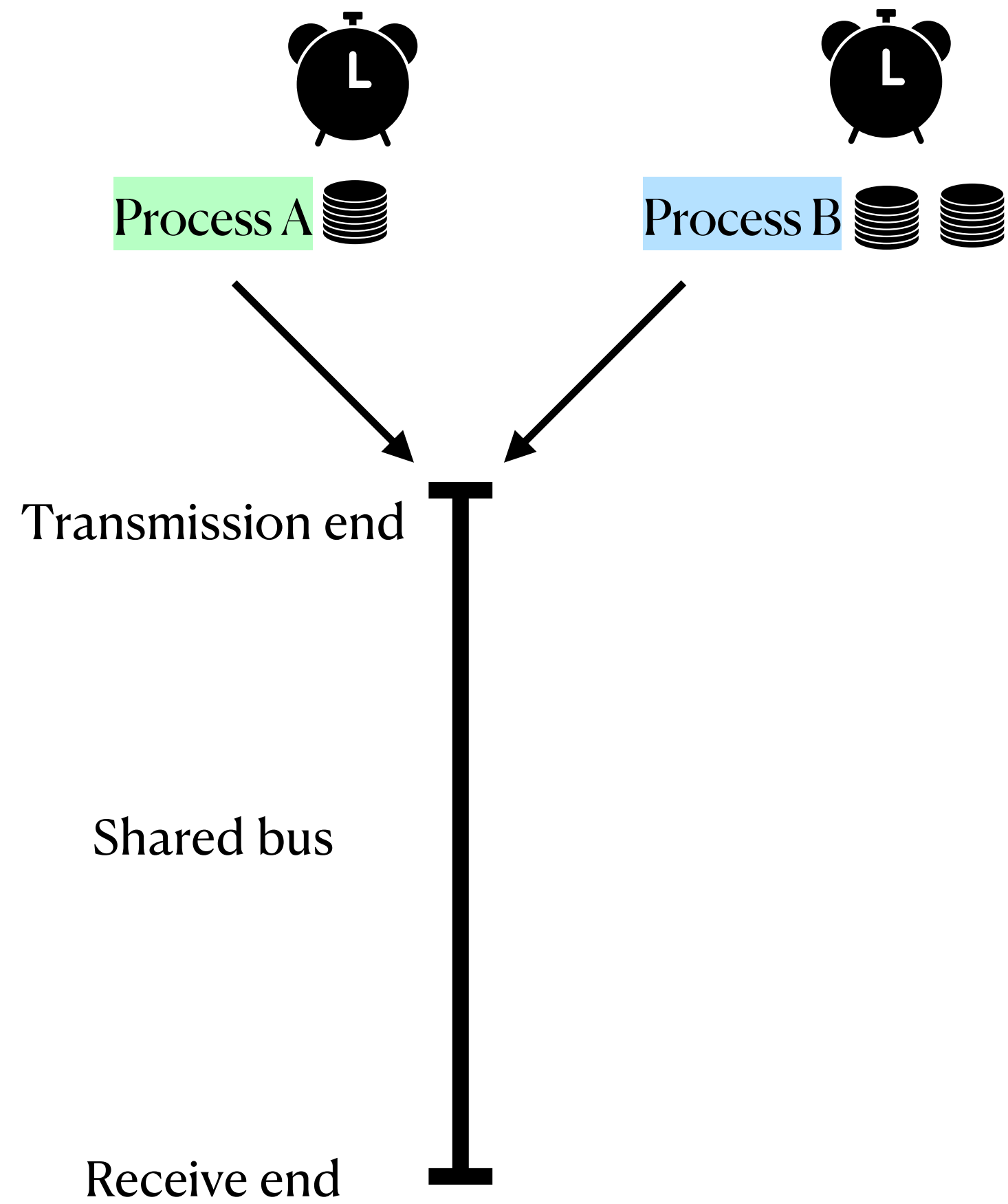
The Mutual Exclusion Example



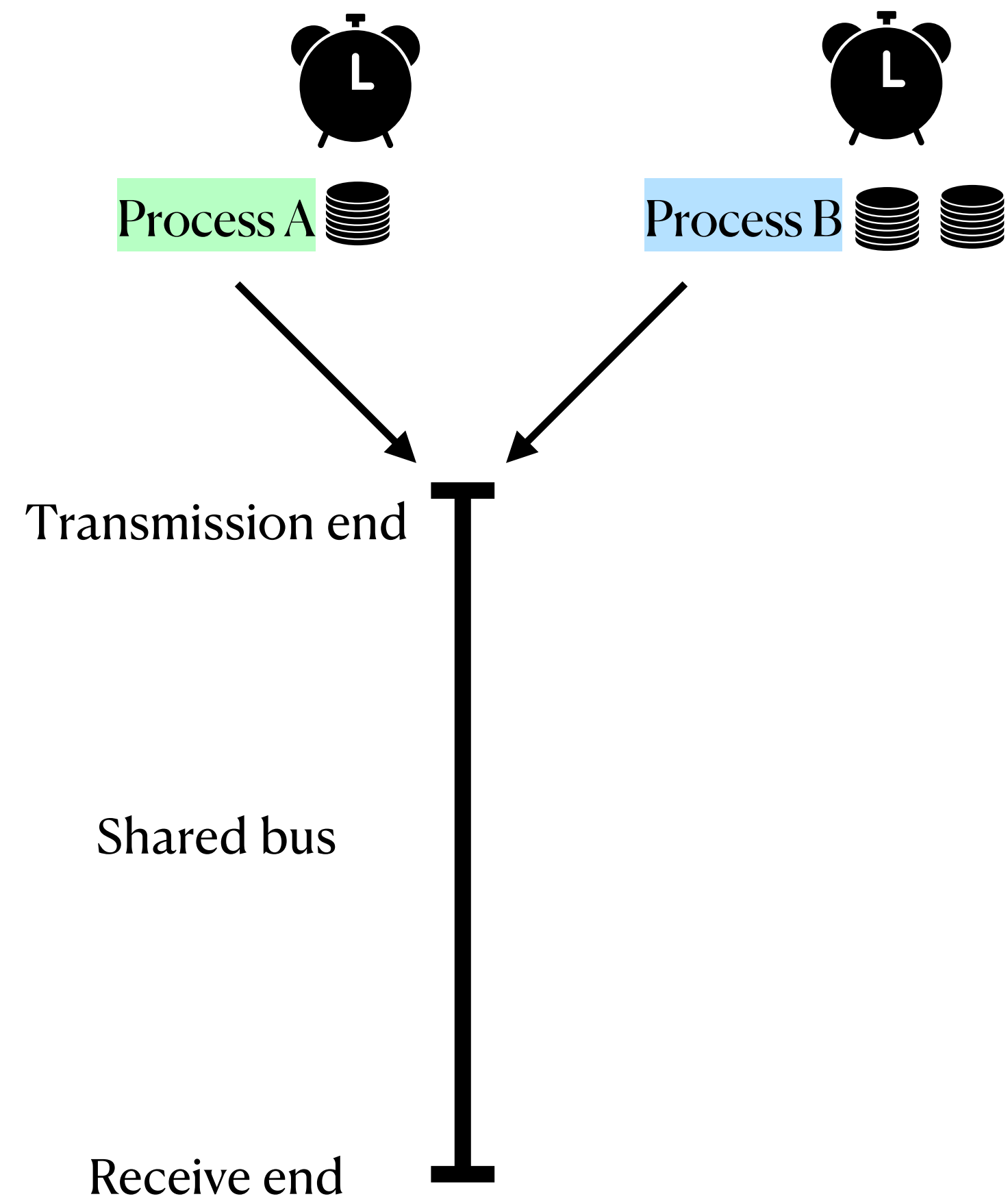
The Mutual Exclusion Example



The Mutual Exclusion Example

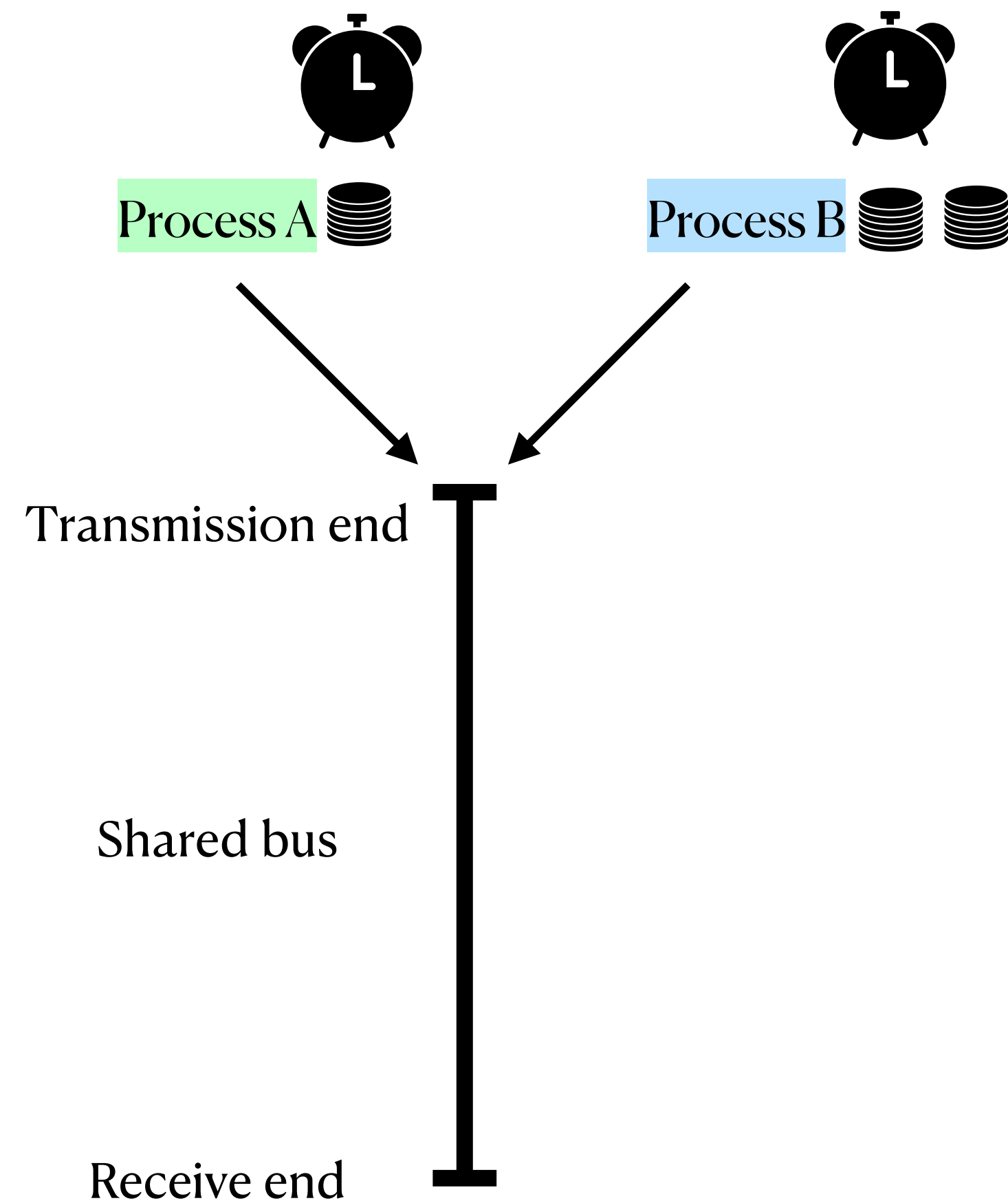


The Mutual Exclusion Example



Contracts need to ensure that the processes **do not write at the same time.**

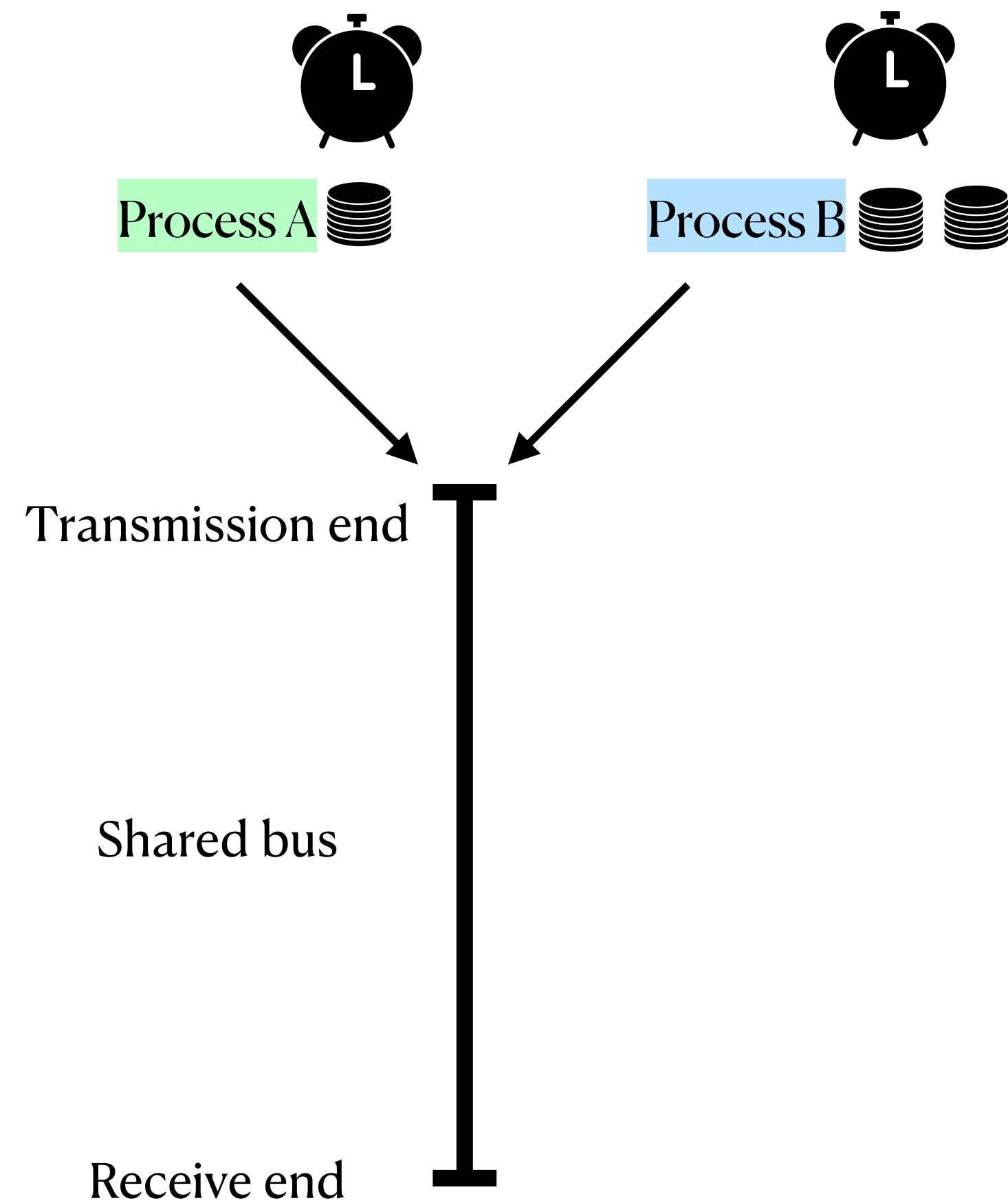
The Mutual Exclusion Example



Contracts need to ensure that the processes **do not write at the same time.**

The assume-guarantee contract for the Process A:
(Pr. B does not write at even time steps, do not write at odd time steps)

The Mutual Exclusion Example

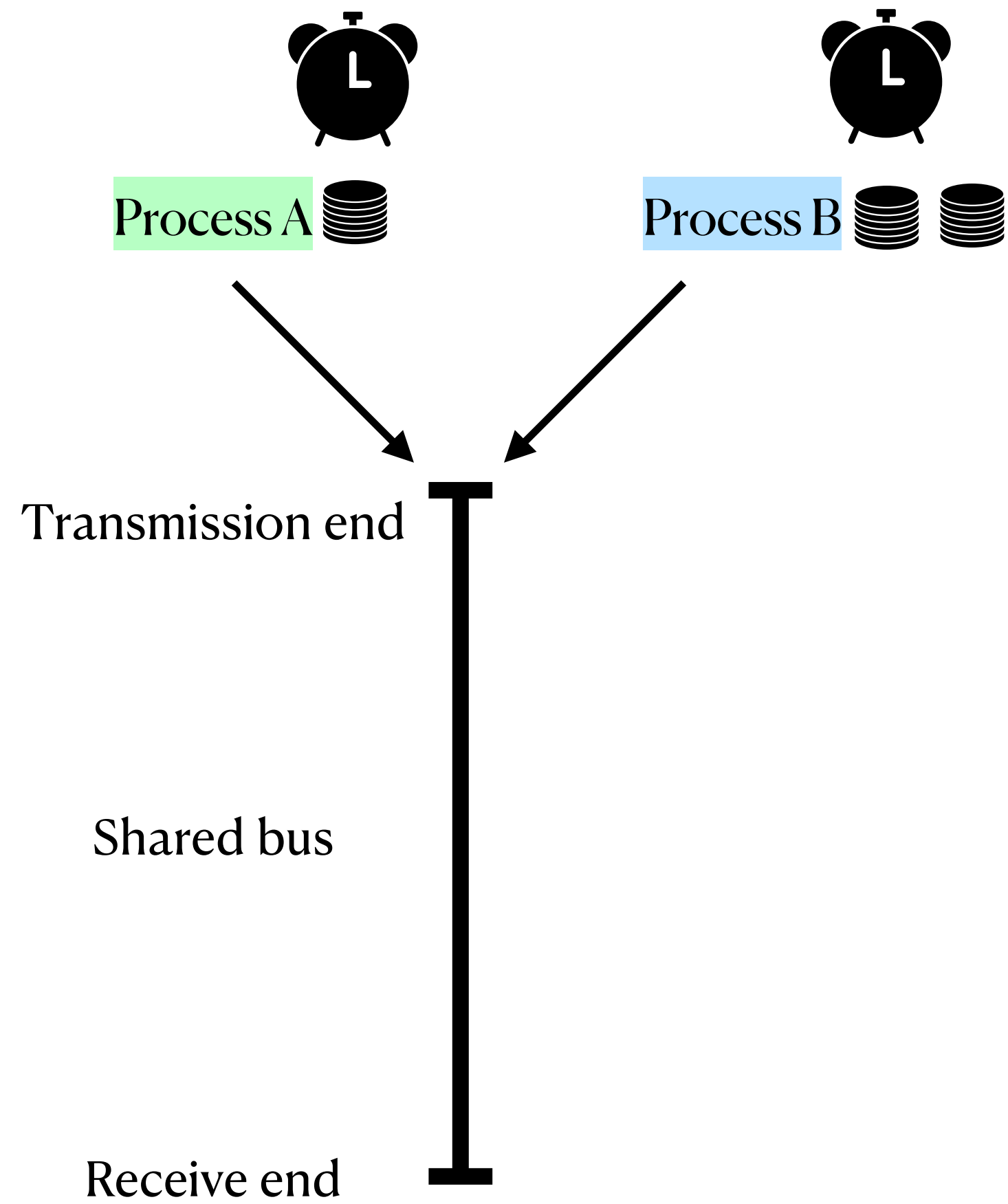


Contracts need to ensure that the processes **do not write at the same time.**

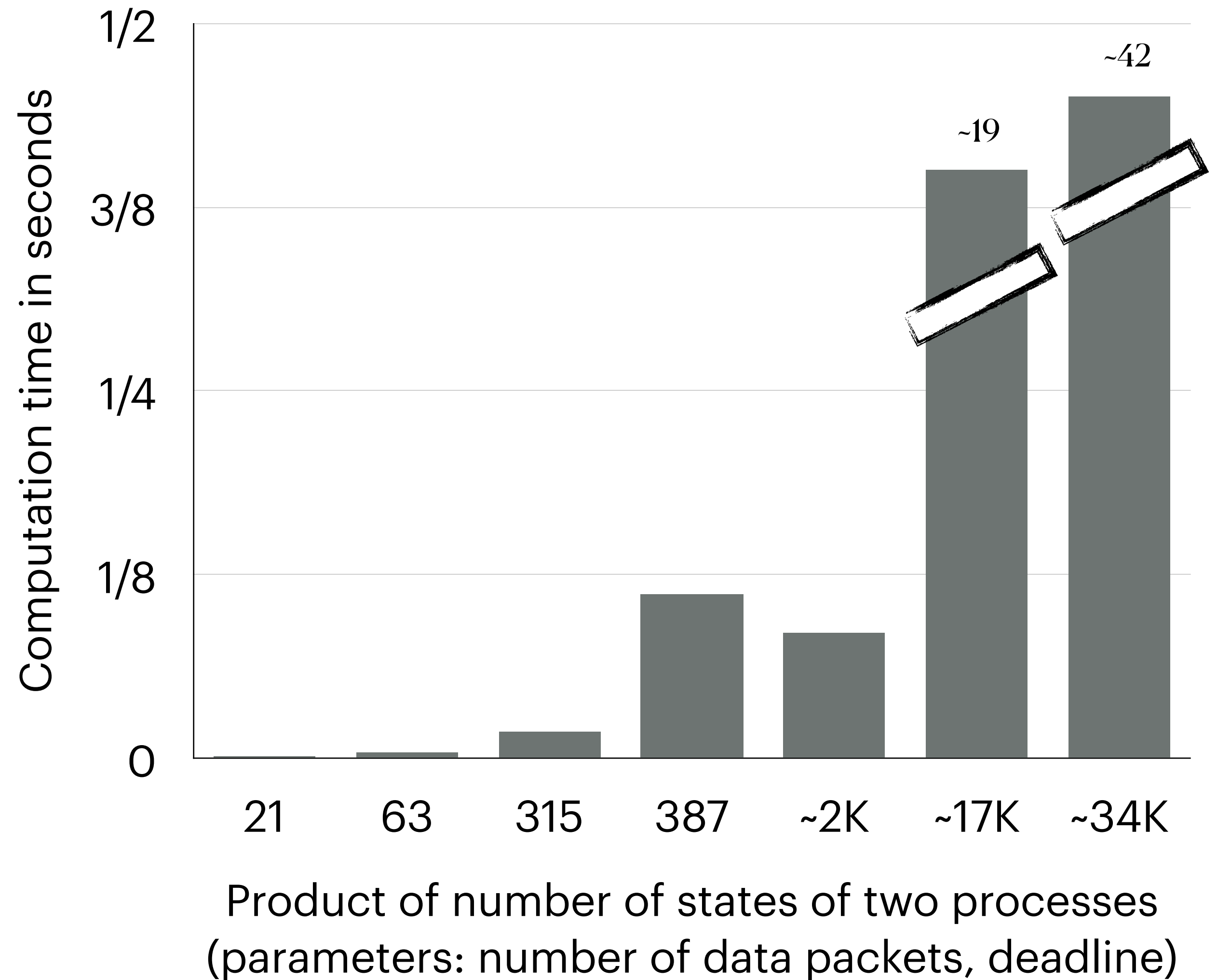
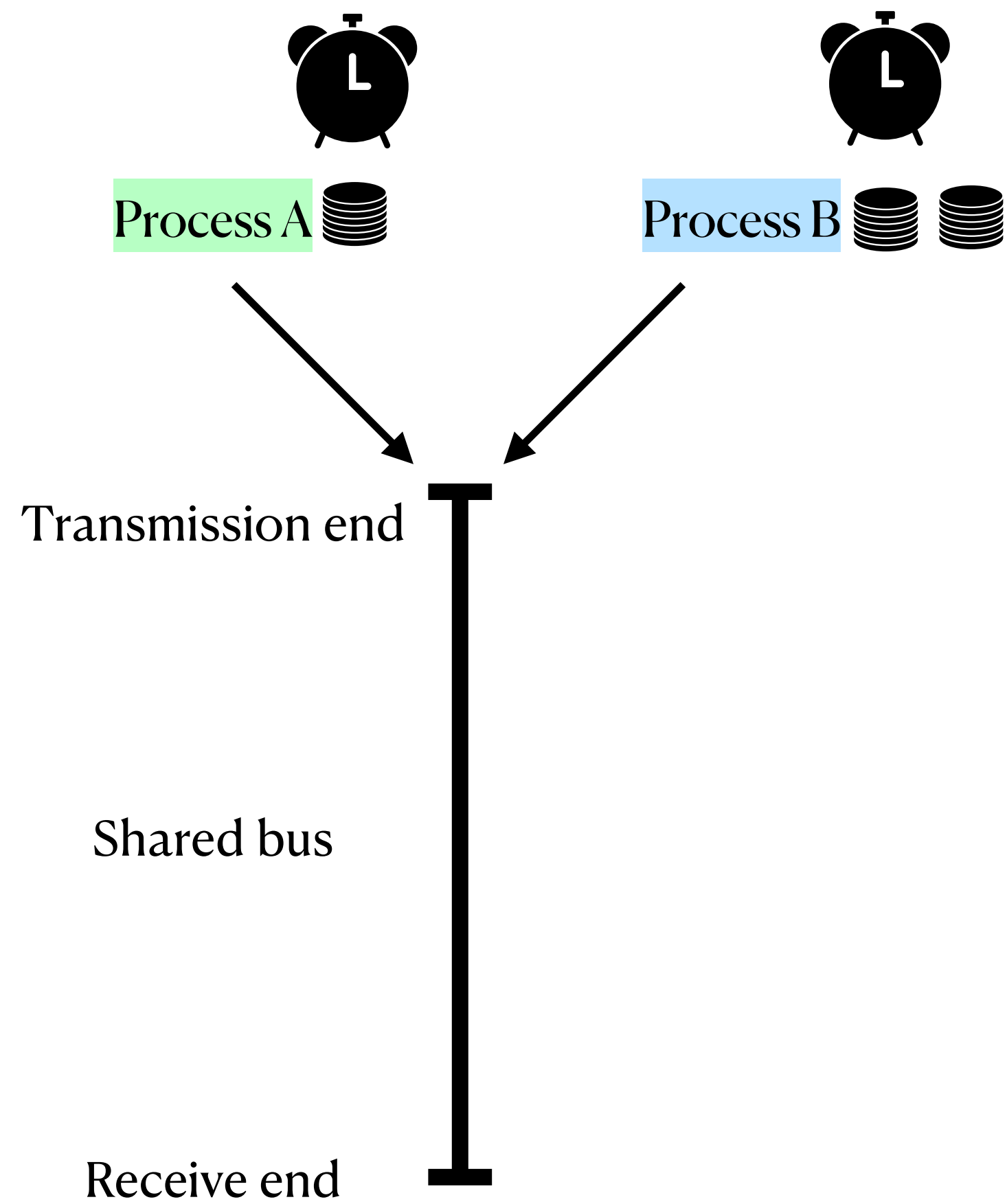
The assume-guarantee contract for the Process A:
(Pr. B does not write at even time steps, do not write at odd time steps)

The assume-guarantee contract for the Process B:
(Pr. A does not write at odd time steps, do not write at even time steps)

The Mutual Exclusion Example



The Mutual Exclusion Example



Summary

- Negotiation: a **modular**, **sound**, and **incomplete** approach for distributed synthesis.
- Input:
 - The systems' models + omega-regular **local** specifications.
 - Systems have partial view of each other.
- Output: **Assume-Guarantee Contracts**.
- Optimization heuristics for faster computation.
- Implemented in the tool called Agnes (<https://github.com/kmallik/Agnes>).

Summary

- Negotiation: a **modular**, **sound**, and **incomplete** approach for distributed synthesis.
- Input:
 - The systems' models + omega-regular **local** specifications.
 - Systems have partial view of each other.
- Output: **Assume-Guarantee Contracts**.
- Optimization heuristics for faster computation.
- Implemented in the tool called Agnes (<https://github.com/kmallik/Agnes>).

Thank you for listening!