Act for Your Duties but Maintain Your Rights

Shufang Zhu, Giuseppe De Giacomo

KR 2022

Sapienza University of Rome

Autonomous Agents

- Autonomy, one of the grand objectives AI
  - Autonomous system, react to environment changes

- Automated strategy synthesis [Pnueli & Rosner, 1989]
  - **Given**: a task expressed in a declarative specification
  - **Obtain**: a strategy (system model)

Agent Strategy

- Strategy

Agent Strategy

- Strategy
    - prescribes for **env action sequence**, makes **agn response**, the generated trace satisfies specification

Agent Strategy

- Strategy
  - prescribes for **env action sequence**, makes **agn response**, the generated trace satisfies specification
  - Deterministic: $\sigma : (2^{\mathcal{X}})^{+} \rightarrow 2^{\mathcal{Y}}$, returns a single agent action

Agent Strategy

- Strategy
  - prescribes for **env action sequence**, makes **agn response**, the generated trace satisfies specification
  - Deterministic: $\sigma : (2^{\mathcal{X}})^{+} \to 2^{\mathcal{Y}}$, returns a single agent action
  - Nondeterministic: $\Pi : (2^{\mathcal{X}})^{+} \to 2^{2^{\mathcal{Y}}}$, returns a set of agent actions to choose from nondeterministically

Agent Strategy

- Strategy
  - prescribes for **env action sequence**, makes **agn response**, the generated trace satisfies specification
  - Deterministic: $\sigma : (2^{\mathcal{X}})^+ \to 2^{\mathcal{Y}}$, returns a single agent action
  - Nondeterministic: $\Pi : (2^{\mathcal{X}})^+ \to 2^{2^{\mathcal{Y}}}$, returns a set of agent actions to choose from nondeterministically
- Every nondet. strategy Π captures a set of det. strategies σ

Duties and Rights

- Standard agents
  - Work in a *fixed* manner: actions only serve the assigned task
  - Highly limits the freedom of the agent

Duties and Rights

- Standard agents
  - Work in a *fixed* manner: actions only serve the assigned task
  - Highly limits the freedom of the agent

- Intelligent agents
  - Work in a *flexible* manner: actions only serve the assigned task + agent demands (if the agent decides to)
  - Allow more freedom of the agent

Duties and Rights

- Standard agents
  - Work in a *fixed* manner: actions only serve the assigned task
  - Highly limits the freedom of the agent

- Intelligent agents
  - Work in a *flexible* manner: actions only serve the assigned task + agent demands (if the agent decides to)
  - Allow more freedom of the agent

- Duties and Rights
  - Duties: assigned tasks, guaranteed to be accomplished
  - Rights: agent demands, accomplished on the agent's own

Act for Your Duties But Maintain Your Rights

- Duties and rights, LTL_f synthesis under environment specifications
  - Tasks in LTL_f formulas, environment in safety specifications
  - Definition, computational properties, and formally well-founded algorithms

- Further duties and rights, while agent in execution
  - Live synthesis in Formal Methods [Finkbeiner, Klein, & Metzger 2021]
  - Definition, computational properties, and formally well-founded algorithms

Synthesis with Duties and Rights

Given:

Environment specification *env*, agent duties φ_{d},

Obtain:

Agent strategy $\sigma : (2^{\mathcal{X}})^+ \to 2^{\mathcal{Y}}$, a function from past history of environment behaviors to agent actions

$$\forall \gamma \ \ \triangleright \mathit{env}, \mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}}$$

φ_{d} describes the mandatory goal/task when the environment behaves as its specification *env*

Synthesis with Duties and Rights

Given:

Environment specification *env*, agent duties φ_{d}, agent rights φ_{r}

Obtain:

Agent strategy $\sigma : (2^{\mathcal{X}})^{+} \to 2^{\mathcal{Y}}$, such that $\forall \gamma \ \triangleright \mathit{env}$,

- either $\mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}}$
- or $\mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ if the agent decides to achieve φ_{r} as well

Synthesis with Duties and Rights

Given:

Environment specification *env*, agent duties φ_{d},

Obtain:

Agent strategy $\sigma : (2^{\mathcal{X}})^+ \to 2^{\mathcal{Y}}$, such that $\forall \gamma \ \triangleright \mathit{env}$,

- either $\mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}}$
- or $\mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ if the agent decides to achieve φ_{r} as well

The agent decides **whether**

Synthesis with Duties and Rights

Given:

Environment specification *env*, agent duties φ_{d},

Obtain:

Agent strategy $\sigma : (2^{\mathcal{X}})^{+} \to 2^{\mathcal{Y}}$, such that $\forall \gamma \ \triangleright \mathit{env}$,

- either $\mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}}$
- or $\mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ if the agent decides to achieve φ_{r} as well

The agent decides **whether** & **when**

When: Strategy Enforcing Task wrt History

Environment specification *env*, agent task φ, history h

Agent strategy $\sigma : (2^{\mathcal{X}})^+ \rightarrow 2^{\mathcal{Y}}$ enforces φ, with respect to history h, denoted by $\sigma \ \triangleright_h \varphi$

- IF $\forall \gamma \ \triangleright$ *env* such that Play(σ, γ) has h as a prefix
- THEN Play$(\sigma, \gamma) \models \varphi$

Whether: Strategy Being Right-aware

Environment specification *env*, agent duties φ_{d}, agent rights φ_{r}

Agent strategy $\sigma : (2^{\mathcal{X}})^{+} \to 2^{\mathcal{Y}}$ enforcing φ_{d} is right-aware for φ_{r}, if $\forall \gamma \ \rhd \mathit{env}$

- $\mathsf{Play}(\sigma, \gamma) \models \varphi_{\mathsf{d}}$
- for every prefix h of $\mathsf{Play}(\sigma, \gamma)$, there exists a strategy σ_h that enforces $\varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ wrt h

Synthesis With Rights

Given:

Environment specification *env*, agent duties φ_{d}, agent rights φ_{r}

Obtain:

Agent strategy $\sigma : (2^{\mathcal{X}})^{+} \to 2^{\mathcal{Y}}$ enforcing φ_{d} that is right-aware of φ_{r}

Synthesis With Rights: Example

A cleaning robot working in a circular hallway, the charging station is close to the entrance.
Duty "cleaning room A" $\varphi_{\mathsf{d}} = \Diamond(\neg \mathit{Dust\_A} \wedge \mathit{RobotOut\_A})$
Rights "fully charging battery" $\varphi_{\mathsf{r}} = \Diamond(\mathit{BatteryFull})$

1. Take the direction that passes the charging station to room A and clean it. The remaining battery after enforcing φ_{d} still allows the robot to reach the charging station

2. Take the other direction to reach room A and clean it. The remaining battery after enforcing φ_{d} is not enough for the robot to reach the charging station

Synthesis With Rights: Example

A cleaning robot working in a circular hallway, the charging station is close to the entrance.
Duty "cleaning room A" $\varphi_{\mathsf{d}} = \Diamond(\neg \mathit{Dust\_A} \wedge \mathit{RobotOut\_A})$
Rights "fully charging battery" $\varphi_{\mathsf{r}} = \Diamond(\mathit{BatteryFull})$

1. Take the direction that passes the charging station to room A and clean it. The remaining battery after enforcing φ_{d} still allows the robot to reach the charging station ✓✓

2. Take the other direction to reach room A and clean it. The remaining battery after enforcing φ_{d} is not enough for the robot to reach the charging station ✗✗

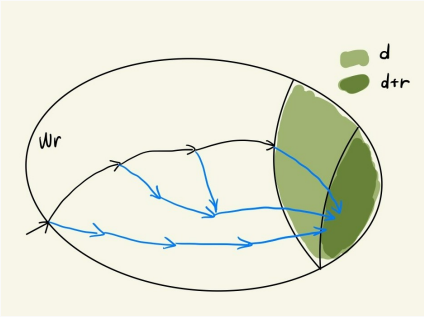
Synthesis Technique

Key idea

- **Winning region of reachability game**, maintaining the ability to reach the goal states

Solution

1. Compute W_r wrt env. spec. *env*, duties φ_d and rights φ_r (2exp time)
   - A_{env} as $\{\gamma \mid \gamma \triangleright env\}$ [De Giacomo et al. 2021]
   - $A_{env} \times A_{\varphi_d} \times A_{\varphi_r}$, reachability game
2. **A det. strategy σ_d pursing** φ_d (black arrows), which always acts within W_r
   - while achieving φ_d
   - having the capability of achieving also φ_r



Synthesis Technique

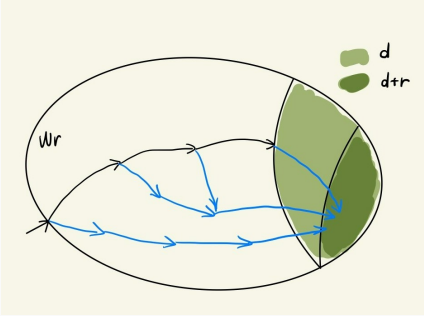
Key idea

- **Winning region of reachability game**, maintaining the ability to reach the goal states

Solution

1. Compute W_r wrt env. spec. *env*, duties φ_d and rights φ_r (2exp time)
   - A_{env} as $\{\gamma \mid \gamma \rhd env\}$ [De Giacomo et al. 2021]
   - $A_{env} \times A_{\varphi_d} \times A_{\varphi_r}$, reachability game
2. **A det. strategy σ_d pursing** φ_d (black arrows), which always acts within W_r
   - while achieving φ_d
   - having the capability of achieving also φ_r

Computationally, linear time

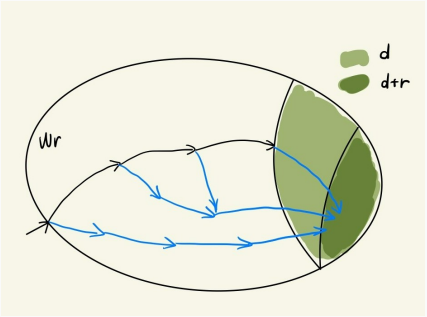


Synthesis Technique

Achieve rights in execution

- **a nondet. strategy $\Pi_{d \wedge r}$ pursing** $\varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$

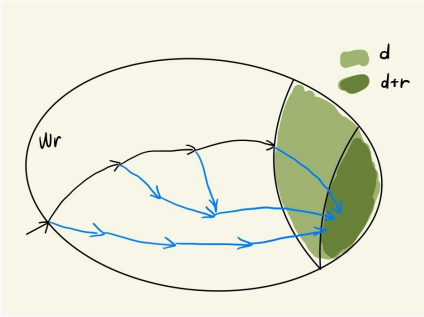
Computationally, linear time



Synthesis Technique

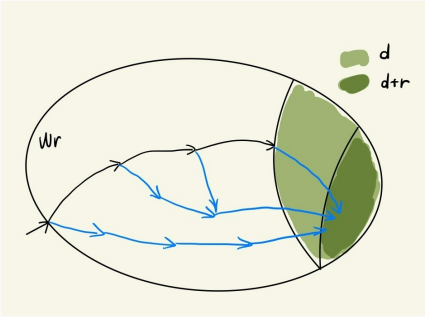
How does the ultimate strategy work?

- **Follow σ_d to achieve duties** φ_{d} (light green zone)
- Choose one strategy from $\Pi_{d \wedge r}$ to achieve also rights φ_{r} (dark green zone), **whenever the agent decides to do so**



Synthesis Technique

Computationally, only requires a constant overhead!



Further Duties and Rights While Executing

- Standard agents
  - Only one task (duty) to accomplish, terminates after finishing
- Intelligent agents
  - Take further tasks (duties and rights) while in execution
  - E.g., a new room to clean while the robot is cleaning the rooms it got assigned at the beginning
- Handling further duties and rights while in execution

Ongoing Duties and New Duties

- Agent and environment start acting when the first duty arrives

Ongoing Duties and New Duties

- Agent and environment start acting when the first duty arrives
- New duties arrive after a history h

Ongoing Duties and New Duties

- Agent and environment start acting when the first duty arrives
- New duties arrive after a history h
- Agent enforces **ongoing duties** with respect to history h

Ongoing Duties and New Duties

- Agent and environment start acting when the first duty arrives
- New duties arrive after a history h
- Agent enforces **ongoing duties** with respect to history h
- Agent enforces **new duties** after history h

Synthesis with Further Duties and Rights

Given: Environment specification *env*, agent duties φ_{d}, rights φ_{r}, further duties φ_{fd}, further rights φ_{fr}, h

Synthesis with Further Duties and Rights

Given: Environment specification *env*, agent duties φ_{d}, rights φ_{r}, further duties φ_{fd}, further rights φ_{fr}, h

Obtain: Agent strategy $\sigma : (2^{\mathcal{X}})^{+} \to 2^{\mathcal{Y}}$, such that

- σ_d enforces φ_{d} and is right-aware for φ_{r} wrt h
- σ_d enforces φ_{fd} and is right-aware for φ_{fr} after h

Synthesis with Further Duties and Rights

Given: Environment specification *env*, agent duties φ_{d}, rights φ_{r}, further duties φ_{fd}, further rights φ_{fr}, h

Obtain: Agent strategy $\sigma : (2^{\mathcal{X}})^{+} \to 2^{\mathcal{Y}}$, such that

- σ_d enforces φ_{d} and is right-aware for φ_{r} wrt h
- σ_d enforces φ_{fd} and is right-aware for φ_{fr} after h

The agent decides **whether** achieving rights (φ_{r}, φ_{fr}) & **when**

Synthesis With Further Duties: Example

Duty "cleaning room A" $\varphi_{\mathsf{d}} = \Diamond(\neg \mathit{Dust\_A} \wedge \mathit{RobotOut\_A})$
Rights "fully charging battery" $\varphi_{\mathsf{r}} = \Diamond(\mathit{BatteryFull})$

New duty "cleaning room B" $\varphi_{\mathsf{fd}} = \Diamond(\neg \mathit{Dust\_B} \wedge \mathit{RobotOut\_B})$

- Take the direction that passes the charging station to reach room A and clean it. Then go to room B and clean it. The remaining battery after cleaning is not enough for the robot to reach the charging station

Synthesis With Further Duties: Example

Duty "cleaning room A" $\varphi_{\mathsf{d}} = \Diamond(\neg \mathit{Dust\_A} \wedge \mathit{RobotOut\_A})$
Rights "fully charging battery" $\varphi_{\mathsf{r}} = \Diamond(\mathit{BatteryFull})$

New duty "cleaning room B" $\varphi_{\mathsf{fd}} = \Diamond(\neg \mathit{Dust\_B} \wedge \mathit{RobotOut\_B})$

- Take the direction that passes the charging station to reach room A and clean it. Then go to room B and clean it. The remaining battery after cleaning is not enough for the robot to reach the charging station ✗✗

Synthesis Technique

Given: Environment specification *env*, agent duties φ_{d}, rights φ_{r}, further duties φ_{fd}, further rights φ_{fr}, h

Recall: winning region W_r, maintaining the ability to achieve φ_{r}

1. Compute $W_{r \wedge \mathsf{fr}}$, maintaining the ability to achieve φ_{r} wrt h, and φ_{fr} after h (2exp time)

Synthesis Technique

Given: Environment specification *env*, agent duties φ_{d}, rights φ_{r}, further duties φ_{fd}, further rights φ_{fr}, h

Recall: winning region W_r, maintaining the ability to achieve φ_{r}

1. Compute $W_{r \wedge \mathsf{fr}}$, maintaining the ability to achieve φ_{r} wrt h, and φ_{fr} after h (2exp time)
   - Synchronize the environment with the new duties and new rights!
2. **A det. strategy σ_d pursing φ_{d} and φ_{fd}**, which always acts within $W_{r \wedge \mathsf{fr}}$
   - while achieving φ_{d} and φ_{fd}
   - having the capability of achieving also φ_{r} and φ_{fr}

Synthesis Technique

Given: Environment specification *env*, agent duties φ_{d}, rights φ_{r}, further duties φ_{fd}, further rights φ_{fr}, h

Recall: winning region W_r, maintaining the ability to achieve φ_{r}

1. Compute $W_{r \wedge \mathsf{fr}}$, maintaining the ability to achieve φ_{r} wrt h, and φ_{fr} after h (2exp time)
   - Synchronize the environment with the new duties and new rights!
2. **A det. strategy σ_d pursing φ_{d} and φ_{fd}**, which always acts within $W_{r \wedge \mathsf{fr}}$
   - while achieving φ_{d} and φ_{fd}
   - having the capability of achieving also φ_{r} and φ_{fr}

Computationally, linear time

Synthesis Technique

Achieve rights in execution

- **a nondet. strategy** $\Pi_{d \wedge fd \wedge r}$ **pursing** $\varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ **wrt** h**, and** φ_{fd} **after** h
- **a nondet. strategy** $\Pi_{d \wedge fd \wedge fr}$ **pursing** φ_{d} **wrt** h**, and** $\varphi_{\mathsf{fd}} \wedge \varphi_{\mathsf{fr}}$ **after** h
- **a nondet. strategy** $\Pi_{d \wedge fd \wedge r \wedge fr}$ **pursing** $\varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ **wrt** h**, and** $\varphi_{\mathsf{fd}} \wedge \varphi_{\mathsf{fr}}$ **after** h

Synthesis Technique

Achieve rights in execution

- **a nondet. strategy** $\Pi_{d \wedge fd \wedge r}$ **pursing** $\varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ **wrt** h**, and** φ_{fd} **after** h
- **a nondet. strategy** $\Pi_{d \wedge fd \wedge fr}$ **pursing** φ_{d} **wrt** h**, and** $\varphi_{\mathsf{fd}} \wedge \varphi_{\mathsf{fr}}$ **after** h
- **a nondet. strategy** $\Pi_{d \wedge fd \wedge r \wedge fr}$ **pursing** $\varphi_{\mathsf{d}} \wedge \varphi_{\mathsf{r}}$ **wrt** h**, and** $\varphi_{\mathsf{fd}} \wedge \varphi_{\mathsf{fr}}$ **after** h

Arbitrary subsets of rights: maintain the information and compute on demands

Conclusions and Future Directions

- Agents handling duties and rights, mandatory tasks and optional tasks
- Further duties and rights arrive in execution
- Multiple duties and multiple rights

- More expressive environment specifications
- More expressive task categories, concepts from Deontic Logic