A Compositional Framework for Preference-aware Agents

Tobias Kappé[1,2] Farhad Arbab[2,1] Carolyn Talcott[3]

[1]Leiden Institute of Advanced Computer Science

[2]Centrum Wiskunde en Informatica

[3]SRI International

V2CPS2016; June 4, 2016

This work was partially supported by ONR grant N00014–15–1–2202.

Problem

A Cyber-Physical System (CPS) consists of components that...

- ...carry out *physical* tasks
- ...perform *cyber* computations
- ...coordinate *interaction* of components

Problem

A Cyber-Physical System (CPS) consists of components that. . .

- . . . carry out *physical* tasks
- . . . perform *cyber* computations
- . . . coordinate *interaction* of components

Ideally, we want to design a CPS. . .

- . . . *compositionally*
- . . . in a *uniform* fashion
- . . . to be *robust*
- . . . amenable to *verification*
- . . . that is easy to *extend*

Running example

Suppose we design an agent that…

- …should patrol between two designated points
- …may try to avoid obstacles on its path
- …has a finite amount of energy
- …can recharge at some location

Running example

Suppose we design an agent that...

- ...should patrol between two designated points
- ...may try to avoid obstacles on its path
- ...has a finite amount of energy
- ...can recharge at some location

Different concerns, different components:

- moving towards the next waypoint
- staying on track as much as possible
- not running out of energy

Robustness

A component (e.g. *movement to waypoint*) has a set of possible actions.

- Some actions have higher preference than others.
  - move *towards* or *away* from the waypoint, or *remain*.
- Components want the *best available* action.
  - we want to move towards the waypoint most of all.
- More alternatives $\Rightarrow$ more robustness!
  - if we cannot move towards the waypoint, we want to remain.

Robustness

A component (e.g. *movement to waypoint*) has a set of possible actions.

- Some actions have higher preference than others.
  - move *towards* or *away* from the waypoint, or *remain*.
- Components want the *best available* action.
  - we want to move towards the waypoint most of all.
- More alternatives $\Rightarrow$ more robustness!
  - if we cannot move towards the waypoint, we want to remain.

With concurrent components:

- Some actions may be incompatible (e.g. *move* and *turn*).
- Composable actions need a *composed preference*.

Preferences

How do we attach preferences to actions?

A c-semiring [Bistarelli, 2004] is a structure for preferences.

- Preferences are contained in the *carrier* set E.
- Values $\mathbf{0}, \mathbf{1} \in E$ are the minimal, respectively maximal preferences.
- The operator $\bigoplus : \mathcal{P}(E) \to E$ models *choice* between preferences.
- The binary operator $\otimes$ models *composition* of preferences.

Preferences

As an example c-semiring, consider the *probabilistic semiring*:

$$\mathbb{P} = \langle [0,1], \sup, \cdot, 0, 1 \rangle$$

- $\sup$ is the supremum within $[0,1]$, with $\sup \emptyset = 0$
- $\cdot$ is multiplication of real numbers

There is also the *weighted semiring*:

$$\mathbb{W} = \langle \mathbb{R}_{\geq 0} \cup \{\infty\}, \inf, +, \infty, 0 \rangle$$

- $\inf$ is the infimum of real numbers
- $+$ is addition of real numbers

Preferences

A c-semiring E induces partial order $\leq_E$, by $e \leq_E e' \stackrel{def.}{\iff} e \oplus e' = e'$

- $\mathbb{P}$: $e \leq_{\mathbb{P}} e' \iff \sup\{e, e'\} = e' \iff e \leq e'$. *Better odds* are preferred.
- $\mathbb{W}$: $e \leq_{\mathbb{W}} e' \iff \inf\{e, e'\} = e' \iff e \geq e'$. *Lower weights* are preferred.

Preferences

A c-semiring E induces partial order $\leq_E$, by $e \leq_E e' \overset{def.}{\iff} e \oplus e' = e'$

- $\mathbb{P}$: $e \leq_{\mathbb{P}} e' \iff \sup\{e, e'\} = e' \iff e \leq e'$. *Better odds* are preferred.
- $\mathbb{W}$: $e \leq_{\mathbb{W}} e' \iff \inf\{e, e'\} = e' \iff e \geq e'$. *Lower weights* are preferred.

If $E' \subseteq E$ has a unique $\leq_E$-maximal value, it is $\bigoplus E'$.

In any case, $\bigoplus E'$ is the *least upper bound* of E'.

Preferences

We can compose c-semirings...

- ...independently: $\odot$ ("smash product")
- ...lexicographically[1]: $\triangleright$

[1] Subject to some technical details [Gadducci et al., 2013].

Preferences

We can compose c-semirings...

- ...independently: $\odot$ ("smash product")
- ...lexicographically[1]: $\rhd$

Examples:

- The order of $\mathbb{P} \odot \mathbb{P}$ is the product order; the carrier is

$$\{(x, y) \in [0, 1]^2 : x \cdot y > 0\} \cup \{\langle 0, 0 \rangle\}$$

- The order of $\mathbb{P} \rhd \mathbb{P}$ is the lexicographic order; the carrier is

$$\{(x, y) \in [0.1]^2 : x > 0\} \cup \{\langle 0, 0 \rangle\}$$

[1] Subject to some technical details [Gadducci et al., 2013].

Soft Constraint Automata

Soft Constraint Automata [Arbab and Santini, 2012] used as components.

An SCA over a c-semiring E is an LTS with labels from $\mathcal{A} \times E$.[2]

Transitions $q \xrightarrow{\alpha,\ e} q'$ with $e = \mathbf{0}$ are called *infeasible*.

[2] $\mathcal{A}$ is a set representing possible actions; refer to the paper for details.

Composition

Let A_1 and A_2 be SCAs over E with...

- ...state spaces Q_1 and Q_2
- ...transition relations $\rightarrow_1$ and $\rightarrow_2$

respectively.

Their composition, $A_1 \otimes A_2$, is the SCA over E with...

- ...state space $Q_1 \times Q_2$
- ...the transition relation generated by:

$$\frac{q_1 \xrightarrow{\alpha_1,\ e_1}_1 q_1' \qquad q_2 \xrightarrow{\alpha_2,\ e_2}_2 q_2' \qquad \alpha_1, \alpha_2 \text{ compatible}}{\langle q_1, q_2 \rangle \xrightarrow{\alpha_1 : \alpha_2,\ e_1 \otimes e_2} \langle q_1', q_2' \rangle}$$

Example: *move* and *turn* are incompatible, *signal* could be compatible with either.

Intermezzo: preferences and composition

actions with maximal preference in the composition

$$\neq$$

compositions of components' actions with maximal preference

This goes two ways:

- Actions with maximal preference in the composition may be compositions of components' actions with non-maximal preference (*compromise*)
  - *move* and *turn* have highest preference, but are incompatible.
- Not all compositions of components' actions are actions that have maximal preference (*harmonize*)
  - *move* and *turn* may compose less preferably than *signal* and *turn*.

In the end, *what is best for a single component may not be best for the composition*.

Composition

We can move SCAs between c-semirings smoothly with *homomorphisms*.

If A is an SCA over E, then $h(A)$ is an SCA over $h(E)$.

Simply transform preferences in A by h to obtain $h(A)$.

Composition

We can define new composition operators now.

Let A_1, A_2 be SCAs over E_1 and E_2 respectively.

$$A_1 \odot A_2 \stackrel{def.}{=} h_1(A_1) \otimes h_2(A_2) \qquad (h_i : E_i \to E_1 \odot E_2)$$

$$A_1 \triangleright A_2 \stackrel{def.}{=} g_1(A_1) \otimes g_2(A_2) \qquad (g_i : E_i \to E_1 \triangleright E_2)$$

Composition

A matter of *which concerns are at play*:

- A_1 and A_2 model the same concern $\Rightarrow \otimes$
  - e.g. both are concerned with energy consumption
- A_1 and A_2 model equally important concerns: $\Rightarrow \odot$
  - e.g. energy consumption and movement towards the waypoint
- A_1's concern outweighs A_2's: $\Rightarrow \triangleright$[3]
  - e.g. movement towards the waypoint and staying on track

[3] Here, A_2 acts as a tie-breaker of sorts.

Composition

The operators allow more techniques:

- *Veto/downgrade* an action by $\otimes$-composition.
  - if energy is low, energy component vetoes moves away from charging station
- *Suppress* either concern of a $\odot$-composite by using $\otimes$.
  - if energy is low, the preferences of the energy component are leading

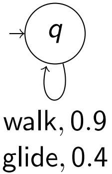
Examples

Suppose our patrol agent that can move by walking or hanggliding (downhill only).

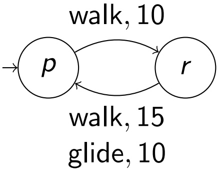
The actions *walk* and *glide* are incompatible.

Recall: in $\mathbb{P}$, a *higher* value is better, while in $\mathbb{W}$, a *lower* value is better.

A_1 (over $\mathbb{P}$):



A_2 (over $\mathbb{W}$):



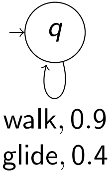
Examples

Suppose our patrol agent that can move by walking or hanggliding (downhill only).

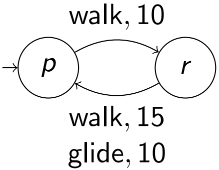
The actions *walk* and *glide* are incompatible.

Recall: in $\mathbb{P}$, a *higher* value is better, while in $\mathbb{W}$, a *lower* value is better.

A_1 (over $\mathbb{P}$):



A_2 (over $\mathbb{W}$):



The SCA A_1 models modes of movement and their (cyber) probability of success, A_2 models actual (physical) movement and its cost in terms of energy.

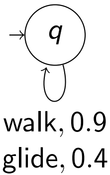
Examples

Suppose our patrol agent that can move by walking or hanggliding (downhill only).

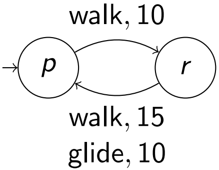
The actions *walk* and *glide* are incompatible.

Recall: in $\mathbb{P}$, a *higher* value is better, while in $\mathbb{W}$, a *lower* value is better.

A_1 (over $\mathbb{P}$):



A_2 (over $\mathbb{W}$):



In $A_1 \odot A_2$, the agent avoids unnecessary risk; from $\langle q, r \rangle$ both $walk : walk$ and $glide : glide$ have maximal preference: $\langle 0.9, 15 \rangle$ and $\langle 0.4, 10 \rangle$ are unordered in $\mathbb{P} \odot \mathbb{W}$.

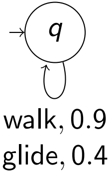
Examples

Suppose our patrol agent that can move by walking or hanggliding (downhill only).

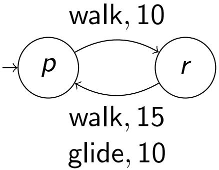
The actions *walk* and *glide* are incompatible.

Recall: in $\mathbb{P}$, a *higher* value is better, while in $\mathbb{W}$, a *lower* value is better.

A_1 (over $\mathbb{P}$):



A_2 (over $\mathbb{W}$):



In $A_1 \triangleright A_2$ the agent tries to maximize probability of success first. In $\langle q, r \rangle$ only *walk* : *walk* has maximal preference: $\langle 0.4, 10 \rangle$ is dominated by $\langle 0.9, 15 \rangle$ in $\mathbb{P} \triangleright \mathbb{W}$.

Conclusion

Soft Constraint Automata...

- ...provide robustness against
  - internal (other components) contexts
  - external (environmental) circumstances
- ...are compositional, with an easily extensible set of composition operators.
- ...are uniform: cyber, physical and coordination components in one format.

Further work

- Our actions (and their preferences) are generated by Soft Constraint Satisfaction Problems [Bistarelli et al., 1995]. Our simulator contains a rudimentary SCSP-solver; improvements to this solver could be useful.
- Integrate with *Soft Agent*s [Talcott et al., 2015].
- Most importantly: *model checking*. May be tough to do compositionally, due to compromise and harmonization. Interplay with compositional operators will have a role, too.

Bonus example: harmonization

Let Σ be a set. We define the *privilege semiring* $\mathbb{L}_\Sigma$ as the c-semiring

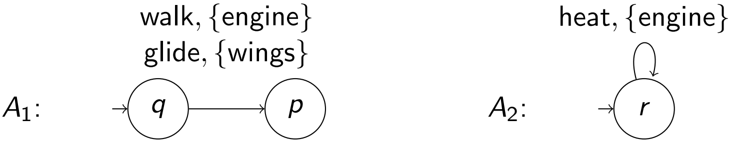
$$\left\langle \mathcal{P}(\Sigma), \bigcap, \cup, \Sigma, \emptyset \right\rangle$$

Note that in this c-semiring, $A \leq B$ if and only if $B \subseteq A$.

This c-semiring encodes the *principle of least privilege*: an action α is preferred over another action β if the privileges for α are a strict subset of those for β.

Bonus example: harmonization

Consider the following SCAs A_1 and A_2, over the privilege semiring $\mathbb{L}_\Sigma$ for $\Sigma = \{\text{engine}, \text{wings}\}$. The action *heat* composes with *walk* and *glide*.



In $A_1 \otimes A_2$, the action *walk* : *heat* is more preferable than the action *glide* : *heat*, for its preference is $\{\text{engine}\}$ rather than $\{\text{wings}, \text{engine}\}$.