Modular Basic Action Theories

Yilan Gu
Dept. of Computer Science
University of Toronto
Toronto, ON, Canada

Mikhail Soutchanski
Dept. of Computer Science
Ryerson University
Toronto, ON, Canada
Outline

- Motivation
- Action Hierarchy
- Modular Basic Action Theories (BATs)
- The Correctness
Motivation
Action Hierarchy

- Acyclic
- Antisymmetric
- Multiple inheritance

A1

specialization(A2, A1)

isA(A5, A1)
A Cooking Example

- prepareMeal
  - reheat
    - microwave
  - lowOilCook
    - stew
    - steam
    - ovenCook
    - boil
  - oilyCook
    - pressureCook
    - fry
    - stir
    - deepFry
  - reheat
  - microwave
  - lowOilCook
  - oilyCook
  - pressureCook
  - deepFry
Examples of Action Hierarchy Axioms

- **Examples of direct specializations:**
  - specialization(reheat(x), prepareMeal(x)).
  - specialization(cook(x), prepareMeal(x)).
  - specialization(microwave(x), reheat(x)).
  - specialization(lowOilCook(x), reheat(x)).
  - specialization(oilyCook(x), reheat(x)).
  - specialization(lowOilCook(x), cook(x)).
  - specialization(oilyCook(x), cook(x)).
  - specialization(stew(x), lowOilCook(x)).
  - specialization(deepFry(x), fry(x)).

- **Examples of isA:**
  - isA(cook(x), cook(x)).
  - isA(deepFry(x), fry(x)).
  - isA(deepFry(x), prepareMeal(x)).
Modular BAT Representation

**Precondition Axioms:**

\[ \text{Poss}(a,s) \equiv \exists x ( \text{isA}(a, \text{reheat}(x)) \land \text{food}(x) \land \text{cooked}(x,s) ) \lor \]
\[ \exists x ( \text{isA}(a, \text{cook}(x)) \land \text{food}(x) \land \lnot \text{cooked}(x,s) ) \lor \]
\[ \exists x ( a = \text{prepareMeal}(x) \land \text{food}(x) ). \]

**Successor State axioms:**

\[ \text{cooked}(x,\text{do}(a,s)) \equiv \text{isA}(a, \text{cook}(x)) \lor \]
\[ \text{cooked}(x,s). \]

\[ \text{mealReady}(x,\text{do}(a,s)) \equiv \text{isA}(a, \text{prepareMeal}(x)) \lor \]
\[ \text{mealReady}(x,s). \]
Comparison: Reiter’s BAT Representation

**Precondition Axioms:**
Poss(deepFry(x),s) ≡ food(x). Poss(fry(x),s) ≡ food(x). … …
Poss(prepareMeal(x),s) ≡ food(x). Poss(reheat(x),s) ≡ food(x) ∧ cooked(x).
Poss(reheat(x),s) ≡ food(x) ∧ ¬ cooked(x).

**Successor State axioms:**
cooked(x,do(a,s)) ≡ a = cook(x) ∨ a = lowOilCook(x) ∨ a = oilyOilCook(x) ∨
a = steam(x) ∨ a = boil(x) ∨ a = stew(x) ∨ a = broil(x) ∨ a = bake(x) ∨
a = ovenCook(x) ∨ a = roast(x) ∨ a = pressureCook(x) ∨ a = fry(x) ∨
a = deepFry(x) ∨ a = stir(x) ∨ cooked(x,s).

mealReady(x,do(a,s)) ≡ a = prepareMeal(x) ∨ a = reheat(x) ∨ a = cook(x) ∨
a = microwave(x) ∨ a = lowOilCook(x) ∨ a = oilyOilCook(x) ∨
a = steam(x) ∨ a = boil(x) ∨ a = stew(x) ∨ a = broil(x) ∨ a = bake(x) ∨
a = ovenCook(x) ∨ a = roast(x) ∨ a = pressureCook(x) ∨ a = fry(x) ∨
a = deepFry(x) ∨ a = stir(x) ∨ mealReady(x,s).
Correctness of the New BATs

A modular BAT $D^H = D_0 \cup D^H_{ap} \cup D^H_{ss} \cup D_{una} \cup \Sigma \cup H$

1. $D_0$ – the (usual) initial theory
2. $D^H_{ap}$ – the modular precondition axioms
3. $D^H_{ss}$ – the modular successor state axioms
4. $D_{una}$ – the (usual) unique name axioms for actions
5. $\Sigma$ – the (usual) foundational axioms
6. $H$ – the specialization axioms and the definition of isA

**Theorem:** For any modular BAT $D^H$ there exists an equivalent $D$ of Reiter’s BAT format, where equivalence means that for any FO regressable sentence $W$, $D^H \models W$ iff $D \models W$.

Although the formal definition of isA is second-order, the reasoning in $D^H$ can be reduced to a FOL reasoning only.

A regression theorem similar to Reiter’s regression theorem is proved.
The End

Thank you!