Warsaw University of Technology's
Faculty of Mathematics and Information Science

Knowledge Representation and Reasoning

Project number 2:
Deterministic Action With Cost
Supervisor: Dr Anna Radzikowska

Created By
Rishabh Jain, Rahul Tomer, Kuldeep Shankar,
Alaa Abboushi, Haran Dev Murugan,
Bui Tuan Anh.

Contents

1 Introduction

A dynamic system (DS) is viewed as

- a collection of objects, together with their properties, and
- a collection of actions which, while performed, change properties of objects (in consequence, the state of the world).

Let C2 be a class of dynamic systems satisfying the following assumptions:

1. Inertia law
2. Complete information about all actions and fluent.
3. Only Determinism
4. Only sequential actions are allowed.
5. Characterizations of actions:
   - Precondition represented by set of literals(a fluent or its negation);if a precondition does not hold, the action is executed but with empty effect
   - Postcondition (effect of an action) represented by a set of literals.
   - Cost $k \in N$ of an action, actions with empty effects cost 0. Each action has a fixed cost, if it leads to non-empty effects.
6. Effects of an action depends on the state where the action starts.
7. All actions are performed in all states.
8. Partial description of any state of the system are allowed.
9. No constraints are defined.

2 Syntax

2.1 Signature :

A signature is a triplet Υ = (F, Ac, K) where F is a set of fluents; Ac is a set of actions as follows A1,A2,...,An where Ai $\in$ Ac and i = 1 to n and K is a set of positive integers representing Cost of each action Ai $\in$ Ac as follows k1,k2,...kn where ki $\in$ K and i = 1 to n.

2.2 Literal :

A literal is either a fluent f or its negation ¬f.
Notation: for a fluent f ϵ F, we write $\overline{f}$ to denote the literal corresponding to f, i.e., either f or ¬f.

2.3 Statements :

The system and changes occurring within can be described through a sequence of statements defined in the table:

Statement	Format	Description
Value Statement	$\overline{f}$ after A1 ....An where $\overline{f} \in$ F and Ai $\in$ Ac, for i = 1, ... ,n	$\overline{f}$ holds after performing the sequence A1 ....An of actions in the initial state.
Abbreviation	initially $\overline{f}$	in the initial state $\overline{f}$ holds
Effect Statement	Ai causes $\overline{f}$ if $\overline{g1}$, . . . , $\overline{gk}$	If the action Ai is performed in any state satisfying $\overline{g1}$, . . . , $\overline{gk}$, then in the resulting state $\overline{f}$ holds.
Cost Statement	Ai costs ki (if Ai causes $\overline{f}$ if $\overline{g1}$,....,$\overline{gn}$), ki $\in$ K for i = 1,...,n	If the action Ai is performed in any state satisfying $\overline{g1}$, ...,$\overline{gk}$, then change in state results in fixed cost ki.

Table 1: Syntax Table

3 Semantics

- A state is a mapping σ : F $\to$ {0, 1}. For any f $\in$ F, if σ(f) = 1, then we say that f holds in σ and write $\sigma \models$ f. If σ(f) = 0, then we write σ $\models$ ¬f and say that f does not hold in σ. Let $\sum$ stand for the set of all states.

- A state transition function is a mapping Ψ : Ac $\times \sum \to \sum$. For any $\sigma \in \sum$, for any action Ai $\in$ Ac where i = 1,...,n, Ψ(Ai , σ) is the state resulting from performing the action Ai in the state σ. Also Ψ(Ai,σ) will results in same state if the effect of action is empty.

- A cost transition function is a mapping Γ : Ac $\times \sum \to$ K.For any $\sigma \in \sum$ and for any action Ai $\in$ Ac where i = 1,...,n, Γ(Ai, σ) is the fixed cost ki corresponding to the action Ai, where ki $\in$ K and i=1,...,n, resulting from performing the action Ai in the state σ. Also Γ(Ai,σ) will results in 0 cost if there is no change in state.

- A transition function is generalized to the mapping $\Psi^* : Ac^* \times \sum \rightarrow \sum$ as follows:

  1. Ψ^* (ε , σ) = σ,
  2. Ψ^* ((A1, . . . , An), σ) = Ψ(An, Ψ^* (A1, . . . , An-1)).

- Let L be an action language of the class A over the signature Υ = (F, Ac, K). A structure for L is a triplet S = (Ψ, σ_0, Γ) where Ψ is a state transition function, Γ is a cost transition function and $\sigma_0 \in \sum$ is the initial state

- Let S = (Ψ, σ_0, Γ) be a structure for L. A statement s is true in S, in symbols S $\models$ s, iff

  1. s is of the form $\overline{f}$ after A1, . . . , An, then Ψ((A1, . . . , An), σ_0)) $\models \overline{f}$;
  2. if s is of the form Ai causes $\overline{f}$ if $\overline{g1}$, . . . , $\overline{gk}$, then for every $\sigma \in \sum$ such that $\sigma \models \overline{gj}$, j = 1, . . . , k, Ψ(Ai, σ) $\models \overline{f}$.
  3. if s is of the form A1,...,An costs k1,...,kn respectively where ki $\in$ K, Ai $\in$ Ac and i = 1,...n, then every $\sigma \in \sum$ such that $\sigma \models \overline{gi}$, i = 1,...,k, Γ(Ai,σ) $\models$ ki.

Let D be an action domain in the language L over the signature Υ=(F,Ac,K). A structure S = (Ψ,σ_0, Γ) is a model of D iff
(M1) for every statement s $\in$ D, S $\models$ s;
(M2) for every Ai $\in$ Ac for every f,g1,...,gn $\in$ F, for every ki $\in$ K and for every $\sigma \in \sum$, if one of the following conditions holds:
(i) D contains an effect statement and a cost statement as follows:

- **Ai causes** $\overline{f}$ **if** $\overline{g1}$,...,$\overline{gk}$,$\sigma \not\models \overline{gj}$ for some j = 1,...,k

- if Ψ(Ai,σ) $\not\models \sigma$ then Γ(Ai,σ) = ki.

(ii) D does not contain an effect statement but contains a 0 cost statement, as follows:

- **Ai causes** $\overline{f}$ **if** $\overline{g1}$,...,$\overline{gk}$ then $\sigma \models$ f iff Ψ(Ai,σ) $\models$ f.

- if Ψ(Ai,σ) = σ then Γ(Ai,σ) = 0.

4 Query

4.1 Syntax

- Query set Q1:
  necessary σ after A1,...,Ai on σ_0
  possibly σ after A1,....,Ai on σ_0

  The first statement says that state σ will always occurs after executing the program on initial state.
  The second statement says that state σ may occurs after executing the program on initial state.

- Query set Q2:
  necessary cost C for A1,....,Ai
  sufficient cost C for A1,....,Ai

  The first statement says that cost C is necessary to execute the program
  The second statement says that cost C is sufficient to execute the program

4.2 Semantics

- The query set Q1 is defined as Q1 = (σ,P,σ_0) where σ_0 is the initial state, σ is the resulting state from the execution of program P and P is a sequence defined as P = (A1,...,An), n $\geq$ 0 of actions and σ, $\sigma_0 \in \sum$
- The Query set Q2 is defined as Q2=(C,P) where C is the cost to execute the program P and P is a sequence defined as P=(A1,....,An), n $\geq$ 0 of actions

5 Examples

5.1 Example 01

5.1.1 Description

Andrew wants to travel by his car to a place. Travelling costs him 50$ when there is fuel in car tank. Travelling costs him 50$ when there is fuel in reserve. When there is no fuel in any of it, Andrew can fuy fuel. Fuel costs him 40$

5.1.2 Representation

Fluents: F = {fuel, reserve}
Actions: Ac = {buy, travel}
Costs: K = {40, 50}

Initially: fuel;
Initially: reserve;
travel causes ¬fuel if fuel, reserve;
travel causes ¬reserve if ¬fuel, reserve;
travel causes ¬fuel if fuel, ¬reserve
travel costs 50;
buy causes fuel if ¬ fuel,reserve;
buy causes fuel if ¬ fuel,¬reserve;
buy causes reserve if fuel,¬reserve;
buy costs 40;

5.1.3 Calculation

$\sum = \{ \sigma_0,\sigma_1,\sigma_2,\sigma_3 \}$

$\sigma_0 = \{$ fuel, reserve $\}$ $\sigma_1 = \{$ ¬fuel, reserve $\}$
$\sigma_2 = \{$ ¬fuel, ¬reserve $\}$ $\sigma_3 = \{$ fuel, ¬reserve $\}$
Ψ(buy, σ_0)=σ_0
Ψ(travel, σ_0)=σ_1
Γ(buy, σ_0) = 0
Γ(travel, σ_0) = 50

Ψ(buy, σ_1)=σ_0

Ψ(travel, σ_1)=σ_2
Γ(buy, σ_1) = 40
Γ(travel, σ_1) = 50

Ψ(buy, σ_2)=σ_3
Ψ(travel, σ_2)=σ_2
Γ(buy, σ_2) = 40
Γ(travel, σ_2) = 0

Ψ(buy, σ_3)=σ_0
Ψ(travel, σ_3)=σ_2
Γ(buy, σ_3) = 40
Γ(travel, σ_3) = 50

5.1.4 Graph

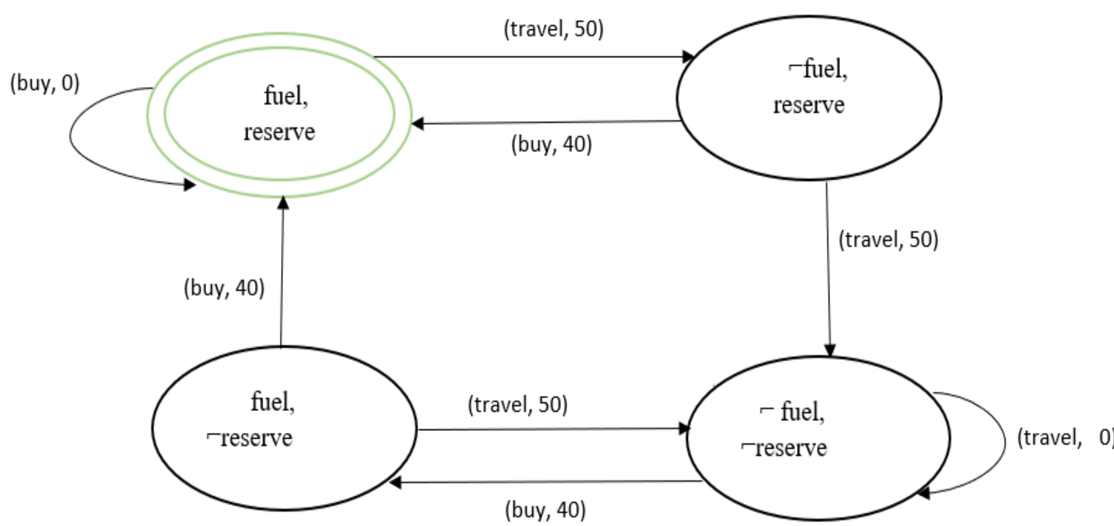

Figure 1: Example 01

5.1.5 Queries

necessary σ_3 for travel, travel, buy on σ_0
possibly σ_1 for buy, buy, travel on σ_0
necessary σ_2 for travel, travel, buy, travel on σ_0

necessary cost 140 for travel, travel, buy
sufficient cost 100 for buy, travel, buy
necessary cost 190 for travel, travel, buy, travel

5.2 Example 02

5.2.1 Description

John visits a painter to buy a specific painting. The cost of painting is 200$ if its available in the shop. But if painting is not available then John needs to order a new one to be painted and will buy once its available.Order costs 50$ At any time only one copy of painting is available and another one to be ordered once sold.

5.2.2 Representation:

Fluents: F = {available, sold}
Actions: Ac = {buy, order}
Costs: K = {200, 50}
Initially: ¬available;
Initially: ¬sold;
buy causes sold if available;
buy causes ¬available;
buy costs 200$;
order causes available if ¬availabl;
order costs 50$;

5.2.3 Calculation:

$\sum = \{ \sigma_0, \sigma_1, \sigma_2, \sigma_3\}$
$\sigma_0 = \{ \neg\text{available}, \neg\text{sold}\}$
$\sigma_1 = \{ \text{available}, \neg\text{sold}\}$
$\sigma_2 = \{ \neg\text{available}, \text{sold}\}$

$\sigma_3 = \{$ available, sold$\}$

Ψ (buy, $\sigma 0) = \sigma 0$
Ψ (order, $\sigma 0) = \sigma 1$
Γ(buy, $\sigma_0) = 0$
Γ(order, $\sigma_0) = 50$

Ψ (buy $\sigma 1) = \sigma 2$
Ψ (order, $\sigma 1) = \sigma 1$
Γ(buy, $\sigma_1) = 200$
Γ(order, $\sigma_1) = 0$

Ψ (buy, $\sigma 2) = \sigma 2$
Ψ (order, $\sigma 2) = \sigma 1$
Γ(buy, $\sigma_2) = 0$
Γ(order, $\sigma_2) = 50$

Ψ (buy, $\sigma 3) = \sigma 2$
Ψ (order, $\sigma 3) = \sigma 3$
Γ(buy, $\sigma_3) = 200$
Γ(order, $\sigma_3) = 0$

5.2.4 Graph

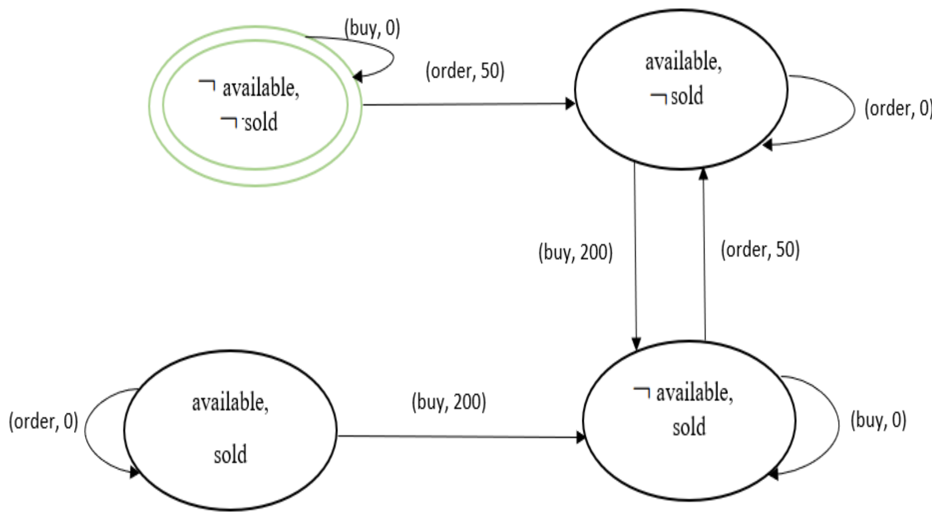

Figure 2: Example 02

5.2.5 Queries

necessary σ_1 for order, buy, order on σ_0
possibly σ_1 for buy, buy, order on σ_0
necessary σ_2 for order, buy on σ_0

necessary cost 300 for order, buy, order
sufficient cost 60 for buy, buy, order
necessary cost 250 for order, buy

5.3 Example 03

5.3.1 Description

There is a man. He can cook, eat, and play. Cooking makes food cooked. he can eat food if it is cooked. After eating he feels not hungry, and food is not cooked again. He can play. Playing makes him hungry. He just can play if he is not hungry. He just cooks when there is no food is cooked. Initially, he is hungry, and no food is cooked. In terms of energy, eating costs 5, cooking costs 15, playing costs 20.

5.3.2 Representation in language

Fluents: F = {cooked, hungry}
Actions: Ac = {cook, eat, play}
Costs: K = {15, 5, 20}

initially ¬cooked;
initially hungry;
cook causes cooked if ¬cooked;
cook costs 15;
eat causes ¬cooked if cooked;
eat causes ¬hungry if cooked;
eat costs 5;
play causes hungry if ¬hungry;
play costs 20;

5.3.3 Calculation

$\sum = \{\sigma_0 ,\sigma_1, \sigma_2, \sigma_3\}$

$\sigma_0 = \{\neg\text{cooked, hungry}\}$
$\sigma_1 = \{\text{cooked, hungry}\}$
$\sigma_2 = \{\neg\text{cooked}, \neg\text{hungry}\}$
$\sigma_3 = \{\text{cooked}, \neg\text{hungry}\}$

$\Psi(\text{eat}, \sigma_0) = \sigma_0$
$\Psi(\text{cook}, \sigma_0) = \sigma_1$
$\Psi(\text{play}, \sigma_0) = \sigma_0$
$\Gamma(\text{eat}, \sigma_0) = 0$
$\Gamma(\text{cook}, \sigma_0) = 15$
$\Gamma(\text{play}, \sigma_0) = 0$

$\Psi(\text{eat}, \sigma_1) = \sigma_2$

$\Psi(\text{cook}, \sigma_1) = \sigma_1$
$\Psi(\text{play}, \sigma_1) = \sigma_1$
$\Gamma(\text{eat}, \sigma_1) = 5$
$\Gamma(\text{cook}, \sigma_1) = 0$
$\Gamma(\text{play}, \sigma_1) = 0$

$\Psi(\text{eat}, \sigma_2) = \sigma_2$
$\Psi(\text{cook}, \sigma_2) = \sigma_3$
$\Psi(\text{play}, \sigma_2) = \sigma_1$
$\Gamma(\text{eat}, \sigma_2) = 0$
$\Gamma(\text{cook}, \sigma_2) = 15$
$\Gamma(\text{play}, \sigma_2) = 20$

$\Psi(\text{eat}, \sigma_3) = \sigma_2$
$\Psi(\text{cook}, \sigma_3) = \sigma_3$
$\Psi(\text{play}, \sigma_3) = \sigma_1$
$\Gamma(\text{eat}, \sigma_3) = 5$
$\Gamma(\text{cook}, \sigma_3) = 0$
$\Gamma(\text{play}, \sigma_3) = 20$

5.3.4 Graph

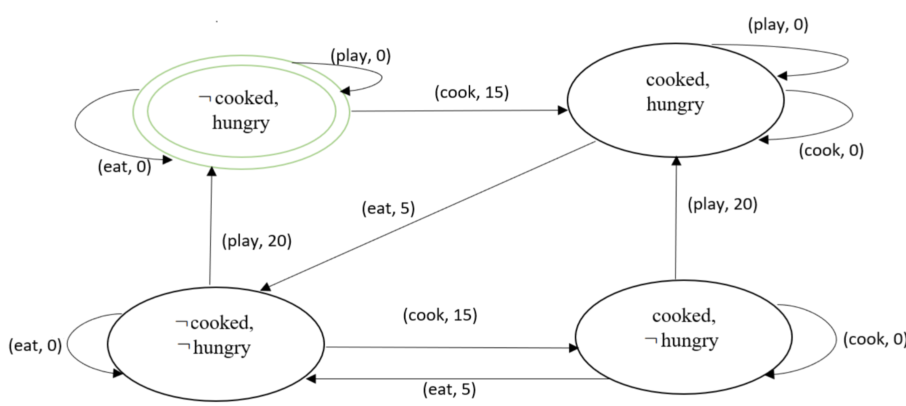

Figure 3: Example 03

5.3.5 Queries

necessary σ_2 for cook, eat, cook on σ_0
possibly σ_3 for play, cook, play, eat on σ_0
necessary σ_1 for cook, eat, play, cook on σ_0

necessary cost 35 for cook, eat, cook
sufficient cost 25 for play, cook, play, eat
necessary cost 55 for cook, eat, play, cook

6 Appendix

List of Figures

List of Tables