

Planning-Notes

Nitin Yadav

July 2020

1 PDDL Translation to SAS using FD parser

```
(define (domain test)
  (:requirements :adl)
  (:predicates
    (a)
    (b)
    (c)
    (d)
    (e)
  )

  (:action action
    :parameters ()
    :precondition (a)
    :effect (and
      (when (b) (c))
      (when (c) (e))
    )
  )
)

(define (problem problem_test)
  (:domain test)

  (:init
    (a)
    (b)
  )

  (:goal
    (e)
  )
)
```

```
2
begin_variable
var0
-1
2
Atom c()
NegatedAtom c()
end_variable
begin_variable
var1
-1
2
Atom e()
NegatedAtom e()
end_variable
0
begin_state
1
1
end_state
begin_goal
1
1 0
end_goal
1
begin_operator
action
0
2
0 0 -1 0
1 0 0 1 -1 0
0
end_operator
0
```

Listing 1: SAS encoding

```

(define (domain test)
  (:requirements :adl)
  (:predicates
    (a)
    (b)
    (c)
    (d)
    (e)
  )

  (:action action
    :parameters ()
    :precondition (a)
    :effect (and
      (when (b) (c))
      (when (c) (e))
    )
  )

  (:action actionA
    :parameters ()
    :precondition (not (a))
    :effect (a)
  )
)

(define (problem problem_test)
  (:domain test)

  (:init
    (not a)
    (b)
  )

  (:goal
    (e)
  )
)

```

```

3
begin_variable
var0
-1
2
Atom a()
NegatedAtom a()
end_variable
begin_variable
var1
-1
2
Atom c()
NegatedAtom c()
end_variable
begin_variable
var2
-1
2
Atom e()
NegatedAtom e()
end_variable
0
begin_state
1
1
1
end_state
begin_goal
1
2 0
end_goal
2
begin_operator
action
1
0 0
2
0 1 -1 0
1 1 0 2 -1 0
0
end_operator
begin_operator
actiona
0
1
0 0 1 0
0
end_operator
0

```

Listing 2: SAS encoding

```

(define (domain test)

  (:requirements :adl)

  (:predicates
    (a)
    (b)
    (c)
    (d)
    (e)
  )

  (:action action
    :parameters ()
    :precondition (a)
    :effect (and
      (when (not (e)) (e))
      (when (c) (e))
    )
  )

  (:action actionA
    :parameters ()
    :precondition (not (a))
    :effect (a)
  )
)

(define (problem problem_test)

  (:domain test)

  (:init
    (not a)
    (b)
  )

  (:goal
    (e)
  )
)

```

```

2
begin_variable
var0
-1
2
Atom a()
NegatedAtom a()
end_variable
begin_variable
var1
-1
2
Atom e()
NegatedAtom e()
end_variable
0
begin_state
1
1
end_state
begin_goal
1
1 0
end_goal
2
begin_operator
action
1
0 0
1
0 1 -1 0
0
end_operator
begin_operator
actiona
0
1
0 0 1 0
0
end_operator
0

```

Listing 3: SAS encoding

```

(define (domain test)

  (:requirements :adl)

  (:predicates
    (a)
    (b)
    (c)
    (d)
    (e)
  )

  (:action action
    :parameters ()
    :precondition (a)
    :effect (oneof
      (when (b) (c))
      (when (c) (e))
    )
  )

  (:action actionA
    :parameters ()
    :precondition (not (a))
    :effect (a)
  )
)

(define (problem problem_test)

  (:domain test)

  (:init
    (not a)
    (b)
  )

  (:goal
    (e)
  )
)

```

```

3
begin_variable
var0
-1
2
Atom a()
NegatedAtom a()
end_variable
begin_variable
var1
-1
2
Atom c()
NegatedAtom c()
end_variable
begin_variable
var2
-1
2
Atom e()
NegatedAtom e()
end_variable
0
begin_state
1
1
1
end_state
begin_goal
1
2 0
end_goal
3
begin_operator
action_DETDUP_0
1
0 0
1
0 1 -1 0
0
end_operator
begin_operator
action_DETDUP_1
1
0 0
1
1 1 0 2 -1 0
0
end_operator
begin_operator
actiona
0
1
0 0 1 0
0
end_operator
0

```

Listing 4: SAS encoding