

# Description Logics State Features for Planning (dlplan)

Dominik Drexler  
Linköping University, Linköping, Sweden  
dominik.drexler@liu.se

September 1, 2021

## 1 Introduction

This document aims at providing a better understanding of the different types of elements and different elements itself that can be used with this library. There are four different types of elements: concepts, roles, boolean, and numericals. The main difference is that each type returns different types of objects when evaluated on a given planning state. Evaluating a concept returns a set of unary relations over objects, evaluating a role returns a set of binary relations over objects, evaluating a boolean returns a Boolean, and evaluating a numerical returns a natural number. The following section shows how elements can be composed to obtain more complex elements.

## 2 Available Elements

In this section, we describe the elements that are available to construct more complex elements. We make use of description logics [1] and include Boolean and numerical elements as described in [2].

Consider concepts  $C, D$ , roles  $R, S, T$ , the universe  $\Delta$  containing all objects, and  $X$  to be either a concept or a role. Also consider some predicate  $p(c_0, \dots, c_{n-1})$  with arity  $n$ , integers  $0 \leq i, j \leq n-1$ , integer  $k \in \{0, 1\}$ , and some constant  $a$  that occurs in every instance.

## 2.1 Syntax Overview

### 2.1.1 Concepts

Name	Abstract syntax	Concrete syntax
Primitive concept	$p(i)$	<code>p(i)</code>
Top	$\top$	<code>c_top</code>
Bottom	$\perp$	<code>c_bot</code>
Intersection	$C \sqcap D$	<code>c_and(C, D)</code>
Union	$C \sqcup D$	<code>c_or(C, D)</code>
Negation	$\neg C$	<code>c_not(C)</code>
Difference	$C \setminus D$	<code>c_diff(C, D)</code>
Value restriction	$\forall R.C$	<code>c_all(R, C)</code>
Existential quantification	$\exists R.C$	<code>c_some(R, C)</code>
Role-value-map	$R \subseteq S$	<code>c_subset(R, S)</code>
One-of	$a$	<code>c_one-of(a)</code>
Projection	$R[k]$	<code>c_projection(R, k)</code>

Table 1: Concepts

### 2.1.2 Roles

Name	Abstract syntax	Concrete syntax
Primitive role	$p(i, j)$	<code>p(i,j)</code>
Universal role	$\top$	<code>r_top</code>
Intersection	$R \sqcap S$	<code>r_and(R, S)</code>
Union	$R \sqcup S$	<code>r_or(R, S)</code>
Complement	$\neg R$	<code>r_not(R)</code>
Difference	$R \setminus S$	<code>r_diff(R, S)</code>
Inverse	$R^{-1}$	<code>r_inverse(R)</code>
Composition	$R \circ S$	<code>r_compose(R, S)</code>
Transitive closure	$R^+$	<code>r_transitive_closure(R)</code>
Transitive reflexive closure	$R^*$	<code>r_transitive_reflexive_closure(R)</code>
Role restriction	$R _C$	<code>r_restrict(R, C)</code>
Identity	$id(C)$	<code>r_identity(C)</code>

Table 2: Roles

### 2.1.3 Booleans

Name	Abstract syntax	Concrete syntax
Empty	$Empty(X)$	<code>b_empty(X)</code>

Table 3: Booleans

### 2.1.4 Numericals

Name	Abstract syntax	Concrete syntax
Count	$Count(X)$	$n\_count(X)$
Concept distance	$ConceptDistance(C, R, D)$	$n\_concept\_distance(C, R, D)$
Sum concept distance	$SumConceptDistance(C, R, D)$	$n\_sum\_concept\_distance(C, R, D)$
Role distance	$RoleDistance(R, S, T)$	$n\_role\_distance(R, S, T)$
Sum role distance	$SumRoleDistance(R, S, T)$	$n\_sum\_role\_distance(R, S, T)$

Table 4: Numericals

## 2.2 Semantics

### 2.2.1 Concepts

- $(p(i))^s$  is a concept that denotes the set of objects occuring at index  $i$  in ground atoms of predicate  $p$  in state  $s$ .
- $\top^s = \Delta^s$
- $\perp^s = \emptyset$
- $(C \sqcap D)^s = C^s \cap D^s$
- $(C \sqcup D)^s = C^s \cup D^s$
- $(\neg C)^s = \Delta^s \setminus C^s$
- $(C \setminus D)^s = (C^s \setminus D^s)$
- $(\forall R.C)^s = \{a \mid \forall b : (a, b) \in R^s \rightarrow b \in C^s\}$
- $(\exists R.C)^s = \{a \mid \exists b : (a, b) \in R^s \wedge b \in C^s\}$
- $(R \subseteq S)^s = \{a \mid \forall b : (a, b) \in R^s \rightarrow (a, b) \in S^s\}$
- $a^s = \{a\}$
- $(R[k])^s = \begin{cases} (\exists R.\top)^s & \text{if } k = 0 \\ (\exists R^{-1}.\top)^s & \text{if } k = 1 \end{cases}$

### 2.2.2 Roles

- $p(i, j)^s$  is a role that denotes the set of pairs of objects occuring at index  $i$  and  $j$  in ground atoms of  $p$  in state  $s$ .
- $\top^s = \Delta^s \times \Delta^s$
- $(R \sqcap S)^s = R^s \cap S^s$
- $(R \sqcup S)^s = R^s \cup S^s$

- $(\neg R)^s = \top^s \setminus R^s$
- $(R \setminus S)^s = (R^s \setminus S^s)$
- $(R^{-1})^s = \{(b, a) \in \top^s \mid (a, b) \in R^s\}$
- $(R \circ S)^s = R^s \circ S^s$
- $(R^+)^s = \bigcup_{n \geq 1} (R^s)^n$
- $(R^*)^s = \bigcup_{n \geq 0} (R^s)^n$
- $(R|_C)^s = R^s \sqcap (\Delta^s \times C^s)$
- $id(C) = \{(d, d) \mid d \in C^s\}$

where the iterated composition  $(R^s)^n$  is inductively defined as  $(R^s)^0 = \{(d, d) \mid d \in \Delta^s\}$  and  $(R^s)^{n+1} = (R^s)^n \circ R^s$ .

### 2.2.3 Booleans

- $Empty(X)^s$  is true iff  $|X^s| = 0$

### 2.2.4 Numericals

- $Count(X)^s \equiv |X^s|$
- $ConceptDistance(C, R, D)^s$  is the smallest  $n \in \mathbb{N}_0$  s.t. there are objects  $x_0, \dots, x_n$  with  $x_0 \in C^s$ ,  $x_n \in D^s$  and  $(x_i, x_{i+1}) \in R^s$  for  $i = 0, \dots, n-1$ . Special cases: If  $C^s$  is empty then the element evaluates to 0 and if no such  $n$  exists then it evaluates to  $\infty$ .
- $SumConceptDistance(C, R, D)^s := \sum_{x \in C^s} ConceptDistance(\{x\}, R, D)^s$  where the sum evaluates to  $\infty$  if any term is  $\infty$ .
- $RoleDistance(R, S, T)^s$  is the smallest  $n \in \mathbb{N}_0$  s.t. there are objects  $x_0, \dots, x_n$ , there exists some  $(a, x_0) \in R^s$ ,  $(a, x_n) \in T^s$ , and  $(x_i, x_{i+1}) \in S^s$  for  $i = 0, \dots, n-1$ . Special cases: If  $R^s$  is empty then the element evaluates to 0 and if no such  $n$  exists then it evaluates to  $\infty$ .
- $SumRoleDistance(R, S, T)^s := \sum_{r \in R^s} RoleDistance(\{r\}, S, T)^s$ , where the sum evaluates to  $\infty$  if any term is  $\infty$ .

## References

- [1] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.
- [2] Dominik Drexler, Jendrik Seipp, and Hector Geffner. Expressing and exploiting the common subgoal structure of classical planning domains using sketches: Extended version. arXiv:2105.04250 [cs.AI], 2021.