A Logic-based Framework for Explainable Agent Scheduling Problems

Supplement

1 Illustrating Example Encoding

Recall the employee shift assignment problem presented in Section 2: four employees $A = \{Thanos, Irene, Vicky, Rose\}$, three shift types $R = \{morning, afternoon, evening\}$, and three time steps $S = \{Monday, Tuesday, Wednesday\}$.

To represent the problem in (propositional) logic, we introduce the Boolean decision variables $x_{i,j,t}$ for all $a_i \in A$, $r_j \in R$, and $s_t \in S$, where each variable is set to true if and only if agent a_i is assigned shift r_j on day s_t. Otherwise, it is set to false. These variables comprise the domain constraints C_D and employee constraints C_A.

Specifically, the domain constraints C_D are:[1]

C_1: *All employees must be assigned a total of two shifts*:

$$exactly_2(x_{i,1,1}, x_{i,1,2}, \ldots, x_{i,3,3}\}) \; \forall a_i \in A \tag{1}$$

C_2: *Employees cannot be assigned multiple shifts per day*:

$$atmost_1(\{x_{i,1,t}, x_{i,2,t}, x_{i,3,t}\}) \; \forall a_i \in A, s_t \in S \tag{2}$$

C_3: *No two employees can be assigned the same shift on the same day*:

$$atmost_1(\{x_{1,j,t}, x_{2,j,t}, x_{3,j,t}, x_{4,j,t}\}) \; \forall r_j \in R, s_t \in S \tag{3}$$

C_4: *Employees cannot be assigned a morning shift right after an evening shift*:

$$\{\neg x_{i,3,1} \vee \neg x_{i,1,2}, \neg x_{i,3,2} \vee \neg x_{i,1,3}\} \; \forall a_i \in A \tag{4}$$

Further, the employee constraints C_A with weights to indicate their priorities are:

C_T: *Thanos wants only morning or afternoon shifts*:

$$(15, \{\neg x_{1,3,1}, \neg x_{1,3,2}, \neg x_{1,3,3}\}) \tag{5}$$

C_I: *Irene does not want evening shifts*:

$$(10, \{\neg x_{2,3,1}, \neg x_{2,3,2}, \neg x_{2,3,3}\}) \tag{6}$$

C_V: *Vicky wants the afternoon shift on Tue. and Wed.*:

$$(35, \{x_{3,2,2}, x_{3,2,3}\}) \tag{7}$$

C_R: *Rose wants the morning shift on Tue. and Wed.*:

$$(35, \{x_{4,1,2}, x_{4,1,3}\}) \tag{8}$$

Constraints (1) to (8) form the knowledge base KB. As discussed in Section 3, to generate an (optimal) schedule Σ_μ for this example, we can simply look for a model μ of KB that satisfies all constraints in C_D and maximizes the cumulative sum of weights of satisfied constraints in C_A. Table 1 depicts a derived optimal schedule Σ_μ.

Employee Name	Monday	Tuesday	Wednesday
Thanos	*morning*	*evening*	–
Irene	*afternoon*	–	*evening*
Vicky	–	*afternoon*	*afternoon*
Rose	–	*morning*	*morning*

Table 1: An optimal schedule Σ_μ for the example employee shift assignment problem.

[1] For compactness, we use *cardinality constraints* $exactly_k(\{v_i \mid v_i \in V\})$ and $atmost_k(\{v_i \mid v_i \in V\})$ to represent constraints that limit the truth values assigned to the variables to *exactly* k and *at most* k.