splitting. For any program $\Pi$ and any epistemic constraint $r$, we can always take the whole set of atoms $U = Atoms(\Pi \cup \{r\})$ as epistemic splitting set for $\Pi' = \Pi \cup \{r\}$ and take $B_U(\Pi') = \Pi$ and $T_U(\Pi') = \{r\}$. For any world view $W$ of $B_U(\Pi')$ two things may happen. A first possibility is $W \models r$, and so the body of $r$ has some false subjective literal in $W$, so $E_U(\Pi', W)$ would be equivalent to $\bot \leftarrow \bot$. Then, the unique world view for the top would be $W_t = [\emptyset]$ and $W \sqcup W_t = W$. A second case is $W \not\models r$, so all literals in the body are satisfied and $E_U(\Pi', W)$ would be equivalent to $\bot \leftarrow \top$ which has no world views. To sum up, we get exactly those world views $W$ of $\Pi$ that satisfy $r$. □

To conclude the exploration of consequences of epistemic splitting, let us consider a possible application to conformant planning. To this aim, consider the following simple example.

**Example 2.** *To turn on the light in a room, we can toggle one of two lamps $l_1$ or $l_2$. In the initial state, lamp $l_1$ is plugged but we ignore the state of $l_2$. Our goal is finding a plan that guarantees we get light in the room in one step.*

A possible logic program that encodes this scenario for a single transition[2] could be $\Pi_4$:

$$plugged(l_1)$$
$$plugged(l_2) \vee {\sim} plugged(l_2)$$

$$\begin{array}{rcl} light & \leftarrow & toggle(L), plugged(L) \\ \bot & \leftarrow & toggle(l_1), toggle(l_2) \end{array}$$

for $L \in \{l_1, l_2\}$. As we can see, $toggle(l_1)$ would constitute a conformant plan, since we obtain *light* regardless of the initial state, while this does not happen with plan $toggle(l_2)$. In order to check whether a given sequence of actions $A_0, \ldots, A_n$ is a valid conformant plan one would expect that, if we added those facts to the program, a subjective constraint should be sufficient to check that the goal holds in all the possible outcomes. In our example, we would just use:

$$\bot \leftarrow \texttt{not}\ \mathbf{K}\, light \tag{14}$$

and check that the program $\Pi_4 \cup \{toggle(L)\} \cup \{(14)\}$ has some world view, varying $L \in \{l_1, l_2\}$. Subjective constraint monotonicity guarantees that the addition of this "straighforward" formalisation has the expected meaning.

This method would only allow testing if the sequence of actions constitutes a conformant plan, but does not allow generating those actions. A desirable feature would be the possibility of applying the well-known ASP methodology of separating the program into three sections: generate, define and test. In our case, the "define" and the "test" sections would respectively be $\Pi_4$ and (14), but we still miss a "generate" part, capable of considering different alternative conformant plans. The problem in this case is that we cannot use a simple choice:

$$toggle(L) \vee {\sim} toggle(L)$$

because this would allow a same action to be executed in some of the stable models and not executed in others, all inside a *same* world view. Let us assume that our epistemic semantics has some way to non-deterministically generate a world view in which either $\mathbf{K}\, a$ or $\mathbf{K}\, \texttt{not}\ a$ holds using a given set of rules[3] $Choice(a)$. Then, take the program $\Pi_5$ consisting of rules

$$Choice(toggle(L)) \tag{15}$$

with $L \in \{l_1, l_2\}$ plus $\Pi_4$ and (14). If our semantics satisfies epistemic splitting, it is safe to obtain the world views in three steps: generate first the alternative world views for $toggle(l_1)$ and $toggle(l_2)$ using (15), apply $\Pi_4$ and rule out those world views not satisfying the goal *light* in all situations using (14). To fulfill the preconditions for applying splitting, we would actually need to replace regular literal $toggle(L)$ by $\mathbf{K}\, toggle(L)$ in all the bodies of $\Pi_4$, but this is safe in the current context. Now, we take the bottom program to obtain 4 possible world views $W_0 = [\{toggle(l_1)\}]$, $W_1 = [\{toggle(l_2)\}]$, $W_2 = [\{toggle(l_1), toggle(l_2)\}]$ and $W_3 = [\emptyset]$. When we combine them with the top $\Pi_4$ we obtain $W_0'$ consisting of two stable models:

$$\{toggle(l_1), plugged(l_2), light, \ldots\}$$
$$\{toggle(l_1), {\sim} plugged(l_2), light, \ldots\}$$

and $W_1'$ consisting of other two stable models:

$$\{toggle(l_2), plugged(l_2), light, \ldots\}$$
$$\{toggle(l_2), {\sim} plugged(l_2), \ldots\}$$

where the latter does not contain *light*. Finally, constraint (14) would rule out $W_1'$.

To sum up, epistemic splitting provides a natural way of formulating conformant planning problems by a separation into three sections: a generation part, the usual encoding of the actions scenario and a test part consisting of a subjective constraint to guarantee that the goal is always reached.

## Splitting in some existing semantics

In this section we study the property of epistemic splitting for the approaches mentioned in the introduction. We will begin by proving that G91 actually satisfies this property. To this aim, we start with some definitions and auxiliary results. Given a set of propositional interpretations $W \subseteq 2^{At}$ and a set of atoms $U$, by $W_{|U} \stackrel{\text{def}}{=} \{\ I \cap U \mid I \in U\ \}$, we denote the restriction of $W$ to $U$. Given a set of atoms $U$, by $\overline{U}$, we denote its complement $At \setminus U$.

[2] For simplicity, we omit time arguments or inertia, as they are not essential for the discussion.

[3] For instance, in Gelfond (1991), this could be just the rule $a \leftarrow \texttt{not}\ \mathbf{K}\, \texttt{not}\ a$. Other semantics may have alternative ways of expressing this intended behaviour.