COMP4418, 2017 – Exercise

Weeks 6, 7, 8, 9

---

1 Answer Set Programming

1.1 Modelling

A *set cover* of a set S of sets $s_1, \dots, s_n$ is a set of sets $C \subseteq S$ such that $\bigcup_{s \in S} s = \bigcup_{s \in C} s$. A *k-set cover* is a set cover of size k, that is, $|C| = k$.

For instance, for an input $S = \{\{1,2\},\{2,3\},\{4,5\},\{1,2,3\}\}$, there is a 2-set cover $C = \{\{1,2,3\},\{4,5\}\}$ since $\bigcup_{s \in S} s = \{1,2\} \cup \{2,3\} \cup \{4,5\} \cup \{1,2,3\} = \{1,2,3\} \cup \{4,5\} = \bigcup_{s \in C} s$.

Write an ASP program that decides the k-Set-Cover problem:

> Input: a set of sets and a natural number $k \geq 0$.
>
> Problem: decide if there is a k-set cover.

Assume the input parameter $S = \{s_1, \dots, s_n\}$ is encoded by a binary predicate `s` in the way that $x \in s_i$ iff `s(`i`,`x`)`. The input parameter k is given as constant symbol `k`. Use a unary predicate `c` to represent the output C in the way that $s_i \in C$ iff `c(`i`)`.

Solution

```
% Instance encoding of the above example:
% s(1, (1;2)). is a shorthand for s(1,1). s(1,2).
s(1, (1;2)).
s(2, (2;3)).
s(3, (4;5)).
s(4, (1;2;3)).

% Helper predicates.
universe(X) :- s(S,X).
covered(X) :- c(S), s(S,X).

% Generate candidate of cardinality k.
k { c(S) : s(S,X) } k.

% Test that the candidate covers the whole universe.
:- universe(X), not covered(X).

#show c/1.
```

1.2 Semantics

Consider the following program P.

$$a.$$
$$c \;\text{:-}\; \text{not}\, b, \text{not}\, d.$$
$$d \;\text{:-}\; a, \text{not}\, c.$$

Determine the stable models of S.

Solution

Candidate S	Reduct P^S	Stable model?
$\{a, b, c, d\}$	a.	✗
$\{a, b, c\}$	a.	✗
$\{a, b, d\}$	$a. \quad d \text{ :- } a.$	✗
$\{a, c, d\}$	a.	✗
$\{b, c, d\}$	a.	✗
$\{a, b\}$	$a. \quad d \text{ :- } a.$	✗
$\{a, c\}$	$a. \quad c.$	✓
$\{a, d\}$	$a. \quad d \text{ :- } a.$	✓
$\{b, c\}$	a.	✗
$\{b, d\}$	$a. \quad d \text{ :- } a.$	✗
$\{c, d\}$	a.	✗
$\{a\}$	$a. \quad c. \quad d \text{ :- } a.$	✗
$\{b\}$	$a. \quad d \text{ :- } a.$	✗
$\{c\}$	$a. \quad c.$	✗
$\{d\}$	$a. \quad d \text{ :- } a.$	✗
$\{\}$	$a. \quad c. \quad d \text{ :- } a.$	✗

2 Reasoning about Knowledge

2.1 Cardinality of different sets related to $\mathcal{OL}$

(This question is not relevant for the exam, but a good exercise to think a bit about the logic.)

Is the...

- set of formulas of $\mathcal{OL}_{\text{PL}}$
- set of worlds of $\mathcal{OL}_{\text{PL}}$
- set of epistemic states $\mathcal{OL}_{\text{PL}}$
- set of formulas of $\mathcal{OL}$
- set of worlds of $\mathcal{OL}$
- set of epistemic states $\mathcal{OL}$

... finite, countably infinite, or uncountable?

Solution

Claim *The set of formulas of $\mathcal{OL}_{\text{PL}}$ is countably infinite.*

Proof. Let F be the set of formulas of $\mathcal{OL}_{\text{PL}}$. Note that there are countably infinitely many propositions $\{p, q, r, \ldots\}$ in $\mathcal{OL}_{\text{PL}}$. Hence F is clearly infinite, and it only remains to be shown that F is countable.

Let F_n be the formulas of length n. Every formula in F_n is a string over the alphabet $A = \{\neg, (,), \vee, \mathbf{K}, \mathbf{O}, p, q, r, \ldots\}$ and can hence be identified with an n-tuple from $\underbrace{A \times \ldots \times A}_{n\text{-times}}$. Since the cartesian product of finitely many countable sets is countable, F_n is countable. Since the set of formulas of $\mathcal{OL}_{\text{PL}}$ is $\bigcup_{n \geq 1} F_n$ and all F_n are countable, the set of formulas is countable as well. □

Claim *The set of worlds of $\mathcal{OL}_{\text{PL}}$ is uncountable.*

Proof. Let s be a set of atomic propositions. We can identify every world w with a set s such that for every atomic proposition p, $w[p] = 1$ iff $p \in s$. The set of worlds is hence bijective to the set of sets of propositions. As the powerset of a countably infinite set is uncountable according to Cantor's theorem, and since the set of propositions is countably infinite, the set of sets of propositions is uncountable and hence the set of worlds is uncountable as well. □

Claim *The set of epistemic states of $\mathcal{OL}_{\text{PL}}$ is uncountable.*

Proof. An epistemic state is a set of worlds. The set of epistemic states is hence the powerset of the set of worlds, which is uncountable, so the set of epistemic states is uncountable as well. □

Claim *The set of formulas of $\mathcal{OL}$ is countably infinite.*

Proof. There are countably infinitely many variables, standard names, function symbols, and predicate symbols in $\mathcal{OL}$. The proof then follows the same argument as for $\mathcal{OL}_{\text{PL}}$. □

Claim *The set of worlds of $\mathcal{OL}$ is uncountable.*

Proof. The argument is analogous to $\mathcal{OL}_{\text{PL}}$. □

Claim *The set of epistemic states of $\mathcal{OL}$ is uncountable.*

Proof. The argument is analogous to $\mathcal{OL}_{\text{PL}}$. □

2.2 Introspection

Prove the following results from Slide 26:

- $\models \exists x \mathbf{K} \alpha \rightarrow \mathbf{K} \exists x \alpha$.
- $\not\models \mathbf{K} \exists x \alpha \rightarrow \exists x \mathbf{K} \alpha$.

Solution

Theorem $\models \exists x \mathbf{K} \alpha \rightarrow \mathbf{K} \exists x \alpha$.

Proof. We need to show that for all e, $e \models \exists x \mathbf{K} \alpha \rightarrow \mathbf{K} \exists x \alpha$. Expanding the $\rightarrow$ and applying the rules for $\vee$ and $\neg$ gives us that this is equivalent to showing that for all e, $e \not\models \exists x \mathbf{K} \alpha$ or $e \models \mathbf{K} \exists x \alpha$.

Suppose that $e \models \exists x \mathbf{K} \alpha$, for otherwise the claim holds immediately. Then by the rule for $\exists$, for some standard name n, $e \models \mathbf{K} \alpha^x_n$. By the rule for $\mathbf{K}$, for some standard name n, for all worlds w, if $w \in e$, $e, w \models \alpha^x_n$. Then for all worlds w, if $w \in e$, for some standard name n, $e, w \models \alpha^x_n$. By the rule for $\exists$, for all worlds w, if $w \in e$, $e, w \models \exists x \alpha$. By the rule for $\mathbf{K}$, $e \models \mathbf{K} \exists x \alpha$. □

Theorem $\not\models \mathbf{K} \exists x \alpha \rightarrow \exists x \mathbf{K} \alpha$.

Proof. We need to show that for some e, $e \not\models \mathbf{K}\exists x\,\alpha \to \exists x\,\mathbf{K}\,\alpha$. This is equivalent to $e \models \neg(\mathbf{K}\exists x\,\alpha \to \exists x\,\mathbf{K}\,\alpha)$, which is equivalent to showing $e \models \mathbf{K}\exists x\,\alpha \wedge \neg\exists x\,\mathbf{K}\,\alpha$, which further reduces to $e \models \mathbf{K}\exists x\,\alpha$ and $e \not\models \exists x\,\mathbf{K}\,\alpha$.

Let α be $P(x)$ and $e = \{w_1, w_2\}$, where $w_1[P(n)] \neq w_2[P(n)]$ and $w_1[P(\#1)] = 1$ and $w_2[P(\#2)] = 1$.

By construction, there is a standard name n such that $w_1[P(n)] = 1$, and there is another standard name n such that $w_2[P(n)] = 1$. Thus for every $w \in e$, there is a standard name n, such that $w \models P(n)$. Thus by the rule for $\exists$, for every world w, if $w \in e$, then $w \models \exists P(x)$. Thus by the rule for $\mathbf{K}$, $e \models \mathbf{K}\exists P(x)$.

On the other hand, by construction, there is no standard name n such that $w_1[P(n)] = 1$ and $w_2[P(n)] = 1$. Thus there is no standard name n such that for every $w \in e$, $w \models P(n)$. Thus by the rule for $\mathbf{K}$, there is no standard name n such that $e \models \mathbf{K}\,P(n)$. Thus by the rule for $\exists$, it is not the case that $e \models \exists x\,\mathbf{K}\,P(x)$, that is, $e \not\models \exists x\,\mathbf{K}\,P(x)$. $\square$

2.3 Only-Knowing

Suppose all you know is

- the father of Sally is Frank or Fred, and
- Sally's father is rich.

Formalise this statement in $\mathcal{OL}$. Show that this statement does entail that Frank or Fred is known to be rich, but it is not known who of them is rich.

Solution

Perhaps the most direct translation to $\mathcal{OL}$ is:

$$\mathbf{O}\big((\mathrm{fatherOf(Sally)} = \mathrm{Frank} \vee \mathrm{fatherOf(Sally)} = \mathrm{Fred}) \wedge \\ \mathrm{Rich(fatherOf(Sally))}\big)$$

where Sally, Frank, Fred shall denote standard names. Let KB denote that sentence within $\mathbf{O}$.

We now need to prove the entailment

$$\mathbf{O}\mathrm{KB} \models \mathbf{K}\,(\mathrm{Rich(Frank)} \vee \mathrm{Rich(Fred)}) \wedge \neg\mathbf{K}\,\mathrm{Rich(Frank)} \wedge \neg\mathbf{K}\,\mathrm{Rich(Fred)}$$

Suppose $e \models \mathbf{O}\mathrm{KB}$. We need to show $e \models \mathbf{K}\,(\mathrm{Rich(Frank)} \vee \mathrm{Rich(Fred)})$ and $e \not\models \mathbf{K}\,\mathrm{Rich(Frank)}$ and $e \not\models \mathbf{K}\,\mathrm{Rich(Fred)}$.

By the rule for $\mathbf{O}$, we have $w \in e$ iff $w \models \mathrm{KB}$. That is, for all w, $w \in e$ iff $w \models \mathrm{fatherOf(Sally)} = \mathrm{Frank} \vee w\mathrm{fatherOf(Sally)} = \mathrm{Fred}$ and $w \models \mathrm{Rich(fatherOf(Sally))}$. Thus for all w, $w \in e$ iff (1) $w[\mathrm{fatherOf(Sally)}] = \mathrm{Frank}$ or $w[\mathrm{fatherOf(Sally)}] = \mathrm{Fred}$ and (2) $w[\mathrm{Rich}(n)] = 1$ where $n = w[\mathrm{fatherOf(Sally)}]$. Thus for all w, $w \in e$ iff either $w[\mathrm{fatherOf(Sally)}] = \mathrm{Frank}$ and $w[\mathrm{Rich(Frank)}] = 1$ or $w[\mathrm{fatherOf(Sally)}] = \mathrm{Fred}$ and $w[\mathrm{Rich(Fred)}] = 1$ (*).

We first show that $e \models \mathbf{K}\,(\mathrm{Rich(Frank)} \vee \mathrm{Rich(Fred)})$. This reduces to showing that for all w, if $w \in e$, then $w \models \mathrm{Rich(Frank)} \vee \mathrm{Rich(Fred)}$. That is, for all $w \in e$, either $w \models \mathrm{Rich(Frank)}$ or $w \models \mathrm{Rich(Fred)}$, which is true by (*).

Next we show that $e \not\models \mathbf{K}\,\mathrm{Rich(Frank)}$. This is equivalent to showing that for some $w \in e$, $w \not\models \mathrm{Rich(Frank)}$. Let w be such that $w[\mathrm{fatherOf(Sally)}] = \mathrm{Fred}$ and $w[\mathrm{Rich(Fred)}] = 1$ and $w[\mathrm{Rich(Frank)}] = 0$. Clearly, w satisfies the right-hand side of (*), so $w \in e$, and clearly $w \not\models \mathrm{Rich(Frank)}$.

Analogously to the case for Frank, we can show that $e \not\models \mathbf{K}\,\mathrm{Rich(Fred)}$.

2.4 Representation Theorem

Suppose you have a wedding database that tells you who is married to whom.[1]

$$\begin{aligned}&\text{Married}(\text{Mia}, \text{Frank}) \land \\ &\exists x\, \text{Married}(x, \text{Fred}) \land \\ &\text{Married}(\text{motherOf}(\text{Sally}), \text{fatherOf}(\text{Sally}))\end{aligned}$$

where Frank, Fred, Mia, Sally are standard names. Call this sentence KB.

(a) Who is not known to be married to Sally?

1. What is the set of tuples of standard names N such that $n \in N$ iff $\mathbf{O}\text{KB} \models \neg\mathbf{K}\,\text{Married}(\text{Sally}, n)$?
2. Determine $\mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, x)]$.
3. Determine whether $\mathbf{O}\text{KB} \models \exists x\, \neg\mathbf{K}\,\text{Married}(\text{Sally}, x)$ using the representation theorem (Slide 31), that is, by checking whether $\models \|\exists x\, \neg\mathbf{K}\,\text{Married}(\text{Sally}, x)\|_{\text{KB}}$.

(b) Who is known to be married?

1. What is the set of standard names N such that $\mathbf{O}\text{KB} \models \mathbf{K}\,\exists y\, (\text{Married}(n, y) \lor \text{Married}(y, n))$?
2. Determine $\mathsf{RES}[\text{KB}, \exists y\, (\text{Married}(x, y) \lor \text{Married}(y, x))]$. (Note: there is one free variable, x.)
3. Determine whether $\mathbf{O}\text{KB} \models \exists x\, \mathbf{K}\, \exists y\, (\text{Married}(x, y) \lor \text{Married}(y, x))$ using the representation theorem, that is, by checking whether $\models \|\exists x\, \mathbf{K}\, \exists y\, (\text{Married}(x, y) \lor \text{Married}(y, x))\|_{\text{KB}}$.

(c) Who is known to be married to an unknown person?

1. What is the set of tuples of standard names N such that $(n_1, n_2) \in N$ iff $\mathbf{O}\text{KB} \models \mathbf{K}\,\text{Married}(n_1, n_2)$?
2. What is the set of standard names N such that $n \in N$ iff $\mathbf{O}\text{KB} \models \mathbf{K}\,\exists x\, (\text{Married}(x, n) \land \neg\mathbf{K}\,\text{Married}(x, n))$?
3. Determine $\mathsf{RES}[\text{KB}, \text{Married}(x, y)]$. (Note: there are two free variables, x and y.)
4. Determine $\mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, y) \land \neg(x = \text{Mia} \land y = \text{Frank}))]$. (Note: there is one free variable, y.)
5. Determine whether $\mathbf{O}\text{KB} \models \exists y\, \mathbf{K}\, \exists x\, (\text{Married}(x, y) \land \neg\mathbf{K}\,\text{Married}(x, y))$ using the representation theorem, that is, by checking whether $\models \|\exists y\, \mathbf{K}\, \exists x\, (\text{Married}(x, y) \land \neg\mathbf{K}\,\text{Married}(x, y))\|_{\text{KB}}$.

Solution

To save some space, we allow ourselves to simplify the end result of $\mathsf{RES}[\text{KB}, \phi]$ sometimes. Any simplification we use should preserve equivalence, that is, we can simplify α to β only when $\models \alpha \leftrightarrow \beta$. A typical simplification is to reduce $(\alpha_1 \land \text{FALSE}) \lor (\alpha_2 \land \text{TRUE})$ to α_2. I'll write $\alpha \overset{\text{simpl}}{=} \beta$ to indicate that α simplifies to β, that is, that $\models \alpha \leftrightarrow \beta$.

[1] In a realistic scenario, we would add $\forall x \forall y\, (\text{Married}(x, y) \leftrightarrow \text{Married}(y, x))$ to formalise that marriage is a symmetric relation. For the sake of this example, we do not add this symmetry constraint to our knowledge.

(a) Who is not known to be married to Sally?

1. N contains all standard names.

2. The standard names that occur in the KB and the query are $\{\text{Mia}, \text{Frank}, \text{Fred}, \text{Sally}\}$.

$$
\begin{aligned}
& \mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, x)] \\
= \; & (x = \text{Mia} \wedge \mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, \text{Mia})]) \vee \\
& (x = \text{Frank} \wedge \mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, \text{Frank})]) \vee \\
& (x = \text{Fred} \wedge \mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, \text{Fred})]) \vee \\
& (x = \text{Sally} \wedge \mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, \text{Sally})]) \vee \\
& (x \neq \text{Mia} \wedge x \neq \text{Frank} \wedge x \neq \text{Fred} \wedge x \neq \text{Sally} \wedge \mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, \#7)]_x^{\#7}) \\
= \; & (x = \text{Mia} \wedge \text{FALSE}) \vee \\
& (x = \text{Frank} \wedge \text{FALSE}) \vee \\
& (x = \text{Fred} \wedge \text{FALSE}) \vee \\
& (x = \text{Sally} \wedge \text{FALSE}) \vee \\
& (x \neq \text{Mia} \wedge x \neq \text{Frank} \wedge x \neq \text{Fred} \wedge x \neq \text{Sally} \wedge \text{FALSE}_x^{\#7}) \\
& (\text{because KB} \not\models \text{Married}(\text{Sally}, \text{Mia}), \text{KB} \not\models \text{Married}(\text{Sally}, \text{Frank}), \ldots, \text{KB} \not\models \text{Married}(\text{Sally}, \#7)) \\
\overset{\mathsf{simpl}}{=} \; & \text{FALSE}
\end{aligned}
$$

3. $\models \|\exists x \, \neg \mathbf{K} \, \text{Married}(\text{Sally}, x)\|_{\text{KB}}$ because:

$$
\begin{aligned}
& \|\exists x \, \neg \mathbf{K} \, \text{Married}(\text{Sally}, x)\|_{\text{KB}} \\
= \; & \exists x \, \neg \|\mathbf{K} \, \text{Married}(\text{Sally}, x)\|_{\text{KB}} \\
= \; & \exists x \, \neg \mathsf{RES}[\text{KB}, \|\text{Married}(\text{Sally}, x)\|_{\text{KB}}] \\
= \; & \exists x \, \neg \mathsf{RES}[\text{KB}, \text{Married}(\text{Sally}, x)] \\
= \; & \exists x \, \neg \text{FALSE}
\end{aligned}
$$

(b) Who is known to be married?

1. $N = \{\text{Mia}, \text{Frank}, \text{Fred}\}$

2. The standard names that occur in the KB and the query are $\{\text{Mia}, \text{Frank}, \text{Fred}, \text{Sally}\}$.

$$
\begin{aligned}
& \mathsf{RES}[\text{KB}, \exists y \, (\text{Married}(x, y) \vee \text{Married}(y, x))] \\
= \; & (x = \text{Mia} \wedge \mathsf{RES}[\text{KB}, \exists y \, (\text{Married}(\text{Mia}, y) \vee \text{Married}(y, \text{Mia}))]) \vee \\
& (x = \text{Frank} \wedge \mathsf{RES}[\text{KB}, \exists y \, (\text{Married}(\text{Frank}, y) \vee \text{Married}(y, \text{Frank}))]) \vee \\
& (x = \text{Fred} \wedge \mathsf{RES}[\text{KB}, \exists y \, (\text{Married}(\text{Fred}, y) \vee \text{Married}(y, \text{Fred}))]) \vee \\
& (x = \text{Sally} \wedge \mathsf{RES}[\text{KB}, \exists y \, (\text{Married}(\text{Sally}, y) \vee \text{Married}(y, \text{Sally}))]) \vee \\
& (x \neq \text{Mia} \wedge x \neq \text{Frank} \wedge x \neq \text{Fred} \wedge x \neq \text{Sally} \wedge \mathsf{RES}[\text{KB}, \exists y \, (\text{Married}(\#7, y) \vee \text{Married}(y, \#7))]_x^{\#7}) \\
= \; & (x = \text{Mia} \wedge \text{TRUE}) \vee \\
& (x = \text{Frank} \wedge \text{TRUE}) \vee \\
& (x = \text{Fred} \wedge \text{TRUE}) \vee \\
& (x = \text{Sally} \wedge \text{FALSE}) \vee \\
& (x \neq \text{Mia} \wedge x \neq \text{Frank} \wedge x \neq \text{Fred} \wedge x \neq \text{Sally} \wedge \text{FALSE}_x^{\#7}) \\
& (\text{because KB} \models \exists y \, (\text{Married}(\text{Mia}, y) \vee \text{Married}(y, \text{Mia})), \text{ etc.}) \\
\overset{\mathsf{simpl}}{=} \; & (x = \text{Mia} \vee x = \text{Frank} \vee x = \text{Fred})
\end{aligned}
$$

3. $\models \|\exists x \, \mathbf{K} \, \exists y \, (\text{Married}(x, y) \vee \text{Married}(y, x))\|_{\text{KB}}$ because:

$$
\begin{aligned}
&\|\exists x\, \mathbf{K}\, \exists y\, (\mathrm{Married}(x,y) \vee \mathrm{Married}(y,x))\|_{\mathrm{KB}} \\
=\ & \exists x\, \|\mathbf{K}\, \exists y\, (\mathrm{Married}(x,y) \vee \mathrm{Married}(y,x))\|_{\mathrm{KB}} \\
=\ & \exists x\, \mathsf{RES}[\mathrm{KB}, \|\exists y\, (\mathrm{Married}(x,y) \vee \mathrm{Married}(y,x))\|_{\mathrm{KB}}] \\
=\ & \exists x\, \mathsf{RES}[\mathrm{KB}, \exists y\, (\mathrm{Married}(x,y) \vee \mathrm{Married}(y,x))] \\
=\ & \exists x\, (x = \mathrm{Mia} \vee x = \mathrm{Frank} \vee x = \mathrm{Fred})
\end{aligned}
$$

(c) Who is known to be married to an unknown person?

1. $N = \{(\mathrm{Mia}, \mathrm{Frank})\}$
2. $N = \{\mathrm{Fred}\}$
3. The standard names that occur in the KB and the query are $\{\mathrm{Mia}, \mathrm{Frank}, \mathrm{Fred}, \mathrm{Sally}\}$.

$$
\begin{aligned}
& \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(x,y)] \\
=\ & (x = \mathrm{Mia} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Mia}, y)]) \vee \\
& (x = \mathrm{Frank} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Frank}, y)]) \vee \\
& (x = \mathrm{Fred} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Fred}, y)]) \vee \\
& (x = \mathrm{Sally} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Sally}, y)]) \vee \\
& (x \neq \mathrm{Mia} \wedge x \neq \mathrm{Frank} \wedge x \neq \mathrm{Fred} \wedge x \neq \mathrm{Sally} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, y)]_x^{\#7}) \\
=\ & (x = \mathrm{Mia} \wedge y = \mathrm{Frank}) \vee \\
& (x = \mathrm{Frank} \wedge \textsc{false}) \vee \\
& (x = \mathrm{Fred} \wedge \textsc{false}) \vee \\
& (x = \mathrm{Sally} \wedge \textsc{false}) \vee \\
& (x \neq \mathrm{Mia} \wedge x \neq \mathrm{Frank} \wedge x \neq \mathrm{Fred} \wedge x \neq \mathrm{Sally} \wedge \textsc{false}_x^{\#7})
\end{aligned}
$$

because the recursive calls to $\mathsf{RES}[\mathrm{KB}, \mathrm{Married}(n, y)]$ yield the following:

$$
\begin{aligned}
& \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Mia}, y)] \\
=\ & (y = \mathrm{Mia} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Mia}, \mathrm{Mia})]) \vee \\
& (y = \mathrm{Frank} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Mia}, \mathrm{Frank})]) \vee \\
& (y = \mathrm{Fred} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Mia}, \mathrm{Fred})]) \vee \\
& (y = \mathrm{Sally} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Mia}, \mathrm{Sally})]) \vee \\
& (y \neq \mathrm{Mia} \wedge y \neq \mathrm{Frank} \wedge y \neq \mathrm{Fred} \wedge y \neq \mathrm{Sally} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\mathrm{Mia}, \#8)]_y^{\#8}) \\
\overset{\mathsf{simpl}}{=}\ & (y = \mathrm{Frank})
\end{aligned}
$$

for $n \in \{\mathrm{Frank}, \mathrm{Fred}, \mathrm{Sally}\}$:

$$
\begin{aligned}
& \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(n, y)] \\
=\ & (y = \mathrm{Mia} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(n, \mathrm{Mia})]) \vee \\
& (y = \mathrm{Frank} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(n, \mathrm{Frank})]) \vee \\
& (y = \mathrm{Fred} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(n, \mathrm{Fred})]) \vee \\
& (y = \mathrm{Sally} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(n, \mathrm{Sally})]) \vee \\
& (y \neq \mathrm{Mia} \wedge y \neq \mathrm{Frank} \wedge y \neq \mathrm{Fred} \wedge y \neq \mathrm{Sally} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(n, \#8)]_y^{\#8}) \\
\overset{\mathsf{simpl}}{=}\ & \textsc{false}
\end{aligned}
$$

$$
\begin{aligned}
& \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, y)] \\
=\ & (y = \mathrm{Mia} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, \mathrm{Mia})]) \vee \\
& (y = \mathrm{Frank} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, \mathrm{Frank})]) \vee \\
& (y = \mathrm{Fred} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, \mathrm{Fred})]) \vee \\
& (y = \mathrm{Sally} \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, \mathrm{Sally})]) \vee \\
& (y = \#7 \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, \#7)]) \vee \\
& (y \neq \mathrm{Mia} \wedge y \neq \mathrm{Frank} \wedge y \neq \mathrm{Fred} \wedge y \neq \mathrm{Sally} \wedge y \neq \#7 \wedge \mathsf{RES}[\mathrm{KB}, \mathrm{Married}(\#7, \#8)]_y^{\#8})
\end{aligned}
$$

$\stackrel{\text{simpl}}{=}$ FALSE[2]

$\stackrel{\text{simpl}}{=} (x = \text{Mia} \wedge y = \text{Frank})$

4. The standard names that occur in the KB and the query are $\{\text{Mia}, \text{Frank}, \text{Fred}, \text{Sally}\}$.

$$
\begin{aligned}
& \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, y) \wedge \neg(x = \text{Mia} \wedge y = \text{Frank}))] \\
= \; & (y = \text{Mia} \wedge \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Mia}) \wedge \neg(x = \text{Mia} \wedge \text{Mia} = \text{Frank}))]) \vee \\
& (y = \text{Frank} \wedge \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Frank}) \wedge \neg(x = \text{Mia} \wedge \text{Frank} = \text{Frank}))]) \vee \\
& (y = \text{Fred} \wedge \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Fred}) \wedge \neg(x = \text{Mia} \wedge \text{Fred} = \text{Frank}))]) \vee \\
& (y = \text{Sally} \wedge \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Sally}) \wedge \neg(x = \text{Mia} \wedge \text{Sally} = \text{Frank}))]) \vee \\
& (y \neq \text{Mia} \wedge y \neq \text{Frank} \wedge y \neq \text{Fred} \wedge y \neq \text{Sally} \wedge \\
& \qquad \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \#8) \wedge \neg(x = \text{Mia} \wedge \#8 = \text{Frank}))]^{\#8}_{y}) \\
= \; & (y = \text{Mia} \wedge \text{FALSE}) \vee \\
& (y = \text{Frank} \wedge \text{FALSE}) \vee \\
& (y = \text{Fred} \wedge \text{TRUE}) \vee \\
& (y = \text{Sally} \wedge \text{FALSE}) \vee \\
& (y \neq \text{Mia} \wedge y \neq \text{Frank} \wedge y \neq \text{Fred} \wedge y \neq \text{Sally} \wedge \text{FALSE}^{\#8}_{y})
\end{aligned}
$$

because the recursive calls to $\mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, n) \wedge \neg(x = \text{Mia} \wedge n = \text{Frank}))]$ yield the following:

$$
\begin{aligned}
& \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Mia}) \wedge \neg(x = \text{Mia} \wedge \text{Mia} = \text{Frank}))] \\
= \; & \text{FALSE (because KB} \not\models \exists x\, \text{Married}(x, \text{Mia})) \\
& \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Frank}) \wedge \neg(x = \text{Mia} \wedge \text{Frank} = \text{Frank}))] \\
= \; & \text{FALSE (because KB} \not\models \exists x\, (\text{Married}(x, \text{Frank}) \wedge x \neq \text{Mia})) \\
& \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Fred}) \wedge \neg(x = \text{Mia} \wedge \text{Fred} = \text{Frank}))] \\
= \; & \text{TRUE (because KB} \models \exists x\, (\text{Married}(x, \text{Mia})) \\
& \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \text{Sally}) \wedge \neg(x = \text{Mia} \wedge \text{Sally} = \text{Frank}))] \\
= \; & \text{FALSE (because KB} \not\models \exists x\, \text{Married}(x, \text{Sally})) \\
& \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, \#8) \wedge \neg(x = \text{Mia} \wedge \#8 = \text{Frank}))] \\
= \; & \text{FALSE (because KB} \not\models \exists x\, \text{Married}(x, \#8))
\end{aligned}
$$

$\stackrel{\text{simpl}}{=} y = \text{Fred}$

5. $\models \|\exists y\, \mathbf{K}\, \exists x\, (\text{Married}(x, y) \wedge \neg \mathbf{K}\, \text{Married}(x, y))\|_{\text{KB}}$ because:

$$
\begin{aligned}
& \|\exists y\, \mathbf{K}\, \exists x\, (\text{Married}(x, y) \wedge \neg \mathbf{K}\, \text{Married}(x, y))\|_{\text{KB}} \\
= \; & \exists y\, \|\mathbf{K}\, \exists x\, (\text{Married}(x, y) \wedge \neg \mathbf{K}\, \text{Married}(x, y))\|_{\text{KB}} \\
= \; & \exists y\, \mathsf{RES}[\text{KB}, \|\exists x\, (\text{Married}(x, y) \wedge \neg \mathbf{K}\, \text{Married}(x, y))\|_{\text{KB}}] \\
= \; & \exists y\, \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, y) \wedge \neg \|\mathbf{K}\, \text{Married}(x, y)\|_{\text{KB}})] \\
= \; & \exists y\, \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, y) \wedge \neg \mathsf{RES}[\text{KB}, \|\text{Married}(x, y)\|_{\text{KB}}])] \\
= \; & \exists y\, \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, y) \wedge \neg \mathsf{RES}[\text{KB}, \text{Married}(x, y)])] \\
= \; & \exists y\, \mathsf{RES}[\text{KB}, \exists x\, (\text{Married}(x, y) \wedge \neg(x = \text{Mia} \wedge y = \text{Frank}))] \\
= \; & \exists y\, y = \text{Fred}
\end{aligned}
$$

[2] Note that when determining $\mathsf{RES}[\text{KB}, \text{Married}(\#7, y)]$, the standard names occuring in the KB and in the query are $\{\text{Mia}, \text{Frank}, \text{Fred}, \text{Sally}, \#7\}$. Hence $\mathsf{RES}[\text{KB}, \text{Married}(\#7, y)]$ does mention #7 as well – it is only our simplification to FALSE that made #7 disappear. In any case, #7 does not appear in $\mathsf{RES}[\text{KB}, \text{Married}(x, y)]$, because it substitutes x back for #7 after the recursive call to $\mathsf{RES}[\text{KB}, \text{Married}(\#7, y)]$ by writing $\mathsf{RES}[\text{KB}, \text{Married}(\#7, y)]^{\#7}_{x}$.

3 Limited Reasoning

3.1 Unit Propagation and Subsumption

Determine $\mathrm{UP}(s)$, $\mathsf{UP}^{+}(s)$, $\mathsf{UP}^{-}(s)$, whether s is obviously inconsistent, and whether s is obviously consistent, for...

1. $s = \{\}$
2. $s = \{p, \neg p\}$
3. $s = \{(p \lor q), (\neg q \lor \neg r), r\}$
4. $s = \{(p \lor q), (p \lor \neg q), (\neg p \lor q), (\neg p \lor \neg q)\}$

Solution

1. $s = \{\}$
   - $\mathrm{UP}(s) = \{\}$
   - $\mathsf{UP}^{+}(s) = \{\}$
   - $\mathsf{UP}^{-}(s) = \{\}$
   - s is not obviously inconsistent because $\Box \notin \mathrm{UP}(s)$
   - s is obviously consistent because $\mathsf{UP}^{-}(s)$ doesn't contain $\Box$ nor it mentions any P and $\neg P$
2. $s = \{p, \neg p\}$
   - $\mathrm{UP}(s) = \{\Box, p, \neg p\}$
   - $\mathsf{UP}^{+}(s) = \{c \mid c \text{ is a clause}\}$ because $\Box \in \mathrm{UP}(s)$ subsumes every clause (recall that we identify a clause $(\ell_1 \lor \ldots \lor \ell_k)$ with the set $\{\ell_1, \ldots, \ell_k\}$, so the empty clause corresponds to the empty set, and a clause c_1 subsumes a clause c_2 iff $c_1 \subseteq c_2$, e.g., $(p \lor q)$ subsumes $(p \lor q \lor r)$, but $(p \lor q)$ does not subsume $(\neg p \lor \neg q)$).
   - $\mathsf{UP}^{-}(s) = \{\Box\}$ (because we remove all the clauses from $\mathrm{UP}(s)$ that are subsumed by another clause, and here p and $\neg p$ both are subsumed by $\Box$)
   - s is obviously inconsistent because $\Box \in \mathrm{UP}(s)$
   - s is not obviously consistent because $\Box \in \mathsf{UP}^{-}(s)$[3]
3. $s = \{(p \lor q), (\neg q \lor \neg r), r\}$
   - $\mathrm{UP}(s) = \{r, \neg q, p, (p \lor q), (\neg q \lor \neg r)\}$
   - $\mathsf{UP}^{+}(s) = \{r, \neg q, p\} \cup \{c \mid c \text{ contains } r, \neg q, \text{ or } p\}$
   - $\mathsf{UP}^{-}(s) = \{r, \neg q, p\}$
   - s is not obviously inconsistent because $\Box \notin \mathrm{UP}(s)$
   - s is obviously consistent because $\mathsf{UP}^{-}(s)$ doesn't contain $\Box$ nor it mentions any P and $\neg P$
4. $s = \{(p \lor q), (p \lor \neg q), (\neg p \lor q), (\neg p \lor \neg q)\}$
   - $\mathrm{UP}(s) = \{(p \lor q), (p \lor \neg q), (\neg p \lor q), (\neg p \lor \neg q)\}$

[3] The definition of *obviously consistent* on Slide 25 was missing a case for the empty clause: when the emptpy clause is in $\mathsf{UP}^{-}(s)$, then s is not obviously consistent (and in fact s is obviously inconsistent). The slide has been updated.

- $\mathsf{UP}^{+}(s) = \{(p\vee q), (p\vee\neg q), (\neg p\vee q), (\neg p\vee\neg q), (p\vee q\vee r), (p\vee\neg q\vee r), (\neg p\vee q\vee r), (\neg p\vee\neg q\vee r), \ldots\} = \{c \mid p, q \in c \text{ or } p, \neg q \in c \text{ or } \neg p, q \in c \text{ or } \neg p, \neg q \in c\}$
- $\mathsf{UP}^{-}(s) = \{(p \vee q), (p \vee \neg q), (\neg p \vee q), (\neg p \vee \neg q)\}$
- s is not obviously inconsistent because $\Box \notin \mathrm{UP}(s)$
- s is not obviously consistent because $\mathsf{UP}^{-}(s)$ mentions p and $\neg p$ (as well as q and $\neg q$)

3.2 Minimal Belief Level

1. Let $s = \{\}$. Find the minimal k such that $s \mathrel{|\!\approx} \mathbf{K}_k(p \vee \neg p)$.
2. Let $s = \{p, \neg p\}$. Find the minimal k such that $s \mathrel{|\!\approx} \mathbf{K}_k q$.
3. Let $s = \{(p \vee q), (\neg p \vee r)\}$. Find the minimal k such that $s \mathrel{|\!\approx} \mathbf{K}_k(q \vee r)$.
4. Let $s = \{(o \vee p \vee r), (o \vee \neg p \vee r), (\neg o \vee q), (\neg o \vee \neg q)\}$. Find the minimal k such that $s \mathrel{|\!\approx} \mathbf{K}_k r$.

Solution

1. Let $s = \{\}$.
   - $k = 0$:

     $s \mathrel{|\!\approx} \mathbf{K}_0(p \vee \neg p)$
     iff s is obv. inconsistent or $s \mathrel{|\!\approx} (p \vee \neg p)$
     iff s is obv. inconsistent or $(p \vee \neg p) \in \mathsf{UP}^{+}(s)$
     ✗
   - $k = 1$:

     $s \mathrel{|\!\approx} \mathbf{K}_1(p \vee \neg p)$
     iff for some proposition P,
     $s \cup \{P\} \mathrel{|\!\approx} \mathbf{K}_0(p \vee \neg p)$ and $s \cup \{\neg P\} \mathrel{|\!\approx} \mathbf{K}_0(p \vee \neg p)$
     if (split on p)
     $s \cup \{p\} \mathrel{|\!\approx} \mathbf{K}_0(p \vee \neg p)$ and $s \cup \{\neg p\} \mathrel{|\!\approx} \mathbf{K}_0(p \vee \neg p)$
     iff $s \cup \{p\}$ is obv. incons. or $s \cup \{p\} \mathrel{|\!\approx} (p \vee \neg p)$, and
     $s \cup \{\neg p\}$ is obv. incons. or $s \cup \{\neg p\} \mathrel{|\!\approx} (p \vee \neg p)$
     iff $s \cup \{p\}$ is obv. incons. or $(p \vee \neg p) \in \mathsf{UP}^{+}(s \cup \{p\})$, and
     $s \cup \{\neg p\}$ is obv. incons. or $(p \vee \neg p) \in \mathsf{UP}^{+}(s \cup \{\neg p\})$
     ✓ (because p and $\neg p$ both subsume $(p \vee \neg q)$)
2. Let $s = \{p, \neg p\}$.
   - $k = 0$:

     $s \mathrel{|\!\approx} \mathbf{K}_0 q$
     iff s is obv. inconsistent or $s \mathrel{|\!\approx} q$
     ✓ (because $\mathsf{UP}^{+}(s)$ contains the empty clause)
3. Let $s = \{(p \vee q), (\neg p \vee r)\}$.
   - $k = 0$:

     $s \mathrel{|\!\approx} \mathbf{K}_0(q \vee r)$

iff s is obv. inconsistent or $s \models\!\!\approx (q \vee r)$

✗

- $k = 1$:

$s \models\!\!\approx \mathbf{K}_1(q \vee r)$

iff for some proposition P,
$s \cup \{P\} \models\!\!\approx \mathbf{K}_0(q \vee r)$ and $s \cup \{\neg P\} \models\!\!\approx \mathbf{K}_0(q \vee r)$

iff for some proposition P,
$s \cup \{P\}$ is obv. inconsistent or $(p \vee q) \in \mathsf{UP}^+(s \cup \{P\})$, and
$s \cup \{\neg P\}$ is obv. inconsistent or $(p \vee q) \in \mathsf{UP}^+(s \cup \{\neg P\})$

if (split on p)
$s \cup \{p\}$ is obv. inconsistent or $(p \vee q) \in \mathsf{UP}^+(s \cup \{p\})$, and
$s \cup \{\neg p\}$ is obv. inconsistent or $(p \vee q) \in \mathsf{UP}^+(s \cup \{\neg p\})$

✓ (because $r \in \mathsf{UP}^+(s \cup \{p\})$ subsumes $(q \vee r)$, and $q \in \mathsf{UP}^+(s \cup \{\neg p\})$ subsumes $(q \vee r)$)

4. Let $s = \{(o \vee p \vee r), (o \vee \neg p \vee r), (\neg o \vee q), (\neg o \vee \neg q)\}$.

- $k = 0$:

$s \models\!\!\approx \mathbf{K}_0 r$

iff s is obv. inconsistent or $s \models\!\!\approx r$

✗

- $k = 1$:

$s \models\!\!\approx \mathbf{K}_1 r$

iff for some proposition P,
$s \cup \{P\} \models\!\!\approx \mathbf{K}_0 r$ and
$s \cup \{\neg P\} \models\!\!\approx \mathbf{K}_0 r$

iff for some proposition P,
$s \cup \{P\}$ is obv. incons. or $s \cup \{P\} \models\!\!\approx r$ and
$s \cup \{\neg P\}$ is obv. incons. or $s \cup \{\neg P\} \models\!\!\approx r$

✗ (splitting on o gives us the empty clause in the o-branch, but in the $\neg o$-branch unit propagation stops after producing $(p \vee r), (\neg p \vee r)$; splitting on p only gives us $(o \vee r)$ in either branch but nothing further; splitting on q produces $\neg o$ and $(p \vee r), (\neg p \vee r)$ in either branch but nothing more)

- $k = 2$:

$s \models\!\!\approx \mathbf{K}_2 r$

iff for some proposition P_1,
$s \cup \{P_1\} \models\!\!\approx \mathbf{K}_1 r$ and
$s \cup \{\neg P_1\} \models\!\!\approx \mathbf{K}_1 r$

iff for some proposition P_1,
for some proposition P_2,
$s \cup \{P_1, P_2\} \models\!\!\approx \mathbf{K}_0 r$ and
$s \cup \{P_1, \neg P_2\} \models\!\!\approx \mathbf{K}_0 r$
and
for some proposition P_2,
$s \cup \{\neg P_1, P_2\} \models\!\!\approx \mathbf{K}_0 r$ and
$s \cup \{\neg P_1, \neg P_2\} \models\!\!\approx \mathbf{K}_0 r$

iff for some proposition P_1,

for some proposition P_2,
$s \cup \{P_1, P_2\}$ is obv. incons. or $s \cup \{P_1, P_2\} \not\approx r$ and
$s \cup \{P_1, \neg P_2\}$ is obv. incons. or $s \cup \{P_1, \neg P_2\} \not\approx r$
and
for some proposition P_2,
$s \cup \{\neg P_1, P_2\}$ is obv. incons. or $s \cup \{\neg P_1, P_2\} \not\approx r$ and
$s \cup \{\neg P_1, \neg P_2\}$ is obv. incons. or $s \cup \{\neg P_1, \neg P_2\} \not\approx r$

if (split on o first; in the positive case, we're done; in the negative case, split on p next)
for arbitrary P_2,
$s \cup \{o, P_2\}$ is obv. incons. or $r \in \mathsf{UP}^+(s \cup \{o, P_2\})$ and
$s \cup \{o, \neg P_2\}$ is obv. incons. or $r \in \mathsf{UP}^+(s \cup \{o, P_2\})$
and
$r \in \mathsf{UP}^+(s \cup \{\neg o, p\})$ or $r \in \mathsf{UP}^+(s \cup \{\neg o, p\})$ and
$r \in \mathsf{UP}^+(s \cup \{\neg o, \neg p\})$ or $r \in \mathsf{UP}^+(s \cup \{\neg o, \neg p\})$

✓ (because $s \cup \{o\}$ is obviously inconsistent, and $\mathsf{UP}^+(s \cup \{\neg o, p\})$ and $\mathsf{UP}^+(s \cup \{\neg o, \neg p\})$ both include r)

4 Reasoning about Actions

4.1 Basic Action Theories

- Consider a light switch. Model that the fluent LightOn is toggled by an action switch.
- Consider some object that may contain other objects. Setting the containing object alight also sets alight the objects in the box. Model a Burning(x) fluent using an action setAlight(x) and another predicate In(x, y) that indicates that x is in y.
- You're participating in a drug trial: you're sick; you take a some medication, which may be placebo or not; and you see whether or not you feel better afterwards. Model the Sick fluent, which is "disabled" when you take medication x, represented by action take(x), provided that x is not placebo, that is, ¬Placebo(x). Also model the sensing axiom for the feel action, which shall tell you whether you're still sick or not.

Solution

- $\Box[a]\mathrm{LightOn} \leftrightarrow (a = \mathrm{switch} \wedge \neg\mathrm{LightOn}) \vee (\mathrm{LightOn} \wedge a \neq \mathrm{switch})$
  - Positive effect axiom: $a = \mathrm{switch} \wedge \neg\mathrm{LightOn} \rightarrow [a]\mathrm{LightOn}$
  - Negative effect axiom: $a = \mathrm{switch} \wedge \mathrm{LightOn} \rightarrow [a]\neg\mathrm{LightOn}$
  - Succ.-state axiom: $\Box[a]\mathrm{LightOn} \leftrightarrow (a{=}\mathrm{switch} \wedge \neg\mathrm{LightOn}) \vee (\mathrm{LightOn} \wedge \neg(a{=}\mathrm{switch} \wedge \mathrm{LightOn}))$
    simplies to: $\Box[a]\mathrm{LightOn} \leftrightarrow (a = \mathrm{switch} \wedge \neg\mathrm{LightOn}) \vee (\mathrm{LightOn} \wedge (a \neq \mathrm{switch} \vee \neg\mathrm{LightOn}))$
    simplies to: $\Box[a]\mathrm{LightOn} \leftrightarrow (a = \mathrm{switch} \wedge \neg\mathrm{LightOn}) \vee (\mathrm{LightOn} \wedge a \neq \mathrm{switch})$
- $\Box[a]\mathrm{Burning}(x) \leftrightarrow a = \mathrm{setAlight}(x) \vee \exists y\, (a = \mathrm{setAlight}(y) \wedge \mathrm{In}(x, y)) \vee \mathrm{Burning}(x)$
  - Positive effect axiom: $a = \mathrm{setAlight}(x) \vee \exists y\, (a = \mathrm{setAlight}(y) \wedge \mathrm{In}(x, y)) \rightarrow \mathrm{Burning}(x)$
  - Negative effect axiom: $\textsc{false} \rightarrow [a]\neg\mathrm{Burning}(x)$ (the question doesn't say anything about ways to put out a fire)

    - Succ.-state axiom: $\Box[a]\text{Burning}(x) \leftrightarrow (a = \text{setAlight}(x) \lor \exists y\,(a = \text{setAlight}(y) \land \text{In}(x,y))) \lor (\text{Burning}(x) \land \neg\textsc{false})$
      simplies to: $\Box[a]\text{Burning}(x) \leftrightarrow a = \text{setAlight}(x) \lor \exists y\,(a = \text{setAlight}(y) \land \text{In}(x,y)) \lor \text{Burning}(x)$

- $\Box[a]\text{Sick} \leftrightarrow \text{Sick} \land \forall x\,(a \neq \text{take}(x) \lor \text{Placebo}(x))$
    - Positive effect axiom: $\textsc{false} \rightarrow [a]\neg\text{Sick}$
    - Negative effect axiom: $\exists x\,(a = \text{take}(x) \land \neg\text{Placebo}(x)) \rightarrow [a]\neg\text{Sick}$
    - Succ.-state axiom: $\Box[a]\text{Sick} \leftrightarrow \textsc{false} \lor (\text{Sick} \land \neg(\exists x\,(a = \text{take}(x) \land \neg\text{Placebo}(x))))$
      simplifies: $\Box[a]\text{Sick} \leftrightarrow \textsc{false} \lor (\text{Sick} \land (\forall x\,(a \neq \text{take}(x) \lor \text{Placebo}(x))))$
      simplifies: $\Box[a]\text{Sick} \leftrightarrow \text{Sick} \land (\forall x\,(a \neq \text{take}(x) \lor \text{Placebo}(x)))$

  $\Box\text{SF}(a) \leftrightarrow (a = \text{feel} \rightarrow \text{Sick})$

4.2 Regression

Consider the following basic action theory, where γ and φ are the right-hand sides of the successor-state axiom of Sick and the axiom for SF from the previous task.

$$\begin{aligned}
\Sigma_0 &= \{\text{Sick} \land \neg\text{Placebo}(\#1) \land \text{Placebo}(\#2)\}\\
\Sigma_1 &= \{\textsc{true}\}\\
\Sigma_{\text{dyn}} &= \{\Box[a]\text{Sick} \leftrightarrow \gamma,\\
&\qquad \Box[a]\text{Placebo}(x) \leftrightarrow \text{Placebo}(x),\\
&\qquad \Box\text{Poss}(a) \leftrightarrow \textsc{true},\\
&\qquad \Box\text{SF}(a) \leftrightarrow \varphi\}
\end{aligned}$$

(a) Prove that $\Sigma_0 \land \Sigma_{\text{dyn}} \models [\text{take}(\#1)]\neg\text{Sick}$ using regression.[4]

(b) Prove that $\Sigma_0 \land \Sigma_{\text{dyn}} \land \mathbf{O}(\Sigma_1 \land \Sigma_{\text{dyn}}) \models [\text{take}(\#1)]\neg\mathbf{K}\,\neg\text{Sick}$.

(c) Prove that $\Sigma_0 \land \Sigma_{\text{dyn}} \land \mathbf{O}(\Sigma_1 \land \Sigma_{\text{dyn}}) \models [\text{take}(\#1)][\text{feel}]\mathbf{K}\,\neg\text{Sick}$.

Solution

(a) By the Theorem from Slide 25, we have

$$\Sigma_0 \land \Sigma_{\text{dyn}} \models [\text{take}(\#1)]\neg\text{Sick} \;\text{ iff }\; \Sigma_0 \models \mathcal{R}[\langle\rangle, [\text{take}(\#1)]\neg\text{Sick}]$$

Let's determine the regression first:

$$\begin{aligned}
&\mathcal{R}[\langle\rangle, [\text{take}(\#1)]\neg\text{Sick}]\\
=\;& \mathcal{R}[\text{take}(\#1), \neg\text{Sick}]\\
=\;& \neg\mathcal{R}[\text{take}(\#1), \text{Sick}]\\
=\;& \neg\mathcal{R}[\langle\rangle, \gamma^a_{\text{take}(\#1)}]\\
=\;& \neg\mathcal{R}[\langle\rangle, (\text{Sick} \land \forall x\,(a \neq \text{take}(x) \lor \text{Placebo}(x)))^a_{\text{take}(\#1)}]
\end{aligned}$$

[4] We defined Σ_0 and Σ_{dyn} as sets of sentences. We identify such a set of sentences with the conjunction of its elements. That is, writing $\Sigma_0 \land \Sigma_{\text{dyn}} \models \alpha$ stands for $\bigwedge_{\phi\in\Sigma_0} \phi \land \bigwedge_{\psi\in\Sigma_{\text{dyn}}} \psi \models \alpha$.

$$= \neg\mathcal{R}[\langle\rangle, (\text{Sick} \wedge \forall x\, (\text{take}(\#1) \neq \text{take}(x) \vee \text{Placebo}(x)))]$$

$$= \neg(\mathcal{R}[\langle\rangle, \text{Sick}] \wedge \forall x\, (\mathcal{R}[\langle\rangle, \text{take}(\#1) \neq \text{take}(x)] \vee \mathcal{R}[\langle\rangle, \text{Placebo}(x)]))$$

$$= \neg(\text{Sick} \wedge \forall x\, (\text{take}(\#1) \neq \text{take}(x) \vee \text{Placebo}(x)))$$

$$\overset{\text{simpl}}{=} (\neg\text{Sick} \vee \neg\forall x\, (\text{take}(\#1) \neq \text{take}(x) \vee \text{Placebo}(x)))$$

$$\overset{\text{simpl}}{=} (\neg\text{Sick} \vee \exists x\, (\text{take}(\#1) = \text{take}(x) \wedge \neg\text{Placebo}(x)))$$

$$\overset{\text{simpl}}{=} (\neg\text{Sick} \vee \exists x\, (\#1 = x \wedge \neg\text{Placebo}(x)))$$

$$\overset{\text{simpl}}{=} (\neg\text{Sick} \vee \neg\text{Placebo}(\#1))$$

So we only need to prove that $\Sigma_0 \models (\neg\text{Sick} \vee \neg\text{Placebo}(\#1))$, that is, for all w, if $w \models \Sigma_0$, then $w \models (\neg\text{Sick} \vee \neg\text{Placebo}(\#1))$, which holds since for every w such that $w \models \Sigma_0$, we have $w \models \neg\text{Placebo}(\#1)$, and hence $w \models (\neg\text{Sick} \vee \neg\text{Placebo}(\#1))$.

(b) By the Theorem from Slide 35, we have

$$\Sigma_0 \wedge \Sigma_{\text{dyn}} \wedge \mathbf{O}(\Sigma_1 \wedge \Sigma_{\text{dyn}}) \models [\text{take}(\#1)]\neg\mathbf{K}\neg\text{Sick} \;\text{ iff }\; \Sigma_0 \wedge \mathbf{O}\Sigma_1 \models \mathcal{R}[\langle\rangle, [\text{take}(\#1)]\neg\mathbf{K}\neg\text{Sick}]$$

Let's determine the regression first:

$$\mathcal{R}[\langle\rangle, [\text{take}(\#1)]\neg\mathbf{K}\neg\text{Sick}]$$

$$= \neg\mathcal{R}[\text{take}(\#1), \mathbf{K}\neg\text{Sick}]$$

$$\begin{aligned} = \neg(&\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1)) \to \mathbf{K}(\text{SF}(\text{take}(\#1)) \to [\text{take}(\#1)]\neg\text{Sick})] \wedge \\ &\mathcal{R}[\langle\rangle, \neg\text{SF}(\text{take}(\#1)) \to \mathbf{K}(\neg\text{SF}(\text{take}(\#1)) \to [\text{take}(\#1)]\neg\text{Sick})]) \end{aligned}$$

$$\begin{aligned} = \neg(&\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathcal{R}[\langle\rangle, \mathbf{K}(\text{SF}(\text{take}(\#1)) \to [\text{take}(\#1)]\neg\text{Sick})] \wedge \\ &\neg\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathcal{R}[\langle\rangle, \mathbf{K}(\neg\text{SF}(\text{take}(\#1)) \to [\text{take}(\#1)]\neg\text{Sick})]) \end{aligned}$$

$$\begin{aligned} = \neg(&\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathbf{K}(\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathcal{R}[\langle\rangle, [\text{take}(\#1)]\neg\text{Sick}]) \wedge \\ &\neg\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathbf{K}(\neg\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathcal{R}[\langle\rangle, [\text{take}(\#1)]\neg\text{Sick}])) \end{aligned}$$

$$\begin{aligned} = \neg(&\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathbf{K}(\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \neg\mathcal{R}[\text{take}(\#1), \text{Sick}]) \wedge \\ &\neg\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathbf{K}(\neg\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \neg\mathcal{R}[\text{take}(\#1), \text{Sick}])) \end{aligned}$$

$$\begin{aligned} = \neg(&\mathcal{R}[\langle\rangle, \varphi^a_{\text{take}(\#1)}] \to \mathbf{K}(\mathcal{R}[\langle\rangle, \varphi^a_{\text{take}(\#1)}] \to \neg\mathcal{R}[\text{take}(\#1), \text{Sick}]) \wedge \\ &\neg\mathcal{R}[\langle\rangle, \varphi^a_{\text{take}(\#1)}] \to \mathbf{K}(\neg\mathcal{R}[\langle\rangle, \varphi^a_{\text{take}(\#1)}] \to \neg\mathcal{R}[\text{take}(\#1), \text{Sick}])) \end{aligned}$$

$$\begin{aligned} = \neg(&(\text{take}(\#1) = \text{feel} \to \text{Sick}) \to \mathbf{K}((\text{take}(\#1) = \text{feel} \to \text{Sick}) \to \neg(\neg\text{Sick} \vee \neg\text{Placebo}(\#1))) \wedge \\ &\neg(\text{take}(\#1) = \text{feel} \to \text{Sick}) \to \mathbf{K}(\neg(\text{take}(\#1) = \text{feel} \to \text{Sick}) \to \neg(\neg\text{Sick} \vee \neg\text{Placebo}(\#1)))) \end{aligned}$$

(by reusing the the result from (a))

$$\overset{\text{simpl}}{=} \neg\mathbf{K}\neg(\neg\text{Sick} \vee \neg\text{Placebo}(\#1))$$

(because $\text{take}(\#1) = \text{feel}$ is unsatisfiable, so $(\text{take}(\#1) = \text{feel} \to \text{Sick})$ is valid, and hence $\neg(\text{take}(\#1) = \text{feel} \to \text{Sick}) \to \mathbf{K} \ldots$ is valid)

$$\overset{\text{simpl}}{=} \neg\mathbf{K}(\text{Sick} \wedge \text{Placebo}(\#1))$$

So we only need to show that $\Sigma_0 \wedge \mathbf{O}\Sigma_1 \models \neg\mathbf{K}(\text{Sick} \wedge \text{Placebo}(\#1))$. Suppose $e, w \models \Sigma_0 \wedge \mathbf{O}\Sigma_1$. Then e is the set of all worlds, since Σ_1 is just TRUE. Clearly there is a world $w \in e$ such that $w \not\models (\text{Sick} \wedge \neg\text{Placebo}(\#1))$. Hence $e \not\models \mathbf{K}(\text{Sick} \wedge \text{Placebo}(\#1))$. So $e \models \neg\mathbf{K}(\text{Sick} \wedge \text{Placebo}(\#1))$.

(c) By the Theorem from Slide 35, we have

$$\Sigma_0 \land \Sigma_{\text{dyn}} \land \mathbf{O}(\Sigma_1 \land \Sigma_{\text{dyn}}) \models [\text{take}(\#1)][\text{feel}]\mathbf{K}\neg\text{Sick} \;\text{ iff }\; \Sigma_0 \land \mathbf{O}\Sigma_1 \models \mathcal{R}[\langle\rangle, [\text{take}(\#1)][\text{feel}]\mathbf{K}\neg\text{Sick}]$$

Let's determine the regression first (to keep the presentation shorter, I'll simplify a bit more aggressively):

$$\begin{aligned}
& \mathcal{R}[\langle\rangle, [\text{take}(\#1)][\text{feel}]\mathbf{K}\neg\text{Sick}] \\
= {} & \mathcal{R}[\text{take}(\#1)\cdot\text{feel}, \mathbf{K}\neg\text{Sick}] \\
= {} & \mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel}) \to \mathbf{K}(\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick})] \land \\
& \mathcal{R}[\text{take}(\#1), \neg\text{SF}(\text{feel}) \to \mathbf{K}(\neg\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick})] \\
= {} & (\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \mathcal{R}[\text{take}(\#1), \mathbf{K}(\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick})]) \land \\
& (\neg\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \mathcal{R}[\text{take}(\#1), \mathbf{K}(\neg\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick})]) \\
= {} & (\text{Sick} \land \text{Placebo}(\#1) \to \mathbf{K}(\neg\text{Sick} \lor \neg\text{Placebo}(\#1))) \land (\neg(\text{Sick} \land \text{Placebo}(\#1)) \to \textsc{true}) \\
& \text{because}
\end{aligned}$$

$$\begin{aligned}
& \mathcal{R}[\text{take}(\#1), \text{Sick}] \\
= {} & \mathcal{R}[\langle\rangle, \text{Sick} \land \forall x\,(\text{take}(\#1) \neq \text{take}(x) \lor \text{Placebo}(x))] \\
= {} & \mathcal{R}[\langle\rangle, \text{Sick}] \land \forall x\,(\text{take}(\#1) \neq \text{take}(x) \lor \mathcal{R}[\langle\rangle, \text{Placebo}(x)]) \\
= {} & \text{Sick} \land \forall x\,(\text{take}(\#1) \neq \text{take}(x) \lor \text{Placebo}(x)) \\
\stackrel{\text{simpl}}{=} {} & \text{Sick} \land \text{Placebo}(\#1)
\end{aligned}$$

$$\begin{aligned}
& \mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \\
= {} & \mathcal{R}[\text{take}(\#1), \text{feel} = \text{feel} \to \text{Sick}] \\
= {} & \text{feel} = \text{feel} \to \mathcal{R}[\text{take}(\#1), \text{Sick}] \\
\stackrel{\text{simpl}}{=} {} & \mathcal{R}[\text{take}(\#1), \text{Sick}] \\
= {} & \text{Sick} \land \text{Placebo}(\#1)
\end{aligned}$$

$$\begin{aligned}
& \mathcal{R}[\text{take}(\#1), \mathbf{K}(\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick})] \\
= {} & \mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1)) \to \mathbf{K}(\text{SF}(\text{take}(\#1)) \to [\text{take}(\#1)](\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick}))] \land \\
& \mathcal{R}[\langle\rangle, \neg\text{SF}(\text{take}(\#1)) \to \mathbf{K}(\neg\text{SF}(\text{take}(\#1)) \to [\text{take}(\#1)](\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick}))] \\
= {} & (\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathbf{K}(\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to (\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg\mathcal{R}[\text{take}(\#1)\cdot\text{feel}, \text{Sick}]))) \land \\
& (\neg\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to \mathbf{K}(\neg\mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1))] \to (\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg\mathcal{R}[\text{take}(\#1)\cdot\text{feel}, \text{Sick}]))) \\
= {} & (\textsc{true} \to \mathbf{K}(\textsc{true} \to (\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg\mathcal{R}[\text{take}(\#1)\cdot\text{feel}, \text{Sick}]))) \land \\
& (\neg\textsc{true} \to \mathbf{K}(\neg\textsc{true} \to (\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg\mathcal{R}[\text{take}(\#1)\cdot\text{feel}, \text{Sick}]))) \\
& \text{because } \mathcal{R}[\langle\rangle, \text{SF}(\text{take}(\#1)] = (\text{take}(\#1) = \text{feel} \to \text{Sick}) \stackrel{\text{simpl}}{=} \textsc{true} \\
\stackrel{\text{simpl}}{=} {} & \mathbf{K}(\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg\mathcal{R}[\text{take}(\#1)\cdot\text{feel}, \text{Sick}]) \\
= {} & \mathbf{K}(\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg\mathcal{R}[\text{take}(\#1), \text{Sick} \land \forall x\,(\text{feel} \neq \text{take}(x) \lor \text{Placebo}(x))]) \\
= {} & \mathbf{K}(\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg(\mathcal{R}[\text{take}(\#1), \text{Sick}] \land \forall x\,(\text{feel} \neq \text{take}(x) \lor \mathcal{R}[\text{take}(\#1), \text{Placebo}(x)]))) \\
\stackrel{\text{simpl}}{=} {} & \mathbf{K}(\mathcal{R}[\text{take}(\#1), \text{SF}(\text{feel})] \to \neg\mathcal{R}[\text{take}(\#1), \text{Sick}]) \\
= {} & \mathbf{K}(\text{Sick} \land \text{Placebo}(\#1) \to \neg(\text{Sick} \land \text{Placebo}(\#1))) \\
\stackrel{\text{simpl}}{=} {} & \mathbf{K}\neg(\text{Sick} \land \text{Placebo}(\#1)) \\
\stackrel{\text{simpl}}{=} {} & \mathbf{K}(\neg\text{Sick} \lor \neg\text{Placebo}(\#1))
\end{aligned}$$

$$\mathcal{R}[\text{take}(\#1), \mathbf{K}(\neg\text{SF}(\text{feel}) \to \neg[\text{feel}]\text{Sick})]$$

$$
\begin{aligned}
&= \text{(analogous to the above)}\\
&= \mathbf{K}\,(\neg(\text{Sick} \land \text{Placebo}(\#1)) \to \neg(\text{Sick} \land \text{Placebo}(\#1)))\\
&\overset{\text{simpl}}{=} \mathbf{K}\,\text{TRUE}\\
&\overset{\text{simpl}}{=} \text{TRUE}
\end{aligned}
$$

$$\overset{\text{simpl}}{=} \text{Sick} \land \text{Placebo}(\#1) \to \mathbf{K}\,(\neg\text{Sick} \lor \neg\text{Placebo}(\#1))$$

It remains to be shown that $\Sigma_0 \land \mathbf{O}\Sigma_1 \models \text{Sick} \land \text{Placebo}(\#1) \to \mathbf{K}\,(\neg\text{Sick} \lor \neg\text{Placebo}(\#1))$. Suppose $e, w \models \Sigma_0 \land \mathbf{O}\Sigma_1$. Then $w \models \text{Sick} \land \neg\text{Placebo}(\#1)$. Thus $w \not\models \text{Sick} \land \text{Placebo}(\#1)$, and hence $e, w \models \text{Sick} \land \text{Placebo}(\#1) \to \mathbf{K}\,(\neg\text{Sick} \lor \neg\text{Placebo}(\#1))$.

4.3 Knowledge after Actions

Prove the theorem from Slide 34, which is crucial for the regression of knowledge:

$$
\begin{aligned}
\models \Box[a]\mathbf{K}\,\alpha \leftrightarrow\ &(\text{SF}(a) \to \mathbf{K}\,(\text{SF}(a) \to [a]\alpha)) \land\\
&(\neg\text{SF}(a) \to \mathbf{K}\,(\neg\text{SF}(a) \to [a]\alpha))
\end{aligned}
$$

Solution

Theorem

$$
\begin{aligned}
\models \Box[a]\mathbf{K}\,\alpha \leftrightarrow\ &(\text{SF}(a) \to \mathbf{K}\,(\text{SF}(a) \to [a]\alpha)) \land\\
&(\neg\text{SF}(a) \to \mathbf{K}\,(\neg\text{SF}(a) \to [a]\alpha))
\end{aligned}
$$

Proof. We need to show that $e, w, z \models [n]\mathbf{K}\,\alpha \leftrightarrow (\text{SF}(n) \to \mathbf{K}\,(\text{SF}(n) \to [n]\alpha)) \land (\neg\text{SF}(n) \to \mathbf{K}\,(\neg\text{SF}(n) \to [n]\alpha))$ for all e, w, z.

For the only-if direction suppose $e, w, z \models [n]\mathbf{K}\,\alpha$. Then by the rules for $[n]$ and $\mathbf{K}$, for all $w' \in e$ with $w \simeq_{z\cdot n} w'$, $e, w', z \cdot n \models \alpha$. Suppose $w[\text{SF}(n)] = 1$ (the case for $w[\text{SF}(n)] = 0$ is analogous). By definition of $\simeq_{z\cdot n}$ and by the rule for $[n]$, for all $w' \in e$ with $w \simeq_z w'$ and $w'[\text{SF}(n)] = 1$, $e, w', z \models [n]\alpha$. Then for all $w' \in e$ with $w \simeq_z w'$, $e, w', z \models \text{SF}(n) \to [n]\alpha$. By the rule for $\mathbf{K}$, $e, w, z \models \mathbf{K}\,(\text{SF}(n) \to [n]\alpha)$. Thus and since $w[\text{SF}(n)] = 1$, the right-hand side holds.

Conversely, suppose $e, w, z \models (\text{SF}(n) \to \mathbf{K}\,(\text{SF}(n) \to [n]\alpha)) \land (\neg\text{SF}(n) \to \mathbf{K}\,(\neg\text{SF}(n) \to [n]\alpha))$. Suppose $w \models \text{SF}(n)$ (the case for $w \models \neg\text{SF}(n)$ is analogous). Then $e, w, z \models \mathbf{K}\,(\text{SF}(n) \to [n]\alpha)$. By the rule for $\mathbf{K}$, for all $w' \in e$ with $w \simeq_z w'$, $e, w', z \models \text{SF}(n) \to [n]\alpha$. Then for all $w' \in e$ with $w \simeq_z w'$ and $w[\text{SF}(n)] = 1$, $e, w', z \models [n]\alpha$. By definition of $\simeq_{z\cdot n}$ and by the rule for $[n]$, for all $w' \in e$ with $w \simeq_{z\cdot n} w'$, $e, w', z \cdot n \models \alpha$. By the rule for $\mathbf{K}$, $e, w, z \cdot n \models \mathbf{K}\,\alpha$. By the rule for $[n]$, $e, w, z \models [n]\mathbf{K}\,\alpha$, so the left-hand side holds. $\Box$