

Group Knowing-How

Raul Fervari^{1,2}, Rustam Galimullin³, and Fernando R. Velázquez-Quesada³

¹ Universidad Nacional de Córdoba, Argentina

² Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

³ University of Bergen, Bergen, Norway

Abstract Knowing-how is an epistemic notion that captures the abilities of agents to achieve a certain goal. Most of logical approaches formalising knowing-how deal with abilities of individual agents, with only a few exceptions discussing the abilities of groups. This paper focuses on the latter, presenting a logical framework in which the abilities a group are given in terms of the *sequential composition* of the abilities of its members. We discuss three ways to capture this form of group knowing-how, studying their relative expressive power and model checking procedures.

Keywords: Knowing-How, Group Knowledge, Model-Checking, Expressivity

1 Introduction

The concept of *knowing-how*/*know-how*, related to the *ability*⁴ to achieve a given goal, has been extensively discussed in the literature. While it was first approached from a philosophical perspective (see, e.g., [32] and references therein), in recent years it has also become an important topic for computer science and, in particular, for AI (e.g., to account for automated planning and strategic reasoning, [36,2]).

Logical frameworks for representing and reasoning about this concept have typically followed two main directions. On the one hand, there is a long tradition of research on logics for *strategic reasoning*, with prime examples being Coalition Logic (CL) [31], Alternating-Time Temporal Logic (ATL) [4,12], Strategy Logic (SL) [29], and their numerous variations (see, e.g., [37,1,30,3,27,10,8]). Semantically, these proposals treat transitions between states in a model as *joint actions* whose outcome depends on the individual choices of strategies of the agents. Then, a group has the ability to guarantee an outcome if and only if its members have a joint strategy (consisting of individual strategies for each member) that produces the outcome regardless of what the rest of the agents

⁴ In the philosophical literature, the concepts of *knowing-how* and *ability* have been seen sometimes as different. Some authors (e.g., [11,34,28]) have argued that abilities can exist in the absence of knowing-how. Some others (e.g., [17,35,9]) have argued that knowing-how does not entail having the ability. We take no side on this discussion, choosing a computational perspective instead and using both terms interchangeably.

do. On the other hand, more recent logics for *knowing-how* [39,22,23,24,25,26] (which have mostly considered only the single-agent case) take a planning-based

While considering groups, or coalitions, is quite standard in logics for strategic reasoning, the *knowledge-how* of a group has been only explored recently in [26]. In both contexts, a crucial question is how the strategies/knowledge-how of individuals can be ‘combined’ to give raise to the strategies/knowledge-how of a group. The *parallel/concurrent* execution of individual strategies has been *de facto* the standard approach in the logics for strategic reasoning, and it is also used in the distributed knowledge-how studied in [26].

Contribution. This paper concerns the gap between individual and group know-how. It presents several notions of group know-how that arise from individual know-how. The particular focus is on *sequential* forms of group know-how (considered also in [26]), which we believe are closer to the knowledge-how literature described above. With our approach, we can model scenarios in which a group of agents can achieve more than its individual members by following a ‘*assembly/production line*’ approach: the group can use *sequences* of actions whose components can be performed by its members/subgroups. We argue that, in many settings, this is quite a natural reading of group know-how. For example, it is relevant whenever work can be divided among agents, each specialising in different tasks, or when there are dependencies between actions. Typical instances are those from which we borrow the name: assembly lines enable the combination of distinct forms of expertise, and they are a necessity when the execution of certain actions depends on the successful completion of previous ones. Finally, our proposal is also well suited to model the *compositionality of abilities*, resembling program composition as in Propositional Dynamic Logic (PDL) (see, e.g., [19]). With this idea in mind, the fundamental point of discussion is the way the plans available to individual agents are ‘combined’ to define the collection of plans that are available for a given group.

In this text, Sec. 2 introduces an underlying framework for individual knowing-how. Then, Sec. 3 starts by providing some natural minimal conditions that the abilities of a group must fulfill, giving two instantiations of that. While in these first two alternatives agents have a form of bounded contribution to the group’s abilities, the section also discusses a further option in which agents can contribute as much as needed/possible, closer to the intuition behind *common knowledge* [15]. For all the proposals, we study their model-checking problems (Sec. 4) and establish their relative expressive power (Sec. 5).

2 Basic Definitions

The multi-agent framework for individual *knowing-how* introduced in this section relies on the goal-directed approach of [39], incorporating also ideas from [7,6]. Throughout the text, let **Prop** be a countable set of propositional symbols, **Act** a countable set of symbols for ‘basic’ actions, and **Agt** $\neq \emptyset$ a finite set of agents.

Definition 1 (Plans). Let Act^* denote the set of finite sequences over **Act**, with ϵ being the empty sequence. The elements of Act^* are called plans. The

length of a plan $\sigma \in \text{Act}^*$, denoted as $|\sigma|$, is inductively defined as usual: $|\epsilon| := 0$ and, given $\sigma \in \text{Act}^*$ and $a \in \text{Act}$, $|\sigma a| := |\sigma| + 1$. Given $0 \leq k \leq |\sigma|$, the plan σ^k is σ 's initial segment up to (and including) the k -th position (with $\sigma^0 := \epsilon$). Given $0 < k \leq |\sigma|$, the action $\sigma[k]$ is the one in σ 's k -th position.

The models we are interested in are labelled transition systems.

Definition 2 (Labelled Transition System). A labelled transition system (LTS) (defined over Prop and Act) is a tuple $\langle S, R, V \rangle$ where S is a non-empty set of states (the domain), $R = \{R_a \subseteq S \times S \mid a \in \text{Act}\}$ is a collection of binary relations on S , and $V : S \rightarrow 2^{\text{Prop}}$ is a valuation function. As usual, R is extended to elements of Act^* by taking $R_\epsilon := \{(s, s) \mid s \in S\}$ and $R_{\sigma a} := R_\sigma \circ R_a$ (first R_σ and then R_a).

A multi-agent LTS (MLTS, sometimes called simply a model) (defined over Prop , Act and Agt) is a tuple $\langle S, R, A, V \rangle$ where $\langle S, R, V \rangle$ is an LTS and $A : \text{Agt} \rightarrow 2^{\text{Act}^*}$ assigns, to each agent in Agt , a collection of plans that always includes the empty plan (so, $\epsilon \in A(i)$ for every $i \in \text{Agt}$). Given an MLTS $\mathfrak{M}$ and a state s in it, the pair $(\mathfrak{M}, s)$ (parentheses usually dropped) is called a pointed MLTS.

An LTS is simply a multi-relational model in which each relation describes the effects of a particular ‘basic’ action. The additional component in an MLTS, the function indicating the plans available to each agent $i \in \text{Agt}$, allows us to define the knowledge-how of the individual agents.⁵ The language for describing the latter is defined below.

Definition 3. Formulas of the language L_{Kh_i} are defined by the following BNF:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \text{Kh}_i(\varphi, \varphi),$$

with $p \in \text{Prop}$ and $i \in \text{Agt}$. Other Boolean connectives are defined as usual, and the standard conventions for omitting parentheses apply. Formulas of the form $\text{Kh}_i(\varphi, \varphi)$ state that “agent i knows how to make φ true, provided that φ holds”.

For the semantic interpretation of *knowledge-how* formulas, the idea in the original proposal from [38,39] (note: over LTSs) is the following. An agent i knows how to achieve φ when ψ is the case, $\text{Kh}_i(\psi, \varphi)$, if and only if there is a plan $\sigma \in \text{Act}^*$ satisfying the following: (1) it is, executability-wise, ‘appropriate’, and (2) all its executions starting at any ψ -state lead only to φ -states. Here (recall: over MLTSs), the requirements are stronger: an agent i knows how to achieve φ when ψ is the case if and only if there is a plan $\sigma \in \text{Act}^*$ satisfying the following: (1) it is available to the agent, (2) it is, executability-wise, ‘appropriate’, and (3) all its executions starting on any ψ -state lead only to φ -states.⁶

Concerning what an executability-wise ‘appropriate’ plan is, the intuition is that a plan is appropriate when it can be completed every time it can be started (see, e.g., [38,6] for discussions). The following definition makes this precise.

⁵ An MLTS is a particular instance of the *uncertainty-based* LTSs from [7,6].

⁶ The requirements in the uncertainty-based LTSs from [7,6] are even stronger: conditions (1)-(3) must be satisfied for a *class of indistinguishable* plans.

Definition 4 (Strong executability). Let $\mathfrak{M} = \langle S, R, A, V \rangle$ be an MLTS, with $R = \{R_a \subseteq S \times S \mid a \in \text{Act}\}$. A plan $\sigma \in \text{Act}^*$ is strongly executable (SE) at $u \in S$ if and only if, for every $k \in [0..|\sigma|-1]$, from $v \in R_{\sigma^k}(u)$ it follows that $R_{\sigma^{k+1}}(v) \neq \emptyset$. The set $\text{SE}^{\mathfrak{M}}(\sigma) := \{s \in S \mid \sigma \text{ is SE at } s\}$ contains the states in $\mathfrak{M}$ where σ is SE. We will drop the superscript when no confusion arises.

Thus, a plan is strongly executable (at a state) when all its partial executions (from that state) can be completed. With this, the semantic interpretation for *knowing-how* formulas over pointed MLTS is defined as follows. (The cases for atoms and Boolean operators are as usual, and thus omitted, for brevity.)

Definition 5. Let $\mathfrak{M} = \langle S, R, A, V \rangle$ be an MLTS and $s \in S$.

$\mathfrak{M}, s \models \text{Kh}_i(\psi, \varphi)$ iff there is a $\sigma \in A(i)$ such that:
 $(i) \llbracket \psi \rrbracket^{\mathfrak{M}} \subseteq \text{SE}(\sigma)$, and $(ii) R_{\sigma}(\llbracket \psi \rrbracket^{\mathfrak{M}}) \subseteq \llbracket \varphi \rrbracket^{\mathfrak{M}}$,

with $\llbracket \chi \rrbracket^{\mathfrak{M}} := \{s \in S \mid \mathfrak{M}, s \models \chi\}$ the set of states in S satisfying χ . Any $\sigma \in \text{Act}^*$ making true the existential statement in the semantic clause of $\text{Kh}_i(\psi, \varphi)$ is called a witness for $\text{Kh}_i(\psi, \varphi)$. Define $\mathfrak{M} \models \varphi$ iff $\llbracket \varphi \rrbracket^{\mathfrak{M}} = S$, and $\models \varphi$ iff $\mathfrak{M} \models \varphi$, for all MLTS $\mathfrak{M}$. These notions are extended as expected for all languages in the rest of the article.

Note how the semantic clause for *knowing-how* formulas does not refer to the evaluation point, focusing rather on *global* properties of the model. Moreover, note that L_{Kh_i} is a very ‘simple’ language. In particular, even though the Kh_i modality has fairly complex requirements for its witness, the witness itself is not featured in the formula. Thus, L_{Kh_i} is actually ‘blind’ to it.

3 Logics of Group Know-How

In an MLTS, the abilities of an agent i are determined by her available plans (namely, those in $A(i) \subseteq \text{Act}^*$). Thus, a reasonable strategy for defining a notion of ability for a group $G \subseteq \text{Agt}$ is to provide an ‘aggregated’ set of plans $A(G) \subseteq \text{Act}^*$. Then, the semantic interpretation of a ‘group ability’ modality Kh_G can be given almost exactly as that for Kh_i , the only difference being the use of $A(G)$ instead of $A(i)$:

$\mathfrak{M}, s \models \text{Kh}_G(\psi, \varphi)$ iff there is $\sigma \in A(G)$ such that:
 $(i) \llbracket \psi \rrbracket^{\mathfrak{M}} \subseteq \text{SE}(\sigma)$, and $(ii) R_{\sigma}(\llbracket \psi \rrbracket^{\mathfrak{M}}) \subseteq \llbracket \varphi \rrbracket^{\mathfrak{M}}$.

Of course, not every such ‘aggregation’ of plans would produce a reasonable notion of ability for a group G . A first natural requirement is that the abilities of any singleton group should be the same as the abilities of its lone member:

(R1) For every $i \in \text{Agt}$, $A(\{i\}) = A(i)$.

Another requirement is that a group should have at least the same abilities as any of its subgroups. Although this does not hold universally, it is a reasonable requirement in a setting in which no agent ‘diminishes’ a group’s abilities.

(R2) For every $G, H \subseteq \text{Agt}$, if $G \subseteq H$ then $A(G) \subseteq A(H)$.

Besides being natural, these properties are also common in the literature. One example, within epistemic logic [20,15], is a group G ’s *distributed knowledge* $D_G\varphi$, intuitively understood as what follows from the aggregated knowledge of the group’s members (see, e.g., [18,16]). Indeed, an individual’s distributed knowledge is the same as her individual knowledge ($\models D_{\{i\}}\varphi \leftrightarrow K_i\varphi$), and what is known distributively by a group is also known distributively by its supergroups (if $G \subseteq H$ then $\models D_G\varphi \rightarrow D_H\varphi$). Another example, within logics for strategic reasoning (see, e.g., [36,2]), is a group G ’s *abilities* $\langle\!\langle G \rangle\!\rangle\varphi$, intuitively understood as what the group can guarantee will be the case (see, e.g., [31] for the case of CL). Indeed, a singleton group and its lone member have the same abilities, and if a group can guarantee a certain outcome, so can any of its supergroups (if $G \subseteq H$ then $\models \langle\!\langle G \rangle\!\rangle\varphi \rightarrow \langle\!\langle H \rangle\!\rangle\varphi$).

Below, we discuss several options to define the set $A(G)$ from an underlying $\{A(i) \mid i \in \text{Agt}\}$. The following situation will be our running example.

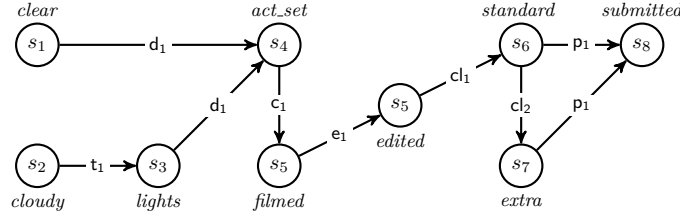


Figure 1: Model $\mathfrak{M}$.

Example 1. Imagine that a small film team would like to shoot a short movie to submit to a local film festival. It was decided that the whole movie would take place in a single room during the golden hour to have actors immersed in a soft, warm light and to enhance shadows. The filming crew consists of a director D , cinematographer C , editor E , technician T , colourist CL , and producer P .

Consider the MLTS $\mathfrak{M}$ in Fig. 1. On the filming day, if the sky is clear (state s_1 , where atom *clear* holds), the director sets the actors (action d_1 , leading to state s_4 , where atom *act.set* holds) and then the cinematographer films the scene (action c_1 , leading from s_4 to s_5). After that, the editor prepares the final edit (action e_1) and the colourist does the standard footage treatment (action cl_1 , leading to state s_6 , where atom *standard* holds); she can also choose to perform extra colour-grading (action cl_2). Finally, the producer submits the film to the festival (action p_1 from s_6 and s_7 to s_8). If the sky is overcast during the golden

hour (state s_2 with atom *cloudy*), the film crew can still shoot the film by first letting the technician prepare studio lights to simulate the golden hour (action $\mathbf{t}_1$). Then, the director sets the actors (action $\mathbf{d}_1$ from state s_3 to state s_4), and the rest follows as in the first option, while still allowing the colourist to perform extra colour-grading (action $\mathbf{cl}_2$). Taking $\mathbf{Agt} = \{D, T, C, E, CL, P\}$ and $\mathbf{Act} = \{\mathbf{d}_1, \mathbf{t}_1, \mathbf{c}_1, \mathbf{e}_1, \mathbf{cl}_1, \mathbf{cl}_2, \mathbf{p}_1\}$, this model's plan-assigning function is given by

$$\begin{aligned} A(D) &= \{\epsilon, \mathbf{d}_1\}, & A(C) &= \{\epsilon, \mathbf{c}_1\}, & A(CL) &= \{\epsilon, \mathbf{cl}_1, \mathbf{cl}_1\mathbf{cl}_2\}, \\ A(T) &= \{\epsilon, \mathbf{t}_1\}, & A(E) &= \{\epsilon, \mathbf{e}_1\}, & A(P) &= \{\epsilon, \mathbf{p}_1\}. \end{aligned}$$

Clearly, no individual agent can, on her own, finish and submit the film: for every $i \in \mathbf{Agt}$ we have $\mathfrak{M} \models \neg \text{Kh}_i(\text{clear}, \text{submitted}) \wedge \neg \text{Kh}_i(\text{cloudy}, \text{submitted})$. At the same time, individual agents know their part of the job: e.g., the colourist can, once the footage is edited, perform both standard colour-grading as well as standard colouring followed by extra colour-grading, expressed as $\mathfrak{M} \models \text{Kh}_{CL}(\text{edited}, \text{standard}) \wedge \text{Kh}_{CL}(\text{edited}, \text{extra})$ (witnesses, $\mathbf{cl}_1$ and $\mathbf{cl}_1\mathbf{cl}_2$).

3.1 An assembly-line approach

As discussed in the introduction, a natural idea for combining the abilities of a group of agents is that of an *assembly-line*: a plan is available to a group if and only if it is the *sequential composition* of plans available to its members. In a first version to this approach, each group member gets to participate in the process *at most once*. For a formal definition, it is useful to take the set of agents to be $\mathbf{Agt} = \{1, \dots, n\}$, so any subset $G \subseteq \mathbf{Agt}$ can be canonically associated with the sequence of its members in increasing order, which will be denoted by α_G . For instance, the group $\{2, 7, 5\}$ can be canonically associated with $\alpha_{\{2,5,7\}} = 2.5.7$.

Definition 6 (Assembly-line plans). Let $G \subseteq \mathbf{Agt}$ be a set of agents. Let α_G be its associated sequence, with α'_G any of its permutations. An assembly-line plan for the permutation α'_G is a sequence $\sigma_1 \cdots \sigma_{|G|}$ where $\sigma_t \in A(\alpha'_G[t])$ for each $t \in [1 \dots |G|]$: the agents take turns following the order indicated by α'_G , and at each turn t the corresponding agent $\alpha'_G[t]$ gets to choose one of her plans. The set of assembly-line plans for G , defined as

$$\text{ALP}(G) := \{\sigma_1 \cdots \sigma_{|G|} \in \mathbf{Act}^* \mid \sigma_t \in A(\alpha'_G[t]) \text{ with } \alpha'_G \text{ a permutation of } \alpha_G\},$$

contains all assembly-line plans for some permutation of G 's associated sequence α_G . To denote the assembly-line semantics ($A(G) = \text{ALP}(G)$), we will decorate the Kh_G modality and write Kh_G^{AL} . The resulting logic will be denoted $\mathbf{L}_{\text{Kh}_G}^{\text{AL}}$.

Each plan in $\text{ALP}(G)$ is built by asking every member $i \in G$ to choose, at their turn, a plan in $A(i)$. Since $\epsilon \in A(i)$ for all $i \in \mathbf{Agt}$, while choosing a plan in $A(i) \setminus \{\epsilon\}$ means agent i executes one of her 'proper' plans, choosing ϵ means she 'does nothing'. This approach satisfies both **(R1)** and **(R2)** (the latter because an assembly-line plan for G is an assembly-line plan for any $H \supseteq G$ in which the 'new' agents do nothing). Finally, note how, while an assembly-line plan for G might contain more than one occurrence of the same subplan σ' , this can only occur when more than one of G 's members has σ' at their disposal.

218 *Example 2.* Let $G = \{1, 2\}$ with $A(1) = \{\epsilon, a, b\}$ and $A(2) = \{\epsilon, ab\}$. The se-
 219 quence associated with $\{1, 2\}$ has two permutations: “1.2” and “2.1”.

- 220 – The set of assembly-line plans for “1.2” (1 choosing first and 2 choosing
 221 second) is $\{\epsilon, ab, a, aab, b, bab\}$.
- 222 – The set of assembly-line plans for “2.1” is (2 choosing first and 1 choosing
 223 second) is $\{\epsilon, a, b, ab, aba, abb\}$.

224 Then, the set of assembly-line plans for $\{1, 2\}$ is

$$225 \quad \{\epsilon, ab, a, aab, b, bab, aba, abb\}.$$

226 *Example 3.* Going back to the filming example in Fig. 1, take $G = \{D, T, C,$
 227 $E, CL, P\}$. The set of assembly-line plans for G is then all permutations of
 228 individual plans, one for each agent in G :

$$\{\epsilon, d_1c_1e_1cl_1p_1, t_1d_1c_1e_1cl_1cl_2p_1, \dots\}.$$

229 Note: no individual film crew member knows how to shoot a film and submit it
 230 to a film festival. Moreover, no proper subset of G has an assembly-line plan to
 231 achieve the goal. Still, the group as a whole does: $\mathfrak{M} \models \text{Kh}_G^{\text{AL}}(\text{clear}, \text{submitted}) \wedge$
 232 $\text{Kh}_G^{\text{AL}}(\text{cloudy}, \text{submitted})$, so G can eventually submit the film in both cases of a
 233 clear and cloudy sky. The witnesses are the plans $d_1c_1e_1cl_1p_1$ and $t_1d_1c_1e_1cl_1cl_2p_1$,
 234 which are strongly executable in s_1 and s_2 , respectively, and lead only to states
 235 satisfying *submitted*. Additionally, the filming crew has the ability to submit the
 236 film from any production stage. For example, if they receive a raw footage of
 237 the film (i.e., they are in state s_5), there is a group assembly-line plan to reach
 238 state s_8 ($\mathfrak{M} \models \text{Kh}_G^{\text{AL}}(\text{filmed}, \text{submitted})$). A witness for this is $e_1cl_1cl_2p_1$ (agents
 239 D and T execute empty plans).

240 3.2 A flexible assembly-line approach

241 In the assembly line proposal, the group gets one turn for each of its members.
 242 Then, during their turn, a member of G gets to decide between executing one of
 243 their plans or doing nothing. An alternative proposal is one in which these turns
 244 are seen as group resources that the group gets to use ‘as it sees fit’: the group
 245 can decide that the same agent will use two or more of the group’s available
 246 turns. For this, given $G \subseteq \text{Agt}$, let the disjoint union of the sets in $\{A(i) \mid i \in G\}$
 247 be denoted by $\text{AP}^{\sqcup}(G)$.

248 **Definition 7 (Flexible assembly-line plans).** Let $G \subseteq \text{Agt}$ be a set of agents.
 249 A flexible assembly-line plan for G is a sequence $\sigma_1 \cdots \sigma_{|G|}$ where $\sigma_t \in \text{AP}^{\sqcup}(G)$
 250 for each $t \in [1 \dots |G|]$: the group as a whole gets to choose, at each turn, a plan
 251 in $\text{AP}^{\sqcup}(G)$. The set of flexible assembly-line plans for G is defined as

$$252 \quad \text{FALP}(G) := \{\sigma_1 \cdots \sigma_{|G|} \in \text{Act}^* \mid \sigma_t \in \text{AP}^{\sqcup}(G)\}.$$

253 To denote the flexible assembly-line semantics ($A(G) = \text{FALP}(G)$), the Kh_G
 254 modality will be written as Kh_G^{FAL} . The obtained logic will be denoted $\text{L}_{\text{Kh}_G}^{\text{FAL}}$.

Each plan in $\text{FALP}(G)$ is built by asking the group G to choose, at each one of its turns, a plan in $\text{AP}^\sqcup(G)$, the group's set of *collectively available* plans. Requirements **(R1)** and **(R2)** are both clearly satisfied. Note how a flexible assembly-line plan for G might contain more than one occurrence of the same subplan σ' , even if σ' is only available to one of G 's members.

Example 4. Take $G = \{1, 2\}$ with $A(1) = \{\epsilon, a, b\}$ and $A(2) = \{\epsilon, ab\}$. Since (using multiset notation) $\text{AP}^\sqcup(G) = [\epsilon, a, b, \epsilon, ab]$, the set of flexible assembly-line plans for $\{1, 2\}$ is

$$\{\epsilon, a, b, ab, aa, aab, ba, bb, bab, aba, abb, abab\}.$$

Example 5. Recall the filming example from Fig. 1, and assume that the director has now decided to go more independent and handle the production and film dissemination on her own (i.e., the producer P is no longer part of the film crew). In other words, take $A(D) = \{\epsilon, d_1, p_1\}$. The set of flexible assembly-line plans for $G' = \{D, T, C, E, CL\}$ is

$$\{\epsilon, d_1 c_1 e_1 c_1 p_1, t_1 d_1 c_1 e_1 c_1 c_2 p_1, \dots\}.$$

All these plans have $|G'|$ steps (in the second, T performs ϵ), but now D performs two tasks: d_1 and then p_1 . The reduced crew G' has a flexible assembly-line plan to submit the film regardless of the weather conditions, in formulas, $\mathfrak{M} \models \text{Kh}_{G'}^{\text{FAL}}(\text{clear}, \text{submitted}) \wedge \text{Kh}_{G'}^{\text{FAL}}(\text{cloudy}, \text{submitted})$. However, they do not have (non-flexible) assembly-line plans to do so: $\mathfrak{M} \models \neg \text{Kh}_{G'}^{\text{AL}}(\text{clear}, \text{submitted}) \wedge \neg \text{Kh}_{G'}^{\text{AL}}(\text{cloudy}, \text{submitted})$.

3.3 Iterated plans

Finally, we also consider scenarios in which plans can be reused by agents and groups thereof. This approach is closer to actions in various logics for strategic reasoning, like ATL, where modelled systems are assumed to be reactive (i.e., they are assumed to run indefinitely). In such approaches, there is no limit on how many times groups of agents can execute their available actions. We apply these intuitions to our setting, bringing group know-how closer to the intuitions from the literature on strategic logics. We call the corresponding plans *iterated closed* plans, or simply **-plans*.

Definition 8 (*-plans). Let $G \subseteq \text{Agt}$ be a set of agents. The set of iterated plans (*-plans) for G is defined as

$$I(G) := \left(\bigcup_{i \in G} A(i) \right)^*.$$

Whenever Kh_G is interpreted under this (i.e., $A(G) = I(G)$), we denote it as Kh_G^I . We call $\text{L}_{\text{Kh}_G}^I$ the logic obtained under this semantics.

Thus, the set of $*$ -plans for a group G is the Kleene star closure of the union of the individual plans of members of G . Observe that, from an epistemic perspective, the idea of iterated closed plans has a *common knowledge* flavour.

Still, note: while condition **(R2)** is satisfied, condition **(R1)** is not. Indeed, an agent i with $A(i) = \{\epsilon, a\}$ can execute a only once, but the group $\{i\}$ (with, note, $I(\{i\}) = \{\epsilon, a, aa, \dots\}$) can execute a any finite number of times. While there might be justifications for such a scenario, we find it more natural to regain **(R1)**. We do so by discussing $*$ -plans only when individual agents can already reuse the plans at their disposal. Thus, we restrict this alternative to the class of models in which $A(i) = (A(i))^*$. This class will be denoted by $\mathbf{M}^*$, with its members denoted as $\mathfrak{M}^*$ and their plan-assigning function denoted by A^* .

Example 6. Take $G = \{1, 2\}$ with $A(1) = \{\epsilon, a, b\}^*$ and $A(2) = \{\epsilon, ab\}^*$. Hence, we have $I(G) = (\{\epsilon, a, b\} \cup \{\epsilon, ab\})^* = \{\epsilon, a, ab, aab, bab, \dots\}$.

Example 7. Let us revisit the filming example (Fig. 1) one last time, now assuming the filming budget is so low that there are only two crew members, D and C . The latter can still film the raw footage (c_1), but can now also edit it (e_1) and do a standard colour-grading (cl_1). The former, in addition to directing (d_1), can also submit the film to the festival (p_1). Hence, the set of $*$ -plans is $I(\{D, C\}) = (\{\epsilon, d_1, p_1\} \cup \{\epsilon, c_1, e_1, cl_1\})^* = \{\epsilon, d_1 c_1 e_1 cl_1 p_1, \dots\}$. Thus, the new crew can achieve their goal when the weather is sunny: $\mathfrak{M} \models \text{Kh}_G^I(\text{clear}, \text{submitted})$, with $d_1 c_1 e_1 cl_1 p_1$ being the witness. Still, the goal cannot be reached if the weather is cloudy: they lack the technical expertise to set up the light simulating the golden hour (i.e., all their plans lack action t_1): $\mathfrak{M} \models \neg \text{Kh}_G^I(\text{cloudy}, \text{submitted})$.

4 Model Checking Logics of Group Know-How

This section investigates the computational complexity of the model-checking problem, defined for a logic $\mathcal{L}$ as follows.

Input: A finite MLTS $\mathfrak{M}$, a state s of $\mathfrak{M}$ and a formula φ of $\mathcal{L}$.

Output: True, if $\mathfrak{M}, s \models \varphi$; False, otherwise.

We will show how the different notions of group know-how introduced above lead to very different complexity results.

4.1 (Flexible) Assembly Line Plans

We design an iterative algorithm that, given an input formula φ and a model $\mathfrak{M}$, labels each state with the subformulas of φ it satisfies. To do so, w.l.o.g., we order φ 's subformulas in increase order, $\text{ord}(\varphi) = \{\psi_1, \dots, \psi_m\}$, so the labelling algorithm works bottom-up. For a given state s , we will denote the set of subformulas the state is labelled with as $\ell(s)$. Initially, each state in the given model is labelled with the set of propositional atoms it satisfies, as indicated by the valuation function.

Algorithm 1 Global Model Checking for Assembly Line Know-How

```

1: procedure MC( $\mathfrak{M}, s, \varphi$ )
2:   for  $\psi_j \in \{\psi_1, \dots, \psi_m\} = \text{ord}(\varphi)$  do
3:     case  $\psi_j = \text{Kh}_G(\psi_k, \psi_{k'})$ , with  $k, k' < j$  ▷ The other cases are standard
4:       if  $\text{PLANVERIF}(\mathfrak{M}, \text{Kh}_G^{\text{AL}}(\psi_k, \psi_{k'}))$  then
5:         add  $\psi_j$  to  $\ell(s)$  for all  $s \in S$ 
6:   procedure  $\text{PLANVERIF}(\mathfrak{M}, \text{Kh}_G^{\text{AL}}(\psi_k, \psi_{k'}))$ 
7:     for  $i \in G$  do ▷ We guess a plan for each agent
8:       guess  $\sigma$  s.t.  $\sigma \in A(i)$ 
9:       guess  $\alpha'_G$  ▷ Guess an order for the agents to act
10:      let  $\sigma'$  be s.t.  $\sigma'_t \in A(\alpha'_G[t])$  ▷ This is built from the two guesses above
11:       $kh \leftarrow \text{true}$ 
12:      for  $v \in \ell(\psi_k)$  do
13:         $kh \leftarrow kh \ \& \ v \in \text{SE}(\sigma')$ 
14:        for  $u \in R_{\sigma'}(v)$  do
15:           $kh \leftarrow kh \ \& \ \psi_{k'} \in \ell(u)$ 
16:      if  $kh$  then return True else return False

```

Model checking $\mathbb{L}_{\text{Kh}_G}^{\text{AL}}$ is done in Alg. 1, where we omit Boolean cases for brevity, and focus only on the group know-how case. In the algorithm, we do not explicitly compute the set of assembly-line plans for a group, since such a set is exponential in the number of agents in the group and the number of individual plans. Instead, we use an NP-oracle to guess such plans.

Theorem 1. *The model-checking problem for $\mathbb{L}_{\text{Kh}_G}^{\text{AL}}$ is in P^{NP} .*

Proof. The correctness of the algorithm is guaranteed by the following claim: $\mathfrak{M}, s \models \psi$ iff $\psi \in \ell(s)$, for all subformulas ψ of φ . It is clear that the number of loops performed by MC is polynomial with respect to the size of the input formula and the size of the model. For the case of group know-how, we turn to an NP-oracle PLANVERIF, which guesses a group plan of linear length in the size of a given group and the sizes of plans of its members, and verifies that the guessed plan satisfies the semantic conditions (i) and (ii) in Def. 5. Checking strong executability (line 13) can be done in polynomial time (see, e.g., [5,14]). Hence, the whole procedure PLANVERIF executes a polynomial number of steps. If the oracle returns True for a given $\text{Kh}_G(\psi_k, \psi'_{k'})$, we label all states of the model with $\text{Kh}_G(\psi_k, \psi'_{k'})$. Finally, we make at most a linear number of calls to the NP-oracle, and hence the total running time of the algorithm is in P^{NP} .

For the flexible assembly-line ($\mathbb{L}_{\text{Kh}_G}^{\text{FAL}}$), we get a similar result by a slight adaptation of the algorithm since the witness for a group G is bounded by $|G|$.

4.2 Iterated plans

Turning to the *-plans, one can wonder whether we can reuse (an adaptation of) Alg. 1 with the NP-oracle. The crucial idea behind the algorithm was that

assembly-line plans have the *polynomial plan property*, i.e., if $\mathfrak{M}, s \models \text{Kh}_G^{\text{AL}}(\psi, \varphi)$, then the corresponding witness is of at most polynomial size (and similarly for the flexible assembly-line plans). This allowed the oracle guess and verify such a plan in polynomial time. Interestingly, $*$ -plans do not have such a property, and witnesses for $\text{Kh}_{\{i\}}^{\text{I}}(\psi, \varphi)$ can be of exponential length.

Proposition 1. *There are $\text{Kh}_G^{\text{I}}(\psi, \varphi)$ and $\mathfrak{M}, s$, such that $\mathfrak{M}, s \models \text{Kh}_G^{\text{I}}(\psi, \varphi)$ and the shortest witness is of exponential size, even for $|G| = 1$.*

To illustrate this fact, consider $\text{Kh}_{\{1\}}^{\text{I}}(p, q)$ and take $n \in \mathbb{N}^+$. We can construct a model $\mathfrak{M}_n$ of size polynomial in n such that the witness for $\text{Kh}_{\{1\}}^{\text{I}}(p, q)$ is of exponential size in n . This model consists of n pairs of states l_i (for *left*) and r_i (for *right*), $A(1) = (\{\epsilon, a_1, \dots, a_n\})^*$, atom p holds in l states, and atom q holds in r states. Given $1 \leq i \leq n$, there is a transition from l_i to r_i via a_i and for $1 \leq j < i$ there are self-loops for l_i labelled with a_j . For r_i , there are self-loops for a_j , and for $j < k \leq n$, there are transitions from r_i to l_i via a_k . The model intuitively consists of a stack of left-right pairs for each number up to n such that to perform a switch from left to right, the agent must use a special action for each pair. The transition function is constructed in such a way that after an agent makes a switch at some level i , she has to re-switch on all earlier levels, thus modelling a binary counter. Models $\mathfrak{M}_4$ and $\mathfrak{M}_5$ are shown in Fig. 2.

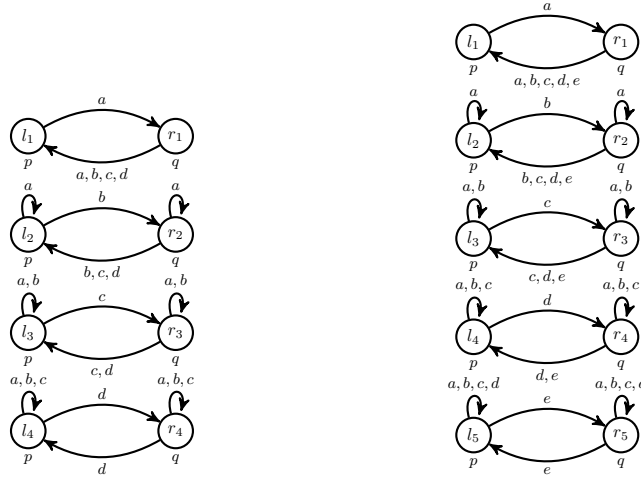


Figure 2: Models $\mathfrak{M}_4$ (left) with $A(1) = (\{\epsilon, a, b, c, d\})^*$ and $\mathfrak{M}_5$ (right) with $A(1) = (\{\epsilon, a, b, c, d, e\})^*$.

A witness to satisfy $\text{Kh}_{\{1\}}^{\text{I}}(p, q)$ in $\mathfrak{M}_4$ would allow the agent to switch from all l -states to r -states. However, observe that if we arrive in a situation in which at each level we can execute action c , it means that we can, in particular, execute

the transition from l_3 to r_3 . This scenario will force the two pairs above to ‘un-switch’, i.e., the agent will enter states l_1 and l_2 . Hence, the intuition here is that after each switch, the agent is forced to redo all the switches on earlier pairs.

Let us construct the shortest witness for $\text{Kh}_{\{1\}}^I(p, q)$ in $\mathfrak{M}_4$. Since the witness should be strongly executable, it must start with a (we omit ϵ since it is not relevant). After transition a , the ‘global state’ of the stack of pairs is (l_4, l_3, l_2, r_1) . Next, executing action a one more time takes us one step back in satisfying the formula, so the only meaningful strongly executable choice here is action b , leading to the global state (l_4, l_3, r_2, l_1) . The only strongly executable action after that is a again, putting the global state to (l_4, l_3, r_2, r_1) , i.e., we re-switched the earlier pair at level 1. Following this reasoning, the shortest path that reaches the global state (r_4, r_3, r_2, r_1) , and hence witnesses $\text{Kh}_{\{1\}}^I(p, q)$, is $abacabadabacaba$. Hence, for $n = 4$, the length of the shortest witness for $\text{Kh}_{\{1\}}^I(p, q)$ is 15.

Generalizing this idea, for model $\mathfrak{M}_n$ the corresponding witness is $\text{witness}_n = \text{witness}_{n-1} + a_n + \text{witness}_{n-1}$, i.e., a witness for $\mathfrak{M}_{n-1}$ concatenated with the switch action for level n and then again the witness for $\mathfrak{M}_{n-1}$. For instance, for $n = 5$, the witness is $abacabadabacaba + e + abacabadabacaba$, whose length is 31. With this at hand, we have shown that models $\mathfrak{M}_n$ grow polynomially w.r.t. n , but required witnesses grow exponentially, their length is $2^n - 1$ to be precise. Remarkably, the argument above applies for the logic L_{Kh} over LTS from [38].

Even though witness for $\ast$ -plans group know-how can be of exponential size, we will show that for model checking one can still leverage polynomial space. In particular, we will show that the problem is PSPACE -complete, by adapting a similar result from [14] for the logic of knowing how from [38]. Notice that we consider model-checking w.r.t. an arbitrary model $\mathfrak{M}$.

Theorem 2. *The model-checking problem for $\text{L}_{\text{Kh}_G}^I$ is PSPACE -complete.*

Proof. First, we can consider that the Kh operator over LTS from [38] can be exactly captured by $\text{Kh}_{\{i\}}^I$, where $\text{Agt} = \{i\}$ and $\text{A}(i) = \text{Act}$ (notice Act is finite). Thus, from the results from [14], it follows that the model-checking problem for $\text{L}_{\text{Kh}_G}^I$ is PSPACE -hard. To establish PSPACE -membership, we will again rely on results from [14]. Therein, a procedure running in polynomial space is given for Kh over LTS. The procedure relies on, given a formula $\text{Kh}(\varphi, \psi)$: 1) the construction of a finite-state automaton (FSA) $\mathcal{A}_\varphi$ whose language is $L(\mathcal{A}_\varphi) = \{\sigma \in \text{Act}^* \mid \sigma \text{ is SE at } \llbracket \varphi \rrbracket^{\mathfrak{M}}\}$; and 2) the construction of an FSA $\mathcal{A}_{(\varphi, \psi)}$ with language $L(\mathcal{A}_{(\varphi, \psi)}) = \{\sigma \in \text{Act}^* \mid \text{R}_\sigma(\llbracket \varphi \rrbracket^{\mathfrak{M}}) \subseteq \llbracket \psi \rrbracket^{\mathfrak{M}}\}$, and then checking non-emptiness of the intersection of the two automata (which can be done in PSPACE).⁷ We only need to reuse this argument over Kh_G^I . The only critical part is to make sure that a plan σ in such intersection is also such that $\sigma \in \text{I}(G)$.

To do so, recall that for each $i \in \text{Agt}$, $\text{A}(i)$ is a finite set of plans, thus we can easily define an FSA $\mathcal{A}_i$ such that $L(\mathcal{A}_i) = \text{A}(i)$, and for $G \subseteq \text{Agt}$, define $\mathcal{A}_G$ as the union of all $\mathcal{A}_i$ such that $i \in G$. Moreover, we can extend $\mathcal{A}_G$ to $\mathcal{A}_G^*$, where $L(\mathcal{A}_G^*) = (L(\mathcal{A}_G))^*$. All these manipulations can be done using standard

⁷ For basic notions associated to automata theory, we refer, e.g., to [21].

415 automata techniques (see e.g. [21]), and the resulting FSA $\mathcal{A}_G^*$ is of polynomial
 416 size w.r.t. the input model.

417 To check whether $\mathfrak{M}, s \models \text{Kh}^I(\varphi, \psi)$, we can check the non-emptiness of the
 418 product between the FSA $\mathcal{A}_\varphi$, $\mathcal{A}_{(\varphi, \psi)}$ and $\mathcal{A}_G^*$. This can be done in **NLogSpace**
 419 by checking the existence of an accepting plan on the fly without building the
 420 whole product FSA, which gives us an **NPSpace** procedure. Thus, by Savitch's
 421 theorem [33], model-checking for $L_{\text{Kh}_G}^I$ is in **PSpace**.

422 It is worth noticing that both in the lower and in the upper bound, we are
 423 using arbitrary models. However, we pointed out that for the logic $L_{\text{Kh}_G}^I$, it was
 424 reasonable to work with models in the class $\mathbf{M}^*$. However, a model $\mathfrak{M}^*$ is not
 425 a finite object, since each $A^*(i)$ contains an infinite number of plans. Given a
 426 model $\mathfrak{M}$, for model-checking each $A(i)$ (for $i \in \text{Agt}$) must be finite. We can easily
 427 construct a FSA $\mathcal{A}_i^*$ such that $L(\mathcal{A}_i^*)$ is $A^*(i)$. Then, we can repeat the method
 428 above, and therefore, model-checking over the class $\mathbf{M}^*$ is also **PSpace**-complete.

429 5 Expressivity

430 In this section, we investigate how logics with different notions of group know-
 431 how compare to each other with respect to their expressive power.

432 **Definition 9 (Expressivity).** Let L_1 and L_2 be two languages, and let $\varphi_1 \in L_1$
 433 and $\varphi_2 \in L_2$. Formulas φ_1 and φ_2 are equivalent, if for any MLTS $\mathfrak{M}$ and state
 434 w it holds that $\mathfrak{M}, w \models \varphi_1$ iff $\mathfrak{M}, w \models \varphi_2$. If for all $\varphi_1 \in L_1$ there is an equivalent
 435 $\varphi_2 \in L_2$, then we say that L_2 is at least as expressive as L_1 (denoted $L_1 \preceq L_2$).
 436 We say that L_2 is strictly more expressive than L_1 (denoted $L_1 \prec L_2$), if $L_1 \preceq L_2$
 437 and $L_2 \not\preceq L_1$. Finally, if $L_1 \not\preceq L_2$ and $L_2 \not\preceq L_1$, then L_1 and L_2 are incomparable.

438 First, we show that the logics of assembly-line group know-how are strictly
 439 more expressive than the logic of individual know-how.

440 **Proposition 2.** $L_{\text{Kh}_i} \prec L_{\text{Kh}_G}^{\text{AL}}$ and $L_{\text{Kh}_i} \prec L_{\text{Kh}_G}^{\text{FAL}}$.

441 *Proof.* That $L_{\text{Kh}_i} \preceq L_{\text{Kh}_G}^{\text{AL}}$ and $L_{\text{Kh}_i} \preceq L_{\text{Kh}_G}^{\text{FAL}}$ follows trivially from the fact that
 442 $L_{\text{Kh}_i} \subset L_{\text{Kh}_G}^{\text{AL}}$ and $L_{\text{Kh}_i} \subset L_{\text{Kh}_G}^{\text{FAL}}$ by taking $i = \{i\}$ plus the fact that both semantics
 443 satisfy property **(R1)**.

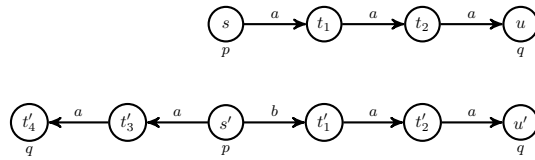


Figure 3: Models $\mathfrak{M}_1$ (above) and $\mathfrak{M}_2$ (below).

444 To see that $L_{\text{Kh}_G}^{\text{AL}} \not\preceq L_{\text{Kh}_i}$, consider two models, $\mathfrak{M}_1$ and $\mathfrak{M}_2$, depicted in Fig. 3,
 445 and assume that $A(1) = A(2) = \{\epsilon, a\}$. First, it is clear that $\mathfrak{M}_1, s \not\models \text{Kh}_{\{1,2\}}^{\text{AL}}(p, q)$

and $\mathfrak{M}_2, s \models \text{Kh}_{\{1,2\}}^{\text{AL}}(p, q)$. Indeed, the set $\text{ALP}(\{1, 2\})$ is $\{\epsilon, a, aa\}$. Plan aa is strongly executable in the p -state s' and allows reaching the q -state t'_4 in $\mathfrak{M}_2$. No plan of $\{1, 2\}$ allows reaching the q -state u from the p -state s in $\mathfrak{M}_1$.

To show that no L_{Kh_i} formula distinguishes the two models, we sketch an argument by induction. Let $A = \{s\}$, $A' = \{s'\}$, $B = \{t_1, t_2\}$, $B' = \{t'_1, t'_2, t'_3\}$, $C = \{u\}$, and $C' = \{u', t'_4\}$. We will denote an arbitrary union of states A , B , and C , as $D \in \{\bigcup E \mid E \in \mathcal{P}(\{A, B, C\})\}$. Given a D , we will denote by D' the same union as D but with primes added (e.g. if $D = A \cup B$, then $D' = A' \cup B'$).

For the base case, by the construction of models it is immediate that that if all $s \in D$ satisfy a propositional atom, then all $s' \in D'$ satisfy the same propositional atom, and vice versa.

For the induction hypothesis (IH), states in D all agree on ψ if and only if states in D' all agree on ψ . Boolean cases follow from the induction hypothesis and we focus on the know-how modality $\text{Kh}_i(\psi, \chi)$.

For both agents 1 and 2, there are two plans to consider: ϵ and a . Plan ϵ is strongly executable everywhere, and it keeps the system in the same state. Hence, for any state from any D that satisfies ψ and χ , we have that all states in the corresponding D' satisfy ψ and χ by the IH, and vice versa.

What is left is to consider plan a for both agents. Let ψ be such that it is satisfied in states in A , i.e. state s . By the IH, ψ is then satisfied in s' . The plan is executable in both states, and it leads to states t_1 and t'_3 . Since both states belong to B and B' respectively (and their unions with other sets), they both agree on χ by the induction hypothesis.

Next, let ψ be satisfied in states in B and B' . The plan is strongly executable in all the states, and hence in $\mathfrak{M}_1$ the reachable states are $\{t_2, u\}$, and in $\mathfrak{M}_2$ the reachable states are $\{t'_2, u', t'_4\}$. Observe that in both cases, the sets of reachable states include states from both B and C , and B' and C' . Hence, by the IH, all states of $B \cup C$ and $B' \cup C'$ agree on χ . Plan a is not strongly executable in any of the C and C' states. Thus, we get that that the models are indistinguishable by any agent using plan a . Finally, because there are no plans left, we conclude that no formula of L_{Kh_i} can distinguish models $\mathfrak{M}_1$ and $\mathfrak{M}_2$.

Because for the given models and sets of agents' plans, ALP and FALP plans coincide, we also get that $\text{L}_{\text{Kh}_i} \prec \text{L}_{\text{Kh}_G}^{\text{FAL}}$.

Next, we claim that assembly-line and flexible assembly-line plans indeed allow us to capture different notions of group know-how.

Proposition 3. $\text{L}_{\text{Kh}_G}^{\text{AL}}$ and $\text{L}_{\text{Kh}_G}^{\text{FAL}}$ are incomparable.

Proof. (sketch) To show $\text{L}_{\text{Kh}_G}^{\text{FAL}} \not\preceq \text{L}_{\text{Kh}_G}^{\text{AL}}$, we use again $\mathfrak{M}_1$ and $\mathfrak{M}_2$ from Fig. 3, and assume that $A(1) = \{\epsilon, a\}$ and $A(2) = \{\epsilon\}$. As we have seen in the proof of Prop. 2, single-agent plans a and ϵ are not enough to distinguish the models. Moreover, since $\text{ALP}(\{1, 2\}) = \{\epsilon, a\} = A(1)$, no $\text{L}_{\text{Kh}_G}^{\text{AL}}$ formula can distinguish the models. However, $\mathfrak{M}_1, s \not\models \text{Kh}_{\{1,2\}}^{\text{FAL}}(p, q)$ and $\mathfrak{M}_2, s \models \text{Kh}_{\{1,2\}}^{\text{FAL}}(p, q)$.

For $\text{L}_{\text{Kh}_G}^{\text{AL}} \not\preceq \text{L}_{\text{Kh}_G}^{\text{FAL}}$, we assume that we have three agents and that $A(1) = \{\epsilon, a\}$, $A(2) = \{\epsilon, a\}$, and $A(3) = \{\epsilon\}$. Under these conditions, no formula of

489 $\mathbb{L}_{\text{Kh}_G}^{\text{FAL}}$ can distinguish the models. However, we have that $\mathfrak{M}_1, s \not\models \text{Kh}_{\{1,2,3\}}^{\text{AL}}(p, q)$
 490 and $\mathfrak{M}_2, s \models \text{Kh}_{\{1,2,3\}}^{\text{AL}}(p, q)$. Full details can be found in App. A.

491 Finally, we focus on iterated group plans and claim that the resulting logic of
 492 group know-how is strictly more expressive than the logic of single-agent know-
 493 how. Following the discussion in Sec. 3.3, we compare the logics on the same
 494 class of models where individual plans are also closed under the Kleene star.

495 **Proposition 4.** *Interpreted on models from $\mathbf{M}^*$, $\mathbb{L}_{\text{Kh}_i} \prec \mathbb{L}_{\text{Kh}_G}^{\text{I}}$.*

496 6 Discussion and Future Work

497 In this paper, we proposed logics for reasoning about group *knowing-how*, which
 498 until now has predominantly been considered only for individual agents. Our
 499 proposal is based on a sequential ‘assembly-line’ approach, with the abilities of
 500 a group of agents being sequential compositions of individual plans in which
 501 each agent contributes in turns. We introduce three natural formalisations of
 502 such group knowing-how: assembly-line plans, flexible assembly-line plans, and
 503 iterated plans. For all alternatives, we study their model-checking problems and
 504 their relative expressive power.

505 To the best of our knowledge, the only other work that touches on group
 506 knowing-how is [26], which considers the notion of distributed knowing-how. The
 507 main difference from our approach is the set of plans a group G has available.
 508 In the cited paper, the set of ‘atomic’ actions a group G can use is given by

- 509 1. the ‘atomic’ actions available to any of its subgroups, and
- 510 2. the ‘parallel’ joint actions by (mutually exclusive) subgroups.

511 The sequential combination of these ‘atomic’ actions gives rise to (uniform)
 512 *plans/strategies*, over which the abilities of a group are defined. Thus, [26] is a
 513 ‘broader’ approach that deals with both sequential and parallel composition of
 514 individual plans. Still, the current proposal has its merits. The main one is that
 515 focusing only on sequential composition allows us to explore some of the ways this
 516 idea can be effectively implemented, as illustrated by the different forms of group
 517 plans that have been proposed. Moreover, while the iterated plans approach
 518 describes ‘the full sequential composition’ power, the assembly-line and flexible
 519 assembly-line approaches can be seen as resource-bounded versions modelling
 520 more realistic scenarios, still more restricted but with a better computational
 521 behaviour than e.g. [13].

522 There is a host of open problems worthwhile to tackle in the future. First,
 523 continuing with the computational aspect, we would like to investigate the satis-
 524 fiability problems for the logics. It is also interesting to provide sound and com-
 525 plete axiomatisations for the different languages, and define appropriate forms of
 526 bisimulation. Finally, we left as an open question whether, for the assembly-line
 527 semantics, the model-checking problem is P^{NP} -hard.

Acknowledgments. We would like to thank the reviewers for their comments and suggestions that helped us to improve the quality of the paper. This work was partially supported by Agencia I+D+i grant 2021-00400, the EU H2020 R&I programme under the MSC agreement 101008233 (MISSION), the IRP SINFIN, and SeCyT-UNC grant 33620230100178CB.

References

1. Ågotnes, T., Balbiani, P., van Ditmarsch, H., Seban, P.: Group announcement logic. *Journal of Applied Logic* **8**(1), 62–81 (2009). <https://doi.org/10.1016/J.JAL.2008.12.002>
2. Ågotnes, T., Goranko, V., Jamroga, W., Wooldridge, M.: Knowledge and ability. In: *Handbook of Epistemic Logic*, chap. 11, pp. 543–589. College Publications, UK (2015)
3. Ågotnes, T., Alechina, N.: Coalition logic with individual, distributed and common knowledge. *Journal of Logic and Computation* **29**(7), 1041–1069 (2019). <https://doi.org/10.1093/logcom/exv085>
4. Alur, R., Henzinger, T., Kupferman, O.: Alternating-time temporal logic. *Journal of the ACM* **49**(5), 672–713 (2002). <https://doi.org/10.1145/585265.585270>
5. Areces, C., Fervari, R., Saravia, A., Velázquez-Quesada, F.R.: The dynamics of knowing how. *Journal of Logic, Language, Information and Computation* (2025). <https://doi.org/10.1093/logcom/exad056>
6. Areces, C., Fervari, R., Saravia, A., Velázquez-Quesada, F.R.: Uncertainty-based knowing how logic. *Journal of Logic and Computation* **35**(1), 1–35 (2025). <https://doi.org/10.1093/LOGCOM/EXAD056>
7. Areces, C., Fervari, R., Saravia, A.R., Velázquez-Quesada, F.R.: Uncertainty-based semantics for multi-agent knowing how logics. In: Halpern, J.Y., Perea, A. (eds.) *Proceedings of the 18th TARK*. pp. 23–37. EPTCS (2021). <https://doi.org/10.4204/EPTCS.335.3>
8. Belardinelli, F., Jamroga, W., Kurpiewski, D., Malvone, V., Murano, A.: Strategy logic with simple goals: Tractable reasoning about strategies. In: Kraus, S. (ed.) *Proceedings of the 28th IJCAI*. pp. 88–94 (2019). <https://doi.org/10.24963/IJCAI.2019/13>
9. Bengson, J., Moffet, M.A.: Nonpropositional intellectualism. In: Bengson, J., Moffet, M.A. (eds.) *Knowing How. Essays on Knowledge, Mind and Action*, pp. 161–195. OUP (2011)
10. Catta, D., Galimullin, R., Murano, A.: First-order coalition logic. In: Kwok, J. (ed.) *Proceedings of the 34th IJCAI*. pp. 4410–4418. ijcai.org (2025). <https://doi.org/10.24963/IJCAI.2025/491>
11. Danto, A.C.: Basic actions. *American Philosophical Quarterly* **2**(2), 141–148 (1965)
12. Demri, S.: *Concise Introduction to Alternating-Time Temporal Logics - A Guide for Understanding the Model-Checking Problem*. Springer (2026). <https://doi.org/10.1007/978-3-032-11885-1>, <https://doi.org/10.1007/978-3-032-11885-1>
13. Demri, S., Doyen, L., Fervari, R.: Knowing-how reasoning with budgets recasted: Universal reachability problem on VASS. In: Ganty, P., Mansutti, A. (eds.) *Reachability Problems - 19th International Conference, RP 2025*. LNCS, vol. 16230, pp. 140–155. Springer (2025). https://doi.org/10.1007/978-3-032-09524-4_10
14. Demri, S., Fervari, R.: Model-checking for ability-based logics with constrained plans. In: Williams, B., Chen, Y., Neville, J. (eds.) *Proceedings of the 37th AAAI*. pp. 6305–6312. AAAI Press (2023). <https://doi.org/10.1609/AAAI.V37I5.25776>

- 576 15. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: Reasoning about knowledge. The MIT
577 Press (1995)
- 578 16. Galimullin, R., Kuijter, L.B.: Varieties of distributed knowledge. In: Ciabattoni, A.,
579 Gabelaia, D., Sedlár, I. (eds.) Proceedings of the 15th AiML. pp. 379–400. College
580 Publications (2024)
- 581 17. Ginet, C.: Knowledge, perception and memory. D. Reidel Publishing Company
582 (1975)
- 583 18. Halpern, J.Y., Moses, Y.: Knowledge and common knowledge in a distributed
584 environment. *Journal of the ACM* **37**(3), 549–587 (1990). [https://doi.org/10.](https://doi.org/10.1145/79147.79161)
585 [1145/79147.79161](https://doi.org/10.1145/79147.79161)
- 586 19. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. The MIT Press (2000)
- 587 20. Hintikka, J.: Knowledge and Belief. Cornell University Press (1962)
- 588 21. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to automata theory, lan-
589 guages, and computation, 3rd Edition. Pearson International Edition, Addison-
590 Wesley (2007)
- 591 22. Li, Y.: Stopping means achieving: A weaker logic of knowing how. *Studies in Logic*
592 **9**(4), 34–54 (2017)
- 593 23. Li, Y., Wang, Y.: Achieving while maintaining: – A logic of knowing how
594 with intermediate constraints. In: Ghosh, S., Prasad, S. (eds.) Proceedings of
595 the 7th ICLA. pp. 154–167. LNCS, Springer (2017). [https://doi.org/10.1007/](https://doi.org/10.1007/978-3-662-54069-5_12)
596 [978-3-662-54069-5_12](https://doi.org/10.1007/978-3-662-54069-5_12)
- 597 24. Li, Y., Wang, Y.: Multi-agent knowing how via multi-step plans: A dynamic epi-
598 stemic planning based approach. In: Blackburn, P., Lorini, E., Guo, M. (eds.)
599 Proceedings of the 7th LORI. LNCS, vol. 11813, pp. 126–139. Springer (2019).
600 https://doi.org/10.1007/978-3-662-60292-8_10
- 601 25. Li, Y., Wang, Y.: Planning-based knowing how: A unified approach. *Artificial In-*
602 *telligence* **296**, 103487 (2021). <https://doi.org/10.1016/j.artint.2021.103487>
- 603 26. Liu, B., Wang, Y.: Distributed knowing how. In: Bjørndahl, A. (ed.) Proceedings
604 of the 20th TARK. pp. 80–97. EPTCS (2025)
- 605 27. Lopes, A.D.C., Laroussinie, F., Markey, N.: ATL with strategy contexts: Express-
606 iveness and model checking. In: Lodaya, K., Mahajan, M. (eds.) Proceedings of
607 the 30th FSTTCS. pp. 120–132. LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum für
608 Informatik (2010). <https://doi.org/10.4230/LIPICS.FSTTCS.2010.120>
- 609 28. Miller, S.: Joint abilities, joint know-how and collective knowledge. *Social Epistem-*
610 *ology* **34**(3), 197–212 (2020)
- 611 29. Mogavero, F., Murano, A., Vardi, M.Y.: Reasoning about strategies. In: Lodaya,
612 K., Mahajan, M. (eds.) Proceedings of the 30th FSTTCS. LIPIcs, vol. 8, pp. 133–
613 144. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2010). [https://doi.org/](https://doi.org/10.4230/LIPICS.FSTTCS.2010.133)
614 [10.4230/LIPICS.FSTTCS.2010.133](https://doi.org/10.4230/LIPICS.FSTTCS.2010.133)
- 615 30. Naumov, P., Tao, J.: Together we know how to achieve: An epistemic logic of
616 know-how. *Artificial Intelligence* **262**, 279–300 (2018). [https://doi.org/10.1016/](https://doi.org/10.1016/J.ARTINT.2018.06.007)
617 [J.ARTINT.2018.06.007](https://doi.org/10.1016/J.ARTINT.2018.06.007)
- 618 31. Pauly, M.: A modal logic for coalitional power in games. *Journal of Logic and Com-*
619 *putation* **12**(1), 149–166 (2002). <https://doi.org/10.1093/logcom/12.1.149>
- 620 32. Pavese, C.: Knowledge how. In: Zalta, E., Nodelman, U. (eds.) The Stanford En-
621 cyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, USA,
622 Fall 2022 edn. (2022)
- 623 33. Savitch, W.J.: Relationships between nondeterministic and deterministic tape
624 complexities. *Journal of Computer and System Sciences* **4**(2), 177–192 (1970).
625 [https://doi.org/10.1016/S0022-0000\(70\)80006-X](https://doi.org/10.1016/S0022-0000(70)80006-X)

- 626 34. Setiya, K.: Knowing how. *Proceedings of the Aristotelian Society* **112**(3), 285–307
 627 (2012). <https://doi.org/10.1111/j.1467-9264.2012.00336.x>
- 628 35. Stanley, J., Williamson, T.: Knowing how. *The Journal of Philosophy* pp. 411–444
 629 (2001)
- 630 36. van der Hoek, W., Pauly, M.: Modal logic for games and information. In: Blackburn,
 631 P., Van Benthem, J., Wolter, F. (eds.) *Handbook of Modal Logic, Studies in Logic*
 632 *and Practical Reasoning*, vol. 3, pp. 1077–1148. Elsevier (2007). [https://doi.org/](https://doi.org/10.1016/S1570-2464(07)80023-1)
 633 [10.1016/S1570-2464\(07\)80023-1](https://doi.org/10.1016/S1570-2464(07)80023-1)
- 634 37. van der Hoek, W., Wooldridge, M.: Cooperation, knowledge, and time: Alternating-
 635 time temporal epistemic logic and its applications. *Studia Logica* **75**(1), 125–157
 636 (2003). <https://doi.org/10.1023/A:1026185103185>
- 637 38. Wang, Y.: A logic of knowing how. In: *Proceedings of the 5th LORI*.
 638 *LNCS*, vol. 9394, pp. 392–405. Springer (2015). [https://doi.org/10.1007/](https://doi.org/10.1007/978-3-662-48561-3_32)
 639 [978-3-662-48561-3_32](https://doi.org/10.1007/978-3-662-48561-3_32)
- 640 39. Wang, Y.: A logic of goal-directed knowing how. *Synthese* **195**(10), 4419–4439
 641 (2018). <https://doi.org/10.1007/s11229-016-1272-0>

642 A Technical Appendix

643 A.1 Proof of Prop. 3

644 We start with the proof for $L_{Kh_G}^{FAL} \not\equiv L_{Kh_G}^{AL}$. For this, we again employ the models
 645 $\mathfrak{M}_1$ and $\mathfrak{M}_2$ from Fig. 3, and assume that $A(1) = \{\epsilon, a\}$ and $A(2) = \{\epsilon\}$. As we
 646 have seen in the proof of Prop. 2, single-agent plans a and ϵ are not enough to
 647 distinguish the models. Moreover, since the set of ALP plans is $\{\epsilon, a\}$, i.e., equal
 648 to the set of plans of agent 1, we have that no $L_{Kh_G}^{AL}$ can distinguish the models.
 649 To see that the models are distinguishable by a $L_{Kh_G}^{FAL}$ formula, take $Kh_{\{1,2\}}^{FAL}(p, q)$.
 650 The set of FALP plans for $\{1, 2\}$ is $\{\epsilon, a, aa\}$. With this, it is immediate that
 651 $\mathfrak{M}_1, s \not\models Kh_{\{1,2\}}^{FAL}(p, q)$ and $\mathfrak{M}_2, s \models Kh_{\{1,2\}}^{FAL}(p, q)$.

652 Next, we show that $L_{Kh_G}^{AL} \not\equiv L_{Kh_G}^{FAL}$ by using the same models, $\mathfrak{M}_1$ and $\mathfrak{M}_2$,
 653 from Fig. 3. But now we assume that we have three agents and that $A(1) = \{\epsilon, a\}$,
 654 $A(2) = \{\epsilon, a\}$, and $A(3) = \{\epsilon\}$. As above, we argue by induction that no formula
 655 of $L_{Kh_G}^{FAL}$ can distinguish the models.

656 The set of all possible FALP plans for $\{1, 2, 3\}$ is $\{\epsilon, a, aa, aaa\}$. That plans a
 657 and ϵ for individual agents do not allow us to distinguish the two models follows
 658 from the proof of Prop. 2. Hence, we need to check only the new plans aa and
 659 aaa (this will take care of group know-how with group sizes 2 and 3).

660 We start with the case of ψ being satisfied in s . This means that if there is
 661 a plan in $\mathfrak{M}_1$ that reaches some χ -state, there should be a plan that reaches a
 662 χ -state in $\mathfrak{M}_2$ from s' . It is easy to see that by implementing plans aa and aaa ,
 663 we can reach t_2 and u respectively. Hence, χ should be true in either t_2 , u , or
 664 their union. If χ holds in t_2 , then the same effect can be achieved in $\mathfrak{M}_2$ by the
 665 group executing plan a and reaching state t'_3 , since by the IH χ holds in t_2 iff
 666 it holds in t'_3 . If χ is true in u , then the corresponding plan for the group to
 667 implement in $\mathfrak{M}_2$ from state s' is aa . Formula χ then holds in u iff it holds in t'_4
 668 by the IH. The other direction can be shown similarly.

669 If ψ is being satisfied in B or B' , then, among aa and aaa , there is no strongly
 670 executable plan that is executable in all of the states of either B and B' . Finally,
 671 apart from the case of ϵ , that we have tackled in the proof of Prop. 2, there
 672 are no strongly executable plans for the group in C and C' . Hence, there is no
 673 formula of $\mathbf{L}_{\mathbf{Kh}_G}^{\text{FALP}}$ that is able to tell apart the two models.

674 On the other hand, take formula $\mathbf{Kh}_{\{1,2,3\}}^{\text{AL}}(p, q)$. The set of ALP plans for the
 675 group is $\{\epsilon, a, aa\}$. From this, it is immediate that $\mathfrak{M}_1, s \not\models \mathbf{Kh}_{\{1,2,3\}}^{\text{AL}}(p, q)$ and
 676 $\mathfrak{M}_2, s \models \mathbf{Kh}_{\{1,2,3\}}^{\text{AL}}(p, q)$.

677 A.2 Proof of Prop. 4

678 That $\mathbf{L}_{\mathbf{Kh}_i} \preceq \mathbf{L}_{\mathbf{Kh}_G}^{\text{I}}$ follows from the fact that in $\mathbf{M}^*$, $\mathbf{Kh}_i(\psi, \chi)$ is equivalent to
 679 $\mathbf{Kh}_{\{i\}}^{\text{I}}(\psi, \chi)$, and hence $\mathbf{L}_{\mathbf{Kh}_i} \subset \mathbf{L}_{\mathbf{Kh}_G}^{\text{I}}$.

680 For the other direction, consider $\mathbf{Kh}_G^{\text{I}}(p, q)$ with $G = \{1, 2\}$, and the two
 models in Fig. 4.

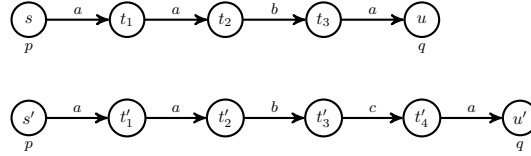


Figure 4: Models $\mathfrak{M}_1$ (above), $\mathfrak{M}_2$ (below), and the sets plans for both agents
 are $A(1) = (\{\epsilon, a\})^*$ and $A(2) = (\{\epsilon, bc\})^*$.

681 It is immediate that $\mathfrak{M}_1, s \not\models \mathbf{Kh}_G^{\text{I}}(p, q)$ and that $\mathfrak{M}_2, s \models \mathbf{Kh}_G^{\text{I}}(p, q)$. It can
 682 be proved by induction that no formula from $\mathbf{L}_{\mathbf{Kh}_i}$ distinguishes the two model.
 683 To show that no formula of $\mathbf{L}_{\mathbf{Kh}_i}$ can distinguish the two models, we provide a
 684 sketch of an argument by induction. For the base case, all states in set X satisfy
 685 the same propositional variables as states in the set X' , with $X \in \{A, B, C\}$
 686 and $A = \{s\}$, $A' = \{s'\}$, $B = \{t_1, t_2, t_3\}$, $B' = \{t'_1, t'_2, t'_3, t'_4\}$, $C = \{u\}$, and
 687 $C' = \{u'\}$.

688 For the induction hypothesis (IH), we have that for all ψ , all states in D
 689 agree on ψ iff all states in D' agree on ψ , with $D \in \{\bigcup E \mid E \in \mathcal{P}(\{A, B, C\})\}$,
 690 and D' being exactly like D but with primes added.

691 Boolean cases follow from the IH. Now, consider the case $\mathbf{Kh}_i(\psi, \chi)$ with $i \in$
 692 $\{1, 2\}$. Agent 1 has the set of available plans $\{\epsilon, a, aa, \dots\}$. Since the empty plan
 693 is strongly executable anywhere and it does not change the state of evaluation,
 694 executing such a plan in a state of any given set does not allow us to leave that
 695 set (recalling that all states in a given set agree on ψ and χ by the IH). Hence,
 696 by the IH, using empty plans does not allow us to distinguish the models.

697 Plan a of agent 1 is strongly executable in states s, s', t_1, t'_1, t_3 , and t'_4 .
 698 By the definition of the semantics, this plan should be strongly executable in *all*
 699

700 states satisfying ψ , and all executions should lead to a χ -state. This, in particular
 701 implies, that the only way $\text{Kh}_1(\psi, \chi)$ can be satisfied once evaluated in one the
 702 t - and t' -states is when all of t_2 , t'_2 , u , and u' satisfy χ , i.e., χ holds in $B \cup C$
 703 and $B' \cup C'$ by the IH. Hence, the two models cannot be distinguished by agent
 704 1 implementing plan a . Similar reasoning holds for the plan aa , which is only
 705 strongly executable in states s and s' , and executions from both states lead to
 706 states from B and B' respectively, which agree on χ by the IH. Finally, there
 707 are no other strongly executable plans for agent 1.

708 For agent 2, it is enough to consider only plan bc . The plan is only strongly
 709 executable in state t'_2 of $\mathfrak{M}_2$, and it reaches state t'_4 . By the semantics of the
 710 know-how modality, this would mean that t'_2 has to satisfy ψ and t'_4 has to satisfy
 711 χ . However, since these states both belong to B' and hence are indistinguishable
 712 by the IH, both t'_2 and t'_4 agree on the same formulas. Therefore, in model $\mathfrak{M}_1$, it
 713 is enough to find a plan for agent 2 that starts in a t -state and reaches a t -state,
 714 i.e., the plan that start and ends in B -states. The empty plan does this job.