stream (there a skipped 45° gave the appearance of a 90° turn). With a monoidal variator one can talk meaningfully of a single symbol from the variator expressing accumulated changes over time: the monoidal variator will take you from any configuration to any other, regardless of how many elements in the sequence of 45° turns have occurred.

Question 4 is challenging because it is rather cumbrous. The source of this is Definition 30: it expresses the idea that asynchronous queries might happen at any time in a very direct and unwieldy way. So, instead, we will consider the 'accumulated changes' intuition just described in the previous remark. This gives a new relation (actually a function) and, following the pattern employed for the pump and shrinking cases, we will then form a connection between the two.

Definition 31 (monoid integrator)**.** Given the monoid $(D,\oplus,1_D)$ and observation set Y, the associated *monoid integrator* is a function $\int\oplus\big|_Y$ is given by:

$$\begin{aligned}\int\oplus\big|_Y \colon \{\varepsilon\}\cup(Y\cdot D^*) &\to \{\varepsilon\}\cup Y\cup(Y\cdot D)\\ \varepsilon &\mapsto \varepsilon\\ y_0 &\mapsto y_0\\ y_0 d_1 d_2\ldots d_m &\mapsto y_0(d_1\oplus d_2\oplus\cdots\oplus d_m).\end{aligned}$$

Notice that $\int\oplus\big|_Y$ is a rather different relation from the previous ones. All the relations express an alteration under which we wish the device to be invariant. Or more precisely: in which we seek to determine whether the requisite information processing *can* be invariant. The relations prior to Definition 31 all describe transformations which we may envision being produced and processed directly—it is easy to think of the robot operating in the world and tracing strings in the relation's image. Not so for the monoid integrator: it serves mostly as an abstract definition. All the strings are short: the robot gets at most two symbols. Nevertheless, the usual questions still apply:

Question 5. For any device F with monoidal variator $(D,\bullet)$, of $Y(F)$, is it $\left(\nabla\big|^D_F \fatsemi \int\oplus\big|_{Y(F)}\right)$-simulatable?

Question 5 (Constructive). Given F with monoidal variator $(D,\bullet)$, find a sensori-computational device F' such that $F'\sim F\ \left(\mathrm{mod}\,\nabla\big|^D_F \fatsemi \int\oplus\big|_{Y(F)}\right)$, or indicate none exist.

To answer these questions, a constructive procedure is given in Algorithm 4: it builds a three-layer tree, where the first layer is the initial state, the second layer consists of the states reached by a single label in $Y(F)$, the third layer consists of the states reached by strings yd where $y\in Y(F)$ and $d\in D$. The first layer is produced through lines 1–4, and the second layer through lines 6–7. The third layer is constructed via a depth-first search on the derivative F' as shown between lines 8–23 by keeping track of the accumulated change d_{acc} and creating the new state reached by yd_{acc} accordingly. The correctness of the algorithm follows next.

Lemma 32. Algorithm 4 gives a sensori-computational device that output simulates F modulo $\nabla\big|^D_F \fatsemi \int\oplus\big|_{Y(F)}$ if and only if there exists a solution for Question 5.

Algorithm 4: Monoid Integrator: $\textsc{Int}_D(F)\rightarrowtail G$

Input : A sensori-computational device F, a monoid variator with monoid $(D,\oplus,1_D)$
Output: A deterministic sensori-computational device G if G output simulates F modulo $\nabla\big|^D_F \fatsemi \int\oplus\big|_Y$; otherwise, return 'No Solution'

1 $F' \coloneqq \textsc{Delta}_D(F)$
2 **if** F' *is 'No Solution'* **then**
3 | **return** 'No Solution'
4 Initialize G with an initial state v_0
5 **for** *edge* $v'_0 \xrightarrow{y} w'$ *in* F' **do**
6 | Create a state w in G and add edge $v_0 \xrightarrow{y} w$
7 | Associate w with w', $c(w) \coloneqq c(w')$
8 | $q' \coloneqq \mathit{Queue}([(1_D, w')])$
9 | **while** $q' \neq$ **do**
10 | | $(d_{\mathrm{acc}}, v') \coloneqq q'.\mathit{pop}()$
11 | | **for** $d \in v'.\mathit{OutgoingLabels}()$ **do**
12 | | | Let w' be the state such that $v' \xrightarrow{d} w'$
13 | | | **if** *there is no state* v_d *in* G **then**
14 | | | | Create a state v_d in G, $c(v_d) \coloneqq c(w')$
15 | | | | Add edge $w \xrightarrow{d} v_d$
16 | | | **else**
17 | | | | $X \coloneqq c(v_d) \cap c(w')$
18 | | | | **if** $X = \varnothing$ **then**
19 | | | | | **return** 'No Solution'
20 | | | | **else**
21 | | | | | $c(v_d) \coloneqq X$
22 | | | **if** $(d_{\mathrm{acc}} \oplus d, w')$ *is not visited* **then**
23 | | | | Add $(d_{\mathrm{acc}} \oplus d, w')$ to q'
24 **return** G

Proof: $\Longrightarrow$: We show that if Algorithm 4 gives a sensori-computational device G, then G output simulates F modulo $\nabla\big|^D_F \fatsemi \int\oplus\big|_{Y(F)}$. First, we need to show that for every string $s\in\mathscr{L}(F)$, $\nabla\big|^D_F \fatsemi \int\oplus\big|_{Y(F)}(s)\subseteq\mathscr{L}(G)$. If s has length 0 or 1, this holds trivially. Let $T_s=\left\{t \mid s\nabla\big|^D_F t\right\}$, i.e. the set of images of string s under the delta relation. Suppose $s=s_1s_2\ldots s_n$, having length greater than 1. This s can be traced in F', for otherwise we will get a 'No Solution' on line 3, contradicting the assumption that G was produced. Since D is a monoid, every string in T_s, denoted as $yd_1d_2\ldots d_n$, will be mapped to a string yd_{acc} where $d_{\mathrm{acc}}=d_1\oplus d_2\ldots\oplus d_n$. The depth-first search procedure between lines 8–23 will traverse each string $yd_1d_2\ldots d_n$ in F' and compute its image yd_{acc}. It creates a new state v_d that is reached by yd_{acc}. The algorithm continues to increase the depth until the current string arrives not only in a state in F' that is visited, but one visited under the same accumulated change d_{acc}. (This is why the queue q' contains pairs.) Therefore, all images of T_s after $\int\oplus\big|_{Y(F)}(s)$ can be traced in G. Hence, $\nabla\big|^D_F \fatsemi \int\oplus\big|_{Y(F)}(s)\subseteq\mathscr{L}(G)$.

Second, we will show that for every string $s\in\mathscr{L}(F)$ and $r\in\nabla\big|^D_F \fatsemi \int\oplus\big|_{Y(F)}(s)$, we have $\mathscr{C}(F,s)\supseteq\mathscr{C}(G,r)$. Via Lemma 17, for every string $s\in\mathscr{L}(F)$, $\mathscr{C}(F',t)\subseteq\mathscr{C}(F,s)$ holds for every string $t\in T_s$. In line 21, the image of string t outputs a subset of $\mathscr{C}(F',t)$. As a consequence, for every string $s\in\mathscr{L}(F)$ and

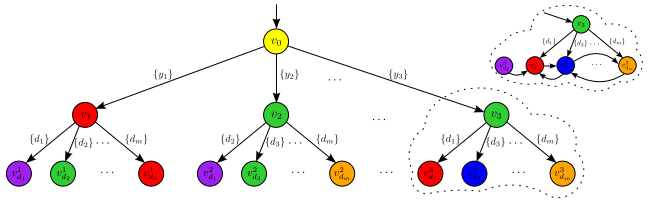

Fig. 11: A depiction of the tree with three layers employed in Algorithm 4 for the monoid integrator. Top right: the inset shows how Algorithm 5 modifies sub-trees so they gain extra edges (and potentially extra vertices) to make composite leaves as self-contained blocks with transitions to encode $(D,\bullet)$.

$r \in \nabla\big|_F^D \fatsemi \int\oplus\big|_{Y(F)}(s)$, we have $\mathscr{C}(F,s) \supseteq \mathscr{C}(G,r)$.

$\Longleftarrow$: If there exists a solution G for Question 5, then we will show that Algorithm 4 will return a sensori-computational device. Suppose instead, it returns 'No Solution'. If 'No Solution' is returned at line 3 then it fails because no F' exists (Lemma 17). But this will also be an obstruction to the existence of a G because either: (*i*) there exists a string in $\mathscr{L}(F)$ that fails to be mapped to any string via relation $\nabla\big|_F^D$ (line 28 of Algorithm 1); or (*ii*) there exists a string s in F with $\cap_{s\in S}\mathscr{C}(F,s) = \varnothing$ (line 37 of Algorithm 1). But then we obtain contradictions for relation $\nabla\big|_F^D \fatsemi \int\oplus\big|_{Y(F)}$: the same string in (*i*) makes it impossible for G to satisfy condition 1 of Definition 5; similarly for (*ii*), condition 2 of Definition 5 cannot be met by any G.

If 'No Solution' is returned at line 19 then there is a set of strings T with the same image, say r, after function $\int\oplus\big|_{Y(F)}$, with no output common to all elements of T. Since those strings in T are mapped from a set of strings $S \subseteq \mathscr{L}(F)$ via relation $\nabla\big|_F^D$, the strings in S do not share any common output in F. Otherwise, Algorithm 4 (line 17) and Algorithm 1 (line 35) will identify a suitable output, since the set of all possible common outputs is maintained by the intersection operations. But this contradicts G being a solution: $r \in \mathscr{L}(G)$ but there is no appropriate output for r to yield, so condition 2 of Definition 5 cannot be met for relation $\nabla\big|_F^D \fatsemi \int\oplus\big|_{Y(F)}$. Hence, if there exists a solution G for Question 5, Algorithm 4 will return a sensori-computational device. ∎

We remark on the fact that Algorithm 4 is unlike the previous algorithms. The preceding ones act as a sort of mutator: they begin by making a copy of their input graph, and local modifications are made before, finally, being determinized to catch inappropriate outputs (via the $c(\cdot)$ function) and to ensure that the resulting sensori-computational device is deterministic. This means that structure of their input F influences the structure of their result. But Algorithm 4 produces an sensori-computational device *de novo*, and its structure is essentially fixed: it is a 3-layer tree regardless of the specifics of F. The F serves really to define the input–output behavior. Figure 11 offers a visualization of the structure of F: the three-layer tree produces outputs for ε, the y_i elements, and strings of the form $y_j d_k$.

Now we return to consideration of Question 4.

Algorithm 5: Disaggregator: $\textsc{FillLeaves}_D(F') \rightarrowtail G$

Input : A monoid integrator device F' from Algorithm 4, and monoid variator D
Output: A deterministic device G that accepts additional elements in D

1 Initialize an empty graph M // Graph for D
2 **for** $d \in D$ **do**
3 | Create a state w_d in M
4 **for** $d_1, d_2 \in D$ **do**
5 | $d := d_1 \oplus d_2$
6 | Form an edge $w_{d_1} \xrightarrow{d} w_{d_2}$ in M
7 Create graph G with a single vertex v_0, with output $c(v_0) := c(v'_0)$
8 Let v'_0 be the initial state of F'
9 **for** *every outgoing edge* $v'_0 \xrightarrow{y} v'_y$ *in* F' **do**
10 | Add a vertex v_y to G
11 | Set $c(v_y) := c(v'_y)$, and connect $v_0 \xrightarrow{y} v_y$
12 | Create M^y as a copy of M, adding M^y to G
13 | **for** *every outgoing edge* $v'_y \xrightarrow{d} v'_d$ *in* F' **do**
14 | | Form an edge $v_y \xrightarrow{d} w^y_d$
15 | | $c(w^y_d) := c(v'_d)$
16 | Assign an arbitrary color to the remaining vertices in M^y
17 **return** G

Lemma 33. Given a sensori-computational device F with monoidal variator $(D,\bullet)$, F being $\left(\nabla\big|_F^D \fatsemi \partial\oplus\big|_F\right)$-simulatable implies that F is $\left(\nabla\big|_F^D \fatsemi \int\oplus\big|_F\right)$-simulatable.

Proof: As $\partial\oplus\big|_F \supseteq \int\oplus\big|_F$, what suffices for $\nabla\big|_F^D \fatsemi \partial\oplus\big|_F$ must also serve for $\nabla\big|_F^D \fatsemi \int\oplus\big|_F$, via Property 7. ∎

Lemma 34. Given a sensori-computational device F with monoidal variator $(D,\bullet)$, F being $\left(\nabla\big|_F^D \fatsemi \int\oplus\big|_F\right)$-simulatable implies that F is $\left(\nabla\big|_F^D \fatsemi \partial\oplus\big|_F\right)$-simulatable.

Proof: Form the monoidal integrator $F' = \textsc{Int}_D(F)$. Given the lemma's premise and the correctness of Algorithm 4, some F' must be produced. Now, construct a device G via $\textsc{FillLeaves}_D(F')$, as defined in Algorithm 5. It first creates a structure (called M) to capture the transition between different monoid elements in D (lines 1–6), and then affixes a copy of this structure below each y to enable the tracing of additional elements in D. Following this algorithm, we obtain a sensori-computational device G, and $\mathscr{L}(G) = \{\varepsilon\} \cup (Y \cdot D^*)$. To show that G output simulates F modulo $\nabla\big|_F^D \fatsemi \partial\oplus\big|_F$, consider a string $s \in \mathscr{L}(F)$ with $|s| = m$. When $m = 0$ or $m = 1$, the requirements are clearly met since G behaves as F' does on those strings. For $m > 1$, the string s will correspond to the some non-empty set of strings under $\nabla\big|_F^D$, each of them of the form $t = y d_1 d_2 \ldots d_{m-1}$; thus, $t \in \mathscr{L}(G)$. Let $d_{\text{acc}} = d_1 \oplus d_2 \oplus \ldots \oplus d_{m-1}$, then we know that $t' = y d_{\text{acc}}$ reaches the same state as that of t in G, according to the construction of G (lines 1–6). But t' on G is identical to t' on F', hence: $\mathscr{C}(G,t) = \mathscr{C}(G,t') = \mathscr{C}(F',t') \subseteq \mathscr{C}(F,s)$, the last being because $F' \sim F \pmod{\nabla\big|_F^D \fatsemi \int\oplus\big|_F}$. Therefore, G output simulates F modulo relation $\nabla\big|_F^D \fatsemi \partial\oplus\big|_F$. ∎

Theorem 35. For any sensori-computational device F with monoidal variator $(D,\bullet)$,

$$F \text{ is } \left(\nabla\big|_F^D \fatsemi \partial\oplus\big|_F\right)\text{-simulatable}$$
$$\Updownarrow$$
$$F \text{ is } \left(\nabla\big|_F^D \fatsemi \textstyle\int\oplus\big|_F\right)\text{-simulatable.}$$

Proof: Combine Lemma 33. and Lemma 34 ∎

The construction in Lemma 34 (via Algorithms 4 and 5) has a specific form: two layers, as a tree, reaching 'composite-leaves' formed by connected blocks. The inset to Figure 11 illustrates this. Since, for any F that is $\left(\nabla\big|_F^D \fatsemi \partial\oplus\big|_{Y(F)}\right)$-simulatable, there is a F' possessing this structure, we term it the *universal* monoid integrator. That same integrator will also output simulate F under $\nabla\big|_F^D \fatsemi \textstyle\int\oplus\big|_{Y(F)}$. (Heed: the universality refers to its structural form, the particular outputs at each vertex will depend on the F and $(D,\bullet)$ used.) This structure is will lead to Theorem 39.

VII. Chatter-free behavior

In this brief section, we introduce two additional concepts that resemble some aspects of the relations in Section V, and which also connect with the topics just discussed in Section VI. We start by presenting a new, detailed example which will motivate a pair of additional definitions.

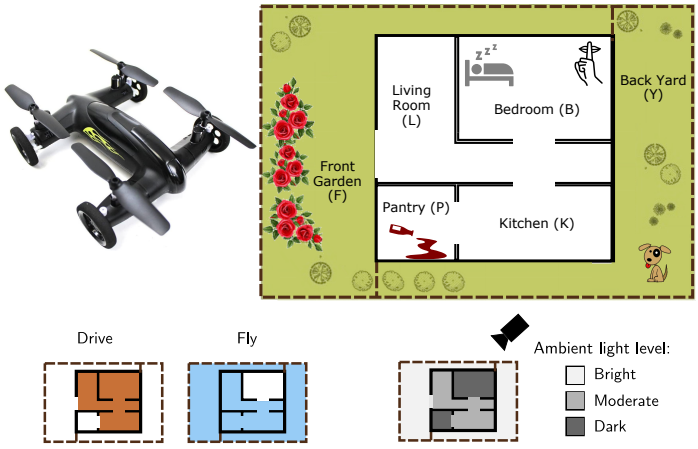

Fig. 12: A driving drone monitors a home environment. The robot is capable of both flight and wheeled locomotion and is equipped with a single-pixel camera. As an occupied residence, the space imposes complex constraints on how the vehicle may move. It must fly to avoid grass outdoors (F and Y) and liquids in the pantry (P); in the bedroom (B), it should drive to minimize noise. The robot is initially located in either the front garden (F) or the living room (L). To determine its state, the robot uses its single-pixel camera, which is capable of discerning just three different ambient light levels (Bright (B), Moderate (M), Dark (D)). The insets show: (left) the motion constraints and (right) the various light levels.

Example 15 (Driving drone)**.** The Syma X9 Flying Quadcopter Car, shown in Figure 12, is a robot marketed as a 'driving drone'. It is capable of switching between driving and flying modes, the idea being that it can make use of either mode of locomotion and determine what best suits the demands of its task. Imagine deploying such a robot in the scenario shown as the map on right-hand side of the figure. The robot, starting in either the front garden (F) or the living room (L), will move about the home and garden. Its size and construction, along with task constraints, mean that it must adjust its mode of locomotion depending on where it is. The robot has a single-pixel camera with which it determines different levels of ambient light. (The figure's caption provides specific details, and further explanation.)

It uses a sensori-computational device that processes the light readings as input, and outputs the appropriate mode (driving or flying). Figure 13a gives such a device; it is essentially a state diagram encoding the problem constraints, topological structure, and raw sensor readings. As Figure 13a is essentially a transcription of the problem, it serves as a type of specification for acceptable input–output functionality.

An observation variator $D_\ell = \{+,-,=\}$ uses $+$ to capture the brightness increase (from dark to moderate, from moderate to bright), $-$ for brightness decrease (from bright to moderate, from moderate to dark), $=$ for brightness equivalence. A derivative device obtained by applying Algorithm 1 to Figure 13a with this variator is shown in Figure 13b. □

Figure 13c presents a device that, though different from the straightforward derivative in Figure 13b, also implements the functionality in Figure 13a under delta relation associated with D_ℓ. That is to say, it also output simulates modulo $\nabla\big|_F^{D_\ell}$ and is, thus, also a derivative. Figure 13c, being smaller that either 13a or 13b, might be desirable for practical purposes. But now consider that the '$=$' element of the variator is produced when there is no change in light levels. If events are triggered when the robot moves from one room (or region) to the next, then there may not be too many of them. On the other hand, if the robot is using these readings to localize, that is, to actually determine that it may have transitioned from one room or region to the next, then many such elements will likely be generated.

Particularly when there are cycles on elements such as the '$=$' symbol, as these are 'neutral' changes in the signal, this may induce oscillatory behavior in the device as it fluctuates between states, flip-flopping rapidly. One may ask, thus, whether there are sensori-computational devices that can avoid this issue.

Definition 36 (vertex stable)**.** For an $F = (V,V_0,Y,\tau,C,c)$ and a set $\mathcal{N} \subseteq Y(F)$, we say F is *vertex stable with respect to* $\mathcal{N}$ when, for all $s_1s_2\ldots s_{n-1}s_n \in \mathscr{L}(F)$ with $s_n \in \mathcal{N}$, $\mathscr{V}_F(s_1s_2\ldots s_{n-1}) = \mathscr{V}_F(s_1s_2\ldots s_n)$.

Intuitively: having handled an input stream of symbols, processing an additional element from $\mathcal{N}$ does not cause a vertex stable device to move to a new vertex.