Composability in Cognitive Hierarchies

David Rajaratnam, Bernhard Hengst, Maurice Pagnucco, Claude Sammut, and Michael Thielscher

University of New South Wales, Australia
{daver,bernhardh,morri,claude,mit}@cse.unsw.edu.au

Abstract. This paper develops a theory of node composition in a formal framework for cognitive hierarchies. It builds on an existing model for the integration of symbolic and sub-symbolic representations in a robot architecture consisting of nodes in a hierarchy. A notion of behaviour equivalence between cognitive hierarchies is introduced and node composition operators that preserve this equivalence are defined. This work is significant in two respects. Firstly, it opens the way for a formal comparison between cognitive robotic systems. Secondly, composition, more precisely *decomposition*, has been shown to be important to many fields, and may therefore prove of practical benefit in the context of cognitive systems.

1 Introduction

Building robots capable of interacting with humans and operating in unstructured environments is an open challenge. Such a robot needs to combine low-level (sub-symbolic) sensor processing with high-level (symbolic) decision making. Currently, there are two basic approaches to this challenge. The first is to combine both symbolic and sub-symbolic representations into a rich language, such as Belle and Levesque's recent integration of probabilistic uncertainty into the Situation Calculus for robot localisation [4]. The second approach is to separate the required representations into interconnected sub-systems, for example using an architecture such as the Robot Control System (RCS) [1].

The advantage of a combined representation is that it allows for formal systems analysis. On the other hand such languages can be difficult to master and implement efficiently. For example, while GPU based processing is required for scalable 3D Simultaneous Localisation and Mapping (SLAM) [15], incorporating such specialised techniques into a rich expressive reasoner poses serious implementation challenges.

The alternative to the rich representation approach is to construct an architecture of loosely connected sub-systems. However, while having some practical advantages, typical architecture based approaches either impose a particular representation language or are only described informally using text descriptions and diagrams. The former lacks the flexibility required to combine disparate reasoning techniques, while the latter makes it difficult to formally establish properties of such a system.

The contribution of this paper is to take a step towards unifying these two approaches. We do this by extending a recently developed formal framework for integrating symbolic and sub-symbolic reasoning processes [9]. In particular, we develop a formal notion of behaviour equivalence between two cognitive systems, and prove theoretical properties for node composability that preserves behaviour equivalence.

The rest of this paper proceeds as follows. First, we introduce related work (Section 2) highlighting both the similarities and differences to existing research. We then provide a summary of the main features of the formal approach developed in [9]. In Section 4 we develop a notion of behaviour equivalence between cognitive hierarchies. Section 5 represents the main technical contribution of this paper, where node composition operators are introduced and formally shown to satisfy behaviour equivalence. It is further established that an arbitrary cognitive hierarchy can be reduced to a system consisting of a single node that nevertheless is formally behaviourally equivalent to the original. We conclude with a discussion and directions for future research.

2 Related Work

In this section we briefly highlight the different approaches to cognitive hierarchies, focusing on issues of formalisation and representation. We also briefly examine the varying notions of node composition that are prevalent in artificial intelligence (AI).

The use of hierarchies to build reasoning systems has a rich history in robotics and AI. While approaches vary, they can be broadly categorised into two groups: either they impose a particular representation language, or they are described informally.

The category of fixed language approaches include the popular cognitive architectures SOAR [14] and ACT-R [3], both of which employ symbolic based representations. Other examples include the logic-based Nilsson's triple-tower architecture [16], and the robot focused *dual dynamic* (DD) hierarchical behaviour system [12] formalised in terms of differential equations. The weakness of fixed language systems is that the representation language imposes limits on the type of problems that can be expressed. For example, the languages of SOAR and ACT-R are not easily applicable for representing the probabilistic uncertainty of robot localisation, while the differential equations used in the DD system cannot easily be adapted to perform traditional logical inference.

The alternative has been to describe architectures in an informal manner. The influential Robot Control System (RCS) model [1] consists only of textual descriptions and diagrams. While the seminal *subsumption architecture* [7] uses finite state machines for representing individual levels within the hierarchy [6], nevertheless the integration of these levels into an overall system is purely informal. An informal component integration is also true of the recent SOAR extension for dealing with sensor data [13]. Unfortunately, the informal approach has a number of weaknesses. In particular, it provides no basis on which to prove properties of the system as a whole, and there is no clear distinction between the architectural properties of an informal system and properties that arise from arbitrary implementation decisions.

Closely related to the construction of hierarchies is the notion of problem decomposition, where its benefits were evident in the context of hierarchical planning [17]. The more general notion of *factored planning* has been associated with the study of how to decompose a domain into sub-domains (know as *factors*) that can be solved independently and for which the solutions to each sub-domain can be combined [2, 5]. This work is generalised in the field of general game playing, where a game is decomposed into sub-games that are solved independently, ensuring that the combined solution solves the original game [8]. As the dual to decomposition, *behaviour composition* [10]

considers the problem of synthesising target behaviours from existing behaviours. However, the primitive behaviours in this case operate over a common transition system (i.e., the encoding of the environment) where as, composition in general-game playing considers composition over different transition systems. Our work is closer in intention to that of sub-game composition in general game playing than it is to behaviour composition.

3 Formal Architecture

In this section we summarise the formalisation developed in [9]. It avoids the weakness of existing formal models (discussed in Section 2) by adopting a meta-theoretic approach to cognitive hierarchies. In essence, it formalises the interaction between cognitive nodes in a hierarchy while making no commitments about the representation and reasoning mechanism within individual nodes. As such this framework complements, rather than competes with, existing hierarchical approaches.

3.1 Nodes

At the most basic level a cognitive hierarchy consists of a set of nodes. Nodes are tasked to achieve a goal or maximise future value. They have two primary functions: world-modelling and behaviour-generation. World-modelling is achieve through the maintenance of a *belief state*. A belief state is updated from lower-level nodes through *sensing*, which is the process of extracting *observations*. Behaviour-generation is achieved through a set of *policies*, where a policy maps a state to a set of actions and the current policy is determined by the actions of higher level nodes. A selected set of actions can also update a belief state, often referred to as an *expectation update*.

Definition 1. *A cognitive language is a tuple $\mathcal{L} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{O})$, where $\mathcal{S}$ is a set of belief states, $\mathcal{A}$ is a set of actions, $\mathcal{T}$ is a set of task parameters, and $\mathcal{O}$ is a set of observations. A cognitive node is a tuple $N = (\mathcal{L}, \Pi, \lambda, \tau, \gamma, s^0, \pi^0)$ s.t:*

- *$\mathcal{L}$ is the cognitive language for N, with initial belief state $s^0 \in \mathcal{S}$.*
- *Π a set of policies such that for all $\pi \in \Pi$, $\pi : \mathcal{S} \to 2^{\mathcal{A}}$, with initial policy $\pi^0 \in \Pi$.*
- *A policy selection function $\lambda \colon 2^{\mathcal{T}} \to \Pi$, s.t. $\lambda(\{\}) = \pi^0$.*
- *A observation update operator $\tau : 2^{\mathcal{O}} \times \mathcal{S} \to \mathcal{S}$.*
- *An action update operator $\gamma : 2^{\mathcal{A}} \times \mathcal{S} \to \mathcal{S}$.*

Note that Definition 1 provides a very abstract characterisation that captures only the interaction between nodes. No restrictions are made on internal representation and reasoning mechanisms, allowing, for example, a symbolic node to be created to encode a logical planner, or a stochastic node to encode a Kalman filter for robot localisation.

3.2 Cognitive Hierarchy

Nodes in the model are interlinked in a hierarchy, where the lowest level node corresponds to the interface to the real world, consisting of physical sensors and actuators (Figure 1). Sensing data is passed up the *abstraction hierarchy*, while actions are sent down the hierarchy, eventually resulting in physical actions.

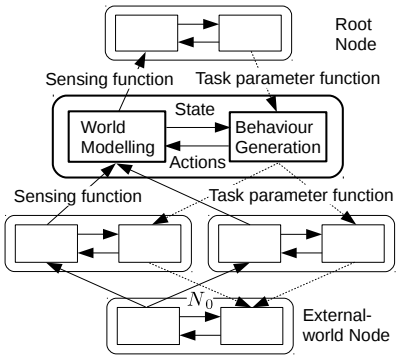


Fig. 1. An example cognitive hierarchy highlighting the sensing and action graphs.

Definition 2. *A cognitive hierarchy is a tuple $H = (\mathcal{N}, N_0, F)$ s.t:*

- *$\mathcal{N}$ is a set of cognitive nodes and $N_0 \in \mathcal{N}$ is a distinguished node corresponding to the external environment.*
- *F is a set of function pairs $\langle \phi_{i,j}, \psi_{j,i} \rangle \in F$ that connect nodes $N_i, N_j \in \mathcal{N}$ where:*
  - *$\phi_{i,j} : \mathcal{S}_i \rightarrow 2^{\mathcal{O}_j}$ is a sensing function, and*
  - *$\psi_{j,i} : 2^{\mathcal{A}_j} \rightarrow 2^{\mathcal{T}_i}$ is a task parameter function.*
- *Sensing graph: each $\phi_{i,j}$ represents an edge from node N_i to N_j and forms a directed acyclic graph (DAG) with N_0 as the unique source node of the graph.*
- *Action graph: the set of task parameter functions forms a converse to the sensing graph such that N_0 is the unique sink node of the graph.*

Definition 2 establishes how the sensing and task parameter functions connect nodes in the hierarchy; forming sensing and action graphs. Sensing functions extract observations from lower-level nodes, while task parameter functions translate higher-level actions into task parameters for lower level nodes. N_0 models the external world.

3.3 Active Cognitive Hierarchy

The notions of a cognitive node and cognitive hierarchy capture only static components of a system and require additional details to model dynamic operational behaviour.

Definition 3. *An active cognitive node is a tuple $Q = (N, s, \pi, a)$ where: 1) N is a cognitive node with $\mathcal{S}$, Π, and $\mathcal{A}$ being the corresponding set of belief states, set of policies, set of actions respectively, 2) $s \in \mathcal{S}$ is the current belief state, $\pi \in \Pi$ is the current policy, and $a \in 2^{\mathcal{A}}$ is the current set of actions.*

Essentially an active cognitive node couples a (static) cognitive node with some dynamic information; in particular the current belief state, policy and set of actions.

Definition 4. *An active cognitive hierarchy is a tuple $\mathcal{X} = (H, \mathcal{Q})$ where H is a cognitive hierarchy with set of cognitive nodes $\mathcal{N}$ such that for each $N \in \mathcal{N}$ there is a corresponding active cognitive node $Q = (N, s, \pi, a) \in \mathcal{Q}$ and vice-versa.*

The active cognitive hierarchy captures the dynamic state of the entire hierarchy at some instance in time, where each node in the hierarchy has a current state, current policy and current set of actions. An *initial active cognitive hierarchy* specifies initial configurations for every node in the hierarchy initialised with each node's initial belief state and initial policy, as well as an empty set of actions.

3.4 Cognitive Process Model

In order to operate in an environment the cognitive system requires a *process model* that applies the various functions of Definitions 1 and 2 to update the active cognitive hierarchy. For brevity, in this paper we provide only the main definition and an intuitive explanation; the interested reader is instead referred to [9] for more complete details.

Definition 5. *Let $\mathcal{X} = (H, \mathcal{Q})$ be an active cognitive hierarchy with $H = (\mathcal{N}, N_0, F)$. The process update of $\mathcal{X}$, written* **Update**$(\mathcal{X})$*, is an active cognitive hierarchy:*

$$\textbf{Update}(\mathcal{X}) \stackrel{\text{def}}{=} \textbf{ActionUpdate}(\textbf{SensingUpdate}(\mathcal{X}))$$

The functions **SensingUpdate** and **ActionUpdate** are intuitively easy to understand. **SensingUpdate** passes sensing information up the cognitive hierarchy, successively updating the belief states of the nodes in the hierarchy, while **ActionUpdate** passes actions down the hierarchy, causing the active policies and actions to be changed. These actions ultimately lead to task parameter changes for node N_0, which correspond to signals to the robot actuators. Crucially, both **SensingUpdate** and **ActionUpdate** update the nodes according to the partial ordering specified by the sensing and action graphs respectively. This guarantees that the functions are well-defined, since any sequence of node updates that satisfies the partial ordering will produce the same result.

The above formalism completes the technical background to this paper as developed in [9]. The remainder of this paper extends the formalism to allow for the specification of properties of behaviour equivalence and node composition.

4 Behaviour Equivalence

In this section we introduce the notion of behaviour equivalence between two cognitive hierarchies. This first involves establishing what it means for two cognitive systems to be behaviourally equivalent (with respect to the real-world) at some instant in time.

Definition 6. *Let $\mathcal{X}_i = (H_i, \mathcal{Q}_i)$ and $\mathcal{X}_j = (H_j, \mathcal{Q}_j)$ be two active cognitive hierarchies with $H_i = (\mathcal{N}_i, N_0, F_i)$ and $H_j = (\mathcal{N}_j, N_0, F_j)$. Then $\mathcal{X}_i$ and $\mathcal{X}_j$ are behaviour equivalent iff $T_i = T_j$, where:*

$$T_i = \bigcup \{\psi_{x,0}(s_x) \mid \langle \phi_{0,x}, \psi_{x,0} \rangle \in F_i \text{ for each } Q_x = (N_x, s_x, \pi_x, a_x) \in \mathcal{Q}_i\}$$
$$T_j = \bigcup \{\psi_{x,0}(s_x) \mid \langle \phi_{0,x}, \psi_{x,0} \rangle \in F_j \text{ for each } Q_x = (N_x, s_x, \pi_x, a_x) \in \mathcal{Q}_j\}$$

Definition 6 formalises the intuition that behaviour equivalence concerns how two cognitive systems behave with respect to the external world (i.e., N_0). The sets of task parameters T_i and T_j represent the actions that are carried out by the two cognitive systems. If the two sets are the same then the two systems perform the same actions.

Behaviour equivalence here is a theoretical notion, in effect saying that if we were able to observe the behaviour of the two systems operating at the same instant in time we would not be able to distinguish between them. This may not be possible to verify in practice, since we cannot actually replay time. Nevertheless the conceptual meaning is clear. We now consider how two active cognitive systems evolve *over* time.

Definition 7. *Let H_i and H_j be two cognitive hierarchies and $\mathcal{X}_i^0 = (H_i, \mathcal{Q}_i^0)$ and $\mathcal{X}_j^0 = (H_j, \mathcal{Q}_j^0)$ be their corresponding initial active cognitive hierarchies, such that N_0 is the distinguished node for both H_i and H_j. Then H_i and H_j are behaviour equivalent with respect to a cognitive process model Γ iff for every pair of sequences $[\mathcal{X}_i^0, \ldots, \mathcal{X}_i^n]$ and $[\mathcal{X}_j^0, \ldots, \mathcal{X}_j^n]$, where $\mathcal{X}_x^{k+1} = \Gamma(\mathcal{X}_x^k)$, each corresponding pair of active cognitive hierarchies $\mathcal{X}_i^k$ and $\mathcal{X}_j^k$ $(0 \leq k \leq n)$ are behaviour equivalent.*

Intuitively, Definition 7 establishes that two behaviourally equivalent cognitive hierarchies will be interchangeable in their operations over time. It is worth noting that because this property deals with operational systems as they evolve, therefore it is necessary to reference the process model that is used for updating the active nodes.

5 Node Composition

The idea behind node composition is to replace a pair of nodes in a cognitive hierarchy with a single node that nevertheless encapsulates the behavioural properties of the original. Formally this takes the form of *composition operators*. We now introduce *parallel* and *sequential* composition operators and establish their foundational properties.

5.1 Parallel and Sequential Composition Operators

As a technical preliminary we first provide the following convenience functions that will be used in the definitions to follow. Given a pair $p = \langle x, y \rangle$, let $fst(p) = x$ and $snd(p) = y$. Furthermore, we overload these definitions in the obvious way when applied to a set of pairs. Namely, $fst(S) = \{x | \langle x, y \rangle \in S\}$ and $snd(S) = \{y | \langle x, y \rangle \in S\}$.

The parallel and sequential composition operators have a number of commonalities which we capture by introducing a *partial* composition operator. The definition of this operator is somewhat lengthy and notationally heavy, however it is intuitively straightforward. Essentially, a new hierarchy is created that is identical to the old except that the two nodes being composed are replaced with a single new node. The representational aspects of the new node (i.e., the set of states, actions, task parameters, and observations) needs to combine the representational aspects of the original. The most lengthy aspect of the definition involves updating the sensing and task parameter functions that connect nodes. Functions in the original hierarchy that connect to at least one of the original two nodes must be integrated into a new function for the composed node.

Definition 8 (Partial Composition). *Let $H = (\mathcal{N}, N_0, F)$ be a cognitive hierarchy, and $N_i = (\mathcal{L}_i, \Pi_i, \lambda_i, \tau_i, \gamma_i, s_i^0, \pi_i^0)$ and $N_j = (\mathcal{L}_j, \Pi_j, \lambda_j, \tau_j, \gamma_j, s_j^0, \pi_j^0)$ be two cognitive nodes in H, where $\mathcal{L}_i = (\mathcal{S}_i, \mathcal{A}_i, \mathcal{T}_i, \mathcal{O}_i)$ and $\mathcal{L}_j = (\mathcal{S}_j, \mathcal{A}_j, \mathcal{T}_j, \mathcal{O}_j)$ are the cognitive languages for N_i and N_j respectively. The partial composition of N_i and N_j with respect to H is a cognitive hierarchy $H' = (\mathcal{N}', N_0, F')$ s.t:*

– $\mathcal{N}' = \mathcal{N} \setminus \{N_i, N_j\} \cup \{N_x\}$ *where* $N_x = (\mathcal{L}_x, \Pi_x, \lambda_x, \tau_x, \gamma_x, s^0_x, \pi^0_x)$ *where:*
  - $\mathcal{L}_x = (\mathcal{S}_x, \mathcal{A}_x, \mathcal{T}_x, \mathcal{O}_x)$,
  - $\mathcal{S}_x = \mathcal{S}_i \times \mathcal{S}_j$, *with initial state* $s_x = \langle s_i, s_j \rangle$.
  - $\mathcal{A}_x = 2^{\mathcal{A}_i} \times 2^{\mathcal{A}_j}$, $\mathcal{T}_x = 2^{\mathcal{T}_i} \times 2^{\mathcal{T}_j}$, $\mathcal{O}_x = 2^{\mathcal{O}_i} \times 2^{\mathcal{O}_j}$
  - $\gamma_x(a_x, \langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \begin{cases} \langle s_i, s_j \rangle & \text{if } |a_x| \neq 1, \\ \langle \gamma_i(a_i, s_i), \gamma_j(a_j, s_j) \rangle & \text{otherwise, for } a_x = \{\langle a_i, a_j \rangle\}. \end{cases}$

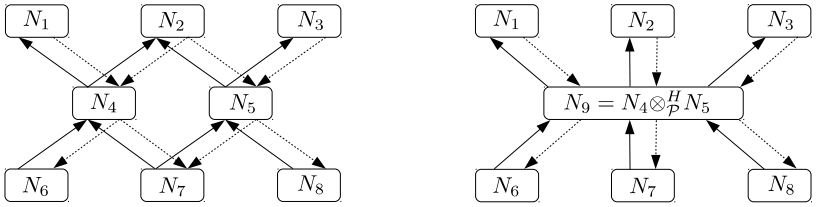
– $F' = F \setminus \{\langle \phi_{k,l}, \psi_{l,k} \rangle \in F \mid k \in \{i,j\} \text{ or } l \in \{i,j\}\} \cup \bigcup_{k=1,\ldots,6} F_k$ *where:*

  - $F_1 = \{\langle \phi_{w,x}, \psi_{x,w} \rangle \mid w$ *is a node;* $\langle \phi_{w,i}, \psi_{i,w} \rangle \in F$*;* $\langle \phi_{w,j}, \psi_{j,w} \rangle \in F\}$ *s.t:*
    * $\phi_{w,x}(s_w) \stackrel{\text{def}}{=} \{\langle \phi_{w,i}(s_w), \phi_{w,j}(s_w) \rangle\}$
    * $\psi_{x,w}(a_x) \stackrel{\text{def}}{=} \begin{cases} \{\} & \text{if } |a_x| \neq 1, \\ \psi_{i,w}(a_i) \cup \psi_{j,w}(a_j) & \text{otherwise, where } a_x = \{\langle a_i, a_j \rangle\}. \end{cases}$
  - $F_2 = \{\langle \phi_{w,x}, \psi_{x,w} \rangle \mid w$ *is a node;* $\langle \phi_{w,i}, \psi_{i,w} \rangle \in F$*;* $\langle \phi_{w,j}, \psi_{j,w} \rangle \notin F\}$ *s.t:*
    * $\phi_{w,x}(s_w) \stackrel{\text{def}}{=} \{\langle \phi_{w,i}(s_w), \{\} \rangle\}$
    * $\psi_{x,w}(a_s) \stackrel{\text{def}}{=} \begin{cases} \{\} & \text{if } |a_x| \neq 1, \\ \psi_{i,w}(a_i) & \text{otherwise, where } a_x = \{\langle a_i, a_j \rangle\}. \end{cases}$
  - $F_3 = \{\langle \phi_{w,x}, \psi_{x,w} \rangle \mid w$ *is a node;* $\langle \phi_{w,i}, \psi_{i,w} \rangle \notin F$*;* $\langle \phi_{w,j}, \psi_{j,w} \rangle \in F\}$ *s.t:*
    * $\phi_{w,x}(s_w) \stackrel{\text{def}}{=} \{\langle \{\}, \phi_{w,j}(s_w) \rangle\}$
    * $\psi_{x,w}(a_x) \stackrel{\text{def}}{=} \begin{cases} \{\} & \text{if } |a_x| \neq 1, \\ \psi_{j,w}(a_j) & \text{otherwise, where } a_x = \{\langle a_i, a_j \rangle\}. \end{cases}$
  - $F_4 = \{\langle \phi_{x,w}, \psi_{w,x} \rangle \mid w$ *is a node;* $\langle \phi_{i,w}, \psi_{w,i} \rangle \in F$*;* $\langle \phi_{j,w}, \psi_{w,j} \rangle \in F\}$ *s.t:*
    * $\phi_{x,w}(\langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \phi_{i,w}(s_i) \cup \phi_{j,w}(s_j)$
    * $\psi_{w,x}(a_w) \stackrel{\text{def}}{=} \{\langle \psi_{w,i}(a_w), \psi_{w,j}(a_w) \rangle\}$.
  - $F_5 = \{\langle \phi_{x,w}, \psi_{w,x} \rangle \mid w$ *is a node;* $\langle \phi_{i,w}, \psi_{w,i} \rangle \in F$*;* $\langle \phi_{j,w}, \psi_{w,j} \rangle \notin F\}$ *s.t:*
    * $\phi_{x,w}(\langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \phi_{i,w}(s_i)$
    * $\psi_{w,x}(a_w) \stackrel{\text{def}}{=} \{\langle \psi_{w,i}(a_w), \{\} \rangle\}$.
  - $F_6 = \{\langle \phi_{x,w}, \psi_{w,x} \rangle \mid w$ *is a node;* $\langle \phi_{i,w}, \psi_{w,i} \rangle \notin F$*;* $\langle \phi_{j,w}, \psi_{w,j} \rangle \in F\}$ *s.t:*
    * $\phi_{x,w}(\langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \phi_{j,w}(s_j)$
    * $\psi_{w,x}(a_w) \stackrel{\text{def}}{=} \{\langle \{\}, \psi_{w,j}(a_w) \rangle\}$

It is worth highlighting some significant aspects of Definition 8. Firstly, the set of states for the composed node consists of the cross-product of the set of states for the original nodes. Hence exactly the same combination of states that can be represented by the two nodes in the original hierarchy can also be represented by the new node. Secondly, the set of actions (resp. task parameters and observations) of the new node consist of the cross-product of the *powerset* of the same property in the original two nodes. This is necessary to allow for the arbitrary combination of subsets of the original set of actions (resp. task parameters and observations). Finally, some of the connections in the old hierarchy need to be preserved while others need to be merged. $F_1, \ldots, F_6$ in

the definition cover the cases where connector functions need to be merged. The modified functions are defined in terms of the originals in order to preserve their behaviour, but are modified to deal with the merged representation of the new node.

We now turn to completing the definitions of the different composition operators. The more straightforward case is the parallel composition operator. This captures the situation where the two nodes being composed are not reachable from each other in the cognitive hierarchy (e.g., nodes N_4 and N_5 in Figure 2(a)).



(a) Pre-composition (b) Post-composition

Fig. 2. Parallel composition of nodes N_4 and N_5 in a cognitive hierarchy. Solid lines represent the sensing functions between nodes and dotted-lines represent the task parameter functions.

Definition 9 (Parallel Composition). *Let $H = (\mathcal{N}, N_0, F)$ be a cognitive hierarchy, and $N_i = (\mathcal{L}_i, \Pi_i, \lambda_i, \tau_i, \gamma_i, s_i^0, \pi_i^0)$ and $N_j = (\mathcal{L}_j, \Pi_j, \lambda_j, \tau_j, \gamma_j, s_j^0, \pi_j^0)$, where $\mathcal{L}_i = (\mathcal{S}_i, \mathcal{A}_i, \mathcal{T}_i, \mathcal{O}_i)$ and $\mathcal{L}_j = (\mathcal{S}_j, \mathcal{A}_j, \mathcal{T}_j, \mathcal{O}_j)$, be cognitive nodes in $\mathcal{N}$, that are distinct from each other and N_0, and furthermore that $N_i \not\leq N_j$ and $N_j \not\leq N_i$ for the partial order $\leq$ induced by the sensing graph of H. The parallel composition of N_i and N_j with respect to H (written $N_i \otimes^H_{\mathcal{P}} N_j$) is a partial composition operator (Definition 8) with the additional requirements that for the composed node $N_x = (\mathcal{L}_x, \Pi_x, \lambda_x, \tau_x, \gamma_x, s_x^0, \pi_x^0)$:*

- $\Pi_x = \{\pi_i \diamond \pi_j : \mathcal{S}_x \to 2^{\mathcal{A}_x} \mid \pi_i \in \Pi_i \text{ and } \pi_j \in \Pi_j\}$, *and* $\pi_x^0 = \pi_i^0 \diamond \pi_j^0$.
- $\lambda_x(t_x) \stackrel{\text{def}}{=} \lambda_i(T_i) \diamond \lambda_i(T_j)$, *for any* $t_x \subseteq \mathcal{T}_x$, *and where* $T_i = \bigcup_{T \in fst(t_x)} T$ *and* $T_j = \bigcup_{T \in snd(t_x)} T$.
- $\tau_x(o_x, \langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \langle \tau_i(O_i, s_i), \tau_j(O_j, s_j) \rangle$, *for any* $o_x \subseteq \mathcal{O}_x, s_i \in \mathcal{S}_i, s_j \in \mathcal{S}_j$, *and where* $O_i = \bigcup_{O \in fst(o_x)} O$ *and* $O_j = \bigcup_{O \in snd(o_x)} O$.

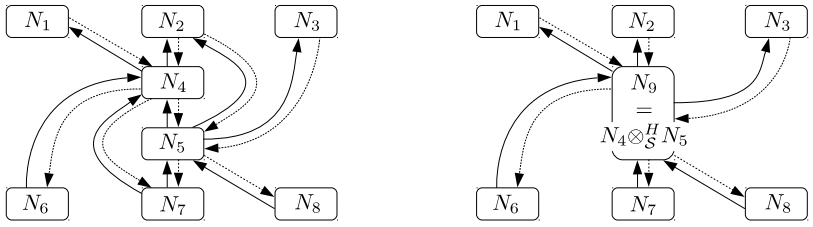
where the composition $\pi_x = \pi_i \diamond \pi_j$ is defined for $\pi_i \in \Pi_i$ and $\pi_j \in \Pi_j$ as:

$$\pi_x(\langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \{\langle \pi_i(s_i), \pi_j(s_j) \rangle\}, \text{ for any } s_i \in \mathcal{S}_i, s_j \in \mathcal{S}_j.$$

There are a number of aspects to the parallel composition operator (Definition 9) that are worth highlighting. Firstly, the restriction that the two nodes being composed are not comparable under the sensing (or action) graph's partial ordering means that the two nodes are unrelated and the result (sensing or actions) of one node will not influence the results of the other node. This makes the definition of various node functions simpler. Secondly, the set of policies of the composed node are the cross-product of

the policies of the two original nodes, and each composed policy simply applies the underlying policy to the appropriate component.

We now consider the more complicated case where the nodes to be composed are related by the partial ordering and therefore the results of one node effects the behaviour of the other node (e.g., nodes N_4 and N_5 in Figure 3(a)).



(a) Pre-composition (b) Post-composition

Fig. 3. Sequential composition of nodes N_4 and N_5 in a cognitive hierarchy. Solid lines represent the sensing functions between nodes and dotted-lines represent the task parameter functions.

Definition 10 (Sequential Composition). *Let $H = (\mathcal{N}, N_0, F)$ be a cognitive hierarchy, and $N_i = (\mathcal{L}_i, \Pi_i, \lambda_i, \tau_i, \gamma_i, s_i^0, \pi_i^0)$ and $N_j = (\mathcal{L}_j, \Pi_j, \lambda_j, \tau_j, \gamma_j, s_j^0, \pi_j^0)$, where $\mathcal{L}_i = (\mathcal{S}_i, \mathcal{A}_i, \mathcal{T}_i, \mathcal{O}_i)$ and $\mathcal{L}_j = (\mathcal{S}_j, \mathcal{A}_j, \mathcal{T}_j, \mathcal{O}_j)$, be cognitive nodes in $\mathcal{N}$, that are distinct from each other and N_0, and furthermore, $N_j \leq N_i$ for the partial order $\leq$ induced by the sensing graph of H and there does not exist a distinct node N_k such that $N_j \leq N_k$ and $N_k \leq N_i$. The sequential composition of N_i and N_j with respect to H (written $N_i \otimes_{\mathcal{S}}^{H} N_j$) is a partial composition operator (Definition 8) with the additional requirements that for the composed node $N_x = (\mathcal{L}_x, \Pi_x, \lambda_x, \tau_x, \gamma_x, s_x^0, \pi_x^0)$:*

- $\Pi_x = \{\pi_i \diamond t_j : \mathcal{S}_x \to 2^{\mathcal{A}_x} | t_j \subseteq \mathcal{T}_j \text{ and } \pi_i \in \Pi_i\}$, *and* $\pi_x^0 = \pi_i^0 \diamond \{\}$.
- $\lambda_x(t_x) \stackrel{\text{def}}{=} \lambda_i(T_i) \diamond T_j$, *for any* $t_x \subseteq \mathcal{T}_x$, *and where* $T_i = \bigcup_{T \in fst(t_x)} T$ *and* $T_j = \bigcup_{T \in snd(t_x)} T$.
- $\tau_x(o_x, \langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \langle \tau_i(O_i \cup \phi_{j,i}(s_j'), s_i), s_j' \rangle$, *for any* $o_x \subseteq \mathcal{O}_x, s_i \in \mathcal{S}_i, s_j \in \mathcal{S}_j$, *and where* $O_i = \bigcup_{O \in fst(o_x)} O$, *and* $s_j' = \tau_j(O_j, s_j)$ *such that* $O_j = \bigcup_{O \in snd(o_x)} O$.

where the composition $\pi_x = \pi_i \diamond t_j$ is defined for $\pi_i \in \Pi_i$ and $t_j \subseteq \mathcal{T}_j$ as:

$$\pi_x(\langle s_i, s_j \rangle) \stackrel{\text{def}}{=} \{\langle \pi_i(s_i), \lambda_j(\psi_{i,j}(\pi_i(s_i)) \cup t_j)(s_j) \rangle\}, \text{ for any } s_i \in \mathcal{S}_i, s_j \in \mathcal{S}_j.$$

The definition of the sequential composition operator is more complicated than the parallel case because of the need to capture the interaction between the original two nodes. For example, consider the construction of the policies for the sequentially composed node in Figure 3. Every action taken in N_4 will (potentially) change the policy for N_5. This behaviour needs to be preserved by the composition operator. So when the policy is applied for the composed node N_9, internally the behaviour that is applied to the N_5 component of the composed state depends on the result of the N_4 component of the composed state. A similar dependency exists for the sensing update operator.

5.2 Properties

We now consider the properties of the composition operators. Firstly, it is necessary to establish that they are in fact well-defined and generate valid cognitive hierarchies.

Lemma 1. *For a cognitive hierarchy $H = (\mathcal{N}, N_0, F)$ and nodes $N_i, N_j \in \mathcal{N}$, the compositions $N_i \otimes^H_{\mathcal{P}} N_j$ and $N_i \otimes^H_{\mathcal{S}} N_j$ are well-defined cognitive hierarchies, when applied subject to the restrictions for parallel and sequential composition respectively.*

Proof Sketch. By inspection. The signatures satisfy the requirements of a cognitive node (Definition 1) and hierarchy (Definition 2). The restrictions on applying $N_i \otimes^H_{\mathcal{P}} N_j$ and $N_i \otimes^H_{\mathcal{S}} N_j$ ensures that the resulting sensing/action graphs are appropriate DAGs. □

Now we establish that the composition operators satisfy the property of behaviour equivalence. We do this separately for each composition operator.

Theorem 1. *Let $H = (\mathcal{N}, N_0, F)$ be a cognitive hierarchy and N_i and N_j be nodes in H that satisfy the requirements of the parallel composition operator (i.e., N_i and N_j are distinct and unrelated under the sensing graph partial ordering). Then H and $N_i \otimes^H_{\mathcal{P}} N_j$ are behaviour equivalent with respect to the cognitive process model* **Update**.

Proof Sketch. Let $H_c = N_i \otimes^H_{\mathcal{P}} N_j$ and let $\mathcal{X}^0 = (H, \mathcal{Q}^0)$ and $\mathcal{X}^0_c = (H_c, \mathcal{Q}^0_c)$ be the initial active cognitive hierarchies for H and H_c respectively. Now let $\mathcal{X}^{i+1} = \textbf{Update}(\mathcal{X}^i)$ and $\mathcal{X}^{i+1}_c = \textbf{Update}(\mathcal{X}^i_c)$. We can show by induction that for all i that $\mathcal{X}^{i+1}$ is behaviour equivalent to $\mathcal{X}^{i+1}_c$. Proving the base case requires comparing $\mathcal{Q}^0$ and $\mathcal{Q}^0_c$ to ensure that the set of task parameters of $\mathcal{Q}^0$ and $\mathcal{Q}^0_c$ w.r.t. N_0 are the same (Definition 6). The induction step requires following every aspect of the application of the **Update** function to confirm that the two cognitive hierarchies are updated so as to preserve the beliefs and actions of the original hierarchy. This is lengthy but essentially straightforward. □

Theorem 2. *Let $H = (\mathcal{N}, N_0, F)$ be a cognitive hierarchy and nodes N_i and N_j be nodes in H that satisfy the requirements of the sequential composition operator (i.e., that N_i and N_j are distinct, $N_j \leq N_i$ and there is no distinct node N_k where $N_j \leq N_k$ and $N_k \leq N_i$ under the sensing graph partial ordering). Then H and $N_i \otimes^H_{\mathcal{S}} N_j$ are behaviour equivalent with respect to the cognitive process model* **Update**.

Proof Sketch. The proof follows the same pattern as the parallel case (Theorem 1). □

Theorems 1 and 2 establish that the application of the parallel and sequential composition operators does indeed preserve behaviour equivalence. But a composed cognitive hierarchy is just another cognitive hierarchy. Consequently, the composition process can potentially be repeated and it is not difficult to see that this process can be repeated successively until the hierarchy consists of only a single non-N_0 node.

Theorem 3. *Let $H = (\mathcal{N}, N_0, F)$ be an arbitrary cognitive hierarchy. Then there exists a cognitive hierarchy $H' = (\mathcal{N}', N_0, F')$ such that H and H' are behaviour equivalent with respect to cognitive process model* **Update** *and $|\mathcal{N}'| = 2$.*

Proof Sketch. Trivial for $|\mathcal{N}| = 2$. For other cases successively construct cognitive hierarchies by applying the parallel or sequential composition operators until a behaviour equivalent hierarchy of size 2 is reached. Showing that this is possible reduces to a graph property of a DAG that allows one or the other composition operator to be applied. □

6 Discussion

The formal properties established in this paper have a number of interesting consequences. In the first place, the fact that any cognitive hierarchy is equivalent to a hierarchy with only two nodes (Theorem 3) establishes that the size of the cognitive hierarchy is unrelated to its behaviour. Rather the choice of the hierarchy is influenced by other factors such as the designers familiarity with a particular representational language, or theoretical and practical computational concerns, such as being able to apply pre-existing tools and techniques in components of a node.

Secondly, while our composition formalism does not in itself provide a method to automatically decompose existing systems it does provide some understanding about what sorts of features to look for in determining whether or not decomposition is possible. If the belief state of an existing node can be factored into two orthogonal components (of size N and M) and if the updating of these components can be performed either independently or sequentially then decomposition will likely follow. Importantly this could have computational benefits; for example instead of searching over an $N \times M$ state space the combined search for the two nodes would be over an $N + M$ state space. This is one of the key motivations that has lead to the study of decomposition in other AI fields (e.g., factored planning [5], general game playing [8]), and in hierarchical reinforcement learning (HRL) [11].

7 Conclusion and Future Work

In this paper we developed a theory of node composition for a formal model of cognitive hierarchies. Two node composition operators were defined and shown to satisfy a notion of behaviour equivalence. The property of behaviour equivalence was established with respect to changes manifested by a cognitive system on an external environment. Such a definition is very general because it makes no reference to the internal state or representation used by a particular cognitive system. Hence, for example, two cognitive systems could be implemented using very different algorithms and techniques but could nevertheless satisfy the property of behaviour equivalence.

A second development of this paper was to establish that for any cognitive hierarchy there is an equivalent one consisting of only a single node (excluding the node that represents the external world). Furthermore, the combination of the parallel and sequential node composition operators was shown to be sufficient to capture this property.

Finally, we argued that the results developed in this paper can serve as a tool to bridge between cognitive systems. This opens up interesting directions for future research to validate these ideas. In particular it is our intention to examine the extent to which a system modelled using a rich expressive language that can deal with both symbolic and sub-symbolic information, such as the language introduced in [4], can be functionally decomposed into a behaviourally equivalent system consisting of a hierarchy of nodes with simpler individual representations; potentially separating the symbolic and sub-symbolic components. This would allow a system modelled formally using an expressive language, from which formal properties can be established, to then be efficiently implemented as a provably equivalent system consisting of decomposed nodes that can take advantage of existing high-performance tools and techniques.

Acknowledgements This material is based upon work supported by the Asian Office of Aerospace Research and Development (AOARD) under Award No: FA2386-15-1-0005. This research was also supported under Australian Research Council's (ARC) *Discovery Projects* funding scheme (project number DP 150103035). Michael Thielscher is also affiliated with the University of Western Sydney.

Disclaimer Opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the AOARD.

References

1. Albus, J.S.: Engineering of Mind: An Introduction to the Science of Intelligent Systems. Wiley (2001)
2. Amir, E., Engelhardt, B.: Factored planning. In: Proc. of IJCAI. pp. 929–935. Morgan Kaufmann (2003)
3. Anderson, J.R.: Rules of the Mind. Lawrence Erlbaum Associates Inc (July 1993)
4. Belle, V., Levesque, H.J.: Robot location estimation in the situation calculus. Journal of Applied Logic 13(4), 397–413 (2015)
5. Brafman, R.I., Domshlak, C.: Factored planning: How, when, and when not. In: Proc. of AAAI. pp. 809–814 (2006)
6. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2(1), 14–23 (Mar 1986)
7. Brooks, R.A.: Elephants don't play chess. Robotics and Autonomous Systems 6 pp. 3–15 (1990)
8. Cerexhe, T.J., Rajaratnam, D., Saffidine, A., Thielscher, M.: A systematic solution to the (de-)composition problem in general game playing. In: Proc. of ECAI. pp. 195–200 (2014)
9. Clark, K., Hengst, B., Pagnucco, M., Rajaratnam, D., Robinson, P., Sammut, C., Thielscher, M.: A framework for integrating symbolic and sub-symbolic representations. In: Proc. of IJCAI. pp. 2486–2492 (2016)
10. De Giacomo, G., Sardiña, S.: Automatic synthesis of new behaviors from a library of available behaviors. In: Proc. of IJCAI. pp. 1866–1871 (2007)
11. Hengst, B.: Hierarchical approaches. In: Wiering, M., van Otterlo, M. (eds.) Reinforcement Learning: State of the Art, Adaptation, Learning, and Optimization, vol. 12. Springer (2011)
12. Jaeger, H., Christaller, T.: Dual dynamics: Designing behavior systems for autonomous robots. Artificial Life and Robotics 2(3), 108–112 (1998)
13. Laird, J.E., Kinkade, K.R., Mohan, S., Xu, J.Z.: Cognitive robotics using the soar cognitive architecture. Cognitive Robotics AAAI Technical Report WS-12-06 pp. 46–54 (2012)
14. Laird, J.E., Newell, A., Rosenbloom, P.S.: SOAR: An architecture for general intelligence. Artif. Intell. 33(1), 1–64 (Sep 1987)
15. Lee, D., Kim, H., Myung, H.: GPU-based real-time RGB-D 3D SLAM. In: Proc. Ubiquitous Robots and Ambient Intelligence (URAI). pp. 46–48. IEEE (2012)
16. Nilsson, N.: Teleo-reactive programs and the triple-tower architecture. Electronic Transactions on Artificial Intelligence 5, 99–110 (2001)
17. Sacerdoti, E.D.: Planning in a hierarchy of abstraction spaces. Artificial Intelligence 5(2), 115 – 135 (1974)