A Ten Year Roadmap to Machines with Common Sense

Push Singh and Marvin Minsky

16 September 2002

1 Introduction

Today, we have programs that beat people at chess, or diagnose serious heart attacks. Some can solve complicated equations. Others can recognize voices and faces, assemble cars in factories, and even pilot ships and planes. But none of them can make a bed, or clean a house, or baby-sit—or do most of the things that our two-year-olds do. Why can't we make our computers do so many things that most persons can do?

The trouble is that each modern program solves only one kind of problem—whereas children learn tens of thousands of skills. Why are our programs less versatile? We suspect that computers already have enough power and speed to compete with us—and what's wrong is in how we're been programming them.

1. We program them only for specialized jobs, without giving them any more "general knowledge". This is why, when we interact with them, they usually can't understand what we mean.

2. The structure of our programs are generally too "procedural." They mainly execute predetermined routines, without thinking about when these are appropriate. In other words, they are not "reflective" enough.

So, we shouldn't expect much resourcefulness from programs designed to work only within some particular realm, by using only a single technique. If a program works in only one way, then it gets stuck when that method fails. But a program with multiple "ways to think" could behave as humans do: whenever anything goes wrong, a person can switch to some other approach.

How can we make them more resourceful? We can help them with this by endowing them with large collections of general knowledge, and with multiple methods for using these—in short, by giving them more kinds of "ways to think."

Answer: we need to develop an Architecture for doing such things—a framework which lets us integrate what we have learned so far in artificial intelligence to build systems far more diverse, robust and resourceful than those we have been able to build to date.

This document outlines a 10 year roadmap to such an Architecture[1].

2 Key Components

Our roadmap will begin with a description of the key components of any commonsense computing system. These can be divided roughly into the substrate, core techniques, composite techniques, meta-management, and large-scale ways of thinking.

- **Framework.** We must develop a framework that allows many types of problem solving methods and representations to interoperate. Such a framework is likely to take the form of a multiagent system framework that consists of a set of resources and protocols that support basic methods for linking multiple representations, representing and invoking many types of agents, agent communication, and self-representation so that the system can describe, reflect on, and modify its own processing.

- **Core inference methods.** We must develop a large library of 'core' inference methods out of which more complex skills can be assembled. These will span the range of known inference methods, for example, Bayesian inference, logical theorem proving, analogical and case-based reasoning, neural

[1] This roadmap is based on the conclusions of two symposia we held earlier this year to discuss building a system with common sense (McCarthy et al. (2002); Minsky and Singh (2002)).

networks, rule-based systems, and other forms of inference. This library will serve as the basic resource for building more specialized agents that reason about more specific domains.

- **Core realms of thinking.** We must develop a large library of agents that can reason effectively about the most basic realms of human common sense: the spatial, visual, bodily, motor, social, psychological, reflective, linguistic, and other realms. Each agent will be built using the library of core inference methods and embody the commonsense knowledge required to model some particular realm. Each such realm will likely require its own special representations, methods of reasoning, techniques for learning, and so forth.

- **Core learning methods.** We must develop a large collection of procedures for learning various types of common sense. Just as a single method of inference is unlikely to work across a wide range of problems, a single method of learning is unlikely to be able to learn a broad range of knowledge. We need a broad range of techniques for learning about the shapes of objects, about the structure of typical events, semantic structures like verb frames, mappings between visual and linguistic descriptions, and so forth.

- **Composite inference methods.** We must develop problem solving methods that capture ways of thinking too complex and heterogeneous to be based on any single core inference method. Instead, such problem solving methods will capture those thought processes that require as constituents a variety of inference types, for example, for diagnosis and debugging, scheduling and planning, natural language parsing and generation, scene analysis, and other complex 'composite' skills.

- **Composite learning methods.** We must develop learning methods that embody more complicated learning skills, such as notions of practice, knowledge organization and repair, posing new goals, seeking out information from books and other minds, and the other more complex strategies that humans use to make improvements in themselves. Such learning skills require composite systems, and cannot be embodied by any primitive machine learning algorithm.

- **Composite commonsense knowledge.** We must develop ways to acquire knowledge that cut across many realms of thinking. For example, many verbs denote actions that have consequences in the physical realm, the social realm, and so on. Similarly, descriptions of objects, events, situations, scenes, and stories, plans, and so forth will need to be partly described in each of many realms. Such knowledge may be acquired in many ways, for example through multi-model knowledge entry (of facts, stories, diagrams, and so forth), by data mining large text corpora, and through direct learning from exposure to and interaction with the environment through rich arrays of sensors and robots with sensory-motor systems.

- **Self-reflection and meta-management.** We must develop techniques for selecting the appropriate societies of agents (composed of both core and composite methods) for solving particular types of problems. These techniques are likely to consist largely of certain special types of meta-knowledge that can characterize problems according to their general features, and that map those problems onto particular types of solution techniques. The highest level 'meta-managements' layers of a commonsense computing system will use this knowledge to control which agents should be made active, as a function of the current problem the system faces.

- **Large-scale ways of thinking.** We must develop the types of large-scale societies of agents appropriate to solving different kinds of practical problems. For example, we will want to assemble societies of agents that can learn about a new subject, look for unusual situations in a narrative, converse with the user to help them learn about some topic, and so forth. These large-scale ways of thinking will constitute the "applications" of commonsense computing systems.

Figure 1 below summarizes the dependencies between these components.

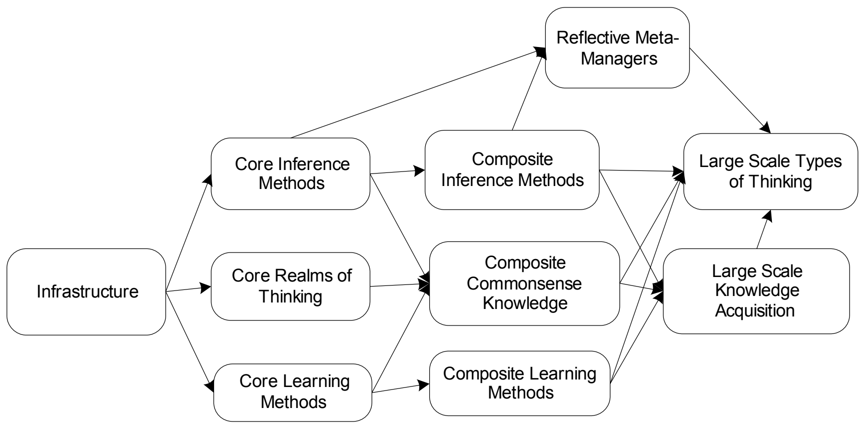


Figure 1. Dependencies between components of commonsense system

3 Main Obstacles

We will now list some of the main obstacles that any such effort is bound to encounter.

- **Can we build very large societies of agents?** As we build larger and larger systems, they eventually 'collapse under their own weight' by becoming too unwieldy to manage, too large to debug, or too complicated to understand. We must develop new ways to manage the complexity of artificial intelligence systems as they accumulate more representations, knowledge, methods of inference, and so forth. This will likely require developing new organizational strategies for agents, and also finding ways to hand over some of the responsibility for managing the system from the programmer to the system itself.

- **Can we find a mapping between problem types and solution methods?** Artificial intelligence can be regarded as a vast toolbox of techniques, but we still have a poor understanding of what types of problems each tool is suited for and what types it is not. Levesque and Brachman (1987) suggested one approach to answering this question, which was to understand the tradeoff between the expressivity and tractability of various representation languages, and Minsky (2001) suggests a different approach of characterizing general features of problem spaces and how those features could be used to predict the success of particular inference methods.

- **Can we determine the kinds of representations we need?** There have been few efforts to produce a comprehensive catalog of the kinds of knowledge needed by a commonsense computing system. One of the few forays into this area is the upper level ontology of Cyc, which provides an overall sense of the structure of commonsense knowledge. However, this approach lacks the kinds of procedural knowledge needed to understand how to use knowledge, for example, the problems that a knowledge item could help solve, ways of thinking that a knowledge item could participate in, known arguments for and against a knowledge item, and means of adapting a knowledge item to new contexts.

- **Can we find ways to bridge multiple representations?** To build systems with common sense we will need to find ways to map between different ways of describing things, for example, between linguistic and visual descriptions of scenes and events. This idea goes well beyond such overt mental systems, for in any interpretation task the more representations you have, the better, for every representation will contribute new kinds of suggestions and constraints on the problem. Nevertheless, today there are few systems that link multiple representations because of the complexity and unknowns involved in using many distinct but linked representations in a system.

- **Can we build sufficiently large commonsense knowledge bases?** It would be terrible if our systems needed to be trained from scratch every time they were placed in a new environment or given a new problem. We must renew our efforts to build the large commonsense knowledge bases that will provide our systems with the models they need to understand the world and give meanings to our words. In addition to traditional knowledge acquisition methods, we must explore new avenues like acquiring knowledge from large text corpora, from the general public and other non-knowledge engineers, and through novel machine learning techniques that allow direct learning from exposure to and interaction with the environment through passive sensory systems and robotic sensory-motor systems.

- **Can we develop components that can manage the problems that appear within them?** Our systems need the capacity to reflect on their own processing so that they can autonomously manage their own resources. We lack knowledge in this area—while we already have techniques for instrumenting and tracing processes, we lack a good understanding of how to teach our processes how to make the right changes to themselves as they begin to falter. To do this we must develop ways to not just encode the kinds of problem-solution mappings described previously, but we must also encode the various ways those solutions might fail, and how to adapt those solutions depending on the nature of the failures.

4 Ten Year Roadmap

Appendix A outlines a ten year roadmap for building the requisite components of the Commonsense Computing System.

5 References

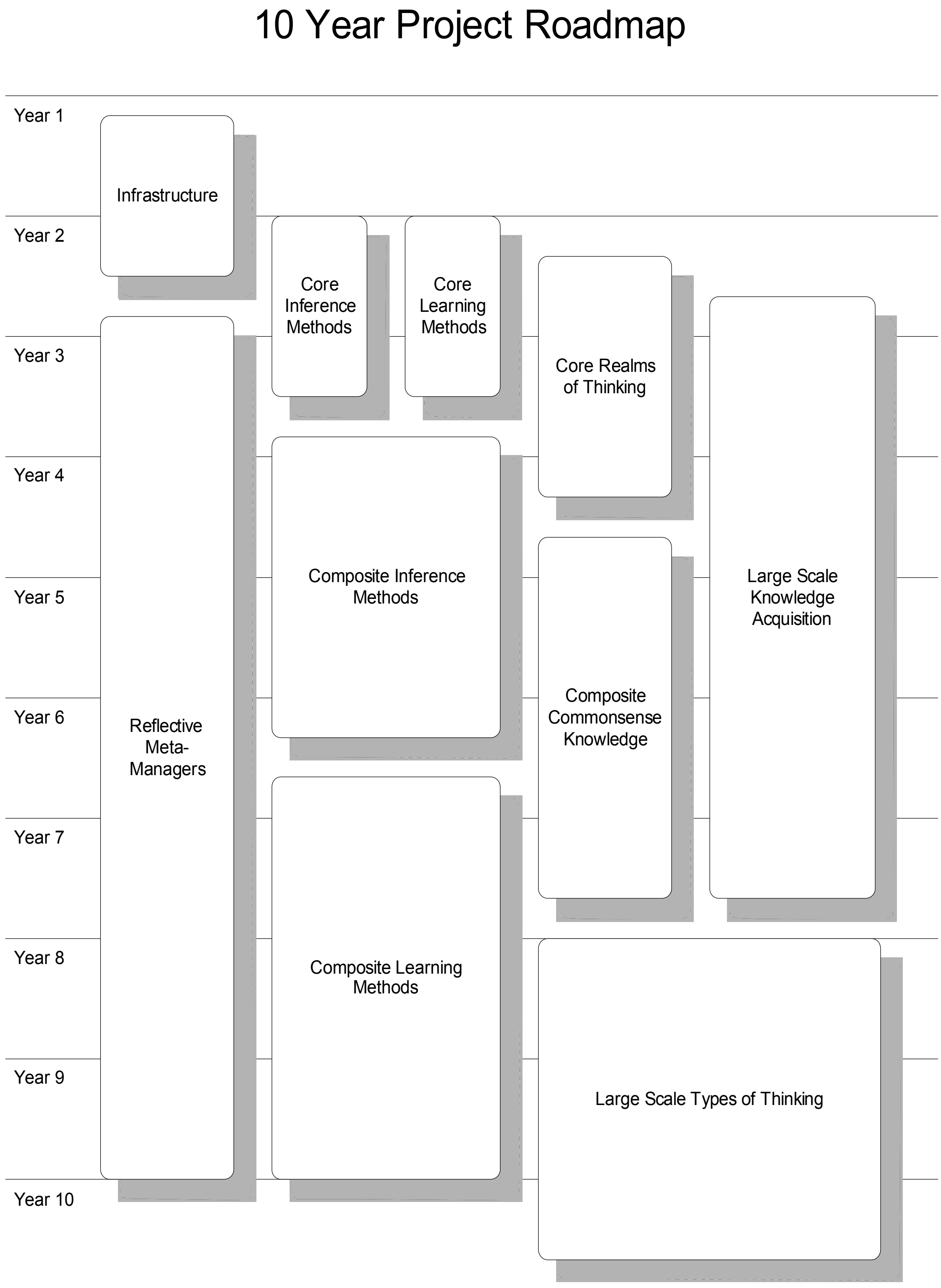
Levesque, Hector and Brachman, Ron. (1987). Expressiveness and tractability in knowledge representation and reasoning. *Computational Intelligence*, 3, 78-93.

McCarthy, John, Minsky, Marvin, Sloman, Aaron, Gong, Leiguang, Lau, Tessa, Morgenstern, Leora, Mueller, Erik T., Riecken, Doug, Singh, Moninder, & Singh, Push (2002). An architecture of diversity for commonsense reasoning. *IBM Systems Journal*, *41*(3), 530-539.

Minsky, Marvin (2000). Commonsense-based interfaces. *Communications of the ACM,* 43(8), 67-73.

Minsky, Marvin and Singh, Push (2002). The St. Thomas Commonsense Symposium. Unpublished report. Retrieved from http://www.media.mit.edu/people/push/StThomasSymposium.pdf

6 Appendix A


10 Year Project Roadmap
Year 1
Year 2
Year 3
Year 4
Year 5
Year 6
Year 7
Year 8
Year 9
Year 10
Infrastructure
Core Inference Methods
Core Learning Methods
Core Realms of Thinking
Large Scale Knowledge Acquisition
Reflective Meta-Managers
Composite Inference Methods
Composite Commonsense Knowledge
Composite Learning Methods
Large Scale Types of Thinking