Introduction to Cognitive Science: Notes

III: Representing Action in the World (Planning)

- Readings for this section: *McCarthy and Hayes 1969; *Shanahan 1997

IV: Representing Action in the World (Planning)

- Basic Dynamic Logic:

  (1) $n \geq 0 \Rightarrow [\alpha](y = F(n))$

  "If n is positive, α-ing always sets y equal to $F(n)$".

- In the real world, such rules are *defaults*, but they are still *deterministic*.

- The particular dynamic logic that we are dealing with here is one that includes the following dynamic axiom (the operator ; is *sequence*, the composition of functions of type *situation* $\rightarrow$ *situation*):

  (2) $[\alpha][\beta]P \Rightarrow [\alpha;\beta]P$

- Composition is one of the most primitive *combinators*, or operations combining functions, which Curry and Feys (1958) call **B**, writing the above sequence $\alpha;\beta$ as $\mathbf{B}\beta\alpha$, where

  (3) $\mathbf{B}\beta\alpha \equiv \lambda s.\beta(\alpha(s))$

Dynamic Logic: Actions as Accessibility

- The actions $\alpha, \beta, \ldots$ can be seen as defining the accessibility relation for a modal logic with an S4 model:

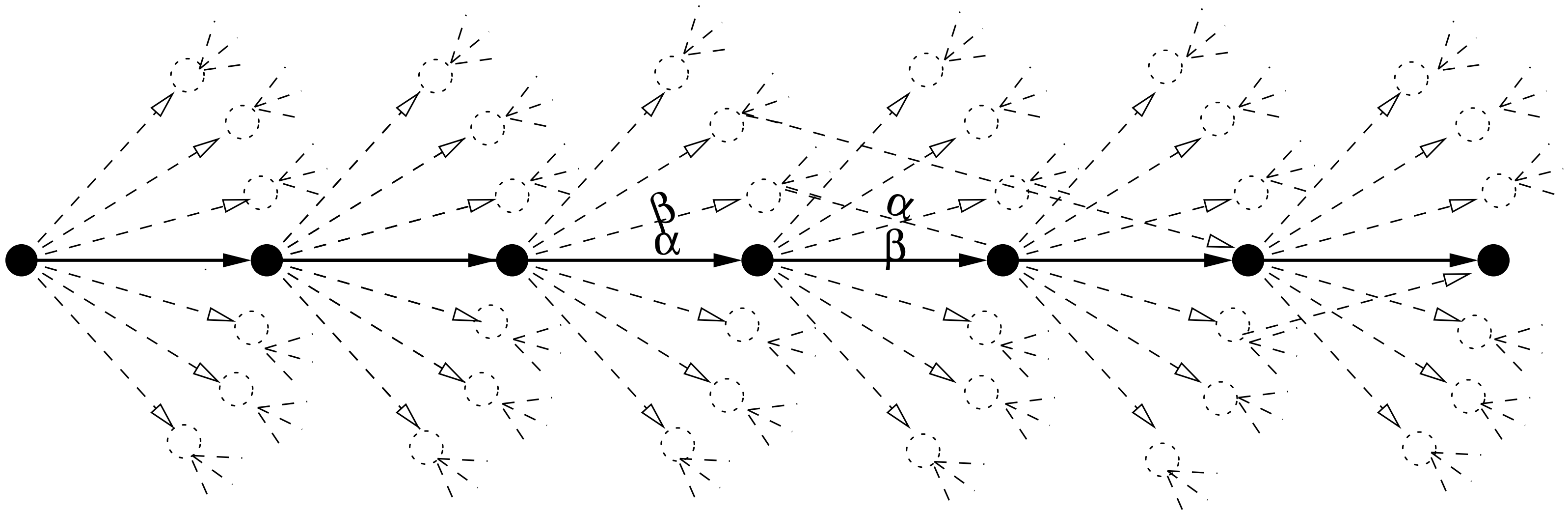

Figure 1: Kripke Model of Causal Accessibility Relation

Situation/Event Calculi and the Frame Problem

- The Situation Calculus (McCarthy and Hayes 1969) and its descendants can be seen as versions of Dynamic Logic.
- These calculi are heir to the “Frame Problem,” which arises from the fact that humans conceptualize events in terms of very localized changes to situations.
- For example, the effects of an event of *My eating a hamburger* are confined to the hamburger and aspects of myself like hunger. The color of the walls, the day of the week, the leadership of the Conservative and Unionist party, and countless other aspects of the situation remain unchanged.

This character of the knowledge representation raises the Frame Problem in two forms: the “Representational” and “Inferential” versions.

The Representational Frame Problem

- Since change is local, it is cumbersome to explicitly represent the input effect of each event on each fact by innumerable rules such as

  (4) $color(wall,x) \Rightarrow [eat(hamburger)]color(wall,x)$

- Kowalski (1979) solved the representational problem using reified Frame Axioms Equivalent in the present notation to the following:

  (5) $p \wedge (p \neq hungry) \wedge (p \neq here(hamburger)) \Rightarrow [eat(hamburger)]p$

- This keeps rules defining the positive effects of eating hamburgers simple. (Note that p is "overloaded," standing for both the fact that p *holds* and for the term p as an individual, as is standard in logic programming.)

- But if we ever need to know what the color of the walls is after a sequence of, say, five hamburger eating events, then we have to do costly theorem-proving search. This is the *Inferential* form of the Frame Problem.

STRIPS and the Inferential Frame Problem

- The STRIPS program (Fikes and Nilsson 1971) solved both representational and inferential problems by representing change as sets of *preconditions* and localized *database updates*, as in the following definition of the operator *eat*:

PRECONDITIONS:	$hamburger(x)$
	$here(x)$
	$hungry$
DELETIONS:	$here(x)$
	$hungry$
ADDITIONS:	$thirsty$

Such representations were initially derided by logicians (because of their nonmonotonicity) . . .

- . . . but then Girard (1995) came along with Linear Logic, and update was logically respectable after all!

The Linear Dynamic Event Calculus (LDEC)

- We can represent events involving boxes in this notation.
- The preconditions of putting something on something else can be defined as follows using standard implication and an *affords* predicate:

(6) $box(x) \wedge box(y) \wedge \neg on(z,x) \wedge \neg on(w,y) \wedge (x \neq y) \Rightarrow affords(puton(x,y))$

- A situation *affords* an action (in the sense of Gibson 1966 discussed below) if it satisfies its preconditions.
- To define the update consequences of putting something *on* something else in a situtaion that affords that action we need a different, linear implication $\multimap$:

(7) $\{affords(puton(x,y))\} \wedge on(x,z) \multimap [puton(x,y)]on(x,y)$

- Linear implication, $\multimap$, treats positive ground literals or “facts” in the antecedent as consumable resources, removing them from database and replacing them by the consequent.

STRIPS updates as Linear Implication (Contd.)

- The braces in marks $\{affords(puton(x,y))\}$ mark the affordance as a nonconsumable precondition: the truth of this condition after a *puton* event is not defined by the linear implication, and is a matter for further inference, via rules like (6).
- It is related to Girards ! exponential ("Of course!").
- Thus we use the $\{affords(\ldots)\}$ notation to "fibre" the intuitionistic and linear components of the logic.

STRIPS Planning in LDEC

- The transitivity axiom of the affordance relation is defined as follows:

(8) $affords(\alpha) \wedge [\alpha] affords(\beta) \Rightarrow affords(\alpha;\beta)$

- Consider the following initial situation:

(9) $block(a) \wedge block(b) \wedge block(c) \wedge on(a, table) \wedge on(b, table) \wedge on(c, table)$

- The following conjunctive goal (10), given a search control, can be made to deliver a constructive proof that (11) is one such plan:

(10) $goal(affords(\alpha) \wedge [\alpha](on(a,b) \wedge on(b,c)))$

(11) $\alpha = puton(b,c); puton(a,b)$

- The result of executing this plan in situation (9) is that the following conjunction of facts is directly represented by the database:

(12) $block(a) \wedge block(b) \wedge block(c) \wedge on(a,b) \wedge on(b,c) \wedge on(c, table)$

LDEC Avoids a Ramification Problem

- If durative events like the agent *moving* are represented as instantaneous transitions to and from a progressive state represented as a fluent $in\_progress(move(me, there))$, LDEC is well behaved with respect to standard examples of the ramification problem such as the one that arises from moving through a paint-spray.

- In event calculi in which intervals are primitive, it is hard to specify frame axioms that capture the common-sense knowledge that if you move, your color is unaffected, and if someone sprays you with paint your color is affected, and that if you move through a paint-spray, your color is affected.

- Because in LDEC durative events are represented in terms of initiating and terminating instants and intervening states, such knowledge is easy to represent. Suppose the situation is $at(me, here) \wedge color(me, green)$:

LDEC Avoids a Ramification Problem

- Axioms for events of spraying someone some color:

(13) $\mathit{affords}(\mathit{start}(\mathit{spray}(y,c)))$

(14) $\{\mathit{affords}(\mathit{start}(\mathit{spray}(y,c)))\} \wedge \mathit{color}(x)$
$\quad \multimap [\mathit{start}(\mathit{spray}(y,c))]\mathit{in\_progress}(\mathit{spray}(y,c))$

(15) $\mathit{in\_progress}(\mathit{spray}(y,c)) \Rightarrow \mathit{affords}(\mathit{stop}(\mathit{spray}(y,c)))$

(16) $\{\mathit{affords}(\mathit{stop}(\mathit{spray}(y,c)))\} \wedge \mathit{in\_progress}(\mathit{spray}(y,c)))$
$\quad \multimap [\mathit{stop}(\mathit{spray}(y,c))]\mathit{color}(y,c)$

LDEC Avoids a Ramification Problem

- For a situation in which $at(me, here) \wedge color(me, green)$, we correctly prove the following without encountering inconsistency:

(17) $[start(move(me, there)); start(spray(me, pink)); stop(spray(me, pink)); stop(move(me, there))]color(me, pink)$

(18) $[start(spray(me, pink)); start(move(me, there)); stop(move(me, there)); stop(spray(me, pink))]at(me, there)$

(19) $[start(spray(me, pink)); start(move(me, there)); stop(spray(me, pink)); stop(move(me, there))]color(me, pink)$

(20) $[start(move(me, there)); start(spray(me, pink)); stop(move(me, there)); stop(spray(me, pink))]at(me, there)$

STRIPS updates as Linear Implication (Again)

- Using linear implication (or the equivalent rewriting logic devices or state update axioms of Thielscher (1999) and Martí-Oliet and Meseguer (1999)) for STRIPS-like rules eliminates frame axioms along the lines of (5).
- Instead, they are theorems of the linear logic representation.
- There is a model theory, in which “the basic idea is to interpret all the operations of linear logic by operations on facts” (cf. Girard 1995:23).
- LDEC rules are reminiscent of Hoare (1969) triples, and the nodes of Petri (1962) nets.
- LDEC rules also resemble the rules in a Production System language, such as SOAR (Newell 1990).

References

Curry, Haskell B. and Feys, Robert, 1958. *Combinatory Logic: Vol. I*. Amsterdam: North Holland.

Gibson, James, 1966. *The Senses Considered as Perceptual Systems*. Boston, MA: Houghton-Mifflin Co.

Girard, Jean-Yves, 1995. "Linear Logic: its Syntax and Semantics." In Jean-Yves Grirard, Yves Lafont, and Laurent Regnier (eds.), *Advances in Linear Logic*, Cambridge: Cambridge University Press, volume 222 of *London Mathematical Society Lecture Notes*. 1–42.

Hoare, C. Anthony, 1969. "An Axiomatic Basis for Computer Programming." *Communications of the Association for Computing Machinery* 30:576–583.

Kowalski, Robert, 1979. *Logic for Problem Solving*. Amsterdam: North Holland.

Martí-Oliet, Narciso and Meseguer, José, 1999. "Action and Change in Rewriting Logic." In Remo Pareschi and B. Fronhöfer (eds.), *Dynamic Worlds*, Dordrecht: Kluwer. 1–53.

McCarthy, John and Hayes, Patrick, 1969. “Some Philosophical Problems from the Standpoint of Artificial Intelligence.” In Bernard Meltzer and Donald Michie (eds.), *Machine Intelligence*, Edinburgh: Edinburgh University Press, volume 4. 473–502.

Newell, Alan, 1990. *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Petri, Carl Adam, 1962. “Kommunikation mit Automaten.” Schriften des IMM 2, Institut für Instrumentelle Mathematik, Bonn.

Shanahan, Murray, 1997. “Event Calculus Planning Revisited.” In *Proceedings of the Workshop on Robots, Softbots, Immobots: Theories of Action, Planning, and Control*. AAAI, 96–104. held in conjunction with the 14th National Conference on Artificial Intelligence, Providence, RI, July 1997.

Thielscher, Michael, 1999. “From Situation Calculus to Fluent Calculus: State Update Axioms as a Solution to the Inferential Frame Problem.” *Artificial Intelligence* 111:277–299.