Transition Logic

Prof. W. Bibel
TU Darmstadt
UBC Canada

Idea of Logic

- Knowledge state: $KB \wedge K_0 \models_{cog} K_i$
- $\models_{cog}$ formally describable by $\models$
- $\models$ studied for 2000+ years
- relationship of $\models_{cog}$ and $\models$ studied scientifically only for a few decades in CogSci (before through introspection and commonsense psychology)

Dimensions of Logic Space

- Structural explorations in a given logic
  - theorem proving, proof theory, semantics
- Problems of representation in a logic
  - eg. how to represent change in FOL
- Design of a logic for modelling cognitive phenomena
  - find $\models$ modelling cognitive logical knowledge relations (eg. including change)

Knowledge Processing

- . . . . for agent in an environment
- Representation for fast retrieval
- Exploiting logical relations
  - static aspect for knowledge compression
- Dynamics of knowledge state
  - dynamic aspect (need for compression)
- Both aspects fundamental for reasoning
  - eg. in planning, explanation, diagnosis etc.

What Is a Logic?

- A formal language for representation +
- a relationship $\models$: $KB \wedge K_0 \models K_i$
- Enables reduction of KB (compression)
- But what if KB changes to KB'?
- Natural reasoning suggests:
  - „ $KB \wedge K_0 \wedge t_0 \models K_i$ "
- Our solution: $KB \wedge K_0 \models_{t_0} K_i$
  - solves compression and frame problem

Natural Reasoning Under Changes

- Example: purse with credit card in pants pocket; put on pants at home; go to restaurant: credit card with me?
- Formally *static*: in(cc,purse); in(purse,pants); loc(me,athome); loc(pants,athome); in(x,y)∧loc(y,z)→loc(x,z); *transitions*: loc(me,x)∧loc(pants,x) ⇒ loc(me,x)∧loc(pants,x)∧in(pants,me); loc(me,athome) ⇒ loc(me,rest); loc(cc,rest)?

A Simple Didactic Example E_1

- $U, U \to F \vdash F \,/\, H, H \to A \vdash A \,/\, D, D \to O \vdash O$
  - U: block is uncovered
  - F: block is free
  - H: hand holds block
  - A: block is airborne
  - D: block is put down
  - O: block is on table
- $U, U \to F$ is short for $U \wedge (U \to F)$

Transitional Problems

- A *transition* is a pair (E,F) of conjunctions of atoms
- A *transitional problem* (W,T,I,G)
  - W: world knowledge (definite nonrec. cls.)
  - T: set of transitions
  - I: initial state (set of atoms)
  - G: goal state (set of atoms)

(finite sets; set, conjunction interchangeable)

Source informally

- $\sigma(F)=\{\{U\}\}$ in example
- for $U_1 \wedge U_2 \rightarrow F$, $\sigma(F)=\{\{U_1\},\{U_2\}\}$
  - U_1 musty, U_2 lack of oxygen, F stuffy
- for $U_1 \rightarrow F$, $U_2 \rightarrow F$, ie. $U_1 \vee U_2 \rightarrow F$, $\sigma(F)=\{\{U_1,U_2\}\}$
- disjunction of conjunctions of …
- nf: normal form, ie. nesting depth is 2
- $subs\{S_1,S_2\}=S_i$ if $S_i \subseteq S_j$ else identical

Source Definition

- For W, I, and
  atom A (without restricting generality):
- $\sigma(A) = \{\{A\}\}$ if $A \in I$
- $\sigma(A) = subs\ [\ nf\ \{\ \{\sigma(B_{1j_1}),\ldots,\sigma(B_{nj_n})\} \mid 1 \leq j_1 \leq m_1,\ldots,1 \leq j_n \leq m_n\}\ ]$ if $B_{11} \wedge \ldots \wedge B_{1m_1} \vee \ldots \vee B_{n1} \wedge \ldots \wedge B_{nm_m} \rightarrow A$
- $\sigma(A) = \perp$ else

Illustration of Definition

- Recursive replacement of atoms by defining bodies, form dual formula, transform to DNF, remove subsumed clauses, interpret as set of sets of atoms
- Semantically, $\sigma(A)$ is set of minimal sets M: I\M does no more entail A under W
- $U_1 \wedge U_2 \rightarrow F$: $\sigma(F) = subs[nf\{\{\sigma(U_1)\},\{\sigma(U_2)\}\}] = subs[nf\{\{\{\{U_1\}\}\},\{\{\{U_2\}\}\}\}] = \{\{U_1\},\{U_2\}\}$
- $U_1 \vee U_2 \rightarrow F$: $\sigma(F) = subs[nf\{\{\sigma(U_1),\sigma(U_2)\}\}] = subs[nf\{\{\{\{U_1\}\},\{\{U_2\}\}\}\}] = \{\{U_1,U_2\}\}$

Inference Relation

- For $s=\{S_1,\ldots,S_m\}$ and $t=(\{P,D\},\{P,A\})$:

  $s` = \bigcup_{i=1}^{m} \{(S_i \setminus D`) \cup A \mid D` \in \sigma(D)\}$
- $S = S_1 \vee \ldots \vee S_m$ state set formula, whereby S_i conjunction of atoms in it
- Inductive definition of $\vdash_{T^n}$ for *sequences*
  - $\vdash_{()} = \vdash_{T^0} = \vdash$
  - step n-1 to n : $W, I \vdash_{(t_1,\ldots,t_m)} F$ iff $W, S` \vdash F$

Deductions Modulo Transitions

- $W_1, U \vdash_{(t_1,t_2)} O$ in example E_1
- E_2 like E_1 but additional $t_3 = U \Rightarrow H$:
  $W_1, U \vdash_{(t_3,t_2)} O$, ie. inference absorbable by additional transitions (STRIPS)
- Similar for *sets* of transitions: resulting plans (transitions) are partially ordered
- General case: $\vdash_{(T,\leq)}$ for partial order $\leq$ on set T

Lifting to FOL

- $C(x) \wedge O(x,y) \wedge C(z) \Rightarrow C(x) \wedge O(x,z) \wedge C(y)$
- $I = \{U(b), \exists y O(b,y), C(c)\}$
- $W = \{U(x) \rightarrow C(x), \ldots\}$
- Variables in transitions treated as parameters, recursion and negation by stratification
- Generalization to more general formulas is harder, left open for future research

Historical Background

- Leibniz conceived „possible worlds"
- Wittgenstein's „state of affairs"
- Carnap's „state descriptions" (1946)
- Kripke's semantics for modal logic (1959ff)
- but modal logic studies global truths
  - quasi top-down (instead of bottom-up)
- and originated with different intuition

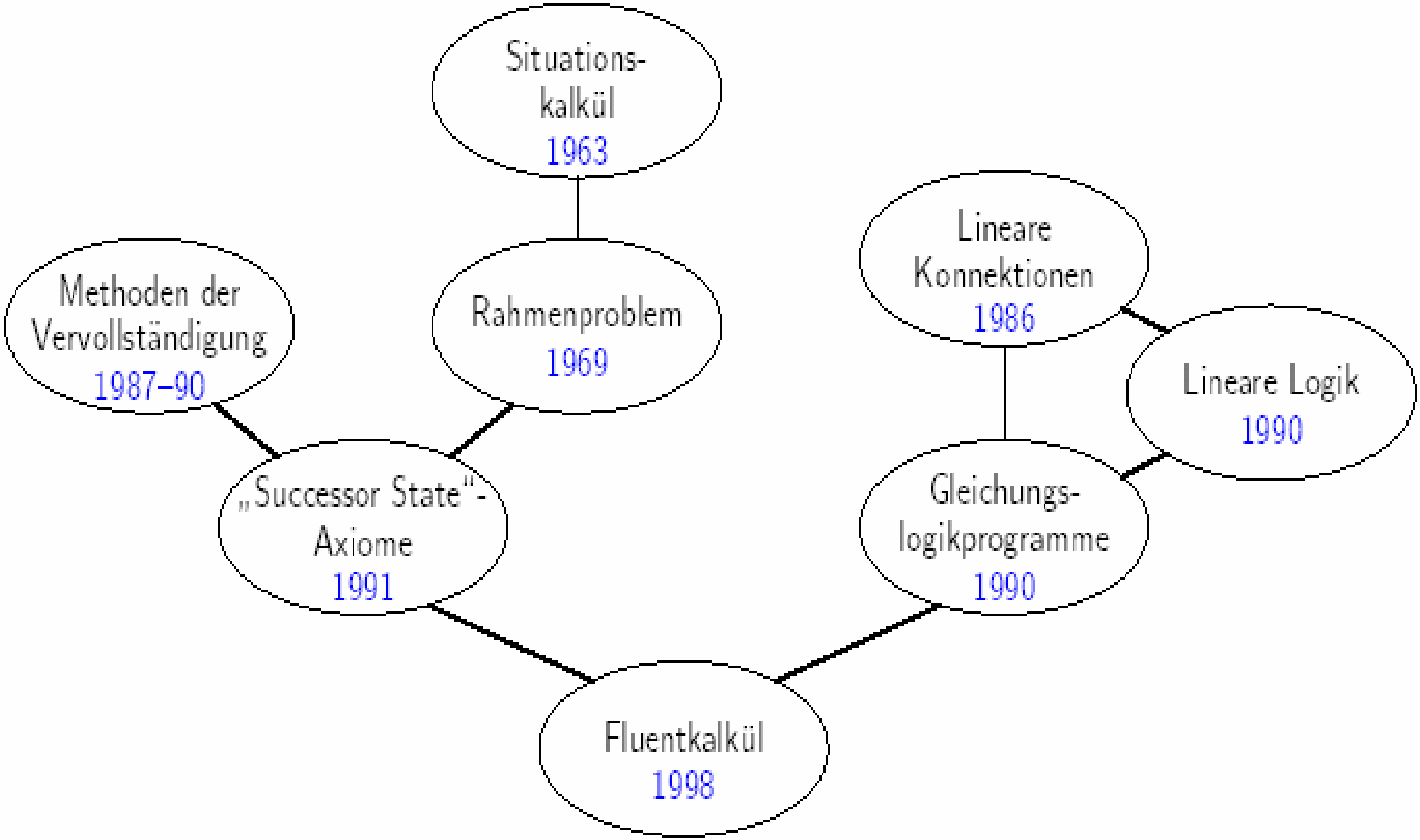
Recent History

- STRIPS 1971
  - classical deduction was never clarified and is not contained in modern systems
  - Thiébaux et al. 2003 (FF planner) introduced $\vdash$ within states with exponential gains but not beyond transitions as here
  - TL shares Lifschitz` semantics with STRIPS
- linear connection method 1986, linear logic 1987, transition logic 1998

Recent History ctd.

- Situation calculus 1963
  - extra situation parameter
  - suffers from inferential frame problem
- Fluent calculus with language FLUX
  - like TL and its predecessors solves inferential frame problem
  - transitions on term level; $\vdash$ not considered
- Modal approaches, active logics etc.

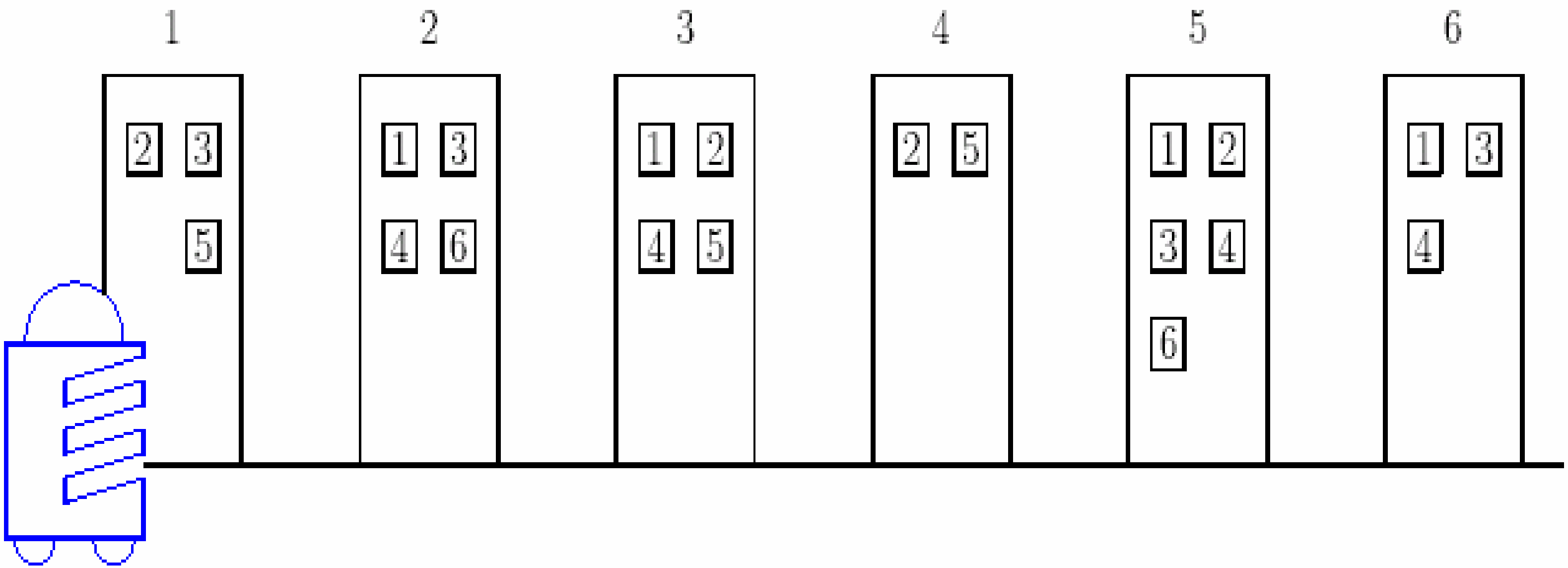
A Partial History Tree



Fluent Calculus (FC)

- state represented as term
- axioms and specification of state and actions imply constraints on solution/action and new state
- deductive plan generation
- language FLUX (Fluent Executor)
- realised in Prolog with constraints
- includes many features from KR

A Simple Office Robot



Aufgabe des Roboters sei es, mit seinen 3 Postfächern die Hauspost einzusammeln und auszuteilen.

Elementaraktionen des Roboters sind:

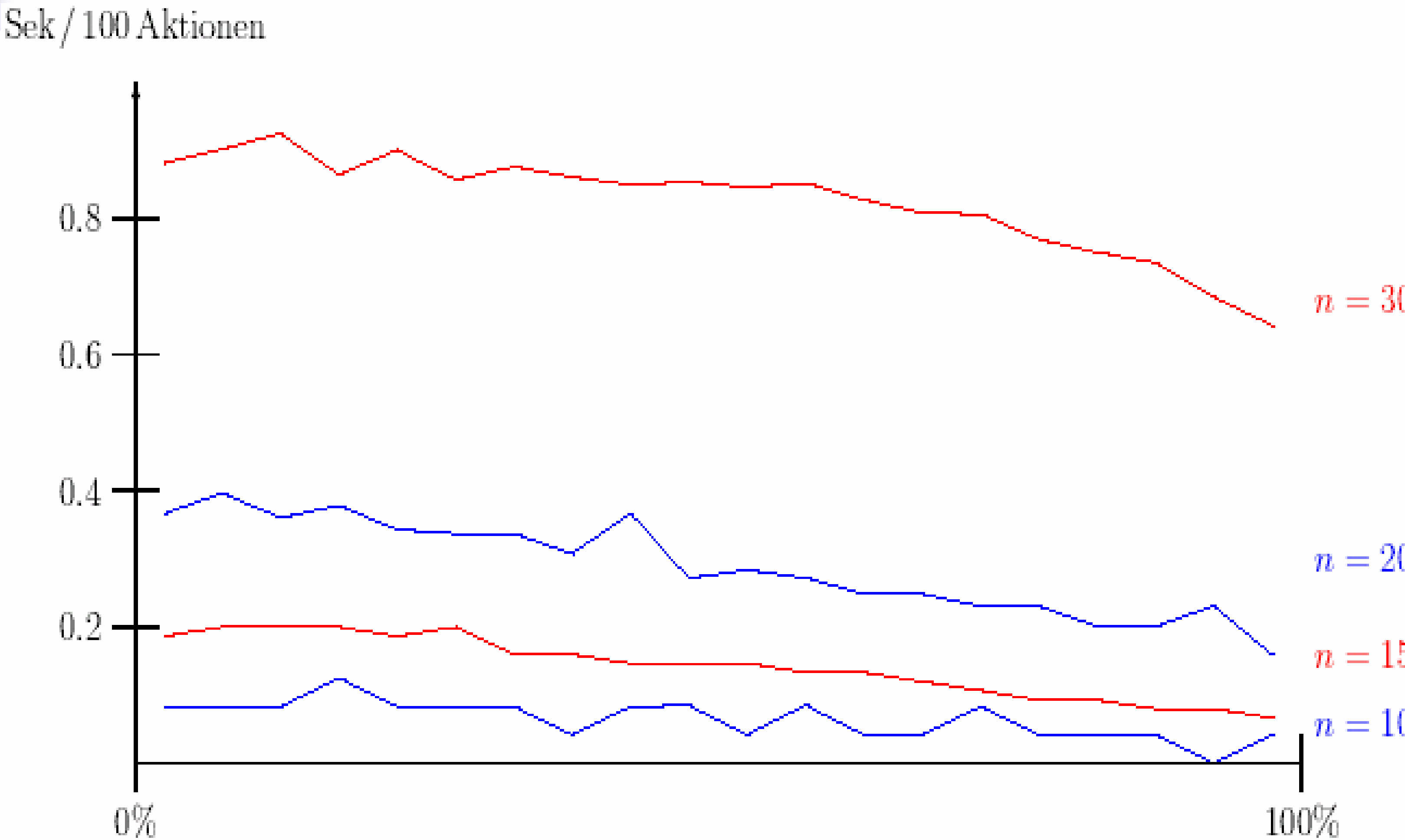
	fülle(x, y)	$x \in \{1, 2, 3\}, \; y \in \{1, \ldots, 6\}$
	leere(x)	$x \in \{1, 2, 3\}$
	gehe(x)	$x \in \{$vor, zurück$\}$

Simple FLUX Program

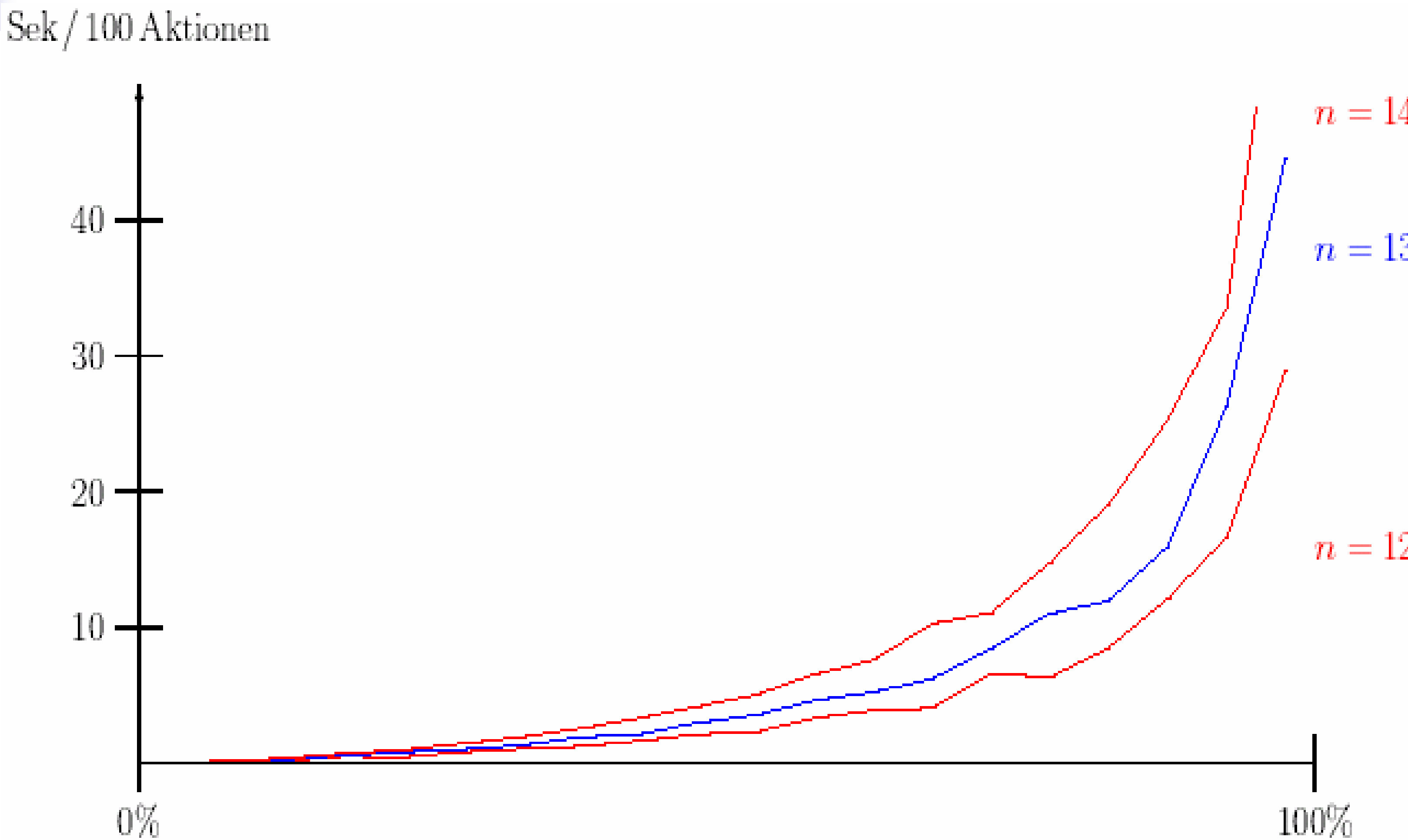
```
main :- init(Z), main_loop(Z).

main_loop(Z) :-
   poss(leere(X),Z)     -> execute(leere(X),Z,Z1),     main_loop(Z1) ;
   poss(fuelle(X,Y),Z) -> execute(fuelle(X,Y),Z,Z1), main_loop(Z1) ;
   weiter(X,Z)          -> execute(gehe(X),Z,Z1),      main_loop(Z1) ;
   true.
```

```
weiter(X,Z) :-
  ( holds(leer(F),Z), holds(sendung(R1,R2),Z) ; holds(traegt(F,R1),Z) ),
  holds(position(R),Z), ( R < R1 -> X = vor ; X = zurueck ).
```

FLUX Runtime Behavior (n,3)



Golog Runtime Behavior (n,3)



Conclusions

- Deductive approach extremely attractive
- Only competitive with a solution for both the representational and inferential frame problem (FC) and with an integration of classical $\vdash$ for compression (TL, partially FF)
- Efficient integration of TL into a theorem prover yet to be done

Midterm Perspective

- Hundreds of millions of facts in brain
- Goal: KBs with 100 mio. facts
- CYC so far 1.5 mio. facts
- Open Mind Common Sense database:
  - everyone can enter knowledge facts
  - input in natural language possible
  - within 2 years 0.5 mio. facts
- Deductive planning system on top

Trend in Full Swing . . .

- CYC (100k concepts, 10k predicates)
- Wordnet 1.6: Englisch, word/meaning
- Enterprise Ontology (modelling)
- Gene Ontology (biological concepts)
- Process Ontology (engineering)
- Cancer Ontology (100k concepts/terms)
- IEEE Standard Upper Ontology

A Practical Example

- Problem specification
  - given a standard laptop under worm attack
  - given an address for a patch program
  - goal: protected machine
  - find plan to achieve goal
- Requires several (static) reasoning steps in order to determine transitions