Modelling search and evaluating strategies in theorem proving

(Colloquium given at TU-Graz in May 2000)

MARIA PAOLA BONACINA
DEPT. OF COMPUTER SCIENCE
THE UNIVERSITY OF IOWA

What is theorem proving and how does it relate to software technology?

Theorem proving

H: assumptions

What follows from H?

φ: conjecture

Does φ follow from H?

$(H \overset{?}{\models} \varphi)$

H may be:

- mathematical theory
  (e.g., algebra
  geometry
  analysis)

- system specification
  (e.g., message-passing)

Refutational theorem proving

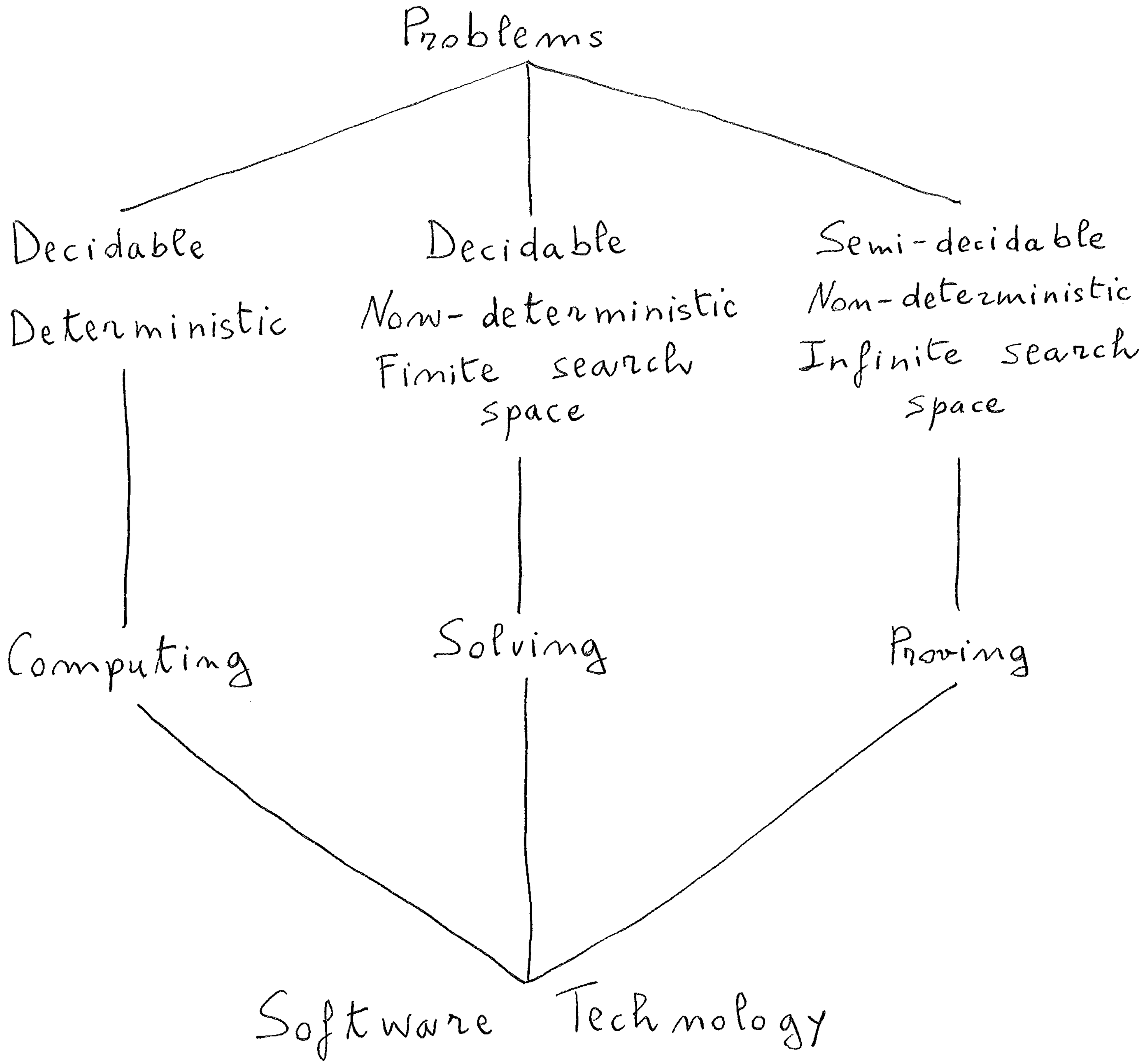
$$H \cup \{\neg\varphi\}$$

either prove φ by generating a refutation of $H \cup \{\neg\varphi\}$

$$H \cup \{\neg\varphi\} \vdash \perp$$

or disprove φ by generating a counterexample (a model of $H \cup \{\neg\varphi\}$)

T.P. and S.T.:


Problems
Decidable
Deterministic
Decidable
Non-deterministic
Finite search space
Semi-decidable
Non-deterministic
Infinite search space
Computing
Solving
Proving
Software Technology

<u>T.P. and S.T.:</u>

Proving helps computing:

Formal Methods

Verification / Synthesis

Computing helps proving:

Algorithms

Human - Computer Interfaces

T.P. and S.T.:

Solving helps proving:

Constraint solving (e.g., SAT)

Symbolic Computation
(e.g., Computer Algebra)

Proving helps solving:

Deductive proofs

Inductive proofs

Models to work in

Different axiomatizations

Problems in proving

Refutationally complete method

Exhaustive search

Done !?

NO!

Too much redundant information

Brute-force search not adequate

Remedies:

Inference systems with less redundancy

Better search techniques

My research program

Common theme: control of deduction

Research directions:

- Combination of forward and backward reasoning
  Target-oriented equational reasoning
  Lemmatization in semantic strategies

- Distributed deduction
  Clause-Diffusion (Aquarius, Peers)
  Modified Clause-Diffusion (Peers-mcd)
  Distributed search: criteria to partition search space

- Analysis of strategies
  (both inference system and search plan)
  Search space reduction by contraction
  Distributed-search contraction-based strategies

Why modelling search and evaluating strategies in T.P. ?

Theorem proving is difficult

Semi-decidable problem

Infinite search space

Finite resources

but:

it works!

In mathematics, e.g.:

- Moufang identities in rings
  S. Anantharaman, J. Hsiang
  SBR2 1990

- Axiomatizations of Lukasiewicz many-valued logic
  S. Anantharaman, M.P. Bonacina
  SBR 3 1989-91

- Single axioms for groups
  W. McCune OTTER 1993

- Robbins algebras are Boolean
  W. McCune EQP 1996

And not only in math:

- Deductive composition of SW from subroutine libraries
  (M.E. Stickel et al.
  SNARK 1994)

- Verification of cryptographic protocols
  (J. Schumann SETHEO 1997)
  (C. Weidenbach SPASS 1999)

- Modelling + verification of message-passing systems
  (W. McCune IVY 1999)
  O. Shumsky

These systems implement many different strategies:

inference rules	I
search plan	Σ
T.P. strategy	$\mathcal{C} = \langle I ; \Sigma \rangle$

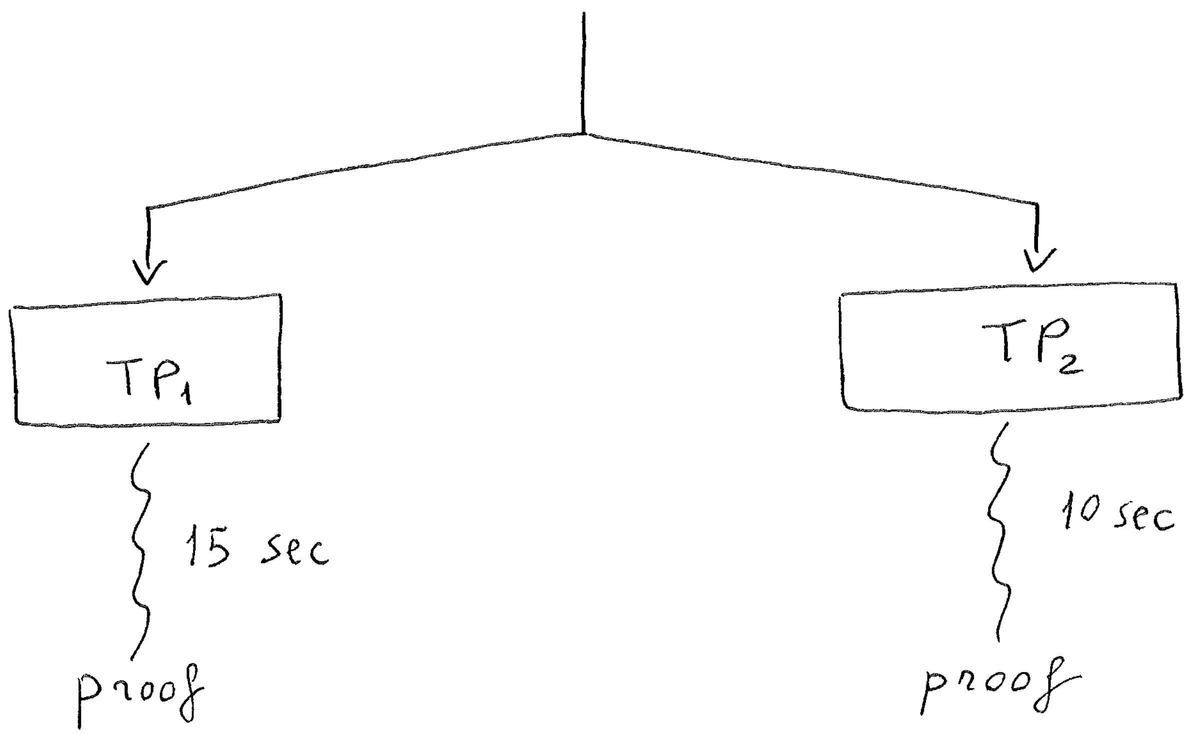
Why do they work?

How to evaluate them?

Traditional approach:

implement $\mathcal{C}_1$ in TP_1

implement $\mathcal{C}_2$ in TP_2



$\mathcal{C}_2$ is better!

Clearly not satisfactory.

Conventional complexity analysis does not apply

- infinite search space
- undecidable problem domain

Can't do worst-case nor average-case analysis.

- complexity not proportional to input (e.g., input length)
- complexity not proportional to output (e.g., proof length)

Need a way to analyze the process of finding a proof.

A key feature of today T.P. strategies:

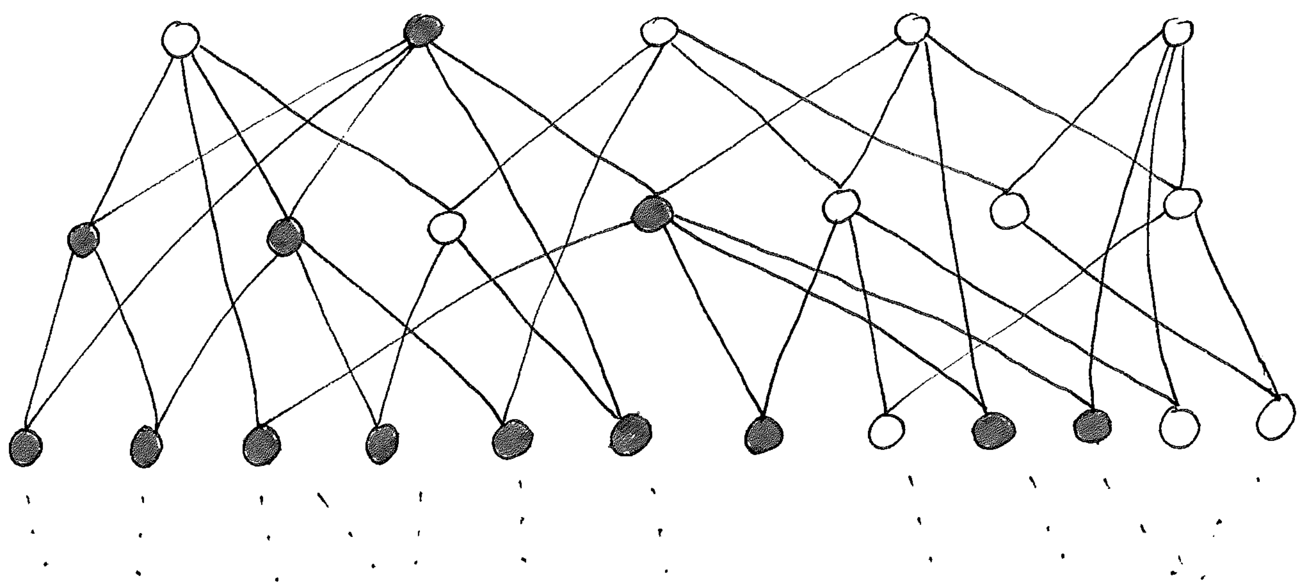
contraction

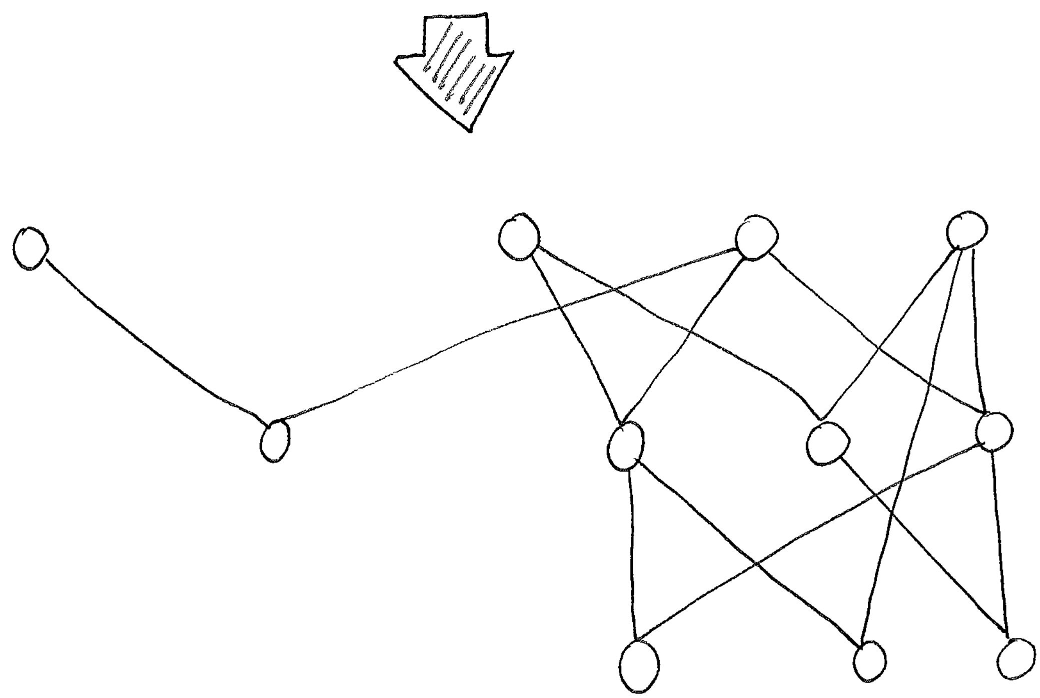
Assume forward reasoning:
generate (e.g., resolution)
and keep clauses.

Contraction:
deletion / replacement rules,

e.g., subsumption
simplification
(term rewriting)

Contraction reduces search space





But how do we compare two infinite graphs?

T.P. strategy: $\mathcal{C} = \langle I, \Sigma \rangle$

I: set of inference rules

Expansion (e.g., resolution)

$$\begin{array}{ccc} \neg M(a,b,r) & & \neg M(x,y,z) \vee M(y,x,z) \\ & \searrow \quad \swarrow & \\ & \neg M(b,a,r) & \end{array}$$

Contraction (e.g., simplification)

$$\begin{array}{c} P(ffffo) \\ \vee \Big\downarrow * \\ P(fo) \end{array} \qquad ffx \xrightarrow{>} fx$$

$>$: well-founded ordering

Search plan and derivation

Σ: search plan

reduce the non-determinism of I

- state: multiset of clauses
- $\Sigma = < \zeta, \xi >$

ζ : States$^* \longrightarrow I$ (decides next rule)

ξ : States$^* \longrightarrow \mathcal{L}^*$ (decides next premises)

Derivation: $S_0 \vdash S_1 \vdash \ldots S_i \vdash \ldots$

Eager-contraction search plan

Contraction-based strategy

Second part:

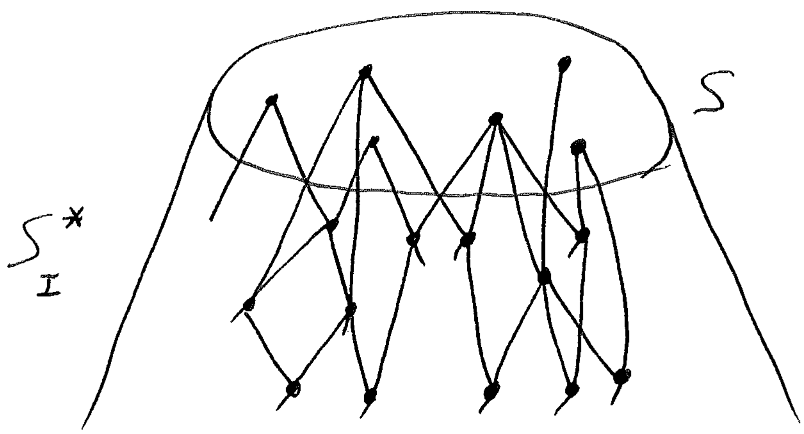
modelling search

Representation of search space

I: inference system

S: input clauses

S_I^*: closure of S w.r.t. I



Search graph: $G(S_I^*) = (V, E, \ell, h)$

Vertices V : clauses

$\ell: V \longrightarrow \mathcal{L}/\doteq$

Hyperarcs E: inferences

$h: E \longrightarrow I$

Examples:

$$M(a,a,b) \qquad \neg M(x,y,z) \vee M(z,y,x\cdot y)$$

$$M(b,a,a\cdot a) \qquad x\cdot x \longrightarrow x$$

$$M(b,a,a) \qquad M(x,y,y)$$

$$\top$$

$$\neg A(x) \vee \neg B(y) \vee R(x,y)$$

$$A(t) \qquad B(s)$$

$$R(t,s)$$

How to represent the evolution of the search space?

- Markings:

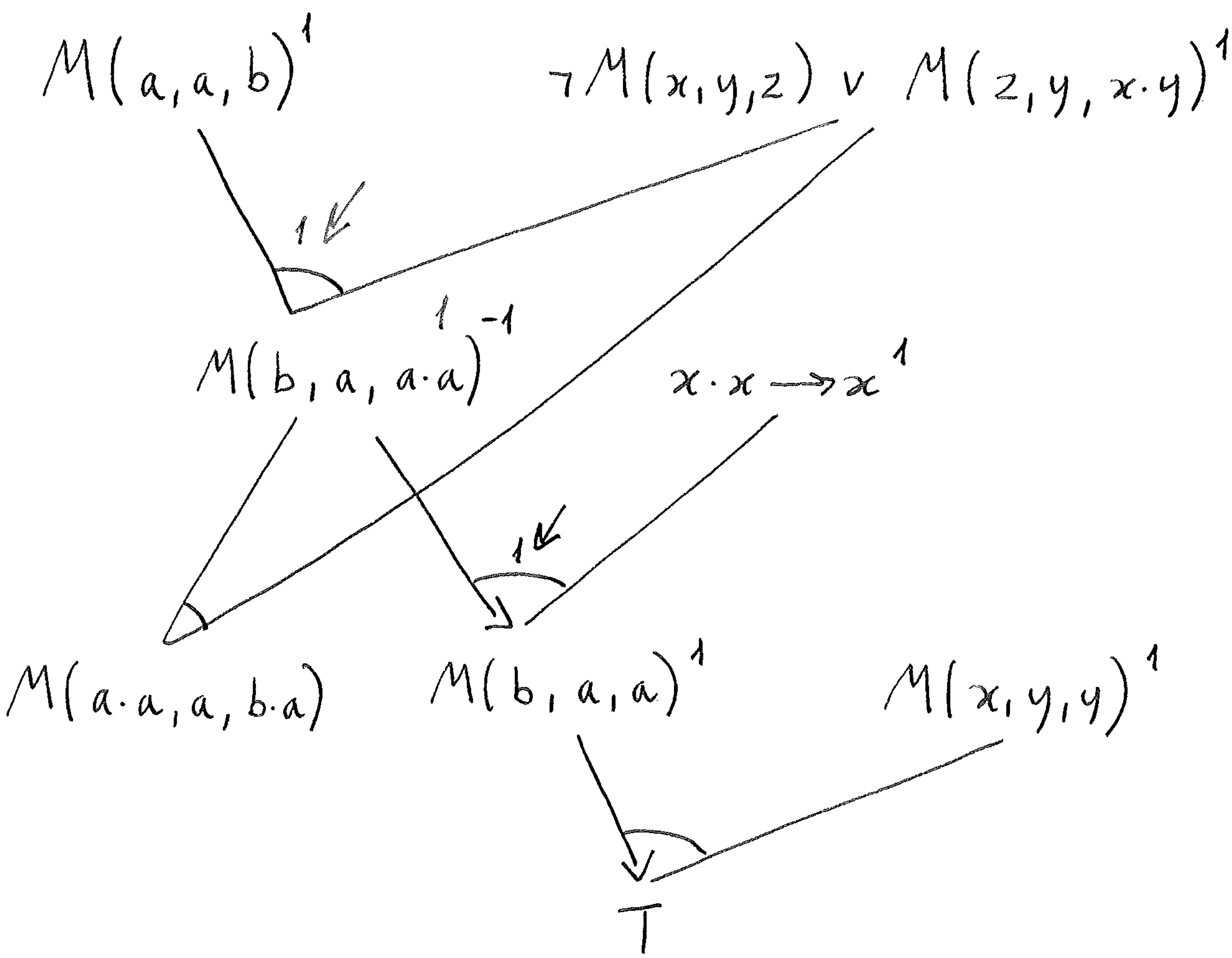
  marked graph $G = (V, E, \ell, h, s, c)$

- s: # 'copies' (variants) of a clause

$$s(\varphi) = \begin{cases} m & \text{if } m \text{ variants } (m>0) \text{ of } \varphi \text{ are present} \\ -1 & \text{if all variants of } \varphi \text{ are deleted} \\ 0 & \text{otherwise} \end{cases}$$

- $c(e)$ = # of times arc e has been executed

Example:



Evolution of search space

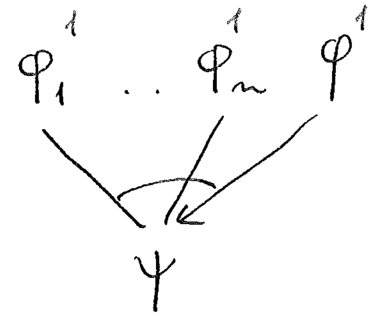
$$S_0 \vdash S_1 \vdash \dots \quad S_i \vdash S_{i+1} \vdash \dots$$
$$G_0 \quad G_1 \qquad\quad G_i \quad G_{i+1}$$

At stage 0:

$$s_0(\varphi) = \begin{cases} 1 & \text{if } \varphi \in S_0 \\ 0 & \text{otherwise} \end{cases}$$

At stage i:

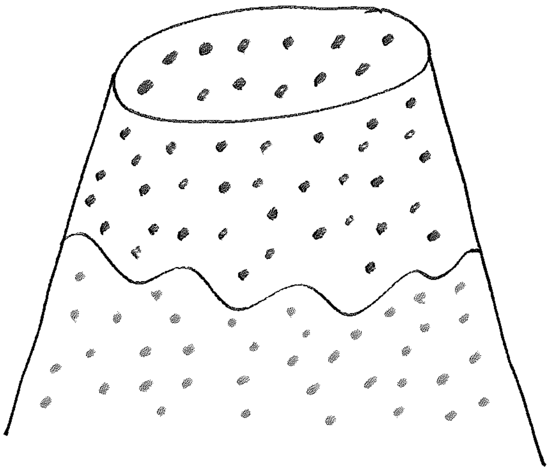
select hyperarc e:



$$S_{i+1}(x) = \begin{cases} S_i(x)+1 & \text{if } x=\psi \wedge s_i(x) \geq 0 \\ 1 & \text{if } x=\psi \wedge s_i(x) < 0 \\ S_i(x)-1 & \text{if } x=\varphi \wedge s_i(x) > 1 \\ -1 & \text{if } x=\varphi \wedge s_i(x) = 1 \\ S_i(x) & \text{otherwise} \end{cases}$$

$$C_{i+1}(e) = c_i(e) + 1$$

Marked search graph



- Active: $s(\varphi) > 0$
- Generated: $s(\varphi) \neq 0$

Advantages:

- Graph does not change marking does
- Easy to represent contraction
- Also extended to parallel search (one marking per process)

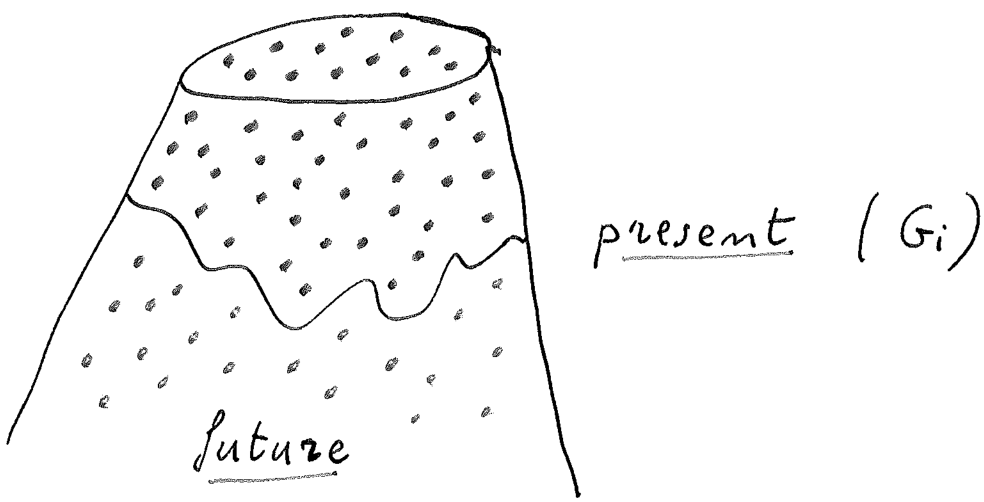
Third part:

evaluating strategies

Analysis of strategy behaviour

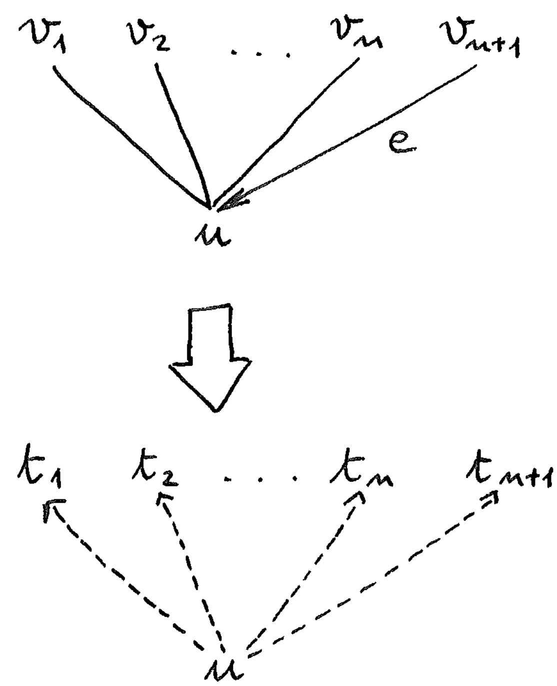
How to capture the effect of the steps performed up to stage i on the search space including the part that remains to be explored after stage i ?

G:



The future is infinite but we have something finite if we look back.

Ancestor-graph



An *ancestor-graph* of u is

$$t = (u; e; (t_1 \dots t_{n+1}))$$

where t_i is an ancestor-graph of v_i.

$at_G(u)$ or $at_G(\varphi)$: set of ancestor-graphs of u
/* $\varphi = l(u)$ */

Remark: an ancestor-graph of u represents a *generation-path* that generates φ from S_0.

Relevance

The nodes relevant to the generation of φ

Given $u \in V$

$$t = (u; e; (t_1 \dots t_{n+1}))$$

$$e = (v_1 \dots v_n; v_{n+1}; u)$$

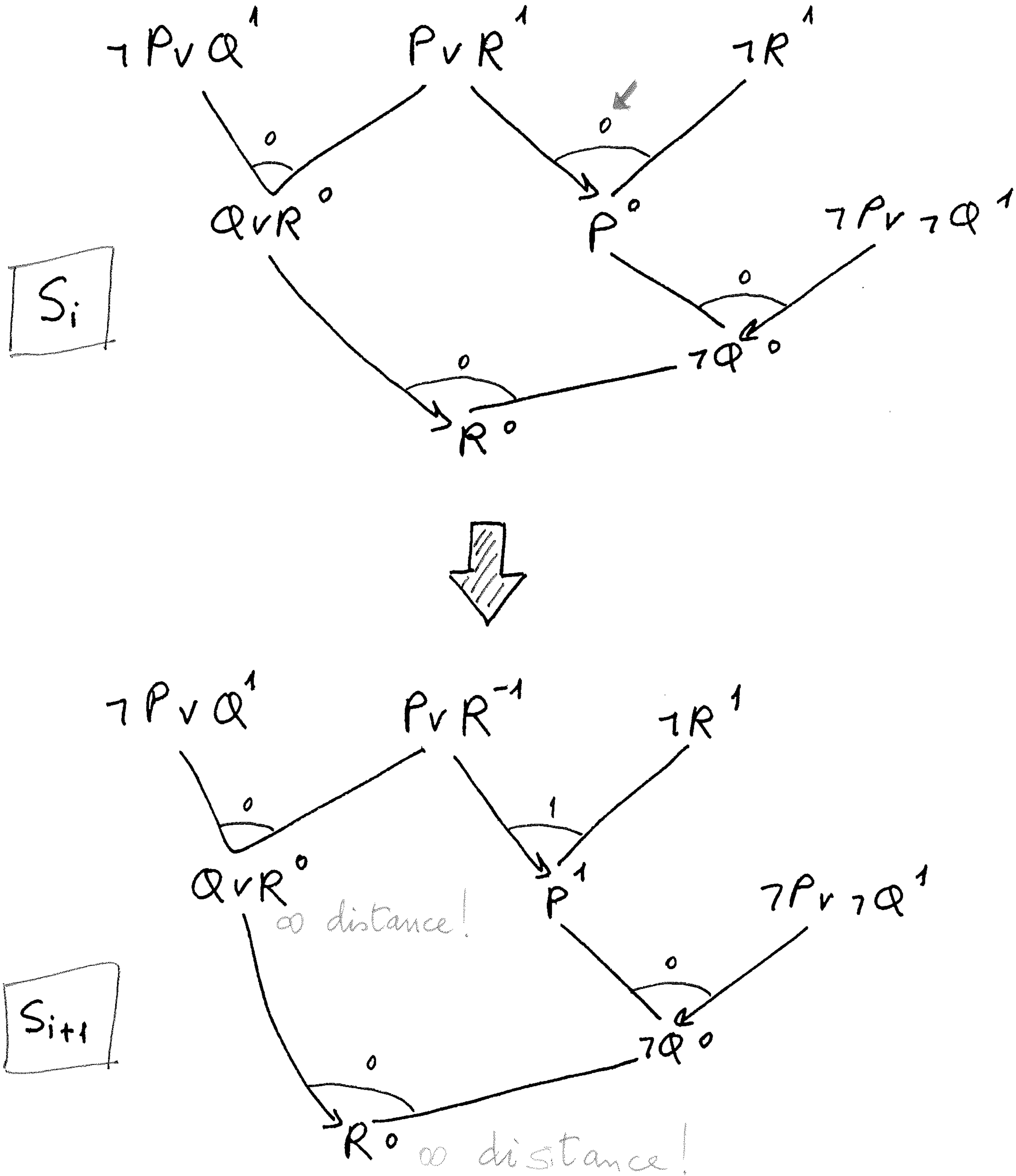
$w \in t$ is <u>relevant</u> to v in t

if

- $w \in \{v_1 \dots v_{n+1}\}$ and $c(e) = 0$ or
- w is relevant to v_i in t_i for some i.

$Rev_G(t)$: set of relevant nodes in t.

Example:



¬P∨Q 1
P∨R 1
¬R 1
0
0
Q∨R 0
P 0
¬P∨¬Q 1
0
¬Q 0
0
R 0
Si
¬P∨Q 1
P∨R −1
¬R 1
0
1
Q∨R 0
P 1
¬P∨¬Q 1
∞ distance!
0
¬Q 0
0
R 0
∞ distance!
Si+1

A motion of distance

Given G, φ, $t \in at_G(\varphi)$

- Past distance of φ in t:

$$pdist_G(t) = |\{w \mid w \in t,\ s(w) \neq 0\}|$$

- Future distance of φ in t:

$$fdist_G(t) = \begin{cases} \infty & \text{if } s(\varphi) < 0 \text{ or } \\ & \exists w \in Rev_G(t)\ s(w) < 0 \\ |\{w \mid w \in t,\ s(w) = 0\}| & o/w \end{cases}$$

- Global distance of φ in t:

$$gdist_G(t) = pdist_G(t) + fdist_G(t)$$

- f-distance of φ: $\quad fdist_G(\varphi) = \min\limits_{t \in at_G(\varphi)} fdist_G(t)$

- g-distance of φ: $\quad gdist_G(\varphi) = \min\limits_{t \in at_G(\varphi)} gdist_G(t)$

Remarks:

(1) Dynamic distance:

if ∞ then unreachable!

(∞ $\Rightarrow$ redundant)

(2) $\mathrm{fdist}_\sigma(t)$ measures the portion of t that needs to be traversed in order to reach φ

(3) Alternative definitions:

use multisets instead of cardinalities.

Bounded search spaces

Slice infinite graph G into sequence of finite layers:

at stage i $(\forall i)$ of derivation, define the bounded search space reachable within distance j $(j>0)$ (from the beginning):

$$\boxed{\mathrm{space}(G,j) = \sum_{\substack{v \in V \\ v \notin T}} mul_G(v,j) \cdot \ell(v)}$$

where

$$mul_G(v,j) = |\{ t : t \in at_G(v), \; 0 < gdist_G(t) \leq j \}|$$

space (G_i, j) is dynamic

Expansion inferences visit the search space:

if $S_i \vdash S_i \cup \{\psi\}$ then

$$space\,(G_i, j) = space\,(G_{i+1}, j) \quad \forall j$$

Contraction inferences visit and modify (prune) the search space:

if $S_i \cup \{\varphi\} \vdash S_i \cup \{\psi\}$ then

- $space\,(G_{i+1}, j) \preccurlyeq_{mul} space\,(G_i, j) \quad \forall j$

- if $s_i(\varphi) = 1$ and $D_{i+1}(\varphi) \neq \phi$

  $\exists\, k > 0 \quad \forall j \geq k$

  $space\,(G_{i+1}, j) \prec_{mul} space\,(G_i, j)$

  where:

  $D_{i+1}(\varphi) = \{\varphi' \mid \exists t \in at_\sigma(\varphi'),\ \varphi \in Rev_{G_{i+1}}(t)\}$

Analysis of search space reduction by contraction: compare strategies of different contraction power

Given $\varphi_1 = \langle I_1, \Sigma_1 \rangle$ $\varphi_2 = \langle I_2, \Sigma_2 \rangle$

input set S_0

Assume same expansion rules:

$$G^1 \neq G^2$$

$$\text{space}(G_0^1, j) \neq \text{space}(G_0^2, j)$$

because of different contraction rules.

What can we compare?

Compare the variations

Given derivation

$$S_0 \vdash_{\varphi} S_1 \vdash_{\varphi} \dots S_i \vdash_{\varphi} S_{i+1} \vdash_{\varphi} \dots$$

$$G_i = (V, E, \ell, h, s_i, c_i)$$

$$\Delta mul_{G_i}(v, j) = mul_{G_0}(v, j) - mul_{G_i}(v, j) \qquad (\geq 0)$$

- $$\Delta space(G_i, j) = \sum_{\substack{v \in V \\ v \neq T}} \Delta mul_{G_i}(v, j) \cdot \ell(v)$$

  the portion of space (G_0, j) that is pruned up to stage i of the derivation.

Use $\Delta space(G, j)$ as measure to compare contraction-based strategies.

- I_1 and I_2 are <u>equipollent</u> if $S^*_{I_1} = S^*_{I_2}$ $\forall S$

- Consider $\mathcal{C}_1 = \langle I_e \cup I_{R_1} ; \Sigma \rangle$

  $\mathcal{C}_2 = \langle I_e \cup I_{R_2} ; \Sigma \rangle$

where (i) I_1 and I_2 are <u>equipollent</u>

(ii) Σ is <u>eager-contraction</u> + <u>fair</u>

(iii) $\mathcal{C}_2$ <u>has more contraction power than $\mathcal{C}_1$</u> $(R_1(S) \subseteq R_2(S) \ \forall S)$

Use $$S_0^1 \vdash_{\mathcal{C}_1} S_1^1 \vdash_{\mathcal{C}_1} S_2^1 \vdash_{\mathcal{C}_1} \dots S_i^1 \vdash_{\mathcal{C}_1} S_{i+1}^1 \dots$$

G_i^1

$$S_0^2 \vdash_{\mathcal{C}_2} S_1^2 \vdash_{\mathcal{C}_2} S_2^2 \vdash_{\mathcal{C}_2} \dots S_i^2 \vdash_{\mathcal{C}_2} S_{i+1}^2 \dots$$

G_i^2

<u>Note:</u> the (unmarked) search graphs $G^1 \neq G^2$

Lemmas:

- $\forall i \geq 0 \quad \forall \varphi \in S_i^1 \quad \exists k \geq 0$ s.t. $\varphi \in S_k^2$ or $\varphi \in R_2(S_k^2)$
  and vice versa

- $\forall i \geq 0 \quad \forall \varphi \in R_1(S_i^1) \quad \exists k \geq 0$ s.t. $\varphi \in R_2(S_k^2)$
  (but not vice versa)

- $\forall i \geq 0 \quad \forall \varphi \in G^1$ if $S_i^1(\varphi) = -1$ then
  $\exists k \geq 0$
  - $S_k^2(\varphi) = -1$ or
  - $S_k^2(\varphi) = 0$ and $fdist_{G_k^2}(\varphi) = \infty$

- $\forall i \geq 0 \quad \forall \varphi \in G^1 \quad \forall t \in at_{G^1}(\varphi)$,
  if $fdist_{G_i^1}(t) = \infty$ then $\exists k \geq 0 \quad fdist_{G_k^2}(t) = \infty$

- More contraction eventually prunes more space

Thm : $\forall i \geq 0 \ \exists k \geq 0 \ \forall j > 0$

$$\Delta space(G^2_{k,j}) \geq_{mul} \Delta space(G^1_{i,j}).$$

- More contraction eventually causes fewer things to be generated

$at^e_G(v) =$ ancestor-graphs of v made of only expansion steps

$$emul_G(v,j) = |\{t : t \in at^e_G(v),\ 0 \leq gdist_G(t) \leq j\}|$$

$$espace(G,j) = \sum_{\substack{v \in V \\ v \notin T}} emul_G(v,j) \cdot \ell(v)$$

Thm : $\forall i \geq 0 \ \exists k \geq 0 \ \forall j > 0 \quad espace(G^2_{k,j}) \leq_{mul} espace(G^1_{i,j})$

- If all rules in $I_2 - I_1$ are deletion rules

<u>Thm</u>: $\forall i \geq 0 \quad \exists k \geq 0 \quad \forall j > 0$

$$\mathrm{space}(G^2_{k}, j) \lesssim_{mul} \mathrm{space}(G^1_{i}, j).$$

Summary

Stronger contraction $(R_1(S) \subseteq R_2(S))$ induces more reduction of the bounded search spaces, thus higher reduction of search complexity.

All theorems so far in the form

$\forall i \geq 0 \quad \exists k \geq 0 \quad \forall j > 0 \quad \dots$

bound on search space

stage in derivation by $\mathcal{C}_2$

stage in derivation by $\mathcal{C}_1$

Further results in the form

$\forall j > 0 \quad \exists m \geq 0 \quad \forall i \geq m$

all subsequent stages

stage where $\mathcal{C}_2$ surpasses $\mathcal{C}_1$

bound on search space

- $\forall j > 0 \ \exists m \geq 0 \quad \forall i \geq m$

  $\mathrm{espace}(G^2_{i}, j) \leq_{mue} \mathrm{espace}(G^1_{i}, j)$.

- $\forall j > 0 \quad \exists m \geq 0 \quad \forall i \geq m$

  $\mathrm{space}(G^2_{i}, j) \leq_{mue} \mathrm{space}(G^1_{i}, j)$ if all rules in $I_2 - I_1$ are deletion rules.

Discussion

- Lack of ways to analyze / compare strategies ("strategy analysis") has hampered T.P. (A.I.).

- Main difficulty: infinite search space.

- Representation of search space
  - all possible inferences form a static infinite graph
  - dynamics of the search described by marking, essential for contraction.

Discussion

- Slice infinite search graph into sequence of (non-static) finite search graphs.

- Notion of complexity based on WFO (e.g., multiset orderings) not only linear ordering $(\mathbb{N}, >)$.

- Comparison of contraction-based strategies:
give a formal justification of why contraction is effective.

Future work

- The beginning of the journey ...
  stimulate interest
  more concepts
  more notions

- Apply to other strategies
  (e.g., subgoal-reduction)

- Comparison of search plans

- Already extended to parallel deduction
  (distributed-search contraction-based
  strategies)

- Asymptotic analysis ?

Related work

- Search space representation
  [Kowalski 1969]

- Proof complexity
  NP $\neq$ co-NP iff no p-bounded proof system
  $f(x) = y$ for propositional tautologies
  [Cook-Reckhow 1979, Urquhart 1995]

- Lower bounds based on Herbrand Theorem
  [Statman 1979, Orevkov 1982, Goubault 1994]

- Search complexity
  [Plaisted 1994, Plaisted-Zhu 1997]

[Leitsch 1997]