On the representation and analysis of distributed search in theorem proving

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

Research program

Area: Automated Reasoning

Emphasis: Control of Deduction

Directions:

* Combination of forward and backward reasoning, e.g.,
  Target-oriented equational reasoning
  Lemmatization in semantic strategies

* Distributed automated deduction, e.g.,
  Clause-Diffusion methodology
  Modified Clause-Diffusion
  AGO-criteria
  Combination of distributed search and multi-search
  Systems built: Aquarius, Peers, Peers-mcd

* Strategy analysis, e.g.,
  Search space reduction by contraction
  Distributed search for contraction-based strategies

Motivation:

- Applications
- Impact on other areas in C.S.
- Basic investigation

Theorem proving and parallelism

- More power:

  faster proofs

  more proofs

- Search plan design:

  investigation of new forms

  of control of deduction

Many approaches:

$S \Rightarrow \boxed{\text{parallelization approach}} \Rightarrow P$

S: sequential method

P: parallel method

Evaluation:

- performance evaluation
- analysis ?

Outline

Contraction - based T.P. strategies

Distributed search

Representation

Analysis

Comparison of distributed strategy and sequential base

Discussion

What are contraction - based strategies and why are they important ?

Contraction-based strategies

History:

resolution + paramodulation

term rewriting + Knuth-Bendix

contraction-based strategies

Forward reasoning:

generate and keep clauses

Ordering-based:

well founded < on terms and clauses

Good for equality reasoning

Contraction-based strategies

$$\mathscr{C} = \langle I, \Sigma \rangle$$

I: inference rules

Expansion (e.g., resolution)

Contraction (e.g., simplification)

Σ: search plan

$$\Sigma = \langle \zeta, \xi, \omega \rangle$$

ζ: select rule [ζ: $States^* \longrightarrow I$]

ξ: select premises [ξ: $States^* \longrightarrow \mathcal{L}$]

ω: detect success [ω: $States \longrightarrow Boolean$]

$\left.\begin{array}{l} I \text{ refutationally complete} \\ \Sigma \text{ fair} \end{array}\right\} \mathscr{C}$ complete

$$S_0 \vdash S_1 \vdash \dots S_i \vdash \dots$$

Contraction-based strategies

Forward contraction

normalize every new clauses w.r.t.
existing ones

Backward contraction

normalize every existing clause w.r.t.
new insertions $\Rightarrow$ inter-reduction

Eager contraction

Σ does not select expansion until
contraction exhausted

$S_0 \vdash \dots S_i \vdash \dots \qquad \forall i \qquad \forall \varphi \in S_i$

if $\exists f \;\; \exists \bar{x} \in S_i$ f applied to $(\bar{x}, \varphi)$ deletes φ
then $\exists l \geq i \quad S_l \vdash S_{l+1}$ deletes φ and
$\forall j \;\; i \leq j \leq l$ no expansion unless succeeds
sooner

Some results of CBS

- Moufang identities in rings
  Anantharaman, Hsiang SBR2 1990

- Axiomatization of Lukasiewicz many-valued logic
  Anantharaman, Bonacina SBR3 1989-91

- Single axioms for groups
  W. McCune OTTER 1993

- Robbins algebras are Boolean
  W. McCune EQP 1996

- Verification of cryptographic protocols
  C. Weidenbach SPASS 1999

What is distributed search and why do we use it?

Parallelism at the term level

Parallelize the single inference
(e.g., parallel rewriting)

Motivation: speed-up frequent operations

Good for: concurrent rewriting

Not for CBS:

- New equations generated dynamically

$$\Downarrow$$

special pre-processing too onerous

- Many terms, equations, steps:
  too fine-grained

Parallelism at the clause level

Parallel inferences within one search
(e.g., parallel resolution steps,
OR-parallelism)

Motivation: speed-up given search

Good for: expansion-oriented T.P.
(e.g., hyperresolution with
no contraction)

Not for CBS:

backward contraction causes conflicts

$\Downarrow$

do it sequentially: bottleneck

Parallelism at the search level

Parallel derivations:

deductive processes search in parallel the space of the problem

Heterogeneous systems:

different inference systems

motivation: combine forward / backward reasoning

Multi-search:

different search plans

motivation: search in different order

Distributed search:

subdivide search space

motivation: divide work

All need communication

Distributed strategy

$$\mathcal{C} = \langle I, M, \Sigma \rangle$$

I : inference rules
(expansion rules , contraction rules)

M : communication operators
(send , receive ...)

Σ : search plan
sequential : select rule
select premises
distributed : also subdivision
communication

Distributed-search plan

$$\Sigma = \langle \zeta, \xi, \alpha, \omega \rangle$$

ζ : select rule / operator

ξ : select premises

α : subdivision function

ω : detects termination

Subdivision function α

Subdivide inferences among $p_0\ p_1 \dots p_{n-1}$

Search space infinite unknown $\Rightarrow$
dynamic subdivision: at each stage S_i
of derivation subdivide inferences in S_i

$\alpha(S_0 \dots S_i, n, k, f, \bar{x})$ = true / false
means p_k is allowed / forbidden
to apply f to $\bar{x}$

Two requirements:

α is total on generated clauses
(partial function in general)

α monotonic: (w.r.t. i)

$\perp\ \perp\ \perp$ false false false ...
$\perp\ \perp\ \perp$ true true true ...
$\perp\ \perp\ \perp\ \perp\ \perp\ \perp$ ...

Parallelization by subdivision

$\mathcal{P} = \langle I, \Sigma \rangle \qquad \Sigma = \langle \zeta, \xi, \omega \rangle$

$\mathcal{P}' = \langle I, M, \Sigma' \rangle \qquad \Sigma' = \langle \zeta', \xi', \alpha, \omega \rangle$

ζ' and ξ' select inferences like ζ and ξ so that the difference is made by α: forbidden steps $\Rightarrow$ different selections and by the presence of communication.

$P_0, P_1 \ldots P_{n-1}$: different derivations:

$$\left.\begin{array}{l}
S = S_0^0 \vdash S_1^0 \vdash \ldots \; S_i^0 \vdash \ldots \\
\vdots \\
S = S_0^k \vdash S_1^k \vdash \ldots \; S_i^k \vdash \ldots \\
\vdots \\
S = S_0^{n-1} \vdash S_1^{n-1} \vdash \ldots \; S_i^{n-1} \vdash \ldots
\end{array}\right\} \text{distributed derivation}$$

Fairness of distributed derivations

Refutational completeness of $\mathcal{I}$ +
Fairness of Σ = Completeness of $\mathcal{C}$

$\forall \bar{x}$ persistent non-redundant
$\forall f$ expansion rule
$\exists p_k$ such that

(1) p_k has $\bar{x}$ (fairness of communication)

(2) p_k is allowed to apply f to $\bar{x}$
at some stage (fairness of subdivision)

and (3) all local derivations are fair
(local fairness)

Theorem: $(1) + (2) + (3) \Rightarrow$

the distributed derivation is fair

How can we guarantee that a parallelization by a subdivision of a contraction-based strategy is also contraction-based?

Eager contraction in distributed derivation

$$\varphi \xrightarrow{\psi} \varphi' \qquad \text{in } \bigcup_h S_i^h$$

What if $\varphi \in S_i^h$ and $\psi \in S_i^j$?

What if the step is forbidden?

Propagation of clauses up to redundancy:

φ persistent non-redundant

$\varphi \in \bigcup_h S_i^h$ (i first stage)

then $\forall p_h \; \exists j \; \varphi \in S_j^h$ (delay : $j-i \geq 0$)

/* also sufficient for fairness of communication */

Eager contraction in distributed derivation

Distributed global contraction:
$\forall p_k \quad \forall i \quad \forall \varphi \in S_i^k$
if $\exists f \ \exists \bar{x} \in \bigcup_h S_i^h$ f applied to $\bar{x}$ deletes φ
then $\exists l \geq i$ p_k deletes φ at stage l
unless it halts sooner.

Global eager contraction:
no expansion
no communication
in between $(\forall j \ \ i \leq j \leq l)$

Lemmas:

$$\left.\begin{array}{l}\text{Local eager contraction} \\ \\ \text{Propagation of clauses up to redundancy}\end{array}\right\} \Rightarrow \text{Distributed global contraction}$$

$$\left.\begin{array}{l}\text{Local eager contraction} \\ \\ \text{Immediate propagation of clauses up to redundancy}\end{array}\right\} \Rightarrow \text{Global eager contraction}$$

Contraction-based strategies

Sequential : contraction rules
eager contraction

Distributed : contraction rules
distributed global contraction

$\mathcal{C}$: contraction-based
$\mathcal{C}'$: parallelization by subdivision of $\mathcal{C}$

Is $\mathcal{C}'$ contraction-based ?

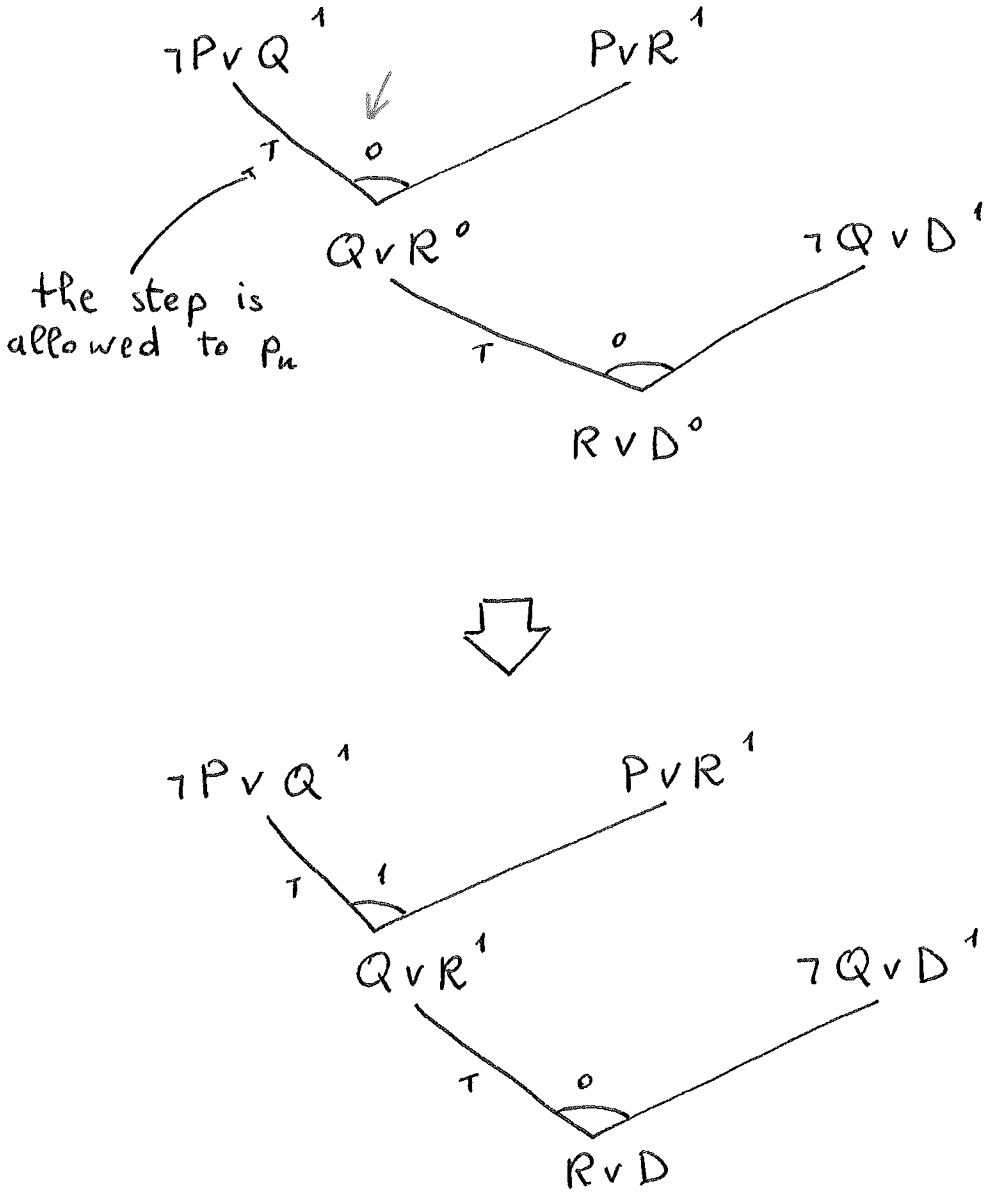
Sufficient conditions : (two sets)

1) Σ' propagates clauses up to redundancy
α does not subdivide contractions

2) Σ' propagates clauses up to redundancy
α subdivides generations, not deletions, by contraction

How can we represent distributed search in the search space of a T.P. problem?

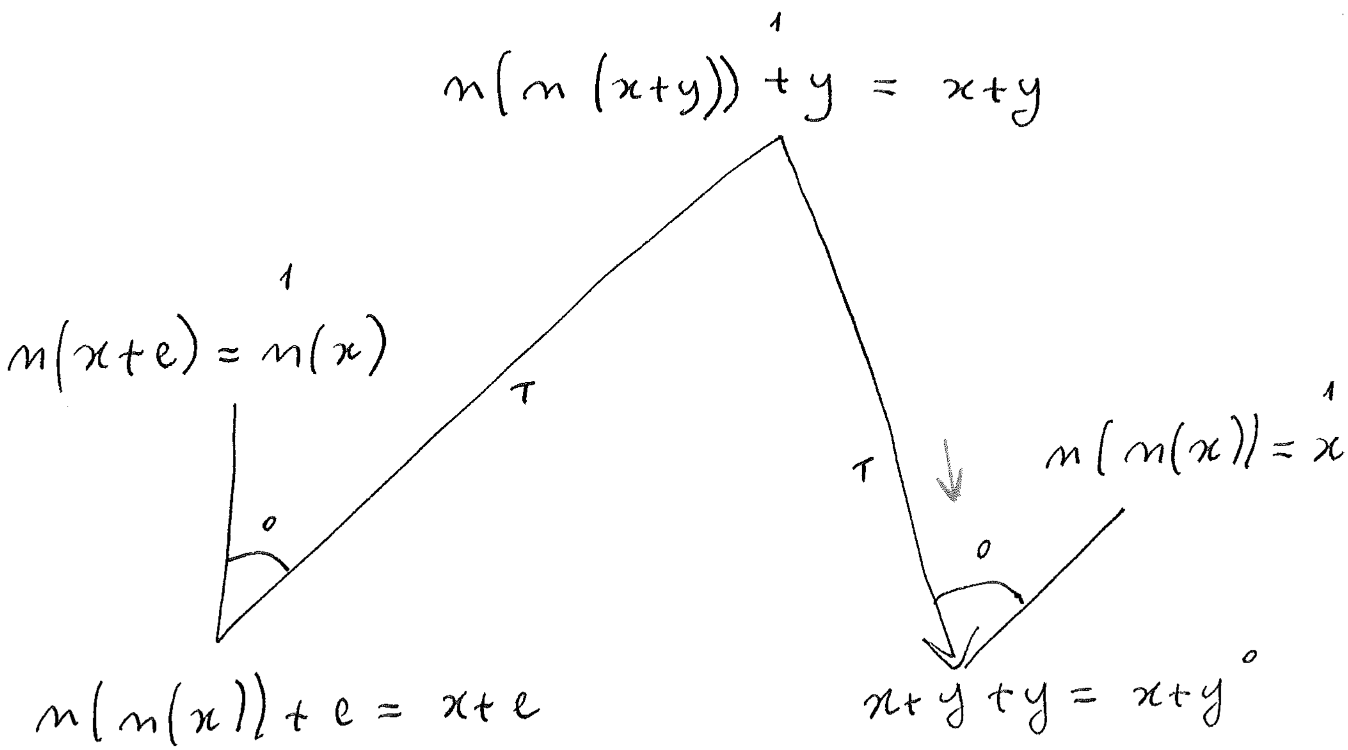
Example : expansion

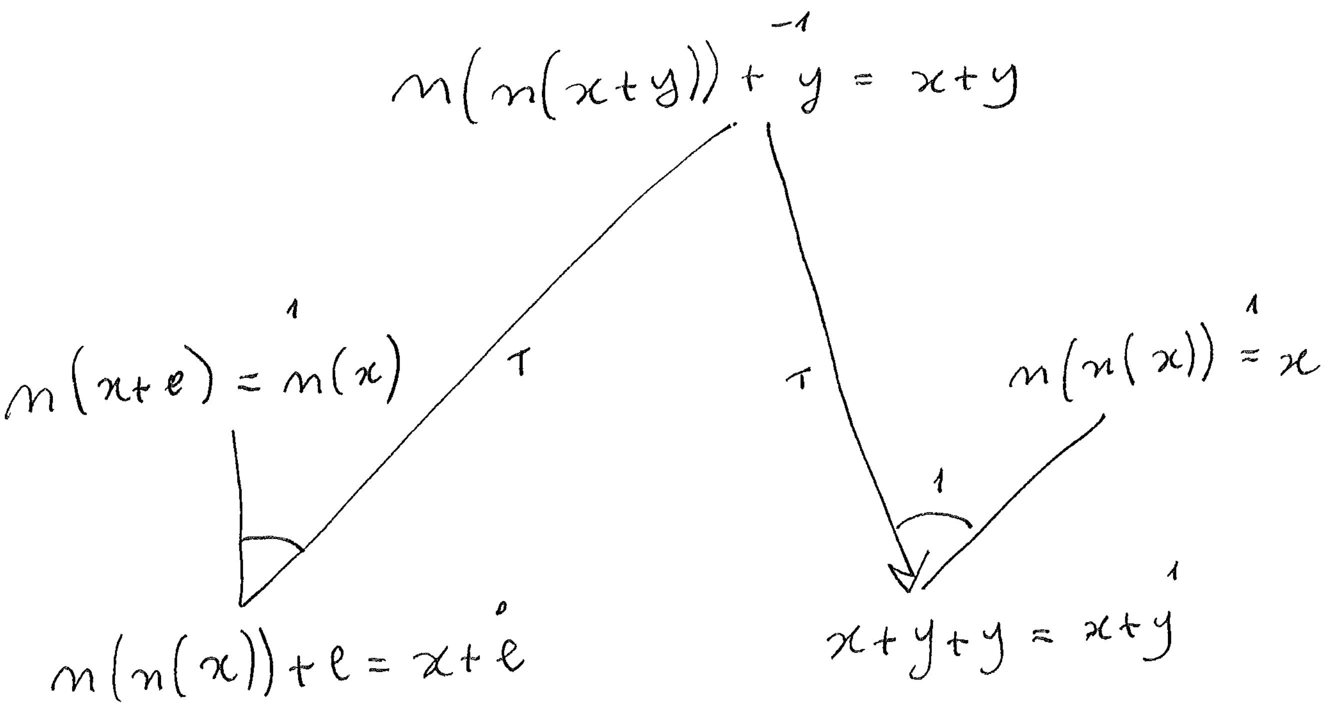


Marking relative to P_k :

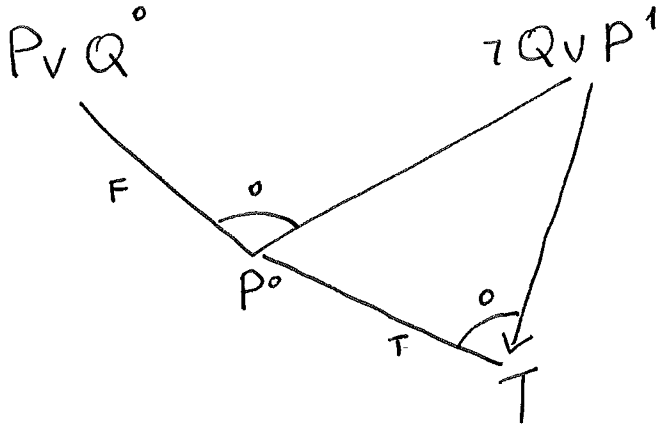
have one per process.

Example : contraction



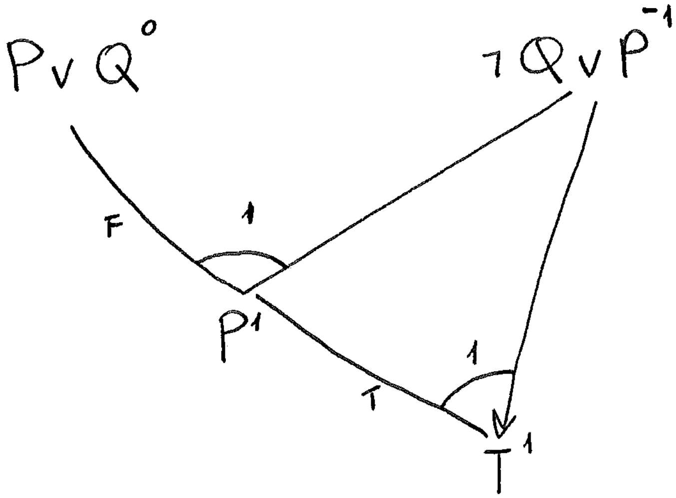


Example: communication



at P_k

receives P from P_j and applies it to subsume $\neg Q \vee P$



at P_k

Representation of search space

Closure: S_I^*

Marked search graph $G(S_I^*) = \langle V, E, l, h, \bar{s}, \bar{c} \rangle$

Vertices V: clauses $(l : V \longrightarrow \mathcal{L}/\doteq)$

Hyperarcs E: inferences $(h : E \longrightarrow I)$

Marking $\bar{s}$ of vertices: $s^k : V \longrightarrow \mathbb{Z}$

$s^k(v)$: # of variants of clause
(-1 if all deleted by p_k)

Marking $\bar{c}$ of arcs: $c^k : E \longrightarrow \mathbb{N} \times Bool$

$\pi_1(c^k(e))$ = # of times p_k executed e or received clauses generated by e

$\pi_2(c^k(e))$ = true / false
(allowed / forbidden)

Evolution of search space

$S_0^k \vdash \dots S_i^k \vdash \dots \qquad k \in [0, n-1]$

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

1) all premises present $(s^k(v) \geq 1)$

2) arc allowed $(\pi_2(c^k(e)) = true)$

$$S_0^k(v) = 0$$

$$S_{i+1}^k(v) = \begin{cases} S_i^k(v) + 1 & \text{if generated or received} \\ S_i^k(v) - 1 & \text{if deleted} \\ (-1 \text{ if last variant}) & \end{cases}$$

$$\pi_1(c_0^k(e)) = 0$$

$$\pi_1(c_{i+1}^k(e)) = \pi_1(c_i^k(e)) + 1 \quad \text{if executed or received}$$

$$\pi_2(c_{i+1}^k(e)) = \begin{cases} \alpha(S_0 .. S_{i+1}, n, k, f, \bar{x}) & \text{if } \neq \perp \\ true & \text{otherwise} \end{cases}$$

How to analyze

T. P. strategies ?

Methodological problems

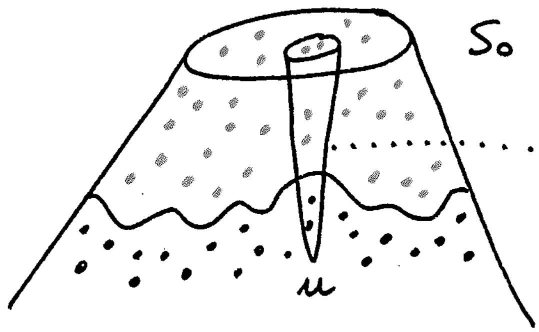
T.P. is only semi-decidable $\Rightarrow$
Search space is infinite
algorithm analysis does not apply.

Complexity proportional
neither to input (e.g., input length)
nor to output (e.g., proof length).

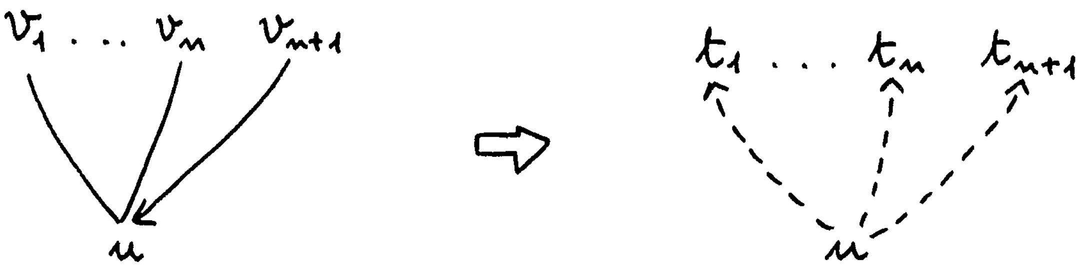
Need to analyze the search process:
for all factors (communication,
overlap, parallel searches, subdivision)
find suitable representation and
measure benefit / cost.

Measuring search complexity

G:



ancestor graph of $\varphi = \ell(u)$ $(at_G(\varphi))$



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

where t_i ancestor-graph of v_i

$w \in t$ is relevant to u in t for p_k if

- $w \in \{v_1 \ldots v_{n+1}\}$ and $\pi_1(c^k(e)) = 0$ or
- w is relevant to v_i in t_i for some i

A notion of distance in search spaces

Past distance:

$$pdist_{G^K}(t) = |\{w \mid w \in t,\ \overset{K}{s}(w) \neq 0\}|$$

Future distance:

$$fdist_{G^K}(t) = \begin{cases} \infty & \text{if } \overset{K}{s}(\varphi) < 0 \text{ or } \\ & \exists w \in Rev_{G^K}(t) \quad \overset{K}{s}(w) < 0 \\ |\{w \mid w \in t,\ s(w) = 0\}| & \end{cases}$$

Global distance:

$$gdist_{G^K}(t) = pdist_{G^K}(t) + fdist_{G^K}(t)$$

$$fdist_{G^K}(\varphi) = \min fdist_{G^K}(t) \qquad t \in at_G(\varphi)$$

Dynamic distance:

$fdist_{G^K}(t)$ measures the part of t that p_K needs to traverse to reach φ via t

if ∞, then unreachable (redundant)

Bounded search spaces

At stage i $(i \geq 0)$ of a derivation define the **bounded search space** reachable (by process P_k) within distance j $(j > 0)$ from the start:

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

where

$$mul_{G^k}(v, j) = \left| \left\{ t : \begin{array}{l} t \in at_G(v), \\ t \text{ allowed for } P_k \\ 0 < gdist_{G^k}(t) \leq j \end{array} \right\} \right|$$

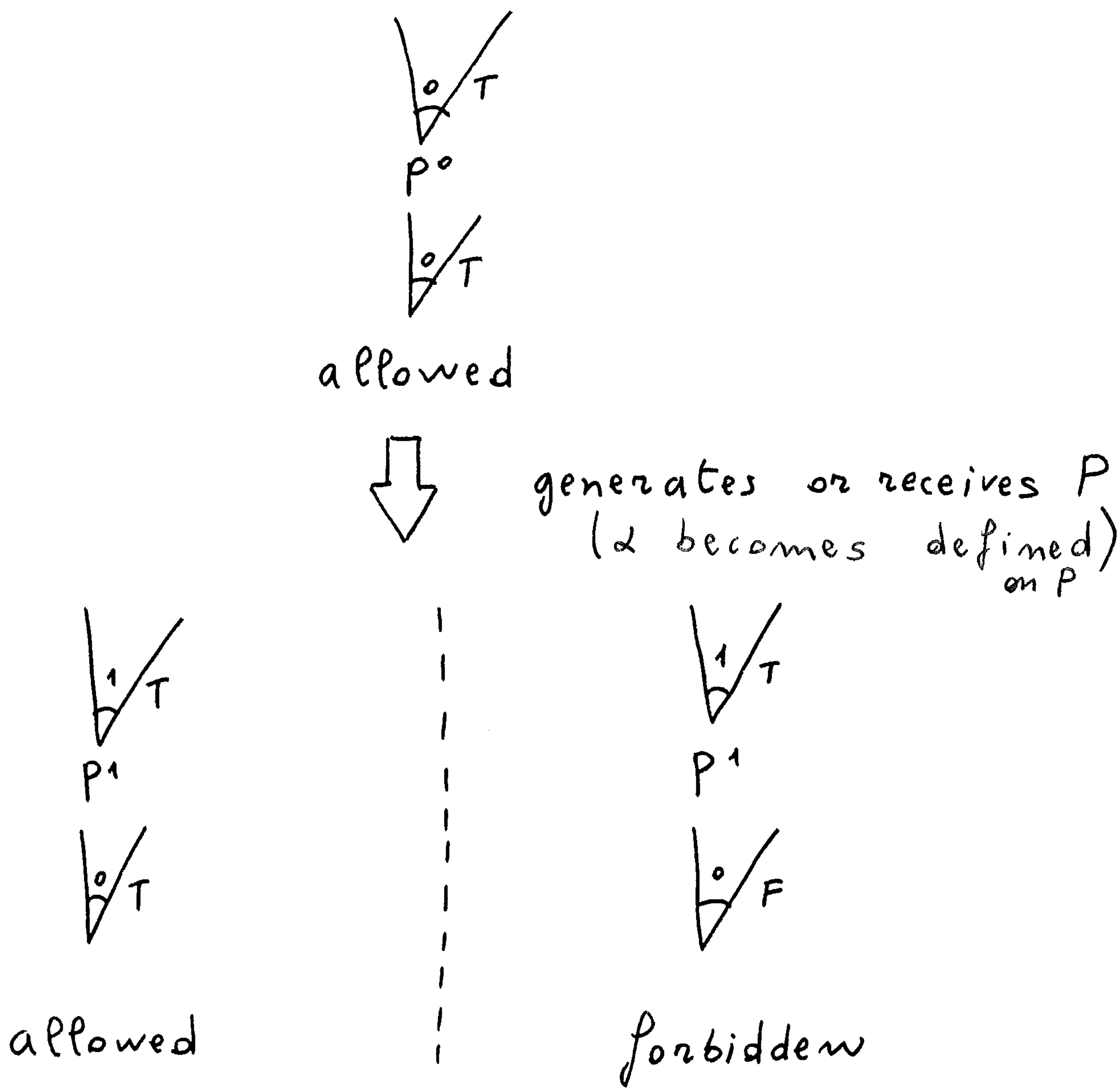
Ancestor-graph **forbidden** for P_k if $\exists e \;\; \pi_1(c^k(e)) = 0$ and $\pi_2(c^k(e)) = false$,

allowed otherwise.

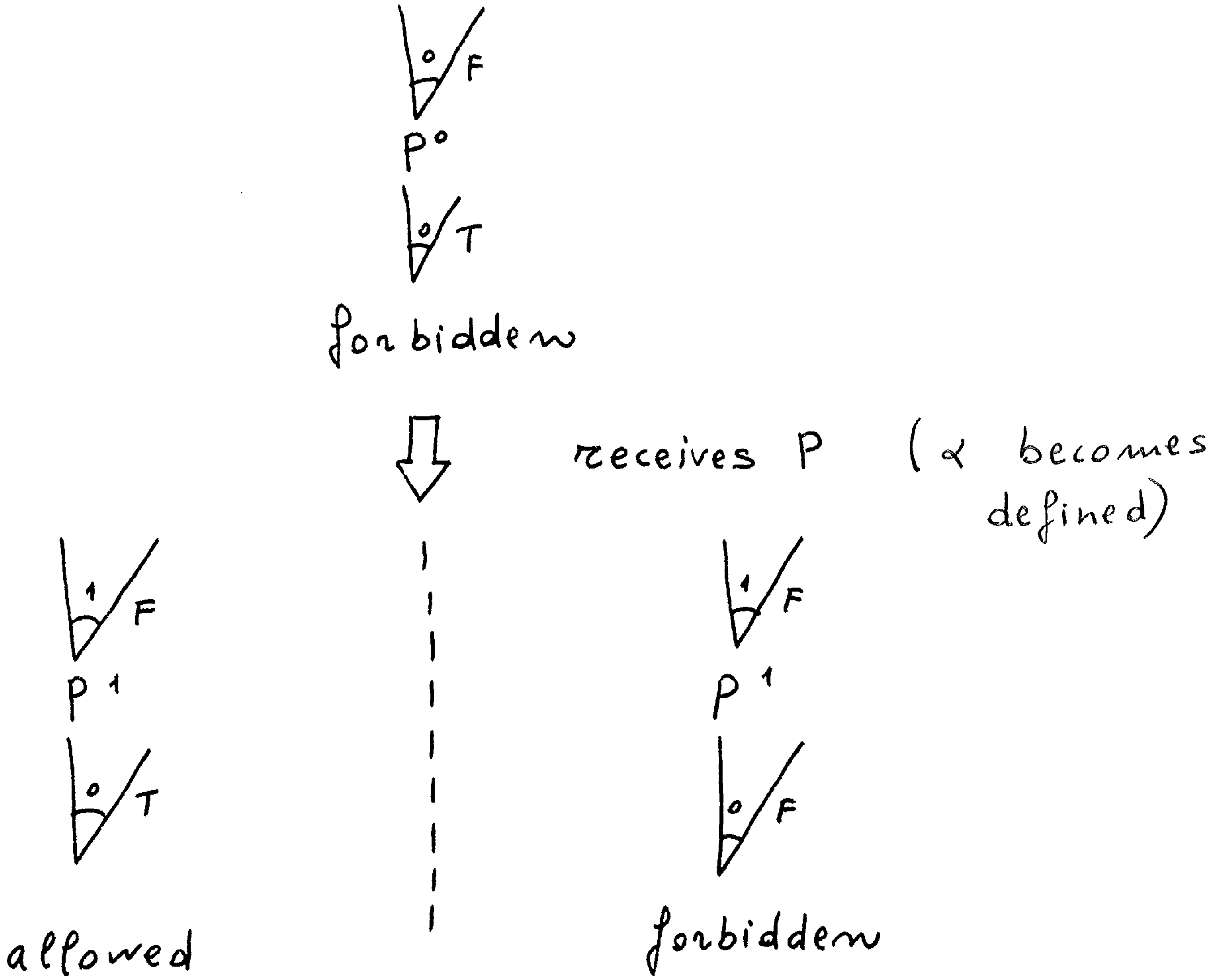
Analysis of the search process in distributed search

Subdivision

Ancestor graph forbidden for p_k if $\exists e \quad \pi_1(c^k(e)) = 0$ and $\pi_2(c^k(e)) = \text{false}$, allowed otherwise.



Communication



P_k and P_h overlap on t if t is allowed for both

Contraction and communication

Sequential contraction - based:
if a deleted φ is re-generated, it is deleted again (monotonic) before being used (eager).

$$\Downarrow$$

$$fdist_i(t) = \infty \quad \text{then} \quad \forall j > i \quad fdist_j(t) = \infty$$

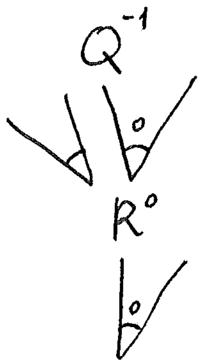
In parallel:
assume local eager contraction:
if a deleted or unreachable φ is re-generated or received, it is deleted again before being used.

$$\Downarrow$$

$$fdist_{G_i^k}(\varphi) = \infty \quad \text{then} \quad \forall j \quad fdist_{G_j^k}(\varphi) = \infty$$

Contraction and communication

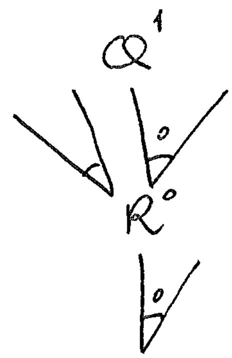
$\boxed{P_k}$



(not redundant)

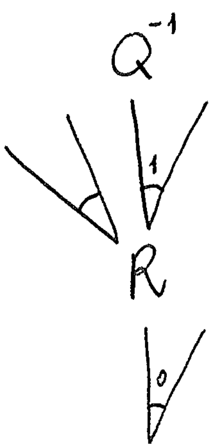
$f\,\mathrm{dist}(t) = \infty$

$\boxed{P_h}$



generates R from Q and sends it

receives R



Q still deleted but no longer relevant

$f\,\mathrm{dist}(t) \neq \infty$

P_h : late contraction

P_k : contraction undone

Evolution of bounded search spaces

1) if $S_i^k \vdash S_{i+1}^k$ generates ψ

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

because of subdivision

2) if $S_i^k \vdash S_{i+1}^k$ replaces ψ by ψ'

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

because of contraction and subdivision

3) if $S_i^k \vdash S_{i+1}^k$ receives ψ

$\forall j \quad \exists l \leq i \quad space(G_{i+1}^k, j) \preccurlyeq_{mul} space(G_l^k, j)$

because of subdivision, subdivision undone and contraction undone

$$\Downarrow$$

Non - monotonic bounded search spaces

Parallel bounded search spaces

$$gmul_G(v,j) = \sum_K mul_{G^K}(v,j)$$

$$pmul_G(v,j) = \lfloor gmul_G(v,j) / n \rfloor$$

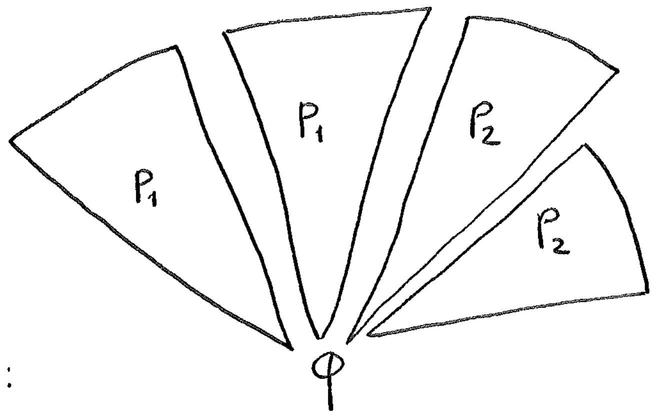
where n is the # of processes

$$\boxed{pspace(G,j) = \sum_{\substack{v \in V \\ v \neq T}} pmul_G(v,j) \cdot \ell(v)}$$

Capture overlap of the processes

Example

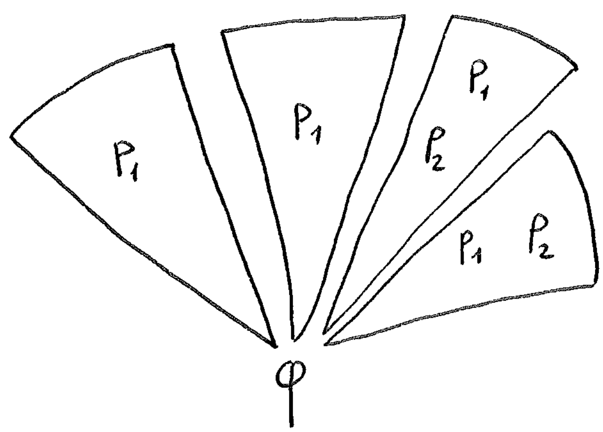
P_1 P_2



No overlap:

$mul_{G^1}(\varphi, j) = 2$ $mul_{G^2}(\varphi, j) = 2$

$gmul_G(\varphi, j) = 4$ $pmul_G(\varphi, j) = 2$



Overlap:

$mul_{G^1}(\varphi, j) = 4$ $mul_{G^2}(\varphi, j) = 2$

$gmul_G(\varphi, j) = 6$ $pmul_G(\varphi, j) = 3$

"average"

Minimize overlap

1) Overlap due to inaccurate subdivision
2) Overlap due to communication

Two properties of α:

No arc-duplication : avoid (1)

No clause-duplication: minimize (2)

Lemma: local eager contraction + no clause-duplication $\Rightarrow$

P_h : allowed to generate φ

$\exists r \quad \forall k \neq h \quad \forall i \geq r \quad \forall j$

$$mul_{G_i^k}(\varphi, j) \leq 1$$

How to compare a distributed-search contraction-based strategy with its sequential base?

Analysis of $\mathcal{C}$ vs. $\mathcal{C}'$

$\mathcal{C} = \langle I, \Sigma \rangle$ sequential contraction-based

$\mathcal{C}' = \langle I, \Sigma' \rangle$ contraction-based parallelization by subdivision of $\mathcal{C}$

Same I $\Rightarrow$ same initial search space

Lemmas:

1) $\varphi \in S_i \Rightarrow \exists p_k \; \exists j \quad \varphi \in S_j^k \cup R(S_j^k)$

2) $\varphi \in R(S_i) \Rightarrow \forall p_k \; \exists j \quad \varphi \in R(S_j^k)$

3) $fdist_{G_i}(\varphi) = \infty \Rightarrow \forall p_k \; \exists j \; \forall \ell \geq j$
either $fdist_{G_\ell^k}(\varphi) = \infty$
or $\forall t \in at_G(\varphi)$ t forbidden

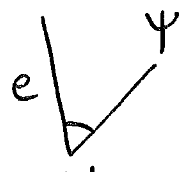
$$\Downarrow$$

Theorem:

$$fdist_{G_i}(\varphi) = \infty \Rightarrow \exists r \; \forall i \geq r \; \forall j \quad pmul_{G_i}(\varphi, j) = 0$$

A limit lemma

Local eager contraction +
immediate propagation
(hence global eager contraction)



$\psi' \in S_\infty - R(S_\infty)$

if $s_i^k(\psi) = -1$ $\quad \psi \in Rev_{G_i^k}(t)$ $\quad$ for P_k then:

1) $\forall P_h \ \forall j \quad s_j^h(\psi) = -1 \quad \Longrightarrow \quad \psi \in Rev_{G_j^h}(t)$

(what is relevant for a process is relevant for all : no late contraction)

2) $\forall j \geqslant i \quad \psi \in Rev_{G_j^k}(t)$

(what is relevant remains relevant : no contraction undone)

$$\Downarrow$$

$f dist_i(t) = \infty$ then $\forall j > i \ \ f dist_j(t) = \infty$

A limit theorem

Assume:

immediate propagation of clauses up to redundancy

no clause-duplication

Lemma:

$$fdist_{G_i}(\varphi) \neq \infty \quad \forall i \quad \Rightarrow \quad \exists r \ \forall i \geq r \ \forall j \quad pmul_{G_i}(\varphi, j) \leq mul_{G_i}(\varphi, j)$$

Theorem:

$$\forall j \ \exists m \ \forall i \geq m \quad pspace(G_i, j) \leq_{mul} space(G_i, j)$$

Significance:

1) "limit theorem" that strategies may approximate (e.g., by reducing overlap)

2) "negative" result which contributes to explain intrinsic difficulty of parallel theorem proving

Discussion

Strategy analysis: study of search in infinite search spaces

Model: marked search graph
Measure: bounded search spaces
Already applied to analysis of contraction
Now: distributed search

Analytic comparison:

"Limit theorem" explains nature of problem (overlap + communication / contraction)

When adopting asynchronous distributed search one expects that contraction may be delayed, but synchronizing on every inference is hopeless, and one may conjecture subdivision compensates for late contraction: not so in general (worst-case scenario).

Relevant to problems where eager contraction is important: not a small class based on experience.

Directions for future work

On analysis of parallel search:

- Reordering of search relevant to both
  - distributed search
  - multi-search

On analysis of theorem proving:

- Comparison of search plans
- Subgoal-reduction strategies
- Reasoning modulo a theory

Distributed search for CBS

Clause-Diffusion (1992)

Aquarius	(Otter 2.2)	1992
Peers	(OPS 1/93)	1993-94

Modified Clause-Diffusion (1994-96)

Peers-mcd	(EQP 0.9)	1996-98
"	(EQP 0.9d)	1999
"	+ hybrid mode	2000

Hybrid mode: distributed search + multi-search

Levi Commutator Problem in group theory:
super-linear speed-up

Robbins algebras are Boolean:
super-linear speed-up

Moufang identities in alternative rings
without cancellation laws built-in
(EQP cannot do)