Parallel Deduction: the Clause-Diffusion method

(joint work with Jieh Hsiang at SUNY Stony Brook)

Maria Paola Bonacina
INRIA Lorraine
Nancy (France)

Outline

1) Analysis of the parallelization of deduction strategies.

   - Size and dynamicity of the data base of clauses versus the granularity of parallelism.

2) The Clause-Diffusion methodology for distributed deduction.

   - Parallelism at the search level.

   - Experiments.

Deduction strategies

First component:

an inference system.

Examples:

resolution (expansion)

$$\neg M(a2, b2, r1) \qquad \neg M(x,y,z) \vee M(y,x,z)$$

$$\neg M(b2, a2, r1)$$

simplification (contraction)

$$P(f(f(f(f(0)))))$$

$$\Big\downarrow {*} \qquad f(f(x)) \rightarrow f(x)$$

$$P(f(0))$$

Deduction strategies

Second component:

a search plan.

Example: given

1. $\neg M(a2, b2, r1)$
2. $\neg M(x, y \cdot y, z) \vee M(y, x, z)$
3. $y \cdot y \simeq y$

which step do first? ➡

Which premises,

which rule

for the next step?

Example:

simplification-first search plan.

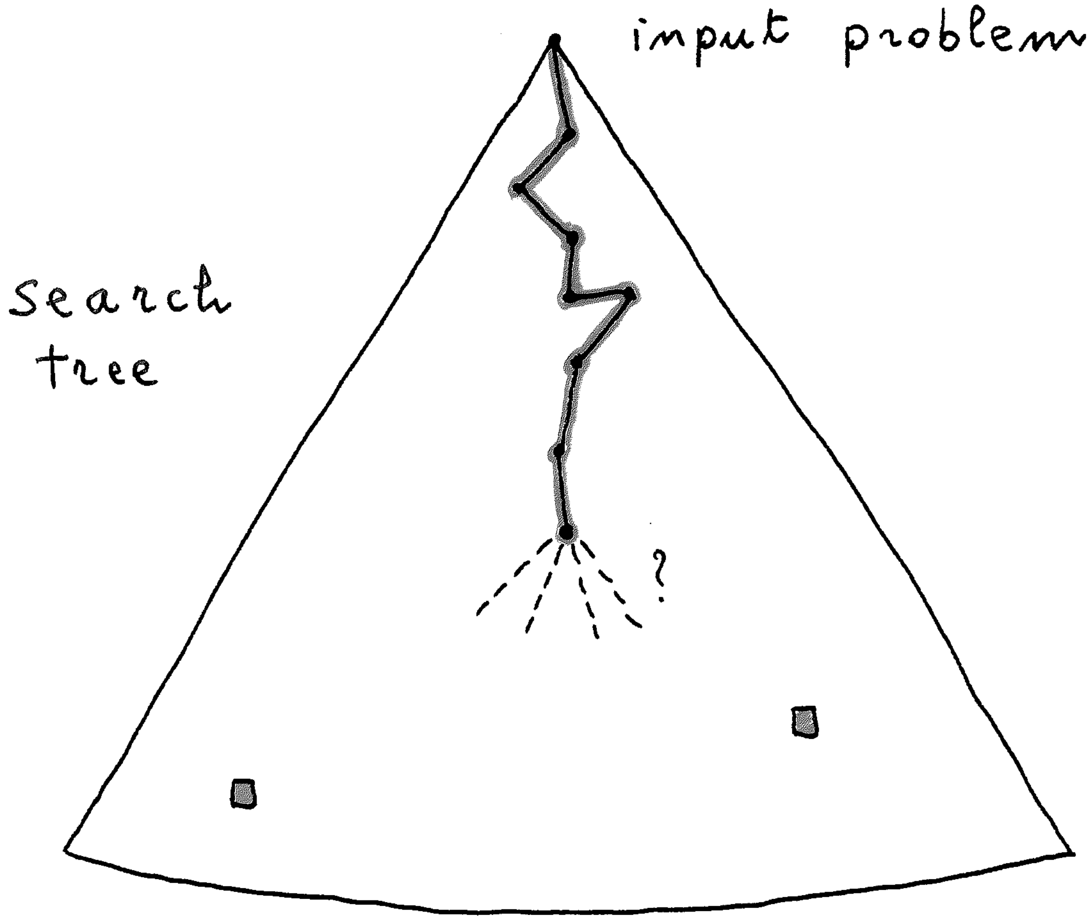
Deduction strategy : $\mathcal{C} = \langle I ; \Sigma \rangle$

Derivation:

$S_0 \vdash_{\mathcal{C}} S_1 \vdash_{\mathcal{C}} \ldots \; S_i \vdash_{\mathcal{C}} S_{i+1} \ldots$

As a search:

■ : solution



Classification of strategies for the purpose of parallelization

- Subgoal-reduction strategies.
  (S-R)

- Expansion-oriented strategies.
  (E-O)

- Contraction-based strategies.
  (C-B)

Subgoal-reduction strategies

Deduction strategies for:

- logic programming
- functional programming (term rewriting)
- Prolog Technology Theorem Proving

...

A derivation has the form:

$$(S ; \varphi_0 ; A_0) \vdash (S ; \varphi_1 ; A_1) \vdash \ldots (S ; \varphi_i ; A_i) \vdash \ldots$$

S: static set of axioms

φ_i : goal

A_i : ancestors of φ_i (backtracking)

Expansion-oriented strategies

Theorem proving strategies with

- resolution
- hyperresolution
- paramodulation

...

A derivation has the form:

$$S_0 \vdash S_1 \vdash S_2 \; \ldots\ldots \; S_i \vdash S_{i+1} \vdash \ldots$$

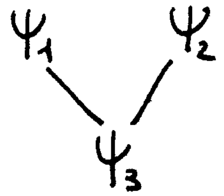
where S is monotonically increasing by expansion steps:

$$S_0 \subseteq S_1 \subseteq \; \ldots\ldots \; S_i \subseteq S_{i+1} \subseteq \ldots$$

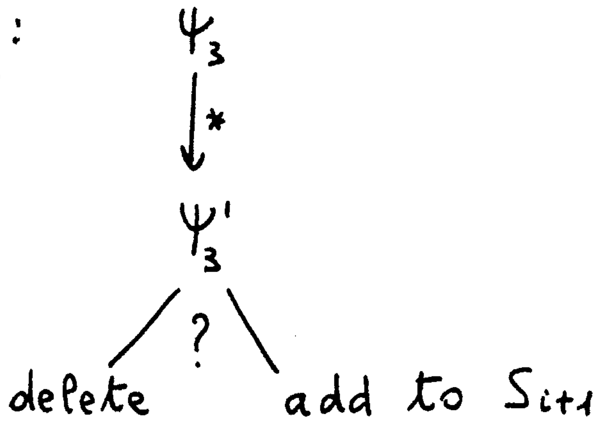
Forward contraction

Expansion-oriented strategies may feature <u>forward contraction</u>:

pick ψ_1, ψ_2 in S_i

generate ψ_3: 

"<u>raw clause</u>"

contract ψ_3: 

Thus:

$$(S_0 ; N_0) \vdash (S_1 ; N_1) \vdash \ldots (S_i ; N_i) \vdash (S_{i+1} ; N_{i+1}) \vdash \ldots$$

S_i : <u>main data base</u> (<u>no contraction</u>)

N_i : "<u>raw clauses</u>"

$$S_0 \subseteq S_1 \subseteq \ldots \quad S_i \subseteq S_{i+1} \subseteq \ldots$$

Contraction-based strategies

Theorem proving strategies with

- subsumption
- simplification / normalization
- conditional simplification

...

- expansion rules (restricted)
- contraction-first search plan.

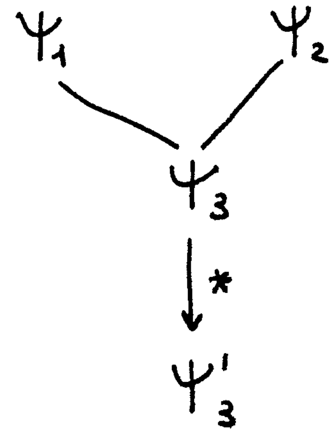
Key concept: have well-founded ordering(s) on terms to define contraction rules and restrict expansion rules.

Theorem provers:

RRL, Otter, Reveal ...

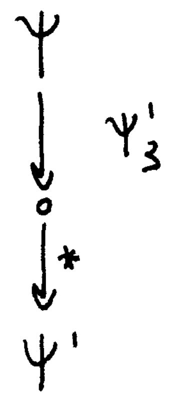
Backward contraction

Contraction-based strategies feature forward and backward contraction:



forward contraction

Use ψ'_3 to contract any ψ in S:



backward contraction

Use ψ' to contract others ...

Contraction-based strategies

A derivation has the form

$$S_0 \vdash S_1 \vdash \ldots \vdash S_i \vdash S_{i+1} \ldots$$

where for all i

$S_i \subseteq S_{i+1}$ expansion

$S_i \nsubseteq S_{i+1}$ contraction (forward or backward)

No monotonicity property:

highly dynamic data base.

Granularity of parallelism

	granularity of data	granularity of operations
parallelism at the term level (fine grain)	TERM	SUBTASK OF INFERENCE STEP
parallelism at the clause level (medium grain)	CLAUSE	INFERENCE STEP
coarse grain parallelism	SET OF CLAUSES	MANY INFERENCE STEPS

	S-R	E-O	C-B
size of the db	small	very large	very large
dynamicity of the db	static	monotonic	dynamic
pre-processing	yes	no	no
read-only data	yes	yes	no
special data structures	yes	no	no
conflicts	no	no	yes

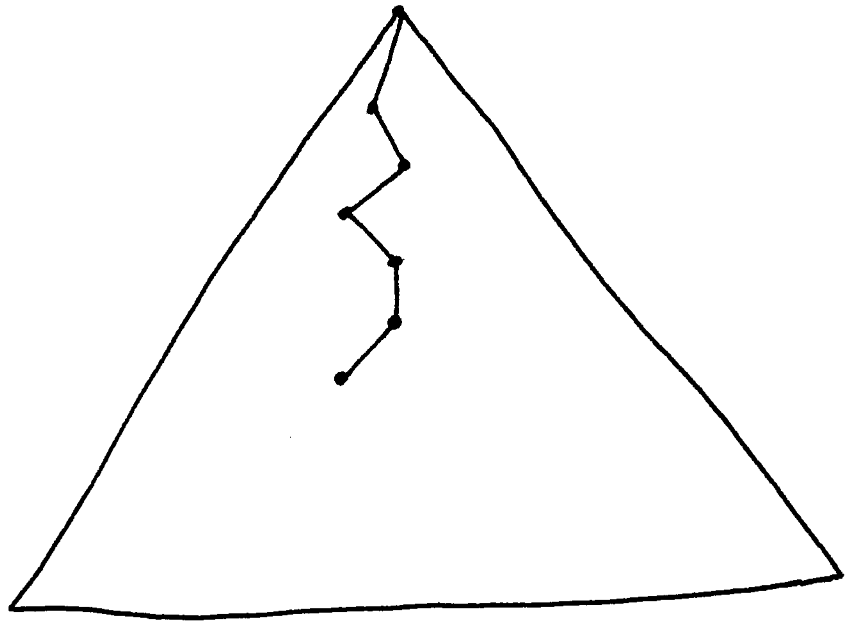
Types of parallelism and strategies

	S-R	E-O	C-B
parallelism at the term level	✓		
parallelism at the clause level	✓	✓	
coarse grain parallelism	✓	✓	✓

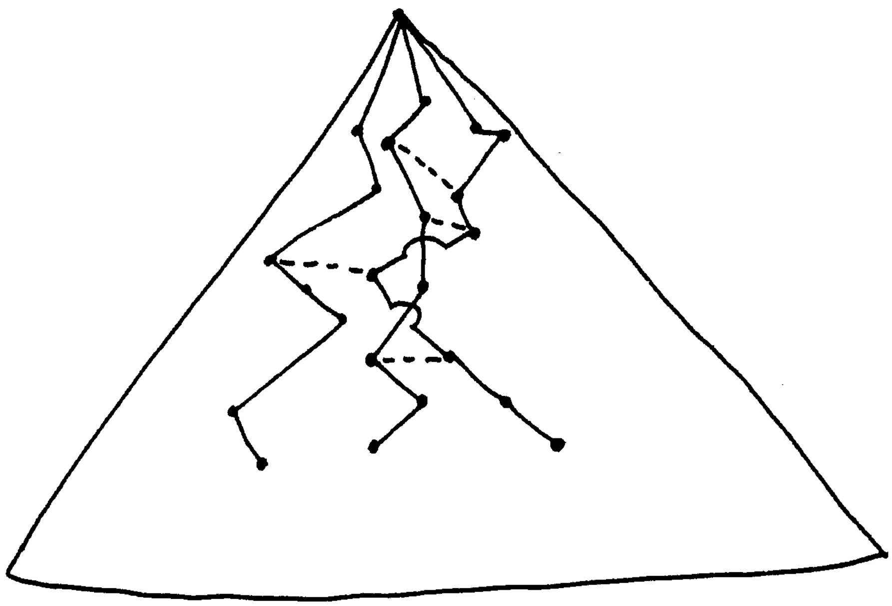
Coarse grain parallelism

is parallelism at the search level

Sequential search:



Parallel search:



Parallelism at the search level in the Clause-Diffusion method

Concurrent, asynchronous, loosely-coupled deductive processes

develop their own derivations

by working on separate sets of clauses (no conflicts)

and

by exchanging clauses as messages.

Success is reached as soon as one of the processes succeeds.

Partition the search space

At the clause level:

subdivide the data base of clauses.

For all ψ assign ψ to a process p_i:

ψ is a resident of p_i.

$\psi \in S^i$

$$\bigcup_{i=0}^{n-1} S^i = S$$

S is the global data base.

Partition the search space

At the inference level:

expansion inferences:

if $\psi_1 \in S^i$ and $\psi_2 \in S^j$

P_i paramodulates ψ_2 into ψ_1
but not vice versa.

It prevents the systematic duplication of expansion steps.

It applies also to other rules, e.g. resolution, hyperresolution, unit resulting resolution ...

Communication of clauses

No process is guaranteed to find a proof by using its residents only.

⇓

Each process sends / broadcasts its residents as "inference messages":

at p_i $(S^i ; M^i)$

S^i: residents

M^i: received inference messages

- Logical partition vs. physical partition.

- Cooperation rather than competition.

No general subdivision of <u>contraction inferences</u> based on ownership of clauses.

In a contraction-based strategy, each process tries to contract as much as possible residents and messages before expansion and communication.

<u>Local</u> contraction (w.r.t. S^i) and <u>global</u> contraction (w.r.t. $\bigcup_i S^i$) by <u>schemes for distributed global contraction</u>.

A Clause-Diffusion strategy

is specified by

- the set of inference rules,
- the search plan which schedules
  - contraction steps
  - expansion steps
  - communication steps

  at each process,
- the algorithm to allocate clauses as residents,
- the mechanism for message-passing,
- the scheme for distributed global contraction.

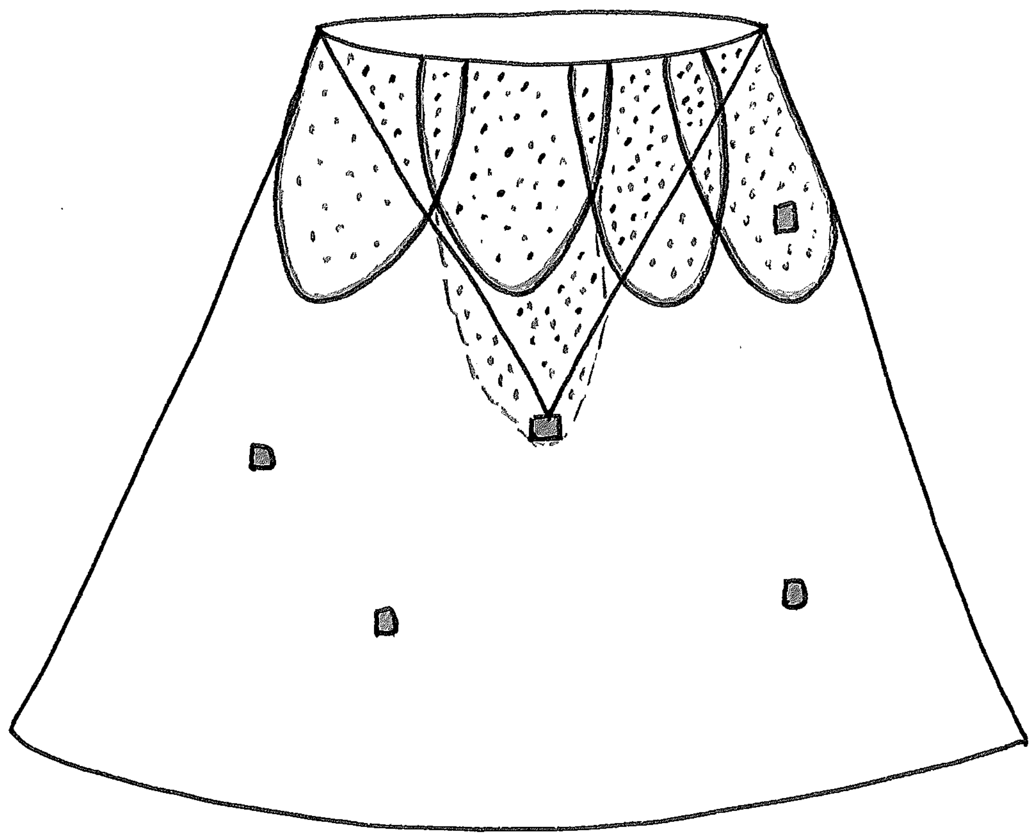
Experiments with Aquarius:

Problem	*Aquarius-1*	*Aquarius-2*	*Aquarius-3*
cd12	104.18	50.98	47.56
cd13	98.79	45.32	51.07
cd90	3.10	0.63	11.87
cn	5.04	8.63	14.50
ec	3.03	1.96	1.77
imp1	6.63	2.64	3.54
imp2	7.25	3.31	7.43
imp3	32.05	17.92	38.89
luka5	844.20	299.24	1079.45
pigeon (ph4)	8.21	7.66	8.14
robbins2	21.62	22.91	24.12
salt	3.89	4.45	5.49
sam's lemma	6.35	5.40	3.90
subgroup	15.55	9.36	17.40
w-sk	3.50	3.52	3.34

Experiments with Peers:

Problem	*1-Peers*	*2-Peers*	*4-Peers*	*6-Peers*	*8-Peers*
	96.45	50.29	43.28	30.66	7.51
	95.86	49.16	44.52	31.65	8.60
	96.28	53.58	46.87	54.04	25.95
x3	96.06	51.37	44.06	43.52	28.06
	96.36	87.64	38.34	24.93	31.02
	39.95	15.69	9.89	33.95	64.15
r2	40.04	16.51	18.74	34.97	22.31
	39.98	15.52	7.75	22.08	51.74
	39.98	18.49	34.26	41.70	34.40
	15.99	7.30	16.06	12.96	9.65
sa1	15.92	30.12	16.03	40.02	20.39
	15.74	14.40	8.94	17.41	16.81
	16.15	36.57	13.16	25.80	18.62
	24.28	20.09	12.76	81.05	20.34
sa2	24.10	23.55	16.55	11.04	46.92
	24.19	49.03	63.07	15.00	74.31

Discussion on the experiments



■ : solutions

black: sequential search

gray: parallel search by Clause - Diffusion

Current and future work

- Study of parallel search
  - partitioning a search space: blind / informed
  - parallel search plans
- Fine-tuning of implementations:
  - heuristics, e.g. for the allocation of clauses
  - reconstructing distributed proofs
  - experiments.

Summary

- Analysis of classes of deduction strategies and parallelism.
- Contraction-based strategies.
- Backward contraction.
- Coarse grain parallelism.
- Parallelism at the search level: no synchronization / duplication.
- Overview of the Clause-Diffusion methodology.
- Experiments.