Ten years of parallel theorem proving: a perspective

Maria Paola Bonacina
Dept. of Computer Science
The University of Iowa

Motivation

Ten years of parallel theorem proving: 1989 - 1999

Strategies in Automated Deduction → Control of deduction ← Parallel theorem proving

⇩

A taxonomy of parallel strategies based on how they search

Relevant to:

Analysis of strategies

Engineering of theorem provers

Outline

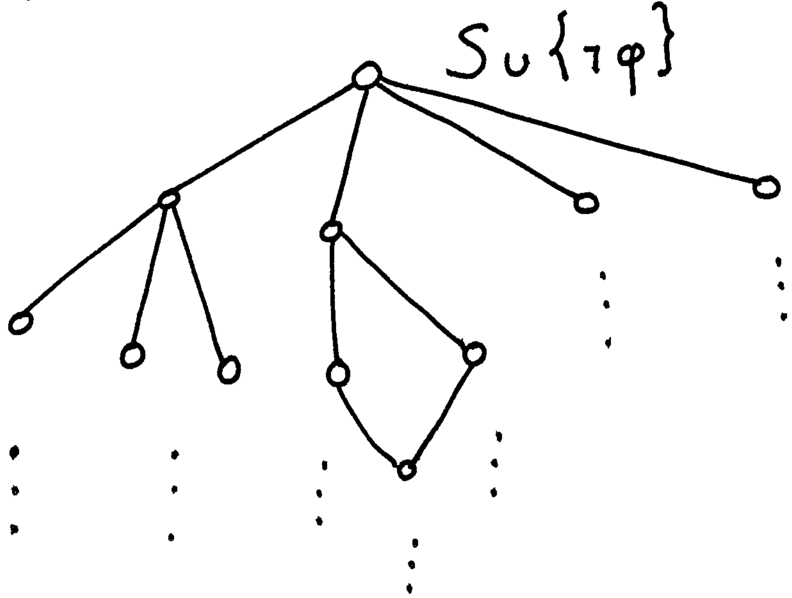
- A taxonomy of sequential theorem proving strategies
- Principles of parallelization
- How they affect the control of search and vice versa: a taxonomy of parallel theorem proving strategies
- Parallel search plans
- Discussion: trends and open problems

A taxonomy of sequential strategies

Theorem proving as search problem

Inference system I

$S \cup \{\neg\varphi\} \overset{?}{\vdash_I} \bot$



vertex : state arc : inference

path : derivation

Search plan Σ : determines unique derivation

Refutationally complete I

Fair Σ

$\Rightarrow$ theorem-proving strategy $\mathcal{C} = \langle I, \Sigma \rangle$ complete

General scheme of search plan

$$\Sigma = \langle \zeta, \xi, \omega \rangle$$ (at least)

- rule-selecting function

  $\zeta : States^* \longrightarrow I$

- premise-selecting function

  $\xi : States^* \longrightarrow \mathcal{P}(\mathcal{L}_{\Theta})$

- termination-detecting function

  $\omega : States \longrightarrow Bool$

Two main classes of strategies

Ordering-based strategies	Subgoal-reduction strategies
work on a set of objects (state)	work on one goal-object at a time (state)
build many proof attempts implicitly	build one proof attempt at a time
no backtracking	backtracking
contraction	no contraction

Ordering-based strategies

Resolution + paramodulation → ordering based strategies ← Term rewriting + Knuth-Bendix

E.g.,

Otter, RRL, Reveal, SNARK,
EQP, Barcelona, CLIN-S, SPASS,
Gandalf, OSHL, daTac ...

Ordering based strategies

$>$: well-founded ordering on data

Expansion: $f: \frac{S}{S'}$ $\quad S \subset S' \quad S < S'$

e.g.,
resolution, hyperresolution, paramodulation, superposition

Contraction: $f: \frac{S}{S'}$ $\quad S \not\subset S' \quad S > S'$

e.g.,
simplification, subsumption, taut. deletion, purity deletion, clausal simplification

Redundancy

Ordered inference rules

Critical pair criteria

"Basic" inference rules

Ordering based strategies

$\Sigma = < \zeta, \xi_1, \xi_2, \omega >$

- $\xi_1 : States^* \longrightarrow \mathcal{L}_\Theta$

  $\xi_1(S_0 \ldots S_i) = \psi_1 \in S_i$

- $\zeta : States^* \times \mathcal{L}_\Theta \longrightarrow I$

  $\zeta(S_0 \ldots S_i, \psi_1) = f$

- $\xi_2 : States^* \times \mathcal{L}_\Theta \times I \longrightarrow \mathcal{P}(\mathcal{L}_\Theta)$

  $\xi_2(S_0 \ldots S_i, \psi_1, f) = \{\psi_2 \ldots \psi_n\} \subseteq S_i$

search plans of Otter and EQP

indexing

forward / backward contraction

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

Expansion - oriented

$(S_0; N_0) \vdash (S_1; N_1) \vdash \ldots (S_i; N_i) \vdash \ldots$

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

forward contraction

Contraction - based

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

$R(S_0) \subseteq R(S_1) \subseteq \ldots R(S_i) \subseteq R(S_{i+1}) \subseteq \ldots$

eager forward and backward contractio

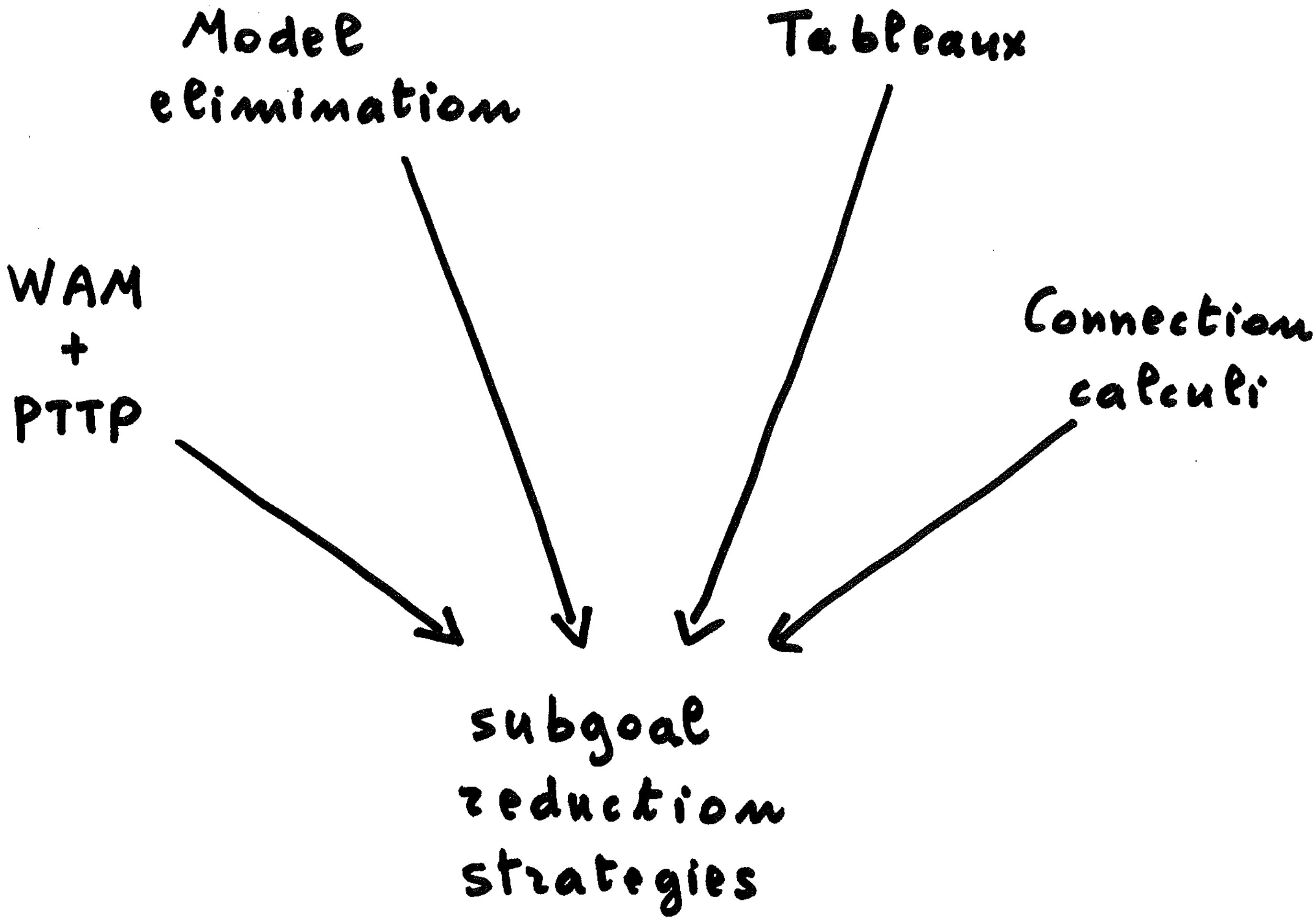
Target - oriented

$(S_0; \varphi_0) \vdash (S_1; \varphi_1) \vdash \ldots (S_i; \varphi_i) \vdash \ldots$

Semantic | Supported

$(T_0; SOS_0) \vdash (T_1; SOS_1) \vdash \ldots (T_i; SOS_i) \vdash \ldots$

Subgoal-reduction strategies



E.g.:

Setheo, METEOR, Protein, TAP,

Tatzelwurm, Mission, KoMeT ...

Subgoal reduction strategies

$$\Sigma = < \zeta, \xi_1, \xi_2, \omega >$$

- $\xi_1 : \text{States}^* \longrightarrow \mathcal{L}_{\textcircled{H}}$

  $\xi_1((T_0; X_0) \ldots (T_i; X_i)) = L \in \text{open}(X_i)$

- $\zeta : \text{States}^* \times \mathcal{L}_{\textcircled{H}} \longrightarrow I \cup \{\text{backtrack}\}$

- $\xi_2 : \text{States}^* \times \mathcal{L}_{\textcircled{H}} \times I \longrightarrow \mathcal{L}_{\textcircled{H}}$

  $\xi_2((T_0; X_0) \ldots (T_i; X_i), L, \varphi) = \psi \in T_i$

$\xi_1 \simeq$ AND-computation rule

$\xi_2 \simeq$ OR-computation rule

$$(T_0 ; X_0) \vdash (T_1 ; X_1) \vdash \ldots (T_i ; X_i) \vdash \ldots$$

Analytic - tableaux

$$X_0 \vdash X_1 \vdash \ldots X_i \vdash \ldots$$

Linear clausal

$$(T_0 ; \varphi_0 ; A_0) \vdash \ldots (T_i ; \varphi_i ; A_i) \vdash \ldots$$

Linear - input clausal

$$(T_0 ; \varphi_0) \vdash \ldots (T_i ; \varphi_i) \vdash \ldots$$

ME - tableaux

$$(T_0 ; X_0) \vdash \ldots (T_i ; X_i) \vdash \ldots$$

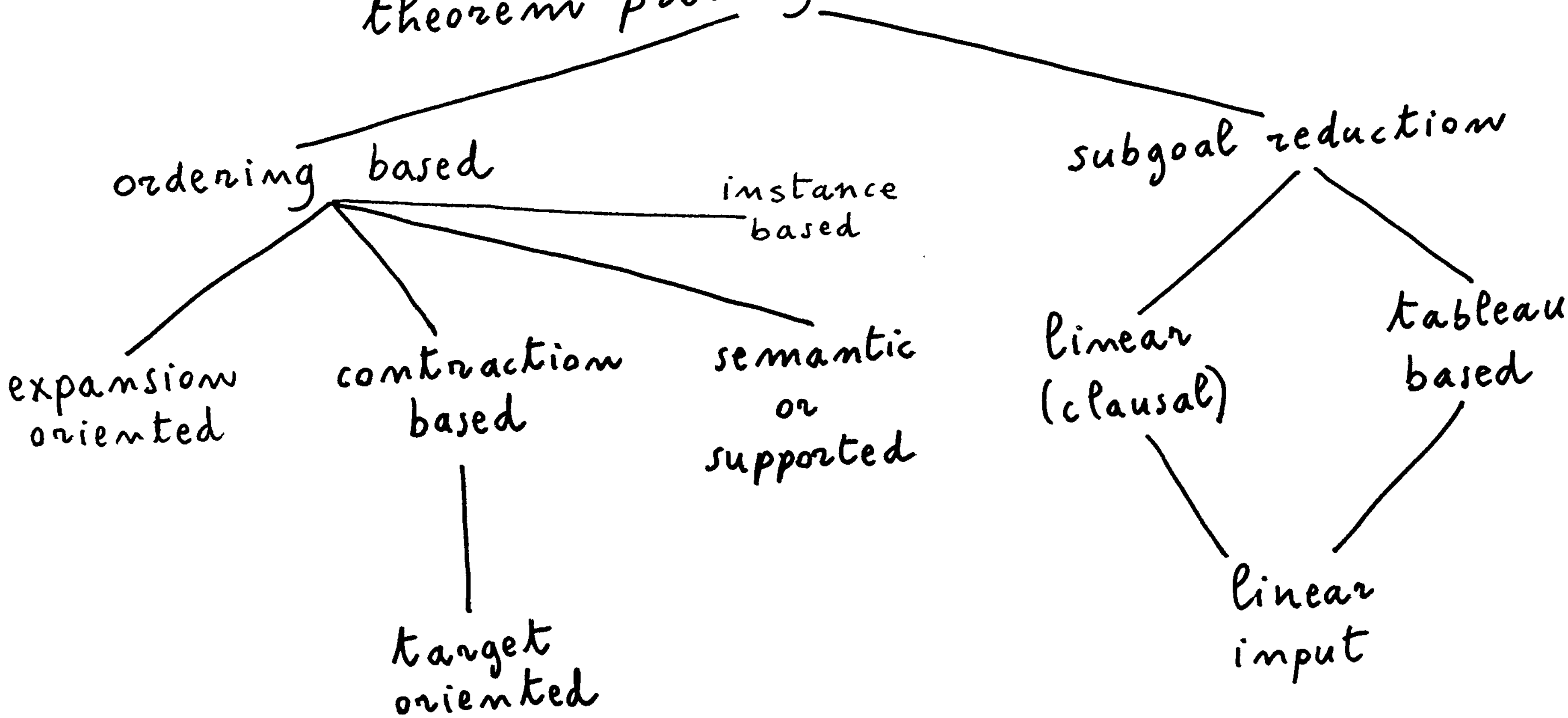
Term rewriting

$$(T_0 ; \varphi_0) \vdash \ldots (T_i ; \varphi_i) \vdash \ldots$$

A taxonomy of strategies

First order, general purpose, fully automated

theorem proving strategies



Principles
of
parallelization

PARALLELISM	DATA ACCESSED IN PARALLEL	PARALLEL OPERATIONS
AT THE TERM LEVEL	subexpressions of formulae	subtasks of inferences
AT THE CLAUSE LEVEL	formulae	inferences
AT THE SEARCH LEVEL	sets of formulae	derivations

Principles of parallelization

Parallelism at the term level:

parallel inner algorithms
below the inference level
speed-up frequent operations

Parallelism at the clause level:

parallel inferences
speed-up the given search

Parallelism at the search level:

generate many different searches
in parallel ⇒ parallel
derivations

Parallel search

How differentiate / combine the parallel search processes?

Distributed search:

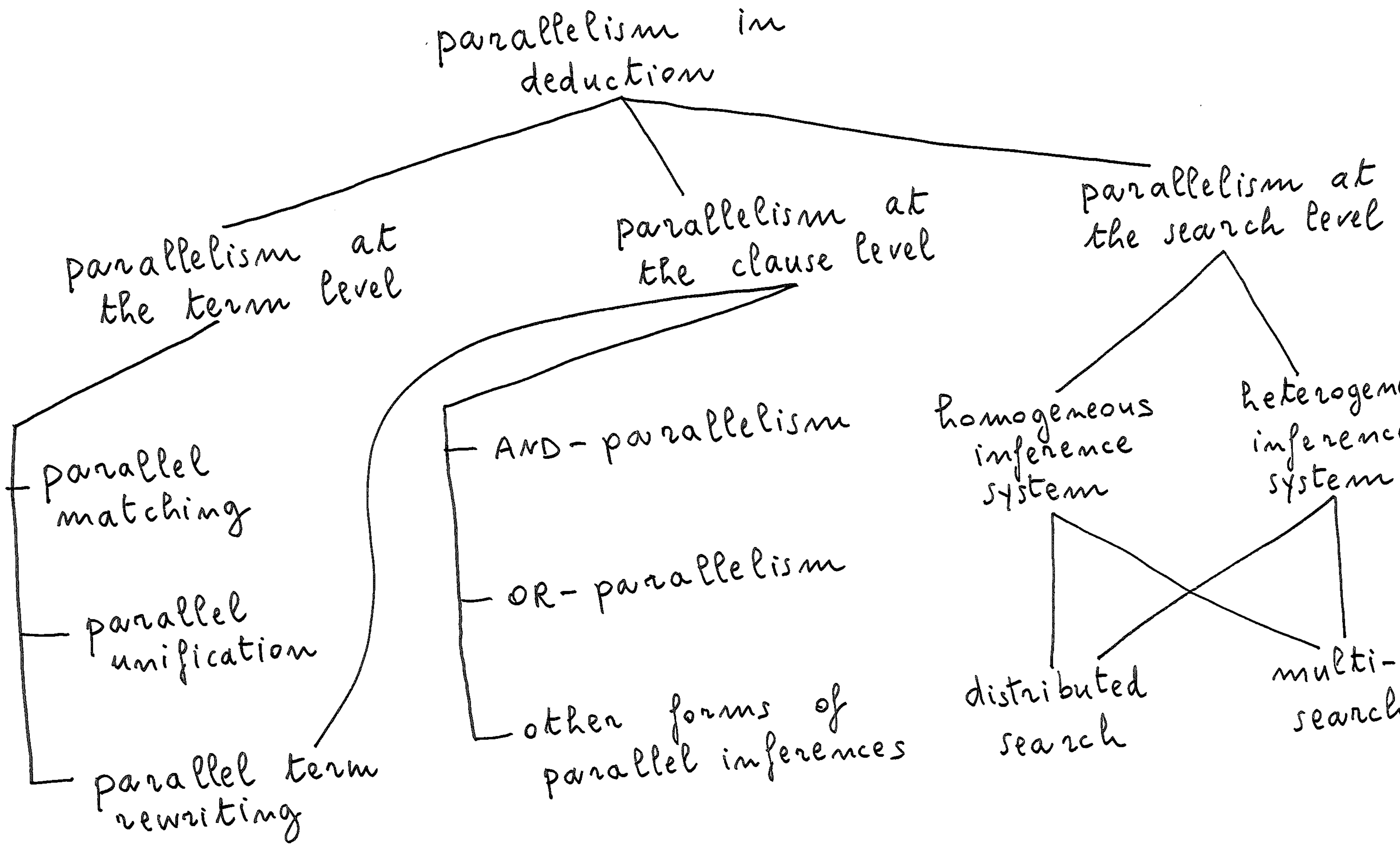
subdivide the search space
decompose the problem
motivation: divide work

Multi-search:

whole problem to each process
different search plans
motivation: different orders

Heterogeneous systems:

different inference systems
motivation: combine backward and forward reasoning, e.g., lemmas


parallelism in deduction
parallelism at the term level
parallelism at the clause level
parallelism at the search level
parallel matching
parallel unification
parallel term rewriting
AND-parallelism
OR-parallelism
other forms of parallel inferences
homogeneous inference system
system
distributed search

A taxonomy of parallel theorem proving strategies

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

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

What is Σ' ?

The search plan and parallelism at the term level

- Concurrent rewriting
  [Kirchner - Viry 1990,
  Alouimi 1995,
  Alouimi - Kirchner 1996]

- Concurrent completion
  [Kirchner - Lynch - Scharff 1996]

- "Concurrent theorem proving"
  [Fisher 1997]

- PaReDuX
  [Bündgen - Göbel - Küchlin
  1994 - 1995 - 1996]

Example concurrent rewriting

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

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

Terms as trees: disjoint redexes

Terms as dags: non-overlapping redexes

∇

(maximal) concurrent rewriting

Data structure dictates what is done in parallel:

Σ' superfluous

Data-driven concurrency

Also:

conditional rewriting

ground completion

unit resolution for Horn propositional clauses

Non-ground inferences?

"Strategy-compliant" parallelization: guarantees same order of steps

The search plan and parallelism at the term level

Parallelism is introduced <u>below</u> the level where the search plan makes decisions

Possibilities:

1) Concurrency <u>replaces</u> the search plan (e.g., concurrent rewriting)

2) "Strategy-compliant" parallelization (e.g., PaReDuX):

$\Sigma' = \Sigma$ $\mathcal{C}' = \mathcal{C}$

The search plan and parallelism at the clause level

ORDERING - BASED

- Expansion-oriented
  PARROT [Jindal-Overbeek-Kabat 1992]

- Instance-based
  Parallel hyper-linking [Lee-Wu 1997]

- Contraction-based
  ROO [Lusk-McCune 1990]
  Transition-based parallel completion [Yelick-Garland 1992]

SUBGOAL-REDUCTION

- PTTP (clausal linear-input)
  PARTHENON [Bose-Clarke-Long-Mychalov 1989]
  METEOR [Astrachan-Loveland 1991]

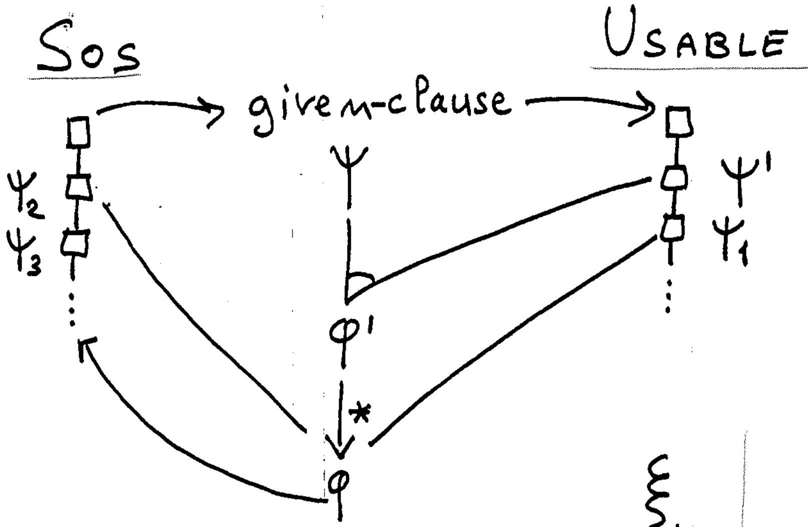
- ME-tableaux (linear-input)
  PARTHEO [Schumann-Letz 1990]

- Analytic tableaux
  HOT [Konrad 1998]

- Gentzen-style subproofs in NUPRL
  [Hickey 1999]

Example: ROO parallelization of Otter

$\mathcal{E} = \langle \zeta, \xi_1, \xi_2, \omega \rangle$: given-clause algorithm



	ξ_1	ζ	ξ_2
Expansion	ψ	exp. rule	ψ'
Forward contraction	φ'	contr. rule	ψ_1, ψ_2
Backward contraction	φ	contr. rule	ψ_3

$\mathcal{E}' = \langle \zeta, \xi'_1, \xi_2, \omega \rangle$: shared Sos and USABLE

Each process does a given-clause (Task A)

Problems: concurrent append / redundant insertions / concurrent deletion

Add lists K-LIST and TO-BE-DELETED : Task B

At most one process does Task B

(Problem: backward contraction bottleneck)

⇨ Parallel plan as task scheduler

Parallel plan as task scheduler

Also:

PTTP with OR-parallelism
(e.g., PARTHENON, METEOR)

ME-tableaux with OR-parallelism
(e.g., PARTHEO)

Analytic tableaux
(e.g., HOT)

Gentzen-style proofs
(e.g., in NUPRL)

Common traits:

Shared data structure:
stack of goals, blackboard ...

Tasks: subgoals, open leaves ..

Control: task scheduling
(e.g., task stealing)

The search plan and parallelism at the clause level

Parallelism is introduced *at* the level where the search plan makes decisions.

Sequential search plan: order the inferences within one derivation.

Parallelism at the clause level: parallelize the inferences within derivation.

⇓

Parallel search plan: scheduler assigning inferences to processes.

The search plan and parallelism at the search level

Each process generates a derivation

▼

Each process needs to execute a search plan

Such search plan needs to control:

- inferences (like sequential)
- subdivision of work (distributed search)
- communication
  (both distributed and multi-search)
  for: completeness
  cooperation
  load-balance

Two approaches:

- hierarchy of processes
  (e.g., master-slaves)
- peer processes

The search plan and parallelism at the search level

Parallel search with master-slaves:

- Multi-search:
  Team-Work (contraction-based)
  [Avenhaus - Denzinger - Fuchs - Kronenburg - Schulz 1993, 1994, 1995, 1996]

- Distributed search:
  PSATO (satisfiability)
  [Zhang - Bonacina - Hsiang 1994, 1996]

- Heterogeneous systems:
  HPDS (subgoal reduction + expansion oriented)
  [Sutcliffe 1992]
  Distributed Larch Prover (contraction-based)
  [Vandevoorde - Kapur 1996]

- Multi-search + "AND-parallelism":
  "Nagging" (subgoal-reduction : PTTP)
  [Sturgill - Segre 1994, 1997]

The search plan and parallelism at the search level

Parallel search with master slaves:

- separate the control of parallelism and the control of deduction;
- each slave executes a sequential search plan to generate its derivation;
- all other control issues (subdivision, communication, selection of data for cooperation, user interface) are dealt with by the master which does not do deductions;

$$\Sigma' = \langle \Sigma_1, \Sigma_2, \ldots \Sigma_n \rangle$$

The search plan and parallelism at the search level

Parallel search with peer processes:

- **Distributed search:**

  DARES (expansion-oriented)
  [Conry - MacIntosh - Meyer 1990]
  Clause-Diffusion (contraction-based)
  [Bonacina, Bonacina-Hsiang 1992, 1993, 1995]
  [Bonacina - McCune 1994]
  Modified Clause-Diffusion
  [Bonacina 1994, 1996, 1997]

- **Heterogeneous systems**
  (combinations of provers):

  SPASS // DISCOUNT (contraction-based)
  [Fuchs 1998]
  CPTHEO = SETHEO // DELTA (subgoal-reduction + expansion-oriented)
  [Fuchs-Wolf-Letz 1998]

The search plan and parallelism at the search level

Parallel search with peer processes:

- no central control;
- the search plan executed by each peer process needs to take care of the control of parallelism (e.g., communication, subdivision) together with the control of deduction

▼

parallel-search plans

Search plan with communication

A set I of inference rules

$$f : \mathcal{P}(\mathcal{L}_\Theta) \longrightarrow \mathcal{P}(\mathcal{L}_\Theta) \times \mathcal{P}(\mathcal{L}_\Theta)$$

A set M of communication operators:

send: $\mathcal{P}(\mathcal{L}_\Theta) \longrightarrow \mathcal{P}(\mathcal{L}_\Theta) \times \mathcal{P}(\mathcal{L}_\Theta)$

s.t. $\forall X \quad send(X) = (\emptyset, \emptyset)$

receive: $\mathcal{P}(\mathcal{L}_\Theta) \longrightarrow \mathcal{P}(\mathcal{L}_\Theta) \times \mathcal{P}(\mathcal{L}_\Theta)$

s.t. $\forall X \quad receive(X) = (X, \emptyset)$

$$\boxed{\Sigma = \langle \zeta, \xi, \omega \rangle}$$

- ζ: $States^* \times \mathbb{N} \times \mathbb{N} \longrightarrow I \cup M$

  select rule/operator

- ξ: $States^* \times \mathbb{N} \times \mathbb{N} \times (I \cup M) \longrightarrow \mathcal{P}(\mathcal{L}_\Theta)$

  select premises

  e.g., $\xi(S_0 \ldots S_i, n, k, f) \subseteq S_i$

- ω: $States \longrightarrow Bool$

  detect success

Subdivision function α

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

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

$\alpha(S_0 .. S_i, n, k, f, X)$ = true / false
means p_k is allowed / forbidden
to apply f to X

Two requirements :

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

α monotonic : (w.r.t. i)

$\bot \; \bot \; \bot$ false false false ...
$\bot \; \bot \; \bot$ true true true ...
$\bot \; \bot \; \bot \; \bot \; \bot \; \bot$ ....

Parallel-search plans

Search plan for multi-search:

multi-plan

a collection of search plans with communication one per process

$$\Sigma = < \Sigma_1 \ldots \Sigma_n >$$

Search plan for distributed search:

distributed-search plan

search plan with communication
+
subdivision function

$$\Sigma = < \zeta, \xi, \alpha, \omega >$$

Parallelizations

Search plan with communication

$\Sigma' = < \zeta', \xi', \omega >$ corresponds to

sequential search plan

$\Sigma = < \zeta, \xi, \omega >$ if

ζ' and ξ' select inferences like ζ and ξ (difference is communication).

A multi-plan $\Sigma' = < \Sigma'_0 \ldots \Sigma'_n >$ is a parallelization by combination of sequential search plans $\Sigma_0 \ldots \Sigma_n$ if Σ'_i corresponds to Σ_i, $0 \leq i \leq n$.

A distributed-search plan

$\Sigma' = < \zeta', \xi', \alpha, \omega >$

is a parallelization by subdivision of sequential search plan $\Sigma = < \zeta, \xi, \omega >$ if $< \zeta', \xi', \omega >$ corresponds to Σ (difference is also made by α).

Parallel strategies

in the sense of parallel search

Homogeneous parallel strategy $\langle I, M, \Sigma \rangle$

multi-search strategy if Σ is multi-plan

distributed strategy if Σ is distributed-search plan

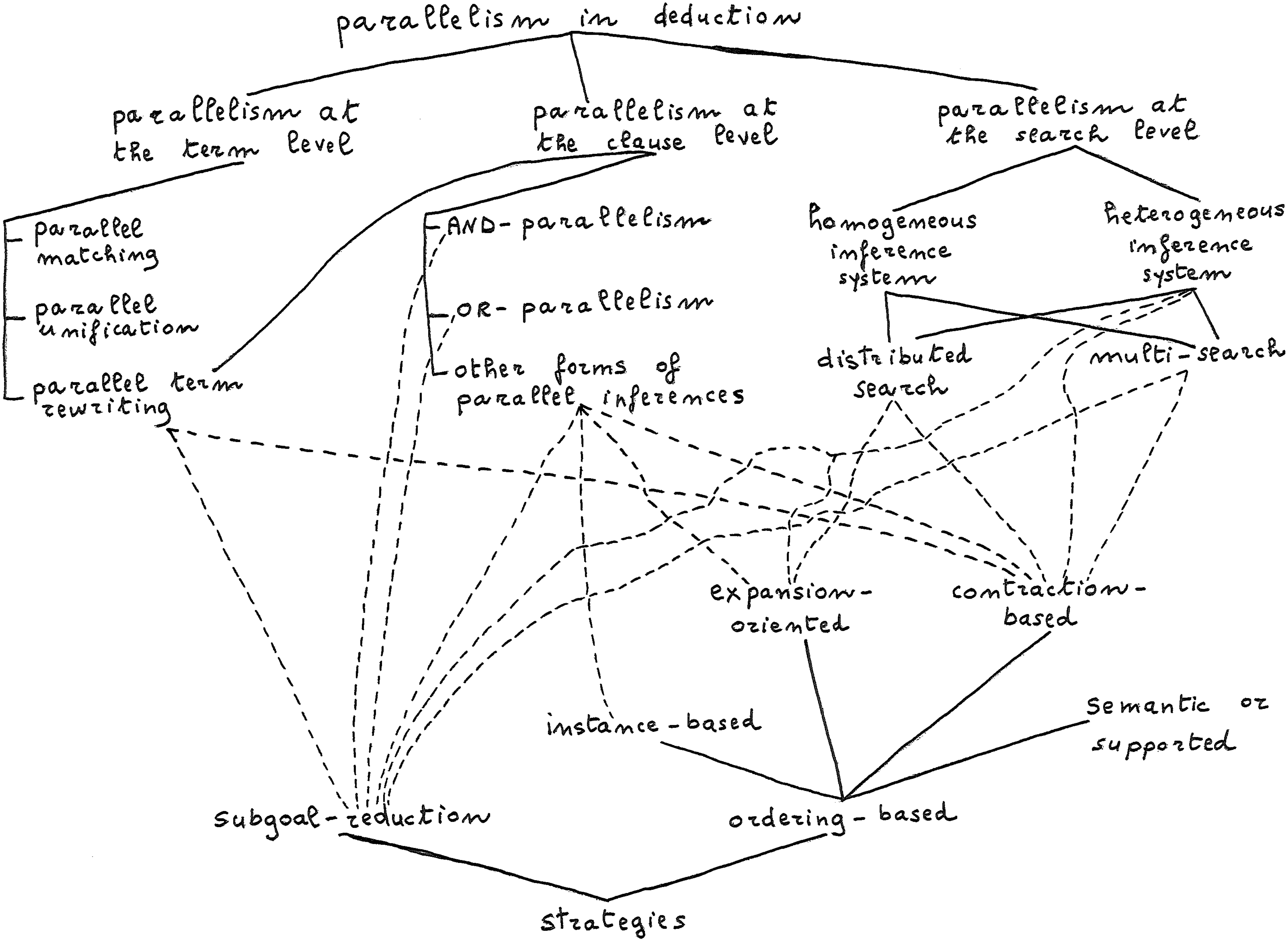
Notions of parallelization by combination/ subdivision extend to strategies assuming same I.

Heterogeneous parallel strategy

$$\langle I_0 \ldots I_{n-1}, \Sigma, M \rangle$$

with same options for Σ.

General notion of parallel derivation which can be specialized to the various cases


parallelism in deduction
parallelism at the term level
parallelism at the clause level
parallelism at the search level
parallel matching
parallel unification
parallel term rewriting
AND-parallelism
OR-parallelism
other forms of parallel inferences
homogeneous inference system
heterogeneous inference system
distributed search
multi-search
expansion-oriented
contraction-based
instance-based
semantic or supported
subgoal-reduction
ordering-based
strategies

First analysis

Size of data base

Dynamics of data base

Pre-processing

Read-only data

Special data structures

Conflicts: backward contraction
"backward-contraction bottleneck"

Parallelism at the search level most suitable for theorem proving especially contraction-based

[Bonacina & Hsiang; "Parallelization of deduction strategies: an analytical study" Journal of Automated Reasoning Vol. 13 1994]

Analysis of control of search

Parallelism at the term level:

- Concurrency replaces search plan:

  loss of control ?

- "Strategy-compliant":

  excess of control ?

Parallelism at the clause level:

- Search plan as scheduler:

  tasks too small ?
  (e.g., given-clause, subgoal)

Evolution 1992 - 1999

Fine-grain → Coarse-grain
Medium-grain → Coarse-grain

Master-slaves → Peers

Examples:

PaReDuX → PaReDuX + TeamWork

PARTHEO → CPTHEO
HPDS → CPTHEO

Also:

Clause-Diffusion → Modified Clause-Diffusion

TeamWork with master-slaves → TeamWork with peers

Parallel search

- Cost of communication ?

+ ignore "redundant" data
- delay ⇒ do redundant steps

- Overlapping searches ?

Distributed search: dynamic heuristic subdivision

Multi-search : too similar plans
all fair, all exhaustive
scarcity of search plans

+ Early difference may suffice

- Scalability ?

Distributed search : subdivision depends on # of processes also qualitatively

Multi-search / Heterogeneous : extra plan or rules may not help

Directions for further research

Design

Parallel semantic strategies?

Distributed search for
subgoal-reduction strategies?

Analysis

Subdivision vs. overlap
Eager contraction vs. communication
in
Distributed-search contraction-based
strategies
by applying
Bounded search spaces methodology
[Bonacina, Strategies '98, JELIA '98]

More formal analyses?