# On the representation of parallel search in theorem proving

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

OCTOBER 1997

# Motivation

Parallel / Distributed theorem proving
(e.g. Clause-Diffusion
Team-Work
Parallel Setheo
...)

Implementation + empirical evaluation

Not satisfactory

Machine-independent evaluation ?

# Strategy analysis is new

Search in AI: design of heuristics

Complexity theory:

* complexity of propositional proofs
  proof system / proof length
  $NP \neq co\text{-}NP$
  [Tseitin 70, Cook-Reckhow 79, Urquhart 95]
* Herbrand theorem
  [Statman 79, Orevkov 82, Goubault 94]
* lower bounds

Strategy analysis: upper bounds

* propositional Horn logic [Plaisted 94]
* beyond ?

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.

# Previous work

Approach to modelling search and measuring search complexity:

Representation of search space with contraction and search process ✓

Turning infinite domains into finite ✓

Complexity measure:
bounded search spaces ✓

Application to compare strategies with different contraction power ✓

# Background

**Contraction-based strategies**

Forward reasoning

Contraction rules

Search plans (eager contraction)

**Coarse-grain parallelism:**

parallel search

- subdivide search space
- use different search plans
- . . .

# Contributions of this work

Extend the bounded search spaces approach to parallel search:

Definitions for parallel theorem proving ✓

Model search space with multiple search processes, subdivision, communication ✓

Complexity measures ✓

Applications to comparison of strategies

# Parallel theorem proving strategy

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

## I: inference system

$$f^n : \mathcal{L}^n \longrightarrow \mathcal{P}(\mathcal{L}) \times \mathcal{P}(\mathcal{L})$$

$$f(\varphi_1 \dots \varphi_n) = (\psi_1, \psi_2)$$

$\psi_1$ added

$\psi_2$ deleted

$$\psi_1 < \psi_2$$

e.g. simplification $(P(f^2\emptyset), fx \to x) =$

$$(P\emptyset, P(f^2\emptyset))$$

Given $S$ and $I$: $\quad S_I^*$

## M: communication operators

send, receive

broadcast, multicast

scatter, gather ...

Basic:

send : $\mathcal{L}^* \longrightarrow \mathcal{P}(\mathcal{L}) \times \mathcal{P}(\mathcal{L})$

$$send(\bar{x}) = (\phi, \phi)$$

receive : $\mathcal{L}^* \longrightarrow \mathcal{P}(\mathcal{L}) \times \mathcal{P}(\mathcal{L})$

$$receive(\bar{x}) = (\bar{x}, \phi)$$

# $\Sigma$: parallel search plan

sequential: select rule
select premises

parallel: also communication
subdivision

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

$\zeta : States^* \times \mathbb{N} \times \mathbb{N} \longrightarrow I \cup M$
(selects next rule or communication operator)

$\xi : States^* \times \mathbb{N} \times \mathbb{N} \times I \cup M \longrightarrow \mathcal{L}^*$
(selects next premises)

$\omega : States \longrightarrow Bool$
(detects termination)

# Subdivision function

Subdivide inferences in search space among $P_0 \cdots P_{n-1}$

Search space: infinite, unknown

Dynamic subdivision:

at each stage $S_i$ of derivation subdivide inferences in $S_i$

$P_k$ : allowed / forbidden

$$\alpha : \mathrm{States}^* \times \mathbb{N} \times \mathbb{N} \times (I \cup M) \times \mathcal{L}^* \longrightarrow \mathrm{Bool}$$

(partial function)

# Parallel derivation

$$\varphi = < I ; M ; \Sigma >$$

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

$$P_0 \ldots P_{n-1}$$

$$S = S_0^0 \vdash S_1^0 \vdash \ldots \vdash S_i^0 \vdash \ldots$$

$$\vdots$$

$$S = S_0^k \vdash S_1^k \vdash \ldots \vdash S_i^k \vdash \ldots$$

$$\vdots$$

$$S = S_0^{n-1} \vdash S_1^{n-1} \vdash \ldots \vdash S_i^{n-1} \vdash \ldots$$

$\forall k \qquad \forall i \geq 0 \qquad$ if

$$\omega(S_i^k) = \mathit{false}$$

$$\zeta(S_0^k \ldots S_i^k, n, k) = f$$

$$\xi(S_0^k \ldots S_i^k, n, k, f) = \bar{x}$$

$$\alpha(S_0^k \ldots S_i^k, n, k, f, \bar{x}) = \mathit{true} \quad \text{(allowed)}$$

$$S_{i+1}^k = S_i^k \cup \pi_1(f(\bar{x})) - \pi_2(f(\bar{x}))$$

# Properties of the subdivision function

## Total on generated clauses:

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

$\bar{x} \in \bigcup_i S_i$

$\alpha(S_0 \dots S_i, n, \kappa, f, \bar{x}) \neq \perp$

## Monotonic:

does not change indefinitely the status (allowed / forbidden) of a step:

if $\alpha(S_0 \dots S_i, n, \kappa, f, \bar{x}) \neq \perp$ then

$\forall j > i \quad \alpha(S_0 \dots S_j, n, \kappa, f, \bar{x}) \neq \perp$ and

$$\exists j \geq i \quad \forall r > j \quad \alpha(S_0 \dots S_r, n, \kappa, f, \bar{x}) = \alpha(S_0 \dots S_j, n, \kappa, f, \bar{x}).$$

# Fairness

Refutational completeness of $\mathcal{I}$ +

Fairness of $\Sigma$ =

Completeness of $\mathcal{C}$

Uniform fairness:

$$S_0 \vdash S_1 \vdash \ldots S_i \vdash$$

$\mathcal{I}$: inference rules

$R$: redundancy criterion

$R(S)$ : clauses redundant in $S$

$$\mathcal{I}(S_\infty - R(S_\infty)) \subseteq \bigcup_j S_j$$

where

$$S_\infty = \bigcup_j \bigcap_{i \geq j} S_i \qquad \text{(persistent clauses)}$$

# Fairness for parallel derivation

for all $\bar{x} \in S_\infty - R(S_\infty)$ $\quad f(\bar{x}) \neq (\emptyset, \emptyset)$

there is a $P_k$

a) $\bar{x}$ together in memory of $P_k$
(fairness of communication)
$\bar{x} \in S_\infty^k - R(S_\infty)$

b) $f(\bar{x})$ allowed
(fairness of subdivision function)
$\exists i \ \forall j \geq i \quad \alpha(S_1^k .. S_j^k, n, k, f, \bar{x}) = true$

c) sequential search plan fair
(local fairness)

Theorem: $a + b + c \Rightarrow$ uniform fairness

Concrete fairness:

stronger requirement because it is not known what is persistent

Propagation of redundancy:

$\varphi \in R(S_i^k)$ for some $P_k$

stage $i$

then $\forall P_h$ $\exists j$

$\varphi \in R(S_j^h)$.

Requirement on communication

# Parallelization by subdivision

$\mathcal{C}' = \langle I; M; \Sigma' \rangle$ $\qquad\qquad$ $\mathcal{C} = \langle I; \Sigma \rangle$

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

$\Sigma'$ selects inferences like $\Sigma$:

if $\zeta'(\bar{S}, m, k) = f$ inference rule

$$\zeta'(\bar{S}, m, k) = \zeta(\bar{S})$$

$$\xi'(\bar{S}, m, k, f) = \xi(\bar{S}, f)$$

Subdivision function $\Rightarrow$ different behavior

forbidden steps $\Rightarrow$ different selections

subdivision $\Rightarrow$ communication

$$\Downarrow$$

Different derivations

## Representation of search space

Multiple processes active in the search space of the problem.

Infinite search space.

Dynamic search space:

Contraction

Subdivision

Communication

All intertwined with selection by search plan.

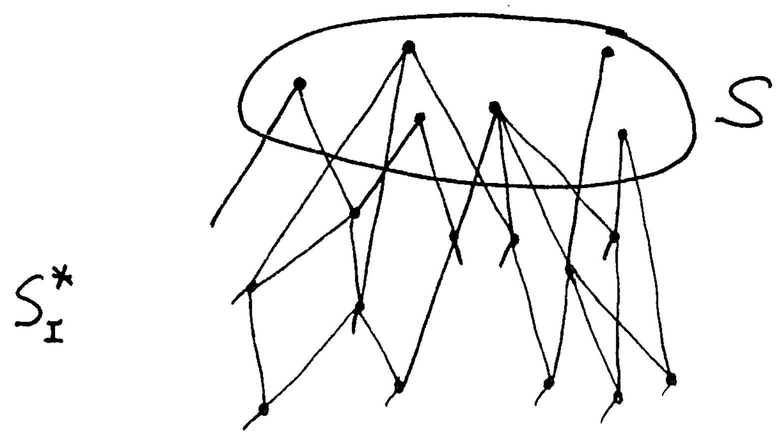
Solution: marked search graph

# Representation of search space

Closure $S_I^*$

$I^0(S) = S \qquad I^k(S) = I(I^{k-1}(S))$

$S_I^* = \bigcup I^k(S)$




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

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

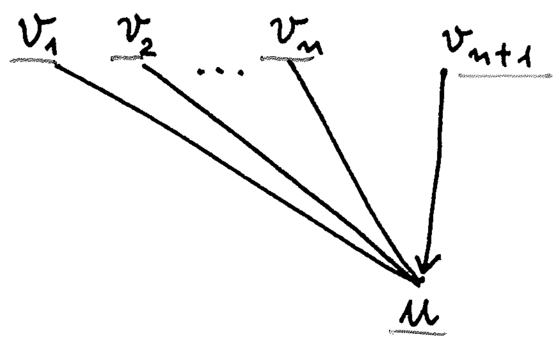
($\doteq$ : equivalence of variants)

# Representation of search space

$$G(S^*_I) = (V, E, \ell, h)$$

$E$: hyperarcs

if $\quad f(\varphi_1 \dots \varphi_n, \varphi) = (\psi, \varphi) \quad$ then




$e \in E$

where
$$h(e) = f$$
$$\ell(v_i) = \varphi_i \qquad i \leq n$$
$$\ell(v_{n+1}) = \varphi$$
$$\ell(u) = \psi$$

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

## Example:

$$
\begin{tikzcd}
M(a,a,b) \arrow[d, no head] & \neg M(x,y,z) \vee M(z,y,x\cdot y) \arrow[dl, no head] \\
M(b,a,a\cdot a) \arrow[d] & x\cdot x \to x \arrow[dl, no head] \\
M(b,a,a) \arrow[d] & M(x,y,y) \arrow[dl, no head] \\
T &
\end{tikzcd}
$$

# Marked search graph

$G = \langle V, E, \ell, h, \bar{s}, \bar{c} \rangle$

Marking $\bar{s}$ of vertices:

$s^k : V \longrightarrow \mathbb{Z}$

$$s^k(v) = \begin{cases} m > 0 & m \text{ variants at } p_k \\ -1 & \text{all deleted by } p_k \\ 0 & \text{otherwise} \end{cases}$$

$s^k(v)$: # of variants of clause

Represents:

contraction

communication

selection by search plan

# Marked search graph

$G = \langle V, E, \ell, h, \bar{s}, \bar{c} \rangle$

Marking $\bar{c}$ of arcs:

$c^k : E \to \mathbb{N} \times Bool$

$\pi_1(c^k(e))$ = # of times arc $e$ executed by $p_k$

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

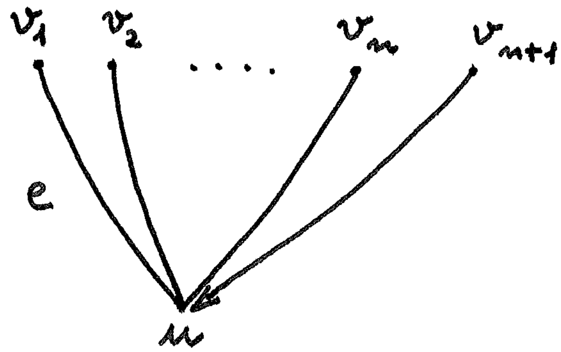
Represents:

subdivision

selection by search plan

# Evolution of search space

## Pre-conditions of a step (e enabled at $p_K$)



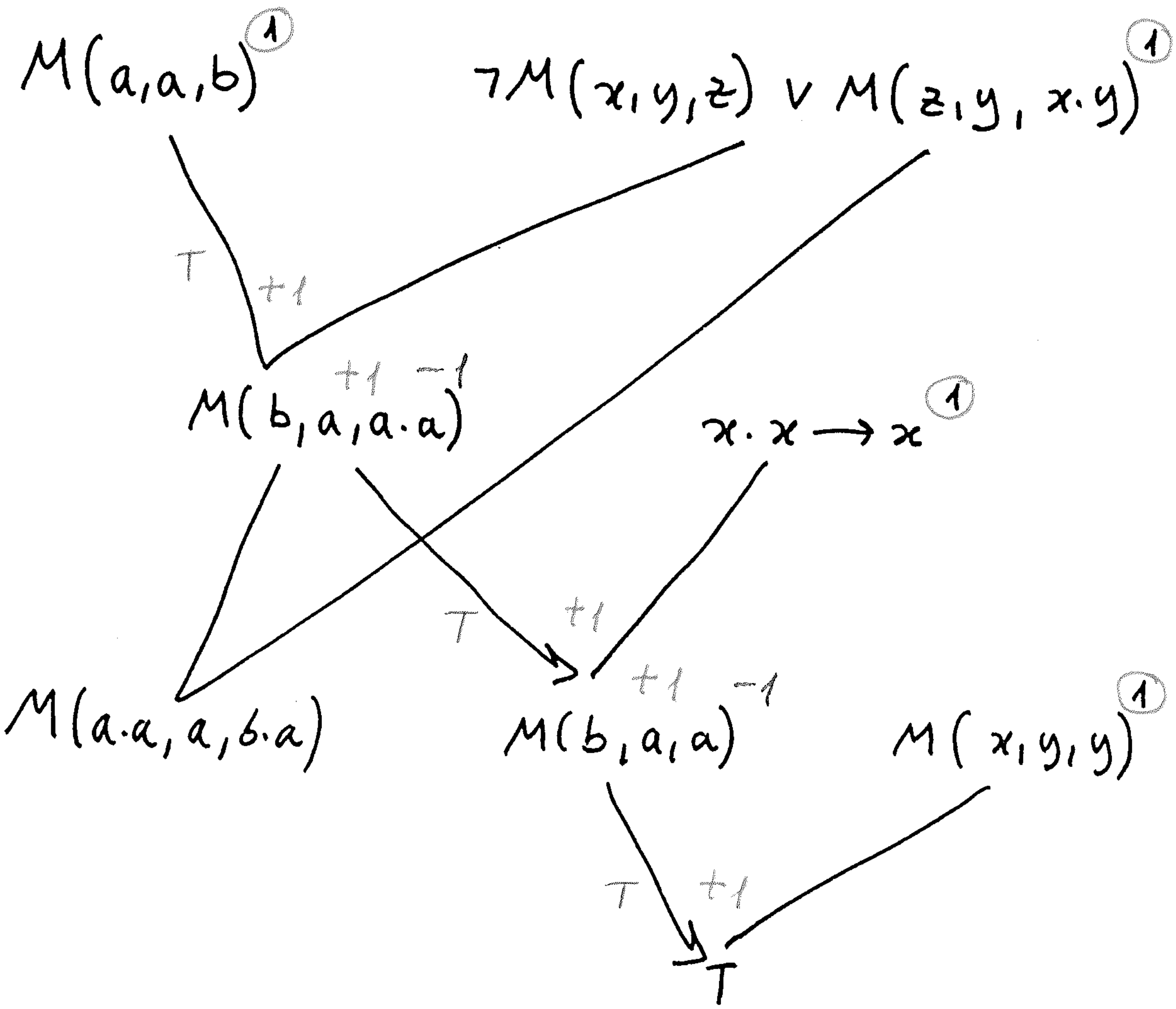

1) all premises present ( $S^K(v_i) \geq 1$ )
2) arc allowed ( $\Pi_2(c^K(e)) = true$ )

## Post-conditions of a step

1) decrement marking of deleted clause (-1 if last variant)

2) increment marking of generated or received clause

## Example:



M(a,a,b) 1
¬M(x,y,z) ∨ M(z,y,x·y) 1
T
+1
M(b,a,a·a)
+1
−1
x·x → x 1
T
+1
M(a·a,a,b·a)
M(b,a,a)
+1
−1
M(x,y,y) 1
T
+1
T

# Evolution of search space

$$S_0^k(v) = \begin{cases} 1 & \text{if input clause} \\ 0 & \text{otherwise} \end{cases}$$

$S_{i+1}^k(v)$ decremented by contraction
incremented by expansion
communication

$$\Pi_1(c_0^k(e)) = 0$$

$\Pi_1(c_{i+1}^k(e))$ incremented if executed

$$\Pi_2(c_{i+1}^k(e)) = \begin{cases} \alpha(S_0 \ldots S_{i+1}, n, k, f, \bar{x}) & \text{if} \neq \perp \\ \text{true} & \text{otherwise} \end{cases}$$

<u>As the derivation progresses, the strategy dynamically forbids arcs.</u>

# Discussion

Marked search graph:

good to represent search space

made dynamic by

contraction

subdivision

communication

Advantages: graph does not change

markings change

all processes on one graph

# Measure of search complexity for parallel search

Advantage of parallelization:

subdivision of search space

Overhead of parallelization:

duplication
communication

Infinite search space:

cannot compare total size of search spaces

# Proposed approach

Extend the bounded search spaces approach used for contraction

Introduce a new notion of overlap to capture the overhead of parallelism:

duplication

dependence (communication)

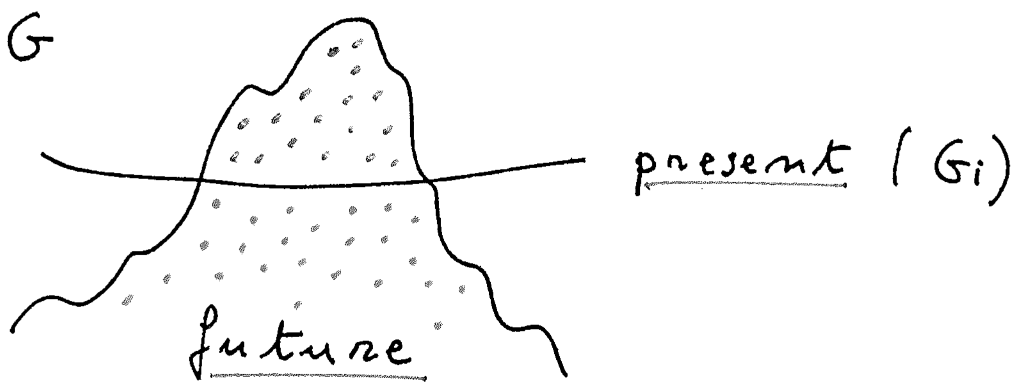
Parallel bounded search spaces reflect both subdivision and overlap

Applications to comparison of strategies

# Complexity measures

$(A, <)$ : well-founded ordering

Idea:




or:

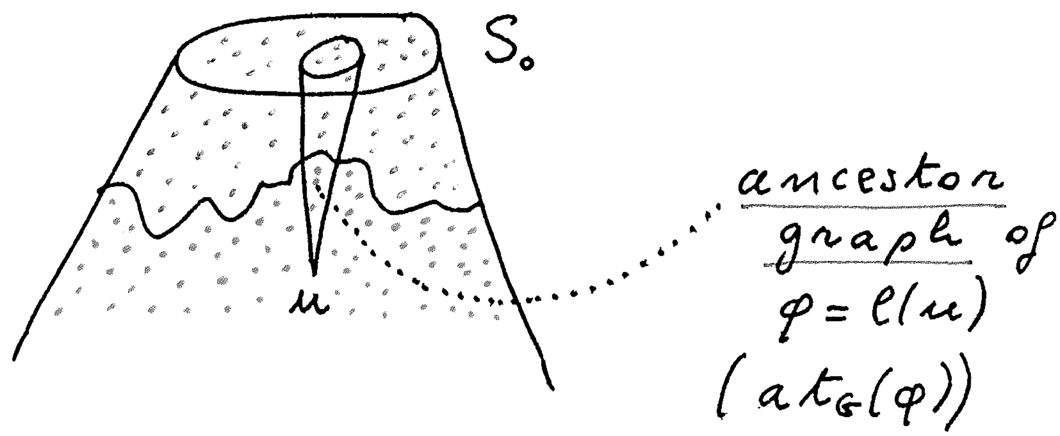
$S_0 \vdash S_1 \vdash \ldots S_i \vdash$ future

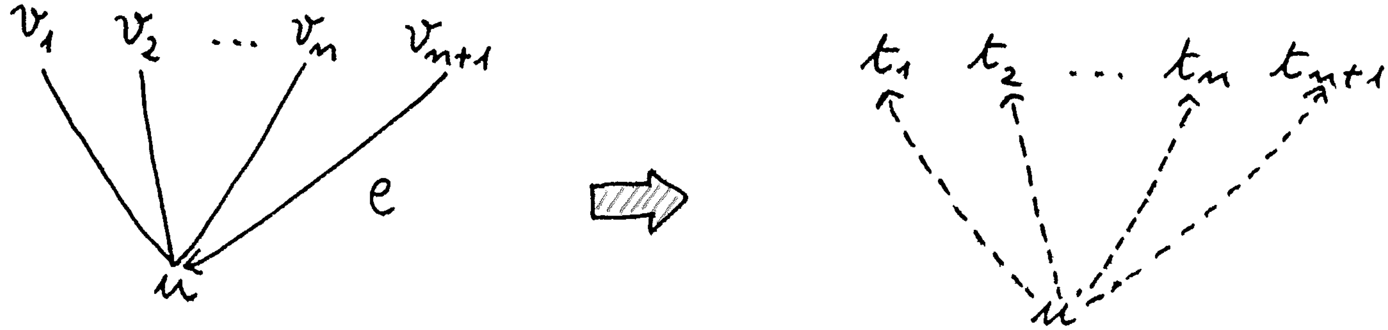
present

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

# Looking back: ancestor-graph

$G$:




- ancestor-graph of $u$ is

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

where $t_i$ is ancestor graph of $v_i$

- $w \in t$ is <u>relevant</u> to $u$ in $t$ if
  - $w \in \{v_1 \dots v_{n+1}\}$ and $\pi_1(c(e)) = 0$ or
  - $w$ is relevant to $v_i$ in $t_i$ for some $i$

# A metric for search graph

- Past distance of $\varphi$ in $t$:

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

- Future distance of $\varphi$ in $t$:

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

- Global distance: $gdist_G(t) = pdist_G(t) + fdist_G(t)$

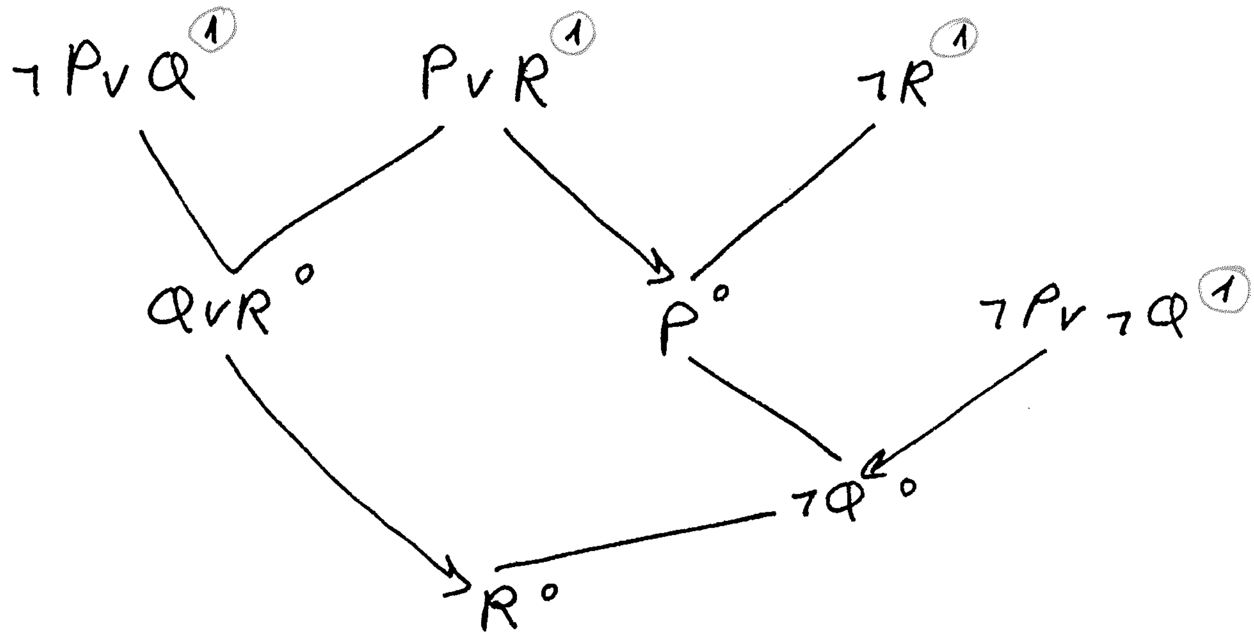
Dynamic distance:

- $fdist_G(t)$ measures the portion of $t$ that needs to be traversed to reach $\varphi$
- if $\infty$, then unreachable! (redundant)
- alternative definitions:

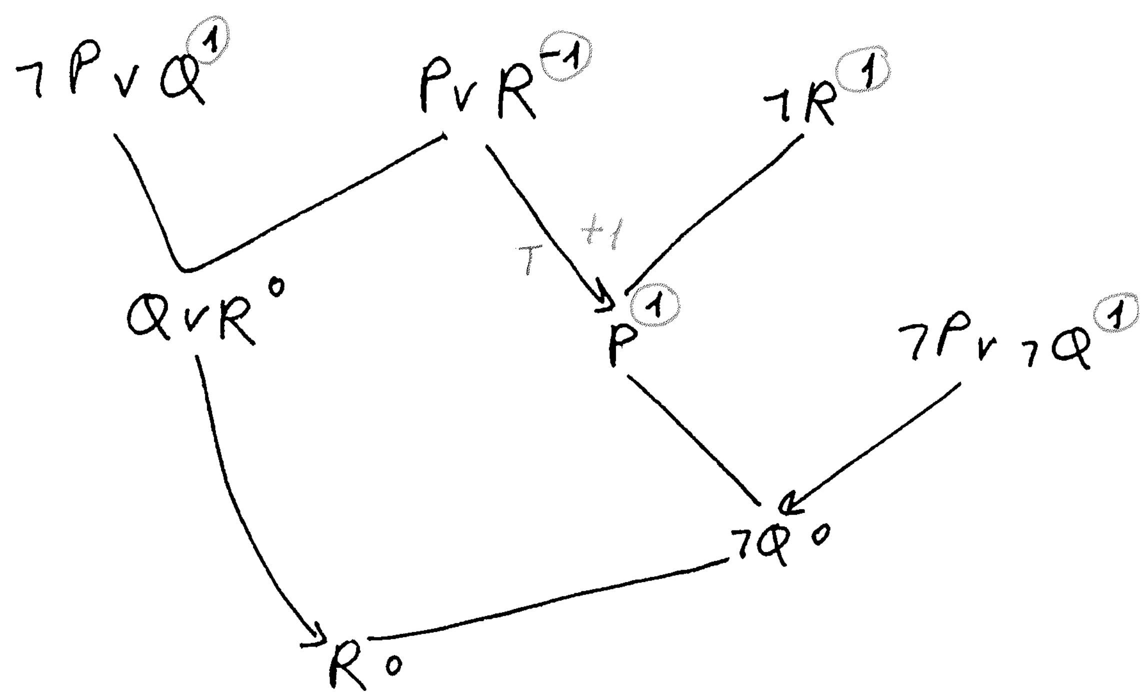
  use multisets instead of cardinalities

## Example:

$S_i$




$S_{i+1}$



## Continuing the example:

as $S_i \vdash S_{i+1}$

- $fdist_{G_i}(\neg Q) = 2 \implies fdist_{G_{i+1}}(\neg Q) = 1$

  $gdist_{G_i}(\neg Q) = gdist_{G_{i+1}}(\neg Q) = 5$

- $fdist_{G_i}(Q \vee R) = 1 \implies fdist_{G_{i+1}}(Q \vee R) = \infty$ !

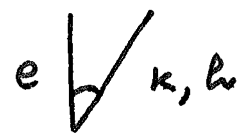
  $gdist_{G_i}(Q \vee R) = 3 \implies gdist_{G_{i+1}}(Q \vee R) = \infty$

- $fdist_{G_i}(R) = 4 \implies fdist_{G_{i+1}}(R) = \infty$ !

  $gdist_{G_i}(R) = 8 \implies gdist_{G_{i+1}}(R) = \infty$

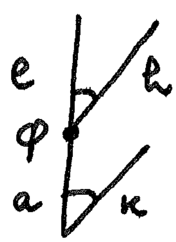
# Overlap

## Duplication:




$$\pi_2(c_k(e)) = \pi_2(c_l(e)) = true$$

## Dependence:




$$e <_G a$$

$\pi_2(c_k(e)) = false$ $\qquad$ $\pi_2(c_k(a)) = true$

$\pi_2(c_l(e)) = true$ $\qquad$ $\pi_2(c_l(a)) = false$

## Cost of communication:

forbidding e does not exclude $p_k$ from this path because if it receives $\varphi$ it can continue since a is allowed.

# Overlap

Ancestor-graph $t$ allowed for $p_k$ iff
$\forall e$ in $t$ forbidden for $p_k$
$\exists a >_G e$ in $t$ allowed for $p_k$.

Process $p_h$ overlaps with process $p_k$ on ancestor-graph $t$ if
$t$ is allowed for $p_k$ and
there exists a subgraph $t'$ of $t$ allowed for $p_h$.

$t = t'$ : duplication

proper subgraph: dependence

# Bounded search spaces

Slice the infinite graph in a sequence of finite layers.

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

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

where

$$mul_G(v,k,j) = |\{ t \mid t \in at_G(v),\ t \text{ allowed for } p_k,\ 0 < gdist_G(t) \leq j \}|$$

# Parallel bounded search spaces

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

where

$$\mathrm{pmul}_G(v, j) = \lceil \mathrm{gmul}_G(v, j) / w \rceil$$

$w$ → # of processes

$$\mathrm{gmul}_G(v, j) = \sum_{k} \mathrm{mul}_G(v, k, j).$$

## Summary

Infinite distance: captures contraction

Forbidden ancestor-graphs:

capture subdivision

Allowed ancestor-graphs:

capture overlap

(duplication,

communication)

# Comparison of strategies

Expansion inferences do not change the bounded search spaces.

Contraction inferences reduce the bounded search spaces (w.r.t. multiset ordering).

Subdivision reduces the bounded search spaces.

Overlap counter the effect of subdivision.

---

Compare $\mathcal{C}$ and $\mathcal{C}'$

Compare different $\mathcal{C}'$ of given $\mathcal{C}$.