# Parallel logical inference search in Algebraic Programming System (APS)

J.Kapitonova, A.Letichevski, V.German, M.Morokhovets

(Glushkov Institute of Cybernetics)

# OBJECTIVES:

- **to develop parallel prover as an application of APS**
- **to investigate the implemented version of that prover**

# Logical calculus 1

**A prime sequent:**

**$x \Rightarrow y$, where x,y are propositional formulas.**

**A conditional sequent:**

**(w,Q), where w is conjunction of literals, Q is conjunction of prime sequents.**

**An auxiliary goal: aux($v,u \Rightarrow z,P$), where z is a literal, u,v are propositional formulas, P is conjunction of prime sequents**

**Axioms:**

**(w, 1);**

**($w, u \Rightarrow 1$);**

**($w, 0 \Rightarrow P$);**

**(0, Q),**

**where w is conjunction of literals, P is conjunction of prime sequents, 1 denotes empty conjunction.**

# Logical calculus
2

**Rules of Inference:**

**(R1)** $(w, u \Rightarrow 0) \mid\!- (w, 1 \Rightarrow \neg u)$,

**(R2)** $(w, u \Rightarrow x \wedge y) \mid\!- (w, (u \Rightarrow x) \wedge (u \Rightarrow y))$,

**(R3)** $(w, u \Rightarrow x \vee y) \mid\!- (w, \neg x \wedge u \Rightarrow y)$,

**(R4)** $(w, x \wedge y \Rightarrow z \mid\!- (w \wedge x, y \Rightarrow z)$, **x is a literal,**

**(R5)**
$$\frac{(w,F) \mid\!- (w',F')}{(w,F \wedge H) \mid\!- (w,H)}$$

**where (w′, F′) is an axiom,**

**(R6)**
$$\frac{aux(1, w \wedge u \Rightarrow z, 1) \mid\!- aux(v, z \wedge y \Rightarrow z, P)}{(w, u \Rightarrow z) \mid\!- (w \wedge \neg z, P)}$$

**where z is a literal, P is conjunction of prime sequents.**

# Logical calculus
3

**(RA1)** $aux(v, x \wedge y \Rightarrow z, P) \vdash aux(v \wedge y, x \Rightarrow z, P)$,

**(RA2)** $aux(v, x \vee y \Rightarrow z, P) \vdash aux(v, x \Rightarrow z, (v \Rightarrow \neg y) \wedge P)$.

**Formula P is a tautology IFF $(1, 1 \Rightarrow P) \vdash Q$, where Q is an axiom.**

# EXAMPLE

**Is $(a2 \wedge a1 \vee \neg a2 \wedge a1 \vee \neg a1) \wedge (a1 \vee \neg a1)$ a tautology?**

**$(a2 \wedge a1 \vee \neg a2 \wedge a1 \vee \neg a1) \wedge (a1 \vee \neg a1)$ is a tautology IFF**
**$(1, 1 \Rightarrow (a2 \wedge a1 \vee \neg a2 \wedge a1 \vee \neg a1) \wedge (a1 \vee \neg a1)) \mid\!- Q$, where Q is an axiom.**

**/ Proof /**

**$(1, 1 \Rightarrow (a2 \wedge a1 \vee \neg a2 \wedge a1 \vee \neg a1) \wedge (a1 \vee \neg a1))$ (initial sequent)**

**$(1, (1 \Rightarrow (a2 \wedge a1 \vee \neg a2 \wedge a1 \vee \neg a1)) \wedge (1 \Rightarrow (a1 \vee \neg a1)))$ (R2)**

**/ Now R5 is applicable: if it is possible to prove**

**$(1, 1 \Rightarrow (a2 \wedge a1 \vee \neg a2 \wedge a1 \vee \neg a1)) \mid\!- (w', F')$, where $(w', F')$ is an axiom, then there is only to prove that**

**$(1, 1 \Rightarrow (a1 \vee \neg a1))$. (*) /**

**$(1, 1 \Rightarrow (a2 \wedge a1 \vee \neg a2 \wedge a1 \vee \neg a1))$** **(R5, begin)**

**$(1, \neg(a2 \wedge a1) \Rightarrow \neg a2 \wedge a1 \vee \neg a1)$** **(for $x=a2 \wedge a1$, $y=\neg a2 \wedge a1 \vee \neg a1$ R3)**

**$(1, \neg a2 \vee \neg a1 \Rightarrow \neg a2 \wedge a1 \vee \neg a1)$** **(substitute: $\neg(a2 \wedge a1) \rightarrow \neg a2 \vee \neg a1$)**

**$(1, (\neg a2 \vee \neg a1) \wedge (a2 \vee \neg a1) \Rightarrow \neg a1)$** **(R3)**

**$aux(1, (\neg a2 \vee \neg a1) \wedge (a2 \vee \neg a1) \Rightarrow \neg a1, 1)$** **(R6, begin)**

**$aux((\neg a2 \vee \neg a1), a2 \vee \neg a1 \Rightarrow \neg a1, 1)$** **(RA1)**

**$aux((\neg a2 \vee \neg a1), \neg a1 \Rightarrow \neg a1, \neg a2 \vee \neg a1 \Rightarrow \neg a2)$** **(RA2)**

**$(a1, \neg a2 \vee \neg a1 \Rightarrow \neg a2)$** **(R6, end)**

**$aux(1, a1 \wedge (\neg a2 \vee \neg a1) \Rightarrow \neg a2, 1)$** **(R6, begin)**

**$aux(a1, \neg a2 \vee \neg a1 \Rightarrow \neg a2, 1)$** **(RA1)**

**$aux(a1, \neg a2 \Rightarrow \neg a2, a1 \Rightarrow a1)$** **(RA2)**

**$(a1 \wedge a2, a1 \Rightarrow a1)$** **(R6, end)**

aux(1,a1$\wedge$a2$\wedge$a1 $\Rightarrow$ a1,1) (R6, begin)

aux(1,a1$\wedge$a2 $\Rightarrow$ a1,1) (substitute: a1$\wedge$a2$\wedge$a1 $\rightarrow$ a1$\wedge$a2)

aux( a2, a1 $\Rightarrow$ a1,1) (RA1)

(a1$\wedge$a2$\wedge\neg$a1,1) (R6, end; axiom)

/ Prove (1, 1 $\Rightarrow$ (a1 $\vee$ $\neg$a1)) /

(1, (1 $\Rightarrow$ (a1 $\vee$ $\neg$a1)) (sequent (*))

(1, a1 $\Rightarrow$ a1) (R3)

aux(1,a1 $\Rightarrow$ a1,1) (R6, begin)

($\neg$a1,1) (R6, end; R5, end; axiom)

/ Proved /

# PARALLELIZATION IN THEOREM-PROVING

- AND-PARALLELISM
- OR-PARALLELISM

- COOPERATIVE THEOREM PROVING
- COMPETITIVE THEOREM PROVING

## COMPUTATIONAL ENVIRONMENT

**SUPERCOMPUTER FOR INFORMATION TECHNOLOGIES (SCIT):**

**cluster-type computer system**

**32 processors + Network File System + Message Passing Interface**

# TYPES OF INPUT FORMULAS

- F1&…&Fn
- F1&…&Fn $\rightarrow$ F

# EXPERIMENTS

**Series 1: Exp 1 = ( a_7 & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 ∨
~(a_7) & a_6 & a_5 & a_4 & a_3 & a_2 & a_1 ∨
~(a_6) & a_5 & a_4 & a_3 & a_2 & a_1 ∨ ~(a_5) & a_4 & a_3 & a_2 & a_1 ∨
~(a_4) & a_3 & a_2 & a_1 ∨ ~(a_3) & a_2 & a_1 ∨ ~(a_2) & a_1 ∨ ~(a_1)) &
( a_6 & a_5 & a_4 & a_3 & a_2 & a_1 ∨ ~(a_6) & a_5 & a_4 & a_3 & a_2 & a_1
∨ ~(a_5) & a_4 & a_3 & a_2 & a_1 ∨ ~(a_4) & a_3 & a_2 & a_1 ∨ ~(a_3) & a_2
& a_1 ∨ ~(a_2) & a_1 ∨ ~(a_1)) &
( a_5 & a_4 & a_3 & a_2 & a_1 ∨ ~(a_5) & a_4 & a_3 & a_2 & a_1 ∨ ~(a_4) &
a_3 & a_2 & a_1 ∨ ~(a_3) & a_2 & a_1 ∨ ~(a_2) & a_1 ∨ ~(a_1)) &
( a_4 & a_3 & a_2 & a_1 ∨ ~(a_4) & a_3 & a_2 & a_1 ∨ ~(a_3) & a_2 & a_1 ∨
~(a_2) & a_1 ∨ ~(a_1)) &
( a_3 & a_2 & a_1 ∨ ~(a_3) & a_2 & a_1 ∨ ~(a_2) & a_1 ∨ ~(a_1)) &
( a_2 & a_1 ∨ ~(a_2) & a_1 ∨ ~(a_1)) & ( a_1 ∨ ~(a_1)).**

**Exp i = (a_f(i) & a_(f(i)-1) & … & a_1 ∨ ~(a_f(i)) & a_(f(i)-1) & … & a_1 ∨ … ∨ ~(a_2) & a_1 ∨ ~(a_1)) & (a_(f(i)-1) & … & a_1 ∨ ~(a_(f(i)-1)) &… & a_1 ∨ … ∨ ~(a_2) & a_1 ∨ ~(a_1)) & Exp i-1.**

**f(i) = 7+2(i-1), i is positive integer, 1≤ i≤ 6, f(7)=18.**

**Exp i = Exp i.j w.r.t. commutativity and associativity**

**Series 2.**

**A tree T1 is represented by propositional formula:**

**$T1 = (\sim(a\_0) \vee a\_1) \& (\sim(a\_0) \vee a\_2) \& (\sim(a\_0) \vee a\_3) \& (\sim(a\_0) \vee a\_4) \& (\sim(a\_0) \vee a\_5) \& (\sim(a\_1) \vee a\_6) \& (\sim(a\_2) \vee a\_7) \& (\sim(a\_3) \vee a\_8) \& (\sim(a\_4) \vee a\_9) \& (\sim(a\_5) \vee a\_10)$.**

**A path between nodes v and u exists, when $T1 \rightarrow (u \rightarrow v)$ is a tautology. A tree Ti ($1 \leq i \leq 10$) has $10+5(i-1)$ nodes.**

**Graph Gri ($1 \leq i \leq 4$) has $10+5(i-1)$ nodes; it is built by re-directing a node (a_(5(i-1)+1),a_(5i+1)) in Ti.**

## Table1 Speed-up in series 1 (time in seconds)

| | Exp1 | Exp1.1 | Exp2 | Exp2.1 | Exp3 | Exp3.1 | Exp4 | Exp4.1 | Exp5 | Ep5.1 | Exp 6 | Exp6.1 | Exp6.2 | Exp 7 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Time for 2 processors | 7 | 9 | 15 | 15 | 31 | 34 | 65 | 90 | 141 | 142 | 294 | 296 | 293 | 2091 |
| Minimal time | 5 | 6 | 8 | 9 | 15 | 9 | 24 | 37 | 43 | 46 | 63 | 85 | 86 | 1711 |
| Number of Proc. | 4 | 3 | 4 | 4 | 3 | 3 | 5 | 4 | 5 | 6 | 8 | 10 | 5 | 8 |
| Speed-up | 1.8 | 1.5 | 1.9 | 1.7 | 6.2 | 3.8 | 2.7 | 2.4 | 3.3 | 3 | 4.7 | 3.5 | 3.4 | 1.2 |

**Table 2** Speed-up in series 2 (paths in trees)

| | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | T10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Time for 2 processors | 15 | 22 | 52 | 125 | 237 | 440 | 652 | 1054 | 1343 | 2234 |
| Min time | 8 | 13 | 18 | 25 | 36 | 42 | 60 | 77 | 132 | 132 |
| Number of proc. | 6 | 7 | 6 | 10 | 13 | 26 | 18 | 20 | 15 | 27 |
| Speed-up | 1.9 | 1.7 | 2.9 | 5.0 | 6.7 | 10.5 | 10.9 | 13.9 | 10.2 | 16.9 |

**Table 3** Speed-up in series 3 (paths in graphs)

| | Gr1 | Gr2 | Gr3 | Gr4 |
|---|---|---|---|---|
| Time for 2 processors | 19 | 28 | 64 | 146 |
| Min. time | 12 | 16 | 28 | 41 |
| Number of processors | 4 | 4 | 11 | 17 |
| Speed-up | 1.6 | 1.8 | 2.3 | 3.6 |