

題名	証明支援システム「CAP-LA」
目的	CAP-LAシステムは、定理の証明、数式の操作などの作業を補佐し、人間と協調して問題を解決していくためのコンピュータ・システムの実現をめざしている。
概要	<p>CAP-LAシステムは、SYC-WGの支援の下にICOTによって開発された。その中心となる機能は以下の通りである。</p> <p>①線型代数を中心とする定理証明の検査 本プロジェクトのため設計された証明記述言語(POL)で記述された証明の妥当性を検査し、誤りを発見する。</p> <p>②証明の記述の支援 専用のエディタ(構造エディタ)を用いて、入力、編集など証明の記述作業を支援する。</p>
特徴	<p>①証明の行間補完機能・・・等式がらみの証明の行間を、項書き換えの技術を用いて補完する機能。</p> <p>②証明中の論理のチェック機能・・・証明に用いられた論理の妥当性をチェックする機能。チェックをユーザと対話的に行うデバグ風の機能。チェック箇所の表示機能。</p> <p>③構造編集機能・・・構造エディタを用いて、ユーザによる証明入力や編集の作業を支援する機能。</p>
構成	<p>汎用構造エディタSEMACS (文法ガイド, 部分構文解析, 構造編集)</p> <p>TEX出力      自然言語風表示</p> <p>等式エディタ      チェック箇所表示      証明戦略展開</p> <p>等式検査系</p> <p>項書き換え系</p> <p>証明検査系</p> <p>証明木生成機能</p> <p>証明戦略展開機能</p> <p>証明木補完機能</p> <p>ルール生成機能</p> <p>証明知識ベース系</p>

PDLによる記述例

```

theorem
det_trans:
all A:squ_mat.
  det(A)=det(trans(A))
proof
let A:squ_mat be arbitrary;
row(A)=col(trans(A));
col(A)=row(trans(A));
det(A)=sigma p:perm<col(A)>.
  (sgn(p)*pi i:1..col(A).
   A[p(i), i])
=sigma p:perm<col(A)>.
  (sgn(inv(p))*pi i:1..col(A).
   A[(inv(p))(i), i])
using sigma_pi
=sigma p:perm<col(A)>.
  (sgn(p)*pi i:1..col(A).
   A[(inv(p))(i), p(i)])
using sigma_pi
=sigma p:perm<col(A)>.
  (sgn(p)*pi i:1..col(A).
   A[(inv(p))(p(i)), p(i)])
using sigma_pi
=sigma p:perm<col(A)>.
  (sgn(p)*pi i:1..col(A).
   A[i, p(i)])
using sigma_pi
=sigma p:perm<col(A)>.
  (sgn(p)*pi i:1..col(A).
   (trans(A))[p(i), i])
using sigma_pi
=det(trans(A))
end_proof
end_theorem

```

TEX 出力

THEOREM: det.trans:  
For all A Esquare matrix,  $\det(A) = \det({}^tA)$

PROOF:  
Now let A Esquare matrix be arbitrary.

$$\begin{aligned}
 \text{row}(A) &= \text{col}({}^tA) \\
 \text{col}(A) &= \text{row}({}^tA) \\
 \det(A) &= \sum_{p \in S_{\text{col}(A)}} \text{sgn}(p) \prod_{i=1}^{\text{col}(A)} A_{p(i),i} \\
 &= \sum_{p \in S_{\text{col}(A)}} \text{sgn}(p^{-1}) \prod_{i=1}^{\text{col}(A)} A_{p^{-1}(i),i} \\
 &\quad \text{using sigma\_pi} \\
 &= \sum_{p \in S_{\text{col}(A)}} \text{sgn}(p) \prod_{i=1}^{\text{col}(A)} A_{p^{-1}(i),p(i)} \\
 &\quad \text{using sigma\_pi} \\
 &= \sum_{p \in S_{\text{col}(A)}} \text{sgn}(p) \prod_{i=1}^{\text{col}(A)} A_{p^{-1}(p(i)),p(i)} \\
 &\quad \text{using sigma\_pi}
 \end{aligned}$$

## 等式エディタ

### ① 起動

```

CAP-LA_48 ==> Structure Mode...
Theorem
det_trans:
all A:squ_mat.
det(A)=det(trans(A))
since
let A:squ_mat be arbitrary:
det(A)=det(trans(A))
end_since
end_theorem
SEMACS (esp) [95,15] demoll sys>user:
Read: icps1180::>sys>user>semacs>t

EQUALITY EDITOR ==> RIGHT HAND SIDE
det(A)
det(trans(A))
SEMACS (esp) [57,6] #48/2# --Top-- #
    
```

### ② 等式の変形

```

EQUALITY EDITOR ==> LEFT HAND SIDE
sigma p:perm(col(A)).
((sign(p)*pi i:1..col(A).
(A[i, (p) (i)])))
det(trans(A))
SEMACS (esp) [57,6] #48/1# --Top-- #
this rule (y/n/a(bort))?y
replace (i_1): p
replace (i_2): i

RULE WINDOW
Rule Tag: def func det
Condition: (A:squ_mat)
    
```

### ③ 変形結果の表示

```

CAP-LA_48 ==> Structure Mode...
all A:squ_mat.
det(A)=det(trans(A))
since
let A:squ_mat be arbitrary:
det(A)=sigma p:perm(col(A)).
((sign(p)*pi i:1..col(A).
(A[i, (p) (i)])))
sigma p:perm(col(trans(A)).
((sign(p)*pi i:1..col(trans(A)).
(A[(p) (i), i])))
sigma p:perm(col(trans(A)).
((sign(p)*pi i:1..col(trans(A)).
((trans(A))[i, (p) (i)])))
=det(trans(A))
end_since
SEMACS (esp) [95,15] demoll sys>user>semacs>text>det_trans..1 --4#-- #
    
```

構造エディタによる証明の展開

(この例は全称記号の除去)

展開前

```

CAP-LA_48 ==> Structure Mode...
theorem
  det_trans:
  all A:squ_mat.
    det(A)=det(trans(A))
end_theorem

menu
  <rule_menu>
  univ_elim
  contradiction
    
```

展開後

```

CAP-LA_48 ==> Structure Mode...
Theorem
  det_trans:
  all A:squ_mat.
    det(A)=det(trans(A))
  since
    let A:squ_mat be arbitrary;
      det(A)=det(trans(A))
    end_since
end_theorem
    
```

証明検査 (添字の誤りの指摘)

```

CAP-LA_55 ==> Structure Mode...
theorem
  trans_trans:
  all m, n:pos, A:matrix(m, n).
    (trans(trans(A))=A)
  since
    let m, n:pos, A:matrix(m, n) be arbitrary;
      trans(trans(A))=A
      since using theorem mat_EQ;
        col(trans(trans(A)))=row(trans(A))
          =col(A);
        row(trans(trans(A)))=col(trans(A))
          =row(A);
        all i:1..col(trans(trans(A))), j:1..row(A).
          (trans(trans(A))) [i, j] = (A) [i, j]
        since
          let i:1..col(trans(trans(A))), j:1..row(A) be arbitrary;
            (trans(trans(A))) [i, j] = (trans(A)) [i, j]
              = (A) [i, j]
          end_since
      end_since
    end_since
end_theorem
    
```

---

```

SEMACS (esp) [96.19] sys:user>semacs>text>TRTR.88.1 --29X-- *
Grammar name? >pd1
Top Category name(theory)? >
Buffer (TRTR): *G
    
```

<p>CAP CHECKER</p> <p>AUNIFY MONITOR</p> <pre> goal:(trans(trans(A))) [i, j] = (trans(A)) [i, j] rule:(trans(A)) [_B, _C] = _A[_C, _B] found difference of... i and _B, j and _C, i and _C, j and _B         </pre> <p>RULE POOL MONITOR</p> <pre> restore function : trans ... O.K.         </pre> <p>EQUALITY TRS MONITOR</p> <pre> (6,2) Goal &gt; (trans(trans(A))) [i, j] = (trans(A)) [i, j] : (by equality)         </pre> <p>press any key</p>	<p>TRS RULES</p> <pre> &lt;EQUALITY&gt; &gt;&gt;col(trans(A)) = n &gt;&gt;row(trans(trans(A))) = n &gt;&gt;col(trans(trans(A))) = n &gt;&gt;row(trans(A)) = n &gt;&gt;row(A) = n &gt;&gt;col(A) = m &lt;INEQUALITY&gt; &gt;&gt;n &gt;= 1 &gt;&gt;n &gt;= j &gt;&gt;j &gt;= 1 &gt;&gt;m &gt;= 1 &gt;&gt;m &gt;= i &gt;&gt;i &gt;= 1         </pre>
--	---