

ACP: System Description for CoCo 2012

**Takahito Aoto and Yoshihito Toyama
(Tohoku University)**

ACP (Automated Confluence Prover)

- Developed in Toyama-Aoto group in Tohoku University

ACP (Automated Confluence Prover)

- Developed in Toyama-Aoto group in Tohoku University
- Written in Standard ML of New Jersey

ACP (Automated Confluence Prover)

- Developed in Toyama-Aoto group in Tohoku University
- Written in Standard ML of New Jersey
- Provided as (not an executable binary but) a heap image that can be loaded into an SML/NJ runtime system (source code is available)

ACP (Automated Confluence Prover)

- Developed in Toyama-Aoto group in Tohoku University
- Written in Standard ML of New Jersey
- Provided as (not an executable binary but) a heap image that can be loaded into an SML/NJ runtime system (source code is available)
- ACP uses external provers
 - a SAT solver (such as **MiniSAT**)
 - an SMT solver **YICES**

```
(*** SML/NJ needs to be installed ***)
```

```
$ which sml
```

```
/usr/local/bin/sml
```

```
(*** acp.x86-linux is a heap image of ACP ***)
```

```
(*** minisat and yices are required ***)
```

```
$ ls
```

```
acp.x86-linux  minisat  sample.trs  yices
```

```
(*** input file specifies a TRS in TPDB format ***)
```

```
$ cat sample.trs
```

```
(VAR x y)
```

```
(RULES
```

```
  +(x,0)  -> x
```

```
  +(x,s(y)) -> s(+(x,y))
```

```
)
```

```
$ sml @SMLload=acp.x86-linux sample.trs
```

```
YES
```

```
Rewrite Rules:
```

```
[ +( ?x, 0) -> ?x,  
  +( ?x, s(?y)) -> s(+( ?x, ?y)) ]
```

```
Apply Direct Methods...
```

```
Inner CPs:
```

```
[ ]
```

```
Outer CPs:
```

```
[ ]
```

```
Overlay, check Innermost Termination...
```

```
Innermost Terminating (hence Terminating), WCR
```

```
Knuth&Bendix
```

```
Direct Methods: CR
```

```
Final result: CR
```

```
sample.trs: Success(CR)
```

```
(0 msec.)
```

Inside of ACP

ACP uses **multiple direct methods and divide-and-conquer methods** for proving confluence.

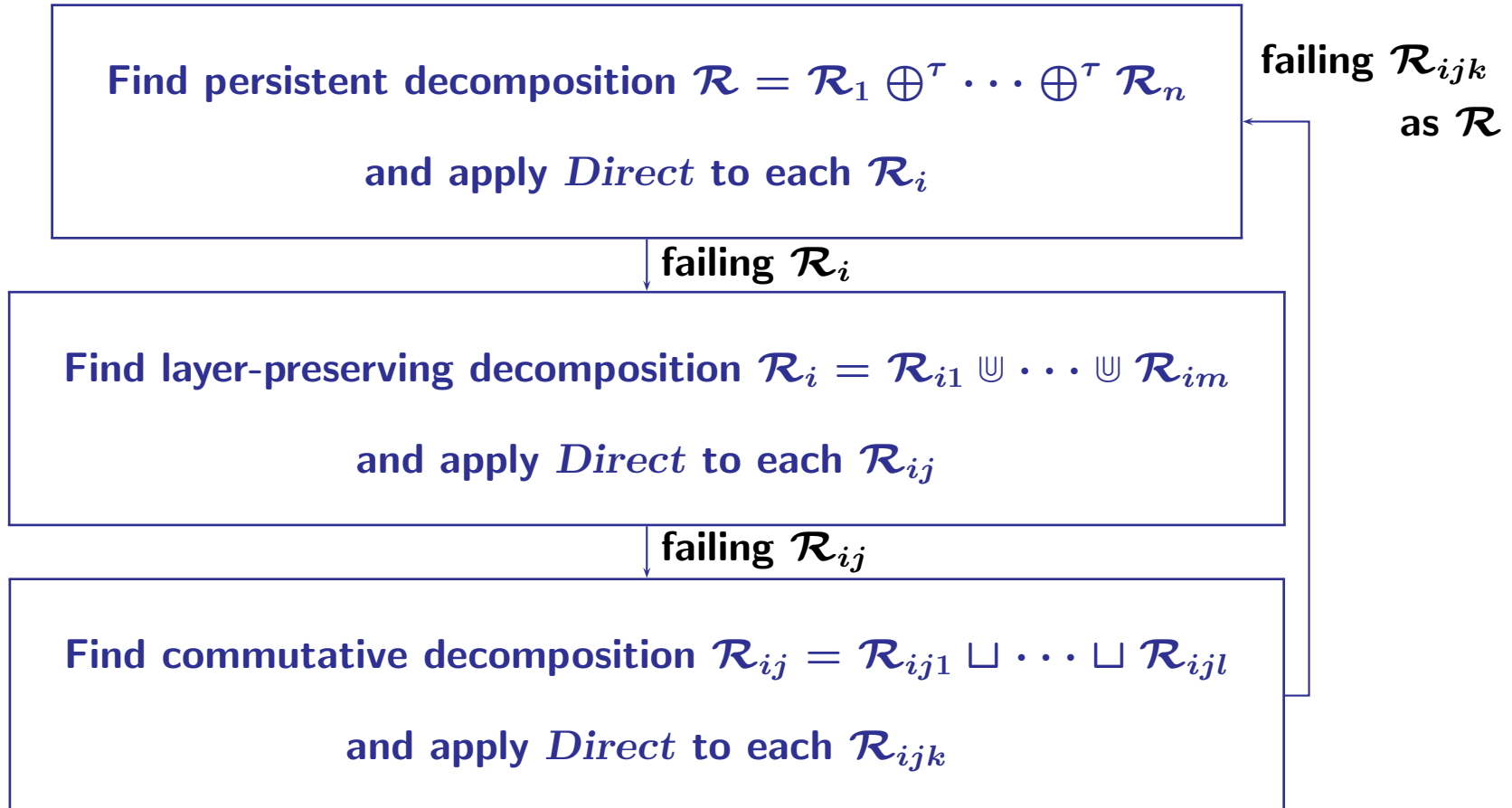
- **Divide-and-conquer methods:**
 - **Persistency**
 - **Layer-preservation**
 - **Commutation**
- **Direct methods:**
 - **Knuth–Bendix’s criterion**
 - **Huet–Toyama–van Oostrom criterion**
 - ...

Inside of ACP (1)

ACP uses multiple direct methods and divide-and-conquer methods for proving confluence.

- **Divide-and-conquer methods:**
 - **Persistency**
 - **Layer-preservation**
 - **Commutation**
- **Direct methods:**
 - **Knuth–Bendix’s criterion**
 - **Huet–Toyama–van Oostrom criterion**
 - ...

Organizing Divide-and-Conquer Methods



(Three successive failures of decomposition \Rightarrow FAIL)

Inside of ACP

ACP uses **multiple direct methods and divide-and-conquer methods** for proving confluence.

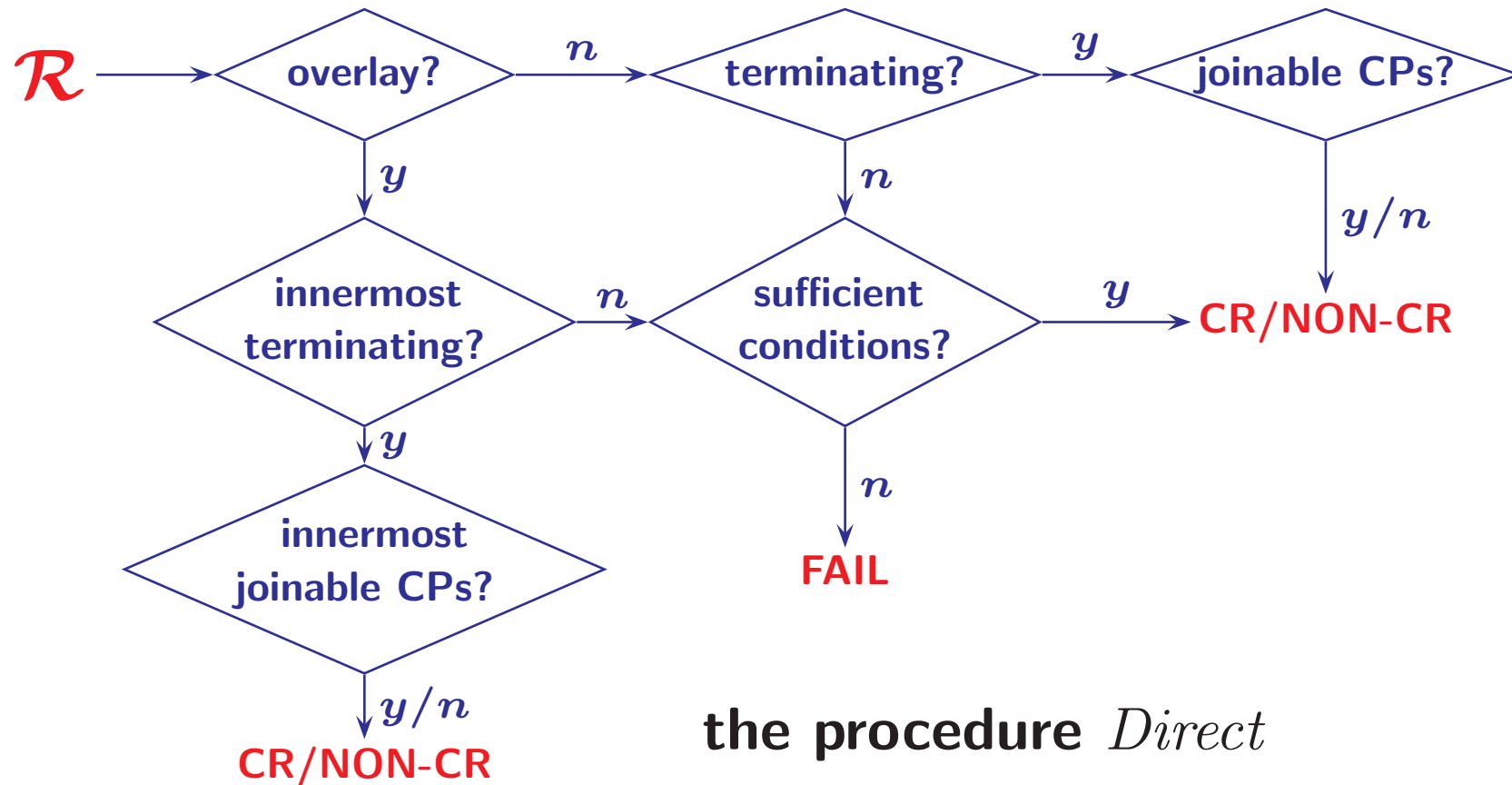
- **Divide-and-conquer methods:**
 - **Persistency**
 - **Layer-preservation**
 - **Commutation**
- **Direct methods:**
 - **Knuth–Bendix’s criterion**
 - **Huet–Toyama–van Oostrom criterion**
 - **...**

Inside of ACP (2)

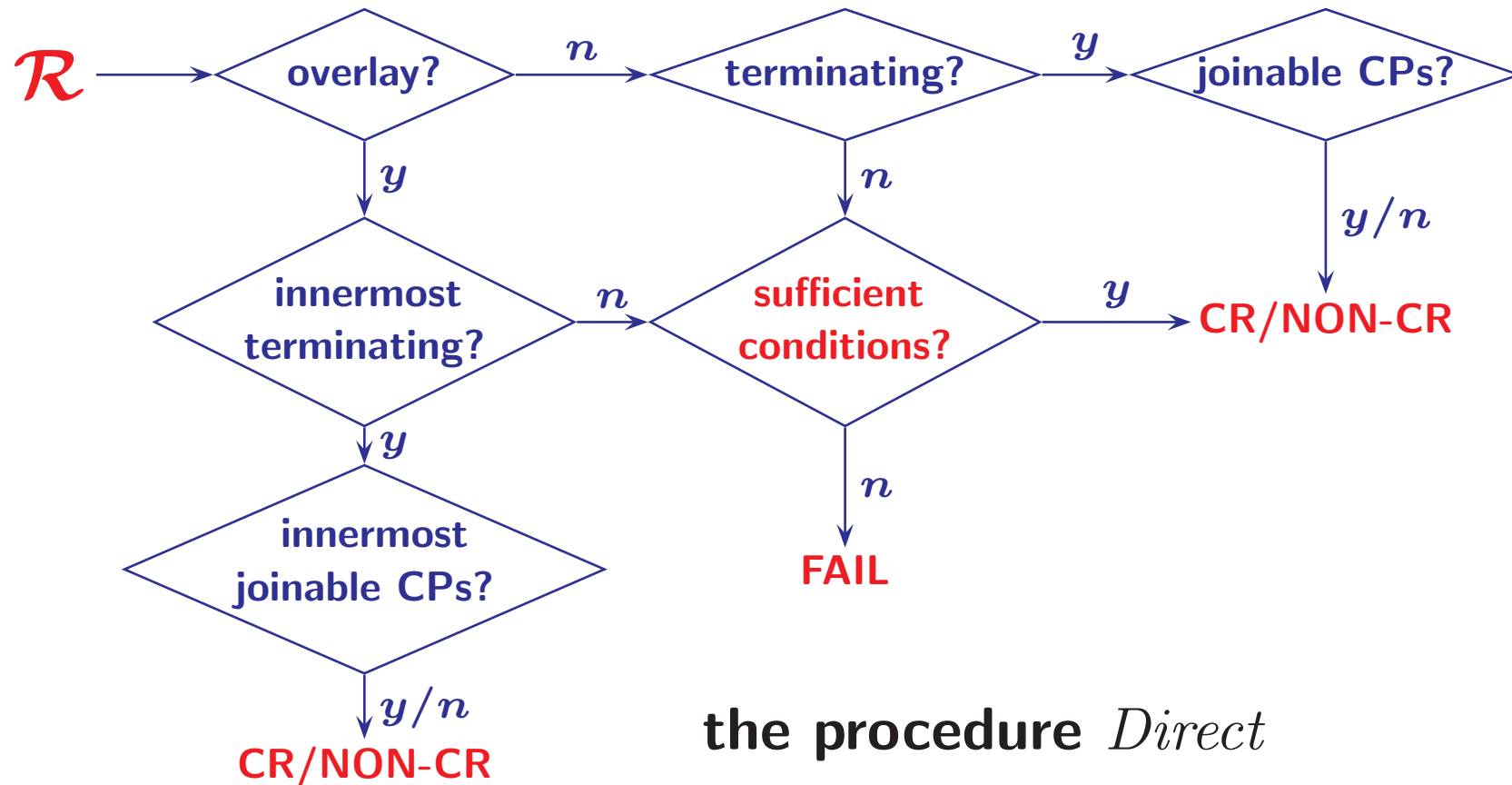
ACP uses multiple direct methods and divide-and-conquer methods for proving confluence.

- **Divide-and-conquer methods:**
 - **Persistency**
 - **Layer-preservation**
 - **Commutation**
- **Direct methods:**
 - **Knuth–Bendix’s criterion**
 - **Huet–Toyama–van Oostrom criterion**
 - **...**

Organizing Direct Methods



Organizing Direct Methods



Sufficient conditions: (with possibly approximations)

- Linear strongly closed TRSs [Huet,1980]
- Criterion based on parallel critical pairs [Toyama,1981]
- Simple-right-linear non-E-overlapping TRSs [Ohta&Oyamaguchi&Toyama,1995]
- Left-linear development closed TRSs

[Huet,1980–Toyama,1988–van Oostrom,1997]

- Criterion based on simultaneous critical pairs [Okui,1998]
- Strongly depth-preserving non-E-overlapping TRSs [Gomi&Oyamaguchi&Ohta,1998]
- Left-linear upside-parallel-closed or outside-closed TRSs [Oyamaguchi&Ohta,2004]
- Decreasing diagrams based on rule-labelling [van Oostrom,2008–Aoto,2010]
- Reduction-preserving completion [Aoto&Toyama,2012]
- Non-confluence check

```
$ sml @SMLload=acp.x86-linux --help
```

```
acp v0.31
```

```
-h          --help          print this message
-v 0|1|2|3  --verbose=0|1|2|3    verbosity level [0: YES/NO/MAYBE, 1(default): YES/NO/MAYBE+explanatio
--ps=0|1|no|yes      use persistent decomposition [default:yes]
--lp=0|1|no|yes      use layer-preserving decomposition [default:yes]
--cm=0|1|no|yes      use commutative decomposition [default:yes]
--nm=0|1|no|yes      with non-minimal commutative decomposition [default:yes]
-d          --direct          equivalent to --ps=0 --lp=0 --cm=0
-t int      --timer=int      specify the time limit in seconds [default:0 (no-limit)]
--minisat-path=path  specify an executable minisat [default:./minisat]
--yices-path=path    specify an executable yices [default:./yices]
--tmp-dir=path       specify a directory for putting temporal files [default:.]
--termination-prover=path  specify an external termination prover
--relative-termination-prover=path  specify an external relative termination prover
-r          --reset          turn off all criteria by default (use before enable options)
--enable-KB          (Knuth&Bendix,1970)
--enable-strong      (Huet,1980) strongly closed
--enable-parallel    (Toyama,1981) parallel CPs
--enable-simple      (Ohta&Oyamaguch&Toyama,1995) simple-right-linear
--enable-development (Huet,1980)-(Toyama,1988)-(van Oostrom,1997) development closed
--enable-simultaneous (Okui,1998) simultaneous CPs
--enable-depth       (Gomi&Oyamaguch&Ohta,1998) strongly depth-preserving
--enable-upside      (Oyamaguchi&Ohta,2004) upside-parallel-closed and outside-closed
--enable-decreasing  (van Oostrom,2008)-(Aoto,2010) decreasing diagram with rule-labelling
--enable-modulo      (Huet,1980) Church-Rosser modulo
--enable-completion  (Aoto&Toyama,2012) reduction-preserving completion
--enable-disproof    disproving for non-terminating system
```


Setting for CoCo 2012:

- **Version 0.31**
- **Internal termination prover is employed**

How to obtain:

- **the source code and heap images can be obtained from**

<http://www.nue.riec.tohoku.ac.jp/tools/acp/>

Setting for CoCo 2012:

- **Version 0.31**
- **Internal termination prover is employed**

How to obtain:

- **the source code and heap images can be obtained from**

<http://www.nue.riec.tohoku.ac.jp/tools/acp/>

Thank you!