An automatic off-line feature interaction detection method by static analysis of specifications

Hélène Jouve, Pascale Le Gall and Sophie Coudert

ICFI'05 Leicester

France Telecom R&D

Université d'Évry - LaMI

ENST - Laboratoire SoC

Introduction

Service : extension of a basic system

Exemple : POTS, CFB, TCS, CW, etc.

Separately designed but integrated on the same network => interactions

Solve or prevent interactions => detect them before integration of services of at runtime

New challenges : internet telephony

Our objectives

1. **Detect** interactions,
2. **At the earliest**, from **graphical specifications**
   - $\rightsquigarrow$ *minimize impact of conflicts*
   - $\rightsquigarrow$ *graphical is easily readable*
3. **Automatically**
   - $\rightsquigarrow$ without supplementary knowledge than the specifications
4. **Give an explanation** of detected conflict
   - $\rightsquigarrow$ Helps solving conflicts or modify specification

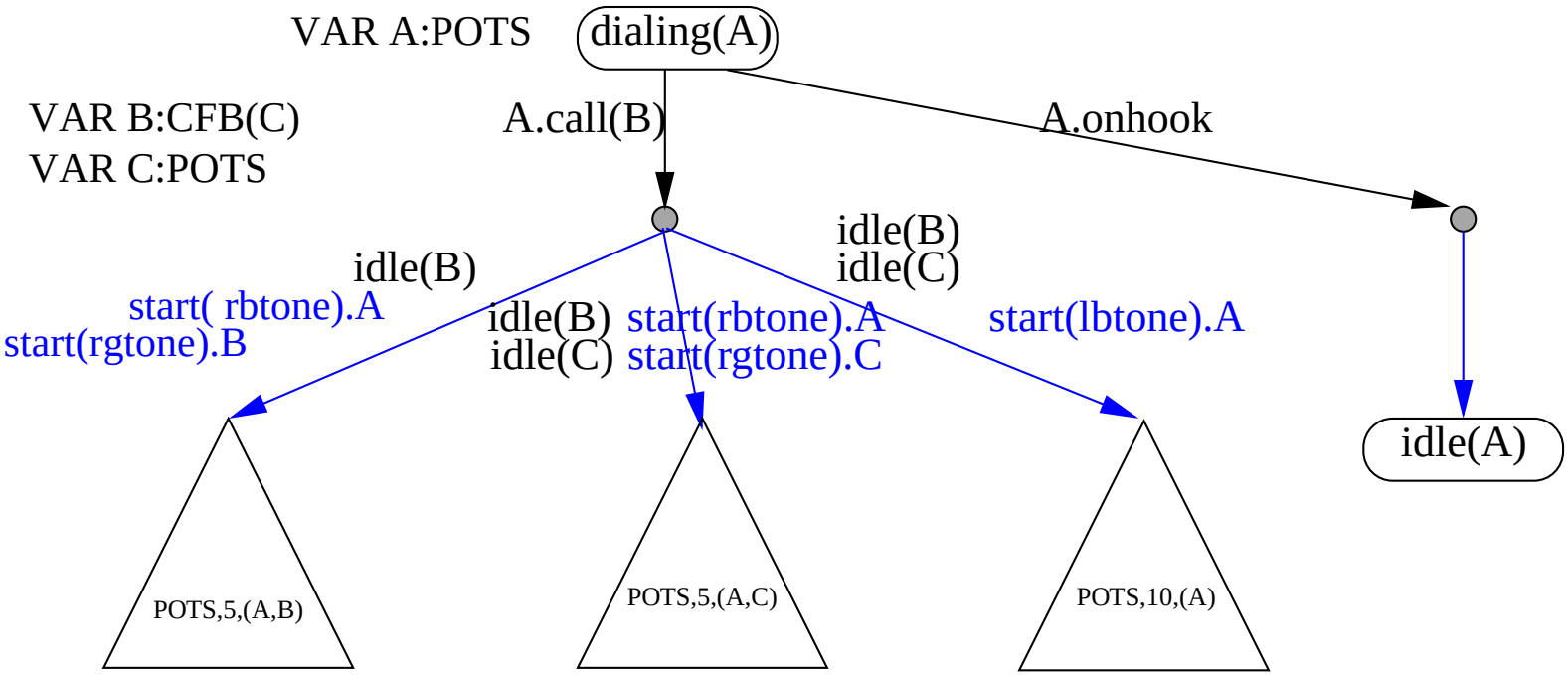
Plan

1. Our method for phone services
2. Current and futur works : SIP

Our choices

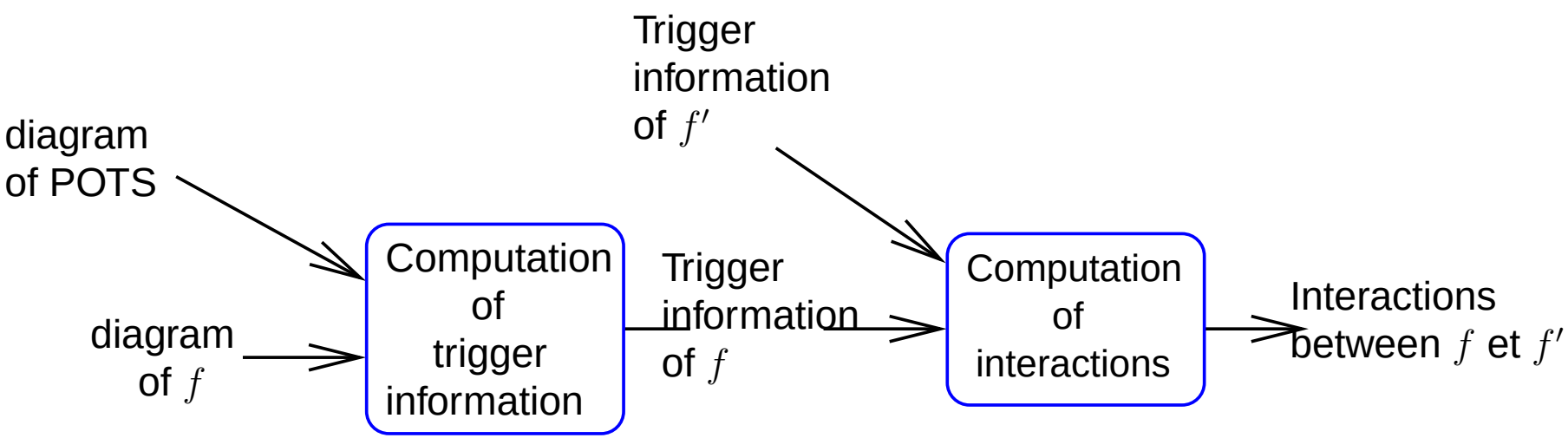
- operational and graphical specifications
  - compromise between expressiveness and feasibility of computations
  - user point of view, abstraction of the network
- simple properties
  - formulas such as $\mathit{emitting}(A, \mathit{ringingtone}) \Rightarrow \neg \mathit{idle}(A)$
- No integration, combination choices, no computed model
  - static method

Example of specification : focus on CFB



- Aliases
- Subscriptions and variables declarations
- Determinism of transitions
- Invariant properties

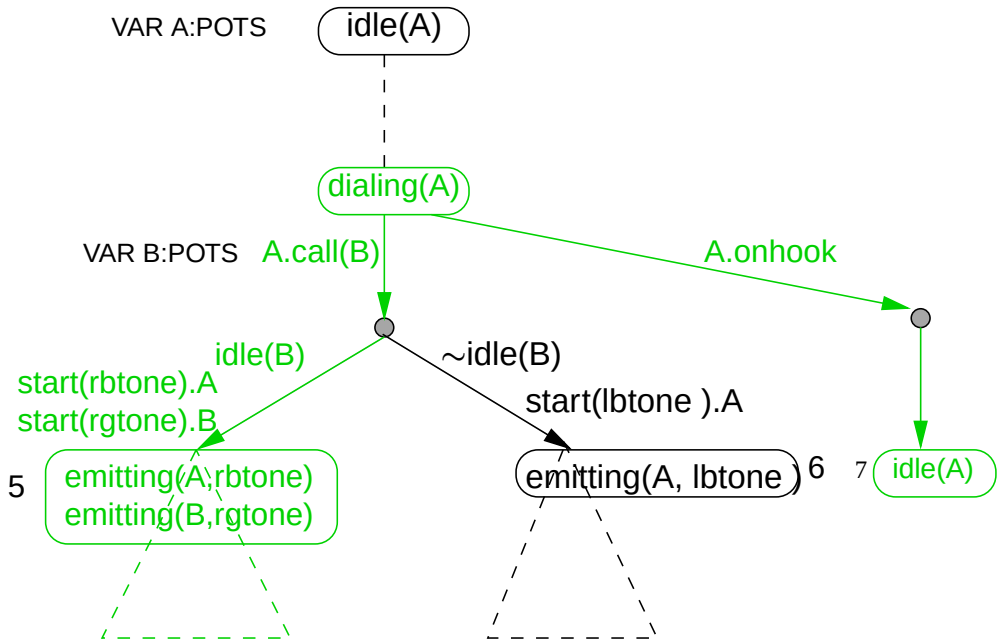
Main steps of our method

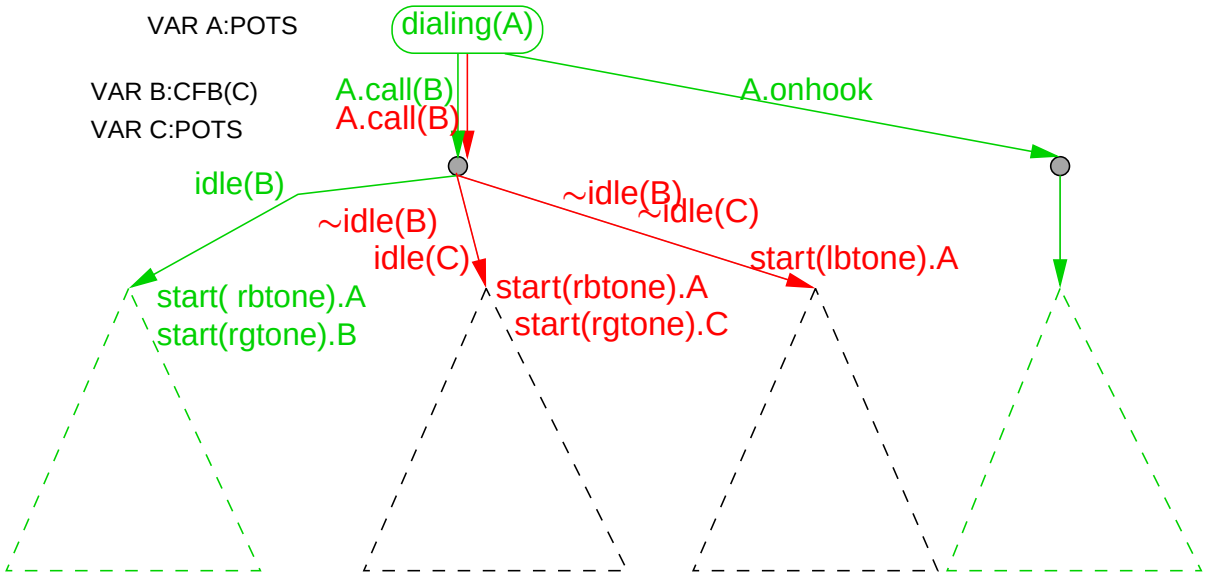


Computing and modelling trigger information

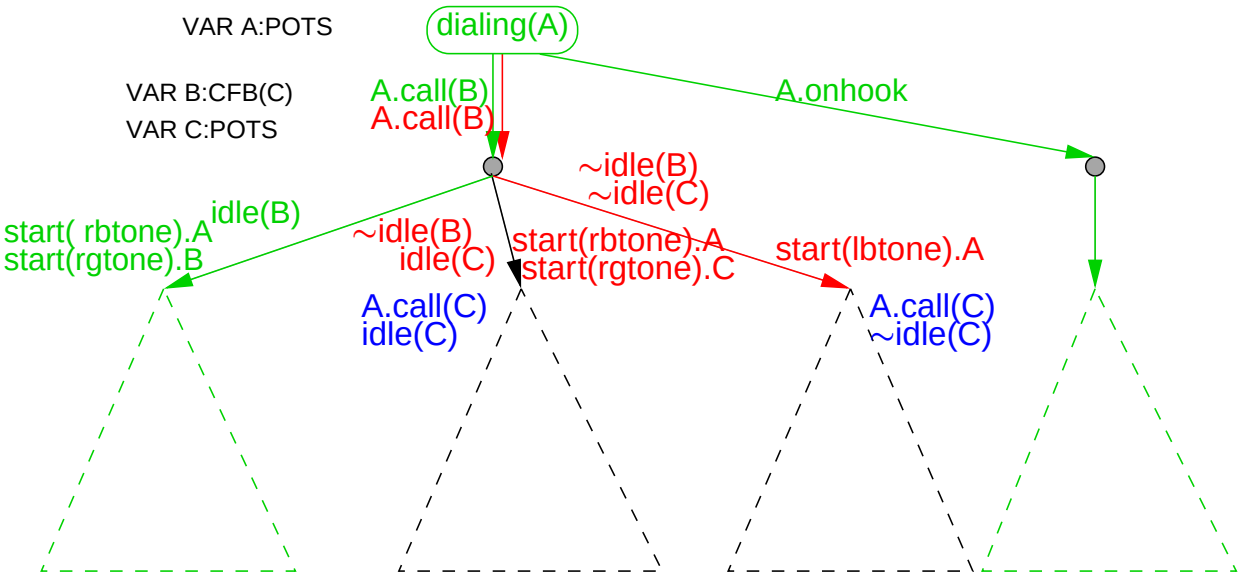
- Step 1: supress aliases (except loops)
- Step 2: pair branches and extract non paired branches (trigger)
- Step 3: annotate non paired branches with “intentions”

Computing trigger information: pairing diagrams

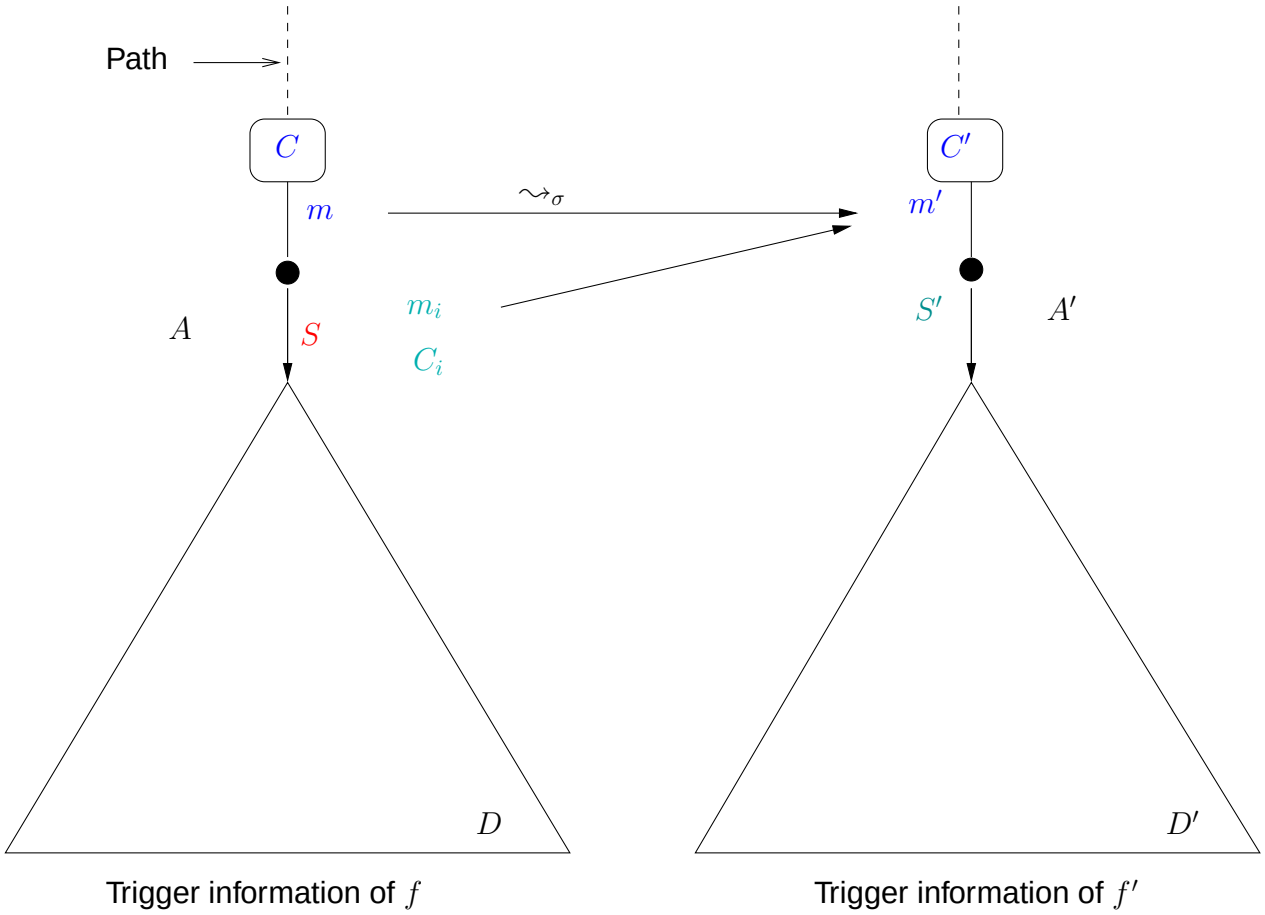




Computing trigger information: intentions



Computing interactions: direct and indirect at trigger



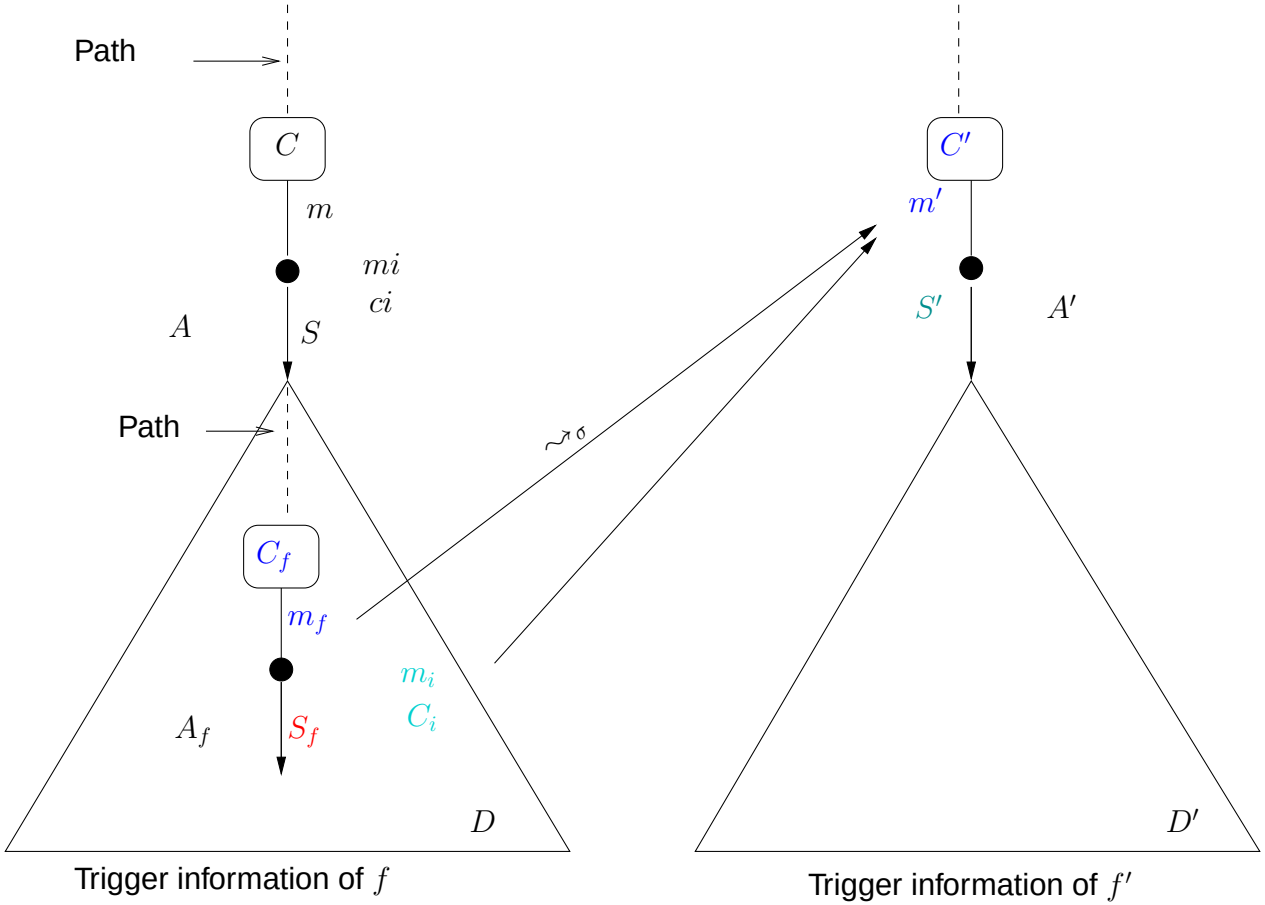
Trigger information of f Trigger information of f'

1. direct interactions:
   If *Compat* $^{\sigma}(C, C')$

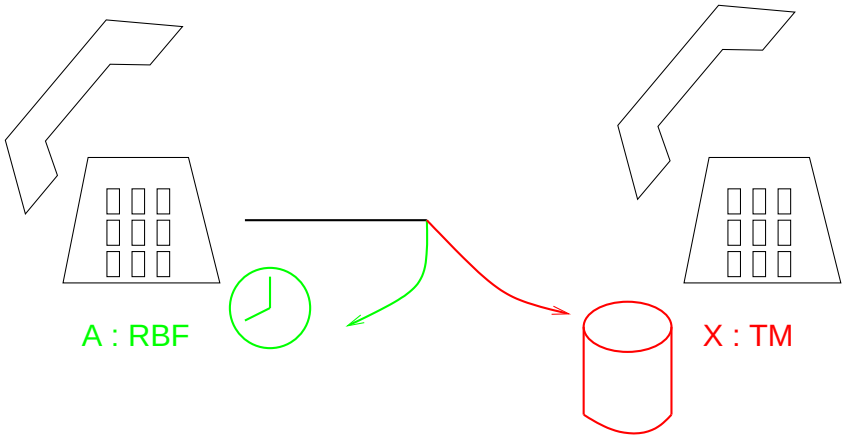
Subs.	$\sigma(A) \cup A'$
Cond.	$\sigma(C) \cup C'$
Msg.	$\sigma(m) = m'$
Conflict	$\sigma(S)$ ou S'

2. indirect interactions using intentional message

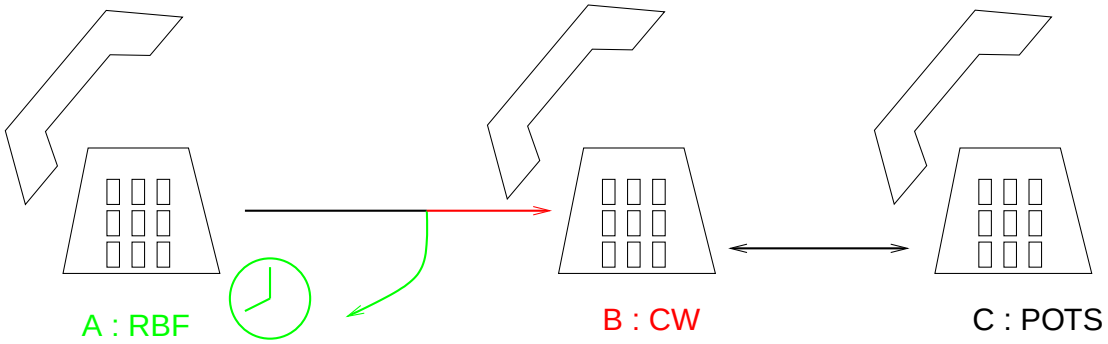
Computing interactions : generalization



Example of computed interactions (RBF vs TM et RBF vs CW)

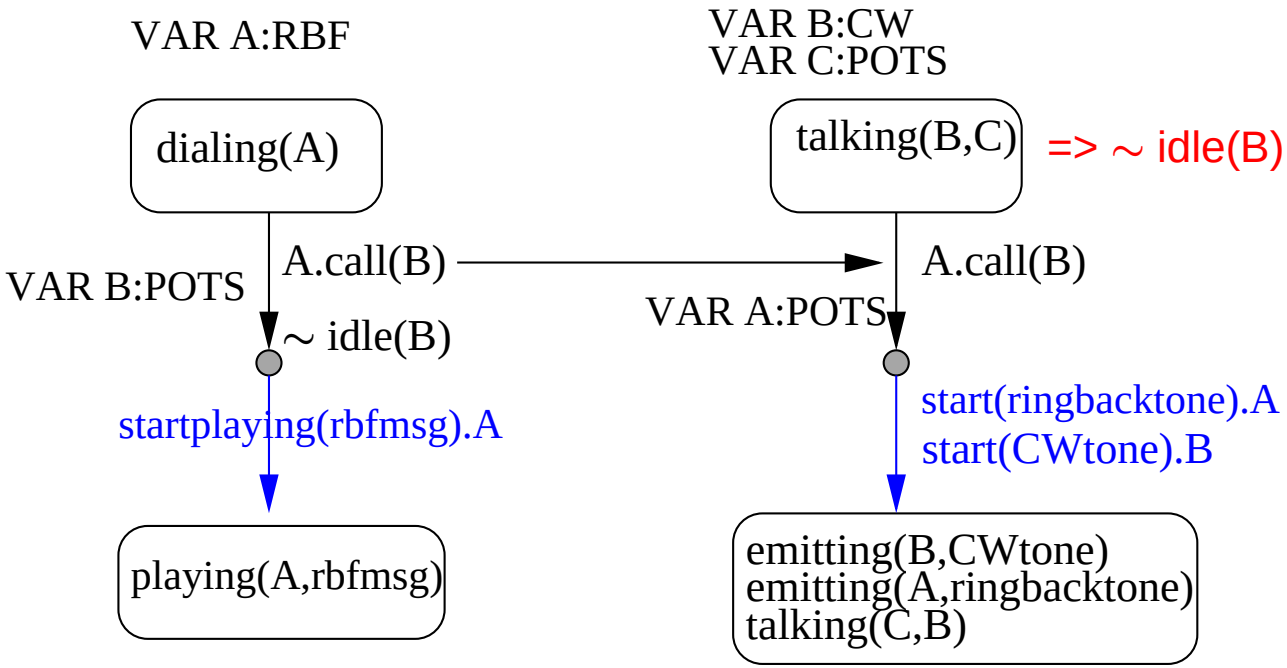


S	A : *RBF* B : *TM*
C	*dialing*(A) $\sim$ *idle*(B)
m	A.*call*(B)
	startplaying(*RBFmsg*).A *startplaying*(*TMaccueilmsg*).A



S	A : *RBF* B : *CW* C:*POTS*
C	*dialing*(A) *talking*(B,C)
m	A.*call*(B)
	startplaying(*RBFmsg*).A *start*(*CWtone*).B

RBF vs CW



- $\{dialing(A), \sim idle(B)\}$, and $\{talking(B, C)\})$ are compatible
- Trigger messages match
- Network answers are different

Conclusion for this technique

- number and expressiveness of computed interactions
- rapidity of computations
- simplicity of use
- run on 23 services (FIW contests)

Plan

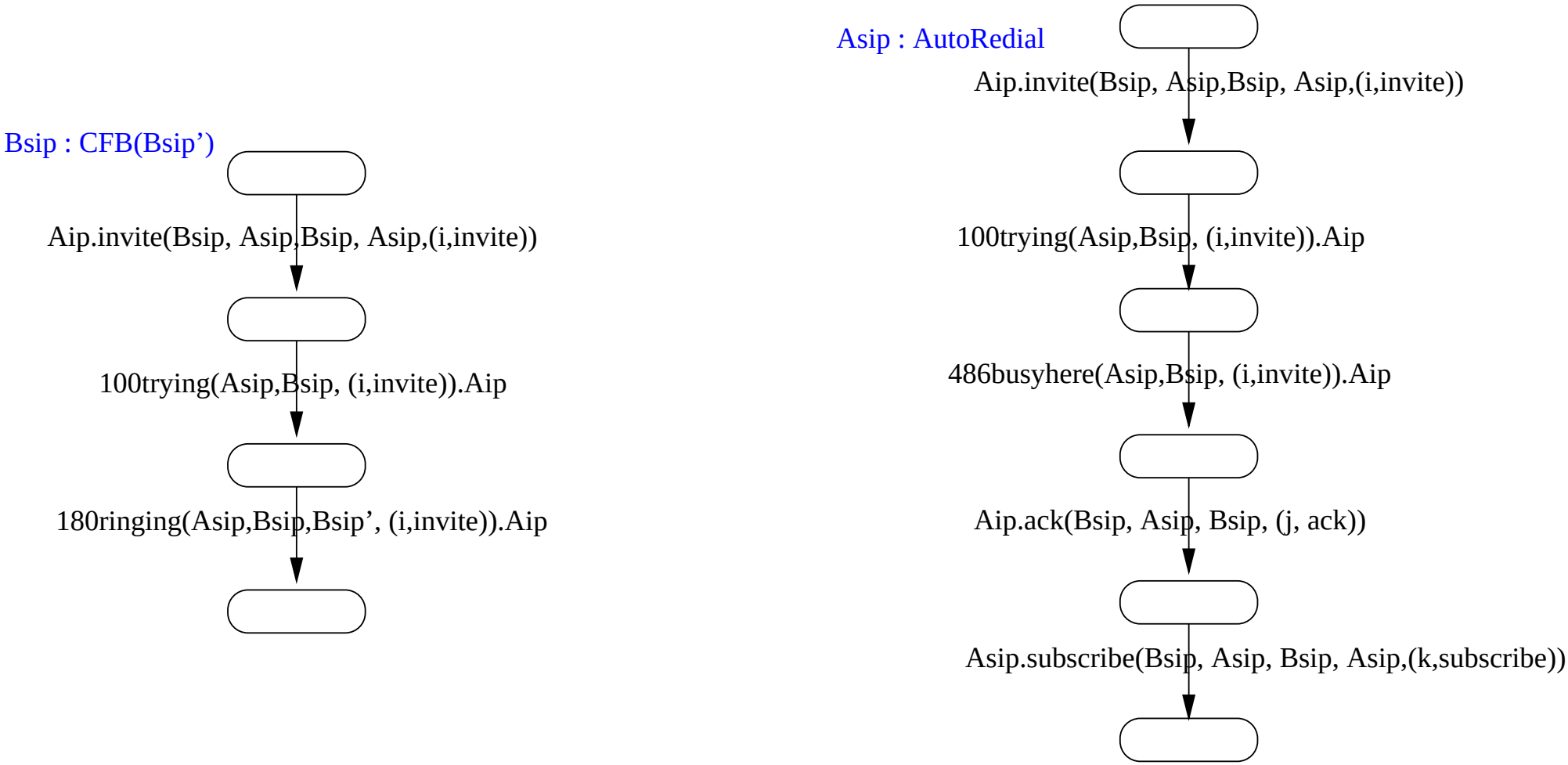
1. Our method for phone services
2. Current and futur works : SIP

SIP and VoIP services

- Services deployed on devices, proxies, etc
- End systems are more “intelligent” than phones
- Work in progress at France Telecom
  - formalism to be adapted
  - algorithms to be adapted

  $\rightsquigarrow$ Intellectual property issues...

CFB and Auto Redial



Interactions (by hand)

Configuration	Bsip : CBF(Bsip') Asip : AutoRedial
Path	Aip.invite(. . .) 100trying(. . .).Aip
Trigger request	Aip.invite(. . .)
Conflicting answers	180ringing(. . .).Aip 486busyhere(. . .).Aip

Conclusion

- Graphic and formal specifications adapted to the domain
- Fully automatic method
  - automatic computation of problematic configurations
  - no integration
- Quality and number of found interactions
- Adaptation of formalism and algorithms : in progress