# Abduction and Contraction in Description Logics

## *What, How, and Why*

***Francesco M Donini***

DISCOM

Dipartimento di Studi sulla Comunicazione

Università della Tuscia – Viterbo

# Abstract

In the notes of this slide

The graft of Semantic Annotation into Electronic Commerce (EC) brings new opportunities for the application of Knowledge Representation techniques, originally devised for Knowledge Bases. Description Logics (DL) are one of the formal basis for Semantic Annotation, and the reasoning services they provide can be extended to cope with problems stemming from EC.

In this talk, I first give the EC scenario, recall the available Semantic Annotation technology, and highlight reasoning problems. Then, I introduce Concept Abduction and Concept Contraction as extensions of DL reasoning services.

In the second part of the talk, I present a Tableaux-based method to compute (some) abductions and contractions.

# Outline of the talk

- Motivation: Electronic Commerce

# Outline of the talk

- Motivation: Electronic Commerce
- Abduction and Contraction: Definitions

# Outline of the talk

- Motivation: Electronic Commerce
- Abduction and Contraction: Definitions
  - Logical and computational properties

# Outline of the talk

- Motivation: Electronic Commerce
- Abduction and Contraction: Definitions
  - Logical and computational properties
- A Tableaux-based calculus

# Outline of the talk

- Motivation: Electronic Commerce
- Abduction and Contraction: Definitions
    - Logical and computational properties
- A Tableaux-based calculus
- Implementation

# Outline of the talk

- Motivation: Electronic Commerce
- Abduction and Contraction: Definitions
    - Logical and computational properties
- A Tableaux-based calculus
- Implementation
- What next?

# Motivation

# Peer-to-Peer Electronic Commerce

(*P2P EC* from now on)

# Peer-to-Peer Electronic Commerce

(*P2P EC* from now on)

offers (supplies)
requests (demands)
services

meet in
⟹

Electronic
Marketplace

+ trusted
third party

# Peer-to-Peer Electronic Commerce

(*P2P EC* from now on)

| offers (supplies)<br>requests (demands)<br>services | meet in<br>$\Longrightarrow$ | Electronic Marketplace<br>+ trusted third party |
|---|---|---|

- Marketplace: mostly, Web Site with human interaction

# Peer-to-Peer Electronic Commerce

(*P2P EC* from now on)

offers (supplies)
requests (demands)
services

meet in ⟹

Electronic Marketplace

+ trusted third party

- Marketplace: mostly, Web Site with human interaction
- Renowned example: eBay `http://www.ebay.com`

# Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on `www.automobili.com`

# Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on `www.automobili.com`
- a room to share in Rome: 851 offers on `www.easystanza.com`

# Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on `www.automobili.com`
- a room to share in Rome: 851 offers on `www.easystanza.com`
- a used Notebook PC: 2361 offers on `informatica.ebay.it`

# Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on `www.automobili.com`
- a room to share in Rome: 851 offers on `www.easystanza.com`
- a used Notebook PC: 2361 offers on `informatica.ebay.it`

. . . how did you choose?

# Some figures

Did you ever tried to find . . .

- a used Fiat Panda gasoline: 109 offers on `www.automobili.com`
- a room to share in Rome: 851 offers on `www.easystanza.com`
- a used Notebook PC: 2361 offers on `informatica.ebay.it`

. . . *which reasoning* did you employed?

# P2P is not B2C

- B2C: Business-to-Consumer

- P2P: *Peer*-to-*Peer*

# P2P is not B2C

- B2C: Business-to-Consumer
- usually, the seller owns the Web Site

- P2P: *Peer*-to-*Peer*
- the Web Site is of some *third party*

# P2P is not B2C

- B2C: Business-to-Consumer
- usually, the seller owns the Web Site
- the seller publishes offers

- P2P: *Peer*-to-*Peer*
- the Web Site is of some *third party*
- *both* parties can publish on the Web Site

# P2P is not B2C

- B2C: Business-to-Consumer
- usually, the seller owns the Web Site
- the seller publishes offers
- the client browses...

- P2P: *Peer*-to-*Peer*
- the Web Site is of some *third party*
- *both* parties can publish on the Web Site
- *Both* parties may take initiative (and browse...)

# Available Technology

# Semantic Annotation

- “The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”

# Semantic Annotation

- “The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”

— OWL - Web Ontology Language Overview

# Semantic Annotation

- “The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”

  — OWL - Web Ontology Language Overview
- *DAML* - DARPA Agent Markup Language

# Semantic Annotation

- “The Semantic Web is a vision for the future of the Web in which information is given explicit meaning, making it easier for machines to automatically process and integrate information available on the Web.”
  — OWL - Web Ontology Language Overview
- *DAML* - DARPA Agent Markup Language
- Web Services can be described through languages like *DAML-S, OWL-S*,. . .

# An Example in OWL — more precisely, OWL-Lite

“On-sale PCs are . . .

home PCs with at most one OS, of type WinX”

# An Example in OWL — more precisely, OWL-Lite

“On-sale PCs are . . .

home PCs with at most one OS, of type WinX”

---

```
< owl : Class rdf : ID = ”onSalePC”/ >
    < rdfs : subClassOf >
        < owl : intersectionOf rdf : parseType = ”Collection” >
            < owl : Class rdf : ID = ”homePC”/ >
            < owl : Restriction >
                < owl : onProperty rdf : resource = ”hasOS” / >
                < owl : maxCardinality
                rdf : datatype = ”&xsd; nonNegativeInteger” >
                    1
                < /owl : maxCardinality >
                < owl : allValuesFrom
                rdf : resource = ”#winX”/ >
            < /owl : Restriction >
        < /owl : intersectionOf >
    < /rdfs : subClassOf >
< /owl : Class >
```

# An Example in OWL — more precisely, OWL-Lite

“On-sale PCs *are* . . .

home PCs with at most one OS, of type WinX”

---

```
< owl : Class rdf : ID = ”onSalePC”/ >
    < rdfs : subClassOf >
        < owl : intersectionOf rdf : parseType = ”Collection” >
            < owl : Class rdf : ID = ”homePC”/ >
            < owl : Restriction >
                < owl : onProperty rdf : resource = ”hasOS” / >
                < owl : maxCardinality
                rdf : datatype = ”&xsd; nonNegativeInteger” >
                    1
                < /owl : maxCardinality >
                < owl : allValuesFrom
                rdf : resource = ”#winX”/ >
            < /owl : Restriction >
        < /owl : intersectionOf >
    < /rdfs : subClassOf >
< /owl : Class >
```

# An Example in OWL — more precisely, OWL-Lite

“On-sale PCs are . . .

home PCs *with at most one OS,* of type WinX”

---

```
< owl : Class rdf : ID = ”onSalePC”/ >
    < rdfs : subClassOf >
        < owl : intersectionOf rdf : parseType = ”Collection” >
            < owl : Class rdf : ID = ”homePC”/ >
            < owl : Restriction >
                < owl : onProperty rdf : resource = "hasOS" / >
                < owl : maxCardinality
                rdf : datatype = ”&xsd; nonNegativeInteger” >
                     1
                < /owl : maxCardinality >
                < owl : allValuesFrom
                rdf : resource = ”#winX”/ >
            < /owl : Restriction >
        < /owl : intersectionOf >
    < /rdfs : subClassOf >
< /owl : Class >
```

# General Assumptions

Based on Semantic Annotation, we assume that

- Offers, Requests, Services are *logic formulas* $O, R, S, \ldots$

# General Assumptions

Based on Semantic Annotation, we assume that

- Offers, Requests, Services are *logic formulas* $O, R, S, \ldots$
- The marketplace ontology is a *logic Theory* $\mathcal{T}$

# General Assumptions

Based on Semantic Annotation, we assume that

- Offers, Requests, Services are *logic formulas* $O, R, S, \ldots$
- The marketplace ontology is a *logic Theory* $\mathcal{T}$
- an agreement betw. $O$ and $R$ is either a...

# General Assumptions

Based on Semantic Annotation, we assume that

- Offers, Requests, Services are *logic formulas* $O, R, S, \ldots$
- The marketplace ontology is a *logic Theory* $\mathcal{T}$
- an agreement betw. $O$ and $R$ is either a...
    - model of $\mathcal{T} \cup \{O, R\}$, or a...

# General Assumptions

Based on Semantic Annotation, we assume that

- Offers, Requests, Services are *logic formulas* $O, R, S, \ldots$
- The marketplace ontology is a *logic Theory* $\mathcal{T}$
- an agreement betw. $O$ and $R$ is either a...
  - model of $\mathcal{T} \cup \{O, R\}$, or a...
  - set of models of $\mathcal{T} \cup \{O, R\}$, or a...

# General Assumptions

Based on Semantic Annotation, we assume that

- Offers, Requests, Services are *logic formulas* $O, R, S, \ldots$
- The marketplace ontology is a *logic Theory* $\mathcal{T}$
- an agreement betw. $O$ and $R$ is either a...
  - model of $\mathcal{T} \cup \{O, R\}$, or a...
  - set of models of $\mathcal{T} \cup \{O, R\}$, or a...
  - formula consistent with $\mathcal{T} \cup \{O, R\}$

# Matchmaking

# What’s Matchmaking?

*First phase* in a Bilateral Commercial Transaction:

1. *Matchmaking* (find counterpart)
2. Negotiation (agree/tradeoff details)
3. Exchange (goods, services, money)

# Reasoning for Matchmaking

Which kind of reasoning is necessary for matching offers and demands?

# An Example — a cognitive experiment

From *Sunday Times, online marketplace*

- Request: Ferrari 430 Coupe/Spider urgently required.Best prices paid. Immediate decision.

# An Example — a cognitive experiment

From *Sunday Times, online marketplace*

- Request: Ferrari 430 Coupe/Spider urgently required.Best prices paid. Immediate decision.
- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

# An Example — a cognitive experiment

From *Sunday Times, online marketplace*

- Request: Ferrari 430 Coupe/Spider urgently required.Best prices paid. Immediate decision.
- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

Do they match?

# An Example — a cognitive experiment

From *Sunday Times, online marketplace*

- Request: Ferrari 430 Coupe/Spider urgently required.Best prices paid. Immediate decision.
- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

*How well* they match? (compared to other offers/requests)

# Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a *facilitator* system

—But: which reasoning?

# Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a *facilitator* system

—But: which reasoning?

- *Compare* (possibly with deduction)

# Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a *facilitator* system

—But: which reasoning?

- *Compare* (possibly with deduction)
- *Posit* missing information

# Aim: less browsing in P2P EC

Solution: move the reasoning methods from persons browsing ads into a *facilitator* system

—But: which reasoning?

- *Compare* (possibly with deduction)
- *Posit* missing information
- *Revise* conflicting issues

# A first classification based on $\models$

[Trastour *et al.*, 2002],[Di Noia *et al.*, 2003]
An offer $O$ and a request $R$ match...

- *exactly* if $\mathcal{T} \models O \equiv R$

# A first classification based on $\models$

[Trastour *et al.*, 2002],[Di Noia *et al.*, 2003]
An offer $O$ and a request $R$ match...

- *exactly* if $\mathcal{T} \models O \equiv R$
- *potentially* if $\mathcal{T} \not\models \neg(O \wedge R)$

# A first classification based on $\models$

[Trastour *et al.*, 2002],[Di Noia *et al.*, 2003]
An offer $O$ and a request $R$ match...

- *exactly* if $\mathcal{T} \models O \equiv R$
- *potentially* if $\mathcal{T} \not\models \neg(O \wedge R)$
  - *i.e.*, if $O \wedge R$ is consistent with $\mathcal{T}$

# A first classification based on $\models$

[Trastour *et al.*, 2002],[Di Noia *et al.*, 2003]
An offer $O$ and a request $R$ match...

- *exactly* if $\mathcal{T} \models O \equiv R$
- *potentially* if $\mathcal{T} \not\models \neg(O \wedge R)$
  - *i.e.*, if $O \wedge R$ is consistent with $\mathcal{T}$
- *partially* if $\mathcal{T} \models \neg(O \wedge R)$

# A first classification based on $\models$

[Trastour *et al.*, 2002],[Di Noia *et al.*, 2003]
An offer $O$ and a request $R$ match...

- *exactly* if $\mathcal{T} \models O \equiv R$
- *potentially* if $\mathcal{T} \not\models \neg(O \wedge R)$
    - *i.e.*, if $O \wedge R$ is consistent with $\mathcal{T}$
- *partially* if $\mathcal{T} \models \neg(O \wedge R)$
    - significant if only some details conflict

# A first classification based on $\models$

[Trastour *et al.*, 2002],[Di Noia *et al.*, 2003]
An offer $O$ and a request $R$ match...

- *exactly* if $\mathcal{T} \models O \equiv R$
- *potentially* if $\mathcal{T} \not\models \neg(O \wedge R)$
  - *i.e.*, if $O \wedge R$ is consistent with $\mathcal{T}$
- *partially* if $\mathcal{T} \models \neg(O \wedge R)$
  - significant if only some details conflict
- *“plug-in”* (w.r.t. $R$) if $\mathcal{T} \models R \Rightarrow O$ [Zaremski and Wing, 1997]

# A first classification based on $\models$

[Trastour *et al.*, 2002],[Di Noia *et al.*, 2003]
An offer $O$ and a request $R$ match...

- *exactly* if $\mathcal{T} \models O \equiv R$
- *potentially* if $\mathcal{T} \not\models \neg(O \wedge R)$
  - *i.e.*, if $O \wedge R$ is consistent with $\mathcal{T}$
- *partially* if $\mathcal{T} \models \neg(O \wedge R)$
  - significant if only some details conflict
- *“plug-in”* (w.r.t. $R$) if $\mathcal{T} \models R \Rightarrow O$ [Zaremski and Wing, 1997]
- *fully* (w.r.t. $R$) if $\mathcal{T} \models O \Rightarrow R$

# Evaluating the match

- Request: Ferrari 430 Coupe/Spider urgently required. Best prices paid. Immediate decision.
- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

# Evaluating the match

- Request: Ferrari *430* Coupe/Spider urgently required. Best prices paid. Immediate decision.
- Offer: 2000/V FERRARI *360* Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

conflicting info: *430* vs. *360* (different models)

# Evaluating the match

- Request: Ferrari 430 *Coupe/Spider urgently required.* Best prices paid. Immediate decision.
- Offer: 2000/V FERRARI 360 Modena F1 Argento Nurburgring with Bordeaux Leather 22,700 £65,000 NE England

in $R$, not in $O$: *Coupe/Spider, urgently required*

# Evaluating the match

- Request: Ferrari 430 Coupe/Spider urgently required. Best prices paid. Immediate decision.
- Offer: 2000/V FERRARI 360 Modena F1 *Argento Nurburgring with Bordeaux Leather* *22,700* £65,000 NE England

in $O$, not in $R$: color *Argento, Bordeaux Leather* seats, *22,700* miles, $\ldots$

# Abduction and Contraction

# Abduction (history)

- C. S. Peirce (1839–1914)
  From $A \Rightarrow B$ and $B$, *abduce* $A$
- Abduction was the first step of scientific reasoning, the other two being
  - Deduction
  - Induction
- since [Pople, 1973] has been used to formalize Diagnosis in AI

# Abduction for Diagnosis— A crash course

[Levesque, 1989]: *Given*

- symptoms $S$, *e.g.*, yellow-eyes

# Abduction for Diagnosis— A crash course

[Levesque, 1989]: *Given*

- symptoms $S$, *e.g.*, yellow-eyes
- a theory $T$, *e.g.*,

$$\begin{array}{l}\text{hepatitis} \Rightarrow \text{yellow-eyes,}\\ \text{jaundice} \Rightarrow \text{yellow-eyes}\end{array}$$

# Abduction for Diagnosis— A crash course

[Levesque, 1989]: *Given*

- symptoms $S$, *e.g.*, yellow-eyes
- a theory $T$, *e.g.*,
  hepatitis $\Rightarrow$ yellow-eyes,
  jaundice $\Rightarrow$ yellow-eyes

*Find* a (minimal) hypothesis $H$ such that

# Abduction for Diagnosis — A crash course

[Levesque, 1989]: *Given*

- symptoms $S$, *e.g.*, yellow-eyes
- a theory $T$, *e.g.*, $\begin{array}{l}\text{hepatitis} \Rightarrow \text{yellow-eyes,} \\ \text{jaundice} \Rightarrow \text{yellow-eyes}\end{array}$

*Find* a (minimal) hypothesis $H$ such that

- $T \wedge H$ is satisfiable, and

# Abduction for Diagnosis— A crash course

[Levesque, 1989]: *Given*

- symptoms $S$, *e.g.*, yellow-eyes
- a theory $T$, *e.g.*, hepatitis $\Rightarrow$ yellow-eyes, jaundice $\Rightarrow$ yellow-eyes

*Find* a (minimal) hypothesis $H$ such that

- $T \wedge H$ is satisfiable, and
- $T \wedge H \models S$

# Example of Diagnosis— still a crash course

- symptoms: yellow-eyes

# Example of Diagnosis— still a crash course

- symptoms: yellow-eyes
- theory: $\begin{array}{l} \text{hepatitis} \Rightarrow \text{yellow-eyes}, \\ \text{jaundice} \Rightarrow \text{yellow-eyes} \end{array}$

# Example of Diagnosis— still a crash course

- symptoms: yellow-eyes
- theory: $\begin{array}{l}\text{hepatitis} \Rightarrow \text{yellow-eyes},\\ \text{jaundice} \Rightarrow \text{yellow-eyes}\end{array}$
- $\begin{array}{l}H_1 = \text{hepatitis},\\ H_2 = \text{jaundice}\end{array}$ are minimal hypotheses;

# Example of Diagnosis— still a crash course

- symptoms: yellow-eyes
- theory: hepatitis $\Rightarrow$ yellow-eyes,
  jaundice $\Rightarrow$ yellow-eyes
- $H_1 =$ hepatitis,
  $H_2 =$ jaundice
  are minimal hypotheses;
- $H_3 =$ hepatitis $\wedge$ jaundice is not minimal

# Example of Diagnosis— still a crash course

- symptoms: yellow-eyes
- theory: hepatitis $\Rightarrow$ yellow-eyes,<br>jaundice $\Rightarrow$ yellow-eyes
- $H_1 =$ hepatitis, $H_2 =$ jaundice are minimal hypotheses;
- $H_3 =$ hepatitis $\wedge$ jaundice is not minimal

Lots of minimality criteria!

[Eiter and Gottlob, 1995]

# Why Abduction for P2P EC— Intuition

| Request(buyer) | Offers (sellers) |
|---|---|
| | $O_1 = \boxed{\text{cellphone}}$ |
| $R = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$ | $O_2 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth}\end{array}}$ |
| | $O_3 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$ |

# Why Abduction for P2P EC— Intuition

| Request(buyer) | Offers (sellers) |
|---|---|
|  | $O_1 = \boxed{\text{cellphone}}$ |
| $R = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$ | $O_2 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth}\end{array}}$ |
| $\mathbf{H_3}=\{\}$ $\nwarrow$ | $O_3 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$ |

# Why Abduction for P2P EC— Intuition

| Request(buyer) | | Offers (sellers) |
|---|---|---|
| | | $O_1 = \boxed{\text{cellphone}}$ |
| $R = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$ | $\mathbf{H_2{=}\{s\}} \leftarrow$ | $O_2 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth}\end{array}}$ |
| | | $O_3 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$ |

# Why Abduction for P2P EC — Intuition

Request(buyer) Offers (sellers)

---

$O_1 = \boxed{\text{cellphone}}$

$\mathbf{H_1 = \{b,s\}} \swarrow$

$R = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$

$O_2 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth}\end{array}}$

$O_3 = \boxed{\begin{array}{l}\text{cellphone,}\\ \text{bluetooth,}\\ \text{shutdown}\end{array}}$

# Propositional Abduction 4 P2P EC

- Let $\mathcal{L}$ be a propositional language
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology

# Propositional Abduction 4 P2P EC

- Let $\mathcal{L}$ be a propositional language
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology

---

- find a hypothesis $H \in \mathcal{H}$ such that

# Propositional Abduction 4 P2P EC

- Let $\mathcal{L}$ be a propositional language
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology

---

- find a hypothesis $H \in \mathcal{H}$ such that
  - $H \wedge O$ is satisfiable in $\mathcal{T}$

# Propositional Abduction 4 P2P EC

- Let $\mathcal{L}$ be a propositional language
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a domain ontology

---

- find a hypothesis $H \in \mathcal{H}$ such that
  - $H \wedge O$ is satisfiable in $\mathcal{T}$
  - $\mathcal{T} \models H \wedge O \Rightarrow R$

# Abd. 4 EC vs. Abd. 4 Diagnosis

- both have a background theory $\mathcal{T}$

# Abd. 4 EC vs. Abd. 4 Diagnosis

- both have a background theory $\mathcal{T}$
- both have a target conclusion (symptoms, request)

# Abd. 4 EC vs. Abd. 4 Diagnosis

- both have a background theory $\mathcal{T}$
- both have a target conclusion (symptoms, request)
- Abd. 4 EC has also *prior facts* (Offer $O$)

# Abd. 4 EC vs. Abd. 4 Diagnosis

- both have a background theory $\mathcal{T}$
- both have a target conclusion (symptoms, request)
- Abd. 4 EC has also *prior facts* (Offer $O$)
- Abd. 4 EC includes Abd. 4 Diag. when $O \equiv \top$ (no prior facts)

# Intuition

- When $R$ evaluates its possible transaction with $O$, before concluding the transaction, $R$ and $O$ should agree on $H$

# Intuition

- When $R$ evaluates its possible transaction with $O$, before concluding the transaction, $R$ and $O$ should agree on $H$
- will $O$ accept $H$?

# Intuition

- When $R$ evaluates its possible transaction with $O$, before concluding the transaction, $R$ and $O$ should agree on $H$
- will $O$ accept $H$?
- *vice versa* for $O$, with a different $H'$ such that $\mathcal{T} \models R \wedge H' \Rightarrow O$

# Intuition

- When $R$ evaluates its possible transaction with $O$, before concluding the transaction, $R$ and $O$ should agree on $H$
- will $O$ accept $H$?
- *vice versa* for $O$, with a different $H'$ such that $\mathcal{T} \models R \wedge H' \Rightarrow O$
  - works also exchanging $O$ and $R$!

# Intuition

- When $R$ evaluates its possible transaction with $O$, before concluding the transaction, $R$ and $O$ should agree on $H$
- will $O$ accept $H$?
- *vice versa* for $O$, with a different $H'$ such that $\mathcal{T} \models R \wedge H' \Rightarrow O$
  - works also exchanging $O$ and $R$!
  - Preference queries [Kießling, 2002] do not.

# What Abduction is good for?

- compute a *score* for each counteroffer

# What Abduction is good for?

- compute a *score* for each counteroffer
  - *e.g.*, number of hypotheses in best $H$

# What Abduction is good for?

- compute a *score* for each counteroffer
  - *e.g.*, number of hypotheses in best $H$
  - *e.g.*, probability/plausibility of $H$

# What Abduction is good for?

- compute a *score* for each counteroffer
  - *e.g.*, number of hypotheses in best $H$
  - *e.g.*, probability/plausibility of $H$
- construct an *explanation* for match suggestions

# What Abduction is good for?

- compute a *score* for each counteroffer
    - *e.g.*, number of hypotheses in best $H$
    - *e.g.*, probability/plausibility of $H$
- construct an *explanation* for match suggestions
    - *e.g.*, a facilitator that suggests
      "Offer 213 seems to be the best, but requests *color:blue* and *Credit Card Payment* are not yet assessed"

# Concept Abduction

[Di Noia *et al.*, 2007]: Let. . .

- $\mathcal{L}$ be a Description Logic
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

# Concept Abduction

[Di Noia *et al.*, 2007]: Let. . .

- $\mathcal{L}$ be a Description Logic
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

---

- *find* a hypothesis $H \in \mathcal{H}$ such that

# Concept Abduction

[Di Noia *et al.*, 2007]: Let. . .

- $\mathcal{L}$ be a Description Logic
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

---

- *find* a hypothesis $H \in \mathcal{H}$ such that
  - $H \sqcap O$ is satisfiable in $\mathcal{T}$

# Concept Abduction

[Di Noia *et al.*, 2007]: Let...

- $\mathcal{L}$ be a Description Logic
- $\mathcal{H}$ be a subset of $\mathcal{L}$ (possible hypotheses)
- $R$ a Request, $O$ a possible Offer for $R$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

---

- *find* a hypothesis $H \in \mathcal{H}$ such that
  - $H \sqcap O$ is satisfiable in $\mathcal{T}$
  - $H \sqcap O \sqsubseteq_{\mathcal{T}} R$

# Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set

# Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set
- *subsumption-maximal* $H$ — least committing see also [Cialdea Mayer and Pirri, 1995]

# Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set
- *subsumption-maximal* $H$ — least committing see also [Cialdea Mayer and Pirri, 1995]
  - no other abduction $H'$ is s.t. $H \sqsubseteq_{\mathcal{T}} H'$ and $H \not\equiv H'$

# Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set
- *subsumption-maximal* $H$ — least committing see also [Cialdea Mayer and Pirri, 1995]
    - no other abduction $H'$ is s.t. $H \sqsubseteq_{\mathcal{T}} H'$ and $H \not\equiv H'$
- language-specific

# Best hypotheses

Different criteria:

- *shortest* $H$ — fewer issues to be set
- *subsumption-maximal* $H$ — least committing see also [Cialdea Mayer and Pirri, 1995]
  - no other abduction $H'$ is s.t. $H \sqsubseteq_{\mathcal{T}} H'$ and $H \not\equiv H'$
- language-specific
  - *e.g.*, minimal conjunctions if $\sqcup, \neg \notin \mathcal{H}$

# Comparing criteria

- $R = \mathit{FiatPanda} \sqcap \mathit{radio} \sqcap \mathit{fogLmps}$

# Comparing criteria

- $R = \mathit{FiatPanda} \sqcap \mathit{radio} \sqcap \mathit{fogLmps}$
  - “I want a used Fiat Panda, with fog lamps”

# Comparing criteria

- $R = \mathit{FiatPanda} \sqcap \mathit{radio} \sqcap \mathit{fogLmps}$
  - “I want a used Fiat Panda, with fog lamps”
- $O = \mathit{FiatPanda} \sqcap \mathit{yr2000}$

# Comparing criteria

- $R = \mathit{FiatPanda} \sqcap \mathit{radio} \sqcap \mathit{fogLmps}$
    - “I want a used Fiat Panda, with fog lamps”
- $O = \mathit{FiatPanda} \sqcap \mathit{yr2000}$
- $\mathcal{T} = \{\mathit{bundleOff} \sqsubseteq \mathit{radio} \sqcap \mathit{fogLmps} \sqcap \mathit{alarm}\}$

# Comparing criteria

- $R = \mathit{FiatPanda} \sqcap \mathit{radio} \sqcap \mathit{fogLmps}$
  - “I want a used Fiat Panda, with fog lamps”
- $O = \mathit{FiatPanda} \sqcap \mathit{yr2000}$
- $\mathcal{T} = \{\mathit{bundleOff} \sqsubseteq \mathit{radio} \sqcap \mathit{fogLmps} \sqcap \mathit{alarm}\}$
- $H_1 = \mathit{radio} \sqcap \mathit{fogLmps}$ is *subsumption-maximal*

# Comparing criteria

- $R = \mathit{FiatPanda} \sqcap \mathit{radio} \sqcap \mathit{fogLmps}$
  - “I want a used Fiat Panda, with fog lamps”
- $O = \mathit{FiatPanda} \sqcap \mathit{yr2000}$
- $\mathcal{T} = \{\mathit{bundleOff} \sqsubseteq \mathit{radio} \sqcap \mathit{fogLmps} \sqcap \mathit{alarm}\}$
- $H_1 = \mathit{radio} \sqcap \mathit{fogLmps}$ is *subsumption-maximal*
- $H_2 = \mathit{bundleOff}$ has *minimum length*

# Comparing criteria

- $R = \mathit{FiatPanda} \sqcap \mathit{radio} \sqcap \mathit{fogLmps}$
    - “I want a used Fiat Panda, with fog lamps”
- $O = \mathit{FiatPanda} \sqcap \mathit{yr2000}$
- $\mathcal{T} = \{\mathit{bundleOff} \sqsubseteq \mathit{radio} \sqcap \mathit{fogLmps} \sqcap \mathit{alarm}\}$
- $H_1 = \mathit{radio} \sqcap \mathit{fogLmps}$ is *subsumption-maximal*
- $H_2 = \mathit{bundleOff}$ has *minimum length*
- neither solution is in the other set

# Intermezzo 1

## — Do you need a coffee?

# Belief Revision (history)

- [Gärdenfors, 1988]:
  Revise Knowledge $\mathcal{K}$ with new info $A$ by:
  1. *contracting* $\mathcal{K}$ into $\mathcal{K}^{-}_{\neg A}$ such that $\mathcal{K}^{-}_{\neg A} \not\models \neg A$
  2. *adding* $A$ to $\mathcal{K}^{-}_{\neg A}$
- Intuition: contract the least

# Belief Revision— another crash course!

$$\mathcal{K} = \left\{ \begin{array}{l} \mathit{1stFloor} \wedge \mathit{noSteps} \Rightarrow \mathit{easyAccess} \\ \mathit{1stFloor} \wedge \mathit{noSteps} \end{array} \right\}$$

# Belief Revision— another crash course!

$$\mathcal{K} = \left\{ \begin{array}{l} \mathit{1stFloor} \wedge \mathit{noSteps} \Rightarrow \mathit{easyAccess} \\ \mathit{1stFloor} \wedge \mathit{noSteps} \end{array} \right\}$$

new information: $\neg \mathit{easyAccess}$

# Belief Revision — another crash course!

$$\mathcal{K} = \left\{ \begin{array}{l} \mathit{1stFloor} \wedge \mathit{noSteps} \Rightarrow \mathit{easyAccess} \\ \mathit{1stFloor} \wedge \mathit{noSteps} \end{array} \right\}$$

new information: $\neg \mathit{easyAccess}$

*Syntax-based* revisions:

$\mathcal{K}_1^- = \{\mathit{1stFloor} \wedge \mathit{noSteps} \Rightarrow \mathit{easyAccess}\}$

# Belief Revision— another crash course!

$$\mathcal{K} = \left\{ \begin{array}{l} \mathit{1stFloor} \wedge \mathit{noSteps} \Rightarrow \mathit{easyAccess} \\ \mathit{1stFloor} \wedge \mathit{noSteps} \end{array} \right\}$$

new information: $\neg \mathit{easyAccess}$

*Syntax-based* revisions:

$\mathcal{K}_1^- = \{\mathit{1stFloor} \wedge \mathit{noSteps} \Rightarrow \mathit{easyAccess}\}$

$\mathcal{K}_2^- = \{\mathit{1stFloor} \wedge \mathit{noSteps}\}$ $\boxed{!!}$

# Belief Revision — another crash course!

$$\mathcal{K} = \left\{ \begin{array}{l} \mathit{1stFloor} \land \mathit{noSteps} \Rightarrow \mathit{easyAccess} \\ \mathit{1stFloor} \land \mathit{noSteps} \end{array} \right\}$$

new information: $\neg \mathit{easyAccess}$

*Syntax-based* revisions:

$\mathcal{K}_1^- = \{\mathit{1stFloor} \land \mathit{noSteps} \Rightarrow \mathit{easyAccess}\}$

$\mathcal{K}_2^- = \{\mathit{1stFloor} \land \mathit{noSteps}\}$ $\boxed{!!}$

A *syntax-independent* revision:

$$\mathcal{K}_3^- = \left\{ \begin{array}{l} \mathit{1stFloor} \land \mathit{noSteps} \Rightarrow \mathit{easyAccess} \\ \mathit{1stFloor} \end{array} \right\}$$

# Concept Contraction

- Let $\mathcal{L}$ be a Description Logic
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

# Concept Contraction

- Let $\mathcal{L}$ be a Description Logic
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

---

- find a pair $\langle G, K \rangle$ ($G$ive up, $K$eep) such that

# Concept Contraction

- Let $\mathcal{L}$ be a Description Logic
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

---

- find a pair $\langle G, K \rangle$ ($G$ive up, $K$eep) such that
  - $\mathcal{T} \models R \equiv G \sqcap K$

# Concept Contraction

- Let $\mathcal{L}$ be a Description Logic
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

---

- find a pair $\langle G, K \rangle$ ($G$ive up, $K$eep) such that
  - $\mathcal{T} \models R \equiv G \sqcap K$
  - $O \sqcap K$ is satisfiable in $\mathcal{T}$

# Concept Contraction

- Let $\mathcal{L}$ be a Description Logic
- $R$ a Request, $O$ a possible Offer for $R$ in $\mathcal{L}$
- $\mathcal{T}$ be a TBox in $\mathcal{L}$

---

- find a pair $\langle G, K \rangle$ ($G$ive up, $K$eep) such that
  - $\mathcal{T} \models R \equiv G \sqcap K$
  - $O \sqcap K$ is satisfiable in $\mathcal{T}$
- $\langle G, K \rangle$ is a *contraction* of $R$ w.r.t. $O$

# Contr. 4 EC vs. Contr. 4 Revision

- only Contr. 4 EC has a background theory $\mathcal{T}$ (non-revisable)

# Contr. 4 EC vs. Contr. 4 Revision

- only Contr. 4 EC has a background theory $\mathcal{T}$ (non-revisable)
- both deal with an unwanted contradiction

# Contr. 4 EC vs. Contr. 4 Revision

- only Contr. 4 EC has a background theory $\mathcal{T}$ (non-revisable)
- both deal with an unwanted contradiction
- Contr. 4 EC has also *prior facts* (Offer $O$)

# Contr. 4 EC vs. Contr. 4 Revision

- only Contr. 4 EC has a background theory $\mathcal{T}$ (non-revisable)
- both deal with an unwanted contradiction
- Contr. 4 EC has also *prior facts* (Offer $O$)
- Contr. 4 EC includes Contr. 4 Rev. when $O \equiv \top$ (no prior facts)

# Best contractions

Different criteria:

- *shortest* $G$ — fewer issues to give up

# Best contractions

Different criteria:

- *shortest* $G$ — fewer issues to give up
- *subsumption-maximal* $G$ — minimal consequences

# Best contractions

Different criteria:

- *shortest* $G$ — fewer issues to give up
- *subsumption-maximal* $G$ — minimal consequences
    - no other contraction $\langle G', K' \rangle$ is s.t. $G \sqsubseteq_{\mathcal{T}} G'$ and $G \not\equiv G'$

# Best contractions

Different criteria:

- *shortest* $G$ — fewer issues to give up
- *subsumption-maximal* $G$ — minimal consequences
  - no other contraction $\langle G', K' \rangle$ is s.t. $G \sqsubseteq_{\mathcal{T}} G'$ and $G \not\equiv G'$
- subsumption-minimal $K$ [Calì *et al.*, 2004]

# Best contractions

Different criteria:

- *shortest* $G$ — fewer issues to give up
- *subsumption-maximal* $G$ — minimal consequences
    - no other contraction $\langle G', K' \rangle$ is s.t. $G \sqsubseteq_{\mathcal{T}} G'$ and $G \not\equiv G'$
- subsumption-minimal $K$ [Calì *et al.*, 2004]
    - no other contraction $\langle G', K' \rangle$ is s.t. $K' \sqsubseteq_{\mathcal{T}} K$ and $K \not\equiv K'$

# Example

- $R = \textit{flat} \sqcap \textit{easyAccess}$

# Example

- $R = \mathit{flat} \sqcap \mathit{easyAccess}$
  - “I want a flat with easy access”

# Example

- $R = \textit{flat} \sqcap \textit{easyAccess}$
  - “I want a flat with easy access”
- $O = \textit{1stFloor} \sqcap \textit{lastFloor} \sqcap \textit{garden}$

# Example

- $R = \mathit{flat} \sqcap \mathit{easyAccess}$
  - “I want a flat with easy access”
- $O = \mathit{1stFloor} \sqcap \mathit{lastFloor} \sqcap \mathit{garden}$
- $\mathcal{T} =$

$$\left\{ \begin{array}{rcl} \mathit{1stFloor} \sqcap \mathit{lastFloor} & \equiv & \mathit{house} \\ \mathit{flat} & \equiv & \neg \mathit{house} \\ \mathit{lift} \sqcup \mathit{1stFloor} & \sqsubseteq & \mathit{easyAccess} \end{array} \right\}$$

# Example

- $R = \mathit{flat} \sqcap \mathit{easyAccess}$
  - “I want a flat with easy access”
- $O = \mathit{1stFloor} \sqcap \mathit{lastFloor} \sqcap \mathit{garden}$
- $\mathcal{T} =$
$$\left\{ \begin{array}{rcl} \mathit{1stFloor} \sqcap \mathit{lastFloor} & \equiv & \mathit{house} \\ \mathit{flat} & \equiv & \neg\mathit{house} \\ \mathit{lift} \sqcup \mathit{1stFloor} & \sqsubseteq & \mathit{easyAccess} \end{array} \right\}$$
- $G = \mathit{flat}$

# Example

- $R = \textit{flat} \sqcap \textit{easyAccess}$
  - “I want a flat with easy access”
- $O = \textit{1stFloor} \sqcap \textit{lastFloor} \sqcap \textit{garden}$
- $\mathcal{T} =$
$$\left\{ \begin{array}{rcl} \textit{1stFloor} \sqcap \textit{lastFloor} & \equiv & \textit{house} \\ \textit{flat} & \equiv & \neg\textit{house} \\ \textit{lift} \sqcup \textit{1stFloor} & \sqsubseteq & \textit{easyAccess} \end{array} \right\}$$
- $G = \textit{flat}$
- $K = \textit{easyAccess}$

# Logic-based ranking in P2P EC

- suppose a buyer enters the marketplace with request $R$

# Logic-based ranking in P2P EC

- suppose a buyer enters the marketplace with request $R$
- the facilitator ranks all offers $O_1, O_2, \ldots, O_n$ based on a pair of scores:

# Logic-based ranking in P2P EC

- suppose a buyer enters the marketplace with request $R$
- the facilitator ranks all offers $O_1, O_2, \ldots, O_n$ based on a pair of scores:
  - a score for a *best contraction* $\langle G, K \rangle$ of $R$ w.r.t. $O_i$

# Logic-based ranking in P2P EC

- suppose a buyer enters the marketplace with request $R$
- the facilitator ranks all offers $O_1, O_2, \ldots, O_n$ based on a pair of scores:
  - a score for a *best contraction* $\langle G, K \rangle$ of $R$ w.r.t. $O_i$
  - a score for a *best abduction* $H$ on $O_i$ w.r.t. $K$

# Logic-based ranking in P2P EC

- suppose a buyer enters the marketplace with request $R$
- the facilitator ranks all offers $O_1, O_2, \ldots, O_n$ based on a pair of scores:
  - a score for a *best contraction* $\langle G, K \rangle$ of $R$ w.r.t. $O_i$
  - a score for a *best abduction* $H$ on $O_i$ w.r.t. $K$
- *plus. . .*

# Logic-based ranking in P2P EC (2)

- the facilitator provides $G, K, H$ as an *explanation* of the rank

# Logic-based ranking in P2P EC (2)

- the facilitator provides $G, K, H$ as an *explanation* of the rank
- the buyer (resp. the seller) can *check* $G, K, H$

# Logic-based ranking in P2P EC (2)

- the facilitator provides $G, K, H$ as an *explanation* of the rank
- the buyer (resp. the seller) can *check* $G, K, H$
- *trust* in the mediator is increased

# Logic-based ranking in P2P EC (2)

- the facilitator provides $G, K, H$ as an *explanation* of the rank
- the buyer (resp. the seller) can *check* $G, K, H$
- *trust* in the mediator is increased
- . . . w.r.t. raw numbers

# Simple complexity results

When abducing $H?$ such that $C \sqcap H \sqsubseteq_{\mathcal{T}} D$,

$$H = \top \text{ iff } C \sqsubseteq_{\mathcal{T}} D \text{ already}$$

Note: $H = \top$ is both subsumption-maximal and minimum-length

Concept Abduction (every criteria) is at least as hard as Subsumption

# Simple complexity results (2)

- If $\mathcal{T}$ is at least Definite Horn-Krom (*i.e.*, $A \sqsubseteq B$ with $A, B$ concept *names*),

# Simple complexity results (2)

- If $\mathcal{T}$ is at least Definite Horn-Krom (*i.e.*, $A \sqsubseteq B$ with $A, B$ concept *names*),
- Min-length Concept Abduction includes MIN SET COVER [Reggia *et al.*, 1985]

# Simple complexity results (2)

- If $\mathcal{T}$ is at least Definite Horn-Krom (*i.e.*, $A \sqsubseteq B$ with $A, B$ concept *names*),
- Min-length Concept Abduction includes MIN SET COVER [Reggia *et al.*, 1985]

Min-Length Concept Abduction (every DL) is NP-hard for Definite Horn-Krom $\mathcal{T}$

[Colucci *et al.*, 2004]

# Simple complexity results (3)

When contracting $C \equiv_{\mathcal{T}} G \sqcap K$ such that $D \sqcap K$ is sat. w.r.t. $\mathcal{T}$,

$G = \top$ iff $C \sqcap D$ is already sat. w.r.t. $\mathcal{T}$

Note: $G = \top$ is both subsumption-maximal and minimum-length

Concept Contraction (every criteria) is at least as hard as Satisfiability

# Alternative Approaches

[Lukasiewicz and Schellhase, 2006]

- Variable-strength *preferences*

# Alternative Approaches

[Lukasiewicz and Schellhase, 2006]

- Variable-strength *preferences*
- syntax: $(\alpha > \beta|\phi)[x]$

# Alternative Approaches

[Lukasiewicz and Schellhase, 2006]

- Variable-strength *preferences*
- syntax: $(\alpha > \beta|\phi)[x]$
- formula $\alpha$ is preferred to formula $\beta$ in the context $\phi$ with weight $x \in \mathbb{N}$

# Alternative Approaches

[Lukasiewicz and Schellhase, 2006]

- Variable-strength *preferences*
- syntax: $(\alpha > \beta|\phi)[x]$
- formula $\alpha$ is preferred to formula $\beta$ in the context $\phi$ with weight $x \in \mathbb{N}$
- drawback: how to *elicit* preferences & numbers?

# Alternatives to Abduction

[Benatallah *et al.*, 2005]

- Computing the Concept Difference [Teege, 1994] $O - R$ (when $R \sqsubseteq O$)

# Alternatives to Abduction

[Benatallah *et al.*, 2005]

- Computing the Concept Difference [Teege, 1994] $O - R$ (when $R \sqsubseteq O$)
- choose the offer with minimal difference

# Alternatives to Abduction

[Benatallah *et al.*, 2005]

- Computing the Concept Difference [Teege, 1994] $O - R$ (when $R \sqsubseteq O$)
- choose the offer with minimal difference
- if $R \not\sqsubseteq_{\mathcal{T}} O$, use $R - LCS(R, O)$

# Alternatives to Abduction

[Benatallah *et al.*, 2005]

- Computing the Concept Difference [Teege, 1994] $O - R$ (when $R \sqsubseteq O$)
- choose the offer with minimal difference
- if $R \not\sqsubseteq_{\mathcal{T}} O$, use $R - LCS(R, O)$
- drawback: difference does not consider $\mathcal{T}$

# Alternatives to Abduction

[Benatallah *et al.*, 2005]

- Computing the Concept Difference [Teege, 1994] $O - R$ (when $R \sqsubseteq O$)
- choose the offer with minimal difference
- if $R \not\sqsubseteq_{\mathcal{T}} O$, use $R - LCS(R, O)$
- drawback: difference does not consider $\mathcal{T}$
- drawback: cannot cope with inconsistencies (no Contraction)

# Outline of the talk

- ✓ Motivation: Electronic Commerce
- ✓ Abduction and Contraction: Definitions
  - ✓ Logical and computational properties
- A Tableaux-based calculus
- Implementation
- What next?

# Prefixed Tableaux

## ∼ for ∼

## Concept Abduction and Contraction

# Basic elements

- labels: **T** (true), **F** (false)

# Basic elements

- labels: **T** (true), **F** (false)
- prefixes: $1$, $1R2$, $1R3S4Q8$,. . .
  [Donini and Massacci, 2000]

# Basic elements

- labels: **T** (true), **F** (false)
- prefixes: $1$, $1R2$, $1R3S4Q8$,. . .
  [Donini and Massacci, 2000]
- let $x$ be a prefix, $n$ be an integer

# Basic elements

- labels: **T** (true), **F** (false)
- prefixes: $1$, $1R2$, $1R3S4Q8$,. . . [Donini and Massacci, 2000]
- let $x$ be a prefix, $n$ be an integer
  - concept formulas: $\begin{cases} \textbf{T}) \; x : C \\ \textbf{F}) \; x : C \end{cases}$

# Basic elements

- labels: **T** (true), **F** (false)
- prefixes: $1$, $1R2$, $1R3S4Q8$,. . .
  [Donini and Massacci, 2000]
- let $x$ be a prefix, $n$ be an integer
  - concept formulas: $\begin{cases} \textbf{T}) \; x : C \\ \textbf{F}) \; x : C \end{cases}$
  - role formulas: $\begin{cases} \textbf{T})(x, xRn) : R \\ \textbf{F})(x, xRn) : \neg R \end{cases}$

# Assumptions

- $\mathcal{L}$ = $\mathcal{AL}$ (can be extended to $\mathcal{ALN}$)
- concepts are in Normal Form [Borgida and Patel-Schneider, 1994] (NF)
- $\mathcal{T}$ is normalized:
  $A \sqsubseteq B \sqcap C \quad \rightarrow \quad A \sqsubseteq B, A \sqsubseteq C$

# Tableaux Rules: $\sqcap$, $\sqcup$

$$\frac{\textbf{T})\ x : C \sqcap D}{\begin{array}{c}\textbf{T})\ x : C\\ \textbf{T})\ x : D\end{array}}\ \textbf{T}\sqcap \qquad\qquad \frac{\textbf{F})\ x : C \sqcup D}{\begin{array}{c}\textbf{F})\ x : C\\ \textbf{F})\ x : D\end{array}}\ \textbf{F}\sqcup$$

# Tableaux Rules: $\sqcap$, $\sqcup$

$$\frac{\textbf{T})\ x : C \sqcap D}{\begin{array}{l}\textbf{T})\ x : C\\ \textbf{T})\ x : D\end{array}}\ \textbf{T}\sqcap \qquad\qquad \frac{\textbf{F})\ x : C \sqcup D}{\begin{array}{l}\textbf{F})\ x : C\\ \textbf{F})\ x : D\end{array}}\ \textbf{F}\sqcup$$

$$\frac{\textbf{T})\ x : C \sqcup D}{\textbf{T})\ x : C \qquad \textbf{T})\ x : D}\ \textbf{T}\sqcup \qquad\qquad \frac{\textbf{F})\ x : C \sqcap D}{\textbf{F})\ x : C \qquad \textbf{F})\ x : D}\ \textbf{F}\sqcap$$

(2 branches) (2 branches)

# Tableaux Rules: $\forall$

$$\frac{\mathbf{T})\ x : \forall R.C \qquad \mathbf{T})(x, xRn) : R}{\mathbf{T})\ xRn : C}\ \mathbf{T}\forall_1$$

$$\frac{\mathbf{F})\ x : \exists R.C \qquad \mathbf{T})(x, xRn) : R}{\mathbf{F})\ xRn : C}\ \mathbf{F}\exists_1$$

# Tableaux Rules: $\forall$

$$\frac{\mathsf{T})\ x : \forall R.C \qquad \mathsf{T})(x, xRn) : R}{\mathsf{T})\ xRn : C}\ \mathsf{T}\forall_1 \qquad \frac{\mathsf{F})\ x : \exists R.C \qquad \mathsf{T})(x, xRn) : R}{\mathsf{F})\ xRn : C}\ \mathsf{F}\exists_1$$

$$\frac{\mathsf{T})\ x : \forall R.C \qquad \mathsf{F})(x, xRn) : \neg R}{\mathsf{T})\ xRn : C}\ \mathsf{T}\forall_2 \qquad \frac{\mathsf{F})\ x : \exists R.C \qquad \mathsf{F})(x, xRn) : \neg R}{\mathsf{F})\ xRn : C}\ \mathsf{F}\exists_2$$

# Tableaux Rules: $\forall$

$$\frac{\textbf{T})\ x : \forall R.C \qquad \textbf{T})(x, xRn) : R}{\textbf{T})\ xRn : C}\ \textbf{T}\forall_1 \qquad \frac{\textbf{F})\ x : \exists R.C \qquad \textbf{T})(x, xRn) : R}{\textbf{F})\ xRn : C}\ \textbf{F}\exists_1$$

$$\frac{\textbf{T})\ x : \forall R.C \qquad \textbf{F})(x, xRn) : \neg R}{\textbf{T})\ xRn : C}\ \textbf{T}\forall_2 \qquad \frac{\textbf{F})\ x : \exists R.C \qquad \textbf{F})(x, xRn) : \neg R}{\textbf{F})\ xRn : C}\ \textbf{F}\exists_2$$

Note: the label of the *concept formula* carries over!

# Tableaux Rules: $\exists$

# Tableaux Rules: $\exists$

$$\frac{\mathbf{T})\ x : \exists R.\top}{\mathbf{T})(x, xRn) : R}\ \mathbf{T}\exists$$

$$\frac{\mathbf{F})\ x : \forall R.C}{\begin{array}{c}\mathbf{F})(x, xRn) : \neg R \\ \mathbf{F})\ xRn : C\end{array}}\ \mathbf{F}\forall$$

# Tableaux Rules: $\exists$

$$\frac{\mathbf{T})\ x : \exists R.\top}{\mathbf{T})(x, xRn) : R}\ \mathbf{T}\exists \qquad\qquad \frac{\mathbf{F})\ x : \forall R.C}{\begin{array}{c}\mathbf{F})(x, xRn) : \neg R\\ \mathbf{F})\ xRn : C\end{array}}\ \mathbf{F}\forall$$

where $xRn$ is *new* in the branch / in the tableau

# Tableaux Rules: inclusions

$$\frac{A \sqsubseteq C \in \mathcal{T}}{\textsf{F}) \; x : A \sqcap NF(\neg C)} \; \textsf{F} \sqsubseteq$$

# Tableaux Rules: inclusions

$$\frac{A \sqsubseteq C \in \mathcal{T}}{\mathbf{F})\ x : A \sqcap NF(\neg C)}\ \mathbf{F} \sqsubseteq$$

forward-chain form$^*$

$$\frac{\begin{array}{c}\mathbf{F})\ x : \neg A \\ A \sqsubseteq C \in \mathcal{T}\end{array}}{\mathbf{F})\ x : NF(\neg C)}\ \mathbf{F} \sqsubseteq_1$$

backward-chain form$^*$

$$\frac{\begin{array}{c}\mathbf{F})\ x : C \\ A \sqsubseteq C \in \mathcal{T}\end{array}}{\mathbf{F})\ x : A}\ \mathbf{F} \sqsubseteq_2$$

# Tableaux Rules: inclusions

$$\frac{A \sqsubseteq C \in \mathcal{T}}{\textbf{F})\ x : A \sqcap NF(\neg C)}\ \textbf{F}\sqsubseteq$$

forward-chain form$^*$

$$\frac{\begin{array}{c}\textbf{F})\ x : \neg A \\ A \sqsubseteq C \in \mathcal{T}\end{array}}{\textbf{F})\ x : NF(\neg C)}\ \textbf{F}\sqsubseteq_1$$

backward-chain form$^*$

$$\frac{\begin{array}{c}\textbf{F})\ x : C \\ A \sqsubseteq C \in \mathcal{T}\end{array}}{\textbf{F})\ x : A}\ \textbf{F}\sqsubseteq_2$$

(*) incomplete—useful for optimizations only

# Note an absence. . .

Rules converting **T**) into **F**) and vice versa [Smullyan, 1968]

# Note an absence. . .

Rules converting **T**) into **F**) and vice versa [Smullyan, 1968]

$$\frac{\textbf{T})\ x : \neg C}{\textbf{F})\ x : C}\ \textbf{T}\neg \qquad\qquad \frac{\textbf{F})\ x : \neg C}{\textbf{T})\ x : C}\ \textbf{F}\neg$$

are *not* present!

# Note an absence...

Rules converting **T**) into **F**) and vice versa [Smullyan, 1968]

$$\frac{\textbf{T})\ x : \neg C}{\textbf{F})\ x : C}\ \textbf{T}\neg \qquad\qquad \frac{\textbf{F})\ x : \neg C}{\textbf{T})\ x : C}\ \textbf{F}\neg$$

are *not* present!

(we need to trace a formula back to either $O$ or $R$)

# Clashes in a branch

Let $A$ be a concept name, $\exists R$, $\forall R.\bot$

- *homogeneous* clash:

# Clashes in a branch

Let $A$ be a concept name, $\exists R$, $\forall R.\bot$

- *homogeneous* clash:

$\otimes_{\mathbf{TT}}$: either $\{\mathbf{T})\ x:\bot\}$ or $\{\mathbf{T})\ x:A, \mathbf{T})\ x:\neg A\}$

# Clashes in a branch

Let $A$ be a concept name, $\exists R$, $\forall R.\bot$

- *homogeneous* clash:

$\otimes_{\textbf{TT}}$: either $\{\textsf{\textbf{T}})\ x : \bot\}$ or $\{\textsf{\textbf{T}})\ x : A, \textsf{\textbf{T}})\ x : \neg A\}$

$\otimes_{\textbf{FF}}$: either $\{\textsf{\textbf{F}})\ x : \top\}$ or $\{\textsf{\textbf{F}})\ x : A, \textsf{\textbf{F}})\ x : \neg A\}$

# Clashes in a branch

Let $A$ be a concept name, $\exists R$, $\forall R.\bot$

- *homogeneous* clash:

$\otimes_{\mathbf{TT}}$: either $\{\mathbf{T})\ x : \bot\}$ or $\{\mathbf{T})\ x : A, \mathbf{T})\ x : \neg A\}$

$\otimes_{\mathbf{FF}}$: either $\{\mathbf{F})\ x : \top\}$ or $\{\mathbf{F})\ x : A, \mathbf{F})\ x : \neg A\}$

- *heterogeneous* clash:

# Clashes in a branch

Let $A$ be a concept name, $\exists R$, $\forall R.\bot$

- *homogeneous* clash:

$\otimes_{\mathbf{TT}}$: either $\{\mathbf{T})\ x : \bot\}$ or $\{\mathbf{T})\ x : A, \mathbf{T})\ x : \neg A\}$

$\otimes_{\mathbf{FF}}$: either $\{\mathbf{F})\ x : \top\}$ or $\{\mathbf{F})\ x : A, \mathbf{F})\ x : \neg A\}$

- *heterogeneous* clash:

$\otimes_{\mathbf{TF}}$: $\{\mathbf{T})\ x : A, \mathbf{F})\ x : A\}$

# Clashes in a branch

Let $A$ be a concept name, $\exists R, \forall R.\bot$

- *homogeneous* clash:

$\otimes_{\mathbf{TT}}$: either $\{\mathbf{T})\ x : \bot\}$ or $\{\mathbf{T})\ x : A, \mathbf{T})\ x : \neg A\}$

$\otimes_{\mathbf{FF}}$: either $\{\mathbf{F})\ x : \top\}$ or $\{\mathbf{F})\ x : A, \mathbf{F})\ x : \neg A\}$

- *heterogeneous* clash:

$\otimes_{\mathbf{TF}}$: $\{\mathbf{T})\ x : A, \mathbf{F})\ x : A\}$

$\otimes_{\mathbf{TF}}$: $\{\mathbf{T})\ x : \neg A, \mathbf{F})\ x : \neg A\}$

# Clashes in a branch

Let $A$ be a concept name, $\exists R$, $\forall R.\bot$

- *homogeneous* clash:

**$\otimes_{\mathbf{TT}}$**: either $\{\mathbf{T})\ x : \bot\}$ or $\{\mathbf{T})\ x : A, \mathbf{T})\ x : \neg A\}$

**$\otimes_{\mathbf{FF}}$**: either $\{\mathbf{F})\ x : \top\}$ or $\{\mathbf{F})\ x : A, \mathbf{F})\ x : \neg A\}$

- *heterogeneous* clash:

**$\otimes_{\mathbf{TF}}$**: $\{\mathbf{T})\ x : A, \mathbf{F})\ x : A\}$

**$\otimes_{\mathbf{TF}}$**: $\{\mathbf{T})\ x : \neg A, \mathbf{F})\ x : \neg A\}$

- a branch is closed if it contains a clash

# Clashes in a branch

Let $A$ be a concept name, $\exists R$, $\forall R.\bot$

- *homogeneous* clash:

$\otimes_{\mathbf{TT}}$: either $\{\mathbf{T})\ x : \bot\}$ or $\{\mathbf{T})\ x : A, \mathbf{T})\ x : \neg A\}$

$\otimes_{\mathbf{FF}}$: either $\{\mathbf{F})\ x : \top\}$ or $\{\mathbf{F})\ x : A, \mathbf{F})\ x : \neg A\}$

- *heterogeneous* clash:

$\otimes_{\mathbf{TF}}$: $\{\mathbf{T})\ x : A, \mathbf{F})\ x : A\}$

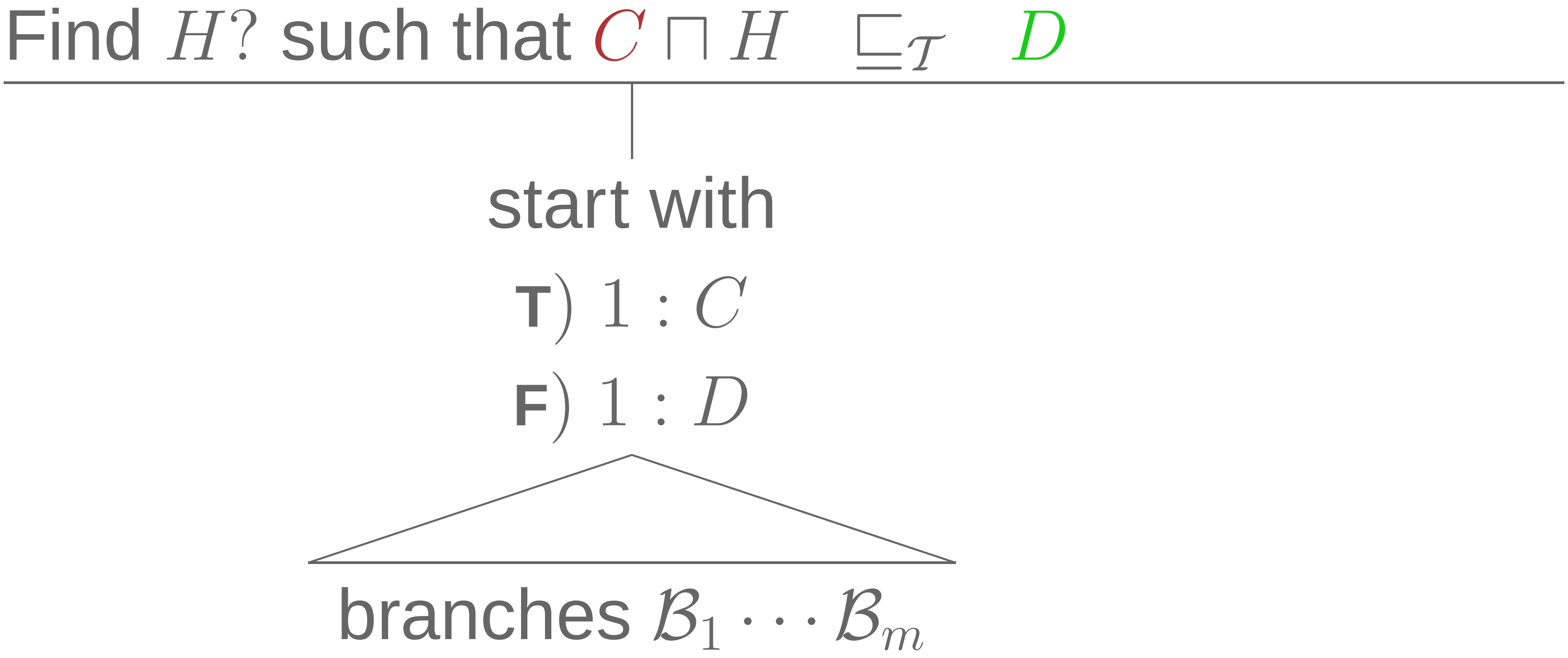
$\otimes_{\mathbf{TF}}$: $\{\mathbf{T})\ x : \neg A, \mathbf{F})\ x : \neg A\}$

- a branch is closed if it contains a clash
- a tableau is closed if every branch is closed

# Start for Abduction

Find $H?$ such that $C \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad D$

# Start for Abduction



Find $H?$ such that $C \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad D$

start with

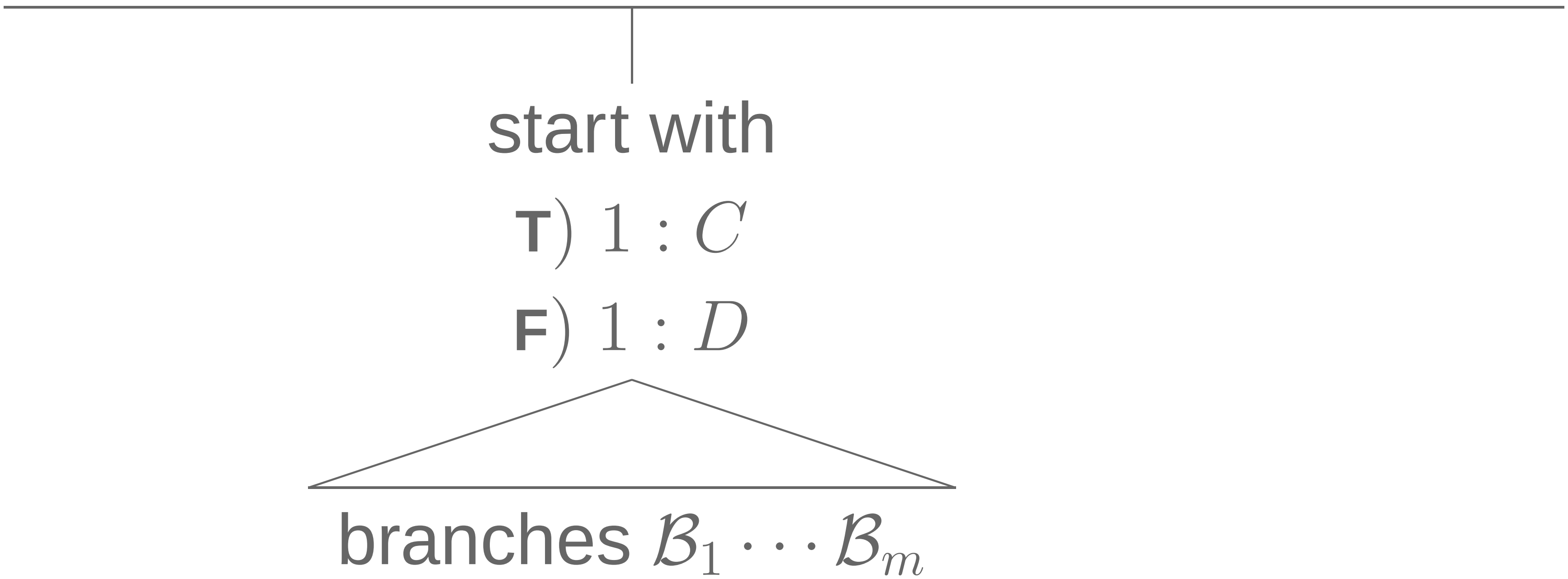
**T**) $1 : C$

**F**) $1 : D$

branches $\mathcal{B}_1 \cdots \mathcal{B}_m$

# Start for Abduction

Find $H?$ such that $C \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad D$



start with

**T**$)\ 1 : C$

**F**$)\ 1 : D$

branches $\mathcal{B}_1 \cdots \mathcal{B}_m$

for every completed, open branch $\mathcal{B}$, *add* all formulas $\boxed{\mathbf{T})\ x : A}$ *ab.* or $\boxed{\mathbf{T})\ x : \neg A}$ *ab.* that yield a *heterogeneous* clash

# Computing Abduction

- *choose* one abducible $\boxed{\textbf{T})\ x_i : E_i}$ *ab.* for each open branch $\mathcal{B}_i$, for $i = 1, \ldots, m$

# Computing Abduction

- *choose* one abducible $\boxed{\textbf{T})\ x_i : E_i}$ *ab.* for each open branch $\mathcal{B}_i$, for $i = 1, \ldots, m$
- let $roles(x) = R_1 \circ \cdots \circ R_k$ for $x = 1R_1n_1 \cdots R_kn_k$

# Computing Abduction

- *choose* one abducible $\boxed{\textbf{T}) \; x_i : E_i}$ *ab.* for each open branch $\mathcal{B}_i$, for $i = 1, \ldots, m$
- let $roles(x) = R_1 \circ \cdots \circ R_k$ for $x = 1R_1n_1 \cdots R_kn_k$
- *e.g.*, $roles(1R4Q6S9) = R \circ Q \circ S$

# Computing Abduction

- *choose* one abducible $\boxed{\textbf{T})\ x_i : E_i}$ *ab.* for each open branch $\mathcal{B}_i$, for $i = 1, \ldots, m$
- let $roles(x) = R_1 \circ \cdots \circ R_k$ for $x = 1R_1n_1 \cdots R_kn_k$
- *e.g.*, $roles(1R4Q6S9) = R \circ Q \circ S$
- let $H = \sqcap_i \forall roles(x_i).E_i$

# Computing Abduction

- *choose* one abducible $\boxed{\textbf{T})\ x_i : E_i}$ *ab.* for each open branch $\mathcal{B}_i$, for $i = 1, \ldots, m$
- let $roles(x) = R_1 \circ \cdots \circ R_k$ for $x = 1R_1n_1 \cdots R_kn_k$
- *e.g.*, $roles(1R4Q6S9) = R \circ Q \circ S$
- let $H = \sqcap_i \forall roles(x_i).E_i$
- several $H$'s, depending on the choice

# One open branch, one choice

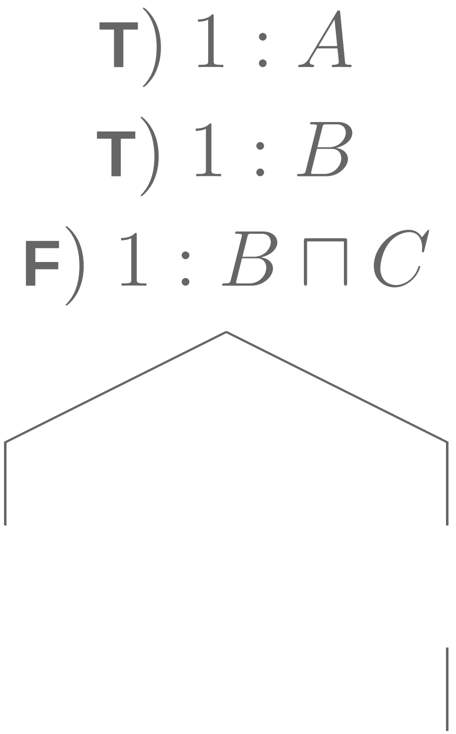
Find $H?$ such that $A \sqcap B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B \sqcap C$

---

# One open branch, one choice

Find $H?$ such that $A \sqcap B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B \sqcap C$

---



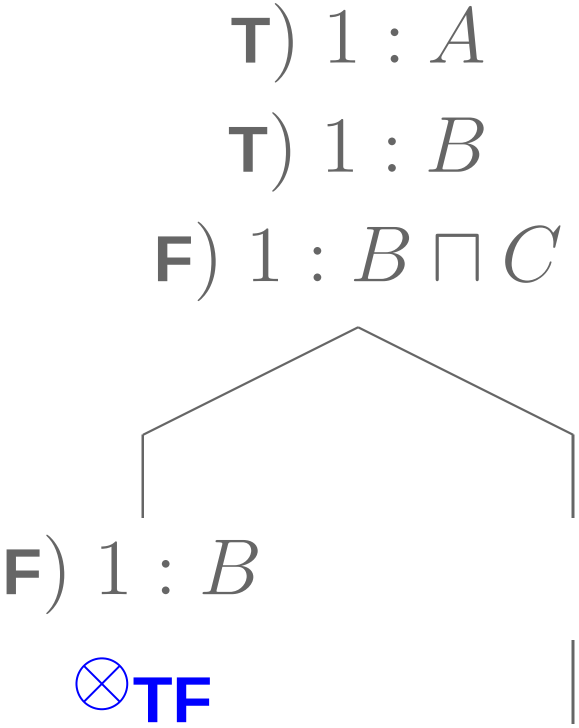
$\textbf{T}) \; 1 : A$

$\textbf{T}) \; 1 : B$

$\textbf{F}) \; 1 : B \sqcap C$

# One open branch, one choice

Find $H?$ such that $A \sqcap B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B \sqcap C$



**T**$) \; 1 : A$

**T**$) \; 1 : B$

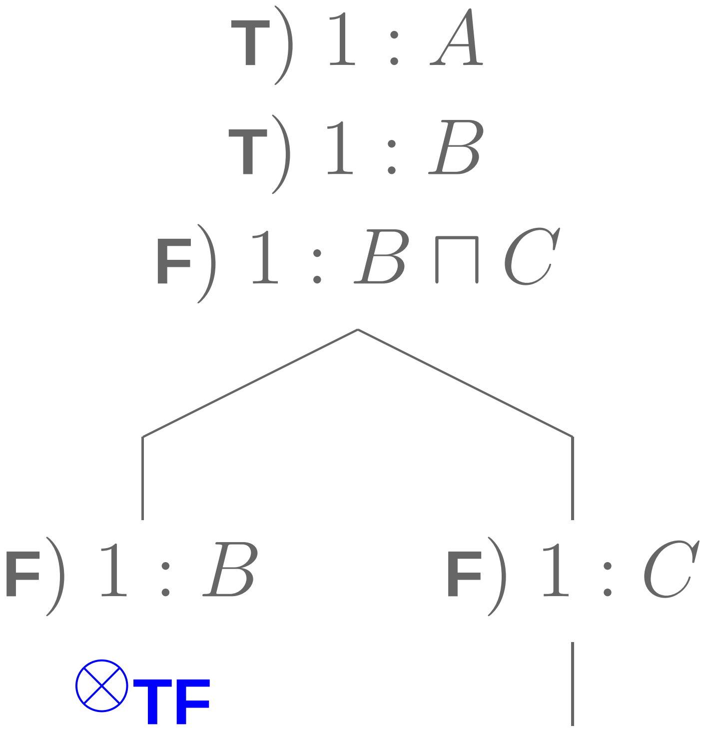
**F**$) \; 1 : B \sqcap C$

**F**$) \; 1 : B$

$\otimes$**TF**

# One open branch, one choice

Find $H?$ such that $A \sqcap B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B \sqcap C$



**T**) $1 : A$

**T**) $1 : B$

**F**) $1 : B \sqcap C$

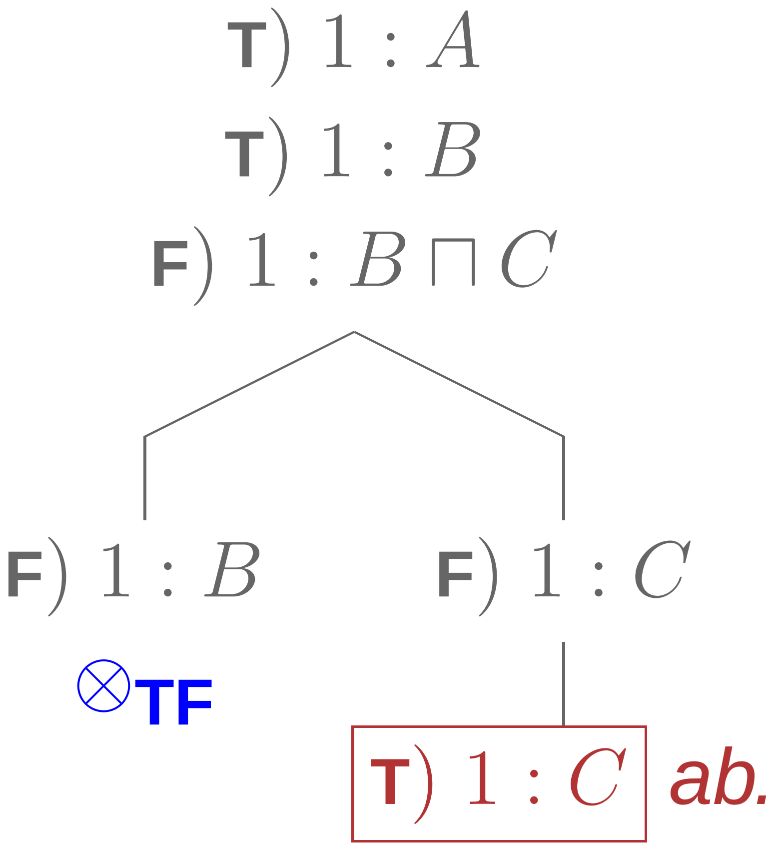
**F**) $1 : B$ **F**) $1 : C$

$\otimes$**TF**

# One open branch, one choice

Find $H?$ such that $A \sqcap B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B \sqcap C$

---



**T**) $1 : A$

**T**) $1 : B$

**F**) $1 : B \sqcap C$

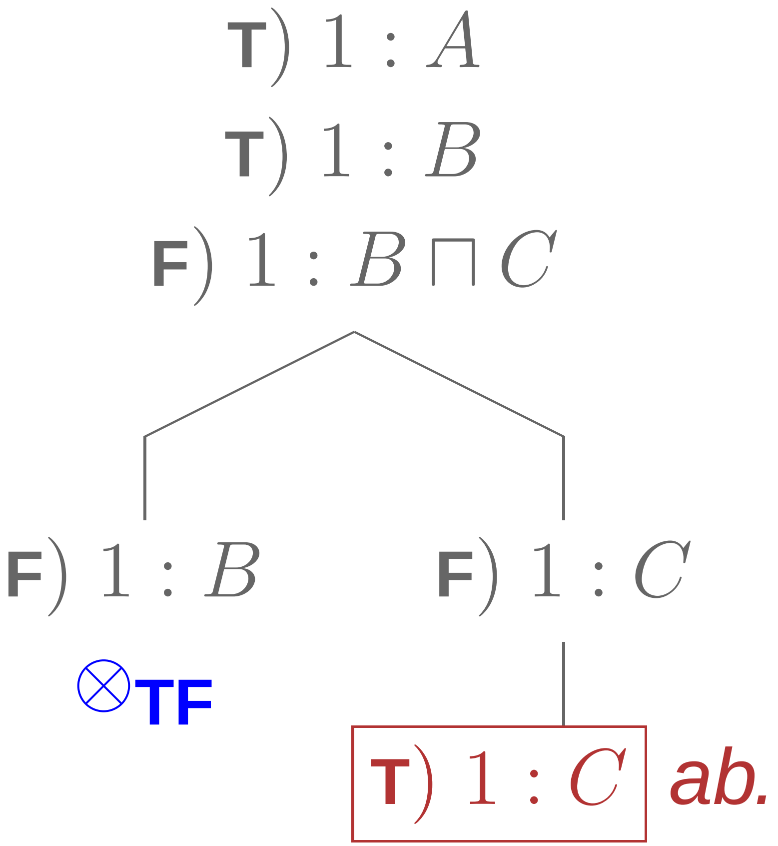
**F**) $1 : B$ $\otimes$**TF**

**F**) $1 : C$

$\boxed{\textbf{T}) \; 1 : C}$ *ab.*

# One open branch, one choice

Find $H$? such that $A \sqcap B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B \sqcap C$



**T**) $1 : A$

**T**) $1 : B$

**F**) $1 : B \sqcap C$

**F**) $1 : B$

$\otimes$**TF**

**F**) $1 : C$

$\boxed{\textbf{T}) \; 1 : C}$ *ab.*

One solution

$H =$

$= \forall roles(1).C$

$= \forall \epsilon.C$

$= C$

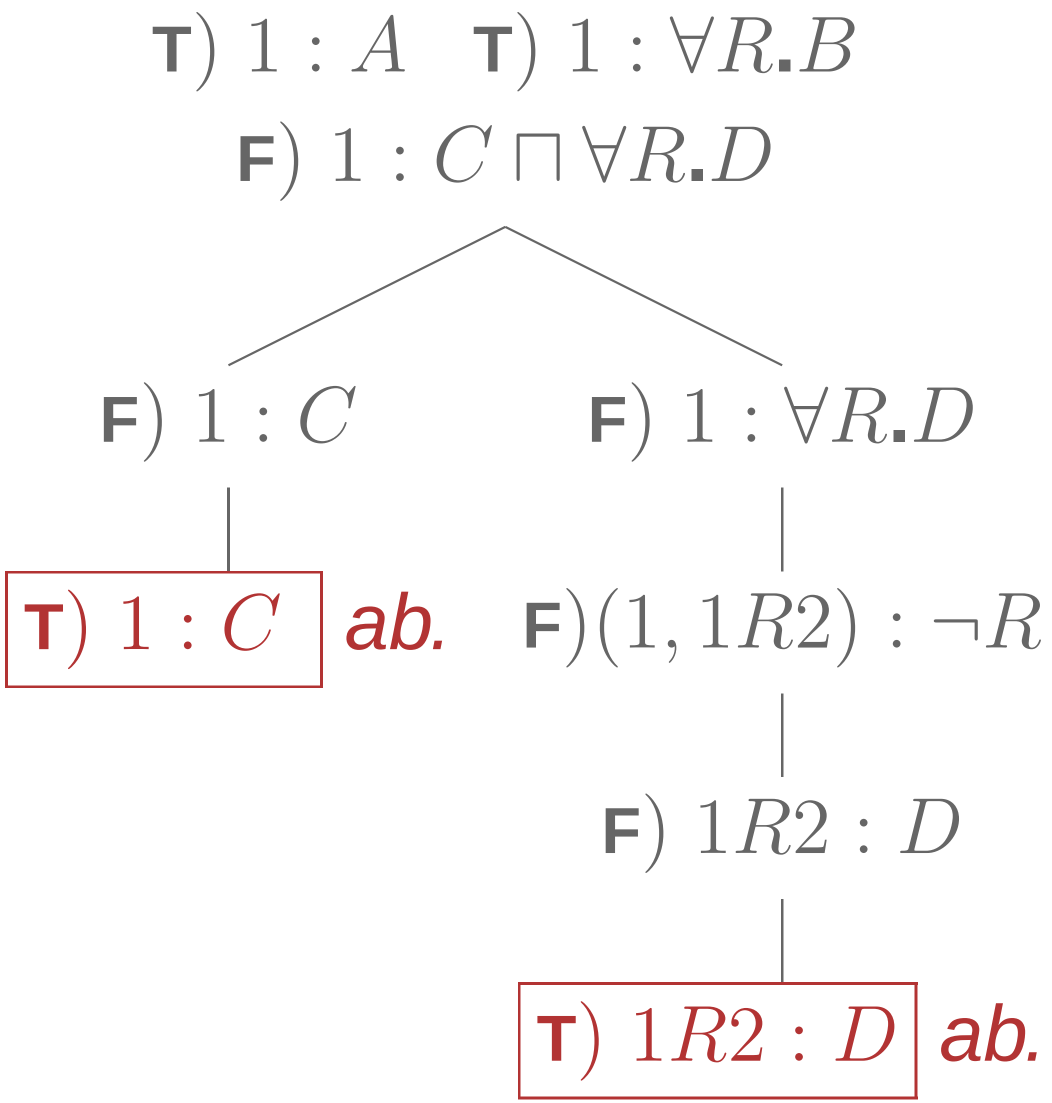
# Two open branches, one choice

Find $H$? such that $A \sqcap \forall R.B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad C \sqcap \forall R.D$

---

# Two open branches, one choice

Find $H$? such that $A \sqcap \forall R.B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad C \sqcap \forall R.D$



**T**) $1 : A$ **T**) $1 : \forall R.B$

**F**) $1 : C \sqcap \forall R.D$

**F**) $1 : C$

**F**) $1 : \forall R.D$

$\boxed{\mathbf{T})\ 1 : C}$ *ab.*

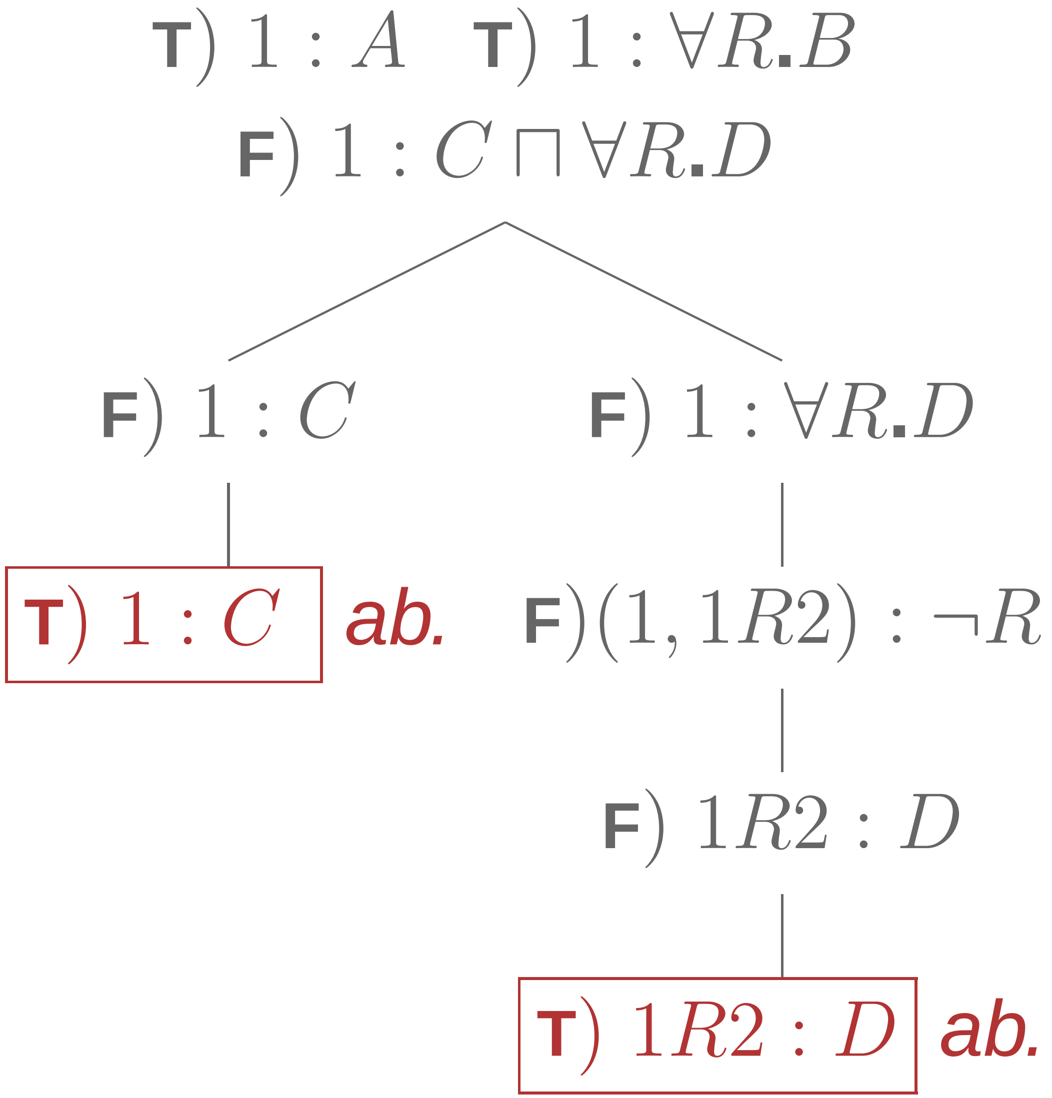
**F**)$(1, 1R2) : \neg R$

**F**) $1R2 : D$

$\boxed{\mathbf{T})\ 1R2 : D}$ *ab.*

# Two open branches, one choice

Find $H$? such that $A \sqcap \forall R.B \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad C \sqcap \forall R.D$



**T**) $1 : A$ **T**) $1 : \forall R.B$

**F**) $1 : C \sqcap \forall R.D$

**F**) $1 : C$

**T**) $1 : C$ *ab.*

**F**) $1 : \forall R.D$

**F**)$(1, 1R2) : \neg R$

**F**) $1R2 : D$

**T**) $1R2 : D$ *ab.*

One solution

$H = \forall roles(1).C \sqcap \forall roles(1R2).D = C \sqcap \forall R.D$

# One open branch, two choices

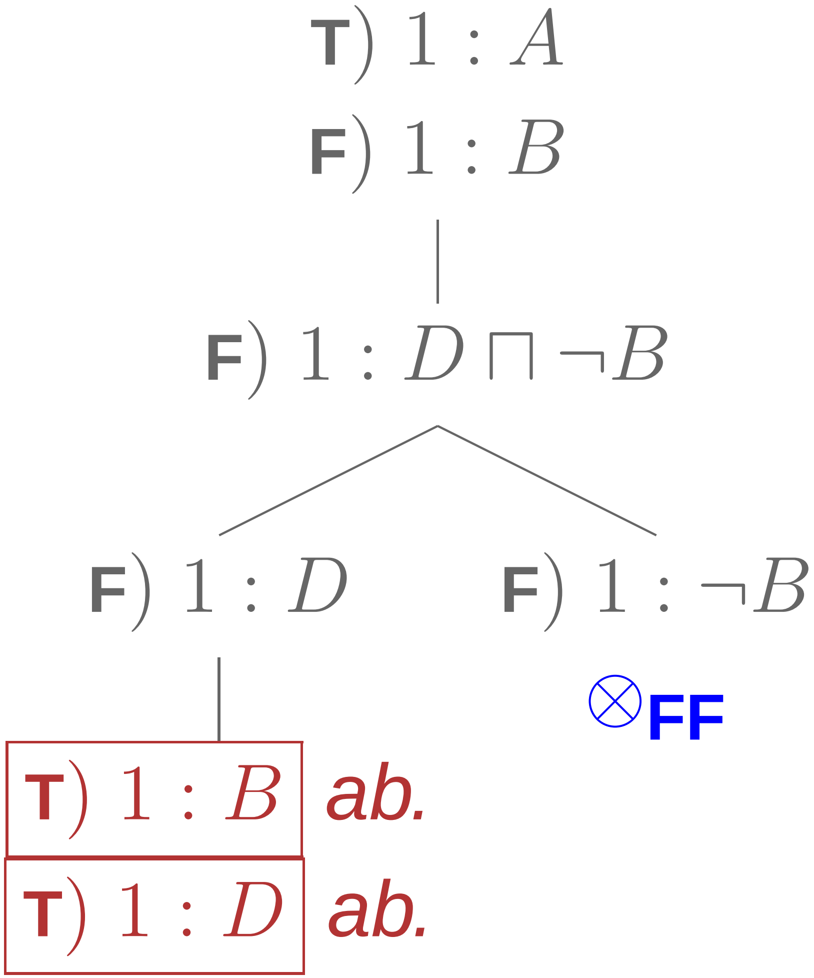
Find $H?$ such that $A \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B$ with $\mathcal{T} = \{D \sqsubseteq B\}$

---

# One open branch, two choices

Find $H?$ such that $A \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B$ with
$\mathcal{T} = \{D \sqsubseteq B\}$

---



**T**$)\ 1 : A$

**F**$)\ 1 : B$

**F**$)\ 1 : D \sqcap \neg B$

**F**$)\ 1 : D$

**F**$)\ 1 : \neg B$

$\otimes$**FF**

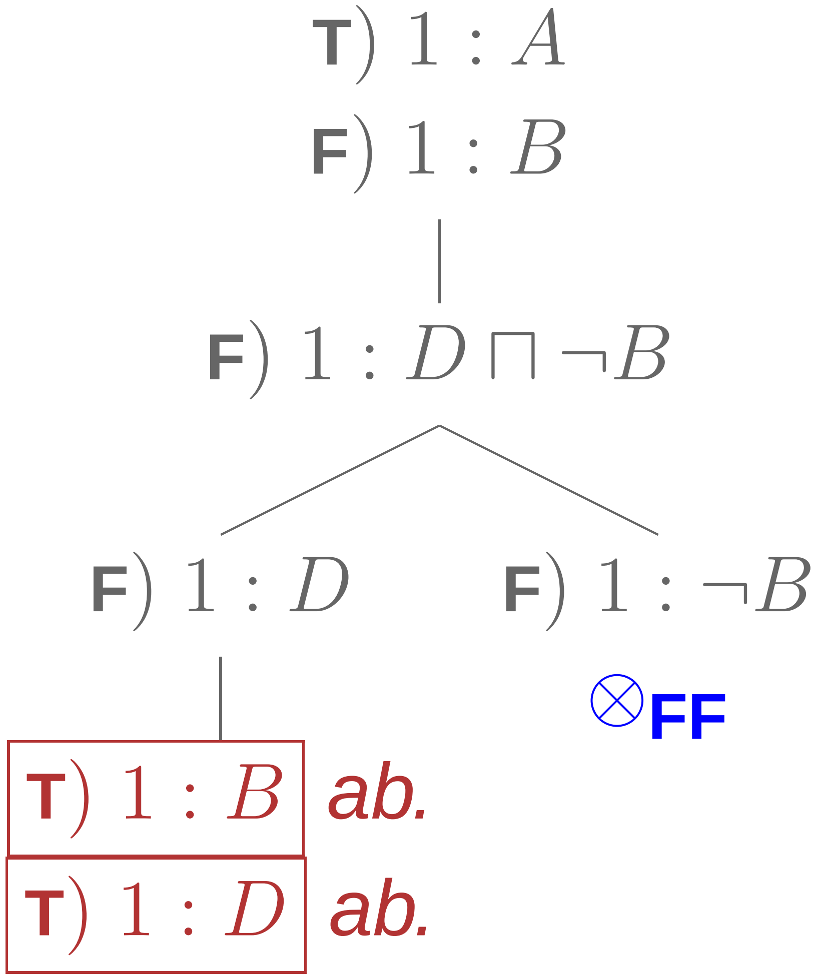
**T**$)\ 1 : B$ *ab.*

**T**$)\ 1 : D$ *ab.*

# One open branch, two choices

Find $H?$ such that $A \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B$ with
$\mathcal{T} = \{D \sqsubseteq B\}$

---



**T**) $1 : A$

**F**) $1 : B$

**F**) $1 : D \sqcap \neg B$

**F**) $1 : D$ **F**) $1 : \neg B$

$\otimes$**FF**

$\boxed{\textbf{T}) \ 1 : B}$ *ab.*

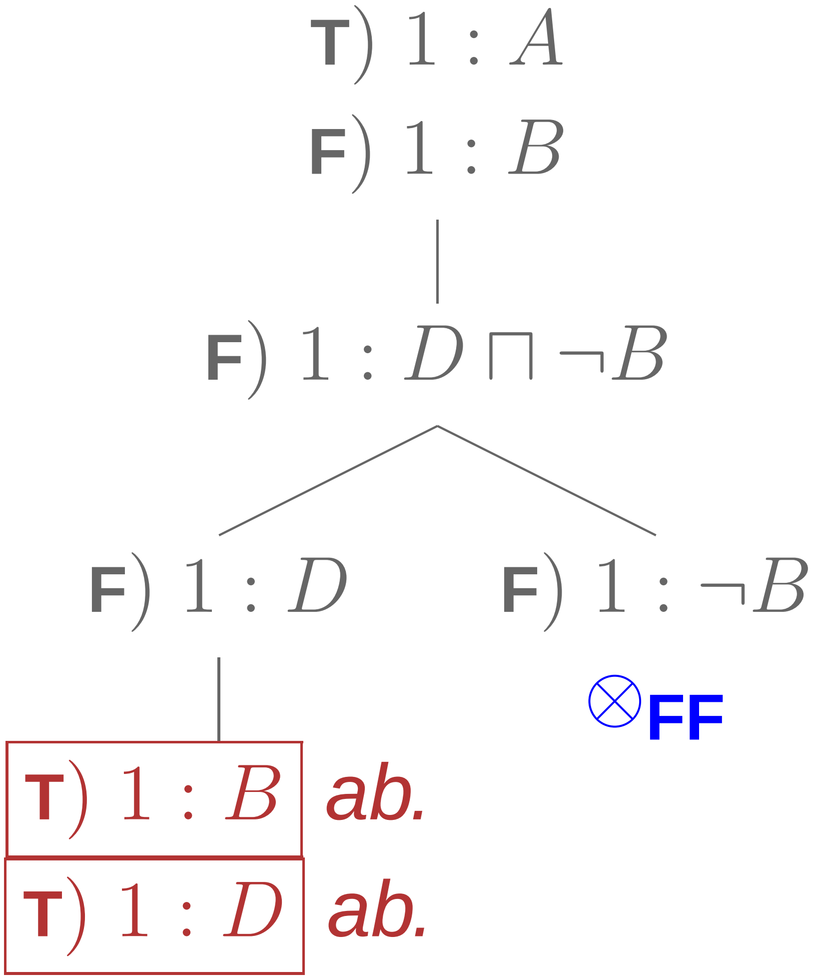
$\boxed{\textbf{T}) \ 1 : D}$ *ab.*

Two solutions
$H_1 = B$,
$H_2 = D$

# One open branch, two choices

Find $H?$ such that $A \sqcap H \quad \sqsubseteq_{\mathcal{T}} \quad B$ with $\mathcal{T} = \{D \sqsubseteq B\}$

---



**T**) $1 : A$

**F**) $1 : B$

**F**) $1 : D \sqcap \neg B$

**F**) $1 : D$ **F**) $1 : \neg B$

$\otimes$**FF**

$\boxed{\textbf{T}) \; 1 : B}$ *ab.*

$\boxed{\textbf{T}) \; 1 : D}$ *ab.*

Two solutions
$H_1 = B$,
$H_2 = D$

Note: $H_2$ is *not* a solution of $B - LCS(A, B)$

[Benatallah *et al.*, 2005],

[Lécué and Delteil, 2007]

# 2 open branches, 2 choices each

Find $H?$ s.t. $(\mathit{FiatPanda}\sqcap)\mathit{yr2000} \sqsubseteq_{\mathcal{T}}$

$(\mathit{FiatPanda}\sqcap)\mathit{radio} \sqcap \mathit{fogLmps}$

with $\mathcal{T} = \{\mathit{bundleOff} \sqsubseteq \mathit{radio} \sqcap \mathit{fogLmps}\}$

---

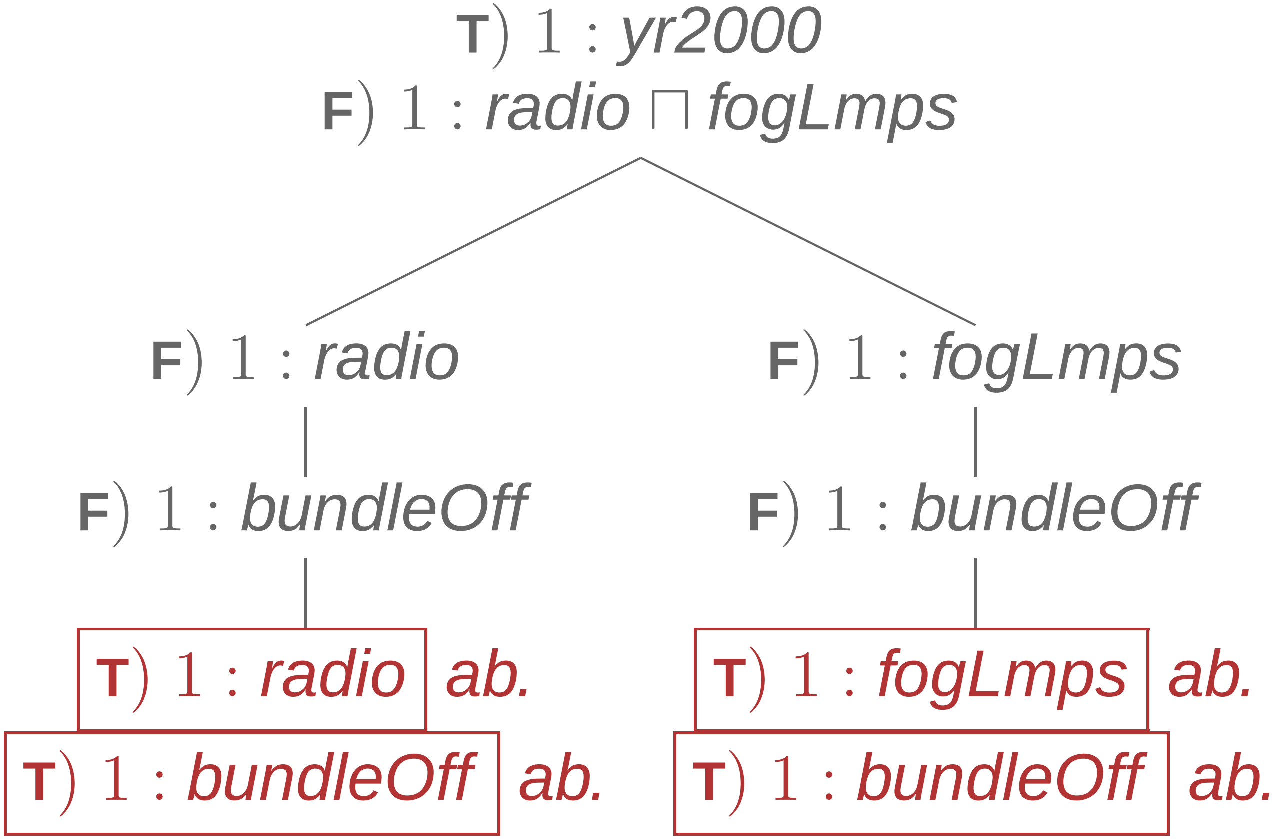
# 2 open branches, 2 choices each

Find $H?$ s.t. $(\textit{FiatPanda}\sqcap)\textit{yr2000} \sqsubseteq_{\mathcal{T}}$

$(\textit{FiatPanda}\sqcap)\textit{radio} \sqcap \textit{fogLmps}$

with $\mathcal{T} = \{\textit{bundleOff} \sqsubseteq \textit{radio} \sqcap \textit{fogLmps}\}$

---



**T**$)\ 1 : \textit{yr2000}$

**F**$)\ 1 : \textit{radio} \sqcap \textit{fogLmps}$

**F**$)\ 1 : \textit{radio}$ | **F**$)\ 1 : \textit{fogLmps}$

**F**$)\ 1 : \textit{bundleOff}$ | **F**$)\ 1 : \textit{bundleOff}$

$\boxed{\textbf{T})\ 1 : \textit{radio}}$ *ab.* | $\boxed{\textbf{T})\ 1 : \textit{fogLmps}}$ *ab.*

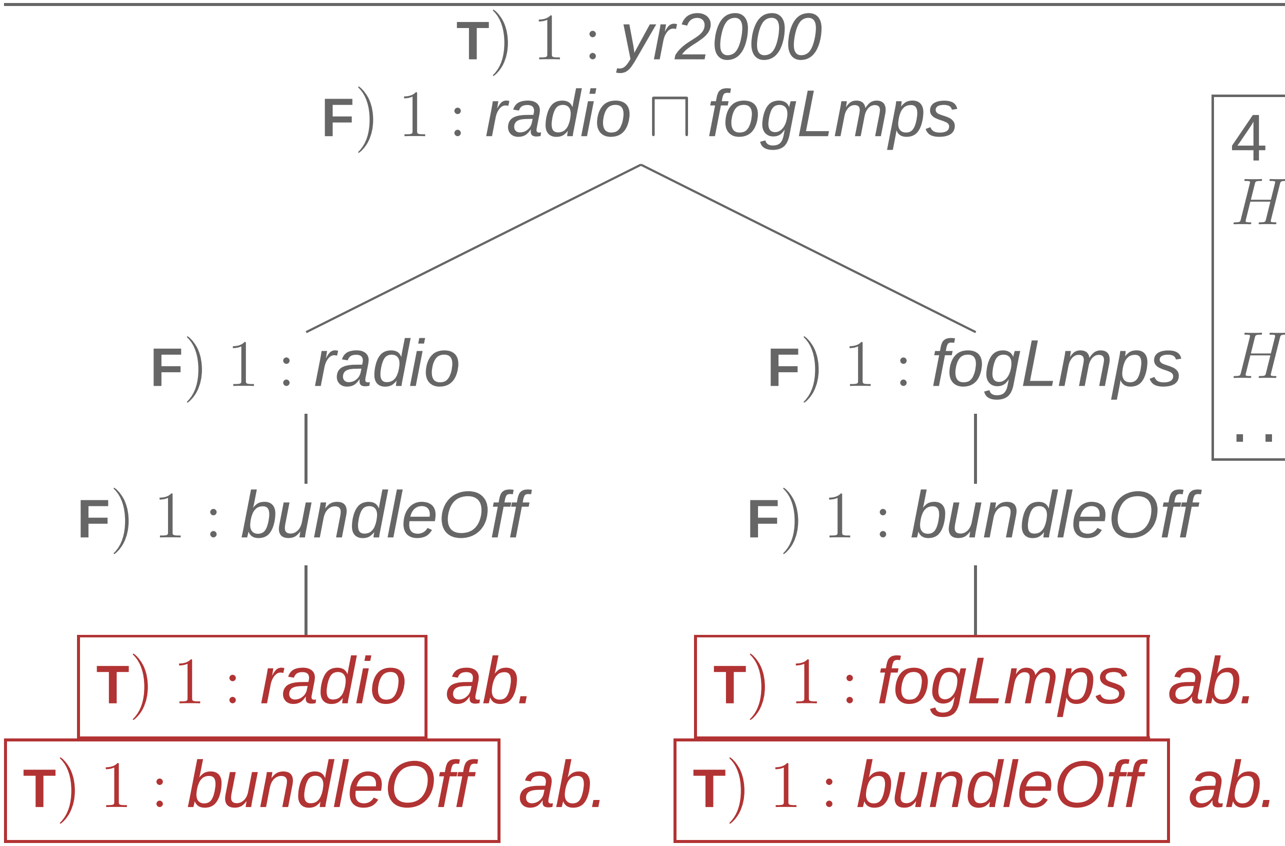
$\boxed{\textbf{T})\ 1 : \textit{bundleOff}}$ *ab.* | $\boxed{\textbf{T})\ 1 : \textit{bundleOff}}$ *ab.*

# 2 open branches, 2 choices each

Find $H?$ s.t. $($*FiatPanda*$\sqcap)$*yr2000* $\sqsubseteq_{\mathcal{T}}$

$($*FiatPanda*$\sqcap)$*radio* $\sqcap$ *fogLmps*

with $\mathcal{T} = \{$*bundleOff* $\sqsubseteq$ *radio* $\sqcap$ *fogLmps*$\}$



**T**$)\ 1 :$ *yr2000*

**F**$)\ 1 :$ *radio* $\sqcap$ *fogLmps*

**F**$)\ 1 :$ *radio*

**F**$)\ 1 :$ *fogLmps*

**F**$)\ 1 :$ *bundleOff*

**F**$)\ 1 :$ *bundleOff*

**T**$)\ 1 :$ *radio* *ab.*

**T**$)\ 1 :$ *fogLmps* *ab.*

**T**$)\ 1 :$ *bundleOff* *ab.*

**T**$)\ 1 :$ *bundleOff* *ab.*

4 solutions:

$H_1 =$ *radio*$\sqcap$ *fogLmps*

$H_2 =$ *bundleOff*

$\ldots$

# Properties

- minimum-length $H$ = minimum hitting set of the abducibles in branches $\mathcal{B}_1, \ldots, \mathcal{B}_m$

# Properties

- minimum-length $H$ = minimum hitting set of the abducibles in branches $\mathcal{B}_1, \ldots, \mathcal{B}_m$
- subsumption-maximal $H$

# Properties

- minimum-length $H$ = minimum hitting set of the abducibles in branches $\mathcal{B}_1, \ldots, \mathcal{B}_m$
- subsumption-maximal $H$
  - after choosing $H$, keep applying rules

# Properties

- minimum-length $H$ = minimum hitting set of the abducibles in branches $\mathcal{B}_1, \ldots, \mathcal{B}_m$
- subsumption-maximal $H$
    - after choosing $H$, keep applying rules
    - $H$ is *not* subs-max iff every branch closes *also* with another clash

# Start for Contraction

Find $G, K?$ s.t. $D \sqcap K$ is sat. and $C \equiv_{\mathcal{T}} G \sqcap K$

# Start for Contraction

Find $G, K?$ s.t. $D \sqcap K$ is sat. and $C \equiv_{\mathcal{T}} G \sqcap K$

start with

**T**$)\ 1 : C$

**F**$)\ 1 : NF(\neg D)$

branches $\mathcal{B}_1 \cdots \mathcal{B}_m$,

complete

# Start for Contraction

Find $G, K?$ s.t. $D \sqcap K$ is sat. and $C \equiv_{\mathcal{T}} G \sqcap K$

start with

**T**) $1 : C$

**F**) $1 : NF(\neg D)$

branches $\mathcal{B}_1 \cdots \mathcal{B}_m$, complete

*choose* one $\mathcal{B}$ with *heterogeneous* clashes only

# Computing Contraction

- take all $\langle E_i, x_i \rangle$ s.t. $\boxed{\textbf{T}) \; x_i : E_i}$ $\boxed{\textbf{F}) \; x_i : NF(\neg E_i)}$ is a heterogeneous clash in the chosen $\mathcal{B}$

# Computing Contraction

- take all $\langle E_i, x_i \rangle$ s.t. $\boxed{\textbf{T}) \ x_i : E_i}$ $\boxed{\textbf{F}) \ x_i : NF(\neg E_i)}$ is a heterogeneous clash in the chosen $\mathcal{B}$
- let $G := \sqcap_i \forall roles(x_i).E_i$

# Computing Contraction

- take all $\langle E_i, x_i\rangle$ s.t. $\boxed{\textbf{T})\ x_i : E_i}$ $\boxed{\textbf{F})\ x_i : NF(\neg E_i)}$ is a heterogeneous clash in the chosen $\mathcal{B}$
- let $G := \sqcap_i \forall roles(x_i).E_i$
- $K := C'$ where each $E_i$ is substituted by $\top$ (*i.e.*, deleted)

# Computing Contraction

- take all $\langle E_i, x_i\rangle$ s.t. $\boxed{\textbf{T})\ x_i : E_i}$ $\boxed{\textbf{F})\ x_i : NF(\neg E_i)}$ is a heterogeneous clash in the chosen $\mathcal{B}$
- let $G := \sqcap_i \forall roles(x_i).E_i$
- $K := C'$ where each $E_i$ is substituted by $\top$ (*i.e.*, deleted)
- several $\langle G, K\rangle$'s, depending on the chosen $\mathcal{B}$

# Computing Contraction

- take all $\langle E_i, x_i\rangle$ s.t. $\boxed{\textbf{T}) \; x_i : E_i}$ $\boxed{\textbf{F}) \; x_i : NF(\neg E_i)}$ is a heterogeneous clash in the chosen $\mathcal{B}$
- let $G := \sqcap_i \forall roles(x_i).E_i$
- $K := C'$ where each $E_i$ is substituted by $\top$ (*i.e.*, deleted)
- several $\langle G, K\rangle$'s, depending on the chosen $\mathcal{B}$
- Note: $\exists R$ is contracted only if it clashes with $\forall R.\bot$

# Two branches

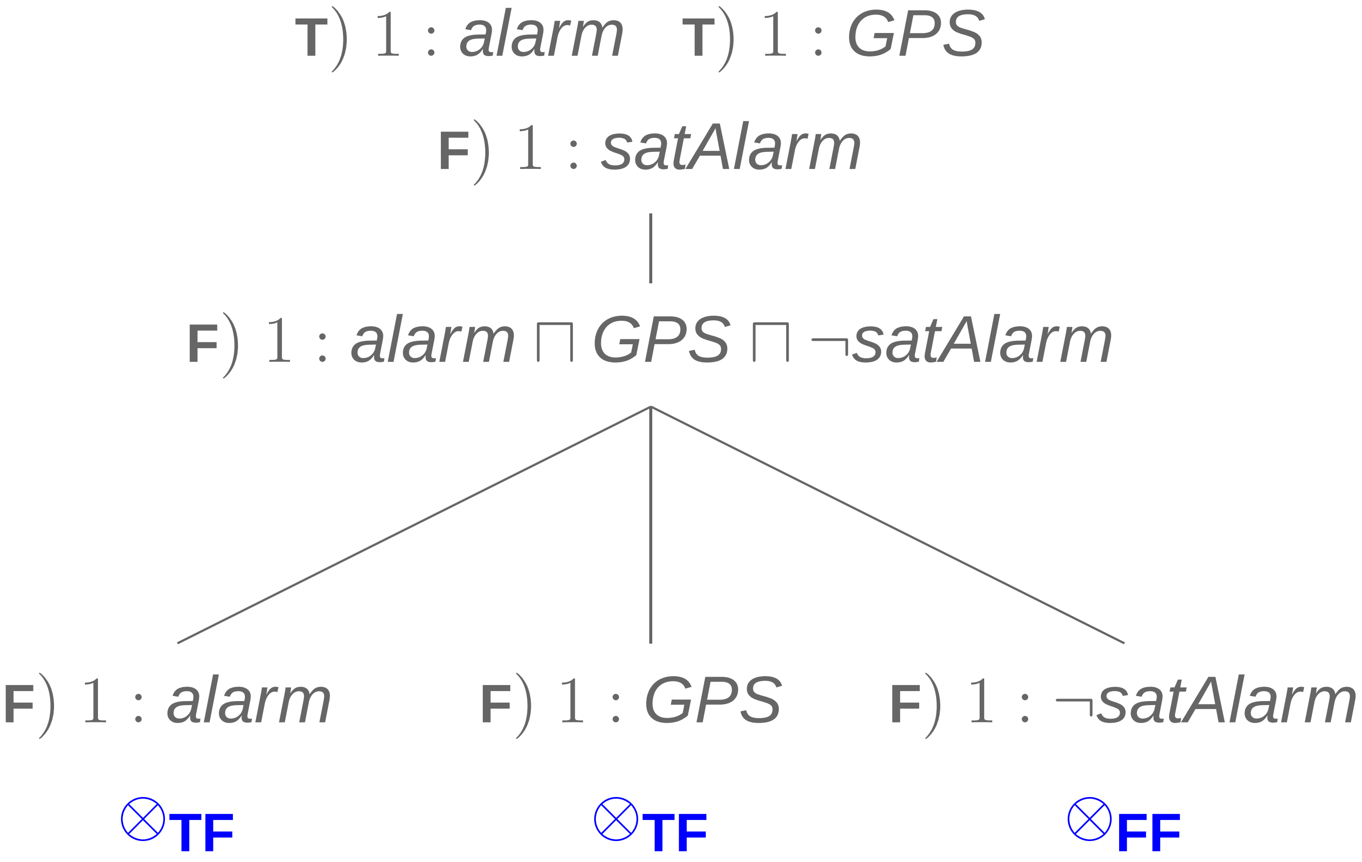
Find $G, K?$ s.t. $\neg$*satAlarm* $\sqcap K$ is sat. and *alarm* $\sqcap$ *GPS* $\equiv_{\mathcal{T}}$ $G \sqcap K$, with $\mathcal{T} = \{$*alarm* $\sqcap$ *GPS* $\sqsubseteq$ *satAlarm*$\}$

---

# Two branches

Find $G, K?$ s.t. $\neg$*satAlarm* $\sqcap K$ is sat. and *alarm* $\sqcap$ *GPS* $\equiv_{\mathcal{T}}$ $G \sqcap K$, with $\mathcal{T} = \{$*alarm* $\sqcap$ *GPS* $\sqsubseteq$ *satAlarm*$\}$

---



**T**) $1 :$ *alarm*   **T**) $1 :$ *GPS*

**F**) $1 :$ *satAlarm*

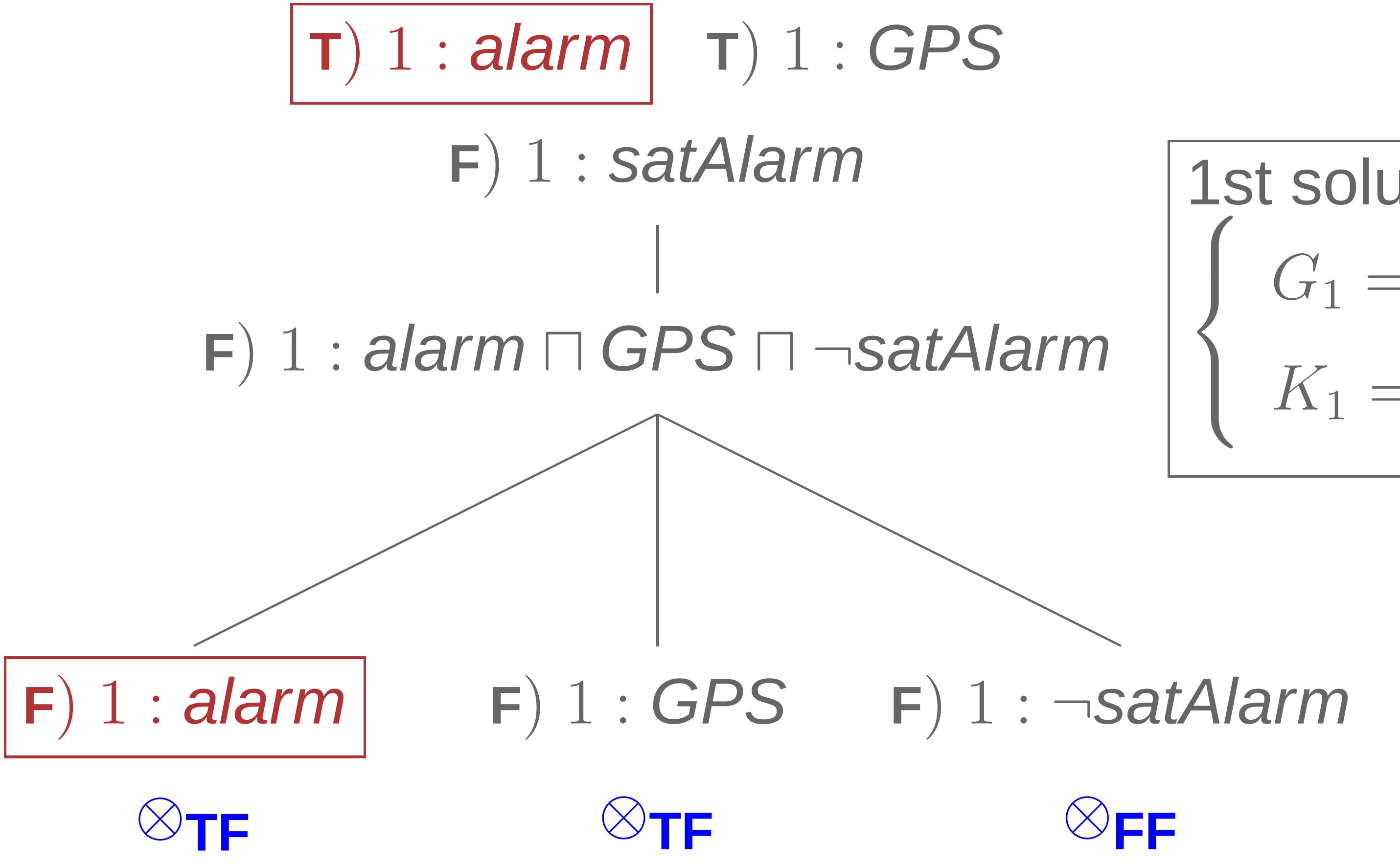
**F**) $1 :$ *alarm* $\sqcap$ *GPS* $\sqcap \neg$*satAlarm*

**F**) $1 :$ *alarm*   **F**) $1 :$ *GPS*   **F**) $1 : \neg$*satAlarm*

$\otimes$**TF**   $\otimes$**TF**   $\otimes$**FF**

# Two branches

Find $G, K?$ s.t. $\neg$*satAlarm* $\sqcap K$ is sat. and *alarm* $\sqcap$ *GPS* $\equiv_{\mathcal{T}} G \sqcap K$, with $\mathcal{T} = \{$*alarm* $\sqcap$ *GPS* $\sqsubseteq$ *satAlarm*$\}$



**T**) $1 :$ *alarm* **T**) $1 :$ *GPS*

**F**) $1 :$ *satAlarm*

**F**) $1 :$ *alarm* $\sqcap$ *GPS* $\sqcap \neg$*satAlarm*

**F**) $1 :$ *alarm* **F**) $1 :$ *GPS* **F**) $1 : \neg$*satAlarm*

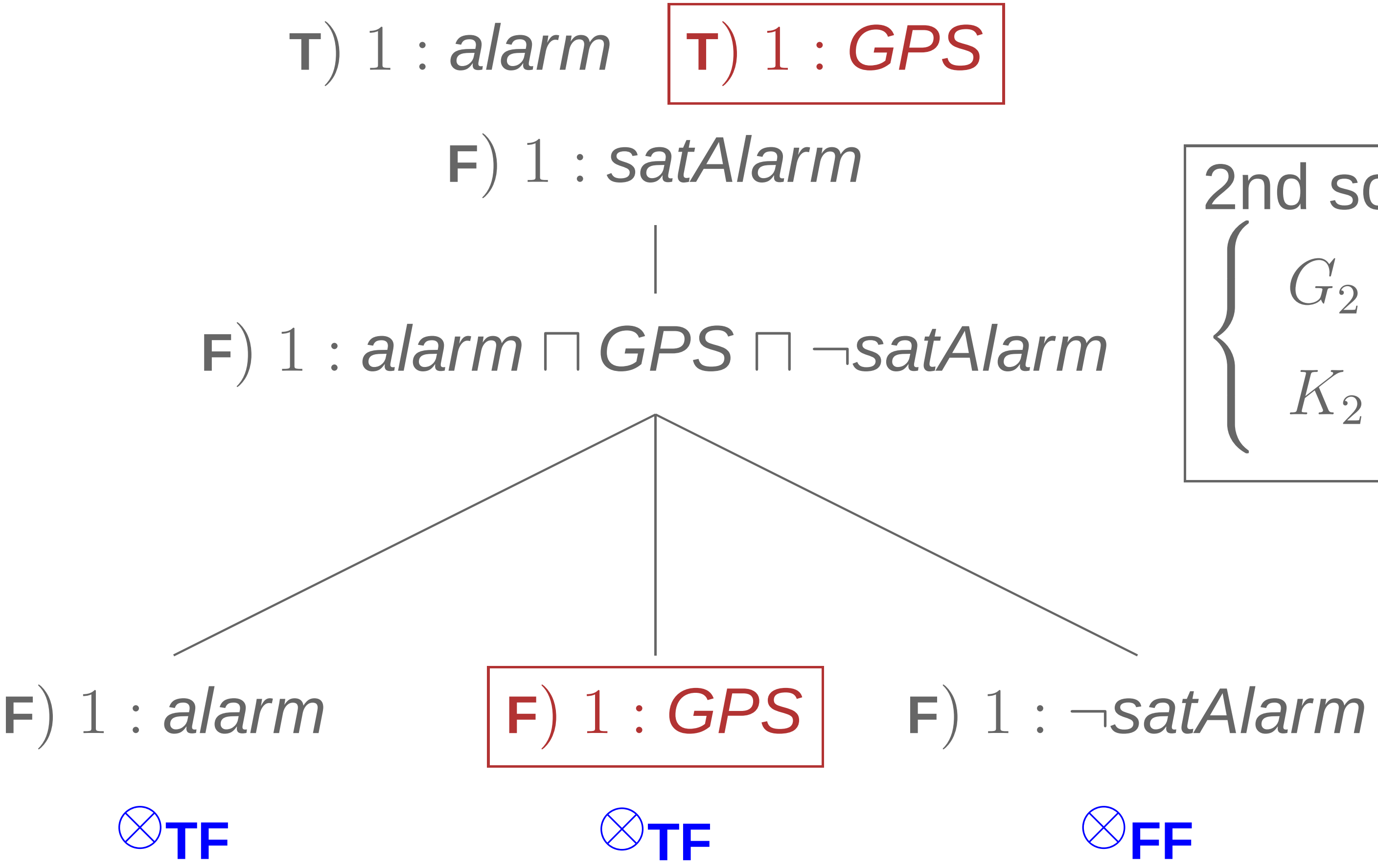
$\otimes$**TF** $\otimes$**TF** $\otimes$**FF**

1st solution:

$$\begin{cases} G_1 = \textit{alarm} \\ K_1 = \textit{GPS} \end{cases}$$

# Two branches

Find $G, K?$ s.t. $\neg \mathit{satAlarm} \sqcap K$ is sat. and $\mathit{alarm} \sqcap \mathit{GPS} \equiv_{\mathcal{T}} G \sqcap K$, with $\mathcal{T} = \{\mathit{alarm} \sqcap \mathit{GPS} \sqsubseteq \mathit{satAlarm}\}$



**T**) $1 : \mathit{alarm}$ $\boxed{\textbf{T}) \ 1 : \mathit{GPS}}$

**F**) $1 : \mathit{satAlarm}$

**F**) $1 : \mathit{alarm} \sqcap \mathit{GPS} \sqcap \neg \mathit{satAlarm}$

**F**) $1 : \mathit{alarm}$ $\boxed{\textbf{F}) \ 1 : \mathit{GPS}}$ **F**) $1 : \neg \mathit{satAlarm}$

$\otimes$**TF** $\otimes$**TF** $\otimes$**FF**

2nd solution:

$$\begin{cases} G_2 = \mathit{GPS} \\ K_2 = \mathit{alarm} \end{cases}$$

# Intermezzo 2

**— Now, *I* need a coffee...**

# Implementation

# MaMaS-tng

- MAtch MAking Service - The Next Generation
- Subsumption, Satisfiability, Concept Abduction and Concept Contraction in $\mathcal{ALN}$
- exposes an exended DIG 1.1 interface
- available as an HTTP service (only HTTP-POST requests)

# OwlEd

- OWL Editor
- supports MaMaS-tng
- also other reasoners endowed of DIG1.1 interface
- OwlEd beta is freely downloadable

# Next issues

- more expressive DLs ($\mathcal{ALC}$)

# Next issues

- more expressive DLs ($\mathcal{ALC}$)
- fuzzy DLs for concrete domains [Ragone *et al.*, 2007], [Ragone *et al.*, 2008]

# Next issues

- more expressive DLs ($\mathcal{ALC}$)
- fuzzy DLs for concrete domains [Ragone *et al.*, 2007], [Ragone *et al.*, 2008]
  - *e.g.*, price, color, delivery time

# Next issues

- more expressive DLs ($\mathcal{ALC}$)
- fuzzy DLs for concrete domains [Ragone *et al.*, 2007], [Ragone *et al.*, 2008]
  - *e.g.*, price, color, delivery time
  - the mediator can *negotiate* conflicting issues

# Future issues

- agents carry *both* an offer *and* a request

# Future issues

- agents carry *both* an offer *and* a request
  - “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002

# Future issues

- agents carry *both* an offer *and* a request
  - “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
  - Dating services

# Future issues

- agents carry *both* an offer *and* a request
  - “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
  - Dating services
- epistemic statements

# Future issues

- agents carry *both* an offer *and* a request
    - “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
    - Dating services
- epistemic statements
    - “Best prices paid”

# Future issues

- agents carry *both* an offer *and* a request
  - “Award-winner chinese calligrapher seeks flat in London” — Sunday Times, August 2002
  - Dating services
- epistemic statements
  - “Best prices paid”
  - “smokers allowed”

# Acknowledgements

All people at SisInfLab, Politecnico di Bari

- Simona Colucci, Tommaso Di Noia,
- Eugenio Di Sciascio, Daniele Maggiore, Agnese Pinto,
- Azzurra Ragone, Michele Ruta,
- Floriano Scioscia, Eufemia Tinelli,
- ... among many others

# References

In the notes of this slide,
references can be found.

Slides are available at
http://sisinflab.poliba.it/donini

# References

[Benatallah *et al.*, 2005] Boualem Benatallah, Mohand-Said Hacid, Alain Léger, Christophe Rey, and Farouk Toumani. On automating web services discovery. *VLDB J.*, 14(1):84–96, 2005.

[Borgida and Patel-Schneider, 1994] A. Borgida and P. F. Patel-Schneider. A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic. *Journal of Artificial Intelligence Research*, 1:277–308, 1994.

[Calì *et al.*, 2004] A. Calì, D. Calvanese, S. Colucci, T. Di Noia, and F. M. Donini. A description logic based approach for matching user profiles. In *Proceedings of the 17th International Workshop on Description Logics (DL'04)*, volume 104 of *CEUR Workshop Proceedings*, 2004.

[Cialdea Mayer and Pirri, 1995] Marta Cialdea Mayer and Fiora Pirri. Propositional abduction in modal logic. *Logic Journal of the IGPL*, 3(6):907–919, 1995.

[Colucci *et al.*, 2004] S. Colucci, T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. Uniform tableaux-based approach to concept abduction and contraction in aln dl. In *Proceedings of the 17th International Workshop on Description Logics (DL'04)*, volume 104 of *CEUR Workshop Proceedings*, 2004.

[Di Noia *et al.*, 2003] T. Di Noia, E. Di Sciascio, F.M. Donini, and M. Mongiello. A system for principled Matchmaking in an electronic marketplace. In *Proc. International World Wide Web Conference (WWW '03)*, pages 321–330. ACM Press, 2003.

[Di Noia *et al.*, 2007] Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M. Donini. Semantic matchmaking as non-monotonic reasoning: A description logic approach. *Journal of Artificial Intelligence Research*, 29(2), 2007.

[Donini and Massacci, 2000] F.M. Donini and F. Massacci. Exptime tableaux for $\mathcal{ALC}$. *Artificial Intelligence*, 124:87–138, 2000.

[Eiter and Gottlob, 1995] T. Eiter and G. Gottlob. The complexity of logic-based abduction. *Journal of the ACM*, 42(1):3–42, 1995.

[Gärdenfors, 1988] P. Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. Bradford Books, MIT Press, Cambridge, MA, 1988.

[Kießling, 2002] W. Kießling. Foundations of preferences in database systems. In *Proc. of 28th VLDB Conference*, 2002.

[Lécué and Delteil, 2007] Freddy Lécué and Alexandre Delteil. Making the difference in semantic web service composition. In *AAAI*, pages 1383–1388. AAAI Press, 2007.

[Levesque, 1989] H. J. Levesque. A knowledge-level account of abduction. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1061–1067, 1989.

[Lukasiewicz and Schellhase, 2006] Thomas Lukasiewicz and Jörg Schellhase. Variable-strength conditional preferences for matchmaking in description logics. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *KR*, pages 164–174. AAAI Press, 2006.

[Pople, 1973] Harry E. Pople. On the mechanization of abductive logic. In *IJCAI*, pages 147–152, 1973.

[Ragone *et al.*, 2007] Azzurra Ragone, Umberto Straccia, Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M. Donini. Vague knowledge bases for matchmaking in p2p e-marketplaces. In Enrico Franconi, Michael Kifer, and Wolfgang May, editors, *ESWC*, volume 4519 of *Lecture Notes in Computer Science*, pages 414–428. Springer, 2007.

[Ragone *et al.*, 2008] Azzurra Ragone, Umberto Straccia, Tommaso Di Noia, Eugenio Di Sciascio, and Francesco M. Donini. Towards a fuzzy logic for automated multi-issue negotiation. In Sven Hartmann and Gabriele Kern-Isberner, editors, *FoIKS*, volume 4932 of *Lecture Notes in Computer Science*, pages 381–396. Springer, 2008.

[Reggia *et al.*, 1985] James A. Reggia, Dana S. Nau, Pearl Y. Wang, and Yun Peng. A formal model of diagnostic inference, ii. algorithmic solution and application. *Information Sciences*, 37(1-3):257–285, 1985.

[Smullyan, 1968] R. M. Smullyan. *First Order Logic*. Springer-Verlag, Berlin, 1968.

[Teege, 1994] G. Teege. Making the difference: A subtraction operation for description logics. In *Proceedings of the Fourth International Conference on the Principles of Knowledge Representation and Reasoning (KR'94)*, pages 540–550. MK, 1994.

[Trastour *et al.*, 2002] D. Trastour, C. Bartolini, and C. Priest. Semantic web support for the Business-to-Business e-commerce lifecycle. In *Proc. International World Wide Web Conference (WWW) '02*, pages 89–98. ACM, 2002.

[Zaremski and Wing, 1997] Amy Moormann Zaremski and Jeannette M. Wing. Specification matching of software components. *ACM Trans. Softw. Eng. Methodol.*, 6(4):333–369, 1997.