Attack–Defense Tree Methodology for Security Assessment

Barbara Kordy

Joint work with Patrick Schweitzer, Sjouke Mauw, Saša Radomirović

UNIVERSITÉ DU LUXEMBOURG

Fonds National de la Recherche Luxembourg

Outline

Outline

Attack trees

Definition

Attack tree (ATree) – tree-like representation of an attacker's goal recursively refined into conjunctive or disjunctive sub-goals.

Methodology to describe security weaknesses of a system

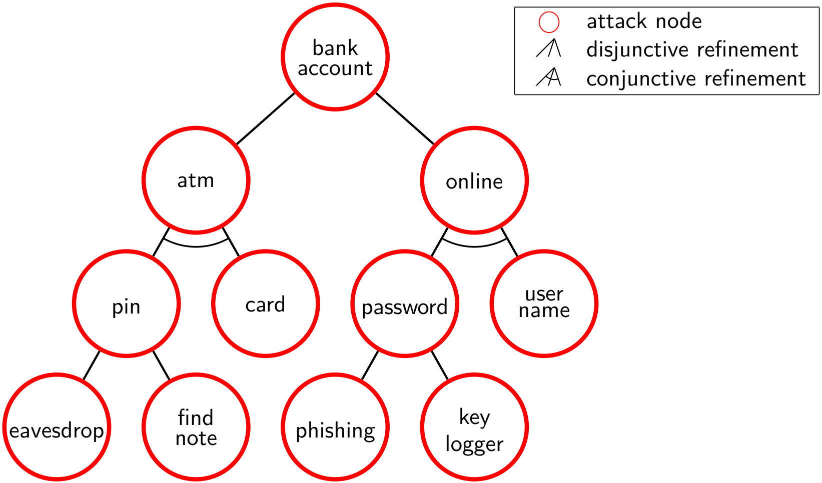
- Proposed by Schneier

  Attack trees: Modeling Security Threats, '99

- Formalized by Mauw and Oostdijk

  Foundations of Attack Trees [ICISC'05]

Example: attacking a bank account



Limitations of attack trees

- Only attacker's point of view
- No defensive measures
- No attacker/defender interactions
- No evolutionary aspects

Attack–defense trees

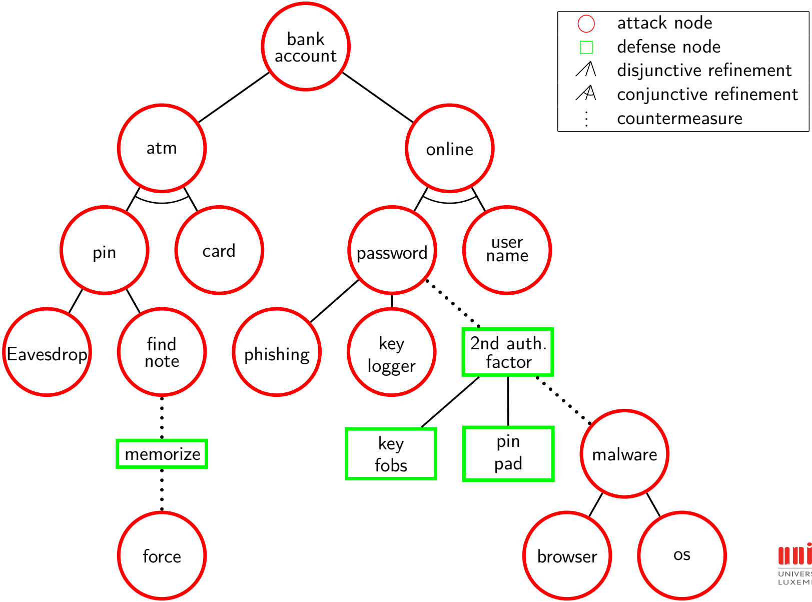
Definition

Attack–defense tree (ADTree) – attack tree extended with possibly refined or countered defensive actions.

Introduced by Kordy et al. in

Foundations of Attack–Defense Trees [FAST'10]

Example: attacking and defending a bank account



Interesting questions

- Equivalent representations of the same scenario (semantics)
- Quantitative analysis (attributes)
- Computational complexity of ATrees and ADTrees (querying)
- Practical applications (case studies)

Outline

UNIVERSITÉ DU LUXEMBOURG

Semantics for ADTrees

Semantics define which ADTrees represent the same scenario.

Definition

Semantics for ADTrees – equivalence relation on ADTrees.

- Propositional semantics
- Semantics induced by a De Morgan lattice
- Multiset semantics
- Equational semantics

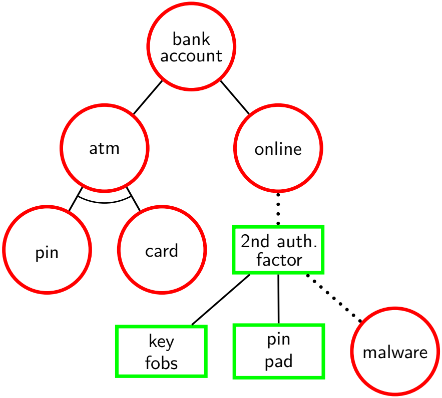
Propositional semantics for ADTrees

In the propositional semantics

ADTrees represent Boolean functions.

Example: propositional interpretation of an ADTree

$$f = (\mathsf{pin} \wedge \mathsf{card}) \vee \Big(\mathsf{online} \wedge \neg((\mathsf{key\ fobs} \vee \mathsf{pin\ pad}) \wedge \neg \mathsf{malware})\Big)$$



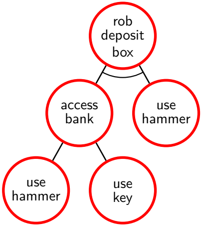
Details

Propositional semantics $\equiv_{\mathcal{P}}$

In the propositional semantics

ADTress represent the same scenario if the corresponding Boolean functions are equivalent.

Example: propositionally equivalent ADTrees



$\equiv_{\mathcal{P}}$

(hammer $\vee$ key) $\wedge$ hammer

hammer

The two trees are equivalent in the propositional semantics, because in propositional logics we have absorption law

$$(\text{hammer} \vee \text{key}) \wedge \text{hammer} \equiv \text{hammer}$$

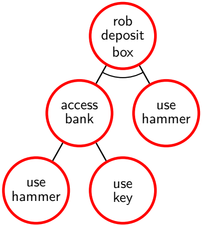
Multiset semantics $\equiv_{\mathcal{M}}$

ADTrees are interpreted as sets of multisets.
Each multiset represents a possible way of attacking.

In the multiset semantics

ADTrees represent the same scenario if the corresponding sets of multisets are equal.

Example: ADTrees not equivalent in the multiset semantics

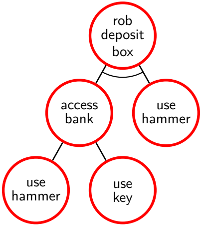


$\not\equiv_{\mathcal{M}}$

$\{\Lbag \mathsf{hammer}, \mathsf{hammer} \Rbag, \Lbag \mathsf{key}, \mathsf{hammer} \Rbag\}$ $\{\Lbag \mathsf{hammer} \Rbag\}$

The two trees are not equivalent in the multiset semantics, because $\{\Lbag \mathsf{hammer}, \mathsf{hammer} \Rbag, \Lbag \mathsf{key}, \mathsf{hammer} \Rbag\} \neq \{\Lbag \mathsf{hammer} \Rbag\}$.

Different semantics – different equivalence classes



$\equiv_{\mathcal{P}}$

$\not\equiv_{\mathcal{M}}$

The choice of an appropriate semantics

depends on considered applications and assumptions.

Outline

Motivation

Quantitative analysis of an attack–defense scenario

- Standard questions
  - What is the minimal cost of an attack?
  - What is the expected impact of a considered attack?
  - Is special equipment required to attack?
- Bivariate questions
  - How long does it take to secure a system, when the attacker has a limited budget?
  - How does the scenario change if both, the attacker and the defender are affected by a power outage?

UNIVERSITÉ DU LUXEMBOURG

Calculation of attributes

Bottom-up algorithm

- Basic assignment – values assigned to basic actions
- Attribute domain – operators specifying how to compute values for other nodes

- Intuitive idea of Schneier
  Attack trees: Modelling Security Threats, '99
- Formalization by Mauw and Oostdijk for attack trees
  Foundations of Attack Trees, [ICISC'05]
- Extension to attack–defense trees by Kordy et al.
  Foundations of Attack–Defense Trees, [FAST'10]

UNIVERSITÉ DU LUXEMBOURG

Attribute: minimal time of an attack

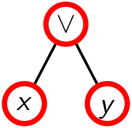
Question:

What is the minimal time needed to achieve a considered attack?

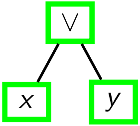
Attribute domain:

- Values from $\mathbb{N} \cup \{\infty\}$
- $\infty =$ action not under control of the attacker
- $(\vee^A, \wedge^A, \vee^D, \wedge^D, c^A, c^D) = (\min, +, +, \min, +, \min)$

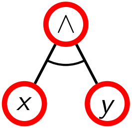
Attribute domain for *minimal time*



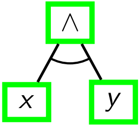
$\vee^A$: $\min\{x, y\}$



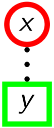
$\vee^D$: $x + y$



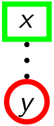
$\wedge^A$: $x + y$



$\wedge^D$: $\min\{x, y\}$



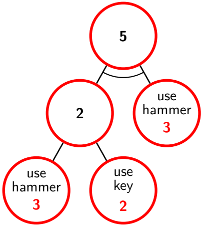
c^A: $x + y$



c^D: $\min\{x, y\}$

Example: computation of *minimal time* on an ADTree

$$(\vee^A, \wedge^A, \vee^D, \wedge^D, c^A, c^D) = (\min, +, +, \min, +, \min)$$

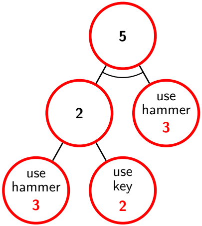


▸ Details

Semantics and attribute domains

Recall: t and t' are equivalent in the propositional semantics.

$t =$



$t' =$

$time(t) = 5$

$time(t') = 3$

- Problem: $t \equiv_{\mathcal{P}} t'$, but $time(t) \neq time(t')$
- Solution: Compatibility notion

Compatibility of an attribute with a semantics

Compatibility defines which semantics should be used in combination with which attribute.

Definition

Attribute α is compatible with semantics $\equiv$ for ADTrees iff $\forall t, t' \in \text{ADTrees}, \quad t \equiv t' \implies \alpha(t) = \alpha(t')$.

- Problem: How to check compatibility?
- Solution: Complete set of axioms for a semantics.

▸ Details

UNIVERSITÉ DU LUXEMBOURG

Axiomatization of semantics

Definition

A set E of ADTree transformations is a complete set of axioms for a semantics for ADTrees iff equivalent ADTrees can be obtained from each other by application of transformations from E.

- Problem: How to find a complete set of axioms for a semantics?
- Solution: This is difficult...

Complete set of axioms

We have identified complete sets of axioms for

- the propositional semantics (44 transformations)
  - using minimal DNF representation of propositional formulas
- the multiset semantics (22 transformations)
  - using term rewriting techniques

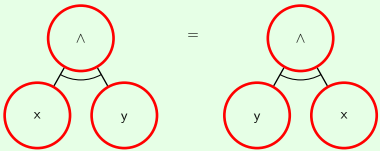
Details can be found in *Attack–Defense Trees* (to appear in JLC'12).

Axiomatization and compatibility

Using a complete set of axioms, compatibility can be decided by performing a finite number of easy checks.

Example

- Transformation – commutativity of attacker's AND refinement



- Corresponding equation for *minimal time* attribute

$$x + y = y + x \quad \text{holds in} \quad \mathbb{N}.$$

Outline

ATrees vs. ADTrees

- ADTrees enrich modeling capabilities of ATrees.
- How much computational power do they require w.r.t. ATrees?

UNIVERSITÉ DU LUXEMBOURG

Boolean functions represented by ATrees and ADTrees

In *Computational Aspects of Attack–Defense Trees* [SIIS'11], we show

Lemma

1. *ATrees represent positive Boolean functions.*
2. *ADTrees represent monotone Boolean functions.*

Theorem

Every monotone Boolean function, which is not positive, can be brought into a positive form in linear time.

Computational complexity of ADTrees vs. ATrees

Corollary (Kordy, Pouly, Schweitzer [SIIS'11])

When the propositional semantics is used, the computational complexity of ADTrees is the same as the computational complexity of ATrees.

Consequences of [SIIS'11]

When the propositional semantics is used

- ADTrees can be processed by algorithms developed for ATrees.
- Complexity of query evaluation on ADTrees is the same as the corresponding complexity on ATrees.
- Queries that can efficiently be solved on ATrees can also efficiently be solved on ADTrees.

UNIVERSITÉ DU LUXEMBOURG

Limitations of the propositional semantics

b $\quad f(b) = b$ $\qquad\qquad$ $\boxed{b}$ $\quad f(b) = b$

- The Boolean function $f\colon \{0,1\}^b \to \{0,1\}$ corresponding to a non-refined node b is of the form $f(b = v) = v$, where $v \in \{1, 0\}$.
- This means that the propositional semantics assumes that each component which is present is fully effective.

- Problem: Such strong assumption is not always desirable.

Example: modeling effectiveness level of an attack

Let $\{F, M, T\}$ be a set of effectiveness levels, where $F < M < T$.

- Boolean function given by $f(\mathrm{d} = 1) = 1$ and $f(\mathrm{d} = 0) = 0$ is not well suited to model effectiveness level of a dictionary attack.
- We need a function of the form $f\colon \{0, 1\}^{\{\mathrm{d}\}} \to \{F, M, T\}$, where $f(\mathrm{d} = 1) = M$ and $f(\mathrm{d} = 0) = F$.

Semantics induced by a De Morgan lattice

In semantics induced by a De Morgan lattice L

ADTrees represent functions of the form $f\colon \{0,1\}^X \to L$, where X is a set of propositional variables.

- De Morgan lattices allow us to use more than only two values 0 and 1.
- Semantics induced by De Morgan lattices allow for more accurate analysis, with respect to the propositional semantics.

▸ Details

Message to remember from [SIIS'11]

Theorem

When a semantics induced by a De Morgan lattice is used, the computational complexity of ADTrees is the same as the computational complexity of ATrees.

When ADTrees represent functions of the form

- $f\colon \{0,1\}^X \to \{0,1\}$
- $f\colon \{0,1\}^X \to L$, with L a De Morgan lattice

enriching the attack tree formalism with defense nodes is not done at the expense of computational complexity.

Outline

1. Attack–defense trees
2. Semantics
3. Quantitative analysis
4. Computational complexity
5. Attack–defense trees in practice

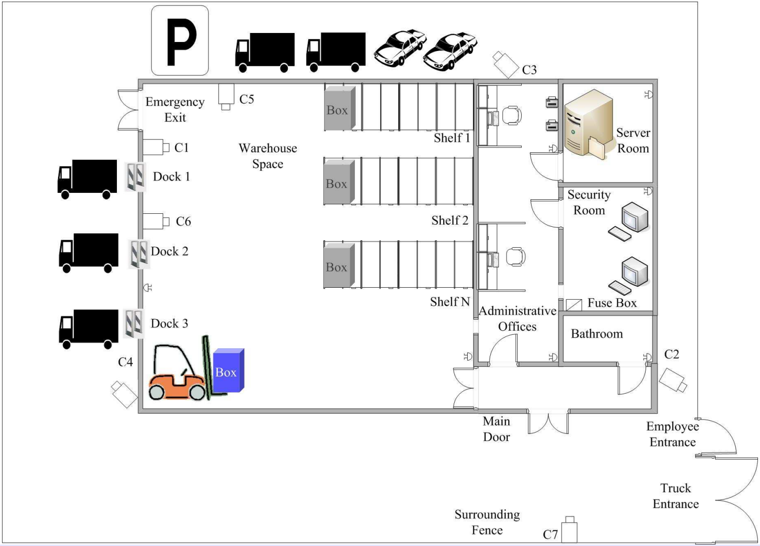
Case study

- Objectives: checking usefulness of the ADTree methodology
  - test
  - validate
  - define tool requirements
  - improve the formalism
- Partners:
  - SINTEF, Norway (Per Håkon Meland)
  - TXT e-solutions, Italy (Alessandra Bagnato)
- Results: *Attribute Decoration of Attack–Defense Trees* [IJSSE'12]

UNIVERSITÉ DU LUXEMBOURG

Case study scenario

DoS in RFID-based goods management system



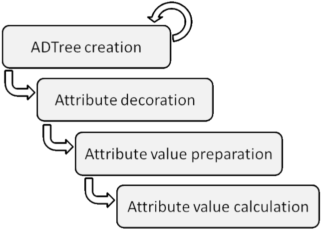
Case study ADTree

- ADTree of 97 nodes
- Taking into account multiple aspects:
  - physical access,
  - social engineering attacks,
  - digital attacks.
- Evaluation of 10 attributes: *cost, time, detectability, penalty, skill level, impact, difficulty, profitability*

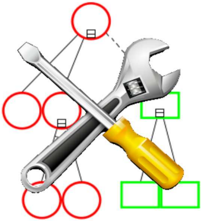
UNIVERSITÉ DU LUXEMBOURG

Case study outcomes

- Guidelines explaining how to use ADTrees in practice



- Requirements for an ADTree software



ADTool

ADTool

Software tool supporting the ADTree methodology.

- Implemented in Java.
- Compatible with multiple platforms.
- Graphical user interface.
- Supports attribute evaluation on ADTrees.

Main features of ADTool

- Creation of ADTrees.
- Modular display of ADTrees – necessary in case of large trees.
- Evaluation of predefined attributes, including:
  - *minimal cost of an attack,*
  - *minimal skill of the winner,*
  - *satisfiability of an attack,*
  - *cheapest satisfiable attack,*
  - *minimal attack time,*
  - *attack satisfiable in less than 10 minutes.*
- Possibility of defining new attributes.

UNIVERSITÉ DU LUXEMBOURG

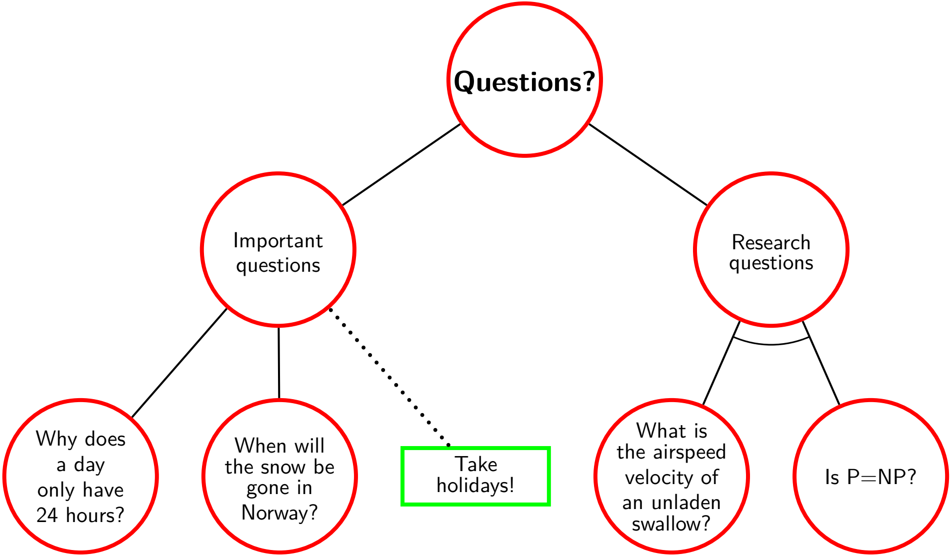
Summary

1. Attack–defense trees
2. Semantics
3. Quantitative analysis
4. Computational complexity
5. Attack–defense trees in practice

Future work

- Research questions
  - Probabilistic analysis: ADTrees & Bayesian networks
  - Access control analysis: ADTrees & policy trees
- Further testing and development of ADTool
  - Release planned for summer 2012
- Future projects: ADTrees for socio–technical security
  - EU: TREsPASS
  - CORE-FNR: STAST

Thank you for your attention!



References

[FAST'10] Kordy, Mauw, Radomirović and Schweitzer
Foundations of Attack–Defense Trees. In Proceedings of FAST 2010, volume 6561 of LNCS. Springer 2011.

[GameSec'10] Kordy, Mauw, Melissen and Schweitzer
Attack-defense trees and two-player binary zero-sum extensive form games are equivalent. In Proceedings of GameSec 2010, volume 6442 of LNCS. Springer, 2010.

[SIIS'11] Kordy, Pouly and Schweitzer
Computational Aspects of Attack–Defense Trees. In Proceedings of SIIS 2011, volume 7053 of LNCS. Springer, 2011.

[JLC'12] Kordy, Mauw, Radomirović and Schweitzer
Attack–Defense Trees. To appear in Journal of Logic and Computation.

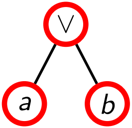
[IJSSE'12] Bagnato, Kordy, Meland and Schweitzer
Attribute Decoration of Attack–Defense Trees. To appear International Journal of Secure Software Engineering, Special Issue on Security Modeling.

APPENDIX

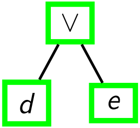
ADTrees as Boolean functions

$f(a) = a$

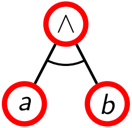
$f(d) = d$



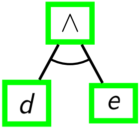
$f(a, b) = a \vee b$



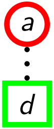
$f(d, e) = d \vee e$



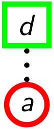
$f(a, b) = a \wedge b$



$f(d, e) = d \wedge e$



$f(a, d) = a \wedge \neg d$

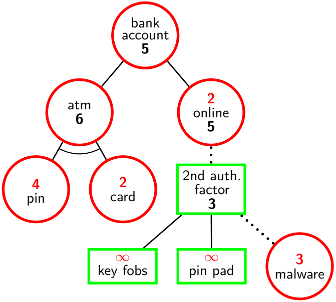


$f(d, a) = d \wedge \neg a$

Back

Example: computation of *minimal time* on an ADTree

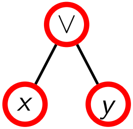
$$(\vee^A, \wedge^A, \vee^D, \wedge^D, \mathsf{c}^A, \mathsf{c}^D) = (\min, +, +, \min, +, \min)$$



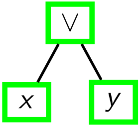
Back

α-expressions for ADTrees

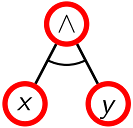
Given an attribute domain $\alpha = (\mathbb{D}, \vee^A, \wedge^A, \vee^D, \wedge^D, c^A, c^D)$ we set



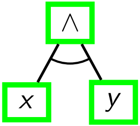
$$t_\alpha = \vee^A(x, y)$$



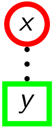
$$t_\alpha = \vee^D(x, y)$$



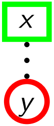
$$t_\alpha = \wedge^A(x, y)$$



$$t_\alpha = \wedge^D(x, y)$$



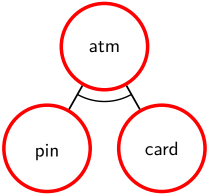
$$t_\alpha = c^A(x, y)$$



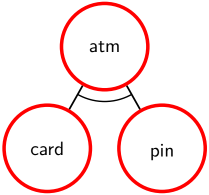
$$t_\alpha = c^D(x, y)$$

Example: *minimal_time*-expressions for ADTrees

$$time = (\mathbb{N} \cup \{\infty\}, \min, +, +, \min, +, \min)$$



$$t_{time} = \text{pin} + \text{card}$$



$$t'_{time} = \text{card} + \text{pin}$$

$$t_{time} = t'_{time} \text{ in } \mathbb{N} \cup \{\infty\}$$

because $+$ is commutative on $\mathbb{N} \cup \{\infty\}$

Semantics preserving attribute values

Definition

Attribute domain $\alpha = (\mathbb{D}, \vee^A, \wedge^A, \vee^D, \wedge^D, \mathsf{c}^A, \mathsf{c}^D)$ is compatible with semantics $\equiv$ if and only if

$$\forall t, t' \in ADTrees, \quad t \equiv t' \Rightarrow t_\alpha = t'_\alpha \text{ holds in } \mathbb{D}.$$

Theorem

If an attribute domain is compatible with a semantics, then equivalent ADTrees yield the same attribute values.

Example: incompatibility of *minimal time* with $\equiv_{\mathcal{P}}$

$$time = (\mathbb{N} \cup \{\infty\}, \min, +, +, \min, +, \min)$$

time is not compatible with the propositional semantics $\equiv_{\mathcal{P}}$

- $(a \vee^{\mathrm{p}} b) \wedge^{\mathrm{p}} a \;\; \equiv_{\mathcal{P}} a$, since $(a \vee b) \wedge a \approx a$
- but $(a \min b) + a \neq a$ in $\mathbb{N} \cup \{\infty\}$.

Checking compatibility

Definition

Attribute domain $\alpha = (\mathbb{D}, \vee^A, \wedge^A, \vee^D, \wedge^D, \mathrm{c}^A, \mathrm{c}^D)$ is compatible with semantics $\equiv$ if and only if

$$\forall t, t' \in ADTrees, \quad t \equiv t' \Rightarrow t_\alpha = t'_\alpha \text{ holds in } \mathbb{D}.$$

- Problem: How to find all t, t', such that $t \equiv t'$?
- Solution: Axiomatization of semantics

▸ Back

De Morgan lattice

L	–	non-empty set
$+, \times$	–	binary operations on L
$\neg$	–	unary operation on L

Definition

$\langle L, +, \times, \neg \rangle$ is a De Morgan lattice if $\langle L, +, \times \rangle$ is a distributive lattice and, for all $a, b \in L$, we have

$$\neg(a + b) = (\neg a) \times (\neg b), \quad \neg(a \times b) = (\neg a) + (\neg b), \quad \neg(\neg a) = a.$$

Example

De Morgan lattice $\langle \{F, M, T\}, \max, \min, \neg \rangle$, with

- $F < M < T$,
- $\neg F = T, \ \neg M = M, \ \neg T = F$,

may represent effectiveness levels.

Semantics induced by a De Morgan lattice

X = finite set of propositional variables
$\langle L, +, \times, \neg \rangle$ = De Morgan lattice

Definition

A De Morgan valuation (DMV) with domain d is a function of the form $f\colon \{0,1\}^X \to L$.

ADTrees form a representation language for De Morgan valuations:

$$f_b(X_b = v) = l_v \qquad f_{\vee^s(t_1,\dots,t_k)} = \sum_{i=1}^{k} f_{t_i},$$

$$f_{\wedge^s(t_1,\dots,t_k)} = \prod_{i=1}^{k} f_{t_i}, \qquad f_{c^s(t_1,t_2)} = f_{t_1} \times \neg f_{t_2},$$

where $v \in \{1, 0\}$, $l_v \in L$ and $s \in \{A, D\}$. Back

UNIVERSITÉ DU LUXEMBOURG