On Attack/Defense Trees

Patrick Schweitzer

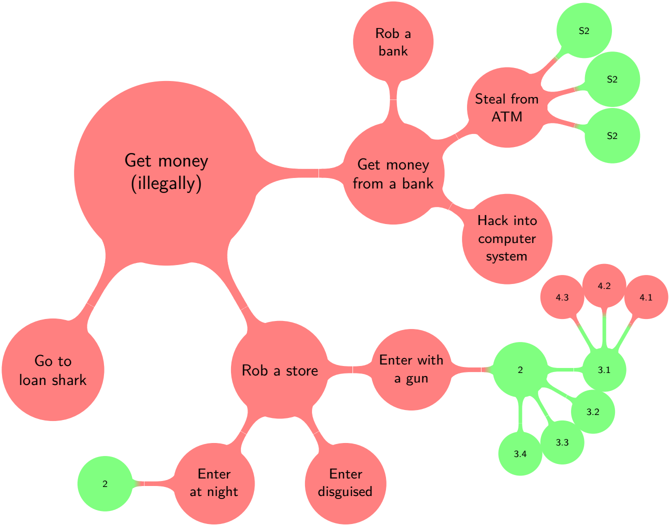
SaToSS, Faculty of Sciences, Communication and Technology

University of Luxembourg

November 17th 2009

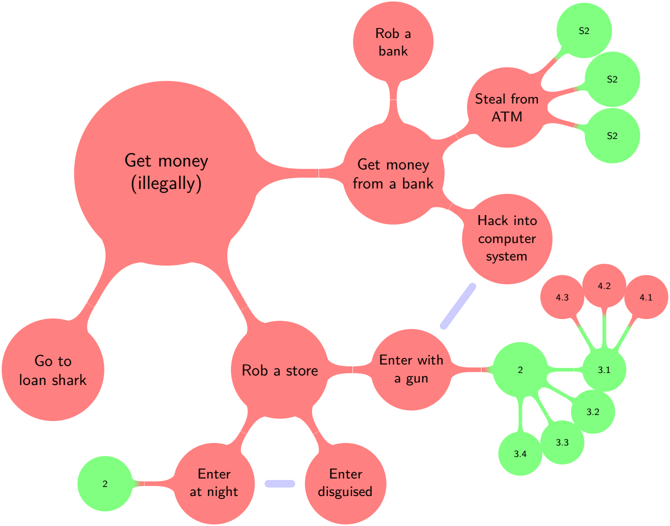
UNIVERSITÉ DU LUXEMBOURG

Intuition

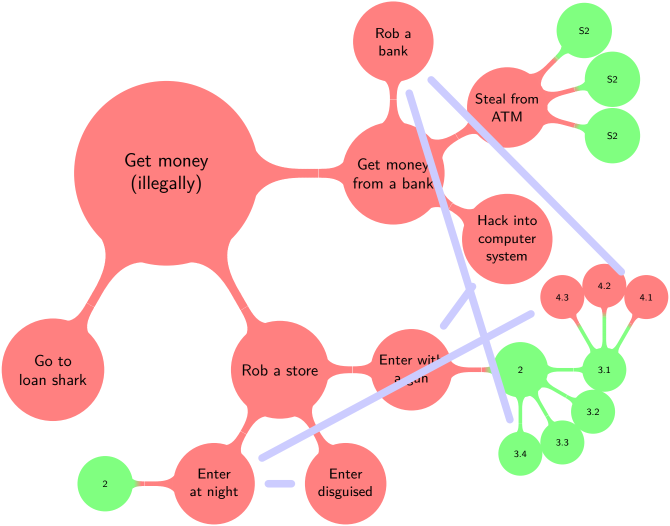


Get money (illegally)
Get money from a bank
Rob a bank
Steal from ATM
S2
S2
S2
Hack into computer system
Go to loan shark
Rob a store
Enter with a gun
2
3.1
3.2
3.3
3.4
4.1
4.2
4.3
Enter at night
2
Enter disguised

Intuition


Get money (illegally)
Get money from a bank
Rob a bank
Steal from ATM
S2
S2
S2
Hack into computer system
Go to loan shark
Rob a store
Enter with a gun
2
3.1
3.2
3.3
3.4
4.1
4.2
4.3
Enter at night
2
Enter disguised

Intuition


Rob a bank
S2
S2
S2
Steal from ATM
Get money (illegally)
Get money from a bank
Hack into computer system
4.3
4.2
4.1
Go to loan shark
Rob a store
2
3.1
3.2
3.3
3.4
2
Enter at night
Enter disguised

Guide to modeling attacks

Intuitive start:	A mindmap (a special graph)
Problem:	Complexity
Solution:	Computer support (requires formalism)
Literature:	Several approaches

Guide to modeling attacks

Intuitive start:	A mindmap (a special graph)
Problem:	Complexity
Solution:	Computer support (requires formalism)
Literature:	Several approaches

Guide to modeling attacks

Intuitive start:	A mindmap (a special graph)
Problem:	Complexity
Solution:	Computer support (requires formalism)
Literature:	Several approaches

Guide to modeling attacks

Intuitive start:	A mindmap (a special graph)
Problem:	Complexity
Solution:	Computer support (requires formalism)
Literature:	Several approaches

Different approaches to modeling attacks

- Attack Trees
  Essentially all information is contained in the leaves.
- Attack Graphs or Attack Nets
  Finite automata that fulfill security properties;
  separation of data and processes
- Security Pattern Descriptions
  Documents that describe in words the possible attacks on a system. They are very long exactly like this text which should never have been on the slide because nobody that listens to the talk reads that much text.
- ...

Different approaches to modeling attacks

- Attack Trees
  Essentially all information is contained in the leaves.
- Attack Graphs or Attack Nets
  Finite automata that fulfill security properties;
  separation of data and processes
- Security Pattern Descriptions
  Documents that describe in words the possible attacks on a system. They are very long exactly like this text which should never have been on the slide because nobody that listens to the talk reads that much text.
- . . .

Different approaches to modeling attacks

- Attack Trees
  Essentially all information is contained in the leaves.
- Attack Graphs or Attack Nets
  Finite automata that fulfill security properties;
  separation of data and processes
- Security Pattern Descriptions
  Documents that describe in words the possible attacks on a system. They are very long exactly like this text which should never have been on the slide because nobody that listens to the talk reads that much text.
- . . .

Attack Trees - the concept

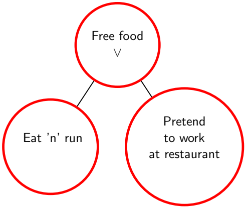
Attack: How to get free food?

Attack Trees - the concept

Attack: How to get free food?

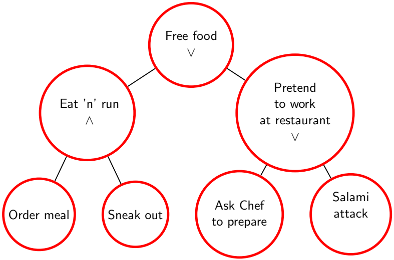
Attack Trees - the concept

Attack: How to get free food?



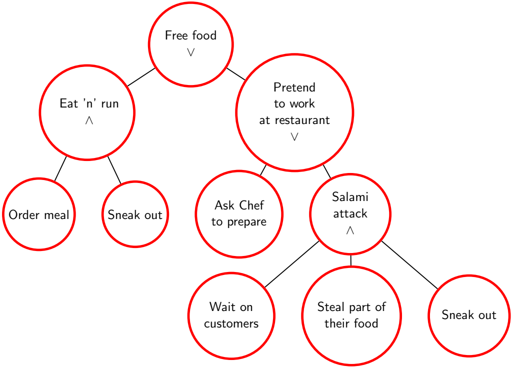
Attack Trees - the concept

Attack: How to get free food?



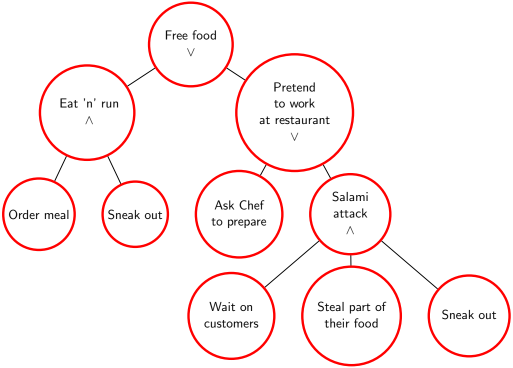
Attack Trees - the concept

Attack: How to get free food?



Attack Trees - the concept

Attack: How to get free food?



Essentially a set of multisets, e.g.:

{⦃Order meal, sneak out⦄,
⦃Ask Chef to prepare⦄,
⦃Wait on customers,
steal part of their food,
sneak out⦄}

Properties of the existing model

Important properties of Attack Trees

- Uses and and or nodes
- Simple normal form: trees of depth 1
- Attributes can be attached to the leaves:
  then the attribute can be calculated for the root
- Projection only works for some attributes
  (Projection = Restriction of an attribute)

Properties of the existing model

Important properties of Attack Trees

- Uses and and or nodes
- Simple normal form: trees of depth 1
- Attributes can be attached to the leaves:
  then the attribute can be calculated for the root
- Projection only works for some attributes
  (Projection = Restriction of an attribute)

Properties of the existing model

Important properties of Attack Trees

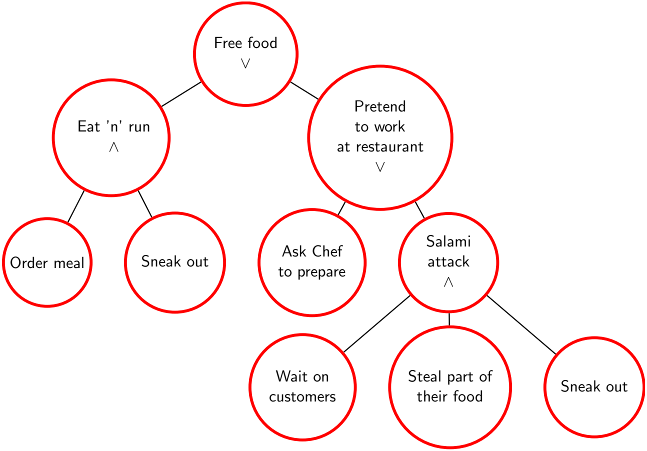
- Uses and and or nodes
- Simple normal form: trees of depth 1
- Attributes can be attached to the leaves:
  then the attribute can be calculated for the root
- Projection only works for some attributes
  (Projection = Restriction of an attribute)

Properties of the existing model

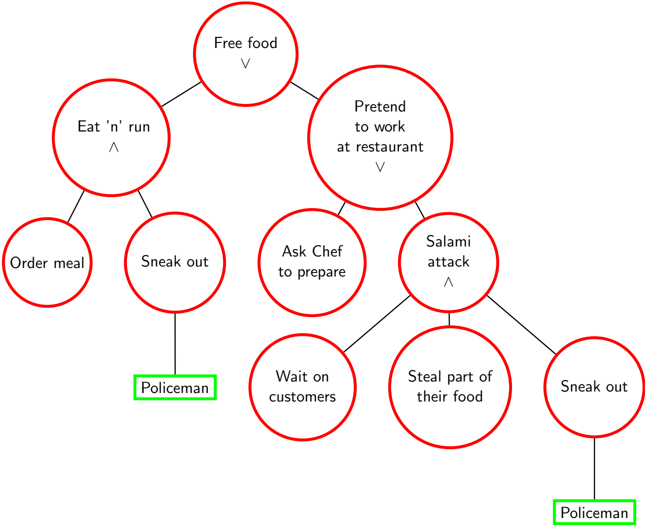
Important properties of Attack Trees

- Uses and and or nodes
- Simple normal form: trees of depth 1
- Attributes can be attached to the leaves: then the attribute can be calculated for the root
- Projection only works for some attributes (Projection = Restriction of an attribute)

Including a defense in the framework



Including a defense in the framework



Attack and Defense Trees

- Consider the Defense Tree 'law enforcement' instead of a policeman.
- Consider the Attack Tree 'Mafia' attached to law enforcement.
- and so on...
- New framework: Attack Tree - Defense Tree - Attack Tree - ...

Attack and Defense Trees

- Consider the Defense Tree 'law enforcement' instead of a policeman.
- Consider the Attack Tree 'Mafia' attached to law enforcement.
- and so on...
- New framework: Attack Tree - Defense Tree - Attack Tree - ...

Attack and Defense Trees

- Consider the Defense Tree 'law enforcement' instead of a policeman.
- Consider the Attack Tree 'Mafia' attached to law enforcement.
- and so on...
- New framework: Attack Tree - Defense Tree - Attack Tree - ...

The general idea: two functions describing the nodes

Structure: rooted tree $T = (V, E, r, \tau, \phi)$
(non-empty, finite, directed, connected, acyclic, rooted)
Type: $\tau\colon V \to \{\bigcirc, \square, \Diamond\}$ Connector $\phi\colon V \to \{\vee, \wedge, \neg, -\}$

The general idea: two functions describing the nodes

Structure: rooted tree $T = (V, E, r, \tau, \phi)$
(non-empty, finite, directed, connected, acyclic, rooted)
Type: $\tau\colon V \to \{\bigcirc, \square, \Diamond\}$ Connector $\phi\colon V \to \{\vee, \wedge, \neg, -\}$

$$\tau(v) \in \{\bigcirc, \square\} \Longrightarrow \tau(w) \in \{\tau(v), \Diamond\} \quad (1)$$

$$\tau(v) \in \{\bigcirc, \square\} \text{ and } |\,\text{Children}_v\,| > 1 \Longleftrightarrow \phi(v) \in \{\vee, \wedge\} \quad (2)$$

$$\tau(v) \in \{\bigcirc, \square\} \text{ and } |\,\text{Children}_v\,| \leq 1 \Longleftrightarrow \phi(v) = - \quad (3)$$

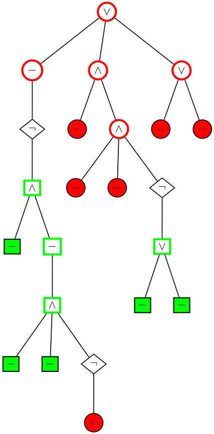
$$\tau(v) = \Diamond \Longrightarrow \tau(w) \in \{f(v), \Diamond\} \quad (4)$$

$$\tau(v) = \Diamond \Longrightarrow |\,\text{Children}_v\,| = 1 \quad (5)$$

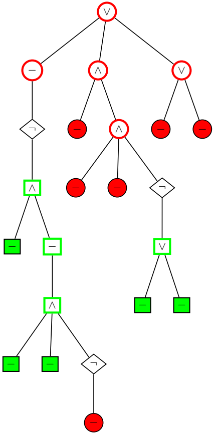
$$\tau(v) = \Diamond \Longleftrightarrow \phi(v) = \neg \quad (6)$$

$v, w \in V$ and $(v, w) \in E$

The additional properties



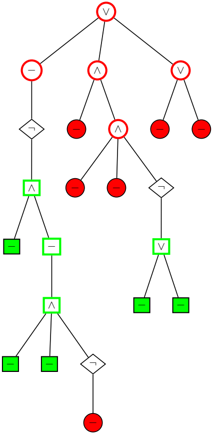
The additional properties



Property (1):

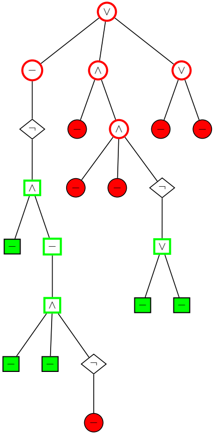
$\tau(v) \in \{\bigcirc, \square\} \implies \tau(w) \in \{\tau(v), \Diamond\}$

The additional properties



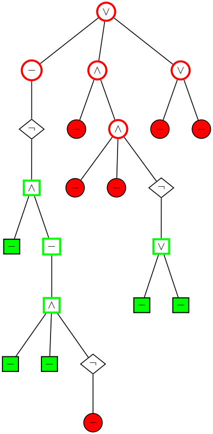
Property (2):
$\tau(v) \in \{\bigcirc, \square\}$ and $|\,\mathrm{Children}_v\,| > 1$
$\Longleftrightarrow$
$\phi(v) \in \{\vee, \wedge\}$

The additional properties



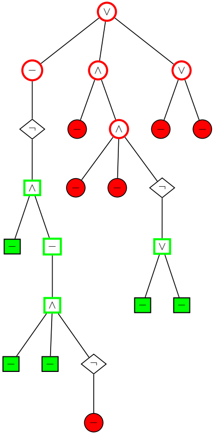
Property (3):
$\tau(v) \in \{\bigcirc, \square\}$ and $|\text{Children}_v| \leq 1$
$\Longleftrightarrow$
$\phi(v) = -$

The additional properties



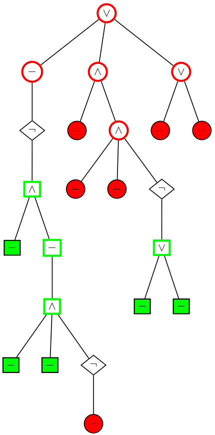
Property (4):
$\tau(v) = \Diamond \Longrightarrow \tau(w) \in \{f(v), \Diamond\}$

The additional properties



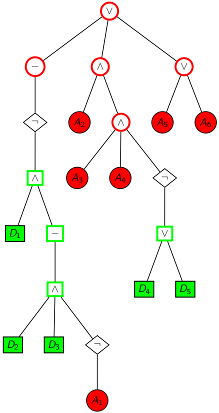
Property (5):
$\tau(v) = \Diamond \implies |\,\text{Children}_v\,| = 1$

The additional properties



Property (6):
$\tau(v) = \Diamond \iff \phi(v) = \neg$

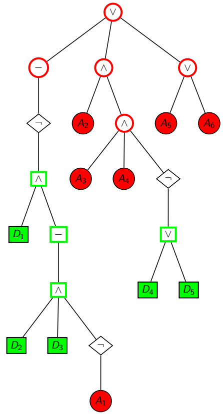
Semantics of the Adtrees



Semantics of the adtree:
Unique variable associated to leaf

$$
[\![v]\!] = \begin{cases} v & \text{if } v \in L(T), \\ \bigvee_{w \in \mathrm{Children}_v} [\![w]\!] & \text{if } \phi(v) = \vee, \\ \bigwedge_{w \in \mathrm{Children}_v} [\![w]\!] & \text{if } \phi(v) = \wedge, \\ [\![w]\!] & \text{if } \phi(v) = - \text{ and} \\ & \mathrm{Children}_v = \{w\}, \\ \neg [\![w]\!] & \text{if } \phi(v) = \neg \text{ and} \\ & \mathrm{Children}_v = \{w\}. \end{cases}
$$

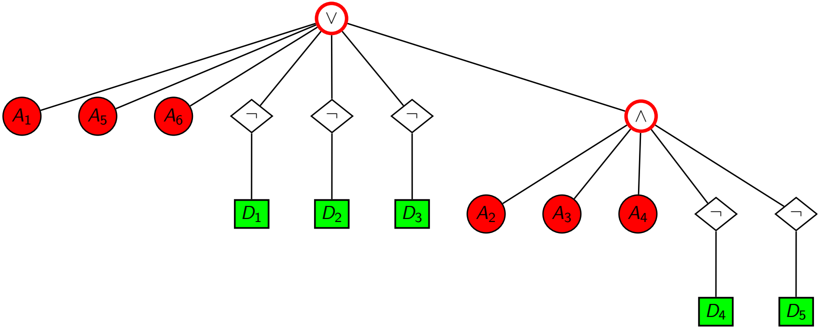
Logical formulas associated to trees



Propositional logic corresponding to the tree:

$$((\neg(D_1 \wedge ((D_2 \wedge D_3 \wedge (\neg A_1))))))\vee$$
$$(A_2 \wedge (A_3 \wedge A_4 \wedge (\neg(D_4 \vee D_5)))\vee$$
$$(A_5 \vee A_6)$$

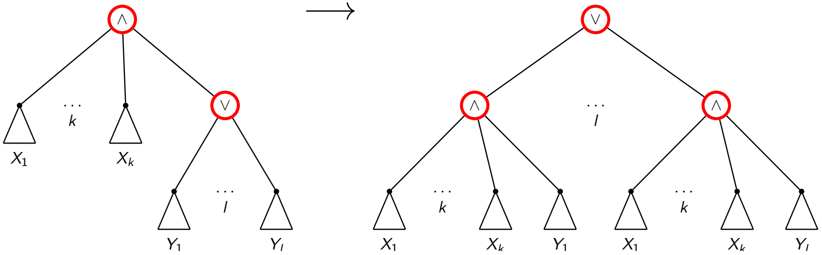
Trees in normal form



Normal form:

$A_1 \vee A_5 \vee A_6 \vee \neg D_1 \vee \neg D_2 \vee \neg D_3 \vee (A_2 \wedge A_3 \wedge A_4 \wedge \neg D_4 \wedge \neg D_5)$

Exemplary transformation: Distributivity $\wedge$ to $\vee$



With $k \geq 1$ and $l \geq 2$

Full set of transformation rules

• Distributivity	$(A \vee B) \wedge C$	$\rightarrow (A \wedge C) \vee (B \wedge C)$
• 1−level absorption	$(A \wedge B) \vee A$	$\rightarrow A$
• 2−level absorption	as above	
• Double negation	$\neg\neg A$	$\rightarrow A$
• Empty refinement	no formula	
• Associativity ($\vee$ and $\wedge$)	$(A \vee B) \vee C$	$\rightarrow A \vee B \vee C$
• De Morgan ($\vee$ and $\wedge$)	$\neg(A \vee B)$	$\rightarrow \neg A \wedge \neg B$
• Idempotency ($\vee$ and $\wedge$)	$X \vee X$	$\rightarrow X$

Full set of transformation rules

- Distributivity $(A \vee B) \wedge C \rightarrow (A \wedge C) \vee (B \wedge C)$
- 1−level absorption $(A \wedge B) \vee A \rightarrow A$
- 2−level absorption as above
- Double negation $\neg\neg A \rightarrow A$
- Empty refinement no formula
- Associativity ($\vee$ and $\wedge$) $(A \vee B) \vee C \rightarrow A \vee B \vee C$
- De Morgan ($\vee$ and $\wedge$) $\neg(A \vee B) \rightarrow \neg A \wedge \neg B$
- Idempotency ($\vee$ and $\wedge$) $X \vee X \rightarrow X$

Full set of transformation rules

• Distributivity	$(A \vee B) \wedge C$	$\rightarrow (A \wedge C) \vee (B \wedge C)$
• 1−level absorption	$(A \wedge B) \vee A$	$\rightarrow A$
• 2−level absorption	as above	
• Double negation	$\neg\neg A$	$\rightarrow A$
• Empty refinement	no formula	
• Associativity ($\vee$ and $\wedge$)	$(A \vee B) \vee C$	$\rightarrow A \vee B \vee C$
• De Morgan ($\vee$ and $\wedge$)	$\neg(A \vee B)$	$\rightarrow \neg A \wedge \neg B$
• Idempotency ($\vee$ and $\wedge$)	$X \vee X$	$\rightarrow X$

Currently working on

Proving the uniqueness of the normal forms

Requires:

- Strong termination (Patrick - almost finished)
  Applying rules indefinitely is not possible
- Local confluence (Barbara - finished)
  Order of applying the rules leads to same result

Currently working on

Proving the uniqueness of the normal forms

Requires:
- Strong termination (Patrick - almost finished)
  Applying rules indefinitely is not possible
- Local confluence (Barbara - finished)
  Order of applying the rules leads to same result

Currently working on

Proving the uniqueness of the normal forms

Requires:
- Strong termination (Patrick - almost finished)
  Applying rules indefinitely is not possible
- Local confluence (Barbara - finished)
  Order of applying the rules leads to same result

Termination function

Termination function:
A function from the trees into a totally ordered set,
s.t. the value before applying a transformation rule $>$
the value after applying a transformation rule.

Termination function

Termination function:
A function from the trees into a totally ordered set, s.t. the value before applying a transformation rule $>$ the value after applying a transformation rule.

Whiteboard

Work on generalizing the framework

- Introduce attributes to the leaves
- Allow directed acyclic graphs
- Consider temporal order of children
- Check out the two existing software packages
- ...

1. Intuition and overview of existing approaches to model attacks
2. Attack Trees
3. The new approach to include defenses
4. Future work