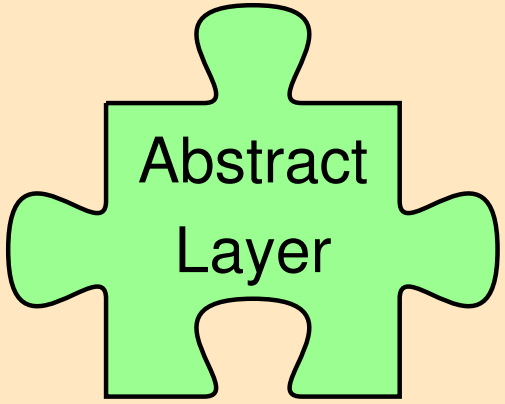
The Task-Layer

Dominik Dietrich

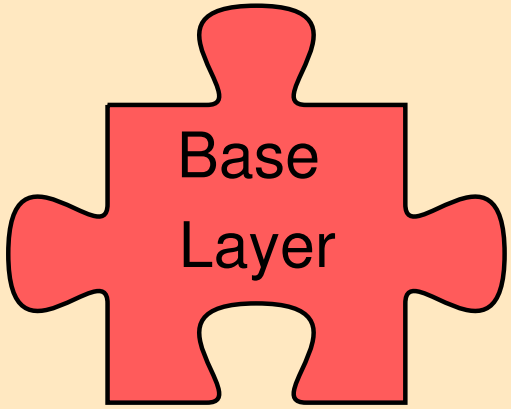
dodi@ags.uni-sb.de

Saarland University, Saarbrücken

The Task-Layer

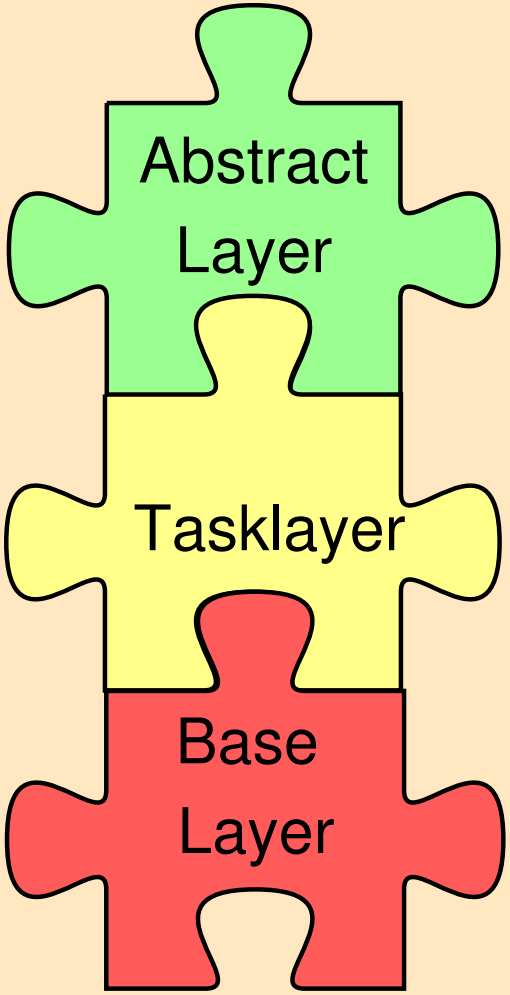


MULTI, Ω-ANTS, user



CORE

The Task-Layer



MULTI, Ω-ANTS, user

- central proof interface
  - proof forest, containing tasks
  - inferences to reduce task to other tasks

CORE

Definition of Tasks

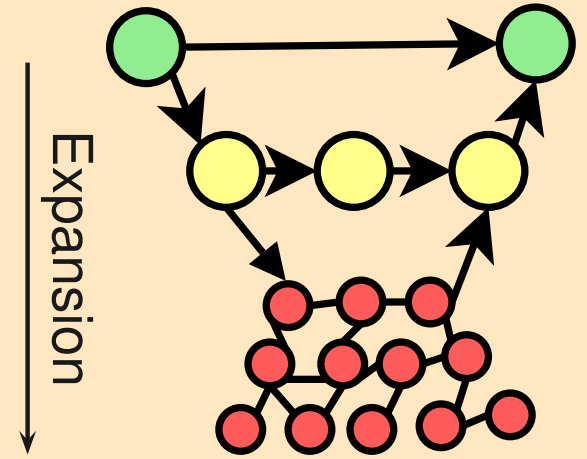
Tasks:

- tasks $\varphi_1, \ldots \varphi_n \rhd \underline{\psi_1}, \psi_2, \ldots, \psi_m$: multi-conclusion sequents with a selected focus of attention
- free variable representation using γ and δ variables
- Advantages:
  - compact representation of available knowledge

Inference steps

- uniform language for user and reasoning tools MULTI, Ω-ANTS

Inference step SIMPLIFY	
premises	$\underbrace{P}_{\text{name}} : \Phi$ $\overbrace{\text{arith-term}(P_{\|\pi})}^{\text{local condition}}$
conlcusion	$\underbrace{C}_{\text{name}} : \Psi$ arith-term$(C_{\|\pi})$
parameter	π:position
check	$C_{\|\pi} =$ $\overbrace{\text{maple-simplify}(P_{\|\pi})}^{\text{global condition}}$ or $P_{\|\pi}$ =maple-simplify$(C_{\|\pi})$
expansion	expand-simplify



Starting the proof

- task:
  - to start a new proof for
    If f, g cont. in a
    then $(f + g)$ cont. in a

Starting the proof

- task:
  - to start a new proof for
    If f, g cont. in a
    then $(f + g)$ cont. in a
- steps:
  - create a new proof tree
  - create initial task from
    assumptions and conjecture

cont. f in a, cont. g in $a \rhd \underline{\text{cont. } (f + g) \text{ in } a}$

Alternative proof ideas

- task
  - alternative proof ideas

Alternative proof ideas

- task
  - alternative proof ideas

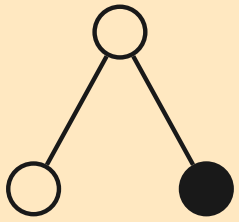
- first approach: $\epsilon - \delta$ criteria for continuity
  - $|x - a| < \delta_f \Rightarrow |f(x) - f(a)| < \epsilon_f$
    $|x - a| < \delta_g \Rightarrow |g(x) - g(a)| < \epsilon_g$
    $\triangleright \underline{|x - a| < \delta_{f+g} \Rightarrow (f + g)(x) - (f + g)(a)| < \epsilon_{f+g}}$

Alternative proof ideas

- task
  - alternative proof ideas

- first approach: $\epsilon - \delta$ criteria for continuity
  - $|x - a| < \delta_f \Rightarrow |f(x) - f(a)| < \epsilon_f$
    $|x - a| < \delta_g \Rightarrow |g(x) - g(a)| < \epsilon_g$
    $\triangleright \underline{|x - a| < \delta_{f+g} \Rightarrow (f + g)(x) - (f + g)(a)| < \epsilon_{f+g}}$
- second approach: neighbourhood characterisation:
  - $nbd(Y_f, f(a)) \Rightarrow nbd(X_f, a) f(X_f) \subseteq Y_f$
    $nbd(Y_g, g(a)) \Rightarrow nbd(X_g, a) f(X_g) \subseteq Y_g$
    $\triangleright \underline{nbd(Y_{f+g}, (f + g)(a)) \Rightarrow nbd(X_{f+g}, a) f(X_{f+g}) \subseteq Y_{f+g}}$

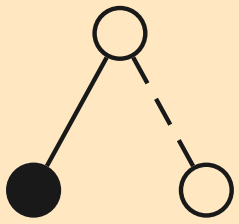
Alternative proof ideas



- task
  - alternative proof ideas
- solution:
  - allow OR branches

- first approach: $\epsilon - \delta$ criteria for continuity
  - $|x - a| < \delta_f \Rightarrow |f(x) - f(a)| < \epsilon_f$
    $|x - a| < \delta_g \Rightarrow |g(x) - g(a)| < \epsilon_g$
    $\triangleright \underline{|x - a| < \delta_{f+g} \Rightarrow (f + g)(x) - (f + g)(a)| < \epsilon_{f+g}}$
- second approach: neighbourhood characterisation:
  - $nbd(Y_f, f(a)) \Rightarrow nbd(X_f, a) f(X_f) \subseteq Y_f$
    $nbd(Y_g, g(a)) \Rightarrow nbd(X_g, a) f(X_g) \subseteq Y_g$
    $\triangleright \underline{nbd(Y_{f+g}, (f + g)(a)) \Rightarrow nbd(X_{f+g}, a) f(X_{f+g}) \subseteq Y_{f+g}}$

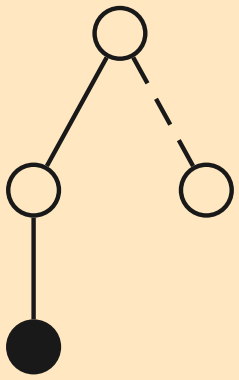
Applying CORE assertions



- Apply definition
  $(f+g)(x) = f(x) + g(x)$
- Rewrite:
  $f(x) + g(x) - (f(a) + g(a))$
  $= f(x) - f(a) + g(x) - g(a)$

$\text{ass}_1, \text{ass}_2 \quad \triangleright \quad \underline{|x-a| < \delta_{f+g} \Rightarrow |(f+g)(x) - (f+g)(a)| < \epsilon_{f+g}}$

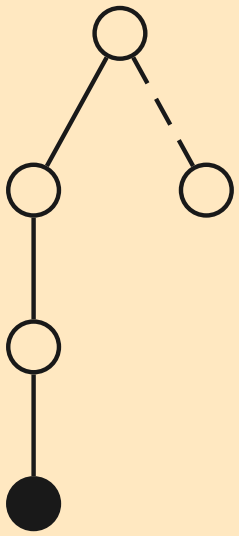
Applying CORE assertions



- Apply definition
  $(f+g)(x) = f(x) + g(x)$
- Rewrite:
  $f(x) + g(x) - (f(a) + g(a))$
  $= f(x) - f(a) + g(x) - g(a)$

$\text{ass}_1, \text{ass}_2 \quad \triangleright \quad \underline{|x-a| < \delta_{f+g} \Rightarrow |f(x) + g(x) - (f(a) + g(a))| < \epsilon_{f+g}}$

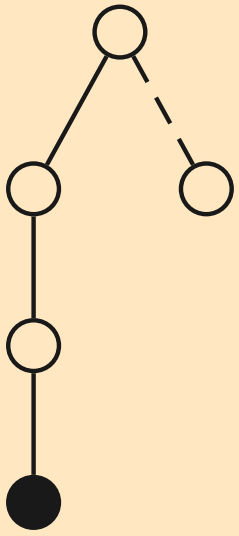
Applying CORE assertions



- Apply definition
  $(f+g)(x) = f(x) + g(x)$
- Rewrite:
  $f(x) + g(x) - (f(a) + g(a))$
  $= f(x) - f(a) + g(x) - g(a)$

$\mathsf{ass}_1, \mathsf{ass}_2 \rhd \underline{|x-a| < \delta_{f+g} \Rightarrow |f(x) - f(a) + g(x) - g(a)| < \epsilon_{f+g}}$

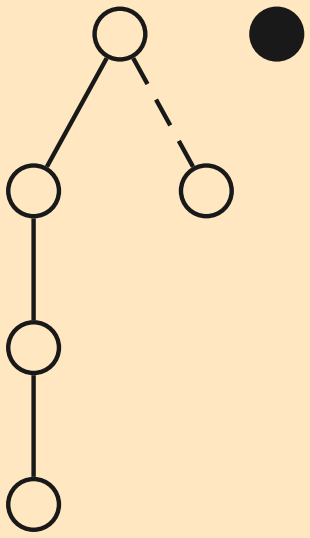
Lemmatization



- problem:
  - need Δ-inequality, postpone proof

$\text{ass}_1, \text{ass}_2 \rhd \underline{|x - a| < \delta_{f+g} \Rightarrow |f(x) - f(a) + g(x) - g(a)| < \epsilon_{f+g}}$

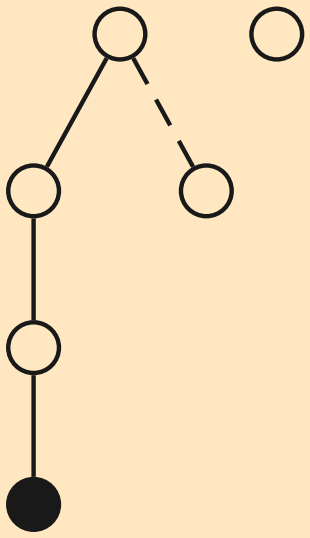
Lemmatization



$\triangleright \underline{|x+y| \leq |x| + |y|}$

- problem:
  - need Δ-inequality, postpone proof
- solution:
  - create a new proof tree
  - prove whenever you want

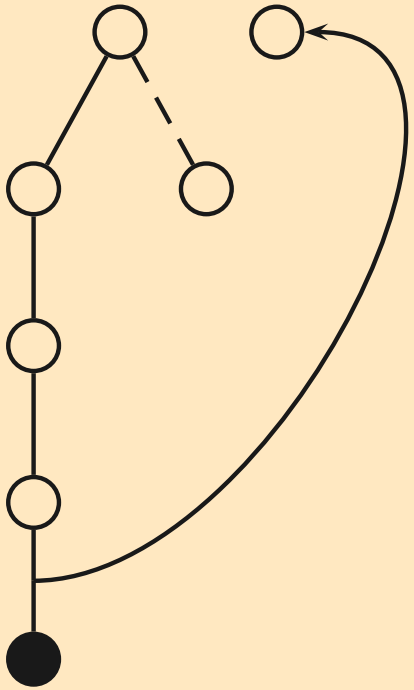
Lemma Application



- problem:
  - apply lemma

$$\mathsf{ass}_1, \mathsf{ass}_2 \rhd |x-a| < \delta_{f+g} \Rightarrow |\underbrace{f(x)-f(a)}_{X} + \underbrace{g(x)-g(a)}_{Y}| < \epsilon_{f+g}$$

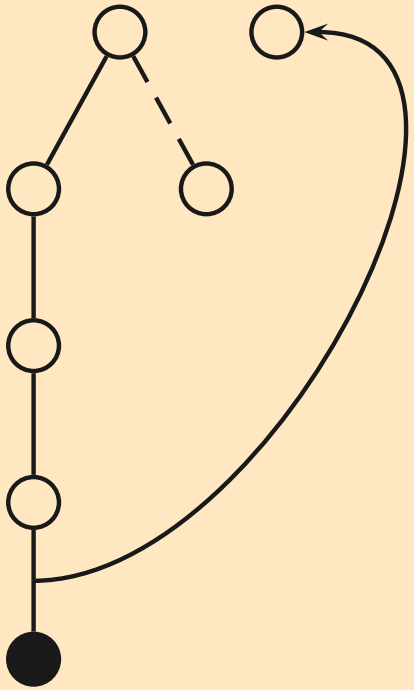
Lemma Application



- problem:
  - apply lemma
- solution:
  - insert a forest link

$$\text{ass}_1, \text{ass}_2 \rhd \underline{|x-a| < \delta_{f+g} \Rightarrow |f(x)-f(a)| + |g(x)-g(a)| < \epsilon_{f+g}}$$

Using Assumptions

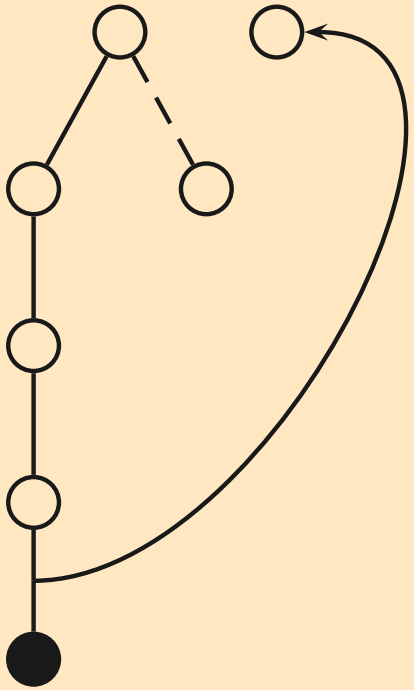


- observation
  - subformula of assumption in goal

$$|x - a| \leq \delta_f \Rightarrow |f(x) - f(a)| < \epsilon_f, \text{ass}_2$$

$$\triangleright \underline{|x - a| < \delta_{f+g} \Rightarrow |f(x) - f(a)| + |g(x) - g(a)| < \epsilon_{f+g}}$$

Using Assumptions

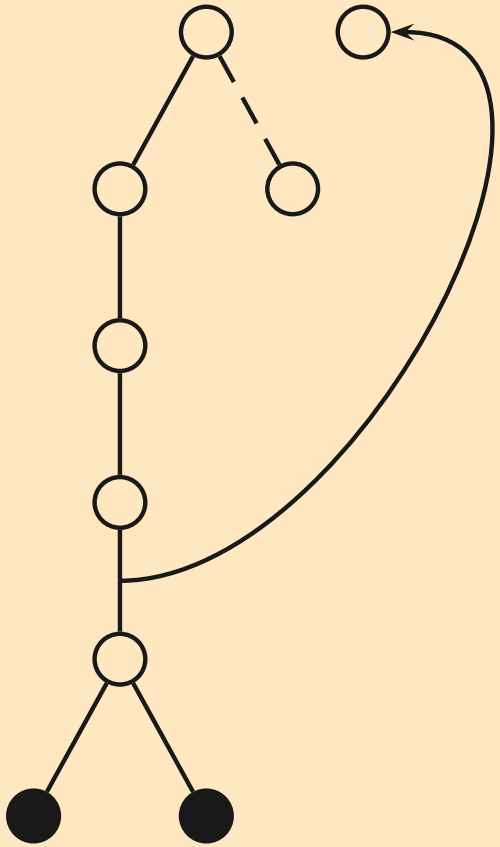


- observation
  - subformula of assumption in goal
- solution
  - automatically extract subformula

$|x-a| \leq \delta_f \Rightarrow |f(x)-f(a)| < \epsilon_f, \text{ass}_2$

$\triangleright$ $|x-a| < \delta_{f+g} \Rightarrow |f(x)-f(a)| + |g(x)-g(a)| < \epsilon_{f+g}$

Using Assumptions



- observation
  - subformula of assumption in goal
- solution
  - automatically extract subformula

$|x - a| < \delta_{f+g}, ass_1, ass_2 \rhd \underline{|x - a| < \delta_f}$

$\mathsf{ass}_1, \mathsf{ass}_2 \rhd \underline{|x - a| < \delta_{f+g} \Rightarrow \epsilon_f + |g(x) - g(a)| < \epsilon_{f+g}}$

Future Work

- Extend task layer to manage constraints
  - substitutions
  - numerical constraints
- Adapt proof planner
- Provide control knowledge for new choice points
  - which alternative to take?
  - when to do parallel exploration?
  - which subformulae to prefer?

Comparison to old ΩMEGA

■ direct access to calculus	■ independent abstract layer ⊕ changing of calculus possible
■ single proof tree ■ alternative granularity	■ proof forest ⊕ lemmatization ■ alternative granularity ■ alternative proof ideas ⊕ parallel exploration
■ commands, agents, methods, tactics	■ one general language ⊕ reducing specification work by 70%
■ Matching on whole formula	■ Matching subformulae ⊕ shorter proofs ⊕ less auxiliary methods needed ⊖ more possibilities to apply methods