Fast plausibility checks of image interpretation maps

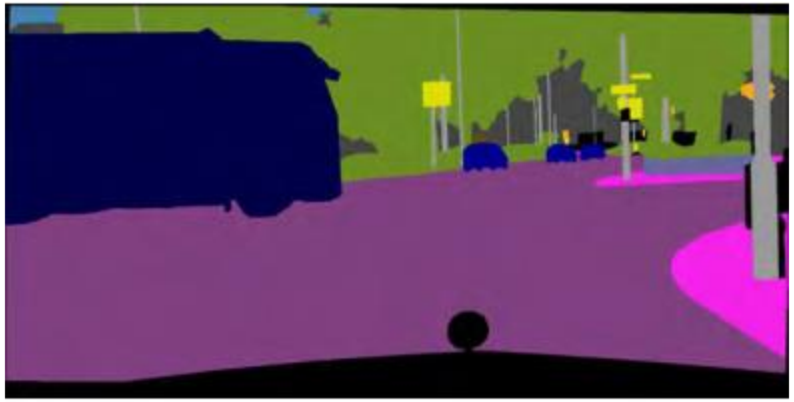
Hardi Hungar
German Aerospace Center
Related to: PhD thesis of Paulin Pekezou Fouopi

DLR

Problem setting

- Image interpretation (for driving assistance or automation)
  - Segmentation of the image into areas corresponding to types of objects: (road, sky, car, lorry, etc.)
  - Segmentation is performed by a (fully convolutional) neural net

Camera image from Cordts et al.



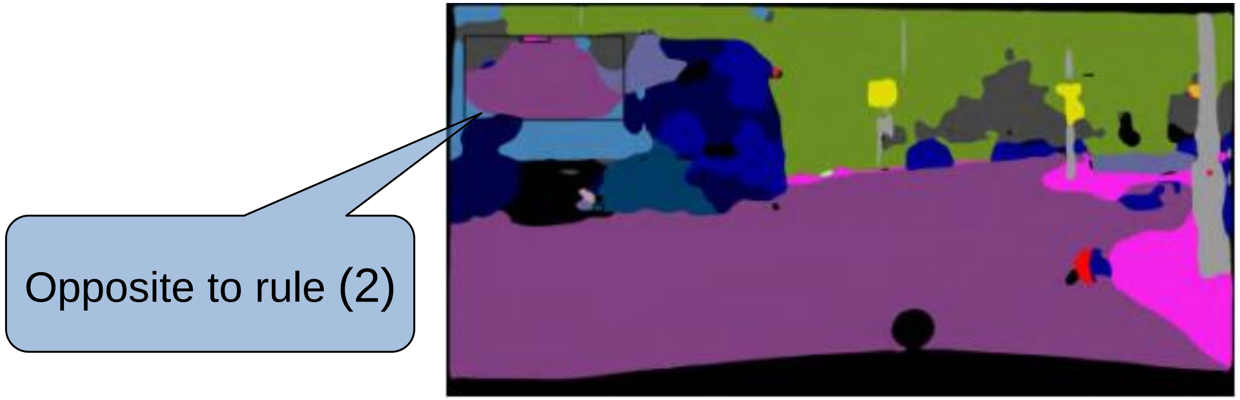
Segmentation Labels from Cordts et al.
„Ground truth“

Road segment inside lorry

Neural net segmentation result

Cordts et al., *The Cityscapes Dataset for Semantic Urban Scene Understanding*, CVPR 2016
Images: Pekezou et al., *Holistic Scene and Situation Assessment Based on Sub-Symbolic, … Approaches*, VDI Tagung, Nürnberg 2018

Problem setting



- Plausibility check of neural net results using a probabilistic knowledge base
  - Logical Rules with probabilistic weights
  - Rule examples:

$$\text{Road}(x) \wedge \text{Sky}(y) \wedge \text{Below}(x,y) \rightarrow \text{Consistent}(x) \quad (1) \qquad (0.45)$$

$$\text{Car}(x) \wedge \text{Road}(y) \wedge \text{Inside}(x,y) \rightarrow \text{Consistent}(x) \wedge \text{Consistent}(y) \quad (2) \qquad (0.37)$$

  - The rules are **probabilistic**, because it is very difficult to formulate statements which are **absolutely true**

Probabilistic Knowledge Bases and their semantics (simplified)

Syntax

- A probabilistic knowledge base (PKB) is given by a set of weighted rules $F_1, \ldots, F_f$
- Weighted rules are formulas made up from predicates P(x), P(x,y)
  - Like Car(x), Inside(x,y), Consistent(x)
- Weights w_i are real numbers (also negative number are permitted) expressing the confidence in the rule
  - Negative means: confidence in the negation

Semantics

- A set of constants $c_1, c_2, \ldots$
  - In our case: constants denote segments of an image
- Assertions $P(c_1)$, $P(c_1,c_2)$: Ground predicates
- Interpretation ***I*** assigning 0 or 1 (for *false* and *true*, resp.) to assertions
  - In our case: a segmentation
- An interpretation of a plausibility probability
- $\mathrm{Prob}(I) = (1/Z)\exp(\sum_{i=1}^{f} w_i\, n_i(I))$, where
  - f number of rules
  - w_i weight of rule i
  - $n_i(I)$ number of true assignments of rule i
  - Z norming factor (sum over all **I**)

DLR

Evaluating PKBs

Computing the plausibility of some segmentation I

$$Prob(I) = (1/Z) \exp(\sum^{f}_{i=1} w_i \, n_i(I)),$$

means essentially computing the $n_i(I)$ – the true groundings of the left-hand-sides of the rules:

- Z depends only on the number of constants in I (maximal number of segments)
- Exp(…) is only a scaling function
- Multiplication w_i with and summation $\sum^{f}_{i=1}$ over the rules is a very fast operation

But computing the $n_i(I)$ entails the application of all rules to all combinations of segments – this is computationally expensive

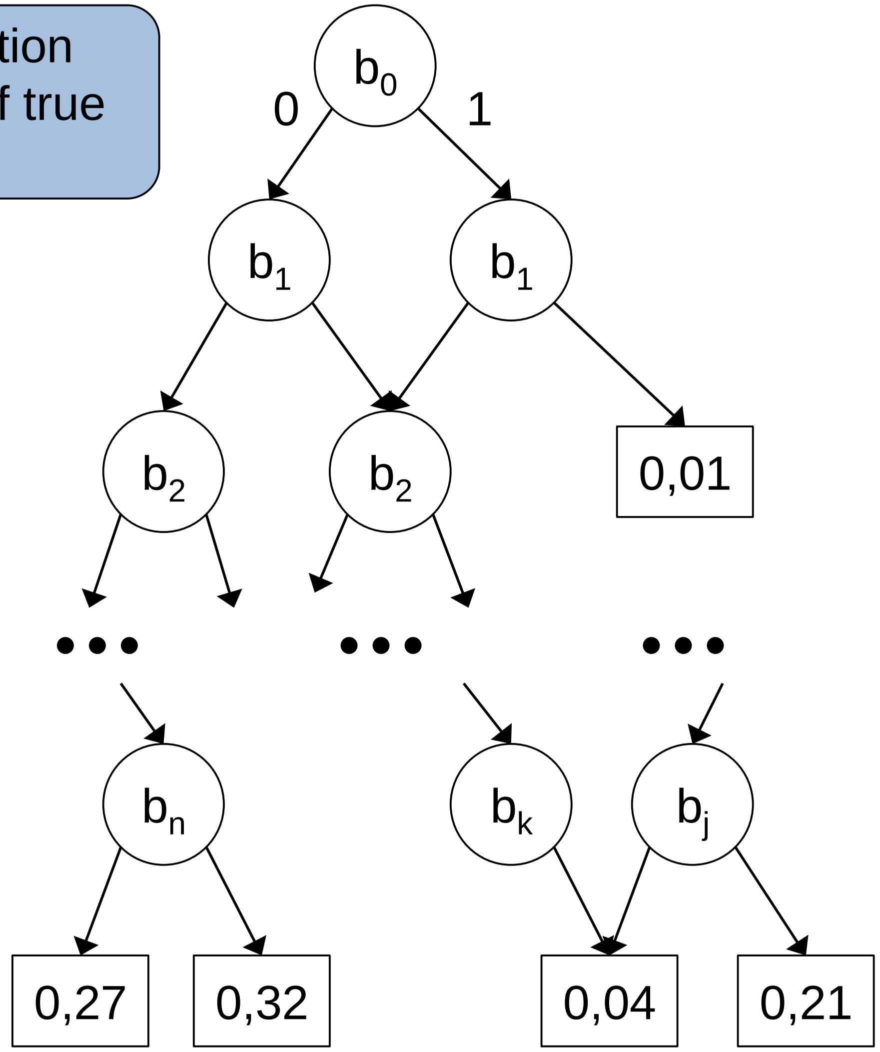
The idea behind the “fast plausibility check“:
Compute the combinational part beforehand.
I.e., make a (cleverly represented) table of the potential logical combinations.

Using ADDs for fast plausibility computation
ADD definition

The “clever” table format

Here: from segmentation properties to number of true rule instances

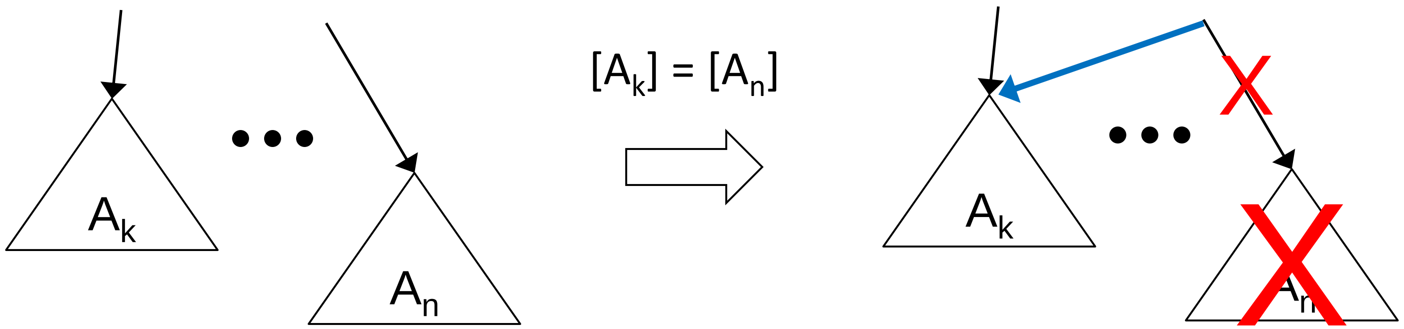
- Algebraic Decision Diagrams (ADDs) are
  - rooted, directed, acyclic graphs
  - with
    - binary inner nodes labeled by variables $b \in B$
      - Nodes have 0- and 1-successors
    - leaves labeled by values $v \in V$
- An ADD A represents a function from an interpretation of the binary variables to the set of values
  - $[A] : I(B) \rightarrow V$
  - $[v]$ = constant v
  - $[(b_k, 0 \rightarrow A_0, 1 \rightarrow A_1)] =$ **if** $I(b_k)$ **then** $[A_0]$ **else** $[A_1]$



Bahar et al., *Algebraic Decision Diagrams and their Applications*. Formal Methods in System Design 1997

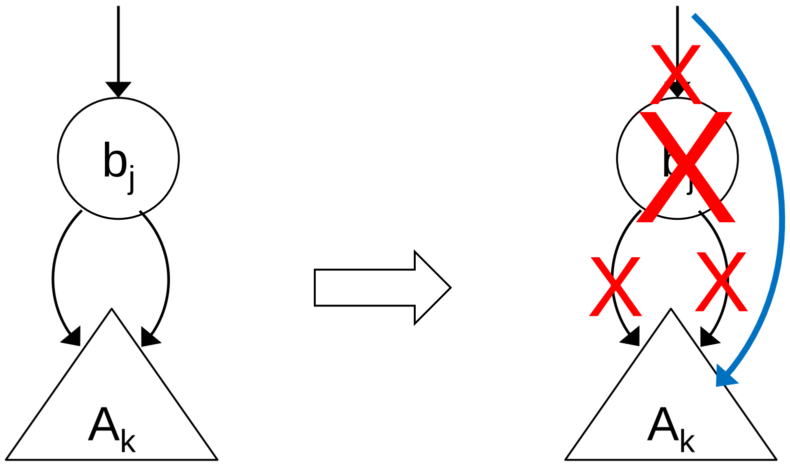
Using ADDs for fast plausibility computation RO-ADDs: Reduced ordered ADDs

- Each ADD may be <u>reduced</u> (without changing ist semantics): by
  - eliminating semantically identical nodes



  - Eliminating irrelevant inner nodes

Part of the “clever”

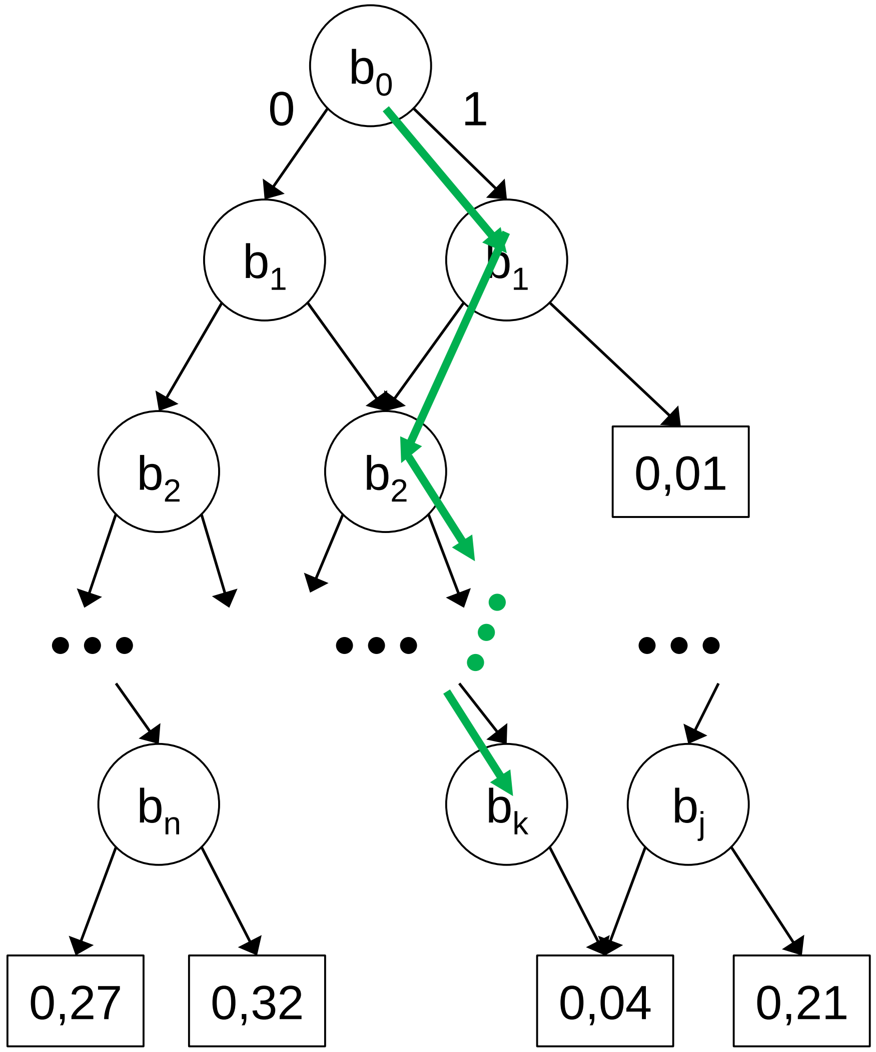


- Reducing saves space

Using ADDs for fast plausibility computation
RO-ADDs: Reduced ordered ADDs

- An ADD is ordered, if on each path, the variables labeling the inner nodes appear according to one fixed order
  - **$b_0 < b_1 < b_2 < \ldots < b_k$**
- Ordering makes several operations on ADDs simple
  - These include combinations of functions via operations like addition

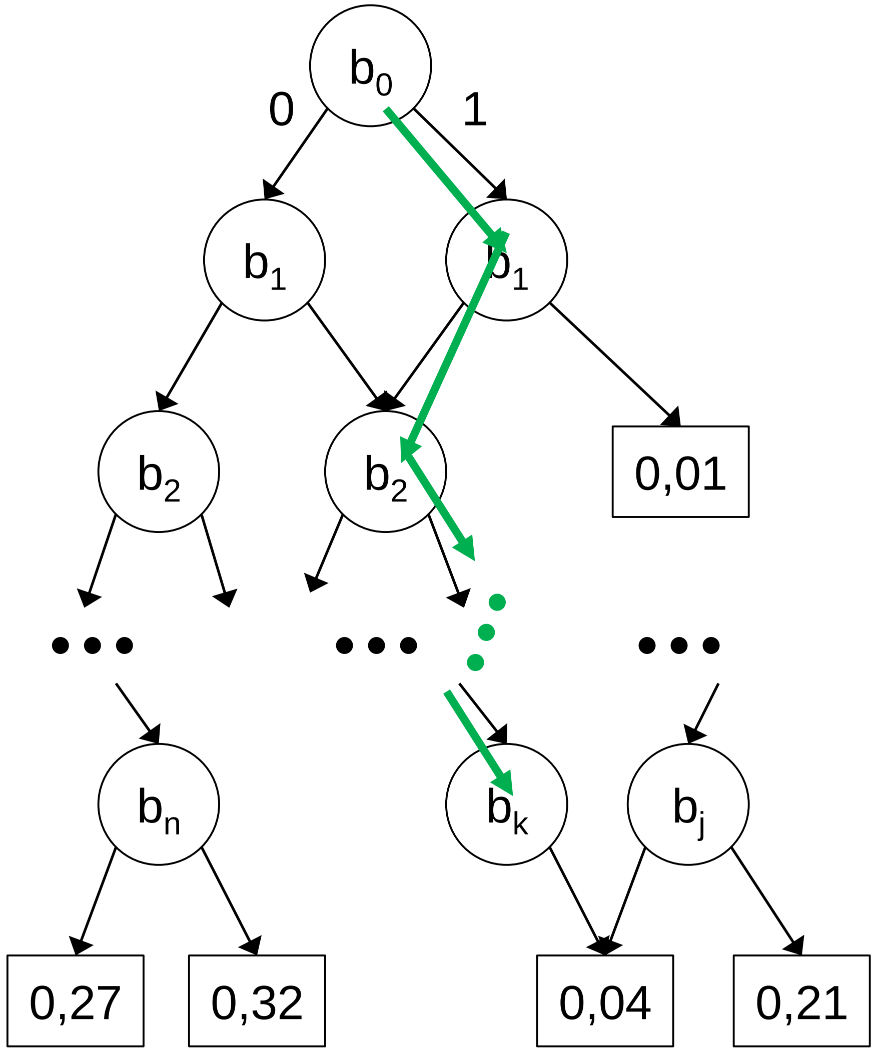
More of the “clever”



Evaluating ADDs

- Given an assignement for the b_i, an ADD is **easy to evaluate**:
  - Just follow the path from the route to the right leaf!
  - **If b_i is true go right else left**

Read off the precomputed result for the rule



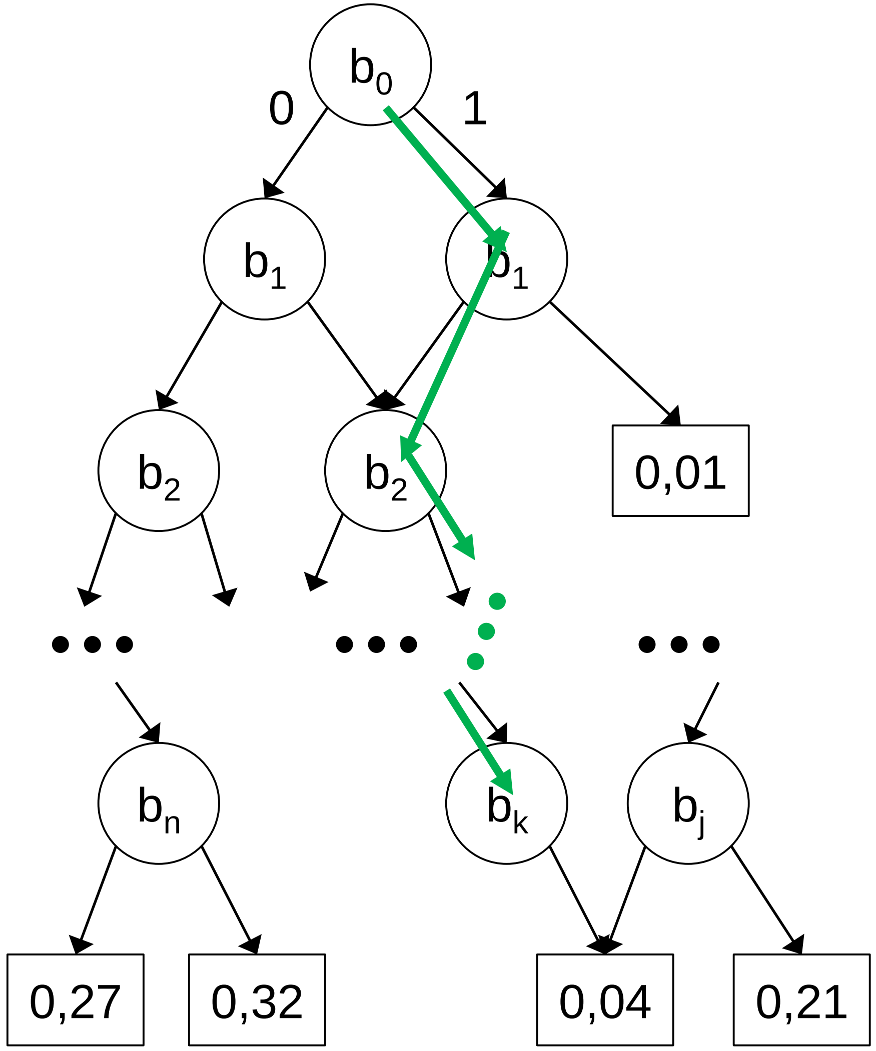
Evaluating ADDs and derive the PKB plausibility

- Given an assignement for the b_i, an ADD is **easy to evaluate**:
  - Just follow the path from the route to the right leaf!
  - **If b_i is true go right else left**

Read off the precomputed result for the rule

- Subsequently, compute the segmentation plausibility
  - Multiply with rule weight w_i
  - Sum over all rules ($\sum^{f}_{i=1}$)
  - Apply scaling

Combine everything



RO-ADDs for representing $n_i(I)$

Precomputation can handle only images with a **limited** number of segments

- Let $P = \{P_1, \dots , P_p\}$ be the predicates in the plausibility PKB

- Let $C = \{c_1, \dots, c_n\}$ be names for the segments in a segmentation (max n segments)

- Let $ground_c(P) = \{P_j(c_{i,1}, \dots, c_{i,k}) \mid c_{i,1},\dots, c_{i,k} \in C, P_j \in P\}$ denote the possible groundings of the predicates
  - These are all possible assertions about segments and their relations, in our case
  - An interpretation I provides a valuation for all these variables

- Then our RO-ADDs for the $n_i(I)$ will have variables in $ground_c(P)$
  - The RO-ADD for $n_i(I)$ depends on the groundings of the predicates appearing in the rule i
    - (at most – reducing might eliminate some variables)
  - Such RO-ADDs can be constructed by standard operations on RO-ADDs

Complexity considerations

- Consider
  - F(x,y) = Road(x) ∧ Sky(y) ∧ Below(x,y) → Consistent(x)

n + n^2 + groundings, i.e. $2*n + n^2$ variables for the ADD

440 variables for n=20

For some rules, the ADD might get too large (precomputation fails for large n)

Complexity considerations

- Consider
  - $F(x,y) = Road(x) \wedge Sky(y) \wedge Below(x,y) \rightarrow Consistent(x)$

$n \quad + \quad n^2 \quad +$ groundings, i.e. $2*n + n^2$ variables for the ADD

- To reduce the size of large ADDs, they may be split up further :
  - $F(x,y) \leftrightarrow x=c_1 \wedge F(c_1,y)$
    $\vee\ x=c_2 \wedge F(c_2,y)$
    …
    $\vee\ x=c_n \wedge F(c_n,y)$

Split up rules
(Alternative view: several
ADDs for one rule)

  - Since the $x=c_i \wedge F(c_i,y)$ are exclusive, the number of true groundings of F is sum of the number of the true groundings of the $x=c_i \wedge F(c_i,y)$.

Conclusion

- **Is the segmentation plausible?**
  - The rules in a probability knowledge base defines via rules a plausibility value, but it is computationally **expensive** to evaluate
  - **Approach:** Precomputation may save online computation time
    - Precomputation: Tables of (partial) results
    - Clever table representation by RO-ADDs
  - **Result**: Reduced online computation effort.

ADDs may help in providing a fast answer

Contact info

PD Dr. Hardi Hungar
German Aerospace Center (DLR)
Institute of Transportation Systems
Lilienthalplatz 7
38108 Brunswick
Germany

Hardi.Hungar@dlr.de