

## Meeting Information

The 75th meeting will be organised by Alberto Pardo.

## Dates

Monday 2017-02-20 to Friday 2017-02-24.

## Participants

Name	Arrival	Departure	Notes
Tom Schrijvers	2017-02-17 23:10 from Sao Paolo on 7632	2017-02-24 14:20 to Sao Paolo on 7631	
Lambert Meertens	2017-02-19 06:46 CM0368 from Panama	2017-02-25 01:26 CM0284 to Panama	
<a href="#">Patrik Jansson</a>	2017-02-15 11:10 from Sao Paolo on JJ8034	2017-02-24 14:20 to Sao Paolo on G37631	
Doaitse Swierstra	2017-02-16 from Madrid on IB6011 Iberia	TND, to Foz do Iguacu and Bolivia	
Jeremy Gibbons	2017-02-19 08:30 on IB6011 from Madrid	2017-02-26 14:35 on IB6012 to Madrid	
Bernhard MlxF6ller	2017-02-19 08:30 IB6011 from Madrid	2017-03-02 14:35 IB6012 to Madrid	
Jose Oliveira	2017-02-19 08:30 IB6011 from Madrid	2017-02-26 14:35 IB6012 to Madrid	
Zhenjiang Hu	2017-02-19 10:15 LH 7378 from Sao Paolo (... Tokyo)	2017-02-25 6:55 JJ8081 to Sao Paolo (... Tokyo)	
Mauro Jaskelioff	2017-02-19 07:30 from Buenos Aires	2017-02-24 18:30 to Rosario	obs
Walter Guttman	2017-02-10 11:45 AR2382 from Buenos Aires	2017-02-24 18:50 LA0905 to Santiago, then NZ	
Carroll Morgan	2017-02-17 00:30 LA0906 from Santiago	2017-02-26 14:20 LA0911 to Santiago	
Fritz Henglein	2017-02-18 09:00 UX045 from Madrid	2017-02-25 13:00 UA046 to Madrid	
Carlos CamarxE3o	2017-02-19 13:20 from Sao Paolo	2017-02-24 14:20 to Sao Paolo on G37631	local obs
Oleg Kiselyov	2017-02-19 12:20 LA902 from Santiago	2017-02-25 17:20 to GRU on JJ8037	obs
James <a href="#">McKinna</a>	2017-02-18 11:25 AF0394 from Paris	2017-02-25 14:45 AF0393 to Paris	obs
Exequiel Rivas	TBD	TBD	local obs
Miguel Pagano	2017-02-20 07:55 from Cordoba	2017-02-24 16:45 to Cordoba	local obs
Marcos Viera			obs
Juan Pablo Garcia Garland			local obs

## Organizational and administrative matters

Friday morning, February 24, 2017.

## Membership

These private matters are not recorded in the public version of the minutes.

## Formal resolution

The members and observers of WG2.1 present at the 75th meeting, in Montevideo, express their gratitude to Alberto Pardo for making it truly a MEAT-ing to remember, one where we were taken into the mysterious regions of "Otras Partes", whose true natures even now are yet to be revealed -- a topic for future research. And his intense energy and dedication spurred us all, but especially himself, to a fever pitch of activity from which we are still recovering. Aiding that recovery is of course that in our wonderful excursions we have seen for ourselves that Montevideo's position as the ["Best city in Latin America for Quality of Life"](#) is fully justified (and, most importantly, that it has been officially judged -well- ahead of Buenos Aires and Santiago de Chile 🤔 )

## Next meetings:

- [M76 \(Oct 16\9620, 2017\): Lesbos \(Meertens\)](#); the [Doodle](#) will be sent around very soon.
- M77 (Jul 18, 2018): NL/Germany? (Hinze).
- M78 (Mar 2019) China/Australia/NZ/Japan (?).

## Technical presentations in scheduled order

Carlos Camar\xE3o, [Haskell: Ambiguity and Multi-Parameter Type Classes](#) (Monday, February 20, 2017, 10:00):

---

We propose a simple definition to Haskell's ambiguity rule that allows ambiguity to have an intuitive, common sense meaning (existence of two or more instances of the same name used in an expression). This allows support for multi-parameter type classes with no additional concepts such as functional dependencies. It also allows overloading resolution to be defined and distinguished from ambiguity, type directed name resolution, and overloaded Ref instead of IOREf and STRef. Together with optional type classes, it supports overloaded record fields in a simple way. Together with instance modularization, it allows distinct instances of a type for a type class, if defined and used in distinct modules, overall support for modular name scoping and newtype wrapping and unwrapping to be avoided.

---

[Patrik Jansson](#), *Sequential Decision Problems and Avoidability using Dependent Types* (Monday, February 20, 2017, 11:05):

### Paper preprint:

[http://www.cse.chalmers.se/~patrikj/papers/CompTheoryPolicyAdviceAvoidability\\_JFP\\_2016-11\\_preprint.pdf](http://www.cse.chalmers.se/~patrikj/papers/CompTheoryPolicyAdviceAvoidability_JFP_2016-11_preprint.pdf)

The "slides" were written on flipcharts: [here are some photographs of what was on them \(and two earlier white-board presentations\)](#).

### Abstract:

The talk is based on a paper presenting a computer-checked generic implementation for solving finite-horizon sequential decision problems. This is a wide class of problems, including inter-temporal optimizations, knapsack, optimal bracketing, scheduling, etc. The implementation can handle time-step dependent control and state spaces, and monadic representations of uncertainty (such as stochastic, non-deterministic, fuzzy, or combinations thereof). This level of genericity is achievable in a programming language with dependent types (we have used both Idris and Agda). Dependent types are also the means that allow us to obtain a formalization and computer-checked proof of the central component of our implementation: Bellman's principle of optimality and the associated backwards induction algorithm. The formalization clarifies certain aspects of backwards induction and, by making explicit notions such as viability and reachability, can serve as a starting point for a theory of controllability of monadic dynamical systems, commonly encountered in, e.g., climate impact research.

---

Jeremy Gibbons, *Profunctor Optics: Modular Data Access* (Monday, February 20, 2017, 12:03):

---

Data accessors allow one to read and write components of a data structure; examples include lenses for accessing the fields of a record, prisms for accessing the variants of a union, and traversals for accessing the elements of a container. These data accessors are collectively known as optics; they are fundamental to programs that manipulate complex data. Individual data accessors for simple data structures are easy to write, for example as pairs of "getter" and "setter" methods. However, it is not obvious how to combine data accessors, in such a way that data accessors for a compound data structure are composed out of smaller data accessors for the parts of that structure. Generally, one has to write a sequence of statements or declarations that navigate step by step through the data structure, accessing one level at a time - which is to say, data accessors are traditionally not first-class citizens, combinable in their own right. We present a framework for modular data access, in which individual data accessors for simple data structures may be freely combined to obtain more complex data accessors for compound data structures. Data accessors become first-class citizens. The framework is based around the notion of profunctors, a flexible generalization of functions. (This is joint work with Matthew Pickering and Nicolas Wu. Here is a [draft of the slides](#).)

---

Walter Guttman, *An Algebraic Framework for Minimum Spanning Tree Problems* (Monday, February 20, 2017, 14:40):

---

It is well known that Warshall's algorithm for transitive closures and Floyd's algorithm for all-pairs shortest paths are instances of a general dynamic-programming algorithm based on semirings. Other instances solve the all-pairs widest path problem or convert finite automata to regular expressions. In this talk, I will show similar (not quite as spectacular) results about Prim's algorithm for minimum spanning trees. The different instances do not vary in the algorithm, but in the specification: the same algorithm solves a number of different problems. This continues work I presented at the G  teborg meeting, and I will also show a generalisation of relation algebras to model weighted graphs that came out of it. (Here are the [slides](#) of my talk.)

---

Oleg Kiselyov, *Having an Effect* (Monday, February 20, 2017, 15:59):

(Side-)Effects are the battle cry in what sometimes seems like a war between pure functional and imperative programming. Functional programmers charge brandishing a *monad* -- a powerful weapon, judging by the exponentially proliferating monad tutorials and its appearances (however fleeting at times) in almost any modern language, natural language included. Monads indeed have many uses and benefits. But this is not the whole story. The talk will try to tell a bigger story, following the tantalizing lead: all the founding papers on monads, free monads, monad transformers, handlers, algebraic effects, etc. -- all of them, down to the title, are about extensible interpreters. What we eventually find is the *interaction* -- between a client and a server, an interpreter and the interpreted code, an expression and its context. The distinction between functional and imperative is hence contingent, an artifact of focusing on a small part of the interacting system. Revisiting the origins of the field and recovering the insights and forgotten alternatives not only help with the wiser attitude towards monads. We see how they help make our programs have the intended effect.

---

Doaitse Swierstra, *Describing the ST Monad using Polymorphic Contexts; Type Rules for PC's*. (Monday, February 20, 2017, 17:10):

---

We show how by an extension of the HM-typing discipline we can define the semantics of the ST-monad, so ST does not have to be built in.

In Gothenburg I presented a simpler motivation for having Polymorphic Contexts (see also paper). In Glasgow I asked for help in formulating the necessary type rules. The are given in the a [paper](#). The thing I want to discuss now is how to prove these rules correct. It seems that the normal approach using conventional type judgements does not work, since types are only partially known.

---

- [Slides of Doaitse's talk](#)

Fritz Henglein, *Bird and Meertens in disguise* (Tuesday, February 21, 2017, 9:03):

---

Futhark is a relatively recent purely functional array programming language based on second-order array combinators with constant-time array update. It generates high-performance GPU-code competitive with and, in some cases, superior to handwritten [OpenCL](#) or CUDA code, without autotuning (yet). Amongst its tools for achieving practically high performance is aggressive application of algebraic equalities for transformations including fusion, fission (vectorization) and streaming chunking. The transformation rules are heavily pointed, but their pointless rendition reveals Bird and Meertens, lurking. This is speculative work ; it is based on work on Futhark by Troels Henriksen and Cosmin Oancea with contributions by Martin Elsman and myself, which is to appear at PLDI '17. See the [slides](#).

---

Mauro Jaskelioff, *Algebraic Effects and Scoped Operations* (Tuesday, February 21, 2017, 9:55):

---

Algebraic effects have many nice properties, but introduce some limitations when dealing with operations with scope. The root of the problem is that operations with scope are treated as semantics, whereas ordinary algebraic operations are syntax. In this talk, I'll discuss an extension of the algebraic approach that allows us to treat scoped operations as syntax, and analyse some of the consequences of doing so. This is joint work with Maciej Pirł, Tom Schrijvers, and Nicolas Wu. (Here are the [slides](#).)

---

James [McKinna](#), *On ringads and foldables* (Tuesday, February 21, 2017, 11:01):

Foldable  $f$  is a class defined in the haskell library, with single method `foldMap :: Monoid m => (a -> m) -> f a -> m` which is analogous to an Eindhoven Quantifier (a reduce over a collection, for a given Monoid structure measuring individuals in the collection)

Ringad is a class introduced by Jeremy after old work of Phil Wadler's, as class `(MonadZero f, MonadPlus f) => Ringad f` which supports the construction of empty, aggregate collections, and hence a suitable target in which to interpret comprehension notation.

Theorem:

For a Functor  $f$ ,  $f$  is Foldable iff forall  $a$ . Monoid  $a \Rightarrow$  Algebra  $f\ a$

For a Monad  $f$ ,  $f$  is a Ringad iff forall  $a$ . Algebra  $f\ a \Rightarrow$  Monoid  $a$

This talk attempts to motivate, explain, and justify these characterisations, and thereby (definitionally) eliminate the Foldable and Ringad classes in favour of the simpler ones of Monoid and Algebra, together with the idea of higher-order class constraint.

Work in progress: there are still some wrinkles and some loose ends.

---

Zhenjiang Hu, *Synchronizing Concrete and Abstract Syntaxes using BiGUL* (Tuesday, February 21, 2017, 11:45):

---

In this talk, I will show that BiGUL, a Haskell library for bidirectional programming, is useful for systematic development of synchronizers between concrete and abstract syntaxes. Concrete syntax is designed to be programmer-friendly and provides convenient syntactic sugar, while abstract syntax is concise, structured, and easily manipulable by the compiler back-end. Synchronizing these two syntaxes would enable us to gain both advantages, for example, for reporting the result of compiler optimization done on abstract syntax to the programmer, who knows only friendly concrete syntax. See the slides ( [hu.pdf](#) ).

---

Jose Oliveira, *Measuring probabilistic contracts* (Tuesday, February 21, 2017, 14:01):

---

In the [TRUST](#) project we are trying to extend Alloy to make it a more comprehensive modelling tool for trustworthy software design. In particular, we wish to be able to express faulty behaviour in software models, in a probabilistic way. Given the relational basis of Alloy, we thought of extending its underlying Boolean matrices to stochastic matrices and of adapting the notion of contract validity accordingly. This talk will present a linear algebraic approach to "measuring" the validity of contracts. In spite of its "algebraicity", there are difficulties in the approach that I would like to discuss. I believe this problem can be traced back to earlier work by Carroll on probabilistic Hoare triples. ( [slides](#) )

---

Tom Schrijvers, *Efficient Compilation of Algebraic Effects and Handlers* (Tuesday, February 21, 2017, 14:55):

The popularity of algebraic effect handlers as a programming language feature for user-defined computational effects, is steadily growing. Yet, even though efficient runtime representations have already been studied, most handler-based programs are still much slower than hand-written code. In this paper we show that the performance gap can be drastically narrowed (in some cases even closed) by means of type- and effect-directed optimising compilation. Our approach consists of two stages. Firstly, we combine elementary source-to-source transformations with judicious function specialisation in order to aggressively reduce handler applications. Secondly, we show how to elaborate the source language into a handler-less target language in a way that incurs no overhead for pure computations. This work comes with a practical implementation: an optimizing compiler from Effy, a small functional language with algebraic effects and handlers, to OCaml. Experimental evaluation with this implementation demonstrates that in a number of benchmarks our approach eliminates much of the overhead of handlers and yields competitive performance with hand-written OCaml code. This is joint work with Matija Pretnar, Amr Hany Saleh Shehata and Axel Faes.

---

Bernhard Möller, *Algebraic approaches to rely/guarantee (RG)* (Tuesday, February 21, 2017, 16:00):

---

This is ongoing work together with Tony Hoare, Martin E. Möller and José Oliveira. RG, initiated by Cliff Jones, is an important paradigm in verifying concurrent programs: a program relies on well-behaviour of its environment and in turn guarantees well-behaviour of its own. There exist already several algebraic formalisations of the RG. Most of them, however, deal only with the case that the guarantees are "absolute", unconnected to the rely part. We are looking whether a recent streamlined approach based on residuals (or factors, as Roland Backhouse would call them) can be carried over to the general case. Here are the [slides](#).

---

Carroll Morgan, *A new rule for almost-certain termination of probabilistic- and demonic programs.* (Wednesday, February 22, 2017, 9:04):

---

In the last year or so there's been a sudden resurgence of interest in proof rules for almost-certain termination of probabilistic loops. This "arms race" consists of making one's rule as general as possible, while preserving soundness of course, and then "reaching a termination that others cannot reach" by treating a well known example that others don't handle. Some of the players in this competition are quite famous.

Annabelle McIver and I have competed in this game for some time but, spurred by the above activity, we re-entered the fray late last year with our own, new rule that goes a little bit further than any of the others we know about. And we have found an intriguing open problem --- appealing to a completeness theorem from the 1950's shows that we can prove termination of the **two**-dimensional random walk in program logic... if only we could find the variant that the completeness result says must exist. But we can't find that variant.

**Can you?**

There's an arXiv of this work at <https://arxiv.org/abs/1612.01091>. But my presentation (if I get to make it) will be on a whiteboard or equivalent (eg flip chart).

---

Tom Schrijvers, *ProbLog is Applicative* (Wednesday, February 22, 2017, 11:00):

---

Probabilistic Programming Languages (PPLs) support constructions to natively express probability distributions, making it easier for researchers to develop, share and reuse probabilistic models. They have a long history in both the functional (e.g., Anglican) and logic programming (e.g., ProbLog) paradigms. Unfortunately, these efforts have been conducted mostly in isolation and little is known about the relative merits of the two approaches, creating much confusion for the un-initiated. In this work we establish a common ground for both approaches in terms of algebraic models of probabilistic computation. It is already well-known that functional PPLs conform to the monadic model. We show that ProbLog's flavour of probabilistic computation is restricted to the applicative functor interface. This means that functional PPLs afford greater expressiveness in terms of dynamic program structure, while ProbLog programs are inherently more amenable to static analysis and thus afford faster inference. This is joint work with Alexander Vandenbroucke.

---

Jose Oliveira, *First steps in a (linear) algebra of (quantum) functional programming* (Wednesday, February 22, 2017, 11:43):

---

A few months ago my lab started a partnership ( [QuantaLab](#)) with the nearby Iberian Nanotechnology Laboratory (INL) whose research programme includes quantum computing. Thus introduced to this interesting field, I am trying to frame quantum functional programming in previous work on typed linear algebra of programming (LAoP). While the latter shifted from functions / relations (standard in AoP) to stochastic matrices, the quantum setting calls for unitary (complex) matrices and their rich algebra. I will talk about my first attempts to (polymorphically) type such (recursive) matrices and to introduce more structure in the definition of quantum gates, usually described pointwise (at qubit-level) in the literature. In particular, reversibility can be described via techniques that we already know from elsewhere (constant-complement functions, bidirectionalization etc). ( [slides](#))

---

Exequiel Rivas, [Dioids and functional programming](#) (Thursday, February 23, 2017, 9:04):

---

In previous work, the [MonadPlus](#) /Alternative type classes were presented as generalized near-semirings. In this talk, a new algebraic structure is introduced: dioids. This new structure provides an account for the [MonadPlus](#) /Alternative type classes subject to the left-catch axiom. We construct the free dioid, and show some partial results regarding the free [MonadPlus](#) /Alternative. This is joint work with Ezequiel Postan and Mauro Jaskelioff.

---

Marcos Viera, [Safe and Lightweight Attribute Grammars for Haskell](#) (Thursday, February 23, 2017, 9:37):

---

I presented a library that defines an EDSL for first-class attribute grammars that ensures a clean public interface for the user with simple types and simple error messages that do not leak implementation details.

---

Oleg Kiselyov, *Sound and efficient language-integrated query: the denotation of ORDER BY* (Thursday, February 23, 2017, 11:00):



Language-integrated query systems like T-LINQ make relational operations on (generally external) data feel like the ordinary iteration over native arrays. As ordinary programs, queries are type-checked, can be abstracted over and composed. To access relational database systems, queries are eventually translated into well-formed, well-typed and efficient SQL. However, most existing language-integrated query systems implement only a small subset of relational operations supported by modern databases. We introduce a full-featured language Quel that supports not only the standard selections, projections, unions and joins but also the operations corresponding to SQL's ORDER BY and LIMIT. We present the denotational semantics of the language, which is, unlike SQL's, is compositional. Quel's type system not only ensures by construction the intricate SQL validity constraints. It also prevents the accidental composition of hard-to-optimize queries. The semantics leads to the normalization-by-evaluation process to produce the efficient SQL code. The denotational treatment goes hand in hand with our OCaml embedding of Quel in the tagless-final style.

Zhenjiang Hu, *Programming Derivatives -- Towards Change-Oriented Programming* (Thursday, February 23, 2017, 11:37):

Change is essential and expensive in software development. Although a significant amount of work has been done on software change, its status as a scientific discipline is still an open challenge. In this talk, I show that it is possible to treat software change as a first-class citizen, where we can directly program changes through coding of derivatives.

Bernhard M\xF6ller, *Functional modelling of geoinformation systems (GIS)* (Thursday, February 23, 2017, 14:20):

I have reported on this in earlier meetings. By now our Haskell-based course on the subject, developed together with Sabine Timpf from the Augsburg Geography Department, has become respectable and even relatively popular. I would show some main ideas, some project work by students and topics from an ongoing master's thesis dealing with temporal aspects (moving bodies in a GIS). Here are the [slides](#).

## Location

### [TRYP Montevideo Hotel](#)

Dr. H\xE9ctor Miranda 2361  
Punta Carretas  
11300 Montevideo

TRYP Hotel is situated in Punta Carretas neighbourhood, next to Montevideo Golf Club.

## Schedule (subject to change)

The meeting will take place at **Tryp Hotel**.

The **meeting room** is situated at the ground floor (take the corridor on the left hand side of the hotel reception).

On **Sunday**, at 8pm we will meet at restaurant **El Berret\xEDn** at the corner of the streets Jos\xE9 Mar\xEDA Montero and Guip\xFAzcoa. Here is a [map](#) that shows you the place. It is not far from Tryp Hotel. Say that you are coming to the tables reserved by Alberto Pardo.

**Lunches** from Monday to Thursday will be at Tryp hotel's restaurant.

Sessions will start at 9:00 on Monday and finish at noon on Friday.

Monday, Tuesday, and Thursday, the schedule is as follows:

- session - 9:00 to 10:30
- coffee break - 10:30 to 11:00
- session - 11:00 to 12:30
- lunch - 12:30 to 14:00
- session - 14:00 to 15:30
- coffee break at 15:30 to 16:00
- session - 16:00 to 18:00
- dinner

Wednesday and Friday we will only have sessions in the morning:

- session - 9:00 to 10:30
- coffee break - 10:30 to 11:00
- session - 11:00 to 12:30
- lunch - 12:30 to 14:00

## Cost and Payment

The *registration fee* has to be payed in cash.

It will consist of: U\$S 208 (US Dollars) + an amount for dinner at El Berret (for those who go to the restaurant).

The fee includes the cost of the meeting room, coffeee breaks, lunches (from Monday to Thursday) and Sunday dinner.

My plan is to pay the global account at El Berret, then divide it by the number of persons that go to and charge them that value as part of the fee. The fact is that at night it is not possible to negotiate a menu with restaurants (like for lunch), but prices at El Berret are quite reasonable.

## Banquet

The banquet will take place on Thursday evening at [VñF1a Varela Zarranz](#), a winery situated just outside Montevideo. We will first have a guided visit to the winery and then the dinner. We have transport to get there. The cost of the banquet is U\$S 55 (US Dollars) per person.

## Excursion

The excursion will be on Wednesday afternoon. The plan is to go to Punta del Este and surroundings. Exact details will be given during the meeting. We depend a lot on the weather to choose the activities to do.

## Accommodation

All room rates are final prices (exempt from taxes). In Uruguay hotel rates are usually given in US Dollars (U\$S). In addition to the meeting's hotel, we list some other alternatives in the near of TRYP Hotel. The list does not pretend to be complete.

[TRYP Hotel](#) has offered us the following special rates (per room, per night):

Single room - U\$S 85,-

Double room - U\$S 95,-

This includes: buffet breakfast, free Wi-Fi, fitness center, swimming pool and sauna.

To book a room at TRYP Hotel write an email to [meliares@confort.com.uy](mailto:meliares@confort.com.uy) mentioning that you come to "IFIP 2.1 event" in order to get the special offer.

#### [Vivaldi Hotel](#)

Blanca del Tabar\xE9 2903

#### [Intercity Hotel](#)

Ibiray 2398

#### [Dazzler Hotel](#)

21 de Setiembre 2752 [Regency Golf](#)

Solano Garcia 2473

#### [BIT Design Hotel](#)

Ram\xF3n Fernandez 265

## Visa

If any of you requires a visa to travel to Uruguay, please tell me and I send you an invitation letter. EU, US and japanese citizens do not need a visa to come, but on the other hand, e.g. chinese citizens do need. My experience (from previous events I organized) is that the visa procedure takes approx. one month. The visa procedure must be started at an uruguayan consulate. I do not know if they require to do it personally or if it is possible by post. The following are the requirements for a visa (taken from a page of our Ministry of Foreign Affairs):

- to fill a form
- invitation letter or hotel reservation
- Passport (expiration date should be 6 months after arrival to Uruguay)
- one photograph (passport size)
- flight reservation
- Cost: US\$ 42.- (US Dollars).

## How to get there

### **By plane.**

[Montevideo's airport](#) ("Aeropuerto Internacional de Carrasco" or simply Carrasco) has direct flights to:

- Madrid (Iberia and Air Europa)
- Paris (Air France)
- Miami (American Airlines)
- Panama City (Copa Airlines)
- São Paulo (Guarulhos airport) (LATAM and GOL)
- Rio de Janeiro (LATAM)
- Santiago de Chile (LATAM and SKY Airline)
- Buenos Aires (Aerolineas Argentinas and Amazonas)
- Lima (Avianca)

If you come via Buenos Aires be careful that the flights from Buenos Aires to Montevideo are actually from *Aeroparque* (the domestic airport) and not from the international one (*Ezeiza*). Air France is the unique airline that connects *Ezeiza* with *Carrasco* simply because the flight from Paris stops both in *Carrasco* and *Ezeiza*. Except for a few cases, the destination of international flights to Buenos Aires is always *Ezeiza*. [Tienda Leon](#) is the company that performs the transfer service (by coach or remise) between *Ezeiza* and *Aeroparque*.

**By ferry.** Buenos Aires and Montevideo are also connected by ferry. [Buquebus](#) is the company that performs the service. There are two alternatives:

- Direct ferry from Buenos Aires to Montevideo.
- Ferry from Buenos Aires to Colonia (a city in Uruguay) and then bus from Colonia to Montevideo. The combined ticket (ferry + bus) is sold altogether. In case you fly to Buenos Aires and decide to come to Montevideo with ferry, then you have to go from *Ezeiza* to Buquebus' terminal in Buenos Aires' port. The easiest way to do so is to take a remise (again from Tienda Leon).

### **How to get TRYP Hotel:**

- [Airport Taxi](#). \$U1330 (uruguayan pesos), approx U\$S 47, the journey from the airport to the hotel.
- [Airport Transfer](#). \$U 400 uruguayan pesos p/person, approx. U\$S 14, the mini-bus departs only when there are at least 5 persons to take.
- *Bus DM1 from the airport*. Take DM1, direction "Punta Carretas", at the bus stop that is close to the airport's entrance (arrivals sector). Continue till the last stop (Punta Carretas Shopping Center). Walk back 50mts till the cross in front of the shopping's entrance, turn left and walk the street (it is Dr. Hector Miranda). The hotel is at approx. 100mts.
- *From the port*, if you come from Buenos Aires with the direct ferry: Take a regular taxi; the fare will cost around \$U 220 (uruguayan pesos), approx. U\$S 7,5.
- *From the bus station* if you come from Buenos Aires with Buquebus via Colonia, or simply because you come by bus from Argentina or Brazil
  - Take a regular taxi; the fare will cost around \$U 190 (uruguayan pesos), approx. U\$S 6.
  - By bus: Take bus nr. 174 at the bus stop that is at the front of the bus terminal. Fare costs \$U 30, 1 U\$S.
- *Remise from the airport*: remises are cheaper than Airport Taxi. I can phone the remises so that they wait for you at the airport. Fare costs:
  - U\$1050 (pesos), approx. U\$S 38, from the airport to the hotel
  - U\$ 850 (pesos), approx. U\$S 30, from the hotel to the airport

## Restaurant recommendations

Alberto recommends the following:

- La Pulperia (Lagunillas and Colonel Mora)
- Bar Tabare (Tabare and JLZ de San Martin)
- La Perdiz (Guipuzcoa and Dr. Bolivar Balinas), behind the shopping. I never stayed there but I have some colleagues that recommended it to me. I looked in [TripAdvisor](#) and in general the people liked it.
- Mandolina (Guipuzcoa and Joaquin Nuñez), I went a pair of times for lunch and it is ok.
- La Cantina del Puertito (Jose Luis Zorrilla de San Martin between Parva Domus and Tabare), never stayed there.
- La Parrilla de Leo (Claudio Williman and Agr. Francisco J. Ros), I went many times there, it is ok.
- Bar Lobo (Montero and Coronel Mora)

If you want to eat something faster, then you should try a "chivito". "Chivo" in spanish means young goat, but a chivito has no goat meat. It is a big sandwich with a thin slice of cow meat and many other things (tomato, mayonese, eggs, sweet pepper, etc, etc). One cannot leave Uruguay without having tasted a chivito. The traditional one is called Canadiense (canadian), although it has nothing to do with Canada (it is matter of name). Fritz told me he ate a chivito and he liked a lot. Many restaurants prepare chivitos, in particular:

- Chivitos Tiqui Taca (Coronel Mora and Gral. Gregorio Suarez), there are good.
- La Pasiva (at the basement the shopping center)
- Another shop at the top floor of the shopping center

-- [JeremyGibbons](#) - 24 Feb 2017

#### **Attachments** 12

This topic: IFIP21 > FutureMeetings > Uruguay

Topic revision: